Command-line clients for Red Hat OpenStack Platform

OpenStack Team
rhos-docs@redhat.com
Legal Notice

Copyright © 2022 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at http://creativecommons.org/licenses/by-sa/3.0/. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation’s permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

A reference to the commands available to the unified OpenStack command-line client.
# Table of Contents

**MAKING OPEN SOURCE MORE INCLUSIVE** ................................................................. 27  
**PROVIDING FEEDBACK ON RED HAT DOCUMENTATION** ........................................... 28  
**CHAPTER 1. THE OPENSTACK CLIENT** .................................................................. 29  
**CHAPTER 2. ACCESS** ......................................................................................... 41  
  2.1. ACCESS RULE DELETE  ............................................................................. 41  
  2.2. ACCESS RULE LIST  ............................................................................... 41  
  2.3. ACCESS RULE SHOW  ........................................................................... 43  
  2.4. ACCESS TOKEN CREATE  ......................................................................... 44  
**CHAPTER 3. ACL** ............................................................................................. 46  
  3.1. ACL DELETE  ............................................................................................ 46  
  3.2. ACL GET  .................................................................................................. 46  
  3.3. ACL SUBMIT  ............................................................................................ 47  
  3.4. ACL USER ADD  ...................................................................................... 49  
  3.5. ACL USER REMOVE  ............................................................................. 51  
**CHAPTER 4. ACTION** ....................................................................................... 54  
  4.1. ACTION DEFINITION CREATE  ............................................................ 54  
  4.2. ACTION DEFINITION DEFINITION SHOW  ........................................... 56  
  4.3. ACTION DEFINITION DELETE  ............................................................... 56  
  4.4. ACTION DEFINITION LIST  .................................................................... 57  
  4.5. ACTION DEFINITION SHOW  .................................................................. 59  
  4.6. ACTION DEFINITION UPDATE  ............................................................... 60  
  4.7. ACTION EXECUTION DELETE  ................................................................. 62  
  4.8. ACTION EXECUTION INPUT SHOW  ....................................................... 63  
  4.9. ACTION EXECUTION LIST  ...................................................................... 63  
  4.10. ACTION EXECUTION OUTPUT SHOW  .................................................. 65  
  4.11. ACTION EXECUTION RUN  ...................................................................... 66  
  4.12. ACTION EXECUTION SHOW  .................................................................. 67  
  4.13. ACTION EXECUTION UPDATE  .............................................................. 69  
**CHAPTER 5. ADDRESS** ....................................................................................... 71  
  5.1. ADDRESS GROUP CREATE  ...................................................................... 71  
  5.2. ADDRESS GROUP DELETE  ...................................................................... 72  
  5.3. ADDRESS GROUP LIST  .......................................................................... 73  
  5.4. ADDRESS GROUP SET  .......................................................................... 74  
  5.5. ADDRESS GROUP SHOW  ........................................................................ 75  
  5.6. ADDRESS GROUP UNSET  ...................................................................... 76  
  5.7. ADDRESS SCOPE CREATE  ...................................................................... 77  
  5.8. ADDRESS SCOPE DELETE  ...................................................................... 79  
  5.9. ADDRESS SCOPE LIST  .......................................................................... 79  
  5.10. ADDRESS SCOPE SET  .......................................................................... 81  
  5.11. ADDRESS SCOPE SHOW  ...................................................................... 81  
**CHAPTER 6. AGGREGATE** .................................................................................. 84  
  6.1. AGGREGATE ADD HOST  .......................................................................... 84  
  6.2. AGGREGATE CACHE IMAGE  .................................................................... 85  
  6.3. AGGREGATE CREATE  ............................................................................. 85  
  6.4. AGGREGATE DELETE  .............................................................................. 87  
  6.5. AGGREGATE LIST  ................................................................................... 87
6.6. AGGREGATE REMOVE HOST .......................... 89
6.7. AGGREGATE SET .................................. 90
6.8. AGGREGATE SHOW ................................. 91
6.9. AGGREGATE UNSET ................................. 92

CHAPTER 7. ALARM ........................................ 94
7.1. ALARM CREATE ..................................... 94
7.2. ALARM DELETE ..................................... 98
7.3. ALARM-HISTORY SEARCH ...................... 98
7.4. ALARM-HISTORY SHOW ......................... 100
7.5. ALARM LIST ......................................... 101
7.6. ALARM QUOTA SET ................................. 103
7.7. ALARM QUOTA SHOW .............................. 104
7.8. ALARM SHOW ....................................... 106
7.9. ALARM STATE GET .................................. 107
7.10. ALARM STATE SET ................................ 108
7.11. ALARM UPDATE ..................................... 110

CHAPTER 8. ALARMING .................................... 115
8.1. ALARMING CAPABILITIES LIST ................. 115

CHAPTER 9. APPLICATION ............................... 117
9.1. APPLICATION CREDENTIAL CREATE .......... 117
9.2. APPLICATION CREDENTIAL DELETE .......... 119
9.3. APPLICATION CREDENTIAL LIST ............... 119
9.4. APPLICATION CREDENTIAL SHOW .............. 121

CHAPTER 10. AVAILABILITY ............................. 123
10.1. AVAILABILITY ZONE LIST ....................... 123

CHAPTER 11. BAREMETAL ............................... 125
11.1. BAREMETAL ALLOCATION CREATE .......... 125
11.2. BAREMETAL ALLOCATION DELETE .......... 127
11.3. BAREMETAL ALLOCATION LIST ............... 127
11.4. BAREMETAL ALLOCATION SET ............... 129
11.5. BAREMETAL ALLOCATION SHOW ............. 130
11.6. BAREMETAL ALLOCATION UNSET .......... 131
11.7. BAREMETAL CHASSIS CREATE ................. 132
11.8. BAREMETAL CHASSIS DELETE ................. 133
11.9. BAREMETAL CHASSIS LIST ..................... 133
11.10. BAREMETAL CHASSIS SET ...................... 135
11.11. BAREMETAL CHASSIS SHOW .................. 136
11.12. BAREMETAL CHASSIS UNSET ................. 137
11.13. BAREMETAL CONDUCTOR LIST ............... 138
11.14. BAREMETAL CONDUCTOR SHOW ............. 140
11.15. BAREMETAL CREATE ............................. 141
11.16. BAREMETAL DEPLOY TEMPLATE CREATE .... 142
11.17. BAREMETAL DEPLOY TEMPLATE DELETE .... 143
11.18. BAREMETAL DEPLOY TEMPLATE LIST ....... 144
11.19. BAREMETAL DEPLOY TEMPLATE SET ....... 146
11.20. BAREMETAL DEPLOY TEMPLATE SHOW ...... 146
11.21. BAREMETAL DEPLOY TEMPLATE UNSET .... 148
11.22. BAREMETAL DRIVER LIST ...................... 148
11.23. BAREMETAL DRIVER PASSTHRU CALL ....... 150
11.24. BAREMETAL DRIVER PASSTHRU LIST 151
11.25. BAREMETAL DRIVER PROPERTY LIST 153
11.26. BAREMETAL DRIVER RAID PROPERTY LIST 154
11.27. BAREMETAL DRIVER SHOW 156
11.28. BAREMETAL INTROSPECTION ABORT 157
11.29. BAREMETAL INTROSPECTION DATA SAVE 158
11.30. BAREMETAL INTROSPECTION INTERFACE LIST 158
11.31. BAREMETAL INTROSPECTION INTERFACE SHOW 160
11.32. BAREMETAL INTROSPECTION LIST 161
11.33. BAREMETAL INTROSPECTION REPROCESS 163
11.34. BAREMETAL INTROSPECTION RULE DELETE 163
11.35. BAREMETAL INTROSPECTION RULE IMPORT 164
11.36. BAREMETAL INTROSPECTION RULE LIST 165
11.37. BAREMETAL INTROSPECTION RULE PURGE 167
11.38. BAREMETAL INTROSPECTION RULE SHOW 167
11.39. BAREMETAL INTROSPECTION START 168
11.40. BAREMETAL INTROSPECTION STATUS 170
11.41. BAREMETAL NODE ABORT 171
11.42. BAREMETAL NODE ADD TRAIT 172
11.43. BAREMETAL NODE ADOPT 172
11.44. BAREMETAL NODE BIOS SETTING LIST 173
11.45. BAREMETAL NODE BIOS SETTING SHOW 174
11.46. BAREMETAL NODE BOOT DEVICE SET 176
11.47. BAREMETAL NODE BOOT DEVICE SHOW 176
11.48. BAREMETAL NODE CLEAN 178
11.49. BAREMETAL NODE CONSOLE DISABLE 178
11.50. BAREMETAL NODE CONSOLE ENABLE 179
11.51. BAREMETAL NODE CONSOLE SHOW 179
11.52. BAREMETAL NODE CREATE 181
11.53. BAREMETAL NODE DELETE 184
11.54. BAREMETAL NODE DEPLOY 185
11.55. BAREMETAL NODE INJECT NMI 186
11.56. BAREMETAL NODE INSPECT 186
11.57. BAREMETAL NODE LIST 187
11.58. BAREMETAL NODE MAINTENANCE SET 189
11.59. BAREMETAL NODE MAINTENANCE UNSET 190
11.60. BAREMETAL NODE MANAGE 190
11.61. BAREMETAL NODE PASSTHRU CALL 191
11.62. BAREMETAL NODE PASSTHRU LIST 191
11.63. BAREMETAL NODE POWER OFF 193
11.64. BAREMETAL NODE POWER ON 194
11.65. BAREMETAL NODE PROVIDE 194
11.66. BAREMETAL NODE REBOOT 195
11.67. BAREMETAL NODE REBUILD 195
11.68. BAREMETAL NODE REMOVE TRAIT 196
11.69. BAREMETAL NODE RESCUE 197
11.70. BAREMETAL NODE SET 197
11.71. BAREMETAL NODE SHOW 201
11.72. BAREMETAL NODE TRAIT LIST 202
11.73. BAREMETAL NODE UNDEPLOY 204
11.74. BAREMETAL NODE UNRESCUE 204
11.75. BAREMETAL NODE UNSET 205
11.76. BAREMETAL NODE VALIDATE 207
11.77. BAREMETAL NODE VIF ATTACH
11.78. BAREMETAL NODE VIF DETACH
11.79. BAREMETAL NODE VIF LIST
11.80. BAREMETAL PORT CREATE
11.81. BAREMETAL PORT DELETE
11.82. BAREMETAL PORT GROUP CREATE
11.83. BAREMETAL PORT GROUP DELETE
11.84. BAREMETAL PORT GROUP LIST
11.85. BAREMETAL PORT GROUP SET
11.86. BAREMETAL PORT GROUP SHOW
11.87. BAREMETAL PORT GROUP UNSET
11.88. BAREMETAL PORT LIST
11.89. BAREMETAL PORT SET
11.90. BAREMETAL PORT SHOW
11.91. BAREMETAL PORT UNSET
11.92. BAREMETAL VOLUME CONNECTOR CREATE
11.93. BAREMETAL VOLUME CONNECTOR DELETE
11.94. BAREMETAL VOLUME CONNECTOR LIST
11.95. BAREMETAL VOLUME CONNECTOR SET
11.96. BAREMETAL VOLUME CONNECTOR SHOW
11.97. BAREMETAL VOLUME CONNECTOR UNSET
11.98. BAREMETAL VOLUME TARGET CREATE
11.99. BAREMETAL VOLUME TARGET DELETE
11.100. BAREMETAL VOLUME TARGET LIST
11.101. BAREMETAL VOLUME TARGET SET
11.102. BAREMETAL VOLUME TARGET SHOW
11.103. BAREMETAL VOLUME TARGET UNSET

CHAPTER 12. CA
12.1. CA GET
12.2. CA LIST

CHAPTER 13. CATALOG
13.1. CATALOG LIST
13.2. CATALOG SHOW

CHAPTER 14. CODE
14.1. CODE SOURCE CONTENT SHOW
14.2. CODE SOURCE CREATE
14.3. CODE SOURCE DELETE
14.4. CODE SOURCE LIST
14.5. CODE SOURCE SHOW
14.6. CODE SOURCE UPDATE

CHAPTER 15. COE
15.1. COE CA ROTATE
15.2. COE CA SHOW
15.3. COE CA SIGN
15.4. COE CLUSTER CONFIG
15.5. COE CLUSTER CREATE
15.6. COE CLUSTER DELETE
15.7. COE CLUSTER LIST
15.8. COE CLUSTER RESIZE
15.9. COE CLUSTER SHOW
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.10. COE CLUSTER TEMPLATE CREATE</td>
<td>261</td>
</tr>
<tr>
<td>15.11. COE CLUSTER TEMPLATE DELETE</td>
<td>265</td>
</tr>
<tr>
<td>15.12. COE CLUSTER TEMPLATE LIST</td>
<td>265</td>
</tr>
<tr>
<td>15.13. COE CLUSTER TEMPLATE SHOW</td>
<td>267</td>
</tr>
<tr>
<td>15.14. COE CLUSTER TEMPLATE UPDATE</td>
<td>268</td>
</tr>
<tr>
<td>15.15. COE CLUSTER UPDATE</td>
<td>270</td>
</tr>
<tr>
<td>15.16. COE CLUSTER UPGRADE</td>
<td>270</td>
</tr>
<tr>
<td>15.17. COE NODEGROUP CREATE</td>
<td>271</td>
</tr>
<tr>
<td>15.18. COE NODEGROUP DELETE</td>
<td>272</td>
</tr>
<tr>
<td>15.19. COE NODEGROUP LIST</td>
<td>273</td>
</tr>
<tr>
<td>15.20. COE NODEGROUP SHOW</td>
<td>274</td>
</tr>
<tr>
<td>15.21. COE NODEGROUP UPDATE</td>
<td>276</td>
</tr>
<tr>
<td>15.22. COE QUOTAS CREATE</td>
<td>276</td>
</tr>
<tr>
<td>15.23. COE QUOTAS DELETE</td>
<td>277</td>
</tr>
<tr>
<td>15.24. COE QUOTAS LIST</td>
<td>277</td>
</tr>
<tr>
<td>15.25. COE QUOTAS SHOW</td>
<td>278</td>
</tr>
<tr>
<td>15.26. COE QUOTAS UPDATE</td>
<td>278</td>
</tr>
<tr>
<td>15.27. COE SERVICE LIST</td>
<td>279</td>
</tr>
<tr>
<td>15.28. COE STATS LIST</td>
<td>280</td>
</tr>
<tr>
<td><strong>CHAPTER 16. COMMAND</strong></td>
<td>281</td>
</tr>
<tr>
<td>16.1. COMMAND LIST</td>
<td>281</td>
</tr>
<tr>
<td><strong>CHAPTER 17. COMPLETE</strong></td>
<td>283</td>
</tr>
<tr>
<td>17.1. COMPLETE</td>
<td>283</td>
</tr>
<tr>
<td><strong>CHAPTER 18. COMPUTE</strong></td>
<td>284</td>
</tr>
<tr>
<td>18.1. COMPUTE AGENT CREATE</td>
<td>284</td>
</tr>
<tr>
<td>18.2. COMPUTE AGENT DELETE</td>
<td>285</td>
</tr>
<tr>
<td>18.3. COMPUTE AGENT LIST</td>
<td>286</td>
</tr>
<tr>
<td>18.4. COMPUTE AGENT SET</td>
<td>287</td>
</tr>
<tr>
<td>18.5. COMPUTE SERVICE DELETE</td>
<td>288</td>
</tr>
<tr>
<td>18.6. COMPUTE SERVICE LIST</td>
<td>288</td>
</tr>
<tr>
<td>18.7. COMPUTE SERVICE SET</td>
<td>290</td>
</tr>
<tr>
<td><strong>CHAPTER 19. CONFIGURATION</strong></td>
<td>292</td>
</tr>
<tr>
<td>19.1. CONFIGURATION SHOW</td>
<td>292</td>
</tr>
<tr>
<td><strong>CHAPTER 20. CONSISTENCY</strong></td>
<td>294</td>
</tr>
<tr>
<td>20.1. CONSISTENCY GROUP ADD VOLUME</td>
<td>294</td>
</tr>
<tr>
<td>20.2. CONSISTENCY GROUP CREATE</td>
<td>294</td>
</tr>
<tr>
<td>20.3. CONSISTENCY GROUP DELETE</td>
<td>296</td>
</tr>
<tr>
<td>20.4. CONSISTENCY GROUP LIST</td>
<td>296</td>
</tr>
<tr>
<td>20.5. CONSISTENCY GROUP REMOVE VOLUME</td>
<td>298</td>
</tr>
<tr>
<td>20.6. CONSISTENCY GROUP SET</td>
<td>298</td>
</tr>
<tr>
<td>20.7. CONSISTENCY GROUP SHOW</td>
<td>299</td>
</tr>
<tr>
<td>20.8. CONSISTENCY GROUP SNAPSHOT CREATE</td>
<td>300</td>
</tr>
<tr>
<td>20.9. CONSISTENCY GROUP SNAPSHOT DELETE</td>
<td>302</td>
</tr>
<tr>
<td>20.10. CONSISTENCY GROUP SNAPSHOT LIST</td>
<td>302</td>
</tr>
<tr>
<td>20.11. CONSISTENCY GROUP SNAPSHOT SHOW</td>
<td>304</td>
</tr>
<tr>
<td><strong>CHAPTER 21. CONSOLE</strong></td>
<td>306</td>
</tr>
<tr>
<td>21.1. CONSOLE LOG SHOW</td>
<td>306</td>
</tr>
<tr>
<td>21.2. CONSOLE URL SHOW</td>
<td>306</td>
</tr>
</tbody>
</table>
36.1. FLOATING IP CREATE
36.2. FLOATING IP DELETE
36.3. FLOATING IP LIST
36.4. FLOATING IP POOL LIST
36.5. FLOATING IP PORT FORWARDING CREATE
36.6. FLOATING IP PORT FORWARDING DELETE
36.7. FLOATING IP PORT FORWARDING LIST
36.8. FLOATING IP PORT FORWARDING SET
36.9. FLOATING IP PORT FORWARDING SHOW
36.10. FLOATING IP SET
36.11. FLOATING IP SHOW
36.12. FLOATING IP UNSET

CHAPTER 37. GROUP
37.1. GROUP ADD USER
37.2. GROUP CONTAINS USER
37.3. GROUP CREATE
37.4. GROUP DELETE
37.5. GROUP LIST
37.6. GROUP REMOVE USER
37.7. GROUP SET
37.8. GROUP SHOW

CHAPTER 38. HELP
38.1. HELP

CHAPTER 39. HOST
39.1. HOST LIST
39.2. HOST SET
39.3. HOST SHOW

CHAPTER 40. HYPERVISOR
40.1. HYPERVISOR LIST
40.2. HYPERVISOR SHOW
40.3. HYPERVISOR STATS SHOW

CHAPTER 41. IDENTITY
41.1. IDENTITY PROVIDER CREATE
41.2. IDENTITY PROVIDER DELETE
41.3. IDENTITY PROVIDER LIST
41.4. IDENTITY PROVIDER SET
41.5. IDENTITY PROVIDER SHOW

CHAPTER 42. IMAGE
42.1. IMAGE ADD PROJECT
42.2. IMAGE CREATE
42.3. IMAGE DELETE
42.4. IMAGE LIST
42.5. IMAGE MEMBER LIST
42.6. IMAGE REMOVE PROJECT
42.7. IMAGE SAVE
42.8. IMAGE SET
42.9. IMAGE SHOW
42.10. IMAGE UNSET
49.19. LOADBALANCER DELETE
49.20. LOADBALANCER FAILOVER
49.21. LOADBALANCER FLAVOR CREATE
49.22. LOADBALANCER FLAVOR DELETE
49.23. LOADBALANCER FLAVOR LIST
49.24. LOADBALANCER FLAVOR SET
49.25. LOADBALANCER FLAVOR SHOW
49.26. LOADBALANCER FLAVOR UNSET
49.27. LOADBALANCER FLAVORPROFILE CREATE
49.28. LOADBALANCER FLAVORPROFILE DELETE
49.29. LOADBALANCER FLAVORPROFILE LIST
49.30. LOADBALANCER FLAVORPROFILE SET
49.31. LOADBALANCER FLAVORPROFILE SHOW
49.32. LOADBALANCER HEALTHMONITOR CREATE
49.33. LOADBALANCER HEALTHMONITOR DELETE
49.34. LOADBALANCER HEALTHMONITOR LIST
49.35. LOADBALANCER HEALTHMONITOR SET
49.36. LOADBALANCER HEALTHMONITOR SHOW
49.37. LOADBALANCER HEALTHMONITOR UNSET
49.38. LOADBALANCER L7POLICY CREATE
49.39. LOADBALANCER L7POLICY DELETE
49.40. LOADBALANCER L7POLICY LIST
49.41. LOADBALANCER L7POLICY SET
49.42. LOADBALANCER L7POLICY SHOW
49.43. LOADBALANCER L7POLICY UNSET
49.44. LOADBALANCER L7RULE CREATE
49.45. LOADBALANCER L7RULE DELETE
49.46. LOADBALANCER L7RULE LIST
49.47. LOADBALANCER L7RULE SET
49.48. LOADBALANCER L7RULE SHOW
49.49. LOADBALANCER L7RULE UNSET
49.50. LOADBALANCER LIST
49.51. LOADBALANCER LISTENER CREATE
49.52. LOADBALANCER LISTENER DELETE
49.53. LOADBALANCER LISTENER LIST
49.54. LOADBALANCER LISTENER SET
49.55. LOADBALANCER LISTENER SHOW
49.56. LOADBALANCER LISTENER STATS SHOW
49.57. LOADBALANCER LISTENER UNSET
49.58. LOADBALANCER MEMBER CREATE
49.59. LOADBALANCER MEMBER DELETE
49.60. LOADBALANCER MEMBER LIST
49.61. LOADBALANCER MEMBER SET
49.62. LOADBALANCER MEMBER SHOW
49.63. LOADBALANCER MEMBER UNSET
49.64. LOADBALANCER POOL CREATE
49.65. LOADBALANCER POOL DELETE
49.66. LOADBALANCER POOL LIST
49.67. LOADBALANCER POOL SET
49.68. LOADBALANCER POOL SHOW
49.69. LOADBALANCER POOL UNSET
49.70. LOADBALANCER PROVIDER CAPABILITY LIST
49.71. LOADBALANCER PROVIDER LIST
54.63. NETWORK SEGMENT RANGE LIST 770
54.64. NETWORK SEGMENT RANGE SET 772
54.65. NETWORK SEGMENT RANGE SHOW 772
54.66. NETWORK SEGMENT SET 774
54.67. NETWORK SEGMENT SHOW 774
54.68. NETWORK SERVICE PROVIDER LIST 776
54.69. NETWORK SET 777
54.70. NETWORK SHOW 779
54.71. NETWORK SUBPORT LIST 780
54.72. NETWORK TRUNK CREATE 782
54.73. NETWORK TRUNK DELETE 784
54.74. NETWORK TRUNK LIST 784
54.75. NETWORK TRUNK SET 785
54.76. NETWORK TRUNK SHOW 786
54.77. NETWORK TRUNK UNSET 787
54.78. NETWORK UNSET 788

CHAPTER 55. OBJECT ................................................................. 789
55.1. OBJECT CREATE 789
55.2. OBJECT DELETE 790
55.3. OBJECT LIST 791
55.4. OBJECT SAVE 793
55.5. OBJECT SET 793
55.6. OBJECT SHOW 794
55.7. OBJECT STORE ACCOUNT SET 795
55.8. OBJECT STORE ACCOUNT SHOW 796
55.9. OBJECT STORE ACCOUNT UNSET 797
55.10. OBJECT UNSET 797

CHAPTER 56. ORCHESTRATION ......................................................... 799
56.1. ORCHESTRATION BUILD INFO 799
56.2. ORCHESTRATION RESOURCE TYPE LIST 800
56.3. ORCHESTRATION RESOURCE TYPE SHOW 801
56.4. ORCHESTRATION SERVICE LIST 803
56.5. ORCHESTRATION TEMPLATE FUNCTION LIST 804
56.6. ORCHESTRATION TEMPLATE VALIDATE 806
56.7. ORCHESTRATION TEMPLATE VERSION LIST 808

CHAPTER 57. OVERCLOUD ............................................................... 810
57.1. OVERCLOUD ADMIN AUTHORIZE 810
57.2. OVERCLOUD BACKUP 811
57.3. OVERCLOUD CELL EXPORT 812
57.4. OVERCLOUD CEPH DEPLOY 813
57.5. OVERCLOUD CEPH SPEC 817
57.6. OVERCLOUD CEPH USER DISABLE 819
57.7. OVERCLOUD CEPH USER ENABLE 820
57.8. OVERCLOUD CONTAINER IMAGE BUILD 821
57.9. OVERCLOUD CONTAINER IMAGE PREPARE 822
57.10. OVERCLOUD CONTAINER IMAGE TAG DISCOVER 824
57.11. OVERCLOUD CONTAINER IMAGE UPLOAD 824
57.12. OVERCLOUD CREDENTIALS 825
57.13. OVERCLOUD DELETE 826
57.14. OVERCLOUD DEPLOY 827
57.15. OVERCLOUD EXPORT CEPH 832
57.16. OVERCLOUD EXPORT
57.17. OVERCLOUD EXTERNAL-UPDATE RUN
57.18. OVERCLOUD EXTERNAL-UPGRADE RUN
57.19. OVERCLOUD GENERATE FENCING
57.20. OVERCLOUD IMAGE BUILD
57.21. OVERCLOUD IMAGE UPLOAD
57.22. OVERCLOUD NETENV VALIDATE
57.23. OVERCLOUD NETWORK EXTRACT
57.24. OVERCLOUD NETWORK PROVISION
57.25. OVERCLOUD NETWORK UNPROVISION
57.26. OVERCLOUD NETWORK VIP EXTRACT
57.27. OVERCLOUD NETWORK VIP PROVISION
57.28. OVERCLOUD NODE BIOS CONFIGURE
57.29. OVERCLOUD NODE BIOS RESET
57.30. OVERCLOUD NODE CLEAN
57.31. OVERCLOUD NODE CONFIGURE
57.32. OVERCLOUD NODE DELETE
57.33. OVERCLOUD NODE DISCOVER
57.34. OVERCLOUD NODE EXTRACT PROVISIONED
57.35. OVERCLOUD NODE IMPORT
57.36. OVERCLOUD NODE INTROSPECT
57.37. OVERCLOUD NODE PROVIDE
57.38. OVERCLOUD NODE PROVISION
57.39. OVERCLOUD NODE UNPROVISION
57.40. OVERCLOUD PROFILES LIST
57.41. OVERCLOUD PROFILES MATCH
57.42. OVERCLOUD RAID CREATE
57.43. OVERCLOUD RESTORE
57.44. OVERCLOUD ROLE LIST
57.45. OVERCLOUD ROLE SHOW
57.46. OVERCLOUD ROLES GENERATE
57.47. OVERCLOUD STATUS
57.48. OVERCLOUD SUPPORT REPORT COLLECT
57.49. OVERCLOUD UPDATE PREPARE
57.50. OVERCLOUD UPDATE RUN
57.51. OVERCLOUD UPGRADE CONVERGE
57.52. OVERCLOUD UPGRADE PREPARE
57.53. OVERCLOUD UPGRADE RUN

CHAPTER 58. POLICY

58.1. POLICY CREATE
58.2. POLICY DELETE
58.3. POLICY LIST
58.4. POLICY SET
58.5. POLICY SHOW

CHAPTER 59. PORT

59.1. PORT CREATE
59.2. PORT DELETE
59.3. PORT LIST
59.4. PORT SET
59.5. PORT SHOW
59.6. PORT UNSET
68.3. ROLE CREATE
68.4. ROLE DELETE
68.5. ROLE LIST
68.6. ROLE REMOVE
68.7. ROLE SET
68.8. ROLE SHOW

CHAPTER 69. ROUTER ........................................... 964
  69.1. ROUTER ADD PORT ................................. 964
  69.2. ROUTER ADD ROUTE ................................ 964
  69.3. ROUTER ADD SUBNET ................................ 966
  69.4. ROUTER CREATE ...................................... 966
  69.5. ROUTER DELETE ...................................... 968
  69.6. ROUTER LIST .......................................... 968
  69.7. ROUTER REMOVE PORT ............................ 971
  69.8. ROUTER REMOVE ROUTE ............................ 971
  69.9. ROUTER REMOVE SUBNET ......................... 973
  69.10. ROUTER SET .......................................... 973
  69.11. ROUTER SHOW ....................................... 975
  69.12. ROUTER UNSET ..................................... 976

CHAPTER 70. SECRET .............................................. 978
  70.1. SECRET CONTAINER CREATE ....................... 978
  70.2. SECRET CONTAINER DELETE ....................... 979
  70.3. SECRET CONTAINER GET ............................. 979
  70.4. SECRET CONTAINER LIST ........................... 981
  70.5. SECRET DELETE ....................................... 982
  70.6. SECRET GET ........................................... 983
  70.7. SECRET LIST ........................................... 984
  70.8. SECRET ORDER CREATE ............................. 986
  70.9. SECRET ORDER DELETE ............................. 988
  70.10. SECRET ORDER GET .................................. 988
  70.11. SECRET ORDER LIST ............................... 990
  70.12. SECRET STORE ....................................... 991
  70.13. SECRET UPDATE ..................................... 993

CHAPTER 71. SECURITY ........................................... 994
  71.1. SECURITY GROUP CREATE ........................... 994
  71.2. SECURITY GROUP DELETE ........................... 995
  71.3. SECURITY GROUP LIST ............................... 996
  71.4. SECURITY GROUP RULE CREATE ................. 998
  71.5. SECURITY GROUP RULE DELETE .................... 1000
  71.6. SECURITY GROUP RULE LIST ....................... 1001
  71.7. SECURITY GROUP RULE SHOW ...................... 1002
  71.8. SECURITY GROUP SET ................................ 1004
  71.9. SECURITY GROUP SHOW ............................. 1005
  71.10. SECURITY GROUP UNSET ......................... 1006

CHAPTER 72. SERVER ............................................. 1007
  72.1. SERVER ADD FIXED IP ............................... 1007
  72.2. SERVER ADD FLOATING IP ......................... 1007
  72.3. SERVER ADD NETWORK .............................. 1008
  72.4. SERVER ADD PORT .................................... 1008
  72.5. SERVER ADD SECURITY GROUP ..................... 1009
<table>
<thead>
<tr>
<th>Number</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>72.6</td>
<td>SERVER ADD VOLUME</td>
</tr>
<tr>
<td>72.7</td>
<td>SERVER BACKUP CREATE</td>
</tr>
<tr>
<td>72.8</td>
<td>SERVER CREATE</td>
</tr>
<tr>
<td>72.9</td>
<td>SERVER DELETE</td>
</tr>
<tr>
<td>72.10</td>
<td>SERVER DUMP CREATE</td>
</tr>
<tr>
<td>72.11</td>
<td>SERVER EVACUATE</td>
</tr>
<tr>
<td>72.12</td>
<td>SERVER EVENT LIST</td>
</tr>
<tr>
<td>72.13</td>
<td>SERVER EVENT SHOW</td>
</tr>
<tr>
<td>72.14</td>
<td>SERVER GROUP CREATE</td>
</tr>
<tr>
<td>72.15</td>
<td>SERVER GROUP DELETE</td>
</tr>
<tr>
<td>72.16</td>
<td>SERVER GROUP LIST</td>
</tr>
<tr>
<td>72.17</td>
<td>SERVER GROUP SHOW</td>
</tr>
<tr>
<td>72.18</td>
<td>SERVER IMAGE CREATE</td>
</tr>
<tr>
<td>72.19</td>
<td>SERVER LIST</td>
</tr>
<tr>
<td>72.20</td>
<td>SERVER LOCK</td>
</tr>
<tr>
<td>72.21</td>
<td>SERVER MIGRATE CONFIRM</td>
</tr>
<tr>
<td>72.22</td>
<td>SERVER MIGRATE REVERT</td>
</tr>
<tr>
<td>72.23</td>
<td>SERVER MIGRATE</td>
</tr>
<tr>
<td>72.24</td>
<td>SERVER MIGRATION ABORT</td>
</tr>
<tr>
<td>72.25</td>
<td>SERVER MIGRATION CONFIRM</td>
</tr>
<tr>
<td>72.26</td>
<td>SERVER MIGRATION FORCE COMPLETE</td>
</tr>
<tr>
<td>72.27</td>
<td>SERVER MIGRATION LIST</td>
</tr>
<tr>
<td>72.28</td>
<td>SERVER MIGRATION REVERT</td>
</tr>
<tr>
<td>72.29</td>
<td>SERVER MIGRATION SHOW</td>
</tr>
<tr>
<td>72.30</td>
<td>SERVER PAUSE</td>
</tr>
<tr>
<td>72.31</td>
<td>SERVER REBOOT</td>
</tr>
<tr>
<td>72.32</td>
<td>SERVER REBUILD</td>
</tr>
<tr>
<td>72.33</td>
<td>SERVER REMOVE FIXED IP</td>
</tr>
<tr>
<td>72.34</td>
<td>SERVER REMOVE FLOATING IP</td>
</tr>
<tr>
<td>72.35</td>
<td>SERVER REMOVE NETWORK</td>
</tr>
<tr>
<td>72.36</td>
<td>SERVER REMOVE PORT</td>
</tr>
<tr>
<td>72.37</td>
<td>SERVER REMOVE SECURITY GROUP</td>
</tr>
<tr>
<td>72.38</td>
<td>SERVER REMOVE VOLUME</td>
</tr>
<tr>
<td>72.39</td>
<td>SERVER RESCUE</td>
</tr>
<tr>
<td>72.40</td>
<td>SERVER RESIZE CONFIRM</td>
</tr>
<tr>
<td>72.41</td>
<td>SERVER RESIZE REVERT</td>
</tr>
<tr>
<td>72.42</td>
<td>SERVER RESIZE</td>
</tr>
<tr>
<td>72.43</td>
<td>SERVER RESTORE</td>
</tr>
<tr>
<td>72.44</td>
<td>SERVER RESUME</td>
</tr>
<tr>
<td>72.45</td>
<td>SERVER SET</td>
</tr>
<tr>
<td>72.46</td>
<td>SERVER SHELVE</td>
</tr>
<tr>
<td>72.47</td>
<td>SERVER SHOW</td>
</tr>
<tr>
<td>72.48</td>
<td>SERVER SSH</td>
</tr>
<tr>
<td>72.49</td>
<td>SERVER START</td>
</tr>
<tr>
<td>72.50</td>
<td>SERVER STOP</td>
</tr>
<tr>
<td>72.51</td>
<td>SERVER SUSPEND</td>
</tr>
<tr>
<td>72.52</td>
<td>SERVER UNLOCK</td>
</tr>
<tr>
<td>72.53</td>
<td>SERVER UNPAUSE</td>
</tr>
<tr>
<td>72.54</td>
<td>SERVER UNRESCUE</td>
</tr>
<tr>
<td>72.55</td>
<td>SERVER UNSET</td>
</tr>
<tr>
<td>72.56</td>
<td>SERVER UNSHELVE</td>
</tr>
<tr>
<td>72.57</td>
<td>SERVER VOLUME LIST</td>
</tr>
<tr>
<td>72.58</td>
<td>SERVER VOLUME UPDATE</td>
</tr>
</tbody>
</table>
CHAPTER 73. SERVICE
73.1. SERVICE CREATE .................................................. 1062
73.2. SERVICE DELETE ................................................... 1063
73.3. SERVICE LIST ....................................................... 1064
73.4. SERVICE PROVIDER CREATE .................................... 1065
73.5. SERVICE PROVIDER DELETE .................................... 1067
73.6. SERVICE PROVIDER LIST ........................................... 1067
73.7. SERVICE PROVIDER SET ............................................ 1069
73.8. SERVICE PROVIDER SHOW ....................................... 1069
73.9. SERVICE SET ....................................................... 1071
73.10. SERVICE SHOW .................................................... 1072

CHAPTER 74. SFC
74.1. SFC FLOW CLASSIFIER CREATE ................................ 1074
74.2. SFC FLOW CLASSIFIER DELETE ................................ 1076
74.3. SFC FLOW CLASSIFIER LIST ...................................... 1076
74.4. SFC FLOW CLASSIFIER SET ...................................... 1078
74.5. SFC FLOW CLASSIFIER SHOW ................................... 1078
74.6. SFC PORT CHAIN CREATE ......................................... 1080
74.7. SFC PORT CHAIN DELETE ......................................... 1081
74.8. SFC PORT CHAIN LIST .............................................. 1082
74.9. SFC PORT CHAIN SET .............................................. 1083
74.10. SFC PORT CHAIN SHOW .......................................... 1084
74.11. SFC PORT CHAIN UNSET .......................................... 1085
74.12. SFC PORT PAIR CREATE .......................................... 1086
74.13. SFC PORT PAIR DELETE .......................................... 1088
74.14. SFC PORT PAIR GROUP CREATE ................................. 1088
74.15. SFC PORT PAIR GROUP DELETE ................................. 1090
74.16. SFC PORT PAIR GROUP LIST .................................... 1090
74.17. SFC PORT PAIR GROUP SET .................................... 1092
74.18. SFC PORT PAIR GROUP SHOW ................................ 1092
74.19. SFC PORT PAIR GROUP UNSET ................................ 1094
74.20. SFC PORT PAIR LIST .............................................. 1094
74.21. SFC PORT PAIR SET .............................................. 1096
74.22. SFC PORT PAIR SHOW ............................................ 1096
74.23. SFC SERVICE GRAPH CREATE ................................. 1098
74.24. SFC SERVICE GRAPH DELETE ................................. 1099
74.25. SFC SERVICE GRAPH LIST .................................... 1100
74.26. SFC SERVICE GRAPH SET .................................... 1101
74.27. SFC SERVICE GRAPH SHOW ................................ 1102

CHAPTER 75. SHARE
75.1. SHARE ABANDON .................................................... 1104
75.2. SHARE ACCESS CREATE .......................................... 1104
75.3. SHARE ACCESS DELETE .......................................... 1106
75.4. SHARE ACCESS LIST .............................................. 1106
75.5. SHARE ACCESS SET .............................................. 1108
75.6. SHARE ACCESS SHOW ............................................ 1108
75.7. SHARE ACCESS UNSET ............................................ 1110
75.8. SHARE ADOPT ....................................................... 1110
75.9. SHARE CREATE ..................................................... 1112
75.10. SHARE DELETE ..................................................... 1114
75.11. SHARE EXPORT LOCATION LIST ................................ 1115
<table>
<thead>
<tr>
<th>Chapter 75. Share</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>75.12. Share Export Location Show</td>
<td>1116</td>
</tr>
<tr>
<td>75.13. Share List</td>
<td>1118</td>
</tr>
<tr>
<td>75.14. Share Message Delete</td>
<td>1120</td>
</tr>
<tr>
<td>75.15. Share Message List</td>
<td>1121</td>
</tr>
<tr>
<td>75.16. Share Message Show</td>
<td>1123</td>
</tr>
<tr>
<td>75.17. Share Properties Show</td>
<td>1124</td>
</tr>
<tr>
<td>75.18. Share Quota Delete</td>
<td>1126</td>
</tr>
<tr>
<td>75.19. Share Quota Set</td>
<td>1126</td>
</tr>
<tr>
<td>75.20. Share Quota Show</td>
<td>1127</td>
</tr>
<tr>
<td>75.21. Share Resize</td>
<td>1129</td>
</tr>
<tr>
<td>75.22. Share Revert</td>
<td>1129</td>
</tr>
<tr>
<td>75.23. Share Set</td>
<td>1130</td>
</tr>
<tr>
<td>75.24. Share Show</td>
<td>1131</td>
</tr>
<tr>
<td>75.25. Share Snapshot Abandon</td>
<td>1132</td>
</tr>
<tr>
<td>75.26. Share Snapshot Access Create</td>
<td>1132</td>
</tr>
<tr>
<td>75.27. Share Snapshot Access Delete</td>
<td>1134</td>
</tr>
<tr>
<td>75.28. Share Snapshot Access List</td>
<td>1134</td>
</tr>
<tr>
<td>75.29. Share Snapshot Adopt</td>
<td>1136</td>
</tr>
<tr>
<td>75.30. Share Snapshot Create</td>
<td>1137</td>
</tr>
<tr>
<td>75.31. Share Snapshot Delete</td>
<td>1139</td>
</tr>
<tr>
<td>75.32. Share Snapshot Export Location List</td>
<td>1139</td>
</tr>
<tr>
<td>75.33. Share Snapshot Export Location Show</td>
<td>1141</td>
</tr>
<tr>
<td>75.34. Share Snapshot List</td>
<td>1142</td>
</tr>
<tr>
<td>75.35. Share Snapshot Set</td>
<td>1144</td>
</tr>
<tr>
<td>75.36. Share Snapshot Show</td>
<td>1145</td>
</tr>
<tr>
<td>75.37. Share Snapshot Unset</td>
<td>1146</td>
</tr>
<tr>
<td>75.38. Share Type Access Create</td>
<td>1147</td>
</tr>
<tr>
<td>75.39. Share Type Access Delete</td>
<td>1147</td>
</tr>
<tr>
<td>75.40. Share Type Access List</td>
<td>1148</td>
</tr>
<tr>
<td>75.41. Share Type Create</td>
<td>1149</td>
</tr>
<tr>
<td>75.42. Share Type Delete</td>
<td>1151</td>
</tr>
<tr>
<td>75.43. Share Type List</td>
<td>1152</td>
</tr>
<tr>
<td>75.44. Share Type Set</td>
<td>1153</td>
</tr>
<tr>
<td>75.45. Share Type Show</td>
<td>1154</td>
</tr>
<tr>
<td>75.46. Share Type Unset</td>
<td>1155</td>
</tr>
<tr>
<td>75.47. Share Unset</td>
<td>1156</td>
</tr>
</tbody>
</table>

**Chapter 76. Software**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>76.1. Software Config Create</td>
<td>1157</td>
</tr>
<tr>
<td>76.2. Software Config Delete</td>
<td>1157</td>
</tr>
<tr>
<td>76.3. Software Config List</td>
<td>1158</td>
</tr>
<tr>
<td>76.4. Software Config Show</td>
<td>1159</td>
</tr>
<tr>
<td>76.5. Software Deployment Create</td>
<td>1160</td>
</tr>
<tr>
<td>76.6. Software Deployment Delete</td>
<td>1161</td>
</tr>
<tr>
<td>76.7. Software Deployment List</td>
<td>1162</td>
</tr>
<tr>
<td>76.8. Software Deployment Metadata Show</td>
<td>1163</td>
</tr>
<tr>
<td>76.9. Software Deployment Output Show</td>
<td>1164</td>
</tr>
<tr>
<td>76.10. Software Deployment Show</td>
<td>1165</td>
</tr>
</tbody>
</table>

**Chapter 77. Stack**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>77.1. Stack Abandon</td>
<td>1169</td>
</tr>
<tr>
<td>77.2. Stack Adopt</td>
<td>1169</td>
</tr>
<tr>
<td>77.3. Stack Cancel</td>
<td>1170</td>
</tr>
<tr>
<td>Chapter</td>
<td>Section</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>77.1</td>
<td>STACK CHECK</td>
</tr>
<tr>
<td>77.2</td>
<td>STACK CREATE</td>
</tr>
<tr>
<td>77.3</td>
<td>STACK DELETE</td>
</tr>
<tr>
<td>77.4</td>
<td>STACK ENVIRONMENT SHOW</td>
</tr>
<tr>
<td>77.5</td>
<td>STACK EVENT LIST</td>
</tr>
<tr>
<td>77.6</td>
<td>STACK EVENT SHOW</td>
</tr>
<tr>
<td>77.7</td>
<td>STACK EXPORT</td>
</tr>
<tr>
<td>77.8</td>
<td>STACK FAILURES LIST</td>
</tr>
<tr>
<td>77.9</td>
<td>STACK FILE LIST</td>
</tr>
<tr>
<td>77.10</td>
<td>STACK HOOK CLEAR</td>
</tr>
<tr>
<td>77.11</td>
<td>STACK HOOK POLL</td>
</tr>
<tr>
<td>77.12</td>
<td>STACK LIST</td>
</tr>
<tr>
<td>77.13</td>
<td>STACK OUTPUT LIST</td>
</tr>
<tr>
<td>77.14</td>
<td>STACK OUTPUT SHOW</td>
</tr>
<tr>
<td>77.15</td>
<td>STACK RESOURCE LIST</td>
</tr>
<tr>
<td>77.16</td>
<td>STACK RESOURCE MARK UNHEALTHY</td>
</tr>
<tr>
<td>77.17</td>
<td>STACK RESOURCE METADATA</td>
</tr>
<tr>
<td>77.18</td>
<td>STACK RESOURCE SHOW</td>
</tr>
<tr>
<td>77.19</td>
<td>STACK RESOURCE SIGNAL</td>
</tr>
<tr>
<td>77.20</td>
<td>STACK RESUME</td>
</tr>
<tr>
<td>77.21</td>
<td>STACK SHOW</td>
</tr>
<tr>
<td>77.22</td>
<td>STACK SNAPSHOTS CREATE</td>
</tr>
<tr>
<td>77.23</td>
<td>STACK SNAPSHOTS DELETE</td>
</tr>
<tr>
<td>77.24</td>
<td>STACK SNAPSHOTS LIST</td>
</tr>
<tr>
<td>77.25</td>
<td>STACK SNAPSHOTS RESTORE</td>
</tr>
<tr>
<td>77.26</td>
<td>STACK SNAPSHOTS SHOW</td>
</tr>
<tr>
<td>77.27</td>
<td>STACK SUSPEND</td>
</tr>
<tr>
<td>77.28</td>
<td>STACK TEMPLATE SHOW</td>
</tr>
<tr>
<td>77.29</td>
<td>STACK UPDATE</td>
</tr>
</tbody>
</table>

**CHAPTER 78. SUBNET** | 1213
| 78.1    | SUBNET CREATE | 1213 |
| 78.2    | SUBNET DELETE | 1216 |
| 78.3    | SUBNET LIST | 1216 |
| 78.4    | SUBNET POOL CREATE | 1219 |
| 78.5    | SUBNET POOL DELETE | 1221 |
| 78.6    | SUBNET POOL LIST | 1221 |
| 78.7    | SUBNET POOL SET | 1224 |
| 78.8    | SUBNET POOL SHOW | 1225 |
| 78.9    | SUBNET POOL UNSET | 1226 |
| 78.10   | SUBNET SET | 1227 |
| 78.11   | SUBNET SHOW | 1229 |
| 78.12   | SUBNET UNSET | 1230 |

**CHAPTER 79. TASK** | 1232
| 79.1    | TASK EXECUTION LIST | 1232 |
| 79.2    | TASK EXECUTION PUBLISHED SHOW | 1234 |
| 79.3    | TASK EXECUTION RERUN | 1234 |
| 79.4    | TASK EXECUTION RESULT SHOW | 1236 |
| 79.5    | TASK EXECUTION SHOW | 1236 |

**CHAPTER 80. TLD** | 1238
| 80.1    | TLD CREATE | 1238 |
| 80.2    | TLD DELETE | 1239 |
CHAPTER 81. TOKEN ................................................................. 1245
  81.1. TOKEN ISSUE 1245
  81.2. TOKEN REVOKE 1246

CHAPTER 82. TRIPLEO .......................................................... 1247
  82.1. TRIPLEO CONFIG GENERATE ANSIBLE 1247
  82.2. TRIPLEO CONTAINER IMAGE BUILD 1247
  82.3. TRIPLEO CONTAINER IMAGE DELETE 1249
  82.4. TRIPLEO CONTAINER IMAGE HOTFIX 1250
  82.5. TRIPLEO CONTAINER IMAGE LIST 1250
  82.6. TRIPLEO CONTAINER IMAGE PREPARE DEFAULT 1252
  82.7. TRIPLEO CONTAINER IMAGE PREPARE 1253
  82.8. TRIPLEO CONTAINER IMAGE PUSH 1254
  82.9. TRIPLEO CONTAINER IMAGE SHOW 1255
  82.10. TRIPLEO DEPLOY 1256
  82.11. TRIPLEO LAUNCH HEAT 1260
  82.12. TRIPLEO UPGRADE 1262
  82.13. TRIPLEO VALIDATOR GROUP INFO 1265
  82.14. TRIPLEO VALIDATOR INIT 1266
  82.15. TRIPLEO VALIDATOR LIST 1267
  82.16. TRIPLEO VALIDATOR RUN 1269
  82.17. TRIPLEO VALIDATOR SHOW HISTORY 1271
  82.18. TRIPLEO VALIDATOR SHOW PARAMETER 1272
  82.19. TRIPLEO VALIDATOR SHOW RUN 1274
  82.20. TRIPLEO VALIDATOR SHOW 1275

CHAPTER 83. TRUST ............................................................... 1277
  83.1. TRUST CREATE 1277
  83.2. TRUST DELETE 1278
  83.3. TRUST LIST 1279
  83.4. TRUST SHOW 1280

CHAPTER 84. TSIGKEY ............................................................. 1282
  84.1. TSIGKEY CREATE 1282
  84.2. TSIGKEY DELETE 1283
  84.3. TSIGKEY LIST 1284
  84.4. TSIGKEY SET 1285
  84.5. TSIGKEY SHOW 1287

CHAPTER 85. UNDERCLOUD .................................................... 1289
  85.1. UNDERCLOUD BACKUP 1289
  85.2. UNDERCLOUD INSTALL 1290
  85.3. UNDERCLOUD UPGRADE 1291

CHAPTER 86. USAGE ............................................................... 1293
  86.1. USAGE LIST 1293
  86.2. USAGE SHOW 1294

CHAPTER 87. USER ............................................................... 1296
  87.1. USER CREATE 1296
  87.2. USER DELETE 1299
| 92.21. ZONE SHOW                           | 1419 |
| 92.22. ZONE TRANSFER ACCEPT LIST         | 1421 |
| 92.23. ZONE TRANSFER ACCEPT REQUEST      | 1422 |
| 92.24. ZONE TRANSFER ACCEPT SHOW         | 1424 |
| 92.25. ZONE TRANSFER REQUEST CREATE      | 1425 |
| 92.26. ZONE TRANSFER REQUEST DELETE      | 1427 |
| 92.27. ZONE TRANSFER REQUEST LIST        | 1427 |
| 92.28. ZONE TRANSFER REQUEST SET         | 1429 |
| 92.29. ZONE TRANSFER REQUEST SHOW        | 1431 |
Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Tell us how we can make it better.

Using the Direct Documentation Feedback (DDF) function

Use the Add Feedback DDF function for direct comments on specific sentences, paragraphs, or code blocks.

1. View the documentation in the Multi-page HTML format.
2. Ensure that you see the Feedback button in the upper right corner of the document.
3. Highlight the part of text that you want to comment on.
4. Click Add Feedback.
5. Complete the Add Feedback field with your comments.
6. Optional: Add your email address so that the documentation team can contact you for clarification on your issue.
7. Click Submit.
CHAPTER 1. THE OPENSTACK CLIENT

The openstack client is a common OpenStack command-line interface (CLI). This chapter documents the main options for openstack version 5.5.1.

Command-line interface to the OpenStack APIs

Usage:

[--os-cloud <cloud-config-name>]
[--os-region-name <auth-region-name>]
[--os-cacert <ca-bundle-file>] [--os-cert <certificate-file>]
[--os-key <key-file>] [--verify | --insecure]
[--os-default-domain <auth-domain>]
[--os-interface <interface>]
[--os-service-provider <service_provider>]
[--os-remote-project-name <remote_project_name> | --os-remote-project-id <remote_project_id>]
[--os-remote-project-domain-name <remote_project_domain_name> | --os-remote-project-domain-id <remote_project_domain_id>]
[--timing] [--os-beta-command] [--os-profile hmac-key]
[--os-compute-api-version <compute-api-version>]
[--os-identity-api-version <identity-api-version>]
[--os-image-api-version <image-api-version>]
[--os-network-api-version <network-api-version>]
[--os-object-api-version <object-api-version>]
[--os-volume-api-version <volume-api-version>]
[--os-metrics-api-version <metrics-api-version>]
[--os-dns-api-version <dns-api-version>]
[--os-key-manager-api-version <key-manager-api-version>]
[--os-orchestration-api-version <orchestration-api-version>]
[--os-loadbalancer-api-version <loadbalancer-api-version>]
[--os-data-processing-api-version <data-processing-api-version>]
[--os-data-processing-url OS_DATA_PROCESSING_URL]
[--os-workflow-api-version <workflow-api-version>]
[--os-database-api-version <database-api-version>]
[--os-share-api-version <shared-file-system-api-version>]
[--os-tripleoclient-api-version <tripleoclient-api-version>]
[--os-queues-api-version <queues-api-version>]
[--os-container-infra-api-version <container-infra-api-version>]
[--os-baremetal-api-version <baremetal-api-version>]
[--inspector-api-version INSPECTOR_API_VERSION]
[--inspector-url INSPECTOR_URL]
[--os-alarming-api-version <alarming-api-version>]
[--os-auth-type <auth-type>] [--os-endpoint <auth-endpoint>]
[--os-auth-url <auth-url>]
[--os-system-scope <auth-system-scope>]
[--os-domain-id <auth-domain-id>]
[--os-domain-name <auth-domain-name>]
[--os-project-id <auth-project-id>]
[--os-project-name <auth-project-name>]
[--os-project-domain-id <auth-project-domain-id>]
[--os-project-domain-name <auth-project-domain-name>]
[--os-trust-id <auth-trust-id>]
[--os-default-domain-id <auth-default-domain-id>]
Table 1.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--version</td>
<td>Show program’s version number and exit</td>
</tr>
<tr>
<td>-v, --verbose</td>
<td>Increase verbosity of output. can be repeated.</td>
</tr>
<tr>
<td>-q, --quiet</td>
<td>Suppress output except warnings and errors.</td>
</tr>
<tr>
<td>--log-file LOG_FILE</td>
<td>Specify a file to log output. disabled by default.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show help message and exit.</td>
</tr>
<tr>
<td>--debug</td>
<td>Show tracebacks on errors.</td>
</tr>
<tr>
<td>--os-cloud &lt;cloud-config-name&gt;</td>
<td>Cloud name in clouds.yaml (env: os_cloud)</td>
</tr>
<tr>
<td>--os-region-name &lt;auth-region-name&gt;</td>
<td>Authentication region name (env: os_region_name)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--os-cacert &lt;ca-bundle-file&gt;</code></td>
<td>Ca certificate bundle file (env: os_cacert)</td>
</tr>
<tr>
<td><code>--os-cert &lt;certificate-file&gt;</code></td>
<td>Client certificate bundle file (env: os_cert)</td>
</tr>
<tr>
<td><code>--os-key &lt;key-file&gt;</code></td>
<td>Client certificate key file (env: os_key)</td>
</tr>
<tr>
<td><code>--verify</code></td>
<td>Verify server certificate (default)</td>
</tr>
<tr>
<td><code>--insecure</code></td>
<td>Disable server certificate verification</td>
</tr>
<tr>
<td><code>--os-default-domain &lt;auth-domain&gt;</code></td>
<td>Default domain id, default=default. (env: OS_DEFAULT_DOMAIN)</td>
</tr>
<tr>
<td><code>--os-interface &lt;interface&gt;</code></td>
<td>Select an interface type. valid interface types: [admin, public, internal]. default=public, (Env: OS_INTERFACE)</td>
</tr>
<tr>
<td><code>--os-service-provider &lt;service_provider&gt;</code></td>
<td>Authenticate with and perform the command on a service provider using Keystone-to-keystone federation. Must also specify the remote project option.</td>
</tr>
<tr>
<td><code>--os-remote-project-name &lt;remote_project_name&gt;</code></td>
<td>Project name when authenticating to a service provider if using Keystone-to-Keystone federation.</td>
</tr>
<tr>
<td><code>--os-remote-project-id &lt;remote_project_id&gt;</code></td>
<td>Project id when authenticating to a service provider if using Keystone-to-Keystone federation.</td>
</tr>
<tr>
<td><code>--os-remote-project-domain-name &lt;remote_project_domain_name&gt;</code></td>
<td>Domain name of the project when authenticating to a service provider if using Keystone-to-Keystone federation.</td>
</tr>
<tr>
<td><code>--os-remote-project-domain-id &lt;remote_project_domain_id&gt;</code></td>
<td>Domain id of the project when authenticating to a service provider if using Keystone-to-Keystone federation.</td>
</tr>
<tr>
<td><code>--timing</code></td>
<td>Print api call timing info</td>
</tr>
<tr>
<td><code>--os-beta-command</code></td>
<td>Enable beta commands which are subject to change</td>
</tr>
<tr>
<td><code>--os-profile hmac-key</code></td>
<td>Hmac key for encrypting profiling context data</td>
</tr>
<tr>
<td><code>--os-compute-api-version &lt;compute-api-version&gt;</code></td>
<td>Compute api version, default=2.1 (env: OS_COMPUTE_API_VERSION)</td>
</tr>
<tr>
<td><code>--os-identity-api-version &lt;identity-api-version&gt;</code></td>
<td>Identity api version, default=3 (env: OS_IDENTITY_API_VERSION)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--os-image-api-version &lt;image-api-version&gt;</td>
<td>Image api version, default=2 (env: OS_IMAGE_API_VERSION)</td>
</tr>
<tr>
<td>--os-network-api-version &lt;network-api-version&gt;</td>
<td>Network api version, default=2.0 (env: OS_NETWORK_API_VERSION)</td>
</tr>
<tr>
<td>--os-object-api-version &lt;object-api-version&gt;</td>
<td>Object api version, default=1 (env: OS_OBJECT_API_VERSION)</td>
</tr>
<tr>
<td>--os-volume-api-version &lt;volume-api-version&gt;</td>
<td>Volume api version, default=3 (env: OS_VOLUME_API_VERSION)</td>
</tr>
<tr>
<td>--os-metrics-api-version &lt;metrics-api-version&gt;</td>
<td>Metrics api version, default=1 (env: OS_METRICS_API_VERSION)</td>
</tr>
<tr>
<td>--os-dns-api-version &lt;dns-api-version&gt;</td>
<td>Dns api version, default=2 (env: os_dns_api_version)</td>
</tr>
<tr>
<td>--os-key-manager-api-version &lt;key-manager-api-version&gt;</td>
<td>Barbican api version, default=1 (env: OS_KEY_MANAGER_API_VERSION)</td>
</tr>
<tr>
<td>--os-orchestration-api-version &lt;orchestration-api-version&gt;</td>
<td>Orchestration api version, default=1 (env: OS_ORCHESTRATION_API_VERSION)</td>
</tr>
<tr>
<td>--os-loadbalancer-api-version &lt;loadbalancer-api-version&gt;</td>
<td>Osc plugin api version, default=2.0 (env: OS_LOADBALANCER_API_VERSION)</td>
</tr>
<tr>
<td>--os-data-processing-url</td>
<td>Data processing api url, (env: OS_DATA_PROCESSING_API_URL)</td>
</tr>
<tr>
<td>--os-workflow-api-version &lt;workflow-api-version&gt;</td>
<td>Workflow api version, default=2 (env: OS_WORKFLOW_API_VERSION)</td>
</tr>
<tr>
<td>--os-database-api-version &lt;database-api-version&gt;</td>
<td>Database api version, default=1 (env: OS_DATABASE_API_VERSION)</td>
</tr>
<tr>
<td>--os-share-api-version &lt;shared-file-system-api-version&gt;</td>
<td>Shared file system api version, default=2.63version supported by both the client and the server). (Env: OS_SHARE_API_VERSION)</td>
</tr>
<tr>
<td>--os-tripleoclient-api-version &lt;tripleoclient-api-version&gt;</td>
<td>Tripleo client api version, default=2 (env: OS_TRIPLEOCLIENT_API_VERSION)</td>
</tr>
<tr>
<td>--os-queues-api-version &lt;queues-api-version&gt;</td>
<td>Queues api version, default=2 (env: OS_QUEUES_API_VERSION)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--os-container-infra-api-version &lt;container-infra-api-version&gt;</td>
<td>Container-infra api version, default=1 (env: OS_CONTAINER_INFRA_API_VERSION)</td>
</tr>
<tr>
<td>--os-baremetal-api-version &lt;baremetal-api-version&gt;</td>
<td>Bare metal api version, default=&quot;latest&quot; (the maximum version supported by both the client and the server). (Env: OS_BAREMETAL_API_VERSION)</td>
</tr>
<tr>
<td>--inspector-api-version INSPECTOR_API_VERSION</td>
<td>Inspector api version, only 1 is supported now (env: INSPECTOR_VERSION).</td>
</tr>
<tr>
<td>--inspector-url INSPECTOR_URL</td>
<td>Inspector url, defaults to localhost (env: INSPECTOR_URL).</td>
</tr>
<tr>
<td>--os-alarming-api-version &lt;alarming-api-version&gt;</td>
<td>Queues api version, default=2 (env: OS_ALARMING_API_VERSION)</td>
</tr>
<tr>
<td>--os-endpoint &lt;auth-endpoint&gt;</td>
<td>With none: the endpoint that will always be used with gnocchi-basic: Gnocchi endpoint With noauth: Cinder endpoint With http_basic: The endpoint that will always be used With gnocchi-noauth: Gnocchi endpoint With admin_token: The endpoint that will always be used (Env: OS_ENDPOINT)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--os-system-scope &lt;auth-system-scope&gt;</code></td>
<td>With password: scope for system operations with v3password: Scope for system operations With v3token: Scope for system operations With v3oidcpassword: Scope for system operations With v3multifactor: Scope for system operations With v3adfspassword: Scope for system operations With v3totp: Scope for system operations With v3oidcauthcode: Scope for system operations With v3applicationcredential: Scope for system operations With v3oidcclientcredentials: Scope for system operations With v3oidcaccesstoken: Scope for system operations With v3samlpassword: Scope for system operations (Env: OS_SYSTEM_SCOPE)</td>
</tr>
<tr>
<td><code>--os-domain-id &lt;auth-domain-id&gt;</code></td>
<td>With password: domain id to scope to with v3password: Domain ID to scope to With v3token: Domain ID to scope to With v3tokenlessauth: Domain ID to scope to With v3oidcpassword: Domain ID to scope to With v3multifactor: Domain ID to scope to With v3adfspassword: Domain ID to scope to With v3totp: Domain ID to scope to With v3oidcauthcode: Domain ID to scope to With v3applicationcredential: Domain ID to scope to With v3oidcclientcredentials: Domain ID to scope to With v3oidcaccesstoken: Domain ID to scope to With v3samlpassword: Domain ID to scope to (Env: OS_DOMAIN_ID)</td>
</tr>
<tr>
<td><code>--os-domain-name &lt;auth-domain-name&gt;</code></td>
<td>With password: domain name to scope to with v3password: Domain name to scope to With v3token: Domain name to scope to With v3tokenlessauth: Domain name to scope to With v3oidcpassword: Domain name to scope to With v3multifactor: Domain name to scope to With v3adfspassword: Domain name to scope to With v3totp: Domain name to scope to With v3oidcauthcode: Domain name to scope to With v3applicationcredential: Domain name to scope to With v3oidcclientcredentials: Domain name to scope to With v3oidcaccesstoken: Domain name to scope to With v3samlpassword: Domain name to scope to (Env: OS_DOMAIN_NAME)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--os-project-id &lt;auth-project-id&gt;</td>
<td>With password: project id to scope to with v3password: Project ID to scope to With v3token: Project ID to scope to With noauth: Project ID With v3tokenlessauth: Project ID to scope to With v3oidcpassword: Project ID to scope to With v3multifactor: Project ID to scope to With v3adfspassword: Project ID to scope to With v3totp: Project ID to scope to With v3oidcauthcode: Project ID to scope to With v3oidcclientcredentials: Project ID to scope to With gnocchi-noauth: Project ID With token: Project ID to scope to With v3oidcaccesstoken: Project ID to scope to (Env: OS_PROJECT_ID)</td>
</tr>
<tr>
<td>--os-project-name &lt;auth-project-name&gt;</td>
<td>With password: project name to scope to with v3password: Project name to scope to With v3token: Project name to scope to With v3tokenlessauth: Project name to scope to With v3oidcpassword: Project name to scope to With v3multifactor: Project name to scope to With v3adfspassword: Project name to scope to With v3totp: Project name to scope to With v3oidcauthcode: Project name to scope to With v3oidcclientcredentials: Project name to scope to With v3applicationcredential: Project name to scope to With v3oidcaccesstoken: Project name to scope to With v3samlpassword: Project name to scope to (Env: OS_PROJECT_NAME)</td>
</tr>
<tr>
<td>--os-project-domain-id &lt;auth-project-domain-id&gt;</td>
<td>With password: domain id containing project with v3password: Domain ID containing project With v3token: Domain ID containing project With v3tokenlessauth: Domain ID containing project With v3oidcpassword: Domain ID containing project With v3multifactor: Domain ID containing project With v3adfspassword: Domain ID containing project With v3totp: Domain ID containing project With v3oidcauthcode: Domain ID containing project With v3oidcclientcredentials: Domain ID containing project With v3applicationcredential: Domain ID containing project With v3oidcaccesstoken: Domain ID containing project With v3samlpassword: Domain ID containing project (Env: OS_PROJECT_DOMAIN_ID)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--os-project-domain-name &lt;auth-project-domain-name&gt;</code></td>
<td>With password: domain name containing project with v3password: Domain name containing project With v3token: Domain name containing project With v3tokenlessauth: Domain name containing project With v3oidcpassword: Domain name containing project With v3multifactor: Domain name containing project With v3adfspassword: Domain name containing project With v3totp: Domain name containing project With v3oidcauthcode: Domain name containing project With v3applicationcredential: Domain name containing project With v3oidclientcredentials: Domain name containing project With token: Domain name containing project With v3oidcaccessstoken: Domain name containing project With v3samlpassword: Domain name containing project (Env: OS_PROJECT_DOMAIN_NAME)</td>
</tr>
<tr>
<td><code>--os-default-domain-id &lt;auth-default-domain-id&gt;</code></td>
<td>With password: optional domain id to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication. With token: Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication. (Env: OS_DEFAULT_DOMAIN_ID)</td>
</tr>
<tr>
<td><code>--os-default-domain-name &lt;auth-default-domain-name&gt;</code></td>
<td>With password: optional domain name to use with v3 api and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication. With token: Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication. (Env: OS_DEFAULT_DOMAIN_NAME)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--os-user-domain-id &lt;auth-user-domain-id&gt;</code></td>
<td>With password: user’s domain id with v3password: User’s domain id With v3totp: User’s domain id With v3applicationcredential: User’s domain id (Env: OS_USER_DOMAIN_ID)</td>
</tr>
<tr>
<td><code>--os-user-domain-name &lt;auth-user-domain-name&gt;</code></td>
<td>With password: user’s domain name with v3password: User’s domain name With v3totp: User’s domain name With v3applicationcredential: User’s domain name (Env: OS_USER_DOMAIN_NAME)</td>
</tr>
<tr>
<td><code>--os-password &lt;auth-password&gt;</code></td>
<td>With password: user’s password with v3password: user’s password With v2password: Password to use With v3oidcpassword: Password With v3adfspassword: Password With v3applicationcredential: Password (Env: OS_PASSWORD)</td>
</tr>
<tr>
<td><code>--os-token &lt;auth-token&gt;</code></td>
<td>With v3token: token to authenticate with With v2token: Token To authenticate With With admin_token: The token that will always be used (Env: OS_TOKEN)</td>
</tr>
<tr>
<td><code>--os-user &lt;auth-user&gt;</code></td>
<td>With gnocchi-basic: user (env: os_user)</td>
</tr>
<tr>
<td><code>--os-identity-provider &lt;auth-identity-provider&gt;</code></td>
<td>With v3oidcpassword: identity provider’s name with v3applicationcredential: Identity Provider’s name With v3oidcauthcode: Identity Provider’s name With v3oidcclientcredentials: Identity Provider’s name With v3oidcpassword: Identity Provider’s name (Env: OS_IDENTITY_PROVIDER)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--os-client-id &lt;auth-client-id&gt;</code></td>
<td>With <code>v3oidcpassword</code>: OAuth 2.0 Client ID With <code>v3oidcauthcode</code>: OAuth 2.0 Client ID (Env: <code>OS_CLIENT_ID</code>)</td>
</tr>
<tr>
<td><code>--os-client-secret &lt;auth-client-secret&gt;</code></td>
<td>With <code>v3oidcpassword</code>: OAuth 2.0 Client Secret With <code>v3oidcauthcode</code>: OAuth 2.0 Client Secret (Env: <code>OS_CLIENT_SECRET</code>)</td>
</tr>
<tr>
<td><code>--os-openid-scope &lt;auth-openid-scope&gt;</code></td>
<td>With <code>v3oidcpassword</code>: OpenID Connect scope that is requested from authorization server. Note that the OpenID Connect specification states that &quot;openid&quot; must be always specified. With <code>v3oidcauthcode</code>: OpenID Connect scope that is requested from authorization server. Note that the OpenID Connect specification states that &quot;openid&quot; must be always specified. With <code>v3oidcclientcredentials</code>: OpenID Connect scope that is requested from authorization server. Note that the OpenID Connect specification states that &quot;openid&quot; must be always specified. (Env: <code>OS_OPENID_SCOPE</code>)</td>
</tr>
<tr>
<td><code>--os-access-token-endpoint &lt;auth-access-token-endpoint&gt;</code></td>
<td>With <code>v3oidcpassword</code>: OpenID Connect provider token Endpoint. Note that if a discovery document is being passed this option will override the endpoint provided by the server in the discovery document. With <code>v3oidcauthcode</code>: OpenID Connect Provider Token Endpoint. Note that if a discovery document is being passed this option will override the endpoint provided by the server in the discovery document. With <code>v3oidcclientcredentials</code>: OpenID Connect Provider Token Endpoint. Note that if a discovery document is being passed this option will override the endpoint provided by the server in the discovery document. (Env: <code>OS_ACCESS_TOKEN_ENDPOINT</code>)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--os-discovery-endpoint &lt;auth-discovery-endpoint&gt; </code></td>
<td>With v3oidcpassword: openid connect discovery document URL. The discovery document will be used to obtain the values of the access token endpoint and the authentication endpoint. This URL should look like <code>https://idp.example.org/.well-known/openid-configuration</code>. With v3oidcauthcode: OpenID Connect Discovery Document URL. The discovery document will be used to obtain the values of the access token endpoint and the authentication endpoint. This URL should look like <code>https://idp.example.org/.well-known/openid-configuration</code>. With v3oidcclientcredentials: OpenID Connect Discovery Document URL. The discovery document will be used to obtain the values of the access token endpoint and the authentication endpoint. This URL should look like <code>https://idp.example.org/.well-known/openid-configuration</code>. (Env: <code>OS_DISCOVERY_ENDPOINT</code>)</td>
</tr>
<tr>
<td><code>--os-access-token-type &lt;auth-access-token-type&gt; </code></td>
<td>With v3oidcpassword: oauth 2.0 authorization server Introspection token type, it is used to decide which type of token will be used when processing token introspection. Valid values are: &quot;access_token&quot; or &quot;id_token&quot;. With v3oidcauthcode: OAuth 2.0 Authorization Server Introspection token type, it is used to decide which type of token will be used when processing token introspection. Valid values are: &quot;access_token&quot; or &quot;id_token&quot;. With v3oidcclientcredentials: OAuth 2.0 Authorization Server Introspection token type, it is used to decide which type of token will be used when processing token introspection. Valid values are: &quot;access_token&quot; or &quot;id_token&quot;. (Env: <code>OS_ACCESS_TOKEN_TYPE</code>)</td>
</tr>
<tr>
<td><code>--os-auth-methods &lt;auth-auth-methods&gt; </code></td>
<td>With v3multifactor: methods to authenticate with. (Env: <code>OS_AUTH_METHODS</code>)</td>
</tr>
<tr>
<td><code>--os-roles &lt;auth-roles&gt; </code></td>
<td>With aodh-noauth: roles with gnocchi-noauth: roles (Env: <code>OS_ROLES</code>)</td>
</tr>
<tr>
<td><code>--os-aodh-endpoint &lt;auth-aodh-endpoint&gt; </code></td>
<td>With aodh-noauth: aodh endpoint (env: <code>OS_AODH_ENDPOINT</code>)</td>
</tr>
<tr>
<td><code>--os-identity-provider-url &lt;auth-identity-provider-url&gt; </code></td>
<td>With v3adfspassword: an identity provider url, where the SAML authentication request will be sent. With v3samlpassword: An Identity Provider URL, where the SAML2 authentication request will be sent. (Env: <code>OS_IDENTITY_PROVIDER_URL</code>)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--os-service-provider-endpoint &lt;auth-service-provider-endpoint&gt;</code></td>
<td>With v3adfspassword: service provider’s endpoint (env: OS_SERVICE_PROVIDER_ENDPOINT)</td>
</tr>
<tr>
<td><code>--os-service-provider-entity-id &lt;auth-service-provider-entity-id&gt;</code></td>
<td>With v3adfspassword: service provider’s saml entity id (Env: OS_SERVICE_PROVIDER_ENTITY_ID)</td>
</tr>
<tr>
<td><code>--os-passcode &lt;auth-passcode&gt;</code></td>
<td>With v3totp: user’s totp passcode (env: os_passcode)</td>
</tr>
<tr>
<td><code>--os-redirect-uri &lt;auth-redirect-uri&gt;</code></td>
<td>With v3oidcauthcode: openid connect redirect url (env: OS_REDIRECT_URI)</td>
</tr>
<tr>
<td><code>--os-code &lt;auth-code&gt;</code></td>
<td>With v3oidcauthcode: oauth 2.0 authorization code (Env: OS_CODE)</td>
</tr>
<tr>
<td><code>--os-application-credential-secret &lt;auth-application-credential-secret&gt;</code></td>
<td>With v3applicationcredential: application credential auth secret (Env: OS_APPLICATION_CREDENTIAL_SECRET)</td>
</tr>
<tr>
<td><code>--os-application-credential-id &lt;auth-application-credential-id&gt;</code></td>
<td>With v3applicationcredential: application credential ID (Env: OS_APPLICATION_CREDENTIAL_ID)</td>
</tr>
<tr>
<td><code>--os-application-credential-name &lt;auth-application-credential-name&gt;</code></td>
<td>With v3applicationcredential: application credential name (Env: OS_APPLICATION_CREDENTIAL_NAME)</td>
</tr>
<tr>
<td><code>--os-consumer-key &lt;auth-consumer-key&gt;</code></td>
<td>With v3oauth1: oauth consumer id/key (env: OS_CONSUMER_KEY)</td>
</tr>
<tr>
<td><code>--os-consumer-secret &lt;auth-consumer-secret&gt;</code></td>
<td>With v3oauth1: oauth consumer secret (env: OS_CONSUMER_SECRET)</td>
</tr>
<tr>
<td><code>--os-access-key &lt;auth-access-key&gt;</code></td>
<td>With v3oauth1: oauth access key (env: os_access_key)</td>
</tr>
<tr>
<td><code>--os-access-secret &lt;auth-access-secret&gt;</code></td>
<td>With v3oauth1: oauth access secret (env: OS_ACCESS_SECRET)</td>
</tr>
<tr>
<td><code>--os-access-token &lt;auth-access-token&gt;</code></td>
<td>With v3oidcaccesstoken: oauth 2.0 access token (env: OS_ACCESS_TOKEN)</td>
</tr>
</tbody>
</table>
CHAPTER 2. ACCESS

This chapter describes the commands under the `access` command.

2.1. ACCESS RULE DELETE

Delete access rule(s)

Usage:

```
openstack access rule delete [-h] <access-rule> [...]
```

Table 2.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access-rule&gt;</td>
<td>Access rule(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 2.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

2.2. ACCESS RULE LIST

List access rules

Usage:

```

Table 2.3. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>User whose access rules to list (name or id)</td>
</tr>
</tbody>
</table>
### Table 2.4. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 2.5. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 2.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 2.7. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
2.3. ACCESS RULE SHOW

Display access rule details

Usage:

openstack access rule show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [-m max-width <integer>] [--fit-width] [--print-empty]

<access-rule>

Table 2.8. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access-rule&gt;</td>
<td>Access rule to display (name or id)</td>
</tr>
</tbody>
</table>

Table 2.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 2.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 2.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 2.12. Shell formatter options
**Value** | **Summary**
---|---
--prefix PREFIX | Add a prefix to all variable names

Table 2.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 2.4. ACCESS TOKEN CREATE

Create an access token

**Usage:**

```
openstack access token create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] --consumer-key <consumer-key> --consumer-secret <consumer-secret> --request-key <request-key> --request-secret <request-secret> --verifier <verifier>
```

Table 2.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--consumer-key &lt;consumer-key&gt;</td>
<td>Consumer key (required)</td>
</tr>
<tr>
<td>--consumer-secret &lt;consumer-secret&gt;</td>
<td>Consumer secret (required)</td>
</tr>
<tr>
<td>--request-key &lt;request-key&gt;</td>
<td>Request token to exchange for access token (required)</td>
</tr>
<tr>
<td>--request-secret &lt;request-secret&gt;</td>
<td>Secret associated with &lt;request-key&gt; (required)</td>
</tr>
</tbody>
</table>
### Table 2.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--verifier &lt;verifier&gt;</td>
<td>Verifier associated with &lt;request-key&gt; (required)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 2.16. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 2.17. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 2.18. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 3. ACL

This chapter describes the commands under the acl command.

3.1. ACL DELETE

Delete ACLs for a secret or container as identified by its href.

Usage:

```
openstack acl delete [-h] URI
```

Table 3.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret or container.</td>
</tr>
</tbody>
</table>

Table 3.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

3.2. ACL GET

Retrieve ACLs for a secret or container by providing its href.

Usage:

```
```

Table 3.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret or container.</td>
</tr>
</tbody>
</table>

Table 3.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
Table 3.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 3.6. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 3.7. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 3.8. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

3.3. ACL SUBMIT

Submit ACL on a secret or container as identified by its href.
Usage:

openstack acl submit [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
[-quote {all, minimal, none, nonnumeric}]
[-noindent] [-max-width <integer>] [--fit-width]
[-print-empty] [--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--user [USERS]]
[--project-access | --no-project-access]
[--operation-type {read}]

Table 3.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret or container.</td>
</tr>
</tbody>
</table>

Table 3.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--user [USERS], -u [USERS]</td>
<td>Keystone userid(s) for acl.</td>
</tr>
<tr>
<td>--project-access</td>
<td>Flag to enable project access behavior.</td>
</tr>
<tr>
<td>--no-project-access</td>
<td>Flag to disable project access behavior.</td>
</tr>
<tr>
<td>--operation-type {read}, -o {read}</td>
<td>Type of barbican operation acl is set for</td>
</tr>
</tbody>
</table>

Table 3.11. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
Table 3.12. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 3.13. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 3.14. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

3.4. ACL USER ADD

Add ACL users to a secret or container as identified by its href.

Usage:

```bash
openstack acl user add [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [-q {all, minimal, none, nonnumeric}] [-noindent] [-m max-width <integer>] [-f-fit-width] [-p print-empty] [-s {sort-column} [-a ascending] [-d descending]] [-u {USERS}] [-p project-access] [-n no-project-access] [-o {read}] [--operation-type {read}] URI
```
Table 3.15. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret or container.</td>
</tr>
</tbody>
</table>

Table 3.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--user [USERS], -u [USERS]</td>
<td>Keystone userid(s) for acl.</td>
</tr>
<tr>
<td>--project-access</td>
<td>Flag to enable project access behavior.</td>
</tr>
<tr>
<td>--no-project-access</td>
<td>Flag to disable project access behavior.</td>
</tr>
<tr>
<td>--operation-type {read}, -o {read}</td>
<td>Type of barbican operation acl is set for</td>
</tr>
</tbody>
</table>

Table 3.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 3.18. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 3.19. JSON formatter options
### Value Summary

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table 3.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 3.5. ACL USER REMOVE

Remove ACL users from a secret or container as identified by its href.

**Usage:**

```
```

#### Table 3.21. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret or container.</td>
</tr>
</tbody>
</table>

#### Table 3.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--user [USERS], -u [USERS]</td>
<td>Keystone userid(s) for acl.</td>
</tr>
<tr>
<td>--project-access</td>
<td>Flag to enable project access behavior.</td>
</tr>
<tr>
<td>--no-project-access</td>
<td>Flag to disable project access behavior.</td>
</tr>
<tr>
<td>--operation-type {read}, -o {read}</td>
<td>Type of barbican operation acl is set for</td>
</tr>
</tbody>
</table>

**Table 3.23. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 3.24. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 3.25. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 3.26. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 4. ACTION

This chapter describes the commands under the action command.

4.1. ACTION DEFINITION CREATE

Create new action.

Usage:

openstack action definition create [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent] [-max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--marker [MARKER]]
    [--limit [LIMIT]]
    [--sort_keys [SORT_KEYS]]
    [--sort_dirs [SORT_DIRS]]
    [--filter FILTERS] [--public]
    [--namespace [NAMESPACE]]
    definition

Table 4.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Action definition file</td>
</tr>
</tbody>
</table>

Table 4.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list - --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
</tbody>
</table>
**Table 4.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 4.4. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 4.5. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 4.6. Table formatter options**
### 4.2. ACTION DEFINITION DEFINITION SHOW

Show action definition.

**Usage:**

```
openstack action definition definition show [-h] 
  [--namespace [NAMESPACE]]
  name
```

**Table 4.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Action name</td>
</tr>
</tbody>
</table>

**Table 4.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the action.</td>
</tr>
</tbody>
</table>

### 4.3. ACTION DEFINITION DELETE

Delete action.

**Usage:**

```
openstack action definition delete [-h] [--namespace [NAMESPACE]]
  action [action ...]
```

**Table 4.9. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action [action ...]</td>
<td>Action names to delete.</td>
</tr>
</tbody>
</table>
4.4. ACTION DEFINITION LIST

List all actions.

Usage:

```
```

Table 4.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Name or id of action(s).</td>
</tr>
</tbody>
</table>

Table 4.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays</td>
</tr>
<tr>
<td></td>
<td>list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by.</td>
</tr>
<tr>
<td></td>
<td>Default: created_at. Example: mistral execution-list -</td>
</tr>
<tr>
<td></td>
<td>-sort_keys=id,description</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the actions.</td>
</tr>
</tbody>
</table>

**Table 4.12. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 4.13. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 4.14. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 4.15. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 4.5. ACTION DEFINITION SHOW

Show specific action.

**Usage:**

```
openstack action definition show [-h]
  [-f {json,shell,table,value,yaml}]  
  [-c COLUMN] [-noindent]
  [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  [--namespace [NAMESPACE]]
  action
```

Table 4.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>Action (name or id)</td>
</tr>
</tbody>
</table>

Table 4.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the action within.</td>
</tr>
</tbody>
</table>

Table 4.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}; --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### Table 4.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 4.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 4.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 4.6. ACTION DEFINITION UPDATE

Update action.

**Usage:**

```
```
[--sort_dirs [SORT_DIRS]]
[--filter FILTERS] [--id ID]
[--public] [--namespace [NAMESPACE]]
definition

Table 4.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Action definition file</td>
</tr>
</tbody>
</table>

Table 4.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list - -sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--id ID</td>
<td>Action id.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag action will be marked as &quot;public&quot;.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the action.</td>
</tr>
</tbody>
</table>

Table 4.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### 4.25. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### 4.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### 4.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 4.7. ACTION EXECUTION DELETE

Delete action execution.

Usage:

```sh
openstack action execution delete [-h]
  action_execution
  [action_execution ...]
```
Table 4.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_execution</td>
<td>Id of action execution identifier(s).</td>
</tr>
</tbody>
</table>

Table 4.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

4.8. ACTION EXECUTION INPUT SHOW

Show Action execution input data.

Usage:

```
openstack action execution input show [-h] id
```

Table 4.30. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Action execution id.</td>
</tr>
</tbody>
</table>

Table 4.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

4.9. ACTION EXECUTION LIST

List all Action executions.

Usage:

```
```
Table 4.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>task_execution_id</td>
<td>Task execution id.</td>
</tr>
</tbody>
</table>

Table 4.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list - --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list -- sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--oldest</td>
<td>Display the executions starting from the oldest entries instead of the newest</td>
</tr>
</tbody>
</table>

Table 4.34. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
Table 4.35. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 4.36. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 4.37. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

4.10. ACTION EXECUTION OUTPUT SHOW

Show Action execution output data.

Usage:

openstack action execution output show [-h] id

Table 4.38. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Action execution id.</td>
</tr>
</tbody>
</table>
### Table 4.39. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 4.11. ACTION EXECUTION RUN

Create new Action execution or just run specific action.

**Usage:**

```
openstack action execution run [-h] [-f {json,shell,table,value,yaml}] 
  [-c COLUMN] [--noindent] 
  [--prefix PREFIX] 
  [--max-width <integer>] [--fit-width] 
  [--print-empty] [-s] [-run-sync] 
  [-t TARGET] [--namespace [NAMESPACE]] 
  name [input]
```

### Table 4.40. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Action name to execute.</td>
</tr>
<tr>
<td>input</td>
<td>Action input.</td>
</tr>
</tbody>
</table>

### Table 4.41. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-s, --save-result</td>
<td>Save the result into db.</td>
</tr>
<tr>
<td>--run-sync</td>
<td>Run the action synchronously.</td>
</tr>
<tr>
<td>-t TARGET, --target TARGET</td>
<td>Action will be executed on &lt;target&gt; executor.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the action(s).</td>
</tr>
</tbody>
</table>

### Table 4.42. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 4.43. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 4.44. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 4.45. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

4.12. ACTION EXECUTION SHOW

Show specific Action execution.

Usage:

```
```

Table 4.46. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action_execution</td>
<td>Action execution id.</td>
</tr>
</tbody>
</table>

Table 4.47. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 4.48. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 4.49. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 4.50. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 4.51. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
4.13. ACTION EXECUTION UPDATE

Update specific Action execution.

Usage:

```
openstack action execution update [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    [--state {PAUSED,RUNNING,SUCCESS,ERROR,CANCELLED}]
    [--output OUTPUT]
    id
```

Table 4.52. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Action execution id.</td>
</tr>
</tbody>
</table>

Table 4.53. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--state</td>
<td>Action execution state</td>
</tr>
<tr>
<td>{PAUSED,RUNNING,SUCCESS,ERROR,CANCELLED}</td>
<td></td>
</tr>
<tr>
<td>--output OUTPUT</td>
<td>Action execution output</td>
</tr>
</tbody>
</table>

Table 4.54. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 4.55. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 4.56. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 4.57. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 5. ADDRESS

This chapter describes the commands under the `address` command.

5.1. ADDRESS GROUP CREATE

Create a new Address Group

Usage:

```
```

Table 5.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New address group name</td>
</tr>
</tbody>
</table>

Table 5.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New address group description</td>
</tr>
<tr>
<td>--address &lt;ip-address&gt;</td>
<td>Ip address or cidr (repeat option to set multiple addresses)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner's project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 5.3. Output formatter options
### 5.2. ADDRESS GROUP DELETE

Delete address group(s)

**Usage:**

```bash
openstack address group delete [-h]
<address-group> [<address-group> ...]
```

**Table 5.7. Positional arguments**
5.3. ADDRESS GROUP LIST

List address groups

Usage:

```
```

Table 5.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-group&gt;</td>
<td>Address group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 5.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List only address groups of given name in output</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List address groups according to their project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 5.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### Table 5.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 5.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 5.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 5.4. ADDRESS GROUP SET

Set address group properties

#### Usage:

```
openstack address group set [-h] [--name <name>] [--description <description>]
```
Table 5.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-group&gt;</td>
<td>Address group to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 5.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set address group name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set address group description</td>
</tr>
<tr>
<td>--address &lt;ip-address&gt;</td>
<td>Ip address or cidr (repeat option to set multiple addresses)</td>
</tr>
</tbody>
</table>

5.5. ADDRESS GROUP SHOW

Display address group details

Usage:

```bash
openstack address group show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] <address-group>
```

Table 5.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-group&gt;</td>
<td>Address group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 5.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 5.18. Output formatter options
Table 5.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 5.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 5.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 5.6. ADDRESS GROUP UNSET

Unset address group properties

**Usage:**

```
openstack address group unset [-h] [--address <ip-address>] <address-group>
```

Table 5.22. Positional arguments
5.7. ADDRESS SCOPE CREATE

Create a new Address Scope

Usage:

```
```

Table 5.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-group&gt;</td>
<td>Address group to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 5.24. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New address scope name</td>
</tr>
</tbody>
</table>

Table 5.25. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--ip-version {4,6}</td>
<td>Ip version (default is 4)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner's project (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--share</td>
<td>Share the address scope between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>Do not share the address scope between projects (default)</td>
</tr>
</tbody>
</table>

Table 5.26. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 5.27. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 5.28. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 5.29. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
5.8. ADDRESS SCOPE DELETE

Delete address scope(s)

Usage:

    openstack address scope delete [-h]  
        <address-scope> [ <address-scope> ... ]

Table 5.30. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-scope&gt;</td>
<td>Address scope(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 5.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

5.9. ADDRESS SCOPE LIST

List address scopes

Usage:

    openstack address scope list [-h] [-f {csv, json, table, value, yaml}]
    [-c COLUMN]  
        [--quote {all, minimal, none, nonnumeric}]  
        [--noindent] [-max-width <integer>]  
        [--fit-width] [--print-empty]  
        [--sort-column SORT_COLUMN]  
        [--sort-ascending | --sort-descending]  
        [--name <name>]  
        [--ip-version <ip-version>]  
        [--project <project>]  
        [--project-domain <project-domain>]  
        [--share | --no-share]

Table 5.32. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List only address scopes of given name in output</td>
</tr>
<tr>
<td>--ip-version &lt;ip-version&gt;</td>
<td>List address scopes of given ip version networks (4 or 6)</td>
</tr>
</tbody>
</table>
List address scopes according to their project (name or ID)

Domain the project belongs to (name or id). This can be used in case collisions between project names exist.

List address scopes shared between projects

List address scopes not shared between projects

### Table 5.33. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--project &lt;project&gt;</td>
<td>List address scopes according to their project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--share</td>
<td>List address scopes shared between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>List address scopes not shared between projects</td>
</tr>
</tbody>
</table>

The output format, defaults to table

Specify the column(s) to include, can be repeated to show multiple columns

Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

Sort the column(s) in ascending order

Sort the column(s) in descending order

### Table 5.34. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 5.35. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 5.36. Table formatter options
5.10. ADDRESS SCOPE SET

Set address scope properties

Usage:

```
openstack address scope set [-h] [--name <name>] [--share | --no-share] <address-scope>
```

Table 5.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-scope&gt;</td>
<td>Address scope to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 5.38. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set address scope name</td>
</tr>
<tr>
<td>--share</td>
<td>Share the address scope between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>Do not share the address scope between projects</td>
</tr>
</tbody>
</table>

5.11. ADDRESS SCOPE SHOW

Display address scope details

Usage:

```
openstack address scope show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-n noindent] [-p prefix PREFIX] [-s max-width <integer>] [-w fit-width]
```
Table 5.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address-scope&gt;</td>
<td>Address scope to display (name or id)</td>
</tr>
</tbody>
</table>

Table 5.40. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 5.41. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 5.42. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 5.43. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 5.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 6. AGGREGATE

This chapter describes the commands under the `aggregate` command.

6.1. AGGREGATE ADD HOST

Add host to aggregate

Usage:
```
```

Table 6.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate (name or id)</td>
</tr>
<tr>
<td>&lt;host&gt;</td>
<td>Host to add to &lt;aggregate&gt;</td>
</tr>
</tbody>
</table>

Table 6.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 6.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 6.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
6.2. AGGREGATE CACHE IMAGE

Request image caching for aggregate

Usage:

```
openstack aggregate cache image [-h] <aggregate> <image> [<image> ...]
```

Table 6.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate (name or id)</td>
</tr>
<tr>
<td>&lt;image&gt;</td>
<td>Image id to request caching for aggregate (name or id). may be specified multiple times.</td>
</tr>
</tbody>
</table>

Table 6.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

6.3. AGGREGATE CREATE

Create a new aggregate

Usage:

Table 6.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New aggregate name</td>
</tr>
</tbody>
</table>

Table 6.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--zone &lt;availability-zone&gt;</td>
<td>Availability zone name</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to add to this aggregate (repeat option to set multiple properties)</td>
</tr>
</tbody>
</table>

Table 6.11. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 6.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 6.13. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 6.14. Table formatter options
## 6.4. AGGREGATE DELETE

Delete existing aggregate(s)

**Usage:**

```
openstack aggregate delete [-h] <aggregate> [<aggregate> ...]
```

**Table 6.15. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 6.16. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

## 6.5. AGGREGATE LIST

List all aggregates

**Usage:**

```
```

**Table 6.17. Command arguments**
### Table 6.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 6.19. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 6.20. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 6.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
6.6. AGGREGATE REMOVE HOST

Remove host from aggregate

Usage:


Table 6.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate (name or id)</td>
</tr>
<tr>
<td>&lt;host&gt;</td>
<td>Host to remove from &lt;aggregate&gt;</td>
</tr>
</tbody>
</table>

Table 6.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 6.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 6.25. JSON formatter options
Table 6.26. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 6.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

6.7. AGGREGATE SET

Set aggregate properties

Usage:

```
openstack aggregate set [-h] [--name <name>] [--zone <availability-zone>] [--property <key=value>] [--no-property] <aggregate>
```

Table 6.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 6.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### 6.8. AGGREGATE SHOW

Display aggregate details

**Usage:**

```
```

**Table 6.30. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 6.31. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 6.32. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

---

**Value**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set aggregate name</td>
</tr>
<tr>
<td>--zone &lt;availability-zone&gt;</td>
<td>Set availability zone name</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to set on &lt;aggregate&gt; (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--no-property</td>
<td>Remove all properties from &lt;aggregate&gt; (specify both --property and --no-property to overwrite the current properties)</td>
</tr>
</tbody>
</table>
Table 6.33. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 6.34. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 6.35. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

6.9. AGGREGATE UNSET

Unset aggregate properties

Usage:

```
openstack aggregate unset [-h] [--property <key>] <aggregate>
```

Table 6.36. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;aggregate&gt;</td>
<td>Aggregate to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 6.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>--property <code>&lt;key&gt;</code></td>
<td>Property to remove from aggregate (repeat option</td>
</tr>
<tr>
<td></td>
<td>to remove multiple properties)</td>
</tr>
</tbody>
</table>
CHAPTER 7. ALARM

This chapter describes the commands under the `alarm` command.

7.1. ALARM CREATE

Create an alarm

**Usage:**

```
```

**Table 7.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;NAME&gt;</td>
<td>Name of the alarm</td>
</tr>
<tr>
<td>-t &lt;TYPE&gt;, --type &lt;TYPE&gt;</td>
<td>Type of alarm, should be one of: event, composite, \</td>
</tr>
<tr>
<td></td>
<td>threshold, gnocchi_resources_threshold, \</td>
</tr>
<tr>
<td></td>
<td>gnocchi_aggregation_by_metrics_threshold, \</td>
</tr>
<tr>
<td></td>
<td>gnocchi_aggregation_by_resources_threshold, \</td>
</tr>
<tr>
<td></td>
<td>loadbalancer_member_health.</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--project-id &lt;PROJECT_ID&gt;</td>
<td>Project to associate with alarm (configurable by admin users only)</td>
</tr>
<tr>
<td>--user-id &lt;USER_ID&gt;</td>
<td>User to associate with alarm (configurable by admin users only)</td>
</tr>
<tr>
<td>--description &lt;DESCRIPTION&gt;</td>
<td>Free text description of the alarm</td>
</tr>
<tr>
<td>--state &lt;STATE&gt;</td>
<td>State of the alarm, one of: [ok, alarm, insufficient data]</td>
</tr>
<tr>
<td>--severity &lt;SEVERITY&gt;</td>
<td>Severity of the alarm, one of: [low, moderate, critical]</td>
</tr>
<tr>
<td>--enabled {True</td>
<td>False}</td>
</tr>
<tr>
<td>--alarm-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to alarm. may be used multiple times</td>
</tr>
<tr>
<td>--ok-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to ok. may be used multiple times</td>
</tr>
<tr>
<td>--insufficient-data-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to insufficient data. May be used multiple times</td>
</tr>
<tr>
<td>--time-constraint &lt;Time Constraint&gt;</td>
<td>Only evaluate the alarm if the time at evaluation is within this time constraint. Start point(s) of the constraint are specified with a cron expression, whereas its duration is given in seconds. Can be specified multiple times for multiple time constraints, format is: name=&lt;CONSTRAINT_NAME&gt;;start=&lt;CRON&gt;;duration=&lt;SECONDS&gt;[:description=&lt;DESCRIPTION&gt;][:timezone=&lt;IANA Timezone&gt;]]</td>
</tr>
<tr>
<td>--repeat-actions {True</td>
<td>False}</td>
</tr>
</tbody>
</table>

**Table 7.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 7.3. JSON formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 7.4. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 7.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 7.6. common alarm rules**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
| --query <QUERY>       | For alarms of type threshold or event: key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean. For alarms of type gnocchi_aggregation_by_resources_threshold: need to specify a complex query json string, like: {"and": [{"=": {"ended_at": null}, …​
| --comparison-operator <OPERATOR> | Operator to compare with, one of: [lt, le, eq, ne, ge, gt] |
| --evaluation-periods <EVAL_PERIODS> | Number of periods to evaluate over |
| --threshold <THRESHOLD> | Threshold to evaluate against. |

**Table 7.7. event alarm**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--event-type &lt;EVENT_TYPE&gt;</td>
<td>Event type to evaluate against</td>
</tr>
</tbody>
</table>
Table 7.8. threshold alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m &lt;METER NAME&gt;, --meter-name &lt;METER NAME&gt;</td>
<td>Meter to evaluate against</td>
</tr>
<tr>
<td>--period &lt;PERIOD&gt;</td>
<td>Length of each period (seconds) to evaluate over.</td>
</tr>
<tr>
<td>--statistic &lt;STATISTIC&gt;</td>
<td>Statistic to evaluate, one of: [max, min, avg, sum, count]</td>
</tr>
</tbody>
</table>

Table 7.9. common gnocchi alarm rules

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--granularity &lt;GRANULARITY&gt;</td>
<td>The time range in seconds over which to query.</td>
</tr>
<tr>
<td>--aggregation-method &lt;AGGR_METHOD&gt;</td>
<td>The aggregation_method to compare to the threshold.</td>
</tr>
<tr>
<td>--metric &lt;METRIC&gt;, --metrics &lt;METRIC&gt;</td>
<td>The metric id or name depending of the alarm type</td>
</tr>
</tbody>
</table>

Table 7.10. gnocchi resource threshold alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--resource-type &lt;RESOURCE_TYPE&gt;</td>
<td>The type of resource.</td>
</tr>
<tr>
<td>--resource-id &lt;RESOURCE_ID&gt;</td>
<td>The id of a resource.</td>
</tr>
</tbody>
</table>

Table 7.11. composite alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--composite-rule &lt;COMPOSITE_RULE&gt;</td>
<td>Composite threshold rule with json format, the form can be a nested dict which combine threshold/gnocchi rules by &quot;and&quot;, &quot;or&quot;. For example, the form is like: {&quot;or&quot;:[RULE1, RULE2, {&quot;and&quot;: [RULE3, RULE4]}]}. The RULEx can be basic threshold rules but must include a &quot;type&quot; field, like this: {&quot;threshold&quot;: 0.8,&quot;meter_name&quot;:&quot;cpu_util&quot;,&quot;type&quot;:&quot;threshold&quot;}</td>
</tr>
</tbody>
</table>

Table 7.12. loadbalancer member health alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
**7.2. ALARM DELETE**

Delete an alarm

**Usage:**

```
openstack alarm delete [-h] [--name <NAME>] [<ALARM ID or NAME>]
```

<table>
<thead>
<tr>
<th>Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ALARM ID or NAME&gt;</td>
</tr>
</tbody>
</table>

Id or name of an alarm.

<table>
<thead>
<tr>
<th>Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--name &lt;NAME&gt;</td>
</tr>
</tbody>
</table>

Name of the alarm

**7.3. ALARM-HISTORY SEARCH**

Show history for all alarms based on query

**Usage:**

```
openstack alarm-history search [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
```

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
### Table 7.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--query QUERY</code></td>
<td>Rich query supported by aodh, e.g. project_id!=my-id</td>
</tr>
<tr>
<td></td>
<td>user_id=foo or user_id=bar</td>
</tr>
</tbody>
</table>

### Table 7.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 7.17. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 7.18. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 7.19. Table formatter options
### 7.4. ALARM-HISTORY SHOW

Show history for an alarm

**Usage:**

```bash
```

#### Table 7.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;alarm-id&gt;</td>
<td>Id of an alarm</td>
</tr>
</tbody>
</table>

#### Table 7.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of resources to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing. Return the next results after this value, the supported marker is event_id.</td>
</tr>
<tr>
<td>--sort &lt;SORT_KEY:SORT_DIR&gt;</td>
<td>Sort of resource attribute. e.g. timestamp:desc</td>
</tr>
</tbody>
</table>
Table 7.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv, json, table, value, yaml], --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 7.23. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 7.24. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 7.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

7.5. ALARM LIST

List alarms
Usage:

```
openstack alarm list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
   [--quote {all, minimal, none, nonnumeric}] [-noindent] [-max-width <integer>]
   [--fit-width] [-print-empty] [---sort-column SORT_COLUMN]
   [-sort-ascending | --sort-descending]
   [--query QUERY | --filter <KEY1=VALUE1;KEY2=VALUE2...>]
   [--limit <LIMIT>] [---marker <MARKER>]
   [-sort <SORT_KEY:SORT_DIR>]
```

Table 7.26. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--query QUERY</td>
<td>Rich query supported by aodh, e.g. project_id!=my-id</td>
</tr>
<tr>
<td></td>
<td>user_id=foo or user_id=bar</td>
</tr>
<tr>
<td>--filter &lt;KEY1=VALUE1;KEY2=VALUE2...&gt;</td>
<td>Filter parameters to apply on returned alarms.</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of resources to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing. return the next results after this value, the supported marker is alarm_id.</td>
</tr>
<tr>
<td>--sort &lt;SORT_KEY:SORT_DIR&gt;</td>
<td>Sort of resource attribute, e.g. name:asc</td>
</tr>
</tbody>
</table>

Table 7.27. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
### Table 7.28. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 7.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 7.30. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also</td>
</tr>
<tr>
<td></td>
<td>use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 7.6. ALARM QUOTA SET

Command base class for displaying data about a single object.

**Usage:**

```
```

### Table 7.31. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>project</td>
<td>Project id.</td>
</tr>
</tbody>
</table>

### Table 7.32. Command arguments
## 7.7. ALARM QUOTA SHOW

Show quota for a project
Usage:


Table 7.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project PROJECT</td>
<td>Project id. if not specified, get quota for the current project.</td>
</tr>
</tbody>
</table>

Table 7.38. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 7.39. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 7.40. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 7.41. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 7.8. ALARM SHOW

Show an alarm

**Usage:**

```bash
openstack alarm show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] [--name <NAME>] [<ALARM ID or NAME>]
```

**Table 7.42. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ALARM ID or NAME&gt;</td>
<td>Id or name of an alarm.</td>
</tr>
</tbody>
</table>

**Table 7.43. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;NAME&gt;</td>
<td>Name of the alarm</td>
</tr>
</tbody>
</table>

**Table 7.44. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 7.45. JSON formatter options**
Table 7.46. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 7.47. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

7.9. ALARM STATE GET

Get state of an alarm

Usage:

```
openstack alarm state get [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] [--name <NAME>] [<ALARM ID or NAME>]
```

Table 7.48. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ALARM ID or NAME&gt;</td>
<td>Id or name of an alarm.</td>
</tr>
</tbody>
</table>

Table 7.49. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### Table 7.50. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name &lt;NAME&gt;</td>
<td>Name of the alarm</td>
</tr>
</tbody>
</table>

#### Value

- `-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}`
  The output format, defaults to table
- `-c COLUMN, --column COLUMN`
  Specify the column(s) to include, can be repeated to show multiple columns

### Table 7.51. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 7.52. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 7.53. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 7.10. ALARM STATE SET

Set state of an alarm

**Usage:**

Table 7.54. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ALARM ID or NAME&gt;</td>
<td>Id or name of an alarm.</td>
</tr>
</tbody>
</table>

Table 7.55. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;NAME&gt;</td>
<td>Name of the alarm</td>
</tr>
<tr>
<td>--state &lt;STATE&gt;</td>
<td>State of the alarm, one of: [ok, alarm, insufficient data]</td>
</tr>
</tbody>
</table>

Table 7.56. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 7.57. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 7.58. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 7.59. Table formatter options
7.11. ALARM UPDATE

Update an alarm

Usage:

```
```

Table 7.60. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ALARM ID or NAME&gt;</td>
<td>Id or name of an alarm.</td>
</tr>
</tbody>
</table>

Table 7.61. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;NAME&gt;</td>
<td>Name of the alarm</td>
</tr>
<tr>
<td>-t &lt;TYPE&gt;, --type &lt;TYPE&gt;</td>
<td>Type of alarm, should be one of: event, composite, threshold, gnocchi_resources_threshold, gnocchi_aggregation_by_metrics_threshold, gnocchi_aggregation_by_resources_threshold, loadbalancer_member_health.</td>
</tr>
<tr>
<td>--project-id &lt;PROJECT_ID&gt;</td>
<td>Project to associate with alarm (configurable by admin users only)</td>
</tr>
<tr>
<td>--user-id &lt;USER_ID&gt;</td>
<td>User to associate with alarm (configurable by admin users only)</td>
</tr>
<tr>
<td>--description &lt;DESCRIPTION&gt;</td>
<td>Free text description of the alarm</td>
</tr>
<tr>
<td>--state &lt;STATE&gt;</td>
<td>State of the alarm, one of: [ok, alarm, insufficient data]</td>
</tr>
<tr>
<td>--severity &lt;SEVERITY&gt;</td>
<td>Severity of the alarm, one of: [low, moderate, critical]</td>
</tr>
<tr>
<td>--enabled {True</td>
<td>False}</td>
</tr>
<tr>
<td>--alarm-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to alarm. may be used multiple times</td>
</tr>
<tr>
<td>--ok-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to ok. may be used multiple times</td>
</tr>
<tr>
<td>--insufficient-data-action &lt;Webhook URL&gt;</td>
<td>Url to invoke when state transitions to insufficient data. May be used multiple times</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--time-constraint &lt;Time Constraint&gt;</td>
<td>Only evaluate the alarm if the time at evaluation is within this time constraint. Start point(s) of the constraint are specified with a cron expression, whereas its duration is given in seconds. Can be specified multiple times for multiple time constraints, format is: name=&lt;CONSTRAINT_NAME&gt;;start=&lt;CRON&gt;;duration=&lt;SECONDS&gt;;[description=&lt;DESCRIPTION&gt;;timezone=&lt;IANA Timezone&gt;]</td>
</tr>
<tr>
<td>--repeat-actions {True</td>
<td>False}</td>
</tr>
</tbody>
</table>

Table 7.62. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 7.63. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 7.64. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 7.65. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
Value | Summary
--- | ---
--fit-width | Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty | Print empty table if there is no data to show.

Table 7.66. common alarm rules

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--query &lt;QUERY&gt;</td>
<td>For alarms of type threshold or event: key[op]data_type::value; list. data_type is optional, but if supplied must be string, integer, float, or boolean. For alarms of type gnocchi_aggregation_by_resources_threshold: need to specify a complex query json string, like: {&quot;and&quot;: [&quot;=&quot;: {&quot;ended_at&quot;: null}], ...}</td>
</tr>
<tr>
<td>--comparison-operator &lt;OPERATOR&gt;</td>
<td>Operator to compare with, one of: [lt, le, eq, ne, ge, gt]</td>
</tr>
<tr>
<td>--evaluation-periods &lt;EVAL_PERIODS&gt;</td>
<td>Number of periods to evaluate over</td>
</tr>
<tr>
<td>--threshold &lt;THRESHOLD&gt;</td>
<td>Threshold to evaluate against</td>
</tr>
</tbody>
</table>

Table 7.67. event alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--event-type &lt;EVENT_TYPE&gt;</td>
<td>Event type to evaluate against</td>
</tr>
</tbody>
</table>

Table 7.68. threshold alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-m &lt;METER NAME&gt;, --meter-name &lt;METER NAME&gt;</td>
<td>Meter to evaluate against</td>
</tr>
<tr>
<td>--period &lt;PERIOD&gt;</td>
<td>Length of each period (seconds) to evaluate over</td>
</tr>
<tr>
<td>--statistic &lt;STATISTIC&gt;</td>
<td>Statistic to evaluate, one of: [max, min, avg, sum, count]</td>
</tr>
</tbody>
</table>

Table 7.69. common gnocchi alarm rules
--granularity <GRANULARITY>
The time range in seconds over which to query.

--aggregation-method <AGGR_METHOD>
The aggregation_method to compare to the threshold.

--metric <METRIC>, --metrics <METRIC>
The metric id or name depending of the alarm type.

Table 7.70. gnocchi resource threshold alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--resource-type &lt;RESOURCE_TYPE&gt;</td>
<td>The type of resource.</td>
</tr>
<tr>
<td>--resource-id &lt;RESOURCE_ID&gt;</td>
<td>The id of a resource.</td>
</tr>
</tbody>
</table>

Table 7.71. composite alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--composite-rule &lt;COMPOSITE_RULE&gt;</td>
<td>Composite threshold rule with json format, the form can be a nested dict which combine threshold/gnocchi rules by &quot;and&quot;, &quot;or&quot;. For example, the form is like: {&quot;or&quot;:[RULE1, RULE2, {&quot;and&quot;:[RULE3, RULE4]}}. The RULEx can be basic threshold rules but must include a type field, like this: {&quot;threshold&quot;: 0.8,&quot;meter_name&quot;:&quot;cpu_util&quot;,&quot;type&quot;:&quot;threshold&quot;}</td>
</tr>
</tbody>
</table>

Table 7.72. loadbalancer member health alarm

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--stack-id &lt;STACK_NAME_OR_ID&gt;</td>
<td>Name or id of the root / top level heat stack containing the loadbalancer pool and members. An update will be triggered on the root Stack if an unhealthy member is detected in the loadbalancer pool.</td>
</tr>
<tr>
<td>--pool-id &lt;LOADBALANCER_POOL_NAME_OR_ID&gt;</td>
<td>Name or id of the loadbalancer pool for which the health of each member will be evaluated.</td>
</tr>
<tr>
<td>--autoscaling-group-id &lt;AUTOSCALING_GROUP_NAME_OR_ID&gt;</td>
<td>Id of the heat autoscaling group that contains the loadbalancer members. Unhealthy members will be marked as such before an update is triggered on the root stack.</td>
</tr>
</tbody>
</table>
CHAPTER 8. ALARMING

This chapter describes the commands under the `alarming` command.

8.1. ALARMING CAPABILITIES LIST

List capabilities of alarming service

**Usage:**

```bash
openstack alarming capabilities list [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
```

**Table 8.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 8.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 8.3. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 8.4. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 8.5. Table formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 9. APPLICATION

This chapter describes the commands under the `application` command.

9.1. APPLICATION CREDENTIAL CREATE

Create new application credential

Usage:

```
openstack application credential create [-h]
    [-f {json,shell,table,value,yaml}]
    [-C COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--secret <secret>]
    [--role <role>]
    [--expiration <expiration>]
    [--description <description>]
    [--unrestricted] [--restricted]
    [--access-rules <access-rules>]
<name>
```

Table 9.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the application credential</td>
</tr>
</tbody>
</table>

Table 9.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--secret &lt;secret&gt;</td>
<td>Secret to use for authentication (if not provided, one will be generated)</td>
</tr>
<tr>
<td>--role &lt;role&gt;</td>
<td>Roles to authorize (name or id) (repeat option to set multiple values)</td>
</tr>
<tr>
<td>--expiration &lt;expiration&gt;</td>
<td>Sets an expiration date for the application credential, format of YYYY-mm-ddTHH:MM:SS (if not provided, the application credential will not expire)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Application credential description</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--unrestricted</td>
<td>Enable application credential to create and delete other application credentials and trusts (this is potentially dangerous behavior and is disabled by default)</td>
</tr>
<tr>
<td>--restricted</td>
<td>Prohibit application credential from creating and deleting other application credentials and trusts (this is the default behavior)</td>
</tr>
</tbody>
</table>
| --access-rules <access-rules> | Either a string or file path containing a json-formatted list of access rules, each containing a request method, path, and service, for example ```
```  ```
       ``` | |

---

### Table 9.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

---

### Table 9.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

---

### Table 9.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

---

### Table 9.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
9.2. APPLICATION CREDENTIAL DELETE

Delete application credentials(s)

Usage:

openstack application credential delete [-h]
<application-credential>
[<application-credential> ...]

Table 9.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;application-credential&gt;</td>
<td>Application credentials(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 9.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

9.3. APPLICATION CREDENTIAL LIST

List application credentials

Usage:

openstack application credential list [-h]
  [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [-quote {all, minimal, none, nonnumeric}]
  [-noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--user <user>]
  [--user-domain <user-domain>]

Table 9.9. Command arguments
## Table 9.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--user &lt;user&gt;</code></td>
<td>User whose application credentials to list (name or ID)</td>
</tr>
<tr>
<td><code>--user-domain &lt;user-domain&gt;</code></td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

## Table 9.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

## Table 9.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

## Table 9.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
9.4. APPLICATION CREDENTIAL SHOW

Display application credential details

Usage:

openstack application credential show [-h] 
  [-f {json,shell,table,value,yaml}] 
  [-c COLUMN] [--noindent] 
  [--prefix PREFIX] 
  [--max-width <integer>] 
  [--fit-width] [--print-empty]  
  <application-credential>

Table 9.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;application-credential&gt;</td>
<td>Application credential to display (name or id)</td>
</tr>
</tbody>
</table>

Table 9.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 9.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>[json,shell,table,value,yaml]</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

Print empty table if there is no data to show.
Table 9.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 9.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 9.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
This chapter describes the commands under the `availability` command.

### 10.1. AVAILABILITY ZONE LIST

List availability zones and their status

**Usage:**

```
```

**Table 10.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--compute</td>
<td>List compute availability zones</td>
</tr>
<tr>
<td>--network</td>
<td>List network availability zones</td>
</tr>
<tr>
<td>--volume</td>
<td>List volume availability zones</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

**Table 10.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specifying the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specifying the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 10.3. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 10.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 10.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 11. BAREMETAL

This chapter describes the commands under the baremetal command.

11.1. BAREMETAL ALLOCATION CREATE

Create a new baremetal allocation.

Usage:

openstack baremetal allocation create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--resource-class RESOURCE_CLASS]
    [--trait TRAITS]
    [--candidate-node CANDIDATE_NODES]
    [--name NAME] [--uuid UUID]
    [--owner OWNER]
    [--extra <key=value>]
    [--wait [<time-out>]]
    [--node NODE]

Table 11.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-class RESOURCE_CLASS</td>
<td>Resource class to request.</td>
</tr>
<tr>
<td>--trait TRAITS</td>
<td>A trait to request. can be specified multiple times.</td>
</tr>
<tr>
<td>--candidate-node CANDIDATE_NODES</td>
<td>A candidate node for this allocation. can be specified multiple times. If at least one is specified, only the provided candidate nodes are considered for the allocation.</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Unique name of the allocation.</td>
</tr>
<tr>
<td>--uuid UUID</td>
<td>Uuid of the allocation.</td>
</tr>
<tr>
<td>--owner OWNER</td>
<td>Owner of the allocation.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
</tbody>
</table>
--wait [<time-out>] Wait for the new allocation to become active. An error is returned if allocation fails and --wait is used. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.

--node NODE Backfill this allocation from the provided node that has already been deployed. Bypasses the normal allocation process.

Table 11.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
11.2. BAREMETAL ALLOCATION DELETE

Unregister baremetal allocation(s).

Usage:

```
openstack baremetal allocation delete [-h] <allocation> [<allocation> ...]
```

Table 11.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;allocation&gt;</td>
<td>allocations(s) to delete (name or uuid).</td>
</tr>
</tbody>
</table>

Table 11.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.3. BAREMETAL ALLOCATION LIST

List baremetal allocations.

Usage:

```
```

Table 11.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of allocations to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;allocation&gt;</td>
<td>Allocation uuid (for example, of the last allocation in the list from a previous request). Returns the list of allocations after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified allocation fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--node &lt;node&gt;</td>
<td>Only list allocations of this node (name or uuid).</td>
</tr>
<tr>
<td>--resource-class &lt;resource_class&gt;</td>
<td>Only list allocations with this resource class.</td>
</tr>
<tr>
<td>--state &lt;state&gt;</td>
<td>Only list allocations in this state.</td>
</tr>
<tr>
<td>--owner &lt;owner&gt;</td>
<td>Only list allocations with this owner.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the allocations.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [... ]</td>
<td>One or more allocation fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>

Table 11.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 11.10. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.4. BAREMETAL ALLOCATION SET

Set baremetal allocation properties.

Usage:

```
openstack baremetal allocation set [-h] [--name <name>] [--extra <key=value>] <allocation>
```

Table 11.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;allocation&gt;</td>
<td>Name or uuid of the allocation</td>
</tr>
</tbody>
</table>

Table 11.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### 11.5. BAREMETAL ALLOCATION SHOW

Show baremetal allocation details.

**Usage:**

```
openstack baremetal allocation show [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--fields <field> [<field> ...]]
{id}
```

**Table 11.15. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Uuid or name of the allocation</td>
</tr>
</tbody>
</table>

**Table 11.16. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more allocation fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

**Table 11.17. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 11.18. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.19. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.6. BAREMETAL ALLOCATION UNSET

Unset baremetal allocation properties.

Usage:

```
openstack baremetal allocation unset [-h] [--name] [--extra <key>] <allocation>
```

Table 11.21. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;allocation&gt;</td>
<td>Name or uuid of the allocation</td>
</tr>
</tbody>
</table>

Table 11.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
11.7. BAREMETAL CHASSIS CREATE

Create a new chassis.

Usage:

```
```

Table 11.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the chassis</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Unique uuid of the chassis</td>
</tr>
</tbody>
</table>

Table 11.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.25. JSON formatter options
## 11.8. BAREMETAL CHASSIS DELETE

Delete a chassis.

**Usage:**

```bash
openstack baremetal chassis delete [-h] <chassis> [<chassis> ...]
```

### Table 11.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;chassis&gt;</td>
<td>Uuids of chassis to delete</td>
</tr>
</tbody>
</table>

### Table 11.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

## 11.9. BAREMETAL CHASSIS LIST
List the chassis.

Usage:

```
```

Table 11.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [ &lt;field&gt; ...]</td>
<td>One or more chassis fields. only these fields will be fetched from the server. Cannot be used when --long is specified.</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of chassis to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the chassis</td>
</tr>
<tr>
<td>--marker &lt;chassis&gt;</td>
<td>Chassis uuid (for example, of the last chassis in the list from a previous request). Returns the list of chassis after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[&lt;direction&gt;]</td>
<td>Sort output by specified chassis fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
</tbody>
</table>

Table 11.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

#### Table 11.32. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

#### Table 11.33. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table 11.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.10. BAREMETAL CHASSIS SET

Set chassis properties.

**Usage:**

```
openstack baremetal chassis set [-h] [--description <description>] [--extra <key=value>] <chassis>
```
### Table 11.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;chassis&gt;</code></td>
<td>Uuid of the chassis</td>
</tr>
</tbody>
</table>

### Table 11.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>Set the description of the chassis</td>
</tr>
<tr>
<td><code>--extra &lt;key=value&gt;</code></td>
<td>Extra to set on this chassis (repeat option to set multiple extras)</td>
</tr>
</tbody>
</table>

### 11.11. BAREMETAL CHASSIS SHOW

Show chassis details.

**Usage:**

```
openstack baremetal chassis show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]

  <chassis>
```

### Table 11.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;chassis&gt;</code></td>
<td>Uuid of the chassis</td>
</tr>
</tbody>
</table>

### Table 11.38. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--fields &lt;field&gt; [&lt;field&gt; ...]</code></td>
<td>One or more chassis fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

### Table 11.39. Output formatter options

Table 11.40. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.41. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.42. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.12. BAREMETAL CHASSIS UNSET

Unset chassis properties.

Usage:

```
openstack baremetal chassis unset [-h] [--description] [--extra <key>] <chassis>
```

Table 11.43. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;chassis&gt;</td>
<td>Uuid of the chassis</td>
</tr>
</tbody>
</table>

Table 11.44. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description</td>
<td>Clear the chassis description</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset on this chassis (repeat option to unset multiple extras)</td>
</tr>
</tbody>
</table>

11.13. BAREMETAL CONDUCTOR LIST

List baremetal conductors

Usage:

```
openstack baremetal conductor list [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent] [-max-width <integer>]
    [-fit-width] [-print-empty]
    [-sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [-limit <limit>]
    [--marker <conductor>]
    [--sort <key>[:<direction>]]
    [--long | --fields <field> [ <field> ...]]
```

Table 11.45. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of conductors to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;conductor&gt;</td>
<td>Hostname of the conductor (for example, of the last conductor in the list from a previous request). Returns the list of conductors after this conductor.</td>
</tr>
</tbody>
</table>
Sort output by specified conductor fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.

Show detailed information about the conductors.

One or more conductor fields. only these fields will be fetched from the server. Can not be used when --long is specified.

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified conductor fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the conductors.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [...&lt;field&gt;]</td>
<td>One or more conductor fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>

Table 11.46. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.47. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.49. Table formatter options
11.14. BAREMETAL CONDUCTOR SHOW

Show baremetal conductor details

Usage:

openstack baremetal conductor show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   [--fields <field> [<field> ...]]
   <conductor>

Table 11.50. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;conductor&gt;</td>
<td>Hostname of the conductor</td>
</tr>
</tbody>
</table>

Table 11.51. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt;</td>
<td>One or more conductor fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

Table 11.52. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 11.53. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.54. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.55. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.15. BAREMETAL CREATE

Create resources from files

Usage:

```
openstack baremetal create [-h] <file> [<file> ...]
```

Table 11.56. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;file&gt;</td>
<td>File (.yaml or .json) containing descriptions of the resources to create. Can be specified multiple times.</td>
</tr>
</tbody>
</table>
### 11.16. BAREMETAL DEPLOY TEMPLATE CREATE

Create a new deploy template

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 11.58. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Unique name for this deploy template. must be a valid trait name</td>
</tr>
</tbody>
</table>

**Table 11.59. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Uuid of the deploy template.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
<tr>
<td>--steps &lt;steps&gt;</td>
<td>The deploy steps. may be the path to a yaml file containing the deploy steps; OR -, with the deploy steps being read from standard input; OR a JSON string. The value should be a list of deploy-step dictionaries; each dictionary should have keys interface, step, args and priority.</td>
</tr>
</tbody>
</table>

**Table 11.60. Output formatter options**
**Table 11.61. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 11.62. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 11.63. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**11.17. BAREMETAL DEPLOY TEMPLATE DELETE**

Delete deploy template(s).

**Usage:**

```
openstack baremetal deploy template delete [-h] <template> [<template> ...]
```

**Table 11.64. Positional arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;template&gt;</td>
<td>Name(s) or uuid(s) of the deploy template(s) to delete.</td>
</tr>
</tbody>
</table>

Table 11.65. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 11.18. BAREMETAL DEPLOY TEMPLATE LIST

List baremetal deploy templates.

**Usage:**

```
openstack baremetal deploy template list [-h]
  [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--limit <limit>]
  [--marker <template>]
  [--sort <key>[:<direction>]]
  [--long | --fields <field> [<field> ...]]
```

Table 11.66. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of deploy templates to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;template&gt;</td>
<td>Deploytemplate uuid (for example, of the last deploy template in the list from a previous request). Returns the list of deploy templates after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified deploy template fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference 144
### Table 11.67. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--long</td>
<td>Show detailed information about deploy templates.</td>
</tr>
<tr>
<td>--fields (&lt;field&gt; [&lt;field&gt; ...])</td>
<td>One or more deploy template fields. only these fields will be fetched from the server. Can not be used when (--long) is specified.</td>
</tr>
</tbody>
</table>

### Table 11.68. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 11.69. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.70. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width (&lt;integer&gt;)</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
11.19. BAREMETAL DEPLOY TEMPLATE SET

Set baremetal deploy template properties.

Usage:

```
openstack baremetal deploy template set [-h] [--name <name>] [--steps <steps>] [--extra <key=value>] <template>
```

Table 11.71. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;template&gt;</td>
<td>Name or uuid of the deploy template</td>
</tr>
</tbody>
</table>

Table 11.72. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set unique name of the deploy template. must be a valid trait name.</td>
</tr>
<tr>
<td>--steps &lt;steps&gt;</td>
<td>The deploy steps. may be the path to a yaml file containing the deploy steps; OR -, with the deploy steps being read from standard input; OR a JSON string. The value should be a list of deploy-step dictionaries; each dictionary should have keys interface, step, args and priority.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Extra to set on this baremetal deploy template (repeat option to set multiple extras).</td>
</tr>
</tbody>
</table>

11.20. BAREMETAL DEPLOY TEMPLATE SHOW

Show baremetal deploy template details.
Usage:

```
openstack baremetal deploy template show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   [--fields <field> [<field> ...]]
<template>
```

Table 11.73. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;template&gt;</td>
<td>Name or uuid of the deploy template.</td>
</tr>
</tbody>
</table>

Table 11.74. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more deploy template fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

Table 11.75. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.76. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.77. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 11.78. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.21. BAREMETAL DEPLOY TEMPLATE UNSET

Unset baremetal deploy template properties.

Usage:

```
openstack baremetal deploy template unset [-h] [--extra <key>] <template>
```

Table 11.79. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;template&gt;</td>
<td>Name or uuid of the deploy template</td>
</tr>
</tbody>
</table>

Table 11.80. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset on this baremetal deploy template (repeat option to unset multiple extras).</td>
</tr>
</tbody>
</table>

11.22. BAREMETAL DRIVER LIST

List the enabled drivers.

Usage:

```
openstack baremetal driver list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [--quote {all,minimal,none,nonnumeric}] [--noindent] [--max-width <integer>] [--fit-width] [--print-empty]
```
Table 11.81. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>Type of driver (&quot;classic&quot; or &quot;dynamic&quot;). the default is to list all of them.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the drivers.</td>
</tr>
</tbody>
</table>

Table 11.82. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.83. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.84. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.85. Table formatter options
### 11.23. BAREMETAL DRIVER PASSTHRU CALL

Call a vendor passthru method for a driver.

**Usage:**

```bash
openstack baremetal driver passthru call [-h] 
[-f {json,shell,table,value,yaml}] 
[-c COLUMN] [-noindent] 
[-prefix PREFIX] 
[--max-width <integer>] 
[--fit-width] [--print-empty] 
[--arg <key=value>] 
[--http-method <http-method>] 
<driver> <method>
```

#### Table 11.86. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;driver&gt;</td>
<td>Name of the driver.</td>
</tr>
<tr>
<td>&lt;method&gt;</td>
<td>Vendor passthru method to be called.</td>
</tr>
</tbody>
</table>

#### Table 11.87. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--arg &lt;key=value&gt;</td>
<td>Argument to pass to the passthru method (repeat option to specify multiple arguments).</td>
</tr>
<tr>
<td>--http-method &lt;http-method&gt;</td>
<td>The http method to use in the passthru request. one of DELETE, GET, PATCH, POST, PUT. Defaults to POST.</td>
</tr>
</tbody>
</table>
Table 11.88. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.89. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.90. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.91. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.24. BAREMETAL DRIVER PASSTHRU LIST

List available vendor passthrough methods for a driver.

Usage:

```bash
openstack baremetal driver passthru list [-h]
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
```
Table 11.92. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;driver&gt;</td>
<td>Name of the driver.</td>
</tr>
</tbody>
</table>

Table 11.93. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.94. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv,json,table,value,yaml], --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.95. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.96. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 11.97. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.25. BAREMETAL DRIVER PROPERTY LIST

List the driver properties.

Usage:

```
openstack baremetal driver property list [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  <driver>
```

Table 11.98. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;driver&gt;</td>
<td>Name of the driver.</td>
</tr>
</tbody>
</table>

Table 11.99. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.100. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Specify the column(s) to include, can be repeated to show multiple columns

Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

Sort the column(s) in ascending order

Sort the column(s) in descending order

Table 11.101. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.102. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.103. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.26. BAREMETAL DRIVER RAID PROPERTY LIST

List a driver’s RAID logical disk properties.

Usage:

```
openstack baremetal driver raid property list [-h] [-f {csv,json,table,value,yaml}]
```
Table 11.104. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;driver&gt;</td>
<td>Name of the driver.</td>
</tr>
</tbody>
</table>

Table 11.105. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.106. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.107. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.108. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.109. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.27. BAREMETAL DRIVER SHOW

Show information about a driver.

Usage:

```
```

Table 11.110. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;driver&gt;</td>
<td>Name of the driver.</td>
</tr>
</tbody>
</table>

Table 11.111. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.112. Output formatter options
### Table 11.113. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the JSON</td>
</tr>
</tbody>
</table>

### Table 11.114. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 11.115. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.28. BAREMETAL INTROSPECTION ABORT

Abort running introspection for node.

**Usage:**

```
openstack baremetal introspection abort [-h] node
```

### Table 11.116. Positional arguments
### Table 11.117. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Baremetal node uuid or name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 11.29. BAREMETAL INTROSPECTION DATA SAVE

Save or display raw introspection data.

**Usage:**

```
openstack baremetal introspection data save [-h] [--file <filename>] [--unprocessed] node
```

### Table 11.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Baremetal node uuid or name</td>
</tr>
</tbody>
</table>

### Table 11.119. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--file &lt;filename&gt;</td>
<td>Downloaded introspection data filename (default: stdout)</td>
</tr>
<tr>
<td>--unprocessed</td>
<td>Download the unprocessed data</td>
</tr>
</tbody>
</table>

### 11.30. BAREMETAL INTROSPECTION INTERFACE LIST

List interface data including attached switch port information.

**Usage:**

```
openstack baremetal introspection interface list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [--quote {all,minimal,none,nonnumeric}]
```
Table 11.120. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_ident</td>
<td>Baremetal node uuid or name</td>
</tr>
</tbody>
</table>

Table 11.121. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--vlan VLAN</td>
<td>List only interfaces configured for this vlan id, can be repeated</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about interfaces.</td>
</tr>
<tr>
<td>--fields &lt;field&gt;</td>
<td>Display one or more fields. can not be used when --long is specified</td>
</tr>
</tbody>
</table>

Table 11.122. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.123. CSV formatter options
Table 11.124. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.125. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.31. BAREMETAL INTROSPECTION INTERFACE SHOW

Show interface data including attached switch port information.

Usage:

```
openstack baremetal introspection interface show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--fields <field> [<field> ...]]
    node_ident interface
```

Table 11.126. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node_ident</td>
<td>Baremetal node uuid or name</td>
</tr>
<tr>
<td>interface</td>
<td>Interface name</td>
</tr>
</tbody>
</table>
Table 11.127. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [field...]</td>
<td>Display one or more fields.</td>
</tr>
</tbody>
</table>

Table 11.128. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.129. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.130. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.131. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.32. BAREMETAL INTROSPECTION LIST

List introspection statuses
**Usage:**

```bash
openstack baremetal introspection list [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--marker MARKER]
    [--limit LIMIT]
```

**Table 11.132. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker MARKER</td>
<td>Uuid of the last item on the previous page</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>The amount of items to return</td>
</tr>
</tbody>
</table>

**Table 11.133. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 11.134. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 11.135. JSON formatter options**
### Table 11.136. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 11.33. BAREMETAL INTROSPECTION REPROCESS

Reprocess stored introspection data

**Usage:**

```
openstack baremetal introspection reprocess [-h] node
```

**Table 11.137. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Baremetal node uuid or name</td>
</tr>
</tbody>
</table>

**Table 11.138. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### 11.34. BAREMETAL INTROSPECTION RULE DELETE

Delete an introspection rule.

**Usage:**

```
openstack baremetal introspection rule delete [-h] uuid
```

**Table 11.139. Positional arguments**
Table 11.140. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>Rule uuid</td>
</tr>
</tbody>
</table>

Table 11.141. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Json or yaml file to import, may contain one or several rules</td>
</tr>
</tbody>
</table>

Table 11.142. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.143. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 11.144. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 11.145. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 11.146. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.36. BAREMETAL INTROSPECTION RULE LIST

List all introspection rules.

**Usage:**

```
openstack baremetal introspection rule list [-h] [-f {csv,json,table,value,yaml}]
```
Table 11.147. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.148. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.149. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.150. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.151. Table formatter options
### 11.37. BAREMETAL INTROSPECTION RULE PURGE

Drop all introspection rules.

**Usage:**

```
openstack baremetal introspection rule purge [-h]
```

**Table 11.152. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 11.38. BAREMETAL INTROSPECTION RULE SHOW

Show an introspection rule.

**Usage:**

```
openstack baremetal introspection rule show [-h]
```

**Table 11.153. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>uuid</td>
<td>Rule uuid</td>
</tr>
</tbody>
</table>

**Table 11.154. Command arguments**
**Value** | **Summary**
--- | ---
-h, --help | Show this help message and exit

**Table 11.155. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 11.156. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 11.157. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 11.158. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**11.39. BAREMETAL INTROSPECTION START**

Start the introspection.

**Usage:**
openstack baremetal introspection start [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--wait] [--check-errors]
  node [node ...]

Table 11.159. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Baremetal node uuid(s) or name(s)</td>
</tr>
</tbody>
</table>

Table 11.160. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for introspection to finish; the result will be displayed in the end</td>
</tr>
<tr>
<td>--check-errors</td>
<td>Check if errors occurred during the introspection; if any error occurs only the errors are displayed; can only be used with --wait</td>
</tr>
</tbody>
</table>

Table 11.161. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 11.162. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.163. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.164. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.40. BAREMETAL INTROSPECTION STATUS

Get introspection status.

Usage:

```bash
openstack baremetal introspection status [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [-noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  node
```

Table 11.165. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>Baremetal node uuid or name</td>
</tr>
</tbody>
</table>
### Table 11.167. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

- **f** `{json,shell,table,value,yaml}`  
  **--format** `{json,shell,table,value,yaml}`  
  The output format, defaults to table

- **c COLUMN, --column COLUMN**  
  Specify the column(s) to include, can be repeated to show multiple columns

### Table 11.168. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.169. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 11.170. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

| --fit-width | Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable |

| --print-empty | Print empty table if there is no data to show. |

### 11.41. BAREMETAL NODE ABORT

Set provision state of baremetal node to *abort*

**Usage:**
openstack baremetal node abort [-h] <node>

Table 11.171. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.172. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.42. BAREMETAL NODE ADD TRAIT

Add traits to a node.

Usage:

openstack baremetal node add trait [-h] <node> <trait> [<trait> ...]

Table 11.173. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;trait&gt;</td>
<td>Trait(s) to add</td>
</tr>
</tbody>
</table>

Table 11.174. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.43. BAREMETAL NODE ADOPT

Set provision state of baremetal node to adopt

Usage:

openstack baremetal node adopt [-h] [--wait [<time-out>]] <node>

Table 11.175. Positional arguments
Table 11.176. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.177. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.178. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.179. Output formatter options

11.44. BAREMETAL NODE BIOS SETTING LIST

List a node's BIOS settings.

Usage:

```bash
openstack baremetal node bios setting list [-h]
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    <node>
```

Table 11.177. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.178. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.179. Output formatter options
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 11.180. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 11.181. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 11.182. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**11.45. BAREMETAL NODE BIOS SETTING SHOW**

Show a specific BIOS setting for a node.

**Usage:**
openstack baremetal node bios setting show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    <node> <setting name>

Table 11.183. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;setting name&gt;</td>
<td>Setting name to show</td>
</tr>
</tbody>
</table>

Table 11.184. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.185. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to</td>
</tr>
<tr>
<td></td>
<td>show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.186. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.187. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.188. Table formatter options
### 11.46. BAREMETAL NODE BOOT DEVICE SET

Set the boot device for a node

**Usage:**

```
openstack baremetal node boot device set [-h] [--persistent] <node> <device>
```

**Table 11.189. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;device&gt;</td>
<td>One of bios, cdrom, disk, pxe, safe, wanboot</td>
</tr>
</tbody>
</table>

**Table 11.190. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--persistent</td>
<td>Make changes persistent for all future boots</td>
</tr>
</tbody>
</table>

### 11.47. BAREMETAL NODE BOOT DEVICE SHOW

Show the boot device information for a node

**Usage:**

```
openstack baremetal node boot device show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>]```

---

**Value**

<table>
<thead>
<tr>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
</tr>
<tr>
<td>--fit-width</td>
</tr>
<tr>
<td>--print-empty</td>
</tr>
</tbody>
</table>
Table 11.191. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.192. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--supported</td>
<td>Show the supported boot devices</td>
</tr>
</tbody>
</table>

Table 11.193. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.194. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.195. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.196. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
11.48. BAREMETAL NODE CLEAN

Set provision state of baremetal node to clean

Usage:

```
openstack baremetal node clean [-h] [--wait [<time-out>]] --clean-steps <clean-steps> <node>
```

Table 11.197. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.198. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [&lt;time-out&gt;]</td>
<td>Wait for a node to reach the desired state, manageable. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
<tr>
<td>--clean-steps &lt;clean-steps&gt;</td>
<td>The clean steps. may be the path to a yaml file containing the clean steps; OR -, with the clean steps being read from standard input; OR a JSON string. The value should be a list of clean-step dictionaries; each dictionary should have keys <code>interface</code> and <code>step</code>, and optional key <code>args</code>.</td>
</tr>
</tbody>
</table>

11.49. BAREMETAL NODE CONSOLE DISABLE
Disable console access for a node

**Usage:**

```
openstack baremetal node console disable [-h] <node>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

**Table 11.199. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 11.200. Command arguments**

Enable console access for a node

**Usage:**

```
openstack baremetal node console enable [-h] <node>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

**Table 11.201. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 11.202. Command arguments**

Show console information for a node

**Usage:**

```
openstack baremetal node console show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [-noindent]
  [-prefix PREFIX]
```

**Table 11.203. Command arguments**

11.50. BAREMETAL NODE CONSOLE ENABLE

11.51. BAREMETAL NODE CONSOLE SHOW
Table 11.203. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.204. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.205. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.206. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.207. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.208. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
11.52. BAREMETAL NODE CREATE

Register a new node with the baremetal service

Usage:


Table 11.209. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--chassis-uuid &lt;chassis&gt;</td>
<td>Uuid of the chassis that this node belongs to.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--driver &lt;driver&gt;</td>
<td>Driver used to control the node [required].</td>
</tr>
<tr>
<td>--driver-info &lt;key=value&gt;</td>
<td>Key/value pair used by the driver, such as out-of-band management credentials. Can be specified multiple times.</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Key/value pair describing the physical characteristics of the node. This is exported to Nova and used by the scheduler. Can be specified multiple times.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Unique uuid for the node.</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Unique name for the node.</td>
</tr>
<tr>
<td>--bios-interface &lt;bios_interface&gt;</td>
<td>Bios interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--boot-interface &lt;boot_interface&gt;</td>
<td>Boot interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--console-interface &lt;console_interface&gt;</td>
<td>Console interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--deploy-interface &lt;deploy_interface&gt;</td>
<td>Deploy interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--inspect-interface &lt;inspect_interface&gt;</td>
<td>Inspect interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--management-interface &lt;management_interface&gt;</td>
<td>Management interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--network-data &lt;network_data&gt;</td>
<td>Json string or a yaml file or - for stdin to read static network configuration for the baremetal node associated with this ironic node. Format of this file should comply with Nova network data metadata (network_data.json). Depending on ironic boot interface capabilities being used, network configuration may or may not been served to the node for offline network configuration.</td>
</tr>
<tr>
<td>--network-interface &lt;network_interface&gt;</td>
<td>Network interface used for switching node to cleaning/provisioning networks.</td>
</tr>
<tr>
<td>--power-interface &lt;power_interface&gt;</td>
<td>Power interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--raid-interface &lt;raid_interface&gt;</td>
<td>Raid interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--rescue-interface &lt;rescue_interface&gt;</td>
<td>Rescue interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--storage-interface &lt;storage_interface&gt;</td>
<td>Storage interface used by the node’s driver.</td>
</tr>
<tr>
<td>--vendor-interface &lt;vendor_interface&gt;</td>
<td>Vendor interface used by the node’s driver. this is only applicable when the specified --driver is a hardware type.</td>
</tr>
<tr>
<td>--resource-class &lt;resource_class&gt;</td>
<td>Resource class for mapping nodes to nova flavors</td>
</tr>
<tr>
<td>--conductor-group &lt;conductor_group&gt;</td>
<td>Conductor group the node will belong to</td>
</tr>
<tr>
<td>--automated-clean</td>
<td>Enable automated cleaning for the node</td>
</tr>
<tr>
<td>--no-automated-clean</td>
<td>Explicitly disable automated cleaning for the node</td>
</tr>
<tr>
<td>--owner &lt;owner&gt;</td>
<td>Owner of the node.</td>
</tr>
<tr>
<td>--lessee &lt;lessee&gt;</td>
<td>Lessee of the node.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the node.</td>
</tr>
</tbody>
</table>

Table 11.210. Output formatter options
11.53. BAREMETAL NODE DELETE

Unregister baremetal node(s)

Usage:

```
openstack baremetal node delete [-h] <node> [<node> ...]
```

Table 11.214. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Node(s) to delete (name or uuid)</td>
</tr>
</tbody>
</table>
### Table 11.215. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 11.54. BAREMETAL NODE DEPLOY

Set provision state of baremetal node to *deploy*

#### Usage:

```
openstack baremetal node deploy [-h] [--wait [<time-out>]]
   [--config-drive <config-drive>]
   [--deploy-steps <deploy-steps>]
   <node>
```

#### Table 11.216. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

#### Table 11.217. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [&lt;time-out&gt;]</td>
<td>Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
<tr>
<td>--config-drive &lt;config-drive&gt;</td>
<td>A gzipped, base64-encoded configuration drive string OR the path to the configuration drive file OR the path to a directory containing the config drive files OR a JSON object to build config drive from. In case it's a directory, a config drive will be generated from it. In case it's a JSON object with optional keys <em>meta_data</em>, <em>user_data</em> and <em>network_data</em>, a config drive will be generated on the server side (see the bare metal API reference for more details).</td>
</tr>
</tbody>
</table>
The deploy steps may be the path to a yaml file containing the deploy steps; OR, with the deploy steps being read from standard input; OR a JSON string. The value should be a list of deploy-step dictionaries; each dictionary should have keys `interface` and `step`, and optional key `args`.

### 11.55. BAREMETAL NODE INJECT NMI

Inject NMI to baremetal node

**Usage:**

```bash
openstack baremetal node inject nmi [-h] <node>
```

**Table 11.218. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.219. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 11.56. BAREMETAL NODE INSPECT

Set provision state of baremetal node to *inspect*

**Usage:**

```bash
openstack baremetal node inspect [-h] [--wait [<time-out>]] <node>
```

**Table 11.220. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.221. Command arguments**
### 11.57. BAREMETAL NODE LIST

List baremetal nodes

**Usage:**

```
```

#### Table 11.222. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of nodes to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;node&gt;</td>
<td>Node uuid (for example, of the last node in the list from a previous request). Returns the list of nodes after this UUID.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified node fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--maintenance</td>
<td>Limit list to nodes in maintenance mode</td>
</tr>
<tr>
<td>--no-maintenance</td>
<td>Limit list to nodes not in maintenance mode</td>
</tr>
<tr>
<td>--retired</td>
<td>Limit list to retired nodes</td>
</tr>
<tr>
<td>--no-retired</td>
<td>Limit list to not retired nodes</td>
</tr>
<tr>
<td>--fault &lt;fault&gt;</td>
<td>List nodes in specified fault</td>
</tr>
<tr>
<td>--associated</td>
<td>List only nodes associated with an instance</td>
</tr>
<tr>
<td>--unassociated</td>
<td>List only nodes not associated with an instance</td>
</tr>
<tr>
<td>--provision-state &lt;provision state&gt;</td>
<td>List nodes in specified provision state.</td>
</tr>
<tr>
<td>--driver &lt;driver&gt;</td>
<td>Limit list to nodes with driver &lt;driver&gt;</td>
</tr>
<tr>
<td>--resource-class &lt;resource class&gt;</td>
<td>Limit list to nodes with resource class &lt;resource class&gt;</td>
</tr>
<tr>
<td>--conductor-group &lt;conductor_group&gt;</td>
<td>Limit list to nodes with conductor group &lt;conductor group&gt;</td>
</tr>
<tr>
<td>--conductor &lt;conductor&gt;</td>
<td>Limit list to nodes with conductor &lt;conductor&gt;</td>
</tr>
<tr>
<td>--chassis &lt;chassis UUID&gt;</td>
<td>Limit list to nodes of this chassis</td>
</tr>
<tr>
<td>--owner &lt;owner&gt;</td>
<td>Limit list to nodes with owner &lt;owner&gt;</td>
</tr>
<tr>
<td>--lessee &lt;lessee&gt;</td>
<td>Limit list to nodes with lessee &lt;lessee&gt;</td>
</tr>
<tr>
<td>--description-contains &lt;description_contains&gt;</td>
<td>Limit list to nodes with description contains &lt;description_contains&gt;</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the nodes.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [... ]</td>
<td>One or more node fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>

Table 11.223. Output formatter options
### 11.224. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### 11.225. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 11.226. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.58. BAREMETAL NODE MAINTENANCE SET

Set baremetal node to maintenance mode

**Usage:**
openstack baremetal node maintenance set [-h] [--reason <reason>] <node>

Table 11.227. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.228. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--reason</td>
<td>Reason for setting maintenance mode.</td>
</tr>
</tbody>
</table>

11.59. BAREMETAL NODE MAINTENANCE UNSET

Unset baremetal node from maintenance mode

Usage:

openstack baremetal node maintenance unset [-h] <node>

Table 11.229. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.230. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.60. BAREMETAL NODE MANAGE

Set provision state of baremetal node to manage

Usage:

openstack baremetal node manage [-h] [--wait [<time-out>]] <node>

Table 11.231. Positional arguments
### Table 11.232. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [&lt;time-out&gt;]</td>
<td>Wait for a node to reach the desired state, manageable. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
</tbody>
</table>

### 11.61. BAREMETAL NODE PASSTHRU CALL

Call a vendor passthru method for a node

**Usage:**

```
```

### Table 11.233. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;method&gt;</td>
<td>Vendor passthru method to be executed</td>
</tr>
</tbody>
</table>

### Table 11.234. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--arg &lt;key=value&gt;</td>
<td>Argument to pass to the passthru method (repeat option to specify multiple arguments)</td>
</tr>
<tr>
<td>--http-method &lt;http-method&gt;</td>
<td>The http method to use in the passthru request. one of DELETE, GET, PATCH, POST, PUT. Defaults to POST.</td>
</tr>
</tbody>
</table>

### 11.62. BAREMETAL NODE PASSTHRU LIST
List vendor passthru methods for a node

Usage:

```bash
```

Table 11.235. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.236. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.237. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.238. CSV formatter options
Table 11.239. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.240. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.63. BAREMETAL NODE POWER OFF

Power off a node

Usage:

```plaintext
openstack baremetal node power off [-h]
   [--power-timeout <power-timeout>]
   [--soft]
   <node>
```

Table 11.241. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.242. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
Power on a node

**Usage:**

```
openstack baremetal node power on [-h] [--power-timeout <power-timeout>] <node>
```

**Table 11.243. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.244. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--power-timeout &lt;v&gt;</td>
<td>Timeout (in seconds, positive integer) to wait for the target power state before erroring out</td>
</tr>
</tbody>
</table>

---

Set provision state of baremetal node to *provide*

**Usage:**

```
openstack baremetal node provide [-h] [--wait [time-out]] <node>
```

**Table 11.245. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.246. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [&lt;time-out&gt;]</td>
<td>Wait for a node to reach the desired state, available. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
</tbody>
</table>

### 11.66. BAREMETAL NODE REBOOT

Reboot baremetal node

**Usage:**

```plaintext
openstack baremetal node reboot [-h] [--soft] [--power-timeout <power-timeout>] <node>
```

#### Table 11.247. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

#### Table 11.248. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--soft</td>
<td>Request graceful reboot.</td>
</tr>
<tr>
<td>--power-timeout &lt;power-timeout&gt;</td>
<td>Timeout (in seconds, positive integer) to wait for the target power state before erring out.</td>
</tr>
</tbody>
</table>

### 11.67. BAREMETAL NODE REBUILD

Set provision state of baremetal node to `rebuild`

**Usage:**

```plaintext
openstack baremetal node rebuild [-h] [-wait [<time-out>]] [--config-drive <config-drive>] [--deploy-steps <deploy-steps>] <node>
```

#### Table 11.249. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.250. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [&lt;time-out&gt;]</td>
<td>Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
<tr>
<td>--config-drive &lt;config-drive&gt;</td>
<td>A gzipped, base64-encoded configuration drive string OR the path to the configuration drive file OR the path to a directory containing the config drive files OR a JSON object to build config drive from. In case it’s a directory, a config drive will be generated from it. In case it’s a JSON object with optional keys meta_data, user_data and network_data, a config drive will be generated on the server side (see the bare metal API reference for more details).</td>
</tr>
<tr>
<td>--deploy-steps &lt;deploy-steps&gt;</td>
<td>The deploy steps in json format. may be the path to a file containing the deploy steps; OR -, with the deploy steps being read from standard input; OR a string. The value should be a list of deploy-step dictionaries; each dictionary should have keys interface, step, priority and optional key args.</td>
</tr>
</tbody>
</table>

### 11.68. BAREMETAL NODE REMOVE TRAIT

Remove trait(s) from a node.

**Usage:**

```
openstack baremetal node remove trait [-h] [--all] <node> [<trait> ...]
```

**Table 11.251. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;trait&gt;</td>
<td>Trait(s) to remove</td>
</tr>
</tbody>
</table>

**Table 11.252. Command arguments**
### 11.69. BAREMETAL NODE RESCUE

Set provision state of baremetal node to *rescue*

**Usage:**

```
openstack baremetal node rescue [-h] [--wait [<time-out>]]
    --rescue-password <rescue-password>
    <node>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--all</code></td>
<td>Remove all traits</td>
</tr>
</tbody>
</table>

Table 11.253. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;node&gt;</code></td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.254. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--wait [&lt;time-out&gt;]</code></td>
<td>Wait for a node to reach the desired state, rescue. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
<tr>
<td><code>--rescue-password &lt;rescue-password&gt;</code></td>
<td>The password that will be used to login to the rescue ramdisk. The value should be a non-empty string.</td>
</tr>
</tbody>
</table>

### 11.70. BAREMETAL NODE SET

Set baremetal properties

**Usage:**

```
openstack baremetal node set [-h] [--instance-uuid <uuid>]
    [--name <name>]
    [--chassis-uuid <chassis UUID>]
    [--driver <driver>]
    [--bios-interface <bios_interface> | --reset-bios-interface]
    [--boot-interface <boot_interface> | --reset-boot-interface]
    [--console-interface <console_interface> | --reset-console-interface]
    [--deploy-interface <deploy_interface> | --reset-deploy-interface]
```
Red Hat OpenStack Platform 17.0 Command Line Interface Reference

--inspect-interface <inspect_interface> | --reset-inspect-interface

--management-interface <management_interface> | --reset-management-interface

--network-interface <network_interface> | --reset-network-interface

--network-data <network data>

--power-interface <power_interface> | --reset-power-interface

--raid-interface <raid_interface> | --reset-raid-interface

--rescue-interface <rescue_interface> | --reset-rescue-interface

--storage-interface <storage_interface> | --reset-storage-interface

--vendor-interface <vendor_interface> | --reset-vendor-interface

--reset-interfaces

--resource-class <resource_class>

--conductor-group <conductor_group>

--automated-clean | --no-automated-clean

--protected

--protected-reason <protected_reason>

--retired

--retired-reason <retired_reason>

--target-raid-config <target_raid_config>

--property <key=value>

--extra <key=value>

--driver-info <key=value>

--instance-info <key=value>

--owner <owner> | --lessee <lessee>

--description <description>

<node>

Table 11.255. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.256. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--instance-uuid &lt;uuid&gt;</td>
<td>Set instance uuid of node to &lt;uuid&gt;</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the node</td>
</tr>
<tr>
<td>--chassis-uuid &lt;chassis UUID&gt;</td>
<td>Set the chassis for the node</td>
</tr>
<tr>
<td>--driver &lt;driver&gt;</td>
<td>Set the driver for the node</td>
</tr>
<tr>
<td>--bios-interface &lt;bios_interface&gt;</td>
<td>Set the bios interface for the node</td>
</tr>
<tr>
<td>--reset-bios-interface</td>
<td>Reset the bios interface to its hardware type default</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--boot-interface &lt;boot_interface&gt;</td>
<td>Set the boot interface for the node</td>
</tr>
<tr>
<td>--reset-boot-interface</td>
<td>Reset the boot interface to its hardware type default</td>
</tr>
<tr>
<td>--console-interface &lt;console_interface&gt;</td>
<td>Set the console interface for the node</td>
</tr>
<tr>
<td>--reset-console-interface</td>
<td>Reset the console interface to its hardware type default</td>
</tr>
<tr>
<td>--deploy-interface &lt;deploy_interface&gt;</td>
<td>Set the deploy interface for the node</td>
</tr>
<tr>
<td>--reset-deploy-interface</td>
<td>Reset the deploy interface to its hardware type default</td>
</tr>
<tr>
<td>--inspect-interface &lt;inspect_interface&gt;</td>
<td>Set the inspect interface for the node</td>
</tr>
<tr>
<td>--reset-inspect-interface</td>
<td>Reset the inspect interface to its hardware type default</td>
</tr>
<tr>
<td>--management-interface &lt;management_interface&gt;</td>
<td>Set the management interface for the node</td>
</tr>
<tr>
<td>--reset-management-interface</td>
<td>Reset the management interface to its hardware type default</td>
</tr>
<tr>
<td>--network-interface &lt;network_interface&gt;</td>
<td>Set the network interface for the node</td>
</tr>
<tr>
<td>--reset-network-interface</td>
<td>Reset the network interface to its hardware type default</td>
</tr>
<tr>
<td>--network-data &lt;network data&gt;</td>
<td>Json string or a yaml file or - for stdin to read static network configuration for the baremetal node associated with this ironic node. Format of this file should comply with Nova network data metadata (network_data.json). Depending on ironic boot interface capabilities being used, network configuration may or may not been served to the node for offline network configuration.</td>
</tr>
<tr>
<td>--power-interface &lt;power_interface&gt;</td>
<td>Set the power interface for the node</td>
</tr>
<tr>
<td>--reset-power-interface</td>
<td>Reset the power interface to its hardware type default</td>
</tr>
<tr>
<td>--raid-interface &lt;raid_interface&gt;</td>
<td>Set the raid interface for the node</td>
</tr>
<tr>
<td>--reset-raid-interface</td>
<td>Reset the raid interface to its hardware type default</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--rescue-interface &lt;rescue_interface&gt;</td>
<td>Set the rescue interface for the node</td>
</tr>
<tr>
<td>--reset-rescue-interface</td>
<td>Reset the rescue interface to its hardware type default</td>
</tr>
<tr>
<td>--storage-interface &lt;storage_interface&gt;</td>
<td>Set the storage interface for the node</td>
</tr>
<tr>
<td>--reset-storage-interface</td>
<td>Reset the storage interface to its hardware type default</td>
</tr>
<tr>
<td>--vendor-interface &lt;vendor_interface&gt;</td>
<td>Set the vendor interface for the node</td>
</tr>
<tr>
<td>--reset-vendor-interface</td>
<td>Reset the vendor interface to its hardware type default</td>
</tr>
<tr>
<td>--reset-interfaces</td>
<td>Reset all interfaces not specified explicitly to their default implementations. Only valid with --driver.</td>
</tr>
<tr>
<td>--resource-class &lt;resource_class&gt;</td>
<td>Set the resource class for the node</td>
</tr>
<tr>
<td>--conductor-group &lt;conductor_group&gt;</td>
<td>Set the conductor group for the node</td>
</tr>
<tr>
<td>--automated-clean</td>
<td>Enable automated cleaning for the node</td>
</tr>
<tr>
<td>--no-automated-clean</td>
<td>Explicitly disable automated cleaning for the node</td>
</tr>
<tr>
<td>--protected</td>
<td>Mark the node as protected</td>
</tr>
<tr>
<td>--protected-reason &lt;protected_reason&gt;</td>
<td>Set the reason of marking the node as protected</td>
</tr>
<tr>
<td>--retired</td>
<td>Mark the node as retired</td>
</tr>
<tr>
<td>--retired-reason &lt;retired_reason&gt;</td>
<td>Set the reason of marking the node as retired</td>
</tr>
<tr>
<td>--target-raid-config &lt;target_raid_config&gt;</td>
<td>Set the target raid configuration (json) for the node. This can be one of: 1. a file containing YAML data of the RAID configuration; 2. &quot;-&quot; to read the contents from standard input; or 3. a valid JSON string.</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to set on this baremetal node (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Extra to set on this baremetal node (repeat option to set multiple extras)</td>
</tr>
</tbody>
</table>
11.71. BAREMETAL NODE SHOW

Show baremetal node details

Usage:

```
```

Table 11.257. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node (or instance uuid if --instance is specified)</td>
</tr>
</tbody>
</table>

Table 11.258. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--instance</td>
<td>&lt;node&gt; is an instance uuid.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more node fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>
### Table 11.260. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 11.261. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 11.262. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.72. BAREMETAL NODE TRAIT LIST

List a node’s traits.

**Usage:**

```bash
openstack baremetal node trait list [-h]  
[-f {csv,json,table,value,yaml}]  
[-c COLUMN]  
[--quote {all,minimal,none,nonnumeric}]  
[--noindent]  
[--max-width <integer>]  
[--fit-width] [--print-empty]
```
Table 11.263. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.264. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.265. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.266. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.267. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.268. Table formatter options
Set provision state of baremetal node to \textit{deleted}

**Usage:**

```
openstack baremetal node undeploy [-h] [--wait [<time-out>]] <node>
```

**Table 11.269. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;node&gt;</code></td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

**Table 11.270. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--wait [&lt;time-out&gt;]</code></td>
<td>Wait for a node to reach the desired state, available. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
</tbody>
</table>

Set provision state of baremetal node to \textit{unrescue}

**Usage:**

```
openstack baremetal node unrescue [-h] [--wait [<time-out>]] <node>
```

**Table 11.271. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;node&gt;</code></td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>
Table 11.272. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.273. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node.</td>
</tr>
</tbody>
</table>

Table 11.274. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait [time-out]</td>
<td>Wait for a node to reach the desired state, active. Optionally takes a timeout value (in seconds). The default value is 0, meaning it will wait indefinitely.</td>
</tr>
</tbody>
</table>

11.75. BAREMETAL NODE UNSET

Unset baremetal properties

Usage:

```
openstack baremetal node unset [-h] [--instance-uuid] [--name]                  
  [--resource-class] 
  [--target-raid-config] 
  [--property <key>] [--extra <key>] 
  [--driver-info <key>] 
  [--instance-info <key>] [--chassis-uuid] 
  [-bios-interface] [-boot-interface] 
  [--console-interface] 
  [--deploy-interface] 
  [-inspect-interface] [-network-data] 
  [-management-interface] 
  [-network-interface] 
  [-power-interface] [-raid-interface] 
  [--rescue-interface] 
  [--storage-interface] 
  [-vendor-interface] [-conductor-group] 
  [-automated-clean] [-protected] 
  [-protected-reason] [-retired] 
  [-retired-reason] [-owner] [-lessee] 
  [--description] 
  <node>                                                                 
```
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--instance-uuid</td>
<td>Unset instance uuid on this baremetal node</td>
</tr>
<tr>
<td>--name</td>
<td>Unset the name of the node</td>
</tr>
<tr>
<td>--resource-class</td>
<td>Unset the resource class of the node</td>
</tr>
<tr>
<td>--target-raid-config</td>
<td>Unset the target raid configuration of the node</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to unset on this baremetal node (repeat option to unset multiple properties)</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset on this baremetal node (repeat option to unset multiple extras)</td>
</tr>
<tr>
<td>--driver-info &lt;key&gt;</td>
<td>Driver information to unset on this baremetal node (repeat option to unset multiple driver informations)</td>
</tr>
<tr>
<td>--instance-info &lt;key&gt;</td>
<td>Instance information to unset on this baremetal node (repeat option to unset multiple instance informations)</td>
</tr>
<tr>
<td>--chassis-uuid</td>
<td>Unset chassis uuid on this baremetal node</td>
</tr>
<tr>
<td>--bios-interface</td>
<td>Unset bios interface on this baremetal node</td>
</tr>
<tr>
<td>--boot-interface</td>
<td>Unset boot interface on this baremetal node</td>
</tr>
<tr>
<td>--console-interface</td>
<td>Unset console interface on this baremetal node</td>
</tr>
<tr>
<td>--deploy-interface</td>
<td>Unset deploy interface on this baremetal node</td>
</tr>
<tr>
<td>--inspect-interface</td>
<td>Unset inspect interface on this baremetal node</td>
</tr>
<tr>
<td>--network-data</td>
<td>Unset network data on this baremetal port.</td>
</tr>
<tr>
<td>--management-interface</td>
<td>Unset management interface on this baremetal node</td>
</tr>
<tr>
<td>--network-interface</td>
<td>Unset network interface on this baremetal node</td>
</tr>
<tr>
<td>--power-interface</td>
<td>Unset power interface on this baremetal node</td>
</tr>
<tr>
<td>--raid-interface</td>
<td>Unset raid interface on this baremetal node</td>
</tr>
</tbody>
</table>
## Value Summary

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--rescue-interface</td>
<td>Unset rescue interface on this baremetal node</td>
</tr>
<tr>
<td>--storage-interface</td>
<td>Unset storage interface on this baremetal node</td>
</tr>
<tr>
<td>--vendor-interface</td>
<td>Unset vendor interface on this baremetal node</td>
</tr>
<tr>
<td>--conductor-group</td>
<td>Unset conductor group for this baremetal node (the default group will be used)</td>
</tr>
<tr>
<td>--automated-clean</td>
<td>Unset automated clean option on this baremetal node (the value from configuration will be used)</td>
</tr>
<tr>
<td>--protected</td>
<td>Unset the protected flag on the node</td>
</tr>
<tr>
<td>--protected-reason</td>
<td>Unset the protected reason (gets unset automatically when protected is unset)</td>
</tr>
<tr>
<td>--retired</td>
<td>Unset the retired flag on the node</td>
</tr>
<tr>
<td>--retired-reason</td>
<td>Unset the retired reason (gets unset automatically when retired is unset)</td>
</tr>
<tr>
<td>--owner</td>
<td>Unset the owner field of the node</td>
</tr>
<tr>
<td>--lessee</td>
<td>Unset the lessee field of the node</td>
</tr>
<tr>
<td>--description</td>
<td>Unset the description field of the node</td>
</tr>
</tbody>
</table>

### 11.76. BAREMETAL NODE VALIDATE

Validate a node’s driver interfaces

**Usage:**

```bash
```

Table 11.275. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;node&gt;</code></td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.276. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.277. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN  </code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN   </code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending            </code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending           </code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.278. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.279. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.280. Table formatter options
## 11.77. BAREMETAL NODE VIF ATTACH

Attach VIF to a given node

**Usage:**

```
openstack baremetal node vif attach [-h] [--port-uuid <port-uuid>] [--vif-info <key=value>] <node> <vif-id>
```

**Value**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;node&gt;</code></td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td><code>&lt;vif-id&gt;</code></td>
<td>Name or uuid of the vif to attach to a node.</td>
</tr>
</tbody>
</table>

**Table 11.282. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--port-uuid &lt;port-uuid&gt;</code></td>
<td>Uuid of the baremetal port to attach the vif to.</td>
</tr>
<tr>
<td><code>--vif-info &lt;key=value&gt;</code></td>
<td>Record arbitrary key/value metadata. can be specified multiple times. The mandatory id parameter cannot be specified as a key.</td>
</tr>
</tbody>
</table>

## 11.78. BAREMETAL NODE VIF DETACH

Detach VIF from a given node
Usage:

openstack baremetal node vif detach [-h] <node> <vif-id>

Table 11.283. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
<tr>
<td>&lt;vif-id&gt;</td>
<td>Name or uuid of the vif to detach from a node.</td>
</tr>
</tbody>
</table>

Table 11.284. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.79. BAREMETAL NODE VIF LIST

Show attached VIFs for a node

Usage:


Table 11.285. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node&gt;</td>
<td>Name or uuid of the node</td>
</tr>
</tbody>
</table>

Table 11.286. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 11.287. Output formatter options
### CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 11.80. BAREMETAL PORT CREATE

Create a new port

**Usage:**

<address>

Table 11.291. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;address&gt;</td>
<td>Mac address for this port.</td>
</tr>
</tbody>
</table>

Table 11.292. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Uuid of the node that this port belongs to.</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Uuid of the port.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. argument can be specified multiple times.</td>
</tr>
<tr>
<td>--local-link-connection &lt;key=value&gt;</td>
<td>Key/value metadata describing local link connection information. Valid keys are switch_info, switch_id, port_id and hostname. The keys switch_id and port_id are required. In case of a Smart NIC port, the required keys are port_id and hostname. Argument can be specified multiple times.</td>
</tr>
<tr>
<td>-l &lt;key=value&gt;</td>
<td>Deprecated. please use --local-link-connection instead. Key/value metadata describing Local link connection information. Valid keys are switch_info, switch_id, and port_id. The keys switch_id and port_id are required. Can be specified multiple times.</td>
</tr>
<tr>
<td>--pxe-enabled &lt;boolean&gt;</td>
<td>Indicates whether this port should be used when pxe booting this Node.</td>
</tr>
<tr>
<td>--port-group &lt;uuid&gt;</td>
<td>Uuid of the port group that this port belongs to.</td>
</tr>
</tbody>
</table>
**Value** | **Summary**
---|---
--physical-network <physical network> | Name of the physical network to which this port is connected.
--is-smartnic | Indicates whether this port is a smart nic port

**Table 11.293. Output formatter options**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 11.294. JSON formatter options**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 11.295. Shell formatter options**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 11.296. Table formatter options**

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**11.81. BAREMETAL PORT DELETE**

Delete port(s).
Usage:

openstack baremetal port delete [-h] <port> [<port> ...]

Table 11.297. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Uuid(s) of the port(s) to delete.</td>
</tr>
</tbody>
</table>

Table 11.298. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.82. BAREMETAL PORT GROUP CREATE

Create a new baremetal port group.

Usage:

openstack baremetal port group create [-h]

[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
--node <uuid>
[--address <mac-address>]
[--name NAME] [--uuid UUID]
[--extra <key=value>]
[--mode MODE]
[--property <key=value>]
[--support-standalone-ports | --unsupport-standalone-ports]

Table 11.299. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Uuid of the node that this port group belongs to.</td>
</tr>
<tr>
<td>--address &lt;mac-address&gt;</td>
<td>Mac address for this port group.</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Name of the port group.</td>
</tr>
<tr>
<td>--uuid UUID</td>
<td>Uuid of the port group.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
<tr>
<td>--mode MODE</td>
<td>Mode of the port group. for possible values, refer to <a href="https://www.kernel.org/doc/Documentation/networking/binding.txt">https://www.kernel.org/doc/Documentation/networking/binding.txt</a>.</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Key/value property related to this port group’s configuration. Can be specified multiple times.</td>
</tr>
<tr>
<td>--support-standalone-ports</td>
<td>Ports that are members of this port group can be used as stand-alone ports. (default)</td>
</tr>
<tr>
<td>--unsupport-standalone-ports</td>
<td>Ports that are members of this port group cannot be used as stand-alone ports.</td>
</tr>
</tbody>
</table>

**Table 11.300. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 11.301. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 11.302. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 11.303. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
11.83. BAREMETAL PORT GROUP DELETE

Unregister baremetal port group(s).

Usage:

```bash
openstack baremetal port group delete [-h] <port group> [<port group> ...]
```

Table 11.304. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port group&gt;</td>
<td>Port group(s) to delete (name or uuid).</td>
</tr>
</tbody>
</table>

Table 11.305. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.84. BAREMETAL PORT GROUP LIST

List baremetal port groups.

Usage:

```bash
```
Table 11.306. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of port groups to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;port group&gt;</td>
<td>Port group uuid (for example, of the last port group in the list from a previous request). Returns the list of port groups after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified port group fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--address &lt;mac-address&gt;</td>
<td>Only show information for the port group with this mac address.</td>
</tr>
<tr>
<td>--node &lt;node&gt;</td>
<td>Only list port groups of this node (name or uuid).</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about the port groups.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more port group fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>

Table 11.307. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
Value | Summary
--- | ---
--sort-descending | Sort the column(s) in descending order

Table 11.308. CSV formatter options

Value | Summary
--- | ---
--quote {all,minimal,none,nonnumeric} | When to include quotes, defaults to nonnumeric

Table 11.309. JSON formatter options

Value | Summary
--- | ---
--noindent | Whether to disable indenting the json

Table 11.310. Table formatter options

Value | Summary
--- | ---
--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty | Print empty table if there is no data to show.

11.85. BAREMETAL PORT GROUP SET

Set baremetal port group properties.

Usage:

```
openstack baremetal port group set [-h] [--node <uuid>]
  [--address <mac-address>]
  [--name <name>]
  [--extra <key=value>] [--mode MODE]
  [--property <key=value>]
  [--support-standalone-ports | --unsupport-standalone-ports]
  <port group>
```

Table 11.311. Positional arguments
### Table 11.312. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port group&gt;</td>
<td>Name or uuid of the port group.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Update uuid of the node that this port group belongs to.</td>
</tr>
<tr>
<td>--address &lt;mac-address&gt;</td>
<td>Mac address for this port group.</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of the port group.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Extra to set on this baremetal port group (repeat option to set multiple extras).</td>
</tr>
<tr>
<td>--mode MODE</td>
<td>Mode of the port group. For possible values, refer to</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Key/value property related to this port group’s configuration (repeat option to set multiple properties).</td>
</tr>
<tr>
<td>--support-standalone-ports</td>
<td>Ports that are members of this port group can be used as stand-alone ports.</td>
</tr>
<tr>
<td>--unsupport-standalone-ports</td>
<td>Ports that are members of this port group cannot be used as stand-alone ports.</td>
</tr>
</tbody>
</table>

### 11.86. BAREMETAL PORT GROUP SHOW

Show baremetal port group details.

**Usage:**

```bash
openstack baremetal port group show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [-p --print-empty]
    [-a --address]
    [-f fields <field> [<field> ...]]
    <id>
```
### Table 11.313. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;id&gt;</code></td>
<td>Uuid or name of the port group (or mac address if <code>--address</code> is specified).</td>
</tr>
</tbody>
</table>

### Table 11.314. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--address</code></td>
<td><code>&lt;id&gt;</code> is the mac address (instead of uuid or name) of the port group.</td>
</tr>
<tr>
<td><code>--fields &lt;field&gt; [...]]</code></td>
<td>One or more port group fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

### Table 11.315. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 11.316. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.317. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 11.318. Table formatter options
11.87. BAREMETAL PORT GROUP UNSET

Unset baremetal port group properties.

Usage:

```
openstack baremetal port group unset [-h] [--name] [--address]
    [--extra <key>] [--property <key>]
    <port group>
```

Table 11.319. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port group&gt;</td>
<td>Name or uuid of the port group.</td>
</tr>
</tbody>
</table>

Table 11.320. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name</td>
<td>Unset the name of the port group.</td>
</tr>
<tr>
<td>--address</td>
<td>Unset the address of the port group.</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset on this baremetal port group (repeat option to unset multiple extras).</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to unset on this baremetal port group (repeat option to unset multiple properties).</td>
</tr>
</tbody>
</table>

11.88. BAREMETAL PORT LIST
List baremetal ports.

**Usage:**

```
openstack baremetal port list [-h] [-f {csv,json,table,value,yaml}]  
  [-c COLUMN] 
  [--quote {all,minimal,none,nonnumeric}] 
  [--noindent] 
  [--max-width <integer>] 
  [--fit-width] 
  [--sort-column SORT_COLUMN] 
  [--sort-ascending | --sort-descending] 
  [--address <mac-address>] 
  [--node <node>] 
  [--port-group <port group>] 
  [--limit <limit>] 
  [--marker <port>] 
  [--sort <key>[<direction>]] 
  [--long | --fields <field> [...]]
```

Table 11.321. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--address &lt;mac-address&gt;</td>
<td>Only show information for the port with this mac address.</td>
</tr>
<tr>
<td>--node &lt;node&gt;</td>
<td>Only list ports of this node (name or uuid).</td>
</tr>
<tr>
<td>--port-group &lt;port group&gt;</td>
<td>Only list ports of this port group (name or uuid).</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of ports to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;port&gt;</td>
<td>Port uuid (for example, of the last port in the list from a previous request). Returns the list of ports after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[&lt;direction&gt;]</td>
<td>Sort output by specified port fields and directions (asc or desc) (default: asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about ports.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [...]]</td>
<td>One or more port fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>
### Value Summary

- `-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}`
  - The output format, defaults to table

- `-c COLUMN, --column COLUMN`
  - Specify the column(s) to include, can be repeated to show multiple columns

- `--sort-column SORT_COLUMN`
  - Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- `--sort-ascending`
  - Sort the column(s) in ascending order

- `--sort-descending`
  - Sort the column(s) in descending order

#### Table 11.323. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

#### Table 11.324. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table 11.325. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.89. BAREMETAL PORT SET

Set baremetal port properties.

**Usage:**
openstack baremetal port set [-h] [--node <uuid>] [--address <address>]
  [--extra <key=value>]
  [--port-group <uuid>]
  [--local-link-connection <key=value>]
  [--pxe-enabled | --pxe-disabled]
  [--physical-network <physical network>]
  [--is-smartnic]
  <port>

Table 11.326. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Uuid of the port</td>
</tr>
</tbody>
</table>

Table 11.327. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Set uuid of the node that this port belongs to</td>
</tr>
<tr>
<td>--address &lt;address&gt;</td>
<td>Set mac address for this port</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Extra to set on this baremetal port (repeat option to set multiple extras)</td>
</tr>
<tr>
<td>--port-group &lt;uuid&gt;</td>
<td>Set uuid of the port group that this port belongs to.</td>
</tr>
<tr>
<td>--local-link-connection &lt;key=value&gt;</td>
<td>Key/value metadata describing local link connection information. Valid keys are switch_info, switch_id, port_id and hostname. The keys switch_id and port_id are required. In case of a Smart NIC port, the required keys are port_id and hostname. Argument can be specified multiple times.</td>
</tr>
<tr>
<td>--pxe-enabled</td>
<td>Indicates that this port should be used when pxe booting this node (default)</td>
</tr>
<tr>
<td>--pxe-disabled</td>
<td>Indicates that this port should not be used when pxe booting this node</td>
</tr>
<tr>
<td>--physical-network &lt;physical network&gt;</td>
<td>Set the name of the physical network to which this port is connected.</td>
</tr>
<tr>
<td>--is-smartnic</td>
<td>Set port to be smart nic port</td>
</tr>
</tbody>
</table>

11.90. BAREMETAL PORT SHOW
Show baremetal port details.

Usage:


Table 11.328. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Uuid of the port (or mac address if --address is specified).</td>
</tr>
</tbody>
</table>

Table 11.329. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--address</td>
<td>&lt;id&gt; is the mac address (instead of the uuid) of the port.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more port fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

Table 11.330. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.331. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.332. Shell formatter options
Table 11.333. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.91. BAREMETAL PORT UNSET

Unset baremetal port properties.

Usage:


Table 11.334. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Uuid of the port.</td>
</tr>
</tbody>
</table>

Table 11.335. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset on this baremetal port (repeat option to unset multiple extras)</td>
</tr>
<tr>
<td>--port-group</td>
<td>Remove port from the port group</td>
</tr>
<tr>
<td>--physical-network</td>
<td>Unset the physical network on this baremetal port.</td>
</tr>
</tbody>
</table>
### 11.92. BAREMETAL VOLUME CONNECTOR CREATE

Create a new baremetal volume connector.

**Usage:**

```
openstack baremetal volume connector create [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty] --node <uuid> --type <type> --connector-id <connector id>
  [--uuid <uuid>] [--extra <key=value>]
```

**Table 11.336. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Uuid of the node that this volume connector belongs to.</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>Type of the volume connector. can be iqn, ip, mac, wwnn, wwpn, port, portgroup.</td>
</tr>
<tr>
<td>--connector-id &lt;connector id&gt;</td>
<td>Id of the volume connector in the specified type. for example, the iSCSI initiator IQN for the node if the type is iqn.</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Uuid of the volume connector.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
</tbody>
</table>

**Table 11.337. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### Table 11.338. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 11.339. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.340. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 11.93. BAREMETAL VOLUME CONNECTOR DELETE

Unregister baremetal volume connector(s).

**Usage:**

```
openstack baremetal volume connector delete [-h] <volume connector> [<volume connector> ...]
```

### Table 11.341. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume connector&gt;</td>
<td>Uuid(s) of the volume connector(s) to delete.</td>
</tr>
</tbody>
</table>
Table 11.342. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.94. BAREMETAL VOLUME CONNECTOR LIST

List baremetal volume connectors.

Usage:

```
```

Table 11.343. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;node&gt;</td>
<td>Only list volume connectors of this node (name or UUID).</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of volume connectors to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;volume connector&gt;</td>
<td>Volume connector uuid (for example, of the last volume connector in the list from a previous request). Returns the list of volume connectors after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by specified volume connector fields and directions (asc or desc) (default:asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about volume connectors.</td>
</tr>
</tbody>
</table>
### Table 11.344. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more volume connector fields. Only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 11.345. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 11.346. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.347. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 11.95. BAREMETAL VOLUME CONNECTOR SET

Set baremetal volume connector properties.

**Usage:**

```bash
openstack baremetal volume connector set [-h] [--node <uuid>] [--type <type>] [--connector-id <connector id>] [--extra <key=value>] <volume connector>
```

**Table 11.348. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;volume connector&gt;</code></td>
<td>Uuid of the volume connector.</td>
</tr>
</tbody>
</table>

**Table 11.349. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--node &lt;uuid&gt;</code></td>
<td>Uuid of the node that this volume connector belongs to.</td>
</tr>
<tr>
<td><code>--type &lt;type&gt;</code></td>
<td>Type of the volume connector. can be <code>iqn, ip, mac, wwnn, wwpn, port, portgroup</code>.</td>
</tr>
<tr>
<td><code>--connector-id &lt;connector id&gt;</code></td>
<td>Id of the volume connector in the specified type.</td>
</tr>
<tr>
<td><code>--extra &lt;key=value&gt;</code></td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
</tbody>
</table>

### 11.96. BAREMETAL VOLUME CONNECTOR SHOW

Show baremetal volume connector details.

**Usage:**

```bash
openstack baremetal volume connector show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent]
```
Table 11.350. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Uuid of the volume connector.</td>
</tr>
</tbody>
</table>

Table 11.351. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more volume connector fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

Table 11.352. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.353. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.354. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 11.355. Table formatter options
11.97. BAREMETAL VOLUME CONNECTOR UNSET

Unset baremetal volume connector properties.

**Usage:**

```
openstack baremetal volume connector unset [-h] [--extra <key>] <volume connector>
```

**Table 11.356. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume connector&gt;</td>
<td>Uuid of the volume connector.</td>
</tr>
</tbody>
</table>

**Table 11.357. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset (repeat option to unset multiple extras)</td>
</tr>
</tbody>
</table>

11.98. BAREMETAL VOLUME TARGET CREATE

Create a new baremetal volume target.

**Usage:**

```
```
Table 11.358. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Uuid of the node that this volume target belongs to.</td>
</tr>
<tr>
<td>--type &lt;volume type&gt;</td>
<td>Type of the volume target, e.g. iscsi, fibre_channel.</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Key/value property related to the type of this volume target. Can be specified multiple times.</td>
</tr>
<tr>
<td>--boot-index &lt;boot index&gt;</td>
<td>Boot index of the volume target.</td>
</tr>
<tr>
<td>--volume-id &lt;volume id&gt;</td>
<td>Id of the volume associated with this target.</td>
</tr>
<tr>
<td>--uuid &lt;uuid&gt;</td>
<td>Uuid of the volume target.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
</tbody>
</table>

Table 11.359. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 11.360. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.361. Shell formatter options
11.99. BAREMETAL VOLUME TARGET DELETE

Unregister baremetal volume target(s).

Usage:

```
openstack baremetal volume target delete [-h] <volume target> [...]
```

Table 11.363. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume target&gt;</td>
<td>Uuid(s) of the volume target(s) to delete.</td>
</tr>
</tbody>
</table>

Table 11.364. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

11.100. BAREMETAL VOLUME TARGET LIST

List baremetal volume targets.

Usage:

```
openstack baremetal volume target list [-h]
```
Table 11.365. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;node&gt;</td>
<td>Only list volume targets of this node (name or uuid).</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of volume targets to return per request, 0 for no limit. Default is the maximum number used by the Baremetal API Service.</td>
</tr>
<tr>
<td>--marker &lt;volume target&gt;</td>
<td>Volume target uuid (for example, of the last volume target in the list from a previous request). Returns the list of volume targets after this UUID.</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[&lt;direction&gt;]</td>
<td>Sort output by specified volume target fields and directions (asc or desc) (default:asc). Multiple fields and directions can be specified, separated by comma.</td>
</tr>
<tr>
<td>--long</td>
<td>Show detailed information about volume targets.</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; ...]</td>
<td>One or more volume target fields. only these fields will be fetched from the server. Can not be used when --long is specified.</td>
</tr>
</tbody>
</table>

Table 11.366. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 11.367. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 11.368. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 11.369. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

11.101. BAREMETAL VOLUME TARGET SET

Set baremetal volume target properties.

Usage:

```
openstack baremetal volume target set [-h] [-node <uuid>] [-type <volume type>] [-property <key=value>] [-boot-index <boot index>]
```
Table 11.370. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume target&gt;</td>
<td>Uuid of the volume target.</td>
</tr>
</tbody>
</table>

Table 11.371. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--node &lt;uuid&gt;</td>
<td>Uuid of the node that this volume target belongs to.</td>
</tr>
<tr>
<td>--type &lt;volume type&gt;</td>
<td>Type of the volume target, e.g. iscsi, fibre_channel.</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Key/value property related to the type of this volume target. Can be specified multiple times.</td>
</tr>
<tr>
<td>--boot-index &lt;boot index&gt;</td>
<td>Boot index of the volume target.</td>
</tr>
<tr>
<td>--volume-id &lt;volume id&gt;</td>
<td>Id of the volume associated with this target.</td>
</tr>
<tr>
<td>--extra &lt;key=value&gt;</td>
<td>Record arbitrary key/value metadata. can be specified multiple times.</td>
</tr>
</tbody>
</table>

11.102. BAREMETAL VOLUME TARGET SHOW

Show baremetal volume target details.

Usage:

```
oopenstack baremetal volume target show [-h]
 [\-f \{json,shell,table,value,yaml\}]
 [-c COLUMN] [-N oindent]
 [\-prefix PREFIX]
 [\-max-width \<integer\>]
 [\-fit-width] [-print-empty]
 [\-fields \<field\> [\<field\> ...]]
 <id>
```

Table 11.372. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Uuid of the volume target.</td>
</tr>
</tbody>
</table>

### Table 11.373. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fields &lt;field&gt; [&lt;field&gt; …​]</td>
<td>One or more volume target fields. only these fields will be fetched from the server.</td>
</tr>
</tbody>
</table>

### Table 11.374. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 11.375. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 11.376. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 11.377. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
11.103. BAREMETAL VOLUME TARGET UNSET

Unset baremetal volume target properties.

Usage:

```
openstack baremetal volume target unset [-h] [--extra <key>] [--property <key>] <volume target>
```

Table 11.378. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;volume target&gt;</code></td>
<td>Uuid of the volume target.</td>
</tr>
</tbody>
</table>

Table 11.379. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--extra &lt;key&gt;</td>
<td>Extra to unset (repeat option to unset multiple extras)</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to unset on this baremetal volume target (repeat option to unset multiple properties).</td>
</tr>
</tbody>
</table>
CHAPTER 12. CA

This chapter describes the commands under the ca command.

12.1. CA GET

Retrieve a CA by providing its URI.

Usage:

```
```

Table 12.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the ca.</td>
</tr>
</tbody>
</table>

Table 12.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 12.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 12.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 12.5. Shell formatter options
Table 12.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

12.2. CA LIST

List CAs.

Usage:

```
```

Table 12.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit LIMIT, -l LIMIT</td>
<td>Specify the limit to the number of items to list per page (default: 10; maximum: 100)</td>
</tr>
<tr>
<td>--offset OFFSET, -o OFFSET</td>
<td>Specify the page offset (default: 0)</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>Specify the ca name (default: none)</td>
</tr>
</tbody>
</table>

Table 12.8. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 12.9. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 12.10. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 12.11. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 13. CATALOG

This chapter describes the commands under the catalog command.

13.1. CATALOG LIST

List services in the service catalog

Usage:


Table 13.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 13.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 13.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
### Table 13.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 13.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Service to display (type or name)</td>
</tr>
</tbody>
</table>

### Table 13.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 13.8. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 13.2. CATALOG SHOW

Display service catalog details

Usage:

```
```

Table 13.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Service to display (type or name)</td>
</tr>
</tbody>
</table>

Table 13.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 13.9. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 13.10. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 13.11. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 14. CODE

This chapter describes the commands under the `code` command.

14.1. CODE SOURCE CONTENT SHOW

Show workflow definition.

Usage:

```
openstack code source content show [-h] [--namespace [NAMESPACE]] identifier
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Code source id or name.</td>
</tr>
</tbody>
</table>

Table 14.1. Positional arguments

Table 14.2. Command arguments

```
-h, --help
--namespace [NAMESPACE]
```

Show this help message and exit

Namespace to get the code source from.

14.2. CODE SOURCE CREATE

Create new code source.

Usage:

```
```

Table 14.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Code source name.</td>
</tr>
<tr>
<td>content</td>
<td>Code source content file.</td>
</tr>
</tbody>
</table>
### Table 14.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the code source within.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag the code source will be marked as &quot;public&quot;.</td>
</tr>
</tbody>
</table>

### Table 14.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 14.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 14.7. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 14.8. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
14.3. CODE SOURCE DELETE

Delete workflow.

Usage:

```
openstack code source delete [-h] [--namespace [NAMESPACE]]
identifier [identifier ...]
```

Table 14.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Code source name or id (can be repeated multiple times).</td>
</tr>
</tbody>
</table>

Table 14.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to delete the code source(s) from.</td>
</tr>
</tbody>
</table>

14.4. CODE SOURCE LIST

List all workflows.

Usage:

```
openstack code source list [-h] [-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent] [--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--marker [MARKER]] [--limit [LIMIT]]
[--sort_keys [SORT_KEYS]]
[--sort_dirs [SORT_DIRS]] [--filter FILTERS]
```

Table 14.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
</tbody>
</table>

**Table 14.12. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 14.13. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 14.14. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 14.15. Table formatter options**
### 14.5. CODE SOURCE SHOW

Show specific code source.

**Usage:**

```
```

**Table 14.16. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Code source id or name.</td>
</tr>
</tbody>
</table>

**Table 14.17. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to get the code source from.</td>
</tr>
</tbody>
</table>

**Table 14.18. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 14.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 14.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 14.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

14.6. CODE SOURCE UPDATE

Update workflow.

Usage:

```
```

Table 14.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Code source identifier (name or id).</td>
</tr>
<tr>
<td>content</td>
<td>Code source content</td>
</tr>
</tbody>
</table>

Table 14.23. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--id ID</td>
<td>Workflow id.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the workflow.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag workflow will be marked as &quot;public&quot;.</td>
</tr>
</tbody>
</table>

Table 14.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 14.25. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 14.26. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 14.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 15. COE

This chapter describes the commands under the `coe` command.

15.1. COE CA ROTATE

Rotate the CA certificate for cluster to revoke access.

Usage:

```
openstack coe ca rotate [-h] <cluster>
```

Table 15.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster</td>
</tr>
</tbody>
</table>

Table 15.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

15.2. COE CA SHOW

Show details about the CA certificate for a cluster.

Usage:

```
openstack coe ca show [-h] <cluster>
```

Table 15.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster</td>
</tr>
</tbody>
</table>

Table 15.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

15.3. COE CA SIGN

Generate the CA certificate for a cluster.
Usage:

```bash
openstack coe ca sign [-h] <cluster> <csr>
```

Table 15.5. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster</td>
</tr>
<tr>
<td>&lt;csr&gt;</td>
<td>File path of csr file to send to magnum to get signed.</td>
</tr>
</tbody>
</table>

Table 15.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

15.4. COE CLUSTER CONFIG

Get Configuration for a Cluster

Usage:

```bash
```

Table 15.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>The name or uuid of cluster to update</td>
</tr>
</tbody>
</table>

Table 15.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--dir &lt;dir&gt;</td>
<td>Directory to save the certificate and config files.</td>
</tr>
<tr>
<td>--force</td>
<td>Overwrite files if existing.</td>
</tr>
<tr>
<td>--output-certs</td>
<td>Output certificates in separate files.</td>
</tr>
</tbody>
</table>
15.5. COE CLUSTER CREATE

Create a cluster

Usage:

```
openstack coe cluster create [-h] --cluster-template <cluster-template>
    [--discovery-url <discovery-url>]
    [--docker-volume-size <docker-volume-size>]
    [--labels <KEY1=VALUE1,KEY2=VALUE2;KEY3=VALUE3...>]
    [--keypair <keypair>]
    [--master-count <master-count>]
    [--node-count <node-count>]
    [--timeout <timeout>]
    [--master-flavor <master-flavor>]
    [--flavor <flavor>]
    [--fixed-network <fixed-network>]
    [--fixed-subnet <fixed-subnet>]
    [--floating-ip-enabled]
    [--floating-ip-disabled] [--merge-labels]
    [--master-lb-enabled]
    <name>
```

Table 15.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the cluster to create.</td>
</tr>
</tbody>
</table>

Table 15.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--cluster-template &lt;cluster-template&gt;</td>
<td>Id or name of the cluster template.</td>
</tr>
<tr>
<td>--discovery-url &lt;discovery-url&gt;</td>
<td>Specifies custom delivery url for node discovery.</td>
</tr>
<tr>
<td>--docker-volume-size &lt;docker-volume-size&gt;</td>
<td>The size in gb for the docker volume to use.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--labels &lt;KEY1=VALUE1,KEY2=VALUE2,KEY3=VALUE3...&gt;</td>
<td>Arbitrary labels in the form of key=value pairs to associate with a cluster template. May be used multiple times.</td>
</tr>
<tr>
<td>--keypair &lt;keypair&gt;</td>
<td>Uuid or name of the keypair to use.</td>
</tr>
<tr>
<td>--master-count &lt;master-count&gt;</td>
<td>The number of master nodes for the cluster.</td>
</tr>
<tr>
<td>--node-count &lt;node-count&gt;</td>
<td>The cluster node count.</td>
</tr>
<tr>
<td>--timeout &lt;timeout&gt;</td>
<td>The timeout for cluster creation time. The default is 60 minutes.</td>
</tr>
<tr>
<td>--master-flavor &lt;master-flavor&gt;</td>
<td>The nova flavor name or uuid to use when launching the master node of the Cluster.</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>The nova flavor name or uuid to use when launching the Cluster.</td>
</tr>
<tr>
<td>--fixed-network &lt;fixed-network&gt;</td>
<td>The private neutron network name to connect to this Cluster template.</td>
</tr>
<tr>
<td>--fixed-subnet &lt;fixed-subnet&gt;</td>
<td>The private neutron subnet name to connect to cluster.</td>
</tr>
<tr>
<td>--floating-ip-enabled</td>
<td>Indicates whether created clusters should have a floating ip.</td>
</tr>
<tr>
<td>--floating-ip-disabled</td>
<td>Disables floating ip creation on the new cluster</td>
</tr>
<tr>
<td>--merge-labels</td>
<td>The labels provided will be merged with the labels configured in the specified cluster template.</td>
</tr>
<tr>
<td>--master-lb-enabled</td>
<td>Indicates whether created clusters should have a load balancer for API.</td>
</tr>
</tbody>
</table>

### 15.6. COE CLUSTER DELETE

Delete a cluster

**Usage:**

```
openstack coe cluster delete [-h] <cluster> [<cluster> ...]
```

Table 15.11. Positional arguments
**15.7. COE CLUSTER LIST**

List clusters

**Usage:**

```
openstack coe cluster list [-h] [-f {csv, json, table, value, yaml}] 
   [-c COLUMN] [-quote {all, minimal, none, nonnumeric}] 
   [-noindent] [-max-width <integer>] 
   [-fit-width] [-print-empty] 
   [--sort-column SORT_COLUMN] 
   [--sort-ascending | --sort-descending] 
   [--limit <limit>] 
   [--sort-key <sort-key>] 
   [--sort-dir <sort-dir>]
```

**Table 15.13. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of clusters to return</td>
</tr>
<tr>
<td>--sort-key &lt;sort-key&gt;</td>
<td>Column to sort results by</td>
</tr>
<tr>
<td>--sort-dir &lt;sort-dir&gt;</td>
<td>Direction to sort. &quot;asc&quot; or &quot;desc&quot;.</td>
</tr>
</tbody>
</table>

**Table 15.14. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
| -f {csv, json, table, value, yaml}, --format 
   {csv, json, table, value, yaml} | The output format, defaults to table                  |
| -c COLUMN, --column COLUMN        | Specify the column(s) to include, can be repeated to show multiple columns |
### Table 15.15. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 15.16. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 15.17. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 15.8. COE CLUSTER RESIZE

Resize a Cluster

Usage:

```bash
openstack coe cluster resize [-h] [--nodes-to-remove <Server UUID>] [--nodegroup <nodegroup>] <cluster> node_count
```

### Table 15.18. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>The name or uuid of cluster to update</td>
</tr>
<tr>
<td>node_count</td>
<td>Desired node count of the cluster.</td>
</tr>
</tbody>
</table>

**Table 15.19. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--nodes-to-remove &lt;Server UUID&gt;</td>
<td>Server id of the nodes to be removed. repeat to add more server ID</td>
</tr>
<tr>
<td>--nodegroup &lt;nodegroup&gt;</td>
<td>The name or uuid of the nodegroup of current cluster.</td>
</tr>
</tbody>
</table>

**15.9. COE CLUSTER SHOW**

Show a Cluster

**Usage:**

```
```

**Table 15.20. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster to show.</td>
</tr>
</tbody>
</table>

**Table 15.21. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 15.22. Output formatter options**
Table 15.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 15.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 15.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

15.10. COE CLUSTER TEMPLATE CREATE

Create a Cluster Template.

Usage:

```bash
```
--coe <coe> --image <image>
--external-network
<external-network>
[--keypair <keypair>]
[--fixed-network <fixed-network>]
[--fixed-subnet <fixed-subnet>]
[--network-driver <network-driver>]
[--volume-driver <volume-driver>]
[--dns-nameserver <dns-nameserver>]
[--flavor <flavor>]
[--master-flavor <master-flavor>]
[--docker-volume-size <docker-volume-size>]
[--docker-storage-driver <docker-storage-driver>]
[--http-proxy <http-proxy>]
[--https-proxy <https-proxy>]
[--no-proxy <no-proxy>]
[--labels <KEY1=VALUE1,KEY2=VALUE2;KEY3=VALUE3...>]
[--tls-disabled] [--public]
[--registry-enabled]
[--server-type <server-type>]
[--master-lb-enabled]
[--floating-ip-enabled]
[--floating-ip-disabled]
[--hidden] [--visible]
[--tags <--tags tag1 --tags tag2,tag3>]
<name>

Table 15.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the cluster template to create.</td>
</tr>
</tbody>
</table>

Table 15.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--coe &lt;coe&gt;</td>
<td>Specify the container orchestration engine to use.</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>The name or uuid of the base image to customize for the Cluster.</td>
</tr>
<tr>
<td>--external-network &lt;external-network&gt;</td>
<td>The external neutron network name or uuid to connect to this Cluster Template.</td>
</tr>
<tr>
<td>--keypair &lt;keypair&gt;</td>
<td>The name or uuid of the ssh keypair to load into the Cluster nodes.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fixed-network &lt;fixed-network&gt;</td>
<td>The private neutron network name to connect to this Cluster model.</td>
</tr>
<tr>
<td>--fixed-subnet &lt;fixed-subnet&gt;</td>
<td>The private neutron subnet name to connect to cluster.</td>
</tr>
<tr>
<td>--network-driver &lt;network-driver&gt;</td>
<td>The network driver name for instantiating container networks.</td>
</tr>
<tr>
<td>--volume-driver &lt;volume-driver&gt;</td>
<td>The volume driver name for instantiating container volume.</td>
</tr>
<tr>
<td>--dns-nameserver &lt;dns-nameserver&gt;</td>
<td>The dns nameserver to use for this cluster template.</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>The nova flavor name or uuid to use when launching the Cluster.</td>
</tr>
<tr>
<td>--master-flavor &lt;master-flavor&gt;</td>
<td>The nova flavor name or uuid to use when launching the master node of the Cluster.</td>
</tr>
<tr>
<td>--docker-volume-size &lt;docker-volume-size&gt;</td>
<td>Specify the number of size in gb for the docker volume to use.</td>
</tr>
<tr>
<td>--docker-storage-driver &lt;docker-storage-driver&gt;</td>
<td>Select a docker storage driver. supported: devicemapper, overlay, overlay2. Default: overlay2</td>
</tr>
<tr>
<td>--http-proxy &lt;http-proxy&gt;</td>
<td>The http_proxy address to use for nodes in cluster.</td>
</tr>
<tr>
<td>--https-proxy &lt;https-proxy&gt;</td>
<td>The https_proxy address to use for nodes in cluster.</td>
</tr>
<tr>
<td>--no-proxy &lt;no-proxy&gt;</td>
<td>The no_proxy address to use for nodes in cluster.</td>
</tr>
<tr>
<td>--labels</td>
<td>Arbitrary labels in the form of key=value pairs to associate with a cluster template. May be used multiple times.</td>
</tr>
<tr>
<td>--tls-disabled</td>
<td>Disable tls in the cluster.</td>
</tr>
<tr>
<td>--public</td>
<td>Make cluster template public.</td>
</tr>
<tr>
<td>--registry-enabled</td>
<td>Enable docker registry in the cluster</td>
</tr>
<tr>
<td>--server-type &lt;server-type&gt;</td>
<td>Specify the server type to be used for example vm. for this release default server type will be vm.</td>
</tr>
</tbody>
</table>
--master-lb-enabled
Indicates whether created clusters should have a load balancer for master nodes or not.

--floating-ip-enabled
Indicates whether created clusters should have a floating ip.

--floating-ip-disabled
Disables floating ip creation on the new cluster

--hidden
Indicates the cluster template should be hidden.

--visible
Indicates the cluster template should be visible.

--tags tags tag1 --tags tag2,tag3
Tags to be added to the cluster template.

Table 15.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 15.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 15.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 15.31. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 15.11. COE CLUSTER TEMPLATE DELETE

Delete a Cluster Template.

**Usage:**

```bash
openstack coe cluster template delete [-h]
<cluster-templates>
[<cluster-templates> ...]
```

**Table 15.32. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster-templates&gt;</td>
<td>Id or name of the (cluster template)s to delete.</td>
</tr>
</tbody>
</table>

**Table 15.33. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 15.12. COE CLUSTER TEMPLATE LIST

List Cluster Templates.

**Usage:**

```bash
openstack coe cluster template list [-h]
 [-f {csv,json,table,value,yaml}]
 [-c COLUMN]
 [--quote {all,minimal,none,nonnumeric}]
 [-noindent]
 [-max-width <integer>]
 [-fit-width] [-print-empty]
 [--sort-column SORT_COLUMN]
 [--sort-ascending | --sort-descending]
 [-limit <limit>]
```
Table 15.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of cluster templates to return</td>
</tr>
<tr>
<td>--sort-key &lt;sort-key&gt;</td>
<td>Column to sort results by</td>
</tr>
<tr>
<td>--sort-dir &lt;sort-dir&gt;</td>
<td>Direction to sort. &quot;asc&quot; or &quot;desc&quot;.</td>
</tr>
<tr>
<td>--fields &lt;fields&gt;</td>
<td>Comma-separated list of fields to display. available fields: uuid, name, coe, image_id, public, link, apiserver_port, server_type, tls_disabled, registry_enabled</td>
</tr>
</tbody>
</table>

Table 15.35. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT.COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 15.36. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 15.37. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 15.38. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**15.13. COE CLUSTER TEMPLATE SHOW**

Show a Cluster Template.

**Usage:**

```
openstack coe cluster template show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    <cluster-template>
```

**Table 15.39. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster-template&gt;</td>
<td>Id or name of the cluster template to show.</td>
</tr>
</tbody>
</table>

**Table 15.40. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 15.41. Output formatter options**
### Table 15.42. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 15.43. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 15.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width, implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 15.14. COE CLUSTER TEMPLATE UPDATE

Update a Cluster Template.

**Usage:**

```
openstack coe cluster template update [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  <cluster-template> <op>
  <path=value> [<path=value> ...]
```
### Table 15.45. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;cluster-template&gt;</code></td>
<td>The name or uuid of cluster template to update</td>
</tr>
<tr>
<td><code>&lt;op&gt;</code></td>
<td>Operations: one of <code>add</code>, <code>replace</code> or <code>remove</code></td>
</tr>
<tr>
<td><code>&lt;path=value&gt;</code></td>
<td>Attributes to add/replace or remove (only path is necessary on remove)</td>
</tr>
</tbody>
</table>

### Table 15.46. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 15.47. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 15.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 15.49. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 15.50. Table formatter options
15.15. COE CLUSTER UPDATE

Update a Cluster

Usage:

```
openstack coe cluster update [-h] [--rollback]
<cluster> <op> <path=value>
[<path=value> ...]
```

Table 15.51. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;cluster&gt;</code></td>
<td>The name or uuid of cluster to update</td>
</tr>
<tr>
<td><code>&lt;op&gt;</code></td>
<td>Operations: one of <code>add</code>, <code>replace</code> or <code>remove</code></td>
</tr>
<tr>
<td><code>&lt;path=value&gt;</code></td>
<td>Attributes to add/replace or remove (only path is necessary on remove)</td>
</tr>
</tbody>
</table>

Table 15.52. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--rollback</code></td>
<td>Rollback cluster on update failure.</td>
</tr>
</tbody>
</table>

15.16. COE CLUSTER UPGRADE

Upgrade a Cluster

Usage:
openstack coe cluster upgrade [-h] [--max-batch-size <max_batch_size>]
  [--nodegroup <nodegroup>]
  <cluster> cluster_template

Table 15.53. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>The name or uuid of cluster to update</td>
</tr>
<tr>
<td>cluster_template</td>
<td>The new cluster template id will be upgraded to.</td>
</tr>
</tbody>
</table>

Table 15.54. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--max-batch-size &lt;max_batch_size&gt;</td>
<td>The max batch size for upgrading each time.</td>
</tr>
<tr>
<td>--nodegroup &lt;nodegroup&gt;</td>
<td>The name or uuid of the nodegroup of current cluster.</td>
</tr>
</tbody>
</table>

15.17. COE NODEGROUP CREATE

Create a nodegroup

Usage:

openstack coe nodegroup create [-h]
  [--docker-volume-size <docker-volume-size>]
  [--labels <KEY1=VALUE1,KEY2=VALUE2;KEY3=VALUE3...>]
  [--node-count <node-count>]
  [-min-nodes <min-nodes>]
  [-max-nodes <max-nodes>]
  [--role <role>] [-image <image>]
  [--flavor <flavor>] [--merge-labels]
  <cluster> <name>

Table 15.55. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Name of the nodegroup to create.</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the nodegroup to create.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--docker-volume-size &lt;docker-volume-size&gt;</td>
<td>The size in gb for the docker volume to use.</td>
</tr>
<tr>
<td>--labels &lt;KEY1=VALUE1,KEY2=VALUE2;KEY3=VALUE3...&gt;</td>
<td>Arbitrary labels in the form of key=value pairs to associate with a nodegroup. May be used multiple times.</td>
</tr>
<tr>
<td>--node-count &lt;node-count&gt;</td>
<td>The nodegroup node count.</td>
</tr>
<tr>
<td>--min-nodes &lt;min-nodes&gt;</td>
<td>The nodegroup minimum node count.</td>
</tr>
<tr>
<td>--max-nodes &lt;max-nodes&gt;</td>
<td>The nodegroup maximum node count.</td>
</tr>
<tr>
<td>--role &lt;role&gt;</td>
<td>The role of the nodegroup</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>The name or uuid of the base image to customize for the NodeGroup.</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>The nova flavor name or uuid to use when launching the nodes in this NodeGroup.</td>
</tr>
<tr>
<td>--merge-labels</td>
<td>The labels provided will be merged with the labels configured in the specified cluster.</td>
</tr>
</tbody>
</table>

### 15.18. COE NODEGROUP DELETE

Delete a nodegroup

**Usage:**

```
openstack coe nodegroup delete [-h] <cluster> <nodegroup> [<nodegroup> ...]
```

**Table 15.57. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster where the nodegroup(s) belong(s).</td>
</tr>
<tr>
<td>&lt;nodegroup&gt;</td>
<td>Id or name of the nodegroup(s) to delete.</td>
</tr>
</tbody>
</table>
15.19. COE NODEGROUP LIST

List nodegroups

Usage:

```bash
```

Table 15.59. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster where the nodegroup belongs.</td>
</tr>
</tbody>
</table>

Table 15.60. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of nodegroups to return</td>
</tr>
<tr>
<td>--sort-key &lt;sort-key&gt;</td>
<td>Column to sort results by</td>
</tr>
<tr>
<td>--sort-dir &lt;sort-dir&gt;</td>
<td>Direction to sort. &quot;asc&quot; or &quot;desc&quot;.</td>
</tr>
<tr>
<td>--role &lt;role&gt;</td>
<td>List the nodegroups in the cluster with this role</td>
</tr>
</tbody>
</table>

Table 15.61. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**Table 15.62. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 15.63. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 15.64. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0, Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 15.20. COE NODEGROUP SHOW

Show a nodegroup

**Usage:**

```
openstack coe nodegroup show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX]
```
Table 15.65. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;cluster&gt;</code></td>
<td>Id or name of the cluster where the nodegroup belongs.</td>
</tr>
<tr>
<td><code>&lt;nodegroup&gt;</code></td>
<td>Id or name of the nodegroup to show.</td>
</tr>
</tbody>
</table>

Table 15.66. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 15.67. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 15.68. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 15.69. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 15.70. Table formatter options
Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

---

Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.

---

Print empty table if there is no data to show.

---

15.21. COE NODEGROUP UPDATE

Update a Nodegroup

Usage:

openstack coe nodegroup update [-h]

<cluster> <nodegroup> <op> <path=value>
[<path=value> ...]

Table 15.71. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;cluster&gt;</td>
<td>Id or name of the cluster where the nodegroup belongs.</td>
</tr>
<tr>
<td>&lt;nodegroup&gt;</td>
<td>The name or uuid of cluster to update</td>
</tr>
<tr>
<td>&lt;op&gt;</td>
<td>Operations: one of add, replace or remove</td>
</tr>
<tr>
<td>&lt;path=value&gt;</td>
<td>Attributes to add/replace or remove (only path is necessary on remove)</td>
</tr>
</tbody>
</table>

Table 15.72. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

15.22. COE QUOTAS CREATE

Create a quota.

Usage:
openstack coe quotas create [-h] --project-id <project-id> --resource <resource> [--hard-limit <hard-limit>]

Table 15.73. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-id &lt;project-id&gt;</td>
<td>Project id</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Resource name.</td>
</tr>
<tr>
<td>--hard-limit &lt;hard-limit&gt;</td>
<td>Max resource limit (default: hard-limit=1)</td>
</tr>
</tbody>
</table>

15.23. COE QUOTAS DELETE

Delete specified resource quota.

Usage:

openstack coe quotas delete [-h] --project-id <project-id> --resource <resource>

Table 15.74. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-id &lt;project-id&gt;</td>
<td>Project id</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Resource name.</td>
</tr>
</tbody>
</table>

15.24. COE QUOTAS LIST

Print a list of available quotas.

Usage:


Table 15.75. Command arguments
15.25. COE QUOTAS SHOW

Show details about the given project resource quota.

**Usage:**

```bash
openstack coe quotas show [-h] --project-id <project-id> --resource <resource>
```

**Table 15.76. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-id &lt;id&gt;</td>
<td>Project id</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Resource name.</td>
</tr>
</tbody>
</table>

15.26. COE QUOTAS UPDATE

Update information about the given project resource quota.

**Usage:**

```bash
openstack coe quotas update [-h] --project-id <project-id> --resource <resource> [--hard-limit <hard-limit>]
```

**Table 15.77. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-id &lt;id&gt;</td>
<td>Project id</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Resource name.</td>
</tr>
</tbody>
</table>
15.27. COE SERVICE LIST

Print a list of Magnum services.

Usage:


Table 15.78. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 15.79. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
### Table 15.80. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 15.81. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 15.82. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 15.28. COE STATS LIST

Show stats for the given project_id

**Usage:**

```
openstack coe stats list [-h] <project>
```

### Table 15.83. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;project&gt;</code></td>
<td>Project id</td>
</tr>
</tbody>
</table>

### Table 15.84. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
CHAPTER 16. COMMAND

This chapter describes the commands under the command command.

16.1. COMMAND LIST

List recognized commands by group

Usage:


Table 16.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--group &lt;group-keyword&gt;</td>
<td>Show commands filtered by a command group, for example: identity, volume, compute, image, network and other keywords</td>
</tr>
</tbody>
</table>

Table 16.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 16.3. CSV formatter options
**Table 16.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 16.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
This chapter describes the commands under the `complete` command.

17.1. COMPLETE

print bash completion command

Usage:

```
openstack complete [-h] [--name <command_name>] [--shell <shell>]
```

Table 17.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;command_name&gt;</td>
<td>Command name to support with command completion</td>
</tr>
<tr>
<td>--shell &lt;shell&gt;</td>
<td>Shell being used. use none for data only (default: bash)</td>
</tr>
</tbody>
</table>
CHAPTER 18. COMPUTE

This chapter describes the commands under the `compute` command.

18.1. COMPUTE AGENT CREATE

Create compute agent. The compute agent functionality is hypervisor specific and is only supported by the XenAPI hypervisor driver. It was removed from nova in the 23.0.0 (Wallaby) release.

Usage:


Table 18.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;os&gt;</code></td>
<td>Type of os</td>
</tr>
<tr>
<td><code>&lt;architecture&gt;</code></td>
<td>Type of architecture</td>
</tr>
<tr>
<td><code>&lt;version&gt;</code></td>
<td>Version</td>
</tr>
<tr>
<td><code>&lt;url&gt;</code></td>
<td>Url</td>
</tr>
<tr>
<td><code>&lt;md5hash&gt;</code></td>
<td>Md5 hash</td>
</tr>
<tr>
<td><code>&lt;hypervisor&gt;</code></td>
<td>Type of hypervisor</td>
</tr>
</tbody>
</table>

Table 18.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 18.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
-c COLUMN, --column COLUMN

Specify the column(s) to include, can be repeated to show multiple columns

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 18.2. COMPUTE AGENT DELETE

Delete compute agent(s). The compute agent functionality is hypervisor specific and is only supported by the XenAPI hypervisor driver. It was removed from nova in the 23.0.0 (Wallaby) release.

Usage:

```
openstack compute agent delete [-h] <id> [<id> ...]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Id of agent(s) to delete</td>
</tr>
</tbody>
</table>
**18.3. COMPUTE AGENT LIST**

List compute agents. The compute agent functionality is hypervisor specific and is only supported by the XenAPI hypervisor driver. It was removed from nova in the 23.0.0 (Wallaby) release.

**Usage:**

```
```

**Table 18.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 18.9. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--hypervisor</td>
<td>Type of hypervisor</td>
</tr>
<tr>
<td>&lt;hypervisor&gt;</td>
<td></td>
</tr>
</tbody>
</table>

**Table 18.10. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 18.11. CSV formatter options**
Table 18.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 18.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

18.4. COMPUTE AGENT SET

Set compute agent properties. The compute agent functionality is hypervisor specific and is only supported by the XenAPI hypervisor driver. It was removed from nova in the 23.0.0 (Wallaby) release.

Usage:

```
```

Table 18.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;id&gt;</td>
<td>Id of the agent</td>
</tr>
</tbody>
</table>

Table 18.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
18.5. COMPUTE SERVICE DELETE

Delete compute service(s)

Usage:

```
openstack compute service delete [-h] <service> [<service> ...]
```

Table 18.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Compute service(s) to delete (id only). if using <code>--os-compute-api-version</code> 2.53 or greater, the ID is a UUID which can be retrieved by listing compute services using the same 2.53+ microversion.</td>
</tr>
</tbody>
</table>

Table 18.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

18.6. COMPUTE SERVICE LIST

List compute services. Using `--os-compute-api-version` 2.53 or greater will return the ID as a UUID value which can be used to uniquely identify the service in a multi-cell deployment.

Usage:

```
```
### Table 18.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>List services on specified host (name only)</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>List only specified service binaries (name only). for example, <code>nova-compute</code> , <code>nova-conductor</code> , etc.</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

### Table 18.19. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 18.20. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 18.21. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 18.22. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

18.7. COMPUTE SERVICE SET

Set compute service properties

Usage:

```
openstack compute service set [-h] [--enable | --disable] [--disable-reason <reason>] [--up | --down] <host> <service>
```

Table 18.23. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;host&gt;</td>
<td>Name of host</td>
</tr>
<tr>
<td>&lt;service&gt;</td>
<td>Name of service (binary name), for example <code>nova-compute</code></td>
</tr>
</tbody>
</table>

Table 18.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable service</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable service</td>
</tr>
<tr>
<td>--disable-reason &lt;reason&gt;</td>
<td>Reason for disabling the service (in quotes). should be used with --disable option.</td>
</tr>
<tr>
<td>--up</td>
<td>Force up service. requires <code>--os-compute-api-version</code> <code>2.11</code> or greater.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--down</td>
<td>Force down service. requires <code>--os-compute-api-version</code> 2.11 or greater.</td>
</tr>
</tbody>
</table>
CHAPTER 19. CONFIGURATION

This chapter describes the commands under the `configuration` command.

19.1. CONFIGURATION SHOW

Display configuration details

Usage:

```bash
```

Table 19.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--mask</td>
<td>Attempt to mask passwords (default)</td>
</tr>
<tr>
<td>--unmask</td>
<td>Show password in clear text</td>
</tr>
</tbody>
</table>

Table 19.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 19.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 19.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 19.5. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 20. CONSISTENCY

This chapter describes the commands under the consistency command.

20.1. CONSISTENCY GROUP ADD VOLUME

Add volume(s) to consistency group

Usage:

openstack consistency group add volume [-h] <consistency-group> <volume> [<volume> ...]

Table 20.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consistency-group&gt;</td>
<td>Consistency group to contain &lt;volume&gt; (name or id)</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume(s) to add to &lt;consistency-group&gt; (name or id) (repeat option to add multiple volumes)</td>
</tr>
</tbody>
</table>

Table 20.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

20.2. CONSISTENCY GROUP CREATE

Create new consistency group.

Usage:


Table 20.3. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of new consistency group (default to none)</td>
</tr>
</tbody>
</table>

### Table 20.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--volume-type &lt;volume-type&gt;</td>
<td>Volume type of this consistency group (name or id)</td>
</tr>
<tr>
<td>--consistency-group-source &lt;consistency-group&gt;</td>
<td>Existing consistency group (name or id)</td>
</tr>
<tr>
<td>--consistency-group-snapshot &lt;consistency-group-snapshot&gt;</td>
<td>Existing consistency group snapshot (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of this consistency group</td>
</tr>
<tr>
<td>--availability-zone &lt;availability-zone&gt;</td>
<td>Availability zone for this consistency group (not available if creating consistency group from source)</td>
</tr>
</tbody>
</table>

### Table 20.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 20.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 20.7. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 20.8. Table formatter options
### 20.3. CONSISTENCY GROUP DELETE

Delete consistency group(s).

**Usage:**

```
openstack consistency group delete [-h] [--force] <consistency-group> [...]
```

**Table 20.9. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consistency-group&gt;</td>
<td>Consistency group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 20.10. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Allow delete in state other than error or available</td>
</tr>
</tbody>
</table>

### 20.4. CONSISTENCY GROUP LIST

List consistency groups.

**Usage:**

```
openstack consistency group list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [--quote {all, minimal, none, nonnumeric}] [--noindent] [--max-width <integer>] [--fit-width] [--print-empty]
```
Table 20.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--all-projects</code></td>
<td>Show details for all projects. admin only. (defaults to False)</td>
</tr>
<tr>
<td><code>--long</code></td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 20.12. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 20.13. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 20.14. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 20.15. Table formatter options
20.5. CONSISTENCY GROUP REMOVE VOLUME

Remove volume(s) from consistency group

Usage:

```bash
openstack consistency group remove volume [-h] <consistency-group> <volume> [<volume> ...]
```

Table 20.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consistency-group&gt;</td>
<td>Consistency group containing &lt;volume&gt; (name or id)</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume(s) to remove from &lt;consistency-group&gt; (name or ID) (repeat option to remove multiple volumes)</td>
</tr>
</tbody>
</table>

Table 20.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

20.6. CONSISTENCY GROUP SET

Set consistency group properties

Usage:

```bash
openstack consistency group set [-h] [--name <name>] [--description <description>] <consistency-group>
```
Table 20.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;consistency-group&gt;</code></td>
<td>Consistency group to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 20.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>New consistency group name</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>New consistency group description</td>
</tr>
</tbody>
</table>

20.7. CONSISTENCY GROUP SHOW

Display consistency group details.

Usage:

```
openstack consistency group show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    <consistency-group>
```

Table 20.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;consistency-group&gt;</code></td>
<td>Consistency group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 20.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 20.22. Output formatter options
Table 20.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 20.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 20.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

20.8. CONSISTENCY GROUP SNAPSHOT CREATE

Create new consistency group snapshot.

Usage:

```
openstack consistency group snapshot create [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN [--noindent]]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
```

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
Table 20.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot-name&gt;</td>
<td>Name of new consistency group snapshot (default to None)</td>
</tr>
</tbody>
</table>

Table 20.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--consistency-group &lt;consistency-group&gt;</td>
<td>Consistency group to snapshot (name or id) (default to be the same as &lt;snapshot-name&gt;)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of this consistency group snapshot</td>
</tr>
</tbody>
</table>

Table 20.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 20.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 20.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 20.31. Table formatter options
20.9. CONSISTENCY GROUP SNAPSHOT DELETE

Delete consistency group snapshot(s).

Usage:

```
openstack consistency group snapshot delete [-h]
<consistency-group-snapshot>  
[<consistency-group-snapshot> ...]
```

Table 20.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consistency-group-snapshot&gt;</td>
<td>Consistency group snapshot(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 20.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

20.10. CONSISTENCY GROUP SNAPSHOT LIST

List consistency group snapshots.

Usage:

```
openstack consistency group snapshot list [-h]  
[ -f {csv, json, table, value, yaml}]  
[ -c COLUMN]  
[ --quote {all, minimal, none, nonnumeric}]  
[ --noindent]  
[ --max-width <integer>]  
[ --fit-width]  
[ --print-empty]  
[ --sort-column SORT_COLUMN]
```
[--sort-ascending | --sort-descending]
[--all-projects] [--long]
[--status <status>]
[--consistency-group <consistency-group>]

Table 20.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show detail for all projects (admin only) (defaults to False)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Filters results by a status (&quot;available&quot;, &quot;error&quot;, &quot;creating&quot;, &quot;deleting&quot; or &quot;error_deleting&quot;)</td>
</tr>
<tr>
<td>--consistency-group &lt;group&gt;</td>
<td>Filters results by a consistency group (name or id)</td>
</tr>
</tbody>
</table>

Table 20.35. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 20.36. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 20.37. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 20.38. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

20.11. CONSISTENCY GROUP SNAPSHOT SHOW

Display consistency group snapshot details

Usage:

```
openstack consistency group snapshot show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   <consistency-group-snapshot>
```

Table 20.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consistency-group-snapshot&gt;</td>
<td>Consistency group snapshot to display (name or id)</td>
</tr>
</tbody>
</table>

Table 20.40. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 20.41. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>--c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 20.42. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 20.43. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 20.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 21. CONSOLE

This chapter describes the commands under the console command.

21.1. CONSOLE LOG SHOW

Show server’s console output

Usage:

openstack console log show [-h] [--lines <num-lines>] <server>

Table 21.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to show console log (name or id)</td>
</tr>
</tbody>
</table>

Table 21.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--lines &lt;num-lines&gt;</td>
<td>Number of lines to display from the end of the log (default=all)</td>
</tr>
</tbody>
</table>

21.2. CONSOLE URL SHOW

Show server’s remote console URL

Usage:


Table 21.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to show url (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--novnc</code></td>
<td>Show novnc console url (default)</td>
</tr>
<tr>
<td><code>--xvpvnc</code></td>
<td>Show xvpvnc console url</td>
</tr>
<tr>
<td><code>--spice</code></td>
<td>Show spice console url</td>
</tr>
<tr>
<td><code>--rdp</code></td>
<td>Show rdp console url</td>
</tr>
<tr>
<td><code>--serial</code></td>
<td>Show serial console url</td>
</tr>
<tr>
<td><code>--mks</code></td>
<td>Show webmks console url</td>
</tr>
</tbody>
</table>

Table 21.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>--c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 21.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 21.7. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 21.8. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 22. CONSUMER

This chapter describes the commands under the `consumer` command.

22.1. CONSUMER CREATE

Create new consumer

Usage:

```
```

Table 22.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New consumer description</td>
</tr>
</tbody>
</table>

Table 22.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 22.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 22.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 22.5. Table formatter options
22.2. CONSUMER DELETE

Delete consumer(s)

Usage:

```
openstack consumer delete [-h] <consumer> [<consumer> ...]
```

Table 22.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consumer&gt;</td>
<td>Consumer(s) to delete</td>
</tr>
</tbody>
</table>

Table 22.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

22.3. CONSUMER LIST

List consumers

Usage:

```
```

Table 22.8. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 22.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 22.10. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 22.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 22.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 22.4. CONSUMER SET

Set consumer properties

**Usage:**

```bash
openstack consumer set [-h] [--description <description>] <consumer>
```

**Table 22.13. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consumer&gt;</td>
<td>Consumer to modify</td>
</tr>
</tbody>
</table>

**Table 22.14. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New consumer description</td>
</tr>
</tbody>
</table>

### 22.5. CONSUMER SHOW

Display consumer details

**Usage:**

```bash
```

**Table 22.15. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;consumer&gt;</td>
<td>Consumer to display</td>
</tr>
</tbody>
</table>

**Table 22.16. Command arguments**
### Table 22.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 22.18. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 22.19. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 22.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 23. CONTAINER

This chapter describes the commands under the container command.

23.1. CONTAINER CREATE

Create new container

Usage:


Table 23.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container-name&gt;</td>
<td>New container name(s)</td>
</tr>
</tbody>
</table>

Table 23.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public</td>
<td>Make the container publicly accessible</td>
</tr>
<tr>
<td>--storage-policy STORAGE_POLICY</td>
<td>Specify a particular storage policy to use.</td>
</tr>
</tbody>
</table>

Table 23.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
### Table 23.4. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 23.5. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 23.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 23.2. CONTAINER DELETE

Delete container

**Usage:**

```
openstack container delete [-h] [--recursive] <container> [<container> ...]
```

#### Table 23.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Container(s) to delete</td>
</tr>
</tbody>
</table>
23.3. CONTAINER LIST

List containers

Usage:

```
```

Table 23.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--prefix &lt;prefix&gt;</td>
<td>Filter list using &lt;prefix&gt;</td>
</tr>
<tr>
<td>--marker &lt;marker&gt;</td>
<td>Anchor for paging</td>
</tr>
<tr>
<td>--end-marker &lt;end-marker&gt;</td>
<td>End anchor for paging</td>
</tr>
<tr>
<td>--limit &lt;num-containers&gt;</td>
<td>Limit the number of containers returned</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--all</td>
<td>List all containers (default is 10000)</td>
</tr>
</tbody>
</table>

Table 23.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv, json, table, value, yaml], --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
-c COLUMN, --column COLUMN  Specify the column(s) to include, can be repeated to show multiple columns

--sort-column SORT_COLUMN  Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending  Sort the column(s) in ascending order

--sort-descending  Sort the column(s) in descending order

Table 23.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 23.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 23.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

23.4. CONTAINER SAVE

Save container contents locally

Usage:
openstack container save [-h] <container>

Table 23.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Container to save</td>
</tr>
</tbody>
</table>

Table 23.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

23.5. CONTAINER SET

Set container properties

Usage:

openstack container set [-h] --property <key=value> <container>

Table 23.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Container to modify</td>
</tr>
</tbody>
</table>

Table 23.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property</td>
<td>Set a property on this container (repeat option to set multiple properties)</td>
</tr>
</tbody>
</table>

23.6. CONTAINER SHOW

Display container details

Usage:

Table 23.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Container to display</td>
</tr>
</tbody>
</table>

Table 23.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 23.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 23.21. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 23.22. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 23.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
## 23.7. CONTAINER UNSET

Unset container properties

**Usage:**

```
openstack container unset [-h] --property <key> <container>
```

**Table 23.24. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;container&gt;</code></td>
<td>Container to modify</td>
</tr>
</tbody>
</table>

**Table 23.25. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--property &lt;key&gt;</code></td>
<td>Property to remove from container (repeat option to remove multiple properties)</td>
</tr>
</tbody>
</table>
CHAPTER 24. CREDENTIAL

This chapter describes the commands under the **credential** command.

### 24.1. CREDENTIAL CREATE

Create new credential

**Usage:**

```bash
```

#### Table 24.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;user&gt;</td>
<td>User that owns the credential (name or id)</td>
</tr>
<tr>
<td>&lt;data&gt;</td>
<td>New credential data</td>
</tr>
</tbody>
</table>

#### Table 24.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>New credential type: cert, ec2, totp and so on</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project which limits the scope of the credential (name or ID)</td>
</tr>
</tbody>
</table>

#### Table 24.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

#### Table 24.4. JSON formatter options
### Table 24.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 24.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAXTERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 24.2. CREDENTIAL DELETE

Delete credential(s)

**Usage:**

```
openstack credential delete [-h] <credential-id> [<credential-id> ...]
```

### Table 24.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;credential-id&gt;</td>
<td>Id of credential(s) to delete</td>
</tr>
</tbody>
</table>

### Table 24.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### 24.3. CREDENTIAL LIST
List credentials

Usage:

```
openstack credential list [-h] [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent] [-max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORTCOLUMN]
    [--sort-ascending | --sort-descending]
    [--user <user>] [--user-domain <user-domain>]
    [--type <type>]
```

<table>
<thead>
<tr>
<th>Table 24.9. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 24.10. Output formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
</tr>
<tr>
<td>--sort-ascending</td>
</tr>
<tr>
<td>--sort-descending</td>
</tr>
</tbody>
</table>

| Table 24.11. CSV formatter options |
### 24.4. CREDENTIAL SET

Set credential properties

**Usage:**

```
openstack credential set [-h] --user <user> --type <type> --data <data> [--project <project>] <credential-id>
```

**Table 24.14. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;credential-id&gt;</code></td>
<td>Id of credential to change</td>
</tr>
</tbody>
</table>

**Table 24.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--user &lt;user&gt;</code></td>
<td>User that owns the credential (name or id)</td>
</tr>
</tbody>
</table>
24.5. CREDENTIAL SHOW

Display credential details

**Usage:**

```
```

**Table 24.16. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;credential-id&gt;</td>
<td>Id of credential to display</td>
</tr>
</tbody>
</table>

**Table 24.17. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 24.18. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 24.19. JSON formatter options**
### Table 24.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 24.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 25. CRON

This chapter describes the commands under the `cron` command.

25.1. CRON TRIGGER CREATE

Create new trigger.

Usage:

```
openstack cron trigger create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] [--params PARAMS] [--pattern <* * * * *>] [--first-time <YYYY-MM-DD HH:MM>] [--count <integer>] [--utc] name workflow_identifier [workflow_input]
```

Table 25.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Cron trigger name</td>
</tr>
<tr>
<td>workflow_identifier</td>
<td>Workflow name or id</td>
</tr>
<tr>
<td>workflow_input</td>
<td>Workflow input</td>
</tr>
</tbody>
</table>

Table 25.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--params PARAMS</td>
<td>Workflow params</td>
</tr>
<tr>
<td>--pattern &lt;*&gt;</td>
<td>Cron trigger pattern</td>
</tr>
<tr>
<td>--first-time &lt;YYYY-MM-DD HH:MM&gt;</td>
<td>Date and time of the first execution. time is treated as local time unless --utc is also specified</td>
</tr>
<tr>
<td>--count &lt;integer&gt;</td>
<td>Number of wanted executions</td>
</tr>
<tr>
<td>--utc</td>
<td>All times specified should be treated as utc</td>
</tr>
</tbody>
</table>

Table 25.3. Output formatter options
Table 25.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 25.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 25.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

25.2. CRON TRIGGER DELETE

Delete trigger.

Usage:

```
openstack cron trigger delete [-h] cron_trigger [cron_trigger ...]
```

Table 25.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cron_trigger</td>
<td>Name of cron trigger(s).</td>
</tr>
</tbody>
</table>
Table 25.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

25.3. CRON TRIGGER LIST

List all cron triggers.

Usage:

```
```

Table 25.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_dirs=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
</tbody>
</table>
### Table 25.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 25.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 25.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 25.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 25.4. CRON TRIGGER SHOW

Show specific cron trigger.
Usage:


Table 25.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cron_trigger</td>
<td>Cron trigger name</td>
</tr>
</tbody>
</table>

Table 25.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 25.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 25.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 25.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 25.19. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 26. DATASTORE

This chapter describes the commands under the `datastore` command.

26.1. DATASTORE DELETE

Deletes a datastore

Usage:

```
openstack datastore delete [-h] <datastore>
```

Table 26.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore&gt;</td>
<td>Id or name of the datastore</td>
</tr>
</tbody>
</table>

Table 26.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

26.2. DATASTORE LIST

List available datastores

Usage:

```
```

Table 26.3. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 26.4. Output formatter options
Table 26.5. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 26.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 26.7. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

26.3. DATASTORE SHOW

Shows details of a datastore
Usage:

```
<datastore>
```

Table 26.8. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore&gt;</td>
<td>Id of the datastore</td>
</tr>
</tbody>
</table>

Table 26.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 26.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 26.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 26.12. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 26.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
26.4. DATASTORE VERSION CREATE

Creates a datastore version.

Usage:

```
openstack datastore version create [-h] [--active] [-image-tags IMAGE_TAGS] [--default] [-version-number VERSION_NUMBER] version_name datastore_name datastore_manager image_id
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 26.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>version_name</td>
<td>Datastore version name.</td>
</tr>
<tr>
<td>datastore_name</td>
<td>Datastore name. the datastore is created automatically if does not exist.</td>
</tr>
<tr>
<td>datastore_manager</td>
<td>Datastore manager, e.g. mysql</td>
</tr>
<tr>
<td>image_id</td>
<td>Id of the datastore image in glance. this can be empty string if --image-tags is specified.</td>
</tr>
</tbody>
</table>

Table 26.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--active</td>
<td>Enable the datastore version.</td>
</tr>
</tbody>
</table>
26.5. DATASTORE VERSION DELETE

Deletes a datastore version.

Usage:

openstack datastore version delete [-h] <datastore_version>

Table 26.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore_version&gt;</td>
<td>Id of the datastore version.</td>
</tr>
</tbody>
</table>

Table 26.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

26.6. DATASTORE VERSION LIST

Lists available versions for a datastore

Usage:


Table 26.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore&gt;</td>
<td></td>
</tr>
</tbody>
</table>
### Table 26.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore&gt;</td>
<td>Id or name of the datastore</td>
</tr>
</tbody>
</table>

### Table 26.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 26.21. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 26.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 26.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
26.7. DATASTORE VERSION SET

Updates a datastore version.

Usage:

```bash
openstack datastore version set [-h]
   [--datastore-manager DATASTORE_MANAGER]
   [--image IMAGE]
   [--image-tags IMAGE_TAGS]
   [--version-name VERSION_NAME]
   [--enable | --disable]
   [--default | --non-default]
   datastore_version_id
```

Table 26.24. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>datastore_version_id</td>
<td>Datastore version id.</td>
</tr>
</tbody>
</table>

Table 26.25. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--datastore-manager DATASTORE_MANAGER</td>
<td>Datastore manager name.</td>
</tr>
<tr>
<td>--image IMAGE</td>
<td>Id of the datastore image in glance.</td>
</tr>
<tr>
<td>--image-tags IMAGE_TAGS</td>
<td>List of image tags separated by comma, e.g.</td>
</tr>
<tr>
<td></td>
<td>trove,mysql</td>
</tr>
</tbody>
</table>
26.8. DATASTORE VERSION SHOW

Shows details of a datastore version.

**Usage:**

```bash
```

**Table 26.26. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;datastore_version&gt;</td>
<td>Id or name of the datastore version.</td>
</tr>
</tbody>
</table>

**Table 26.27. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--datastore &lt;datastore&gt;</td>
<td>Id or name of the datastore, optional if the id of the datastore_version is provided.</td>
</tr>
</tbody>
</table>

**Table 26.28. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 26.29. JSON formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 26.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 26.31. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the <code>CLIFF_MAX_TERM_WIDTH</code> environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 27. DNS

This chapter describes the commands under the `dns` command.

27.1. DNS QUOTA LIST

List quotas

Usage:

```bash
```

Table 27.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
<tr>
<td>--project-id PROJECT_ID</td>
<td>Project id default: current project</td>
</tr>
</tbody>
</table>

Table 27.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 27.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 27.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

27.2. DNS QUOTA RESET

Reset quotas

Usage:

```
openstack dns quota reset [-h] [--all-projects]
                          [--sudo-project-id SUDO_PROJECT_ID]
                          [--project-id PROJECT_ID]
```

Table 27.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
<tr>
<td>--project-id PROJECT_ID</td>
<td>Project id</td>
</tr>
</tbody>
</table>

27.3. DNS QUOTA SET

Set quotas

Usage:

```
openstack dns quota set [-h] [-f {json,shell,table,value,yaml}]```


Table 27.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
<tr>
<td>--project-id PROJECT_ID</td>
<td>Project id</td>
</tr>
<tr>
<td>--api-export-size &lt;api-export-size&gt;</td>
<td>New value for the api-export-size quota</td>
</tr>
<tr>
<td>--recordset-records &lt;recordset-records&gt;</td>
<td>New value for the recordset-records quota</td>
</tr>
<tr>
<td>--zone-records &lt;zone-records&gt;</td>
<td>New value for the zone-records quota</td>
</tr>
<tr>
<td>--zone-recordsets &lt;zone-recordsets&gt;</td>
<td>New value for the zone-recordsets quota</td>
</tr>
<tr>
<td>--zones &lt;zones&gt;</td>
<td>New value for the zones quota</td>
</tr>
</tbody>
</table>

Table 27.8. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### Table 27.10. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 27.11. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 27.4. DNS SERVICE LIST

List service statuses

**Usage:**

```
```

### Table 27.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### Table 27.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--hostname HOSTNAME</td>
<td>Hostname</td>
</tr>
<tr>
<td>--service_name SERVICE_NAME</td>
<td>Service name</td>
</tr>
<tr>
<td>--status STATUS</td>
<td>Status</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

### Table 27.14. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 27.15. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 27.16. Table formatter options
27.5. DNS SERVICE SHOW

Show service status details

Usage:

```
```

Table 27.17. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Service status id</td>
</tr>
</tbody>
</table>

Table 27.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 27.19. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 27.20. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 27.21. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 27.22. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 28. DOMAIN

This chapter describes the commands under the domain command.

28.1. DOMAIN CREATE

Create new domain

Usage:


Table 28.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;domain-name&gt;</td>
<td>New domain name</td>
</tr>
</tbody>
</table>

Table 28.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New domain description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable domain (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable domain</td>
</tr>
<tr>
<td>--or-show</td>
<td>Return existing domain</td>
</tr>
<tr>
<td>--immutable</td>
<td>Make resource immutable. an immutable project may not be deleted or modified except to remove the immutable flag</td>
</tr>
<tr>
<td>--no-immutable</td>
<td>Make resource mutable (default)</td>
</tr>
</tbody>
</table>

Table 28.3. Output formatter options
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

#### Table 28.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table 28.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

#### Table 28.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 28.2. DOMAIN DELETE

Delete domain(s)

**Usage:**

```
openstack domain delete [-h] <domain> [<domain> ...]
```

#### Table 28.7. Positional arguments
### 28.3. DOMAIN LIST

List domains

**Usage:**

```bash
```

**Table 28.9. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>The domain name</td>
</tr>
<tr>
<td>--enabled</td>
<td>The domains that are enabled will be returned</td>
</tr>
</tbody>
</table>

**Table 28.10. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
### Table 28.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 28.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 28.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 28.4. DOMAIN SET

Set domain properties

**Usage:**

```
```

**Table 28.14. Positional arguments**
Table 28.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;domain&gt;</td>
<td>Domain to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 28.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New domain name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New domain description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable domain</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable domain</td>
</tr>
<tr>
<td>--immutable</td>
<td>Make resource immutable. An immutable project may not be deleted or modified except to remove the immutable flag</td>
</tr>
<tr>
<td>--no-immutable</td>
<td>Make resource mutable (default)</td>
</tr>
</tbody>
</table>

28.5. DOMAIN SHOW

Display domain details

Usage:


Table 28.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;domain&gt;</td>
<td>Domain to display (name or id)</td>
</tr>
</tbody>
</table>

Table 28.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### Table 28.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 28.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 28.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 28.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 29. DYNAMIC

This chapter describes the commands under the `dynamic` command.

29.1. DYNAMIC ACTION CREATE

Create new action.

Usage:

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Dynamic action name</td>
</tr>
<tr>
<td>class_name</td>
<td>Dynamic action class name</td>
</tr>
<tr>
<td>code_source</td>
<td>Code source id or name</td>
</tr>
</tbody>
</table>

Table 29.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag an action will be marked as &quot;public&quot;.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the action within.</td>
</tr>
</tbody>
</table>

Table 29.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 29.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 29.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 29.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

29.2. DYNAMIC ACTION DELETE

Delete action.

Usage:

```
openstack dynamic action delete [-h] [--namespace [NAMESPACE]] identifier [identifier ...]
```

Table 29.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Dynamic action name or id (can be repeated multiple times).</td>
</tr>
</tbody>
</table>

Table 29.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the dynamic action(s).</td>
</tr>
</tbody>
</table>

### 29.3. DYNAMIC ACTION LIST

List all dynamic actions.

**Usage:**

```
openstack dynamic action list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
[--quote {all,minimal,none,nonnumeric}] [-noindent] [-max-width <integer>]
 [--fit-width] [-print-empty]
 [--sort-column SORT_COLUMN]
 [--sort-ascending | --sort-descending]
 [--marker [MARKER]] [--limit [LIMIT]]
 [--sort_keys [SORT_KEYS]]
 [--sort_dirs [SORT_DIRS]]
 [--filter FILTERS]
 [--namespace [NAMESPACE]]
```

**Table 29.9. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of dynamic actions.</td>
</tr>
</tbody>
</table>

**Table 29.10. Output formatter options**
Table 29.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 29.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 29.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

29.4. DYNAMIC ACTION SHOW

Show specific dynamic action.

Usage:

**Table 29.14. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Dynamic action identifier (name or id)</td>
</tr>
</tbody>
</table>

**Table 29.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the dynamic action within.</td>
</tr>
</tbody>
</table>

**Table 29.16. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 29.17. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 29.18. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 29.19. Table formatter options**
29.5. DYNAMIC ACTION UPDATE

Update dynamic action.

Usage:

```
```

Table 29.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Dynamic action identifier (id or name)</td>
</tr>
</tbody>
</table>

Table 29.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--class-name [CLASS_NAME]</td>
<td>Dynamic action class name.</td>
</tr>
<tr>
<td>--code-source [CODE_SOURCE]</td>
<td>Code source identifier (id or name).</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag action will be marked as &quot;public&quot;.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the action.</td>
</tr>
</tbody>
</table>
### Table 29.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 29.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 29.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 29.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 30. EC2

This chapter describes the commands under the `ec2` command.

30.1. EC2 CREDENTIALS CREATE

Create EC2 credentials

Usage:

```
openstack ec2 credentials create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty] [--project <project>]
    [--user <user>]
    [--user-domain <user-domain>]
    [--project-domain <project-domain>]
```

Table 30.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--project &lt;project&gt;</code></td>
<td>Create credentials in project (name or id; default: current authenticated project)</td>
</tr>
<tr>
<td><code>--user &lt;user&gt;</code></td>
<td>Create credentials for user (name or id; default: current authenticated user)</td>
</tr>
<tr>
<td><code>--user-domain &lt;user-domain&gt;</code></td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td><code>--project-domain &lt;project-domain&gt;</code></td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 30.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 30.3. JSON formatter options
Value | Summary
---|---
--noindent | Whether to disable indenting the json

Table 30.4. Shell formatter options

Value | Summary
---|---
--prefix PREFIX | Add a prefix to all variable names

Table 30.5. Table formatter options

Value | Summary
---|---
--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty | Print empty table if there is no data to show.

30.2. EC2 CREDENTIALS DELETE

Delete EC2 credentials

Usage:

```bash
openstack ec2 credentials delete [-h] [--user <user>] [--user-domain <user-domain>] <access-key> [<access-key> ...]
```

Table 30.6. Positional arguments

Value | Summary
---|---
<access-key> | Credentials access key(s)

Table 30.7. Command arguments

Value | Summary
---|---
-h, --help | Show this help message and exit
--user <user> | Delete credentials for user (name or id)
### 30.3. EC2 CREDENTIALS LIST

List EC2 credentials

**Usage:**

```
```

#### Table 30.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--user &lt;user&gt;</td>
<td>Filter list by user (name or id)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

#### Table 30.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
### Table 30.10. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 30.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 30.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 30.4. EC2 CREDENTIALS SHOW

Display EC2 credentials details

**Usage:**

```
```

### Table 30.13. Positional arguments
### Table 30.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access-key&gt;</td>
<td>Credentials access key</td>
</tr>
</tbody>
</table>

### Table 30.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Show credentials for user (name or id)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

### Table 30.16. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 30.17. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 30.18. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 31. ENDPOINT

This chapter describes the commands under the endpoint command.

31.1. ENDPOINT ADD PROJECT

Associate a project to an endpoint

Usage:

openstack endpoint add project [-h] [-project-domain <project-domain>]
  <endpoint> <project>

Table 31.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint&gt;</td>
<td>Endpoint to associate with specified project (name or ID)</td>
</tr>
<tr>
<td>&lt;project&gt;</td>
<td>Project to associate with specified endpoint name or ID</td>
</tr>
</tbody>
</table>

Table 31.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

31.2. ENDPOINT CREATE

Create new endpoint

Usage:

openstack endpoint create [-h] [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent] [--prefix PREFIX]
  [-max-width <integer>] [--fit-width]
  [--print-empty] [--region <region-id>]
  [--enable | --disable]
  <service> <interface> <url>

Table 31.3. Positional arguments
**Table 31.4. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;service&gt;</code></td>
<td>Service to be associated with new endpoint (name or ID)</td>
</tr>
<tr>
<td><code>&lt;interface&gt;</code></td>
<td>New endpoint interface type (admin, public or internal)</td>
</tr>
<tr>
<td><code>&lt;url&gt;</code></td>
<td>New endpoint url</td>
</tr>
</tbody>
</table>

**Table 31.5. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--region &lt;region-id&gt;</code></td>
<td>New endpoint region id</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable endpoint (default)</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable endpoint</td>
</tr>
</tbody>
</table>

**Table 31.6. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 31.7. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 31.8. Table formatter options**
31.3. ENDPOINT DELETE

Delete endpoint(s)

Usage:

openstack endpoint delete [-h] <endpoint-id> [<endpoint-id> ...]

Table 31.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint-id&gt;</td>
<td>Endpoint(s) to delete (id only)</td>
</tr>
</tbody>
</table>

Table 31.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

31.4. ENDPOINT GROUP ADD PROJECT

Add a project to an endpoint group

Usage:

openstack endpoint group add project [-h] [\|--project-domain <project-domain>\] <endpoint-group> <project>

Table 31.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint-group&gt;</td>
<td>Endpoint group (name or id)</td>
</tr>
</tbody>
</table>
Table 31.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Project to associate (name or id)</td>
</tr>
</tbody>
</table>

Table 31.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the endpoint group</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
<td>Filename that contains a new set of filters</td>
</tr>
</tbody>
</table>

Table 31.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description of the endpoint group</td>
</tr>
</tbody>
</table>

Table 31.15. Output formatter options

31.5. ENDPOINT GROUP CREATE

Create new endpoint group

Usage:

### 31.6. ENDPOINT GROUP DELETE

Delete endpoint group(s)

**Usage:**

```bash
openstack endpoint group delete [-h] <endpoint-group> [<endpoint-group> ...]
```

**Table 31.19. Positional arguments**
31.7. ENDPOINT GROUP LIST

List endpoint groups

Usage:


Table 31.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint-group&gt;</td>
<td>Endpoint group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 31.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--endpointgroup &lt;endpoint-group&gt;</td>
<td>Endpoint group (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project (name or id)</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;project&gt; (name or id)</td>
</tr>
</tbody>
</table>

Table 31.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
**Table 31.23. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 31.24. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 31.25. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 31.8. ENDPOINT GROUP REMOVE PROJECT

Remove project from endpoint group

**Usage:**

```
openstack endpoint group remove project [-h]
    [--project-domain <project-domain>]
    <endpoint-group> <project>
```

**Table 31.26. Positional arguments**
### 31.9. ENDPOINT GROUP SET

Set endpoint group properties

**Usage:**

```
openstack endpoint group set [-h] [--name <name>] [--filters <filename>] [--description <description>] <endpoint-group>
```

**Table 31.28. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint-group&gt;</td>
<td>Endpoint group to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 31.29. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New endpoint group name</td>
</tr>
<tr>
<td>--filters &lt;filename&gt;</td>
<td>Filename that contains a new set of filters</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New endpoint group description</td>
</tr>
</tbody>
</table>

### 31.10. ENDPOINT GROUP SHOW
Display endpoint group details

Usage:

```bash
```

Table 31.30. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpointgroup&gt;</td>
<td>Endpoint group (name or id)</td>
</tr>
</tbody>
</table>

Table 31.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 31.32. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 31.33. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 31.34. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 31.35. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 31.11. ENDPOINT LIST

List endpoints

**Usage:**

```
```

**Table 31.36. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Filter by service (type, name or id)</td>
</tr>
<tr>
<td>--interface &lt;interface&gt;</td>
<td>Filter by interface type (admin, public or internal)</td>
</tr>
<tr>
<td>--region &lt;region-id&gt;</td>
<td>Filter by region id</td>
</tr>
<tr>
<td>--endpoint &lt;endpoint-group&gt;</td>
<td>Endpoint to list filters</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project to list filters (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>
### Table 31.37. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv,json,table,value,yaml], --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 31.38. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 31.39. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 31.40. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 31.12. ENDPOINT REMOVE PROJECT

Dissociate a project from an endpoint
Usage:

openstack endpoint remove project [-h]
    [--project-domain <project-domain>]
    <endpoint> <project>

Table 31.41. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint&gt;</td>
<td>Endpoint to dissociate from specified project (name or ID)</td>
</tr>
<tr>
<td>&lt;project&gt;</td>
<td>Project to dissociate from specified endpoint name or ID</td>
</tr>
</tbody>
</table>

Table 31.42. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

31.13. ENDPOINT SET

Set endpoint properties

Usage:

openstack endpoint set [-h] [--region <region-id>]
    [--interface <interface>] [--url <url>]
    [--service <service>] [--enable | --disable]
    <endpoint-id>

Table 31.43. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint-id&gt;</td>
<td>Endpoint to modify (id only)</td>
</tr>
</tbody>
</table>

Table 31.44. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### 31.14. ENDPOINT SHOW

Display endpoint details

**Usage:**

```
```

**Table 31.45. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;endpoint&gt;</td>
<td>Endpoint to display (endpoint id, service id, service name, service type)</td>
</tr>
</tbody>
</table>

**Table 31.46. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 31.47. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 31.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 31.49. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 31.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 32. EVENT

This chapter describes the commands under the event command.

32.1. EVENT TRIGGER CREATE

Create new trigger.

Usage:

```
```

Table 32.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Event trigger name</td>
</tr>
<tr>
<td>workflow_id</td>
<td>Workflow id</td>
</tr>
<tr>
<td>exchange</td>
<td>Event trigger exchange</td>
</tr>
<tr>
<td>topic</td>
<td>Event trigger topic</td>
</tr>
<tr>
<td>event</td>
<td>Event trigger event name</td>
</tr>
<tr>
<td>workflow_input</td>
<td>Workflow input</td>
</tr>
</tbody>
</table>

Table 32.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--params PARAMS</td>
<td>Workflow params</td>
</tr>
</tbody>
</table>

Table 32.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 32.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 32.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 32.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

32.2. EVENT TRIGGER DELETE

Delete trigger.

Usage:

```
openstack event trigger delete [-h]
    event_trigger_id [event_trigger_id ...]
```

Table 32.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_trigger_id</td>
<td>Id of event trigger(s).</td>
</tr>
</tbody>
</table>
32.3. EVENT TRIGGER LIST

List all event triggers.

Usage:

```
```

Table 32.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list -sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
</tbody>
</table>

Table 32.10. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 32.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 32.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 32.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

32.4. EVENT TRIGGER SHOW

Show specific event trigger.

Usage:

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>event_trigger</td>
<td>Event trigger id</td>
</tr>
</tbody>
</table>

**Table 32.14. Positional arguments**

**Table 32.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 32.16. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 32.17. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 32.18. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 32.19. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 33. EXTENSION

This chapter describes the commands under the `extension` command.

33.1. EXTENSION LIST

List API extensions

**Usage:**

```
```

**Table 33.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--compute</td>
<td>List extensions for the compute api</td>
</tr>
<tr>
<td>--identity</td>
<td>List extensions for the identity api</td>
</tr>
<tr>
<td>--network</td>
<td>List extensions for the network api</td>
</tr>
<tr>
<td>--volume</td>
<td>List extensions for the block storage api</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

**Table 33.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv, json, table, value, yaml], --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 33.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 33.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 33.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

33.2. EXTENSION SHOW

Show API extension

Usage:

```shell
```

Table 33.6. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;extension&gt;</code></td>
<td>Extension to display. Currently, only network extensions are supported. (Name or Alias)</td>
</tr>
</tbody>
</table>

**Table 33.7. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 33.8. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 33.9. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 33.10. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 33.11. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width. <code>&lt;1</code> to disable. You can also use the <code>CLIFF_MAX_TERM_WIDTH</code> environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if <code>--max-width</code> greater than <code>0</code>. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 34. FEDERATION

This chapter describes the commands under the `federation` command.

### 34.1. FEDERATION DOMAIN LIST

List accessible domains

**Usage:**

```
openstack federation domain list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
[--quote {all, minimal, none, nonnumeric}] [--noindent] [--max-width <integer>]
[--fit-width] [--print-empty] [--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
```

**Table 34.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 34.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 34.3. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 34.4. JSON formatter options**
### 34.2. FEDERATION PROJECT LIST

List accessible projects

#### Usage:

```
```

#### Table 34.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### Table 34.7. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
---sort-column \( \text{SORT\_COLUMN} \)

Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

---sort-ascending

Sort the column(s) in ascending order

---sort-descending

Sort the column(s) in descending order

**Table 34.8. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 34.9. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 34.10. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width (&lt;\text{integer})</td>
<td>Maximum display width, (&lt;1) to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 34.3. FEDERATION PROTOCOL CREATE

Create new federation protocol

**Usage:**

```
openstack federation protocol create [-h]
   \[-f \{json,shell,table,value,yaml\}\]
   \[-c COLUMN\] \[-noindent\]
   \[-prefix PREFIX\]
   \[-max-width \(<\text{integer}\>\]
   \[-fit-width\] \[-print-empty\]
```
Table 34.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New federation protocol name (must be unique per identity provider)</td>
</tr>
</tbody>
</table>

Table 34.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--identity-provider &lt;identity-provider&gt;</td>
<td>Identity provider that will support the new federation protocol (name or ID) (required)</td>
</tr>
<tr>
<td>--mapping &lt;mapping&gt;</td>
<td>Mapping that is to be used (name or id) (required)</td>
</tr>
</tbody>
</table>

Table 34.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 34.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 34.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 34.6. Table formatter options
34.4. FEDERATION PROTOCOL DELETE

Delete federation protocol(s)

Usage:

openstack federation protocol delete [-h] --identity-provider
  <identity-provider>
  <federation-protocol>
  [<federation-protocol> ...]

Table 34.17. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;federation-protocol&gt;</td>
<td>Federation protocol(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 34.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
| --identity-provider
  <identity-provider>         | Identity provider that supports <federation-protocol>    |
|                              | (name or ID) (required)                                   |

34.5. FEDERATION PROTOCOL LIST

List federation protocols

Usage:

openstack federation protocol list [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent] [--max-width <integer>]
Table 34.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--identity-provider &lt;identity-provider&gt;</td>
<td>Identity provider to list (name or id) (required)</td>
</tr>
</tbody>
</table>

Table 34.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 34.21. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 34.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 34.23. Table formatter options
### 34.6. FEDERATION PROTOCOL SET

Set federation protocol properties

**Usage:**

```
openstack federation protocol set [-h] --identity-provider <identity-provider> [-f {json,shell,table,value,yaml}]
```

**Table 34.24. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Federation protocol to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 34.25. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--identity-provider &lt;provider&gt;</td>
<td>Identity provider that supports &lt;federation-protocol&gt; (name or ID) (required)</td>
</tr>
<tr>
<td>--mapping &lt;mapping&gt;</td>
<td>Mapping that is to be used (name or id)</td>
</tr>
</tbody>
</table>

### 34.7. FEDERATION PROTOCOL SHOW

Display federation protocol details

**Usage:**

```
openstack federation protocol show [-h] [-f {json,shell,table,value,yaml}]
```
Table 34.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;federation-protocol&gt;</td>
<td>Federation protocol to display (name or id)</td>
</tr>
</tbody>
</table>

Table 34.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--identity-provider &lt;identity-provider&gt;</td>
<td>Identity provider that supports &lt;federation-protocol&gt; (name or ID) (required)</td>
</tr>
</tbody>
</table>

Table 34.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 34.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 34.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 34.31. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 35. FLAVOR

This chapter describes the commands under the `flavor` command.

### 35.1. FLAVOR CREATE

Create new flavor

Usage:

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flavor-name&gt;</code></td>
<td>New flavor name</td>
</tr>
</tbody>
</table>

Table 35.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--id <code>&lt;id&gt;</code></td>
<td>Unique flavor id</td>
</tr>
<tr>
<td>--ram <code>&lt;size-mb&gt;</code></td>
<td>Memory size in mb (default 256m)</td>
</tr>
<tr>
<td>--disk <code>&lt;size-gb&gt;</code></td>
<td>Disk size in gb (default 0g)</td>
</tr>
<tr>
<td>--ephemeral <code>&lt;size-gb&gt;</code></td>
<td>Ephemeral disk size in gb (default 0g)</td>
</tr>
<tr>
<td>--swap <code>&lt;size-mb&gt;</code></td>
<td>Additional swap space size in mb (default 0m)</td>
</tr>
<tr>
<td>--vcpus <code>&lt;vcpus&gt;</code></td>
<td>Number of vcpus (default 1)</td>
</tr>
<tr>
<td>--rxtx-factor <code>&lt;factor&gt;</code></td>
<td>Rx/tx factor (default 1.0)</td>
</tr>
<tr>
<td>--public</td>
<td>Flavor is available to other projects (default)</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--private</td>
<td>Flavor is not available to other projects</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to add for this flavor (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Allow &lt;project&gt; to access private flavor (name or id) (Must be used with --private option)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the flavor.(supported by api versions 2.55 - 2.latest)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

**Table 35.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 35.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 35.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 35.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
35.2. FLAVOR DELETE

Delete flavor(s)

Usage:

openstack flavor delete [-h] <flavor> [<flavor> ...]

Table 35.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 35.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

35.3. FLAVOR LIST

List flavors

Usage:


Table 35.9. Command arguments
### Table 35.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public</td>
<td>List only public flavors (default)</td>
</tr>
<tr>
<td>--private</td>
<td>List only private flavors</td>
</tr>
<tr>
<td>--all</td>
<td>List all flavors, whether public or private</td>
</tr>
<tr>
<td>--min-disk &lt;min-disk&gt;</td>
<td>Filters the flavors by a minimum disk space, in gib.</td>
</tr>
<tr>
<td>--min-ram &lt;min-ram&gt;</td>
<td>Filters the flavors by a minimum ram, in mib.</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--marker &lt;flavor-id&gt;</td>
<td>The last flavor id of the previous page</td>
</tr>
<tr>
<td>--limit &lt;num-flavors&gt;</td>
<td>Maximum number of flavors to display. this is also configurable on the server. The actual limit used will be the lower of the user-supplied value and the server configuration-derived value</td>
</tr>
</tbody>
</table>

### Table 35.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv,json,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

--quote {all,minimal,none,nonnumeric}      | When to include quotes, defaults to nonnumeric                                                     |
### Table 35.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 35.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 35.4. FLAVOR SET

Set flavor properties

**Usage:**

```
```

### Table 35.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to modify (name or id)</td>
</tr>
</tbody>
</table>

### Table 35.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--no-property</td>
<td>Remove all properties from this flavor (specify both --no-property and --property to remove the current properties before setting new properties.)</td>
</tr>
</tbody>
</table>
### 35.5. FLAVOR SHOW

Display flavor details

**Usage:**

```
```

**Table 35.16. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 35.17. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 35.18. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 35.19. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 35.20. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 35.21. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 35.6. FLAVOR UNSET

Unset flavor properties

**Usage:**

```bash
openstack flavor unset [-h] [--property <key>] [--project <project>]
[--project-domain <project-domain>]
<flavor>
```

**Table 35.22. Positional arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 35.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to remove from flavor (repeat option to unset multiple properties)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Remove flavor access from project (name or id) (admin only)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>
CHAPTER 36. FLOATING

This chapter describes the commands under the `floating` command.

36.1. FLOATING IP CREATE

Create floating IP

Usage:

```
openstack floating ip create [-h] [-f {json,shell,table,value,yaml}] 
[-c COLUMN] [-noindent] [-prefix PREFIX] 
[--max-width <integer>] [--fit-width] 
[--print-empty] [-subnet <subnet>] 
[-port <port>] 
[--floating-ip-address <ip-address>] 
[--fixed-ip-address <ip-address>] 
[--qos-policy <qos-policy>] 
[--description <description>] 
[--project <project>] 
[--dns-domain <dns-domain>] 
[--dns-name <dns-name>] 
[--project-domain <project-domain>] 
[--tag <tag> | --no-tag] 
<network>
```

Table 36.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Network to allocate floating ip from (name or id)</td>
</tr>
</tbody>
</table>

Table 36.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--subnet &lt;subnet&gt;</td>
<td>Subnet on which you want to create the floating ip (name or ID)</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>Port to be associated with the floating ip (name or ID)</td>
</tr>
<tr>
<td>--floating-ip-address &lt;ip-address&gt;</td>
<td>Floating ip address</td>
</tr>
<tr>
<td>--fixed-ip-address &lt;ip-address&gt;</td>
<td>Fixed ip address mapped to the floating ip</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
<td>Attach qos policy to the floating ip (name or id)</td>
</tr>
</tbody>
</table>
### Table 36.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set floating ip description</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--dns-domain &lt;dns-domain&gt;</td>
<td>Set dns domain for this floating ip</td>
</tr>
<tr>
<td>--dns-name &lt;dns-name&gt;</td>
<td>Set dns name for this floating ip</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the floating ip (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the floating ip</td>
</tr>
</tbody>
</table>

### Table 36.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 36.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 36.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
36.2. FLOATING IP DELETE

Delete floating IP(s)

Usage:

```bash
openstack floating ip delete [-h] <floating-ip> [<floating-ip> ...]
```

Table 36.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip(s) to delete (ip address or id)</td>
</tr>
</tbody>
</table>

Table 36.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

36.3. FLOATING IP LIST

List floating IP(s)

Usage:

```bash
```
Table 36.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>List floating ip(s) according to given network (name or ID)</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>List floating ip(s) according to given port (name or ID)</td>
</tr>
<tr>
<td>--fixed-ip-address &lt;ip-address&gt;</td>
<td>List floating ip(s) according to given fixed ip address</td>
</tr>
<tr>
<td>--floating-ip-address &lt;ip-address&gt;</td>
<td>List floating ip(s) according to given floating ip address</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>List floating ip(s) according to given status (ACTIVE, DOWN)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List floating ip(s) according to given project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--router &lt;router&gt;</td>
<td>List floating ip(s) according to given router (name or ID)</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List floating ip which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List floating ip which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude floating ip which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>
Table 36.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude floating ip which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 36.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 36.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 36.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
36.4. FLOATING IP POOL LIST

List pools of floating IP addresses

Usage:

```bash
```

Table 36.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 36.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 36.16. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
### 36.5. FLOATING IP PORT FORWARDING CREATE

Create floating IP port forwarding

#### Usage:

```
openstack floating ip port forwarding create [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width]
   [--print-empty]
   --internal-ip-address
   <internal-ip-address>
   --port <port>
   --internal-protocol-port
   <port-number>
   --external-protocol-port
   <port-number> --protocol
   <protocol>
   [--description <description>]
   <floating-ip>
```

#### Table 36.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip that the port forwarding belongs to (ip address or ID)</td>
</tr>
</tbody>
</table>
### Table 36.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--internal-ip-address &lt;internal-ip-address&gt;</td>
<td>The fixed ipv4 address of the network port associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>The name or id of the network port associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--internal-protocol-port &lt;port-number&gt;</td>
<td>The protocol port number of the network port fixed IPv4 address associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--external-protocol-port &lt;port-number&gt;</td>
<td>The protocol port number of the port forwarding’s floating IP address</td>
</tr>
<tr>
<td>--protocol &lt;protocol&gt;</td>
<td>The protocol used in the floating ip port forwarding, for instance: TCP, UDP</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>A text to describe/contextualize the use of the port forwarding configuration</td>
</tr>
</tbody>
</table>

### Table 36.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 36.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 36.23. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 36.24. Table formatter options
### 36.6. FLOATING IP PORT FORWARDING DELETE

Delete floating IP port forwarding

**Usage:**

```
openstack floating ip port forwarding delete [-h]
<floating-ip>
<port-forwarding-id>
[<port-forwarding-id> ...]
```

**Table 36.25. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;floating-ip&gt;</code></td>
<td>Floating ip that the port forwarding belongs to (ip address or ID)</td>
</tr>
<tr>
<td><code>&lt;port-forwarding-id&gt;</code></td>
<td>The id of the floating ip port forwarding(s) to delete</td>
</tr>
</tbody>
</table>

**Table 36.26. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 36.7. FLOATING IP PORT FORWARDING LIST

List floating IP port forwarding

**Usage:**

```
openstack floating ip port forwarding list [-h]
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
```
Table 36.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip that the port forwarding belongs to (ip address or ID)</td>
</tr>
</tbody>
</table>

Table 36.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--port &lt;port&gt;</code></td>
<td>Filter the list result by the id or name of the internal network port</td>
</tr>
<tr>
<td><code>--external-protocol-port &lt;port-number&gt;</code></td>
<td>Filter the list result by the protocol port number of the floating IP</td>
</tr>
<tr>
<td><code>--protocol protocol</code></td>
<td>Filter the list result by the port protocol</td>
</tr>
</tbody>
</table>

Table 36.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>--c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 36.30. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 36.31. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 36.32. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

36.8. FLOATING IP PORT FORWARDING SET

Set floating IP Port Forwarding Properties

Usage:

```bash
openstack floating ip port forwarding set [-h] [-port <port>]
    [-internal-ip-address <internal-ip-address>]
    [-internal-protocol-port <port-number>]
    [-external-protocol-port <port-number>]
    [-protocol <protocol>]
    [-description <description>]
    <floating-ip>
    <port-forwarding-id>
```

Table 36.33. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip that the port forwarding belongs to (ip address or ID)</td>
</tr>
</tbody>
</table>
Table 36.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-forwarding-id&gt;</td>
<td>The id of the floating ip port forwarding</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>The id of the network port associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--internal-ip-address &lt;internal-ip-address&gt;</td>
<td>The fixed ipv4 address of the network port associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--internal-protocol-port &lt;port-number&gt;</td>
<td>The tcp/udp/other protocol port number of the network port fixed IPv4 address associated to the floating IP port forwarding</td>
</tr>
<tr>
<td>--external-protocol-port &lt;port-number&gt;</td>
<td>The tcp/udp/other protocol port number of the port forwarding’s floating IP address</td>
</tr>
<tr>
<td>--protocol &lt;protocol&gt;</td>
<td>The ip protocol used in the floating ip port forwarding</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>A text to describe/contextualize the use of the port forwarding configuration</td>
</tr>
</tbody>
</table>

### 36.9. FLOATING IP PORT FORWARDING SHOW

Display floating IP Port Forwarding details

**Usage:**

```
openstack floating ip port forwarding show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [-prefix PREFIX]
    [-max-width <integer>]
    [-fit-width]
    [-print-empty]
    <floating-ip>
    <port-forwarding-id>
```

Table 36.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td></td>
</tr>
<tr>
<td>&lt;port-forwarding-id&gt;</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>&lt;floating-ip&gt;</code></td>
<td>Floating ip that the port forwarding belongs to (ip address or ID)</td>
</tr>
<tr>
<td><code>&lt;port-forwarding-id&gt;</code></td>
<td>The id of the floating ip port forwarding</td>
</tr>
</tbody>
</table>

**Table 36.36. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 36.37. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 36.38. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 36.39. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 36.40. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
---fit-width
Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty
Print empty table if there is no data to show.

### 36.10. FLOATING IP SET

Set floating IP Properties

**Usage:**

```bash
```

**Table 36.41. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip to modify (ip address or id)</td>
</tr>
</tbody>
</table>

**Table 36.42. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>Associate the floating ip with port (name or id)</td>
</tr>
<tr>
<td>--fixed-ip-address &lt;ip-address&gt;</td>
<td>Fixed ip of the port (required only if port has multiple IPs)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set floating ip description</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
<td>Attach qos policy to the floating ip (name or id)</td>
</tr>
<tr>
<td>--no-qos-policy</td>
<td>Remove the qos policy attached to the floating ip</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the floating ip (repeat option to set multiple tags)</td>
</tr>
</tbody>
</table>
36.11. FLOATING IP SHOW

Display floating IP details

Usage:

```bash
```

Table 36.43. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip to display (ip address or id)</td>
</tr>
</tbody>
</table>

Table 36.44. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 36.45. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 36.46. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 36.47. Shell formatter options
Table 36.48. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

36.12. FLOATING IP UNSET

Unset floating IP Properties

Usage:

```
```

Table 36.49. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;floating-ip&gt;</td>
<td>Floating ip to disassociate (ip address or id)</td>
</tr>
</tbody>
</table>

Table 36.50. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--port</td>
<td>Disassociate any port associated with the floating ip</td>
</tr>
<tr>
<td>--qos-policy</td>
<td>Remove the qos policy attached to the floating ip</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the floating ip (repeat option to remove multiple tags)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the floating ip</td>
</tr>
</tbody>
</table>
CHAPTER 37. GROUP

This chapter describes the commands under the `group` command.

### 37.1. GROUP ADD USER

Add user to group

**Usage:**

```
openstack group add user [-h] [--group-domain <group-domain>]
                   [--user-domain <user-domain>]
                   <group> <user> [...]
```

**Table 37.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Group to contain &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>&lt;user&gt;</td>
<td>User(s) to add to &lt;group&gt; (name or id) (repeat option to add multiple users)</td>
</tr>
</tbody>
</table>

**Table 37.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--group-domain &lt;group-domain&gt;</td>
<td>Domain the group belongs to (name or id). this can be used in case collisions between group names exist.</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

### 37.2. GROUP CONTAINS USER

Check user membership in group

**Usage:**

```
openstack group contains user [-h] [--group-domain <group-domain>]
                            [--user-domain <user-domain>]
                            <group> <user>
```

**Table 37.3. Positional arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;group&gt;</code></td>
<td>Group to check (name or id)</td>
</tr>
<tr>
<td><code>&lt;user&gt;</code></td>
<td>User to check (name or id)</td>
</tr>
</tbody>
</table>

**Table 37.4. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--group-domain &lt;group-domain&gt;</code></td>
<td>Domain the group belongs to (name or id). this can be used in case collisions between group names exist.</td>
</tr>
<tr>
<td><code>--user-domain &lt;user-domain&gt;</code></td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

### 37.3. GROUP CREATE

Create new group

**Usage:**

```
```

**Table 37.5. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;group-name&gt;</code></td>
<td>New group name</td>
</tr>
</tbody>
</table>

**Table 37.6. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--domain &lt;domain&gt;</code></td>
<td>Domain to contain new group (name or id)</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>New group description</td>
</tr>
</tbody>
</table>
### 37.4. GROUP DELETE

Delete group(s)

**Usage:**

```bash
--or-show
```

**Summary:**

Return existing group

### Table 37.7. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 37.8. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 37.9. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 37.10. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>


openstack group delete [-h] [--domain <domain>] <group> [<group> ...]

Table 37.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 37.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain containing group(s) (name or id)</td>
</tr>
</tbody>
</table>

37.5. GROUP LIST

List groups

Usage:

openstack group list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] 
    [-o {all, minimal, none, nonnumeric}] 
    [--noindent] [-m -w <integer>] [-w] 
    [-p empty] [-s COLUMN] 
    [--ascending | --descending] 
    [--domain <domain>] [--user <user>] 
    [--user-domain <user-domain>] [--long]

Table 37.13. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Filter group list by &lt;domain&gt; (name or id)</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Filter group list by &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 37.14. Output formatter options
### 37.15. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### 37.16. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 37.17. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 37.6. GROUP REMOVE USER

Remove user from group

Usage:
openstack group remove user [-h] [--group-domain <group-domain>]
    [--user-domain <user-domain>]
    <group> <user> [<user> ...]

Table 37.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Group containing &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>&lt;user&gt;</td>
<td>User(s) to remove from &lt;group&gt; (name or id) (repeat option to remove multiple users)</td>
</tr>
</tbody>
</table>

Table 37.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--group-domain &lt;group-domain&gt;</td>
<td>Domain the group belongs to (name or id). this can be used in case collisions between group names exist.</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

37.7. GROUP SET

Set group properties

Usage:

openstack group set [-h] [--domain <domain>] [--name <name>]
    [--description <description>]
    <group>

Table 37.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Group to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 37.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain containing &lt;group&gt; (name or id)</td>
</tr>
</tbody>
</table>
37.8. GROUP SHOW

Display group details

Usage:

```
```

Table 37.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 37.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain containing &lt;group&gt; (name or id)</td>
</tr>
</tbody>
</table>

Table 37.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 37.25. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 37.26. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 37.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 38. HELP

This chapter describes the commands under the `help` command.

38.1. HELP

print detailed help for another command

Usage:

```
openstack help [-h] [cmd ...]
```

Table 38.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd</td>
<td>Name of the command</td>
</tr>
</tbody>
</table>

Table 38.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
CHAPTER 39. HOST

This chapter describes the commands under the `host` command.

39.1. HOST LIST

List hosts

Usage:

```
openstack host list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                  [--quote {all,minimal,none,nonnumeric}]
                  [--noindent] [-m <integer>] [--fit-width]
                  [--print-empty] [--sort-column SORT_COLUMN]
                  [--sort-ascending | --sort-descending]
                  [--zone <zone>]
```

Table 39.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--zone &lt;zone&gt;</td>
<td>Only return hosts in the availability zone</td>
</tr>
</tbody>
</table>

Table 39.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 39.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 39.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 39.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

39.2. HOST SET

Set host properties

Usage:

```
openstack host set [-h] [--enable | --disable]
   [--enable-maintenance | --disable-maintenance]
   <host>
```

Table 39.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;host&gt;</td>
<td>Host to modify (name only)</td>
</tr>
</tbody>
</table>

Table 39.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the host</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the host</td>
</tr>
<tr>
<td>--enable-maintenance</td>
<td>Enable maintenance mode for the host</td>
</tr>
</tbody>
</table>
### 39.3. HOST SHOW

Display host details

**Usage:**

```
openstack host show [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] 
[--quote {all, minimal, none, nonnumeric}] 
[--noindent] [--max-width <integer>] [--fit-width] 
[--print-empty] [--sort-column SORT_COLUMN] 
[--sort-ascending | --sort-descending] 
<host>
```

**Table 39.8. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;host&gt;</td>
<td>Name of host</td>
</tr>
</tbody>
</table>

**Table 39.9. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 39.10. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 39.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 39.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 39.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 40. HYPERVISOR

This chapter describes the commands under the hypervisor command.

40.1. HYPERVISOR LIST

List hypervisors

Usage:


Table 40.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--matching &lt;hostname&gt;</td>
<td>Filter hypervisors using &lt;hostname&gt; substring</td>
</tr>
<tr>
<td>--marker &lt;marker&gt;</td>
<td>The uuid of the last hypervisor of the previous page; displays list of hypervisors after marker. (supported with --os-compute-api-version 2.33 or above)</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of hypervisors to display. note that there is a configurable max limit on the server, and the limit that is used will be the minimum of what is requested here and what is configured in the server. (supported with --os-compute-api-version 2.33 or above)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 40.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 40.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 40.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 40.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

40.2. HYPERVISOR SHOW

Display hypervisor details

Usage:

```
```
### Table 40.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;hypervisor&gt;</td>
<td>Hypervisor to display (name or id)</td>
</tr>
</tbody>
</table>

### Table 40.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 40.8. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 40.9. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 40.10. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 40.11. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
40.3. HYPERVISOR STATS SHOW

Display hypervisor stats details

Usage:


<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 40.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 40.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 40.14. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 40.15. Shell formatter options

Table 40.16. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 41. IDENTITY

This chapter describes the commands under the `identity` command.

41.1. IDENTITY PROVIDER CREATE

Create new identity provider

Usage:

```
openstack identity provider create [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   [--remote-id <remote-id> | --remote-id-file <file-name>]
   [--description <description>]
   [--domain <domain>]
   [--enable | --disable]
   <name>
```

Table 41.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New identity provider name (must be unique)</td>
</tr>
</tbody>
</table>

Table 41.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--remote-id &lt;remote-id&gt;</td>
<td>Remote ids to associate with the identity provider (repeat option to provide multiple values)</td>
</tr>
<tr>
<td>--remote-id-file &lt;file-name&gt;</td>
<td>Name of a file that contains many remote ids to associate with the identity provider, one per line</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New identity provider description</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain to associate with the identity provider. If not specified, a domain will be created automatically. (Name or ID)</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable identity provider (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the identity provider</td>
</tr>
</tbody>
</table>
Table 41.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 41.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 41.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 41.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

41.2. IDENTITY PROVIDER DELETE

Delete identity provider(s)

Usage:

    openstack identity provider delete [-h] [identity-provider] ...

Table 41.7. Positional arguments
41.3. IDENTITY PROVIDER LIST

List identity providers

Usage:

```
```

Table 41.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--id &lt;id&gt;</td>
<td>The identity providers' id attribute</td>
</tr>
<tr>
<td>--enabled</td>
<td>The identity providers that are enabled will be returned</td>
</tr>
</tbody>
</table>

Table 41.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Sort the column(s) in ascending order

Table 41.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 41.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 41.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

41.4. IDENTITY PROVIDER SET

Set identity provider properties

Usage:

```
```
### Table 41.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;identity-provider&gt;</code></td>
<td>Identity provider to modify</td>
</tr>
</tbody>
</table>

### Table 41.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>Set identity provider description</td>
</tr>
<tr>
<td><code>--remote-id &lt;remote-id&gt;</code></td>
<td>Remote ids to associate with the identity provider (repeat option to provide multiple values)</td>
</tr>
<tr>
<td><code>--remote-id-file &lt;file-name&gt;</code></td>
<td>Name of a file that contains many remote ids to associate with the identity provider, one per line</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable the identity provider</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable the identity provider</td>
</tr>
</tbody>
</table>

### 41.5. IDENTITY PROVIDER SHOW

Display identity provider details

**Usage:**

```bash
```

### Table 41.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;identity-provider&gt;</code></td>
<td>Identity provider to display</td>
</tr>
</tbody>
</table>

### Table 41.17. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 41.18. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 41.19. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 41.20. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 41.21. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 42. IMAGE

This chapter describes the commands under the `image` command.

### 42.1. IMAGE ADD PROJECT

Associate project with image

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;image&gt;</code></td>
<td>Image to share (name or id)</td>
</tr>
<tr>
<td><code>&lt;project&gt;</code></td>
<td>Project to associate with image (id)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--project-domain &lt;project-domain&gt;</code></td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 42.4. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 42.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 42.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, (&lt;1) to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**42.2. IMAGE CREATE**

Create/upload an image

**Usage:**

```bash
```

**Table 42.7. Positional arguments**
Table 42.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;image-name&gt;</code></td>
<td>New image name</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--id <code>&lt;id&gt;</code></td>
<td>Image id to reserve</td>
</tr>
<tr>
<td>--container-format <code>&lt;container-format&gt;</code></td>
<td>Image container format. The supported options are: ami, ari, aki, bare, docker, ova, ovf. The default format is: bare</td>
</tr>
<tr>
<td>--disk-format <code>&lt;disk-format&gt;</code></td>
<td>Image disk format. The supported options are: ami, ari, aki, vhd, vmdk, raw, qcow2, vhdx, vdi, iso, ploop. The default format is: raw</td>
</tr>
<tr>
<td>--min-disk <code>&lt;disk-gb&gt;</code></td>
<td>Minimum disk size needed to boot image, in gigabytes</td>
</tr>
<tr>
<td>--min-ram <code>&lt;ram-mb&gt;</code></td>
<td>Minimum ram size needed to boot image, in megabytes</td>
</tr>
<tr>
<td>--file <code>&lt;file&gt;</code></td>
<td>Upload image from local file</td>
</tr>
<tr>
<td>--volume <code>&lt;volume&gt;</code></td>
<td>Create image from a volume</td>
</tr>
<tr>
<td>--force</td>
<td>Force image creation if volume is in use (only meaningful with --volume)</td>
</tr>
<tr>
<td>--progress</td>
<td>Show upload progress bar</td>
</tr>
<tr>
<td>--sign-key-path <code>&lt;sign-key-path&gt;</code></td>
<td>Sign the image using the specified private key. Only use in combination with --sign-cert-id</td>
</tr>
<tr>
<td>--sign-cert-id <code>&lt;sign-cert-id&gt;</code></td>
<td>The specified certificate uuid is a reference to the certificate in the key manager that corresponds to the public key and is used for signature validation. Only use in combination with --sign-key-path</td>
</tr>
<tr>
<td>--protected</td>
<td>Prevent image from being deleted</td>
</tr>
<tr>
<td>--unprotected</td>
<td>Allow image to be deleted (default)</td>
</tr>
<tr>
<td>--public</td>
<td>Image is accessible to the public</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--private</td>
<td>Image is inaccessible to the public (default)</td>
</tr>
<tr>
<td>--community</td>
<td>Image is accessible to the community</td>
</tr>
<tr>
<td>--shared</td>
<td>Image can be shared</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on this image (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Set a tag on this image (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Set an alternate project on this image (name or id)</td>
</tr>
<tr>
<td>--import</td>
<td>Force the use of glance image import instead of direct upload</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id), this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

**Table 42.9. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 42.10. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 42.11. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 42.12. Table formatter options**

Red Hat OpenStack Platform 17.0 Command Line Interface Reference

452
42.3. IMAGE DELETE

Delete image(s)

Usage:

```
openstack image delete [-h] <image> [<image> ...]
```

Table 42.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 42.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

42.4. IMAGE LIST

List available images

Usage:

```
openstack image list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
[-quote {all, minimal, none, nonnumeric}]
[--noindent] [--max-width <integer>] [--fit-width]
[-print-empty] [-sort-column SORT_COLUMN]
[-sort-ascending | --sort-descending]
[-public | --private | --community | --shared]
[-property <key=value>] [-name <name>]
[-status <status>]
[--member-status <member-status>]
[-project <project>]
```
Table 42.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public</td>
<td>List only public images</td>
</tr>
<tr>
<td>--private</td>
<td>List only private images</td>
</tr>
<tr>
<td>--community</td>
<td>List only community images</td>
</tr>
<tr>
<td>--shared</td>
<td>List only shared images</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Filter output based on property (repeat option to filter on multiple properties)</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Filter images based on name.</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Filter images based on status.</td>
</tr>
<tr>
<td>--member-status &lt;member-status&gt;</td>
<td>Filter images based on member status. the supported options are: accepted, pending, rejected, all.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Search by project (admin only) (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Filter images based on tag.</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[::&lt;direction&gt;]</td>
<td>Sort output by selected keys and directions(asc or desc) (default: name:asc), multiple keys and directions can be specified separated by comma</td>
</tr>
<tr>
<td>--limit &lt;num-images&gt;</td>
<td>Maximum number of images to display.</td>
</tr>
<tr>
<td>--marker &lt;image&gt;</td>
<td>The last image of the previous page. display list of images after marker. Display all images if not specified. (name or ID)</td>
</tr>
</tbody>
</table>

Table 42.16. Output formatter options
### Value Summary

- **-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}**
  - The output format, defaults to table

- **-c COLUMN, --column COLUMN**
  - Specify the column(s) to include, can be repeated to show multiple columns

- **--sort-column SORT_COLUMN**
  - Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

- **--sort-ascending**
  - Sort the column(s) in ascending order

- **--sort-descending**
  - Sort the column(s) in descending order

<table>
<thead>
<tr>
<th>Table 42.17. CSV formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 42.18. JSON formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>--noindent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 42.19. Table formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
</tr>
<tr>
<td>--fit-width</td>
</tr>
<tr>
<td>--print-empty</td>
</tr>
</tbody>
</table>

### 42.5. IMAGE MEMBER LIST

List projects associated with image

**Usage:**
openstack image member list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [--quote {all,minimal,none,nonnumeric}] [--noindent] [-m <image>] [-w <integer>] [-f {csv,json,table,value,yaml}] [--sort-column SORT_COLUMN] [--sort-ascending | --sort-descending] [--project-domain <project-domain>]

Table 42.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image (name or id)</td>
</tr>
</tbody>
</table>

Table 42.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 42.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 42.23. CSV formatter options
### Table 42.24. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 42.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 42.6. IMAGE REMOVE PROJECT

Disassociate project with image

**Usage:**

```
openstack image remove project [-h] [--project-domain <project-domain>] <image> <project>
```

**Table 42.26. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image to unshare (name or id)</td>
</tr>
<tr>
<td>&lt;project&gt;</td>
<td>Project to disassociate with image (name or id)</td>
</tr>
</tbody>
</table>

**Table 42.27. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
42.7. IMAGE SAVE

Save an image locally

Usage:

openstack image save [-h] [--file <filename>] <image>

Table 42.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image to save (name or id)</td>
</tr>
</tbody>
</table>

Table 42.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--file &lt;filename&gt;</td>
<td>Downloaded image save filename (default: stdout)</td>
</tr>
</tbody>
</table>

42.8. IMAGE SET

Set image properties

Usage:

openstack image set [-h] [--name <name>] [--min-disk <disk-gb>]      
  [--min-ram <ram-mb>]     
  [--container-format <container-format>]  
  [--disk-format <disk-format>]  
  [--protected | --unprotected]  
  [--public | --private | --community | --shared]  
  [--property <key=value>] [--tag <tag>]  
  [--architecture <architecture>]  
  [--instance-id <instance-id>]  
  [--kernel-id <kernel-id>] [--os-distro <os-distro>]  
  [--os-version <os-version>]  
  [--ramdisk-id <ramdisk-id>]  
  [--deactivate | --activate] [--project <project>]

Red Hat OpenStack Platform 17.0 Command Line Interface Reference

458
Table 42.30. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 42.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New image name</td>
</tr>
<tr>
<td>--min-disk &lt;disk-gb&gt;</td>
<td>Minimum disk size needed to boot image, in gigabytes</td>
</tr>
<tr>
<td>--min-ram &lt;ram-mb&gt;</td>
<td>Minimum ram size needed to boot image, in megabytes</td>
</tr>
<tr>
<td>--container-format &lt;container-format&gt;</td>
<td>Image container format. The supported options are: ami, ari, aki, bare, docker, ova, ovf</td>
</tr>
<tr>
<td>--disk-format &lt;disk-format&gt;</td>
<td>Image disk format. The supported options are: ami, ari, aki, vhd, vmdk, raw, qcow2, vhdx, vdi, iso, ploop</td>
</tr>
<tr>
<td>--protected</td>
<td>Prevent image from being deleted</td>
</tr>
<tr>
<td>--unprotected</td>
<td>Allow image to be deleted (default)</td>
</tr>
<tr>
<td>--public</td>
<td>Image is accessible to the public</td>
</tr>
<tr>
<td>--private</td>
<td>Image is inaccessible to the public (default)</td>
</tr>
<tr>
<td>--community</td>
<td>Image is accessible to the community</td>
</tr>
<tr>
<td>--shared</td>
<td>Image can be shared</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on this image (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Set a tag on this image (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>--architecture &lt;architecture&gt;</td>
<td>Operating system architecture</td>
</tr>
<tr>
<td>--instance-id &lt;instance-id&gt;</td>
<td>Id of server instance used to create this image</td>
</tr>
<tr>
<td>--kernel-id &lt;kernel-id&gt;</td>
<td>Id of kernel image used to boot this disk image</td>
</tr>
<tr>
<td>--os-distro &lt;os-distro&gt;</td>
<td>Operating system distribution name</td>
</tr>
<tr>
<td>--os-version &lt;os-version&gt;</td>
<td>Operating system distribution version</td>
</tr>
<tr>
<td>--ramdisk-id &lt;ramdisk-id&gt;</td>
<td>Id of ramdisk image used to boot this disk image</td>
</tr>
<tr>
<td>--deactivate</td>
<td>Deactivate the image</td>
</tr>
<tr>
<td>--activate</td>
<td>Activate the image</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Set an alternate project on this image (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--accept</td>
<td>Accept the image membership</td>
</tr>
<tr>
<td>--reject</td>
<td>Reject the image membership</td>
</tr>
<tr>
<td>--pending</td>
<td>Reset the image membership to pending</td>
</tr>
</tbody>
</table>

42.9. IMAGE SHOW

Display image details

Usage:

```
```

Table 42.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image to display (name or id)</td>
</tr>
</tbody>
</table>
Table 42.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--human-readable</td>
<td>Print image size in a human-friendly format.</td>
</tr>
</tbody>
</table>

Table 42.34. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 42.35. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 42.36. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 42.37. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFFFITWIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

42.10. IMAGE UNSET

Unset image tags and properties
Usage:

openstack image unset [-h] [--tag <tag>] [--property <property-key>] <image>

Table 42.38. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image&gt;</td>
<td>Image to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 42.39. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Unset a tag on this image (repeat option to unset multiple tags)</td>
</tr>
<tr>
<td>--property &lt;property-key&gt;</td>
<td>Unset a property on this image (repeat option to unset multiple properties)</td>
</tr>
</tbody>
</table>
CHAPTER 43. IMPLIED

This chapter describes the commands under the implied command.

43.1. IMPLIED ROLE CREATE

Creates an association between prior and implied roles

Usage:

```
```

Table 43.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role (name or id) that implies another role</td>
</tr>
</tbody>
</table>

Table 43.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--implied-role &lt;role&gt;</td>
<td>&lt;role&gt; (name or id) implied by another role</td>
</tr>
</tbody>
</table>

Table 43.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 43.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 43.5. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 43.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**43.2. IMPLIED ROLE DELETE**

Deletes an association between prior and implied roles

Usage:

```
openstack implied role delete [-h] --implied-role <role> <role>
```

**Table 43.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role (name or id) that implies another role</td>
</tr>
</tbody>
</table>

**Table 43.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--implied-role &lt;role&gt;</td>
<td>&lt;role&gt; (name or id) implied by another role</td>
</tr>
</tbody>
</table>

**43.3. IMPLIED ROLE LIST**

List implied roles

Usage:

```
openstack implied role list [-h] [-f {csv, json, table, value, yaml}]
```
Table 43.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 43.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 43.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 43.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 43.13. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width <code>&lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 44. IP

This chapter describes the commands under the **ip** command.

### 44.1. IP AVAILABILITY LIST

List IP availability for network

**Usage:**

```bash
```

**Table 44.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--ip-version &lt;ip-version&gt;</td>
<td>List ip availability of given ip version networks (default is 4)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List ip availability of given project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

**Table 44.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
44.2. IP AVAILABILITY SHOW

Show network IP availability details

**Usage:**

```
```

**Table 44.6. Positional arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Show IP availability for a specific network (name or ID)</td>
</tr>
</tbody>
</table>

**Table 44.7. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 44.8. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 44.9. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 44.10. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 44.11. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 45. KEYPAIR

This chapter describes the commands under the **keypair** command.

### 45.1. KEYPAIR CREATE

Create new public or private key for server ssh access

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Table 45.1. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>&lt;name&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 45.2. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--public-key &lt;file&gt;</td>
</tr>
<tr>
<td>--private-key &lt;file&gt;</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
</tr>
</tbody>
</table>

| Table 45.3. Output formatter options |
45.2. KEYPAIR DELETE

Delete public or private key(s)

Usage:

    openstack keypair delete [-h] [--user <user>] [-user-domain <user-domain>] <key> [<key> ...]

Table 45.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 45.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 45.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 45.7. Positional arguments
### Table 45.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;key&gt;</code></td>
<td>Name of key(s) to delete (name only)</td>
</tr>
</tbody>
</table>

### 45.3. KEYPAIR LIST

List key fingerprints

**Usage:**

```
```

**Table 45.9. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--user &lt;user&gt;</code></td>
<td>Show keypairs for another user (admin only) (name or ID). Requires <code>--os-compute-api-version</code> 2.10 or greater.</td>
</tr>
<tr>
<td><code>--user-domain &lt;user-domain&gt;</code></td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>
### Table 45.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--project &lt;project&gt;</td>
<td>Show keypairs for all users associated with project (admin only) (name or ID). Requires <code>--os-compute-api-version</code> 2.10 or greater.</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--marker MARKER</td>
<td>The last keypair id of the previous page</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>Maximum number of keypairs to display</td>
</tr>
</tbody>
</table>

### Table 45.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 45.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 45.13. Table formatter options
### 45.4. KEYPAIR SHOW

Display key details

**Usage:**

```bash
```

**Table 45.14. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;key&gt;</td>
<td>Public or private key to display (name only)</td>
</tr>
</tbody>
</table>

**Table 45.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public-key</td>
<td>Show only bare public key paired with the generated key</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>The owner of the keypair. (admin only) (name or id). Requires <code>--os-compute-api-version</code> 2.10 or greater.</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). this can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

**Table 45.16. Output formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 45.17. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 45.18. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 45.19. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 46. L2GW

This chapter describes the commands under the `l2gw` command.

46.1. L2GW CONNECTION CREATE

Create l2gateway-connection

Usage:

```bash
openstack l2gw connection create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty]
[--default-segmentation-id SEG_ID]
<GATEWAY-NAME/UUID>
<NETWORK-NAME/UUID>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;GATEWAY-NAME/UUID&gt;</td>
<td>Descriptive name for logical gateway.</td>
</tr>
<tr>
<td>&lt;NETWORK-NAME/UUID&gt;</td>
<td>Network name or uuid.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--default-segmentation-id SEG_ID</td>
<td>Default segmentation-id that will be applied to the interfaces for which segmentation id was not specified in l2-gateway-create command.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 46.1. Positional arguments

Table 46.2. Command arguments

Table 46.3. Output formatter options

Table 46.4. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 46.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 46.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 46.2. L2GW CONNECTION DELETE

Delete a given l2gateway-connection

**Usage:**

```plaintext
openstack l2gw connection delete [-h]
<L2_GATEWAY_CONNECTIONS>
[<L2_GATEWAY_CONNECTIONS> ...]
```

**Table 46.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;L2_GATEWAYgetConnections&gt;</td>
<td>Id(s) of l2_gateway_connections(s) to delete.</td>
</tr>
</tbody>
</table>

**Table 46.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
46.3. L2GW CONNECTION LIST

List l2gateway-connections

Usage:


Table 46.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 46.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 46.11. CSV formatter options
### 46.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 46.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 46.4. L2GW CONNECTION SHOW

Show information of a given l2gateway-connection

**Usage:**

```bash
code
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;L2_GATEWAY_CONNECTION&gt;</td>
<td>Id of l2_gateway_connection to look up.</td>
</tr>
</tbody>
</table>
### Table 46.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

The output format, defaults to table

Specify the column(s) to include, can be repeated to show multiple columns

### Table 46.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 46.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 46.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 46.5. L2GW CREATE

Create l2gateway resource

**Usage:**

Table 46.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;GATEWAY-NAME&gt;</td>
<td>Descriptive name for logical gateway.</td>
</tr>
</tbody>
</table>

Table 46.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--device name=name,interface_names=INTERFACE-DETAILS</td>
<td>Device name and interface-names of l2gateway. INTERFACE-DETAILS is of form &quot;&lt;interface_name1&gt;; [&lt;interface_name2&gt;][&lt;seg_id1&gt;[#&lt;seg_id2&gt;]]&quot; (--device option can be repeated)</td>
</tr>
</tbody>
</table>

Table 46.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 46.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 46.24. Shell formatter options
46.6. L2GW DELETE

Delete a given l2gateway

Usage:

openstack l2gw delete [-h] <L2_GATEWAY> [<L2_GATEWAY> ...]

Table 46.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;L2_GATEWAY&gt;</td>
<td>Id(s) or name(s) of l2_gateway to delete.</td>
</tr>
</tbody>
</table>

Table 46.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

46.7. L2GW LIST

List l2gateway that belongs to a given tenant

Usage:

openstack l2gw list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]  
  [--quote {all,minimal,none,nonnumeric}]  
  [--noindent] [--max-width <integer>] [--fit-width]
Table 46.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 46.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 46.30. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 46.31. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 46.32. Table formatter options
### 46.8. L2GW SHOW

Show information of a given l2gateway

**Usage:**

```
openstack l2gw show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] 
[--noindent] [--prefix PREFIX] 
[--max-width <integer>] [--fit-width] 
[--print-empty] 
<L2_GATEWAY>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 46.33. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;L2_GATEWAY&gt;</td>
<td>Id or name of l2_gateway to look up.</td>
</tr>
</tbody>
</table>

Table 46.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 46.35. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
| -f {json,shell,table,value,yaml}, --format 
{json,shell,table,value,yaml} | The output format, defaults to table                      |
| -c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns |

Table 46.36. JSON formatter options
Table 46.37. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 46.38. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

46.9. L2GW UPDATE

Update a given l2gateway

Usage:

```
```

Table 46.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;L2_GATEWAY&gt;</td>
<td>Id or name of l2_gateway to update.</td>
</tr>
</tbody>
</table>

Table 46.40. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name name</td>
<td>Descriptive name for logical gateway.</td>
</tr>
<tr>
<td>--device name=name,interface_names=INTERFACE-DETAILS</td>
<td>Device name and interface-names of l2gateway. INTERFACE-DETAILS is of form &quot;&lt;interface_name1&gt;; [&lt;interface_name2&gt;][&lt;seg_id1&gt;[#&lt;seg_id2&gt;]]&quot; (--device option can be repeated)</td>
</tr>
</tbody>
</table>

Table 46.41. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 46.42. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 46.43. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 46.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 47. LIMIT

This chapter describes the commands under the limit command.

47.1. LIMIT CREATE

Create a limit

Usage:


Table 47.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;resource-name&gt;</td>
<td>The name of the resource to limit</td>
</tr>
</tbody>
</table>

Table 47.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the limit</td>
</tr>
<tr>
<td>--region &lt;region&gt;</td>
<td>Region for the limit to affect.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project to associate the resource limit to</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Service responsible for the resource to limit</td>
</tr>
<tr>
<td>--resource-limit &lt;resource-limit&gt;</td>
<td>The resource limit for the project to assume</td>
</tr>
</tbody>
</table>

Table 47.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**Table 47.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 47.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 47.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 47.2. LIMIT DELETE

Delete a limit

**Usage:**

```bash
openstack limit delete [-h] <limit-id> [<limit-id> ...]
```

**Table 47.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;limit-id&gt;</td>
<td>Limit to delete (id)</td>
</tr>
</tbody>
</table>
Table 47.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

47.3. LIMIT LIST

List limits

Usage:

```
openstack limit list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
[--quote {all, minimal, none, nonnumeric}]
[--noindent] [--max-width <integer>] [--fit-width]
[--print-empty] [--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--service <service>]
[--resource-name <resource-name>]
[--region <region>] [--project <project>]
```

Table 47.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Service responsible for the resource to limit</td>
</tr>
<tr>
<td>--resource-name &lt;resource-name&gt;</td>
<td>The name of the resource to limit</td>
</tr>
<tr>
<td>--region &lt;region&gt;</td>
<td>Region for the registered limit to affect.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List resource limits associated with project</td>
</tr>
</tbody>
</table>

Table 47.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
Table 47.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 47.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 47.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

47.4. LIMIT SET

Update information about a limit

Usage:

```
```

Table 47.14. Positional arguments
### Table 47.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;limit-id&gt;</code></td>
<td>Limit to update (id)</td>
</tr>
</tbody>
</table>

### Table 47.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>Description of the limit</td>
</tr>
<tr>
<td><code>--resource-limit &lt;resource-limit&gt;</code></td>
<td>The resource limit for the project to assume</td>
</tr>
</tbody>
</table>

### Table 47.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 47.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 47.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
47.5. LIMIT SHOW

Display limit details

Usage:

```Shell
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 47.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;limit-id&gt;</td>
<td>Limit to display (id)</td>
</tr>
</tbody>
</table>

Table 47.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 47.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 47.23. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 47.24. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 47.25. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 48. LIMITS

This chapter describes the commands under the `limits` command.

48.1. LIMITS SHOW

Show compute and block storage limits

Usage:

```
openstack limits show [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
   [--quote {all,minimal,none,nonnumeric}] [-noindent] [-max-width <integer>]
   [-fit-width] [-print-empty] [--sort-column SORT_COLUMN]
   [--sort-ascending | --sort-descending]
   (--absolute | --rate) [--reserved] [--project <project>] [--domain <domain>]
```

Table 48.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--absolute</td>
<td>Show absolute limits</td>
</tr>
<tr>
<td>--rate</td>
<td>Show rate limits</td>
</tr>
<tr>
<td>--reserved</td>
<td>Include reservations count [only valid with --absolute]</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Show limits for a specific project (name or id) [only valid with --absolute]</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the project belongs to (name or id) [only valid with --absolute]</td>
</tr>
</tbody>
</table>

Table 48.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv,json,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
**Table 48.3. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 48.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 48.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 49. LOADBALANCER

This chapter describes the commands under the loadbalancer command.

49.1. LOADBALANCER AMPHORA CONFIGURE

Update the amphora agent configuration

Usage:

openstack loadbalancer amphora configure [-h] [--wait] <amphora-id>

Table 49.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;amphora-id&gt;</td>
<td>Uuid of the amphora to configure.</td>
</tr>
</tbody>
</table>

Table 49.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

49.2. LOADBALANCER AMPHORA DELETE

Delete an amphora

Usage:

openstack loadbalancer amphora delete [-h] [--wait] <amphora-id>

Table 49.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;amphora-id&gt;</td>
<td>Uuid of the amphora to delete.</td>
</tr>
</tbody>
</table>

Table 49.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>
49.3. LOADBALANCER AMPHORA FAILOVER

Force failover an amphora

Usage:

openstack loadbalancer amphora failover [-h] [--wait] <amphora-id>

Table 49.5. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;amphora-id&gt;</td>
<td>Uuid of the amphora.</td>
</tr>
</tbody>
</table>

Table 49.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

49.4. LOADBALANCER AMPHORA LIST

List amphorae

Usage:

openstack loadbalancer amphora list [-h]

[-f {csv, json, table, value, yaml}]
[-c COLUMN]
[--quote {all, minimal, none, nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--loadbalancer <loadbalancer>]
[--compute-id <compute-id>]
[--role {BACKUP, MASTER, STANDALONE}]
[--status

{ALLOCATED, BOOTING, DELETED, ERROR, PENDING_CREATE, PENDING_DELETE, READY}]
[--long]

Table 49.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--loadbalancer &lt;loadbalancer&gt;</td>
<td>Filter by load balancer (name or id).</td>
</tr>
<tr>
<td>--compute-id &lt;compute-id&gt;</td>
<td>Filter by compute id.</td>
</tr>
<tr>
<td>--role {BACKUP, MASTER, STANDALONE}</td>
<td>Filter by role.</td>
</tr>
<tr>
<td>--status {ALLOCATED, BOOTING, DELETED, ERROR, PENDING_CREATE, PENDING_DELETE, READY}, --provisioning-status {ALLOCATED, BOOTING, DELETED, ERROR, PENDING_CREATE, PENDING_DELETE, READY}</td>
<td>Filter by amphora provisioning status.</td>
</tr>
<tr>
<td>--long</td>
<td>Show additional fields.</td>
</tr>
</tbody>
</table>

**Table 49.8. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 49.9. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 49.10. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### Table 49.11. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 49.5. LOADBALANCER AMPHORA SHOW

Show the details of a single amphora

**Usage:**

```
openstack loadbalancer amphora show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
<amphora-id>
```

### Table 49.12. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;amphora-id&gt;</td>
<td>Uuid of the amphora.</td>
</tr>
</tbody>
</table>

### Table 49.13. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 49.14. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
49.6. LOADBALANCER AMPHORA STATS SHOW

Shows the current statistics for an amphora.

Usage:

```
openstack loadbalancer amphora stats show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [---noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--listener <listener>]
    <amphora-id>
```

Table 49.18. Positional arguments
Table 49.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;amphora-id&gt;</td>
<td>Uuid of the amphora</td>
</tr>
</tbody>
</table>

Table 49.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--listener &lt;listener&gt;</td>
<td>Filter by listener (name or id)</td>
</tr>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.21. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.22. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, 0 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
49.7. LOADBALANCER AVAILABILITYZONE CREATE

Create an octavia availability zone

Usage:

```
openstack loadbalancer availabilityzone create [-h]
[--f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty] --name
<name>
--availabilityzoneprofile
<availabilityzone_profile>
[--description <description>]
[--enable | --disable]
```

Table 49.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 49.25. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 49.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.27. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.28. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.8. LOADBALANCER AVAILABILITYZONE DELETE

Delete an availability zone

Usage:

```
openstack loadbalancer availabilityzone delete [-h] <availabilityzone>
```

Table 49.29. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone&gt;</td>
<td>Name of the availability zone to delete.</td>
</tr>
</tbody>
</table>

Table 49.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
49.9. LOADBALANCER AVAILABILITYZONE LIST

List availability zones

Usage:

openstack loadbalancer availabilityzone list [-h]
  [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--name <name>]
  [--availabilityzoneprofile <availabilityzone_profile>]
  [--enable | --disable]

Table 49.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List availability zones according to their name.</td>
</tr>
<tr>
<td>--availabilityzoneprofile &lt;availabilityzone_profile&gt;</td>
<td>List availability zones according to their az profile.</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled availability zones.</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled availability zones.</td>
</tr>
</tbody>
</table>

Table 49.32. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
Table 49.33. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.34. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.35. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.10. LOADBALANCER AVAILABILITYZONE SET

Update an availability zone

Usage:

```
openstack loadbalancer availabilityzone set [-h]  
  [--description <description>]  
  [--enable | --disable]  
  <availabilityzone>
```

Table 49.36. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone&gt;</td>
<td>Name of the availability zone to update.</td>
</tr>
</tbody>
</table>
Table 49.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set the description of the availability zone.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the availability zone.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the availability zone.</td>
</tr>
</tbody>
</table>

49.11. LOADBALANCER AVAILABILITYZONE SHOW

Show the details for a single availability zone

Usage:

    openstack loadbalancer availabilityzone show [-h]
                        [-f {json,shell,table,value,yaml}]
                        [-c COLUMN] [--noindent]
                        [--prefix PREFIX]
                        [--max-width <integer>]
                        [--fit-width]
                        [--print-empty]
                        <availabilityzone>

Table 49.38. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone&gt;</td>
<td>Name of the availability zone.</td>
</tr>
</tbody>
</table>

Table 49.39. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.40. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format json,shell,table,value,yaml</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 49.41. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.42. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.43. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.12. LOADBALANCER AVAILABILITYZONE UNSET

Clear availability zone settings

Usage:

```
openstack loadbalancer availabilityzone unset [-h] [--description] <availabilityzone>
```

Table 49.44. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone&gt;</td>
<td>Name of the availability zone to update.</td>
</tr>
</tbody>
</table>

Table 49.45. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
49.13. LOADBALANCER AVAILABILITYZONEPROFILE CREATE

Create an octavia availability zone profile

**Usage:**

    openstack loadbalancer availabilityzoneprofile create
          [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent]
          [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty]
          --name <name> --provider <provider name> --availability-zone-data
          <availability_zone_data>

**Value**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--name &lt;name&gt;</td>
<td>New octavia availability zone profile name.</td>
</tr>
<tr>
<td>--provider &lt;provider name&gt;</td>
<td>Provider name for the availability zone profile.</td>
</tr>
<tr>
<td>--availability-zone-data</td>
<td>The json string containing the availability zone metadata.</td>
</tr>
</tbody>
</table>

**Summary**

| --description               | Clear the availability zone description. |

**Table 49.46. Command arguments**

**Table 49.47. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.48. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.49. Shell formatter options**
### Table 49.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 49.14. LOADBALANCER AVAILABILITYZONEPROFILE DELETE

Delete an availability zone profile

**Usage:**

```
openstack loadbalancer availabilityzoneprofile delete
  [-h] <availabilityzone_profile>
```

**Table 49.51. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone_profile&gt;</td>
<td>Availability zone profile to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 49.52. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### 49.15. LOADBALANCER AVAILABILITYZONEPROFILE LIST

List availability zone profiles

**Usage:**

```
openstack loadbalancer availabilityzoneprofile list [-h]
  [-f {csv, json, table, value, yaml}]```
Table 49.53. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List availabilityzone profiles by profile name.</td>
</tr>
<tr>
<td>--provider &lt;provider_name&gt;</td>
<td>List availability zone profiles according to their provider.</td>
</tr>
</tbody>
</table>

Table 49.54. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.55. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.57. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.16. LOADBALANCER AVAILABILITYZONEPROFILE SET

Update an availability zone profile

Usage:

```plaintext
openstack loadbalancer availabilityzoneprofile set [-h] [--name <name>] [--provider <provider_name>] [--availabilityzone-data <availabilityzone_data>] <availabilityzone_profile>
```

Table 49.58. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone_profile&gt;</td>
<td>Name or uuid of the availability zone profile to update.</td>
</tr>
</tbody>
</table>

Table 49.59. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the availability zone profile.</td>
</tr>
<tr>
<td>--provider &lt;provider_name&gt;</td>
<td>Set the provider of the availability zone profile.</td>
</tr>
<tr>
<td>--availabilityzone-data &lt;availabilityzone_data&gt;</td>
<td>Set the availability zone data of the profile.</td>
</tr>
</tbody>
</table>
Show the details of a single availability zone profile

**Usage:**

```
openstack loadbalancer availabilityzoneprofile show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN]
   [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width]
   [--print-empty]
   <availabilityzone_profile>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;availabilityzone_profile&gt;</td>
<td>Name or uuid of the availability zone profile to show.</td>
</tr>
</tbody>
</table>

**Table 49.60. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 49.61. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.62. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### Table 49.65. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 49.18. LOADBALANCER CREATE

Create a load balancer

**Usage:**

```bash
```

### Table 49.66. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New load balancer name.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set load balancer description.</td>
</tr>
<tr>
<td>--vip-address &lt;vip_address&gt;</td>
<td>Set the vip ip address.</td>
</tr>
<tr>
<td>--vip-qos-policy-id &lt;vip_qos_policy_id&gt;</td>
<td>Set qos policy id for vip port. unset with none.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project for the load balancer (name or id).</td>
</tr>
<tr>
<td>--provider &lt;provider&gt;</td>
<td>Provider name for the load balancer.</td>
</tr>
<tr>
<td>--availability-zone &lt;availability_zone&gt;</td>
<td>Availability zone for the load balancer.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable load balancer (default).</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable load balancer.</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>The name or id of the flavor for the load balancer.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the load balancer (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the load balancer</td>
</tr>
</tbody>
</table>

**Table 49.67. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.68. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.69. Shell formatter options**
Table 49.70. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 49.71. VIP Network

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least one of the following arguments is required. --vip-port-id &lt;vip_port_id&gt;</td>
<td>Set port for the load balancer (name or id).</td>
</tr>
<tr>
<td>--vip-subnet-id &lt;vip_subnet_id&gt;</td>
<td>Set subnet for the load balancer (name or id).</td>
</tr>
<tr>
<td>--vip-network-id &lt;vip_network_id&gt;</td>
<td>Set network for the load balancer (name or id).</td>
</tr>
</tbody>
</table>

49.19. LOADBALANCER DELETE

Delete a load balancer

Usage:

```bash
openstack loadbalancer delete [-h] [--cascade] [--wait] <load_balancer>
```

Table 49.72. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;load_balancer&gt;</td>
<td>Load balancers to delete (name or id)</td>
</tr>
</tbody>
</table>
49.20. LOADBALANCER FAILOVER

Trigger load balancer failover

Usage:

```
openstack loadbalancer failover [-h] [--wait] <load_balancer>
```

Table 49.74. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;load_balancer&gt;</td>
<td>Name or uuid of the load balancer.</td>
</tr>
</tbody>
</table>

Table 49.75. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

49.21. LOADBALANCER FLAVOR CREATE

Create a octavia flavor

Usage:

```
openstack loadbalancer flavor create [-h] 
[f {json,shell,table,value,yaml}] 
[-c COLUMN] [--noindent] 
[--prefix PREFIX] 
[--max-width <integer>] 
[--fit-width] [--print-empty] 
--name <name> --flavorprofile <flavor_profile> 
[--description <description>] 
[--enable | --disable]
```
### Table 49.76. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New flavor name.</td>
</tr>
<tr>
<td>--flavorprofile &lt;flavor_profile&gt;</td>
<td>Flavor profile to add the flavor to (name or id).</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set flavor description.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable flavor.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable flavor.</td>
</tr>
</tbody>
</table>

### Table 49.77. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 49.78. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 49.79. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 49.80. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
49.22. LOADBALANCER FLAVOR DELETE

Delete a flavor

Usage:

```
openstack loadbalancer flavor delete [-h] <flavor>
```

Table 49.81. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 49.82. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

49.23. LOADBALANCER FLAVOR LIST

List flavor

Usage:

```
```

Table 49.83. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List flavors according to their name.</td>
</tr>
<tr>
<td>--flavorprofile &lt;flavor_profile&gt;</td>
<td>List flavors according to their flavor profile.</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled flavors.</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled flavors.</td>
</tr>
</tbody>
</table>

**Table 49.84. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 49.85. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 49.86. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.87. Table formatter options**
Value | Summary
---|---
--max-width <integer> | Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width | Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.

--print-empty | Print empty table if there is no data to show.

### 49.24. LOADBALANCER FLAVOR SET

Update a flavor

**Usage:**

```bash
openstack loadbalancer flavor set [-h] [--name <name>] [--enable | --disable] <flavor>
```

**Table 49.88. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Name or uuid of the flavor to update.</td>
</tr>
</tbody>
</table>

**Table 49.89. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the flavor.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable flavor.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable flavor.</td>
</tr>
</tbody>
</table>

### 49.25. LOADBALANCER FLAVOR SHOW

Show the details for a single flavor

**Usage:**

```bash
openstack loadbalancer flavor show [-h]
```
Table 49.90. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Name or uuid of the flavor.</td>
</tr>
</tbody>
</table>

Table 49.91. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.92. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.93. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.94. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.95. Table formatter options
### 49.26. LOADBALANCER FLAVOR UNSET

Clear flavor settings

**Usage:**

```bash
openstack loadbalancer flavor unset [-h] [--description] <flavor>
```

**Table 49.96. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to update (name or id).</td>
</tr>
</tbody>
</table>

**Table 49.97. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description</td>
<td>Clear the flavor description.</td>
</tr>
</tbody>
</table>

### 49.27. LOADBALANCER FLAVORPROFILE CREATE

Create a octavia flavor profile

**Usage:**

```bash
openstack loadbalancer flavorprofile create [-h]
                                             [-f {json,shell,table,value,yaml}]
                                             [-c COLUMN] [--noindent]
                                             [--prefix PREFIX]
                                             [--max-width <integer>]
                                             [--fit-width]
                                             [--print-empty] --name
```
<name> --provider <provider name> --flavor-data <flavor_data>

Table 49.98. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New octavia flavor profile name.</td>
</tr>
<tr>
<td>--provider &lt;provider name&gt;</td>
<td>Provider name for the flavor profile.</td>
</tr>
<tr>
<td>--flavor-data &lt;flavor_data&gt;</td>
<td>The json string containing the flavor metadata.</td>
</tr>
</tbody>
</table>

Table 49.99. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.100. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.101. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.102. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 49.28. LOADBALANCER FLAVORPROFILE DELETE

Delete a flavor profile

**Usage:**

```
openstack loadbalancer flavorprofile delete [-h] <flavor_profile>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flavor_profile&gt;</code></td>
<td>Flavor profiles to delete (name or id)</td>
</tr>
</tbody>
</table>

### 49.29. LOADBALANCER FLAVORPROFILE LIST

List flavor profile

**Usage:**

```
openstack loadbalancer flavorprofile list [-h]  
  [-f {csv,json,table,value,yaml}]  
  [-c COLUMN]  
  [--quote {all,minimal,none,nonnumeric}]  
  [-noindent]  
  [-max-width <integer>]  
  [-fit-width] [-print-empty]  
  [--sort-column SORT_COLUMN]  
  [--sort-ascending | --sort-descending]  
  [-name <name>]  
  [-provider <provider_name>]  
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>List flavor profiles by flavor profile name.</td>
</tr>
<tr>
<td><code>--provider &lt;provider_name&gt;</code></td>
<td>List flavor profiles according to their provider.</td>
</tr>
</tbody>
</table>

Table 49.106. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.107. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.108. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.109. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
49.30. LOADBALANCER FLAVORPROFILE SET

Update a flavor profile

Usage:

openstack loadbalancer flavorprofile set [-h] [--name <name>]
[--provider <provider_name>]
[--flavor-data <flavor_data>]
<flavor_profile>

Table 49.110. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor_profile&gt;</td>
<td>Name or uuid of the flavor profile to update.</td>
</tr>
</tbody>
</table>

Table 49.111. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the flavor profile.</td>
</tr>
<tr>
<td>--provider &lt;provider_name&gt;</td>
<td>Set the provider of the flavor profile.</td>
</tr>
<tr>
<td>--flavor-data &lt;flavor_data&gt;</td>
<td>Set the flavor data of the flavor profile.</td>
</tr>
</tbody>
</table>

49.31. LOADBALANCER FLAVORPROFILE SHOW

Show the details for a single flavor profile

Usage:

openstack loadbalancer flavorprofile show [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
Table 49.112. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flavor_profile&gt;</code></td>
<td>Name or uuid of the flavor profile to show.</td>
</tr>
</tbody>
</table>

Table 49.113. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.114. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.115. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.116. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.117. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
49.32. LOADBALANCER HEALTHMONITOR CREATE

Create a health monitor

Usage:

```
openstack loadbalancer healthmonitor create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
[--name <name>] --delay <delay>
[--domain-name <domain_name>]
[--expected-codes <codes>]
[--http-method {GET,POST,DELETE,PUT,HEAD,OPTIONS,PATCH,CONNECT,TRACE}]
[--http-version <http_version>]
--timeout <timeout>
--max-retries <max_retries>
[--url-path <url_path>]
--type {PING,HTTP,TCP,HTTPS,TLS-HELLO,UDP-CONNECT,SCTP}
[--max-retries-down <max_retries_down>]
[--enable | --disable]
[--wait]
[--tag <tag> | --no-tag]
<pool>
```

Table 49.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Set the pool for the health monitor (name or id).</td>
</tr>
</tbody>
</table>

Table 49.119. Command arguments

<p>| --fit-width | Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable |
| --print-empty | Print empty table if there is no data to show. |</p>
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>Set the health monitor name.</td>
</tr>
<tr>
<td><code>--delay &lt;delay&gt;</code></td>
<td>Set the time in seconds, between sending probes to members.</td>
</tr>
<tr>
<td><code>--domain-name &lt;domain_name&gt;</code></td>
<td>Set the domain name, which be injected into the http Host Header to the backend server for HTTP health check.</td>
</tr>
<tr>
<td><code>--expected-codes &lt;codes&gt;</code></td>
<td>Set the list of http status codes expected in response from the member to declare it healthy.</td>
</tr>
<tr>
<td><code>--http-method</code></td>
<td>Set the http method that the health monitor uses for requests.</td>
</tr>
<tr>
<td><code>{GET,POST,DELETE,PUT,HEAD,OPTIONS,PATCH,CONNECT,TRACE}</code></td>
<td></td>
</tr>
<tr>
<td><code>--http-version &lt;http_version&gt;</code></td>
<td>Set the http version.</td>
</tr>
<tr>
<td><code>--timeout &lt;timeout&gt;</code></td>
<td>Set the maximum time, in seconds, that a monitor waits to connect before it times out. This value must be less than the delay value.</td>
</tr>
<tr>
<td><code>--max-retries &lt;max_retries&gt;</code></td>
<td>The number of successful checks before changing the operating status of the member to ONLINE.</td>
</tr>
<tr>
<td><code>--url-path &lt;url_path&gt;</code></td>
<td>Set the http url path of the request sent by the monitor to test the health of a backend member.</td>
</tr>
<tr>
<td><code>--type {PING,HTTP,TCP,HTTPS,TLS-HELLO,UDP-CONNECT,SCTP}</code></td>
<td>Set the health monitor type.</td>
</tr>
<tr>
<td><code>--max-retries-down &lt;max_retries_down&gt;</code></td>
<td>Set the number of allowed check failures before changing the operating status of the member to ERROR.</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable health monitor (default).</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable health monitor.</td>
</tr>
<tr>
<td><code>--wait</code></td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td><code>--tag &lt;tag&gt;</code></td>
<td>Tag to be added to the health monitor (repeat option to set multiple tags)</td>
</tr>
</tbody>
</table>
### Table 49.120. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--no-tag</td>
<td>No tags associated with the health monitor</td>
</tr>
</tbody>
</table>

The output format, defaults to table

- f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}

Specify the column(s) to include, can be repeated to show multiple columns

### Table 49.121. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 49.122. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 49.123. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 49.33. LOADBALANCER HEALTHMONITOR DELETE

Delete a health monitor

**Usage:**
openstack loadbalancer healthmonitor delete [-h] [--wait]
  <health_monitor>

Table 49.124. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;health_monitor&gt;</td>
<td>Health monitor to delete (name or id).</td>
</tr>
</tbody>
</table>

Table 49.125. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

49.34. LOADBALANCER HEALTHMONITOR LIST

List health monitors

Usage:

openstack loadbalancer healthmonitor list [-h]
  [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--tags <tag>[,<tag>,...]]
  [--any-tags <tag>[,<tag>,...]]
  [--not-tags <tag>[,<tag>,...]]
  [--not-any-tags <tag>[,<tag>,...]]

Table 49.126. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]]</td>
<td>List health monitor which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]]</td>
<td>List health monitor which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>
Table 49.127. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude health monitor which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude health monitor which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.128. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.129. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.130. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 49.35. LOADBALANCER HEALTHMONITOR SET

Update a health monitor

**Usage:**

```
openstack loadbalancer healthmonitor set [-h] [--name <name>]
   [-delay <delay>]
   [--domain-name <domain_name>]
   [--expected-codes <codes>]
   [--http-method {GET,POST,DELETE,PUT,HEAD,OPTIONS,PATCH,CONNECT,TRACE}]
   [--http-version <http_version>]
   [--timeout <timeout>]
   [--max-retries <max_retries>]
   [--max-retries-down <max_retries_down>]
   [--url-path <url_path>]
   [--enable] [--disable]
   [--wait] [--tag <tag>]
   [--no-tag]
   <health_monitor>
```

**Table 49.131. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;health_monitor&gt;</td>
<td>Health monitor to update (name or id).</td>
</tr>
</tbody>
</table>

**Table 49.132. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set health monitor name.</td>
</tr>
<tr>
<td>--delay &lt;delay&gt;</td>
<td>Set the time in seconds, between sending probes to members.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--domain-name &lt;domain_name&gt;</td>
<td>Set the domain name, which be injected into the http Host Header to the backend server for HTTP health check.</td>
</tr>
<tr>
<td>--expected-codes &lt;codes&gt;</td>
<td>Set the list of http status codes expected in response from the member to declare it healthy.</td>
</tr>
<tr>
<td>--http-method {GET,POST,DELETE,PUT,HEAD,OPTIONS,PATCH,CONNECT,TRACE}</td>
<td>Set the http method that the health monitor uses for requests.</td>
</tr>
<tr>
<td>--http-version &lt;http_version&gt;</td>
<td>Set the http version.</td>
</tr>
<tr>
<td>--timeout &lt;timeout&gt;</td>
<td>Set the maximum time, in seconds, that a monitor waits to connect before it times out. This value must be less than the delay value.</td>
</tr>
<tr>
<td>--max-retries &lt;max_retries&gt;</td>
<td>Set the number of successful checks before changing the operating status of the member to ONLINE.</td>
</tr>
<tr>
<td>--max-retries-down &lt;max_retries_down&gt;</td>
<td>Set the number of allowed check failures before changing the operating status of the member to ERROR.</td>
</tr>
<tr>
<td>--url-path &lt;url_path&gt;</td>
<td>Set the http url path of the request sent by the monitor to test the health of a backend member.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable health monitor.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable health monitor.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the health monitor (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the health monitor. specify both --tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>

49.36. LOADBALANCER HEALTHMONITOR SHOW

Show the details of a single health monitor

Usage:

Table 49.133. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;health_monitor&gt;</td>
<td>Name or uuid of the health monitor.</td>
</tr>
</tbody>
</table>

Table 49.134. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.135. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.136. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.137. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.138. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
**49.37. LOADBALANCER HEALTHMONITOR UNSET**

Clear health monitor settings

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 49.139. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;health_monitor&gt;</td>
<td>Health monitor to update (name or id).</td>
</tr>
</tbody>
</table>

**Table 49.140. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain-name</td>
<td>Clear the health monitor domain name.</td>
</tr>
<tr>
<td>--expected-codes</td>
<td>Reset the health monitor expected codes to the api default.</td>
</tr>
<tr>
<td>--http-method</td>
<td>Reset the health monitor http method to the api default.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--http-version</td>
<td>Reset the health monitor http version to the api default.</td>
</tr>
<tr>
<td>--max-retries-down</td>
<td>Reset the health monitor max retries down to the api default.</td>
</tr>
<tr>
<td>--name</td>
<td>Clear the health monitor name.</td>
</tr>
<tr>
<td>--url-path</td>
<td>Clear the health monitor url path.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the health monitor (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the health monitor</td>
</tr>
</tbody>
</table>

49.38. LOADBALANCER L7POLICY CREATE

Create a l7policy

Usage:

openstack loadbalancer l7policy create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [-noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--name <name>]
[--description <description>]
--action

(REDIRECT_TO_URL,REDIRECT_TO_POOL,REDIRECT_PREFIX,REJECT)
[--redirect-pool <pool> | --redirect-url <url> | --redirect-prefix <url>]
[--redirect-http-code <redirect_http_code>]
[--position <position>]
[--enable | --disable] [--wait]
[--tag <tag> | --no-tag]
<listener>

Table 49.141. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Listener to add l7policy to (name or id).</td>
</tr>
</tbody>
</table>
### Table 49.142. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>Set the l7policy name.</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>Set l7policy description.</td>
</tr>
<tr>
<td><code>--action</code></td>
<td>Set the action of the policy.</td>
</tr>
<tr>
<td><code>{REDIRECT_TO_URL,REDIRECT_TO_POOL,REDIRECT_PREFIX,REJECT}</code></td>
<td></td>
</tr>
<tr>
<td><code>--redirect-pool &lt;pool&gt;</code></td>
<td>Set the pool to redirect requests to (name or id).</td>
</tr>
<tr>
<td><code>--redirect-url &lt;url&gt;</code></td>
<td>Set the url to redirect requests to.</td>
</tr>
<tr>
<td><code>--redirect-prefix &lt;url&gt;</code></td>
<td>Set the url prefix to redirect requests to.</td>
</tr>
<tr>
<td><code>--redirect-http-code &lt;redirect_http_code&gt;</code></td>
<td>Set the http response code for redirect_url orREDIRECT_PREFIX action.</td>
</tr>
<tr>
<td><code>--position &lt;position&gt;</code></td>
<td>Sequence number of this l7 policy.</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable l7policy (default).</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable l7policy.</td>
</tr>
<tr>
<td><code>--wait</code></td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td><code>--tag &lt;tag&gt;</code></td>
<td>Tag to be added to the l7policy (repeat option to set multiple tags)</td>
</tr>
<tr>
<td><code>--no-tag</code></td>
<td>No tags associated with the l7policy</td>
</tr>
</tbody>
</table>

### Table 49.143. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 49.144. JSON formatter options
Table 49.145. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.146. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.39. LOADBALANCER L7POLICY DELETE

Delete a l7policy

Usage:

```
openstack loadbalancer l7policy delete [-h] [--wait] <policy>
```

Table 49.147. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>L7policy to delete (name or id).</td>
</tr>
</tbody>
</table>

Table 49.148. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>
49.40. LOADBALANCER L7POLICY LIST

List L7policies

Usage:

```
openstack loadbalancer l7policy list [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--listener LISTENER]
    [--tags <tag>[,<tag>,...]]
    [--any-tags <tag>[,<tag>,...]]
    [--not-tags <tag>[,<tag>,...]]
    [--not-any-tags <tag>[,<tag>,...]]
```

Table 49.149. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--listener LISTENER</td>
<td>List L7policies that applied to the given listener (name or ID).</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List L7policy which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List L7policy which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude L7policy which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude L7policy which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.150. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 49.151. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.152. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.153. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.41. LOADBALANCER L7POLICY SET

Update a l7policy

Usage:

```
```
Table 49.154. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>L7policy to update (name or id).</td>
</tr>
</tbody>
</table>

Table 49.155. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set l7policy name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set l7policy description.</td>
</tr>
<tr>
<td>--action</td>
<td>Set the action of the policy.</td>
</tr>
<tr>
<td></td>
<td>{REDIRECT_TO_URL,REDIRECT_TO_POOL,REDIRECT_PREFIX,REJECT}</td>
</tr>
<tr>
<td>--redirect-pool &lt;pool&gt;</td>
<td>Set the pool to redirect requests to (name or id).</td>
</tr>
<tr>
<td>--redirect-url &lt;url&gt;</td>
<td>Set the url to redirect requests to.</td>
</tr>
<tr>
<td>--redirect-prefix &lt;url&gt;</td>
<td>Set the url prefix to redirect requests to.</td>
</tr>
<tr>
<td>--redirect-http-code &lt;redirect_http_code&gt;</td>
<td>Set the http response code for redirect_url orREDIRECT_PREFIX action.</td>
</tr>
<tr>
<td>--position &lt;position&gt;</td>
<td>Set sequence number of this l7 policy.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable l7policy.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable l7policy.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the l7policy (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the l7policy. specify both - -tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>
49.42. LOADBALANCER L7POLICY SHOW

Show the details of a single l7policy

Usage:

```
openstack loadbalancer l7policy show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  <policy>
```

Table 49.156. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>Name or uuid of the l7policy.</td>
</tr>
</tbody>
</table>

Table 49.157. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.158. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.159. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.160. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 49.161. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.43. LOADBALANCER L7POLICY UNSET

Clear l7policy settings

Usage:

```
```

Table 49.162. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>L7policy to update (name or id).</td>
</tr>
</tbody>
</table>

Table 49.163. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description</td>
<td>Clear the l7policy description.</td>
</tr>
<tr>
<td>--name</td>
<td>Clear the l7policy name.</td>
</tr>
<tr>
<td>--redirect-http-code</td>
<td>Clear the l7policy redirect http code.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the l7policy (repeat option to remove multiple tags)</td>
</tr>
</tbody>
</table>
49.44. LOADBALANCER L7RULE CREATE

Create a l7rule

Usage:

```
openstack loadbalancer l7rule create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    --compare-type
    {REGEX,EQUAL_TO,CONTAINS,ENDS_WITH,STARTS_WITH}
    [--invert] --value <value>
    [--key <key>] --type
    {FILE_TYPE,PATH,COOKIE,HOST_NAME,HEADER,SSL_CONN_HAS_CERT,SSL_VERIFY_RESULT,SSL_DN_FIELD}
    [--enable | --disable] [--wait]
    [--tag <tag> | --no-tag]
    <l7policy>
```

Table 49.164. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;l7policy&gt;</td>
<td>L7policy to add l7rule to (name or id).</td>
</tr>
</tbody>
</table>

Table 49.165. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--compare-type</td>
<td>Set the compare type for the l7rule.</td>
</tr>
<tr>
<td>{REGEX,EQUAL_TO,CONTAINS,ENDS_WITH,STARTS_WITH}</td>
<td></td>
</tr>
<tr>
<td>--invert</td>
<td>Invert l7rule.</td>
</tr>
<tr>
<td>--value &lt;value&gt;</td>
<td>Set the rule value to match on.</td>
</tr>
<tr>
<td>--key &lt;key&gt;</td>
<td>Set the key for the l7rule's value to match on.</td>
</tr>
</tbody>
</table>

---all-tag
Clear all tags associated with the l7policy
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--type</code> {FILE_TYPE,PATH,COOKIE,HOST_NAME,HEADER,SSL_CONN_HAS_CERT,SSL_VERIFY_RESULT,SSL_DN_FIELD}</td>
<td>Set the type for the l7rule.</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable l7rule (default).</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable l7rule.</td>
</tr>
<tr>
<td><code>--wait</code></td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td><code>--tag &lt;tag&gt;</code></td>
<td>Tag to be added to the l7rule (repeat option to set multiple tags)</td>
</tr>
<tr>
<td><code>--no-tag</code></td>
<td>No tags associated with the l7rule</td>
</tr>
</tbody>
</table>

**Table 49.166. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.167. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.168. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.169. Table formatter options**
### 49.45. LOADBALANCER L7RULE DELETE

Delete a L7rule

**Usage:**

```
openstack loadbalancer l7rule delete [-h] [--wait] <l7policy> <rule_id>
```

**Table 49.170. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;l7policy&gt;</code></td>
<td>L7policy to delete rule from (name or id).</td>
</tr>
<tr>
<td><code>&lt;rule_id&gt;</code></td>
<td>L7rule to delete.</td>
</tr>
</tbody>
</table>

**Table 49.171. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

### 49.46. LOADBALANCER L7RULE LIST

List L7rules for L7policy

**Usage:**

```
openstack loadbalancer l7rule list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [--quote {all,minimal,none,nonnumeric}]
```
Table 49.172. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;l7policy&gt;</td>
<td>L7policy to list rules for (name or id).</td>
</tr>
</tbody>
</table>

Table 49.173. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List l7rule which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List l7rule which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude l7rule which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude l7rule which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.174. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.175. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.176. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.177. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.47. LOADBALANCER L7RULE SET

Update a l7rule

Usage:

```
openstack loadbalancer l7rule set [-h]
    [--compare-type
        {REGEX,EQUAL_TO,CONTAINS,ENDS_WITH,STARTS_WITH}]
        [--invert] [--value <value>]
        [--key <key>]
        [--type
            {FILE_TYPE,PATH,COOKIE,HOST_NAME,HEADER,SSL_CONN_HAS_CERT,SSL_VERIFY_RESULT,SSL_DN_FIELD}]
        [--enable | --disable] [--wait]
        [-tag <tag>] [-no-tag]
        <l7policy> <l7rule_id>
```
Table 49.178. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;l7policy&gt;</td>
<td>L7policy to update l7rule on (name or id).</td>
</tr>
<tr>
<td>&lt;l7rule_id&gt;</td>
<td>L7rule to update.</td>
</tr>
</tbody>
</table>

Table 49.179. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--compare-type</td>
<td>Set the compare type for the l7rule.</td>
</tr>
<tr>
<td>{REGEX,EQUAL_TO,CONTAINS,ENDS_WITH,STARTsWith}</td>
<td></td>
</tr>
<tr>
<td>--invert</td>
<td>Invert l7rule.</td>
</tr>
<tr>
<td>--value &lt;value&gt;</td>
<td>Set the rule value to match on.</td>
</tr>
<tr>
<td>--key &lt;key&gt;</td>
<td>Set the key for the l7rule's value to match on.</td>
</tr>
<tr>
<td>--type</td>
<td>Set the type for the l7rule.</td>
</tr>
<tr>
<td>{FILE_TYPE,PATH,COOKIE,HOST_NAME,HEADER,SSL_CONN_HAS_CERT,SSL_VERIFY_RESULT,SSL_DN_FIELD}</td>
<td></td>
</tr>
<tr>
<td>--enable</td>
<td>Enable l7rule.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable l7rule.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the l7rule (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the l7rule. specify both --tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>

49.48. LOADBALANCER L7RULE SHOW

Show the details of a single l7rule

Usage:

```
openstack loadbalancer l7rule show [-h]
[-f {json,shell,table,value,yaml}]
```
### Table 49.180. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;l7policy&gt;</code></td>
<td>L7policy to show rule from (name or id).</td>
</tr>
<tr>
<td><code>&lt;l7rule_id&gt;</code></td>
<td>L7rule to show.</td>
</tr>
</tbody>
</table>

### Table 49.181. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 49.182. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 49.183. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 49.184. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 49.185. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 49.49. LOADBALANCER L7RULE UNSET

Clear L7 rule settings

**Usage:**

```bash
openstack loadbalancer l7rule unset [-h] [--invert] [--key] [--wait]
[--tag <tag> | --all-tag]
<l7policy> <l7rule_id>
```

**Table 49.186. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;l7policy&gt;</code></td>
<td>L7 policy to update (name or id).</td>
</tr>
<tr>
<td><code>&lt;l7rule_id&gt;</code></td>
<td>L7 rule to update.</td>
</tr>
</tbody>
</table>

**Table 49.187. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--invert</td>
<td>Reset the L7 rule invert to the API default.</td>
</tr>
<tr>
<td>--key</td>
<td>Clear the L7 rule key.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the L7 rule (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the L7 rule</td>
</tr>
</tbody>
</table>
49.50. LOADBALANCER LIST

List load balancers

Usage:

```bash
openstack loadbalancer list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent] [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--name <name>] [--enable | --disable]
    [--project <project-id>]
    [--vip-network-id <vip_network_id>]
    [--vip-subnet-id <vip_subnet_id>]
    [--vip-qos-policy-id <vip_qos_policy_id>]
    [--vip-port-id <vip_port_id>]
    [--provisioning-status {ACTIVE,DELETED,ERROR,PENDING_CREATE,PENDING_UPDATE,PENDING_DELETE}]
    [--operating-status {ONLINE,DRAINING,OFFLINE,DEGRADED,ERROR,NO_MONITOR}]
    [--provider <provider>] [--flavor <flavor>]
    [--availability-zone <availability_zone>]
    [--tags <tag>,<tag>,...]
    [--any-tags <tag>,<tag>,...]
    [--not-tags <tag>,<tag>,...]
    [--not-any-tags <tag>,<tag>,...]
```

Table 49.188. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List load balancers according to their name.</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled load balancers.</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled load balancers.</td>
</tr>
<tr>
<td>--project &lt;project-id&gt;</td>
<td>List load balancers according to their project (name or ID).</td>
</tr>
<tr>
<td>--vip-network-id &lt;vip_network_id&gt;</td>
<td>List load balancers according to their vip network (name or ID).</td>
</tr>
<tr>
<td>--vip-subnet-id &lt;vip_subnet_id&gt;</td>
<td>List load balancers according to their vip subnet (name or ID).</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--vip-qos-policy-id &lt;vip_qos_policy_id&gt;</td>
<td>List load balancers according to their vip qos policy (name or ID).</td>
</tr>
<tr>
<td>--vip-port-id &lt;vip_port_id&gt;</td>
<td>List load balancers according to their vip port (name or ID).</td>
</tr>
<tr>
<td>--provisioning-status</td>
<td>List load balancers according to their provisioning status.</td>
</tr>
<tr>
<td>[ACTIVE,DELETED,ERROR,PENDING_CREATE,PENDING_UPDATE,PENDING_DELETE]</td>
<td></td>
</tr>
<tr>
<td>--operating-status</td>
<td>List load balancers according to their operating status.</td>
</tr>
<tr>
<td>[ONLINE,DRAINING,OFFLINE,DEGRADED,ERROR,N O_MONITOR]</td>
<td></td>
</tr>
<tr>
<td>--provider &lt;provider&gt;</td>
<td>List load balancers according to their provider.</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>List load balancers according to their flavor.</td>
</tr>
<tr>
<td>--availability-zone &lt;availability_zone&gt;</td>
<td>List load balancers according to their availability zone.</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List load balancer which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List load balancer which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude load balancer which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude load balancer which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>

### Table 49.189. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

Sort the column(s) in ascending order

Sort the column(s) in descending order

Table 49.190. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.191. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.192. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 49.51. LOADBALANCER LISTENER CREATE

Create a listener

**Usage:**

```bash
```
[--name <name>]
[--description <description>]
--protocol
{TCP,HTTP,HTTPS,TERMINATED_HTTPS,UDP,SCTP}
[--connection-limit <limit>]
[--default-pool <pool>]
[--default-tls-container-ref <container_ref>]
[--sni-container-refs [<container_ref> ...]]
[--insert-headers <header=value,...>]
--protocol-port <port>
[--timeout-client-data <timeout>]
[--timeout-member-connect <timeout>]
[--timeout-member-data <timeout>]
[--timeout-tcp-inspect <timeout>]
[--enable | --disable]
[--client-ca-tls-container-ref <container_ref>]
[--client-authentication {NONE,OPTIONAL,MANDATORY}]
[--client-crl-container-ref <client_crl_container_ref>]
[--allowed-cidr [<allowed_cidr>]]
[--wait]
[--tls-ciphers <tls_ciphers>]
[--tls-version [<tls_versions>]]
[--alpn-protocol [<alpn_protocols>]]
[--tag <tag> | --no-tag]
<loadbalancer>

Table 49.193. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;loadbalancer&gt;</td>
<td>Load balancer for the listener (name or id).</td>
</tr>
</tbody>
</table>

Table 49.194. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the listener name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set the description of this listener.</td>
</tr>
<tr>
<td>--protocol</td>
<td>The protocol for the listener.</td>
</tr>
<tr>
<td>{TCP,HTTP,HTTPS,TERMINATED_HTTPS,UDP,SCTP}</td>
<td></td>
</tr>
<tr>
<td>--connection-limit &lt;limit&gt;</td>
<td>Set the maximum number of connections permitted for this listener.</td>
</tr>
<tr>
<td>--default-pool &lt;pool&gt;</td>
<td>Set the name or id of the pool used by the listener if no L7 policies match.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--default-tls-container-ref &lt;container_ref&gt;</code></td>
<td>The uri to the key manager service secrets container containing the certificate and key for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td><code>--sni-container-refs [ &lt;container_ref&gt; ... ]</code></td>
<td>A list of uris to the key manager service secrets containers containing the certificates and keys for TERMINATED_TLS the listener using Server Name Indication.</td>
</tr>
<tr>
<td><code>--insert-headers &lt;header=value,...&gt;</code></td>
<td>A dictionary of optional headers to insert into the request before it is sent to the backend member.</td>
</tr>
<tr>
<td><code>--protocol-port &lt;port&gt;</code></td>
<td>Set the protocol port number for the listener.</td>
</tr>
<tr>
<td><code>--timeout-member-connect &lt;timeout&gt;</code></td>
<td>Backend member connection timeout in milliseconds. Default: 5000.</td>
</tr>
<tr>
<td><code>--timeout-member-data &lt;timeout&gt;</code></td>
<td>Backend member inactivity timeout in milliseconds. Default: 50000.</td>
</tr>
<tr>
<td><code>--timeout-tcp-inspect &lt;timeout&gt;</code></td>
<td>Time, in milliseconds, to wait for additional tcp packets for content inspection. Default: 0.</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable listener (default).</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable listener.</td>
</tr>
<tr>
<td><code>--client-ca-tls-container-ref &lt;container_ref&gt;</code></td>
<td>The uri to the key manager service secrets container containing the CA certificate for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td><code>--client-authentication {NONE,OPTIONAL,MANDATORY}</code></td>
<td>The tls client authentication verify options for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td><code>--client-crl-container-ref &lt;client_crl_container_ref&gt;</code></td>
<td>The uri to the key manager service secrets container containing the CA revocation list file for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td><code>--allowed-cidr [ &lt;allowed_cidr&gt;]</code></td>
<td>Cidr to allow access to the listener (can be set multiple times).</td>
</tr>
<tr>
<td><code>--wait</code></td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--tls-ciphers &lt;tls_ciphers&gt;</td>
<td>Set the tls ciphers to be used by the listener in OpenSSL format.</td>
</tr>
<tr>
<td>--tls-version [&lt;tls_versions&gt;]</td>
<td>Set the tls protocol version to be used by the listener (can be set multiple times).</td>
</tr>
<tr>
<td>--alpn-protocol [&lt;alpn_protocols&gt;]</td>
<td>Set the alpn protocol to be used by the listener (can be set multiple times).</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the listener (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the listener</td>
</tr>
</tbody>
</table>

**Table 49.195. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.196. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.197. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.198. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 49.52. LOADBALANCER LISTENER DELETE

Delete a listener

**Usage:**

```bash
openstack loadbalancer listener delete [-h] [--wait] <listener>
```

**Table 49.199. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Listener to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 49.200. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

### 49.53. LOADBALANCER LISTENER LIST

List listeners

**Usage:**

```bash
openstack loadbalancer listener list [-h]
   [-f {csv, json, table, value, yaml}]
   [-c COLUMN]
   [--quote {all, minimal, none, nonnumeric}]
   [--noindent]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   [--sort-column SORT_COLUMN]
   [--sort-ascending | --sort-descending]
   [--name <name>]
   [--loadbalancer <loadbalancer>]
```
Table 49.201. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List listeners by listener name.</td>
</tr>
<tr>
<td>--loadbalancer &lt;loadbalancer&gt;</td>
<td>Filter by load balancer (name or id).</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled listeners.</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled listeners.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List listeners by project id.</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>List listener which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>List listener which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>Exclude listener which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>Exclude listener which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.202. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 49.203. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 49.204. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 49.205. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 49.54. LOADBALANCER LISTENER SET

Update a listener

**Usage:**

```
openstack loadbalancer listener set [-h] [--name <name>] 
  [-d <description>] 
  [-c <connection-limit>] 
  [-p <default-pool>] 
  [-t <default-tls-container-ref>] 
  [-i <sni-container-refs>] 
  [-i <insert-headers>] 
  [-t <timeout-client-data>] 
  [-t <timeout-member-connect>] 
  [-t <timeout-member-data>]
```
```plaintext
[--timeout-tcp-inspect <timeout>]
[--enable | --disable]
[--client-ca-tls-container-ref <container_ref>]
[--client-authentication {NONE, OPTIONAL, MANDATORY}]
[--client-crl-container-ref <client_crl_container_ref>]
[--allowed-cidr <allowed_cidr>]
[--wait]
[--tls-ciphers <tls_ciphers>]
[--tls-version [<tls_versions>]]
[--alpn-protocol [alpn_protocols>]]
[--tag <tag>] [--no-tag]
<listener>
```

Table 49.206. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Listener to modify (name or id).</td>
</tr>
</tbody>
</table>

Table 49.207. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the listener name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set the description of this listener.</td>
</tr>
<tr>
<td>--connection-limit &lt;limit&gt;</td>
<td>The maximum number of connections permitted for this listener. Default value is -1 which represents infinite connections.</td>
</tr>
<tr>
<td>--default-pool &lt;pool&gt;</td>
<td>The id of the pool used by the listener if no l7 policies match.</td>
</tr>
<tr>
<td>--default-tls-container-ref &lt;container-ref&gt;</td>
<td>The uri to the key manager service secrets container containing the certificate and key for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td>--sni-container-refs [&lt;container-ref&gt; ...]</td>
<td>A list of uris to the key manager service secrets containers containing the certificates and keys for TERMINATED_TLS the listener using Server Name Indication.</td>
</tr>
<tr>
<td>--insert-headers &lt;header=value&gt;</td>
<td>A dictionary of optional headers to insert into the request before it is sent to the backend member.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--timeout-member-connect &lt;timeout&gt;</td>
<td>Backend member connection timeout in milliseconds. Default: 5000.</td>
</tr>
<tr>
<td>--timeout-member-data &lt;timeout&gt;</td>
<td>Backend member inactivity timeout in milliseconds. Default: 50000.</td>
</tr>
<tr>
<td>--timeout-tcp-inspect &lt;timeout&gt;</td>
<td>Time, in milliseconds, to wait for additional tcp packets for content inspection. Default: 0.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable listener.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable listener.</td>
</tr>
<tr>
<td>--client-ca-tls-container-ref &lt;container_ref&gt;</td>
<td>The uri to the key manager service secrets container containing the CA certificate for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td>--client-authentication {NONE,OPTIONAL,MANDATORY}</td>
<td>The tls client authentication verify options for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td>--client-crl-container-ref &lt;client_crl_container_ref&gt;</td>
<td>The uri to the key manager service secrets container containing the CA revocation list file for TERMINATED_TLS listeners.</td>
</tr>
<tr>
<td>--allowed-cidr [allowed_cidr]</td>
<td>Cidr to allow access to the listener (can be set multiple times).</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tls-ciphers &lt;tls_ciphers&gt;</td>
<td>Set the tls ciphers to be used by the listener in OpenSSL format.</td>
</tr>
<tr>
<td>--tls-version [&lt;tls_versions&gt;]</td>
<td>Set the tls protocol version to be used by the listener (can be set multiple times).</td>
</tr>
<tr>
<td>--alpn-protocol [&lt;alpn_protocols&gt;]</td>
<td>Set the alpn protocol to be used by the listener (can be set multiple times).</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the listener (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the listener. specify both -tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>

49.55. LOADBALANCER LISTENER SHOW

Show the details of a single listener
Usage:

openstack loadbalancer listener show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   <listener>

Table 49.208. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Name or uuid of the listener</td>
</tr>
</tbody>
</table>

Table 49.209. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.210. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.211. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.212. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.213. Table formatter options
### 49.56. LOADBALANCER LISTENER STATS SHOW

Shows the current statistics for a listener.

#### Usage:

```
```

#### Table 49.214. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Name or uuid of the listener</td>
</tr>
</tbody>
</table>

#### Table 49.215. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### Table 49.216. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### Table 49.217. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 49.218. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 49.219. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 49.57. LOADBALANCER LISTENER UNSET

Clear listener settings

**Usage:**

```bash
openstack loadbalancer listener unset [-h] [--name] [--description]
    [--connection-limit]
    [--default-pool]
    [--default-tls-container-ref]
    [--sni-container-refs]
    [--insert-headers]
    [--timeout-client-data]
    [--timeout-member-connect]
    [--timeout-member-data]
    [--timeout-tcp-inspect]
    [--client-ca-tls-container-ref]
    [--client-authentication]
    [--client-crl-container-ref]
    [--allowed-cidrs]
    [--tls-versions] [--tls-ciphers]
    [--wait] [--alpn-protocols]
    [--tag <tag> | --all-tag]
    <listener>
```

""
### Table 49.220. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;listener&gt;</td>
<td>Listener to modify (name or id).</td>
</tr>
</tbody>
</table>

### Table 49.221. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name</td>
<td>Clear the listener name.</td>
</tr>
<tr>
<td>--description</td>
<td>Clear the description of this listener.</td>
</tr>
<tr>
<td>--connection-limit</td>
<td>Reset the connection limit to the api default.</td>
</tr>
<tr>
<td>--default-pool</td>
<td>Clear the default pool from the listener.</td>
</tr>
<tr>
<td>--default-tls-container-ref</td>
<td>Remove the default tls container reference from the listener.</td>
</tr>
<tr>
<td>--sni-container-reffs</td>
<td>Remove the tls sni container references from the listener.</td>
</tr>
<tr>
<td>--insert-headers</td>
<td>Clear the insert headers from the listener.</td>
</tr>
<tr>
<td>--timeout-client-data</td>
<td>Reset the client data timeout to the api default.</td>
</tr>
<tr>
<td>--timeout-member-connect</td>
<td>Reset the member connect timeout to the api default.</td>
</tr>
<tr>
<td>--timeout-member-data</td>
<td>Reset the member data timeout to the api default.</td>
</tr>
<tr>
<td>--timeout-tcp-inspect</td>
<td>Reset the tcp inspection timeout to the api default.</td>
</tr>
<tr>
<td>--client-ca-tls-container-ref</td>
<td>Clear the client ca tls container reference from the listener.</td>
</tr>
<tr>
<td>--client-authentication</td>
<td>Reset the client authentication setting to the api default.</td>
</tr>
<tr>
<td>--client-crl-container-ref</td>
<td>Clear the client crl container reference from the listener.</td>
</tr>
<tr>
<td>--allowed-cidrs</td>
<td>Clear all allowed cidrs from the listener.</td>
</tr>
</tbody>
</table>
Clear all TLS versions from the listener.

Clear all TLS ciphers from the listener.

Wait for action to complete.

Clear all ALPN protocols from the listener.

Tag to be removed from the listener (repeat option to remove multiple tags)

Clear all tags associated with the listener

49.58. LOADBALANCER MEMBER CREATE

Creating a member in a pool

Usage:

```
openstack loadbalancer member create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--name <name>]
    [--disable-backup | --enable-backup]
    [--weight <weight>] --address <ip_address>
    [--subnet-id <subnet_id>]
    --protocol-port <protocol_port>
    [--monitor-port <monitor_port>]
    [--monitor-address <monitor_address>]
    [--enable | --disable] [--wait]
    [--tag <tag> | --no-tag]
    <pool>
```

Table 49.222. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Id or name of the pool to create the member for.</td>
</tr>
</tbody>
</table>

Table 49.223. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of the member.</td>
</tr>
<tr>
<td>--disable-backup</td>
<td>Disable member backup (default)</td>
</tr>
<tr>
<td>--enable-backup</td>
<td>Enable member backup</td>
</tr>
<tr>
<td>--weight &lt;weight&gt;</td>
<td>The weight of a member determines the portion of</td>
</tr>
<tr>
<td></td>
<td>requests or connections it services compared to</td>
</tr>
<tr>
<td></td>
<td>the other members of the pool.</td>
</tr>
<tr>
<td>--address &lt;ip_address&gt;</td>
<td>The ip address of the backend member server</td>
</tr>
<tr>
<td>--subnet-id &lt;subnet_id&gt;</td>
<td>The subnet id the member service is accessible from.</td>
</tr>
<tr>
<td>--protocol-port &lt;protocol_port&gt;</td>
<td>The protocol port number the backend member server is listening on.</td>
</tr>
<tr>
<td>--monitor-port &lt;monitor_port&gt;</td>
<td>An alternate protocol port used for health monitoring a backend member.</td>
</tr>
<tr>
<td>--monitor-address &lt;monitor_address&gt;</td>
<td>An alternate ip address used for health monitoring a backend member.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable member (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable member</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the member (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the member</td>
</tr>
</tbody>
</table>

**Table 49.224. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 49.225. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.226. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.227. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.59. LOADBALANCER MEMBER DELETE

Delete a member from a pool

Usage:

```
openstack loadbalancer member delete [-h] [--wait] <pool> <member>
```

Table 49.228. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool name or id to delete the member from.</td>
</tr>
<tr>
<td>&lt;member&gt;</td>
<td>Name or id of the member to be deleted.</td>
</tr>
</tbody>
</table>

Table 49.229. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
49.60. LOADBALANCER MEMBER LIST

List members in a pool

Usage:

openstack loadbalancer member list [-h]

[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent] [---max-width <integer>]
[--fit-width] [---print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--tags <tag>,<tag>,...]
[--any-tags <tag>,<tag>,...]
[--not-tags <tag>,<tag>,...]
[--not-any-tags <tag>,<tag>,...]

Value | Summary
--- | ---
--wait | Wait for action to complete

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>List member which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>List member which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>Exclude member which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;,&lt;tag&gt;,...</td>
<td>Exclude member which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.230. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool name or id to list the members of.</td>
</tr>
</tbody>
</table>

Table 49.231. Command arguments

Table 49.232. Output formatter options
Table 49.233. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.234. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.235. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.61. LOADBALANCER MEMBER SET

Update a member

Usage:
openstack loadbalancer member set [-h] [-name <name>]
[-disable-backup | --enable-backup]
[-weight <weight>]
[-monitor-port <monitor_port>]
[-monitor-address <monitor_address>]
[-enable | --disable] [--wait]
[--tag <tag>] [--no-tag]
<pool> <member>

Table 49.236. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool that the member to update belongs to (name or ID).</td>
</tr>
<tr>
<td>&lt;member&gt;</td>
<td>Name or id of the member to update</td>
</tr>
</tbody>
</table>

Table 49.237. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the member</td>
</tr>
<tr>
<td>--disable-backup</td>
<td>Disable member backup (default)</td>
</tr>
<tr>
<td>--enable-backup</td>
<td>Enable member backup</td>
</tr>
<tr>
<td>--weight &lt;weight&gt;</td>
<td>Set the weight of member in the pool</td>
</tr>
<tr>
<td>--monitor-port &lt;monitor_port&gt;</td>
<td>An alternate protocol port used for health monitoring a backend member</td>
</tr>
<tr>
<td>--monitor-address &lt;monitor_address&gt;</td>
<td>An alternate ip address used for health monitoring a backend member.</td>
</tr>
<tr>
<td>--enable</td>
<td>Set the admin_state_up to true</td>
</tr>
<tr>
<td>--disable</td>
<td>Set the admin_state_up to false</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the member (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the member. specify both --tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>
49.62. LOADBALANCER MEMBER SHOW

Shows details of a single Member

Usage:

```
openstack loadbalancer member show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    <pool> <member>
```

Table 49.238. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool name or id to show the members of.</td>
</tr>
<tr>
<td>&lt;member&gt;</td>
<td>Name or id of the member to show.</td>
</tr>
</tbody>
</table>

Table 49.239. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.240. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.241. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.242. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.243. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.63. LOADBALANCER MEMBER UNSET

Clear member settings

Usage:

```
```

Table 49.244. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool that the member to update belongs to (name or id).</td>
</tr>
<tr>
<td>&lt;member&gt;</td>
<td>Member to modify (name or id).</td>
</tr>
</tbody>
</table>

Table 49.245. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--backup</td>
<td>Clear the backup member flag.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>--monitor-address</td>
<td>Clear the member monitor address.</td>
</tr>
<tr>
<td>--monitor-port</td>
<td>Clear the member monitor port.</td>
</tr>
<tr>
<td>--name</td>
<td>Clear the member name.</td>
</tr>
<tr>
<td>--weight</td>
<td>Reset the member weight to the api default.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the member (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the member</td>
</tr>
</tbody>
</table>

### 49.64. LOADBALANCER POOL CREATE

Create a pool

**Usage:**

```bash

{TCP,HTTP,HTTPS,TERMINATED_HTTPS,PROXY,PROXYV2,UDP,SCTP}

(--listener <listener> | --loadbalancer <load_balancer>)

[--session-persistence <session persistence>]

--lb-algorithm

{SOURCE_IP,ROUND_ROBIN,LEAST_CONNECTIONS,SOURCE_IP_PORT}

[--enable | --disable]

[--tls-container-ref <container-ref>]

[--ca-tls-container-ref <ca_tls_container_ref>]

[--crl-container-ref <crl_container_ref>]

[--enable-tls | --disable-tls]

[--wait]

[--tls-ciphers <tls_ciphers>]

[--tls-version [<tls_versions>]]

[--alpn-protocol [<alpn_protocols>]]

[--tag <tag> | --no-tag]
```

Table 49.246. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set pool name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set pool description.</td>
</tr>
<tr>
<td>--protocol [TCP,HTTP,HTTPS,TERMINATED_HTTPS,PROXY,PROXYV2,UDP,SCTP]</td>
<td>Set the pool protocol.</td>
</tr>
<tr>
<td>--listener &lt;listener&gt;</td>
<td>Listener to add the pool to (name or id).</td>
</tr>
<tr>
<td>--loadbalancer &lt;load_balancer&gt;</td>
<td>Load balancer to add the pool to (name or id)</td>
</tr>
<tr>
<td>--session-persistence &lt;session persistence&gt;</td>
<td>Set the session persistence for the listener (key=value).</td>
</tr>
<tr>
<td>--lb-algorithm [SOURCE_IP,ROUND_ROBIN,LEAST_CONNECTION_S,SOURCE_IP_PORT]</td>
<td>Load balancing algorithm to use.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable pool (default).</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable pool.</td>
</tr>
<tr>
<td>--tls-container-ref &lt;container-ref&gt;</td>
<td>The reference to the key manager service secrets container containing the certificate and key for <code>tls_enabled</code> pools to re-encrypt the traffic to backend member servers.</td>
</tr>
<tr>
<td>--ca-tls-container-ref &lt;ca_tls_container_ref&gt;</td>
<td>The reference to the key manager service secrets container containing the CA certificate for <code>tls_enabled</code> pools to check the backend member servers certificates</td>
</tr>
<tr>
<td>--crl-container-ref &lt;crl_container_ref&gt;</td>
<td>The reference to the key manager service secrets container containing the CA revocation list file for <code>tls_enabled</code> pools to validate the backend member servers certificates</td>
</tr>
<tr>
<td>--enable-tls</td>
<td>Enable backend member re-encryption.</td>
</tr>
<tr>
<td>--disable-tls</td>
<td>Disable backend member re-encryption.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--tls-ciphers &lt;tls_ciphers&gt;</td>
<td>Set the tls ciphers to be used by the pool in openssl cipher string format.</td>
</tr>
<tr>
<td>--tls-version [&lt;tls_versions&gt;]</td>
<td>Set the tls protocol version to be used by the pool (can be set multiple times).</td>
</tr>
<tr>
<td>--alpn-protocol [&lt;alpn_protocols&gt;]</td>
<td>Set the alpn protocol to be used by the pool (can be set multiple times).</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the pool (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the pool</td>
</tr>
</tbody>
</table>

Table 49.247. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.248. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.249. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.250. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
49.65. LOADBALANCER POOL DELETE

Delete a pool

Usage:

```
openstack loadbalancer pool delete [-h] [--wait] <pool>
```

Table 49.251. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool to delete (name or id).</td>
</tr>
</tbody>
</table>

Table 49.252. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
</tbody>
</table>

49.66. LOADBALANCER POOL LIST

List pools

Usage:

```
```
[--any-tags <tag>[,<tag>,...]]
[--not-tags <tag>[,<tag>,...]]
[--not-any-tags <tag>[,<tag>,...]]

Table 49.253. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--loadbalancer &lt;loadbalancer&gt;</td>
<td>Filter by load balancer (name or id).</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List pool which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List pool which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude pool which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude pool which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 49.254. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.255. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 49.256. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.257. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.67. LOADBALANCER POOL SET

Update a pool

Usage:

```
```

Table 49.258. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool to update (name or id).</td>
</tr>
</tbody>
</table>

Table 49.259. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the name of the pool.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set the description of the pool.</td>
</tr>
<tr>
<td>--session-persistence &lt;session_persistence&gt;</td>
<td>Set the session persistence for the listener (key=value).</td>
</tr>
<tr>
<td>--lb-algorithm [SOURCE_IP,ROUND_ROBIN,LEAST_CONNECTION_S,SOURCE_IP_PORT]</td>
<td>Set the load balancing algorithm to use.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable pool.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable pool.</td>
</tr>
<tr>
<td>--tls-container-ref &lt;container-ref&gt;</td>
<td>The uri to the key manager service secrets container containing the certificate and key for TERMINATED_TLS pools to re-encrypt the traffic from TERMINATED_TLS listener to backend servers.</td>
</tr>
<tr>
<td>--ca-tls-container-ref &lt;ca_tls_container_ref&gt;</td>
<td>The uri to the key manager service secrets container containing the CA certificate for TERMINATED_TLS listeners to check the backend servers certificates in ssl traffic.</td>
</tr>
<tr>
<td>--crl-container-ref &lt;crl_container_ref&gt;</td>
<td>The uri to the key manager service secrets container containing the CA revocation list file for TERMINATED_TLS listeners to valid the backend servers certificates in ssl traffic.</td>
</tr>
<tr>
<td>--enable-tls</td>
<td>Enable backend associated members re-encryption.</td>
</tr>
<tr>
<td>--disable-tls</td>
<td>Disable backend associated members re-encryption.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tls-ciphers &lt;tls_ciphers&gt;</td>
<td>Set the tls ciphers to be used by the pool in openssl cipher string format.</td>
</tr>
<tr>
<td>--tls-version [&lt;tls_versions&gt;]</td>
<td>Set the tls protocol version to be used by the pool (can be set multiple times).</td>
</tr>
<tr>
<td>--alpn-protocol [&lt;alpn_protocols&gt;]</td>
<td>Set the alpn protocol to be used by the pool (can be set multiple times).</td>
</tr>
</tbody>
</table>
49.68. LOADBALANCER POOL SHOW

Show the details of a single pool

Usage:

    openstack loadbalancer pool show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    <pool>

Table 49.260. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Name or uuid of the pool.</td>
</tr>
</tbody>
</table>

Table 49.261. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.262. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.263. JSON formatter options
Whether to disable indenting the json

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.264. Shell formatter options**

Add a prefix to all variable names

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.265. Table formatter options**

Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>

Print empty table if there is no data to show.

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**49.69. LOADBALANCER POOL UNSET**

Clear pool settings

**Usage:**

```bash
openstack loadbalancer pool unset [-h] [--name] [--description] 
[-ca-tls-container-ref] 
[-crl-container-ref] 
[--session-persistence] 
[-tls-container-ref] 
[-tls-versions] [-tls-ciphers] 
[--wait] [--alpn-protocols] 
[-tag <tag> | --all-tag] 
<pool>
```

**Table 49.266. Positional arguments**

Pool to modify (name or id).

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool&gt;</td>
<td>Pool to modify (name or id).</td>
</tr>
</tbody>
</table>

**Table 49.267. Command arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name</td>
<td>Clear the pool name.</td>
</tr>
<tr>
<td>--description</td>
<td>Clear the description of this pool.</td>
</tr>
<tr>
<td>--ca-tls-container-ref</td>
<td>Clear the certificate authority certificate reference on this pool.</td>
</tr>
<tr>
<td>--crl-container-ref</td>
<td>Clear the certificate revocation list reference on this pool.</td>
</tr>
<tr>
<td>--session-persistence</td>
<td>Disables session persistence on the pool.</td>
</tr>
<tr>
<td>--tls-container-ref</td>
<td>Clear the certificate reference for this pool.</td>
</tr>
<tr>
<td>--tls-versions</td>
<td>Clear all tls versions from the pool.</td>
</tr>
<tr>
<td>--tls-ciphers</td>
<td>Clear all tls ciphers from the pool.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--alpn-protocols</td>
<td>Clear all alpn protocols from the pool.</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the pool (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the pool</td>
</tr>
</tbody>
</table>

### 49.70. LOADBALANCER PROVIDER CAPABILITY LIST

List specified provider driver’s capabilities.

**Usage:**

```bash
openstack loadbalancer provider capability list [-h]  
[-f {csv, json, table, value, yaml}]  
[-c COLUMN]  
[--quote {all, minimal, none, nonnumeric}]  
[--noindent]  
[--max-width <integer>]  
[--fit-width]  
[--print-empty]  
[--sort-column SORT_COLUMN]  
[--sort-ascending | --sort-descending]  
[--flavor | --availability-zone]  
<provider_name>
```
### Table 49.268. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;provider_name&gt;</code></td>
<td>Name of the provider driver.</td>
</tr>
</tbody>
</table>

### Table 49.269. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--flavor</code></td>
<td>Get capabilities for flavor only.</td>
</tr>
<tr>
<td><code>--availability-zone</code></td>
<td>Get capabilities for availability zone only.</td>
</tr>
</tbody>
</table>

### Table 49.270. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>--column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 49.271. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 49.272. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 49.273. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.71. LOADBALANCER PROVIDER LIST

List all providers

Usage:

openstack loadbalancer provider list [-h]  
[ -f {csv,json,table,value,yaml}]  
[ -c COLUMN]  
[ --quote {all,minimal,none,nonnumeric}]  
[ --noindent]  
[ --max-width <integer>]  
[ --fit-width] [ --print-empty]  
[ --sort-column SORT_COLUMN]  
[ --sort-ascending | --sort-descending]

Table 49.274. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.275. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
Table 49.276. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.277. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.278. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

49.72. LOADBALANCER QUOTA DEFAULTS SHOW

Show quota defaults

Usage:

```
```

Table 49.279. Command arguments
- `h, --help`  
  Show this help message and exit

**Table 49.280. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format</code> {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.281. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.282. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.283. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**49.73. LOADBALANCER QUOTA LIST**

List quotas

**Usage:**

Table 49.284. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project-id&gt;</td>
<td>Name or uuid of the project.</td>
</tr>
</tbody>
</table>

Table 49.285. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 49.286. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 49.287. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.288. Table formatter options
49.74. LOADBALANCER QUOTA RESET

Resets quotas to default quotas

Usage:

openstack loadbalancer quota reset [-h] <project>

Table 49.289. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Project to reset quotas (name or id)</td>
</tr>
</tbody>
</table>

Table 49.290. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

49.75. LOADBALANCER QUOTA SET

Update a quota

Usage:

openstack loadbalancer quota set [-h]
   [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent]
   [--prefix PREFIX] [--max-width <integer>] [--fit-width]
   [--print-empty] [--healthmonitor <health_monitor>]
   [--listener <listener>]
   [--loadbalancer <load_balancer>]
   [--member <member>] [--pool <pool>]
Table 49.291. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Name or uuid of the project.</td>
</tr>
</tbody>
</table>

Table 49.292. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.293. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.294. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.295. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 49.296. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 49.297. Quota limits

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>At least one of the following arguments is required.--healthmonitor &lt;health_monitor&gt;</td>
<td>New value for the health monitor quota. value -1 means unlimited.</td>
</tr>
<tr>
<td>--listener &lt;listener&gt;</td>
<td>New value for the listener quota. value -1 means unlimited.</td>
</tr>
<tr>
<td>--loadbalancer &lt;load_balancer&gt;</td>
<td>New value for the load balancer quota limit. value -1 means unlimited.</td>
</tr>
<tr>
<td>--member &lt;member&gt;</td>
<td>New value for the member quota limit. value -1 means unlimited.</td>
</tr>
<tr>
<td>--pool &lt;pool&gt;</td>
<td>New value for the pool quota limit. value -1 means unlimited.</td>
</tr>
<tr>
<td>--l7policy &lt;l7policy&gt;</td>
<td>New value for the l7policy quota limit. value -1 means unlimited.</td>
</tr>
<tr>
<td>--l7rule &lt;l7rule&gt;</td>
<td>New value for the l7rule quota limit. value -1 means unlimited.</td>
</tr>
</tbody>
</table>

49.76. LOADBALANCER QUOTA SHOW

Show the quota details for a project

Usage:

```
openstack loadbalancer quota show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  <project>
```

Table 49.298. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Name or uuid of the project.</td>
</tr>
</tbody>
</table>

**Table 49.299. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 49.300. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.301. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.302. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.303. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
49.77. LOADBALANCER QUOTA UNSET

Clear quota settings

Usage:

```
openstack loadbalancer quota unset [-h] [--loadbalancer] [--listener] 
[--pool] [--member] 
[--healthmonitor] [--l7policy] 
[--l7rule]
<project>
```

Table 49.304. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Name or uuid of the project.</td>
</tr>
</tbody>
</table>

Table 49.305. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--loadbalancer</td>
<td>Reset the load balancer quota to the default.</td>
</tr>
<tr>
<td>--listener</td>
<td>Reset the listener quota to the default.</td>
</tr>
<tr>
<td>--pool</td>
<td>Reset the pool quota to the default.</td>
</tr>
<tr>
<td>--member</td>
<td>Reset the member quota to the default.</td>
</tr>
<tr>
<td>--healthmonitor</td>
<td>Reset the health monitor quota to the default.</td>
</tr>
<tr>
<td>--l7policy</td>
<td>Reset the l7policy quota to the default.</td>
</tr>
<tr>
<td>--l7rule</td>
<td>Reset the l7rule quota to the default.</td>
</tr>
</tbody>
</table>

49.78. LOADBALANCER SET

Update a load balancer

Usage:

```
openstack loadbalancer set [-h] [--name <name>] 
[--description <description>]
[--vip-qos-policy-id <vip_qos_policy_id>]
```
Table 49.306. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;load_balancer&gt;</td>
<td>Name or uuid of the load balancer to update.</td>
</tr>
</tbody>
</table>

Table 49.307. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set load balancer name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set load balancer description.</td>
</tr>
<tr>
<td>--vip-qos-policy-id &lt;vip_qos_policy_id&gt;</td>
<td>Set qos policy id for vip port. unset with none.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable load balancer.</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable load balancer.</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the load balancer (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the load balancer. specify both --tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>

49.79. LOADBALANCER SHOW

Show the details for a single load balancer

Usage:

```
```

Table 49.308. Positional arguments
### Value

| <load_balancer> | Name or uuid of the load balancer. |

**Table 49.309. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 49.310. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 49.311. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 49.312. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 49.313. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
49.80. LOADBALANCER STATS SHOW

Shows the current statistics for a load balancer

Usage:

```
```

Table 49.314. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;load_balancer&gt;</code></td>
<td>Name or uuid of the load balancer.</td>
</tr>
</tbody>
</table>

Table 49.315. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 49.316. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 49.317. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 49.318. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
### 49.81. LOADBALANCER STATUS SHOW

Display load balancer status tree in json format

**Usage:**

```bash
openstack loadbalancer status show [-h] <load_balancer>
```

#### Table 49.320. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;load_balancer&gt;</td>
<td>Name or uuid of the load balancer.</td>
</tr>
</tbody>
</table>

#### Table 49.321. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 49.82. LOADBALANCER UNSET

Clear load balancer settings

**Usage:**

```bash
```

#### Table 49.322. Positional arguments
<table>
<thead>
<tr>
<th>Name or uuid of the load balancer to update.</th>
</tr>
</thead>
</table>

### Table 49.323. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name</code></td>
<td>Clear the load balancer name.</td>
</tr>
<tr>
<td><code>--description</code></td>
<td>Clear the load balancer description.</td>
</tr>
<tr>
<td><code>--vip-qos-policy-id</code></td>
<td>Clear the load balancer qos policy.</td>
</tr>
<tr>
<td><code>--wait</code></td>
<td>Wait for action to complete</td>
</tr>
<tr>
<td><code>--tag &lt;tag&gt;</code></td>
<td>Tag to be removed from the load balancer (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td><code>--all-tag</code></td>
<td>Clear all tags associated with the load balancer</td>
</tr>
</tbody>
</table>
CHAPTER 50. MAPPING

This chapter describes the commands under the `mapping` command.

50.1. MAPPING CREATE

Create new mapping

Usage:

```
```

Table 50.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New mapping name (must be unique)</td>
</tr>
</tbody>
</table>

Table 50.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--rules &lt;filename&gt;</td>
<td>Filename that contains a set of mapping rules (required)</td>
</tr>
</tbody>
</table>

Table 50.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 50.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 50.5. Shell formatter options
50.2. MAPPING DELETE

Delete mapping(s)

Usage:

```
openstack mapping delete [-h] <mapping> [<mapping> ...]
```

Table 50.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mapping&gt;</td>
<td>Mapping(s) to delete</td>
</tr>
</tbody>
</table>

Table 50.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

50.3. MAPPING LIST

List mappings

Usage:

```
openstack mapping list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [--quote {all, minimal, none, nonnumeric}]
```
Table 50.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 50.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 50.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 50.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 50.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 50.4. MAPPING SET

Set mapping properties

**Usage:**

```
openstack mapping set [-h] [--rules <filename>] <name>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Mapping to modify</td>
</tr>
</tbody>
</table>

#### Table 50.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--rules &lt;filename&gt;</td>
<td>Filename that contains a new set of mapping rules</td>
</tr>
</tbody>
</table>

### 50.5. MAPPING SHOW

Display mapping details

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mapping&gt;</td>
<td></td>
</tr>
</tbody>
</table>
### Table 50.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;mapping&gt;</td>
<td>Mapping to display</td>
</tr>
</tbody>
</table>

### Table 50.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 50.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 50.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 50.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 51. MESSAGING

This chapter describes the commands under the `messaging` command.

51.1. MESSAGING CLAIM CREATE

Create claim and return a list of claimed messages

Usage:

```
<queue_name>
```

Table 51.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;queue_name&gt;</code></td>
<td>Name of the queue to be claim</td>
</tr>
</tbody>
</table>

Table 51.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--ttl &lt;ttl&gt;</code></td>
<td>Time to live in seconds for claim</td>
</tr>
<tr>
<td><code>--grace &lt;grace&gt;</code></td>
<td>The message grace period in seconds</td>
</tr>
<tr>
<td><code>--limit &lt;limit&gt;</code></td>
<td>Claims a set of messages, up to limit</td>
</tr>
</tbody>
</table>

Table 51.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 51.4. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 51.5. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.2. MESSAGING CLAIM QUERY

Display claim details

Usage:

```bash
openstack messaging claim query [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [--quote {all, minimal, none, nonnumeric}] [--noindent] [-max-width <integer>]
```
### Table 51.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;queue_name&gt;</code></td>
<td>Name of the claimed queue</td>
</tr>
<tr>
<td><code>&lt;claim_id&gt;</code></td>
<td>Id of the claim</td>
</tr>
</tbody>
</table>

### Table 51.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 51.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 51.10. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 51.11. JSON formatter options
### Table 51.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 51.3. MESSAGING CLAIM RELEASE

Delete a claim

**Usage:**

```
openstack messaging claim release [-h] <queue_name> <claim_id>
```

**Table 51.13. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the claimed queue</td>
</tr>
<tr>
<td>&lt;claim_id&gt;</td>
<td>Claim id to delete</td>
</tr>
</tbody>
</table>

**Table 51.14. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 51.4. MESSAGING CLAIM RENEW

Renew a claim

**Usage:**

```
openstack messaging claim renew [-h] [-f {csv, json, table, value, yaml}]`
```
Table 51.15. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the claimed queue</td>
</tr>
<tr>
<td>&lt;claim_id&gt;</td>
<td>Claim id</td>
</tr>
</tbody>
</table>

Table 51.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--ttl &lt;ttl&gt;</td>
<td>Time to live in seconds for claim</td>
</tr>
<tr>
<td>--grace &lt;grace&gt;</td>
<td>The message grace period in seconds</td>
</tr>
</tbody>
</table>

Table 51.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv,json,table,value,yaml], --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 51.18. CSV formatter options
### Table 51.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 51.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 51.5. MESSAGING FLAVOR CREATE

Create a pool flavor

**Usage:**

```
```

### Table 51.21. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor_name&gt;</td>
<td>Name of the flavor</td>
</tr>
</tbody>
</table>

### Table 51.22. Command arguments
Table 51.23. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pool_list &lt;pool_list&gt;</td>
<td>Pool list for flavor</td>
</tr>
<tr>
<td>--capabilities &lt;capabilities&gt;</td>
<td>Describes flavor-specific capabilities, this option is only available in client api version &lt; 2</td>
</tr>
</tbody>
</table>

Table 51.24. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.25. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.26. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.6. MESSAGING FLAVOR DELETE
Delete a pool flavor

Usage:

openstack messaging flavor delete [-h] <flavor_name>

Table 51.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor_name&gt;</td>
<td>Name of the flavor</td>
</tr>
</tbody>
</table>

Table 51.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

51.7. MESSAGING FLAVOR LIST

List available pool flavors

Usage:


Table 51.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker &lt;flavor_name&gt;</td>
<td>Flavor's paging marker</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Page size limit</td>
</tr>
<tr>
<td>--detailed</td>
<td>If show detailed capabilities of flavor</td>
</tr>
</tbody>
</table>

Table 51.30. Output formatter options
Value | Summary
--- | ---
-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml} | The output format, defaults to table
-c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns
--sort-column SORT_COLUMN | Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
--sort-ascending | Sort the column(s) in ascending order
--sort-descending | Sort the column(s) in descending order

Table 51.31. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 51.32. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.33. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.8. MESSAGING FLAVOR SHOW

Display pool flavor details

Usage:
openstack messaging flavor show [-h] [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  <flavor_name>

Table 51.34. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor_name&gt;</td>
<td>Flavor to display (name)</td>
</tr>
</tbody>
</table>

Table 51.35. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.36. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.37. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.38. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.39. Table formatter options
51.9. MESSAGING FLAVOR UPDATE

Update a flavor’s attributes

Usage:

openstack messaging flavor update [-h]
                [-f {json,shell,table,value,yaml}]
                [-c COLUMN] [--noindent]
                [--prefix PREFIX]
                [--max-width <integer>] [--fit-width]
                [--print-empty]
                [--pool_list <pool_list>]
                [--capabilities <capabilities>]
                <flavor_name>

Table 51.40. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor_name&gt;</td>
<td>Name of the flavor</td>
</tr>
</tbody>
</table>

Table 51.41. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pool_list &lt;pool_list&gt;</td>
<td>Pool list the flavor sits on</td>
</tr>
<tr>
<td>--capabilities &lt;capabilities&gt;</td>
<td>Describes flavor-specific capabilities.</td>
</tr>
</tbody>
</table>

Table 51.42. Output formatter options
Table 51.43. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.44. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.45. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.10. MESSAGING HEALTH

Display detailed health status of Zaqar server

Usage:

openstack messaging health [-h]

Table 51.46. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
51.11. MESSAGING HOMEDOC

Display detailed resource doc of Zaqar server

Usage:

openstack messaging homedoc [-h]

Table 51.47. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

51.12. MESSAGING MESSAGE LIST

List all messages for a given queue

Usage:


Table 51.48. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.49. Command arguments
Table 51.50. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv, json, table, value, yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 51.51. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 51.52. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 51.53. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.13. MESSAGING MESSAGE POST

Post messages for a given queue

Usage:

```
openstack messaging message post [-h] [--client-id <client_id>] <queue_name> <messages>
```

Table 51.54. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
<tr>
<td>&lt;messages&gt;</td>
<td>Messages to be posted.</td>
</tr>
</tbody>
</table>

Table 51.55. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--client-id &lt;client_id&gt;</td>
<td>A uuid for each client instance.</td>
</tr>
</tbody>
</table>

51.14. MESSAGING PING

Check if Zaqar server is alive or not

Usage:

```
```
### Table 51.56. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 51.57. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 51.58. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 51.59. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 51.60. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 51.15. MESSAGING POOL CREATE

Create a pool
Usage:

```
```

Table 51.61. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool_name&gt;</td>
<td>Name of the pool</td>
</tr>
<tr>
<td>&lt;pool_uri&gt;</td>
<td>Storage engine uri</td>
</tr>
<tr>
<td>&lt;pool_weight&gt;</td>
<td>Weight of the pool</td>
</tr>
</tbody>
</table>

Table 51.62. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>Flavor of the pool</td>
</tr>
<tr>
<td>--pool_options &lt;pool_options&gt;</td>
<td>An optional request component related to storage-specific options</td>
</tr>
</tbody>
</table>

Table 51.63. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.64. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.65. Shell formatter options
### Table 51.66. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 51.16. MESSAGING POOL DELETE

Delete a pool

**Usage:**

```
openstack messaging pool delete [-h] <pool_name>
```

**Table 51.67. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool_name&gt;</td>
<td>Name of the pool</td>
</tr>
</tbody>
</table>

**Table 51.68. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 51.17. MESSAGING POOL LIST

List available Pools

**Usage:**

```
openstack messaging pool list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [-quote {all,minimal,none,nonnumeric}]
```
Table 51.69. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker &lt;pool_name&gt;</td>
<td>Pool’s paging marker</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Page size limit</td>
</tr>
<tr>
<td>--detailed</td>
<td>Detailed output</td>
</tr>
</tbody>
</table>

Table 51.70. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 51.71. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 51.72. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 51.73. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.18. MESSAGING POOL SHOW

Display pool details

Usage:

```
```

Table 51.74. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool_name&gt;</td>
<td>Pool to display (name)</td>
</tr>
</tbody>
</table>

Table 51.75. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.76. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 51.77. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.78. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.79. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.19. MESSAGING POOL UPDATE

Update a pool attribute

Usage:

```
```

Table 51.80. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;pool_name&gt;</td>
<td>Name of the pool</td>
</tr>
</tbody>
</table>
### Table 51.81. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pool_uri &lt;pool_uri&gt;</td>
<td>Storage engine uri</td>
</tr>
<tr>
<td>--pool_weight &lt;pool_weight&gt;</td>
<td>Weight of the pool</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>Flavor of the pool</td>
</tr>
<tr>
<td>--pool_options &lt;pool_options&gt;</td>
<td>An optional request component related to storage-specific options</td>
</tr>
</tbody>
</table>

### Table 51.82. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 51.83. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 51.84. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 51.85. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
**51.20. MESSAGING QUEUE CREATE**

Create a queue

**Usage:**

```
openstack messaging queue create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [-noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty]
<queue_name>
```

Table 51.86. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.87. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.88. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.89. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.90. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.91. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.21. MESSAGING QUEUE DELETE

Delete a queue

Usage:

```
openstack messaging queue delete [-h] <queue_name>
```

Table 51.92. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.93. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

51.22. MESSAGING QUEUE GET METADATA
Get queue metadata

Usage:

```
openstack messaging queue get metadata [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [-noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   <queue_name>
```

Table 51.94. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.95. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.96. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.97. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.98. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.99. Table formatter options
51.23. MESSAGING QUEUE LIST

List available queues

Usage:


Table 51.100. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker &lt;queue_id&gt;</td>
<td>Queue’s paging marker</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Page size limit</td>
</tr>
<tr>
<td>--detailed</td>
<td>If show detailed information of queue</td>
</tr>
<tr>
<td>--with_count</td>
<td>If show amount information of queue</td>
</tr>
</tbody>
</table>

Table 51.101. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### 51.102. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### 51.103. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 51.104. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 51.24. MESSAGING QUEUE PURGE

**Purge a queue**

**Usage:**
openstack messaging queue purge [-h]
    [--resource_types <resource_types>]
    <queue_name>

Table 51.105. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.106. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource_types &lt;resource_types&gt;</td>
<td>Resource types want to be purged.</td>
</tr>
</tbody>
</table>

51.25. MESSAGING QUEUE SET METADATA

Set queue metadata

Usage:

openstack messaging queue set metadata [-h]
    <queue_name> <queue_metadata>

Table 51.107. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
<tr>
<td>&lt;queue_metadata&gt;</td>
<td>Queue metadata, all the metadata of the queue will be replaced by queue_metadata</td>
</tr>
</tbody>
</table>

Table 51.108. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

51.26. MESSAGING QUEUE SIGNED URL

Create a pre-signed url

Usage:

-
openstack messaging queue signed url [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--paths <paths>]
[--ttl-seconds <ttl_seconds>]
[--methods <methods>]
<queue_name>

Table 51.109. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.110. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--paths &lt;paths&gt;</td>
<td>Allowed paths in a comma-separated list. options: messages, subscriptions, claims</td>
</tr>
<tr>
<td>--ttl-seconds &lt;ttl_seconds&gt;</td>
<td>Length of time (in seconds) until the signature expires</td>
</tr>
<tr>
<td>--methods &lt;methods&gt;</td>
<td>Http methods to allow as a comma-separated list. Options: GET, HEAD, OPTIONS, POST, PUT, DELETE</td>
</tr>
</tbody>
</table>

Table 51.111. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 51.112. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 51.113. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.114. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.27. MESSAGING QUEUE STATS

Get queue stats

Usage:

```
```

Table 51.115. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue</td>
</tr>
</tbody>
</table>

Table 51.116. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.117. Output formatter options
Table 51.118. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 51.119. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 51.120. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the \texttt{CLIFF_MAX_TERM_WIDTH} environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable \texttt{CLIFF_FIT_WIDTH}=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.28. MESSAGING SUBSCRIPTION CREATE

Create a subscription for queue

Usage:

```
```
### Table 51.121. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;queue_name&gt;</code></td>
<td>Name of the queue to subscribe to</td>
</tr>
<tr>
<td><code>&lt;subscriber&gt;</code></td>
<td>Subscriber which will be notified</td>
</tr>
<tr>
<td><code>&lt;ttl&gt;</code></td>
<td>Time to live of the subscription in seconds</td>
</tr>
</tbody>
</table>

### Table 51.122. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--options &lt;options&gt;</code></td>
<td>Metadata of the subscription in json format</td>
</tr>
</tbody>
</table>

### Table 51.123. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 51.124. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 51.125. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 51.126. Table formatter options
### 51.29. MESSAGING SUBSCRIPTION DELETE

Delete a subscription

**Usage:**

```bash
genius messaging subscription delete [-h] <queue_name> <subscription_id>
```

**Table 51.127. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;queue_name&gt;</code></td>
<td>Name of the queue for the subscription</td>
</tr>
<tr>
<td><code>&lt;subscription_id&gt;</code></td>
<td>Id of the subscription</td>
</tr>
</tbody>
</table>

**Table 51.128. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 51.30. MESSAGING SUBSCRIPTION LIST

List available subscriptions

**Usage:**

```bash
genius messaging subscription list [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [-max-width <integer>]
  [--fit-width] [-print-empty]
  [-sort-column SORT_COLUMN]
```
Table 51.129. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue to subscribe to</td>
</tr>
</tbody>
</table>

Table 51.130. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker &lt;subscription_id&gt;</td>
<td>Subscription’s paging marker, the id of the last subscription of the previous page</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Page size limit, default value is 20</td>
</tr>
</tbody>
</table>

Table 51.131. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 51.132. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 51.133. JSON formatter options
Table 51.134. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

51.31. MESSAGING SUBSCRIPTION SHOW

Display subscription details

Usage:

```
```

Table 51.135. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;queue_name&gt;</td>
<td>Name of the queue to subscribe to</td>
</tr>
<tr>
<td>&lt;subscription_id&gt;</td>
<td>Id of the subscription</td>
</tr>
</tbody>
</table>

Table 51.136. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 51.137. Output formatter options
51.32. MESSAGING SUBSCRIPTION UPDATE

Update a subscription

Usage:

```
```
### Table 51.141. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;queue_name&gt;</code></td>
<td>Name of the queue to subscribe to</td>
</tr>
<tr>
<td><code>&lt;subscription_id&gt;</code></td>
<td>Id of the subscription</td>
</tr>
</tbody>
</table>

### Table 51.142. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--subscriber &lt;subscriber&gt;</code></td>
<td>Subscriber which will be notified</td>
</tr>
<tr>
<td><code>--ttl &lt;ttl&gt;</code></td>
<td>Time to live of the subscription in seconds</td>
</tr>
<tr>
<td><code>--options &lt;options&gt;</code></td>
<td>Metadata of the subscription in json format</td>
</tr>
</tbody>
</table>

### Table 51.143. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 51.144. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 51.145. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 51.146. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 52. METRIC

This chapter describes the commands under the `metric` command.

52.1. METRIC AGGREGATES

Get measurements of aggregated metrics.

Usage:

```bash
```

Table 52.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>operations</td>
<td>Operations to apply to time series</td>
</tr>
<tr>
<td>search</td>
<td>A query to filter resource. the syntax is a combination of attribute, operator and value. For example: id=90d58eea-70d7-4294-a49a-170dcdf44c3c would filter resource with a certain id. More complex queries can be built, e.g.: not (flavor_id!=&quot;1&quot; and memory&gt;=24). Use &quot;&quot; to force data to be interpreted as string. Supported operators are: not, and, or, ∧, ∨, &gt;=, &lt;=, !=, &gt;, &lt;, =, ==, eq, ne, lt, gt, le, ge, in, like, ≠, ≥, ≤, like, in.</td>
</tr>
</tbody>
</table>

Table 52.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-type RESOURCE_TYPE</td>
<td>Resource type to query</td>
</tr>
<tr>
<td>--start START</td>
<td>Beginning of the period</td>
</tr>
<tr>
<td>--stop STOP</td>
<td>End of the period</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--granularity GRANULARITY</td>
<td>Granularity to retrieve</td>
</tr>
<tr>
<td>--needed-overlap NEEDED_OVERLAP</td>
<td>Percentage of overlap across datapoints</td>
</tr>
<tr>
<td>--groupby GROUPBY</td>
<td>Attribute to use to group resources</td>
</tr>
<tr>
<td>--fill FILL</td>
<td>Value to use when backfilling timestamps with missing values in a subset of series. Value should be a float or null.</td>
</tr>
</tbody>
</table>

Table 52.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 52.4. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 52.5. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.6. Table formatter options
## 52.2. METRIC ARCHIVE-POLICY CREATE

Create an archive policy.

**Usage:**

```plaintext
```

### Table 52.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the archive policy</td>
</tr>
</tbody>
</table>

### Table 52.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-d &lt;DEFINITION&gt;, --definition &lt;DEFINITION&gt;</td>
<td>Two attributes (separated by ,) of an archive policy definition with its name and value separated with a :</td>
</tr>
<tr>
<td>-b BACK_WINDOW, --back-window BACK_WINDOW</td>
<td>Back window of the archive policy</td>
</tr>
<tr>
<td>-m AGGREGATION_METHODS, --aggregation-method AGGREGATION_METHODS</td>
<td>Aggregation method of the archive policy</td>
</tr>
</tbody>
</table>
Table 52.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.10. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.11. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.3. METRIC ARCHIVE-POLICY DELETE

Delete an archive policy.

Usage:

```
openstack metric archive-policy delete [-h] name
```

Table 52.13. Positional arguments
52.4. METRIC ARCHIVE-POLICY LIST

List archive policies.

Usage:

```
openstack metric archive-policy list [-h]
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
```

Table 52.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 52.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
### 52.17. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### 52.18. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 52.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 52.5. METRIC ARCHIVE-POLICY-RULE CREATE

Create an archive policy rule.

**Usage:**

```
```
Table 52.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Rule name</td>
</tr>
</tbody>
</table>

Table 52.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-a ARCHIVE_POLICY_NAME, --archive-policy-name ARCHIVE_POLICY_NAME</td>
<td>Archive policy name</td>
</tr>
<tr>
<td>-m METRIC_PATTERN, --metric-pattern METRIC_PATTERN</td>
<td>Wildcard of metric name to match</td>
</tr>
</tbody>
</table>

Table 52.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.25. Table formatter options

| Value | Summary |
52.6. METRIC ARCHIVE-POLICY-RULE DELETE

Delete an archive policy rule.

Usage:

```
openstack metric archive-policy-rule delete [-h] name
```

Table 52.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the archive policy rule</td>
</tr>
</tbody>
</table>

Table 52.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

52.7. METRIC ARCHIVE-POLICY-RULE LIST

List archive policy rules.

Usage:

```
openstack metric archive-policy-rule list [-h]
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [-quote {all,minimal,none,nonnumeric}]
    [-noindent]
    [-max-width <integer>]
    [-fit-width] [-print-empty]
    [-sort-column SORT_COLUMN]
    [-sort-ascending | --sort-descending]
```
## Table 52.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

## Table 52.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

## Table 52.30. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

## Table 52.31. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

## Table 52.32. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 52.8. METRIC ARCHIVE-POLICY-RULE SHOW

Show an archive policy rule.

**Usage:**

```
openstack metric archive-policy-rule show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
name
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 52.33. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the archive policy rule</td>
</tr>
</tbody>
</table>

**Table 52.34. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 52.35. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 52.36. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.37. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 52.38. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.9. METRIC ARCHIVE-POLICY SHOW

Show an archive policy.

Usage:

```bash
openstack metric archive-policy show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [-noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  name
```

Table 52.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the archive policy</td>
</tr>
</tbody>
</table>

Table 52.40. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 52.41. Output formatter options
52.10. METRIC ARCHIVE-POLICY UPDATE

Update an archive policy.

Usage:

```
openstack metric archive-policy update [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty] -d
    <DEFINITION>
    name
```
### Table 52.45. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the archive policy</td>
</tr>
</tbody>
</table>

### Table 52.46. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-d &lt;DEFINITION&gt;, --definition &lt;DEFINITION&gt;</td>
<td>Two attributes (separated by ,) of an archive policy definition with its name and value separated with a :</td>
</tr>
</tbody>
</table>

### Table 52.47. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 52.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 52.49. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 52.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX.Term_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
52.11. METRIC BENCHMARK MEASURES ADD

Do benchmark testing of adding measurements.

Usage:

```
openstack metric benchmark measures add [-h]
    [--resource-id RESOURCE_ID]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--workers WORKERS] --count
    COUNT [-batch BATCH]
    [--timestamp-start TIMESTAMP_START]
    [--timestamp-end TIMESTAMP_END]
    [--wait]
metric
```

Table 52.51. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metric</td>
</tr>
</tbody>
</table>

Table 52.52. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>--workers WORKERS, -w WORKERS</td>
<td>Number of workers to use</td>
</tr>
<tr>
<td>--count COUNT, -n COUNT</td>
<td>Number of total measures to send</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--batch BATCH, -b BATCH</td>
<td>Number of measures to send in each batch</td>
</tr>
<tr>
<td>--timestamp-start TIMESTAMP_START, -s TIMESTAMP_START</td>
<td>First timestamp to use</td>
</tr>
<tr>
<td>--timestamp-end TIMESTAMP_END, -e TIMESTAMP_END</td>
<td>Last timestamp to use</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for all measures to be processed</td>
</tr>
</tbody>
</table>

**Table 52.53. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 52.54. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.55. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 52.56. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
52.12. METRIC BENCHMARK MEASURES SHOW

Do benchmark testing of measurements show.

Usage:

openstack metric benchmark measures show [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--utc]
[--resource-id RESOURCE_ID]
[--aggregation AGGREGATION]
[--start START] [--stop STOP]
[--granularity GRANULARITY]
[--refresh]
[--resample RESAMPLE]
[--workers WORKERS] --count
COUNT
metric

Table 52.57. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metric</td>
</tr>
</tbody>
</table>

Table 52.58. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--utc</td>
<td>Return timestamps as utc</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>--aggregation AGGREGATION</td>
<td>Aggregation to retrieve</td>
</tr>
<tr>
<td>--start START</td>
<td>Beginning of the period</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>--stop STOP</td>
<td>End of the period</td>
</tr>
<tr>
<td>--granularity GRANULARITY</td>
<td>Granularity to retrieve</td>
</tr>
<tr>
<td>--refresh</td>
<td>Force aggregation of all known measures</td>
</tr>
<tr>
<td>--resample RESAMPLE</td>
<td>Granularity to resample time-series to (in seconds)</td>
</tr>
<tr>
<td>--workers WORKERS, -w WORKERS</td>
<td>Number of workers to use</td>
</tr>
<tr>
<td>--count COUNT, -n COUNT</td>
<td>Number of total measures to send</td>
</tr>
</tbody>
</table>

Table 52.59. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 52.60. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.61. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.62. Table formatter options
52.13. METRIC BENCHMARK METRIC CREATE

Do benchmark testing of metric creation.

Usage:


Table 52.63. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>--archive-policy-name ARCHIVE_POLICY_NAME, -a ARCHIVE_POLICY_NAME</td>
<td>Name of the archive policy</td>
</tr>
<tr>
<td>--workers WORKERS, -w WORKERS</td>
<td>Number of workers to use</td>
</tr>
<tr>
<td>--count COUNT, -n COUNT</td>
<td>Number of metrics to create</td>
</tr>
<tr>
<td>--keep, -k</td>
<td>Keep created metrics</td>
</tr>
</tbody>
</table>

Table 52.64. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
52.65. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

52.66. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

52.67. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.14. METRIC BENCHMARK METRIC SHOW

Do benchmark testing of metric show.

Usage:

```bash
```
[--workers WORKERS] --count COUNT metric [metric ...]

Table 52.68. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metrics</td>
</tr>
</tbody>
</table>

Table 52.69. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>--workers WORKERS, -w WORKERS</td>
<td>Number of workers to use</td>
</tr>
<tr>
<td>--count COUNT, -n COUNT</td>
<td>Number of metrics to get</td>
</tr>
</tbody>
</table>

Table 52.70. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.71. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.72. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.73. Table formatter options
### 52.15. METRIC CAPABILITIES LIST

List capabilities.

**Usage:**

```
openstack metric capabilities list [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 52.74. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 52.75. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.76. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
### Table 52.78. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIIF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 52.16. METRIC CREATE

Create a metric.

**Usage:**

```plaintext
```

#### Table 52.79. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>METRIC_NAME</td>
<td>Name of the metric</td>
</tr>
</tbody>
</table>

#### Table 52.80. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>
Value | Summary
--- | ---
--archive-policy-name ARCHIVE_POLICY_NAME, -a ARCHIVE_POLICY_NAME | Name of the archive policy
--unit UNIT, -u UNIT | Unit of the metric

### Table 52.81. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 52.82. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 52.83. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 52.84. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 52.17. METRIC DELETE

Delete a metric.
Usage:

openstack metric delete [-h] [--resource-id RESOURCE_ID] metric [metric ...]

Table 52.85. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Ids or names of the metric</td>
</tr>
</tbody>
</table>

Table 52.86. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>

52.18. METRIC LIST

List metrics.

Usage:


Table 52.87. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of metrics to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing, return the next results after this value</td>
</tr>
<tr>
<td>--sort &lt;SORT&gt;</td>
<td>Sort of metric attribute (example: user_id:desc-nullslast)</td>
</tr>
</tbody>
</table>
Table 52.88. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 52.89. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 52.90. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.91. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.19. METRIC MEASURES ADD

Add measurements to a metric.
Usage:

openstack metric measures add [-h] [--resource-id RESOURCE_ID] -m MEASURE

Table 52.92. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metric</td>
</tr>
</tbody>
</table>

Table 52.93. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>-m MEASURE, --measure MEASURE</td>
<td>Timestamp and value of a measure separated with a @</td>
</tr>
</tbody>
</table>

52.20. METRIC MEASURES AGGREGATION

Get measurements of aggregated metrics.

Usage:

openstack metric measures aggregation [-h]

[f {csv,json,table,value,yaml}]
[-c COLUMN]
[-quote {all,minimal,none,nonnumeric}]
[-noindent]
[-max-width <integer>]
[-fit-width] [-print-empty]
[-sort-column SORT_COLUMN]
[-sort-ascending | --sort-descending]
[-utc] -m METRIC [METRIC ...]
[--aggregation AGGREGATION]
[--reaggregation REAGGREGATION]
[--start START] [--stop STOP]
[--granularity GRANULARITY]
[--needed-overlap NEEDED_OVERLAP]
[--query QUERY]
[--resource-type RESOURCE_TYPE]
[--groupby GROUPBY] [--refresh]
[--resample RESAMPLE]
[--fill FILL]

Table 52.94. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--utc</td>
<td>Return timestamps as utc</td>
</tr>
<tr>
<td>-m METRIC [METRIC ...], --metric METRIC [METRIC ...]</td>
<td>Metrics ids or metric name</td>
</tr>
<tr>
<td>--aggregation AGGREGATION</td>
<td>Granularity aggregation function to retrieve</td>
</tr>
<tr>
<td>--reaggregation REAGGREGATION</td>
<td>Groupby aggregation function to retrieve</td>
</tr>
<tr>
<td>--start START</td>
<td>Beginning of the period</td>
</tr>
<tr>
<td>--stop STOP</td>
<td>End of the period</td>
</tr>
<tr>
<td>--granularity GRANULARITY</td>
<td>Granularity to retrieve</td>
</tr>
<tr>
<td>--needed-overlap NEEDED_OVERLAP</td>
<td>Percent of datapoints in each metrics required</td>
</tr>
<tr>
<td>--query QUERY</td>
<td>A query to filter resource. the syntax is a combination of attribute, operator and value. For example: id=90d58eea-70d7-4294-a49a-170dcdf44c3c would filter resource with a certain id. More complex queries can be built, e.g.: not (flavor_id!=&quot;1&quot; and memory&gt;=24). Use &quot;&quot; to force data to be interpreted as string. Supported operators are: not, and, \land, \lor, \geq, \leq, \neq, &gt;, &lt;, =, eq, ne, it, gt, ge, le, in, like, ≠, ≥, ≤, like, in.</td>
</tr>
<tr>
<td>--resource-type RESOURCE_TYPE</td>
<td>Resource type to query</td>
</tr>
<tr>
<td>--groupby GROUPBY</td>
<td>Attribute to use to group resources</td>
</tr>
<tr>
<td>--refresh</td>
<td>Force aggregation of all known measures</td>
</tr>
<tr>
<td>--resample RESAMPLE</td>
<td>Granularity to resample time-series to (in seconds)</td>
</tr>
<tr>
<td>--fill FILL</td>
<td>Value to use when backfilling timestamps with missing values in a subset of series. Value should be a float or null.</td>
</tr>
</tbody>
</table>

Table 52.95. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 52.96. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 52.97. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.98. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**52.21. METRIC MEASURES BATCH-METRICS**

**Usage:**

```bash
openstack metric measures batch-metrics [-h] file
```
Table 52.99. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>File containing measurements to batch or - for stdin (see Gnocchi REST API docs for the format)</td>
</tr>
</tbody>
</table>

Table 52.100. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

52.22. METRIC MEASURES BATCH-RESOURCES-METRICS

Usage:

```
openstack metric measures batch-resources-metrics [-h] [--create-metrics] file
```

Table 52.101. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>File containing measurements to batch or - for stdin (see Gnocchi REST API docs for the format)</td>
</tr>
</tbody>
</table>

Table 52.102. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--create-metrics</td>
<td>Create unknown metrics</td>
</tr>
</tbody>
</table>

52.23. METRIC MEASURES SHOW

Get measurements of a metric.

Usage:

```
openstack metric measures show [-h] [-f {csv,json,table,value,yaml}]
[-c COLUMN] [-quote {all,minimal,none,nonnumeric}]
[--noindent] [-max-width <integer>] [-fit-width] [-print-empty]
```
Table 52.103. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metric</td>
</tr>
</tbody>
</table>

Table 52.104. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--utc</td>
<td>Return timestamps as utc</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
<tr>
<td>--aggregation AGGREGATION</td>
<td>Aggregation to retrieve</td>
</tr>
<tr>
<td>--start START</td>
<td>Beginning of the period</td>
</tr>
<tr>
<td>--stop STOP</td>
<td>End of the period</td>
</tr>
<tr>
<td>--granularity GRANULARITY</td>
<td>Granularity to retrieve</td>
</tr>
<tr>
<td>--refresh</td>
<td>Force aggregation of all known measures</td>
</tr>
<tr>
<td>--resample RESAMPLE</td>
<td>Granularity to resample time-series to (in seconds)</td>
</tr>
</tbody>
</table>

Table 52.105. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
---sort-column SORT_COLUMN

Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

---sort-ascending

Sort the column(s) in ascending order

---sort-descending

Sort the column(s) in descending order

### Table 52.106. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 52.107. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 52.108. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 52.24. METRIC RESOURCE BATCH DELETE

Delete a batch of resources based on attribute values.

#### Usage:

```
openstack metric resource batch delete [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-noindent]
```
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--type RESOURCE_TYPE]

query

Table 52.109. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A query to filter resource. The syntax is a combination of attribute, operator and value. For example: id=90d58eea-70d7-4294-a49a-170dcd44c3c would filter resource with a certain id. More complex queries can be built, e.g.: not (flavor_id=&quot;1&quot; and memory&gt;=24). Use &quot;&quot; to force data to be interpreted as string. Supported operators are: not, and, or, &gt;=, &lt;=, !=, &gt;, &lt;, =, ==, eq, ne, lt, gt, in, like, ≠, ≥, ≤, like, in.</td>
</tr>
</tbody>
</table>

Table 52.110. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
</tbody>
</table>

Table 52.111. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.112. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.113. Shell formatter options
52.25. METRIC RESOURCE CREATE

Create a resource.

Usage:

```
```

Table 52.115. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>

Table 52.116. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-a ATTRIBUTE, --attribute ATTRIBUTE</td>
<td>Name and value of an attribute separated with a :</td>
</tr>
<tr>
<td>-m ADD_METRIC, --add-metric ADD_METRIC</td>
<td>Name:id of a metric to add</td>
</tr>
<tr>
<td>-n CREATE_METRIC, --create-metric CREATE_METRIC</td>
<td>Name:archive_policy_name of a metric to create</td>
</tr>
</tbody>
</table>

**Table 52.117. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 52.118. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.119. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 52.120. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**52.26. METRIC RESOURCE DELETE**
Delete a resource.

Usage:

    openstack metric resource delete [-h] resource_id

Table 52.121. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>

Table 52.122. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

52.27. METRIC RESOURCE HISTORY

Show the history of a resource.

Usage:


Table 52.123. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Id of a resource</td>
</tr>
</tbody>
</table>

Table 52.124. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--details</td>
<td>Show all attributes of generic resources</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of resources to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing. return the next results after this value</td>
</tr>
<tr>
<td>--sort &lt;SORT&gt;</td>
<td>Sort of resource attribute (example: user_id:desc-nullslast)</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
</tbody>
</table>

Table 52.125. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 52.126. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 52.127. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.128. Table formatter options
52.28. METRIC RESOURCE LIST

List resources.

Usage:

```
```

Table 52.129. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-t RESOURCE_TYPE, --type RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
<tr>
<td>--details</td>
<td>Show all attributes of generic resources</td>
</tr>
<tr>
<td>--history</td>
<td>Show history of the resources</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of resources to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing. return the next results after this value</td>
</tr>
<tr>
<td>--sort &lt;SORT&gt;</td>
<td>Sort of resource attribute (example: user_id:desc-nullslast)</td>
</tr>
</tbody>
</table>

---

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.

Print empty table if there is no data to show.
**Table 52.130. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 52.131. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 52.132. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.133. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**52.29. METRIC RESOURCE SEARCH**

Search resources with specified query rules.
Usage:


Table 52.134. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>query</td>
<td>A query to filter resource. The syntax is a combination of attribute, operator and value. For example: id=90d8eaa-70d7-4294-a49a-170dc94c3c would filter resource with a certain id. More complex queries can be built, e.g.: not (flavor_id!=&quot;1&quot; and memory&gt;24). Use &quot;&quot; to force data to be interpreted as string. Supported operators are: not, and, or,∧,≥,⇐,≠, &gt;, &lt;, =, ==, eq, ne, le, in, like, i, ≥, ≤, like, in.</td>
</tr>
</tbody>
</table>

Table 52.135. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--details</td>
<td>Show all attributes of generic resources</td>
</tr>
<tr>
<td>--history</td>
<td>Show history of the resources</td>
</tr>
<tr>
<td>--limit &lt;LIMIT&gt;</td>
<td>Number of resources to return (default is server default)</td>
</tr>
<tr>
<td>--marker &lt;MARKER&gt;</td>
<td>Last item of the previous listing. return the next results after this value</td>
</tr>
<tr>
<td>--sort &lt;SORT&gt;</td>
<td>Sort of resource attribute (example: user_id:desc-nullslast)</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
</tbody>
</table>

Table 52.136. Output formatter options
### Table 52.137. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 52.138. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 52.139. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 52.30. METRIC RESOURCE SHOW

Show a resource.

**Usage:**

Table 52.140. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Id of a resource</td>
</tr>
</tbody>
</table>

Table 52.141. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
</tbody>
</table>

Table 52.142. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.143. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.144. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.145. Table formatter options
### 52.31. METRIC RESOURCE-TYPE CREATE

Create a resource type.

**Usage:**

```bash
code
```

<table>
<thead>
<tr>
<th>Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>-a ATTRIBUTE, --attribute ATTRIBUTE</td>
</tr>
</tbody>
</table>

| Output formatter options |
Value | Summary  
---|---  
-f \{json,shell,table,value,yaml\}, --format \{json,shell,table,value,yaml\} | The output format, defaults to table  
-c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns  

Table 52.149. JSON formatter options

Value | Summary  
---|---  
--noindent | Whether to disable indenting the json  

Table 52.150. Shell formatter options

Value | Summary  
---|---  
--prefix PREFIX | Add a prefix to all variable names  

Table 52.151. Table formatter options

Value | Summary  
---|---  
--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.  
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable  
--print-empty | Print empty table if there is no data to show.  

52.32. METRIC RESOURCE-TYPE DELETE

Delete a resource type.

Usage:

```
openstack metric resource-type delete [-h] name
```

Table 52.152. Positional arguments
Table 52.153. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the resource type</td>
</tr>
</tbody>
</table>

Table 52.154. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 52.33. METRIC RESOURCE-TYPE LIST

List resource types.

**Usage:**

```
openstack metric resource-type list [-h]
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
```

Table 52.155. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
Table 52.156. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 52.157. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 52.158. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.34. METRIC RESOURCE-TYPE SHOW

Show a resource type.

Usage:

```
```

Table 52.159. Positional arguments
Table 52.160. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 52.161. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.162. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.163. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.164. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
52.35. METRIC RESOURCE-TYPE UPDATE

Create a resource type.

Usage:


name

Table 52.165. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Name of the resource type</td>
</tr>
</tbody>
</table>

Table 52.166. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-a ATTRIBUTE, --attribute ATTRIBUTE</td>
<td>Attribute definition, attribute_name:attribute_type:attribute_is_required:attribute_type_option_name=attribute_type_option_value:… For example: display_name:string:true:max_length=255</td>
</tr>
<tr>
<td>-r REMOVE_ATTRIBUTE, --remove-attribute REMOVE_ATTRIBUTE</td>
<td>Attribute name</td>
</tr>
</tbody>
</table>

Table 52.167. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.168. JSON formatter options
Table 52.169. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.170. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.36. METRIC RESOURCE UPDATE

Update a resource.

**Usage:**

```bash
```

Table 52.171. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>

Table 52.172. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type RESOURCE_TYPE, -t RESOURCE_TYPE</td>
<td>Type of resource</td>
</tr>
<tr>
<td>-a ATTRIBUTE, --attribute ATTRIBUTE</td>
<td>Name and value of an attribute separated with a :</td>
</tr>
<tr>
<td>-m ADD_METRIC, --add-metric ADD_METRIC</td>
<td>Name:id of a metric to add</td>
</tr>
<tr>
<td>-n CREATE_METRIC, --create-metric CREATE_METRIC</td>
<td>Name:archive_policy_name of a metric to create</td>
</tr>
<tr>
<td>-d DELETE_METRIC, --delete-metric DELETE_METRIC</td>
<td>Name of a metric to delete</td>
</tr>
</tbody>
</table>

**Table 52.173. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 52.174. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 52.175. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 52.176. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
52.37. METRIC SERVER VERSION

Show the version of Gnocchi server.

Usage:

```
openstack metric server version [-h] [-f {json,shell,table,value,yaml}] 
[-c COLUMN] [--noindent] 
[-prefix PREFIX] 
[--max-width <integer>] [--fit-width] 
[--print-empty]
```

Table 52.177. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 52.178. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 52.179. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 52.181. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.38. METRIC SHOW

Show a metric.

Usage:

```
```

Table 52.182. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>metric</td>
<td>Id or name of the metric</td>
</tr>
</tbody>
</table>

Table 52.183. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID, -r RESOURCE_ID</td>
<td>Id of the resource</td>
</tr>
</tbody>
</table>
-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}  
   The output format, defaults to table

-c COLUMN, --column COLUMN  
   Specify the column(s) to include, can be repeated to show multiple columns

Table 52.185. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 52.186. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 52.187. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

52.39. METRIC STATUS

Show the status of measurements processing.

Usage:

```
```

Table 52.188. Command arguments
### Table 52.189. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 52.190. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 52.191. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 52.192. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 53. MODULE

This chapter describes the commands under the module command.

53.1. MODULE LIST

List module versions

Usage:


Table 53.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all</td>
<td>Show all modules that have version information</td>
</tr>
</tbody>
</table>

Table 53.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 53.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 53.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 53.5. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 54. NETWORK

This chapter describes the commands under the network command.

54.1. NETWORK AGENT ADD NETWORK

Add network to an agent

Usage:

openstack network agent add network [-h] [--dhcp] <agent-id> <network>

Table 54.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;agent-id&gt;</td>
<td>Agent to which a network is added (id only)</td>
</tr>
<tr>
<td>&lt;network&gt;</td>
<td>Network to be added to an agent (name or id)</td>
</tr>
</tbody>
</table>

Table 54.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--dhcp</td>
<td>Add network to a dhcp agent</td>
</tr>
</tbody>
</table>

54.2. NETWORK AGENT ADD ROUTER

Add router to an agent

Usage:

openstack network agent add router [-h] [-l3] <agent-id> <router>

Table 54.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;agent-id&gt;</td>
<td>Agent to which a router is added (id only)</td>
</tr>
<tr>
<td>&lt;router&gt;</td>
<td>Router to be added to an agent (name or id)</td>
</tr>
</tbody>
</table>

Table 54.4. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--l3</td>
<td>Add router to an l3 agent</td>
</tr>
</tbody>
</table>

### 54.3. NETWORK AGENT DELETE

Delete network agent(s)

**Usage:**

```
openstack network agent delete [-h] <network-agent> [...]
```

**Table 54.5. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-agent&gt;</td>
<td>Network agent(s) to delete (id only)</td>
</tr>
</tbody>
</table>

**Table 54.6. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 54.4. NETWORK AGENT LIST

List network agents

**Usage:**

```
```

**Table 54.7. Command arguments**
Table 54.8. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.9. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.10. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
54.5. NETWORK AGENT REMOVE NETWORK

Remove network from an agent.

Usage:

```bash
openstack network agent remove network [-h] [--dhcp] <agent-id> <network>
```

Table 54.12. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;agent-id&gt;</td>
<td>Agent to which a network is removed (id only)</td>
</tr>
<tr>
<td>&lt;network&gt;</td>
<td>Network to be removed from an agent (name or id)</td>
</tr>
</tbody>
</table>

Table 54.13. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--dhcp</td>
<td>Remove network from dhcp agent</td>
</tr>
</tbody>
</table>

54.6. NETWORK AGENT REMOVE ROUTER

Remove router from an agent

Usage:

```bash
openstack network agent remove router [-h] [--l3] <agent-id> <router>
```

Table 54.14. Positional arguments
### 54.7. NETWORK AGENT SET

Set network agent properties

**Usage:**

```
```

### Table 54.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network-agent&gt;</code></td>
<td>Network agent to modify (id only)</td>
</tr>
</tbody>
</table>

### Table 54.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set network agent description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable network agent</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable network agent</td>
</tr>
</tbody>
</table>

### 54.8. NETWORK AGENT SHOW

Display network agent details

**Usage:**

Table 54.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-agent&gt;</td>
<td>Network agent to display (id only)</td>
</tr>
</tbody>
</table>

Table 54.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.21. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.22. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
54.9. NETWORK AUTO ALLOCATED TOPOLOGY CREATE

Create the auto allocated topology for project

Usage:

    openstack network auto allocated topology create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN]
    [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--project <project>]
    [--project-domain <project-domain>]
    [--check-resources]
    [--or-show]

Table 54.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Return the auto allocated topology for a given project. Default is current project</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--check-resources</td>
<td>Validate the requirements for auto allocated topology. Does not return a topology.</td>
</tr>
<tr>
<td>--or-show</td>
<td>If topology exists returns the topology's information (Default)</td>
</tr>
</tbody>
</table>
Table 54.25. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.27. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.28. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.10. NETWORK AUTO ALLOCATED TOPOLOGY DELETE

Delete auto allocated topology for project

Usage:

```
openstack network auto allocated topology delete [-h]
    [--project <project>]
    [--project-domain <project-domain>]
```

Table 54.29. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Delete auto allocated topology for a given project. Default is the current project</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

### 54.11. NETWORK CREATE

Create new network

**Usage:**

```bash
```

<table>
<thead>
<tr>
<th>Table 54.30. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 54.31. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--share</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>----------------------------</td>
</tr>
<tr>
<td>--no-share</td>
</tr>
<tr>
<td>--enable</td>
</tr>
<tr>
<td>--disable</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
</tr>
<tr>
<td>--mtu &lt;mtu&gt;</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
</tr>
<tr>
<td>--availability-zone-hint &lt;availability-zone&gt;</td>
</tr>
<tr>
<td>--enable-port-security</td>
</tr>
<tr>
<td>--disable-port-security</td>
</tr>
<tr>
<td>--external</td>
</tr>
<tr>
<td>--internal</td>
</tr>
<tr>
<td>--default</td>
</tr>
<tr>
<td>--no-default</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
</tr>
<tr>
<td>--transparent-vlan</td>
</tr>
<tr>
<td>--no-transparent-vlan</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>--provider-network-type &lt;provider-network-type&gt;</td>
</tr>
<tr>
<td>--provider-physical-network &lt;provider-physical-network&gt;</td>
</tr>
<tr>
<td>--provider-segment &lt;provider-segment&gt;</td>
</tr>
<tr>
<td>--dns-domain &lt;dns-domain&gt;</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
</tr>
<tr>
<td>--no-tag</td>
</tr>
</tbody>
</table>

Table 54.32. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.33. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.34. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.35. Table formatter options
### 54.12. NETWORK DELETE

Delete network(s)

**Usage:**

```
openstack network delete [-h] <network> [<network> ...]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 54.36. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network&gt;</code></td>
<td>Network(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 54.37. Command arguments**

### 54.13. NETWORK FLAVOR ADD PROFILE

Add a service profile to a network flavor

**Usage:**

```
openstack network flavor add profile [-h] <flavor> <service-profile>
```

**Table 54.38. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flavor&gt;</code></td>
<td>Network flavor (name or id)</td>
</tr>
</tbody>
</table>
54.14. NETWORK FLAVOR CREATE

Create new network flavor

**Usage:**

```bash
```

**Table 54.40. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name for the flavor</td>
</tr>
</tbody>
</table>

**Table 54.41. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type to which the flavor applies to: e.g. vpn (See openstack network service provider list for loaded examples.)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description for the flavor</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the flavor (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the flavor</td>
</tr>
</tbody>
</table>

**Table 54.42. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 54.43. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.44. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 54.45. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
54.15. NETWORK FLAVOR DELETE

Delete network flavors

Usage:

```
openstack network flavor delete [-h] <flavor> [<flavor> ...]
```

Table 54.46. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 54.47. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.16. NETWORK FLAVOR LIST

List network flavors

Usage:

```
```

Table 54.48. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.49. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 54.50. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.51. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.52. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.17. NETWORK FLAVOR PROFILE CREATE

Create new network flavor profile

Usage:

    openstack network flavor profile create [-h] [-f {json,shell,table,value,yaml}]
Table 54.53. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the flavor profile</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the flavor profile</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the flavor profile</td>
</tr>
<tr>
<td>--driver DRIVER</td>
<td>Python module path to driver. this becomes required if --metainfo is missing and vice versa</td>
</tr>
<tr>
<td>--metainfo METAINFO</td>
<td>Metainfo for the flavor profile. this becomes required if --driver is missing and vice versa</td>
</tr>
</tbody>
</table>

Table 54.54. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format json,shell,table,value,yaml</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.55. JSON formatter options
### Table 54.56. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 54.57. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.18. NETWORK FLAVOR PROFILE DELETE

Delete network flavor profile

**Usage:**

```
openstack network flavor profile delete [-h] <flavor-profile> [<flavor-profile> ...]
```

### Table 54.58. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor-profile&gt;</td>
<td>Flavor profile(s) to delete (id only)</td>
</tr>
</tbody>
</table>

### Table 54.59. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
54.19. NETWORK FLAVOR PROFILE LIST

List network flavor profile(s)

Usage:

```
openstack network flavor profile list [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
```

Table 54.60. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.61. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.62. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 54.64. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.20. NETWORK FLAVOR PROFILE SET

Set network flavor profile properties

Usage:

```
openstack network flavor profile set [-h]
  [--project-domain <project-domain>]
  [--description <description>]
  [--enable | --disable]
  [--driver DRIVER]
  [--metainfo METAINFO]
  <flavor-profile>
```

Table 54.65. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor-profile&gt;</td>
<td>Flavor profile to update (id only)</td>
</tr>
</tbody>
</table>

Table 54.66. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>
### 54.21. NETWORK FLAVOR PROFILE SHOW

Display network flavor profile details

**Usage:**

```bash
openstack network flavor profile show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    <flavor-profile>
```

**Table 54.67. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor-profile&gt;</td>
<td>Flavor profile to display (id only)</td>
</tr>
</tbody>
</table>

**Table 54.68. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 54.69. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### 54.22. NETWORK FLAVOR REMOVE PROFILE

Remove service profile from network flavor

**Usage:**

```
openstack network flavor remove profile [-h] <flavor> <service-profile>
```

**Table 54.73. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Network flavor (name or id)</td>
</tr>
</tbody>
</table>
Table 54.74. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-profile&gt;</td>
<td>Service profile (id only)</td>
</tr>
</tbody>
</table>

54.23. NETWORK FLAVOR SET

Set network flavor properties

Usage:

```
openstack network flavor set [-h] [--description DESCRIPTION] [--disable | --enable] [--name <name>] <flavor>
```

Table 54.75. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to update (name or id)</td>
</tr>
</tbody>
</table>

Table 54.76. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Set network flavor description</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable network flavor</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable network flavor</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set flavor name</td>
</tr>
</tbody>
</table>

54.24. NETWORK FLAVOR SHOW

Display network flavor details

Usage:

```
openstack network flavor show [-h] [-f {json,shell,table,value,yaml}]
```
[-c COLUMN] [--noindent]
[-prefix PREFIX] [--max-width <integer>]
[-fit-width] [--print-empty]

Table 54.77. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flavor&gt;</td>
<td>Flavor to display (name or id)</td>
</tr>
</tbody>
</table>

Table 54.78. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.79. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.80. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.81. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.82. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 54.25. NETWORK LIST

List networks

### Usage:

```
openstack network list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] 
[-quote {all, minimal, none, nonnumeric}] 
[-noindent] [-max-width <integer>]
[-fit-width] [-print-empty] 
[-sort-column SORT_COLUMN] 
[-sort-ascending | --sort-descending] 
[-external | --internal] [-long] 
[-name <name>] [-enable | --disable] 
[-project <project>] 
[-project-domain <project-domain>] 
[-share | --no-share] [-status <status>]  
[-provider-network-type <provider-network-type>]  
[-provider-physical-network <provider-physical-network>]  
[-provider-segment <provider-segment>] 
[-agent <agent-id>] [-tags <tag>[,<tag>,...]] 
[-any-tags <tag>[,<tag>,...]]  
[-not-tags <tag>[,<tag>,...]]  
[-not-any-tags <tag>[,<tag>,...]]
```

### Table 54.83. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--external</td>
<td>List external networks</td>
</tr>
<tr>
<td>--internal</td>
<td>List internal networks</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List networks according to their name</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled networks</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled networks</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List networks according to their project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--share</td>
<td>List networks shared between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>List networks not shared between projects</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>List networks according to their status (active, BUILD, DOWN, ERROR)</td>
</tr>
<tr>
<td>--provider-network-type &lt;provider-network-type&gt;</td>
<td>List networks according to their physical mechanisms. The supported options are: flat, geneve, gre, local, vlan, vxlan.</td>
</tr>
<tr>
<td>--provider-physical-network &lt;provider-physical-network&gt;</td>
<td>List networks according to name of the physical network</td>
</tr>
<tr>
<td>--provider-segment &lt;provider-segment&gt;</td>
<td>List networks according to vlan id for vlan networks or Tunnel ID for GENEVE/GRE/VXLAN networks</td>
</tr>
<tr>
<td>--agent &lt;agent-id&gt;</td>
<td>List networks hosted by agent (id only)</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List networks which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List networks which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude networks which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude networks which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

**Table 54.84. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Value | Summary
--- | ---
--c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns
--sort-column SORT_COLUMN | Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
--sort-ascending | Sort the column(s) in ascending order
--sort-descending | Sort the column(s) in descending order

Table 54.85. CSV formatter options

Value | Summary
--- | ---
--quote {all,minimal,none,nonnumeric} | When to include quotes, defaults to nonnumeric

Table 54.86. JSON formatter options

Value | Summary
--- | ---
--noindent | Whether to disable indenting the json

Table 54.87. Table formatter options

Value | Summary
--- | ---
--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty | Print empty table if there is no data to show.

54.26. NETWORK LOG CREATE

Create a new network log

Usage:

openstack network log create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX]
Table 54.88. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name for the network log</td>
</tr>
</tbody>
</table>

Table 54.89. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the network log</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable this log (default is disabled)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable this log</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--event {ALL,ACCEPT,DROP}</td>
<td>An event to store with log</td>
</tr>
<tr>
<td>--resource-type &lt;resource-type&gt;</td>
<td>Network log type(s). you can see supported type(s) with following command: $ openstack network loggable resources list</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Name or id of resource (security group or firewall group) that used for logging. You can control for logging target combination with --target option.</td>
</tr>
<tr>
<td>--target &lt;target&gt;</td>
<td>Port (name or id) for logging. you can control for logging target combination with --resource option.</td>
</tr>
</tbody>
</table>

Table 54.90. Output formatter options
**Table 54.91. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.92. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 54.93. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**54.27. NETWORK LOG DELETE**

Delete network log(s)

**Usage:**

```bash
openstack network log delete [-h] <network-log> [<network-log> ...]
```

**Table 54.94. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-log&gt;</td>
<td>Network log(s) to delete (name or id)</td>
</tr>
</tbody>
</table>
### NETWORK LOG LIST

List network logs

#### Usage:

```bash
```

#### Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

#### Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
Table 54.98. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.99. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.100. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.29. NETWORK LOG SET

Set network log properties

Usage:

```
```

Table 54.101. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-log&gt;</td>
<td>Network log to set (name or id)</td>
</tr>
</tbody>
</table>

Table 54.102. Command arguments
### Table 54.103. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-log&gt;</td>
<td>Network log to show (name or id)</td>
</tr>
</tbody>
</table>

### Table 54.104. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 54.105. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

## 54.30. NETWORK LOG SHOW

Display network log details

**Usage:**

```
```

### Table 54.103. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-log&gt;</td>
<td>Network log to show (name or id)</td>
</tr>
</tbody>
</table>

### Table 54.104. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 54.105. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 54.107. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.108. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.31. NETWORK LOGGABLE RESOURCES LIST

List supported loggable resources

Usage:

```
```

Table 54.109. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>
### Table 54.110. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 54.111. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 54.112. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 54.113. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.32. NETWORK METER CREATE

Create network meter
Usage:

openstack network meter create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-noindent]
[-prefix PREFIX] [-max-width <integer>] [-fit-width]
[-print-empty] [-description <description>]
[--project <project>]
[-project-domain <project-domain>]
[-share | --no-share]
<name>

Table 54.114. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of meter</td>
</tr>
</tbody>
</table>

Table 54.115. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Create description for meter</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner's project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--share</td>
<td>Share meter between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>Do not share meter between projects</td>
</tr>
</tbody>
</table>

Table 54.116. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.117. JSON formatter options
### Table 54.118. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 54.119. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.33. NETWORK METER DELETE

Delete network meter

**Usage:**

```
openstack network meter delete [-h] <meter> [<meter> ...]
```

**Table 54.120. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;meter&gt;</td>
<td>Meter to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 54.121. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 54.34. NETWORK METER LIST
List network meters

Usage:

```
```

Table 54.122. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.123. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.124. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.125. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### 54.35. NETWORK METER RULE CREATE

Create a new meter rule

**Usage:**

```
openstack network meter rule create [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--project <project>]
  [--project-domain <project-domain>]
  [--exclude | --include]
  [--ingress | --egress]
  [--remote-ip-prefix <remote-ip-prefix>]
  [--source-ip-prefix <remote-ip-prefix>]
  [--destination-ip-prefix <remote-ip-prefix>]
  <meter>
```

**Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;meter&gt;</td>
<td>Label to associate with this metering rule (name or ID)</td>
</tr>
</tbody>
</table>

**Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner's project (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--exclude</td>
<td>Exclude remote ip prefix from traffic count</td>
</tr>
<tr>
<td>--include</td>
<td>Include remote ip prefix from traffic count (default)</td>
</tr>
<tr>
<td>--ingress</td>
<td>Apply rule to incoming network traffic (default)</td>
</tr>
<tr>
<td>--egress</td>
<td>Apply rule to outgoing network traffic</td>
</tr>
<tr>
<td>--remote-ip-prefix &lt;remote-ip-prefix&gt;</td>
<td>The remote ip prefix to associate with this rule</td>
</tr>
<tr>
<td>--source-ip-prefix &lt;remote-ip-prefix&gt;</td>
<td>The source ip prefix to associate with this rule</td>
</tr>
<tr>
<td>--destination-ip-prefix &lt;remote-ip-prefix&gt;</td>
<td>The destination ip prefix to associate with this rule</td>
</tr>
</tbody>
</table>

**Table 54.129. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 54.130. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.131. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
54.36. NETWORK METER RULE DELETE

Delete meter rule(s)

Usage:

```
openstack network meter rule delete [-h] <meter-rule-id> [<meter-rule-id> ...]
```

Table 54.133. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;meter-rule-id&gt;</td>
<td>Meter rule to delete (id only)</td>
</tr>
</tbody>
</table>

Table 54.134. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.37. NETWORK METER RULE LIST

List meter rules

Usage:

```
```

Table 54.135. Command arguments
Table 54.136. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {csv, json, table, value, yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.137. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.138. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.139. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
54.38. NETWORK METER RULE SHOW

Display meter rules details

Usage:

```bash
```

Table 54.140. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;meter-rule-id&gt;</code></td>
<td>Meter rule (id only)</td>
</tr>
</tbody>
</table>

Table 54.141. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.142. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.143. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 54.145. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.39. NETWORK METER SHOW

Show network meter

Usage:

```
```

Table 54.146. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;meter&gt;</td>
<td>Meter to display (name or id)</td>
</tr>
</tbody>
</table>

Table 54.147. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.148. Output formatter options
54.40. NETWORK ONBOARD SUBNETS

Onboard network subnets into a subnet pool

Usage:

```
openstack network onboard subnets [-h] <network> <subnetpool>
```

Table 54.152. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Onboard all subnets associated with this network</td>
</tr>
</tbody>
</table>
54.41. NETWORK QOS POLICY CREATE

Create a QoS policy

Usage:

```sh
openstack network qos policy create [-h]
     [-f {json,shell,table,value,yaml}]
     [-c COLUMN] [--noindent]
     [--prefix PREFIX]
     [--max-width <integer>]
     [--fit-width] [--print-empty]
     [--description <description>]
     [--share | --no-share]
     [--project <project>]
     [--project-domain <project-domain>]
     [--default | --no-default]
     <name>
```

Table 54.154. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of qos policy to create</td>
</tr>
</tbody>
</table>

Table 54.155. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the qos policy</td>
</tr>
<tr>
<td>--share</td>
<td>Make the qos policy accessible by other projects</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--no-share</td>
<td>Make the qos policy not accessible by other projects (default)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--default</td>
<td>Set this as a default network qos policy</td>
</tr>
<tr>
<td>--no-default</td>
<td>Set this as a non-default network qos policy</td>
</tr>
</tbody>
</table>

**Table 54.156. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 54.157. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.158. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 54.159. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 54.42. NETWORK QOS POLICY DELETE

Delete Qos Policy(s)

**Usage:**

```
openstack network qos policy delete [-h] <qos-policy> [qos-policy> ...]
```

Table 54.160. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 54.161. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 54.43. NETWORK QOS POLICY LIST

List QoS policies

**Usage:**

```
```
### Table 54.162. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List qos policies according to their project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--share</td>
<td>List qos policies shared between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>List qos policies not shared between projects</td>
</tr>
</tbody>
</table>

### Table 54.163. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 54.164. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 54.165. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 54.166. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.44. NETWORK QOS POLICY SET

Set QoS policy properties

Usage:


Table 54.167. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 54.168. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set qos policy name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the qos policy</td>
</tr>
<tr>
<td>--share</td>
<td>Make the qos policy accessible by other projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>Make the qos policy not accessible by other projects</td>
</tr>
<tr>
<td>--default</td>
<td>Set this as a default network qos policy</td>
</tr>
</tbody>
</table>
54.45. NETWORK QOS POLICY SHOW

Display QoS policy details

Usage:

```
openstack network qos policy show [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty]
<qos-policy>
```

Table 54.169. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy to display (name or id)</td>
</tr>
</tbody>
</table>

Table 54.170. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.171. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.172. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.173. Shell formatter options
--prefix PREFIX

Add a prefix to all variable names

Table 54.174. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.46. NETWORK QOS RULE CREATE

Create new Network QoS rule

Usage:

openstack network qos rule create [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [-noindent]
   [--prefix PREFIX]
   [--max-width <integer>] [--fit-width]
   [--print-empty] --type <type>
   [--max-kbps <max-kbps>]
   [--max-burst-kbits <max-burst-kbits>]
   [--dscp-mark <dscp-mark>]
   [--min-kbps <min-kbps>]
   [--ingress | --egress]
   <qos-policy>

Table 54.175. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy that contains the rule (name or id)</td>
</tr>
</tbody>
</table>

Table 54.176. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>Qos rule type (minimum-bandwidth, dscp-marking, bandwidth-limit)</td>
</tr>
<tr>
<td>--max-kbps &lt;max-kbps&gt;</td>
<td>Maximum bandwidth in kbps</td>
</tr>
<tr>
<td>--max-burst-kbits &lt;max-burst-kbits&gt;</td>
<td>Maximum burst in kilobits, 0 or not specified means automatic, which is 80% of the bandwidth limit, which works for typical TCP traffic. For details check the QoS user workflow.</td>
</tr>
<tr>
<td>--dscp-mark &lt;dscp-mark&gt;</td>
<td>Dscp mark: value can be 0, even numbers from 8-56, excluding 42, 44, 50, 52, and 54</td>
</tr>
<tr>
<td>--min-kbps &lt;min-kbps&gt;</td>
<td>Minimum guaranteed bandwidth in kbps</td>
</tr>
<tr>
<td>--ingress</td>
<td>Ingress traffic direction from the project point of view</td>
</tr>
<tr>
<td>--egress</td>
<td>Egress traffic direction from the project point of view</td>
</tr>
</tbody>
</table>

**Table 54.177. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 54.178. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.179. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 54.180. Table formatter options**
54.47. NETWORK QOS RULE DELETE

Delete Network QoS rule

Usage:

openstack network qos rule delete [-h] <qos-policy> <rule-id>

Table 54.181. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy that contains the rule (name or id)</td>
</tr>
<tr>
<td>&lt;rule-id&gt;</td>
<td>Network qos rule to delete (id)</td>
</tr>
</tbody>
</table>

Table 54.182. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.48. NETWORK QOS RULE LIST

List Network QoS rules

Usage:

openstack network qos rule list [-h] [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent] [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  <qos-policy>
Table 54.183. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy that contains the rule (name or id)</td>
</tr>
</tbody>
</table>

Table 54.184. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.185. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.186. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.187. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.188. Table formatter options
### 54.49. NETWORK QOS RULE SET

Set Network QoS rule properties

**Usage:**

```
```

**Table 54.189. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy that contains the rule (name or id)</td>
</tr>
<tr>
<td>&lt;rule-id&gt;</td>
<td>Network qos rule to delete (id)</td>
</tr>
</tbody>
</table>

**Table 54.190. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--max-kbps &lt;max-kbps&gt;</td>
<td>Maximum bandwidth in kbps</td>
</tr>
<tr>
<td>--max-burst-kbits &lt;max-burst-kbits&gt;</td>
<td>Maximum burst in kilobits, 0 or not specified means automatic, which is 80% of the bandwidth limit, which works for typical TCP traffic. For details check the QoS user workflow.</td>
</tr>
</tbody>
</table>
# 54.50. NETWORK QOS RULE SHOW

Display Network QoS rule details

**Usage:**

```
```

**Table 54.191. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-policy&gt;</td>
<td>Qos policy that contains the rule (name or id)</td>
</tr>
<tr>
<td>&lt;rule-id&gt;</td>
<td>Network qos rule to delete (id)</td>
</tr>
</tbody>
</table>

**Table 54.192. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 54.193. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 54.194. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.195. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.196. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.51. NETWORK QOS RULE TYPE LIST

List QoS rule types

Usage:

```
openstack network qos rule type list [-h]
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
```

Table 54.197. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 54.198. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv,json,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 54.199. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 54.200. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.201. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
54.52. NETWORK QOS RULE TYPE SHOW

Show details about supported QoS rule type

Usage:

openstack network qos rule type show [-h]
[f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
<qos-rule-type-name>

Table 54.202. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-rule-type-name&gt;</td>
<td>Name of qos rule type</td>
</tr>
</tbody>
</table>

Table 54.203. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.204. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.205. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.206. Shell formatter options
### Table 54.207. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### 54.53. NETWORK RBAC CREATE

Create network RBAC policy

**Usage:**

```
```

**Table 54.208. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rbac-object&gt;</td>
<td>The object to which this rbac policy affects (name or ID)</td>
</tr>
</tbody>
</table>

**Table 54.209. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>Type of the object that rbac policy affects &quot;address_group&quot;, &quot;address_scope&quot;, &quot;security_group&quot;, &quot;subnetpool&quot;, &quot;qos_policy&quot; or &quot;network&quot;</td>
</tr>
<tr>
<td>--action &lt;action&gt;</td>
<td>Action for the rbac policy &quot;access_as_external&quot; or &quot;access_as_shared&quot;</td>
</tr>
<tr>
<td>--target-project &lt;target-project&gt;</td>
<td>The project to which the rbac policy will be enforced (name or ID)</td>
</tr>
<tr>
<td>--target-all-projects</td>
<td>Allow creating rbac policy for all projects.</td>
</tr>
<tr>
<td>--target-project-domain &lt;target-project-domain&gt;</td>
<td>Domain the target project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>The owner project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 54.210. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.211. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.212. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
### Table 54.213. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.54. NETWORK RBAC DELETE

Delete network RBAC policy(s)

**Usage:**

```bash
openstack network rbac delete [-h] <rbac-policy> [<rbac-policy> ...]
```

**Table 54.214. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rbac-policy&gt;</td>
<td>Rbac policy(s) to delete (id only)</td>
</tr>
</tbody>
</table>

**Table 54.215. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 54.55. NETWORK RBAC LIST

List network RBAC policies

**Usage:**

```bash
openstack network rbac list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] 
[--quote {all,minimal,none,nonnumeric}] [-noindent] [-max-width <integer>]
[--fit-width] [-print-empty] 
[--sort-column SORT_COLUMN] 
[--sort-ascending | --sort-descending] 
[--type <type>] [--action <action>] 
[--long]
```
### Table 54.216. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>List network rbac policies according to given object type (&quot;address_group&quot;, &quot;address_scope&quot;, &quot;security_group&quot;, &quot;subnetpool&quot;, &quot;qos_policy&quot; or &quot;network&quot;)</td>
</tr>
<tr>
<td>--action &lt;action&gt;</td>
<td>List network rbac policies according to given action (&quot;access_as_external&quot; or &quot;access_as_shared&quot;)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

### Table 54.217. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 54.218. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
54.56. NETWORK RBAC SET

Set network RBAC policy properties

Usage:

openstack network rbac set [-h] [--target-project <target-project>] [--target-project-domain <target-project-domain>] <rbac-policy>

Table 54.221. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rbac-policy&gt;</td>
<td>Rbac policy to be modified (id only)</td>
</tr>
</tbody>
</table>

Table 54.222. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--target-project &lt;target-project&gt;</td>
<td>The project to which the rbac policy will be enforced (name or ID)</td>
</tr>
<tr>
<td>--target-project-domain &lt;target-project-domain&gt;</td>
<td>Domain the target project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

54.57. NETWORK RBAC SHOW

Display network RBAC policy details

Usage:

openstack network rbac show [-h] [-f {json,shell,table,value,yaml}]
Table 54.223. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rbac-policy&gt;</td>
<td>Rbac policy (id only)</td>
</tr>
</tbody>
</table>

Table 54.224. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.225. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.226. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.227. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.228. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
54.58. NETWORK SEGMENT CREATE

Create new network segment

Usage:

```
openstack network segment create [-h]
        [-f {json,shell,table,value,yaml}]
        [-c COLUMN] [-noindent]
        [--prefix PREFIX]
        [--max-width <integer>] [--fit-width]
        [--print-empty]
        [--description <description>]
        [--physical-network <physical-network>]
        [--segment <segment>] --network
        <network> --network-type
        <network-type>
        <name>
```

Table 54.229. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New network segment name</td>
</tr>
</tbody>
</table>

Table 54.230. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Network segment description</td>
</tr>
<tr>
<td>--physical-network &lt;physical-network&gt;</td>
<td>Physical network name of this network segment</td>
</tr>
<tr>
<td>--segment &lt;segment&gt;</td>
<td>Segment identifier for this network segment which is based on the network type, VLAN ID for vlan network type and tunnel ID for geneve, gre and vxlan network types</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>Network this network segment belongs to (name or id)</td>
</tr>
<tr>
<td>--network-type &lt;network-type&gt;</td>
<td>Network type of this network segment (flat, geneve, gre, local, vlan or vxlan)</td>
</tr>
</tbody>
</table>

**Table 54.231. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 54.232. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 54.233. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 54.234. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.59. NETWORK SEGMENT DELETE
Delete network segment(s)

Usage:

```bash
openstack network segment delete [-h] <network-segment> [...]
```

Table 54.235. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-segment&gt;</td>
<td>Network segment(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 54.236. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.60. NETWORK SEGMENT LIST

List network segments

Usage:

```bash
```

Table 54.237. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>List network segments that belong to this network (name or ID)</td>
</tr>
</tbody>
</table>

Table 54.238. Output formatter options
### Table 54.239. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 54.240. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 54.241. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.61. NETWORK SEGMENT RANGE CREATE

Create new network segment range

**Usage:**
openstack network segment range create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [-prefix PREFIX]
    [-max-width <integer>]
    [-fit-width] [-print-empty]
    [-private | --shared]
    [--project <project>]
    [--project-domain <project-domain>]
    --network-type <network-type>
    [--physical-network <physical-network-name>]
    --minimum
    <minimum-segmentation-id>
    --maximum
    <maximum-segmentation-id>
    <name>

Table 54.242. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of new network segment range</td>
</tr>
</tbody>
</table>

Table 54.243. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--private</td>
<td>Network segment range is assigned specifically to</td>
</tr>
<tr>
<td></td>
<td>the project</td>
</tr>
<tr>
<td>--shared</td>
<td>Network segment range is shared with other projects</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Network segment range owner (name or id). optional</td>
</tr>
<tr>
<td></td>
<td>when the segment range is shared</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can</td>
</tr>
<tr>
<td></td>
<td>be used in case collisions between project names</td>
</tr>
<tr>
<td></td>
<td>exist.</td>
</tr>
<tr>
<td>--network-type &lt;network-type&gt;</td>
<td>Network type of this network segment range</td>
</tr>
<tr>
<td></td>
<td>(geneve, gre, vlan or vxlan)</td>
</tr>
<tr>
<td>--physical-network &lt;physical-network-name&gt;</td>
<td>Physical network name of this network segment range</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference

768
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--minimum &lt;minimum-segmentation-id&gt;</code></td>
<td>Minimum segment identifier for this network segment range which is based on the network type, VLAN ID for vlan network type and tunnel ID for geneve, gre and vxlan network types</td>
</tr>
<tr>
<td><code>--maximum &lt;maximum-segmentation-id&gt;</code></td>
<td>Maximum segment identifier for this network segment range which is based on the network type, VLAN ID for vlan network type and tunnel ID for geneve, gre and vxlan network types</td>
</tr>
</tbody>
</table>

Table 54.244. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.245. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.246. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.247. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
54.62. NETWORK SEGMENT RANGE DELETE

Delete network segment range(s)

Usage:

openstack network segment range delete [-h]
    <network-segment-range>
    [<network-segment-range> ...]

Table 54.248. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-segment-range&gt;</td>
<td>Network segment range(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 54.249. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.63. NETWORK SEGMENT RANGE LIST

List network segment ranges

Usage:

openstack network segment range list [-h]
    [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT.COLUMN]
    [--sort-ascending | --sort-descending]
    [--long] [--used | --unused]
    [--available | --unavailable]

Table 54.250. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--used</td>
<td>List network segment ranges that have segments in use</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--unused</td>
<td>List network segment ranges that have segments not in use</td>
</tr>
<tr>
<td>--available</td>
<td>List network segment ranges that have available segments</td>
</tr>
<tr>
<td>--unavailable</td>
<td>List network segment ranges without available segments</td>
</tr>
</tbody>
</table>

Table 54.251. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.252. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.253. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.254. Table formatter options
### 54.64. NETWORK SEGMENT RANGE SET

Set network segment range properties

**Usage:**

```bash
openstack network segment range set [-h] [--name <name>] [--minimum <minimum-segmentation-id>] [--maximum <maximum-segmentation-id>] <network-segment-range>
```

**Table 54.255. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network-segment-range&gt;</code></td>
<td>Network segment range to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 54.256. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set network segment name</td>
</tr>
<tr>
<td>--minimum &lt;minimum-segmentation-id&gt;</td>
<td>Set network segment range minimum segment identifier</td>
</tr>
<tr>
<td>--maximum &lt;maximum-segmentation-id&gt;</td>
<td>Set network segment range maximum segment identifier</td>
</tr>
</tbody>
</table>

### 54.65. NETWORK SEGMENT RANGE SHOW

Display network segment range details
Usage:

```
openstack network segment range show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    <network-segment-range>
```

Table 54.257. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-segment-range&gt;</td>
<td>Network segment range to display (name or id)</td>
</tr>
</tbody>
</table>

Table 54.258. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.259. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.260. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.261. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.262. Table formatter options
54.66. NETWORK SEGMENT SET

Set network segment properties

Usage:

openstack network segment set [-h] [--description <description>] [--name <name>] <network-segment>

Table 54.263. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network-segment&gt;</td>
<td>Network segment to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 54.264. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set network segment description</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set network segment name</td>
</tr>
</tbody>
</table>

54.67. NETWORK SEGMENT SHOW

Display network segment details

Usage:

openstack network segment show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-noindent] [-prefix PREFIX]
Table 54.265. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;network-segment&gt;</code></td>
<td>Network segment to display (name or id)</td>
</tr>
</tbody>
</table>

Table 54.266. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.267. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.268. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.269. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.270. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
**54.68. NETWORK SERVICE PROVIDER LIST**

List Service Providers

Usage:

```
openstack network service provider list [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [-print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
```

Table 54.271. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 54.272. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 54.273. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.274. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.275. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.69. NETWORK SET

Set network properties

Usage:

```
openstack network set [-h] [--name <name>] [--enable | --disable]
                        [--share | --no-share]
                        [--description <description>] [--mtu <mtu>]
                        [--enable-port-security | --disable-port-security]
                        [--external | --internal]
                        [--default | --no-default]
                        [--qos-policy <qos-policy> | --no-qos-policy]
                        [--tag <tag>] [--no-tag]
                        [--provider-network-type <provider-network-type>]
                        [--provider-physical-network <provider-physical-network>]
                        [--provider-segment <provider-segment>]
                        [--dns-domain <dns-domain>]
                        <network>
```

Table 54.276. Positional arguments
### Table 54.277. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Network to modify (name or id)</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set network name</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable network</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable network</td>
</tr>
<tr>
<td>--share</td>
<td>Share the network between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>Do not share the network between projects</td>
</tr>
<tr>
<td>--description &lt;description</td>
<td>Set network description</td>
</tr>
<tr>
<td>--mtu &lt;mtu</td>
<td>Set network mtu</td>
</tr>
<tr>
<td>--enable-port-security</td>
<td>Enable port security by default for ports created on this network</td>
</tr>
<tr>
<td>--disable-port-security</td>
<td>Disable port security by default for ports created on this network</td>
</tr>
<tr>
<td>--external</td>
<td>Set this network as an external network (external-net extension required)</td>
</tr>
<tr>
<td>--internal</td>
<td>Set this network as an internal network</td>
</tr>
<tr>
<td>--default</td>
<td>Set the network as the default external network</td>
</tr>
<tr>
<td>--no-default</td>
<td>Do not use the network as the default external network</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
<td>Qos policy to attach to this network (name or id)</td>
</tr>
<tr>
<td>--no-qos-policy</td>
<td>Remove the qos policy attached to this network</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the network (repeat option to set multiple tags)</td>
</tr>
</tbody>
</table>
Clear tags associated with the network. Specify both --tag and --no-tag to overwrite current tags.

The physical mechanism by which the virtual network is implemented. For example: flat, geneve, gre, local, vlan, vxlan.

Name of the physical network over which the virtual network is implemented.

Vlan id for vlan networks or tunnel id for GENEVE/GRE/VXLAN networks.

Set dns domain for this network (requires dns integration extension).

---

### 54.70. NETWORK SHOW

Show network details.

**Usage:**

```bash
```

**Table 54.278. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Network to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 54.279. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 54.280. Output formatter options**
Table 54.281. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show</td>
</tr>
<tr>
<td></td>
<td>multiple columns</td>
</tr>
</tbody>
</table>

Table 54.282. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.283. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the</td>
</tr>
<tr>
<td></td>
<td>CLIFF_MAX_TERM_WIDTH environment variable, but the parameter</td>
</tr>
<tr>
<td></td>
<td>takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width</td>
</tr>
<tr>
<td></td>
<td>greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1</td>
</tr>
<tr>
<td></td>
<td>to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.71. NETWORK SUBPORT LIST

List all subports for a given network trunk

Usage:

```
```
Table 54.284. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--trunk &lt;trunk&gt;</td>
<td>List subports belonging to this trunk (name or id)</td>
</tr>
</tbody>
</table>

Table 54.285. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 54.286. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 54.287. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.288. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
54.72. NETWORK TRUNK CREATE

Create a network trunk for a given project

Usage:

```plaintext
```

Table 54.289. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the trunk to create</td>
</tr>
</tbody>
</table>

Table 54.290. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>A description of the trunk</td>
</tr>
<tr>
<td>--parent-port &lt;parent-port&gt;</td>
<td>Parent port belonging to this trunk (name or id)</td>
</tr>
<tr>
<td>--subport &lt;port=,segmentation-type=,segmentation-id=&gt;</td>
<td>Subport to add. subport is of form <code>port=&lt;name or ID&gt;,segmentation-type=,segmentation-id=</code> (<code>subport</code>) option can be repeated</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable trunk (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable trunk</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 54.291. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.292. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.293. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.294. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
54.73. NETWORK TRUNK DELETE

Delete a given network trunk

Usage:

openstack network trunk delete [-h] <trunk> [<trunk> ...]

Table 54.295. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trunk&gt;</td>
<td>Trunk(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 54.296. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

54.74. NETWORK TRUNK LIST

List all network trunks

Usage:


Table 54.297. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>
### Table 54.298. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 54.299. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 54.300. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 54.301. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 54.75. NETWORK TRUNK SET

Set network trunk properties
Usage:


Table 54.302. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trunk&gt;</td>
<td>Trunk to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 54.303. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set trunk name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>A description of the trunk</td>
</tr>
<tr>
<td>--subport &lt;port=,segmentation-type=,segmentation-id=&gt;</td>
<td>Subport to add. subport is of form port=&lt;name or ID&gt;,segmentation-type=,segmentation-ID=(--subport) option can be repeated</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable trunk</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable trunk</td>
</tr>
</tbody>
</table>

54.76. NETWORK TRUNK SHOW

Show information of a given network trunk

Usage:


Table 54.304. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trunk&gt;</td>
<td>Trunk to display (name or id)</td>
</tr>
</tbody>
</table>
Table 54.305. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 54.306. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 54.307. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 54.308. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 54.309. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

54.77. NETWORK TRUNK UNSET

Unset subports from a given network trunk

Usage:
openstack network trunk unset [-h] --subport <subport> <trunk>

Table 54.310. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trunk&gt;</td>
<td>Unset subports from this trunk (name or id)</td>
</tr>
</tbody>
</table>

Table 54.311. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--subport &lt;subport&gt;</td>
<td>Subport to delete (name or id of the port) (--subport) option can be repeated</td>
</tr>
</tbody>
</table>

54.78. NETWORK UNSET

Unset network properties

Usage:

openstack network unset [-h] [--tag <tag> | --all-tag] <network>

Table 54.312. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network&gt;</td>
<td>Network to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 54.313. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the network (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the network</td>
</tr>
</tbody>
</table>
CHAPTER 55. OBJECT

This chapter describes the commands under the `object` command.

55.1. OBJECT CREATE

Upload object to container

**Usage:**

```bash
openstack object create [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
   [--quote {all,minimal,none,nonnumeric}] [-noindent]
   [--max-width <integer>] [--fit-width] [--print-empty]
   [--sort-column SORT_COLUMN] [--sort-ascending | --sort-descending]
   [--name <name>]
   <container> <filename> [<filename> ...]
```

<table>
<thead>
<tr>
<th>Table 55.1. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>&lt;container&gt;</td>
</tr>
<tr>
<td>&lt;filename&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 55.2. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 55.3. Output formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
</tbody>
</table>
| -f {csv,json,table,value,yaml}, --format
   {csv,json,table,value,yaml} | The output format, defaults to table |
| -c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns |
## 55.2. OBJECT DELETE

Delete object from container

**Usage:**

```
openstack object delete [-h] <container> <object> [<object> ...]
```

### Table 55.7. Positional arguments
Table 55.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Delete object(s) from &lt;container&gt;</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>Object(s) to delete</td>
</tr>
</tbody>
</table>

Table 55.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Container to list</td>
</tr>
</tbody>
</table>

Table 55.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--prefix &lt;prefix&gt;</td>
<td>Filter list using &lt;prefix&gt;</td>
</tr>
<tr>
<td>--delimiter &lt;delimiter&gt;</td>
<td>Roll up items with &lt;delimiter&gt;</td>
</tr>
<tr>
<td>--marker &lt;marker&gt;</td>
<td>Anchor for paging</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--end-marker</td>
<td>End anchor for paging</td>
</tr>
<tr>
<td>--limit num-objects</td>
<td>Limit the number of objects returned</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--all</td>
<td>List all objects in container (default is 10000)</td>
</tr>
</tbody>
</table>

**Table 55.11. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 55.12. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 55.13. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 55.14. Table formatter options**
55.4. OBJECT SAVE

Save object locally

Usage:

```bash
openstack object save [-h] [--file <filename>] <container> <object>
```

Table 55.15. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Download &lt;object&gt; from &lt;container&gt;</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>Object to save</td>
</tr>
</tbody>
</table>

Table 55.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--file &lt;filename&gt;</td>
<td>Destination filename (defaults to object name); using - as the filename will print the file to stdout</td>
</tr>
</tbody>
</table>

55.5. OBJECT SET

Set object properties

Usage:

```bash
openstack object set [-h] --property <key=value> <container> <object>
```

Table 55.17. Positional arguments
### Table 55.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;container&gt;</code></td>
<td>Modify <code>&lt;object&gt;</code> from <code>&lt;container&gt;</code></td>
</tr>
<tr>
<td><code>&lt;object&gt;</code></td>
<td>Object to modify</td>
</tr>
</tbody>
</table>

### 55.6. OBJECT SHOW

Display object details

**Usage:**

```
```

### Table 55.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;container&gt;</code></td>
<td>Display <code>&lt;object&gt;</code> from <code>&lt;container&gt;</code></td>
</tr>
<tr>
<td><code>&lt;object&gt;</code></td>
<td>Object to display</td>
</tr>
</tbody>
</table>

### Table 55.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 55.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
55.7. OBJECT STORE ACCOUNT SET

Set account properties

Usage:

openstack object store account set [-h] --property <key=value>

Table 55.25. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
55.8. OBJECT STORE ACCOUNT SHOW

Display account details

Usage:

```
openstack object store account show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
```

Table 55.26. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 55.27. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 55.28. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 55.29. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 55.30. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

55.9. OBJECT STORE ACCOUNT UNSET

Unset account properties

Usage:

```bash
openstack object store account unset [-h] --property <key>
```

Table 55.31. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to remove from account (repeat option to remove multiple properties)</td>
</tr>
</tbody>
</table>

55.10. OBJECT UNSET

Unset object properties

Usage:

```bash
openstack object unset [-h] --property <key> <container> <object>
```

Table 55.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container&gt;</td>
<td>Modify &lt;object&gt; from &lt;container&gt;</td>
</tr>
<tr>
<td>&lt;object&gt;</td>
<td>Object to modify</td>
</tr>
</tbody>
</table>

Table 55.33. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to remove from object (repeat option to remove multiple properties)</td>
</tr>
</tbody>
</table>
CHAPTER 56. ORCHESTRATION

This chapter describes the commands under the orchestration command.

56.1. ORCHESTRATION BUILD INFO

Retrieve build information.

Usage:

openstack orchestration build info [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]

Table 56.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 56.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to</td>
</tr>
<tr>
<td></td>
<td>show multiple columns</td>
</tr>
</tbody>
</table>

Table 56.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 56.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 56.5. Table formatter options
### 56.2. ORCHESTRATION RESOURCE TYPE LIST

List resource types.

#### Usage:

```
```

#### Table 56.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--help [-h]</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--filter &lt;key=value&gt;</td>
<td>Filter parameters to apply on returned resource types. This can be specified multiple times. It can be any of name, version or support_status</td>
</tr>
<tr>
<td>--long</td>
<td>Show resource types with corresponding description of each resource type.</td>
</tr>
</tbody>
</table>

#### Table 56.7. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
### Table 56.8. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 56.9. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 56.10. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 56.3. ORCHESTRATION RESOURCE TYPE SHOW

Show details and optionally generate a template for a resource type.

Usage:
openstack orchestration resource type show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--template-type <template-type>]
    [--long]
    <resource-type>

### Table 56.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;resource-type&gt;</td>
<td>Resource type to show details for</td>
</tr>
</tbody>
</table>

### Table 56.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--template-type &lt;template-type&gt;</td>
<td>Optional template type to generate, hot or cfn</td>
</tr>
<tr>
<td>--long</td>
<td>Show resource type with corresponding description.</td>
</tr>
</tbody>
</table>

### Table 56.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 56.14. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 56.16. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

Print empty table if there is no data to show.

### 56.4. ORCHESTRATION SERVICE LIST

List the Heat engines.

**Usage:**

```bash
```

Table 56.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 56.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 56.19. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 56.20. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 56.21. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

56.5. ORCHESTRATION TEMPLATE FUNCTION LIST

List the available functions.

**Usage:**

```
openstack orchestration template function list [-h]
   [-f {csv,json,table,value,yaml}]
```
Table 56.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;template-version&gt;</td>
<td>Template version to get the functions for</td>
</tr>
</tbody>
</table>

Table 56.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--with_conditions</td>
<td>Show condition functions for template.</td>
</tr>
</tbody>
</table>

Table 56.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 56.25. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 56.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 56.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

56.6. ORCHESTRATION TEMPLATE VALIDATE

Validate a template

Usage:

openstack orchestration template validate [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [-e <environment>]
  [--show-nested]
  [-p <key=value>]
  [-s <files-container>]
  [--ignore-errors <error1,error2,...>]
  -t <template>

Table 56.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-e &lt;environment&gt;, --environment &lt;environment&gt;</td>
<td>Path to the environment. Can be specified multiple times</td>
</tr>
<tr>
<td>--show-nested</td>
<td>Resolve parameters from nested templates as well</td>
</tr>
</tbody>
</table>
Parameter values used to create the stack. this can be specified multiple times

Swift files container name. local files other than root template would be ignored. If other files are not found in swift, heat engine would raise an error.

List of heat errors to ignore

Path to the template

Table 56.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 56.30. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 56.31. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 56.32. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
56.7. ORCHESTRATION TEMPLATE VERSION LIST

List the available template versions.

Usage:

```bash
```

Table 56.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 56.34. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 56.35. CSV formatter options
### Table 56.36. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 56.37. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 57. OVERCLOUD

This chapter describes the commands under the overcloud command.

57.1. OVERCLOUD ADMIN AUTHORIZE

Deploy the ssh keys needed by Mistral.

Usage:

openstack overcloud admin authorize [-h] [--stack STACK]
    [--overcloud-ssh-user OVERCLOUD_SSH_USER]
    [--overcloud-ssh-key OVERCLOUD_SSH_KEY]
    [--overcloud-ssh-network OVERCLOUD_SSH_NETWORK]
    [--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT]
    [--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT]
    [--static-inventory STATIC_INVENTORY]
    [--limit LIMIT_HOSTS]

Table 57.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to overcloud nodes</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes. When undefined the key will be autodetected.</td>
</tr>
<tr>
<td>--overcloud-ssh-network OVERCLOUD_SSH_NETWORK</td>
<td>Network name to use for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td>--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT</td>
<td>This option no longer has any effect.</td>
</tr>
<tr>
<td>--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT</td>
<td>Timeout for the ssh port to become active.</td>
</tr>
<tr>
<td>--static-inventory STATIC_INVENTORY</td>
<td>Path to an existing ansible inventory to use. If not specified, one will be generated in ~/tripleo-ansible-inventory.yaml</td>
</tr>
<tr>
<td>--limit LIMIT_HOSTS</td>
<td>Define which hosts or group of hosts to run the admin Authorize tasks against.</td>
</tr>
</tbody>
</table>
57.2. OVERCLOUD BACKUP

Backup the Overcloud

Usage:

```
openstack overcloud backup [--init [INIT]] [--setup-nfs] [--setup-rear]
                          [--setup-ironic] [--cron]
                          [--inventory INVENTORY]
                          [--storage-ip STORAGE_IP]
                          [--extra-vars EXTRA_VARS]
```

Table 57.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--init [INIT]</td>
<td>Initialize environment for backup, using rear, nfs or ironic as args which will check for package install and configured ReaR or NFS server. Defaults to: rear. i.e. --init rear. WARNING: This flag will be deprecated and replaced by --setup-rear, --setup-nfs and --setup-ironic.</td>
</tr>
<tr>
<td>--setup-nfs</td>
<td>Setup the nfs server on the backup node which will install required packages and configuration on the host BackupNode in the ansible inventory.</td>
</tr>
<tr>
<td>--setup-rear</td>
<td>Setup rear on the overcloud controller hosts which will install and configure ReaR.</td>
</tr>
<tr>
<td>--setup-ironic</td>
<td>Setup rear on the overcloud controller hosts which will install and configure ReaR with ironic</td>
</tr>
<tr>
<td>--cron</td>
<td>Sets up a new cron job that by default will execute a weekly backup at Sundays midnight, but that can be customized by using the tripleo_backup_and_restore_cron extra-var.</td>
</tr>
<tr>
<td>--inventory INVENTORY</td>
<td>Tripleo inventory file generated with tripleo-ansible-inventory command. Defaults to: /root/config-download/overcloud/tripleo-ansible-inventory.yaml</td>
</tr>
<tr>
<td>--storage-ip STORAGE_IP</td>
<td>Storage ip is an optional parameter which allows for an ip of a storage server to be specified, overriding the default undercloud. WARNING: This flag will be deprecated in favor of --extra-vars which will allow to pass this and other variables.</td>
</tr>
</tbody>
</table>


57.3. OVERCLOUD CELL EXPORT

Export cell information used as import of another cell

Usage:

openstack overcloud cell export [-h]
    [--control-plane-stack <control plane stack>]
    [--cell-stack <cell stack>]
    [--output-file <output file>]
    [--working-dir WORKING_DIR]
    [--config-download-dir CONFIG_DOWNLOAD_DIR]
    [--force-overwrite]

Table 57.3. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--extra-vars EXTRA_VARS</td>
<td>Set additional variables as dict or as an absolute path of a JSON or YAML file type. i.e. --extra-vars &quot;key&quot;: &quot;val&quot;, &quot;key2&quot;: &quot;val2&quot; i.e. --extra-vars /path/to/my_vars.yaml i.e. --extra-vars /path/to/my_vars.json. For more information about the variables that can be passed, visit: <a href="https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml">https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml</a>.</td>
</tr>
</tbody>
</table>

-h, --help
    Show this help message and exit

--control-plane-stack <control plane stack>
    Name of the environment main heat stack to export information from. (default=Env: OVERCLOUD_STACK_NAME)

--cell-stack <cell stack>, -e <cell stack>
    Name of the controller cell heat stack to export information from. Used in case of: control plane stack → cell controller stack → multiple compute stacks

--output-file <output file>, -o <output file>
    Name of the output file for the cell data export. it will default to "<name>.yaml"

--working-dir WORKING_DIR
    The working directory for the deployment where all input, output, and generated files are stored. Defaults to "$HOME/overcloud-deploy/<stack>"
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--config-download-dir CONFIG_DOWNLOAD_DIR</td>
<td>Directory to search for config-download export data. Defaults to $HOME/overcloud-deploy/&lt;stack&gt;/config-download</td>
</tr>
<tr>
<td>--force-overwrite, -f</td>
<td>Overwrite output file if it exists.</td>
</tr>
</tbody>
</table>

**57.4. OVERCLOUD CEPH DEPLOY**

**Usage:**

```bash
openstack overcloud ceph deploy [-h] -o <deployed_ceph.yaml> [-y]
  [--skip-user-create]
  [--skip-hosts-config]
  [--skip-container-registry-config]
  [--cephadm-ssh-user CEPHADM_SSH_USER]
  [--stack STACK]
  [--working-dir WORKING_DIR]
  [--roles-data ROLES_DATA]
  [--network-data NETWORK_DATA]
  [--public-network-name PUBLIC_NETWORK_NAME]
  [--cluster-network-name CLUSTER_NETWORK_NAME]
  [--cluster CLUSTER] [--mon-ip MON_IP]
  [--config CONFIG]
  [--cephadm-extra-args CEPHADM_EXTRA_ARGS]
  [--force]
  [--ansible-extra-vars ANSIBLE_EXTRA_VARS]
  [--ceph-client-username CEPH_CLIENT_USERNAME]
  [--ceph-client-key CEPH_CLIENT_KEY]
  [--skip-cephx-keys]
  [--single-host-defaults]
  [--ceph-spec CEPH_SPEC | --osd-spec OSD_SPEC]
  [--crush-hierarchy CRUSH_HIERARCHY]
  [--standalone]
  [--container-image-prepare CONTAINER_IMAGE_PREPARE]
  [--cephadm-default-container]
  [--container-namespace CONTAINER_NAMESPACE]
  [--container-image CONTAINER_IMAGE]
  [--container-tag CONTAINER_TAG]
  [--registry-url REGISTRY_URL]
  [--registry-username REGISTRY_USERNAME]
  [--registry-password REGISTRY_PASSWORD]
  [<deployed_baremetal.yaml>]
```

Table 57.4. Positional arguments
### Table 57.5. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;deployed_baremetal.yaml&gt;</code></td>
<td>Path to the environment file output from &quot;openstack overcloud node provision&quot;. This argument may be excluded only if <code>--ceph-spec</code> is used.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>-o &lt;deployed_ceph.yaml&gt;, --output</code></td>
<td>The path to the output environment file describing the Ceph deployment to pass to the overcloud deployment.</td>
</tr>
<tr>
<td><code>-y, --yes</code></td>
<td>Skip yes/no prompt before overwriting an existing <code>&lt;deployed_ceph.yaml&gt;</code> output file (assume yes).</td>
</tr>
<tr>
<td><code>--skip-user-create</code></td>
<td>Do not create the cephadm ssh user. This user is necessary to deploy but may be created in a separate step via <code>openstack overcloud ceph user enable</code>.</td>
</tr>
<tr>
<td><code>--skip-hosts-config</code></td>
<td>Do not update <code>/etc/hosts</code> on deployed servers. By default this is configured so overcloud nodes can reach each other and the undercloud by name.</td>
</tr>
<tr>
<td><code>--skip-container-registry-config</code></td>
<td>Do not update <code>/etc/containers/registries.conf</code> on deployed servers. By default this is configured so overcloud nodes can pull containers from the undercloud registry.</td>
</tr>
<tr>
<td><code>--cephadm-ssh-user CEPHADM_SSH_USER</code></td>
<td>Name of the ssh user used by cephadm. warning: if this option is used, it must be used consistently for every <code>openstack overcloud ceph</code> call. Defaults to <code>ceph-admin</code>. (default=Env: CEPHADM_SSH_USER)</td>
</tr>
<tr>
<td><code>--stack STACK</code></td>
<td>Name or id of heat stack (default=env: <code>OVERCLOUD_STACK_NAME</code>)</td>
</tr>
<tr>
<td><code>--working-dir WORKING_DIR</code></td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td><code>--roles-data ROLES_DATA</code></td>
<td>Path to an alternative <code>roles_data.yaml</code> used to decide which node gets which Ceph mon, mgr, or osd service based on the node’s role in <code>&lt;deployed_baremetal.yaml&gt;</code>.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--network-data NETWORK_DATA</code></td>
<td>Path to an alternative network_data.yaml used to define Ceph public_network and cluster_network. This file is searched for networks with name_lower values of storage and storage_mgmt. If none found, then search repeats but with service_net_map_replace in place of name_lower. Use <code>--public-network-name</code> or <code>--cluster-network-name</code> options to override name of the searched for network from storage or storage_mgmt to a customized name. If network_data has no storage networks, both default to ctlplane. If found network has &gt;1 subnet, they are all combined (for routed traffic). If a network has ipv6 true, then the ipv6_subnet is retrieved instead of the ip_subnet, and the Ceph global ms_bind_ipv4 is set false and the ms_bind_ipv6 is set true. Use <code>--config</code> to override these defaults if desired.</td>
</tr>
<tr>
<td><code>--public-network-name PUBLIC_NETWORK_NAME</code></td>
<td>Name of the network defined in network_data.yaml which should be used for the Ceph public_network. Defaults to storage.</td>
</tr>
<tr>
<td><code>--cluster-network-name CLUSTER_NETWORK_NAME</code></td>
<td>Name of the network defined in network_data.yaml which should be used for the Ceph cluster_network. Defaults to storage_mgmt.</td>
</tr>
<tr>
<td><code>--cluster CLUSTER</code></td>
<td>Name of the ceph cluster. If set to foo, then the files /etc/ceph/&lt;FSID&gt;/foo.conf and /etc/ceph/&lt;FSID&gt;/foo.client.admin.keyring will be created. Otherwise these files will use the name ceph. Changing this means changing command line calls too, e.g. <code>ceph health</code> will become <code>ceph --cluster foo health</code> unless export CEPH_ARGS=<code>--cluster foo</code> is used.</td>
</tr>
<tr>
<td><code>--mon-ip MON_IP</code></td>
<td>Ip address of the first ceph monitor. If not set, an IP from the Ceph public_network of a server with the mon label from the Ceph spec is used. IP must already be active on server.</td>
</tr>
<tr>
<td><code>--config CONFIG</code></td>
<td>Path to an existing ceph.conf with settings to be assimilated by the new cluster via <code>cephadm bootstrap</code> --config</td>
</tr>
<tr>
<td><code>--cephadm-extra-args CEPHADM_EXTRA_ARGS</code></td>
<td>String of extra parameters to pass cephadm. E.g. if <code>--cephadm-extra-args --log-to-file --skip-prepare-host</code>, then cephadm bootstrap will use those options. Warning: requires --force as not all possible options ensure a functional deployment.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--force</td>
<td>Run command regardless of consequences.</td>
</tr>
<tr>
<td>--ansible-extra-vars ANSIBLE_EXTRA_VARS</td>
<td>Path to an existing ansible vars file which can override any variable in tripleo-ansible. If --ansible-extra-vars vars.yaml is passed, then ansible-playbook -e @vars.yaml ... is used to call tripleo-ansible Ceph roles. Warning: requires --force as not all options ensure a functional deployment.</td>
</tr>
<tr>
<td>--ceph-client-username CEPH_CLIENT_USERNAME</td>
<td>Name of the cephx user. e.g. if openstack is used, then ceph auth get client.openstack will return a working user with key and capabilities on the deployed Ceph cluster. Ignored unless tripleo_cephadm_pools is set via --ansible-extra-vars. If this parameter is not set and tripleo_cephadm_keys is set via --ansible-extra-vars, then openstack will be used. Used to set CephClientUserName in --output.</td>
</tr>
<tr>
<td>--ceph-client-key CEPH_CLIENT_KEY</td>
<td>Value of the cephx key. e.g. AQC*vYNXgDAGAhAAc8UoYt+OTz5uhV77LdtwUw==. Ignored unless tripleo_cephadm_pools is set via --ansible-extra-vars. If this parameter is not set and tripleo_cephadm_keys is set via --ansible-extra-vars, then a random key will be generated. Used to set CephClientKey in --output.</td>
</tr>
<tr>
<td>--skip-cephx-keys</td>
<td>Do not create cephx keys even if tripleo_cephadm_pools is set via --ansible-extra-vars. If this option is used, then even the defaults of --ceph-client-key and --ceph-client-username are ignored, but the pools defined via --ansible-extra-vars are still created.</td>
</tr>
<tr>
<td>--single-host-defaults</td>
<td>Adjust configuration defaults to suit a single-host Ceph cluster.</td>
</tr>
<tr>
<td>--ceph-spec CEPH_SPEC</td>
<td>Path to an existing ceph spec file. if not provided a spec will be generated automatically based on --roles-data and &lt;deployed_baremetal.yaml&gt;. The &lt;deployed_baremetal.yaml&gt; parameter is optional only if --ceph-spec is used.</td>
</tr>
<tr>
<td>--osd-spec OSD_SPEC</td>
<td>Path to an existing osd spec file. mutually exclusive with --ceph-spec. If the Ceph spec file is generated automatically, then the OSD spec in the Ceph spec file defaults to {data_devices: {all: true}} for all service_type osd. Use --osd-spec to override the data_devices value inside the Ceph spec file.</td>
</tr>
</tbody>
</table>
**--crush-hierarchy CRUSH_HIERARCHY**
Path to an existing crush hierarchy spec file.

**--standalone**
Use single host ansible inventory. used only for development or testing environments.

**--container-image-prepare CONTAINER_IMAGE_PREPARE**
Path to an alternative container_image_prepare_defaults.yaml. Used to control which Ceph container is pulled by cephadm via the ceph_namespace, ceph_image, and ceph_tag variables in addition to registry authentication via ContainerImageRegistryCredentials.

**--cephadm-default-container**
Use the default container defined in cephadm instead of container_image_prepare_defaults.yaml. If this is used, cephadm bootstrap is not passed the --image parameter.

### Table 57.6. container-image-prepare overrides

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--crush-hierarchy CRUSH_HIERARCHY</td>
<td>Path to an existing crush hierarchy spec file.</td>
</tr>
<tr>
<td>--standalone</td>
<td>Use single host ansible inventory. used only for development or testing environments.</td>
</tr>
<tr>
<td>--container-image-prepare CONTAINER_IMAGE_PREPARE</td>
<td>Path to an alternative container_image_prepare_defaults.yaml. Used to control which Ceph container is pulled by cephadm via the ceph_namespace, ceph_image, and ceph_tag variables in addition to registry authentication via ContainerImageRegistryCredentials.</td>
</tr>
<tr>
<td>--cephadm-default-container</td>
<td>Use the default container defined in cephadm instead of container_image_prepare_defaults.yaml. If this is used, cephadm bootstrap is not passed the --image parameter.</td>
</tr>
</tbody>
</table>

### 57.5. OVERCLOUD CEPH SPEC

**Usage:**

```
openstack overcloud ceph spec [-h] -o <ceph_spec.yaml> [-y] [-stack STACK] [-working-dir WORKING_DIR] [-roles-data ROLES_DATA]
```
[--mon-ip MON_IP] [--standalone]
[--osd-spec OSD_SPEC | --crush-hierarchy CRUSH_HIERARCHY]
[<deployed_baremetal.yaml>]

Table 57.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;deployed_baremetal.yaml&gt;</td>
<td>Path to the environment file output from &quot;openstack overcloud node provision&quot;. This argument may be excluded only if --standalone is used.</td>
</tr>
</tbody>
</table>

Table 57.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-o &lt;ceph_spec.yaml&gt;, --output &lt;ceph_spec.yaml&gt;</td>
<td>The path to the output cephadm spec file to pass to the &quot;openstack overcloud ceph deploy --ceph-spec &lt;ceph_spec.yaml&gt;&quot; command.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt before overwriting an existing &lt;ceph_spec.yaml&gt; output file (assume yes).</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td>--roles-data ROLES_DATA</td>
<td>Path to an alternative roles_data.yaml. used to decide which node gets which Ceph mon, mgr, or osd service based on the node’s role in &lt;deployed_baremetal.yaml&gt;.</td>
</tr>
<tr>
<td>--mon-ip MON_IP</td>
<td>Ip address of the first ceph monitor. only available with --standalone.</td>
</tr>
<tr>
<td>--standalone</td>
<td>Create a spec file for a standalone deployment. used for single server development or testing environments.</td>
</tr>
<tr>
<td>--osd-spec OSD_SPEC</td>
<td>Path to an existing osd spec file. when the ceph spec file is generated its OSD spec defaults to {data_devices: {all: true}} for all service_type osd. Use --osd-spec to override the data_devices value inside the Ceph spec file.</td>
</tr>
</tbody>
</table>
57.6. OVERCLOUD CEPH USER DISABLE

Usage:

```bash
openstack overcloud ceph user disable [-h] [-y]
    [--cephadm-ssh-user CEPHADM_SSH_USER]
    [--stack STACK]
    [--working-dir WORKING_DIR]
    --fsid <FSID> [--standalone]
    <ceph_spec.yaml>
```

Table 57.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ceph_spec.yaml&gt;</td>
<td>Path to an existing ceph spec file which describes the Ceph cluster where the cephadm SSH user will have their public and private keys removed and cephadm will be disabled. Spec file is necessary to determine which nodes to modify. WARNING: Ceph cluster administration or modification will no longer function.</td>
</tr>
</tbody>
</table>

Table 57.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt before disabling cephadm and its SSH user. (assume yes).</td>
</tr>
<tr>
<td>--cephadm-ssh-user CEPHADM_SSH_USER</td>
<td>Name of the ssh user used by cephadm. warning: if this option is used, it must be used consistently for every openstack overcloud ceph call. Defaults to ceph-admin. (default=Env: CEPHADM_SSH_USER)</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td>--standalone</td>
<td>Use single host ansible inventory. used only for development or testing environments.</td>
</tr>
</tbody>
</table>
Table 57.11. required named arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fsid &lt;FSID&gt;</td>
<td>The fsid of the ceph cluster to be disabled. required for disable option.</td>
</tr>
</tbody>
</table>

57.7. OVERCLOUD CEPH USER ENABLE

Usage:

```
openstack overcloud ceph user enable [-h] [--fsid <FSID>] [--standalone]
[--cephadm-ssh-user CEPHADM_SSH_USER]
[--stack STACK]
[--working-dir WORKING_DIR]
<ceph_spec.yaml>
```

Table 57.12. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ceph_spec.yaml&gt;</td>
<td>Path to an existing ceph spec file which describes the Ceph cluster where the cephadm SSH user will be created (if necessary) and have their public and private keys installed. Spec file is necessary to determine which nodes to modify and if a public or private key is required.</td>
</tr>
</tbody>
</table>

Table 57.13. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fsid &lt;FSID&gt;</td>
<td>The fsid of the ceph cluster to be (re-)enabled. if the user disable option has been used, the FSID may be passed to the user enable option so that cephadm will be re-enabled for the Ceph cluster identified by the FSID.</td>
</tr>
<tr>
<td>--standalone</td>
<td>Use single host ansible inventory. used only for development or testing environments.</td>
</tr>
<tr>
<td>--cephadm-ssh-user CEPHADM_SSH_USER</td>
<td>Name of the ssh user used by cephadm. warning: if this option is used, it must be used consistently for every openstack overcloud ceph call. Defaults to ceph- admin. (default=Env: CEPHADM_SSH_USER)</td>
</tr>
</tbody>
</table>
57.8. OVERCLOUD CONTAINER IMAGE BUILD

Build overcloud container images with kolla-build.

Usage:

```
openstack overcloud container image build [-h]
[-config-file <yaml config file>]
--kolla-config-file <config file> [-list-images]
[-list-dependencies]
[--exclude <container-name>]
[--use-buildah]
[--work-dir <container builds directory>]
[--build-timeout <build timeout in seconds>]
```

Table 57.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config-file &lt;yaml config file&gt;</td>
<td>Yaml config file specifying the images to build. may be specified multiple times. Order is preserved, and later files will override some options in previous files. Other options will append. If not specified, the default set of containers will be built.</td>
</tr>
<tr>
<td>--kolla-config-file &lt;config file&gt;</td>
<td>Path to a kolla config file to use. multiple config files can be specified, with values in later files taking precedence. By default, tripleo kolla conf file /usr/share/tripleo-common/container-images/tripleo_kolla_config_overrides.conf is added.</td>
</tr>
<tr>
<td>--list-images</td>
<td>Show the images which would be built instead of building them.</td>
</tr>
<tr>
<td>--list-dependencies</td>
<td>Show the image build dependencies instead of building them.</td>
</tr>
</tbody>
</table>
57.9. OVERCLOUD CONTAINER IMAGE PREPARE

Generate files defining the images, tags and registry.

Usage:

openstack overcloud container image prepare [-h]
    [--template-file <yaml template file>]
    [--push-destination <location>]
    [--tag <tag>]
    [--tag-from-label <image label>]
    [--namespace <namespace>]
    [--prefix <prefix>]
    [--suffix <suffix>]
    [--set <variable=value>]
    [--exclude <regex>]
    [--include <regex>]
    [--output-images-file <file path>]
    [--environment-file <file path>]
    [--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
    [--output-env-file <file path>]
    [--roles-file ROLES_FILE]
    [--modify-role MODIFY_ROLE]
    [--modify-vars MODIFY_VARS]

Table 57.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--exclude &lt;container-name&gt;</td>
<td>Name of a container to match against the list of containers to be built to skip. Can be specified multiple times.</td>
</tr>
<tr>
<td>--use-buildah</td>
<td>Use buildah instead of docker to build the images with Kolla.</td>
</tr>
<tr>
<td>--work-dir &lt;container builds directory&gt;</td>
<td>Tripleo container builds directory, storing configs and logs for each image and its dependencies.</td>
</tr>
<tr>
<td>--build-timeout &lt;build timeout in seconds&gt;</td>
<td>Build timeout in seconds.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--push-destination &lt;location&gt;</td>
<td>Location of image registry to push images to. If specified, a push_destination will be set for every image entry.</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Override the default tag substitution. If --tag-from-label is specified, start discovery with this tag. Default: 17.0</td>
</tr>
<tr>
<td>--tag-from-label &lt;image label&gt;</td>
<td>Use the value of the specified label(s) to discover the tag. Labels can be combined in a template format, for example: {version}-{release}</td>
</tr>
<tr>
<td>--namespace &lt;namespace&gt;</td>
<td>Override the default namespace substitution. Default: registry.redhat.io/rhosp-rhel9</td>
</tr>
<tr>
<td>--prefix &lt;prefix&gt;</td>
<td>Override the default name prefix substitution. Default: openstack-</td>
</tr>
<tr>
<td>--suffix &lt;suffix&gt;</td>
<td>Override the default name suffix substitution. Default:</td>
</tr>
<tr>
<td>--set &lt;variable=value&gt;</td>
<td>Set the value of a variable in the template, even if it has no dedicated argument such as &quot;--suffix&quot;.</td>
</tr>
<tr>
<td>--exclude &lt;regex&gt;</td>
<td>Pattern to match against resulting imagename entries to exclude from the final output. Can be specified multiple times.</td>
</tr>
<tr>
<td>--include &lt;regex&gt;</td>
<td>Pattern to match against resulting imagename entries to include in final output. Can be specified multiple times, entries not matching any --include will be excluded. --exclude is ignored if --include is used.</td>
</tr>
<tr>
<td>--output-images-file &lt;file path&gt;</td>
<td>File to write resulting image entries to, as well as stdout. Any existing file will be overwritten.</td>
</tr>
<tr>
<td>--environment-file &lt;file path&gt;, -e &lt;file path&gt;</td>
<td>Environment files specifying which services are containerized. Entries will be filtered to only contain images used by containerized services. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--environment-directory &lt;HEAT ENVIRONMENT DIRECTORY&gt;</td>
<td>Environment file directories that are automatically added to the update command. Entries will be filtered to only contain images used by containerized services. Can be specified more than once. Files in directories are loaded in ascending sort order.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--output-env-file &lt;file path&gt;</td>
<td>File to write heat environment file which specifies all image parameters. Any existing file will be overwritten.</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data.yaml in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--modify-role MODIFY_ROLE</td>
<td>Name of ansible role to run between every image upload pull and push.</td>
</tr>
<tr>
<td>--modify-vars MODIFY_VARS</td>
<td>Ansible variable file containing variables to use when invoking the role --modify-role.</td>
</tr>
</tbody>
</table>

### 57.10. OVERCLOUD CONTAINER IMAGE TAG DISCOVER

Discover the versioned tag for an image.

**Usage:**

```bash
openstack overcloud container image tag discover [-h] --image <container image> [--tag-from-label <image label>]
```

Table 57.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--image &lt;container image&gt;</td>
<td>Fully qualified name of the image to discover the tag for (Including registry and stable tag).</td>
</tr>
<tr>
<td>--tag-from-label &lt;image label&gt;</td>
<td>Use the value of the specified label(s) to discover the tag. Labels can be combined in a template format, for example: {version}-{release}</td>
</tr>
</tbody>
</table>

### 57.11. OVERCLOUD CONTAINER IMAGE UPLOAD

Push overcloud container images to registries.

**Usage:**

---
openstack overcloud container image upload [-h] --config-file <yaml config file> [---cleanup <full, partial, none>]

Table 57.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config-file &lt;yaml config file&gt;</td>
<td>Yaml config file specifying the image build. may be specified multiple times. Order is preserved, and later files will override some options in previous files. Other options will append.</td>
</tr>
<tr>
<td>--cleanup &lt;full, partial, none&gt;</td>
<td>Cleanup behavior for local images left after upload. The default full will attempt to delete all local images. partial will leave images required for deployment on this host. none will do no cleanup.</td>
</tr>
</tbody>
</table>

57.12. OVERCLOUD CREDENTIALS

Create the overcloudrc files

Usage:

openstack overcloud credentials [-h] [--directory [DIRECTORY]] [--working-dir WORKING_DIR] stack

Table 57.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack</td>
<td>The name of the stack you want to create rc files for.</td>
</tr>
</tbody>
</table>

Table 57.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--directory [DIRECTORY]</td>
<td>The directory to create the rc files. defaults to the current directory.</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory that contains the input, output, and generated files for the deployment. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
</tbody>
</table>
57.13. OVERCLOUD DELETE

Delete overcloud stack and plan

Usage:

```
```

Table 57.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>stack</td>
<td>Name or id of heat stack to delete (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
</tbody>
</table>

Table 57.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
<tr>
<td>-s, --skip-ipa-cleanup</td>
<td>Skip removing overcloud hosts, services, and DNS records from FreeIPA. This is particularly relevant for deployments using certificates from FreeIPA for TLS. By default, overcloud hosts, services, and DNS records will be removed from FreeIPA before deleting the overcloud. Using this option might require you to manually cleanup FreeIPA later.</td>
</tr>
<tr>
<td>-b &lt;baremetal_deployment.yaml&gt;, --baremetal-deployment &lt;baremetal_deployment.yaml&gt;</td>
<td>Configuration file describing the baremetal deployment</td>
</tr>
<tr>
<td>--networks-file &lt;network_data.yaml&gt;</td>
<td>Configuration file describing the network deployment to enable unprovisioning of networks.</td>
</tr>
<tr>
<td>--network-ports</td>
<td>Enable unprovisioning of network ports</td>
</tr>
<tr>
<td>--heat-type {installed, pod, container, native}</td>
<td>The type of heat process that was used to execute the deployment. pod (Default): Use an ephemeral Heat pod. installed: Use the system installed Heat. container: Use an ephemeral Heat container. native: Use an ephemeral Heat process.</td>
</tr>
</tbody>
</table>
57.14. OVERCLOUD DEPLOY

Deploy Overcloud

Usage:

```
openstack overcloud deploy [--templates [TEMPLATES]] [--stack STACK]
    [--timeout <TIMEOUT>]
    [--libvirt-type {kvm,qemu}]
    [--ntp-server NTP_SERVER]
    [--no-proxy NO_PROXY]
    [--overcloud-ssh-user OVERCLOUD_SSH_USER]
    [--overcloud-ssh-key OVERCLOUD_SSH_KEY]
    [--overcloud-ssh-network OVERCLOUD_SSH_NETWORK]
    [--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT]
    [--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT]
    [--environment-file <HEAT ENVIRONMENT FILE>]
    [--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
    [--roles-file ROLES_FILE]
    [--networks-file NETWORKS_FILE]
    [--vip-file VIP_FILE]
    [--plan-environment-file PLAN_ENVIRONMENT_FILE]
    [--no-cleanup] [--update-plan-only]
    [--validation-errors-nonfatal]
    [--validation-warnings-fatal]
    [--disable-validations]
    [--inflight-validations] [--dry-run]
    [--run-validations] [--skip-postconfig]
    [--force-postconfig]
    [--skip-deploy-identifier]
    [--answers-file ANSWERS_FILE]
    [--disable-password-generation]
    [--deployed-server] [--config-download]
    [--no-config-download]
    [--config-download-only] [--setup-only]
    [--config-dir CONFIG_DIR]
    [--config-type CONFIG_TYPE]
    [--no-preserve-config]
    [--output-dir OUTPUT_DIR]
    [--override-ansible-cfg OVERRIDE_ANSIBLE_CFG]
    [--config-download-timeout CONFIG_DOWNLOAD_TIMEOUT]
    [--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER]
    [-b [baremetal_deployment.yaml]]
    [--network-config] [--limit LIMIT]
    [--tags TAGS] [--skip-tags SKIP_TAGS]
    [--ansible-forks ANSIBLE_FORKS]
    [--disable-container-prepare]
    [--working-dir WORKING_DIR]
    [--heat-type {pod,container,native}]
    [--heat-container-api-image <HEAT_CONTAINER_API_IMAGE>]
    [--heat-container-engine-image <HEAT_CONTAINER_ENGINE_IMAGE>]
    [--rm-heat] [--skip-heat-pull]
    [--disable-protected-resource-types] [-y]
    [--allow-deprecated-network-data]
```
Table 57.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Stack name to create or update</td>
</tr>
<tr>
<td>--timeout &lt;TIMEOUT&gt;, -t &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes.</td>
</tr>
<tr>
<td>--libvirt-type {kvm,qemu}</td>
<td>Libvirt domain type.</td>
</tr>
<tr>
<td>--ntp-server NTP_SERVER</td>
<td>The ntp for overcloud nodes.</td>
</tr>
<tr>
<td>--no-proxy NO_PROXY</td>
<td>A comma separated list of hosts that should not be proxied.</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to overcloud nodes</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes. When undefined the key will be autodetected.</td>
</tr>
<tr>
<td>--overcloud-ssh-network OVERCLOUD_SSH_NETWORK</td>
<td>Network name to use for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td>--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT</td>
<td>This option no longer has any effect.</td>
</tr>
<tr>
<td>--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT</td>
<td>Timeout for the ssh port to become active.</td>
</tr>
<tr>
<td>--environment-file &lt;HEAT ENVIRONMENT FILE&gt;, -e &lt;HEAT ENVIRONMENT FILE&gt;</td>
<td>Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--environment-directory &lt;HEAT ENVIRONMENT DIRECTORY&gt;</td>
<td>Environment file directories that are automatically added to the heat stack-create or heat stack-update commands. Can be specified more than once. Files in directories are loaded in ascending sort order.</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data.yaml in the --templates directory. May be an absolute path or the path relative to --templates</td>
</tr>
<tr>
<td>--networks-file NETWORKS_FILE, -n NETWORKS_FILE</td>
<td>Networks file, overrides the default network_data_default.yaml in the --templates directory</td>
</tr>
<tr>
<td>--vip-file VIP_FILE</td>
<td>Configuration file describing the network virtual ips.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PLAN_ENVIRONMENT_FILE, -p PLAN_ENVIRONMENT_FILE</td>
<td></td>
</tr>
<tr>
<td>--no-cleanup</td>
<td>Don’t cleanup temporary files, just log their location</td>
</tr>
<tr>
<td>--update-plan-only</td>
<td>Deprecated: only update the plan. do not perform the actual deployment. NOTE: Will move to a discrete command in a future release. Not supported anymore.</td>
</tr>
<tr>
<td>--validation-errors-nonfatal</td>
<td>Allow the deployment to continue in spite of validation errors. Note that attempting deployment while errors exist is likely to fail.</td>
</tr>
<tr>
<td>--validation-warnings-fatal</td>
<td>Exit if there are warnings from the configuration pre-checks.</td>
</tr>
<tr>
<td>--disable-validations</td>
<td>Deprecated. disable the pre-deployment validations entirely. These validations are the built-in pre-deployment validations. To enable external validations from tripleo-validations, use the --run-validations flag. These validations are now run via the external validations in tripleo-validations.</td>
</tr>
<tr>
<td>--inflight-validations</td>
<td>Activate in-flight validations during the deploy. in-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Only run validations, but do not apply any changes.</td>
</tr>
<tr>
<td>--run-validations</td>
<td>Run external validations from the tripleo-validations project.</td>
</tr>
<tr>
<td>--skip-postconfig</td>
<td>Skip the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--force-postconfig</td>
<td>Force the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--skip-deploy-identifier</td>
<td>Skip generation of a unique identifier for the DeployIdentifier parameter. The software configuration deployment steps will only be triggered if there is an actual change to the configuration. This option should be used with Caution, and only if there is confidence that the software configuration does not need to be run, such as when scaling out certain roles.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--answers-file ANSWERS_FILE</td>
<td>Path to a yaml file with arguments and parameters.</td>
</tr>
<tr>
<td>--disable-password-generation</td>
<td>Disable password generation.</td>
</tr>
<tr>
<td>--deployed-server</td>
<td>Use pre-provisioned overcloud nodes. Removes baremetal, compute and image services requirements from the undercloud node. Must only be used with the --disable-validations.</td>
</tr>
<tr>
<td>--config-download</td>
<td>Deprecated: run deployment via config-download mechanism. This is now the default, and this CLI options has no effect.</td>
</tr>
<tr>
<td>--no-config-download, --stack-only</td>
<td>Disable the config-download workflow and only create the stack and download the config. No software configuration, setup, or any changes will be applied to overcloud nodes.</td>
</tr>
<tr>
<td>--config-download-only</td>
<td>Disable the stack create and setup, and only run the config-download workflow to apply the software configuration. Requires that config-download setup was previously completed, either with --stack-only and --setup-only or a full deployment.</td>
</tr>
<tr>
<td>--setup-only</td>
<td>Disable the stack and config-download workflow to apply the software configuration and only run the setup to enable ssh connectivity.</td>
</tr>
<tr>
<td>--config-dir CONFIG_DIR</td>
<td>The directory where the configuration files will be pushed.</td>
</tr>
<tr>
<td>--config-type CONFIG_TYPE</td>
<td>Only used when &quot;--setup-only&quot; is invoked. Type of object config to be extract from the deployment, defaults to all keys available.</td>
</tr>
<tr>
<td>--no-preserve-config</td>
<td>Only used when &quot;--setup-only&quot; is invoked. If specified, will delete and recreate the --config-dir if it already exists. Default is to use the existing dir location and overwrite files. Files in --config-dir not from the stack will be preserved by default.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to use for saved output when using --config-download. When not specified, &lt;working-dir&gt;/config-download will be used.</td>
</tr>
<tr>
<td>--override-ansible-cfg OVERRIDE_ANSIBLE_CFG</td>
<td>Path to ansible configuration file. The configuration in the file will override any configuration used by config-download by default.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--config-download-timeout CONFIG_DOWNLOAD_TIMEOUT</td>
<td>Timeout (in minutes) to use for config-download steps. If unset, will default to however much time is leftover from the --timeout parameter after the stack operation.</td>
</tr>
<tr>
<td>--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER</td>
<td>The path to python interpreter to use for the deployment actions. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td>-b [&lt;baremetal_deployment.yaml&gt;], --baremetal-deployment [&lt;baremetal_deployment.yaml&gt;]</td>
<td>Deploy baremetal nodes, network and virtual ip addresses as defined in baremetal_deployment.yaml along with overcloud. If no baremetal_deployment YAML file is given, the tripleo-&lt;stack_name&gt;-baremetal- deployment.yaml file in the working-dir will be used.</td>
</tr>
<tr>
<td>--network-config</td>
<td>Apply network config to provisioned nodes. (implies &quot;--network-ports&quot;)</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: --limit &quot;compute-0,compute-1,compute-5&quot;.</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A list of tags to use when running the the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A list of tags to skip when running the the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
</tbody>
</table>
### --heat-type {pod,container,native}

The type of heat process to use to execute the deployment. `pod` (Default): Use an ephemeral Heat pod. `container` (Experimental): Use an ephemeral Heat container. `native` (Experimental): Use an ephemeral Heat process.

---

### --heat-container-api-image `<HEAT_CONTAINER_API_IMAGE>`

The container image to use when launching the heat-api process. Only used when `--heat-type=pod`. Defaults to `localhost/tripleo/openstack-heat-api:ephemeral`

---

### --heat-container-engine-image `<HEAT_CONTAINER_ENGINE_IMAGE>`

The container image to use when launching the heat-engine process. Only used when `--heat-type=pod`. Defaults to `localhost/tripleo/openstack-heat-engine:ephemeral`

---

### --rm-heat

If specified and `--heat-type` is container or pod any existing container or pod of a previous ephemeral Heat process will be deleted first. Ignored if `--heat-type` is native.

---

### --skip-heat-pull

When `--heat-type` is pod or container, assume the container image has already been pulled

---

### --disable-protected-resource-types

Disable protected resource type overrides. Resources types that are used internally are protected, and cannot be overridden in the user environment. Setting this argument disables the protection, allowing the protected resource types to be override in the user environment.

---

### -y, --yes

Use `-y` or `--yes` to skip any confirmation required before the deploy operation. Use this with caution!

---

### --allow-deprecated-network-data

Set this to allow using deprecated network data yaml definition schema.

---

## 57.15. OVERCLOUD EXPORT CEPH

Export Ceph information used as import of another stack Export Ceph information from one or more stacks to be used as input of another stack. Creates a valid YAML file with the CephExternalMultiConfig parameter populated.

### Usage:

```
openstack overcloud export ceph [-h] [--stack <stack>]
   [--cephx-key-client-name <cephx>]
   [--output-file <output file>]
```
Table 57.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack &lt;stack&gt;</td>
<td>Name of the overcloud stack(s) to export ceph information from. If a comma delimited list of stacks is passed, Ceph information for all stacks will be exported into a single file. (default=Env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--cephx-key-client-name &lt;cephx&gt;, -k &lt;cephx&gt;</td>
<td>Name of the cephx client key to export. (default=openstack)</td>
</tr>
<tr>
<td>--output-file &lt;output file&gt;, -o &lt;output file&gt;</td>
<td>Name of the output file for the ceph data export. Defaults to &quot;ceph-export-&lt;STACK&gt;.yaml&quot; if one stack is provided. Defaults to &quot;ceph-export-&lt;N&gt;-stacks.yaml&quot; if N stacks are provided.</td>
</tr>
<tr>
<td>--force-overwrite, -f</td>
<td>Overwrite output file if it exists.</td>
</tr>
<tr>
<td>--config-download-dir CONFIG_DOWNLOAD_DIR</td>
<td>Directory to search for config-download export data. Defaults to $HOME/overcloud-deploy/stack/config-download</td>
</tr>
</tbody>
</table>

57.16. OVERCLOUD EXPORT

Export stack information used as import of another stack

Usage:

```bash
```

Table 57.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack &lt;stack&gt;</td>
<td>Name of the environment main heat stack to export information from. (default=overcloud)</td>
</tr>
</tbody>
</table>
57.17. OVERCLOUD EXTERNAL-UPDATE RUN

Run external minor update Ansible playbook This will run the external minor update Ansible playbook, executing tasks from the undercloud. The update playbooks are made available after completion of the `overcloud update prepare` command.

**Usage:**

```
openstack overcloud external-update run [-h]
  [--static-inventory STATIC_INVENTORY]
  [--ssh-user SSH_USER]
  [--tags TAGS]
  [--skip-tags SKIP_TAGS]
  [--stack STACK] [-e EXTRA_VARS]
  [-y] [-limit LIMIT]
  [--ansible-forks ANSIBLE_FORKS]
  [--refresh]
```

**Table 57.25. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--output-file &lt;output file&gt;, -o &lt;output file&gt;</code></td>
<td>Name of the output file for the stack data export. It will default to &quot;&lt;name&gt;.yaml&quot;</td>
</tr>
<tr>
<td><code>--force-overwrite, -f</code></td>
<td>Overwrite output file if it exists.</td>
</tr>
<tr>
<td><code>--working-dir WORKING_DIR</code></td>
<td>The working directory for the deployment where all input, output, and generated files are stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td><code>--config-download-dir CONFIG_DOWNLOAD_DIR</code></td>
<td>Directory to search for config-download export data. Defaults to $HOME/overcloud-deploy/&lt;stack&gt;/config-download</td>
</tr>
<tr>
<td><code>--no-password-excludes</code></td>
<td>Don't exclude certain passwords from the password export. Defaults to False in that some passwords will be excluded that are not typically necessary.</td>
</tr>
</tbody>
</table>

Note: The `--static-inventory` and `--ssh-user` arguments are deprecated. Use `tripleo-ansible-inventory.yaml` and `tripleo-admin` as the ssh user respectively.
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--tags TAGS</td>
<td>A string specifying the tag or comma separated list of tags to be passed as --tags to ansible-playbook.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A string specifying the tag or comma separated list of tags to be passed as --skip-tags to ansible-playbook.</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>-e EXTRA_VARS, --extra-vars EXTRA_VARS</td>
<td>Set additional variables as key=value or yaml/json</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Use -y or --yes to skip the confirmation required before any upgrade operation. Use this with caution!</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: --limit &quot;compute-0,compute-1,compute-5&quot;.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--refresh</td>
<td>Deprecated: refresh the config-download playbooks. use overcloud update prepare instead to refresh playbooks.</td>
</tr>
</tbody>
</table>

### 57.18. OVERCLOUD EXTERNAL-UPGRADE RUN

Run external major upgrade Ansible playbook This will run the external major upgrade Ansible playbook, executing tasks from the undercloud. The upgrade playbooks are made available after completion of the overcloud upgrade prepare command.

**Usage:**

```
openstack overcloud external-upgrade run [-h]
  [--static-inventory STATIC_INVENTORY]
  [--ssh-user SSH_USER]
  [--tags TAGS]
  [--skip-tags SKIP_TAGS]
  [--stack STACK]
  [-e EXTRA_VARS] [-y]
  [--limit LIMIT]
  [--ansible-forks ANSIBLE_FORKS]
```

Table 57.26. Command arguments
## 57.19. OVERCLOUD GENERATE FENCING

Generate fencing parameters

**Usage:**

```
openstack overcloud generate fencing [-h] [-a FENCE_ACTION] 
  [--delay DELAY] [--ipmi-lanplus] 
  [--ipmi-no-lanplus] 
  [--ipmi-cipher IPMI_CIPHER] 
  [--ipmi-level IPMI_LEVEL] 
  [--output OUTPUT] 
  instackenv
```

### Table 57.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--static-inventory STATIC_INVENTORY</code></td>
<td>Deprecated: tripleo-ansible-inventory.yaml in working dir will be used.</td>
</tr>
<tr>
<td><code>--ssh-user SSH_USER</code></td>
<td>Deprecated: only tripleo-admin should be used as ssh user.</td>
</tr>
<tr>
<td><code>--tags TAGS</code></td>
<td>A string specifying the tag or comma separated list of tags to be passed as --tags to ansible-playbook.</td>
</tr>
<tr>
<td><code>--skip-tags SKIP_TAGS</code></td>
<td>A string specifying the tag or comma separated list of tags to be passed as --skip-tags to ansible-playbook.</td>
</tr>
<tr>
<td><code>--stack STACK</code></td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td><code>-e EXTRA_VARS, --extra-vars EXTRA_VARS</code></td>
<td>Set additional variables as key=value or yaml/json</td>
</tr>
<tr>
<td><code>-y, --yes</code></td>
<td>Use -y or --yes to skip the confirmation required before any upgrade operation. Use this with caution!</td>
</tr>
<tr>
<td><code>--limit LIMIT</code></td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: --limit &quot;compute-0,compute-1,compute-5&quot;.</td>
</tr>
<tr>
<td><code>--ansible-forks ANSIBLE_FORKS</code></td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
</tbody>
</table>
### Table 57.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instackenv</td>
<td>None</td>
</tr>
</tbody>
</table>

**57.20. OVERCLOUD IMAGE BUILD**

Build images for the overcloud

**Usage:**

```
```

**Table 57.29. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config-file &lt;yaml config file&gt;</td>
<td>Yaml config file specifying the image build. may be specified multiple times. Order is preserved, and later files will override some options in previous files. Other options will append.</td>
</tr>
</tbody>
</table>
57.21. OVERCLOUD IMAGE UPLOAD

Make existing image files available for overcloud deployment.

Usage:

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--image-name &lt;image name&gt;</td>
<td>Name of image to build. may be specified multiple times. If unspecified, will build all images in given YAML files.</td>
</tr>
<tr>
<td>--no-skip</td>
<td>Skip build if cached image exists.</td>
</tr>
<tr>
<td>--output-directory OUTPUT_DIRECTORY</td>
<td>Output directory for images. defaults to $TRIPLEO_ROOT,or current directory if unset.</td>
</tr>
<tr>
<td>--temp-dir TEMP_DIR</td>
<td>Temporary directory to use when building the images. Defaults to $TMPDIR or current directory if unset.</td>
</tr>
</tbody>
</table>

Table 57.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--image-path IMAGE_PATH</td>
<td>Path to directory containing image files</td>
</tr>
<tr>
<td>--os-image-name OS_IMAGE_NAME</td>
<td>Openstack disk image filename</td>
</tr>
<tr>
<td>--ironic-python-agent-name IPA_NAME</td>
<td>Openstack ironic-python-agent (agent) image filename</td>
</tr>
</tbody>
</table>
### --http-boot HTTP_BOOT

Root directory for the ironic-python-agent image. If uploading images for multiple architectures/platforms, vary this argument such that a distinct folder is created for each architecture/platform.

### --update-existing

Update images if already exist

### --whole-disk

When set, the overcloud-full image to be uploaded will be considered as a whole disk one

### --architecture ARCHITECTURE

Architecture type for these images, x86_64, i386 and ppc64le are common options. This option should match at least one arch value in instackenv.json

### --platform PLATFORM

Platform type for these images. platform is a sub-category of architecture. For example you may have generic images for x86_64 but offer images specific to SandyBridge (SNB).

### --image-type {os,ironic-python-agent}

If specified, allows to restrict the image type to upload (os for the overcloud image or ironic-python-agent for the ironic-python-agent one)

### --progress

Show progress bar for upload files action

### --local

Deprecated: copy files locally, even if there is an image service endpoint. The default has been changed to copy files locally.

### --no-local

Upload files to image service.

### --local-path LOCAL_PATH

Root directory for image file copy destination when there is no image endpoint, or when --local is specified

### 57.22. OVERCLOUD NETENV VALIDATE

Validate the network environment file.

**Usage:**

```
openstack overcloud netenv validate [-h] [-f NETENV]
```

**Table 57.31. Command arguments**
### 57.23. OVERCLOUD NETWORK EXTRACT

**Usage:**

```
openstack overcloud network extract [-h] --stack STACK -o <network_deployment.yaml> [-y]
```

**Table 57.32. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>-o &lt;network_deployment.yaml&gt;, --output &lt;network_deployment.yaml&gt;</td>
<td>The output file path describing the network deployment</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt for existing files (assume yes).</td>
</tr>
</tbody>
</table>

### 57.24. OVERCLOUD NETWORK PROVISION

**Usage:**

```
```

**Table 57.33. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network_data.yaml&gt;</td>
<td>Configuration file describing the network deployment.</td>
</tr>
</tbody>
</table>
57.25. OVERCLOUD NETWORK UNPROVISION

Usage:

```
openstack overcloud network unprovision [-h] [-y] <network_data.yaml>
```

Table 57.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;network_data.yaml&gt;</td>
<td>Configuration file describing the network deployment.</td>
</tr>
</tbody>
</table>

Table 57.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
</tbody>
</table>

57.26. OVERCLOUD NETWORK VIP EXTRACT

Usage:

```
openstack overcloud network vip extract [-h] --stack STACK -o <vip_data.yaml> [-y]
```
### Table 57.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name of heat stack (default=env: overcloud_stack_name)</td>
</tr>
<tr>
<td>-o &lt;vip_data.yaml&gt;, --output &lt;vip_data.yaml&gt;</td>
<td>The output file path describing the virtual ip deployment</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt for existing files (assume yes).</td>
</tr>
</tbody>
</table>

### 57.27. OVERCLOUD NETWORK VIP PROVISION

#### Usage:

```
openstack overcloud network vip provision [-h] --stack STACK -o <vip_environment.yaml> [-y] [--templates TEMPLATES] [--working-dir WORKING_DIR] <vip_data.yaml>
```

#### Table 57.38. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;vip_data.yaml&gt;</td>
<td>Configuration file describing the network virtual ips.</td>
</tr>
</tbody>
</table>

#### Table 57.39. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name of heat stack (default=env: overcloud_stack_name)</td>
</tr>
<tr>
<td>-o &lt;vip_environment.yaml&gt;, --output &lt;vip_environment.yaml&gt;</td>
<td>The output virtual ip environment file path.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt for existing files (assume yes).</td>
</tr>
<tr>
<td>--templates TEMPLATES</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
</tbody>
</table>
The working directory for the deployment where all input, output, and generated files will be stored. Defaults to "$HOME/overcloud-deploy-<stack>”

### 57.28. OVERCLOUD NODE BIOS CONFIGURE

Apply BIOS configuration on given nodes

**Usage:**

```
openstack overcloud node bios configure [-h] [--all-manageable] [--configuration <configuration>] [<node_uuid> ...]
```

**Table 57.40. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node_uuid&gt;</td>
<td>Baremetal node uuids for the node(s) to configure bios</td>
</tr>
</tbody>
</table>

**Table 57.41. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-manageable</td>
<td>Configure bios for all nodes currently in manageable state</td>
</tr>
<tr>
<td>--configuration &lt;configuration&gt;</td>
<td>Bios configuration (yaml/json string or file name).</td>
</tr>
</tbody>
</table>

### 57.29. OVERCLOUD NODE BIOS RESET

Reset BIOS configuration to factory default

**Usage:**

```
openstack overcloud node bios reset [-h] [--all-manageable] [<node_uuid> ...]
```

**Table 57.42. Positional arguments**
### 57.30. OVERCLOUD NODE CLEAN

Run node(s) through cleaning.

**Usage:**

```
openstack overcloud node clean [-h] [--all-manageable] [--provide] 
  [<node_uuid> ...]
```

**Table 57.44. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node_uuid&gt;</td>
<td>Baremetal node uuids for the node(s) to be cleaned</td>
</tr>
</tbody>
</table>

**Table 57.45. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-manageable</td>
<td>Reset bios on all nodes currently in <code>manageable</code> state</td>
</tr>
<tr>
<td>--provide</td>
<td>Provide (make available) the nodes once cleaned</td>
</tr>
</tbody>
</table>

### 57.31. OVERCLOUD NODE CONFIGURE

Configure Node boot options.

**Usage:**

```
openstack overcloud node configure [-h] [--all-manageable] 
  [--deploy-kernel DEPLOY_KERNEL] 
  [--deploy-ramdisk DEPLOY_RAMDISK] 
  [--instance-boot-option {local,netboot}]```
[--boot-mode {uefi,bios}]
[--root-device ROOT_DEVICE]
[--root-device-minimum-size ROOT_DEVICE_MINIMUM_SIZE]
[--overwrite-root-device-hints]
[next_node_uuid> ...]

Table 57.46. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node_uuid&gt;</td>
<td>Baremetal node uuids for the node(s) to be configured</td>
</tr>
</tbody>
</table>

Table 57.47. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-manageable</td>
<td>Configure all nodes currently in manageable state</td>
</tr>
<tr>
<td>--deploy-kernel DEPLOY_KERNEL</td>
<td>Image with deploy kernel.</td>
</tr>
<tr>
<td>--deploy-ramdisk DEPLOY_RAMDISK</td>
<td>Image with deploy ramdisk.</td>
</tr>
<tr>
<td>--instance-boot-option {local,netboot}</td>
<td>Whether to set instances for booting from local hard drive (local) or network (netboot).</td>
</tr>
<tr>
<td>--boot-mode {uefi,bios}</td>
<td>Whether to set the boot mode to uefi (uefi) or legacy BIOS (bios)</td>
</tr>
<tr>
<td>--root-device ROOT_DEVICE</td>
<td>Define the root device for nodes, can be either a list of device names (without /dev) to choose from or one of two strategies: largest or smallest. For it to work this command should be run after the introspection.</td>
</tr>
<tr>
<td>--root-device-minimum-size ROOT_DEVICE_MINIMUM_SIZE</td>
<td>Minimum size (in gib) of the detected root device. Used with --root-device.</td>
</tr>
<tr>
<td>--overwrite-root-device-hints</td>
<td>Whether to overwrite existing root device hints when --root-device is used.</td>
</tr>
</tbody>
</table>

57.32. OVERCLOUD NODE DELETE

Delete overcloud nodes.

Usage:
openstack overcloud node delete [-h] [-b <BAREMETAL DEPLOYMENT FILE>] 
[--stack STACK] [--timeout <TIMEOUT>] 
[--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT] 
[-y] 
[node> ...]

Table 57.48. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node&gt;</td>
<td>Node id(s) to delete (otherwise specified in the --baremetal-deployment file)</td>
</tr>
</tbody>
</table>

Table 57.49. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-b &lt;BAREMETAL DEPLOYMENT FILE&gt;, --baremetal-deployment &lt;BAREMETAL DEPLOYMENT FILE&gt;</td>
<td>Configuration file describing the baremetal deployment</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack to scale (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--timeout &lt;TIMEOUT&gt;</td>
<td>Timeout in minutes to wait for the nodes to be deleted. Keep in mind that due to keystone session duration that timeout has an upper bound of 4 hours</td>
</tr>
<tr>
<td>--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT</td>
<td>Timeout for the ssh port to become active.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
</tbody>
</table>

57.33. OVERCLOUD NODE DISCOVER

Discover overcloud nodes by polling their BMCs.

Usage:

openstack overcloud node discover [-h] [-ip <ips> | --range <range>] 
[--credentials <key:value>] 
[--port <ports>] [--introspect] 
[--run-validations] [--provide] 
[--no-deploy-image] 
[--instance-boot-option {local,netboot}] 
[--concurrency CONCURRENCY]
Table 57.50. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--ip &lt;ips&gt;</td>
<td>Ip address(es) to probe</td>
</tr>
<tr>
<td>--range &lt;range&gt;</td>
<td>Ip range to probe</td>
</tr>
<tr>
<td>--credentials <a href="">key:value</a></td>
<td>Key/value pairs of possible credentials</td>
</tr>
<tr>
<td>--port &lt;ports&gt;</td>
<td>Bmc port(s) to probe</td>
</tr>
<tr>
<td>--introspect</td>
<td>Introspect the imported nodes</td>
</tr>
<tr>
<td>--run-validations</td>
<td>Run the pre-deployment validations. these external validations are from the TripleO Validations project.</td>
</tr>
<tr>
<td>--provide</td>
<td>Provide (make available) the nodes</td>
</tr>
<tr>
<td>--no-deploy-image</td>
<td>Skip setting the deploy kernel and ramdisk.</td>
</tr>
<tr>
<td>--instance-boot-option {local,netboot}</td>
<td>Whether to set instances for booting from local hard drive (local) or network (netboot).</td>
</tr>
<tr>
<td>--concurrency CONCURRENCY</td>
<td>Maximum number of nodes to introspect at once.</td>
</tr>
<tr>
<td>--node-timeout NODE_TIMEOUT</td>
<td>Maximum timeout for node introspection.</td>
</tr>
<tr>
<td>--max-retries MAX_RETRIES</td>
<td>Maximum introspection retries.</td>
</tr>
<tr>
<td>--retry-timeout RETRY_TIMEOUT</td>
<td>Maximum timeout between introspection retries.</td>
</tr>
</tbody>
</table>

57.34. OVERCLOUD NODE EXTRACT PROVISIONED

Usage:

openstack overcloud node extract provisioned [-h] [--stack STACK] [-o <baremetal_deployment.yaml>] [-y] [--roles-file ROLES_FILE]

Table 57.51. Command arguments
57.35. OVERCLOUD NODE IMPORT

Import baremetal nodes from a JSON, YAML or CSV file. The node status will be set to manageable by default.

Usage:

```
openstack overcloud node import [-h] [-i] [-r] [-v] [-v] [-c CONCURRENCY]
```

Table 57.52. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>env_file</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 57.53. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--introspect</td>
<td>Introspect the imported nodes</td>
</tr>
<tr>
<td>--run-validations</td>
<td>Run the pre-deployment validations. These external validations are from the TripleO Validations project.</td>
</tr>
</tbody>
</table>
57.36. OVERCLOUD NODE INTROSPECT

Introspect specified nodes or all nodes in manageable state.

Usage:

```
openstack overcloud node introspect [-h] [--all-manageable] [--provide]
[--run-validations] [--concurrency CONCURRENCY] [--node-timeout NODE_TIMEOUT]
[--max-retries MAX_RETRIES] [--retry-timeout RETRY_TIMEOUT] [<node_uuid> ...]
```

Table 57.54. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node_uuid&gt;</td>
<td>Baremetal node uuids for the node(s) to be introspected</td>
</tr>
</tbody>
</table>

Table 57.55. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-manageable</td>
<td>Introspect all nodes currently in manageable state</td>
</tr>
</tbody>
</table>
## 57.37. OVERCLOUD NODE PROVIDE

Mark nodes as available based on UUIDs or current manageable state.

**Usage:**

```
openstack overcloud node provide [-h] [--all-manageable] [<node_uuid> ...]
```

**Table 57.56. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;node_uuid&gt;</td>
<td>Baremetal node uuids for the node(s) to be provided</td>
</tr>
</tbody>
</table>

**Table 57.57. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-manageable</td>
<td>Provide all nodes currently in manageable state</td>
</tr>
</tbody>
</table>

## 57.38. OVERCLOUD NODE PROVISION

Provision new nodes using Ironic.

**Usage:**

```
openstack overcloud node provision [-h] [-o OUTPUT] [-y]
```
Table 57.58. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;baremetal_deployment.yaml&gt;</code></td>
<td>Configuration file describing the baremetal deployment</td>
</tr>
</tbody>
</table>

Table 57.59. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-o OUTPUT, --output OUTPUT</td>
<td>The output environment file path</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt for existing files (assume yes).</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to newly deployed nodes</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes. when undefined the key will be autodetected.</td>
</tr>
<tr>
<td>--concurrency CONCURRENCY</td>
<td>Maximum number of nodes to provision at once. (default=20)</td>
</tr>
<tr>
<td>--timeout TIMEOUT</td>
<td>Number of seconds to wait for the node provision to complete. (default=3600)</td>
</tr>
<tr>
<td>--network-ports</td>
<td>Deprecated! network ports will always be provisioned. Enable provisioning of network ports</td>
</tr>
<tr>
<td>--network-config</td>
<td>Apply network config to provisioned nodes. (implies &quot;--network-ports&quot;)</td>
</tr>
<tr>
<td>--templates TEMPLATES</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
</tbody>
</table>
### 57.39. OVERCLOUD NODE UNPROVISION

Unprovisions nodes using Ironic.

**Usage:**

```
openstack overcloud node unprovision [-h] [--stack STACK] [--all] [-y]
[--network-ports]
<baremetal_deployment.yaml>
```

**Table 57.60. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;baremetal_deployment.yaml&gt;</code></td>
<td>Configuration file describing the baremetal deployment</td>
</tr>
</tbody>
</table>

**Table 57.61. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--stack STACK</code></td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td><code>--all</code></td>
<td>Unprovision every instance in the deployment</td>
</tr>
<tr>
<td><code>-y, --yes</code></td>
<td>Skip yes/no prompt (assume yes)</td>
</tr>
<tr>
<td><code>--network-ports</code></td>
<td>Deprecated! network ports will always be unprovisioned. Enable unprovisioning of network ports</td>
</tr>
</tbody>
</table>

### 57.40. OVERCLOUD PROFILES LIST

List overcloud node profiles

**Usage:**

```
openstack overcloud profiles list [-h] [-f {csv, json, table, value, yaml}]
[-c COLUMN]
```
Table 57.62. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all</td>
<td>List all nodes, even those not available to nova.</td>
</tr>
<tr>
<td>--control-scale CONTROL_SCALE</td>
<td>New number of control nodes.</td>
</tr>
<tr>
<td>--compute-scale COMPUTE_SCALE</td>
<td>New number of compute nodes.</td>
</tr>
<tr>
<td>--ceph-storage-scale CEPH_STORAGE_SCALE</td>
<td>New number of ceph storage nodes.</td>
</tr>
<tr>
<td>--block-storage-scale BLOCK_STORAGE_SCALE</td>
<td>New number of cinder storage nodes.</td>
</tr>
<tr>
<td>--swift-storage-scale SWIFT_STORAGE_SCALE</td>
<td>New number of swift storage nodes.</td>
</tr>
<tr>
<td>--control-flavor CONTROL_FLAVOR</td>
<td>Nova flavor to use for control nodes.</td>
</tr>
<tr>
<td>--compute-flavor COMPUTE_FLAVOR</td>
<td>Nova flavor to use for compute nodes.</td>
</tr>
<tr>
<td>--ceph-storage-flavor CEPH_STORAGE_FLAVOR</td>
<td>Nova flavor to use for ceph storage nodes.</td>
</tr>
<tr>
<td>--block-storage-flavor BLOCK_STORAGE_FLAVOR</td>
<td>Nova flavor to use for cinder storage nodes</td>
</tr>
<tr>
<td>--swift-storage-flavor SWIFT_STORAGE_FLAVOR</td>
<td>Nova flavor to use for swift storage nodes</td>
</tr>
</tbody>
</table>

Table 57.63. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 57.64. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 57.65. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 57.66. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 57.41. OVERCLOUD PROFILES MATCH

Assign and validate profiles on nodes

**Usage:**
openstack overcloud profiles match [-h] [--dry-run]
    [--control-scale CONTROL_SCALE]
    [--compute-scale COMPUTE_SCALE]
    [--ceph-storage-scale CEPH_STORAGE_SCALE]
    [--block-storage-scale BLOCK_STORAGE_SCALE]
    [--swift-storage-scale SWIFT_STORAGE_SCALE]
    [--control-flavor CONTROL_FLAVOR]
    [--compute-flavor COMPUTE_FLAVOR]
    [--ceph-storage-flavor CEPH_STORAGE_FLAVOR]
    [--block-storage-flavor BLOCK_STORAGE_FLAVOR]
    [--swift-storage-flavor SWIFT_STORAGE_FLAVOR]

Table 57.67. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Only run validations, but do not apply any changes.</td>
</tr>
<tr>
<td>--control-scale CONTROL_SCALE</td>
<td>New number of control nodes.</td>
</tr>
<tr>
<td>--compute-scale COMPUTE_SCALE</td>
<td>New number of compute nodes.</td>
</tr>
<tr>
<td>--ceph-storage-scale CEPH_STORAGE_SCALE</td>
<td>New number of ceph storage nodes.</td>
</tr>
<tr>
<td>--block-storage-scale BLOCK_STORAGE_SCALE</td>
<td>New number of cinder storage nodes.</td>
</tr>
<tr>
<td>--swift-storage-scale SWIFT_STORAGE_SCALE</td>
<td>New number of swift storage nodes.</td>
</tr>
<tr>
<td>--control-flavor CONTROL_FLAVOR</td>
<td>Nova flavor to use for control nodes.</td>
</tr>
<tr>
<td>--compute-flavor COMPUTE_FLAVOR</td>
<td>Nova flavor to use for compute nodes.</td>
</tr>
<tr>
<td>--ceph-storage-flavor CEPH_STORAGE_FLAVOR</td>
<td>Nova flavor to use for ceph storage nodes.</td>
</tr>
<tr>
<td>--block-storage-flavor BLOCK_STORAGE_FLAVOR</td>
<td>Nova flavor to use for cinder storage nodes</td>
</tr>
<tr>
<td>--swift-storage-flavor SWIFT_STORAGE_FLAVOR</td>
<td>Nova flavor to use for swift storage nodes</td>
</tr>
</tbody>
</table>

57.42. OVERCLOUD RAID CREATE

Create RAID on given nodes

Usage:

```
openstack overcloud raid create [-h] --node NODE configuration
```

Table 57.68. Positional arguments
**Table 57.69. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration</td>
<td>Raid configuration (yaml/json string or file name).</td>
</tr>
</tbody>
</table>

**57.43. OVERCLOUD RESTORE**

Restore the Overcloud

**Usage:**

```
openstack overcloud restore [--inventory INVENTORY] [--stack [STACK]]
   --node-name NODE_NAME
   [--extra-vars EXTRA_VARS]
```

**Table 57.70. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--inventory INVENTORY</td>
<td>Tripleo inventory file generated with tripleo-ansible-inventory command. Defaults to: /root/config-download/overcloud/tripleo-ansible-inventory.yaml</td>
</tr>
<tr>
<td>--stack [STACK]</td>
<td>Name or id of the stack to be used(default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>--node-name NODE_NAME</td>
<td>Controller name is a required parameter which defines the controller node to be restored.</td>
</tr>
<tr>
<td>--extra-vars EXTRA_VARS</td>
<td>Set additional variables as dict or as an absolute path of a JSON or YAML file type. i.e. --extra-vars &quot;key1&quot;: &quot;val1&quot;, &quot;key2&quot;: &quot;val2&quot; i.e. --extra-vars /path/to/my_vars.yaml i.e. --extra-vars /path/to/my_vars.json. For more information about the variables that can be passed, visit: <a href="https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml">https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml</a>.</td>
</tr>
</tbody>
</table>
57.44. OVERCLOUD ROLE LIST

List available roles.

Usage:

```
openstack overcloud role list [-h] [--roles-path <roles directory>]
```

Table 57.71. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--roles-path &lt;roles directory&gt;</td>
<td>Filesystem path containing the role yaml files. By default this is <code>/usr/share/openstack-tripleo-heat-templates/roles</code></td>
</tr>
</tbody>
</table>

57.45. OVERCLOUD ROLE SHOW

Show information about a given role.

Usage:

```
openstack overcloud role show [-h] [--roles-path <roles directory>] <role>
```

Table 57.72. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role to display more information about.</td>
</tr>
</tbody>
</table>

Table 57.73. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--roles-path &lt;roles directory&gt;</td>
<td>Filesystem path containing the role yaml files. By default this is <code>/usr/share/openstack-tripleo-heat-templates/roles</code></td>
</tr>
</tbody>
</table>

57.46. OVERCLOUD ROLES GENERATE

Generate roles_data.yaml file

Usage:
openstack overcloud roles generate [-h]
  [--roles-path <roles directory>]
  [-o <output file>] [--skip-validate]
  <role> [<role> ...]

Table 57.74. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>List of roles to use to generate the roles_data.yaml file for the deployment. NOTE: Ordering is important if no role has the &quot;primary&quot; and &quot;controller&quot; tags. If no role is tagged then the first role listed will be considered the primary role. This usually is the controller role.</td>
</tr>
</tbody>
</table>

Table 57.75. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--roles-path &lt;roles directory&gt;</td>
<td>Filesystem path containing the role yaml files. by default this is /usr/share/openstack-tripleo-heat-templates/roles</td>
</tr>
<tr>
<td>-o &lt;output file&gt;, --output-file &lt;output file&gt;</td>
<td>File to capture all output to. for example, roles_data.yaml</td>
</tr>
<tr>
<td>--skip-validate</td>
<td>Skip role metadata type validation when generating the roles_data.yaml</td>
</tr>
</tbody>
</table>

57.47. OVERCLOUD STATUS

Get deployment status

Usage:

openstack overcloud status [-h] [--plan PLAN] [--working-dir WORKING_DIR]

Table 57.76. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--plan PLAN, --stack PLAN</td>
<td>Name of the stack/plan. (default: overcloud)</td>
</tr>
</tbody>
</table>
The working directory for the deployment where all input, output, and generated files are stored. Defaults to "$HOME/overcloud-deploy/<stack>".

57.48. OVERCLOUD SUPPORT REPORT COLLECT

Run sosreport on selected servers.

Usage:

```
```

Table 57.77. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_name</td>
<td>Server name, group name, or partial name to match. for example &quot;Controller&quot; will match all controllers for an environment.</td>
</tr>
</tbody>
</table>

Table 57.78. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Stack name to use for log collection.</td>
</tr>
<tr>
<td>-c CONTAINER, --container CONTAINER</td>
<td>This option no-longer has any effect.</td>
</tr>
<tr>
<td>-o DESTINATION, --output DESTINATION</td>
<td>Output directory for the report</td>
</tr>
<tr>
<td>--skip-container-delete</td>
<td>This option no-longer has any effect.</td>
</tr>
<tr>
<td>-t TIMEOUT, --timeout TIMEOUT</td>
<td>This option no-longer has any effect.</td>
</tr>
<tr>
<td>-n CONCURRENCY, --concurrency CONCURRENCY</td>
<td>This option no-longer has any effect.</td>
</tr>
</tbody>
</table>
57.49. OVERCLOUD UPDATE PREPARE

Use Heat to update and render the new Ansible playbooks based on the updated templates. These playbooks will be rendered and used during the update run step to perform the minor update of the overcloud nodes.

Usage:

```
openstack overcloud update prepare [--templates [TEMPLATES]]
[--stack STACK]
[--timeout <TIMEOUT>]
[--libvirt-type {kvm,qemu}]
[--ntp-server NTP_SERVER]
[--no-proxy NO_PROXY]
[--overcloud-ssh-user OVERCLOUD_SSH_USER]
[--overcloud-ssh-key OVERCLOUD_SSH_KEY]
[--overcloud-ssh-network OVERCLOUD_SSH_NETWORK]
[--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT]
[--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT]
[--environment-file <HEAT ENVIRONMENT FILE>]
[--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
[--roles-file ROLES_FILE]
[--networks-file NETWORKS_FILE]
[--vip-file VIP_FILE]
[--plan-environment-file PLAN_ENVIRONMENT_FILE]
[--no-cleanup] [--update-plan-only]
[--validation-errors-nonfatal]
[--validation-warnings-fatal]
[--disable-validations]
[--inflight-validations] [--dry-run]
[--run-validations]
[--skip-postconfig]
[--force-postconfig]
[--skip-deploy-identifier]
[--answers-file ANSWERS_FILE]
[--deployed-server]
[--config-download]
[--no-config-download]
[--config-download-only]
[--config-dir CONFIG_DIR]
[--config-type CONFIG_TYPE]
[--no-preserve-config]
[--output-dir OUTPUT_DIR]
[--override-ansible-cfg OVERRIDE_ANSIBLE_CFG]
[--config-download-timeout CONFIG_DOWNLOAD_TIMEOUT]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--collect-only</td>
<td>This option no-longer has any effect.</td>
</tr>
<tr>
<td>--download-only</td>
<td>This option no-longer has any effect.</td>
</tr>
</tbody>
</table>
Table 57.79. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Stack name to create or update</td>
</tr>
<tr>
<td>--timeout &lt;TIMEOUT&gt;, -t &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes</td>
</tr>
<tr>
<td>--libvirt-type {kvm,qemu}</td>
<td>Libvirt domain type.</td>
</tr>
<tr>
<td>--ntp-server NTP_SERVER</td>
<td>The ntp for overcloud nodes.</td>
</tr>
<tr>
<td>--no-proxy NO_PROXY</td>
<td>A comma separated list of hosts that should not be proxied.</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to overcloud nodes</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td>--overcloud-ssh-network OVERCLOUD_SSH_NETWORK</td>
<td>Network name to use for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td>--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT</td>
<td>This option no longer has any effect.</td>
</tr>
<tr>
<td>--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT</td>
<td>Timeout for the ssh port to become active.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--environment-file <code>&lt;HEAT ENVIRONMENT FILE&gt;</code>, -e <code>&lt;HEAT ENVIRONMENT FILE&gt;</code></td>
<td>Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--environment-directory <code>&lt;HEAT ENVIRONMENT DIRECTORY&gt;</code></td>
<td>Environment file directories that are automatically added to the heat stack-create or heat stack-update commands. Can be specified more than once. Files in directories are loaded in ascending sort order.</td>
</tr>
<tr>
<td>--roles-file <code>ROLES_FILE</code>, -r <code>ROLES_FILE</code></td>
<td>Roles file, overrides the default roles_data.yaml in the --templates directory. May be an absolute path or the path relative to --templates</td>
</tr>
<tr>
<td>--networks-file <code>NETWORKS_FILE</code>, -n <code>NETWORKS_FILE</code></td>
<td>Networks file, overrides the default network_data_default.yaml in the --templates directory</td>
</tr>
<tr>
<td>--vip-file <code>VIP_FILE</code></td>
<td>Configuration file describing the network virtual ips.</td>
</tr>
<tr>
<td>--no-cleanup</td>
<td>Don’t cleanup temporary files, just log their location</td>
</tr>
<tr>
<td>--update-plan-only</td>
<td>Deprecated: only update the plan. do not perform the actual deployment. NOTE: Will move to a discrete command in a future release. Not supported anymore.</td>
</tr>
<tr>
<td>--validation-errors-nonfatal</td>
<td>Allow the deployment to continue in spite of validation errors. Note that attempting deployment while errors exist is likely to fail.</td>
</tr>
<tr>
<td>--validation-warnings-fatal</td>
<td>Exit if there are warnings from the configuration pre-checks.</td>
</tr>
<tr>
<td>--disable-validations</td>
<td>Deprecated. disable the pre-deployment validations entirely. These validations are the built-in pre-deployment validations. To enable external validations from tripleo-validations, use the --run-validations flag. These validations are now run via the external validations in tripleo-validations.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--inflight-validations</td>
<td>Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Only run validations, but do not apply any changes.</td>
</tr>
<tr>
<td>--run-validations</td>
<td>Run external validations from the tripleo-validations project.</td>
</tr>
<tr>
<td>--skip-postconfig</td>
<td>Skip the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--force-postconfig</td>
<td>Force the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--skip-deploy-identifier</td>
<td>Skip generation of a unique identifier for the DeployIdentifier parameter. The software configuration deployment steps will only be triggered if there is an actual change to the configuration. This option should be used with Caution, and only if there is confidence that the software configuration does not need to be run, such as when scaling out certain roles.</td>
</tr>
<tr>
<td>--answers-file ANSWERS_FILE</td>
<td>Path to a yaml file with arguments and parameters.</td>
</tr>
<tr>
<td>--disable-password-generation</td>
<td>Disable password generation.</td>
</tr>
<tr>
<td>--deployed-server</td>
<td>Use pre-provisioned overcloud nodes. Removes baremetal, compute and image services requirements from the undercloud node. Must only be used with the --disable-validations.</td>
</tr>
<tr>
<td>--config-download</td>
<td>Deprecated: run deployment via config-download mechanism. This is now the default, and this CLI options has no effect.</td>
</tr>
<tr>
<td>--no-config-download, --stack-only</td>
<td>Disable the config-download workflow and only create the stack and download the config. No software configuration, setup, or any changes will be applied to overcloud nodes.</td>
</tr>
<tr>
<td>--config-download-only</td>
<td>Disable the stack create and setup, and only run the config-download workflow to apply the software configuration. Requires that config-download setup was previously completed, either with --stack-only and --setup-only or a full deployment</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--setup-only</code></td>
<td>Disable the stack and config-download workflow to apply the software configuration and only run the setup to enable ssh connectivity.</td>
</tr>
<tr>
<td><code>--config-dir CONFIG_DIR</code></td>
<td>The directory where the configuration files will be pushed.</td>
</tr>
<tr>
<td><code>--config-type CONFIG_TYPE</code></td>
<td>Only used when <code>--setup-only</code> is invoked. Type of object config to be extract from the deployment, defaults to all keys available.</td>
</tr>
<tr>
<td><code>--no-preserve-config</code></td>
<td>Only used when <code>--setup-only</code> is invoked. If specified, will delete and recreate the <code>--config-dir</code> if it already exists. Default is to use the existing dir location and overwrite files. Files in <code>--config-dir</code> not from the stack will be preserved by default.</td>
</tr>
<tr>
<td><code>--output-dir OUTPUT_DIR</code></td>
<td>Directory to use for saved output when using <code>--config-download</code>. When not specified, <code>&lt;working-dir&gt;/config-download</code> will be used.</td>
</tr>
<tr>
<td><code>--override-ansible-cfg OVERRIDE_ANSIBLE_CFG</code></td>
<td>Path to ansible configuration file. The configuration in the file will override any configuration used by config-download by default.</td>
</tr>
<tr>
<td><code>--config-download-timeout CONFIG_DOWNLOAD_TIMEOUT</code></td>
<td>Timeout (in minutes) to use for config-download steps. If unset, will default to however much time is leftover from the <code>--timeout</code> parameter after the stack operation.</td>
</tr>
<tr>
<td><code>--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER</code></td>
<td>The path to python interpreter to use for the deployment actions. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td><code>-b [&lt;baremetal_deployment.yaml&gt;], --baremetal-deployment [&lt;baremetal_deployment.yaml&gt;]</code></td>
<td>Deploy baremetal nodes, network and virtual ip addresses as defined in baremetal_deployment.yaml along with overcloud. If no baremetal_deployment YAML file is given, the tripleo-stack_name-baremetal-deployment.yaml file in the working-dir will be used.</td>
</tr>
<tr>
<td><code>--network-config</code></td>
<td>Apply network config to provisioned nodes. (implies <code>--network-ports</code>).</td>
</tr>
<tr>
<td><code>--limit LIMIT</code></td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: <code>--limit &quot;compute-0,compute-1,compute-5&quot;</code>.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A list of tags to use when running the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A list of tags to skip when running the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td>--heat-type {pod,container,native}</td>
<td>The type of heat process to use to execute the deployment. pod (Default): Use an ephemeral Heat pod. container (Experimental): Use an ephemeral Heat container. native (Experimental): Use an ephemeral Heat process.</td>
</tr>
<tr>
<td>--heat-container-api-image &lt;HEAT_CONTAINER_API_IMAGE&gt;</td>
<td>The container image to use when launching the heat-api process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-api:ephemeral</td>
</tr>
<tr>
<td>--heat-container-engine-image &lt;HEAT_CONTAINER_ENGINE_IMAGE&gt;</td>
<td>The container image to use when launching the heat-engine process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-engine:ephemeral</td>
</tr>
<tr>
<td>--rm-heat</td>
<td>If specified and --heat-type is container or pod any existing container or pod of a previous ephemeral Heat process will be deleted first. Ignored if --heat-type is native.</td>
</tr>
<tr>
<td>--skip-heat-pull</td>
<td>When --heat-type is pod or container, assume the container image has already been pulled</td>
</tr>
<tr>
<td>--disable-protected-resource-types</td>
<td>Disable protected resource type overrides. resources types that are used internally are protected, and cannot be overridden in the user environment. Setting this argument disables the protection, allowing the protected resource types to be override in the user environment.</td>
</tr>
</tbody>
</table>
57.50. OVERCLOUD UPDATE RUN

Run minor update ansible playbooks on Overcloud nodes

Usage:

openstack overcloud update run [-h] --limit LIMIT
  [--playbook [PLAYBOOK ...]]
  [--ssh-user SSH_USER]
  [--static-inventory STATIC_INVENTORY]
  [--stack STACK] [--tags TAGS]
  [--skip-tags SKIP_TAGS] [-y]
  [--ansible-forks ANSIBLE_FORKS]

Table 57.80. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: --limit &quot;compute-0,compute-1,compute-5&quot;.</td>
</tr>
<tr>
<td>--playbook [PLAYBOOK ...]</td>
<td>Ansible playbook to use for the minor update. can be used multiple times. Set this to each of those playbooks in consecutive invocations of this command if you prefer to run them manually. Note: make sure to run all playbooks so that all services are updated and running with the target version configuration.</td>
</tr>
<tr>
<td>--ssh-user SSH_USER</td>
<td>Deprecated: only tripleo-admin should be used as ssh user.</td>
</tr>
<tr>
<td>--static-inventory STATIC_INVENTORY</td>
<td>Deprecated: tripleo-ansible-inventory.yaml in working dir will be used.</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A list of tags to use when running the <code>config-download ansible-playbook</code> command.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A list of tags to skip when running the <code>config-download ansible-playbook</code> command.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Use <code>-y</code> or <code>--yes</code> to skip the confirmation required before any update operation. Use this with caution!</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the <code>config-download ansible-playbook</code> command.</td>
</tr>
</tbody>
</table>

## 57.51. OVERCLOUD UPGRADE CONVERGE

Major upgrade converge - reset Heat resources in the stored plan This is the last step for completion of a overcloud major upgrade. The main task is updating the plan and stack to unblock future stack updates. For the major upgrade workflow we have set specific values for some stack Heat resources. This unsets those back to their default values.

**Usage:**

```bash
openstack overcloud upgrade converge [--templates [TEMPLATES]]
  [--stack STACK]
  [--timeout <TIMEOUT>]
  [--libvirt-type {kvm,qemu}]
  [--ntp-server NTP_SERVER]
  [--no-proxy NO_PROXY]
  [--overcloud-ssh-user OVERCLOUD_SSH_USER]
  [--overcloud-ssh-key OVERCLOUD_SSH_KEY]
  [--overcloud-ssh-network OVERCLOUD_SSH_NETWORK]
  [--overcloud-ssh-enable-timeout]
  [--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT]
  [--environment-file <HEAT ENVIRONMENT_FILE>]
  [--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
  [--roles-file ROLES_FILE]
  [--networks-file NETWORKS_FILE]
  [--vip-file VIP_FILE]
  [--plan-environment-file PLAN_ENVIRONMENT_FILE]
  [--no-cleanup]
  [--update-plan-only]
  [--validation-errors-nonfatal]
  [--validation-warnings-fatal]
  [--disable-validations]
  [--inflight-validations]
  [--dry-run] [--run-validations]
  [--skip-postconfig]
  [--force-postconfig]
  [--skip-deploy-identifier]
  [--answers-file ANSWERS_FILE]
```
Table 57.81. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Stack name to create or update</td>
</tr>
<tr>
<td>--timeout &lt;TIMEOUT&gt;, -t &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes.</td>
</tr>
<tr>
<td>--libvirt-type {kvm,qemu}</td>
<td>Libvirt domain type.</td>
</tr>
<tr>
<td>--ntp-server NTP_SERVER</td>
<td>The ntp for overcloud nodes.</td>
</tr>
<tr>
<td>--no-proxy NO_PROXY</td>
<td>A comma separated list of hosts that should not be proxied.</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to overcloud nodes</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td></td>
<td>When undefined the key will be autodetected.</td>
</tr>
</tbody>
</table>
### Overcloud SSH Network

**Value**

```
--overcloud-ssh-network
OVERCLOUD_SSH_NETWORK
```

**Summary**

Network name to use for ssh access to overcloud nodes.

---

**Value**

```
--overcloud-ssh-enable-timeout
OVERCLOUD_SSH_ENABLE_TIMEOUT
```

**Summary**

This option no longer has any effect.

---

**Value**

```
--overcloud-ssh-port-timeout
OVERCLOUD_SSH_PORT_TIMEOUT
```

**Summary**

Timeout for the ssh port to become active.

---

**Value**

```
--environment-file <HEAT ENVIRONMENT FILE>, -e
<HEAT ENVIRONMENT FILE>
```

**Summary**

Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)

---

**Value**

```
--environment-directory <HEAT ENVIRONMENT DIRECTORY>
```

**Summary**

Environment file directories that are automatically added to the heat stack-create or heat stack-update commands. Can be specified more than once. Files in directories are loaded in ascending sort order.

---

**Value**

```
--roles-file ROLES_FILE, -r ROLES_FILE
```

**Summary**

Roles file, overrides the default roles_data.yaml in the --templates directory. May be an absolute path or the path relative to --templates

---

**Value**

```
--networks-file NETWORKS_FILE, -n NETWORKS_FILE
```

**Summary**

Networks file, overrides the default network_data_default.yaml in the --templates directory

---

**Value**

```
--vip-file VIP_FILE
```

**Summary**

Configuration file describing the network virtual ips.

---

**Value**

```
--plan-environment-file
PLAN_ENVIRONMENT_FILE, -p
PLAN_ENVIRONMENT_FILE
```

**Summary**

Plan environment file for derived parameters.

---

**Value**

```
--no-cleanup
```

**Summary**

Don’t cleanup temporary files, just log their location

---

**Value**

```
--update-plan-only
```

**Summary**

Deprecated: only update the plan. do not perform the actual deployment. NOTE: Will move to a discrete command in a future release. Not supported anymore.

---

**Value**

```
--validation-errors-nonfatal
```

**Summary**

Allow the deployment to continue in spite of validation errors. Note that attempting deployment while errors exist is likely to fail.

---

**Value**

```
--validation-warnings-fatal
```

**Summary**

Exit if there are warnings from the configuration pre-checks.
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--disable-validations</td>
<td>Deprecated. disable the pre-deployment validations entirely. These validations are the built-in pre-deployment validations. To enable external validations from tripleo-validations, use the --run-validations flag. These validations are now run via the external validations in tripleo-validations.</td>
</tr>
<tr>
<td>--inflight-validations</td>
<td>Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Only run validations, but do not apply any changes.</td>
</tr>
<tr>
<td>--run-validations</td>
<td>Run external validations from the tripleo-validations project.</td>
</tr>
<tr>
<td>--skip-postconfig</td>
<td>Skip the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--force-postconfig</td>
<td>Force the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td>--skip-deploy-identifier</td>
<td>Skip generation of a unique identifier for the DeployIdentifier parameter. The software configuration deployment steps will only be triggered if there is an actual change to the configuration. This option should be used with Caution, and only if there is confidence that the software configuration does not need to be run, such as when scaling out certain roles.</td>
</tr>
<tr>
<td>--answers-file ANSWERS_FILE</td>
<td>Path to a yaml file with arguments and parameters.</td>
</tr>
<tr>
<td>--disable-password-generation</td>
<td>Disable password generation.</td>
</tr>
<tr>
<td>--deployed-server</td>
<td>Use pre-provisioned overcloud nodes. Removes baremetal,compute and image services requirements from the undercloud node. Must only be used with the --disable-validations.</td>
</tr>
<tr>
<td>--config-download</td>
<td>Deprecated: run deployment via config-download mechanism. This is now the default, and this CLI options has no effect.</td>
</tr>
<tr>
<td>--no-config-download, --stack-only</td>
<td>Disable the config-download workflow and only create the stack and download the config. No software configuration, setup, or any changes will be applied to overcloud nodes.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--config-download-only</td>
<td>Disable the stack create and setup, and only run the config-download workflow to apply the software configuration. Requires that config-download setup was previously completed, either with --stack-only and --setup-only or a full deployment</td>
</tr>
<tr>
<td>--setup-only</td>
<td>Disable the stack and config-download workflow to apply the software configuration and only run the setup to enable ssh connectivity.</td>
</tr>
<tr>
<td>--config-dir CONFIG_DIR</td>
<td>The directory where the configuration files will be pushed</td>
</tr>
<tr>
<td>--config-type CONFIG_TYPE</td>
<td>Only used when &quot;--setup-only&quot; is invoked. type of object config to be extract from the deployment, defaults to all keys available</td>
</tr>
<tr>
<td>--no-preserve-config</td>
<td>Only used when &quot;--setup-only&quot; is invoked. if specified, will delete and recreate the --config-dir if it already exists. Default is to use the existing dir location and overwrite files. Files in --config-dir not from the stack will be preserved by default.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to use for saved output when using --config-download. When not specified, &lt;working-dir&gt;/config-download will be used.</td>
</tr>
<tr>
<td>--override-ansible-cfg OVERRIDE_ANSIBLE_CFG</td>
<td>Path to ansible configuration file. the configuration in the file will override any configuration used by config-download by default.</td>
</tr>
<tr>
<td>--config-download-timeout CONFIG_DOWNLOAD_TIMEOUT</td>
<td>Timeout (in minutes) to use for config-download steps. If unset, will default to however much time is leftover from the --timeout parameter after the stack operation.</td>
</tr>
<tr>
<td>--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER</td>
<td>The path to python interpreter to use for the deployment actions. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td>-b [&lt;baremetal_deployment.yaml&gt;], --baremetal-deployment [&lt;baremetal_deployment.yaml&gt;]</td>
<td>Deploy baremetal nodes, network and virtual ip addresses as defined in baremetal_deployment.yaml along with overcloud. If no baremetal_deployment YAML file is given, the tripleo-&lt;stack_name&gt;-baremetal-deployment.yaml file in the working-dir will be used.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--network-config</td>
<td>Apply network config to provisioned nodes. (implies &quot;--network-ports&quot;)</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: <code>--limit &quot;compute-0,compute-1,compute-5&quot;</code>.</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A list of tags to use when running the the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A list of tags to skip when running the the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td>--heat-type {pod,container,native}</td>
<td>The type of heat process to use to execute the deployment. pod (Default): Use an ephemeral Heat pod. container (Experimental): Use an ephemeral Heat container. native (Experimental): Use an ephemeral Heat process.</td>
</tr>
<tr>
<td>--heat-container-api-image</td>
<td>The container image to use when launching the heat-api process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-api:ephemeral</td>
</tr>
<tr>
<td>--heat-container-engine-image</td>
<td>The container image to use when launching the heat-engine process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-engine:ephemeral</td>
</tr>
<tr>
<td>--rm-heat</td>
<td>If specified and --heat-type is container or pod any existing container or pod of a previous ephemeral Heat process will be deleted first. Ignored if --heat-type is native.</td>
</tr>
</tbody>
</table>
When --heat-type is pod or container, assume the container image has already been pulled

Disable protected resource type overrides. Resources types that are used internally are protected, and cannot be overridden in the user environment. Setting this argument disables the protection, allowing the protected resource types to be override in the user environment.

Use -y or --yes to skip any confirmation required before the deploy operation. Use this with caution!

Set this to allow using deprecated network data yaml definition schema.

57.52. OVERCLOUD UPGRADE PREPARE

Run heat stack update for overcloud nodes to refresh heat stack outputs. The heat stack outputs are what we use later on to generate ansible playbooks which deliver the major upgrade workflow. This is used as the first step for a major upgrade of your overcloud.

Usage:

openstack overcloud upgrade prepare [--templates [TEMPLATES]]
  [-stack STACK]
  [--timeout <TIMEOUT>]
  [--libvirt-type {kvm,qemu}]
  [--ntp-server NTP_SERVER]
  [--no-proxy NO_PROXY]
  [--overcloud-ssh-user OVERCLOUD_SSH_USER]
  [--overcloud-ssh-key OVERCLOUD_SSH_KEY]
  [--overcloud-ssh-network OVERCLOUD_SSH_NETWORK]
  [--overcloud-ssh-enable-timeout OVERCLOUD_SSH_ENABLE_TIMEOUT]
  [--overcloud-ssh-port-timeout OVERCLOUD_SSH_PORT_TIMEOUT]
  [--environment-file <HEAT ENVIRONMENT FILE>]
  [--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
  [--roles-file ROLES_FILE]
  [--networks-file NETWORKS_FILE]
  [--vip-file VIP_FILE]
  [--plan-environment-file PLAN_ENVIRONMENT_FILE]
  [--no-cleanup] [--update-plan-only]
  [--validation-errors-nonfatal]
  [--validation-warnings-fatal]
  [--disable-validations]
  [--inflight-validations]
  [--dry-run] [--run-validations]
  [--skip-postconfig]
  [--force-postconfig]
Table 57.82. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Stack name to create or update</td>
</tr>
<tr>
<td>--timeout &lt;TIMEOUT&gt;, -t &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes.</td>
</tr>
<tr>
<td>--libvirt-type {kvm,qemu}</td>
<td>Libvirt domain type.</td>
</tr>
<tr>
<td>--ntp-server NTP_SERVER</td>
<td>The ntp for overcloud nodes.</td>
</tr>
<tr>
<td>--no-proxy NO_PROXY</td>
<td>A comma separated list of hosts that should not be proxied.</td>
</tr>
<tr>
<td>--overcloud-ssh-user OVERCLOUD_SSH_USER</td>
<td>User for ssh access to overcloud nodes</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--overcloud-ssh-key OVERCLOUD_SSH_KEY</td>
<td>Key path for ssh access to overcloud nodes. When undefined the key will be autodetected.</td>
</tr>
<tr>
<td>--overcloud-ssh-network</td>
<td>Network name to use for ssh access to overcloud nodes.</td>
</tr>
<tr>
<td>--overcloud-ssh-enable-timeout</td>
<td>This option no longer has any effect.</td>
</tr>
<tr>
<td>--overcloud-ssh-port-timeout</td>
<td>Timeout for the ssh port to become active.</td>
</tr>
<tr>
<td>--environment-file &lt;HEAT ENVIRONMENT FILE&gt;, -e &lt;HEAT ENVIRONMENT FILE&gt;</td>
<td>Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--environment-directory &lt;HEAT ENVIRONMENT DIRECTORY&gt;</td>
<td>Environment file directories that are automatically added to the heat stack-create or heat stack-update commands. Can be specified more than once. Files in directories are loaded in ascending sort order.</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data.yaml in the --templates directory. May be an absolute path or the path relative to --templates</td>
</tr>
<tr>
<td>--networks-file NETWORKS_FILE, -n NETWORKS_FILE</td>
<td>Networks file, overrides the default network_data_default.yaml in the --templates directory</td>
</tr>
<tr>
<td>--vip-file VIP_FILE</td>
<td>Configuration file describing the network virtual ips.</td>
</tr>
<tr>
<td>--no-cleanup</td>
<td>Don’t cleanup temporary files, just log their location</td>
</tr>
<tr>
<td>--update-plan-only</td>
<td>Deprecated: only update the plan. do not perform the actual deployment. NOTE: Will move to a discrete command in a future release. Not supported anymore.</td>
</tr>
<tr>
<td>--validation-errors-nonfatal</td>
<td>Allow the deployment to continue in spite of validation errors. Note that attempting deployment while errors exist is likely to fail.</td>
</tr>
<tr>
<td>--validation-warnings-fatal</td>
<td>Exit if there are warnings from the configuration pre-checks.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--disable-validations</code></td>
<td>Deprecated. disable the pre–deployment validations entirely. These validations are the built-in pre–deployment validations. To enable external validations from tripleo–validations, use the <code>--run-validations</code> flag. These validations are now run via the external validations in tripleo–validations.</td>
</tr>
<tr>
<td><code>--inflight-validations</code></td>
<td>Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td><code>--dry-run</code></td>
<td>Only run validations, but do not apply any changes.</td>
</tr>
<tr>
<td><code>--run-validations</code></td>
<td>Run external validations from the tripleo–validations project.</td>
</tr>
<tr>
<td><code>--skip-postconfig</code></td>
<td>Skip the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td><code>--force-postconfig</code></td>
<td>Force the overcloud post-deployment configuration.</td>
</tr>
<tr>
<td><code>--skip-deploy-identifier</code></td>
<td>Skip generation of a unique identifier for the DeployIdentifier parameter. The software configuration deployment steps will only be triggered if there is an actual change to the configuration. This option should be used with Caution, and only if there is confidence that the software configuration does not need to be run, such as when scaling out certain roles.</td>
</tr>
<tr>
<td><code>--answers-file ANSWERS_FILE</code></td>
<td>Path to a yaml file with arguments and parameters.</td>
</tr>
<tr>
<td><code>--disable-password-generation</code></td>
<td>Disable password generation.</td>
</tr>
<tr>
<td><code>--deployed-server</code></td>
<td>Use pre-provisioned overcloud nodes. Removes baremetal, compute and image services requirements from the undercloud node. Must only be used with the <code>--disable-validations</code> option.</td>
</tr>
<tr>
<td><code>--config-download</code></td>
<td>Deprecated: run deployment via config-download mechanism. This is now the default, and this CLI options has no effect.</td>
</tr>
<tr>
<td><code>--no-config-download, --stack-only</code></td>
<td>Disable the config-download workflow and only create the stack and download the config. No software configuration, setup, or any changes will be applied to overcloud nodes.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--config-download-only</td>
<td>Disable the stack create and setup, and only run the config-download workflow to apply the software configuration. Requires that config-download setup was previously completed, either with --stack-only and --setup-only or a full deployment</td>
</tr>
<tr>
<td>--setup-only</td>
<td>Disable the stack and config-download workflow to apply the software configuration and only run the setup to enable ssh connectivity.</td>
</tr>
<tr>
<td>--config-dir CONFIG_DIR</td>
<td>The directory where the configuration files will be pushed</td>
</tr>
<tr>
<td>--config-type CONFIG_TYPE</td>
<td>Only used when &quot;--setup-only&quot; is invoked. type of object config to be extract from the deployment, defaults to all keys available</td>
</tr>
<tr>
<td>--no-preserve-config</td>
<td>Only used when &quot;--setup-only&quot; is invoked. if specified, will delete and recreate the --config-dir if it already exists. Default is to use the existing dir location and overwrite files. Files in --config-dir not from the stack will be preserved by default.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to use for saved output when using --config-download. When not specified, &lt;working-dir&gt;/config-download will be used.</td>
</tr>
<tr>
<td>--override-ansible-cfg OVERRIDE_ANSIBLE_CFG</td>
<td>Path to ansible configuration file. the configuration in the file will override any configuration used by config-download by default.</td>
</tr>
<tr>
<td>--config-download-timeout</td>
<td>Timeout (in minutes) to use for config-download steps. If unset, will default to however much time is leftover from the --timeout parameter after the stack operation.</td>
</tr>
<tr>
<td>--deployment-python-interpreter</td>
<td>The path to python interpreter to use for the deployment actions. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td>-b [baremetal_deployment.yaml], --baremetal-deployment [baremetal_deployment.yaml]</td>
<td>Deploy baremetal nodes, network and virtual ip addresses as defined in baremetal_deployment.yaml along with overcloud. If no baremetal_deployment YAML file is given, the tripleo-&lt;stack_name&gt;-baremetal-deployment.yaml file in the working-dir will be used.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--network-config</td>
<td>Apply network config to provisioned nodes. (implies &quot;--network-ports&quot;)</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: --limit &quot;compute-0,compute-1,compute-5&quot;.</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A list of tags to use when running the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A list of tags to skip when running the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--working-dir WORKING_DIR</td>
<td>The working directory for the deployment where all input, output, and generated files will be stored. Defaults to &quot;$HOME/overcloud-deploy/&lt;stack&gt;&quot;</td>
</tr>
<tr>
<td>--heat-type {pod,container,native}</td>
<td>The type of heat process to use to execute the deployment. pod (Default): Use an ephemeral Heat pod. container (Experimental): Use an ephemeral Heat container. native (Experimental): Use an ephemeral Heat process.</td>
</tr>
<tr>
<td>--heat-container-api-image &lt;HEAT_CONTAINER_API_IMAGE&gt;</td>
<td>The container image to use when launching the heat-api process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-api:ephemeral</td>
</tr>
<tr>
<td>--heat-container-engine-image &lt;HEAT_CONTAINER_ENGINE_IMAGE&gt;</td>
<td>The container image to use when launching the heat-engine process. Only used when --heat-type=pod. Defaults to: localhost/tripleo/openstack-heat-engine:ephemeral</td>
</tr>
<tr>
<td>--rm-heat</td>
<td>If specified and --heat-type is container or pod any existing container or pod of a previous ephemeral Heat process will be deleted first. Ignored if --heat-type is native.</td>
</tr>
</tbody>
</table>
When `--heat-type` is pod or container, assume the container image has already been pulled.

Disable protected resource type overrides. Resources types that are used internally are protected, and cannot be overridden in the user environment. Setting this argument disables the protection, allowing the protected resource types to be override in the user environment.

Use `-y` or `--yes` to skip any confirmation required before the deploy operation. Use this with caution!

Set this to allow using deprecated network data YAML definition schema.

57.53. OVERCLOUD UPGRADE RUN

Run major upgrade ansible playbooks on Overcloud nodes. This will run the major upgrade ansible playbooks on the overcloud. By default all playbooks are executed, that is the `upgrade_steps_playbook.yaml` then the `deploy_steps_playbook.yaml` and then the `post_upgrade_steps_playbook.yaml`. The upgrade playbooks are made available after completion of the `overcloud upgrade prepare` command. This `overcloud upgrade run` command is the second step in the major upgrade workflow.

Usage:

```bash
openstack overcloud upgrade run [-h] --limit LIMIT
   [-playbook [PLAYBOOK ...]]
   [--static-inventory STATIC_INVENTORY]
   [--ssh-user SSH_USER] [--tags TAGS]
   [--skip-tags SKIP_TAGS] [--stack STACK]
   [-y] [--ansible-forks ANSIBLE_FORKS]
```

Table 57.83. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--limit LIMIT</code></td>
<td>A string that identifies a single node or comma-separated list of nodes the config-download Ansible playbook execution will be limited to. For example: <code>--limit &quot;compute-0,compute-1,compute-5&quot;</code>.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--playbook [PLAYBOOK ...]</td>
<td>Ansible playbook to use for the minor update. can be used multiple times. Set this to each of those playbooks in consecutive invocations of this command if you prefer to run them manually. Note: make sure to run all playbooks so that all services are updated and running with the target version configuration.</td>
</tr>
<tr>
<td>--static-inventory STATIC_INVENTORY</td>
<td>Deprecated: tripleo-ansible-inventory.yaml in working dir will be used.</td>
</tr>
<tr>
<td>--ssh-user SSH_USER</td>
<td>Deprecated: only tripleo-admin should be used as ssh user.</td>
</tr>
<tr>
<td>--tags TAGS</td>
<td>A string specifying the tag or comma separated list of tags to be passed as --tags to ansible-playbook.</td>
</tr>
<tr>
<td>--skip-tags SKIP_TAGS</td>
<td>A string specifying the tag or comma separated list of tags to be passed as --skip-tags to ansible-playbook. The currently supported values are validation and pre-upgrade. In particular validation is useful if you must re-run following a failed upgrade and some services cannot be started.</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name or id of heat stack (default=env: OVERCLOUD_STACK_NAME)</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Use -y or --yes to skip the confirmation required before any upgrade operation. Use this with caution!</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
</tbody>
</table>
CHAPTER 58. POLICY

This chapter describes the commands under the `policy` command.

58.1. POLICY CREATE

Create new policy

**Usage:**

```
```

**Table 58.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;filename&gt;</td>
<td>New serialized policy rules file</td>
</tr>
</tbody>
</table>

**Table 58.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>New mime type of the policy rules file (defaults to application/json)</td>
</tr>
</tbody>
</table>

**Table 58.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 58.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 58.5. Shell formatter options**
Table 58.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

58.2. POLICY DELETE

Delete policy(s)

Usage:

```
openstack policy delete [-h] <policy> [<policy> ...]
```

Table 58.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>Policy(s) to delete</td>
</tr>
</tbody>
</table>

Table 58.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

58.3. POLICY LIST

List policies

Usage:

```
openstack policy list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [-quote {all, minimal, none, nonnumeric}] [-noindent] [-max-width <integer>]
```
Table 58.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 58.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 58.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 58.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 58.13. Table formatter options
58.4. POLICY SET
Set policy properties

Usage:

```
openstack policy set [-h] [--type <type>] [--rules <filename>] <policy>
```

Table 58.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>Policy to modify</td>
</tr>
</tbody>
</table>

Table 58.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>New mime type of the policy rules file</td>
</tr>
<tr>
<td>--rules &lt;filename&gt;</td>
<td>New serialized policy rules file</td>
</tr>
</tbody>
</table>

58.5. POLICY SHOW
Display policy details

Usage:

```
openstack policy show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [-max-width <integer>] [-fit-width]
```
### Table 58.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;policy&gt;</td>
<td>Policy to display</td>
</tr>
</tbody>
</table>

### Table 58.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 58.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 58.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 58.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 58.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 59. PORT

This chapter describes the commands under the `port` command.

59.1. PORT CREATE

Create a new port

Usage:

```bash
```

Table 59.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of this port</td>
</tr>
</tbody>
</table>

Table 59.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>Network this port belongs to (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of this port</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--device &lt;device-id&gt;</code></td>
<td>Port device id</td>
</tr>
<tr>
<td><code>--mac-address &lt;mac-address&gt;</code></td>
<td>Mac address of this port (admin only)</td>
</tr>
<tr>
<td><code>--device-owner &lt;device-owner&gt;</code></td>
<td>Device owner of this port. this is the entity that uses the port (for example, network:dhcp).</td>
</tr>
<tr>
<td><code>--vnic-type &lt;vnic-type&gt;</code></td>
<td>Vnic type for this port (direct</td>
</tr>
<tr>
<td><code>--host &lt;host-id&gt;</code></td>
<td>Allocate port on host &lt;host-id&gt; (id only)</td>
</tr>
<tr>
<td><code>--dns-domain dns-domain</code></td>
<td>Set dns domain to this port (requires dns_domain extension for ports)</td>
</tr>
<tr>
<td><code>--dns-name &lt;dns-name&gt;</code></td>
<td>Set dns name for this port (requires dns integration extension)</td>
</tr>
<tr>
<td><code>--numa-policy-required</code></td>
<td>Numa affinity policy required to schedule this port</td>
</tr>
<tr>
<td><code>--numa-policy-preferred</code></td>
<td>Numa affinity policy preferred to schedule this port</td>
</tr>
<tr>
<td><code>--numa-policy-legacy</code></td>
<td>Numa affinity policy using legacy mode to schedule this port</td>
</tr>
<tr>
<td><code>--fixed-ip subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt;</code></td>
<td>Desired ip and/or subnet for this port (name or id): subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt; (repeat option to set multiple fixed IP addresses)</td>
</tr>
<tr>
<td><code>--no-fixed-ip</code></td>
<td>No ip or subnet for this port.</td>
</tr>
<tr>
<td><code>--binding-profile &lt;binding-profile&gt;</code></td>
<td>Custom data to be passed as binding:profile. data may be passed as &lt;key&gt;=&lt;value&gt; or JSON. (repeat option to set multiple binding:profile data)</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable port (default)</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable port</td>
</tr>
<tr>
<td><code>--enable-uplink-status-propagation</code></td>
<td>Enable uplink status propagate</td>
</tr>
<tr>
<td><code>--disable-uplink-status-propagation</code></td>
<td>Disable uplink status propagate (default)</td>
</tr>
<tr>
<td><code>--project &lt;project&gt;</code></td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--extra-dhcp-option name=&lt;name&gt;[,value=&lt;value&gt;,ip-version={4,6}]</td>
<td>Extra dhcp options to be assigned to this port: name=&lt;name&gt;[,value=&lt;value&gt;,ip-version={4,6}] (repeat option to set multiple extra DHCP options)</td>
</tr>
<tr>
<td>--security-group &lt;security-group&gt;</td>
<td>Security group to associate with this port (name or ID) (repeat option to set multiple security groups)</td>
</tr>
<tr>
<td>--no-security-group</td>
<td>Associate no security groups with this port</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
<td>Attach qos policy to this port (name or id)</td>
</tr>
<tr>
<td>--enable-port-security</td>
<td>Enable port security for this port (default)</td>
</tr>
<tr>
<td>--disable-port-security</td>
<td>Disable port security for this port</td>
</tr>
<tr>
<td>--allowed-address ip-address=&lt;ip-address&gt;[,mac-address=&lt;mac-address&gt;]</td>
<td>Add allowed-address pair associated with this port: ip-address=&lt;ip-address&gt;[,mac-address=&lt;mac-address&gt;] (repeat option to set multiple allowed-address pairs)</td>
</tr>
<tr>
<td>--device-profile &lt;device-profile&gt;</td>
<td>Cyborg port device profile</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the port (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the port</td>
</tr>
</tbody>
</table>

Table 59.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 59.4. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 59.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 59.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

59.2. PORT DELETE

Delete port(s)

Usage:

```
openstack port delete [-h] <port> [<port> ...]
```

Table 59.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Port(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 59.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

59.3. PORT LIST
List ports

Usage:

```
openstack port list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
   [--quote {all,minimal,none,nonnumeric}] [-noindent] [-max-width <integer>]
   [--fit-width] [--sort-column SORT_COLUMN]
   [--sort-ascending] [--sort-descending] [--device-owner <device-owner>]
   [-host <host-id>] [--network <network>]
   [--router <router> | --server <server> | --device-id <device-id>]
   [--mac-address <mac-address>] [-long]
   [--project <project>] [--name <name>]
   [--project-domain <project-domain>]
   [--fixed-ip subnet=<subnet>,ip-address=<ip-address>,ip-substring=<ip-substring>]
   [--tags <tag>[,<tag>,....]]
   [--any-tags <tag>[,<tag>,....]]
   [--not-tags <tag>[,<tag>,....]]
   [--not-any-tags <tag>[,<tag>,....]]
```

Table 59.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--device-owner &lt;device-owner&gt;</td>
<td>List only ports with the specified device owner. this is the entity that uses the port (for example, network:dhcp).</td>
</tr>
<tr>
<td>--host &lt;host-id&gt;</td>
<td>List only ports bound to this host id</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>List only ports connected to this network (name or id)</td>
</tr>
<tr>
<td>--router &lt;router&gt;</td>
<td>List only ports attached to this router (name or id)</td>
</tr>
<tr>
<td>--server &lt;server&gt;</td>
<td>List only ports attached to this server (name or id)</td>
</tr>
<tr>
<td>--device-id &lt;device-id&gt;</td>
<td>List only ports with the specified device id</td>
</tr>
<tr>
<td>--mac-address &lt;mac-address&gt;</td>
<td>List only ports with this mac address</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List ports according to their project (name or id)</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List ports according to their name</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--fixed-ip subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt;,ip-substring=&lt;ip-substring&gt;</td>
<td>Desired IP and/or subnet for filtering ports (name or ID): subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt;,ip-substring=&lt;ip-substring&gt; (repeat option to set multiple fixed IP addresses)</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List ports which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List ports which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude ports which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude ports which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 59.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 59.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
### Table 59.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 59.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 59.4. PORT SET

Set port properties

**Usage:**

```
openstack port set [-h] [--description <description>]  
  [--device <device-id>] [--mac-address <mac-address>]  
  [--device-owner <device-owner>]  
  [-vnic-type <vnic-type>] [--host <host-id>]  
  [-dns-domain dns-domain] [-dns-name <dns-name>]  
  [--numa-policy-required | --numa-policy-preferred | --numa-policy-legacy]  
  [--enable | --disable] [--name <name>]  
  [--fixed-ip subnet=<subnet>,ip-address=<ip-address>]  
  [--no-fixed-ip]  
  [--binding-profile <binding-profile>]  
  [--no-binding-profile] [--qos-policy <qos-policy>]  
  [--security-group <security-group>]  
  [--no-security-group]  
  [--enable-port-security | --disable-port-security]  
  [--allowed-address ip-address=<ip-address>[mac-address=<mac-address>]]  
  [--no-allowed-address]  
  [--data-plane-status <status>] [--tag <tag>]  
  [--no-tag]  
  <port>
```

### Table 59.14. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Port to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 59.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of this port</td>
</tr>
<tr>
<td>--device &lt;device-id&gt;</td>
<td>Port device id</td>
</tr>
<tr>
<td>--mac-address &lt;mac-address&gt;</td>
<td>Mac address of this port (admin only)</td>
</tr>
<tr>
<td>--device-owner &lt;device-owner&gt;</td>
<td>Device owner of this port. this is the entity that uses the port (for example, network:dhcp).</td>
</tr>
<tr>
<td>--vnic-type &lt;vnic-type&gt;</td>
<td>Vnic type for this port (direct</td>
</tr>
<tr>
<td>--host &lt;host-id&gt;</td>
<td>Allocate port on host &lt;host-id&gt; (id only)</td>
</tr>
<tr>
<td>--dns-domain dns-domain</td>
<td>Set dns domain to this port (requires dns_domain extension for ports)</td>
</tr>
<tr>
<td>--dns-name &lt;dns-name&gt;</td>
<td>Set dns name for this port (requires dns integration extension)</td>
</tr>
<tr>
<td>--numa-policy-required</td>
<td>Numa affinity policy required to schedule this port</td>
</tr>
<tr>
<td>--numa-policy-preferred</td>
<td>Numa affinity policy preferred to schedule this port</td>
</tr>
<tr>
<td>--numa-policy-legacy</td>
<td>Numa affinity policy using legacy mode to schedule this port</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable port</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable port</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set port name</td>
</tr>
<tr>
<td>--fixed-ip subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt;</td>
<td>Desired ip and/or subnet for this port (name or id): subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt; (repeat option to set multiple fixed IP addresses)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--no-fixed-ip</td>
<td>Clear existing information of fixed ip addresses. Specify both --fixed-ip and --no-fixed-ip to overwrite the current fixed IP addresses.</td>
</tr>
<tr>
<td>--binding-profile &lt;binding-profile&gt;</td>
<td>Custom data to be passed as binding:profile. data may be passed as &lt;key&gt;=&lt;value&gt; or JSON. (repeat option to set multiple binding:profile data)</td>
</tr>
<tr>
<td>--no-binding-profile</td>
<td>Clear existing information of binding:profile. specify both --binding-profile and --no-binding-profile to overwrite the current binding:profile information.</td>
</tr>
<tr>
<td>--qos-policy &lt;qos-policy&gt;</td>
<td>Attach qos policy to this port (name or id)</td>
</tr>
<tr>
<td>--security-group &lt;security-group&gt;</td>
<td>Security group to associate with this port (name or ID) (repeat option to set multiple security groups)</td>
</tr>
<tr>
<td>--no-security-group</td>
<td>Clear existing security groups associated with this port</td>
</tr>
<tr>
<td>--enable-port-security</td>
<td>Enable port security for this port</td>
</tr>
<tr>
<td>--disable-port-security</td>
<td>Disable port security for this port</td>
</tr>
<tr>
<td>--allowed-address ip-address=&lt;ip-address&gt;[,mac-address=&lt;mac-address&gt;]</td>
<td>Add allowed-address pair associated with this port: ip-address=&lt;ip-address&gt;[,mac-address=&lt;mac-address&gt;] (repeat option to set multiple allowed-address pairs)</td>
</tr>
<tr>
<td>--no-allowed-address</td>
<td>Clear existing allowed-address pairs associated with this port. (Specify both --allowed-address and --no-allowed-address to overwrite the current allowed-address pairs)</td>
</tr>
<tr>
<td>--data-plane-status &lt;status&gt;</td>
<td>Set data plane status of this port (active</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the port (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the port. specify both --tag and --no-tag to overwrite current tags</td>
</tr>
</tbody>
</table>

59.5. PORT SHOW

Display port details
Usage:

openstack port show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN]
  [--noindent] [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  <port>

Table 59.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Port to display (name or id)</td>
</tr>
</tbody>
</table>

Table 59.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 59.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 59.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 59.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 59.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
59.6. PORT UNSET

Unset port properties

Usage:

```bash
openstack port unset [-h]  
[--fixed-ip subnet=<subnet>,ip-address=<ip-address>]  
[--binding-profile <binding-profile-key>]  
[--security-group <security-group>]  
[--allowed-address ip-address=<ip-address>,[mac-address=<mac-address>]]  
[--qos-policy] [--data-plane-status]  
[--numa-policy] [--tag <tag> | --all-tag]  
<port>
```

Table 59.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>Port to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 59.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fixed-ip subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt;</td>
<td>Desired ip and/or subnet which should be removed from this port (name or ID): subnet=&lt;subnet&gt;,ip-address=&lt;ip-address&gt; (repeat option to unset multiple fixed IP addresses)</td>
</tr>
<tr>
<td>--binding-profile &lt;binding-profile-key&gt;</td>
<td>Desired key which should be removed from binding:profile (repeat option to unset multiple binding:profile data)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--security-group &lt;security-group&gt;</td>
<td>Security group which should be removed this port (name or ID) (repeat option to unset multiple security groups)</td>
</tr>
<tr>
<td>--allowed-address ip-address=&lt;ip-address&gt;[,mac-address=&lt;mac-address&gt;]</td>
<td>Desired allowed-address pair which should be removed from this port; ip-address=&lt;ip-address&gt; [,mac-address=&lt;mac-address&gt;] (repeat option to unset multiple allowed-address pairs)</td>
</tr>
<tr>
<td>--qos-policy</td>
<td>Remove the qos policy attached to the port</td>
</tr>
<tr>
<td>--data-plane-status</td>
<td>Clear existing information of data plane status</td>
</tr>
<tr>
<td>--numa-policy</td>
<td>Clear existing numa affinity policy</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the port (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the port</td>
</tr>
</tbody>
</table>
CHAPTER 60. PROJECT

This chapter describes the commands under the `project` command.

60.1. PROJECT CLEANUP

Clean resources associated with a project

**Usage:**

```
openstack project cleanup [-h] [--dry-run] 
   (--auth-project | --project <project>) 
   [--created-before <YYYY-MM-DDTHH24:MI:SS>] 
   [--updated-before <YYYY-MM-DDTHH24:MI:SS>] 
   [--project-domain <project-domain>]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--dry-run</td>
<td>List a project’s resources</td>
</tr>
<tr>
<td>--auth-project</td>
<td>Delete resources of the project used to authenticate</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project to clean (name or id)</td>
</tr>
<tr>
<td>--created-before <a href="">YYYY-MM-DDTHH24:MI:SS</a></td>
<td>Drop resources created before the given time</td>
</tr>
<tr>
<td>--updated-before <a href="">YYYY-MM-DDTHH24:MI:SS</a></td>
<td>Drop resources updated before the given time</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can</td>
</tr>
<tr>
<td></td>
<td>be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

60.2. PROJECT CREATE

Create new project

**Usage:**

```
openstack project create [-h] [-f {json,shell,table,value,yaml}] 
   [-c COLUMN] [--noindent] [--prefix PREFIX] 
   [--max-width <integer>] [--fit-width] 
   [--print-empty] [--domain <domain>] 
   [--parent <project>] 
   [--description <description>] 
   [--enable | --disable]
```
Table 60.2. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project-name&gt;</td>
<td>New project name</td>
</tr>
</tbody>
</table>

Table 60.3. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning the project (name or id)</td>
</tr>
<tr>
<td>--parent &lt;project&gt;</td>
<td>Parent of the project (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Project description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable project</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable project</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Add a property to &lt;name&gt; (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--or-show</td>
<td>Return existing project</td>
</tr>
<tr>
<td>--immutable</td>
<td>Make resource immutable. an immutable project may not be deleted or modified except to remove the immutable flag</td>
</tr>
<tr>
<td>--no-immutable</td>
<td>Make resource mutable (default)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the project (repeat option to set multiple tags)</td>
</tr>
</tbody>
</table>

Table 60.4. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
## 60.3. PROJECT DELETE

Delete project(s)

### Usage:

```
openstack project delete [-h] [--domain <domain>] <project> [<project> ...]
```

### Table 60.8. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Project(s) to delete (name or id)</td>
</tr>
</tbody>
</table>
### Table 60.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;project&gt; (name or id)</td>
</tr>
</tbody>
</table>

### 60.4. PROJECT LIST

List projects

**Usage:**

```
openstack project list [-h] [-f {csv,json,table,value,yaml}] 
   [-c COLUMN] 
   [--quote {all,minimal,none,nonnumeric}] 
   [-noindent] [-max-width <integer>] 
   [--fit-width] [-print-empty] 
   [--sort-column SORT_COLUMN] 
   [--sort-ascending | --sort-descending] 
   [--domain <domain>] [--parent <parent>] 
   [--user <user>] [-my-projects] [-long] 
   [-sort <key>[:<direction>]] 
   [-tags <tag>,<tag>,...]] 
   [-tags-any <tag>,<tag>,...]] 
   [-not-tags <tag>,<tag>,...]] 
   [-not-tags-any <tag>,<tag>,...]]
```

### Table 60.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Filter projects by &lt;domain&gt; (name or id)</td>
</tr>
<tr>
<td>--parent &lt;parent&gt;</td>
<td>Filter projects whose parent is &lt;parent&gt; (name or id)</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Filter projects by &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>--my-projects</td>
<td>List projects for the authenticated user. supersedes other filters.</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by selected keys and directions (asc or desc) (default: asc), repeat this option to specify multiple keys and directions.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;...]</td>
<td>List projects which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--tags-any &lt;tag&gt;[,&lt;tag&gt;...]</td>
<td>List projects which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;...]</td>
<td>Exclude projects which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags-any &lt;tag&gt;[,&lt;tag&gt;...]</td>
<td>Exclude projects which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 60.11. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 60.12. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 60.13. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 60.14. Table formatter options
**60.5. PROJECT PURGE**

Clean resources associated with a project

**Usage:**

```
openstack project purge [-h] [--dry-run] [--keep-project]  
  (--auth-project | --project <project>)  
  [--project-domain <project-domain>]
```

**Table 60.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**60.6. PROJECT SET**

Set project properties

**Usage:**

```
openstack project set [-h] [--name <name>] [--domain <domain>]  
  [--description <description>]  
  [--enable | --disable] [--property <key=value>]
```
### Table 60.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Project to modify (name or id)</td>
</tr>
</tbody>
</table>

### Table 60.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set project name</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set project description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable project</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable project</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on &lt;project&gt; (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--immutable</td>
<td>Make resource immutable. an immutable project may not be deleted or modified except to remove the immutable flag</td>
</tr>
<tr>
<td>--no-immutable</td>
<td>Make resource mutable (default)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the project (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--clear-tags</td>
<td>Clear tags associated with the project. specify both -tag and --clear-tags to overwrite current tags</td>
</tr>
<tr>
<td>--remove-tag &lt;tag&gt;</td>
<td>Tag to be deleted from the project (repeat option to delete multiple tags)</td>
</tr>
</tbody>
</table>

### 60.7. PROJECT SHOW

Display project details

**Usage:**

Table 60.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project&gt;</td>
<td>Project to display (name or id)</td>
</tr>
</tbody>
</table>

Table 60.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--parents</td>
<td>Show the project’s parents as a list</td>
</tr>
<tr>
<td>--children</td>
<td>Show project’s subtree (children) as a list</td>
</tr>
</tbody>
</table>

Table 60.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 60.21. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 60.22. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
### Table 60.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 61. PTR

This chapter describes the commands under the `ptr` command.

61.1. PTR RECORD LIST

List floatingip ptr records

Usage:

```
```

Table 61.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 61.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 61.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 61.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 61.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

61.2. PTR RECORD SET

Set floatingip ptr record

Usage:


Table 61.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>floatingip_id</td>
<td>Floating ip id in format region:floatingip_id</td>
</tr>
<tr>
<td>ptrdname</td>
<td>Ptd name</td>
</tr>
</tbody>
</table>
### Table 61.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--description DESCRIPTION</code></td>
<td>Description</td>
</tr>
<tr>
<td><code>--no-description--ttl TTL</code></td>
<td>Ttl</td>
</tr>
<tr>
<td><code>--no-ttl--all-projects</code></td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td><code>--sudo-project-id SUDO_PROJECT_ID</code></td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

### Table 61.8. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 61.9. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 61.10. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 61.11. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
61.3. PTR RECORD SHOW

Show floatingip ptr record details

Usage:


Table 61.12. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>floatingip_id</td>
<td>Floating ip id in format region:floatingip_id</td>
</tr>
</tbody>
</table>

Table 61.13. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 61.14. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 61.15. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 61.16. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 61.17. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 61.4. PTR RECORD UNSET

Unset floatingip ptr record

**Usage:**

```
openstack ptr record unset [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] floatingip_id
```

Table 61.18. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>floatingip_id</td>
<td>Floating ip id in format region:floatingip_id</td>
</tr>
</tbody>
</table>
### Table 61.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>
CHAPTER 62. QUOTA

This chapter describes the commands under the quota command.

62.1. QUOTA LIST

List quotas for all projects with non-default quota values or list detailed quota informations for requested project

Usage:

```
openstack quota list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] 
   [--quote {all, minimal, none, nonnumeric}] 
   [--noindent] [--max-width <integer>] [--fit-width] 
   [--print-empty] [--sort-column SORT_COLUMN] 
   [--sort-ascending | --sort-descending] 
   [--project <project>] [--detail] 
   (--compute | --volume | --network)
```

Table 62.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List quotas for this project &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--detail</td>
<td>Show details about quotas usage</td>
</tr>
<tr>
<td>--compute</td>
<td>List compute quota</td>
</tr>
<tr>
<td>--volume</td>
<td>List volume quota</td>
</tr>
<tr>
<td>--network</td>
<td>List network quota</td>
</tr>
</tbody>
</table>

Table 62.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
### Table 62.3. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 62.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 62.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 62.2. QUOTA SET

Set quotas for project or class

**Usage:**

```
```
Table 62.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;project/class&gt;</td>
<td>Set quotas for this project or class (name/id)</td>
</tr>
</tbody>
</table>

Table 62.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--class</td>
<td>Set quotas for &lt;class&gt;</td>
</tr>
<tr>
<td>--cores &lt;cores&gt;</td>
<td>New value for the cores quota</td>
</tr>
<tr>
<td>--fixed-ips &lt;fixed-ips&gt;</td>
<td>New value for the fixed-ips quota</td>
</tr>
<tr>
<td>--injected-file-size &lt;injected-file-size&gt;</td>
<td>New value for the injected-file-size quota</td>
</tr>
<tr>
<td>--injected-path-size &lt;injected-path-size&gt;</td>
<td>New value for the injected-path-size quota</td>
</tr>
<tr>
<td>--injected-files &lt;injected-files&gt;</td>
<td>New value for the injected-files quota</td>
</tr>
<tr>
<td>--instances &lt;instances&gt;</td>
<td>New value for the instances quota</td>
</tr>
<tr>
<td>--key-pairs &lt;key-pairs&gt;</td>
<td>New value for the key-pairs quota</td>
</tr>
<tr>
<td>--properties &lt;properties&gt;</td>
<td>New value for the properties quota</td>
</tr>
<tr>
<td>--ram &lt;ram&gt;</td>
<td>New value for the ram quota</td>
</tr>
<tr>
<td>--server-groups &lt;server-groups&gt;</td>
<td>New value for the server-groups quota</td>
</tr>
<tr>
<td>--server-group-members &lt;server-group-members&gt;</td>
<td>New value for the server-group-members quota</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--backups &lt;backups&gt;</td>
<td>New value for the backups quota</td>
</tr>
<tr>
<td>--backup-gigabytes &lt;backup-gigabytes&gt;</td>
<td>New value for the backup-gigabytes quota</td>
</tr>
<tr>
<td>--gigabytes &lt;gigabytes&gt;</td>
<td>New value for the gigabytes quota</td>
</tr>
<tr>
<td>--per-volume-gigabytes &lt;per-volume-gigabytes&gt;</td>
<td>New value for the per-volume-gigabytes quota</td>
</tr>
<tr>
<td>--snapshots &lt;snapshots&gt;</td>
<td>New value for the snapshots quota</td>
</tr>
<tr>
<td>--volumes &lt;volumes&gt;</td>
<td>New value for the volumes quota</td>
</tr>
<tr>
<td>--floating-ips &lt;floating-ips&gt;</td>
<td>New value for the floating-ips quota</td>
</tr>
<tr>
<td>--secgroup-rules &lt;secgroup-rules&gt;</td>
<td>New value for the secgroup-rules quota</td>
</tr>
<tr>
<td>--secgroups &lt;secgroups&gt;</td>
<td>New value for the secgroups quota</td>
</tr>
<tr>
<td>--networks &lt;networks&gt;</td>
<td>New value for the networks quota</td>
</tr>
<tr>
<td>--subnets &lt;subnets&gt;</td>
<td>New value for the subnets quota</td>
</tr>
<tr>
<td>--ports &lt;ports&gt;</td>
<td>New value for the ports quota</td>
</tr>
<tr>
<td>--routers &lt;routers&gt;</td>
<td>New value for the routers quota</td>
</tr>
<tr>
<td>--rbac-policies &lt;rbac-policies&gt;</td>
<td>New value for the rbac-policies quota</td>
</tr>
<tr>
<td>--subnetpools &lt;subnetpools&gt;</td>
<td>New value for the subnetpools quota</td>
</tr>
<tr>
<td>--volume-type &lt;volume-type&gt;</td>
<td>Set quotas for a specific &lt;volume-type&gt;</td>
</tr>
<tr>
<td>--force</td>
<td>Force quota update (only supported by compute)</td>
</tr>
</tbody>
</table>

### 62.3. QUOTA SHOW

Show quotas for project or class. Specify `--os-compute-api-version 2.50` or higher to see `server-groups` and `server-group-members` output for a given quota class.

**Usage:**

```
openstack quota show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] [--class | --default] [<project/class>]
```
### Table 62.8. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;project/class&gt;</code></td>
<td>Show quotas for this project or class (name or id)</td>
</tr>
</tbody>
</table>

### Table 62.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--class</code></td>
<td>Show quotas for <code>&lt;class&gt;</code></td>
</tr>
<tr>
<td><code>--default</code></td>
<td>Show default quotas for <code>&lt;project&gt;</code></td>
</tr>
</tbody>
</table>

### Table 62.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 62.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 62.12. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 62.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 63. RECORDSET

This chapter describes the commands under the `recordset` command.

63.1. RECORDSET CREATE

Create new recordset

Usage:

```
```

Table 63.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id</td>
</tr>
<tr>
<td>name</td>
<td>Recordset name</td>
</tr>
</tbody>
</table>

Table 63.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--record RECORD</td>
<td>Recordset record, repeat if necessary</td>
</tr>
<tr>
<td>--type TYPE</td>
<td>Recordset type</td>
</tr>
<tr>
<td>--ttl TTL</td>
<td>Time to live (seconds)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 63.3. Output formatter options
63.2. RECORDSET DELETE

Delete recordset

Usage:

```bash
```
### Table 63.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id</td>
</tr>
<tr>
<td>id</td>
<td>Recordset id</td>
</tr>
</tbody>
</table>

### Table 63.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
<tr>
<td>--edit-managed</td>
<td>Edit resources marked as managed. default: false</td>
</tr>
</tbody>
</table>

### Table 63.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 63.10. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 63.11. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 63.12. Table formatter options
63.3. RECORDSET LIST

List recordsets

Usage:

```
```

Table 63.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id. to list all recordsets specify all</td>
</tr>
</tbody>
</table>

Table 63.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Recordset name</td>
</tr>
<tr>
<td>--type TYPE</td>
<td>Recordset type</td>
</tr>
<tr>
<td>--data DATA</td>
<td>Recordset record data</td>
</tr>
</tbody>
</table>

---

Table 63.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

---

CHAPTER 63. RECORDSET

923
### Table 63.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--ttl TTL</td>
<td>Time to live (seconds)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--status STATUS</td>
<td>Recordset status</td>
</tr>
<tr>
<td>--action ACTION</td>
<td>Recordset action</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

### Table 63.16. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [csv, json, table, value, yaml], --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 63.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 63.18. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 63.4. RECORDSET SET

Set recordset properties

#### Usage:

```
```

#### Table 63.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id</td>
</tr>
<tr>
<td>id</td>
<td>Recordset id</td>
</tr>
</tbody>
</table>

#### Table 63.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--record RECORD</td>
<td>Recordset record, repeat if necessary</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--no-description --ttl TTL</td>
<td>Ttl</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--no-ttl--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
<tr>
<td>--edit-managed</td>
<td>Edit resources marked as managed. default: false</td>
</tr>
</tbody>
</table>

Table 63.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 63.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 63.23. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 63.24. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

63.5. RECORDSET SHOW
Show recordset details

Usage:

```
```

Table 63.25. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id</td>
</tr>
<tr>
<td>id</td>
<td>Recordset id</td>
</tr>
</tbody>
</table>

Table 63.26. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 63.27. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 63.28. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 63.29. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 63.30. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 64. REGION

This chapter describes the commands under the `region` command.

### 64.1. REGION CREATE

Create new region

**Usage:**

```
openstack region create [-h] [-f {json,shell,table,value,yaml}]  
[ -c COLUMN] [--noindent] [--prefix PREFIX]  
[ --max-width <integer> ] [ --fit-width ]  
[ --print-empty ] [ --parent-region <region-id> ]  
[ --description <description> ]  
<region-id>
```

**Table 64.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;region-id&gt;</code></td>
<td>New region id</td>
</tr>
</tbody>
</table>

**Table 64.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--parent-region &lt;region-id&gt;</code></td>
<td>Parent region id</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>New region description</td>
</tr>
</tbody>
</table>

**Table 64.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format</code> <code>{json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 64.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### Table 64.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 64.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 64.2. REGION DELETE

Delete region(s)

**Usage:**

```
openstack region delete [-h] <region-id> [<region-id> ...]
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;region-id&gt;</td>
<td>Region id(s) to delete</td>
</tr>
</tbody>
</table>

### Table 64.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 64.3. REGION LIST

List regions

**Usage:**

```
openstack region list [-h] [\{-f \{csv,json,table,value,yaml\}\} [-c COLUMN] [-quote \{all,minimal,none,nonnumeric\}]
```

930
Table 64.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--parent-region &lt;region-id&gt;</td>
<td>Filter by parent region id</td>
</tr>
</tbody>
</table>

Table 64.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 64.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 64.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 64.13. Table formatter options
### 64.4. REGION SET

Set region properties

**Usage:**

```bash
openstack region set [-h] [--parent-region <region-id>] [--description <description>] <region-id>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 64.14. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;region-id&gt;</td>
<td>Region to modify</td>
</tr>
</tbody>
</table>

**Table 64.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--parent-region &lt;region-id&gt;</td>
<td>New parent region id</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New region description</td>
</tr>
</tbody>
</table>

### 64.5. REGION SHOW

Display region details

**Usage:**

```bash
openstack region show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width]
```
Table 64.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;region-id&gt;</td>
<td>Region to display</td>
</tr>
</tbody>
</table>

Table 64.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 64.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 64.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 64.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 64.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 65. REGISTERED

This chapter describes the commands under the `registered` command.

65.1. REGISTERED LIMIT CREATE

Create a registered limit

Usage:

```
openstack registered limit create [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  [--description <description>]
  [--region <region>] --service
  <service> --default-limit
  <default-limit>
  <resource-name>
```

Table 65.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;resource-name&gt;</td>
<td>The name of the resource to limit</td>
</tr>
</tbody>
</table>

Table 65.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the registered limit</td>
</tr>
<tr>
<td>--region &lt;region&gt;</td>
<td>Region for the registered limit to affect</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Service responsible for the resource to limit</td>
</tr>
<tr>
<td>--default-limit &lt;default-limit&gt;</td>
<td>The default limit for the resources to assume</td>
</tr>
</tbody>
</table>

Table 65.3. Output formatter options
### 65.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>

### 65.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 65.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 65.2. REGISTERED LIMIT DELETE

Delete a registered limit

**Usage:**

```bash
code
openstack registered limit delete [-h] <registered-limit-id> [<registered-limit-id> ...]
```

**Table 65.7. Positional arguments**
65.3. REGISTERED LIMIT LIST

List registered limits

Usage:

```
```

Table 65.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Service responsible for the resource to limit</td>
</tr>
<tr>
<td>--resource-name &lt;resource-name&gt;</td>
<td>The name of the resource to limit</td>
</tr>
<tr>
<td>--region &lt;region&gt;</td>
<td>Region for the limit to affect.</td>
</tr>
</tbody>
</table>

Table 65.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 65.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 65.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 65.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

65.4. REGISTERED LIMIT SET

Update information about a registered limit

Usage:

```
```
Table 65.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;registered-limit-id&gt;</td>
<td>Registered limit to update (id)</td>
</tr>
</tbody>
</table>

Table 65.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Service to be updated responsible for the resource to limit. Either --service, --resource-name or --region must be different than existing value otherwise it will be duplicate entry</td>
</tr>
<tr>
<td>--resource-name &lt;resource-name&gt;</td>
<td>Resource to be updated responsible for the resource to limit. Either --service, --resource-name or --region must be different than existing value otherwise it will be duplicate entry</td>
</tr>
<tr>
<td>--default-limit &lt;default-limit&gt;</td>
<td>The default limit for the resources to assume</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description to update of the registered limit</td>
</tr>
<tr>
<td>--region &lt;region&gt;</td>
<td>Region for the registered limit to affect. either --service, --resource-name or --region must be different than existing value otherwise it will be duplicate entry</td>
</tr>
</tbody>
</table>

Table 65.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 65.17. JSON formatter options

-
Table 65.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 65.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

65.5. REGISTERED LIMIT SHOW

Display registered limit details

Usage:

```
```

Table 65.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;registered-limit-id&gt;</td>
<td>Registered limit to display (id)</td>
</tr>
</tbody>
</table>
### Table 65.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 65.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 65.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 65.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 66. REQUEST

This chapter describes the commands under the request command.

66.1. REQUEST TOKEN AUTHORIZE

Authorize a request token

Usage:

```
openstack request token authorize [-h]
                              [-f {json,shell,table,value,yaml}]
                              [-c COLUMN] [--noindent]
                              [--prefix PREFIX]
                              [--max-width <integer>] [--fit-width]
                              [--print-empty] --request-key
                              <request-key> --role <role>
```

Table 66.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--request-key &lt;request-key&gt;</td>
<td>Request token to authorize (id only) (required)</td>
</tr>
<tr>
<td>--role &lt;role&gt;</td>
<td>Roles to authorize (name or id) (repeat option to set multiple values) (required)</td>
</tr>
</tbody>
</table>

Table 66.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 66.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 66.4. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 66.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 66.2. REQUEST TOKEN CREATE

Create a request token

**Usage:**

```
```

**Table 66.6. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--consumer-key &lt;consumer-key&gt;</td>
<td>Consumer key (required)</td>
</tr>
<tr>
<td>--consumer-secret &lt;consumer-secret&gt;</td>
<td>Consumer secret (required)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project that consumer wants to access (name or id) (required)</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;project&gt; (name or id)</td>
</tr>
</tbody>
</table>
### Table 66.7. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 66.8. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 66.9. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 66.10. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 67. RESOURCE

This chapter describes the commands under the `resource` command.

67.1. RESOURCE MEMBER CREATE

Shares a resource to another tenant.

**Usage:**

```
openstack resource member create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty]
resource_id resource_type member_id
```

**Table 67.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Resource id to be shared.</td>
</tr>
<tr>
<td>resource_type</td>
<td>Resource type.</td>
</tr>
<tr>
<td>member_id</td>
<td>Project id to whom the resource is shared to.</td>
</tr>
</tbody>
</table>

**Table 67.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 67.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 67.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 67.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 67.2. RESOURCE MEMBER DELETE

Delete a resource sharing relationship.

**Usage:**

```
openstack resource member delete [-h] resource resource_type member_id
```

**Table 67.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource</td>
<td>Resource id to be shared.</td>
</tr>
<tr>
<td>resource_type</td>
<td>Resource type.</td>
</tr>
<tr>
<td>member_id</td>
<td>Project id to whom the resource is shared to.</td>
</tr>
</tbody>
</table>

**Table 67.8. Command arguments**
### 67.3. RESOURCE MEMBER LIST

List all members.

#### Usage:

```bash
resource_id resource_type
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Resource id to be shared.</td>
</tr>
<tr>
<td>resource_type</td>
<td>Resource type.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
</tbody>
</table>
Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc

Filters. can be repeated.

Table 67.11. Output formatter options

- The output format, defaults to table

Table 67.12. CSV formatter options

When to include quotes, defaults to nonnumeric

Table 67.13. JSON formatter options

Whether to disable indenting the json

Table 67.14. Table formatter options

Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 67.4. RESOURCE MEMBER SHOW

Show specific member information.

**Usage:**

```
openstack resource member show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [-m MEMBER_ID] [-print-empty] [-m MEMBER_ID]
```

Table 67.15. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource</td>
<td>Resource id to be shared.</td>
</tr>
<tr>
<td>resource_type</td>
<td>Resource type.</td>
</tr>
</tbody>
</table>

Table 67.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-m MEMBER_ID, --member-id MEMBER_ID</td>
<td>Project id to whom the resource is shared to. no need to provide this param if you are the resource member.</td>
</tr>
</tbody>
</table>

Table 67.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### 67.5. RESOURCE MEMBER UPDATE

Update resource sharing status.

**Usage:**

```
openstack resource member update [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty] [-m MEMBER_ID]
    [-s {pending,accepted,rejected}]
    resource_id resource_type
```

**Table 67.21. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 67.18. JSON formatter options**

**Table 67.19. Shell formatter options**

**Table 67.20. Table formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource_id</td>
<td>Resource id to be shared.</td>
</tr>
<tr>
<td>resource_type</td>
<td>Resource type.</td>
</tr>
</tbody>
</table>

Table 67.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-m MEMBER_ID, --member-id MEMBER_ID</td>
<td>Project id to whom the resource is shared to. no need to provide this param if you are the resource member.</td>
</tr>
<tr>
<td>-s {pending,accepted,rejected}, --status {pending,accepted,rejected}</td>
<td>Status of the sharing.</td>
</tr>
</tbody>
</table>

Table 67.23. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 67.24. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 67.25. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 67.26. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 68. ROLE

This chapter describes the commands under the `role` command.

68.1. ROLE ADD

Adds a role assignment to a user or group on the system, a domain, or a project

Usage:

```
openstack role add [-h] [--system <system> | --domain <domain> | --project <project>]
  [--user <user> | --group <group>]
  [--group-domain <group-domain>]
  [--project-domain <project-domain>]
  [--user-domain <user-domain> {[--inherited]}
  [--role-domain <role-domain>]
  <role>
```

Table 68.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role to add to &lt;user&gt; (name or id)</td>
</tr>
</tbody>
</table>

Table 68.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--system &lt;system&gt;</td>
<td>Include &lt;system&gt; (all)</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Include &lt;domain&gt; (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Include &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Include &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>--group &lt;group&gt;</td>
<td>Include &lt;group&gt; (name or id)</td>
</tr>
<tr>
<td>--group-domain &lt;group-domain&gt;</td>
<td>Domain the group belongs to (name or id). this can be used in case collisions between group names exist.</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>
68.2. ROLE ASSIGNMENT LIST

List role assignments

Usage:

```
openstack role assignment list [-h] [-f {csv,json,table,value,yaml}]
  [-c COLUMN] [-quote {all,minimal,none,nonnumeric}]
  [--noindent] [--max-width <integer>] [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN] [--sort-ascending | --sort-descending]
  [--effective] [--role <role>] [--role-domain <role-domain>]
  [--names] [--user <user>] [--user-domain <user-domain>]
  [--group <group>] [--group-domain <group-domain>]
  [--domain <domain> | --project <project> | --system <system>]
  [--project-domain <project-domain>] [--inherited] [--auth-user]
  [--auth-project]
```

Table 68.3. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--inherited</td>
<td>Specifies if the role grant is inheritable to the sub projects</td>
</tr>
<tr>
<td>--role-domain &lt;role-domain&gt;</td>
<td>Domain the role belongs to (name or id). This must be specified when the name of a domain specific role is used.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--names</td>
<td>Display names instead of ids</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>User to filter (name or id)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--group &lt;group&gt;</td>
<td>Group to filter (name or id)</td>
</tr>
<tr>
<td>--group-domain &lt;group-domain&gt;</td>
<td>Domain the group belongs to (name or id). This can be used in case collisions between group names exist.</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain to filter (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project to filter (name or id)</td>
</tr>
<tr>
<td>--system &lt;system&gt;</td>
<td>Filter based on system role assignments</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--inherited</td>
<td>Specifies if the role grant is inheritable to the sub projects</td>
</tr>
<tr>
<td>--auth-user</td>
<td>Only list assignments for the authenticated user</td>
</tr>
<tr>
<td>--auth-project</td>
<td>Only list assignments for the project to which the authenticated user’s token is scoped</td>
</tr>
</tbody>
</table>

**Table 68.4. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
### Table 68.5. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 68.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 68.7. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 68.3. ROLE CREATE

Create new role

**Usage:**

```
openstack role create [-h] [-f {json,shell,table,value,yaml}] 
  [-c COLUMN] [--noindent] [--prefix PREFIX] 
  [--max-width <integer>] [--fit-width] 
  [--print-empty] [--description <description>] 
  [--domain <domain>] [--or-show] 
  [--immutable | --no-immutable] 
  <role-name>
```

### Table 68.8. Positional arguments
### Table 68.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role-name&gt;</td>
<td>New role name</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Add description about the role</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the role belongs to (name or id)</td>
</tr>
<tr>
<td>--or-show</td>
<td>Return existing role</td>
</tr>
<tr>
<td>--immutable</td>
<td>Make resource immutable. An immutable project may not be deleted or modified except to remove the immutable flag</td>
</tr>
<tr>
<td>--no-immutable</td>
<td>Make resource mutable (default)</td>
</tr>
</tbody>
</table>

### Table 68.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 68.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 68.12. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 68.13. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 68.4. ROLE DELETE

Delete role(s)

**Usage:**

```bash
openstack role delete [-h] [--domain <domain>] <role> [<role> ...]
```

**Table 68.14. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 68.15. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the role belongs to (name or id)</td>
</tr>
</tbody>
</table>

### 68.5. ROLE LIST

List roles

**Usage:**

```bash
openstack role list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent] [--max-width <integer>] [--fit-width]
    [--print-empty] [-s sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--domain <domain>]
```

**Table 68.16. Command arguments**
### Table 68.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Include &lt;domain&gt; (name or id)</td>
</tr>
</tbody>
</table>

### Table 68.18. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 68.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 68.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 68.6. ROLE REMOVE

Removes a role assignment from system/domain/project : user/group

**Usage:**

```bash
openstack role remove [-h]
[--system <system> | --domain <domain> | --project <project>]
[--user <user> | --group <group>]
[--group-domain <group-domain>]
[--project-domain <project-domain>]
[--user-domain <user-domain>] [--inherited]
[--role-domain <role-domain>]
<role>
```

**Table 68.21. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role to remove (name or id)</td>
</tr>
</tbody>
</table>

**Table 68.22. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--system &lt;system&gt;</td>
<td>Include &lt;system&gt; (all)</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Include &lt;domain&gt; (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Include &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Include &lt;user&gt; (name or id)</td>
</tr>
<tr>
<td>--group &lt;group&gt;</td>
<td>Include &lt;group&gt; (name or id)</td>
</tr>
<tr>
<td>--group-domain &lt;group-domain&gt;</td>
<td>Domain the group belongs to (name or id). this can be used in case collisions between group names exist.</td>
</tr>
</tbody>
</table>
68.7. ROLE SET

Set role properties

Usage:


Table 68.23. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 68.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Add description about the role</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the role belongs to (name or id)</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set role name</td>
</tr>
</tbody>
</table>
68.8. ROLE SHOW

Display role details

**Usage:**

```
openstack role show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN]
  [--noindent] [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty] [--domain <domain>]
  <role>
```

**Table 68.25. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;role&gt;</td>
<td>Role to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 68.26. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the role belongs to (name or id)</td>
</tr>
</tbody>
</table>

**Table 68.27. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 68.28. JSON formatter options**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 68.29. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 68.30. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 69. ROUTER

This chapter describes the commands under the `router` command.

69.1. ROUTER ADD PORT

Add a port to a router

**Usage:**

```
openstack router add port [-h] <router> <port>
```

**Table 69.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;router&gt;</code></td>
<td>Router to which port will be added (name or id)</td>
</tr>
<tr>
<td><code>&lt;port&gt;</code></td>
<td>Port to be added (name or id)</td>
</tr>
</tbody>
</table>

**Table 69.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

69.2. ROUTER ADD ROUTE

Add extra static routes to a router’s routing table.

**Usage:**

```
```

**Table 69.3. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;router&gt;</code></td>
<td>Router to which extra static routes will be added (name or ID).</td>
</tr>
</tbody>
</table>

**Table 69.4. Command arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--route destination=&lt;subnet&gt;, gateway=&lt;ip-address&gt;</td>
<td>Add extra static route to the router. destination: destination subnet (in CIDR notation), gateway: nexthop IP address. Repeat option to add multiple routes. Trying to add a route that's already present (exactly, including destination and nexthop) in the routing table is allowed and is considered a successful operation.</td>
</tr>
</tbody>
</table>

Table 69.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 69.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 69.7. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 69.8. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
69.3. ROUTER ADD SUBNET
Add a subnet to a router

Usage:
```
openstack router add subnet [-h] <router> <subnet>
```

Table 69.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router to which subnet will be added (name or id)</td>
</tr>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet to be added (name or id)</td>
</tr>
</tbody>
</table>

Table 69.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

69.4. ROUTER CREATE
Create a new router

Usage:
```
```

Table 69.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New router name</td>
</tr>
</tbody>
</table>

Table 69.12. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable router (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable router</td>
</tr>
<tr>
<td>--distributed</td>
<td>Create a distributed router</td>
</tr>
<tr>
<td>--centralized</td>
<td>Create a centralized router</td>
</tr>
<tr>
<td>--ha</td>
<td>Create a highly available router</td>
</tr>
<tr>
<td>--no-ha</td>
<td>Create a legacy router</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set router description</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--availability-zone-hint &lt;availability-zone&gt;</td>
<td>Availability zone in which to create this router (Router Availability Zone extension required, repeat option to set multiple availability zones)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the router (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the router</td>
</tr>
</tbody>
</table>

Table 69.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 69.14. JSON formatter options
### Table 69.15. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 69.16. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the <code>CLIFF_MAX_TERM_WIDTH</code> environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 69.5. ROUTER DELETE

Delete router(s)

**Usage:**

```
openstack router delete [-h] <router> [<router> ...]
```

### Table 69.17. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

### Table 69.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 69.6. ROUTER LIST
List routers

Usage:

openstack router list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
[-quote {all, minimal, none, nonnumeric}]
[--noindent] [-max-width <integer>]
[-fit-width] [-print-empty]
[-sort-column SORT_COLUMN]
[-sort-ascending | --sort-descending]
[-name <name>] [--enable | --disable] [-long]
[--project <project>]
[--project-domain <project-domain>]
[--agent <agent-id>] [--tags <tag>[,<tag>,...]]
[--any-tags <tag>[,<tag>,...]]
[--not-tags <tag>[,<tag>,...]]
[--not-any-tags <tag>[,<tag>,...]]

Table 69.19. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List routers according to their name</td>
</tr>
<tr>
<td>--enable</td>
<td>List enabled routers</td>
</tr>
<tr>
<td>--disable</td>
<td>List disabled routers</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List routers according to their project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--agent &lt;agent-id&gt;</td>
<td>List routers hosted by an agent (id only)</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List routers which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List routers which have any given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude routers which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>Exclude routers which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

### Table 69.20. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 69.21. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 69.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 69.23. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 69.7. ROUTER REMOVE PORT

Remove a port from a router

**Usage:**
```
openstack router remove port [-h] <router> <port>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 69.24. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router from which port will be removed (name or ID)</td>
</tr>
<tr>
<td>&lt;port&gt;</td>
<td>Port to be removed and deleted (name or id)</td>
</tr>
</tbody>
</table>

Table 69.25. Command arguments

### 69.8. ROUTER REMOVE ROUTE

Remove extra static routes from a router’s routing table.

**Usage:**
```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router from which extra static routes will be removed (name or ID).</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</code></td>
<td>Remove extra static route from the router. destination: destination subnet (in CIDR notation), gateway: nexthop IP address. Repeat option to remove multiple routes. Trying to remove a route that’s already missing (fully, including destination and nexthop) from the routing table is allowed and is considered a successful operation.</td>
</tr>
</tbody>
</table>

Table 69.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>--column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 69.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 69.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 69.31. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
69.9. ROUTER REMOVE SUBNET

Remove a subnet from a router

Usage:

```
openstack router remove subnet [-h] <router> <subnet>
```

Table 69.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router from which the subnet will be removed (name or id)</td>
</tr>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet to be removed (name or id)</td>
</tr>
</tbody>
</table>

Table 69.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

69.10. ROUTER SET

Set router properties

Usage:

```
openstack router set [-h] [--name <name>] [--description <description>]
[---enable | --disable]
[---distributed | --centralized]
[---route destination=<subnet>,gateway=<ip-address>]
[---no-route] [---ha | --no-ha]
[---external-gateway <network>]
[---fixed-ip subnet=<subnet>,ip-address=<ip-address>]
[---enable-snat | --disable-snat]
[---qos-policy <qos-policy> | --no-qos-policy]
[---tag <tag>] [---no-tag]
<router>
```

Table 69.34. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 69.35. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set router name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set router description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable router</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable router</td>
</tr>
<tr>
<td>--distributed</td>
<td>Set router to distributed mode (disabled router only)</td>
</tr>
<tr>
<td>--centralized</td>
<td>Set router to centralized mode (disabled router only)</td>
</tr>
</tbody>
</table>
| --route destination=<subnet>,gateway=<ip-address> | Add routes to the router destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to add multiple routes). This is deprecated in favor of `router add/remove route` since it is prone to race conditions between concurrent clients when not used together with `--no-route` to overwrite the current value of `routes`.
| --no-route                    | Clear routes associated with the router. specify both `--route` and `--no-route` to overwrite current value of `routes`. |
| --ha                          | Set the router as highly available (disabled router only)               |
| --no-ha                       | Clear high availability attribute of the router (disabled router only)   |
| --external-gateway <network> | External network used as router’s gateway (name or id)                  |
| --fixed-ip subnet=<subnet>,ip-address=<ip-address> | Desired ip and/or subnet (name or id) on external gateway: `subnet=<subnet>,ip-address=<ip-address>` (repeat option to set multiple fixed IP addresses) |
| --enable-snat                 | Enable source nat on external gateway                                   |
| --disable-snat                | Disable source nat on external gateway                                   |
| --qos-policy <qos-policy>     | Attach qos policy to router gateway ips                                 |
| --no-qos-policy               | Remove qos policy from router gateway ips                               |
69.11. ROUTER SHOW

Display router details

Usage:


Table 69.36. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router to display (name or id)</td>
</tr>
</tbody>
</table>

Table 69.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 69.38. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 69.39. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### Table 69.40. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 69.41. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 69.12. ROUTER UNSET

Unset router properties

**Usage:**

```
openstack router unset [-h]  
  [--route destination=<subnet>,gateway=<ip-address>]  
  [--external-gateway] [-qos-policy]  
  [-tag <tag> | --all-tag]  
  <router>
```

### Table 69.42. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;router&gt;</td>
<td>Router to modify (name or id)</td>
</tr>
</tbody>
</table>

### Table 69.43. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Routes to be removed from the router destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to unset multiple routes)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--external-gateway</td>
<td>Remove external gateway information from the router</td>
</tr>
<tr>
<td>--qos-policy</td>
<td>Remove qos policy from router gateway ips</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the router (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the router</td>
</tr>
</tbody>
</table>
CHAPTER 70. SECRET

This chapter describes the commands under the `secret` command.

70.1. SECRET CONTAINER CREATE

Store a container in Barbican.

Usage:

```bash
openstack secret container create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty] [--name NAME]
[--type TYPE] [--secret SECRET]
```

Table 70.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>A human-friendly name.</td>
</tr>
<tr>
<td>--type TYPE</td>
<td>Type of container to create (default: generic).</td>
</tr>
<tr>
<td>--secret SECRET, -s SECRET</td>
<td>One secret to store in a container (can be set multiple times). Example: --secret &quot;private_key=<a href="https://url.test/v1/secrets/1-2-3-4">https://url.test/v1/secrets/1-2-3-4</a>&quot;</td>
</tr>
</tbody>
</table>

Table 70.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>[json,shell,table,value,yaml]</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 70.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 70.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

70.2. SECRET CONTAINER DELETE

Delete a container by providing its href.

Usage:

openstack secret container delete [-h] URI

Table 70.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the container</td>
</tr>
</tbody>
</table>

Table 70.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

70.3. SECRET CONTAINER GET

Retrieve a container by providing its URI.

Usage:

openstack secret container get [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-noindent] [-prefix PREFIX]
Table 70.8. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the container.</td>
</tr>
</tbody>
</table>

Table 70.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 70.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 70.11. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 70.12. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 70.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
70.4. SECRET CONTAINER LIST

List containers.

Usage:

```
```

Table 70.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit LIMIT, -l LIMIT</td>
<td>Specify the limit to the number of items to list per page (default: 10; maximum: 100)</td>
</tr>
<tr>
<td>--offset OFFSET, -o OFFSET</td>
<td>Specify the page offset (default: 0)</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>Specify the container name (default: none)</td>
</tr>
<tr>
<td>--type TYPE, -t TYPE</td>
<td>Specify the type filter for the list (default: none).</td>
</tr>
</tbody>
</table>

Table 70.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**Value** | **Summary**
---|---
-c COLUMN, --column COLUMN  | Specify the column(s) to include, can be repeated to show multiple columns
--sort-column SORT_COLUMN  | Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated
--sort-ascending  | Sort the column(s) in ascending order
--sort-descending  | Sort the column(s) in descending order

**Table 70.16. CSV formatter options**

**Value** | **Summary**
---|---
--quote {all, minimal, none, nonnumeric}  | When to include quotes, defaults to nonnumeric

**Table 70.17. JSON formatter options**

**Value** | **Summary**
---|---
--noindent  | Whether to disable indenting the json

**Table 70.18. Table formatter options**

**Value** | **Summary**
---|---
--max-width <integer>  | Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence. 
--fit-width  | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty  | Print empty table if there is no data to show.

**70.5. SECRET DELETE**

Delete a secret by providing its URI.

**Usage:**

```
openstack secret delete [-h] URI
```
Table 70.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret</td>
</tr>
</tbody>
</table>

Table 70.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

70.6. SECRET GET

Retrieve a secret by providing its URI.

Usage:

```
```

Table 70.21. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret</td>
</tr>
</tbody>
</table>

Table 70.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--decrypt, -d</td>
<td>If specified, retrieve the unencrypted secret data.</td>
</tr>
<tr>
<td>--payload, -p</td>
<td>If specified, retrieve the unencrypted secret data.</td>
</tr>
<tr>
<td>--file &lt;filename&gt;, -F &lt;filename&gt;</td>
<td>If specified, save the payload to a new file with the given filename.</td>
</tr>
<tr>
<td>--payload_content_type PAYLOAD_CONTENT_TYPE, -t PAYLOAD_CONTENT_TYPE</td>
<td>The content type of the decrypted secret (default: text/plain).</td>
</tr>
</tbody>
</table>
### Table 70.23. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 70.24. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 70.25. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 70.26. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 70.7. SECRET LIST

List secrets.

**Usage:**

```bash
```
Table 70.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit LIMIT, -l LIMIT</td>
<td>Specify the limit to the number of items to list per page (default: 10; maximum: 100)</td>
</tr>
<tr>
<td>--offset OFFSET, -o OFFSET</td>
<td>Specify the page offset (default: 0)</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>Specify the secret name (default: none)</td>
</tr>
<tr>
<td>--algorithm ALGORITHM, -a ALGORITHM</td>
<td>The algorithm filter for the list (default: none).</td>
</tr>
<tr>
<td>--bit-length BIT_LENGTH, -b BIT_LENGTH</td>
<td>The bit length filter for the list (default: 0).</td>
</tr>
<tr>
<td>--mode MODE, -m MODE</td>
<td>The algorithm mode filter for the list (default: None).</td>
</tr>
<tr>
<td>--secret-type SECRET_TYPE, -s SECRET_TYPE</td>
<td>Specify the secret type (default: none).</td>
</tr>
</tbody>
</table>

Table 70.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 70.29. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 70.31. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 70.8. SECRET ORDER CREATE

Create a new order.

**Usage:**

```bash
```

**Table 70.32. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>The type of the order (key, asymmetric, certificate) to create.</td>
</tr>
</tbody>
</table>

**Table 70.33. Command arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>A human-friendly name.</td>
</tr>
<tr>
<td>--algorithm ALGORITHM, -a ALGORITHM</td>
<td>The algorithm to be used with the requested key (default: aes).</td>
</tr>
<tr>
<td>--bit-length BIT_LENGTH, -b BIT_LENGTH</td>
<td>The bit length of the requested secret key (default: 256).</td>
</tr>
<tr>
<td>--mode MODE, -m MODE</td>
<td>The algorithm mode to be used with the requested key (default: cbc).</td>
</tr>
<tr>
<td>--payload-content-type PAYLOAD_CONTENT_TYPE, -t PAYLOAD_CONTENT_TYPE</td>
<td>The type/format of the secret to be generated (default: application/octet-stream).</td>
</tr>
<tr>
<td>--expiration EXPIRATION, -x EXPIRATION</td>
<td>The expiration time for the secret in iso 8601 format.</td>
</tr>
<tr>
<td>--request-type REQUEST_TYPE</td>
<td>The type of the certificate request.</td>
</tr>
<tr>
<td>--subject-dn SUBJECT_DN</td>
<td>The subject of the certificate.</td>
</tr>
<tr>
<td>--source-container-ref SOURCE_CONTAINER_REF</td>
<td>The source of the certificate when using stored-key requests.</td>
</tr>
<tr>
<td>--ca-id CA_ID</td>
<td>The identifier of the ca to use for the certificate request.</td>
</tr>
<tr>
<td>--profile PROFILE</td>
<td>The profile of certificate to use.</td>
</tr>
<tr>
<td>--request-file REQUEST_FILE</td>
<td>The file containing the csr.</td>
</tr>
</tbody>
</table>

Table 70.34. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 70.35. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 70.36. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 70.37. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

70.9. SECRET ORDER DELETE

Delete an order by providing its href.

Usage:

```bash
openstack secret order delete [-h] URI
```

Table 70.38. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the order</td>
</tr>
</tbody>
</table>

Table 70.39. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

70.10. SECRET ORDER GET
Retrieve an order by providing its URI.

**Usage:**

```bash
```

**Table 70.40. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference order.</td>
</tr>
</tbody>
</table>

**Table 70.41. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 70.42. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 70.43. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 70.44. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 70.45. Table formatter options**
### 70.11. SECRET ORDER LIST

List orders.

**Usage:**

```
```

**Table 70.46. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit LIMIT, -l LIMIT</td>
<td>Specify the limit to the number of items to list per page (default: 10; maximum: 100)</td>
</tr>
<tr>
<td>--offset OFFSET, -o OFFSET</td>
<td>Specify the page offset (default: 0)</td>
</tr>
</tbody>
</table>

**Table 70.47. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format [csv, json, table, value, yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 70.48. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 70.49. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 70.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

70.12. SECRET STORE

Store a secret in Barbican.

Usage:

```
```
[--secret-type SECRET_TYPE]
[--payload-content-type PAYLOAD_CONTENT_TYPE]
[--payload-content-encoding PAYLOAD_CONTENT_ENCODING]
[--algorithm ALGORITHM]
[--bit-length BIT_LENGTH] [--mode MODE]
[--expiration EXPIRATION]
[--payload PAYLOAD | --file <filename>]

Table 70.51. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME, -n NAME</td>
<td>A human-friendly name.</td>
</tr>
<tr>
<td>--secret-type SECRET_TYPE, -s SECRET_TYPE</td>
<td>The secret type; must be one of symmetric, public, private, certificate, passphrase, opaque (default)</td>
</tr>
<tr>
<td>--payload-content-type PAYLOAD_CONTENT_TYPE, -t PAYLOAD_CONTENT_TYPE</td>
<td>The type/format of the provided secret data; &quot;text/plain&quot; is assumed to be UTF-8; required when --payload is supplied.</td>
</tr>
<tr>
<td>--payload-content-encoding PAYLOAD_CONTENT_ENCODING, -e PAYLOAD_CONTENT_ENCODING</td>
<td>Required if --payload-content-type is &quot;application/octet-stream&quot;.</td>
</tr>
<tr>
<td>--algorithm ALGORITHM, -a ALGORITHM</td>
<td>The algorithm (default: aes).</td>
</tr>
<tr>
<td>--bit-length BIT_LENGTH, -b BIT_LENGTH</td>
<td>The bit length (default: 256).</td>
</tr>
<tr>
<td>--mode MODE, -m MODE</td>
<td>The algorithm mode; used only for reference (default: cbc)</td>
</tr>
<tr>
<td>--expiration EXPIRATION, -x EXPIRATION</td>
<td>The expiration time for the secret in iso 8601 format.</td>
</tr>
<tr>
<td>--payload PAYLOAD, -p PAYLOAD</td>
<td>The unencrypted secret data.</td>
</tr>
<tr>
<td>--file &lt;filename&gt;, -F &lt;filename&gt;</td>
<td>File containing the secret payload</td>
</tr>
</tbody>
</table>

Table 70.52. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 70.53. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 70.54. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 70.55. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, `&lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

70.13. SECRET UPDATE

Update a secret with no payload in Barbican.

Usage:

```
openstack secret update [-h] URI payload
```

Table 70.56. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URI</td>
<td>The uri reference for the secret.</td>
</tr>
<tr>
<td>payload</td>
<td>The unencrypted secret</td>
</tr>
</tbody>
</table>

Table 70.57. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
CHAPTER 71. SECURITY

This chapter describes the commands under the `security` command.

71.1. SECURITY GROUP CREATE

Create a new security group

Usage:

```bash
```

Table 71.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New security group name</td>
</tr>
</tbody>
</table>

Table 71.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Security group description</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--stateful</td>
<td>--stateless</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>--no-tag</td>
</tr>
</tbody>
</table>
Table 71.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 71.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 71.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

71.2. SECURITY GROUP DELETE

Delete security group(s)

Usage:
openstack security group delete [-h] <group> [<group> ...]

Table 71.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Security group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 71.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

71.3. SECURITY GROUP LIST

List security groups

Usage:

openstack security group list [-h] [-f {csv, json, table, value, yaml}]
[-c COLUMN]
[--quote {all, minimal, none, nonnumeric}]
[--noindent] [-max-width <integer>]
[-fit-width] [-print-empty]
[-sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--project <project>]
[--project-domain <project-domain>]
[--tags <tag>[,<tag>,...]]
[--any-tags <tag>[,<tag>,...]]
[--not-tags <tag>[,<tag>,...]]
[--not-any-tags <tag>[,<tag>,...]]

Table 71.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List security groups according to the project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or ID), this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List security group which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,... ]</td>
<td>List security group which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;[,&lt;tag&gt;,... ]</td>
<td>Exclude security group which have all given tag(s) (Comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags &lt;tag&gt;[,&lt;tag&gt;,... ]</td>
<td>Exclude security group which have any given tag(s) (Comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 71.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 71.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 71.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.13. Table formatter options
71.4. SECURITY GROUP RULE CREATE

Create a new security group rule

Usage:

openstack security group rule create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--remote-ip <ip-address>] | [--remote-group <group>] | [--remote-address-group <group>]
[--dst-port <port-range>]
[--protocol <protocol>]
[--description <description>]
[--icmp-type <icmp-type>]
[--icmp-code <icmp-code>]
[--ingress | --egress]
[--ethertype <ethertype>]
[--project <project>]
[--project-domain <project-domain>]
<group>

Table 71.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Create rule in this security group (name or id)</td>
</tr>
</tbody>
</table>

Table 71.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--remote-ip &lt;ip-address&gt;</td>
<td>Remote ip address block (may use cidr notation; default for IPv4 rule: 0.0.0.0/0, default for IPv6 rule: ::/0)</td>
</tr>
<tr>
<td>--remote-group &lt;group&gt;</td>
<td>Remote security group (name or id)</td>
</tr>
<tr>
<td>--remote-address-group &lt;group&gt;</td>
<td>Remote address group (name or id)</td>
</tr>
<tr>
<td>--dst-port &lt;port-range&gt;</td>
<td>Destination port, may be a single port or a starting and ending port range: 137:139. Required for IP protocols TCP and UDP. Ignored for ICMP IP protocols.</td>
</tr>
<tr>
<td>--protocol &lt;protocol&gt;</td>
<td>Ip protocol (ah, dccp, egp, esp, gre, icmp, igmp, ipv6-encap, ipv6-frag, ipv6-icmp, ipv6-nonxt, ipv6-opts, ipv6-route, ospf, pgm, rsvp, sctp, tcp, udp, udplite, vrrp and integer representations [0-255] or any; default: any (all protocols))</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set security group rule description</td>
</tr>
<tr>
<td>--icmp-type &lt;icmp-type&gt;</td>
<td>Icmp type for icmp ip protocols</td>
</tr>
<tr>
<td>--icmp-code &lt;icmp-code&gt;</td>
<td>Icmp code for icmp ip protocols</td>
</tr>
<tr>
<td>--ingress</td>
<td>Rule applies to incoming network traffic (default)</td>
</tr>
<tr>
<td>--egress</td>
<td>Rule applies to outgoing network traffic</td>
</tr>
<tr>
<td>--ethertype &lt;ethertype&gt;</td>
<td>Ethertype of network traffic (ipv4, ipv6; default: based on IP protocol)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

Table 71.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 71.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 71.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

71.5. SECURITY GROUP RULE DELETE

Delete security group rule(s)

Usage:

```
openstack security group rule delete [-h] <rule> [<rule> ...]
```

Table 71.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rule&gt;</td>
<td>Security group rule(s) to delete (id only)</td>
</tr>
</tbody>
</table>
### 71.6. SECURITY GROUP RULE LIST

List security group rules

**Usage:**

```
openstack security group rule list [-h] 
[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent] [-max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--protocol <protocol>]
[--ethertype <ethertype>]
[--ingress | --egress] [--long]
[group>
```

**Table 71.22. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;group&gt;</code></td>
<td>List all rules in this security group (name or id)</td>
</tr>
</tbody>
</table>

**Table 71.23. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--protocol &lt;protocol&gt;</code></td>
<td>List rules by the ip protocol (ah, dhcp, egp, esp, gre, icmp, igmp, ipv6-encap, ipv6-frag, ipv6-icmp, ipv6- nonxt, ipv6-opts, ipv6-route, ospf, pgm, rsvp, sctp, tcp, udp, udp- lite, vrrp and integer representations [0-255] or any; default: any (all protocols))</td>
</tr>
<tr>
<td><code>--ethertype &lt;ethertype&gt;</code></td>
<td>List rules by the ethertype (ipv4 or ipv6)</td>
</tr>
<tr>
<td><code>--ingress</code></td>
<td>List rules applied to incoming network traffic</td>
</tr>
<tr>
<td><code>--egress</code></td>
<td>List rules applied to outgoing network traffic</td>
</tr>
<tr>
<td><code>--long</code></td>
<td><strong>deprecated</strong> this argument is no longer needed</td>
</tr>
</tbody>
</table>
Table 71.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 71.25. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 71.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

71.7. SECURITY GROUP RULE SHOW

Display security group rule details
Usage:

openstack security group rule show [-h]
   [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
   [--prefix PREFIX]
   [--max-width <integer>]
   [--fit-width] [--print-empty]
   <rule>

Table 71.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;rule&gt;</td>
<td>Security group rule to display (id only)</td>
</tr>
</tbody>
</table>

Table 71.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 71.30. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 71.31. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.32. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 71.33. Table formatter options
### 71.8. SECURITY GROUP SET

Set security group properties

**Usage:**

```bash
```

**Table 71.34. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Security group to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 71.35. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;new-name&gt;</td>
<td>New security group name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New security group description</td>
</tr>
<tr>
<td>--stateful</td>
<td>Security group is stateful (default)</td>
</tr>
<tr>
<td>--stateless</td>
<td>Security group is stateless</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the security group (repeat option to set multiple tags)</td>
</tr>
</tbody>
</table>
Clear tags associated with the security group. Specify both `--tag` and `--no-tag` to overwrite current tags.

### 71.9. SECURITY GROUP SHOW

Display security group details

**Usage:**

```
openstack security group show [-h] [-f {json,shell,table,value,yaml}] 
[cOLUMN] [--noindent] 
[--prefix PREFIX] [--max-width <integer>] 
[--fit-width] [--print-empty]
<group>
```

Table 71.36. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Security group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 71.37. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 71.38. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 71.39. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 71.40. Shell formatter options
Table 71.41. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

71.10. SECURITY GROUP UNSET

Unset security group properties

Usage:

```
openstack security group unset [-h] [--tag <tag> | --all-tag] <group>
```

Table 71.42. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;group&gt;</td>
<td>Security group to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 71.43. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the security group (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the security group</td>
</tr>
</tbody>
</table>
CHAPTER 72. SERVER

This chapter describes the commands under the server command.

72.1. SERVER ADD FIXED IP

Add fixed IP address to server

Usage:

```
openstack server add fixed ip [-h] [--fixed-ip-address <ip-address>] [--tag <tag>] <server> <network>
```

Table 72.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to receive the fixed ip address (name or id)</td>
</tr>
<tr>
<td>&lt;network&gt;</td>
<td>Network to allocate the fixed ip address from (name or ID)</td>
</tr>
</tbody>
</table>

Table 72.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--fixed-ip-address &lt;ip-address&gt;</td>
<td>Requested fixed ip address</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag for the attached interface. (supported by --os-compute-api-version 2.52 or above)</td>
</tr>
</tbody>
</table>

72.2. SERVER ADD FLOATING IP

Add floating IP address to server

Usage:

```
openstack server add floating ip [-h] [--fixed-ip-address <ip-address>] <server> <ip-address>
```

Table 72.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to receive the floating ip address (name or id)</td>
</tr>
</tbody>
</table>
**Table 72.4. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;ip-address&gt;</code></td>
<td>Floating ip address to assign to the first available server port (IP only)</td>
</tr>
</tbody>
</table>

**72.3. SERVER ADD NETWORK**

Add network to server

**Usage:**

```
openstack server add network [-h] [--tag <tag>] <server> <network>
```

**Table 72.5. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;server&gt;</code></td>
<td>Server to add the network to (name or id)</td>
</tr>
<tr>
<td><code>&lt;network&gt;</code></td>
<td>Network to add to the server (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.6. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--tag &lt;tag&gt;</code></td>
<td>Tag for the attached interface. (supported by <code>--os-compute-api-version 2.49 or above)</code></td>
</tr>
</tbody>
</table>

**72.4. SERVER ADD PORT**

Add port to server

**Usage:**

```
openstack server add port [-h] [-tag <tag>] <server> <port>
```
### 72.5. SERVER ADD SECURITY GROUP

Add security group to server.

**Usage:**

```
openstack server add security group [-h] <server> <group>
```

### Table 72.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>&lt;group&gt;</td>
<td>Security group to add (name or id)</td>
</tr>
</tbody>
</table>

### Table 72.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.6. SERVER ADD VOLUME

Add volume to server. Specify `--os-compute-api-version 2.20` or higher to add a volume to a server with status `SHELVED` or `SHELVED_OFFLOADED`.

**Usage:**
openstack server add volume [-h] [--device <device>] [--tag <tag>]  
[--enable-delete-on-termination | --disable-delete-on-termination]  
<server> <volume>

Table 72.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to add (name or id)</td>
</tr>
</tbody>
</table>

Table 72.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--device &lt;device&gt;</td>
<td>Server internal device name for volume</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag for the attached volume (supported by --os-compute-api-version 2.49 or above)</td>
</tr>
<tr>
<td>--enable-delete-on-termination</td>
<td>Delete the volume when the server is destroyed (supported by --os-compute-api-version 2.79 or above)</td>
</tr>
<tr>
<td>--disable-delete-on-termination</td>
<td>Do not delete the volume when the server is destroyed (supported by --os-compute-api-version 2.79 or above)</td>
</tr>
</tbody>
</table>

72.7. SERVER BACKUP CREATE

Create a server backup image

Usage:

openstack server backup create [-h] [-f {json,shell,table,value,yaml}]  
[-c COLUMN] [-noindent]  
[-prefix PREFIX]  
[[-max-width <integer>] [-fit-width]  
[[-print-empty] [-name <image-name>]  
[-type <backup-type>]  
[-rotate <count>] [-wait]  
<server>

Table 72.13. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to back up (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.14. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;image-name&gt;</td>
<td>Name of the backup image (default: server name)</td>
</tr>
<tr>
<td>--type &lt;backup-type&gt;</td>
<td>Used to populate the backup_type property of the backup image (default: empty)</td>
</tr>
<tr>
<td>--rotate &lt;count&gt;</td>
<td>Number of backups to keep (default: 1)</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for backup image create to complete</td>
</tr>
</tbody>
</table>

**Table 72.15. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 72.16. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 72.17. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 72.18. Table formatter options**
### 72.8. SERVER CREATE

Create a new server

**Usage:**

```bash
```

Table 72.19. Positional arguments
### Table 72.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;server-name&gt;</code></td>
<td>New server name</td>
</tr>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--flavor &lt;flavor&gt;</code></td>
<td>Create server with this flavor (name or ID)</td>
</tr>
<tr>
<td><code>--image &lt;image&gt;</code></td>
<td>Create server boot disk from this image (name or id)</td>
</tr>
<tr>
<td><code>--image-property &lt;key=value&gt;</code></td>
<td>Create server using the image that matches the specified property. Property must match exactly one property.</td>
</tr>
<tr>
<td><code>--volume &lt;volume&gt;</code></td>
<td>Create server using this volume as the boot disk (name or ID)</td>
</tr>
<tr>
<td></td>
<td>This option automatically creates a block device mapping with a boot index of 0. On many hypervisors (libvirt/kvm for example) this will be device vda. Do not create a duplicate mapping using <code>--block-device-mapping</code> for this volume.</td>
</tr>
<tr>
<td><code>--snapshot &lt;snapshot&gt;</code></td>
<td>Create server using this snapshot as the boot disk (name or ID)</td>
</tr>
<tr>
<td></td>
<td>This option automatically creates a block device mapping with a boot index of 0. On many hypervisors (libvirt/kvm for example) this will be device vda. Do not create a duplicate mapping using <code>--block-device-mapping</code> for this volume.</td>
</tr>
<tr>
<td><code>--boot-from-volume &lt;volume-size&gt;</code></td>
<td>When used in conjunction with the <code>--image</code> or <code>--image-property</code> option, this option automatically creates a block device mapping with a boot index of 0 and tells the compute service to create a volume of the given size (in GB) from the specified image and use it as the root disk of the server. The root volume will not be deleted when the server is deleted. This option is mutually exclusive with the <code>--volume</code> and <code>--snapshot</code> options.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--block-device-mapping &lt;dev-name=mapping&gt;</code></td>
<td><strong>deprecated</strong> create a block device on the server. Block device mapping in the format <code>&lt;dev-name&gt;=&lt;id&gt;:&lt;type&gt;:&lt;size(GB)&gt;({delete-on-terminate})</code> <code>&lt;dev-name&gt;</code>: block device name, like: vdb, xvdc (required) <code>&lt;id&gt;</code>: Name or ID of the volume, volume snapshot or image (required) <code>&lt;type&gt;</code>: volume, snapshot or image; default: volume (optional) <code>&lt;size(GB)&gt;</code>: volume size if create from image or snapshot (optional) <code>&lt;delete-on-terminate&gt;</code>: true or false; default: false (optional) Replaced by <code>--block-device</code></td>
</tr>
<tr>
<td><code>--block-device</code></td>
<td>Create a block device on the server. Either a path to a JSON file or a CSV-serialized string describing the block device mapping. The following keys are accepted for both: <code>uid=&lt;uuid&gt;</code>: UUID of the volume, snapshot or ID (required if using source image, snapshot or volume), <code>source_type=&lt;source_type&gt;</code>: source type (one of: image, snapshot, volume, blank), <code>destination_type=&lt;destination_type&gt;</code>: destination type (one of: volume, local) (optional), <code>disk_bus=&lt;disk_bus&gt;</code>: device bus (one of: uml, lxc, virtio, ...) (optional), <code>device_type=&lt;device_type&gt;</code>: device type (one of: disk, cdrom, etc. (optional), <code>device_name=&lt;device_name&gt;</code>: name of the device (optional), <code>volume_size=&lt;volume_size&gt;</code>: size of the block device in MiB (for swap) or GiB (for everything else) (optional), <code>guest_format=&lt;guest_format&gt;</code>: format of device (optional), <code>boot_index=&lt;boot_index&gt;</code>: index of disk used to order boot disk (required for volume-backed instances), `delete_on_termination=&lt;true</td>
</tr>
<tr>
<td><code>--swap &lt;swap&gt;</code></td>
<td>Create and attach a local swap block device of <code>&lt;swap_size&gt;</code> MiB.</td>
</tr>
<tr>
<td><code>--ephemeral &lt;size=size[,format=format]&gt;</code></td>
<td>Create and attach a local ephemeral block device of <code>&lt;size&gt;</code> GiB and format it to <code>&lt;format&gt;</code>.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--network &lt;network&gt;</code></td>
<td>Create a nic on the server and connect it to network. Specify option multiple times to create multiple NICs. This is a wrapper for the <code>--nic net-id=&lt;network&gt;</code> parameter that provides simple syntax for the standard use case of connecting a new server to a given network. For more advanced use cases, refer to the <code>--nic</code> parameter.</td>
</tr>
<tr>
<td><code>--port &lt;port&gt;</code></td>
<td>Create a nic on the server and connect it to port. Specify option multiple times to create multiple NICs. This is a wrapper for the <code>--nic port-id=&lt;port&gt;</code> parameter that provides simple syntax for the standard use case of connecting a new server to a given port. For more advanced use cases, refer to the <code>--nic</code> parameter.</td>
</tr>
<tr>
<td><code>--nic &lt;net-id=net-uuid,port-id=port-uuid,v4-fixed-ip=ip-addr,v6-fixed-ip=ip-addr,tag=tag,auto,none&gt;</code></td>
<td>Create a nic on the server. NIC in the format: net-id=&lt;net-uuid&gt;: attach NIC to network with this UUID, port-id=&lt;port-uuid&gt;: attach NIC to port with this UUID, v4-fixed-ip=&lt;ip-addr&gt;: IPv4 fixed address for NIC (optional), v6-fixed-ip=&lt;ip-addr&gt;: IPv6 fixed address for NIC (optional), tag: interface metadata tag (optional) (supported by <code>--os-compute-api-version 2.43</code> or above), none: (v2.37+) no network is attached, auto: (v2.37+) the compute service will automatically allocate a network. Specify option multiple times to create multiple NICs. Specifying a <code>--nic</code> of auto or none cannot be used with any other <code>--nic</code> value. Either net-id or port-id must be provided, but not both.</td>
</tr>
<tr>
<td><code>--password &lt;password&gt;</code></td>
<td>Set the password to this server</td>
</tr>
<tr>
<td><code>--security-group &lt;security-group&gt;</code></td>
<td>Security group to assign to this server (name or id) (repeat option to set multiple groups)</td>
</tr>
<tr>
<td><code>--key-name &lt;key-name&gt;</code></td>
<td>Keypair to inject into this server</td>
</tr>
<tr>
<td><code>--property &lt;key=value&gt;</code></td>
<td>Set a property on this server (repeat option to set multiple values)</td>
</tr>
<tr>
<td><code>--file &lt;dest-filename=source-filename&gt;</code></td>
<td>File(s) to inject into image before boot (repeat option to set multiple files)(supported by <code>--os-compute-api-version 2.57</code> or below)</td>
</tr>
<tr>
<td><code>--user-data &lt;user-data&gt;</code></td>
<td>User data file to serve from the metadata server</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set description for the server (supported by --os-compute-api-version 2.19 or above)</td>
</tr>
<tr>
<td>--availability-zone &lt;zone-name&gt;</td>
<td>Select an availability zone for the server. host and node are optional parameters. Availability zone in the format &lt;zone-name&gt;:&lt;host-name&gt;:&lt;node-name&gt;, &lt;zone-name&gt;:&lt;node-name&gt;, &lt;zone-name&gt;:&lt;host-name&gt; or &lt;zone-name&gt;</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>Requested host to create servers. (admin only) (supported by --os-compute-api-version 2.74 or above)</td>
</tr>
<tr>
<td>--hypervisor-hostname &lt;hypervisor-hostname&gt;</td>
<td>Requested hypervisor hostname to create servers. (admin only) (supported by --os-compute-api-version 2.74 or above)</td>
</tr>
<tr>
<td>--hint &lt;key=value&gt;</td>
<td>Hints for the scheduler</td>
</tr>
<tr>
<td>--use-config-drive</td>
<td>Enable config drive.</td>
</tr>
<tr>
<td>--no-config-drive</td>
<td>Disable config drive.</td>
</tr>
<tr>
<td>--config-drive &lt;config-drive-volume&gt;</td>
<td>True</td>
</tr>
<tr>
<td>--min &lt;count&gt;</td>
<td>Minimum number of servers to launch (default=1)</td>
</tr>
<tr>
<td>--max &lt;count&gt;</td>
<td>Maximum number of servers to launch (default=1)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tags for the server. specify multiple times to add multiple tags. (supported by --os-compute-api-version 2.52 or above)</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for build to complete</td>
</tr>
</tbody>
</table>

**Table 72.21. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
CHAPTER 72. SERVER

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 72.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 72.23. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 72.24. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.9. SERVER DELETE

Delete server(s)

Usage:

```
```

Table 72.25. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to delete (name or id)</td>
</tr>
</tbody>
</table>
### 72.10. SERVER DUMP CREATE

Create a dump file in server(s) Trigger crash dump in server(s) with features like kdump in Linux. It will create a dump file in the server(s) dumping the server(s)' memory, and also crash the server(s). OSC sees the dump file (server dump) as a kind of resource. This command requires `--os-compute-api-version` 2.17 or greater.

**Usage:**

```
openstack server dump create [-h] <server> [<server> ...]
```

Table 72.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to create dump file (name or id)</td>
</tr>
</tbody>
</table>

### 72.11. SERVER EVACUATE

Evacuate a server to a different host. This command is used to recreate a server after the host it was on has failed. It can only be used if the compute service that manages the server is down. This command should only be used by an admin after they have confirmed that the instance is not running on the failed host. If the server instance was created with an ephemeral root disk on non-shared storage the server will be rebuilt using the original glance image preserving the ports and any attached data volumes. If the server uses boot for volume or has its root disk on shared storage the root disk will be preserved and reused for the evacuated instance on the new host.

**Usage:**

Table 72.29. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for evacuation to complete</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>Set the preferred host on which to rebuild the evacuated server. The host will be validated by the scheduler. (supported by --os-compute-api-version 2.29 or above)</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Set the password on the evacuated instance. this option is mutually exclusive with the --shared-storage option</td>
</tr>
<tr>
<td>--shared-storage</td>
<td>Indicate that the instance is on shared storage. this will be auto-calculated with --os-compute-api-version 2.14 and greater and should not be used with later microversions. This option is mutually exclusive with the --password option</td>
</tr>
</tbody>
</table>

Table 72.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 72.32. JSON formatter options
### Table 72.33. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 72.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 72.12. SERVER EVENT LIST

List recent events of a server. Specify `--os-compute-api-version 2.21` or higher to show events for a deleted server, specified by ID only.

**Usage:**

```
```

### Table 72.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to list events (name or id)</td>
</tr>
</tbody>
</table>
### Table 72.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--changes-since &lt;changes-since&gt;</td>
<td>List only server events changed later or equal to a certain point of time. The provided time should be an ISO 8061 formatted time, e.g. <code>2016-03-04T06:27:59Z</code>. (supported with --os-compute-api-version 2.58 or above)</td>
</tr>
<tr>
<td>--changes-before &lt;changes-before&gt;</td>
<td>List only server events changed earlier or equal to a certain point of time. The provided time should be an ISO 8061 formatted time, e.g. <code>2016-03-04T06:27:59Z</code>. (supported with --os-compute-api-version 2.66 or above)</td>
</tr>
<tr>
<td>--marker MARKER</td>
<td>The last server event id of the previous page</td>
</tr>
<tr>
<td>--limit LIMIT</td>
<td>Maximum number of server events to display</td>
</tr>
</tbody>
</table>

### Table 72.37. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 72.38. CSV formatter options
Table 72.39. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 72.40. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.13. SERVER EVENT SHOW

Show server event details. Specify `--os-compute-api-version 2.21` or higher to show event details for a deleted server, specified by ID only. Specify `--os-compute-api-version 2.51` or higher to show event details for non-admin users.

Usage:

```
```

Table 72.41. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to show event details (name or id)</td>
</tr>
<tr>
<td>&lt;request-id&gt;</td>
<td>Request id of the event to show (id only)</td>
</tr>
</tbody>
</table>

Table 72.42. Command arguments
Table 72.43. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 72.44. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 72.45. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 72.46. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.14. SERVER GROUP CREATE

Create a new server group.

Usage:

Table 72.47. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New server group name</td>
</tr>
</tbody>
</table>

Table 72.48. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--policy &lt;policy&gt;</td>
<td>Add a policy to &lt;name&gt; specify --os-compute-api-version 2.15 or higher for the soft-affinity or soft-anti-affinity policy.</td>
</tr>
<tr>
<td>--rule &lt;key=value&gt;</td>
<td>A rule for the policy. currently, only the max_server_per_host rule is supported for the anti-affinity policy.</td>
</tr>
</tbody>
</table>

Table 72.49. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 72.50. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### 72.15. SERVER GROUP DELETE

Delete existing server group(s).

**Usage:**

```bash
openstack server group delete [-h] <server-group> [<server-group> ...]
```

**Table 72.53. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;server-group&gt;</code></td>
<td>Server group(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.54. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.16. SERVER GROUP LIST

List all server groups.

**Usage:**

```bash
openstack server group list [-h] [-f {csv, json, table, value, yaml}]  
[ -c COLUMN]  
[ --quote {all, minimal, none, nonnumeric}]
```
Table 72.55. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Display information from all projects (admin only)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--offset &lt;offset&gt;</td>
<td>Index from which to start listing servers. this should typically be a factor of --limit. Display all server groups if not specified.</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of server groups to display. if limit is greater than osapi_max_limit option of Nova API, osapi_max_limit will be used instead.</td>
</tr>
</tbody>
</table>

Table 72.56. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 72.57. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 72.58. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the JSON</td>
</tr>
</tbody>
</table>

Table 72.59. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.17. SERVER GROUP SHOW

Display server group details.

Usage:

```
```

Table 72.60. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server-group&gt;</td>
<td>Server group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 72.61. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 72.62. Output formatter options
### Table 72.63. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 72.64. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 72.65. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 72.18. SERVER IMAGE CREATE

Create a new server disk image from an existing server

**Usage:**

```bash
```
### Table 72.66. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to create image (name or id)</td>
</tr>
</tbody>
</table>

### Table 72.67. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;image-name&gt;</td>
<td>Name of new disk image (default: server name)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a new property to meta_data.json on the metadata server (repeat option to set multiple values)</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for operation to complete</td>
</tr>
</tbody>
</table>

### Table 72.68. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 72.69. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 72.70. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 72.71. Table formatter options
### 72.19. SERVER LIST

List servers

**Usage:**

```bash
cat
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 72.72. Command arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--reservation-id &lt;reservation-id&gt;</td>
<td>Only return instances that match the reservation</td>
</tr>
<tr>
<td>--ip &lt;ip-address-regex&gt;</td>
<td>Regular expression to match ip addresses</td>
</tr>
<tr>
<td>--ip6 &lt;ip-address-regex&gt;</td>
<td>Regular expression to match ipv6 addresses. Note that this option only applies for non-admin users when using <code>--os-compute-api-version</code> 2.5 or greater.</td>
</tr>
<tr>
<td>--name &lt;name-regex&gt;</td>
<td>Regular expression to match names</td>
</tr>
<tr>
<td>--instance-name &lt;server-name&gt;</td>
<td>Regular expression to match instance name (admin only)</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Search by server status</td>
</tr>
<tr>
<td>--flavor &lt;flavor&gt;</td>
<td>Search by flavor (name or id)</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>Search by image (name or id)</td>
</tr>
<tr>
<td>--host &lt;hostname&gt;</td>
<td>Search by hostname</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Include all projects (admin only) (can be specified using the ALL_PROJECTS envvar)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Search by project (admin only) (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Search by user (name or id) (admin only before microversion 2.83)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--deleted</td>
<td>Only display deleted servers (admin only)</td>
</tr>
<tr>
<td>--availability-zone AVAILABILITY_ZONE</td>
<td>Search by availability zone (admin only before microversion 2.83)</td>
</tr>
<tr>
<td>--key-name KEY_NAME</td>
<td>Search by keypair name (admin only before microversion 2.83)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--config-drive</code></td>
<td>Only display servers with a config drive attached (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--no-config-drive</code></td>
<td>Only display servers without a config drive attached (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--progress PROGRESS</code></td>
<td>Search by progress value (%) (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--vm-state &lt;state&gt;</code></td>
<td>Search by vm_state value (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--task-state &lt;state&gt;</code></td>
<td>Search by task_state value (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--power-state &lt;state&gt;</code></td>
<td>Search by power_state value (admin only before microversion 2.83)</td>
</tr>
<tr>
<td><code>--long</code></td>
<td>List additional fields in output</td>
</tr>
<tr>
<td><code>-n, --no-name-lookup</code></td>
<td>Skip flavor and image name lookup. mutually exclusive with &quot;--name-lookup-one-by-one&quot; option.</td>
</tr>
<tr>
<td><code>--name-lookup-one-by-one</code></td>
<td>When looking up flavor and image names, look them up one by one as needed instead of all together (default). Mutually exclusive with &quot;--no-name-lookup</td>
</tr>
<tr>
<td><code>--marker &lt;server&gt;</code></td>
<td>The last server of the previous page. display list of servers after marker. Display all servers if not specified. When used with <code>--deleted</code>, the marker must be an ID, otherwise a name or ID can be used.</td>
</tr>
<tr>
<td><code>--limit &lt;num-servers&gt;</code></td>
<td>Maximum number of servers to display. if limit equals -1, all servers will be displayed. If limit is greater than <code>osapi_max_limit</code> option of Nova API, <code>osapi_max_limit</code> will be used instead.</td>
</tr>
<tr>
<td><code>--changes-before &lt;changes-before&gt;</code></td>
<td>List only servers changed before a certain point of time. The provided time should be an ISO 8061 formatted time (e.g., 2016-03-05T06:27:59Z). (supported by <code>--os-compute-api-version 2.66 or above</code>)</td>
</tr>
<tr>
<td><code>--changes-since &lt;changes-since&gt;</code></td>
<td>List only servers changed after a certain point of time. The provided time should be an ISO 8061 formatted time (e.g., 2016-03-04T06:27:59Z).</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--locked</td>
<td>Only display locked servers (supported by --os-compute-api-version 2.73 or above)</td>
</tr>
<tr>
<td>--unlocked</td>
<td>Only display unlocked servers (supported by --os-compute-api-version 2.73 or above)</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;</td>
<td>Only list servers with the specified tag. specify multiple times to filter on multiple tags. (supported by --os-compute-api-version 2.26 or above)</td>
</tr>
<tr>
<td>--not-tags &lt;tag&gt;</td>
<td>Only list servers without the specified tag. specify multiple times to filter on multiple tags. (supported by --os-compute-api-version 2.26 or above)</td>
</tr>
</tbody>
</table>

Table 72.73. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 72.74. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 72.75. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 72.76. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.20. SERVER LOCK

Lock server(s). A non-admin user will not be able to execute actions

Usage:

    openstack server lock [-h] [--reason <reason>] <server> [<server> ...]

Table 72.77. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to lock (name or id)</td>
</tr>
</tbody>
</table>

Table 72.78. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--reason &lt;reason&gt;</td>
<td>Reason for locking the server(s). requires <code>--os-compute-api-version</code> 2.73 or greater.</td>
</tr>
</tbody>
</table>

72.21. SERVER MIGRATE CONFIRM

DEPRECATED: Confirm server migration. Use server migration confirm instead.

Usage:

    openstack server migrate confirm [-h] <server>

Table 72.79. Positional arguments
### 72.22. SERVER MIGRATE REVERT

Revert server migration. Use `server migration revert` instead.

Usage:

```
openstack server migrate revert [-h] <server>
```

Table 72.81. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.82. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.23. SERVER MIGRATE

Migrate server to different host. A migrate operation is implemented as a resize operation using the same flavor as the old server. This means that, like resize, migrate works by creating a new server using the same flavor and copying the contents of the original disk into a new one. As with resize, the migrate operation is a two-step process for the user: the first step is to perform the migrate, and the second step is to either confirm (verify) success and release the old server, or to declare a revert to release the new server and restart the old one.

Usage:

```
openstack server migrate [-h] [--live-migration] [--host <hostname>]
                            [--shared-migration | --block-migration]
                            [--disk-overcommit | --no-disk-overcommit]
                            [--wait]
                            <server>
```

Table 72.83. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>
Table 72.84. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--live-migration</td>
<td>Live migrate the server; use the <code>--host</code> option to specify a target host for the migration which will be validated by the scheduler</td>
</tr>
<tr>
<td>--host &lt;hostname&gt;</td>
<td>Migrate the server to the specified host. (supported with <code>--os-compute-api-version</code> 2.30 or above when used with the <code>--live-migration</code> option) (supported with <code>--os-compute-api-version</code> 2.56 or above when used without the <code>--live-migration</code> option)</td>
</tr>
<tr>
<td>--shared-migration</td>
<td>Perform a shared live migration (default before <code>--os-compute-api-version</code> 2.25, auto after)</td>
</tr>
<tr>
<td>--block-migration</td>
<td>Perform a block live migration (auto-configured from <code>--os-compute-api-version</code> 2.25)</td>
</tr>
<tr>
<td>--disk-overcommit</td>
<td>Allow disk over-commit on the destination host(supported with <code>--os-compute-api-version</code> 2.24 or below)</td>
</tr>
<tr>
<td>--no-disk-overcommit</td>
<td>Do not over-commit disk on the destination host (default)(supported with <code>--os-compute-api-version</code> 2.24 or below)</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for migrate to complete</td>
</tr>
</tbody>
</table>

**72.24. SERVER MIGRATION ABORT**

Cancel an ongoing live migration. This command requires `--os-compute-api-version` 2.24 or greater.

**Usage:**

```
openstack server migration abort [-h] <server> <migration>
```

Table 72.85. Positional arguments
### 72.25. SERVER MIGRATION CONFIRM

Confirm server migration. Confirm (verify) success of the migration operation and release the old server.

**Usage:**

```bash
openstack server migration confirm [-h] <server>
```

**Table 72.87. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>&lt;migration&gt;</td>
<td>Migration (id)</td>
</tr>
</tbody>
</table>

Table 72.90. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.27. SERVER MIGRATION LIST

List server migrations

Usage:

```

Table 72.91. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--server &lt;server&gt;</td>
<td>Filter migrations by server (name or id)</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>Filter migrations by source or destination host</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Filter migrations by status</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>Filter migrations by type</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--marker &lt;marker&gt;</td>
<td>The last migration of the previous page; displays list of migrations after marker. Note that the marker is the migration UUID. (supported with --os-compute-api-version 2.59 or above)</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of migrations to display. Note that there is a configurable max limit on the server, and the limit that is used will be the minimum of what is requested here and what is configured in the server. (supported with --os-compute-api-version 2.59 or above)</td>
</tr>
<tr>
<td>--changes-since &lt;changes-since&gt;</td>
<td>List only migrations changed later or equal to a certain point of time. The provided time should be an ISO 8061 formatted time, e.g. <code>2016-03-04T06:27:59Z</code>. (supported with --os-compute-api-version 2.59 or above)</td>
</tr>
<tr>
<td>--changes-before &lt;changes-before&gt;</td>
<td>List only migrations changed earlier or equal to a certain point of time. The provided time should be an ISO 8061 formatted time, e.g. <code>2016-03-04T06:27:59Z</code>. (supported with --os-compute-api-version 2.66 or above)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Filter migrations by project (name or id) (supported with --os-compute-api-version 2.80 or above)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Filter migrations by user (name or id) (supported with --os-compute-api-version 2.80 or above)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
</tbody>
</table>

Table 72.92. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 72.93. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 72.94. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 72.95. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.28. SERVER MIGRATION REVERT

Revert server migration. Revert the migration operation. Release the new server and restart the old one.

Usage:

```
openstack server migration revert [-h] <server>
```

Table 72.96. Positional arguments
72.29. SERVER MIGRATION SHOW

Show a migration for a given server.

Usage:

openstack server migration show [-h] <server> <migration>

Table 72.98. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>&lt;migration&gt;</td>
<td>Migration (id)</td>
</tr>
</tbody>
</table>

72.30. SERVER PAUSE

Pause server(s)

Usage:

openstack server pause [-h] <server> [<server> ...]

Table 72.100. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to pause (name or id)</td>
</tr>
</tbody>
</table>
Table 72.101. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.31. SERVER REBOOT

Perform a hard or soft server reboot

**Usage:**

```
openstack server reboot [-h] [--hard | --soft] [--wait] <server>
```

Table 72.102. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.103. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--hard</td>
<td>Perform a hard reboot</td>
</tr>
<tr>
<td>--soft</td>
<td>Perform a soft reboot</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for reboot to complete</td>
</tr>
</tbody>
</table>

### 72.32. SERVER REBUILD

Rebuild server

**Usage:**

```
```
Table 72.104. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.105. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>Recreate server from the specified image (name or ID). Defaults to the currently used one.</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set the new name of the rebuilt server</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Set the password on the rebuilt server</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a new property on the rebuilt server (repeat option to set multiple values)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set a new description on the rebuilt server (supported by --os-compute-api-version 2.19 or above)</td>
</tr>
<tr>
<td>--preserve-ephemeral</td>
<td>Preserve the default ephemeral storage partition on rebuild.</td>
</tr>
<tr>
<td>--no-preserve-ephemeral</td>
<td>Do not preserve the default ephemeral storage partition on rebuild.</td>
</tr>
<tr>
<td>--key-name &lt;key-name&gt;</td>
<td>Set the key name of key pair on the rebuilt server. Cannot be specified with the --key-unset option. (supported by --os-compute-api-version 2.54 or above)</td>
</tr>
<tr>
<td>--no-key-name</td>
<td>Unset the key name of key pair on the rebuilt server. Cannot be specified with the --key-name option. (supported by --os-compute-api-version 2.54 or above)</td>
</tr>
<tr>
<td>--user-data &lt;user-data&gt;</td>
<td>Add a new user data file to the rebuilt server. cannot be specified with the --no-user-data option. (supported by --os-compute-api-version 2.57 or above)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--no-user-data</td>
<td>Remove existing user data when rebuilding server. Cannot be specified with the --user-data option. (supported by --os-compute-api-version 2.57 or above)</td>
</tr>
<tr>
<td>--trusted-image-cert &lt;trusted-cert-id&gt;</td>
<td>Trusted image certificate ids used to validate certificates during the image signature verification process. Defaults to env[OS_TRUSTED_IMAGE_CERTIFICATE_IDS]. May be specified multiple times to pass multiple trusted image certificate IDs. Cannot be specified with the --no-trusted-certs option. (supported by --os-compute-api-version 2.63 or above)</td>
</tr>
<tr>
<td>--no-trusted-image-certs</td>
<td>Remove any existing trusted image certificates from the server. Cannot be specified with the --trusted-certs option. (supported by --os-compute-api-version 2.63 or above)</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for rebuild to complete</td>
</tr>
</tbody>
</table>

**Table 72.106. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 72.107. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 72.108. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 72.109. Table formatter options**
### 72.33. SERVER REMOVE FIXED IP

Remove fixed IP address from server

**Usage:**

```bash
openstack server remove fixed ip [-h] <server> <ip-address>
```

**Table 72.110. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to remove the fixed ip address from (name or id)</td>
</tr>
<tr>
<td>&lt;ip-address&gt;</td>
<td>Fixed ip address to remove from the server (ip only)</td>
</tr>
</tbody>
</table>

**Table 72.111. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.34. SERVER REMOVE FLOATING IP

Remove floating IP address from server

**Usage:**

```bash
openstack server remove floating ip [-h] <server> <ip-address>
```

**Table 72.112. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to remove the floating ip address from (name or id)</td>
</tr>
<tr>
<td>&lt;ip-address&gt;</td>
<td>Floating ip address to remove from the server (ip only)</td>
</tr>
</tbody>
</table>
### Table 72.113. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to remove the floating ip address from (name or id)</td>
</tr>
<tr>
<td>&lt;ip-address&gt;</td>
<td>Floating ip address to remove from server (ip only)</td>
</tr>
</tbody>
</table>

### 72.35. SERVER REMOVE NETWORK

Remove all ports of a network from server

**Usage:**

```
openstack server remove network [-h] <server> <network>
```

#### Table 72.114. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server to remove the port from (name or id)</td>
</tr>
<tr>
<td>&lt;network&gt;</td>
<td>Network to remove from the server (name or id)</td>
</tr>
</tbody>
</table>

#### Table 72.115. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.36. SERVER REMOVE PORT

Remove port from server

**Usage:**

```
openstack server remove port [-h] <server> <port>
```

#### Table 72.116. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
72.37. SERVER REMOVE SECURITY GROUP

Remove security group from server

Usage:

```
openstack server remove security group [-h] <server> <group>
```

Table 72.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Name or id of server to use</td>
</tr>
<tr>
<td>&lt;group&gt;</td>
<td>Name or id of security group to remove from server</td>
</tr>
</tbody>
</table>

Table 72.119. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

72.38. SERVER REMOVE VOLUME

Remove volume from server. Specify `--os-compute-api-version 2.20` or higher to remove a volume from a server with status `SHELVED` or `SHELVED_OFFLOADED`.

Usage:

```
openstack server remove volume [-h] <server> <volume>
```

Table 72.120. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Name or id of server to use</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to remove from server</td>
</tr>
</tbody>
</table>
### Table 72.121. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to remove (name or id)</td>
</tr>
</tbody>
</table>

### 72.39. SERVER RESCUE

Put server in rescue mode

**Usage:**

```
openstack server rescue [-h] [--image <image>] [--password <password>] <server>
```

### Table 72.122. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

### Table 72.123. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>Image (name or id) to use for the rescue mode. Defaults to the currently used one.</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Set the password on the rescued instance</td>
</tr>
</tbody>
</table>

### 72.40. SERVER RESIZE CONFIRM

Confirm server resize. Confirm (verify) success of resize operation and release the old server.

**Usage:**

```
openstack server resize confirm [-h] <server>
```
### 72.41. SERVER RESIZE REVERT

Revert server resize. Revert the resize operation. Release the new server and restart the old one.

**Usage:**

```
openstack server resize revert [-h] <server>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

### 72.42. SERVER RESIZE

Scale server to a new flavor. A resize operation is implemented by creating a new server and copying the contents of the original disk into a new one. It is a two-step process for the user: the first step is to perform the resize, and the second step is to either confirm (verify) success and release the old server or to declare a revert to release the new server and restart the old one.

**Usage:**

```
openstack server resize [-h] [--flavor <flavor> | --confirm | --revert] [--wait] <server>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### 72.43. SERVER RESTORE

Restore server(s)

**Usage:**

```
openstack server restore [-h] <server> [<server> ...]
```

**Table 72.130. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to restore (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.131. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.44. SERVER RESUME

Resume server(s)

**Usage:**

```
openstack server resume [-h] <server> [<server> ...]
```
Table 72.132. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to resume (name or id)</td>
</tr>
</tbody>
</table>

Table 72.133. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

72.45. SERVER SET

Set server properties

Usage:

```
openstack server set [-h] [--name <new-name>]            
                        [--password PASSWORD | --no-password] 
                        [--property <key=value>] [--state <state>] 
                        [--description <description>] [--tag <tag>] 
                        <server>
```

Table 72.134. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.135. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;new-name&gt;</td>
<td>New server name</td>
</tr>
<tr>
<td>--password PASSWORD</td>
<td>Set the server password</td>
</tr>
<tr>
<td>--no-password</td>
<td>Clear the admin password for the server from the metadata service; note that this action does not actually change the server password</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to add/change for this server (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--state &lt;state&gt;</td>
<td>New server state (valid value: active, error)</td>
</tr>
</tbody>
</table>
72.46. SERVER SHELVE

Shelve and optionally offload server(s). Shelving a server creates a snapshot of the server and stores this snapshot before shutting down the server. This shelved server can then be offloaded or deleted from the host, freeing up remaining resources on the host, such as network interfaces. Shelved servers can be unshelved, restoring the server from the snapshot. Shelving is therefore useful where users wish to retain the UUID and IP of a server, without utilizing other resources or disks. Most clouds are configured to automatically offload shelved servers immediately or after a small delay. For clouds where this is not configured, or where the delay is larger, offloading can be manually specified. This is an admin-only operation by default.

Usage:

```
openstack server shelve [-h] [--offload] [--wait] <server> [<server> ...]
```

Table 72.136. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to shelve (name or id)</td>
</tr>
</tbody>
</table>

Table 72.137. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--offload</td>
<td>Remove the shelved server(s) from the host (admin only). Invoking this option on an unshelved server(s) will result in the server being shelved first</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for shelve and/or offload operation to complete</td>
</tr>
</tbody>
</table>

72.47. SERVER SHOW

Show server details. Specify `--os-compute-api-version 2.47` or higher to see the embedded flavor information for the server.

Usage:

Table 72.138. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.139. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--diagnostics</td>
<td>Display server diagnostics information</td>
</tr>
<tr>
<td>--topology</td>
<td>Include topology information in the output (supported by --os-compute-api-version 2.78 or above)</td>
</tr>
</tbody>
</table>

Table 72.140. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 72.141. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 72.142. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 72.143. Table formatter options
--max-width <integer>

Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width

Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.

--print-empty

Print empty table if there is no data to show.

72.48. SERVER SSH

SSH to server

Usage:

openstack server ssh [-h] [--login <login-name>] [--port <port>]
                  [--identity <keyfile>] [--option <config-options>]
                  [--4 | --6]
                  [--public | --private | --address-type <address-type>]
                  <server>

Table 72.144. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.145. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--login &lt;login-name&gt;</td>
<td>Login name (ssh -l option)</td>
</tr>
<tr>
<td>--port &lt;port&gt;</td>
<td>Destination port (ssh -p option)</td>
</tr>
<tr>
<td>--identity &lt;keyfile&gt;</td>
<td>Private key file (ssh -i option)</td>
</tr>
<tr>
<td>--option &lt;config-options&gt;</td>
<td>Options in ssh_config(5) format (ssh -o option)</td>
</tr>
<tr>
<td>-4</td>
<td>Use only ipv4 addresses</td>
</tr>
<tr>
<td>-6</td>
<td>Use only ipv6 addresses</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>--public</td>
<td>Use public ip address</td>
</tr>
<tr>
<td>--private</td>
<td>Use private ip address</td>
</tr>
<tr>
<td>--address-type &lt;address-type&gt;</td>
<td>Use other ip address (public, private, etc)</td>
</tr>
</tbody>
</table>

### 72.49. SERVER START

Start server(s).

**Usage:**

```
openstack server start [-h] [--all-projects] <server> [<server> ...]
```

**Table 72.146. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to start (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.147. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Start server(s) in another project by name (admin only)(can be specified using the ALL_PROJECTS envvar)</td>
</tr>
</tbody>
</table>

### 72.50. SERVER STOP

Stop server(s).

**Usage:**

```
openstack server stop [-h] [--all-projects] <server> [<server> ...]
```

**Table 72.148. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to stop (name or id)</td>
</tr>
</tbody>
</table>
Table 72.149. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Stop server(s) in another project by name (admin only)(can be specified using the ALL_PROJECTS envvar)</td>
</tr>
</tbody>
</table>

72.51. SERVER SUSPEND

Suspend server(s)

Usage:

```
openstack server suspend [-h] <server> [<server> ...]
```

Table 72.150. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to suspend (name or id)</td>
</tr>
</tbody>
</table>

Table 72.151. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

72.52. SERVER UNLOCK

Unlock server(s)

Usage:

```
openstack server unlock [-h] <server> [<server> ...]
```

Table 72.152. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to unlock (name or id)</td>
</tr>
</tbody>
</table>

Table 72.153. Command arguments
### 72.53. SERVER UNPAUSE

Unpause server(s)

**Usage:**
```
openstack server unpause [-h] <server> [<server> ...]
```

**Table 72.154. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;server&gt;</code></td>
<td>Server(s) to unpause (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.155. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.54. SERVER UNRESCUE

Restore server from rescue mode

**Usage:**
```
openstack server unrescue [-h] <server>
```

**Table 72.156. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;server&gt;</code></td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

**Table 72.157. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 72.55. SERVER UNSET
Unset server properties and tags

Usage:

```
openstack server unset [-h] [--property <key>] [--description] [--tag <tag>] <server>
```

Table 72.158. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server (name or id)</td>
</tr>
</tbody>
</table>

Table 72.159. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property key to remove from server (repeat option to remove multiple values)</td>
</tr>
<tr>
<td>--description</td>
<td>Unset server description (supported by --os-compute-api- version 2.19 or above)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to remove from the server. specify multiple times to remove multiple tags. (supported by --os-compute-api- version 2.26 or above)</td>
</tr>
</tbody>
</table>

72.56. SERVER UNSHELVE

Unshelve server(s)

Usage:

```
openstack server unshelve [-h] [--availability-zone AVAILABILITY_ZONE] [--wait] <server> [<server> ...]
```

Table 72.160. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Server(s) to unshelve (name or id)</td>
</tr>
</tbody>
</table>
72.57. SERVER VOLUME LIST

List all the volumes attached to a server.

Usage:


Table 72.162. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>Server to list volume attachments for (name or id)</td>
</tr>
</tbody>
</table>

Table 72.163. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 72.164. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format csv,json,table,value,yaml</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 72.165. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 72.166. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 72.167. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

72.58. SERVER VOLUME UPDATE

Update a volume attachment on the server.

Usage:

```bash
openstack server volume update [-h] [--delete-on-termination | --preserve-on-termination] server volume
```

Table 72.168. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>server</td>
<td>Server to update volume for (name or id)</td>
</tr>
<tr>
<td>volume</td>
<td>Volume (id)</td>
</tr>
</tbody>
</table>

Table 72.169. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--delete-on-termination</td>
<td>Delete the volume when the server is destroyed (supported by --os-compute-api-version 2.85 or above)</td>
</tr>
<tr>
<td>--preserve-on-termination</td>
<td>Preserve the volume when the server is destroyed (supported by --os-compute-api-version 2.85 or above)</td>
</tr>
</tbody>
</table>
CHAPTER 73. SERVICE

This chapter describes the commands under the `service` command.

73.1. SERVICE CREATE

Create new service

Usage:

```
```

Table 73.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;type&gt;</code></td>
<td>New service type (compute, image, identity, volume, etc)</td>
</tr>
</tbody>
</table>

Table 73.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>New service name</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>New service description</td>
</tr>
<tr>
<td><code>--enable</code></td>
<td>Enable service (default)</td>
</tr>
<tr>
<td><code>--disable</code></td>
<td>Disable service</td>
</tr>
</tbody>
</table>

Table 73.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference

1062
Table 73.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 73.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 73.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

73.2. SERVICE DELETE

Delete service(s)

Usage:

openstack service delete [-h] <service> [<service> ...]

Table 73.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Service(s) to delete (type, name or id)</td>
</tr>
</tbody>
</table>

Table 73.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
73.3. SERVICE LIST

List services

Usage:

```
openstack service list [-h] [-f {csv,json,table,value,yaml}]
    [-c COLUMN]
    [--quote {all,minimal,none,nonnumeric}]
    [--noindent] [-w <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending|--sort-descending] [--long]
```

Table 73.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 73.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 73.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 73.12. JSON formatter options
Table 73.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

73.4. SERVICE PROVIDER CREATE

Create new service provider

Usage:

```sh
```

Table 73.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New service provider name (must be unique)</td>
</tr>
</tbody>
</table>

Table 73.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--auth-url &lt;auth-url&gt;</td>
<td>Authentication url of remote federated service provider (required)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New service provider description</td>
</tr>
<tr>
<td>--service-provider-url &lt;sp-url&gt;</td>
<td>A service url where saml assertions are being sent (required)</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the service provider (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the service provider</td>
</tr>
</tbody>
</table>

**Table 73.16. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 73.17. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 73.18. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 73.19. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
73.5. SERVICE PROVIDER DELETE

Delete service provider(s)

Usage:

openstack service provider delete [-h] <service-provider> ...

Table 73.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-provider&gt;</td>
<td>Service provider(s) to delete</td>
</tr>
</tbody>
</table>

Table 73.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

73.6. SERVICE PROVIDER LIST

List service providers

Usage:


Table 73.22. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 73.23. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 73.24. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 73.25. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 73.26. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 73.7. SERVICE PROVIDER SET

Set service provider properties

**Usage:**

```
openstack service provider set [-h] [--auth-url <auth-url>]
   [-description <description>]
   [--service-provider-url <sp-url>]
   [--enable | --disable]
   <service-provider>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

#### Table 73.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-provider&gt;</td>
<td>Service provider to modify</td>
</tr>
</tbody>
</table>

#### Table 73.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--auth-url &lt;auth-url&gt;</td>
<td>New authentication url of remote federated service provider</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New service provider description</td>
</tr>
<tr>
<td>--service-provider-url &lt;sp-url&gt;</td>
<td>New service provider url, where saml assertions are sent</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable the service provider</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable the service provider</td>
</tr>
</tbody>
</table>

### 73.8. SERVICE PROVIDER SHOW

Display service provider details

**Usage:**

```
openstack service provider show [-h] [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent]
```
Table 73.29. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-provider&gt;</td>
<td>Service provider to display</td>
</tr>
</tbody>
</table>

Table 73.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 73.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 73.32. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 73.33. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 73.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
73.9. SERVICE SET

Set service properties

Usage:

openstack service set [-h] [-type <type>] [-name <service-name>]
[-description <description>]
[-enable | --disable]
<service>

Table 73.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Service to modify (type, name or id)</td>
</tr>
</tbody>
</table>

Table 73.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--type &lt;type&gt;</td>
<td>New service type (compute, image, identity, volume, etc)</td>
</tr>
<tr>
<td>--name &lt;service-name&gt;</td>
<td>New service name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New service description</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable service</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable service</td>
</tr>
</tbody>
</table>
73.10. SERVICE SHOW

Display service details

Usage:

```
openstack service show [-h] [-f {json,shell,table,value,yaml}]
   [-c COLUMN] [--noindent] [-p prefix PREFIX]
   [--max-width <integer>] [--fit-width]
   [-p print-empty]
   <service>
```

Table 73.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service&gt;</td>
<td>Service to display (type, name or id)</td>
</tr>
</tbody>
</table>

Table 73.38. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 73.39. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 73.40. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 73.41. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 73.42. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 74. SFC

This chapter describes the commands under the `sfc` command.

74.1. SFC FLOW CLASSIFIER CREATE

Create a flow classifier

Usage:

```
openstack sfc flow classifier create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--description <description>]
[--protocol <protocol>]
[--ethertype {IPv4,IPv6}]
[--source-port <min-port>:<max-port>]
[--destination-port <min-port>:<max-port>]
[--source-ip-prefix <source-ip-prefix>]
[--destination-ip-prefix <destination-ip-prefix>]
[--logical-source-port <logical-source-port>]
[--logical-destination-port <logical-destination-port>]
[--l7-parameters L7_PARAMETERS]

<name>
```

<table>
<thead>
<tr>
<th>Table 74.1. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>&lt;name&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 74.2. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
</tr>
<tr>
<td>--protocol &lt;protocol&gt;</td>
</tr>
<tr>
<td>--ethertype {IPv4,IPv6}</td>
</tr>
<tr>
<td>--source-port &lt;min-port&gt;:&lt;max-port&gt;</td>
</tr>
</tbody>
</table>
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--destination-port &lt;min-port&gt;:&lt;max-port&gt;</td>
<td>Destination protocol port (allowed range ([1,65535]). Must be specified as a:b, where a=min-port and b=max-port) in the allowed range.</td>
</tr>
<tr>
<td>--source-ip-prefix &lt;source-ip-prefix&gt;</td>
<td>Source ip address in cidr notation</td>
</tr>
<tr>
<td>--destination-ip-prefix &lt;destination-ip-prefix&gt;</td>
<td>Destination ip address in cidr notation</td>
</tr>
<tr>
<td>--logical-source-port &lt;logical-source-port&gt;</td>
<td>Neutron source port (name or id)</td>
</tr>
<tr>
<td>--logical-destination-port &lt;logical-destination-port&gt;</td>
<td>Neutron destination port (name or id)</td>
</tr>
<tr>
<td>--l7-parameters L7_PARAMETERS</td>
<td>Dictionary of l7 parameters. currently, no value is supported for this option.</td>
</tr>
</tbody>
</table>

**Table 74.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 74.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 74.5. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 74.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 74.2. SFC FLOW CLASSIFIER DELETE

Delete a given flow classifier

**Usage:**

```bash
openstack sfc flow classifier delete [-h] <flow-classifier>
```

**Table 74.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;flow-classifier&gt;</td>
<td>Flow classifier to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 74.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 74.3. SFC FLOW CLASSIFIER LIST

List flow classifiers

**Usage:**

```bash
openstack sfc flow classifier list [-h]
    [-f {csv, json, table, value, yaml}]
    [-c COLUMN]
    [--quote {all, minimal, none, nonnumeric}]
    [--noindent] [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--sort-column SORT_COLUMN]
    [--sort-ascending | --sort-descending]
    [--long]
```

**Table 74.9. Command arguments**
### Table 74.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

### Table 74.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 74.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 74.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
74.4. SFC FLOW CLASSIFIER SET

Set flow classifier properties

Usage:

```
openstack sfc flow classifier set [-h] [--name <name>] [--description <description>] <flow-classifier>
```

Table 74.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flow-classifier&gt;</code></td>
<td>Flow classifier to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 74.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of the flow classifier</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the flow classifier</td>
</tr>
</tbody>
</table>

74.5. SFC FLOW CLASSIFIER SHOW

Display flow classifier details

Usage:

```
```
### Table 74.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;flow-classifier&gt;</code></td>
<td>Flow classifier to display (name or id)</td>
</tr>
</tbody>
</table>

### Table 74.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 74.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 74.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 74.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 74.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
74.6. SFC PORT CHAIN CREATE

Create a port chain

Usage:

openstack sfc port chain create [-h] [-f {json,shell,table,value,yaml}] 
  [-c COLUMN] [--noindent] 
  [--prefix PREFIX] 
  [--max-width <integer>] [--fit-width] 
  [--print-empty] 
  [--description <description>] 
  [--flow-classifier <flow-classifier>] 
  [--chain-parameters correlation=<correlation-type>,symmetric=<boolean>] 
  --port-pair-group <port-pair-group> 
  <name>

Table 74.22. Positional arguments

Value | Summary
---|---
=name> | Name of the port chain

Table 74.23. Command arguments

Value | Summary
---|---
-h, --help | Show this help message and exit
--description <description> | Description for the port chain
--flow-classifier <flow-classifier> | Add flow classifier (name or id). this option can be repeated.
--chain-parameters correlation=<correlation-type>,symmetric=<boolean> | Dictionary of chain parameters. supports correlation=(mpls|nsh) (default is mpls) and symmetric=(true|false).
--port-pair-group <port-pair-group> | Add port pair group (name or id). this option can be repeated.

Table 74.24. Output formatter options

---

--print-empty | Print empty table if there is no data to show.
### 74.7. SFC PORT CHAIN DELETE

Delete a given port chain

**Usage:**
```
openstack sfc port chain delete [-h] <port-chain>
```

**Table 74.28. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-chain&gt;</td>
<td>Port chain to delete (name or id)</td>
</tr>
</tbody>
</table>
### 74.8. SFC PORT CHAIN LIST

List port chains

**Usage:**

```bash
```

**Table 74.30. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

**Table 74.31. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td></td>
</tr>
</tbody>
</table>
Table 74.32. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 74.33. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

74.9. SFC PORT CHAIN SET

Set port chain properties

Usage:

```
openstack sfc port chain set [-h] [--name <name>]
  [--description <description>]
  [--flow-classifier <flow-classifier>]
  [--no-flow-classifier]
  [--port-pair-group <port-pair-group>]
  [--no-port-pair-group]
  <port-chain>
```

Table 74.35. Positional arguments
### Table 74.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-chain&gt;</td>
<td>Port chain to modify (name or id)</td>
</tr>
</tbody>
</table>

### Table 74.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-chain&gt;</td>
<td>Port chain to display (name or id)</td>
</tr>
</tbody>
</table>

### 74.10. SFC PORT CHAIN SHOW

Display port chain details

**Usage:**

```
```

**Table 74.38. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-chain&gt;</td>
<td>Port chain to modify (name or id)</td>
</tr>
</tbody>
</table>
Value | Summary
---|---
-h, --help | Show this help message and exit

### Table 74.39. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>(-f {\text{json,shell,table,value,yaml}}, --format {\text{json,shell,table,value,yaml}})</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>(-c \text{COLUMN}, --column \text{COLUMN})</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 74.40. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 74.41. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 74.42. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 74.11. SFC PORT CHAIN UNSET

Unset port chain properties

**Usage:**
Table 74.43. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-chain&gt;</td>
<td>Port chain to unset (name or id)</td>
</tr>
</tbody>
</table>

Table 74.44. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--flow-classifier &lt;flow-classifier&gt;</td>
<td>Remove flow classifier(s) from the port chain (name or ID). This option can be repeated.</td>
</tr>
<tr>
<td>--all-flow-classifier</td>
<td>Remove all flow classifiers from the port chain</td>
</tr>
<tr>
<td>--port-pair-group &lt;port-pair-group&gt;</td>
<td>Remove port pair group(s) from the port chain (name or ID). This option can be repeated.</td>
</tr>
</tbody>
</table>

74.12. SFC PORT PAIR CREATE

Create a port pair

Usage:

```
```

Table 74.45. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the port pair</td>
</tr>
</tbody>
</table>

Table 74.46. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the port pair</td>
</tr>
<tr>
<td>--service-function-parameters correlation= &lt;correlation-type&gt;,weight=&lt;weight&gt;</td>
<td>Dictionary of service function parameters. currently, correlation=(None</td>
</tr>
<tr>
<td>--ingress &lt;ingress&gt;</td>
<td>Ingress neutron port (name or id)</td>
</tr>
<tr>
<td>--egress &lt;egress&gt;</td>
<td>Egress neutron port (name or id)</td>
</tr>
</tbody>
</table>

Table 74.47. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 74.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.49. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 74.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
74.13. SFC PORT PAIR DELETE

Delete a given port pair

**Usage:**

```bash
openstack sfc port pair delete [-h] <port-pair>
```

**Table 74.51. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair&gt;</td>
<td>Port pair to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 74.52. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

74.14. SFC PORT PAIR GROUP CREATE

Create a port pair group

**Usage:**

```bash
openstack sfc port pair group create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--description <description>]
[--port-pair <port-pair>]
[--enable-tap | --disable-tap]
[--port-pair-group-parameters lb-fields=<lb-fields>]
<name>
```

**Table 74.53. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the port pair group</td>
</tr>
</tbody>
</table>

**Table 74.54. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>Description for the port pair group</td>
</tr>
<tr>
<td><code>--port-pair &lt;port-pair&gt;</code></td>
<td>Port pair (name or id). this option can be repeated.</td>
</tr>
<tr>
<td><code>--enable-tap</code></td>
<td>Port pairs of this port pair group are deployed as passive tap service function</td>
</tr>
<tr>
<td><code>--disable-tap</code></td>
<td>Port pairs of this port pair group are deployed as l3 service function (default)</td>
</tr>
<tr>
<td><code>--port-pair-group-parameters lb-fields=&lt;lb-fields&gt;</code></td>
<td>Dictionary of port pair group parameters. currently only one parameter lb-fields is supported. &lt;lb-fields&gt; is a &amp; separated list of load-balancing fields.</td>
</tr>
</tbody>
</table>

**Table 74.55. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 74.56. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 74.57. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 74.58. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

74.15. SFC PORT PAIR GROUP DELETE

Delete a given port pair group

Usage:

```
openstack sfc port pair group delete [-h] <port-pair-group>
```

Table 74.59. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair-group&gt;</td>
<td>Port pair group to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 74.60. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

74.16. SFC PORT PAIR GROUP LIST

List port pair group

Usage:

```
```
### Table 74.61. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

### Table 74.62. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 74.63. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 74.64. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 74.65. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 74.17. SFC PORT PAIR GROUP SET

Set port pair group properties

**Usage:**

```
openstack sfc port pair group set [-h] [--name <name>] [--description <description>] [--port-pair <port-pair>] [--no-port-pair] <port-pair-group>
```

**Table 74.66. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair-group&gt;</td>
<td>Port pair group to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 74.67. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of the port pair group</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description for the port pair group</td>
</tr>
<tr>
<td>--port-pair &lt;port-pair&gt;</td>
<td>Port pair (name or id). this option can be repeated.</td>
</tr>
<tr>
<td>--no-port-pair</td>
<td>Remove all port pair from port pair group</td>
</tr>
</tbody>
</table>

### 74.18. SFC PORT PAIR GROUP SHOW

Display port pair group details

**Usage:**
openstack sfc port pair group show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  <port-pair-group>

Table 74.68. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair-group&gt;</td>
<td>Port pair group to display (name or id)</td>
</tr>
</tbody>
</table>

Table 74.69. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 74.70. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 74.71. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.72. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 74.73. Table formatter options
74.19. SFC PORT PAIR GROUP UNSET

Unset port pairs from port pair group

Usage:

```
openstack sfc port pair group unset [-h] [--port-pair <port-pair> | --all-port-pair] <port-pair-group>
```

Table 74.74. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair-group&gt;</td>
<td>Port pair group to unset (name or id)</td>
</tr>
</tbody>
</table>

Table 74.75. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--port-pair &lt;port-pair&gt;</td>
<td>Remove port pair(s) from the port pair group (name or ID). This option can be repeated.</td>
</tr>
<tr>
<td>--all-port-pair</td>
<td>Remove all port pairs from the port pair group</td>
</tr>
</tbody>
</table>

74.20. SFC PORT PAIR LIST

List port pairs

Usage:

```
openstack sfc port pair list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
```
Table 74.76. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 74.77. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 74.78. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 74.79. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.80. Table formatter options
### 74.21. SFC PORT PAIR SET

Set port pair properties

**Usage:**

```
openstack sfc port pair set [-h] [--name <name>] [--description <description>] <port-pair>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 74.81. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>&lt;port-pair&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 74.82. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
</tr>
</tbody>
</table>

### 74.22. SFC PORT PAIR SHOW

Display port pair details

**Usage:**

```
openstack sfc port pair show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [-noindent] [-prefix PREFIX] [--max-width <integer>] [-fit-width]
```
Table 74.83. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port-pair&gt;</td>
<td>Port pair to display (name or id)</td>
</tr>
</tbody>
</table>

Table 74.84. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 74.85. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 74.86. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.87. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 74.88. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
74.23. SFC SERVICE GRAPH CREATE

Create a service graph.

Usage:

```
openstack sfc service graph create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--description DESCRIPTION]
    --branching-point
    SRC_CHAIN:DST_CHAIN_1,DST_CHAIN_2,DST_CHAIN_N
    <name>
```

Table 74.89. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the service graph.</td>
</tr>
</tbody>
</table>

Table 74.90. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description for the service graph.</td>
</tr>
<tr>
<td>--branching-point SRC_CHAIN:</td>
<td>Service graph branching point: the key is the</td>
</tr>
<tr>
<td>DST_CHAIN_1,DST_CHAIN_2,DST</td>
<td>source Port Chain while the value is a list</td>
</tr>
<tr>
<td>_CHAIN_N</td>
<td>of destination Port Chains. This option can</td>
</tr>
<tr>
<td>&lt;name&gt;</td>
<td>be repeated.</td>
</tr>
</tbody>
</table>

Table 74.91. Output formatter options

---

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
### Value Summary

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

#### Table 74.92. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table 74.93. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

#### Table 74.94. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 74.24. SFC SERVICE GRAPH DELETE

Delete a given service graph.

#### Usage:

```
openstack sfc service graph delete [-h] <service-graph>
```

#### Table 74.95. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-graph&gt;</td>
<td>Id or name of the service graph to delete.</td>
</tr>
</tbody>
</table>
### Table 74.96. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 74.25. SFC SERVICE GRAPH LIST

List service graphs

**Usage:**

```bash
```

### Table 74.97. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

### Table 74.98. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format csv,json,table,value,yaml</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
</tbody>
</table>
---sort-descending  | Sort the column(s) in descending order

Table 74.99. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 74.100. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 74.101. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 74.26. SFC SERVICE GRAPH SET

Set service graph properties

**Usage:**

```
openstack sfc service graph set [-h] [--name <name>] [--description <description>] <service-graph>
```

Table 74.102. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-graph&gt;</td>
<td>Service graph to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 74.103. Command arguments
## 74.27. SFC SERVICE GRAPH SHOW

Show information of a given service graph.

**Usage:**

```
openstack sfc service graph show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    <service-graph>
```

### Table 74.104. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-graph&gt;</td>
<td>Id or name of the service graph to display.</td>
</tr>
</tbody>
</table>

### Table 74.105. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 74.106. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 74.107. JSON formatter options
**Table 74.108. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 74.109. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 75. SHARE

This chapter describes the commands under the `share` command.

75.1. SHARE ABANDON

Abandon a share

Usage:

```
openstack share abandon [-h] [-wait] <share> [<share> ...]
```

Table 75.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of the share(s)</td>
</tr>
</tbody>
</table>

Table 75.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait until share is abandoned</td>
</tr>
</tbody>
</table>

75.2. SHARE ACCESS CREATE

Create new share access rule

Usage:

```
openstack share access create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] [--properties [<key=value> ...]] [--access-level <access_level>] <share> <access_type> <access_to>
```

Table 75.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of the nas share to modify.</td>
</tr>
<tr>
<td>&lt;access_type&gt;</td>
<td>Access rule type (only &quot;ip&quot;, &quot;user&quot; (user or group), &quot;cert&quot; or &quot;cephx&quot; are supported).</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>&lt;access_to&gt;</td>
<td>Value that defines access.</td>
</tr>
</tbody>
</table>

Table 75.4. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--properties [&lt;key=value&gt; ...]</td>
<td>Space separated list of key=value pairs of properties. OPTIONAL: Default=None. Available only for API microversion &gt;= 2.45.</td>
</tr>
<tr>
<td>--access-level &lt;access_level&gt;</td>
<td>Share access level (&quot;rw&quot; and &quot;ro&quot; access levels are supported). Defaults to rw.</td>
</tr>
</tbody>
</table>

Table 75.5. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.6. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.7. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.8. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
75.3. SHARE ACCESS DELETE

Delete a share access rule

Usage:

```bash
openstack share access delete [-h] <share> <id>
```

Table 75.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of the nas share to modify.</td>
</tr>
<tr>
<td>&lt;id&gt;</td>
<td>Id of the access rule to be deleted.</td>
</tr>
</tbody>
</table>

Table 75.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

75.4. SHARE ACCESS LIST

List share access rule

Usage:

```bash
```
### Table 75.11. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of the share.</td>
</tr>
</tbody>
</table>

### Table 75.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--properties [&lt;key=value&gt; ...]</td>
<td>Filters results by properties (key=value). optional: Default=None. Available only for API microversion &gt;= 2.45</td>
</tr>
</tbody>
</table>

### Table 75.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 75.14. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 75.15. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 75.16. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

75.5. SHARE ACCESS SET

Set properties to share access rule. Available for API microversion 2.45 and higher

Usage:

openstack share access set [-h] [--property <key=value>] <access_id>

Table 75.17. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access_id&gt;</td>
<td>Id of the nas share access rule.</td>
</tr>
</tbody>
</table>

Table 75.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property to this share access rule. (Repeat option to set multiple properties) Available only for API microversion &gt;= 2.45.</td>
</tr>
</tbody>
</table>

75.6. SHARE ACCESS SHOW

Display a share access rule. Available for API microversion 2.45 and higher

Usage:

openstack share access show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [-max-width <integer>] [--fit-width] [--print-empty] <access_id>
Table 75.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access_id&gt;</td>
<td>Id of the nas share access rule.</td>
</tr>
</tbody>
</table>

Table 75.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.22. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.23. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.24. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
### 75.7. SHARE ACCESS UNSET

Unset properties of share access rule. Available for API microversion 2.45 and higher.

**Usage:**

```
openstack share access unset [-h] [--property <key>] <access_id>
```

**Table 75.25. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access_id&gt;</td>
<td>Id of the nas share access rule.</td>
</tr>
</tbody>
</table>

**Table 75.26. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Remove property from share access rule. (repeat option to remove multiple properties) Available only for API microversion &gt;= 2.45.</td>
</tr>
</tbody>
</table>

### 75.8. SHARE ADOPT

Adopt a share

**Usage:**

```
```

**Table 75.27. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;access_id&gt;</td>
<td>Id of the nas share access rule.</td>
</tr>
</tbody>
</table>
### Table 75.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;service-host&gt;</td>
<td>Service host: some.host@driver#pool.</td>
</tr>
<tr>
<td>&lt;protocol&gt;</td>
<td>Protocol of the share to manage, such as nfs or cifs.</td>
</tr>
<tr>
<td>&lt;export-path&gt;</td>
<td>Share export path, nfs share such as: 10.0.0.1:/example_path, CIFS share such as: \10.0.0.1\example_cifs_share.</td>
</tr>
</tbody>
</table>

### Table 75.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 75.30. JSON formatter options

- Level of visibility for share. defines whether other projects are able to see it or not. Available only for microversion >= 2.8. (Default=False)
- Share server associated with share when using a share type with "driver_handles_share_servers" extra_spec set to True. Available only for microversion >= 2.49. (Default=False)
### Table 75.31. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 75.32. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.9. SHARE CREATE

Create new share

**Usage:**

```bash
```

### Table 75.33. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_protocol&gt;</td>
<td>Share protocol (nfs, cifs, cephfs, glusterfs or hdfs)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>&lt;size&gt;</td>
<td>Share size in gib.</td>
</tr>
</tbody>
</table>

**Table 75.34. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Optional share name. (default=none)</td>
</tr>
<tr>
<td>--snapshot-id &lt;snapshot-id&gt;</td>
<td>Optional snapshot id to create the share from. (Default=None)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property to this share (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--share-network &lt;network-info&gt;</td>
<td>Optional network info id or name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Optional share description. (default=none)</td>
</tr>
<tr>
<td>--public &lt;public&gt;</td>
<td>Level of visibility for share. defines whether other tenants are able to see it or not. (Default = False)</td>
</tr>
<tr>
<td>--share-type &lt;share-type&gt;</td>
<td>Optional share type. use of optional shares type is deprecated. (Default=Default)</td>
</tr>
<tr>
<td>--availability-zone &lt;availability-zone&gt;</td>
<td>Availability zone in which share should be created.</td>
</tr>
<tr>
<td>--share-group &lt;share-group&gt;</td>
<td>Optional share group name or id in which to create the share. (Default=None).</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for share creation</td>
</tr>
</tbody>
</table>

**Table 75.35. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 75.36. JSON formatter options**
Table 75.37. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.38. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

75.10. SHARE DELETE

Delete a share

Usage:

```bash
openstack share delete [-h] [--share-group <share-group>] [--force] [--wait] <share> [<share> ...]
```

Table 75.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Share(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 75.40. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
Optional share group (name or id) which contains the share

Attempt forced removal of share(s), regardless of state (defaults to False)

Wait for share deletion

75.11. SHARE EXPORT LOCATION LIST

List export location of a share

Usage:

openstack share export location list [-h]
[-f \{csv,json,table,value,yaml\}]
[-c COLUMN]
[--quote \{all,minimal,none,nonnumeric\}]
[--noindent]
[--max-width <integer>]
[--fix-width] [-print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
<share>

Table 75.41. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of share</td>
</tr>
</tbody>
</table>

Table 75.42. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.43. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**Value** | **Summary**
---|---
-c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns

--sort-column SORT_COLUMN | Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending | Sort the column(s) in ascending order

--sort-descending | Sort the column(s) in descending order

### Table 75.44. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--quote {all,minimal,none,nonnumeric} | When to include quotes, defaults to nonnumeric

### Table 75.45. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--noindent | Whether to disable indenting the json

### Table 75.46. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty | Print empty table if there is no data to show.

### 75.12. SHARE EXPORT LOCATION SHOW

Show export location of a share

**Usage:**

```
openstack share export location show [-h]
```
Table 75.47. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of share</td>
</tr>
<tr>
<td>&lt;export-location&gt;</td>
<td>Id of the share export location</td>
</tr>
</tbody>
</table>

Table 75.48. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.49. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.50. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.51. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.52. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.13. SHARE LIST

List shares

**Usage:**

```
openstack share list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] 
  [--quote {all, minimal, none, nonnumeric}] 
  [--noindent] [--max-width <integer>] [--fit-width] 
  [--print-empty] [--sort-column SORT_COLUMN] 
  [--sort-ascending | --sort-descending] 
  [--name <share-name>] [--status <share-status>] 
  [--snapshot <share-network-id>] [--public] 
  [--share-network <share-network-name-or-id>] 
  [--share-type <share-type-name-or-id>] 
  [--share-group <share-group-name-or-id>] 
  [--host <share-host>] 
  [--share-server <share-server-id>] 
  [--project <project>] 
  [--project-domain <project-domain>] 
  [--user <user>] [--user-domain <user-domain>] 
  [--all-projects] [--property <key=value>] 
  [--extra-spec <key=value>] [--long] 
  [--sort <key>[:<direction>]] 
  [--limit <num-shares>] [--marker <share>]
```

**Table 75.53. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;share-name&gt;</td>
<td>Filter shares by share name</td>
</tr>
<tr>
<td>--status &lt;share-status&gt;</td>
<td>Filter shares by status</td>
</tr>
<tr>
<td>--snapshot &lt;share-network-id&gt;</td>
<td>Filter shares by snapshot name or id.</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td><strong>Summary</strong></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>--public</td>
<td>Include public shares</td>
</tr>
<tr>
<td>--share-network &lt;share-network-name-or-id&gt;</td>
<td>Filter shares exported on a given share network</td>
</tr>
<tr>
<td>--share-type &lt;share-type-name-or-id&gt;</td>
<td>Filter shares of a given share type</td>
</tr>
<tr>
<td>--share-group &lt;share-group-name-or-id&gt;</td>
<td>Filter shares belonging to a given share group</td>
</tr>
<tr>
<td>--host &lt;share-host&gt;</td>
<td>Filter shares belonging to a given host (admin only)</td>
</tr>
<tr>
<td>--share-server &lt;share-server-id&gt;</td>
<td>Filter shares exported via a given share server (admin only)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Filter shares by project (name or id) (admin only)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Filter results by user (name or id) (admin only)</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Include all projects (admin only)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Filter shares having a given metadata key=value property (repeat option to filter by multiple properties)</td>
</tr>
<tr>
<td>--extra-spec &lt;key=value&gt;</td>
<td>Filter shares with extra specs (key=value) of the share type that they belong to. (repeat option to filter by multiple extra specs)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[&lt;direction&gt;]</td>
<td>Sort output by selected keys and directions (asc or desc) (default: name:asc), multiple keys and directions can be specified separated by comma</td>
</tr>
<tr>
<td>--limit &lt;num-shares&gt;</td>
<td>Maximum number of shares to display</td>
</tr>
<tr>
<td>--marker &lt;share&gt;</td>
<td>The last share id of the previous page</td>
</tr>
</tbody>
</table>

Table 75.54. Output formatter options
### Value Summary

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f [csv, json, table, value, yaml], --format [csv, json, table, value, yaml]</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

#### CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all, minimal, none, nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

#### JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

#### Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.14. SHARE MESSAGE DELETE

Remove one or more messages

**Usage:**

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
openstack share message delete [-h] <message> [<message> ...]

Table 75.58. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;message&gt;</td>
<td>Id of the message(s).</td>
</tr>
</tbody>
</table>

Table 75.59. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

75.15. SHARE MESSAGE LIST

Lists all messages

Usage:

```
```

Table 75.60. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource-id &lt;resource-id&gt;</td>
<td>Filters results by a resource uuid. default=none.</td>
</tr>
<tr>
<td>--resource-type &lt;resource-type&gt;</td>
<td>Filters results by a resource type. default=none. Example: &quot;openstack message list --resource-type share&quot;</td>
</tr>
<tr>
<td>--action-id &lt;action-id&gt;</td>
<td>Filters results by action id. default=none.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--detail-id &lt;detail-id&gt;</td>
<td>Filters results by detail id. default=none.</td>
</tr>
<tr>
<td>--request-id &lt;request-id&gt;</td>
<td>Filters results by request id. default=none.</td>
</tr>
<tr>
<td>--message-level &lt;message-level&gt;</td>
<td>Filters results by the message level. default=none. Example: &quot;openstack message list --message-level ERROR&quot;.</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Maximum number of messages to return. (default=none)</td>
</tr>
<tr>
<td>--since &lt;since&gt;</td>
<td>Return only user messages created since given date. The date format must be conforming to ISO8601. Available only for microversion &gt;= 2.52.</td>
</tr>
<tr>
<td>--before &lt;before&gt;</td>
<td>Return only user messages created before given date. The date format must be conforming to ISO8601. Available only for microversion &gt;= 2.52.</td>
</tr>
</tbody>
</table>

**Table 75.61. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 75.62. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 75.63. JSON formatter options**
### 75.16. SHARE MESSAGE SHOW

Show details about a message

**Usage:**

```
```

**Table 75.65. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;message&gt;</td>
<td>Id of the message.</td>
</tr>
</tbody>
</table>

**Table 75.66. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 75.67. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 75.64. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
**Value** | **Summary**
---|---
-\( f \) \{json, shell, table, value, yaml\}, `--format` \{json, shell, table, value, yaml\} | The output format, defaults to table
-\( c \) COLUMN, `--column` COLUMN | Specify the column(s) to include, can be repeated to show multiple columns

**Table 75.68. JSON formatter options**

**Value** | **Summary**
---|---
`--noindent` | Whether to disable indenting the json

**Table 75.69. Shell formatter options**

**Value** | **Summary**
---|---
`--prefix` PREFIX | Add a prefix to all variable names

**Table 75.70. Table formatter options**

**Value** | **Summary**
---|---
`--max-width` <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
`--fit-width` | Fit the table to the display width. implied if `--max-width` greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
`--print-empty` | Print empty table if there is no data to show.

### 75.17. SHARE PROPERTIES SHOW

Show share properties

**Usage:**

```
```

**Table 75.71. Positional arguments**
### Table 75.72. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 75.73. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 75.74. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 75.75. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 75.76. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
## 75.18. SHARE QUOTA DELETE

Delete Quota

**Usage:**

```
openstack share quota delete [-h] [--project <project>] [--user <user>]
[--share-type <share-type>]
```

### Table 75.77. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Name or id of the project to delete quotas for.</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Name or id of user to delete the quotas for. Optional. Mutually exclusive with --share-type.</td>
</tr>
<tr>
<td>--share-type &lt;share-type&gt;</td>
<td>Name or id of a share type to delete the quotas for. Optional. Mutually exclusive with --user. Available only for microversion &gt;= 2.39</td>
</tr>
</tbody>
</table>

## 75.19. SHARE QUOTA SET

Set Quota

**Usage:**

```
openstack share quota set [-h] [--project <project>] [--user <user>]
[--shares <shares>] [--snapshots <snapshots>]
[--gigabytes <gigabytes>]
[--snapshot-gigabytes <snapshot-gigabytes>]
[--share-networks <share-networks>]
[--share-groups <share-groups>]
[--share-group-snapshots <share-group-snapshots>]
[--share-replicas <share-replicas>]
[--replica-gigabytes <replica-gigabytes>]
[--share-type <share-type>] [--force]
```

### Table 75.78. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Name or id of the project to set the quotas for.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
<td>Name or id of a user to set the quotas for. optional. Mutually exclusive with --share-type.</td>
</tr>
<tr>
<td>--shares &lt;shares&gt;</td>
<td>New value for the &quot;shares&quot; quota.</td>
</tr>
<tr>
<td>--snapshots &lt;snapshots&gt;</td>
<td>New value for the &quot;snapshots&quot; quota.</td>
</tr>
<tr>
<td>--gigabytes &lt;gigabytes&gt;</td>
<td>New value for the &quot;gigabytes&quot; quota.</td>
</tr>
<tr>
<td>--snapshot-gigabytes &lt;snapshot-gigabytes&gt;</td>
<td>New value for the &quot;snapshot-gigabytes&quot; quota.</td>
</tr>
<tr>
<td>--share-networks &lt;share-networks&gt;</td>
<td>New value for the &quot;share-networks&quot; quota.</td>
</tr>
<tr>
<td>--share-groups &lt;share-groups&gt;</td>
<td>New value for the &quot;share-groups&quot; quota. available only for microversion &gt;= 2.40</td>
</tr>
<tr>
<td>--share-group-snapshots &lt;share-group-snapshots&gt;</td>
<td>New value for the &quot;share-group-snapshots&quot; quota.</td>
</tr>
<tr>
<td>--share-replicas &lt;share-replicas&gt;</td>
<td>Number of share replicas. available only for microversion &gt;= 2.53</td>
</tr>
<tr>
<td>--replica-gigabytes &lt;replica-gigabytes&gt;</td>
<td>Capacity of share replicas in total. available only for microversion &gt;= 2.53</td>
</tr>
<tr>
<td>--share-type &lt;share-type&gt;</td>
<td>Name or id of a share type to set the quotas for. Optional. Mutually exclusive with --user.</td>
</tr>
<tr>
<td>--force</td>
<td>Force update the quota.</td>
</tr>
</tbody>
</table>

### 75.20. SHARE QUOTA SHOW

Show Quota

Usage:

```
```

Table 75.79. Command arguments
### Table 75.80. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 75.81. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 75.82. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 75.83. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
75.21. SHARE RESIZE

Resize a share

Usage:

openstack share resize [-h] [-w] <share> <new-size>

Table 75.84. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of share to resize</td>
</tr>
<tr>
<td>&lt;new-size&gt;</td>
<td>New size of share, in gibbs</td>
</tr>
</tbody>
</table>

Table 75.85. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for share resize</td>
</tr>
</tbody>
</table>

75.22. SHARE REVERT

Revert a share to the specified snapshot.

Usage:

openstack share revert [-h] <snapshot>

Table 75.86. Positional arguments
Table 75.87. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot to restore. The snapshot must be the most recent one known to manila.</td>
</tr>
</tbody>
</table>

75.23. SHARE SET

Set share properties

Usage:

```
```

Table 75.88. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Share to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 75.89. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property to this share (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New share name. (default=none)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New share description. (default=none)</td>
</tr>
<tr>
<td>--public &lt;public&gt;</td>
<td>Level of visibility for share. defines whether other tenants are able to see it or not.</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Explicitly update the status of a share (admin only). Examples include: available, error, creating, deleting, error_deleting.</td>
</tr>
</tbody>
</table>
75.24. SHARE SHOW

Display share details

Usage:

<share>

Table 75.90. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Share to display (name or id)</td>
</tr>
</tbody>
</table>

Table 75.91. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.92. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.93. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.94. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.95. Table formatter options
### 75.25. SHARE SNAPSHOT ABANDON

Abandon share snapshot(s)

Usage:

```
openstack share snapshot abandon [-h] <snapshot> [<snapshot> ...]
```

Table 75.96. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot(s) to be abandoned.</td>
</tr>
</tbody>
</table>

Table 75.97. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 75.26. SHARE SNAPSHOT ACCESS CREATE

Allow access to a snapshot

Usage:

```
openstack share snapshot access create [-h] 
    [-f {json,shell,table,value,yaml}] 
    [-c COLUMN] [-noindent] 
    [-prefix PREFIX] 
    [--max-width <integer>] 
    [--fit-width] [--print-empty] 
    <snapshot> <access_type> 
    <access_to>
```

Table 75.98. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot</td>
</tr>
<tr>
<td>&lt;access_type&gt;</td>
<td>Access rule type (only &quot;ip&quot;, &quot;user&quot; (user or group), &quot;cert&quot; or &quot;cephx&quot; are supported).</td>
</tr>
<tr>
<td>&lt;access_to&gt;</td>
<td>Value that defines access</td>
</tr>
</tbody>
</table>

Table 75.99. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.100. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.101. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.102. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.103. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
75.27. SHARE SNAPSHOT ACCESS DELETE

Delete access to a snapshot

Usage:

openstack share snapshot access delete [-h] <snapshot> <id> [<id> ...]

Table 75.104. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the share snapshot to deny access to.</td>
</tr>
<tr>
<td>&lt;id&gt;</td>
<td>Id(s) of the access rule(s) to be deleted.</td>
</tr>
</tbody>
</table>

Table 75.105. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

75.28. SHARE SNAPSHOT ACCESS LIST

Show access list for a snapshot

Usage:

openstack share snapshot access list [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  <snapshot>
Table 75.106. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the share snapshot to show access list for.</td>
</tr>
</tbody>
</table>

Table 75.107. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.108. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 75.109. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 75.110. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.111. Table formatter options
75.29. SHARE SNAPSHOT ADOPT

Adopt a share snapshot

Usage:

```bash
```

Table 75.112. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;share&gt;</code></td>
<td>Name or id of the share that owns the snapshot to be adopted.</td>
</tr>
<tr>
<td><code>&lt;provider-location&gt;</code></td>
<td>Provider location of the snapshot on the backend.</td>
</tr>
</tbody>
</table>

Table 75.113. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>-name &lt;name&gt;</code></td>
<td>Optional snapshot name (default=None).</td>
</tr>
<tr>
<td><code>-description &lt;description&gt;</code></td>
<td>Optional snapshot description (default=None).</td>
</tr>
<tr>
<td><code>-driver-option &lt;key=value&gt;</code></td>
<td>Set driver options as key=value pairs. (repeat option to set multiple key=value pairs)</td>
</tr>
</tbody>
</table>
Table 75.114. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.115. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.116. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.117. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

75.30. SHARE SNAPSHOT CREATE

Create a snapshot of the given share

Usage:

```
```
Table 75.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Name or id of the share to create snapshot of</td>
</tr>
</tbody>
</table>

Table 75.119. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Optional flag to indicate whether to snapshot a share even if it's busy. (Default=False)</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Add a name to the snapshot (optional).</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Add a description to the snapshot (optional).</td>
</tr>
</tbody>
</table>

Table 75.120. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.121. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.122. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.123. Table formatter options
### 75.31. SHARE SNAPAPSHOT DELETE

Delete one or more share snapshots

**Usage:**

```
openstack share snapshot delete [-h] [--force] <snapshot> [...]  
```

Table 75.124. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot(s) to delete</td>
</tr>
</tbody>
</table>

Table 75.125. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Delete the snapshot(s) ignoring the current state.</td>
</tr>
</tbody>
</table>

### 75.32. SHARE SNAPSHOT EXPORT LOCATION LIST

List export locations of a given snapshot

**Usage:**

```
openstack share snapshot export location list [-h]  
    [-f {csv,json,table,value,yaml}]  
    [-c COLUMN]  
    [--quote {all,minimal,none,nonnumeric}]  
    [-noindent]  
    [--max-width <integer>]  
    [--fit-width]  
    [--print-empty]  
```
Table 75.126. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the share snapshot.</td>
</tr>
</tbody>
</table>

Table 75.127. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.128. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 75.129. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 75.130. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.131. Table formatter options
75.33. SHARE SNAPSHOT EXPORT LOCATION SHOW

Show export location of the share snapshot

Usage:

openstack share snapshot export location show [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width]
[--print-empty]
.snapshot>
<export-location>

Table 75.132. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the share snapshot.</td>
</tr>
<tr>
<td>&lt;export-location&gt;</td>
<td>Id of the share snapshot export location.</td>
</tr>
</tbody>
</table>

Table 75.133. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 75.134. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 75.135. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.136. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.137. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.34. SHARE SNAPSHOT LIST

List snapshots

**Usage:**

```
```
### Table 75.138. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Display snapshots from all projects (admin only).</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Filter results by name.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Filter results by description. available only for microversion &gt;= 2.36.</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Filter results by status</td>
</tr>
<tr>
<td>--share &lt;share&gt;</td>
<td>Name or id of a share to filter results by.</td>
</tr>
<tr>
<td>--usage &lt;usage&gt;</td>
<td>Option to filter snapshots by usage.</td>
</tr>
<tr>
<td>--limit &lt;num-snapshots&gt;</td>
<td>Limit the number of snapshots returned</td>
</tr>
<tr>
<td>--marker &lt;snapshot&gt;</td>
<td>The last share id of the previous page</td>
</tr>
<tr>
<td>--sort &lt;key&gt;:&lt;direction&gt;</td>
<td>Sort output by selected keys and directions (asc or desc) (default: name:asc), multiple keys and directions can be specified separated by comma</td>
</tr>
<tr>
<td>--name~ &lt;name~&gt;</td>
<td>Filter results matching a share snapshot name pattern. Available only for microversion &gt;= 2.36.</td>
</tr>
<tr>
<td>--description~ &lt;description~&gt;</td>
<td>Filter results matching a share snapshot description pattern. Available only for microversion &gt;= 2.36.</td>
</tr>
<tr>
<td>--detail</td>
<td>List share snapshots with details</td>
</tr>
</tbody>
</table>

### Table 75.139. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**75.35. SHARE SNAPSHOT SET**

Set share snapshot properties

**Usage:**

```
openstack share snapshot set [-h] [--name <name>]
```
Table 75.143. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot to set a property for</td>
</tr>
</tbody>
</table>

Table 75.144. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set a name to the snapshot.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set a description to the snapshot.</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Assign a status to the snapshot (admin only). options include: available, error, creating, deleting, manage_starting, manage_error, unmanage_starting, unmanage_error, error_deleting.</td>
</tr>
</tbody>
</table>

**75.36. SHARE SNAPSHOT SHOW**

Show details about a share snapshot

*Usage:*

```
```

Table 75.145. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Name or id of the snapshot to display</td>
</tr>
</tbody>
</table>

Table 75.146. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
Table 75.147. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 75.148. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 75.149. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 75.150. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

75.37. SHARE SNAPSHOT UNSET

Unset a share snapshot property

Usage:

    openstack share snapshot unset [-h] [--name] [--description] <snapshot>

Table 75.151. Positional arguments
### Table 75.152. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;snapshot&gt;</code></td>
<td>Name or id of the snapshot to set a property for</td>
</tr>
</tbody>
</table>

### 75.38. SHARE TYPE ACCESS CREATE

Add access for share type

**Usage:**

```bash
openstack share type access create [-h] <share_type> <project_id>
```

### Table 75.153. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;share_type&gt;</code></td>
<td>Share type name or id to add access to</td>
</tr>
<tr>
<td><code>&lt;project_id&gt;</code></td>
<td>Project id to add share type access for</td>
</tr>
</tbody>
</table>

### Table 75.154. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 75.39. SHARE TYPE ACCESS DELETE

Delete access from share type

**Usage:**

```bash
openstack share type access delete [-h] <share_type> <project_id>
```

### Table 75.155. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;share_type&gt;</code></td>
<td>Share type name or id to add access to</td>
</tr>
<tr>
<td><code>&lt;project_id&gt;</code></td>
<td>Project id to add share type access for</td>
</tr>
</tbody>
</table>
## 75.40. SHARE TYPE ACCESS LIST

Get access list for share type

**Usage:**

```
```

### Table 75.157. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_type&gt;</td>
<td>Share type name or id to get access list for</td>
</tr>
</tbody>
</table>

### Table 75.158. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 75.159. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
75.41. SHARE TYPE CREATE

Create new share type

Usage:

```
openstack share type create [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX]
```
Table 75.163. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Share type name</td>
</tr>
<tr>
<td>&lt;spec_driver_handles_share_servers&gt;</td>
<td>Required extra specification. valid values are true and false</td>
</tr>
</tbody>
</table>

Table 75.164. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Share type description. available only for microversion &gt;= 2.41.</td>
</tr>
<tr>
<td>--snapshot-support &lt;snapshot_support&gt;</td>
<td>Boolean extra spec used for filtering of back ends by their capability to create share snapshots.</td>
</tr>
<tr>
<td>--create-share-from-snapshot-support &lt;create_share_from_snapshot_support&gt;</td>
<td>Boolean extra spec used for filtering of back ends by their capability to create shares from snapshots.</td>
</tr>
<tr>
<td>--revert-to-snapshot-support &lt;revert_to_snapshot_support&gt;</td>
<td>Boolean extra spec used for filtering of back ends by their capability to revert shares to snapshots. (Default is False).</td>
</tr>
<tr>
<td>--mount-snapshot-support &lt;mount_snapshot_support&gt;</td>
<td>Boolean extra spec used for filtering of back ends by their capability to mount share snapshots. (Default is False).</td>
</tr>
<tr>
<td>--extra-specs [key=value&gt; ...]</td>
<td>Extra specs key and value of share type that will be used for share type creation. OPTIONAL: Default=None. example --extra-specs thin_provisioning=&lt;is&gt; True, replication_type=readable.</td>
</tr>
</tbody>
</table>
- **--public <public>**
  Make type accessible to the public (default true).

**Table 75.165. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 75.166. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 75.167. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 75.168. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**75.42. SHARE TYPE DELETE**

Delete a share type

**Usage:**
openstack share type delete [-h] <share_types> [<share_types> ...]

Table 75.169. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_types&gt;</td>
<td>Name or id of the share type(s) to delete</td>
</tr>
</tbody>
</table>

Table 75.170. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

75.43. SHARE TYPE LIST

List share types

Usage:


Table 75.171. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all</td>
<td>Display all share types whatever public or private. Default=False. (Admin only)</td>
</tr>
</tbody>
</table>

| --extra-specs [key=value> ...] | Filter share types with extra specs (key=value). Available only for API microversion >= 2.43. OPTIONAL: Default=None. |

Table 75.172. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### 75.173. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### 75.174. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### 75.175. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.44. SHARE TYPE SET

Set share type properties

**Usage:**

```
openstack share type set [-h] [--extra-specs <key=value> ...] [--public <public>]
```
Table 75.176. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_type&gt;</td>
<td>Name or id of the share type to modify</td>
</tr>
</tbody>
</table>

Table 75.177. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--extra-specs [key=value] ...</td>
<td>Extra specs key and value of share type that will be used for share type creation. OPTIONAL: Default=None. example --extra-specs thin_provisioning=&lt;is&gt; True, replication_type=readable.</td>
</tr>
<tr>
<td>--public &lt;public&gt;</td>
<td>New visibility of the share type. if set to true, share type will be available to all projects in the cloud. Available only for microversion &gt;= 2.50</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New description of share type. available only for microversion &gt;= 2.50</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New name of share type. available only for microversion &gt;= 2.50</td>
</tr>
</tbody>
</table>

75.45. SHARE TYPE SHOW

Display share type details

Usage:

```
```

Table 75.178. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_type&gt;</td>
<td>Share type to display (name or id)</td>
</tr>
</tbody>
</table>
### Table 75.179. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 75.180. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 75.181. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 75.182. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 75.183. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 75.46. SHARE TYPE UNSET

Unset share type extra specs

**Usage:**
openstack share type unset [-h] <share_type> <key> [<key> ...]

Table 75.184. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share_type&gt;</td>
<td>Name or id of the share type to modify</td>
</tr>
<tr>
<td>&lt;key&gt;</td>
<td>Remove extra_specs from this share type</td>
</tr>
</tbody>
</table>

Table 75.185. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

75.47. SHARE UNSET

Unset share properties

Usage:

openstack share unset [-h] [--property <key>] [--name] [--description] <share>

Table 75.186. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;share&gt;</td>
<td>Share to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 75.187. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Remove a property from share (repeat option to remove multiple properties)</td>
</tr>
<tr>
<td>--name</td>
<td>Unset share name.</td>
</tr>
<tr>
<td>--description</td>
<td>Unset share description.</td>
</tr>
</tbody>
</table>
CHAPTER 76. SOFTWARE

This chapter describes the commands under the software command.

76.1. SOFTWARE CONFIG CREATE

Create software config

Usage:

openstack software config create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>] [--fit-width]
[--print-empty]
[--config-file <config-file>]
[--definition-file <destination-file>]
[--group <group>]
<config-name>

Table 76.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;config-name&gt;</td>
<td>Name of the software config to create</td>
</tr>
</tbody>
</table>

Table 76.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config-file &lt;config-file&gt;</td>
<td>Path to json/yaml containing map defining &lt;inputs&gt;, &lt;outputs&gt;, and &lt;options&gt;</td>
</tr>
<tr>
<td>--definition-file &lt;destination-file&gt;</td>
<td>Path to software config script/data</td>
</tr>
<tr>
<td>--group &lt;group&gt;</td>
<td>Group name of tool expected by the software config</td>
</tr>
</tbody>
</table>

Table 76.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to json</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### Table 76.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 76.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 76.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 76.2. SOFTWARE CONFIG DELETE

Delete software configs

**Usage:**

```
openstack software config delete [-h] <config> [<config> ...]
```

### Table 76.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;config&gt;</code></td>
<td>Ids of the software configs to delete</td>
</tr>
</tbody>
</table>

### Table 76.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
76.3. SOFTWARE CONFIG LIST

List software configs

Usage:

```bash
```

Table 76.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Limit the number of configs returned</td>
</tr>
<tr>
<td>--marker &lt;id&gt;</td>
<td>Return configs that appear after the given config id</td>
</tr>
</tbody>
</table>

Table 76.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 76.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 76.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 76.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

76.4. SOFTWARE CONFIG SHOW

Show software config details

Usage:

```
```

Table 76.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;config&gt;</td>
<td>Id of the config</td>
</tr>
</tbody>
</table>

Table 76.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config-only</td>
<td>Only display the value of the &lt;config&gt; property.</td>
</tr>
</tbody>
</table>

Table 76.16. Output formatter options
Create a software deployment.

**Usage:**

```bash
code
openstack software deployment create [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width]
  [--print-empty]
  [--input-value <key=value>]
  [--action <action>]
code
```
```
[--config <config>]
[--signal-transport <signal-transport>]
[--container <container>]
[--timeout <timeout>] --server <server>
<deployment-name>
```

Table 76.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;deployment-name&gt;</td>
<td>Name of the derived config associated with this deployment. This is used to apply a sort order to the list of configurations currently deployed to the server.</td>
</tr>
</tbody>
</table>

Table 76.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--input-value &lt;key=value&gt;</td>
<td>Input value to set on the deployment. this can be specified multiple times.</td>
</tr>
<tr>
<td>--action &lt;action&gt;</td>
<td>Name of an action for this deployment. this can be a custom action, or one of CREATE, UPDATE, DELETE, SUSPEND, RESUME. Default is UPDATE</td>
</tr>
<tr>
<td>--config &lt;config&gt;</td>
<td>Id of the configuration to deploy</td>
</tr>
<tr>
<td>--signal-transport &lt;signal-transport&gt;</td>
<td>How the server should signal to heat with the deployment output values. TEMP_URL_SIGNAL will create a Swift TempURL to be signaled via HTTP PUT. ZAQAR_SIGNAL will create a dedicated zaqar queue to be signaled using the provided keystone credentials.NO_SIGNAL will result in the resource going to the COMPLETE state without waiting for any signal</td>
</tr>
<tr>
<td>--container &lt;container&gt;</td>
<td>Optional name of container to store temp_url_signal objects in. If not specified a container will be created with a name derived from the DEPLOY_NAME</td>
</tr>
<tr>
<td>--timeout &lt;timeout&gt;</td>
<td>Deployment timeout in minutes</td>
</tr>
<tr>
<td>--server &lt;server&gt;</td>
<td>Id of the server being deployed to</td>
</tr>
</tbody>
</table>

Table 76.22. Output formatter options
76.6. SOFTWARE DEPLOYMENT DELETE

Delete software deployment(s) and correlative config(s).

Usage:

```
openstack software deployment delete [-h] <deployment> [<deployment> ...]
```

Table 76.26. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;deployment&gt;</code></td>
<td>Id of the deployment(s) to delete.</td>
</tr>
</tbody>
</table>

Table 76.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 76.7. SOFTWARE DEPLOYMENT LIST

List software deployments.

**Usage:**

```
```

Table 76.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--server &lt;server&gt;</code></td>
<td>Id of the server to fetch deployments for</td>
</tr>
<tr>
<td><code>--long</code></td>
<td>List more fields in output</td>
</tr>
</tbody>
</table>

Table 76.29. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv, json, table, value, yaml}, --format</code> {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 76.30. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 76.31. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 76.32. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**76.8. SOFTWARE DEPLOYMENT METADATA SHOW**

Get deployment configuration metadata for the specified server.

**Usage:**

```
openstack software deployment metadata show [-h] <server>
```

**Table 76.33. Positional arguments**
Table 76.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;server&gt;</td>
<td>Id of the server to fetch deployments for</td>
</tr>
</tbody>
</table>

76.9. SOFTWARE DEPLOYMENT OUTPUT SHOW

Show a specific deployment output.

Usage:

```
openstack software deployment output show [-h] [-all] [--long] <deployment> [<output-name>]
```

Table 76.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;deployment&gt;</td>
<td>Id of deployment to show the output for</td>
</tr>
<tr>
<td>&lt;output-name&gt;</td>
<td>Name of an output to display</td>
</tr>
</tbody>
</table>

Table 76.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all</td>
<td>Display all deployment outputs</td>
</tr>
<tr>
<td>--long</td>
<td>Show full deployment logs in output</td>
</tr>
</tbody>
</table>

76.10. SOFTWARE DEPLOYMENT SHOW

Show SoftwareDeployment Details.

Usage:

```
openstack software deployment show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX]
```
Table 76.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;deployment&gt;</td>
<td>Id of the deployment</td>
</tr>
</tbody>
</table>

Table 76.38. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>Show more fields in output</td>
</tr>
</tbody>
</table>

Table 76.39. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 76.40. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 76.41. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 76.42. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 77. STACK

This chapter describes the commands under the stack command.

77.1. STACK ABANDON

Abandon stack and output results.

Usage:

```bash
```

Table 77.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to abandon</td>
</tr>
</tbody>
</table>

Table 77.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--output-file &lt;output-file&gt;</td>
<td>File to output abandon results</td>
</tr>
</tbody>
</table>

Table 77.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to json</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.5. Shell formatter options
Table 77.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack-name&gt;</td>
<td>Name of the stack to adopt</td>
</tr>
</tbody>
</table>

Table 77.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-e &lt;environment&gt;, --environment &lt;environment&gt;</td>
<td>Path to the environment. can be specified multiple times</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>--timeout &lt;timeout&gt;</td>
<td>Stack creation timeout in minutes</td>
</tr>
<tr>
<td>--enable-rollback</td>
<td>Enable rollback on create/update failure</td>
</tr>
<tr>
<td>--parameter &lt;key=value&gt;</td>
<td>Parameter values used to create the stack. can be specified multiple times</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait until stack adopt completes</td>
</tr>
<tr>
<td>--adopt-file &lt;adopt-file&gt;</td>
<td>Path to adopt stack data file</td>
</tr>
</tbody>
</table>

Table 77.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.10. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.11. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.12. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
77.3. STACK CANCEL

Cancel current task for a stack. Supported tasks for cancellation: * update * create

Usage:

```
```

Table 77.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack(s) to cancel (name or id)</td>
</tr>
</tbody>
</table>

Table 77.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for cancel to complete</td>
</tr>
<tr>
<td>--no-rollback</td>
<td>Cancel without rollback</td>
</tr>
</tbody>
</table>

Table 77.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### 77.4. STACK CHECK

Check a stack.

**Usage:**

```
openstack stack check [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] [--quote {all,minimal,none,nonnumeric}] [--noindent] [--max-width <integer>]
```

---

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 77.16. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 77.17. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 77.18. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width. &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
Table 77.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack(s) to check update (name or id)</td>
</tr>
</tbody>
</table>

Table 77.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for check to complete</td>
</tr>
</tbody>
</table>

Table 77.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 77.22. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.23. JSON formatter options
### 77.5. STACK CREATE

Create a stack.

**Usage:**

```
```

**Table 77.25. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack-name&gt;</td>
<td>Name of the stack to create</td>
</tr>
</tbody>
</table>

**Table 77.26. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>-e &lt;environment&gt;, --environment &lt;environment&gt;</td>
<td>Path to the environment. can be specified multiple times</td>
</tr>
<tr>
<td>-s &lt;files-container&gt;, --files-container &lt;files-container&gt;</td>
<td>Swift files container name. local files other than root template would be ignored. If other files are not found in swift, heat engine would raise an error.</td>
</tr>
<tr>
<td>--timeout &lt;timeout&gt;</td>
<td>Stack creating timeout in minutes</td>
</tr>
<tr>
<td>--pre-create &lt;resource&gt;</td>
<td>Name of a resource to set a pre-create hook to. Resources in nested stacks can be set using slash as a separator: <code>nested_stack/another/my_resource</code>. You can use wildcards to match multiple stacks or resources: <code>nested_stack/an*/_resource</code>. This can be specified multiple times</td>
</tr>
<tr>
<td>--enable-rollback</td>
<td>Enable rollback on create/update failure</td>
</tr>
<tr>
<td>--parameter &lt;key=value&gt;</td>
<td>Parameter values used to create the stack. this can be specified multiple times</td>
</tr>
<tr>
<td>--parameter-file &lt;key=file&gt;</td>
<td>Parameter values from file used to create the stack. This can be specified multiple times. Parameter values would be the content of the file</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait until stack goes to create_complete or CREATE_FAILED</td>
</tr>
<tr>
<td>--poll SECONDS</td>
<td>Poll interval in seconds for use with --wait, defaults to 5.</td>
</tr>
<tr>
<td>--tags &lt;tag1,tag2...&gt;</td>
<td>A list of tags to associate with the stack</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Do not actually perform the stack create, but show what would be created</td>
</tr>
<tr>
<td>-t &lt;template&gt;, --template &lt;template&gt;</td>
<td>Path to the template</td>
</tr>
</tbody>
</table>

Table 77.27. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### Table 77.28. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 77.29. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 77.30. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if <code>--max-width</code> greater than 0. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 77.6. STACK DELETE

Delete stack(s).

**Usage:**

```
openstack stack delete [-h] [-y] [--wait] <stack> [<stack> ...]
```

### Table 77.31. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;stack&gt;</code></td>
<td>Stack(s) to delete (name or id)</td>
</tr>
</tbody>
</table>
77.7. STACK ENVIRONMENT SHOW

Show a stack’s environment.

Usage:

openstack stack environment show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty]
    <NAME or ID>

Table 77.33. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NAME or ID&gt;</td>
<td>Name or id of stack to query</td>
</tr>
</tbody>
</table>

Table 77.34. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 77.35. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.36. JSON formatter options
### 77.8. STACK EVENT LIST

List events.

**Usage:**

```bash
```

#### Table 77.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to show events for</td>
</tr>
</tbody>
</table>
**Table 77.40. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resource &lt;resource&gt;</td>
<td>Name of resource to show events for. note: this cannot be specified with --nested-depth</td>
</tr>
<tr>
<td>--filter &lt;key=value&gt;</td>
<td>Filter parameters to apply on returned events</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>Limit the number of events returned</td>
</tr>
<tr>
<td>--marker &lt;id&gt;</td>
<td>Only return events that appear after the given id</td>
</tr>
<tr>
<td>--nested-depth &lt;depth&gt;</td>
<td>Depth of nested stacks from which to display events. Note: this cannot be specified with --resource</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by selected keys and directions (asc or desc) (default: asc). Specify multiple times to sort on multiple keys. Sort key can be: &quot;event_time&quot; (default), &quot;resource_name&quot;, &quot;links&quot;, &quot;logical_resource_id&quot;, &quot;resource_status&quot;, &quot;resource_status_reason&quot;, &quot;physical_resource_id&quot;, or &quot;id&quot;. You can leave the key empty and specify &quot;:desc&quot; for sorting by reverse time.</td>
</tr>
<tr>
<td>--follow</td>
<td>Print events until process is halted</td>
</tr>
</tbody>
</table>

**Table 77.41. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, log, table, value, yaml}, --format {csv, json, log, table, value, yaml}</td>
<td>The output format, defaults to log</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 77.42. CSV formatter options**
Table 77.43. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.9. STACK EVENT SHOW

Show event details.

Usage:

```
```

Table 77.45. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;stack&gt;</code></td>
<td>Name or id of stack to show events for</td>
</tr>
<tr>
<td><code>&lt;resource&gt;</code></td>
<td>Name of the resource event belongs to</td>
</tr>
<tr>
<td><code>&lt;event&gt;</code></td>
<td>Id of event to display details for</td>
</tr>
</tbody>
</table>

Table 77.46. Command arguments
Table 77.47. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

The output format, defaults to table

Specify the column(s) to include, can be repeated to show multiple columns

Table 77.48. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.49. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.50. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.10. STACK EXPORT

Export stack data json.

Usage:

### Table 77.51. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to export</td>
</tr>
</tbody>
</table>

### Table 77.52. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--output-file &lt;output-file&gt;</td>
<td>File to output export data</td>
</tr>
</tbody>
</table>

### Table 77.53. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to json</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 77.54. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 77.55. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 77.56. Table formatter options
77.11. STACK FAILURES LIST

Show information about failed stack resources.

Usage:

openstack stack failures list [-h] [--long] <stack>

Table 77.57. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack to display (name or id)</td>
</tr>
</tbody>
</table>

Table 77.58. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>Show full deployment logs in output</td>
</tr>
</tbody>
</table>

77.12. STACK FILE LIST

Show a stack’s files map.

Usage:


Table 77.59. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;NAME or ID&gt;</td>
<td>Name or id of stack to query</td>
</tr>
</tbody>
</table>

**Table 77.60. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 77.61. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td></td>
<td>{json,shell,table,value,yaml}</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 77.62. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 77.63. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 77.64. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
### 77.13. STACK HOOK CLEAR

Clear resource hooks on a given stack.

**Usage:**

```
openstack stack hook clear [-h] [--pre-create] [--pre-update]
                           [--pre-delete]
                           <stack> <resource> [<resource> ...]
```

**Table 77.65. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack to display (name or id)</td>
</tr>
</tbody>
</table>
| <resource>| Resource names with hooks to clear. resources in nested stacks can be set using slash as a separator: `\`nested_stack/another/my_resource``. You can use wildcards to match multiple stacks or resources: `\`nested_stack/an*/resource` `\`

**Table 77.66. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pre-create</td>
<td>Clear the pre-create hooks</td>
</tr>
<tr>
<td>--pre-update</td>
<td>Clear the pre-update hooks</td>
</tr>
<tr>
<td>--pre-delete</td>
<td>Clear the pre-delete hooks</td>
</tr>
</tbody>
</table>

### 77.14. STACK HOOK POLL

List resources with pending hook for a stack.

**Usage:**

```
openstack stack hook poll [-h] [-f {csv,json,table,value,yaml}]
                          [-c COLUMN]
                          [--quote {all,minimal,none,nonnumeric}]
                          [-o noindent] [-m width <integer>]
                          [-f width] [--print-empty]
                          [--sort-column SORT_COLUMN]
                          [--sort-ascending | --sort-descending]
                          [-r nested-depth <nested-depth>]
                          <stack>
```

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
### Table 77.67. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack to display (name or id)</td>
</tr>
</tbody>
</table>

### Table 77.68. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--nested-depth &lt;nested-depth&gt;</td>
<td>Depth of nested stacks from which to display hooks</td>
</tr>
</tbody>
</table>

### Table 77.69. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 77.70. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 77.71. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 77.72. Table formatter options
---max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty | Print empty table if there is no data to show.

### 77.15. STACK LIST

List stacks.

#### Usage:

```
```

#### Table 77.73. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--deleted</td>
<td>Include soft-deleted stacks in the stack listing</td>
</tr>
<tr>
<td>--nested</td>
<td>Include nested stacks in the stack listing</td>
</tr>
<tr>
<td>--hidden</td>
<td>Include hidden stacks in the stack listing</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Filter properties to apply on returned stacks (repeat to filter on multiple properties)</td>
</tr>
<tr>
<td>--tags &lt;tag1,tag2...&gt;</td>
<td>List of tags to filter by. can be combined with --tag-mode to specify how to filter tags</td>
</tr>
</tbody>
</table>
Table 77.74. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--tag-mode &lt;mode&gt;</td>
<td>Method of filtering tags. must be one of &quot;any&quot;, &quot;not&quot;, or &quot;not-any&quot;. If not specified, multiple tags will be combined with the boolean AND expression</td>
</tr>
<tr>
<td>--limit &lt;limit&gt;</td>
<td>The number of stacks returned</td>
</tr>
<tr>
<td>--marker &lt;id&gt;</td>
<td>Only return stacks that appear after the given id</td>
</tr>
<tr>
<td>--sort &lt;key&gt;[:&lt;direction&gt;]</td>
<td>Sort output by selected keys and directions (asc or desc) (default: asc). Specify multiple times to sort on multiple properties</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Include all projects (admin only)</td>
</tr>
<tr>
<td>--short</td>
<td>List fewer fields in output</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output, this is implied by --all-projects</td>
</tr>
</tbody>
</table>

Table 77.75. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 77.76. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
**Table 77.77. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 77.16. STACK OUTPUT LIST

List stack outputs.

Usage:

```
```

**Table 77.78. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;stack&gt;</code></td>
<td>Name or id of stack to query</td>
</tr>
</tbody>
</table>

**Table 77.79. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 77.80. Output formatter options**
Table 77.81. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.82. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.83. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.17. STACK OUTPUT SHOW

Show stack output.

Usage:

Table 77.84. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to query</td>
</tr>
<tr>
<td>&lt;output&gt;</td>
<td>Name of an output to display</td>
</tr>
</tbody>
</table>

Table 77.85. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all</td>
<td>Display all stack outputs</td>
</tr>
</tbody>
</table>

Table 77.86. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.87. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.88. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.89. Table formatter options
**Value** | **Summary**
--- | ---
--max-width <integer> | Maximum display width, <1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

--fit-width | Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.

--print-empty | Print empty table if there is no data to show.

### 77.18. STACK RESOURCE LIST

List stack resources.

**Usage:**

```bash
```

Table 77.90. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to query</td>
</tr>
</tbody>
</table>

Table 77.91. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>Enable detailed information presented for each resource in resource list</td>
</tr>
<tr>
<td>-n &lt;nested-depth&gt;, --nested-depth &lt;nested-depth&gt;</td>
<td>Depth of nested stacks from which to display resources</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--filter &lt;key=value&gt;</td>
<td>Filter parameters to apply on returned resources based on their name, status, type, action, id and physical_resource_id</td>
</tr>
</tbody>
</table>

Table 77.92. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, dot, json, table, value, yaml}, --format {csv, dot, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 77.93. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.94. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.95. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
77.19. STACK RESOURCE MARK UNHEALTHY

Set resource’s health.

Usage:

```
openstack stack resource mark unhealthy [-h] [--reset]  
<stack> <resource> [reason]
```

Table 77.96. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack the resource belongs to</td>
</tr>
<tr>
<td>&lt;resource&gt;</td>
<td>Name of the resource</td>
</tr>
<tr>
<td>reason</td>
<td>Reason for state change</td>
</tr>
</tbody>
</table>

Table 77.97. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--reset</td>
<td>Set the resource as healthy</td>
</tr>
</tbody>
</table>

77.20. STACK RESOURCE METADATA

Show resource metadata

Usage:

```
openstack stack resource metadata [-h]  
[ -f {json,shell,table,value,yaml} ]  
[ -c COLUMN ] [ --noindent ]  
[ --prefix PREFIX ]  
[ --max-width <integer> ] [ --fit-width ]  
[ --print-empty ]  
<stack> <resource>
```

Table 77.98. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;stack&gt;</code></td>
<td>Stack to display (name or id)</td>
</tr>
<tr>
<td><code>&lt;resource&gt;</code></td>
<td>Name of the resource to show the metadata for</td>
</tr>
</tbody>
</table>

**Table 77.99. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 77.100. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to json</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 77.101. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 77.102. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 77.103. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
77.21. STACK RESOURCE SHOW

Display stack resource.

Usage:

```
```

Table 77.104. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to query</td>
</tr>
<tr>
<td>&lt;resource&gt;</td>
<td>Name of resource</td>
</tr>
</tbody>
</table>

Table 77.105. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--with-attr &lt;attribute&gt;</td>
<td>Attribute to show, can be specified multiple times</td>
</tr>
</tbody>
</table>

Table 77.106. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.107. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
Table 77.108. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.109. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.22. STACK RESOURCE SIGNAL

Signal a resource with optional data.

Usage:

```
openstack stack resource signal [-h] [--data <data>] [--data-file <data-file>] <stack> <resource>
```

Table 77.110. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack the resource belongs to</td>
</tr>
<tr>
<td>&lt;resource&gt;</td>
<td>Name of the resource to signal</td>
</tr>
</tbody>
</table>

Table 77.111. Command arguments
### 77.23. STACK RESUME

Resume a stack.

**Usage:**

```bash
```

**Table 77.112. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack(s) to resume (name or id)</td>
</tr>
</tbody>
</table>

**Table 77.113. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for resume to complete</td>
</tr>
</tbody>
</table>

**Table 77.114. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 77.115. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.116. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.117. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.24. STACK SHOW

Show stack details.

Usage:

```bash
openstack stack show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width]
```
Table 77.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack to display (name or id)</td>
</tr>
</tbody>
</table>

Table 77.119. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--no-resolve-outputs</td>
<td>Do not resolve outputs of the stack.</td>
</tr>
</tbody>
</table>

Table 77.120. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.121. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.122. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.123. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
---

**Value** | **Summary**
---|---
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty | Print empty table if there is no data to show.

### 77.25. STACK SNAPSHOT CREATE

Create stack snapshot.

**Usage:**

```bash
```

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of snapshot</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Value</strong></th>
<th><strong>Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

---
### Table 77.128. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 77.129. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 77.26. STACK SNAPSHOT DELETE

Delete stack snapshot.

**Usage:**

```
openstack stack snapshot delete [-h] [-y] <stack> <snapshot>
```

### Table 77.130. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack</td>
</tr>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Id of stack snapshot</td>
</tr>
</tbody>
</table>

### Table 77.131. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
77.27. STACK SNAPSHOT LIST

List stack snapshots.

Usage:

```
```

Table 77.132. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack containing the snapshots</td>
</tr>
</tbody>
</table>

Table 77.133. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 77.134. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>
Table 77.135. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.136. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.137. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.28. STACK SNAPSHOT RESTORE

Restore stack snapshot

Usage:

```
openstack stack snapshot restore [-h] <stack> <snapshot>
```

Table 77.138. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack containing the snapshot</td>
</tr>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Id of the snapshot to restore</td>
</tr>
</tbody>
</table>

Table 77.139. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
77.29. STACK SNAPSHOT SHOW

Show stack snapshot.

Usage:

```
openstack stack snapshot show [-h] [-f {json,shell,table,value,yaml}] 
[-c COLUMN] [--noindent] 
[--prefix PREFIX] [--max-width <integer>] 
[--fit-width] [--print-empty] 
<stack> <snapshot>
```

Table 77.140. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack containing the snapshot</td>
</tr>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Id of the snapshot to show</td>
</tr>
</tbody>
</table>

Table 77.141. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 77.142. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.143. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.144. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
### 77.30. STACK SUSPEND

Suspend a stack.

**Usage:**

```bash
```

#### Table 77.146. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Stack(s) to suspend (name or id)</td>
</tr>
</tbody>
</table>

#### Table 77.147. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--wait</td>
<td>Wait for suspend to complete</td>
</tr>
</tbody>
</table>

#### Table 77.148. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>

---

### Table 77.145. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &quot;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

---
The output format, defaults to table

Specify the column(s) to include, can be repeated to show multiple columns

Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

Sort the column(s) in ascending order

Sort the column(s) in descending order

Table 77.149. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 77.150. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.151. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

77.31. STACK TEMPLATE SHOW

Display stack template.

Usage:

<stack>

### Table 77.152. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;stack&gt;</td>
<td>Name or id of stack to query</td>
</tr>
</tbody>
</table>

### Table 77.153. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### Table 77.154. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to yaml</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 77.155. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 77.156. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 77.157. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
--fit-width  
Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

--print-empty  
Print empty table if there is no data to show.

## 77.32. STACK UPDATE

Update a stack.

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-t &lt;template&gt;, --template &lt;template&gt;</td>
<td>Path to the template</td>
</tr>
<tr>
<td>-s &lt;files-container&gt;, --files-container &lt;files-container&gt;</td>
<td>Swift files container name. local files other than root template would be ignored. If other files are not found in swift, heat engine would raise an error.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td><code>--environment &lt;environment&gt;</code></td>
<td>Path to the environment. Can be specified multiple times</td>
</tr>
<tr>
<td><code>--pre-update &lt;resource&gt;</code></td>
<td>Name of a resource to set a pre-update hook to. Resources in nested stacks can be set using slash as a separator: <code>nested_stack/another/my_resource</code>. You can use wildcards to match multiple stacks or resources: <code>nested_stack/an*/_resource</code>. This can be specified multiple times</td>
</tr>
<tr>
<td><code>--timeout &lt;timeout&gt;</code></td>
<td>Stack update timeout in minutes</td>
</tr>
<tr>
<td><code>--rollback &lt;value&gt;</code></td>
<td>Set rollback on update failure. Value &quot;enabled&quot; sets rollback to enabled. Value &quot;disabled&quot; sets rollback to disabled. Value &quot;keep&quot; uses the value of existing stack to be updated (default)</td>
</tr>
<tr>
<td><code>--dry-run</code></td>
<td>Do not actually perform the stack update, but show what would be changed</td>
</tr>
<tr>
<td><code>--show-nested</code></td>
<td>Show nested stacks when performing --dry-run</td>
</tr>
<tr>
<td><code>--parameter &lt;key=value&gt;</code></td>
<td>Parameter values used to create the stack. This can be specified multiple times</td>
</tr>
<tr>
<td><code>--parameter-file &lt;key=file&gt;</code></td>
<td>Parameter values from file used to create the stack. This can be specified multiple times. Parameter value would be the content of the file</td>
</tr>
<tr>
<td><code>--existing</code></td>
<td>Re-use the template, parameters and environment of the current stack. If the template argument is omitted then the existing template is used. If no --environment is specified then the existing environment is used. Parameters specified in --parameter will patch over the existing values in the current stack. Parameters omitted will keep the existing values</td>
</tr>
<tr>
<td><code>--clear-parameter &lt;parameter&gt;</code></td>
<td>Remove the parameters from the set of parameters of current stack for the stack-update. The default value in the template will be used. This can be specified multiple times</td>
</tr>
<tr>
<td><code>--tags &lt;tag1,tag2...&gt;</code></td>
<td>An updated list of tags to associate with the stack</td>
</tr>
</tbody>
</table>
--wait

Wait until stack goes to update_complete or UPDATE_FAILED

--converge

Stack update with observe on reality.

Table 77.160. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 77.161. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 77.162. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 77.163. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 78. SUBNET

This chapter describes the commands under the subnet command.

78.1. SUBNET CREATE

Create a subnet

Usage:


Table 78.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New subnet name</td>
</tr>
</tbody>
</table>

Table 78.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner's project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--subnet-pool &lt;subnet-pool&gt;</td>
<td>Subnet pool from which this subnet will obtain a cidr (Name or ID)</td>
</tr>
<tr>
<td>--use-prefix-delegation USE_PREFIX_DELEGATION</td>
<td>Use prefix-delegation if ip is ipv6 format and ip would be delegated externally</td>
</tr>
<tr>
<td>--use-default-subnet-pool</td>
<td>Use default subnet pool for --ip-version</td>
</tr>
<tr>
<td>--prefix-length &lt;prefix-length&gt;</td>
<td>Prefix length for subnet allocation from subnet pool</td>
</tr>
<tr>
<td>--subnet-range &lt;subnet-range&gt;</td>
<td>Subnet range in cidr notation (required if --subnet-pool is not specified, optional otherwise)</td>
</tr>
<tr>
<td>--dhcp</td>
<td>Enable dhcp (default)</td>
</tr>
<tr>
<td>--no-dhcp</td>
<td>Disable dhcp</td>
</tr>
<tr>
<td>--dns-publish-fixed-ip</td>
<td>Enable publishing fixed ips in dns</td>
</tr>
<tr>
<td>--no-dns-publish-fixed-ip</td>
<td>Disable publishing fixed ips in dns (default)</td>
</tr>
<tr>
<td>--gateway &lt;gateway&gt;</td>
<td>Specify a gateway for the subnet. the three options are: &lt;ip-address&gt;: Specific IP address to use as the gateway, auto: Gateway address should automatically be chosen from within the subnet itself, none: This subnet will not use a gateway, e.g.: --gateway 192.168.9.1, --gateway auto, --gateway none (default is auto).</td>
</tr>
<tr>
<td>--ip-version {4,6}</td>
<td>Ip version (default is 4). note that when subnet pool is specified, IP version is determined from the subnet pool and this option is ignored.</td>
</tr>
<tr>
<td>--ipv6-ra-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}</td>
<td>Ipv6 ra (router advertisement) mode, valid modes: [dhcpv6-stateful, dhcpv6-stateless, slaac]</td>
</tr>
<tr>
<td>--ipv6-address-mode {dhcpv6-stateful,dhcpv6-stateless,slaac}</td>
<td>Ipv6 address mode, valid modes: [dhcpv6-stateful, dhcpv6-stateless, slaac]</td>
</tr>
<tr>
<td>--network-segment &lt;network-segment&gt;</td>
<td>Network segment to associate with this subnet (name or ID)</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>Network this subnet belongs to (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set subnet description</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--allocation-pool start=&lt;ip-address&gt;,end=&lt;ip-address&gt;</td>
<td>Allocation pool ip addresses for this subnet e.g.: start=192.168.199.2,end=192.168.199.254 (repeat option to add multiple IP addresses)</td>
</tr>
<tr>
<td>--dns-nameserver &lt;dns-nameserver&gt;</td>
<td>Dns server for this subnet (repeat option to set multiple DNS servers)</td>
</tr>
<tr>
<td>--host-route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Additional route for this subnet e.g.: destination=10.10.0.0/16,gateway=192.168.71.254 destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to add multiple routes)</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type for this subnet e.g.: network:floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to set multiple service types)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the subnet (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the subnet</td>
</tr>
</tbody>
</table>

Table 78.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 78.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 78.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 78.6. Table formatter options
### 78.2. SUBNET DELETE

Delete subnet(s)

**Usage:**

```bash
openstack subnet delete [-h] <subnet> [<subnet> ...]
```

**Table 78.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

**Table 78.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 78.3. SUBNET LIST

List subnets

**Usage:**

```bash
```
Table 78.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--ip-version &lt;ip-version&gt;</td>
<td>List only subnets of given ip version in output. Allowed values for IP version are 4 and 6.</td>
</tr>
<tr>
<td>--dhcp</td>
<td>List subnets which have dhcp enabled</td>
</tr>
<tr>
<td>--no-dhcp</td>
<td>List subnets which have dhcp disabled</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>List only subnets of a given service type in output</td>
</tr>
<tr>
<td></td>
<td>e.g.: network: floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to list multiple service types)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List only subnets which belong to a given project in output (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--network &lt;network&gt;</td>
<td>List only subnets which belong to a given network in output (name or ID)</td>
</tr>
<tr>
<td>--gateway &lt;gateway&gt;</td>
<td>List only subnets of given gateway ip in output</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List only subnets of given name in output</td>
</tr>
<tr>
<td>--subnet-range &lt;subnet-range&gt;</td>
<td>List only subnets of given subnet range (in cidr notation) in output e.g.: --subnet-range 10.10.0.0/16</td>
</tr>
<tr>
<td>--tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List subnets which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--any-tags &lt;tag&gt;[,&lt;tag&gt;,...]</td>
<td>List subnets which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>
Table 78.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--not-tags <code>&lt;tag&gt;,&lt;tag&gt;,...</code></td>
<td>Exclude subnets which have all given tag(s) (comma-separated list of tags)</td>
</tr>
<tr>
<td>--not-any-tags <code>&lt;tag&gt;,&lt;tag&gt;,...</code></td>
<td>Exclude subnets which have any given tag(s) (comma-separated list of tags)</td>
</tr>
</tbody>
</table>

Table 78.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 78.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 78.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width <code>&lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
78.4. SUBNET POOL CREATE

Create subnet pool

Usage:

```
```

Table 78.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Name of the new subnet pool</td>
</tr>
</tbody>
</table>

Table 78.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pool-prefix &lt;pool-prefix&gt;</td>
<td>Set subnet pool prefixes (in cidr notation) (repeat option to set multiple prefixes)</td>
</tr>
<tr>
<td>--default-prefix-length &lt;default-prefix-length&gt;</td>
<td>Set subnet pool default prefix length</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--min-prefix-length &lt;min-prefix-length&gt;</td>
<td>Set subnet pool minimum prefix length</td>
</tr>
<tr>
<td>--max-prefix-length &lt;max-prefix-length&gt;</td>
<td>Set subnet pool maximum prefix length</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Owner’s project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--address-scope &lt;address-scope&gt;</td>
<td>Set address scope associated with the subnet pool (name or ID), prefixes must be unique across address scopes</td>
</tr>
<tr>
<td>--default</td>
<td>Set this as a default subnet pool</td>
</tr>
<tr>
<td>--no-default</td>
<td>Set this as a non-default subnet pool</td>
</tr>
<tr>
<td>--share</td>
<td>Set this subnet pool as shared</td>
</tr>
<tr>
<td>--no-share</td>
<td>Set this subnet pool as not shared</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set subnet pool description</td>
</tr>
<tr>
<td>--default-quota &lt;num-ip-addresses&gt;</td>
<td>Set default per-project quota for this subnet pool as the number of IP addresses that can be allocated from the subnet pool</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the subnet pool (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>No tags associated with the subnet pool</td>
</tr>
</tbody>
</table>

Table 78.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 78.17. JSON formatter options
Table 78.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 78.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

78.5. SUBNET POOL DELETE

Delete subnet pool(s)

Usage:

openstack subnet pool delete [-h] <subnet-pool> [<subnet-pool> ...]

Table 78.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet-pool&gt;</td>
<td>Subnet pool(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 78.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

78.6. SUBNET POOL LIST
List subnet pools

Usage:

```
openstack subnet pool list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
[--quote {all, minimal, none, nonnumeric}] [--noindent] [--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[-l] [--share | --no-share]
[-d | --no-default]
[--project <project>]
[--project-domain <project-domain>]
[--name <name>]
[--address-scope <address-scope>]
[--tags <tag>[, <tag>,...]]
[--any-tags <tag>[, <tag>,...]]
[--not-tags <tag>[, <tag>,...]]
[--not-any-tags <tag>[, <tag>,...]]
```

Table 78.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--share</td>
<td>List subnet pools shared between projects</td>
</tr>
<tr>
<td>--no-share</td>
<td>List subnet pools not shared between projects</td>
</tr>
<tr>
<td>--default</td>
<td>List subnet pools used as the default external subnet pool</td>
</tr>
<tr>
<td>--no-default</td>
<td>List subnet pools not used as the default external subnet pool</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>List subnet pools according to their project (name or ID)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id), this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>List only subnet pools of given name in output</td>
</tr>
<tr>
<td>--address-scope &lt;address-scope&gt;</td>
<td>List only subnet pools of given address scope in output (name or ID)</td>
</tr>
</tbody>
</table>
### Table 78.23. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 78.24. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 78.25. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 78.26. Table formatter options
78.7. SUBNET POOL SET

Set subnet pool properties

Usage:

```
openstack subnet pool set [-h] [--name <name>]
                         [--pool-prefix <pool-prefix>]
                         [--default-prefix-length <default-prefix-length>]
                         [--min-prefix-length <min-prefix-length>]
                         [--max-prefix-length <max-prefix-length>]
                         [--address-scope <address-scope> | --no-address-scope]
                         [--default | --no-default]
                         [--description <description>]
                         [--default-quota <num-ip-addresses>]
                         [--tag <tag>] [--no-tag]
                         <subnet-pool>
```

Table 78.27. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet-pool&gt;</td>
<td>Subnet pool to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 78.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set subnet pool name</td>
</tr>
<tr>
<td>--pool-prefix &lt;pool-prefix&gt;</td>
<td>Set subnet pool prefixes (in cidr notation) (repeat option to set multiple prefixes)</td>
</tr>
<tr>
<td>--default-prefix-length &lt;default-prefix-length&gt;</td>
<td>Set subnet pool default prefix length</td>
</tr>
</tbody>
</table>
### Value | Summary
---|---
--min-prefix-length <min-prefix-length> | Set subnet pool minimum prefix length

--max-prefix-length <max-prefix-length> | Set subnet pool maximum prefix length

--address-scope <address-scope> | Set address scope associated with the subnet pool (name or ID), prefixes must be unique across address scopes

--no-address-scope | Remove address scope associated with the subnet pool

--default | Set this as a default subnet pool

--no-default | Set this as a non-default subnet pool

--description <description> | Set subnet pool description

--default-quota <num-ip-addresses> | Set default per-project quota for this subnet pool as the number of IP addresses that can be allocated from the subnet pool

--tag <tag> | Tag to be added to the subnet pool (repeat option to set multiple tags)

--no-tag | Clear tags associated with the subnet pool. Specify both --tag and --no-tag to overwrite current tags

---

### 78.8. SUBNET POOL SHOW

Display subnet pool details

**Usage:**

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet-pool&gt;</td>
<td>Subnet pool to display (name or id)</td>
</tr>
</tbody>
</table>

---

Table 78.29. Positional arguments

Table 78.30. Command arguments
Table 78.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 78.32. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 78.33. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 78.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

78.9. SUBNET POOL UNSET

Unset subnet pool properties

Usage:
openstack subnet pool unset [-h] [--tag <tag> | --all-tag] <subnet-pool>

Table 78.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet-pool&gt;</td>
<td>Subnet pool to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 78.36. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the subnet pool (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the subnet pool</td>
</tr>
</tbody>
</table>

78.10. SUBNET SET

Set subnet properties

Usage:

openstack subnet set [-h] [--name <name>] [--dhcp | --no-dhcp]
[--dns-publish-fixed-ip | --no-dns-publish-fixed-ip]
[--gateway <gateway>]
[--network-segment <network-segment>]
[--description <description>] [--tag <tag>]
[--no-tag]
[--allocation-pool start=<ip-address>,end=<ip-address>]
[--no-allocation-pool]
[--dns-nameserver <dns-nameserver>]
[--no-dns-nameservers]
[--host-route destination=<subnet>,gateway=<ip-address>]
[--no-host-route] [--service-type <service-type>]
<subnet>

Table 78.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 78.38. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Updated name of the subnet</td>
</tr>
<tr>
<td>--dhcp</td>
<td>Enable dhcp</td>
</tr>
<tr>
<td>--no-dhcp</td>
<td>Disable dhcp</td>
</tr>
<tr>
<td>--dns-publish-fixed-ip</td>
<td>Enable publishing fixed ips in dns</td>
</tr>
<tr>
<td>--no-dns-publish-fixed-ip</td>
<td>Disable publishing fixed ips in dns</td>
</tr>
<tr>
<td>--gateway &lt;gateway&gt;</td>
<td>Specify a gateway for the subnet, the options are:</td>
</tr>
<tr>
<td></td>
<td>&lt;ip-address&gt;: Specific IP address to use as the gateway, none: This</td>
</tr>
<tr>
<td></td>
<td>subnet will not use a gateway, e.g.: --gateway 192.168.9.1, --gateway</td>
</tr>
<tr>
<td></td>
<td>none.</td>
</tr>
<tr>
<td>--network-segment &lt;network-segment&gt;</td>
<td>Network segment to associate with this subnet (name or ID). It is only</td>
</tr>
<tr>
<td></td>
<td>allowed to set the segment if the current value is None, the network</td>
</tr>
<tr>
<td></td>
<td>must also have only one segment and only one subnet can exist on the</td>
</tr>
<tr>
<td></td>
<td>network.</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set subnet description</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be added to the subnet (repeat option to set multiple tags)</td>
</tr>
<tr>
<td>--no-tag</td>
<td>Clear tags associated with the subnet. specify both --tag and</td>
</tr>
<tr>
<td></td>
<td>--no-tag to overwrite current tags</td>
</tr>
<tr>
<td>--allocation-pool start=&lt;ip-address&gt;,end=</td>
<td>Allocation pool ip addresses for this subnet e.g.: start=192.168.199.2,</td>
</tr>
<tr>
<td>&lt;ip-address&gt;</td>
<td>end=192.168.199.254 (repeat option to add multiple IP addresses)</td>
</tr>
<tr>
<td>--no-allocation-pool</td>
<td>Clear associated allocation-pools from the subnet. Specify both --</td>
</tr>
<tr>
<td></td>
<td>allocation-pool and --no-allocation-pool to overwrite the current</td>
</tr>
<tr>
<td></td>
<td>allocation pool information.</td>
</tr>
<tr>
<td>--dns-nameserver &lt;dns-nameserver&gt;</td>
<td>Dns server for this subnet (repeat option to set multiple DNS servers)</td>
</tr>
<tr>
<td>--no-dns-nameservers</td>
<td>Clear existing information of dns nameservers. specify both --dns-</td>
</tr>
<tr>
<td></td>
<td>nameserver and --no-dns-nameserver to overwrite the current DNS Nameserver information.</td>
</tr>
</tbody>
</table>
### 78.11. SUBNET SHOW

Display subnet details

#### Usage:

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--host-route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Additional route for this subnet e.g.: destination=10.10.0.0/16,gateway=192.168.71.254 destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to add multiple routes)</td>
</tr>
<tr>
<td>--no-host-route</td>
<td>Clear associated host-routes from the subnet. specify both --host-route and --no-host-route to overwrite the current host route information.</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type for this subnet e.g.: network:floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to set multiple service types)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--host-route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Additional route for this subnet e.g.: destination=10.10.0.0/16,gateway=192.168.71.254 destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to add multiple routes)</td>
</tr>
<tr>
<td>--no-host-route</td>
<td>Clear associated host-routes from the subnet. specify both --host-route and --no-host-route to overwrite the current host route information.</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type for this subnet e.g.: network:floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to set multiple service types)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--host-route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Additional route for this subnet e.g.: destination=10.10.0.0/16,gateway=192.168.71.254 destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to add multiple routes)</td>
</tr>
<tr>
<td>--no-host-route</td>
<td>Clear associated host-routes from the subnet. specify both --host-route and --no-host-route to overwrite the current host route information.</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type for this subnet e.g.: network:floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to set multiple service types)</td>
</tr>
</tbody>
</table>

#### Table 78.39. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet to display (name or id)</td>
</tr>
</tbody>
</table>

#### Table 78.40. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### Table 78.41. Output formatter options
### Table 78.42. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the JSON</td>
</tr>
</tbody>
</table>

### Table 78.43. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 78.44. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 78.12. SUBNET UNSET

Unset subnet properties

**Usage:**

```
openstack subnet unset [-h]  
  [-allocation-pool start=<ip-address>,end=<ip-address>]  
  [-gateway] [-dns-nameserver <dns-nameserver>]  
  [-host-route destination=<subnet>,gateway=<ip-address>]  
```
Table 78.45. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;subnet&gt;</td>
<td>Subnet to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 78.46. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--allocation-pool start=&lt;ip-address&gt;,end=&lt;ip-address&gt;</td>
<td>Allocation pool ip addresses to be removed from this subnet e.g.: start=192.168.199.2,end=192.168.199.254 (repeat option to unset multiple allocation pools)</td>
</tr>
<tr>
<td>--gateway</td>
<td>Remove gateway ip from this subnet</td>
</tr>
<tr>
<td>--dns-nameserver &lt;dns-nameserver&gt;</td>
<td>Dns server to be removed from this subnet (repeat option to unset multiple DNS servers)</td>
</tr>
<tr>
<td>--host-route destination=&lt;subnet&gt;,gateway=&lt;ip-address&gt;</td>
<td>Route to be removed from this subnet e.g.: destination=10.10.0.0/16,gateway=192.168.71.254 destination: destination subnet (in CIDR notation) gateway: nexthop IP address (repeat option to unset multiple host routes)</td>
</tr>
<tr>
<td>--service-type &lt;service-type&gt;</td>
<td>Service type to be removed from this subnet e.g.: network:floatingip_agent_gateway. Must be a valid device owner value for a network port (repeat option to unset multiple service types)</td>
</tr>
<tr>
<td>--tag &lt;tag&gt;</td>
<td>Tag to be removed from the subnet (repeat option to remove multiple tags)</td>
</tr>
<tr>
<td>--all-tag</td>
<td>Clear all tags associated with the subnet</td>
</tr>
</tbody>
</table>
CHAPTER 79. TASK

This chapter describes the commands under the `task` command.

79.1. TASK EXECUTION LIST

List all tasks.

Usage:

```
```

Table 79.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow_execution</td>
<td>Workflow execution id associated with list of tasks.</td>
</tr>
</tbody>
</table>

Table 79.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--oldest</td>
<td>Display the executions starting from the oldest entries instead of the newest</td>
</tr>
</tbody>
</table>

**Table 79.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 79.4. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 79.5. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 79.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
**79.2. TASK EXECUTION PUBLISHED SHOW**

Show task published variables.

**Usage:**

```
openstack task execution published show [-h] id
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Task id</td>
</tr>
</tbody>
</table>

**Table 79.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 79.8. Command arguments**

---

**79.3. TASK EXECUTION RERUN**

Rerun an existing task.

**Usage:**

```
openstack task execution rerun [-h] [-f {json,shell,table,value,yaml}] 
[-c COLUMN] [-noindent] 
[-prefix PREFIX] 
[-max-width <integer>] [-fit-width] 
[-print-empty] [-resume] [-e ENV] id
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Task identifier</td>
</tr>
</tbody>
</table>

**Table 79.9. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 79.10. Command arguments**
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--resume</td>
<td>Rerun only failed or unstarted action executions for with-items task</td>
</tr>
<tr>
<td>-e ENV, --env ENV</td>
<td>Environment variables</td>
</tr>
</tbody>
</table>

Table 79.11. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 79.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 79.13. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 79.14. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
79.4. TASK EXECUTION RESULT SHOW

Show task output data.

Usage:

```bash
openstack task execution result show [-h] id
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Task id</td>
</tr>
</tbody>
</table>

79.5. TASK EXECUTION SHOW

Show specific task.

Usage:

```bash
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>task</td>
<td>Task identifier</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 79.19. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 79.20. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 79.21. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 79.22. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 80. TLD

This chapter describes the commands under the `tld` command.

80.1. TLD CREATE

Create new tld

Usage:


Table 80.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Tld name</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 80.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 80.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 80.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 80.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

80.2. TLD DELETE

Delete tld

Usage:

openstack tld delete [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] id

Table 80.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tld name or id</td>
</tr>
</tbody>
</table>

Table 80.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>
80.3. TLD LIST

List tlds

Usage:

openstack tld list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN] 
[-quote {all,minimal,none,nonnumeric}] [--noindent] 
[--max-width <integer>] [--fit-width] 
[--print-empty] [-s {sort-column SORT_COLUMN}] 
[-sort-ascending | --sort-descending] [--name NAME] 
[-description DESCRIPTION] [-all-projects] 
[--sudo-project-id SUDO_PROJECT_ID]

Table 80.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Tld name</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Tld description</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 80.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 80.10. CSV formatter options
80.4. TLD SET

Set tld properties

Usage:


Table 80.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tld name or id</td>
</tr>
</tbody>
</table>

Table 80.14. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Tld name</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--no-description--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 80.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 80.16. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 80.17. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 80.18. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
80.5. TLD SHOW

Show tld details

Usage:

```
openstack tld show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN]
                 [--noindent] [--prefix PREFIX]
                 [--max-width <integer>] [--fit-width]
                 [--print-empty] [--all-projects]
                 [--sudo-project-id SUDO_PROJECT_ID]

id
```

Table 80.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tld name or id</td>
</tr>
</tbody>
</table>

Table 80.20. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 80.21. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 80.22. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 80.23. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 80.24. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
This chapter describes the commands under the `token` command.

### 81.1. TOKEN ISSUE

Issue new token

**Usage:**

```
```

**Table 81.1. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 81.2. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 81.3. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 81.4. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
### 81.2. TOKEN REVOKE

Revoke existing token

**Usage:**

```
openstack token revoke [-h] <token>
```

**Table 81.6. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;token&gt;</td>
<td>Token to be deleted</td>
</tr>
</tbody>
</table>

**Table 81.7. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
CHAPTER 82. TRIPLEO

This chapter describes the commands under the tripleo command.

82.1. TRIPLEO CONFIG GENERATE ANSIBLE

Generate the default ansible.cfg for deployments.

Usage:

openstack tripleo config generate ansible
    [--deployment-user DEPLOYMENT_USER]
    [--output-dir OUTPUT_DIR]

Table 82.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--deployment-user DEPLOYMENT_USER</td>
<td>User who executes the tripleo config generate command. Defaults to stack.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to output ansible.cfg and ansible.log files.</td>
</tr>
</tbody>
</table>

82.2. TRIPLEO CONTAINER IMAGE BUILD

Build tripleo container images with tripleo-ansible.

Usage:

openstack tripleo container image build [-h] [--authfile <authfile>]
    [--base <base-image>]
    [--config-file <config-file>]
    [--config-path <config-path>]
    [--distro <distro>]
    [--tcib-extras <key=val>]
    [--exclude <container-name>]
    [--extra-config <extra-config>]
    [--namespace <registry-namespace>]
    [--registry <registry-url>]
    [--skip-build]
    [--tag <image-tag>]
    [--prefix <image-prefix>]
    [--push] [--label <label-data>]
    [--volume <volume-path>]
    [--repo-dir <repo-dir>]
    [--work-dir <work-directory>]
    [--rhel-modules <rhel-modules>]
    [--build-timeout <build timeout in seconds>]

Table 82.2. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--authfile &lt;authfile&gt;</td>
<td>Path of the authentication file. Use REGISTRY_AUTH_FILE environment variable to override. (default: /root/containers/auth.json)</td>
</tr>
<tr>
<td>--base &lt;base-image&gt;</td>
<td>Base image name, with optional version. Can be centos:8, base name image will be centos but centos:8 will be pulled to build the base image. (default: ubi8)</td>
</tr>
<tr>
<td>--config-file &lt;config-file&gt;</td>
<td>Yaml config file specifying the images to build. (default: tripleo_containers.yaml)</td>
</tr>
<tr>
<td>--config-path &lt;config-path&gt;</td>
<td>Base configuration path. This is the base path for all container-image files. The defined containers must reside within a tcib folder that is in this path. If this option is set, the default path for &lt;config-file&gt; will be modified. (default: /usr/share/tripleo-common/container-images)</td>
</tr>
<tr>
<td>--distro &lt;distro&gt;</td>
<td>Distro name, if undefined the system will build using the host distro. (default: centos)</td>
</tr>
<tr>
<td>--tcib-extras &lt;key=val&gt;</td>
<td>Tcib extra variables you want to pass. They can be later used within TCIB files as conditionals. Can be passed multiple times (default: None)</td>
</tr>
<tr>
<td>--exclude &lt;container-name&gt;</td>
<td>Name of one container to match against the list of containers to be built to skip. Should be specified multiple times when skipping multiple containers. (default: [])</td>
</tr>
<tr>
<td>--extra-config &lt;extra-config&gt;</td>
<td>Apply additional options from a given configuration YAML file. This will apply to all containers built. (default: None)</td>
</tr>
<tr>
<td>--namespace &lt;registry-namespace&gt;</td>
<td>Container registry namespace (default: tripleomaster)</td>
</tr>
<tr>
<td>--registry &lt;registry-url&gt;</td>
<td>Container registry url (default: localhost)</td>
</tr>
<tr>
<td>--skip-build</td>
<td>Skip or not the build of the images (default: false)</td>
</tr>
<tr>
<td>--tag &lt;image-tag&gt;</td>
<td>Image tag (default: latest)</td>
</tr>
<tr>
<td>--prefix &lt;image-prefix&gt;</td>
<td>Image prefix. (default: openstack)</td>
</tr>
</tbody>
</table>
82.3. TRIPLEO CONTAINER IMAGE DELETE

Delete specified image from registry.

Usage:

```
openstack tripleo container image delete [-h]
    [--registry-url <registry url>]
    [--username <username>]
    [--password <password>] [-y]
    <image to delete>
```

Table 82.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image to delete&gt;</td>
<td>Full url of image to be deleted in the form &lt;fqdn&gt;:&lt;port&gt;/path/to/image</td>
</tr>
</tbody>
</table>

Table 82.4. Command arguments
### 82.4. TRIPLEO CONTAINER IMAGE HOTFIX

Hotfix tripleo container images with tripleo-ansible.

**Usage:**

```
openstack tripleo container image hotfix [-h] --image <images> --rpms-path <rpms-path> [--tag <image-tag>]
```

**Table 82.5. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--image &lt;images&gt;</td>
<td>Fully qualified reference to the source image to be modified. Can be specified multiple times (one per image) (default: []).</td>
</tr>
<tr>
<td>--rpms-path &lt;rpms-path&gt;</td>
<td>Path containing rpms to install (default: none).</td>
</tr>
<tr>
<td>--tag &lt;image-tag&gt;</td>
<td>Image hotfix tag (default: latest)</td>
</tr>
</tbody>
</table>

### 82.5. TRIPLEO CONTAINER IMAGE LIST

List images discovered in registry.

**Usage:**

```
openstack tripleo container image list [-h]
  [-l {csv, json, table, value, yaml}]
  [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [-noindent]
  [-max-width <integer>]
```
Table 82.6. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--registry-url &lt;registry url&gt;</td>
<td>Url of registry images are to be listed from in the form &lt;fqdn&gt;:&lt;port&gt;.</td>
</tr>
<tr>
<td>--username &lt;username&gt;</td>
<td>Username for image registry.</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Password for image registry.</td>
</tr>
</tbody>
</table>

Table 82.7. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 82.8. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 82.9. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 82.10. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also</td>
</tr>
<tr>
<td></td>
<td>use the CLIFF_MAX_TERM_WIDTH environment variable, but the</td>
</tr>
<tr>
<td></td>
<td>parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width</td>
</tr>
<tr>
<td></td>
<td>greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1</td>
</tr>
<tr>
<td></td>
<td>to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**82.6. TRIPLEO CONTAINER IMAGE PREPARE DEFAULT**

Generate a default ContainerImagePrepare parameter.

**Usage:**

```
openstack tripleo container image prepare default [-h]
    [--output-env-file <file path>]
    [-llocal-push-destination]
    [--enable-registry-login]
```

**Table 82.11. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--output-env-file &lt;file path&gt;</td>
<td>File to write environment file containing default ContainerImagePrepare value.</td>
</tr>
<tr>
<td>--local-push-destination</td>
<td>Include a push_destination to trigger upload to a local registry.</td>
</tr>
<tr>
<td>--enable-registry-login</td>
<td>Use this flag to enable the flag to have systems attempt to login to a remote registry prior to pulling their containers. This flag should be used when --local-push-destination is NOT used and the target systems will have network connectivity to the remote registries. Do not use this for an overcloud that may not have network connectivity to a remote registry.</td>
</tr>
</tbody>
</table>
82.7. TRIPLEO CONTAINER IMAGE PREPARE

Prepare and upload containers from a single command.

Usage:

```bash
openstack tripleo container image prepare [-h]
   [--environment-file <file path>]
   [--environment-directory <HEAT ENVIRONMENT DIRECTORY>]
   [--roles-file ROLES_FILE]
   [--output-env-file <file path>]
   [--dry-run]
   [--cleanup <full, partial, none>]
   [--log-file LOG_FILE]
```

Table 82.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--environment-file &lt;file path&gt;, -e &lt;file path&gt;</td>
<td>Environment file containing the containerimageprepare parameter which specifies all prepare actions. Also, environment files specifying which services are containerized. Entries will be filtered to only contain images used by containerized services. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--environment-directory &lt;HEAT ENVIRONMENT DIRECTORY&gt;</td>
<td>Environment file directories that are automatically added to the environment. Can be specified more than once. Files in directories are loaded in ascending sort order.</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data.yaml in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--output-env-file &lt;file path&gt;</td>
<td>File to write heat environment file which specifies all image parameters. Any existing file will be overwritten.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Do not perform any pull, modify, or push operations. The environment file will still be populated as if these operations were performed.</td>
</tr>
<tr>
<td>--cleanup &lt;full, partial, none&gt;</td>
<td>Cleanup behavior for local images left after upload. The default full will attempt to delete all local images. partial will leave images required for deployment on this host. none will do no cleanup.</td>
</tr>
</tbody>
</table>
82.8. TRIPLEO CONTAINER IMAGE PUSH

Push specified image to registry.

Usage:

```
openstack tripleo container image push [-h] [--local]
[-registry-url <registry url>]
[--append-tag APPEND_TAG]
[--username <username>]
[--password <password>]
[--source-username <source_username>]
[--source-password <source_password>]
[--dry-run] [--multi-arch]
[--cleanup]
<image to push>
```

Table 82.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image to push&gt;</td>
<td>Container image to upload. should be in the form of &lt;registry&gt;/&lt;namespace&gt;/&lt;name&gt;:&lt;tag&gt;. If tag is not provided, then latest will be used.</td>
</tr>
</tbody>
</table>

Table 82.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--local</td>
<td>Use this flag if the container image is already on the current system and does not need to be pulled from a remote registry.</td>
</tr>
<tr>
<td>--registry-url &lt;registry url&gt;</td>
<td>Url of the destination registry in the form &lt;fqdn&gt;:&lt;port&gt;.</td>
</tr>
<tr>
<td>--append-tag APPEND_TAG</td>
<td>Tag to append to the existing tag when pushing the container.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--username &lt;username&gt;</td>
<td>Username for the destination image registry.</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Password for the destination image registry.</td>
</tr>
<tr>
<td>--source-username &lt;source_username&gt;</td>
<td>Username for the source image registry.</td>
</tr>
<tr>
<td>--source-password &lt;source_password&gt;</td>
<td>Password for the source image registry.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Perform a dry run upload. the upload action is not performed, but the authentication process is attempted.</td>
</tr>
<tr>
<td>--multi-arch</td>
<td>Enable multi arch support for the upload.</td>
</tr>
<tr>
<td>--cleanup</td>
<td>Remove local copy of the image after uploading</td>
</tr>
</tbody>
</table>

### 82.9. TRIPLEO CONTAINER IMAGE SHOW

Show image selected from the registry.

**Usage:**

```
```

**Table 82.15. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;image to inspect&gt;</td>
<td>Image to be inspected, for example: docker.io/library/centos:7 or docker://docker.io/library/centos:7</td>
</tr>
</tbody>
</table>

**Table 82.16. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
Table 82.17. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--username &lt;username&gt;</td>
<td>Username for image registry.</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Password for image registry.</td>
</tr>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to json</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 82.18. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 82.19. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 82.20. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

82.10. TRIPLEO DEPLOY

Deploy containerized Undercloud
Usage:

openstack tripleo deploy [--templates [TEMPLATES]] [--standalone]
  [--upgrade] [-y] [--stack STACK]
  [--output-dir OUTPUT_DIR] [--output-only]
  [--standalone-role STANDALONE_ROLE]
  [-t <TIMEOUT>] [-e <HEAT_ENVIRONMENT_FILE>]
  [--roles-file ROLES_FILE]
  [--networks-file NETWORKS_FILE]
  [--plan-environment-file PLAN_ENVIRONMENT_FILE]
  [--heat-api-port <HEAT_API_PORT>]
  [--heat-user <HEAT_USER>]
  [--deployment-user DEPLOYMENT_USER]
  [--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER]
  [--heat-container-image <HEAT_CONTAINER_IMAGE>]
  [--heat-native <HEAT_NATIVE>]
  [--local-ip <LOCAL_IP>]
  [--control-virtual-ip <CONTROL_VIRTUAL_IP>]
  [--public-virtual-ip <PUBLIC_VIRTUAL_IP>]
  [--local-domain <LOCAL_DOMAIN>] [--cleanup]
  [--hieradata-override [HIERADATA_OVERRIDE]]
  [--keep-running] [--inflight-validations]
  [--transport TRANSPORT]
  [--ansible-forks ANSIBLE_FORKS]
  [--disable-container-prepare]
  [--reproduce-command]
  [--force-stack-update] [--force-stack-create]

Table 82.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--standalone</td>
<td>Deprecated. the --standalone argument is now deprecated. Standalone deployments can now be run without passing --standalone.</td>
</tr>
<tr>
<td>--upgrade</td>
<td>Upgrade an existing deployment.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name for the ephemeral (one-time create and forget) heat stack.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to output state, processed heat templates, ansible deployment files. Defaults to ~/tripleo-deploy/&lt;stack&gt;</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--output-only</td>
<td>Do not execute the ansible playbooks. by default the playbooks are saved to the output-dir and then executed.</td>
</tr>
<tr>
<td>--standalone-role STANDALONE_ROLE</td>
<td>The role to use for standalone configuration when populating the deployment actions.</td>
</tr>
<tr>
<td>-t &lt;TIMEOUT&gt;, --timeout &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes.</td>
</tr>
<tr>
<td>-e &lt;HEAT ENVIRONMENT FILE&gt;, --environment-file &lt;HEAT ENVIRONMENT FILE&gt;</td>
<td>Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data_undercloud.yaml in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--networks-file NETWORKS_FILE, -n NETWORKS_FILE</td>
<td>Roles file, overrides the default /dev/null in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--plan-environment-file PLAN_ENVIRONMENT_FILE, -p PLAN_ENVIRONMENT_FILE</td>
<td>Deprecated: plan environment file, not supported</td>
</tr>
<tr>
<td>--heat-api-port &lt;HEAT_API_PORT&gt;</td>
<td>Heat api port to use for the installers private heat API instance. Optional. Default: 8006.)</td>
</tr>
<tr>
<td>--heat-user &lt;HEAT_USER&gt;</td>
<td>User to execute the non-privileged heat-all process. Defaults to the value of --deployment-user.</td>
</tr>
<tr>
<td>--deployment-user DEPLOYMENT_USER</td>
<td>User who executes the tripleo deploy command. defaults to $SUDO_USER. If $SUDO_USER is unset it defaults to stack.</td>
</tr>
<tr>
<td>--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER</td>
<td>The path to python interpreter to use for the deployment actions. If not specified the python version of the openstackclient will be used. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td>--heat-container-image &lt;HEAT_CONTAINER_IMAGE&gt;</td>
<td>The container image to use when launching the heat-all process. Defaults to: registry.redhat.io/rhosp-rhel9/openstack-heat-all:17.0</td>
</tr>
</tbody>
</table>
--heat-native [HEAT_NATIVE]

Execute the heat-all process natively on this host. This option requires that the heat-all binaries be installed locally on this machine. This option is enabled by default which means heat-all is executed on the host OS directly.

--local-ip <LOCAL_IP>

Local ip/cidr for undercloud traffic. required.

--control-virtual-ip <CONTROL_VIRTUAL_IP>

Control plane vip. this allows the undercloud installer to configure a custom VIP on the control plane.

--public-virtual-ip <PUBLIC_VIRTUAL_IP>

Public nw vip. this allows the undercloud installer to configure a custom VIP on the public (external) NW.

--local-domain <LOCAL_DOMAIN>

Local domain for standalone cloud and its api endpoints

--cleanup

Cleanup temporary files. using this flag will remove the temporary files used during deployment in after the command is run.

--hieradata-override [HIERADATA_OVERRIDE]

Path to hieradata override file. when it points to a heat env file, it is passed in t-h-t via --environment-file. When the file contains legacy instack data, it is wrapped with <role>ExtraConfig and also passed in for t-h-t as a temp file created in --output-dir. Note, instack hiera data may be not t-h-t compatible and will highly likely require a manual revision.

--keep-running

Keep the ephemeral heat running after the stack operation is complete. This is for debugging purposes only. The ephemeral Heat can be used by openstackclient with: OS_AUTH_TYPE=none OS_ENDPOINT=http://127.0.0.1:8006/v1/admin openstack stack list where 8006 is the port specified by --heat-api-port.

--inflight-validations

Activate in-flight validations during the deploy. in-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.

--transport TRANSPORT

Transport mechanism to use for ansible. use "ssh" for multinode deployments. Use "local" for standalone deployments. Defaults to "local".

--ansible-forks ANSIBLE_FORKS

The number of ansible forks to use for the config-download ansible-playbook command.
---disable-container-prepare
Disable the container preparation actions to prevent
container tags from being updated and new
containers from being fetched. If you skip this but do
not have the container parameters configured, the
deployment action may fail.

--reproduce-command
Create a reproducer command with ansible
commandline and all environments variables.

--force-stack-update
Do a virtual update of the ephemeral heat stack (it
cannot take real updates). New or failed
deployments always have the stack_action=CREATE.
This option enforces stack_action=UPDATE.

--force-stack-create
Do a virtual create of the ephemeral heat stack. new
or failed deployments always have the
stack_action=CREATE. This option enforces
stack_action=CREATE.

---tripleo launch heat
Launch ephemeral Heat process.

Usage:

openstack tripleo launch heat [--heat-api-port <HEAT_API_PORT>]
[--heat-user <HEAT_USER>]
[--heat-container-image <HEAT_CONTAINER_IMAGE>]
[--heat-container-api-image <HEAT_CONTAINER_API_IMAGE>]
[--heat-container-engine-image <HEAT_CONTAINER_ENGINE_IMAGE>]
[--kill] [--heat-dir HEAT_DIR]
[--rm-heat] [--skip-heat-pull]
[--restore-db]
[--heat-native | --heat-type {native,container,pod}]

Table 82.22. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--heat-api-port &lt;HEAT_API_PORT&gt;</td>
<td>Heat api port to use for the installers private heat API instance. Optional. (default: 8006)</td>
</tr>
<tr>
<td>--heat-user &lt;HEAT_USER&gt;</td>
<td>User to execute the non-privileged heat-all process. Defaults to current user. If the configuration files /etc/heat/heat.conf or /usr/share/heat/heat-dist.conf exist, the user must have read access to those files. This option is ignored when using --heat-type=container or --heat-type=pod (default: root)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--heat-container-image &lt;HEAT_CONTAINER_IMAGE&gt;</td>
<td>The container image to use when launching the heat-all process. Defaults to: localhost/tripleo/openstack-heat-all:ephemeral (default: localhost/tripleo/openstack-heat-all:ephemeral)</td>
</tr>
<tr>
<td>--kill, -k</td>
<td>Kill the running heat process (if found). (default: False)</td>
</tr>
<tr>
<td>--heat-dir HEAT_DIR</td>
<td>Directory to use for file storage and logs of the running heat process. in the current directory. Can be set to an already existing directory to reuse the environment from a previous Heat process. (default: /root/overcloud-deploy/overcloud/heat-launcher)</td>
</tr>
<tr>
<td>--rm-heat</td>
<td>If specified and --heat-type is container or pod any existing container or pod of a previous ephemeral Heat process will be deleted first. Ignored if --heat-type is native or --kill. (default: False)</td>
</tr>
<tr>
<td>--skip-heat-pull</td>
<td>When --heat-type is pod or container, assume the container image has already been pulled (default: False)</td>
</tr>
<tr>
<td>--restore-db</td>
<td>Restore a database dump if it exists within the directory specified by --heat-dir (default: False)</td>
</tr>
<tr>
<td>--heat-native</td>
<td>(deprecated): execute the heat-all process natively on this host. This option requires that the heat-all binaries be installed locally on this machine. This option is enabled by default which means heat-all is executed on the host OS directly. Conflicts with --heat-type, which deprecates --heat-native. (default: False)</td>
</tr>
</tbody>
</table>
82.12. TRIPLEO UPGRADE

Upgrade TripleO

Usage:

```
openstack tripleo upgrade [--templates [TEMPLATES]] [--standalone]
[-t <TIMEOUT>] [--output-only] [--standalone-role STANDALONE_ROLE]
[-y] [--heat-api-port <HEAT_API_PORT>] [--roles-file ROLES_FILE]
[-e <HEAT ENVIRONMENT FILE>] [--networks-file NETWORKS_FILE]
[-plan-environment-file PLAN_ENVIRONMENT_FILE] [--heat-user <HEAT_USER>]
[-deployment-user DEPLOYMENT_USER] [--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER]
[-heat-container-image <HEAT_CONTAINER_IMAGE>] [--heat-native [HEAT_NATIVE]]
[-local-ip <LOCAL_IP>] [--control-virtual-ip <CONTROL_VIRTUAL_IP>]
[-public-virtual-ip <PUBLIC_VIRTUAL_IP>] [--local-domain <LOCAL_DOMAIN>] [--cleanup]
[-hieradata-override [HIERADATA_OVERRIDE]]
[--keep-running] [--inflight-validations]
[--transport TRANSPORT] [--ansible-forks ANSIBLE_FORKS]
[--deployment-python-interpreter DEPLOYMENT_PYTHON_INTERPRETER]
[--disable-container-prepare]
[--reproduce-command]
[--force-stack-update | --force-stack-create]
```

Table 82.23. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--templates [TEMPLATES]</td>
<td>The directory containing the heat templates to deploy</td>
</tr>
<tr>
<td>--standalone</td>
<td>Deprecated. the --standalone argument is now deprecated. Standalone deployments can now be run without passing --standalone.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--upgrade</td>
<td>Upgrade an existing deployment.</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
<tr>
<td>--stack STACK</td>
<td>Name for the ephemeral (one-time create and forget) heat stack.</td>
</tr>
<tr>
<td>--output-dir OUTPUT_DIR</td>
<td>Directory to output state, processed heat templates, ansible deployment files. Defaults to ~/tripleo-deploy/&lt;stack&gt;</td>
</tr>
<tr>
<td>--output-only</td>
<td>Do not execute the ansible playbooks. By default the playbooks are saved to the output-dir and then executed.</td>
</tr>
<tr>
<td>--standalone-role STANDALONE_ROLE</td>
<td>The role to use for standalone configuration when populating the deployment actions.</td>
</tr>
<tr>
<td>-t &lt;TIMEOUT&gt;, --timeout &lt;TIMEOUT&gt;</td>
<td>Deployment timeout in minutes.</td>
</tr>
<tr>
<td>-e &lt;HEAT ENVIRONMENT FILE&gt;, --environment-file &lt;HEAT ENVIRONMENT FILE&gt;</td>
<td>Environment files to be passed to the heat stack-create or heat stack-update command. (Can be specified more than once.)</td>
</tr>
<tr>
<td>--roles-file ROLES_FILE, -r ROLES_FILE</td>
<td>Roles file, overrides the default roles_data_undercloud.yaml in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--networks-file NETWORKS_FILE, -n NETWORKS_FILE</td>
<td>Roles file, overrides the default /dev/null in the t-h-t templates directory used for deployment. May be an absolute path or the path relative to the templates dir.</td>
</tr>
<tr>
<td>--plan-environment-file PLAN_ENVIRONMENT_FILE, -p PLAN_ENVIRONMENT_FILE</td>
<td>Deprecated: plan environment file, not supported</td>
</tr>
<tr>
<td>--heat-api-port &lt;HEAT_API_PORT&gt;</td>
<td>Heat api port to use for the installers private heat API instance. Optional. Default: 8006.)</td>
</tr>
<tr>
<td>--heat-user &lt;HEAT_USER&gt;</td>
<td>User to execute the non-privileged heat-all process. Defaults to the value of --deployment-user.</td>
</tr>
<tr>
<td>--deployment-user DEPLOYMENT_USER</td>
<td>User who executes the tripleo deploy command. Defaults to $SUDO_USER. If $SUDO_USER is unset it defaults to stack.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--deployment-python-interpreter&lt;br&gt;DEPLOYMENT_PYTHON_INTERPRETER</td>
<td>The path to python interpreter to use for the deployment actions. If not specified the python version of the openstackclient will be used. This may need to be used if deploying on a python2 host from a python3 system or vice versa.</td>
</tr>
<tr>
<td>--heat-container-image&lt;br&gt;&lt;HEAT_CONTAINER_IMAGE&gt;</td>
<td>The container image to use when launching the heat-all process. Defaults to: registry.redhat.io/rhosp-rhel9/openstack-heat-all:17.0</td>
</tr>
<tr>
<td>--heat-native [HEAT_NATIVE]</td>
<td>Execute the heat-all process natively on this host. This option requires that the heat-all binaries be installed locally on this machine. This option is enabled by default which means heat-all is executed on the host OS directly.</td>
</tr>
<tr>
<td>--local-ip &lt;LOCAL_IP&gt;</td>
<td>Local ip/cidr for undercloud traffic. required.</td>
</tr>
<tr>
<td>--control-virtual-ip &lt;CONTROL_VIRTUAL_IP&gt;</td>
<td>Control plane vip. this allows the undercloud installer to configure a custom VIP on the control plane.</td>
</tr>
<tr>
<td>--public-virtual-ip &lt;PUBLIC_VIRTUAL_IP&gt;</td>
<td>Public nw vip. this allows the undercloud installer to configure a custom VIP on the public (external) NW.</td>
</tr>
<tr>
<td>--local-domain &lt;LOCAL_DOMAIN&gt;</td>
<td>Local domain for standalone cloud and its api endpoints</td>
</tr>
<tr>
<td>--cleanup</td>
<td>Cleanup temporary files. using this flag will remove the temporary files used during deployment in after the command is run.</td>
</tr>
<tr>
<td>--hieradata-override [HIERADATA_OVERRIDE]</td>
<td>Path to hieradata override file. when it points to a heat env file, it is passed in t-h-t via --environment-file. When the file contains legacy instack data, it is wrapped with &lt;role&gt;ExtraConfig and also passed in for t-h-t as a temp file created in --output-dir. Note, instack hiera data may be not t-h-t compatible and will highly likely require a manual revision.</td>
</tr>
<tr>
<td>--keep-running</td>
<td>Keep the ephemeral heat running after the stack operation is complete. This is for debugging purposes only. Only the ephemeral Heat can be used by openstackclient with: OS_AUTH_TYPE=none OS_ENDPOINT=<a href="http://127.0.0.1:8006/v1/admin">http://127.0.0.1:8006/v1/admin</a> openstack stack list where 8006 is the port specified by --heat-api-port.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--inflight-validations</td>
<td>Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td>--transport TRANSPORT</td>
<td>Transport mechanism to use for ansible. Use &quot;ssh&quot; for multinode deployments. Use &quot;local&quot; for standalone deployments. Defaults to &quot;local&quot;.</td>
</tr>
<tr>
<td>--ansible-forks ANSIBLE_FORKS</td>
<td>The number of ansible forks to use for the config-download ansible-playbook command.</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--reproduce-command</td>
<td>Create a reproducer command with ansible commandline and all environments variables.</td>
</tr>
<tr>
<td>--force-stack-update</td>
<td>Do a virtual update of the ephemeral heat stack (it cannot take real updates). New or failed deployments always have the stack_action=CREATE. This option enforces stack_action=UPDATE.</td>
</tr>
<tr>
<td>--force-stack-create</td>
<td>Do a virtual create of the ephemeral heat stack. New or failed deployments always have the stack_action=CREATE. This option enforces stack_action=CREATE.</td>
</tr>
</tbody>
</table>

### 82.13. TRIPLEO VALIDATOR GROUP INFO

[DEPRECATED]: Display detailed information about a Group. Please use "validation show group --help" instead.

Usage:

```bash
```
Table 82.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table (default: table)</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric (default: nonnumeric)</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework. (default: /etc/validation.cfg)</td>
</tr>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where validation playbooks are located. (default: /usr/share/ansible/validation-playbooks)</td>
</tr>
</tbody>
</table>

82.14. TRIPLEO VALIDATOR INIT

Create the paths and infrastructure to create a community validation

Usage:
openstack tripleo validator init [-h] [--config CONFIG]

[--validation-dir VALIDATION_DIR]
[--ansible-base-dir ANSIBLE_BASE_DIR]

<validation_name>

Table 82.25. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;validation_name&gt;</td>
<td>The name of the community validation: Validation name is limited to contain only lowercase alphanumeric characters, plus _ or - and starts with an alpha character. Ex: my-val, my_val2. This will generate an Ansible role and a playbook in /root/community-validations. Note that the structure of this directory will be created at the first use.</td>
</tr>
</tbody>
</table>

Table 82.26. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework. (default: /etc/validation.cfg)</td>
</tr>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where the validation playbooks is located. (default: /usr/share/ansible/validation-playbooks)</td>
</tr>
<tr>
<td>--ansible-base-dir ANSIBLE_BASE_DIR</td>
<td>Path where the ansible roles, library and plugins are located. (default: /usr/share/ansible)</td>
</tr>
</tbody>
</table>

82.15. TRIPLEO VALIDATOR LIST

[DEPRECATED]: List the available validations. Please use "validation list --help" instead.

Usage:

openstack tripleo validator list [-h] [-f {csv, json, table, value, yaml}]
[[-c COLUMN]]
[[-quote {all, minimal, none, nonnumeric}]
[
[[-noindent] [--max-width <integer>]
[[-fil-width] [--print-empty]]
[[-sort-column SORT_COLUMN] [--sort-ascending] [--sort-descending]]
[[-config CONFIG]]
[[-group <group_id>[,<group_id>,...]]
[[-category <category_id>[,<category_id>,...]]
[[-product <product_id>[,<product_id>,...]]
[[-validation-dir VALIDATION_DIR]]
# Table 82.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table (default: table)</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--quote {all,minimal,none,nonnumeric}</code></td>
<td>When to include quotes, defaults to nonnumeric (default: nonnumeric)</td>
</tr>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td><code>--sort-ascending</code></td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td><code>--sort-descending</code></td>
<td>Sort the column(s) in descending order</td>
</tr>
<tr>
<td><code>--config CONFIG</code></td>
<td>Config file path for validation framework. (default: /etc/validation.cfg)</td>
</tr>
<tr>
<td><code>--group &lt;group_id&gt;[,&lt;group_id&gt;;...], -g &lt;group_id&gt;[,&lt;group_id&gt;;...]</code></td>
<td>List specific group of validations, if more than one group is required separate the group names with commas.</td>
</tr>
<tr>
<td><code>--category &lt;category_id&gt;[,&lt;category_id&gt;;...]</code></td>
<td>List specific category of validations, if more than one category is required separate the category names with commas.</td>
</tr>
<tr>
<td><code>--product &lt;product_id&gt;[,&lt;product_id&gt;;...]</code></td>
<td>List specific product of validations, if more than one product is required separate the product names with commas.</td>
</tr>
</tbody>
</table>
82.16. TRIPLEO VALIDATOR RUN

[DEPRECATED]: Run the available validations. Please use "validation run --help" instead.

Usage:

```
openstack tripleo validator run [-h] [--config CONFIG] 
    [--limit <host1>,<host2>,<host3>,...]]
    [--ssh-user SSH_USER]
    [--validation-dir VALIDATION_DIR]
    [--ansible-base-dir ANSIBLE_BASE_DIR]
    [--validation-log-dir VALIDATION_LOG_DIR]
    [--inventory INVENTORY]
    [--output-log OUTPUT_LOG]
    [--junitxml JUNITXML]
    [--python-interpreter --python-interpreter
<PYTHON_INTERPRETER_PATH>]
    [--extra-env-vars key1=<val1> [--extra-env-vars key2=<val2>]]
    [--skiplist SKIP_LIST]
    [--extra-vars key1=<val1> [--extra-vars key2=<val2>]
    [--extra-vars-file
    /tmp/my_vars_file.[json|yaml]]
    (--validation <validation_id>[,<validation_id>,...]) | --group <group_id>[,
    <group_id>,...]] | --category <category_id>[,<category_id>,...]) | --product <product_id>[,
    <product_id>,...])
```

Table 82.28. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where validation playbooks are located. (default: /usr/share/ansible/validation-playbooks)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework. (default: /etc/validation.cfg)</td>
</tr>
<tr>
<td>--limit &lt;host1&gt;,&lt;host2&gt;,&lt;host3&gt;,...</td>
<td>A string that identifies a single node or comma-separated list of nodes to be validated in this run invocation.</td>
</tr>
<tr>
<td>--ssh-user SSH_USER</td>
<td>Ssh user name for the ansible ssh connection. (default: root)</td>
</tr>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where validation playbooks are located. (default: /usr/share/ansible/validation-playbooks)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--ansible-base-dir ANSIBLE_BASE_DIR</td>
<td>Path where the ansible roles, library and plugins are located. (default: <code>/usr/share/ansible</code>)</td>
</tr>
<tr>
<td>--validation-log-dir VALIDATION_LOG_DIR</td>
<td>Path where the log files and artifacts are located. (default: <code>/var/log/validations</code>)</td>
</tr>
<tr>
<td>--inventory INVENTORY, -i INVENTORY</td>
<td>Path of the ansible inventory. (default: <code>localhost</code>)</td>
</tr>
<tr>
<td>--output-log OUTPUT_LOG</td>
<td>Path where the run result will be stored.</td>
</tr>
<tr>
<td>--junitxml JUNITXML</td>
<td>Path where the run result in junitxml format will be stored.</td>
</tr>
<tr>
<td>--python-interpreter --python-interpreter &lt;PYTHON_INTERPRETER_PATH&gt;</td>
<td>Python interpreter for ansible execution. (default: <code>/usr/bin/python3</code>)</td>
</tr>
<tr>
<td>--extra-env-vars key1=&lt;val1&gt; [---extra-env-vars key2=&lt;val2&gt;]</td>
<td>Add extra environment variables you may need to provide to your Ansible execution as KEY=VALUE pairs. Note that if you pass the same KEY multiple times, the last given VALUE for that same KEY will override the other(s).</td>
</tr>
<tr>
<td>--skiplist SKIP_LIST</td>
<td>Path where the skip list is stored. An example of the skiplist format could be found at the root of the validations-libs repository.</td>
</tr>
<tr>
<td>--extra-vars key1=&lt;val1&gt; [---extra-vars key2=&lt;val2&gt;]</td>
<td>Add ansible extra variables to the validation(s) execution as KEY=VALUE pair(s). Note that if you pass the same KEY multiple times, the last given VALUE for that same KEY will override the other(s).</td>
</tr>
<tr>
<td>--extra-vars-file /tmp/my_vars_file.[json</td>
<td>yaml]</td>
</tr>
<tr>
<td>--validation &lt;validation_id&gt;[,&lt;validation_id&gt;,...]</td>
<td>Run specific validations, if more than one validation is required separate the names with commas.</td>
</tr>
<tr>
<td>--group &lt;group_id&gt;[,&lt;group_id&gt;,...] -g &lt;group_id&gt;[, &lt;group_id&gt;,...]</td>
<td>Run specific validations by group, if more than one group is required separate the group names with commas.</td>
</tr>
<tr>
<td>--category &lt;category_id&gt;[,&lt;category_id&gt;,...]</td>
<td>Run specific validations by category, if more than one category is required separate the category names with commas.</td>
</tr>
</tbody>
</table>
Run specific validations by product, if more than one product is required separate the product names with commas.

### 82.17. TRIPLEO VALIDATOR SHOW HISTORY

[DEPRECATED]: Display Validations execution history. Please use "validation history list --help" instead.

#### Usage:

```
openstack tripleo validator show history [-h]
  [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending]
  [--sort-descending]
  [--config CONFIG]
  [--validation <validation_id>]
  [--limit HISTORY_LIMIT]
  [--validation-log-dir VALIDATION_LOG_DIR]
```

#### Table 82.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table (default: table)</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric (default: nonnumeric)</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework. (default: /etc/validation.cfg)</td>
</tr>
<tr>
<td>--validation &lt;validation_id&gt;</td>
<td>Display execution history for a validation</td>
</tr>
<tr>
<td>--limit HISTORY_LIMIT</td>
<td>Display &lt;n&gt; most recent runs of the selected &lt;validation&gt;. &lt;n&gt; must be &gt; 0 The default display limit is set to 15.</td>
</tr>
<tr>
<td>--validation-log-dir VALIDATION_LOG_DIR</td>
<td>Path where the log files and artifacts are located. (default: /var/log/validations)</td>
</tr>
</tbody>
</table>

### 82.18. TRIPLEO VALIDATOR SHOW PARAMETER

[DEPRECATED]: Display Validations Parameters. Please use "validation show parameter --help" instead.

**Usage:**

```
openstack tripleo validator show parameter [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [-noindent]
  [-prefix PREFIX]
  [-max-width <integer>]
  [--fit-width]
  [-print-empty]
  [--config CONFIG]
  [--validation-dir VALIDATION_DIR]
  [--validation <validation_id>[,<validation_id>,...]]
  | --group
  | <group_id>[,<group_id>,...]
  | --category
  | <category_id>[,<category_id>,...]
  | --product
```
Table 82.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework.</td>
</tr>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where validation playbooks are located.</td>
</tr>
<tr>
<td>--validation &lt;validation_id&gt;[&lt;validation_id&gt;,...]]</td>
<td>List specific validations, if more than one validation is required separate the names with commas.</td>
</tr>
<tr>
<td>--group &lt;group_id&gt;[&lt;group_id&gt;,...], -g &lt;group_id&gt;[&lt;group_id&gt;,...]</td>
<td>List specific group of validations, if more than one group is required separate the group names with commas.</td>
</tr>
<tr>
<td>--category &lt;category_id&gt;[&lt;category_id&gt;,...]</td>
<td>List specific category of validations, if more than one category is required separate the category names with commas.</td>
</tr>
<tr>
<td>--product &lt;product_id&gt;[&lt;product_id&gt;,...]</td>
<td>List specific product of validations, if more than one product is required separate the product names with commas.</td>
</tr>
<tr>
<td>--download DOWNLOAD</td>
<td>Create a json or a yaml file containing all the variables available for the validations: /tmp/myvars</td>
</tr>
<tr>
<td>--format-output &lt;format_output&gt;</td>
<td>Print representation of the validation. the choices of the output format is json,yaml.</td>
</tr>
</tbody>
</table>

Table 82.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 82.32. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 82.33. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 82.34. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 82.19. TRIPLEO VALIDATOR SHOW RUN

[DEPRECATED]: Display details about a Validation execution. Please use "validation history get --help" instead.

**Usage:**

```
openstack tripleo validator show run [-h] [--config CONFIG] [--full]
    [--validation-log-dir VALIDATION_LOG_DIR]
    <uuid>
```

**Table 82.35. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;uuid&gt;</td>
<td>Validation uuid run</td>
</tr>
</tbody>
</table>

**Table 82.36. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
82.20. TRIPLEO VALIDATOR SHOW

[DEPRECATED]: Display detailed information about a Validation. Please use "validation show --help" instead.

Usage:

```
openstack tripleo validator show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>] [--fit-width]
    [--print-empty] [--config CONFIG]
    [--validation-dir VALIDATION_DIR]
    <validation>
```

Table 82.37. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;validation&gt;</td>
<td>Show a specific validation.</td>
</tr>
</tbody>
</table>

Table 82.38. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework.</td>
</tr>
<tr>
<td>--validation-dir VALIDATION_DIR</td>
<td>Path where validation playbooks are located.</td>
</tr>
</tbody>
</table>

Table 82.39. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--config CONFIG</td>
<td>Config file path for validation framework.</td>
</tr>
<tr>
<td>--validation-log-dir VALIDATION_LOG_DIR</td>
<td>Path where the log files and artifacts are located. (default: /var/log/validations)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 82.40. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 82.41. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 82.42. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 83. TRUST

This chapter describes the commands under the trust command.

83.1. TRUST CREATE

Create new trust

Usage:

```
```

Table 83.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trustor-user&gt;</td>
<td>User that is delegating authorization (name or id)</td>
</tr>
<tr>
<td>&lt;trustee-user&gt;</td>
<td>User that is assuming authorization (name or id)</td>
</tr>
</tbody>
</table>

Table 83.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Project being delegated (name or id) (required)</td>
</tr>
<tr>
<td>--role &lt;role&gt;</td>
<td>Roles to authorize (name or id) (repeat option to set multiple values, required)</td>
</tr>
<tr>
<td>--impersonate</td>
<td>Tokens generated from the trust will represent &lt;trustor&gt; (defaults to False)</td>
</tr>
<tr>
<td>--expiration &lt;expiration&gt;</td>
<td>Sets an expiration date for the trust (format of yyyy-mm-ddTHH:MM:SS)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>
Table 83.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--trustor-domain &lt;trustor-domain&gt;</td>
<td>Domain that contains &lt;trustor&gt; (name or id)</td>
</tr>
<tr>
<td>--trustee-domain &lt;trustee-domain&gt;</td>
<td>Domain that contains &lt;trustee&gt; (name or id)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple</td>
</tr>
<tr>
<td></td>
<td>columns</td>
</tr>
</tbody>
</table>

Table 83.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 83.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 83.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

83.2. TRUST DELETE

Delete trust(s)
### 83.3. TRUST LIST

List trusts

#### Usage:

```
```

#### Table 83.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

#### Table 83.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>

---

**Table 83.7. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;trust&gt;</td>
<td>Trust(s) to delete</td>
</tr>
</tbody>
</table>

**Table 83.8. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

---
### Table 83.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 83.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 83.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 83.4. TRUST SHOW

Display trust details

**Usage:**

```bash
```

### Table 83.14. Positional arguments
Value | Summary
--- | ---
<trust> | Trust to display

Table 83.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--h, --help | Show this help message and exit

Table 83.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
|--c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns

Table 83.17. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--noindent | Whether to disable indenting the json

Table 83.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
|--prefix PREFIX | Add a prefix to all variable names

Table 83.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 84. TSIGKEY

This chapter describes the commands under the tsigkey command.

84.1. TSIGKEY CREATE

Create new tsigkey

Usage:


Table 84.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Tsigkey name</td>
</tr>
<tr>
<td>--algorithm ALGORITHM</td>
<td>Tsigkey algorithm</td>
</tr>
<tr>
<td>--secret SECRET</td>
<td>Tsigkey secret</td>
</tr>
<tr>
<td>--scope SCOPE</td>
<td>Tsigkey scope</td>
</tr>
<tr>
<td>--resource-id RESOURCE_ID</td>
<td>Tsigkey resource_id</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 84.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
### Table 84.3. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 84.4. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 84.5. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAXTERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 84.2. TSIGKEY DELETE

Delete tsigkey

**Usage:**

```plaintext
openstack tsigkey delete [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] id
```

### Table 84.6. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tsigkey id</td>
</tr>
</tbody>
</table>

### Table 84.7. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
84.3. TSIGKEY LIST

List tsigkeys

Usage:


Table 84.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 84.9. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
**Table 84.10. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 84.11. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 84.12. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 84.4. TSIGKEY SET

Set tsigkey properties

Usage:

```
openstack tskigkey set [-h] [-f {json,shell,table,value,yaml}]`
```
Table 84.13. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tsigkey id</td>
</tr>
</tbody>
</table>

Table 84.14. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Tsigkey name</td>
</tr>
<tr>
<td>--algorithm ALGORITHM</td>
<td>Tsigkey algorithm</td>
</tr>
<tr>
<td>--secret SECRET</td>
<td>Tsigkey secret</td>
</tr>
<tr>
<td>--scope SCOPE</td>
<td>Tsigkey scope</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 84.15. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>json,shell,table,value,yaml</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Whether to disable indenting the json

Add a prefix to all variable names

Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.

Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable

Print empty table if there is no data to show.

Show tsigkey details

Usage:

```
```

Tsigkey id

---

Table 84.17. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 84.18. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

Table 84.19. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Tsigkey id</td>
</tr>
</tbody>
</table>

Table 84.20. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

**Table 84.21. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 84.22. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 84.23. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 84.24. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 85. UNDERCLOUD

This chapter describes the commands under the `undercloud` command.

85.1. UNDERCLOUD BACKUP

Backup the undercloud

Usage:

```bash
openstack undercloud backup [--init [INIT]] [--setup-nfs]
   [--setup-rear] [--cron] [--db-only]
   [--inventory INVENTORY]
   [--add-path ADD_PATH]
   [--exclude-path EXCLUDE_PATH]
   [--save-swift] [--extra-vars EXTRA_VARS]
```

Table 85.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--init [INIT]</code></td>
<td>Initialize environment for backup, using <code>rear</code> or <code>nfs</code> as args which</td>
</tr>
<tr>
<td></td>
<td>will check for package install and configured ReaR or NFS server.</td>
</tr>
<tr>
<td></td>
<td>Defaults to: <code>rear</code>. i.e. <code>--init rear</code>. WARNING: This flag will be</td>
</tr>
<tr>
<td></td>
<td>deprecated and replaced by <code>--setup-rear</code> and <code>--setup-nfs</code>.</td>
</tr>
<tr>
<td><code>--setup-nfs</code></td>
<td>Setup the <code>nfs</code> server on the backup node which will install required</td>
</tr>
<tr>
<td></td>
<td>packages and configuration on the host <code>BackupNode</code> in the ansible</td>
</tr>
<tr>
<td></td>
<td>inventory.</td>
</tr>
<tr>
<td><code>--setup-rear</code></td>
<td>Setup <code>rear</code> on the <code>undercloud</code> host which will install and configure</td>
</tr>
<tr>
<td></td>
<td><code>ReaR</code>.</td>
</tr>
<tr>
<td><code>--cron</code></td>
<td>Sets up a new cron job that by default will execute a weekly backup</td>
</tr>
<tr>
<td></td>
<td>at Sundays midnight, but that can be customized by using the</td>
</tr>
<tr>
<td></td>
<td><code>tripleo_backup_and_restore_cron</code> extra-var.</td>
</tr>
<tr>
<td><code>--db-only</code></td>
<td>Perform a db backup of the <code>undercloud</code> host. the db backup file will</td>
</tr>
<tr>
<td></td>
<td>be stored in <code>/home/stack</code> with the name <code>openstack-backup-mysql-&lt;timestamp&gt;.sql</code>.</td>
</tr>
<tr>
<td><code>--inventory INVENTORY</code></td>
<td>Tripleo inventory file generated with <code>tripleo-ansible-inventory</code></td>
</tr>
<tr>
<td></td>
<td>command. Defaults to: <code>/root/config-download/overcloud/tripleo-ansible-inventory.yam</code></td>
</tr>
<tr>
<td><code>--add-path ADD_PATH</code></td>
<td>Add additional files to backup. defaults to: <code>/home/stack/ i.e. </code>--add-path /this/is/a/folder/ --add-path /this/is/a/textfile.txt`.</td>
</tr>
<tr>
<td><code>--exclude-path EXCLUDE_PATH</code></td>
<td>Exclude specified paths from backup.</td>
</tr>
<tr>
<td><code>--save-swift</code></td>
<td>Save backup file to swift.</td>
</tr>
<tr>
<td><code>--extra-vars EXTRA_VARS</code></td>
<td>Extra variables for backup command.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--exclude-path EXCLUDE_PATH</td>
<td>Exclude path when performing the undercloud backup, this option can be specified multiple times. Defaults to: none i.e. --exclude-path /this/is/a/folder/ --exclude-path /this/is/a/textfile.txt.</td>
</tr>
<tr>
<td>--save-swift</td>
<td>Save backup to swift. defaults to: false special attention should be taken that Swift itself is backed up if you call this multiple times the backup size will grow exponentially.</td>
</tr>
<tr>
<td>--extra-vars EXTRA_VARS</td>
<td>Set additional variables as dict or as an absolute path of a JSON or YAML file type. i.e. --extra-vars &quot;key&quot;: &quot;val&quot;, &quot;key2&quot;: &quot;val2&quot; i.e. --extra-vars /path/to/my_vars.yaml i.e. --extra-vars /path/to/my_vars.json. For more information about the variables that can be passed, visit: <a href="https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml">https://opendev.org/openstack/tripleo-ansible/src/branch/master/tripleo_ansible/roles/backup_and_restore/defaults/main.yml</a>.</td>
</tr>
</tbody>
</table>

### 85.2. UNDERCLOUD INSTALL

Install and setup the undercloud

**Usage:**

```
```

**Table 85.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--force-stack-update</td>
<td>Do a virtual update of the ephemeral heat stack. new or failed deployments always have the stack_action=CREATE. This option enforces stack_action=UPDATE.</td>
</tr>
<tr>
<td>--no-validations</td>
<td>Do not perform undercloud configuration validations</td>
</tr>
</tbody>
</table>
Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.

Print the install command instead of running it

Skip yes/no prompt (assume yes).

Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.

Create a reproducer command with ansible commandline and all environments variables.

---

85.3. UNDERCLOUD UPGRADE

Upgrade undercloud

Usage:

```
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--force-stack-update</td>
<td>Do a virtual update of the ephemeral heat stack. new or failed deployments always have the stack_action=CREATE. This option enforces stack_action=UPDATE.</td>
</tr>
<tr>
<td>--no-validations</td>
<td>Do not perform undercloud configuration validations</td>
</tr>
<tr>
<td>--inflight-validations</td>
<td>Activate in-flight validations during the deploy. In-flight validations provide a robust way to ensure deployed services are running right after their activation. Defaults to False.</td>
</tr>
<tr>
<td>--dry-run</td>
<td>Print the install command instead of running it</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--reproduce-command</td>
<td>Create a reproducer command with ansible commandline and all environments variables.</td>
</tr>
<tr>
<td>--skip-package-updates</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-y, --yes</td>
<td>Skip yes/no prompt (assume yes).</td>
</tr>
<tr>
<td>--disable-container-prepare</td>
<td>Disable the container preparation actions to prevent container tags from being updated and new containers from being fetched. If you skip this but do not have the container parameters configured, the deployment action may fail.</td>
</tr>
<tr>
<td>--reproduce-command</td>
<td>Create a reproducer command with ansible commandline and all environments variables.</td>
</tr>
<tr>
<td>--skip-package-updates</td>
<td>Flag to skip the package update when performing upgrades and updates</td>
</tr>
</tbody>
</table>
CHAPTER 86. USAGE

This chapter describes the commands under the `usage` command.

86.1. USAGE LIST

List resource usage per project

Usage:

```
openstack usage list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
[--quote {all,minimal,none,nonnumeric}] [--noindent] [--max-width <integer>] [--fit-width]
[--print-empty] [--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--start <start>] [--end <end>]
```

Table 86.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--start &lt;start&gt;</td>
<td>Usage range start date, ex 2012-01-20 (default: 4 weeks ago)</td>
</tr>
<tr>
<td>--end &lt;end&gt;</td>
<td>Usage range end date, ex 2012-01-20 (default: tomorrow)</td>
</tr>
</tbody>
</table>

Table 86.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 86.3. CSV formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 86.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 86.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 86.2. USAGE SHOW

Show resource usage for a single project

**Usage:**

```
openstack usage show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [-m max-width <integer>] [-t fit-width] [--print-empty] [-p project <project>] [-s start <start>] [-e end <end>]
```

**Table 86.6. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Name or id of project to show usage for</td>
</tr>
<tr>
<td>--start &lt;start&gt;</td>
<td>Usage range start date, ex 2012-01-20 (default: 4 weeks ago)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>--end &lt;end&gt;</strong></td>
<td>Usage range end date, ex 2012-01-20 (default: tomorrow)</td>
</tr>
</tbody>
</table>

**Table 86.7. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 86.8. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 86.9. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 86.10. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 87. USER

This chapter describes the commands under the user command.

87.1. USER CREATE

Create new user

Usage:


Table 87.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New user name</td>
</tr>
</tbody>
</table>

Table 87.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Default domain (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Default project (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>--project-domain &lt;project-domain&gt;</code></td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td><code>--password &lt;password&gt;</code></td>
<td>Set user password</td>
</tr>
<tr>
<td><code>--password-prompt</code></td>
<td>Prompt interactively for password</td>
</tr>
<tr>
<td><code>--email &lt;email-address&gt;</code></td>
<td>Set user email address</td>
</tr>
<tr>
<td><code>--description &lt;description&gt;</code></td>
<td>User description</td>
</tr>
<tr>
<td><code>--ignore-lockout-failure-attempts</code></td>
<td>Opt into ignoring the number of times a user has authenticated and locking out the user as a result</td>
</tr>
<tr>
<td><code>--no-ignore-lockout-failure-attempts</code></td>
<td>Opt out of ignoring the number of times a user has authenticated and locking out the user as a result</td>
</tr>
<tr>
<td><code>--ignore-password-expiry</code></td>
<td>Opt into allowing user to continue using passwords that may be expired</td>
</tr>
<tr>
<td><code>--no-ignore-password-expiry</code></td>
<td>Opt out of allowing user to continue using passwords that may be expired</td>
</tr>
<tr>
<td><code>--ignore-change-password-upon-first-use</code></td>
<td>Control if a user should be forced to change their password immediately after they log into keystone for the first time. Opt into ignoring the user to change their password during first time login in keystone</td>
</tr>
<tr>
<td><code>--no-ignore-change-password-upon-first-use</code></td>
<td>Control if a user should be forced to change their password immediately after they log into keystone for the first time. Opt out of ignoring the user to change their password during first time login in keystone</td>
</tr>
<tr>
<td><code>--enable-lock-password</code></td>
<td>Disables the ability for a user to change its password through self-service APIs</td>
</tr>
<tr>
<td><code>--disable-lock-password</code></td>
<td>Enables the ability for a user to change its password through self-service APIs</td>
</tr>
<tr>
<td><code>--enable-multi-factor-auth</code></td>
<td>Enables the mfa (multi factor auth)</td>
</tr>
<tr>
<td><code>--disable-multi-factor-auth</code></td>
<td>Disables the mfa (multi factor auth)</td>
</tr>
</tbody>
</table>
Set multi-factor auth rules, for example, to set a rule requiring the "password" and "totp" auth methods to be provided, use: 

```
--multi-factor-auth-rule password,totp
```

May be provided multiple times to set different rule combinations.

---

Enable user (default)

---

Disable user

---

Return existing user

### Table 87.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 87.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 87.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 87.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
87.2. USER DELETE

Delete user(s)

Usage:

openstack user delete [-h] [--domain <domain>] <user> [<user> ...]

Table 87.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;user&gt;</td>
<td>User(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 87.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;user&gt; (name or id)</td>
</tr>
</tbody>
</table>

87.3. USER LIST

List users

Usage:

openstack user list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN]
  [--quote {all, minimal, none, nonnumeric}]
  [--noindent] [--max-width <integer>] [--fit-width]  
  [--print-empty] [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--domain <domain>]
  [--group <group> | --project <project>] [--long]

Table 87.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Filter users by &lt;domain&gt; (name or id)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--group &lt;group&gt;</td>
<td>Filter users by &lt;group&gt; membership (name or id)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Filter users by &lt;project&gt; (name or id)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 87.10. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>--column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 87.11. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 87.12. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 87.13. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 87.4. USER PASSWORD SET

Change current user password

**Usage:**

```
openstack user password set [-h] [--password <new-password>] [--original-password <original-password>]
```

**Table 87.14. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--password &lt;new-password&gt;</td>
<td>New user password</td>
</tr>
<tr>
<td>--original-password &lt;original-password&gt;</td>
<td>Original user password</td>
</tr>
</tbody>
</table>

### 87.5. USER SET

Set user properties

**Usage:**

```
```
[--multi-factor-auth-rule <rule>]
[--enable | --disable]

Table 87.15. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;user&gt;</td>
<td>User to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 87.16. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set user name</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain the user belongs to (name or id). This can be used in case collisions between user names exist.</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Set default project (name or id)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--password &lt;password&gt;</td>
<td>Set user password</td>
</tr>
<tr>
<td>--password-prompt</td>
<td>Prompt interactively for password</td>
</tr>
<tr>
<td>--email &lt;email-address&gt;</td>
<td>Set user email address</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set user description</td>
</tr>
<tr>
<td>--ignore-lockout-failure-attempts</td>
<td>Opt into ignoring the number of times a user has authenticated and locking out the user as a result</td>
</tr>
<tr>
<td>--no-ignore-lockout-failure-attempts</td>
<td>Opt out of ignoring the number of times a user has authenticated and locking out the user as a result</td>
</tr>
<tr>
<td>--ignore-password-expiry</td>
<td>Opt into allowing user to continue using passwords that may be expired</td>
</tr>
<tr>
<td>--no-ignore-password-expiry</td>
<td>Opt out of allowing user to continue using passwords that may be expired</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--ignore-change-password-upon-first-use</td>
<td>Control if a user should be forced to change their password immediately after they log into keystone for the first time. Opt into ignoring the user to change their password during first time login in keystone</td>
</tr>
<tr>
<td>--no-ignore-change-password-upon-first-use</td>
<td>Control if a user should be forced to change their password immediately after they log into keystone for the first time. Opt out of ignoring the user to change their password during first time login in keystone</td>
</tr>
<tr>
<td>--enable-lock-password</td>
<td>Disables the ability for a user to change its password through self-service APIs</td>
</tr>
<tr>
<td>--disable-lock-password</td>
<td>Enables the ability for a user to change its password through self-service APIs</td>
</tr>
<tr>
<td>--enable-multi-factor-auth</td>
<td>Enables the mfa (multi factor auth)</td>
</tr>
<tr>
<td>--disable-multi-factor-auth</td>
<td>Disables the mfa (multi factor auth)</td>
</tr>
<tr>
<td>--multi-factor-auth-rule &lt;rule&gt;</td>
<td>Set multi-factor auth rules. For example, to set a rule requiring the &quot;password&quot; and &quot;totp&quot; auth methods to be provided, use: &quot;--multi-factor-auth-rule password,totp&quot;. May be provided multiple times to set different rule combinations.</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable user (default)</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable user</td>
</tr>
</tbody>
</table>

### 87.6. USER SHOW

Display user details

**Usage:**

```bash
```

**Table 87.17. Positional arguments**
### Table 87.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;user&gt;</td>
<td>User to display (name or id)</td>
</tr>
</tbody>
</table>

### Table 87.19. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--domain &lt;domain&gt;</td>
<td>Domain owning &lt;user&gt; (name or id)</td>
</tr>
</tbody>
</table>

### Table 87.20. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 87.21. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 87.22. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 88. VERSIONS

This chapter describes the commands under the `versions` command.

88.1. VERSIONS SHOW

Show available versions of services

Usage:

```
```

Table 88.1. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-interfaces</td>
<td>Show values for all interfaces</td>
</tr>
<tr>
<td>--interface &lt;interface&gt;</td>
<td>Show versions for a specific interface.</td>
</tr>
<tr>
<td>--region-name &lt;region_name&gt;</td>
<td>Show versions for a specific region.</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>Show versions for a specific service. the argument should be either an exact match to what is in the catalog or a known official value or alias from service-types-authority (<a href="https://service-types.openstack.org/">https://service-types.openstack.org/</a>)</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Show versions for a specific status. valid values are: SUPPORTED - CURRENT - DEPRECATED - EXPERIMENTAL</td>
</tr>
</tbody>
</table>

Table 88.2. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 88.3. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 88.4. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 88.5. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 89. VOLUME

This chapter describes the commands under the `volume` command.

89.1. VOLUME BACKUP CREATE

Create new volume backup

Usage:

```
```

Table 89.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to backup (name or id)</td>
</tr>
</tbody>
</table>

Table 89.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Name of the backup</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the backup</td>
</tr>
<tr>
<td>--container &lt;container&gt;</td>
<td>Optional backup container name</td>
</tr>
<tr>
<td>--snapshot &lt;snapshot&gt;</td>
<td>Snapshot to backup (name or id)</td>
</tr>
<tr>
<td>--force</td>
<td>Allow to back up an in-use volume</td>
</tr>
<tr>
<td>--incremental</td>
<td>Perform an incremental backup</td>
</tr>
</tbody>
</table>

Table 89.3. Output formatter options
89.2. VOLUME BACKUP DELETE

Delete volume backup(s)

Usage:

```
openstack volume backup delete [-h] [--force] <backup> [<backup> ...]
```

Table 89.7. Positional arguments
### Table 89.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup&gt;</td>
<td>Backup(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

### 89.3. VOLUME BACKUP LIST

List volume backups

**Usage:**

```
```

**Usage:**

```
```

### Table 89.9. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Filters results by the backup name</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
<td>Filters results by the backup status (creating, available, deleting, error, restoring or error_restoring)</td>
</tr>
<tr>
<td>--volume &lt;volume&gt;</td>
<td>Filters results by the volume which they backup (name or ID)</td>
</tr>
<tr>
<td>--marker &lt;volume-backup&gt;</td>
<td>The last backup of the previous page (name or id)</td>
</tr>
</tbody>
</table>
**Table 89.10. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--limit &lt;num-backups&gt;</td>
<td>Maximum number of backups to display</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Include all projects (admin only)</td>
</tr>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 89.11. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 89.12. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 89.13. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
89.4. VOLUME BACKUP RECORD EXPORT

Export volume backup details. Backup information can be imported into a new service instance to be able to restore.

Usage:

```
openstack volume backup record export [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
<backup>
```

Table 89.14. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup&gt;</td>
<td>Backup to export (name or id)</td>
</tr>
</tbody>
</table>

Table 89.15. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.16. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.17. JSON formatter options
Table 89.18. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.19. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.5. VOLUME BACKUP RECORD IMPORT

Import volume backup details. Exported backup details contain the metadata necessary to restore to a new or rebuilt service instance

Usage:

```
```

Table 89.20. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup_service&gt;</td>
<td>Backup service containing the backup.</td>
</tr>
<tr>
<td>&lt;backup_metadata&gt;</td>
<td>Encoded backup metadata from export.</td>
</tr>
</tbody>
</table>
Table 89.21. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.22. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.23. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.24. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.25. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.6. VOLUME BACKUP RESTORE

Restore volume backup

Usage:

Table 89.26. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup&gt;</td>
<td>Backup to restore (name or id)</td>
</tr>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to restore to (name or id)</td>
</tr>
</tbody>
</table>

Table 89.27. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.28. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.29. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.30. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.31. Table formatter options
### 89.7. VOLUME BACKUP SET

Set volume backup properties

#### Usage:

```
openstack volume backup set [-h] [--name <name>] [--description <description>] [--state <state>] <backup>
```

Table 89.32. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup&gt;</td>
<td>Backup to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.33. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New backup name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New backup description</td>
</tr>
<tr>
<td>--state &lt;state&gt;</td>
<td>New backup state (&quot;available&quot; or &quot;error&quot;) (admin only) (This option simply changes the state of the backup in the database with no regard to actual status, exercise caution when using)</td>
</tr>
</tbody>
</table>

### 89.8. VOLUME BACKUP SHOW

Display volume backup details
Usage:

```
openstack volume backup show [-h] [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent] [--prefix PREFIX]
  [--max-width <integer>] [--fit-width]
  [--print-empty]
  <backup>
```

Table 89.34. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;backup&gt;</td>
<td>Backup to display (name or id)</td>
</tr>
</tbody>
</table>

Table 89.35. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.36. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.37. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.38. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.39. Table formatter options
89.9. VOLUME CREATE

Create new volume

Usage:

```bash
```

Table 89.40. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Volume name</td>
</tr>
</tbody>
</table>

Table 89.41. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--size &lt;size&gt;</td>
<td>Volume size in gb (required unless --snapshot or --source is specified)</td>
</tr>
<tr>
<td>--type &lt;volume-type&gt;</td>
<td>Set the type of volume</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--image &lt;image&gt;</td>
<td>Use &lt;image&gt; as source of volume (name or id)</td>
</tr>
<tr>
<td>--snapshot &lt;snapshot&gt;</td>
<td>Use &lt;snapshot&gt; as source of volume (name or id)</td>
</tr>
<tr>
<td>--source &lt;volume&gt;</td>
<td>Volume to clone (name or id)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Volume description</td>
</tr>
<tr>
<td>--availability-zone &lt;availability-zone&gt;</td>
<td>Create volume in &lt;availability-zone&gt;</td>
</tr>
<tr>
<td>--consistency-group consistency-group&gt;</td>
<td>Consistency group where the new volume belongs to</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property to this volume (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--hint &lt;key=value&gt;</td>
<td>Arbitrary scheduler hint key-value pairs to help boot an instance</td>
</tr>
<tr>
<td></td>
<td>(repeat option to set multiple hints)</td>
</tr>
<tr>
<td>--bootable</td>
<td>Mark volume as bootable</td>
</tr>
<tr>
<td>--non-bootable</td>
<td>Mark volume as non-bootable (default)</td>
</tr>
<tr>
<td>--read-only</td>
<td>Set volume to read-only access mode</td>
</tr>
<tr>
<td>--read-write</td>
<td>Set volume to read-write access mode (default)</td>
</tr>
</tbody>
</table>

Table 89.42. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.43. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.44. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.45. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.10. VOLUME DELETE

Delete volume(s)

Usage:

```bash
openstack volume delete [-h] [--force | --purge] <volume> [<volume> ...]
```

Table 89.46. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 89.47. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Attempt forced removal of volume(s), regardless of state (defaults to False)</td>
</tr>
<tr>
<td>--purge</td>
<td>Remove any snapshots along with volume(s) (defaults to false)</td>
</tr>
</tbody>
</table>

89.11. VOLUME HOST SET
Set volume host properties

**Usage:**

```
openstack volume host set [-h] [--disable | --enable] <host-name>
```

<table>
<thead>
<tr>
<th>Table 89.48. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>&lt;host-name&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 89.49. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--disable</td>
</tr>
<tr>
<td>--enable</td>
</tr>
</tbody>
</table>

### 89.12. VOLUME LIST

List volumes

**Usage:**

```
openstack volume list [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                      [--quote {all, minimal, none, nonnumeric}]
                      [--noindent] [-m <integer>]
                      [-p] [-print-empty]
                      [--sort-column SORT_COLUMN]
                      [--sort-ascending | --sort-descending]
                      [-p] [-project <project>]
                      [-p] [-project-domain <project-domain>]
                      [-u <user>] [-user-domain <user-domain>]
                      [-n <name>] [-status <status>]
                      [-a] [-all-projects] [-l] [-limit <num-volumes>]
```

<table>
<thead>
<tr>
<th>Table 89.50. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
</tr>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
</tr>
<tr>
<td>--user &lt;user&gt;</td>
</tr>
<tr>
<td>--user-domain &lt;user-domain&gt;</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
</tr>
<tr>
<td>--status &lt;status&gt;</td>
</tr>
<tr>
<td>--all-projects</td>
</tr>
<tr>
<td>--long</td>
</tr>
<tr>
<td>--marker &lt;volume&gt;</td>
</tr>
<tr>
<td>--limit &lt;num-volumes&gt;</td>
</tr>
</tbody>
</table>

**Table 89.51. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 89.52. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
### Table 89.53. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 89.54. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 89.13. VOLUME MIGRATE

Migrate volume to a new host

**Usage:**

```
openstack volume migrate [-h] --host <host> [--force-host-copy] [--lock-volume] <volume>
```

### Table 89.55. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to migrate (name or id)</td>
</tr>
</tbody>
</table>

### Table 89.56. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>Destination host (takes the form: host@backend-name#pool)</td>
</tr>
<tr>
<td>--force-host-copy</td>
<td>Enable generic host-based force-migration, which bypasses driver optimizations</td>
</tr>
</tbody>
</table>
If specified, the volume state will be locked and will not allow a migration to be aborted (possibly by another operation)

89.14. VOLUME QOS ASSOCIATE

Associate a QoS specification to a volume type

Usage:

openstack volume qos associate [-h] <qos-spec> <volume-type>

Table 89.57. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-spec&gt;</td>
<td>Qos specification to modify (name or id)</td>
</tr>
<tr>
<td>&lt;volume-type&gt;</td>
<td>Volume type to associate the qos (name or id)</td>
</tr>
</tbody>
</table>

Table 89.58. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

89.15. VOLUME QOS CREATE

Create new QoS specification

Usage:


Table 89.59. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>New qos specification name</td>
</tr>
</tbody>
</table>
Table 89.60. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--consumer &lt;consumer&gt;</td>
<td>Consumer of the qos. valid consumers: back-end, both, front-end (defaults to both)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a qos specification property (repeat option to set multiple properties)</td>
</tr>
</tbody>
</table>

Table 89.61. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.62. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.63. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.64. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
89.16. VOLUME QOS DELETE

Delete QoS specification

Usage:

```
openstack volume qos delete [-h] [--force] <qos-spec> [<qos-spec> ...]
```

<table>
<thead>
<tr>
<th>Table 89.65. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td><code>&lt;qos-spec&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 89.66. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td><code>-h, --help</code></td>
</tr>
<tr>
<td><code>--force</code></td>
</tr>
</tbody>
</table>

89.17. VOLUME QOS DISASSOCIATE

Disassociate a QoS specification from a volume type

Usage:

```
openstack volume qos disassociate [-h] [--volume-type <volume-type> | --all] <qos-spec>
```

<table>
<thead>
<tr>
<th>Table 89.67. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td><code>&lt;qos-spec&gt;</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 89.68. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Value</strong></td>
</tr>
<tr>
<td><code>-h, --help</code></td>
</tr>
<tr>
<td><code>--volume-type &lt;volume-type&gt;</code></td>
</tr>
<tr>
<td><code>--all</code></td>
</tr>
</tbody>
</table>
89.18. VOLUME QOS LIST

List QoS specifications

Usage:

```bash
```

Table 89.69. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.70. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 89.71. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 89.72. JSON formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 89.73. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**89.19. VOLUME QOS SET**

Set QoS specification properties

**Usage:**

```bash
openstack volume qos set [-h] [--property <key=value>] <qos-spec>
```

**Table 89.74. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-spec&gt;</td>
<td>Qos specification to modify (name or id)</td>
</tr>
</tbody>
</table>

**Table 89.75. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to add or modify for this qos specification (repeat option to set multiple properties)</td>
</tr>
</tbody>
</table>

**89.20. VOLUME QOS SHOW**

Display QoS specification details

**Usage:**

Table 89.76. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-spec&gt;</td>
<td>Qos specification to display (name or id)</td>
</tr>
</tbody>
</table>

Table 89.77. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 89.78. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.79. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.80. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.81. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
89.21. VOLUME QOS UNSET

Unset QoS specification properties

Usage:

```bash
openstack volume qos unset [-h] [--property <key>] <qos-spec>
```

Table 89.82. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;qos-spec&gt;</td>
<td>Qos specification to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.83. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to remove from the qos specification. (repeat option to unset multiple properties)</td>
</tr>
</tbody>
</table>

89.22. VOLUME SERVICE LIST

List service command

Usage:

```bash
openstack volume service list [-h] [-f {csv, json, table, value, yaml}]
  [-c COLUMN]
  [-quote {all, minimal, none, nonnumeric}]
  [--noindent] [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
```
Table 89.84. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--host &lt;host&gt;</td>
<td>List services on specified host (name only)</td>
</tr>
<tr>
<td>--service &lt;service&gt;</td>
<td>List only specified service (name only)</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
</tbody>
</table>

Table 89.85. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv,json,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 89.86. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 89.87. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.88. Table formatter options
### 89.23. VOLUME SERVICE SET

Set volume service properties

**Usage:**

```bash
openstack volume service set [-h] [--enable | --disable] [--disable-reason <reason>] <host> <service>
```

**Table 89.89. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;host&gt;</td>
<td>Name of host</td>
</tr>
<tr>
<td>&lt;service&gt;</td>
<td>Name of service (binary name)</td>
</tr>
</tbody>
</table>

**Table 89.90. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--enable</td>
<td>Enable volume service</td>
</tr>
<tr>
<td>--disable</td>
<td>Disable volume service</td>
</tr>
<tr>
<td>--disable-reason &lt;reason&gt;</td>
<td>Reason for disabling the service (should be used with --disable option)</td>
</tr>
</tbody>
</table>

### 89.24. VOLUME SET

Set volume properties

**Usage:**
openstack volume set [-h] [--name <name>] [--size <size>]
[-description <description>] [--no-property]
[--property <key=value>]
[--image-property <key=value>] [--state <state>]
[--attached | --detached] [--type <volume-type>]
[--retype-policy <retype-policy>]
[--bootable | --non-bootable]
[--read-only | --read-write]
<br><volume>

Table 89.91. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.92. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New volume name</td>
</tr>
<tr>
<td>--size &lt;size&gt;</td>
<td>Extend volume size in gb</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New volume description</td>
</tr>
<tr>
<td>--no-property</td>
<td>Remove all properties from &lt;volume&gt; (specify both --no-property and --property to remove the current properties before setting new properties.)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on this volume (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--image-property &lt;key=value&gt;</td>
<td>Set an image property on this volume (repeat option to set multiple image properties)</td>
</tr>
<tr>
<td>--state &lt;state&gt;</td>
<td>New volume state (&quot;available&quot;, &quot;error&quot;, &quot;creating&quot;, &quot;deleting&quot;, &quot;in-use&quot;, &quot;attaching&quot;, &quot;detaching&quot;, &quot;error_deleting&quot; or &quot;maintenance&quot;) (admin only) (This option simply changes the state of the volume in the database with no regard to actual status, exercise caution when using)</td>
</tr>
<tr>
<td>--attached</td>
<td>Set volume attachment status to &quot;attached&quot; (admin only) (This option simply changes the state of the volume in the database with no regard to actual status, exercise caution when using)</td>
</tr>
</tbody>
</table>
---detached

Set volume attachment status to "detached" (admin only) (This option simply changes the state of the volume in the database with no regard to actual status, exercise caution when using)

--type <volume-type>

New volume type (name or id)

--retype-policy <retype-policy>

Migration policy while re-typing volume ("never" or "on-demand", default is "never") (available only when --type option is specified)

--bootable

Mark volume as bootable

--non-bootable

Mark volume as non-bootable

--read-only

Set volume to read-only access mode

--read-write

Set volume to read-write access mode

### 89.25. VOLUME SHOW

Display volume details

**Usage:**

```
```

**Table 89.93. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 89.94. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 89.95. Output formatter options**
89.26. VOLUME SNAPSHOT CREATE

Create new volume snapshot

**Usage:**

```bash
```
Table 89.99. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot-name&gt;</td>
<td>Name of the new snapshot</td>
</tr>
</tbody>
</table>

Table 89.100. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--volume &lt;volume&gt;</td>
<td>Volume to snapshot (name or id) (default is &lt;snapshot-name&gt;)</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Description of the snapshot</td>
</tr>
<tr>
<td>--force</td>
<td>Create a snapshot attached to an instance. default is False</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property to this snapshot (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--remote-source &lt;key=value&gt;</td>
<td>The attribute(s) of the existing remote volume snapshot (admin required) (repeat option to specify multiple attributes) e.g.: --remote-source source-name=test_name --remote-source source-id=test_id</td>
</tr>
</tbody>
</table>

Table 89.101. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.102. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
Table 89.103. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.104. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.27. VOLUME SNAPSHOT DELETE

Delete volume snapshot(s)

Usage:

    openstack volume snapshot delete [-h] [--force] <snapshot> [<snapshot> ...]

Table 89.105. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Snapshot(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 89.106. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Attempt forced removal of snapshot(s), regardless of state (defaults to False)</td>
</tr>
</tbody>
</table>

89.28. VOLUME SNAPSHOT LIST

List volume snapshots
Usage:

```
```

Table 89.107. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--all-projects</code></td>
<td>Include all projects (admin only)</td>
</tr>
<tr>
<td><code>--project &lt;project&gt;</code></td>
<td>Filter results by project (name or id) (admin only)</td>
</tr>
<tr>
<td><code>--project-domain &lt;project-domain&gt;</code></td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td><code>--long</code></td>
<td>List additional fields in output</td>
</tr>
<tr>
<td><code>--marker &lt;volume-snapshot&gt;</code></td>
<td>The last snapshot id of the previous page</td>
</tr>
<tr>
<td><code>--limit &lt;num-snapshots&gt;</code></td>
<td>Maximum number of snapshots to display</td>
</tr>
<tr>
<td><code>--name &lt;name&gt;</code></td>
<td>Filters results by a name.</td>
</tr>
<tr>
<td><code>--status &lt;status&gt;</code></td>
<td>Filters results by a status. (available, error, creating, deleting or error_deleting)</td>
</tr>
<tr>
<td><code>--volume &lt;volume&gt;</code></td>
<td>Filters results by a volume (name or id).</td>
</tr>
</tbody>
</table>

Table 89.108. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Value | Summary
---|---
--c COLUMN, --column COLUMN | Specify the column(s) to include, can be repeated to show multiple columns

--sort-column SORT_COLUMN | Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated

--sort-ascending | Sort the column(s) in ascending order

--sort-descending | Sort the column(s) in descending order

**Table 89.109. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 89.110. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 89.111. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

|--fit-width | Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable |

|--print-empty | Print empty table if there is no data to show. |

### 89.29. VOLUME SNAPSHOT SET

Set volume snapshot properties

**Usage:**

```
openstack volume snapshot set [-h] [--name <name>] [--description <description>]
```
Table 89.112. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Snapshot to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.113. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New snapshot name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>New snapshot description</td>
</tr>
<tr>
<td>--no-property</td>
<td>Remove all properties from &lt;snapshot&gt; (specify both --no-property and --property to remove the current properties before setting new properties.)</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Property to add/change for this snapshot (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--state &lt;state&gt;</td>
<td>New snapshot state. (&quot;available&quot;, &quot;error&quot;, &quot;creating&quot;, &quot;deleting&quot;, or &quot;error_deleting&quot;) (admin only) (This option simply changes the state of the snapshot in the database with no regard to actual status, exercise caution when using)</td>
</tr>
</tbody>
</table>

89.30. VOLUME SNAPSHOT SHOW

Display volume snapshot details

Usage:

```
```

Table 89.114. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Snapshot to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 89.115. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 89.116. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 89.117. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 89.118. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 89.119. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable, you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
89.31. VOLUME SNAPSHOT UNSET

Unset volume snapshot properties

Usage:

```
openstack volume snapshot unset [-h] [--property <key>] <snapshot>
```

Table 89.120. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;snapshot&gt;</td>
<td>Snapshot to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.121. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Property to remove from snapshot (repeat option to remove multiple properties)</td>
</tr>
</tbody>
</table>

89.32. VOLUME TRANSFER REQUEST ACCEPT

Accept volume transfer request.

Usage:

```
openstack volume transfer request accept [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>] [--fit-width] [--print-empty] --auth-key <key> <transfer-request-id>
```

Table 89.122. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;transfer-request-id&gt;</td>
<td>Volume transfer request to accept (id only)</td>
</tr>
</tbody>
</table>

Table 89.123. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
</table>
Table 89.124. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.125. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.126. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.127. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.33. VOLUME TRANSFER REQUEST CREATE

Create volume transfer request.
Usage:

```
openstack volume transfer request create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [-noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--name <name>]
    <volume>
```

Table 89.128. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to transfer (name or id)</td>
</tr>
</tbody>
</table>

Table 89.129. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>New transfer request name (default to none)</td>
</tr>
</tbody>
</table>

Table 89.130. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 89.131. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.132. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
89.34. VOLUME TRANSFER REQUEST DELETE

Delete volume transfer request(s).

Usage:

openstack volume transfer request delete [-h] <transfer-request> [...]

Table 89.134. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;transfer-request&gt;</td>
<td>Volume transfer request(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 89.135. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

89.35. VOLUME TRANSFER REQUEST LIST

Lists all volume transfer requests.

Usage:

openstack volume transfer request list [-h] [-f {csv, json, table, value, yaml}] [-c COLUMN] [--quote {all, minimal, none, nonnumeric}] [-noindent] [-max-width <integer>] [-fit-width] [-print-empty]
Table 89.136. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Include all projects (admin only)</td>
</tr>
</tbody>
</table>

Table 89.137. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 89.138. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 89.139. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.140. Table formatter options
### 89.36. VOLUME TRANSFER REQUEST SHOW

Show volume transfer request details.

**Usage:**

```
openstack volume transfer request show [-h]
  [-f {json,shell,table,value,yaml}]
  [-c COLUMN] [--noindent]
  [--prefix PREFIX]
  [--max-width <integer>]
  [--fit-width] [--print-empty]
  <transfer-request>
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

**Table 89.141. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;transfer-request&gt;</td>
<td>Volume transfer request to display (name or id)</td>
</tr>
</tbody>
</table>

**Table 89.142. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 89.143. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 89.144. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.145. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 89.146. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.37. VOLUME TYPE CREATE

Create new volume type

Usage:


Red Hat OpenStack Platform 17.0 Command Line Interface Reference

1348
Table 89.147. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;name&gt;</td>
<td>Volume type name</td>
</tr>
</tbody>
</table>

Table 89.148. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Volume type description</td>
</tr>
<tr>
<td>--public</td>
<td>Volume type is accessible to the public</td>
</tr>
<tr>
<td>--private</td>
<td>Volume type is not accessible to the public</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on this volume type (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Allow &lt;project&gt; to access private type (name or id) (Must be used with --private option)</td>
</tr>
<tr>
<td>--encryption-provider &lt;provider&gt;</td>
<td>Set the encryption provider format for this volume type (e.g &quot;luks&quot; or &quot;plain&quot;) (admin only) (This option is required when setting encryption type of a volume. Consider using other encryption options such as: &quot;--encryption-cipher&quot;, &quot;--encryption-key-size&quot; and &quot;--encryption-control-location&quot;)</td>
</tr>
<tr>
<td>--encryption-cipher &lt;cipher&gt;</td>
<td>Set the encryption algorithm or mode for this volume type (e.g &quot;aes-xts-plain64&quot;) (admin only)</td>
</tr>
<tr>
<td>--encryption-key-size &lt;key-size&gt;</td>
<td>Set the size of the encryption key of this volume type (e.g &quot;128&quot; or &quot;256&quot;) (admin only)</td>
</tr>
<tr>
<td>--encryption-control-location &lt;control-location&gt;</td>
<td>Set the notional service where the encryption is performed (&quot;front-end&quot; or &quot;back-end&quot;) (admin only) (The default value for this option is &quot;front-end&quot; when setting encryption type of a volume. Consider using other encryption options such as: &quot;--encryption-cipher&quot;, &quot;--encryption-key-size&quot; and &quot;--encryption-provider&quot;)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
</tbody>
</table>

### Table 89.149. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f [json,shell,table,value,yaml], --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 89.150. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 89.151. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 89.152. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 89.38. VOLUME TYPE DELETE

Delete volume type(s)
Usage:

openstack volume type delete [-h] <volume-type> [<volume-type> ...]

Table 89.153. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume-type&gt;</td>
<td>Volume type(s) to delete (name or id)</td>
</tr>
</tbody>
</table>

Table 89.154. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

89.39. VOLUME TYPE LIST

List volume types

Usage:


Table 89.155. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--long</td>
<td>List additional fields in output</td>
</tr>
<tr>
<td>--default</td>
<td>List the default volume type</td>
</tr>
<tr>
<td>--public</td>
<td>List only public types</td>
</tr>
<tr>
<td>--private</td>
<td>List only private types (admin only)</td>
</tr>
<tr>
<td>--encryption-type</td>
<td>Display encryption information for each volume type (admin only)</td>
</tr>
</tbody>
</table>
Table 89.156. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 89.157. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 89.158. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 89.159. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

89.40. VOLUME TYPE SET

Set volume type properties
Usage:

```
openstack volume type set [-h] [--name <name>]
    [--description <description>]
    [--property <key=value>]
    [--project <project>]
    [--project-domain <project-domain>]
    [--encryption-provider <provider>]
    [--encryption-cipher <cipher>]
    [--encryption-key-size <key-size>]
    [--encryption-control-location <control-location>]
    <volume-type>
```

Table 89.160. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume-type&gt;</td>
<td>Volume type to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.161. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name &lt;name&gt;</td>
<td>Set volume type name</td>
</tr>
<tr>
<td>--description &lt;description&gt;</td>
<td>Set volume type description</td>
</tr>
<tr>
<td>--property &lt;key=value&gt;</td>
<td>Set a property on this volume type (repeat option to set multiple properties)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Set volume type access to project (name or id) (admin only)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). This can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--encryption-provider &lt;provider&gt;</td>
<td>Set the encryption provider format for this volume type (e.g. &quot;luks&quot; or &quot;plain&quot;) (admin only) (This option is required when setting encryption type of a volume for the first time. Consider using other encryption options such as: &quot;--encryption-cipher&quot;, &quot;--encryption-key-size&quot; and &quot;--encryption-control-location&quot;)</td>
</tr>
<tr>
<td>--encryption-cipher &lt;cipher&gt;</td>
<td>Set the encryption algorithm or mode for this volume type (e.g. &quot;aes-xts-plain64&quot;) (admin only)</td>
</tr>
</tbody>
</table>
89.41. VOLUME TYPE SHOW

Display volume type details

Usage:


Table 89.162. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume-type&gt;</td>
<td>Volume type to display (name or id)</td>
</tr>
</tbody>
</table>

Table 89.163. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--encryption-type</td>
<td>Display encryption information of this volume type (admin only)</td>
</tr>
</tbody>
</table>

Table 89.164. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
### Table 89.165. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 89.166. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 89.167. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 89.42. VOLUME TYPE UNSET

Unset volume type properties

**Usage:**

```
openstack volume type unset [-h] [--property <key>] [--project <project>] [--project-domain <project-domain>] [--encryption-type] <volume-type>
```

### Table 89.168. Positional arguments
Table 89.169. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume-type&gt;</td>
<td>Volume type to modify (name or id)</td>
</tr>
</tbody>
</table>

**89.43. VOLUME UNSET**

Unset volume properties

**Usage:**

```
openstack volume unset [-h] [--property <key>] [--image-property <key>] <volume>
```

Table 89.170. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;volume&gt;</td>
<td>Volume to modify (name or id)</td>
</tr>
</tbody>
</table>

Table 89.171. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--property &lt;key&gt;</td>
<td>Remove a property from volume (repeat option to remove multiple properties)</td>
</tr>
<tr>
<td>--project &lt;project&gt;</td>
<td>Removes volume type access to project (name or id) (admin only)</td>
</tr>
<tr>
<td>--project-domain &lt;project-domain&gt;</td>
<td>Domain the project belongs to (name or id). this can be used in case collisions between project names exist.</td>
</tr>
<tr>
<td>--encryption-type</td>
<td>Remove the encryption type for this volume type (admin only)</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--image-property &lt;key&gt;</td>
<td>Remove an image property from volume (repeat option to remove multiple image properties)</td>
</tr>
</tbody>
</table>
CHAPTER 90. WORKBOOK

This chapter describes the commands under the `workbook` command.

90.1. WORKBOOK CREATE

Create new workbook.

Usage:

```
```

Table 90.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workbook definition file</td>
</tr>
</tbody>
</table>

Table 90.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag workbook will be marked as &quot;public&quot;.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the workbook within.</td>
</tr>
</tbody>
</table>

Table 90.3. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 90.4. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
### Table 90.5. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 90.6. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, (&lt;1) to disable. You can also use the <code>CLIFF_MAX_TERM_WIDTH</code> environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if (--max-width) greater than 0. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 90.2. WORKBOOK DEFINITION SHOW

Show workbook definition.

**Usage:**

```
openstack workbook definition show [-h] name
```

### Table 90.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Workbook name</td>
</tr>
</tbody>
</table>

### Table 90.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

### 90.3. WORKBOOK DELETE

Delete workbook.

**Usage:**

```
openstack workbook delete [-h] [--namespace [NAMESPACE]] [workbook [workbook ...]]
```
Table 90.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>workbook</td>
<td>Name of workbook(s).</td>
</tr>
</tbody>
</table>

Table 90.10. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to delete the workbook(s) from.</td>
</tr>
</tbody>
</table>

90.4. WORKBOOK LIST

List all workbooks.

Usage:

```
```

Table 90.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
</tbody>
</table>
### --sort_dirs [SORT_DIRS]
Comma-separated list of sort directions. Default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc

### --filter FILTERS
Filters. Can be repeated.

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--filter FILTERS</td>
<td>Filters. Can be repeated.</td>
</tr>
</tbody>
</table>

### Table 90.12. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 90.13. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 90.14. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 90.15. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
### 90.5. WORKBOOK SHOW

Show specific workbook.

**Usage:**

```
workbook
```

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width, implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>workbook</td>
<td>Workbook name</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to get the workbook from.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
Table 90.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 90.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

90.6. WORKBOOK UPDATE

Update workbook.

Usage:

```
```

Table 90.22. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workbook definition file</td>
</tr>
</tbody>
</table>

Table 90.23. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to update the workbook in.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag workbook will be marked as &quot;public&quot;.</td>
</tr>
</tbody>
</table>

Table 90.24. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 90.25. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 90.26. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 90.27. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

90.7. WORKBOOK VALIDATE
Validate workbook.

Usage:

```
definition
```

Table 90.28. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workbook definition file</td>
</tr>
</tbody>
</table>

Table 90.29. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 90.30. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 90.31. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 90.32. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 91. WORKFLOW

This chapter describes the commands under the `workflow` command.

91.1. WORKFLOW CREATE

Create new workflow.

**Usage:**

```
```

**Table 91.1. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workflow definition file.</td>
</tr>
</tbody>
</table>

**Table 91.2. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to create the workflow within.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag workflow will be marked as &quot;public&quot;.</td>
</tr>
</tbody>
</table>

**Table 91.3. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

**Table 91.4. CSV formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

**Table 91.5. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 91.6. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
91.2. WORKFLOW DEFINITION SHOW

Show workflow definition.

Usage:

```
openstack workflow definition show [-h] [--namespace [NAMESPACE]] identifier
```

Table 91.7. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>identifier</td>
<td>Workflow id or name.</td>
</tr>
</tbody>
</table>

Table 91.8. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to get the workflow from.</td>
</tr>
</tbody>
</table>

91.3. WORKFLOW DELETE

Delete workflow.

Usage:

```
openstack workflow delete [-h] [--namespace [NAMESPACE]] workflow [workflow ...]
```

Table 91.9. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow</td>
<td>Name or id of workflow(s).</td>
</tr>
</tbody>
</table>

Table 91.10. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to delete the workflow from.</td>
</tr>
</tbody>
</table>

## 91.4. WORKFLOW ENGINE SERVICE LIST

List all services.

**Usage:**

```bash
openstack workflow engine service list [-h]

[-f {csv,json,table,value,yaml}]
[-c COLUMN]
[--quote {all,minimal,none,nonnumeric}]
[--noindent]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--sort-column SORT_COLUMN]
[--sort-ascending | --sort-descending]
[--marker [MARKER]]
[--limit [LIMIT]]
[--sort_keys [SORT_KEYS]]
[--sort_dirs [SORT_DIRS]]
[--filter FILTERS]
```

### Table 91.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
</tbody>
</table>

### Table 91.12. Output formatter options
### Table 91.13. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 91.14. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 91.15. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 91.5. WORKFLOW ENV CREATE

Create new environment.

Usage:
openstack workflow env create [-h] [-f {json,shell,table,value,yaml}] 
[-c COLUMN] [--noindent] 
[--prefix PREFIX] [--max-width <integer>] 
[--fit-width] [--print-empty] 
file

Table 91.16. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Environment configuration file in json or yaml</td>
</tr>
</tbody>
</table>

Table 91.17. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 91.18. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 91.19. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 91.20. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 91.21. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
91.6. WORKFLOW ENV DELETE

Delete environment.

**Usage:**

```
openstack workflow env delete [-h] environment [environment ...]
```

**Table 91.22. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>environment</td>
<td>Name of environment(s).</td>
</tr>
</tbody>
</table>

**Table 91.23. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

91.7. WORKFLOW ENV LIST

List all environments.

**Usage:**

```
```
### Table 91.24. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT.Keys]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
</tbody>
</table>

### Table 91.25. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 91.26. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 91.27. JSON formatter options
Table 91.28. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

91.8. WORKFLOW ENV SHOW

Show specific environment.

Usage:

```bash
```

Table 91.29. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>environment</td>
<td>Environment name</td>
</tr>
</tbody>
</table>

Table 91.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--export</td>
<td>Export the environment suitable for import</td>
</tr>
</tbody>
</table>

Table 91.31. Output formatter options
### Table 91.32. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 91.33. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 91.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 91.9. WORKFLOW ENV UPDATE

Update environment.

**Usage:**

```
```

### Table 91.35. Positional arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Environment configuration file in json or yaml</td>
</tr>
</tbody>
</table>

**Table 91.36. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

**Table 91.37. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

**Table 91.38. JSON formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 91.39. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 91.40. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
91.10. WORKFLOW EXECUTION CREATE

Create new execution.

Usage:

```bash
openstack workflow execution create [-h]
[-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent]
[--prefix PREFIX]
[--max-width <integer>]
[--fit-width] [--print-empty]
[--namespace [NAMESPACE]]
[-d DESCRIPTION]
[-s [SOURCE_EXECUTION_ID]]
[workflow_identifier] workflow_input [params]
```

<table>
<thead>
<tr>
<th>Table 91.41. Positional arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>workflow_identifier</td>
</tr>
<tr>
<td>workflow_input</td>
</tr>
<tr>
<td>params</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 91.42. Command arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
<tr>
<td>-h, --help</td>
</tr>
<tr>
<td>-d DESCRIPTION, --description DESCRIPTION</td>
</tr>
<tr>
<td>-s [SOURCE_EXECUTION_ID]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 91.43. Output formatter options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
</tr>
</tbody>
</table>

The output format, defaults to table

Specify the column(s) to include, can be repeated to show multiple columns

Table 91.44. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 91.45. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 91.46. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

91.11. WORKFLOW EXECUTION DELETE

Delete execution.

Usage:

```
openstack workflow execution delete [-h] [--force] execution [execution ...]
```

Table 91.47. Positional arguments
### Table 91.48. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>execution</td>
<td>Id of execution identifier(s).</td>
</tr>
</tbody>
</table>

### 91.12. WORKFLOW EXECUTION INPUT SHOW

Show execution input data.

**Usage:**

```
openstack workflow execution input show [-h] id
```

**Table 91.49. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Execution id</td>
</tr>
</tbody>
</table>

**Table 91.50. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--force</td>
<td>Force the deletion of an execution. might cause a cascade of errors if used for running executions.</td>
</tr>
</tbody>
</table>

### 91.13. WORKFLOW EXECUTION LIST

List all executions.

**Usage:**

```
```
Table 91.51. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--oldest</td>
<td>Display the executions starting from the oldest entries instead of the newest</td>
</tr>
<tr>
<td>--task [TASK]</td>
<td>Parent task execution id associated with workflow execution list.</td>
</tr>
<tr>
<td>--rootsonly</td>
<td>Return only root executions</td>
</tr>
</tbody>
</table>

Table 91.52. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
</tbody>
</table>
Value | Summary
--- | ---
--sort-ascending | Sort the column(s) in ascending order
--sort-descending | Sort the column(s) in descending order

Table 91.53. CSV formatter options

Value | Summary
--- | ---
--quote {all,minimal,none,nonnumeric} | When to include quotes, defaults to nonnumeric

Table 91.54. JSON formatter options

Value | Summary
--- | ---
--noindent | Whether to disable indenting the json

Table 91.55. Table formatter options

Value | Summary
--- | ---
--max-width <integer> | Maximum display width, <1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.
--fit-width | Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable
--print-empty | Print empty table if there is no data to show.

91.14. WORKFLOW EXECUTION OUTPUT SHOW

Show execution output data.

Usage:

```
openstack workflow execution output show [-h] id
```

Table 91.56. Positional arguments

Value | Summary
--- | ---
id | Execution id
### 91.15. WORKFLOW EXECUTION PUBLISHED SHOW

Show workflow global published variables.

**Usage:**

```
openstack workflow execution published show [-h] id
```

### Table 91.58. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Workflow id</td>
</tr>
</tbody>
</table>

### 91.16. WORKFLOW EXECUTION REPORT SHOW

Print execution report.

**Usage:**

```
openstack workflow execution report show [-h] [--errors-only]
    [--statistics-only]
    [--no-errors-only]
    [--max-depth [MAX_DEPTH]]
    id
```

### Table 91.60. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Execution id</td>
</tr>
</tbody>
</table>

### Table 91.61. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
## 91.17. WORKFLOW EXECUTION SHOW

Show specific execution.

**Usage:**

```
openstack workflow execution show [-h] [-f {json,shell,table,value,yaml}]
[-c COLUMN] [--noindent] [--prefix PREFIX] [--max-width <integer>]
[--fit-width] [--print-empty] execution
```

Table 91.62. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>execution</td>
<td>Execution identifier</td>
</tr>
</tbody>
</table>

Table 91.63. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>

Table 91.64. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>
Table 91.65. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 91.66. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 91.67. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

91.18. WORKFLOW EXECUTION UPDATE

Update execution.

Usage:

```
```
### Table 91.68. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Execution identifier</td>
</tr>
</tbody>
</table>

### Table 91.69. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>-s {RUNNING,PAUSED,SUCCESS,ERROR,CANCELLED}</td>
<td>Execution state</td>
</tr>
<tr>
<td>-e ENV, --env ENV</td>
<td>Environment variables</td>
</tr>
<tr>
<td>-d DESCRIPTION, --description DESCRIPTION</td>
<td>Execution description</td>
</tr>
</tbody>
</table>

### Table 91.70. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 91.71. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 91.72. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 91.73. Table formatter options
### Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. Implied if <code>--max-width</code> greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable.</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 91.19. WORKFLOW LIST

List all workflows.

**Usage:**

```bash
```

Table 91.74. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--marker [MARKER]</code></td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td><code>--limit [LIMIT]</code></td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td><code>--sort_keys [SORT_KEYS]</code></td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
</tbody>
</table>
### Table 91.75. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

### Table 91.76. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

### Table 91.77. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 91.78. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>

---

Red Hat OpenStack Platform 17.0 Command Line Interface Reference

1388
91.20. WORKFLOW SHOW

Show specific workflow.

Usage:

openstack workflow show [-h] [-f {json,shell,table,value,yaml}] 
[cOLUMN] [-noindent] [-prefix PREFIX] 
[--max-width <integer>] [-fit-width] 
[--print-empty] [-namespace [NAMESPACE]]

Table 91.79. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>workflow</td>
<td>Workflow id or name.</td>
</tr>
</tbody>
</table>

Table 91.80. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace to get the workflow from.</td>
</tr>
</tbody>
</table>

Table 91.81. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format json,shell,table,value,yaml</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 91.82. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>
**Value** | **Summary**
---|---
--prefix PREFIX | Add a prefix to all variable names

Table 91.84. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 91.21. WORKFLOW UPDATE

Update workflow.

**Usage:**

```
```

Table 91.85. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workflow definition</td>
</tr>
</tbody>
</table>

Table 91.86. Command arguments
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--marker [MARKER]</td>
<td>The last execution uuid of the previous page, displays list of executions after &quot;marker&quot;.</td>
</tr>
<tr>
<td>--limit [LIMIT]</td>
<td>Maximum number of entries to return in a single result.</td>
</tr>
<tr>
<td>--sort_keys [SORT_KEYS]</td>
<td>Comma-separated list of sort keys to sort results by. Default: created_at. Example: mistral execution-list --sort_keys=id,description</td>
</tr>
<tr>
<td>--sort_dirs [SORT_DIRS]</td>
<td>Comma-separated list of sort directions. default: asc. Example: mistral execution-list --sort_keys=id,description --sort_dirs=asc,desc</td>
</tr>
<tr>
<td>--filter FILTERS</td>
<td>Filters. can be repeated.</td>
</tr>
<tr>
<td>--id ID</td>
<td>Workflow id.</td>
</tr>
<tr>
<td>--namespace [NAMESPACE]</td>
<td>Namespace of the workflow.</td>
</tr>
<tr>
<td>--public</td>
<td>With this flag workflow will be marked as &quot;public&quot;.</td>
</tr>
</tbody>
</table>

Table 91.87. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 91.88. CSV formatter options
Table 91.89. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 91.90. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

91.22. WORKFLOW VALIDATE

Validate workflow.

Usage:

```
```

Table 91.91. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>definition</td>
<td>Workflow definition file</td>
</tr>
</tbody>
</table>

Table 91.92. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### Table 91.93. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 91.94. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--noindent</code></td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 91.95. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--prefix PREFIX</code></td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 91.96. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--max-width &lt;integer&gt;</code></td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td><code>--fit-width</code></td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td><code>--print-empty</code></td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
CHAPTER 92. ZONE

This chapter describes the commands under the `zone` command.

92.1. ZONE ABANDON

Abandon a zone

Usage:

```
openstack zone abandon [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] id
```

Table 92.1. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

Table 92.2. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

92.2. ZONE AXFR

AXFR a zone

Usage:

```
openstack zone axfr [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] id
```

Table 92.3. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

Table 92.4. Command arguments
### 92.3. ZONE BLACKLIST CREATE

Create new blacklist

**Usage:**

```bash
```

**Table 92.5. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pattern PATTERN</td>
<td>Blacklist pattern</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

**Table 92.6. Output formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
Table 92.7. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.8. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.9. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.4. ZONE BLACKLIST DELETE

Delete blacklist

Usage:

```
openstack zone blacklist delete [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] id
```

Table 92.10. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Blacklist id</td>
</tr>
</tbody>
</table>

Table 92.11. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
</tbody>
</table>
### 92.5. ZONE BLACKLIST LIST

List blacklists

**Usage:**

```
openstack zone blacklist list [-h] [-f {csv,json,table,value,yaml}] 
                [-c COLUMN] 
                [-quote {all,minimal,none,nonnumeric}] 
                [-noindent] [-max-width <integer>] 
                [-fit-width] [-print-empty] 
                [-sort-column SORT_COLUMN] 
                [-sort-ascending | --sort-descending] 
                [-all-projects] 
                [-sudo-project-id SUDO_PROJECT_ID]
```

#### Table 92.12. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-h, --help</code></td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td><code>--all-projects</code></td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td><code>--sudo-project-id SUDO_PROJECT_ID</code></td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

#### Table 92.13. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}</code></td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td><code>-c COLUMN, --column COLUMN</code></td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td><code>--sort-column SORT_COLUMN</code></td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 92.14. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 92.15. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.16. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.6. ZONE BLACKLIST SET

Set blacklist properties

Usage:

```
```

Table 92.17. Positional arguments
### Table 92.18. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--pattern PATTERN</td>
<td>Blacklist pattern</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--no-description—all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

### Table 92.19. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 92.20. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 92.21. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 92.22. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the <code>CLIFF_MAX_TERM_WIDTH</code> environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable <code>CLIFF_FIT_WIDTH=1</code> to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 92.7. ZONE BLACKLIST SHOW

Show blacklist details

**Usage:**

```
```

**Table 92.23. Positional arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Blacklist id</td>
</tr>
</tbody>
</table>

**Table 92.24. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

**Table 92.25. Output formatter options**
Table 92.26. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.27. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.28. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.8. ZONE CREATE

Create new zone

Usage:

```
```
Table 92.29. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Zone name</td>
</tr>
</tbody>
</table>

Table 92.30. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--email EMAIL</td>
<td>Zone email</td>
</tr>
<tr>
<td>--type {PRIMARY,SECONDARY}</td>
<td>Zone type</td>
</tr>
<tr>
<td>--ttl TTL</td>
<td>Time to live (seconds)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--masters MASTERS [MASTERS …​]</td>
<td>Zone masters</td>
</tr>
<tr>
<td>--attributes ATTRIBUTES [ATTRIBUTES ...]</td>
<td>Zone attributes</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.31. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.32. JSON formatter options
Whether to disable indenting the json

Table 92.33. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td></td>
</tr>
</tbody>
</table>

Add a prefix to all variable names

Table 92.34. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.9. ZONE DELETE

Delete zone

Usage:

```bash
```

Table 92.35. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

Table 92.36. Command arguments
Table 92.37. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.38. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.39. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.40. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.10. ZONE EXPORT CREATE
Export a Zone

Usage:


Table 92.41. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

Table 92.42. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.43. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.44. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.45. Shell formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.46. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.11. ZONE EXPORT DELETE

Delete a Zone Export

Usage:

```
openstack zone export delete [-h] [--all-projects]
    [--sudo-project-id SUDO_PROJECT_ID]
    zone_export_id
```

Table 92.47. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_export_id</td>
<td>Zone export id</td>
</tr>
</tbody>
</table>

Table 92.48. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

92.12. ZONE EXPORT LIST
List Zone Exports

Usage:

```
openstack zone export list [-h] [-f {csv,json,table,value,yaml}]
  [-c COLUMN]
  [--quote {all,minimal,none,nonnumeric}]
  [--noindent] [--max-width <integer>]
  [--fit-width] [--print-empty]
  [--sort-column SORT_COLUMN]
  [--sort-ascending | --sort-descending]
  [--all-projects]
  [--sudo-project-id SUDO_PROJECT_ID]
```

Table 92.49. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.50. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{csv,json,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 92.51. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>
Table 92.52. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.53. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.13. ZONE EXPORT SHOW

Show a Zone Export

Usage:

```
```

Table 92.54. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_export_id</td>
<td>Zone export id</td>
</tr>
</tbody>
</table>

Table 92.55. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
</tbody>
</table>
### Table 92.56. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

### Table 92.57. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

### Table 92.58. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

### Table 92.59. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

## 92.14. ZONE EXPORT SHOWFILE

Show the zone file for the Zone Export

**Usage:**

```

Table 92.60. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_export_id</td>
<td>Zone export id</td>
</tr>
</tbody>
</table>

Table 92.61. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.62. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.63. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.64. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 92.65. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.15. ZONE IMPORT CREATE

Import a Zone from a file on the filesystem

Usage:

```
```

Table 92.66. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_file_path</td>
<td>Path to a zone file</td>
</tr>
</tbody>
</table>

Table 92.67. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.68. Output formatter options
Table 92.69. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-c COLUMN, --column COLUMN</th>
<th>Specify the column(s) to include, can be repeated to show multiple columns</th>
</tr>
</thead>
</table>

Table 92.70. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.71. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.16. ZONE IMPORT DELETE

Delete a Zone Import

Usage:

```
openstack zone import delete [-h] [--all-projects] [--sudo-project-id SUDO_PROJECT_ID] zone_import_id
```

Table 92.72. Positional arguments
Table 92.73. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_import_id</td>
<td>Zone import id</td>
</tr>
</tbody>
</table>

Table 92.73. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

92.17. ZONE IMPORT LIST

List Zone Imports

Usage:

```
```

Table 92.74. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.75. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>--c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 92.76. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 92.77. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.78. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.18. ZONE IMPORT SHOW

Show a Zone Import

Usage:

```
openstack zone import show [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] [--noindent] [--prefix PREFIX]
```
Table 92.79. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_import_id</td>
<td>Zone import id</td>
</tr>
</tbody>
</table>

Table 92.80. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.81. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>{json,shell,table,value,yaml}</td>
<td></td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.82. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.83. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.84. Table formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 92.19. ZONE LIST

List zones

**Usage:**

```
```

**Table 92.85. Command arguments**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--name NAME</td>
<td>Zone name</td>
</tr>
<tr>
<td>--email EMAIL</td>
<td>Zone email</td>
</tr>
<tr>
<td>--type {PRIMARY, SECONDARY}</td>
<td>Zone type</td>
</tr>
<tr>
<td>--ttl TTL</td>
<td>Time to live (seconds)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--status STATUS</td>
<td>Zone status</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
</tbody>
</table>
Table 92.86. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 92.87. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all, minimal, none, nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 92.88. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.89. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
## 92.20. ZONE SET

Set zone properties

**Usage:**

```bash
openstack zone set [-h] [-f {json,shell,table,value,yaml}] [-c COLUMN] 
    [--noindent] [--prefix PREFIX] 
    [--max-width <integer>] [--fit-width] 
    [--all-projects] 
    [sudo-project-id SUDO_PROJECT_ID]
```

### Table 92.90. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

### Table 92.91. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--email EMAIL</td>
<td>Zone email</td>
</tr>
<tr>
<td>--ttl TTL</td>
<td>Time to live (seconds)</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--no-description --masters MASTERS [MASTERS ...]</td>
<td>Zone masters</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

### Table 92.92. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>
Table 92.93. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.94. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.95. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.21. ZONE SHOW

Show zone details

Usage:

```
```
Table 92.96. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone id</td>
</tr>
</tbody>
</table>

Table 92.97. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.98. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.99. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.100. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.101. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
</tbody>
</table>
Table 92.102. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.103. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv, json, table, value, yaml}, --format {csv, json, table, value, yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>
## 92.23. ZONE TRANSFER ACCEPT REQUEST

Accept a Zone Transfer Request

### Usage:

```
openstack zone transfer accept request [-h] ...
```

Red Hat OpenStack Platform 17.0 Command Line Interface Reference
Table 92.107. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--transfer-id TRANSFER_ID</td>
<td>Transfer id</td>
</tr>
<tr>
<td>--key KEY</td>
<td>Transfer key</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.108. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.109. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.110. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.111. Table formatter options
92.24. ZONE TRANSFER ACCEPT SHOW

Show Zone Transfer Accept

Usage:

openstack zone transfer accept show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width]
    [--print-empty]
    [--all-projects]
    [--sudo-project-id SUDO_PROJECT_ID]

id

Table 92.112. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone transfer accept id</td>
</tr>
</tbody>
</table>

Table 92.113. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.114. Output formatter options
<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.115. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.116. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.117. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 92.25. ZONE TRANSFER REQUEST CREATE

Create new zone transfer request

**Usage:**

```
openstack zone transfer request create [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--target-project-id TARGET_PROJECT_ID]
    [--description DESCRIPTION]
```
Table 92.118. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>zone_id</td>
<td>Zone id to transfer.</td>
</tr>
</tbody>
</table>

Table 92.119. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--target-project-id TARGET_PROJECT_ID</td>
<td>Target project id to transfer to.</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.120. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.121. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.122. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>
Table 92.123. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

92.26. ZONE TRANSFER REQUEST DELETE

Delete a Zone Transfer Request

Usage:

```
openstack zone transfer request delete [-h] [--all-projects] [-a] id
```

Table 92.124. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone transfer request id</td>
</tr>
</tbody>
</table>

Table 92.125. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

92.27. ZONE TRANSFER REQUEST LIST

List Zone Transfer Requests

Usage:

```
openstack zone transfer request list [-h] [-f {csv,json,table,value,yaml}]
```
Table 92.126. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.127. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {csv,json,table,value,yaml}, --format [csv,json,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>--sort-column SORT_COLUMN</td>
<td>Specify the column(s) to sort the data (columns specified first have a priority, non-existing columns are ignored), can be repeated</td>
</tr>
<tr>
<td>--sort-ascending</td>
<td>Sort the column(s) in ascending order</td>
</tr>
<tr>
<td>--sort-descending</td>
<td>Sort the column(s) in descending order</td>
</tr>
</tbody>
</table>

Table 92.128. CSV formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--quote {all,minimal,none,nonnumeric}</td>
<td>When to include quotes, defaults to nonnumeric</td>
</tr>
</tbody>
</table>

Table 92.129. JSON formatter options
### Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. You can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. Implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>

### 92.28. ZONE TRANSFER REQUEST SET

Set a Zone Transfer Request

Usage:

```bash
```

### Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone transfer request id</td>
</tr>
</tbody>
</table>

### Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>--description DESCRIPTION</td>
<td>Description</td>
</tr>
<tr>
<td>--no-description --target-project-id TARGET_PROJECT_ID</td>
<td>Target project id to transfer to.</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.133. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format {json,shell,table,value,yaml}</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
</tbody>
</table>

Table 92.134. JSON formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

Table 92.135. Shell formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

Table 92.136. Table formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
</tbody>
</table>
92.29. ZONE TRANSFER REQUEST SHOW

Show Zone Transfer Request Details

Usage:

```
openstack zone transfer request show [-h]
    [-f {json,shell,table,value,yaml}]
    [-c COLUMN] [--noindent]
    [--prefix PREFIX]
    [--max-width <integer>]
    [--fit-width] [--print-empty]
    [--all-projects]
    [--sudo-project-id SUDO_PROJECT_ID]
    id
```

Table 92.137. Positional arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Zone transfer request id</td>
</tr>
</tbody>
</table>

Table 92.138. Command arguments

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h, --help</td>
<td>Show this help message and exit</td>
</tr>
<tr>
<td>--all-projects</td>
<td>Show results from all projects. default: false</td>
</tr>
<tr>
<td>--sudo-project-id SUDO_PROJECT_ID</td>
<td>Project id to impersonate for this command. default: None</td>
</tr>
</tbody>
</table>

Table 92.139. Output formatter options

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f {json,shell,table,value,yaml}, --format [json,shell,table,value,yaml]</td>
<td>The output format, defaults to table</td>
</tr>
<tr>
<td>-c COLUMN, --column COLUMN</td>
<td>Specify the column(s) to include, can be repeated to show multiple columns</td>
</tr>
<tr>
<td>Value</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>--noindent</td>
<td>Whether to disable indenting the json</td>
</tr>
</tbody>
</table>

**Table 92.141. Shell formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--prefix PREFIX</td>
<td>Add a prefix to all variable names</td>
</tr>
</tbody>
</table>

**Table 92.142. Table formatter options**

<table>
<thead>
<tr>
<th>Value</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>--max-width &lt;integer&gt;</td>
<td>Maximum display width, &lt;1 to disable. you can also use the CLIFF_MAX_TERM_WIDTH environment variable, but the parameter takes precedence.</td>
</tr>
<tr>
<td>--fit-width</td>
<td>Fit the table to the display width. implied if --max-width greater than 0. Set the environment variable CLIFF_FIT_WIDTH=1 to always enable</td>
</tr>
<tr>
<td>--print-empty</td>
<td>Print empty table if there is no data to show.</td>
</tr>
</tbody>
</table>