Legal Notice

Copyright © 2018 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, 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 Software Collections 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

This guide describes the Red Hat Virtualization Manager Representational State Transfer Application Programming Interface. This guide is generated from documentation comments in the ovirt-engine-api-model code, and is currently partially complete. Updated versions of this documentation will be published as new content becomes available.
# Table of Contents

## CHAPTER 1. INTRODUCTION
- 1.1. REPRESENTATIONAL STATE TRANSFER 44
- 1.2. API PREREQUISITES 44

## CHAPTER 2. AUTHENTICATION AND SECURITY
- 2.1. TLS/SSL CERTIFICATION 46
  - 2.1.1. Obtaining the CA Certificate 46
  - 2.1.2. Importing a Certificate to a Client 48
- 2.2. AUTHENTICATION 48
  - 2.2.1. OAuth Authentication 48
  - 2.2.2. Basic Authentication 50
  - 2.2.3. Authentication Sessions 51
    - 2.2.3.1. Requesting an Authenticated Session 51

## CHAPTER 3. COMMON CONCEPTS
- 3.1. TYPES 53
- 3.2. IDENTIFIED TYPES 53
- 3.3. OBJECTS 53
- 3.4. COLLECTIONS 53
- 3.5. REPRESENTATIONS
  - 3.5.1. XML representation 54
  - 3.5.2. JSON representation 54
- 3.6. SERVICES 55
- 3.7. SEARCHING
  - 3.7.1. Maximum results parameter 58
  - 3.7.2. Case sensitivity 58
  - 3.7.3. Search syntax 59
  - 3.7.4. Wildcards 59
  - 3.7.5. Pagination 59
- 3.8. FOLLOWING LINKS 60
- 3.9. PERMISSIONS 62
- 3.10. HANDLING ERRORS 63

## CHAPTER 4. QUICK START EXAMPLES
- 4.1. ACCESS API ENTRY POINT 64
- 4.2. LIST DATA CENTERS 66
- 4.3. LIST HOST CLUSTERS 67
- 4.4. LIST LOGICAL NETWORKS 68
- 4.5. LIST HOSTS 69
- 4.6. CREATE NFS DATA STORAGE 70
- 4.7. CREATE NFS ISO STORAGE 71
- 4.8. ATTACH STORAGE DOMAINS TO DATA CENTER 73
- 4.9. CREATE VIRTUAL MACHINE 74
- 4.10. CREATE A VIRTUAL MACHINE NIC 76
- 4.11. CREATE VIRTUAL MACHINE DISK 76
- 4.12. ATTACH ISO IMAGE TO VIRTUAL MACHINE 77
- 4.13. START THE VIRTUAL MACHINE 79

## CHAPTER 5. REQUESTS

## CHAPTER 6. SERVICES
- 6.1. AFFINITYGROUP
  - 6.1.1. get GET 110
6.1.1. follow
6.1.2. remove DELETE
6.1.3. update PUT
6.2. AFFINITYGROUPVM
6.2.1. remove DELETE
6.3. AFFINITYGROUPVMS
6.3.1. add POST
6.3.2. list GET
6.3.2.1. follow
6.3.2.2. max
6.4. AFFINITYGROUPS
6.4.1. add POST
6.4.2. list GET
6.4.2.1. follow
6.4.2.2. max
6.5. AFFINITYLABEL
6.5.1. get GET
6.5.1.1. follow
6.5.2. remove DELETE
6.5.3. update PUT
6.6. AFFINITYLABELHOST
6.6.1. get GET
6.6.1.1. follow
6.6.2. remove DELETE
6.7. AFFINITYLABELHOSTS
6.7.1. add POST
6.7.2. list GET
6.7.2.1. follow
6.8. AFFINITYLABELVM
6.8.1. get GET
6.8.1.1. follow
6.8.2. remove DELETE
6.9. AFFINITYLABELVMS
6.9.1. add POST
6.9.2. list GET
6.9.2.1. follow
6.10. AFFINITYLABELS
6.10.1. add POST
6.10.2. list GET
6.10.2.1. follow
6.10.2.2. max
6.11. AREA
6.12. ASSIGNEDAFFINITYLABEL
6.12.1. get GET
6.12.1.1. follow
6.12.2. remove DELETE
6.13. ASSIGNEDAFFINITYLABELS
6.13.1. add POST
6.13.2. list GET
6.13.2.1. follow
6.14. ASSIGNEDCPUPROFILE
6.14.1. get GET
6.14.1.1. follow
6.14.2. remove DELETE

6.15. ASSIGNEDCPU_PROFILES
6.15.1. add POST
6.15.2. list GET
   6.15.2.1. follow
6.15.2.2. max

6.16. ASSIGNED_DISK_PROFILE
6.16.1. get GET
   6.16.1.1. follow
6.16.2. remove DELETE

6.17. ASSIGNED_DISK_PROFILES
6.17.1. add POST
6.17.2. list GET
   6.17.2.1. follow
6.17.2.2. max

6.18. ASSIGNED_PERMISSIONS
6.18.1. add POST
6.18.2. list GET
   6.18.2.1. follow

6.19. ASSIGNED_ROLES
6.19.1. list GET
   6.19.1.1. follow
6.19.1.2. max

6.20. ASSIGNED_TAG
6.20.1. get GET
   6.20.1.1. follow
6.20.2. remove DELETE

6.21. ASSIGNED_TAGS
6.21.1. add POST
6.21.2. list GET
   6.21.2.1. follow
6.21.2.2. max

6.22. ASSIGNED_VNIC_PROFILE
6.22.1. get GET
   6.22.1.1. follow
6.22.2. remove DELETE

6.23. ASSIGNED_VNIC_PROFILES
6.23.1. add POST
6.23.2. list GET
   6.23.2.1. follow
6.23.2.2. max

6.24. ATTACHED_STORAGE_DOMAIN
6.24.1. activate POST
6.24.2. deactivate POST
   6.24.2.1. force
6.24.3. get GET
   6.24.3.1. follow
6.24.4. remove DELETE

6.25. ATTACHED_STORAGE_DOMAIN_DISK
6.25.1. copy POST
6.25.2. export POST
6.25.3. get GET
   6.25.3.1. follow
6.25.4. move POST  
6.25.5. register POST  
6.25.6. remove DELETE  
6.25.7. sparsify POST  
6.25.8. update PUT  

6.26. ATTACHEDSTORAGEDOMAINDISKS  
6.26.1. add POST  
   6.26.1.1. unregistered  
6.26.2. list GET  
   6.26.2.1. disks  
   6.26.2.2. follow  
   6.26.2.3. max  

6.27. ATTACHEDSTORAGEDOMAINS  
6.27.1. add POST  
6.27.2. list GET  
   6.27.2.1. follow  
   6.27.2.2. max  

6.28. BALANCE  
6.28.1. get GET  
   6.28.1.1. follow  
6.28.2. remove DELETE  

6.29. BALANCES  
6.29.1. add POST  
6.29.2. list GET  
   6.29.2.1. follow  
   6.29.2.2. max  

6.30. BOOKMARK  
6.30.1. get GET  
   6.30.1.1. follow  
6.30.2. remove DELETE  
6.30.3. update PUT  

6.31. BOOKMARKS  
6.31.1. add POST  
6.31.2. list GET  
   6.31.2.1. follow  
   6.31.2.2. max  

6.32. CLUSTER  
6.32.1. get GET  
   6.32.1.1. follow  
6.32.2. remove DELETE  
6.32.3. resetemulatedmachine POST  
6.32.4. syncallnetworks POST  
6.32.5. update PUT  

6.33. CLUSTERENABLEDFEATURE  
6.33.1. get GET  
   6.33.1.1. follow  
6.33.2. remove DELETE  

6.34. CLUSTERENABLEDFEATURES  
6.34.1. add POST  
6.34.2. list GET  
   6.34.2.1. follow  

6.35. CLUSTEREXTERNALPROVIDERS  
6.35.1. list GET
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.35.1.1. follow</td>
<td>150</td>
</tr>
<tr>
<td>6.36. CLUSTERFEATURE</td>
<td>150</td>
</tr>
<tr>
<td>6.36.1. get GET</td>
<td>151</td>
</tr>
<tr>
<td>6.36.1.1. follow</td>
<td>151</td>
</tr>
<tr>
<td>6.37. CLUSTERFEATURES</td>
<td>151</td>
</tr>
<tr>
<td>6.37.1. list GET</td>
<td>151</td>
</tr>
<tr>
<td>6.37.1.1. follow</td>
<td>152</td>
</tr>
<tr>
<td>6.38. CLUSTERLEVEL</td>
<td>152</td>
</tr>
<tr>
<td>6.38.1. get GET</td>
<td>152</td>
</tr>
<tr>
<td>6.38.1.1. follow</td>
<td>153</td>
</tr>
<tr>
<td>6.39. CLUSTERLEVELS</td>
<td>153</td>
</tr>
<tr>
<td>6.39.1. list GET</td>
<td>153</td>
</tr>
<tr>
<td>6.39.1.1. follow</td>
<td>154</td>
</tr>
<tr>
<td>6.40. CLUSTERNETWORK</td>
<td>154</td>
</tr>
<tr>
<td>6.40.1. get GET</td>
<td>154</td>
</tr>
<tr>
<td>6.40.1.1. follow</td>
<td>154</td>
</tr>
<tr>
<td>6.40.2. remove DELETE</td>
<td>154</td>
</tr>
<tr>
<td>6.40.3. update PUT</td>
<td>155</td>
</tr>
<tr>
<td>6.41. CLUSTERNETWORKS</td>
<td>155</td>
</tr>
<tr>
<td>6.41.1. add POST</td>
<td>155</td>
</tr>
<tr>
<td>6.41.2. list GET</td>
<td>155</td>
</tr>
<tr>
<td>6.41.2.1. follow</td>
<td>156</td>
</tr>
<tr>
<td>6.41.2.2. max</td>
<td>156</td>
</tr>
<tr>
<td>6.42. CLUSTERS</td>
<td>156</td>
</tr>
<tr>
<td>6.42.1. add POST</td>
<td>156</td>
</tr>
<tr>
<td>6.42.2. list GET</td>
<td>156</td>
</tr>
<tr>
<td>6.42.2.1. case_sensitive</td>
<td>157</td>
</tr>
<tr>
<td>6.42.2.2. follow</td>
<td>158</td>
</tr>
<tr>
<td>6.42.2.3. max</td>
<td>158</td>
</tr>
<tr>
<td>6.43. COPYABLE</td>
<td>158</td>
</tr>
<tr>
<td>6.43.1. copy POST</td>
<td>158</td>
</tr>
<tr>
<td>6.44. CPUPROFILE</td>
<td>158</td>
</tr>
<tr>
<td>6.44.1. get GET</td>
<td>158</td>
</tr>
<tr>
<td>6.44.1.1. follow</td>
<td>159</td>
</tr>
<tr>
<td>6.44.2. remove DELETE</td>
<td>159</td>
</tr>
<tr>
<td>6.44.3. update PUT</td>
<td>159</td>
</tr>
<tr>
<td>6.45. CPUPROFILES</td>
<td>159</td>
</tr>
<tr>
<td>6.45.1. add POST</td>
<td>159</td>
</tr>
<tr>
<td>6.45.2. list GET</td>
<td>160</td>
</tr>
<tr>
<td>6.45.2.1. follow</td>
<td>160</td>
</tr>
<tr>
<td>6.45.2.2. max</td>
<td>160</td>
</tr>
<tr>
<td>6.46. DATACENTER</td>
<td>160</td>
</tr>
<tr>
<td>6.46.1. get GET</td>
<td>160</td>
</tr>
<tr>
<td>6.46.1.1. follow</td>
<td>161</td>
</tr>
<tr>
<td>6.46.2. remove DELETE</td>
<td>161</td>
</tr>
<tr>
<td>6.46.2.1. force</td>
<td>162</td>
</tr>
<tr>
<td>6.46.3. update PUT</td>
<td>162</td>
</tr>
<tr>
<td>6.47. DATACENTERNETWORK</td>
<td>163</td>
</tr>
<tr>
<td>6.47.1. get GET</td>
<td>163</td>
</tr>
<tr>
<td>6.47.1.1. follow</td>
<td>163</td>
</tr>
<tr>
<td>6.47.2. remove DELETE</td>
<td>163</td>
</tr>
<tr>
<td>6.47.3. update PUT</td>
<td>163</td>
</tr>
<tr>
<td>6.48. DATACENTERNETWORKS</td>
<td>164</td>
</tr>
</tbody>
</table>
6.48.1. add POST
6.48.2. list GET
   6.48.2.1. follow
   6.48.2.2. max
6.49. DATACENTERS
   6.49.1. add POST
   6.49.2. list GET
      6.49.2.1. case_sensitive
      6.49.2.2. follow
      6.49.2.3. max
6.50. DISK
   6.50.1. copy POST
      6.50.1.1. disk_profile
      6.50.1.2. quota
      6.50.1.3. storage_domain
   6.50.2. export POST
   6.50.3. get GET
      6.50.3.1. all_content
      6.50.3.2. follow
   6.50.4. move POST
      6.50.4.1. disk_profile
      6.50.4.2. quota
   6.50.5. reduce POST
   6.50.6. refreshlun POST
   6.50.7. remove DELETE
   6.50.8. sparsify POST
   6.50.9. update PUT
6.51. DISKATTACHMENT
   6.51.1. get GET
      6.51.1.1. follow
   6.51.2. remove DELETE
      6.51.2.1. detach_only
   6.51.3. update PUT
6.52. DISKATTACHMENTS
   6.52.1. add POST
   6.52.2. list GET
      6.52.2.1. follow
6.53. DISKPROFILE
   6.53.1. get GET
      6.53.1.1. follow
   6.53.2. remove DELETE
   6.53.3. update PUT
6.54. DISKPROFILES
   6.54.1. add POST
   6.54.2. list GET
      6.54.2.1. follow
      6.54.2.2. max
6.55. DISKSNAPSHOT
   6.55.1. get GET
      6.55.1.1. follow
   6.55.2. remove DELETE
6.56. DISKSNAPSHOTS
   6.56.1. list GET
6.56.1.1. follow
6.56.1.2. max
6.57. DISKS
   6.57.1. add POST
   6.57.2. list GET
      6.57.2.1. case_sensitive
      6.57.2.2. follow
      6.57.2.3. max
6.58. DOMAIN
   6.58.1. get GET
      6.58.1.1. follow
6.59. DOMAINGROUP
   6.59.1. get GET
      6.59.1.1. follow
6.60. DOMAINGROUPS
   6.60.1. list GET
      6.60.1.1. case_sensitive
      6.60.1.2. follow
      6.60.1.3. max
6.61. DOMAINUSER
   6.61.1. get GET
      6.61.1.1. follow
6.62. DOMAINTIERSGROUPS
   6.62.1. list GET
      6.62.1.1. follow
6.63. DOMAINTIERSERS
   6.63.1. list GET
      6.63.1.1. case_sensitive
      6.63.1.2. follow
      6.63.1.3. max
6.64. DOMAINS
   6.64.1. list GET
      6.64.1.1. follow
      6.64.1.2. max
6.65. ENGINEKATELLOERRATA
   6.65.1. list GET
      6.65.1.1. follow
      6.65.1.2. max
6.66. EVENT
   6.66.1. get GET
      6.66.1.1. follow
   6.66.2. remove DELETE
6.67. EVENTS
   6.67.1. add POST
   6.67.2. list GET
      6.67.2.1. case_sensitive
      6.67.2.2. follow
      6.67.2.3. from
      6.67.2.4. max
      6.67.2.5. search
   6.67.3. undelete POST
6.68. EXTERNALCOMPUTERESOURCE
   6.68.1. get GET
6.68.1. follow

6.69. EXTERNALCOMPUTERESOURCES
6.69.1. list GET
   6.69.1.1. follow
   6.69.1.2. max

6.70. EXTERNALDISCOVEREDHOST
6.70.1. get GET
   6.70.1.1. follow

6.71. EXTERNALDISCOVEREDHOSTS
6.71.1. list GET
   6.71.1.1. follow
   6.71.1.2. max

6.72. EXTERNALHOST
6.72.1. get GET
   6.72.1.1. follow

6.73. EXTERNALHOSTGROUP
6.73.1. get GET
   6.73.1.1. follow

6.74. EXTERNALHOSTGROUPS
6.74.1. list GET
   6.74.1.1. follow
   6.74.1.2. max

6.75. EXTERNALHOSTPROVIDER
6.75.1. get GET
   6.75.1.1. follow
   6.75.2. importcertificates POST
   6.75.3. remove DELETE
   6.75.4. testconnectivity POST
   6.75.5. update PUT

6.76. EXTERNALHOSTPROVIDERS
6.76.1. add POST
6.76.2. list GET
   6.76.2.1. follow
   6.76.2.2. max

6.77. EXTERNALHOSTS
6.77.1. list GET
   6.77.1.1. follow
   6.77.1.2. max

6.78. EXTERNALNETWORKPROVIDERCONFIGURATION
6.78.1. get GET
   6.78.1.1. follow

6.79. EXTERNALNETWORKPROVIDERCONFIGURATIONS
6.79.1. list GET
   6.79.1.1. follow

6.80. EXTERNALPROVIDER
6.80.1. importcertificates POST
6.80.2. testconnectivity POST

6.81. EXTERNALPROVIDERCERTIFICATE
6.81.1. get GET
   6.81.1.1. follow

6.82. EXTERNALPROVIDERCERTIFICATES
6.82.1. list GET
   6.82.1.1. follow
6.82.1.2. max
6.83. EXTERNALVMIMPORTS
6.83.1. add POST
6.84. FENCEAGENT
  6.84.1. get GET
  6.84.1.1. follow
  6.84.2. remove DELETE
  6.84.3. update PUT
6.85. FENCEAGENTS
  6.85.1. add POST
  6.85.2. list GET
    6.85.2.1. follow
    6.85.2.2. max
6.86. FILE
  6.86.1. get GET
    6.86.1.1. follow
6.87. FILES
  6.87.1. list GET
    6.87.1.1. case_sensitive
    6.87.1.2. follow
    6.87.1.3. max
6.88. FILTER
  6.88.1. get GET
    6.88.1.1. follow
  6.88.2. remove DELETE
6.89. FILTERS
  6.89.1. add POST
  6.89.2. list GET
    6.89.2.1. follow
    6.89.2.2. max
6.90. FOLLOW
6.91. GLUSTERBRICK
  6.91.1. get GET
    6.91.1.1. follow
  6.91.2. remove DELETE
  6.91.3. replace POST
6.92. GLUSTERBRICKS
  6.92.1. activate POST
  6.92.2. add POST
  6.92.3. list GET
    6.92.3.1. follow
    6.92.3.2. max
  6.92.4. migrate POST
  6.92.5. remove DELETE
  6.92.6. stopmigrate POST
    6.92.6.1. bricks
6.93. GLUSTERHOOK
  6.93.1. disable POST
  6.93.2. enable POST
  6.93.3. get GET
    6.93.3.1. follow
  6.93.4. remove DELETE
  6.93.5. resolve POST
6.94. GLUSTERHOOKS
   6.94.1. list GET
       6.94.1.1. follow
       6.94.1.2. max

6.95. GLUSTERVOLUME
   6.95.1. get GET
       6.95.1.1. follow
   6.95.2. getprofilestatistics POST
   6.95.3. rebalance POST
       6.95.3.1. fix_layout
       6.95.3.2. force
   6.95.4. remove DELETE
   6.95.5. resetalloptions POST
   6.95.6. resetoption POST
   6.95.7. setoption POST
   6.95.8. start POST
       6.95.8.1. force
   6.95.9. startprofile POST
   6.95.10. stop POST
   6.95.11. stopprofile POST
   6.95.12. stoprebalance POST

6.96. GLUSTERVOLUMES
   6.96.1. add POST
   6.96.2. list GET
       6.96.2.1. case_sensitive
       6.96.2.2. follow
       6.96.2.3. max

6.97. GROUP
   6.97.1. get GET
       6.97.1.1. follow
   6.97.2. remove DELETE

6.98. GROUPS
   6.98.1. add POST
   6.98.2. list GET
       6.98.2.1. case_sensitive
       6.98.2.2. follow
       6.98.2.3. max

6.99. HOST
   6.99.1. activate POST
   6.99.2. approve POST
   6.99.3. commitnetconfig POST
   6.99.4. deactivate POST
       6.99.4.1. stop_gluster_service
   6.99.5. enrollcertificate POST
   6.99.6. fence POST
   6.99.7. forceselectspm POST
   6.99.8. get GET
       6.99.8.1. all_content
       6.99.8.2. follow
   6.99.9. install POST
       6.99.9.1. deploy_hosted_engine
       6.99.9.2. undeploy_hosted_engine
   6.99.10. iscsidiscover POST
6.99.10.1. iscsi_targets
6.99.11. iscsilogin POST
6.99.12. refresh POST
6.99.13. remove DELETE
6.99.14. setupnetworks POST
6.99.15. syncallnetworks POST
6.99.16. unregisteredstoragedomainsdiscover POST
6.99.17. update PUT
6.99.18. upgrade POST
   6.99.18.1. image
   6.99.18.2. reboot
6.99.19. upgradecheck POST
6.100. HOSTDEVICE
   6.100.1. get GET
      6.100.1.1. follow
6.101. HOSTDEVICES
   6.101.1. list GET
      6.101.1.1. follow
      6.101.1.2. max
6.102. HOSTHOOK
   6.102.1. get GET
      6.102.1.1. follow
6.103. HOSTHOOKS
   6.103.1. list GET
      6.103.1.1. follow
      6.103.1.2. max
6.104. HOSTNIC
   6.104.1. get GET
      6.104.1.1. follow
6.104.2. updatevirtualfunctionsconfiguration POST
6.105. HOSTNICS
   6.105.1. list GET
      6.105.1.1. follow
      6.105.1.2. max
6.106. HOSTNUMANODE
   6.106.1. get GET
      6.106.1.1. follow
6.107. HOSTNUMANODES
   6.107.1. list GET
      6.107.1.1. follow
      6.107.1.2. max
6.108. HOSTSTORAGE
   6.108.1. list GET
      6.108.1.1. follow
      6.108.1.2. report_status
6.109. HOSTS
   6.109.1. add POST
      6.109.1.1. deploy_hosted_engine
      6.109.1.2. undeploy_hosted_engine
   6.109.2. list GET
      6.109.2.1. all_content
      6.109.2.2. case_sensitive
      6.109.2.3. follow
6.121. get GET
  6.121.1. follow
  6.121.2. remove DELETE
  6.121.3. update PUT

6.122. INSTANCETYPEWATCHDOGS
  6.122.1. add POST
  6.122.2. list GET
    6.122.2.1. follow
    6.122.2.2. max

6.123. INSTANCETYPES
  6.123.1. add POST
  6.123.2. list GET
    6.123.2.1. case_sensitive
    6.123.2.2. follow
    6.123.2.3. max

6.124. ISCSIBOND
  6.124.1. get GET
    6.124.1.1. follow
  6.124.2. remove DELETE
  6.124.3. update PUT

6.125. ISCSIBONDS
  6.125.1. add POST
  6.125.2. list GET
    6.125.2.1. follow
    6.125.2.2. max

6.126. JOB
  6.126.1. clear POST
  6.126.2. end POST
    6.126.2.1. succeeded
  6.126.3. get GET
    6.126.3.1. follow

6.127. JOBS
  6.127.1. add POST
  6.127.2. list GET
    6.127.2.1. case_sensitive
    6.127.2.2. follow
    6.127.2.3. max

6.128. KATELLOERRATA
  6.128.1. list GET
    6.128.1.1. follow
    6.128.1.2. max

6.129. KATELLOERRATUM
  6.129.1. get GET
    6.129.1.1. follow

6.130. LINKLAYERDISCOVERYPROTOCOL
  6.130.1. list GET
    6.130.1.1. elements
    6.130.1.2. follow

6.131. MACPOOL
  6.131.1. get GET
    6.131.1.1. follow
  6.131.2. remove DELETE
  6.131.3. update PUT
6.132. MACPOOLS
  6.132.1. add POST
  6.132.2. list GET
    6.132.2.1. follow
    6.132.2.2. max
  6.133. MEASURABLE
  6.134. MOVEABLE
    6.134.1. move POST
  6.135. NETWORK
    6.135.1. get GET
      6.135.1.1. follow
    6.135.2. remove DELETE
    6.135.3. update PUT
  6.136. NETWORKATTACHMENT
    6.136.1. get GET
      6.136.1.1. follow
    6.136.2. remove DELETE
    6.136.3. update PUT
  6.137. NETWORKATTACHMENTS
    6.137.1. add POST
    6.137.2. list GET
      6.137.2.1. follow
      6.137.2.2. max
  6.138. NETWORKFILTER
    6.138.1. get GET
      6.138.1.1. follow
  6.139. NETWORKFILTERS
    6.139.1. list GET
      6.139.1.1. follow
  6.140. NETWORKLABEL
    6.140.1. get GET
      6.140.1.1. follow
    6.140.2. remove DELETE
  6.141. NETWORKLABELS
    6.141.1. add POST
    6.141.2. list GET
      6.141.2.1. follow
      6.141.2.2. max
  6.142. NETWORKS
    6.142.1. add POST
    6.142.2. list GET
      6.142.2.1. case_sensitive
      6.142.2.2. follow
      6.142.2.3. max
  6.143. NICNETWORKFILTERPARAMETER
    6.143.1. get GET
      6.143.1.1. follow
    6.143.2. remove DELETE
    6.143.3. update PUT
  6.144. NICNETWORKFILTERPARAMETERS
    6.144.1. add POST
    6.144.2. list GET
      6.144.2.1. follow
6.145. OPENSTACKIMAGE
  6.145.1. get GET
    6.145.1.1. follow
  6.145.2. import POST

6.146. OPENSTACKIMAGEPROVIDER
  6.146.1. get GET
    6.146.1.1. follow
  6.146.2. importcertificates POST
  6.146.3. remove DELETE
  6.146.4. testconnectivity POST
  6.146.5. update PUT

6.147. OPENSTACKIMAGEPROVIDERS
  6.147.1. add POST
  6.147.2. list GET
    6.147.2.1. follow
    6.147.2.2. max

6.148. OPENSTACKIMAGES
  6.148.1. list GET
    6.148.1.1. follow
    6.148.1.2. max

6.149. OPENSTACKNETWORK
  6.149.1. get GET
    6.149.1.1. follow
  6.149.2. import POST
    6.149.2.1. data_center

6.150. OPENSTACKNETWORKPROVIDER
  6.150.1. get GET
    6.150.1.1. follow
  6.150.2. importcertificates POST
  6.150.3. remove DELETE
  6.150.4. testconnectivity POST
  6.150.5. update PUT

6.151. OPENSTACKNETWORKPROVIDERS
  6.151.1. add POST
  6.151.2. list GET
    6.151.2.1. follow
    6.151.2.2. max

6.152. OPENSTACKNETWORKS
  6.152.1. list GET
    6.152.1.1. follow
    6.152.1.2. max

6.153. OPENSTACKSUBNET
  6.153.1. get GET
    6.153.1.1. follow
  6.153.2. remove DELETE

6.154. OPENSTACKSUBNETS
  6.154.1. add POST
  6.154.2. list GET
    6.154.2.1. follow
    6.154.2.2. max

6.155. OPENSTACKVOLUMEAUTHENTICATIONKEY
  6.155.1. get GET
    6.155.1.1. follow
6.155.2. remove DELETE
6.155.3. update PUT
6.156. OPENSTACKVOLUMEAUTHENTICATIONKEYS
   6.156.1. add POST
   6.156.2. list GET
      6.156.2.1. follow
      6.156.2.2. max
6.157. OPENSTACKVOLUMEPROVIDER
   6.157.1. get GET
      6.157.1.1. follow
   6.157.2. importcertificates POST
   6.157.3. remove DELETE
      6.157.3.1. force
   6.157.4. testconnectivity POST
   6.157.5. update PUT
6.158. OPENSTACKVOLUMEPROVIDERS
   6.158.1. add POST
   6.158.2. list GET
      6.158.2.1. follow
      6.158.2.2. max
6.159. OPENSTACKVOLUMETYPE
   6.159.1. get GET
      6.159.1.1. follow
6.160. OPENSTACKVOLUMETYPES
   6.160.1. list GET
      6.160.1.1. follow
      6.160.1.2. max
6.161. OPERATINGSYSTEM
   6.161.1. get GET
      6.161.1.1. follow
6.162. OPERATINGSYSTEMS
   6.162.1. list GET
      6.162.1.1. follow
      6.162.1.2. max
6.163. PERMISSION
   6.163.1. get GET
      6.163.1.1. follow
   6.163.2. remove DELETE
6.164. PERMIT
   6.164.1. get GET
      6.164.1.1. follow
   6.164.2. remove DELETE
6.165. PERMITS
   6.165.1. add POST
   6.165.2. list GET
      6.165.2.1. follow
      6.165.2.2. max
6.166. QOS
   6.166.1. get GET
      6.166.1.1. follow
   6.166.2. remove DELETE
   6.166.3. update PUT
6.167. QOSS
6.167.1. add POST
6.167.2. list GET
   6.167.2.1. follow
   6.167.2.2. max
6.168. QUOTA
   6.168.1. get GET
   6.168.1.1. follow
   6.168.2. remove DELETE
   6.168.3. update PUT
6.169. QUOTA.CLUSTERLIMIT
   6.169.1. get GET
   6.169.1.1. follow
   6.169.2. remove DELETE
6.170. QUOTA.CLUSTERLIMITS
   6.170.1. add POST
   6.170.2. list GET
      6.170.2.1. follow
      6.170.2.2. max
6.171. QUOTA.STORAGELIMIT
   6.171.1. get GET
      6.171.1.1. follow
   6.171.2. remove DELETE
6.172. QUOTA.STORAGELIMITS
   6.172.1. add POST
   6.172.2. list GET
      6.172.2.1. follow
      6.172.2.2. max
6.173. QUOTAS
   6.173.1. add POST
   6.173.2. list GET
      6.173.2.1. follow
      6.173.2.2. max
6.174. ROLE
   6.174.1. get GET
      6.174.1.1. follow
   6.174.2. remove DELETE
   6.174.3. update PUT
6.175. ROLES
   6.175.1. add POST
   6.175.2. list GET
      6.175.2.1. follow
      6.175.2.2. max
6.176. SCHEDULINGPOLICIES
   6.176.1. add POST
   6.176.2. list GET
      6.176.2.1. follow
      6.176.2.2. max
6.177. SCHEDULINGPOLICY
   6.177.1. get GET
      6.177.1.1. follow
   6.177.2. remove DELETE
   6.177.3. update PUT
6.178. SCHEDULINGPOLICYUNIT
6.178.1. get GET
6.178.1.1. follow
6.178.2. remove DELETE

6.179. SCHEDULINGPOLICYUNITS
6.179.1. list GET
6.179.1.1. follow
6.179.1.2. max

6.180. SNAPSHOT
6.180.1. get GET
6.180.1.1. follow
6.180.2. remove DELETE
6.180.2.1. all_content
6.180.3. restore POST
6.180.3.1. disks

6.181. SNAPSHOTCDROM
6.181.1. get GET
6.181.1.1. follow

6.182. SNAPSHOTCDROMS
6.182.1. list GET
6.182.1.1. follow
6.182.1.2. max

6.183. SNAPSHOTDISK
6.183.1. get GET
6.183.1.1. follow

6.184. SNAPSHOTDISKS
6.184.1. list GET
6.184.1.1. follow
6.184.1.2. max

6.185. SNAPSHOTNIC
6.185.1. get GET
6.185.1.1. follow

6.186. SNAPSHOTNICS
6.186.1. list GET
6.186.1.1. follow
6.186.1.2. max

6.187. SNAPSHOTS
6.187.1. add POST
6.187.2. list GET
6.187.2.1. all_content
6.187.2.2. follow
6.187.2.3. max

6.188. SSHPUBLICKEY
6.188.1. get GET
6.188.1.1. follow
6.188.2. remove DELETE
6.188.3. update PUT

6.189. SSHPUBLICKEYS
6.189.1. add POST
6.189.2. list GET
6.189.2.1. follow
6.189.2.2. max

6.190. STATISTIC
6.190.1. get GET
6.199.2.2. max
6.199.2.3. unregistered

6.200. STORAGEDOMAINSERVERCONNECTION
6.200.1. get GET
6.200.1.1. follow
6.200.2. remove DELETE

6.201. STORAGEDOMAINSERVERCONNECTIONS
6.201.1. add POST
6.201.2. list GET
6.201.2.1. follow
6.201.2.2. max

6.202. STORAGEDOMAINTEMPLATE
6.202.1. get GET
6.202.1.1. follow
6.202.2. import POST
6.202.2.1. clone
6.202.3. register POST
6.202.3.1. allow_partial_import
6.202.3.2. registration_configuration
6.202.3.3. vnic_profile_mappings
6.202.4. remove DELETE

6.203. STORAGEDOMAINTEMPLATES
6.203.1. list GET
6.203.1.1. follow
6.203.1.2. max
6.203.1.3. unregistered

6.204. STORAGEDOMAINVM
6.204.1. get GET
6.204.1.1. follow
6.204.2. import POST
6.204.2.1. clone
6.204.2.2. collapse_snapshots
6.204.3. register POST
6.204.3.1. allow_partial_import
6.204.3.2. reassign_bad_macs
6.204.3.3. registration_configuration
6.204.3.4. vnic_profile_mappings
6.204.4. remove DELETE

6.205. STORAGEDOMAINVMDISKATTACHMENT
6.205.1. get GET
6.205.1.1. follow

6.206. STORAGEDOMAINVMDISKATTACHMENTS
6.206.1. list GET
6.206.1.1. follow

6.207. STORAGEDOMAINVMS
6.207.1. list GET
6.207.1.1. follow
6.207.1.2. max
6.207.1.3. unregistered

6.208. STORAGEDOMAINS
6.208.1. add POST
6.208.2. list GET
6.208.2.1. case_sensitive
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.208.2.2. follow</td>
<td>398</td>
</tr>
<tr>
<td>6.208.2.3. max</td>
<td>398</td>
</tr>
<tr>
<td>6.209. STORAGESERVERCONNECTION</td>
<td>398</td>
</tr>
<tr>
<td>6.209.1. get GET</td>
<td>398</td>
</tr>
<tr>
<td>6.209.1.1. follow</td>
<td>399</td>
</tr>
<tr>
<td>6.209.2. remove DELETE</td>
<td>399</td>
</tr>
<tr>
<td>6.209.2.1. host</td>
<td>399</td>
</tr>
<tr>
<td>6.209.3. update PUT</td>
<td>399</td>
</tr>
<tr>
<td>6.209.3.1. force</td>
<td>400</td>
</tr>
<tr>
<td>6.210. STORAGESERVERCONNECTIONEXTENSION</td>
<td>400</td>
</tr>
<tr>
<td>6.210.1. get GET</td>
<td>400</td>
</tr>
<tr>
<td>6.210.1.1. follow</td>
<td>401</td>
</tr>
<tr>
<td>6.210.2. remove DELETE</td>
<td>401</td>
</tr>
<tr>
<td>6.210.3. update PUT</td>
<td>401</td>
</tr>
<tr>
<td>6.211. STORAGESERVERCONNECTIONEXTENSIONS</td>
<td>402</td>
</tr>
<tr>
<td>6.211.1. add POST</td>
<td>402</td>
</tr>
<tr>
<td>6.211.2. list GET</td>
<td>403</td>
</tr>
<tr>
<td>6.211.2.1. follow</td>
<td>403</td>
</tr>
<tr>
<td>6.211.2.2. max</td>
<td>403</td>
</tr>
<tr>
<td>6.212. STORAGESERVERCONNECTIONS</td>
<td>403</td>
</tr>
<tr>
<td>6.212.1. add POST</td>
<td>403</td>
</tr>
<tr>
<td>6.212.2. list GET</td>
<td>404</td>
</tr>
<tr>
<td>6.212.2.1. follow</td>
<td>404</td>
</tr>
<tr>
<td>6.212.2.2. max</td>
<td>404</td>
</tr>
<tr>
<td>6.213. SYSTEM</td>
<td>404</td>
</tr>
<tr>
<td>6.213.1. get GET</td>
<td>404</td>
</tr>
<tr>
<td>6.213.1.1. follow</td>
<td>406</td>
</tr>
<tr>
<td>6.213.2. reloadconfigurations POST</td>
<td>406</td>
</tr>
<tr>
<td>6.214. SYSTEMOPTION</td>
<td>406</td>
</tr>
<tr>
<td>6.214.1. get GET</td>
<td>406</td>
</tr>
<tr>
<td>6.214.1.1. version</td>
<td>408</td>
</tr>
<tr>
<td>6.215. SYSTEMOPTIONS</td>
<td>408</td>
</tr>
<tr>
<td>6.216. SYSTEMPERMISSIONS</td>
<td>408</td>
</tr>
<tr>
<td>6.216.1. add POST</td>
<td>408</td>
</tr>
<tr>
<td>6.216.2. list GET</td>
<td>410</td>
</tr>
<tr>
<td>6.216.2.1. follow</td>
<td>410</td>
</tr>
<tr>
<td>6.217. TAG</td>
<td>410</td>
</tr>
<tr>
<td>6.217.1. get GET</td>
<td>411</td>
</tr>
<tr>
<td>6.217.1.1. follow</td>
<td>411</td>
</tr>
<tr>
<td>6.217.2. remove DELETE</td>
<td>411</td>
</tr>
<tr>
<td>6.217.3. update PUT</td>
<td>411</td>
</tr>
<tr>
<td>6.218. TAGS</td>
<td>412</td>
</tr>
<tr>
<td>6.218.1. add POST</td>
<td>412</td>
</tr>
<tr>
<td>6.218.2. list GET</td>
<td>413</td>
</tr>
<tr>
<td>6.218.2.1. follow</td>
<td>414</td>
</tr>
<tr>
<td>6.218.2.2. max</td>
<td>414</td>
</tr>
<tr>
<td>6.219. TEMPLATE</td>
<td>414</td>
</tr>
<tr>
<td>6.219.1. export POST</td>
<td>414</td>
</tr>
<tr>
<td>6.219.1.1. exclusive</td>
<td>415</td>
</tr>
<tr>
<td>6.219.2. get GET</td>
<td>415</td>
</tr>
<tr>
<td>6.219.2.1. follow</td>
<td>415</td>
</tr>
<tr>
<td>6.219.3. remove DELETE</td>
<td>415</td>
</tr>
<tr>
<td>6.219.4. update PUT</td>
<td>416</td>
</tr>
</tbody>
</table>
6.239.11. previewsnapshot POST
  6.239.11.1. disks
  6.239.11.2. lease
6.239.12. reboot POST
6.239.13. remove DELETE
  6.239.13.1. force
6.239.14. reordermacaddresses POST
6.239.15. shutdown POST
6.239.16. start POST
  6.239.16.1. pause
  6.239.16.2. use_cloud_init
  6.239.16.3. use_sysprep
  6.239.16.4. vm
  6.239.16.5. volatile
6.239.17. stop POST
6.239.18. suspend POST
6.239.19. thawfilesystems POST
6.239.20. ticket POST
6.239.21. undosnapshot POST
6.239.22. update PUT
  6.239.22.1. next_run
6.240. VMAPPLICATION
  6.240.1. get GET
    6.240.1.1. application
    6.240.1.2. follow
6.241. VMAPPLICATIONS
  6.241.1. list GET
    6.241.1.1. applications
    6.241.1.2. follow
    6.241.1.3. max
6.242. VMCDROM
  6.242.1. get GET
    6.242.1.1. current
    6.242.1.2. follow
  6.242.2. update PUT
    6.242.2.1. current
6.243. VMCDROMS
  6.243.1. add POST
  6.243.2. list GET
    6.243.2.1. follow
    6.243.2.2. max
6.244. VMDISK
  6.244.1. activate POST
  6.244.2. deactivate POST
  6.244.3. export POST
  6.244.4. get GET
    6.244.4.1. follow
  6.244.5. move POST
  6.244.6. reduce POST
  6.244.7. remove DELETE
  6.244.8. update PUT
6.245. VMDISKS
  6.245.1. add POST
6.245.2. list GET
6.245.2.1. follow
6.245.2.2. max

6.246. VMGRAPHICSCONSOLE
6.246.1. get GET
6.246.1.1. current
6.246.1.2. follow
6.246.2. proxyticket POST
6.246.3. remoteviewerconnectionfile POST
   6.246.3.1. remote_viewer_connection_file
6.246.4. remove DELETE
6.246.5. ticket POST

6.247. VMGRAPHICSCONSOLES
6.247.1. add POST
6.247.2. list GET
   6.247.2.1. current
   6.247.2.2. follow
   6.247.2.3. max

6.248. VMHOSTDEVICE
6.248.1. get GET
   6.248.1.1. follow
6.248.2. remove DELETE

6.249. VMHOSTDEVICES
6.249.1. add POST
6.249.2. list GET
   6.249.2.1. follow
   6.249.2.2. max

6.250. VMNIC
6.250.1. activate POST
6.250.2. deactivate POST
6.250.3. get GET
   6.250.3.1. follow
6.250.4. remove DELETE
6.250.5. update PUT

6.251. VMNICS
6.251.1. add POST
6.251.2. list GET
   6.251.2.1. follow
   6.251.2.2. max

6.252. VMNUMANODE
6.252.1. get GET
   6.252.1.1. follow
6.252.2. remove DELETE
6.252.3. update PUT

6.253. VMNUMANODES
6.253.1. add POST
6.253.2. list GET
   6.253.2.1. follow
   6.253.2.2. max

6.254. VMPOOL
6.254.1. allocatevm POST
6.254.2. get GET
   6.254.2.1. follow
6.254.3. remove DELETE
6.254.4. update PUT
6.255. VMPOOLS
6.255.1. add POST
6.255.2. list GET
   6.255.2.1. case_sensitive
   6.255.2.2. follow
   6.255.2.3. max
6.256. VMREPORTEDDEVICE
6.256.1. get GET
   6.256.1.1. follow
6.257. VMREPORTEDDEVICES
6.257.1. list GET
   6.257.1.1. follow
   6.257.1.2. max
6.258. VMSESSION
6.258.1. get GET
   6.258.1.1. follow
6.259. VMSESSIONS
6.259.1. list GET
   6.259.1.1. follow
   6.259.1.2. max
6.260. VMWATCHDOG
6.260.1. get GET
   6.260.1.1. follow
   6.260.1.2. watchdog
6.260.2. remove DELETE
6.260.3. update PUT
   6.260.3.1. watchdog
6.261. VMWATCHDOGS
6.261.1. add POST
   6.261.1.1. watchdog
6.261.2. list GET
   6.261.2.1. follow
   6.261.2.2. max
   6.261.2.3. watchdogs
6.262. VMS
6.262.1. add POST
   6.262.1.1. clone
   6.262.1.2. clone_permissions
6.262.2. list GET
   6.262.2.1. all_content
   6.262.2.2. case_sensitive
   6.262.2.3. follow
6.263. VNICPROFILE
6.263.1. get GET
   6.263.1.1. follow
6.263.2. remove DELETE
6.263.3. update PUT
6.264. VNICPROFILES
6.264.1. add POST
6.264.2. list GET
   6.264.2.1. follow
7.47. DISKATTACHMENT STRUCT
  7.47.1. active
  7.47.2. logical_name
  7.47.3. read_only
  7.47.4. uses_scsi_reservation
7.48. DISKCONTENTTYPE ENUM
7.49. DISKFORMAT ENUM
7.50. DISKINTERFACE ENUM
  7.50.1. ide
  7.50.2. virtio
  7.50.3. virtio_scsi
7.51. DISKPROFILE STRUCT
7.52. DISKSNAPSHOT STRUCT
  7.52.1. active
  7.52.2. actual_size
  7.52.3. bootable
  7.52.4. initial_size
  7.52.5. interface
  7.52.6. provisioned_size
  7.52.7. qcow_version
  7.52.8. read_only
  7.52.9. shareable
  7.52.10. total_size
  7.52.11. wipe_after_delete
  7.52.12. statistics
  7.52.13. storage_domains
  7.52.14. vms
7.53. DISKSTATUS ENUM
  7.53.1. locked
7.54. DISKSTORAGETYPE ENUM
7.55. DISKTYPE ENUM
7.56. DISPLAY STRUCT
  7.56.1. allow_override
  7.56.2. certificate
  7.56.3. copy_paste_enabled
  7.56.4. disconnect_action
  7.56.5. file_transfer_enabled
  7.56.6. keyboard_layout
  7.56.7. monitors
  7.56.8. proxy
  7.56.9. secure_port
  7.56.10. single_qxl_pci
  7.56.11. smartcard_enabled
7.57. DISPLAYTYPE ENUM
  7.57.1. spice
  7.57.2. vnc
7.58. DNS STRUCT
7.59. DNSSERVICES CONFIGURATION STRUCT
  7.59.1. name_servers
7.60. DOMAIN STRUCT
  7.60.1. users
7.61. ENTITYEXTERNALSTATUS ENUM
  7.61.1. error
7.61.2. failure
7.62. ENTITYPROFILEDETAIL STRUCT
7.63. ERRORHANDLING STRUCT
7.64. EVENT STRUCT
  7.64.1. correlation_id
  7.64.2. flood_rate
  7.64.3. index
  7.64.4. cluster
  7.64.5. data_center
  7.64.6. host
  7.64.7. storage_domain
  7.64.8. template
  7.64.9. user
  7.64.10. vm
7.65. EXTERNALCOMPUTERESOURCE STRUCT
7.66. EXTERNALDISCOVEREDHOST STRUCT
7.67. EXTERNALHOST STRUCT
7.68. EXTERNALHOSTGROUP STRUCT
7.69. EXTERNALHOSTPROVIDER STRUCT
  7.69.1. requires_authentication
  7.69.2. compute_resources
  7.69.3. discovered_hosts
  7.69.4. host_groups
7.70. EXTERNALNETWORKPROVIDERCONFIGURATION STRUCT
7.71. EXTERNALPROVIDER STRUCT
  7.71.1. requires_authentication
7.72. EXTERNALSTATUS ENUM
  7.72.1. error
  7.72.2. failure
  7.72.3. info
  7.72.4. ok
  7.72.5. warning
7.73. EXTERNALSYSTEMTYPE ENUM
7.74. EXTERNALVMIMPORT STRUCT
  7.74.1. url
  7.74.2. cpu_profile
  7.74.3. drivers_iso
  7.74.4. host
  7.74.5. quota
  7.74.6. vm
7.75. EXTERNALVMPROVIDERTYPE ENUM
7.76. FAULT STRUCT
7.77. FENCETYPE ENUM
7.78. FENCINGPOLICY STRUCT
  7.78.1. skip_if_connectivity_broken
  7.78.2. skip_if_gluster_bricks_up
  7.78.3. skip_if_gluster_quorum_not_met
  7.78.4. skip_if_sd_active
7.79. FILE STRUCT
7.80. FILTER STRUCT
7.81. FIREWALLTYPE ENUM
  7.81.1. firewalld
  7.81.2. iptables
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.82</td>
<td>FLOPPY STRUCT</td>
</tr>
<tr>
<td>7.82.1</td>
<td>vms</td>
</tr>
<tr>
<td>7.83</td>
<td>FOPSTATISTIC STRUCT</td>
</tr>
<tr>
<td>7.84</td>
<td>GLUSTERBRICK STRUCT</td>
</tr>
<tr>
<td>7.84.1</td>
<td>vms</td>
</tr>
<tr>
<td>7.85</td>
<td>GLUSTERBRICKADVANCEDDETAILS STRUCT</td>
</tr>
<tr>
<td>7.85.1</td>
<td>vms</td>
</tr>
<tr>
<td>7.86</td>
<td>GLUSTERBRICKMEMORYINFO STRUCT</td>
</tr>
<tr>
<td>7.87</td>
<td>GLUSTERBRICKSTATUS ENUM</td>
</tr>
<tr>
<td>7.88</td>
<td>GLUSTERCLIENT STRUCT</td>
</tr>
<tr>
<td>7.89</td>
<td>GLUSTERHOOK STRUCT</td>
</tr>
<tr>
<td>7.90</td>
<td>GLUSTERHOOKSTATUS ENUM</td>
</tr>
<tr>
<td>7.91</td>
<td>GLUSTERMEMORYPOOL STRUCT</td>
</tr>
<tr>
<td>7.92</td>
<td>GLUSTERSERVERHOOK STRUCT</td>
</tr>
<tr>
<td>7.93</td>
<td>GLUSTERSTATE ENUM</td>
</tr>
<tr>
<td>7.94</td>
<td>GLUSTERVOLUME STRUCT</td>
</tr>
<tr>
<td>7.95</td>
<td>GLUSTERVOLUMEPROFILEDETAILS STRUCT</td>
</tr>
<tr>
<td>7.96</td>
<td>GLUSTERVOLUMESTATUS ENUM</td>
</tr>
<tr>
<td>7.97</td>
<td>GLUSTERVOLUMETYPE ENUM</td>
</tr>
<tr>
<td>7.97.1</td>
<td>disperse</td>
</tr>
<tr>
<td>7.97.2</td>
<td>distribute</td>
</tr>
<tr>
<td>7.97.3</td>
<td>distributed_disperse</td>
</tr>
<tr>
<td>7.97.4</td>
<td>distributed_replicate</td>
</tr>
<tr>
<td>7.97.5</td>
<td>distributed_stripe</td>
</tr>
<tr>
<td>7.97.6</td>
<td>distributed_striped_replicate</td>
</tr>
<tr>
<td>7.97.7</td>
<td>replicate</td>
</tr>
<tr>
<td>7.97.8</td>
<td>stripe</td>
</tr>
<tr>
<td>7.97.9</td>
<td>striped_replicate</td>
</tr>
<tr>
<td>7.98</td>
<td>GRACEPERIOD STRUCT</td>
</tr>
<tr>
<td>7.99</td>
<td>GRAPHICSCONSOLE STRUCT</td>
</tr>
<tr>
<td>7.100</td>
<td>GRAPHICSTYPE ENUM</td>
</tr>
<tr>
<td>7.100.1</td>
<td>spice</td>
</tr>
<tr>
<td>7.100.2</td>
<td>vnc</td>
</tr>
<tr>
<td>7.101</td>
<td>GROUP STRUCT</td>
</tr>
<tr>
<td>7.101.1</td>
<td>roles</td>
</tr>
<tr>
<td>7.102</td>
<td>GUESTOPERATINGSYSTEM STRUCT</td>
</tr>
<tr>
<td>7.103</td>
<td>HARDWAREINFORMATION STRUCT</td>
</tr>
<tr>
<td>7.104</td>
<td>HIGHAVAILABILITY STRUCT</td>
</tr>
<tr>
<td>7.104.1</td>
<td>priority</td>
</tr>
<tr>
<td>7.105</td>
<td>HOOK STRUCT</td>
</tr>
<tr>
<td>7.106</td>
<td>HOOKCONTENTTYPE ENUM</td>
</tr>
<tr>
<td>7.107</td>
<td>HOOKSTAGE ENUM</td>
</tr>
<tr>
<td>7.108</td>
<td>HOOKSTATUS ENUM</td>
</tr>
<tr>
<td>7.109</td>
<td>HOST STRUCT</td>
</tr>
<tr>
<td>7.109.1</td>
<td>external_status</td>
</tr>
<tr>
<td>7.109.2</td>
<td>hosted_engine</td>
</tr>
<tr>
<td>7.109.3</td>
<td>kdump_status</td>
</tr>
<tr>
<td>7.109.4</td>
<td>ksm</td>
</tr>
<tr>
<td>7.109.5</td>
<td>libvirt_version</td>
</tr>
<tr>
<td>7.109.6</td>
<td>override_iptables</td>
</tr>
<tr>
<td>7.109.7</td>
<td>protocol</td>
</tr>
<tr>
<td>7.109.8</td>
<td>se_linux</td>
</tr>
<tr>
<td>7.109.9</td>
<td>spm</td>
</tr>
</tbody>
</table>
7.109.10. status_detail
7.109.11. transparent_huge_pages
7.109.12. version
7.109.13. external_network_provider_configurations
7.109.14. katello_errata
7.109.15. statistics
7.110. HOSTDEVICE STRUCT
7.110.1. driver
7.111. HOSTDEVICEPASSTHROUGH STRUCT
7.112. HOSTNIC STRUCT
7.112.1. ad_aggregator_id
7.112.2. bridged
7.112.3. statistics
7.112.4. network
7.113. HOSTNICVIRTUALFUNCTIONSCONFIGURATION STRUCT
7.113.1. max_number_of_virtual_functions
7.113.2. number_of_virtual_functions
7.114. HOSTPROTOCOL ENUM
7.115. HOSTSTATUS ENUM
7.115.1. error
7.115.2. initializing
7.115.3. install_failed
7.115.4. installing_os
7.115.5. maintenance
7.115.6. non_operational
7.115.7. non_responsive
7.115.8. pending_approval
7.115.9. preparing_for_maintenance
7.116. HOSTSTORAGE STRUCT
7.116.1. nfs_retrans
7.116.2. nfs_timeo
7.117. HOSTTYPE ENUM
7.117.1. ovirt_node
7.118. HOSTEDENGINE STRUCT
7.119. ICON STRUCT
7.119.1. media_type
7.120. IDENTIFIED STRUCT
7.121. IMAGE STRUCT
7.122. IMAGEFILETYPE ENUM
7.122.1. iso
7.123. IMAGETRANSFER STRUCT
7.123.1. direction
7.123.2. inactivity_timeout
7.123.3. phase
7.123.4. proxy_url
7.123.5. transfer_url
7.123.6. host
7.123.7. image
7.124. IMAGETRANSFERDIRECTION ENUM
7.125. IMAGETRANSFERPHASE ENUM
7.125.1. cancelled
7.125.2. finalizing_success
7.125.3. finished_failure
7.143.2. discard_zeros_data
7.144. LUNSTATUS ENUM
7.145. MAC STRUCT
7.146. MACPOOL STRUCT
  7.146.1. allow_duplicates
  7.146.2. default_pool
  7.146.3. ranges
7.147. MEMORYOVERCOMMIT STRUCT
7.148. MEMORYPOLICY STRUCT
  7.148.1. guaranteed
  7.148.2. max
7.149. MESSAGEBROKERTYPE ENUM
7.150. METHOD STRUCT
7.151. MIGRATEONERROR ENUM
7.152. MIGRATIONBANDWIDTH STRUCT
  7.152.1. custom_value
7.153. MIGRATIONBANDWIDTHASSIGNMENTMETHOD ENUM
  7.153.1. auto
7.154. MIGRATIONOPTIONS STRUCT
7.155. MIGRATIONPOLICY STRUCT
7.156. NETWORK STRUCT
  7.156.1. required
  7.156.2. status
  7.156.3. usages
  7.156.4. cluster
  7.156.5. external_provider
  7.156.6. external_provider_physical_network
7.157. NETWORKATTACHMENT STRUCT
  7.157.1. dns_resolver_configuration
  7.157.2. properties
7.158. NETWORKCONFIGURATION STRUCT
7.159. NETWORKFILTER STRUCT
  7.159.1. version
7.160. NETWORKFILTERPARAMETER STRUCT
7.161. NETWORKLABEL STRUCT
7.162. NETWORKPLUGINTYPE ENUM
  7.162.1. open_vswitch
7.163. NETWORKSTATUS ENUM
7.164. NETWORKUSAGE ENUM
  7.164.1. default_route
  7.164.2. management
7.165. NFSPROFILEDETAIL STRUCT
7.166. NFSVERSION ENUM
  7.166.1. v4_2
7.167. NIC STRUCT
  7.167.1. network
  7.167.2. vms
7.168. NICCONFIGURATION STRUCT
7.169. NICINTERFACE ENUM
7.170. NICSTATUS ENUM
7.171. NUMANODE STRUCT
  7.171.1. statistics
7.172. NUMANODEPIN STRUCT
7.198.1. agents
7.198.2. automatic_pm_enabled
7.198.3. kdump_detection
7.198.4. type
7.199. POWERMANAGEMENTSTATUS ENUM
7.200. PRODUCT STRUCT
7.201. PRODUCTINFO STRUCT
  7.201.1. vendor
7.202. PROFILEDETAIL STRUCT
7.203. PROPERTY STRUCT
7.204. PROXYTICKET STRUCT
7.205. QCOWVERSION ENUM
  7.205.1. qcow2_v3
7.206. QOS STRUCT
  7.206.1. cpu_limit
  7.206.2. inbound_average
  7.206.3. inbound_burst
  7.206.4. inbound_peak
  7.206.5. max_iops
  7.206.6. max_read_iops
  7.206.7. max_read_throughput
  7.206.8. max_throughput
  7.206.9. max_write_iops
  7.206.10. max_write_throughput
  7.206.11. outbound_average
  7.206.12. outbound_average_linkshare
  7.206.13. outbound_average_realtime
  7.206.14. outbound_average_upperlimit
  7.206.15. outbound_burst
  7.206.16. outbound_peak
7.207. QOSTYPE ENUM
7.208. QUOTA STRUCT
7.209. QUOTACLUSTERLIMIT STRUCT
7.210. QUOTAMODETYPE ENUM
7.211. QUOTASTORAGELIMIT STRUCT
7.212. RANGE STRUCT
7.213. RATE STRUCT
7.214. REGISTRATIONAFFINITYGROUPMAPPING STRUCT
  7.214.1. from
7.215. REGISTRATIONAFFINITYLABELMAPPING STRUCT
  7.215.1. from
7.216. REGISTRATIONCLUSTERMAPPING STRUCT
  7.216.1. from
  7.216.2. to
7.217. REGISTRATIONCONFIGURATION STRUCT
7.218. REGISTRATIONDOMAINMAPPING STRUCT
  7.218.1. from
7.219. REGISTRATIONLUNMAPPING STRUCT
  7.219.1. from
7.220. REGISTRATIONROLEMAPPING STRUCT
  7.220.1. from
7.221. REGISTRATIONVNICPROFILEMAPPING STRUCT
  7.221.1. from
7.221.2. to
7.222. REPORTEDCONFIGURATION STRUCT
7.223. REPORTEDDEVICE STRUCT
7.224. REPORTEDDEVICETYPE ENUM
7.225. RESOLUTIONTYPE ENUM
7.226. RNGDEVICE STRUCT
7.227. RNGSOURCE ENUM
   7.227.1. urandom
7.228. ROLE STRUCT
   7.228.1. mutable
7.229. ROLETYPE ENUM
7.230. SCHEDULINGPOLICY STRUCT
7.231. SCHEDULINGPOLICYUNIT STRUCT
7.232. SCSIGENERICIO ENUM
7.233. SELINUX STRUCT
7.234. SELINUXMODE ENUM
7.235. SERIALNUMBER STRUCT
7.236. SERIALNUMBERPOLICY ENUM
7.237. SESSION STRUCT
   7.237.1. console_user
   7.237.2. ip
   7.237.3. protocol
   7.237.4. user
7.238. SKIPIFCONNECTIVITYBROKEN STRUCT
   7.238.1. enabled
   7.238.2. threshold
7.239. SKIPIFSDACTIVE STRUCT
   7.239.1. enabled
7.240. SNAPSHOT STRUCT
   7.240.1. cpu
   7.240.2. custom_compatibility_version
   7.240.3. high_availability
   7.240.4. initialization
   7.240.5. large_icon
   7.240.6. lease
   7.240.7. memory
   7.240.8. migration_downtime
   7.240.9. next_run_configuration_exists
   7.240.10. origin
   7.240.11. persist_memorystate
   7.240.12. placement_policy
   7.240.13. small_icon
   7.240.14. sso
   7.240.15. stop_reason
   7.240.16. affinity_labels
   7.240.17. katello_errata
   7.240.18. original_template
   7.240.19. statistics
7.241. SNAPSHOTSTATUS ENUM
   7.241.1. locked
7.242. SNAPSHOTTYPE ENUM
   7.242.1. preview
   7.242.2. stateless
7.243. SPECIALOBJECTS STRUCT
7.244. SPM STRUCT
7.245. SPMSTATUS ENUM
7.246. SSH STRUCT
7.247. SSAUTHENTICATIONMETHOD ENUM
7.248. SSHPUBLICKEY STRUCT
7.249. SSO STRUCT
7.250. SSMETHOD ENUM
7.251. STATISTIC STRUCT
7.252. STATISTICKIND ENUM
7.253. STATISTICUNIT ENUM
7.254. STEP STRUCT
  7.254.1. external
7.255. STEPENUM ENUM
  7.255.1. executing
  7.255.2. finalizing
  7.255.3. rebalancing_volume
  7.255.4. removing_bricks
  7.255.5. unknown
  7.255.6. validating
7.256. STEPSTATUS ENUM
  7.256.1. aborted
  7.256.2. finished
  7.256.3. started
  7.256.4. unknown
7.257. STORAGECONNECTION STRUCT
7.258. STORAGECONNECTIONEXTENSION STRUCT
7.259. STORAGEDOMAIN STRUCT
  7.259.1. backup
  7.259.2. discard_after_delete
  7.259.3. supports_discard
  7.259.4. supports_discard_zeroes_data
  7.259.5. wipe_after_delete
  7.259.6. data_center
7.260. STORAGEDOMAINLEASE STRUCT
7.261. STORAGEDOMAINSTATUS ENUM
7.262. STORAGEDOMAINTYPE ENUM
  7.262.1. data
  7.262.2. export
  7.262.3. image
  7.262.4. iso
  7.262.5. volume
7.263. STORAGEFORMAT ENUM
  7.263.1. v1
  7.263.2. v2
  7.263.3. v3
7.264. STORAGETYPE ENUM
  7.264.1. cinder
  7.264.2. glance
  7.264.3. glusterfs
7.265. SWITCHTYPE ENUM
7.266. SYSTEMOPTION STRUCT
7.267. SYSTEMOPTIONVALUE STRUCT
7.268. TAG STRUCT
7.269. TEMPLATE STRUCT
  7.269.1. cpu
  7.269.2. custom_compatibility_version
  7.269.3. high_availability
  7.269.4. initialization
  7.269.5. large_icon
  7.269.6. lease
  7.269.7. memory
  7.269.8. migration_downtime
  7.269.9. origin
  7.269.10. placement_policy
  7.269.11. small_icon
  7.269.12. sso
7.270. TEMPLATESTATUS ENUM
7.271. TEMPLATEVERSION STRUCT
  7.271.1. version_number
7.272. TICKET STRUCT
7.273. TIMEZONE STRUCT
  7.273.1. utc_offset
7.274. TRANSPARENTHUGE_PAGES STRUCT
7.275. TRANSPORTTYPE ENUM
7.276. UNMANAGEDNETWORK STRUCT
7.277. USB STRUCT
7.278. USBTYPE ENUM
  7.278.1. legacy
  7.278.2. native
7.279. USER STRUCT
  7.279.1. namespace
  7.279.2. principal
  7.279.3. user_name
7.280. VALUE STRUCT
7.281. VALUETYPE ENUM
7.282. VCPUPIN STRUCT
7.283. VENDOR STRUCT
7.284. VERSION STRUCT
7.285. VIRTIOSCSI STRUCT
7.286. VIRTUALNUMANODE STRUCT
  7.286.1. statistics
7.287. VLAN STRUCT
7.288. VM STRUCT
  7.288.1. cpu
  7.288.2. custom_compatibility_version
  7.288.3. high_availability
  7.288.4. initialization
  7.288.5. large_icon
  7.288.6. lease
  7.288.7. memory
  7.288.8. migration_downtime
  7.288.9. next_run_configuration_exists
  7.288.10. origin
  7.288.11. placement_policy
  7.288.12. small_icon
7.288.13. sso
7.288.14. stop_reason
7.288.15. affinity_labels
7.288.16. katello_errata
7.288.17. original_template
7.288.18. statistics

7.289. VMAFFINITY ENUM

7.290. VMBASE STRUCT
7.290.1. cpu
7.290.2. custom_compatibility_version
7.290.3. high_availability
7.290.4. initialization
7.290.5. large_icon
7.290.6. lease
7.290.7. memory
7.290.8. migration_downtime
7.290.9. origin
7.290.10. placement_policy
7.290.11. small_icon
7.290.12. sso

7.291. VMDEVICETYPE ENUM

7.292. VMPLACEMENTPOLICY STRUCT

7.293. VMPOOL STRUCT
7.293.1. auto_storage_select
7.293.2. prestarted_vms
7.293.3. stateful
7.293.4. instance_type
7.293.5. vm

7.294. VMPOOLTYPE ENUM
7.294.1. automatic
7.294.2. manual

7.295. VMSTATUS ENUM
7.295.1. paused
7.295.2. powering_up
7.295.3. restoring_state
7.295.4. saving_state
7.295.5. suspended
7.295.6. unknown
7.295.7. up
7.295.8. wait_for_launch

7.296. VMSTORAGEERRORRESUMEBEHAVIOUR ENUM
7.296.1. auto_resume
7.296.2. kill
7.296.3. leave_paused

7.297. VMSUMMARY STRUCT

7.298. VMTYPE ENUM
7.298.1. desktop
7.298.2. high_performance
7.298.3. server

7.299. VNCPASSTHROUGH STRUCT

7.300. VNCPASSTHROUGHMODE ENUM

7.301. VNICPROFILE STRUCT
7.301.1. migratable
7.301.2. pass_through 770
7.301.3. port_mirroring 770
7.301.4. network_filter 771
7.301.5. qos 771
7.302. VNICPROFILEMAPPING STRUCT 771
  7.302.1. source_network_name 772
  7.302.2. source_network_profile_name 773
  7.302.3. target_vnic_profile 773
7.303. VOLUMEGROUP STRUCT 773
7.304. WATCHDOG STRUCT 773
  7.304.1. model 774
  7.304.2. vms 774
7.305. WATCHDOGACTION ENUM 774
  7.305.1. none 775
7.306. WATCHDOGMODEL ENUM 775
  7.306.1. diag288 775
  7.306.2. i6300esb 775
7.307. WEIGHT STRUCT 775

APPENDIX A. PRIMITIVE TYPES .......................................................... 777.
   A.1. STRING PRIMITIVE 777
   A.2. BOOLEAN PRIMITIVE 777
   A.3. INTEGER PRIMITIVE 777
   A.4. DECIMAL PRIMITIVE 778
   A.5. DATE PRIMITIVE 778

APPENDIX B. CHANGES IN VERSION 4 OF THE API .................................. 780.
   B.1. REMOVED YAML SUPPORT 780
   B.2. RENAMED COMPLEX TYPES 780
   B.3. REPLACED THE STATUS TYPE WITH ENUM TYPES 781
   B.4. REMOVE THE NIC NETWORK AND PORT_MIRRORING PROPERTIES 782
   B.5. REMOVE THE NIC ACTIVE PROPERTY 783
   B.6. REMOVE THE DISK TYPE PROPERTY 783
   B.7. REMOVE THE DISK SIZE PROPERTY 783
   B.8. REMOVED SUPPORT FOR PINNING A VM TO A SINGLE HOST 783
   B.9. REMOVED THE CAPABILITIES.PERMITS ELEMENT 783
   B.10. REMOVED THE STORAGE_MANAGER ELEMENT 784
   B.11. REMOVED THE DATA CENTER STORAGE_TYPE ELEMENT 784
   B.12. REMOVE THE TIMEZONE ELEMENT 784
   B.13. REMOVED THE GUEST_INFO ELEMENT 785
   B.14. REPLACED CPU ID ATTRIBUTE WITH TYPE ELEMENT 786
   B.15. USE ELEMENTS INSTEAD OF ATTRIBUTES IN CPU TOPOLOGY 786
   B.16. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VCPU PIN 786
   B.17. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VCPU PIN 787
   B.18. USE ELEMENTS INSTEAD OF ATTRIBUTES IN MEMORY OVERCOMMIT 787
   B.19. USE ELEMENTS INSTEAD OF ATTRIBUTES IN CONSOLE 787
   B.20. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VIRTIO SCSI 788
   B.21. USE ELEMENT INSTEAD OF ATTRIBUTE FOR POWER MANAGEMENT AGENT TYPE 788
   B.22. USE ELEMENTS INSTEAD OF ATTRIBUTES IN POWER MANAGEMENT AGENT OPTIONS 788
   B.23. USE ELEMENTS INSTEAD OF ATTRIBUTES IN IP ADDRESS: 789
   B.24. USE ELEMENTS INSTEAD OF ATTRIBUTES IN MAC ADDRESS: 789
   B.25. USE ELEMENTS INSTEAD OF ATTRIBUTES IN BOOT DEVICE: 789
   B.26. USE ELEMENT INSTEAD OF ATTRIBUTE FOR OPERATING SYSTEM TYPE 789
B.27. REMOVED THE FORCE PARAMETER FROM THE REQUEST TO RETRIEVE A HOST
B.28. REMOVED DEPRECATED HOST POWER MANAGEMENT CONFIGURATION
B.29. USE MULTIPLE BOOT.DEVICES.DEVICE INSTEAD OF MULTIPLE BOOT
B.30. REMOVED THE DISKS.CLONE AND DISKS.DETACH_ONLY ELEMENTS
B.31. RENAME ELEMENT VMPOOL TO VM_POOL
B.32. USE LOGICAL_UNITS INSTEAD OF MULTIPLE LOGICAL_UNIT
B.33. REMOVED THE SNAPSHOT.COLLAPSE_SNAPSHOTS ELEMENT
B.34. RENAMED STORAGE AND HOST_STORAGE ELEMENTS
B.35. REMOVED THE PERMISSIONS.CLONE ELEMENT
B.36. RENAMED THE RANDOM NUMBER GENERATOR SOURCE ELEMENTS
B.37. REMOVED THE INTERMEDIATE TAG.PARENT ELEMENT
B.38. REMOVE SCHEDULING BUILT-IN NAMES AND THRESHOLDS
B.39. REMOVED THE BRICKS.REPLICA_COUNT AND BRICKS.STRIPE_COUNT ELEMENTS
B.40. RENAMED THE STATISTICS TYPE PROPERTY TO KIND
B.41. USE MULTIPLE VCPU_PINS.VCPU_PIN INSTEAD OF MULTIPLE VCPU_PIN
B.42. USE FORCE PARAMETER TO FORCE REMOVE A DATA CENTER
B.43. USE FORCE PARAMETER TO FORCE REMOVE A HOST
B.44. USE PARAMETERS FOR FORCE REMOVE STORAGE DOMAIN
B.45. USE HOST PARAMETER TO REMOVE STORAGE SERVER CONNECTION
B.46. USE FORCE AND STORAGE_DOMAIN PARAMETERS TO REMOVE TEMPLATE DISKS
B.47. DON'T REMOVE DISKS VIA THE VM DISK API
B.48. USE FORCE QUERY PARAMETER TO FORCE REMOVE A VIRTUAL MACHINE
B.49. USE POST INSTEAD OF DELETE TO REMOVE MULTIPLE BRICKS
B.50. REMOVED THE SCHEDULING_POLICY.POLICY ELEMENT
B.51. ADDED SNAPSHOT.SNAPSHOT_TYPE
B.52. REMOVED MOVE ACTION FROM VM
B.53. MOVED REPORTED_CONFIGURATIONS.IN_SYNC TO NETWORK_ATTACHMENT
B.54. REPLACED CAPABILITIES WITH CLUSTERLEVELS
B.55. REPLACED DISKS WITH DISKATTACHMENTS
B.56. USE ISCSI_TARGETS ELEMENT TO DISCOVER UNREGISTERED STORAGE
CHAPTER 1. INTRODUCTION

The Red Hat Virtualization Manager provides a **Representational State Transfer (REST) API**. The API provides software developers and system administrators with control over their Red Hat Virtualization environment outside of the standard web interface. The API is useful for developers and administrators to integrate the functionality of a Red Hat Virtualization environment with custom scripts or external applications that access the API via the standard Hypertext Transfer Protocol (HTTP).

The benefits of the API are:

- Broad client support - Any programming language, framework, or system with support for HTTP protocol can use the API.
- Self descriptive - Client applications require minimal knowledge of the virtualization infrastructure, as many details are discovered at runtime.
- Resource-based model - The resource-based REST model provides a natural way to manage a virtualization platform.

This provides developers and administrators with the ability to:

- Integrate with enterprise IT systems.
- Integrate with third-party virtualization software.
- Perform automated maintenance or error-checking tasks.
- Automate repetitive tasks in a Red Hat Virtualization environment with scripts.

This documentation acts as a reference for the Red Hat Virtualization API. It aims to provide developers and administrators with instructions and examples to help harness the functionality of their Red Hat Virtualization environment through the API, either directly or using the provided SDKs.

1.1. REPRESENTATIONAL STATE TRANSFER

**Representational State Transfer (REST)** is a design architecture that focuses on resources for a specific service and their representations. A resource representation is a key abstraction of information that corresponds to one specific managed element on a server. A client sends a request to a server element located at a Uniform Resource Identifier (URI) and performs operations with standard HTTP methods, such as **GET**, **POST**, **PUT**, and **DELETE**. This provides a stateless communication between the client and server where each request acts independently of any other request, and contains all the information necessary to complete the request.

1.2. API PREREQUISITES

Prerequisites for using the Red Hat Virtualization API:

- A networked installation of Red Hat Virtualization Manager, which includes the API.
- A client or programming library that initiates and receives HTTP requests from the API server. For example:
  - The oVirt Python SDK.
  - The oVirt Ruby SDK.
- The oVirt Java SDK.
- The cURL command line tool.
- RESTClient, a debugger for RESTful web services.


- Knowledge of Extensible Markup Language (XML) or JavaScript Object Notation (JSON), which the API uses to construct resource representations. The W3C provides a full specification on XML at http://www.w3.org/TR/xml. ECMA International provide a free publication on JSON at http://www.ecma-international.org.
2.1. TLS/SSL CERTIFICATION

The Red Hat Virtualization API requires Hypertext Transfer Protocol Secure (HTTPS) for secure interaction with client software, such as the SDK and CLI components. This involves obtaining the CA certificate used by the server, and importing it into the certificate store of your client.

2.1.1. Obtaining the CA Certificate

You can obtain the CA certificate from the Red Hat Virtualization Manager and transfer it to the client machine using one of these methods:

Method 1

The preferred method for obtaining the CA certificate is to use the `openssl s_client` command line tool to perform a real TLS handshake with the server, and then extract the certificates that it presents. Run a command like this:

```
$ openssl s_client \
   -connect myengine.example.com:443 \
   -showcerts \
   < /dev/null
```

This command will connect to the server and display output similar to the following:

```
CONNECTED(00000003) depth=1 C = US, O = Example Inc., CN = myengine.example.com.23416
verify error:num=19:self signed certificate in certificate chain
---
Certificate chain
  0 s:/C=US/O=Example Inc./CN=myengine.example.com
    i:/C=US/O=Example Inc./CN=myengine.example.com.23416
    -----BEGIN CERTIFICATE-----
MIIEaTCCA1GgAwIBAgICEAQwDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lNDEuZXhhbXBsZS5jMCEGCSqGSIb3DQEBBQUAMHggIwIBAgYIKoZIhvcNAQcGBQEMBQAwCTEAMG0GCSqGSIb3DQEBBQUAAMIGCMGCQwEQQDHAESBQIL
    -----END CERTIFICATE-----

  1 s:/C=US/O=Example Inc./CN=myengine.example.com.23416
    i:/C=US/O=Example Inc./CN=myengine.example.com.23416
    -----BEGIN CERTIFICATE-----
MIITdDCCAsAwAwIBAgIBATANBgkqhkiG9w0BAQsFAAOCAQEECDqA3vWzFrbmFzcXJ5b3QDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lNDEuZXhhbXBsZS5jMCEGCSqGSIb3DQEBBQUAMHggIwIBAgYIKoZIhvcNAQcGBQEMBQAwCTEAMG0GCSqGSIb3DQEBBQUAAMIGCMGCQwEQQDHAESBQIL
    -----END CERTIFICATE-----

The text between the `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` marks shows the certificates presented by the server. The first one is the certificate of the server itself, and the last one is the certificate of the CA. Copy the CA certificate, including the marks, to the `ca.crt` file. The result should look like this:

```
-----BEGIN CERTIFICATE-----
MIIEaTCCA1GgAwIBAgICEAQwDQYJKoZIhvcNAQEFBQAwSTELMAkGA1UEBhMCVVMxFTATBgNVBAoTDEV4YW1wbGUgSW5jLjEjMCEGA1UEAxMaZW5naW5lNDEuZXhhbXBsZS5jMCEGCSqGSIb3DQEBBQUAMHggIwIBAgYIKoZIhvcNAQcGBQEMBQAwCTEAMG0GCSqGSIb3DQEBBQUAAMIGCMGCQwEQQDHAESBQIL
-----END CERTIFICATE-----
```
IMPORTANT

This is the most reliable method to obtain the CA certificate used by the server. The rest of the methods described here will work in most cases, but they will not obtain the correct CA certificate if it has been manually replaced by the administrator of the server.

**Method 2**

If you cannot use the `openssl s_client` method described above, you can instead use a command line tool to download the CA certificate from the Red Hat Virtualization Manager. Examples of command line tools include `curl` and `wget`, both of which are available on multiple platforms.

If using `curl`:

```bash
$ curl \
   --output ca.crt \
   'http://myengine.example.com/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA'
```

If using `wget`:

```bash
$ wget \
   --output-document ca.crt \
   'http://myengine.example.com/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA'
```

**Method 3**

Use a web browser to navigate to the certificate located at:

```bash
given_url
```

Depending on the chosen browser, the certificate either downloads or imports into the browser’s keystore.

1. **If the browser downloads the certificate**: save the file as `ca.crt`.
2. **If the browser imports the certificate**: export it from the browser’s certification options and save it as `ca.crt`.

**Method 4**

Log in to the Red Hat Virtualization Manager, export the certificate from the truststore, and copy it to your client machine.

1. Log in to the Red Hat Virtualization Manager machine as `root`.
2. Export the certificate from the truststore using the Java `keytool` management utility:
This creates a certificate file called `ca.crt`.

3. Copy the certificate to the client machine using the `scp` command:

   ```
   $ scp ca.crt myuser@myclient.example.com:/home/myuser/.
   ```

Each of these methods results in a certificate file named `ca.crt` on your client machine. You must then import this file into the certificate store of the client.

### 2.1.2. Importing a Certificate to a Client

Importing a certificate to a client relies on how the client stores and interprets certificates. See your client documentation for more information on importing a certificate.

### 2.2. AUTHENTICATION

Any user with a Red Hat Virtualization Manager account has access to the API. All requests must be authenticated using either OAuth or basic authentication, as described below.

#### 2.2.1. OAuth Authentication

Since version 4.0 of Red Hat Virtualization the preferred authentication mechanism is OAuth 2.0, as described in RFC 6749.

OAuth is a sophisticated protocol, with several mechanisms for obtaining authorization and access tokens. For use with the Red Hat Virtualization API, the only supported one is the Resource Owner Password Credentials Grant, as described in section 4.3 of RFC 6749.

You must first obtain a token, sending the user name and password to the Red Hat Virtualization Manager single sign-on service:

```
POST /ovirt-engine/sso/oauth/token HTTP/1.1
Host: myengine.example.com
Content-Type: application/x-www-form-urlencoded
Accept: application/json
```

The request body must contain the `grant_type`, `scope`, `username`, and `password` parameters:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>grant_type</td>
<td>password</td>
</tr>
</tbody>
</table>
These parameters must be URL-encoded. For example, the @ character in the user name needs to be encoded as %40. The resulting request body will be something like this:

```
grant_type=password&scope=ovirt-app-api&username=admin%40internal&password=mypassword
```

**IMPORTANT**

The `scope` parameter is described as optional in the OAuth RFC, but when using it with the Red Hat Virtualization API it is mandatory, and its value must be `ovirt-app-api`.

If the user name and password are valid, the Red Hat Virtualization Manager single sign-on service will respond with a JSON document similar to this one:

```
{
  "access_token": "fqbR1ftzh8wBCviLxJcYuV5oSDI=",
  "token_type": "bearer",
  "scope": "...",
  ...
}
```

For API authentication purposes, the only relevant name/value pair is the `access_token`. Do not manipulate this in any way; use it exactly as provided by the SSO service.

Once the token has been obtained, it can be used to perform requests to the API by including it in the HTTP `Authorization` header, and using the `Bearer` scheme. For example, to get the list of virtual machines, send a request like this:

```
GET /ovirt-engine/api/vms HTTP/1.1
Host: myengine.example.com
Accept: application/xml
Authorization: Bearer fqbR1ftzh8wBCviLxJcYuV5oSDI=
```

The token can be used multiple times, for multiple requests, but it will eventually expire. When it expires, the server will reject the request with the 401 HTTP response code:

```
HTTP/1.1 401 Unauthorized
```

When this happens, a new token is needed, as the Red Hat Virtualization Manager single sign-on service does not currently support refreshing tokens. A new token can be requested using the same method described above.
2.2.2. Basic Authentication

IMPORTANT

Basic authentication is supported only for backwards compatibility; it is deprecated since version 4.0 of Red Hat Virtualization, and will be removed in the future.

Each request uses HTTP Basic Authentication[^2] to encode the credentials. If a request does not include an appropriate **Authorization** header, the server sends a **401 Authorization Required** response:

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
HTTP/1.1 401 Authorization Required
```

Request are issued with an **Authorization** header for the specified realm. Encode an appropriate Red Hat Virtualization Manager domain and user in the supplied credentials with the `username@domain:password` convention.

The following table shows the process for encoding credentials in **Base64**.

<table>
<thead>
<tr>
<th>Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>User name</td>
<td>admin</td>
</tr>
<tr>
<td>Domain</td>
<td>internal</td>
</tr>
<tr>
<td>Password</td>
<td>mypassword</td>
</tr>
<tr>
<td>Unencoded credentials</td>
<td>admin@internal:mypassword</td>
</tr>
<tr>
<td>Base64 encoded credentials</td>
<td>YWRtaW5AaW50ZXJuYww6bXlwYXNzd29yZA==</td>
</tr>
</tbody>
</table>

Provide the Base64-encoded credentials as shown:

```
HEAD /ovirt-engine/api HTTP/1.1
Host: myengine.example.com
Authorization: Basic YWRtaW5AaW50ZXJuYww6bXlwYXNzd29yZA==
HTTP/1.1 200 OK
```

IMPORTANT

Basic authentication involves potentially sensitive information, such as passwords, sent as plain text. The API requires Hypertext Transfer Protocol Secure (HTTPS) for transport-level encryption of plain-text requests.
2.2.3. Authentication Sessions

The API also provides authentication session support. Send an initial request with authentication details, then send all subsequent requests using a session cookie to authenticate.

2.2.3.1. Requesting an Authenticated Session

1. Send a request with the Authorization and Prefer: persistent-auth headers:

   ```
   HEAD /ovirt-engine/api HTTP/1.1
   Host: myengine.example.com
   Authorization: Basic YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==
   Prefer: persistent-auth
   
   HTTP/1.1 200 OK
   ...
   ```

   This returns a response with the following header:

   ```
   Set-Cookie: JSESSIONID=5dQja5ubr4yvI2MM2z+LZxrK; Path=/ovirt-engine/api; Secure
   ```

   Take note of the JSESSIONID= value. In this example the value is 5dQja5ubr4yvI2MM2z+LZxrK.

2. Send all subsequent requests with the Prefer: persistent-auth and Cookie headers with the JSESSIONID= value. The Authorization header is no longer needed when using an authenticated session.

   ```
   HEAD /ovirt-engine/api HTTP/1.1
   Host: myengine.example.com
   Prefer: persistent-auth
   Cookie: JSESSIONID=5dQja5ubr4yvI2MM2z+LZxrK
   
   HTTP/1.1 200 OK
   ...
   ```

3. When the session is no longer required, perform a request to the server without the Prefer: persistent-auth header.

   ```
   HEAD /ovirt-engine/api HTTP/1.1
   Host: myengine.example.com
   Authorization: Basic YWRtaW5AaW50ZXJuYWw6bXlwYXNzd29yZA==
   
   HTTP/1.1 200 OK
   ```
HTTPS is described in RFC 2818 HTTP Over TLS.

Basic Authentication is described in RFC 2617 HTTP Authentication: Basic and Digest Access Authentication.
CHAPTER 3. COMMON CONCEPTS

3.1. TYPES

The API uses the type concept to describe the different kinds of objects accepted and returned.

There are three relevant kinds of types:

**Primitive types**
Describe simple kinds of objects, like strings or integers.

**Enumerated types**
Describe lists of valid values like VmStatus or DiskFormat.

**Structured types**
Describe structured objects, with multiple attributes and links, like Vm or Disk.

3.2. IDENTIFIED TYPES

Many of the types used by the API represent identified objects, objects that have an unique identifier and exist independently of other objects. The types used to describe those objects extend the Identified type, which contains the following set of common attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>Each object in the virtualization infrastructure contains an id, which acts as an unique identifier.</td>
</tr>
<tr>
<td>href</td>
<td>String</td>
<td>The canonical location of the object as an absolute path.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A user-supplied human readable name for the object. The name is unique across all objects of the same type.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A free-form user-supplied human readable description of the object.</td>
</tr>
</tbody>
</table>

**IMPORTANT**
Currently for most types of objects the id attribute is actually a randomly generated UUID, but this is an implementation detail, and users should not rely on that, as it may change in the future. Instead users should assume that these identifiers are just strings.

3.3. OBJECTS

Objects are the individual instances of the types supported by the API. For example, the virtual machine with identifier 123 is an object of the Vm type.

3.4. COLLECTIONS

A collection is a set of objects of the same type.
3.5. REPRESENTATIONS

The state of objects needs to be represented when it is transferred between the client and the server. The API supports XML and JSON as the representation of the state of objects, both for input and output.

3.5.1. XML representation

The XML representation of an object consists of an XML element corresponding to the type of the object, XML attributes for the `id` and `href` attributes, and nested XML elements for the rest of the attributes. For example, the XML representation for a virtual machine appears as follows:

```xml
<vm id="123" href="/ovirt-engine/api/vms/123">
  <name>myvm</name>
  <description>My VM</description>
  <memory>1073741824</memory>
  ...
</vm>
```

The XML representation of a collection of objects consists of an XML element, named after the type of the objects, in plural. This contains the representations of the objects of the collection. For example, the XML representation for a collection of virtual machines appears as follows:

```xml
<vms>
  <vm id="123" href="/ovirt-engine/api/vms/123">
    <name>myvm</name>
    <description>My VM</description>
    <memory>1073741824</memory>
    ...
  </vm>
  <vm id="456" href="/ovirt-engine/api/vms/456">
    <name>myname</name>
    <description>My description</description>
    <memory>2147483648</memory>
    ...
  </vm>
  ...
</vms>
```

**IMPORTANT**

In the XML representation of objects the `id` and `href` attributes are the only ones that are represented as XML attributes, the rest are represented as nested XML elements.

3.5.2. JSON representation

The JSON representation of an object consists of a JSON document containing a name/value pair for each attribute (including `id` and `href`). For example, the JSON representation of a virtual machine appears as follows:

```json
{
  "id": "123",
  "href": "/ovirt-engine/api/vms/123",
  "name": "myvm",
  "description": "My VM",
  ...
}
```
The JSON representation of a collection of objects consists of a JSON document containing a name/value pair (named after the type of the objects, in singular) which in turn contains an array with the representations of the objects of the collection. For example, the JSON representation for a collection of virtual machines appears as follows:

```json
{
    "vm": [
    {
        "id": "123",
        "href": "/ovirt-engine/api/vms/123",
        "name": "myvm",
        "description": "My VM",
        "memory": 1073741824,
        ...
    },
    {
        "id": "456",
        "href": "/ovirt-engine/api/vms/456",
        "name": "yourvm",
        "description": "Your VM",
        "memory": 2147483648,
        ...
    }
    ]
}
```

### 3.6. SERVICES

Services are the parts of the server responsible for retrieving, adding, updating, removing and executing actions on the objects supported by the API.

There are two relevant kinds of services:

**Services that manage a collection of objects**

These services are responsible for listing existing objects and adding new objects. For example, the Vms service is responsible for managing the collection of virtual machines available in the system.

**Services that manage a specific object**

These services are responsible for retrieving, updating, deleting and executing actions in specific objects. For example, the Vm service is responsible for managing a specific virtual machine.

Each service is accessible via a particular path within the server. For example, the service that manages the collection of virtual machines available in the system is available in the via the path `/vms`, and the service that manages the virtual machine 123 is available via the path `/vms/123`.

All kinds of services have a set of methods that represent the operations that they can perform. The services that manage collections of objects usually have the list and add methods. The services that manage specific objects usually have the get, update and remove methods. In addition, services may also have action methods, that represent less common operations. For example, the Vm service has a start method that is used to start a virtual machine.
For the more usual methods there is a direct mapping between the name of the method and the name of
the HTTP method:

<table>
<thead>
<tr>
<th>Method name</th>
<th>HTTP method</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>POST</td>
</tr>
<tr>
<td>get</td>
<td>GET</td>
</tr>
<tr>
<td>list</td>
<td>GET</td>
</tr>
<tr>
<td>update</td>
<td>PUT</td>
</tr>
<tr>
<td>remove</td>
<td>DELETE</td>
</tr>
</tbody>
</table>

The path used in the HTTP request is the path of the service, with the /ovirt-engine/api prefix.

For example, the request to list the virtual machines should be like this, using the HTTP GET method
and the path /vms:

```
GET /ovirt-engine/api/vms
```

For action methods the HTTP method is always POST, and the name of the method is added as a suffix
to the path. For example, the request to start virtual machine 123 should look like this, using the HTTP POST method and the path /vms/123/start:

```
POST /ovirt-engine/api/vms/123/start
```

Each method has a set of parameters.

Parameters are classified into two categories:

**Main parameter**

The main parameter corresponds the object or collection that is retrieved, added or updated. This
only applies to the add, get, list and update methods, and there will be exactly one such main
parameter per method.

**Secondary parameters**

The rest of the parameters.

For example, the operation that adds a virtual machine (see here) has three parameters: vm, clone and
clone_permissions. The main parameter is vm, as it describes the object that is added. The clone
and clone_permissions parameters are secondary parameters.

The main parameter, when used for input, must be included in the body of the HTTP request. For
example, when adding a virtual machine, the vm parameter, of type Vm, must be included in the request
body. So the complete request to add a virtual machine, including all the HTTP details, must look like
this:

```
POST /ovirt-engine/api/vms HTTP/1.1
Host: myengine.example.com
```
<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
</vm>

When used for output, the main parameters are included in the response body. For example, when adding a virtual machine, the `vm` parameter will be included in the response body. So the complete response body will look like this:

```
HTTP/1.1 201 Created
Content-Type: application/xml

<vm href="/ovirt-engine/api/vms/123" id="123">
  <name>myvm</name>
  <description>My VM</description>
  ...
</vm>
```

Secondary parameters are only allowed for input (except for action methods, which are described later), and they must be included as query parameters. For example, when adding a virtual machine with the `clone` parameter set to `true`, the complete request must look like this:

```
POST /ovirt-engine/api/vms?clone=true HTTP/1.1
Host: myengine.example.com
Authorization: Bearer fqbR1ftzh8wBCviLxJcYuV5oSDI=
Content-Type: application/xml
Accept: application/xml

<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
</vm>
```

Action methods only have secondary parameters. They can be used for input and output, and they should be included in the request body, wrapped with an `action` element. For example, the action method used to start a virtual machine (see here) has a `vm` parameter to describe how the virtual machine should be started, and a `use_cloud_init` parameter to specify if `cloud-init` should be used to configure the guest operating system. So the complete request to start virtual machine 123 using `cloud-init` will look like this when using XML:

```
```
POST /ovirt-engine/api/vms/123/start HTTP/1.1
Host: myengine.example.com
Authorization: Bearer fgbR1ftzh8wBCviLxJcYuV5oSDI=
Content-Type: application/xml
Accept: application/xml

<action>
  <use_cloud_init>true</use_cloud_init>
  <vm>
    <initialization>
      <nic_configurations>
        <nic_configuration>
          <name>eth0</name>
          <on_boot>true</on_boot>
          <boot_protocol>static</boot_protocol>
          <ip>
            <address>192.168.0.100</address>
            <netmask>255.255.255.0</netmask>
            <gateway>192.168.0.1</gateway>
          </ip>
        </nic_configuration>
      </nic_configurations>
      <dns_servers>192.168.0.1</dns_servers>
    </initialization>
  </vm>
</action>

3.7. SEARCHING

The list method of some services has a search parameter that can be used to specify search criteria. When used, the server will only return objects within the collection that satisfy those criteria. For example, the following request will return only the virtual machine named myvm:

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```

3.7.1. Maximum results parameter

Use the max parameter to limit the number of objects returned. For example, the following request will only return one virtual machine, regardless of how many are available in the system:

```
GET /ovirt-engine/api/vms?max=1
```

A search request without the max parameter will return all the objects. Specifying the max parameter is recommended to reduce the impact of requests in the overall performance of the system.

3.7.2. Case sensitivity

By default queries are not case sensitive. For example, the following request will return the virtual machines named myvm, MyVM and MYVM:

```
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```
The optional **case_sensitive** boolean parameter can be used to change this behaviour. For example, to get exactly the virtual machine named *myhost*, and not *MyHost* or *MYHOST*, send a request like this:

```bash
GET /ovirt-engine/api/vms?search=name%3D=myvm&case_sensitive=true
```

### 3.7.3. Search syntax

The **search** parameters use the same syntax as the Red Hat Virtualization query language:

```
(criteria) [sortby (element) asc|desc]
```

The **sortby** clause is optional and only needed when ordering results.

Example search queries:

<table>
<thead>
<tr>
<th>Collection</th>
<th>Criteria</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hosts</strong></td>
<td>vms.status=up</td>
<td>Returns a list of all hosts running virtual machines that are <strong>up</strong>.</td>
</tr>
<tr>
<td><strong>vms</strong></td>
<td>domain=example.com</td>
<td>Returns a list of all virtual machines running on the specified domain.</td>
</tr>
<tr>
<td><strong>vms</strong></td>
<td>users.name=mary</td>
<td>Returns a list of all virtual machines belonging to users with the user name <strong>mary</strong>.</td>
</tr>
<tr>
<td><strong>events</strong></td>
<td>severity &gt; normal</td>
<td>Returns a list of all events with severity higher than <strong>normal</strong> and sorted by the the value of their <strong>time</strong> attribute.</td>
</tr>
<tr>
<td><strong>events</strong></td>
<td>severity &gt; normal</td>
<td>Returns a list of all events with severity higher than <strong>normal</strong> and sorted by the the value of their <strong>time</strong> attribute in descending order.</td>
</tr>
</tbody>
</table>

The value of the **search** parameter must be **URL-encoded** to translate reserved characters, such as operators and spaces. For example, the equal sign should be encoded as `%3D`:

```bash
GET /ovirt-engine/api/vms?search=name%3Dmyvm
```

### 3.7.4. Wildcards

The asterisk can be used as part of a value, to indicate that any string matches, including the empty string. For example, the following request will return all the virtual machines with names beginning with *myvm*, such as *myvm*, *myvm2*, *myvma* or *myvm-webserver*:

```bash
GET /ovirt-engine/api/vms?search=name%3Dmyvm*
```

### 3.7.5. Pagination
Some Red Hat Virtualization environments contain large collections of objects. Retrieving all of them with one request isn’t practical, and hurts performance. To allow retrieving them page by page the `search` parameter supports an optional `page` clause. This, combined with the `max` parameter, is the basis for paging. For example, to get the first page of virtual machines, with a page size of 10 virtual machines, send request like this:

```
GET /ovirt-engine/api/vms?search=page%201&max=10
```

**NOTE**

The search parameter is URL-encoded, the actual value of the `search` parameter, before encoding, is `page 1`, so this is actually requesting the first page.

Increase the `page` value to retrieve the next page:

```
GET /ovirt-engine/api/vms?search=page%202&max=10
```

The `page` clause can be used in conjunction with other clauses inside the `search` parameter. For example, the following request will return the second page of virtual machines, but sorting by name:

```
GET /ovirt-engine/api/vms?search=sortby%20name%20page%202&max=10
```

**IMPORTANT**

The API is stateless; it is not possible to retain a state between different requests since all requests are independent from each other. As a result, if a status change occurs between your requests, then the page results may be inconsistent.

For example, if you request a specific page from a list of virtual machines, and virtual machines are created or removed before you request the next page, then your results may be missing some of them, or contain duplicates.

### 3.8. FOLLOWING LINKS

The API returns references to related objects as *links*. For example, when a virtual machine is retrieved it contains links to its disk attachments and network interface cards:

```
<vm id="123" href="/ovirt-engine/api/vms/123">
  ...
  <link rel="diskattachments" href="/ovirt-engine/api/vms/123/diskattachments"/>
  <link rel="nics" href="/ovirt-engine/api/vms/123/nics"/>
  ...
</vm>
```

The complete description of those linked objects can be retrieved by sending separate requests:

```
GET /ovirt-engine/api/vms/123/diskattachments
GET /ovirt-engine/api/vms/123/nics
```

However, in some situations it is more convenient for the application using the API to retrieve the linked information in the same request. This is useful, for example, when the additional network round trips
introduce an unacceptable overhead, or when the multiple requests complicate the code of the application in an unacceptable way. For those use cases the API provides a follow parameter that allows the application to retrieve the linked information using only one request.

The value of the follow parameter is a list of strings, separated by commas. Each of those strings is the path of the linked object. For example, to retrieve the disk attachments and the NICs in the example above the request should be like this:

```
GET /ovirt-engine/api/vms/123?follow=disk_attachments,nics
```

That will return an response like this:

```
<vm id="123" href="/ovirt-engine/api/vms/123">
  ...  
  <disk_attachments>
    <disk_attachment id="456" href="/ovirt-engine/api/vms/123/diskattachments/456">
      <active>true</active>
      <bootable>true</bootable>
      <interface>virtio_scsi</interface>
      <pass_discard>false</pass_discard>
      <read_only>false</read_only>
      <uses_scsi_reservation>false</uses_scsi_reservation>
      <disk id="789" href="/ovirt-engine/api/disks/789"/>
    </disk_attachment>
    ...
  </disk_attachments>
  <nics>
    <nic id="234" href="/ovirt-engine/api/vms/123/nics/234">
      <name>eth0</name>
      <interface>virtio</interface>
      <linked>true</linked>
      <mac>
        <address>00:1a:4a:16:01:00</address>
      </mac>
      <plugged>true</plugged>
    </nic>
    ...
  </nics>
  ...
</vm>
```

The path to the linked object can be a single word, as in the previous example, or it can be a sequence of words, separated by dots, to request nested data. For example, the previous example used disk_attachments in order to retrieve the complete description of the disk attachments, but each disk attachment contains a link to the disk, which wasn’t followed. In order to also follow the links to the disks, the following request can be used:

```
POST /ovirt-engine/api/vms/123?follow=disk_attachments.disk
```

That will result in the following response:

```
<vm id="123" href="/ovirt-engine/api/vms/123">
  <disk_attachments>
    <disk_attachment id="456" href="/ovirt-engine/api/disks/789">
      <active>true</active>
      <bootable>true</bootable>
      <interface>virtio_scsi</interface>
      ...  
    </disk_attachment>
  ...
```
The path can be made as deep as needed. For example, to also get the statistics of the disks:

```
POST /ovirt-engine/api/vms/123?follow=disk_attachments.disk.statistics
```

Multiple path elements and multiple paths can be combined. For example, to get the disk attachments and the network interface cards, both with their statistics:

```
POST /ovirt-engine/api/vms/123?
follow=disk_attachments.disk.statistics,nics.statistics
```

**IMPORTANT**

Almost all the operations that retrieve objects support the `follow` parameter, but make sure to explicitly check the reference documentation, as some operations may not support it, or may provide advice on how to use it to get the best performance.

**IMPORTANT**

Using the `follow` parameter moves the overhead from the client side to the server side. When you request additional data, the server must fetch and merge it with the basic data. That consumes CPU and memory in the server side, and will in most cases require additional database queries. That may adversely affect the performance of the server, especially in large scale environments. Make sure to test your application in a realistic environment, and use the `follow` parameter only when justified.

### 3.9. PERMISSIONS

Many of the services that manage a single object provide a reference to a `permissions` service that manages the permissions assigned to that object. Each permission contains links to the user or group,
the role and the object. For example, the permissions assigned to a specific virtual machine can be retrieved sending a request like this:

```
GET /ovirt-engine/api/vms/123/permissions
```

The response body will look like this:

```
<permissions>
  <permission id="456" href="/ovirt-engine/api/vms/123/permissions/456">
    <user id="789" href="/ovirt-engine/api/users/789"/>
    <role id="abc" href="/ovirt-engine/api/roles/abc"/>
    <vm id="123" href="/ovirt-engine/api/vms/123"/>
  </permission>
  ...
</permissions>
```

A permission is added to an object sending a `POST` request with a permission representation to this service. Each new permission requires a role and a user.

### 3.10. HANDLING ERRORS

Some errors require further explanation beyond a standard HTTP status code. For example, the API reports an unsuccessful object state update or action with a `fault` in the response body. The fault contains the `reason` and `detail` attributes. For example, when the server receives a request to create a virtual machine without the mandatory `name` attribute it will respond with the following HTTP response line:

```
HTTP/1.1 400 Bad Request
```

And the following response body:

```
<fault>
  <reason>Incomplete parameters</reason>
  <detail>Vm [name] required for add</detail>
</fault>
```
CHAPTER 4. QUICK START EXAMPLES

The examples in this section show you how to use the REST API to set up a basic Red Hat Virtualization environment and to create a virtual machine. In addition to the standard prerequisites, these examples require the following:

- A networked and configured Red Hat Virtualization installation.
- An ISO file containing the virtual machine operating system you want to install. This chapter uses CentOS 7 for the installation ISO example.

The API examples use curl to demonstrate API requests with a client application. You can use any application that sends HTTP requests.

IMPORTANT

The HTTP request headers in this example omit the Host and Authorization headers. However, these fields are mandatory and require data specific to your installation of Red Hat Virtualization.

The curl examples use admin@internal for the user name, mypassword for the password, /etc/pki/ovirt-engine/ca.pem for the certificate location, and myengine.example.com for the host name. You must replace them with the correct values for your environment.

Red Hat Virtualization generates a unique identifier for the id attribute for each resource. Identifier codes in this example will differ from the identifier codes in your Red Hat Virtualization environment.

In many examples, some attributes of the results returned by the API have been omitted, for brevity. See, for example, the Cluster reference for a complete list of attributes.

4.1. ACCESS API ENTRY POINT

The following request retrieves a representation of the main entry point for version 4 of the API:

GET /ovirt-engine/api HTTP/1.1
Version: 4
Accept: application/xml

The same request, but using the /v4 URL prefix instead of the Version header:

GET /ovirt-engine/api/v4 HTTP/1.1
Accept: application/xml

The same request, using the curl command:

curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api
The result is an object of type Api:

```xml
<api>
  <link href="/ovirt-engine/api/clusters" rel="clusters"/>
  <link href="/ovirt-engine/api/datacenters" rel="datacenters"/>
  ...
  <product_info>
    <name>oVirt Engine</name>
    <vendor>ovirt.org</vendor>
    <version>
      <build>0</build>
      <full_version>4.0.0-0.0.el7</full_version>
      <major>4</major>
      <minor>0</minor>
      <revision>0</revision>
    </version>
  </product_info>
  <special_objects>
    <blank_template href="..." id="..."/>
    <root_tag href="..." id="..."/>
  </special_objects>
  <summary>
    <hosts>
      <active>23</active>
      <total>30</total>
    </hosts>
    <storage_domains>
      <active>5</active>
      <total>6</total>
    </storage_domains>
    <users>
      <active>12</active>
      <total>102</total>
    </users>
    <vms>
      <active>253</active>
      <total>545</total>
    </vms>
  </summary>
  <time>2016-10-06T15:38:18.548+02:00</time>
</api>
```

**IMPORTANT**

When neither the header nor the URL prefix are used, the server will automatically select a version. The default is version 4. You can change the default version using the `ENGINE_API_DEFAULT_VERSION` configuration parameter:

```bash
# echo "ENGINE_API_DEFAULT_VERSION=3" > /
# systemctl restart ovirt-engine
```

Changing this parameter affects all users of the API that don't specify the version explicitly.
The entry point provides a user with links to the collections in a virtualization environment. The `rel` attribute of each collection link provides a reference point for each link. The next step in this example examines the data center collection, which is available through the `datacenters` link.

The entry point also contains other data such as `product_info`, `special_objects` and `summary`. This data is covered in chapters outside this example.

### 4.2. LIST DATA CENTERS

Red Hat Virtualization creates a **Default** data center on installation. This example uses the **Default** data center as the basis for the virtual environment.

The following request retrieves a representation of the data centers:

```
GET /ovirt-engine/api/datacenters HTTP/1.1
Accept: application/xml
```

The same request, using the **curl** command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/datacenters
```

The result will be a list of objects of type `DataCenter`:

```
<data_centers>
  <data_center href="/ovirt-engine/api/datacenters/001" id="001">
    <name>Default</name>
    <description>The default Data Center</description>
    <link href="/ovirt-engine/api/datacenters/001/clusters" rel="clusters"/>
    <link href="/ovirt-engine/api/datacenters/001/storagedomains" rel="storagedomains"/>
    ...     <local>false</local>
    <quota_mode>disabled</quota_mode>
    <status>up</status>
    <supported_versions>
      <version>
        <major>4</major>
        <minor>0</minor>
      </version>
    </supported_versions>
  </data_center>
  ...
</data_centers>
```
Note the **id** of your **Default** data center. It identifies this data center in relation to other resources of your virtual environment.

The data center also contains a link to the **service** that manages the storage domains attached to the data center:

```xml
<link href="/ovirt-engine/api/datacenters/001/storagedomains" rel="storagedomains"/>
```

That service is used to attach storage domains from the main **storagedomains** collection, which this example covers later.

### 4.3. LIST HOST CLUSTERS

Red Hat Virtualization creates a **Default** hosts cluster on installation. This example uses the **Default** cluster to group resources in your Red Hat Virtualization environment.

The following request retrieves a representation of the cluster collection:

```plaintext
GET /ovirt-engine/api/clusters HTTP/1.1
Accept: application/xml
```

The same request, using the **curl** command:

```bash
curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/clusters
```

The result will be a list of objects of type **Cluster**:

```xml
<clusters>
  <cluster href="/ovirt-engine/api/clusters/002" id="002">
    <name>Default</name>
    <description>The default server cluster</description>
    <link href="/ovirt-engine/api/clusters/002/networks" rel="networks"/>
    <link href="/ovirt-engine/api/clusters/002" rel="permissions"/>
    ...
    <cpu>
      <architecture>x86_64</architecture>
      <type>Intel Conroe Family</type>
    </cpu>
    <version>
      <major>4</major>
      <minor>0</minor>
    </version>
    <data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
  </cluster>
  ...
</clusters>
```
Note the id of your Default host cluster. It identifies this host cluster in relation to other resources of your virtual environment.

The Default cluster is associated with the Default data center through a relationship using the id and href attributes of the data_center link:

```xml
<data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
```

The networks link is a reference to the service that manages the networks associated to this cluster. The next section examines the networks collection in more detail.

### 4.4. LIST LOGICAL NETWORKS

Red Hat Virtualization creates a default ovirtmgmt network on installation. This network acts as the management network for Red Hat Virtualization Manager to access hosts.

This network is associated with the Default cluster and is a member of the Default data center. This example uses the ovirtmgmt network to connect the virtual machines.

The following request retrieves the list of logical networks:

```plaintext
GET /ovirt-engine/api/networks HTTP/1.1
Accept: application/xml
```

The same request, using the curl command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/networks
```

The result will be a list of objects of type Network:

```xml
<networks>
  <network href="/ovirt-engine/api/networks/003" id="003">
    <name>ovirtmgmt</name>
    <description>Management Network</description>
    <link href="/ovirt-engine/api/networks/003/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/networks/003/vnicprofiles" rel="vnicprofiles"/>
    <link href="/ovirt-engine/api/networks/003/networklabels" rel="networklabels"/>
    <mtu>0</mtu>
    <stp>false</stp>
    <usages>
      <usage>vm</usage>
    </usages>
    <data_center href="/ovirt-engine/api/datacenters/001" id="001"/>
  </network>
</networks>
```
The ovirtmgmt network is attached to the Default data center through a relationship using the data center’s id.

The ovirtmgmt network is also attached to the Default cluster through a relationship in the cluster’s network sub-collection.

4.5. LIST HOSTS

This example retrieves the list of hosts and shows a host named myhost registered with the virtualization environment:

GET /ovirt-engine/api/hosts HTTP/1.1
Accept: application/xml

The same request, using the curl command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--request GET \
--header 'Version: 4' \
--header 'Accept: application/xml' \
--user 'admin@internal:mypassword' \
https://myengine.example.com/ovirt-engine/api/hosts
```

The result will be a list of objects of type Host:

```
<hosts>
  <host href="/ovirt-engine/api/hosts/004" id="004">
    <name>myhost</name>
    <link href="/ovirt-engine/api/hosts/004/nics" rel="nics"/>
    ...
    <address>node40.example.com</address>
    <cpu>
      <name>Intel Core Processor (Haswell, no TSX)</name>
      <speed>3600</speed>
      <topology>
        <cores>1</cores>
        <sockets>2</sockets>
        <threads>1</threads>
      </topology>
    </cpu>
    <memory>8371830784</memory>
    <os>
      <type>RHEL</type>
      <version>
        <full_version>7 - 2.1511.el7.centos.2.10</full_version>
        <major>7</major>
      </version>
    </os>
    <port>54321</port>
  </host>
  ...
</hosts>
```
Note the id of your host. It identifies this host in relation to other resources of your virtual environment.

This host is a member of the Default cluster and accessing the nics sub-collection shows this host has a connection to the ovirtmgmt network.

### 4.6. CREATE NFS DATA STORAGE

An NFS data storage domain is an exported NFS share attached to a data center and provides storage for virtualized guest images. Creation of a new storage domain requires a POST request, with the storage domain representation included, sent to the URL of the storage domain collection.

You can enable the wipe after delete option by default on the storage domain. To configure this specify `wipe_after_delete` in the POST request. This option can be edited after the domain is created, but doing so will not change the wipe after delete property of disks that already exist.

The request should be like this:

```
POST /ovirt-engine/api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

And the request body should be like this:

```
<storage_domain>
  <name>mydata</name>
  <type>data</type>
  <description>My data</description>
  <storage>
    <type>nfs</type>
    <address>mynfs.example.com</address>
    <path>/exports/mydata</path>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
```

The same request, using the `curl` command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \ 
--user 'admin@internal:mypassword' \ 
--request POST \ 
--header 'Version: 4' \ 
--header 'Content-Type: application/xml' \ 
--header 'Accept: application/xml' \ 
--data '
<storage_domain>
```
https://myengine.example.com/ovirt-engine/api/storagedomains

The server uses host myhost to create a NFS data storage domain called mydata with an export path of mynfs.example.com:/exports/mydata. The API also returns the following representation of the newly created storage domain resource (of type StorageDomain):

```
<storage_domain id="005" href="/ovirt-engine/api/storagedomains/005">
  <name>mydata</name>
  <description>My data</description>
  <available>42949672960</available>
  <committed>0</committed>
  <master>false</master>
  <status>unattached</status>
  <storage>
    <address>mynfs.example.com</address>
    <path>/exports/mydata</path>
    <type>nfs</type>
  </storage>
  <storage_format>v3</storage_format>
  <type>data</type>
  <used>9663676416</used>
</storage_domain>
```

### 4.7. CREATE NFS ISO STORAGE

An NFS ISO storage domain is a mounted NFS share attached to a data center and provides storage for DVD/CD-ROM ISO and virtual floppy disk (VFD) image files. Creation of a new storage domain requires a POST request, with the storage domain representation included, sent to the URL of the storage domain collection:

The request should be like this:

```
POST /ovirt-engine/api/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

And the request body should be like this:

```
<storage_domain>
  <name>myisos</name>
  <description>My ISOS</description>
</storage_domain>
```
The same request, using the `curl` command:

```bash
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \n--user 'admin@internal:mypassword' \n--request POST \n--header 'Version: 4' \n--header 'Content-Type: application/xml' \n--header 'Accept: application/xml' \n--data ' 
<storage_domain> 
  <name>myisos</name> 
  <description>My ISOs</description> 
  <type>iso</type> 
  <storage> 
    <type>nfs</type> 
    <address>mynfs.example.com</address> 
    <path>/exports/myisos</path> 
  </storage> 
  <host> 
    <name>myhost</name> 
  </host> 
</storage_domain> 
' \nhttps://myengine.example.com/ovirt-engine/api/storagedomains
```

The server uses host `myhost` to create a NFS ISO storage domain called `myisos` with an export path of `mynfs.example.com:/exports/myisos`. The API also returns the following representation of the newly created storage domain resource (of type `StorageDomain`):

```xml
<storage_domain href="/ovirt-engine/api/storagedomains/006" id="006"> 
  <name>myiso</name> 
  <description>My ISOs</description> 
  <available>42949672960</available> 
  <committed>0</committed> 
  <master>false</master> 
  <status>unattached</status> 
  <storage> 
    <address>mynfs.example.com</address> 
    <path>/exports/myisos</path> 
    <type>nfs</type> 
  </storage> 
  <storage_format>v1</storage_format>
</storage_domain>
```
ATTACH STORAGE DOMAINS TO DATA CENTER

The following example attaches the mydata and myisos storage domains to the Default data center.

To attach the mydata storage domain, send a request like this:

```
POST /ovirt-engine/api/datacenters/001/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

With a request body like this:

```
<storage_domain>
  <name>mydata</name>
</storage_domain>
```

The same request, using the curl command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '"
<storage_domain>
  <name>mydata</name>
</storage_domain>
' \
https://myengine.example.com/ovirt-engine/api/datacenters/001/storagedomains
```

To attach the myisos storage domain, send a request like this:

```
POST /ovirt-engine/api/datacenters/001/storagedomains HTTP/1.1
Accept: application/xml
Content-type: application/xml

With a request body like this:

```
<storage_domain>
  <name>myisos</name>
</storage_domain>
```

The same request, using the curl command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '"
<storage_domain>
  <name>myisos</name>
</storage_domain>
' \
https://myengine.example.com/ovirt-engine/api/datacenters/001/storagedomains
```
4.9. CREATE VIRTUAL MACHINE

The following example creates a virtual machine called myvm on the Default cluster using the virtualization environment’s Blank template as a basis. The request also defines the virtual machine’s memory as 512 MiB and sets the boot device to a virtual hard disk.

The request should be contain an object of type Vm describing the virtual machine to create:

```xml
<vm>
  <name>myvm</name>
  <description>My VM</description>
  <cluster>
    <name>Default</name>
  </cluster>
  <template>
    <name>Blank</name>
  </template>
  <memory>536870912</memory>
  <os>
    <boot>
      <devices>
        <device>hd</device>
      </devices>
    </boot>
  </os>
</vm>
```

And the request body should be like this:

```bash
# curl
--cacert '/etc/pki/ovirt-engine/ca.pem' \n--user 'admin@internal:mypassword' \n--request POST \n--header 'Version: 4' \n--header 'Content-Type: application/xml' \nPOST /ovirt-engine/api/vms HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

The same request, using the curl command:
The response body will be an object of the `Vm` type:

```
<vm href="/ovirt-engine/api/vms/007" id="007">
    <name>myvm</name>
    <link href="/ovirt-engine/api/vms/007/diskattachments" rel="diskattachments"/>
    <link href="/ovirt-engine/api/vms/007/nics" rel="nics"/>
...
    <cpu>
        <architecture>x86_64</architecture>
        <topology>
            <cores>1</cores>
            <sockets>1</sockets>
            <threads>1</threads>
        </topology>
    </cpu>
    <memory>1073741824</memory>
    <os>
        <boot>
            <devices>
                <device>hd</device>
            </devices>
        </boot>
        <type>other</type>
    </os>
    <type>desktop</type>
    <cluster href="/ovirt-engine/api/clusters/002" id="002"/>
    <status>down</status>
    <original_template href="/ovirt-engine/api/templates/000" id="00"/>
    <template href="/ovirt-engine/api/templates/000" id="00"/>
</vm>
```
4.10. CREATE A VIRTUAL MACHINE NIC

The following example creates a virtual network interface to connect the example virtual machine to the ovirtmgmt network.

The request should be like this:

```plaintext
POST /ovirt-engine/api/vms/007/nics HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

The request body should contain an object of type Nic describing the NIC to be created:

```xml
<nic>
  <name>mynic</name>
  <description>My network interface card</description>
</nic>
```

The same request, using the curl command:

```
# curl \
  --cacert '/etc/pki/ovirt-engine/ca.pem' \
  --user 'admin@internal:mypassword' \
  --request POST \
  --header 'Version: 4' \
  --header 'Content-Type: application/xml' \
  --header 'Accept: application/xml' \
  --data ' \
    <nic>
      <name>mynic</name>
      <description>My network interface card</description>
    </nic>
' \
  https://myengine.example.com/ovirt-engine/api/vms/007/nics
```

4.11. CREATE VIRTUAL MACHINE DISK

The following example creates an 8 GiB copy-on-write disk for the example virtual machine.

The request should be like this:

```plaintext
POST /ovirt-engine/api/vms/007/diskattachments HTTP/1.1
Content-Type: application/xml
Accept: application/xml
```

The request body should be an object of type DiskAttachment describing the disk and how it will be attached to the virtual machine:

```xml
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <active>true</active>
</disk_attachment>
```
The same request, using the **curl** command:

```bash
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '  
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <active>true</active>
  <disk>
    <description>My disk</description>
    <format>cow</format>
    <name>mydisk</name>
    <provisioned_size>8589934592</provisioned_size>
    <storage_domains>
      <storage_domain>
        <name>mydata</name>
      </storage_domain>
    </storage_domains>
  </disk>
</disk_attachment>
```

The same request, using the **curl** command:

```bash
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data '  
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <active>true</active>
  <disk>
    <description>My disk</description>
    <format>cow</format>
    <name>mydisk</name>
    <provisioned_size>8589934592</provisioned_size>
    <storage_domains>
      <storage_domain>
        <name>mydata</name>
      </storage_domain>
    </storage_domains>
  </disk>
</disk_attachment>
```

The **storage_domains** attribute tells the API to store the disk on the **mydata** storage domain.

### 4.12. ATTACH ISO IMAGE TO VIRTUAL MACHINE

The boot media for the following virtual machine example requires a CD-ROM or DVD ISO image for an operating system installation. This example uses a CentOS 7 image.

ISO images must be available in the **myisos** ISO domain for the virtual machines to use. You can use Section 6.114, “ImageTransfers” to create an image transfer and Section 6.113, “ImageTransfer” to upload the ISO image.

Once the ISO image is uploaded, an API can be used to request the list of files from the ISO storage domain:
GET /ovirt-engine/api/storagedomains/006/files HTTP/1.1
Accept: application/xml

The same request, using the `curl` command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \ 
--user 'admin@internal:mypassword' \ 
--request GET \ 
--header 'Version: 4' \ 
--header 'Accept: application/xml' \ 
https://myengine.example.com/ovirt-engine/api/storagedomains/006/files
```

The server returns the following list of objects of type `File`, one for each available ISO (or floppy) image:

```
<files>
  <file href="" id="CentOS-7-x86_64-Minimal.iso">CentOS-7-x86_64-Minimal.iso</name>
</file>
```

An API user attaches the `CentOS-7-x86_64-Minimal.iso` to the example virtual machine. Attaching an ISO image is equivalent to using the `Change CD` button in the administration or user portal applications.

The request should be like this:

```
PUT /ovirt-engine/api/vms/007/cdroms/00000000-0000-0000-0000-000000000000 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

The request body should be an object of type `Cdrom` containing an inner `file` attribute to indicate the identifier of the ISO (or floppy) image:

```
<cdrom>
  <file id="CentOS-7-x86_64-Minimal.iso"/>
</cdrom>
```

The same request, using the `curl` command:

```
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \ 
--user 'admin@internal:mypassword' \ 
--request PUT \ 
--header 'Version: 4' \ 
--header 'Content-Type: application/xml' \ 
--header 'Accept: application/xml' \ 
--data '  
<cdrom>
  <file id="CentOS-7-x86_64-Minimal.iso"/>
</cdrom>
  
```

Red Hat Virtualization 4.2 REST API Guide
78
For more details see the documentation of the service that manages virtual machine CD-ROMS.

### 4.13. START THE VIRTUAL MACHINE

The virtual environment is complete and the virtual machine contains all necessary components to function. This example starts the virtual machine using the start method.

The request should be like this:

```plaintext
POST /ovirt-engine/api/vms/007/start HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

The request body should be like this:

```xml
<action>
  <vm>
    <os>
      <boot>
        <devices>
          <device>cdrom</device>
        </devices>
      </boot>
    </os>
  </vm>
</action>
```

The same request, using the curl command:

```bash
# curl \
--cacert '/etc/pki/ovirt-engine/ca.pem' \
--user 'admin@internal:mypassword' \
--request POST \
--header 'Version: 4' \
--header 'Content-Type: application/xml' \
--header 'Accept: application/xml' \
--data <action>
  <vm>
    <os>
      <boot>
        <devices>
          <device>cdrom</device>
        </devices>
      </boot>
    </os>
  </vm>
</action>
```

https://myengine.example.com/ovirt-engine/api/vms/007/start/cdroms/00000000-0000-0000-0000-000000000000
The additional request body sets the virtual machine's boot device to CD-ROM for this boot only. This enables the virtual machine to install the operating system from the attached ISO image. The boot device reverts back to disk for all future boots.
CHAPTER 5. REQUESTS

This section enumerates all the requests that are available in the API.

- POST /affinitylabels
- GET /affinitylabels
- GET /affinitylabels/{label:id}
- PUT /affinitylabels/{label:id}
- DELETE /affinitylabels/{label:id}
- POST /affinitylabels/{label:id}/hosts
- GET /affinitylabels/{label:id}/hosts
- DELETE /affinitylabels/{label:id}/hosts/{host:id}
- GET /affinitylabels/{label:id}/hosts/{host:id}
- POST /affinitylabels/{label:id}/vms
- GET /affinitylabels/{label:id}/vms
- DELETE /affinitylabels/{label:id}/vms/{vm:id}
- GET /affinitylabels/{label:id}/vms/{vm:id}
- POST /bookmarks
- GET /bookmarks
- GET /bookmarks/{bookmark:id}
- PUT /bookmarks/{bookmark:id}
- DELETE /bookmarks/{bookmark:id}
- GET /clusterlevels
- GET /clusterlevels/{level:id}
- GET /clusterlevels/{level:id}/clusterfeatures
- GET /clusterlevels/{level:id}/clusterfeatures/{feature:id}
- POST /clusters
- GET /clusters
- GET /clusters/{cluster:id}
- PUT /clusters/{cluster:id}
- DELETE /clusters/{cluster:id}
- POST /clusters/{cluster:id}/affinitygroups
- GET /clusters/{cluster:id}/affinitygroups
- GET /clusters/{cluster:id}/affinitygroups/{group:id}
- PUT /clusters/{cluster:id}/affinitygroups/{group:id}
- DELETE /clusters/{cluster:id}/affinitygroups/{group:id}
- POST /clusters/{cluster:id}/affinitygroups/{group:id}/vms
- GET /clusters/{cluster:id}/affinitygroups/{group:id}/vms
- DELETE /clusters/{cluster:id}/affinitygroups/{group:id}/vms/{vm:id}
- POST /clusters/{cluster:id}/cpuprofiles
- GET /clusters/{cluster:id}/cpuprofiles
- GET /clusters/{cluster:id}/cpuprofiles/{profile:id}
- DELETE /clusters/{cluster:id}/cpuprofiles/{profile:id}
- GET /clusters/{cluster:id}/enabledfeatures
- POST /clusters/{cluster:id}/enabledfeatures
- GET /clusters/{cluster:id}/enabledfeatures/{feature:id}
- DELETE /clusters/{cluster:id}/enabledfeatures/{feature:id}
- GET /clusters/{cluster:id}/externalnetworkproviders
- GET /clusters/{cluster:id}/glusterhooks
- GET /clusters/{cluster:id}/glusterhooks/{hook:id}
- DELETE /clusters/{cluster:id}/glusterhooks/{hook:id}
- POST /clusters/{cluster:id}/glusterhooks/{hook:id}/disable
- POST /clusters/{cluster:id}/glusterhooks/{hook:id}/enable
- POST /clusters/{cluster:id}/glusterhooks/{hook:id}/resolve
- POST /clusters/{cluster:id}/glustervolumes
- GET /clusters/{cluster:id}/glustervolumes
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}
- DELETE /clusters/{cluster:id}/glustervolumes/{volume:id}
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/getprofilestatistics
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- DELETE /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/activate
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/migrate
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/stopmigrate
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}
- DELETE /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/replace
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/{brick:id}/statistics/{statistic:id}
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/rebalance
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/resetalloptions
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/resetoption
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/setoption
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/start
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/startprofile
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/statistics
- GET /clusters/{cluster:id}/glustervolumes/{volume:id}/statistics/{statistic:id}
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/stop
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/stopprofile
- POST /clusters/{cluster:id}/glustervolumes/{volume:id}/stoprebalance
- GET /clusters/{cluster:id}/networkfilters
- GET /clusters/{cluster:id}/networkfilters/{networkfilter:id}
- POST /clusters/{cluster:id}/networks
- GET /clusters/{cluster:id}/networks
- GET /clusters/{cluster:id}/networks/{network:id}
- DELETE /clusters/{cluster:id}/networks/{network:id}
- PUT /clusters/{cluster:id}/networks/{network:id}
- POST /clusters/{cluster:id}/permissions
- GET /clusters/{cluster:id}/permissions
- GET /clusters/{cluster:id}/permissions/{permission:id}
- DELETE /clusters/{cluster:id}/permissions/{permission:id}
- POST /clusters/{cluster:id}/resetemulatedmachine
- POST /clusters/{cluster:id}/syncallnetworks
- POST /cpuprofiles
- GET /cpuprofiles
- GET /cpuprofiles/{profile:id}
- PUT /cpuprofiles/{profile:id}
- DELETE /cpuprofiles/{profile:id}
- POST /cpuprofiles/{profile:id}/permissions
- GET /cpuprofiles/{profile:id}/permissions
- GET /cpuprofiles/{profile:id}/permissions/{permission:id}
- DELETE /cpuprofiles/{profile:id}/permissions/{permission:id}
- POST /datacenters
- GET /datacenters
- GET /datacenters/{datacenter:id}
- PUT /datacenters/{datacenter:id}
- DELETE /datacenters/{datacenter:id}
- POST /datacenters/{datacenter:id}/clusters
- GET /datacenters/{datacenter:id}/clusters
- GET /datacenters/{datacenter:id}/clusters/{cluster:id}
- PUT /datacenters/{datacenter:id}/clusters/{cluster:id}
- DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}
- POST /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups
- GET /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups
- GET /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}
- PUT /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}
- DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}/vms
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}/vms
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/affinitygroups/{group:id}/vms/{vm:id}
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/cpuprofiles
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/cpuprofiles
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/cpuprofiles/{profile:id}
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/enabledfeatures
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/enabledfeatures
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/enabledfeatures/{feature:id}
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/enabledfeatures/{feature:id}
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/externalnetworkproviders
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/glusterhooks
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/glusterhooks/{hook:id}
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glusterhooks/{hook:id}/disable
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glusterhooks/{hook:id}/enable
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glusterhooks/{hook:id}/resolve
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/getprofilestatistics
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
GET /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
DELETE /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks
POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/activate
- POST /datacenters/{datacenter:id}/clusters/{cluster:id}/glustervolumes/{volume:id}/glusterbricks/migrate
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/glusterbricks/stopmigration
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/glusterbricks/{brick.id}
- DELETE /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/glusterbricks/{brick.id}
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/glusterbricks/{brick.id}/reset
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/glusterbricks/{brick.id}/reset
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/rebalance
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/resetalloptions
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/resetoption
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/setoption
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/start
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/startprofile
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/statistics
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/statistics/{statistic:id}
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/stop
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/stopprofile
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/glustervolumes/{volume.id}/stoprebalance
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/networkfilters
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/networkfilters/{networkfilter:id}
- POST /datacenters/{datacenter.id}/clusters/{cluster.id}/networks
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/networks
- GET /datacenters/{datacenter.id}/clusters/{cluster.id}/networks/{network.id}
- DELETE /datacenters/{datacenter.id}/clusters/{cluster.id}/networks/{network.id}
- PUT /datacenters/{datacenter.id}/clusters/{cluster.id}/networks/{network.id}


- **POST** /datacenters/{datacenter:id}/clusters/{cluster:id}/permissions

- **GET** /datacenters/{datacenter:id}/clusters/{cluster:id}/permissions

- **GET** /datacenters/{datacenter:id}/clusters/{cluster:id}/permissions/{permission:id}

- **DELETE** /datacenters/{datacenter:id}/clusters/{cluster:id}/permissions/{permission:id}

- **POST** /datacenters/{datacenter:id}/clusters/{cluster:id}/resetemulatedmachine

- **POST** /datacenters/{datacenter:id}/clusters/{cluster:id}/syncallnetworks

- **POST** /datacenters/{datacenter:id}/iscsibonds

- **GET** /datacenters/{datacenter:id}/iscsibonds

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}

- **PUT** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}

- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}

- **POST** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}

- **PUT** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}

- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}

- **POST** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/networklabels

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/networklabels

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/networklabels/{label:id}

- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/networklabels/{label:id}

- **POST** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/permissions

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/permissions

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/permissions/{permission:id}

- **DELETE** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/permissions/{permission:id}

- **POST** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles

- **GET** /datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}
- **DELETE**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}`

- **POST**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}`

- **GET**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}`

- **GET**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}`

- **DELETE**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/networks/{network:id}/vnicprofiles/{profile:id}`

- **POST**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections`

- **GET**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections`

- **GET**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storageconnection:id}`

- **PUT**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storageconnection:id}`

- **DELETE**
  `/datacenters/{datacenter:id}/iscsibonds/{iscsibond:id}/storageserverconnections/{storageconnection:id}`

- **POST**
  `/datacenters/{datacenter:id}/networks`

- **GET**
  `/datacenters/{datacenter:id}/networks`

- **GET**
  `/datacenters/{datacenter:id}/networks/{network:id}`

- **DELETE**
  `/datacenters/{datacenter:id}/networks/{network:id}`

- **PUT**
  `/datacenters/{datacenter:id}/networks/{network:id}`

- **POST**
  `/datacenters/{datacenter:id}/permissions`

- **GET**
  `/datacenters/{datacenter:id}/permissions`

- **GET**
  `/datacenters/{datacenter:id}/permissions/{permission:id}`

- **DELETE**
  `/datacenters/{datacenter:id}/permissions/{permission:id}`

- **POST**
  `/datacenters/{datacenter:id}/qoss`

- **GET**
  `/datacenters/{datacenter:id}/qoss`

- **GET**
  `/datacenters/{datacenter:id}/qoss/{qos:id}`

- **PUT**
  `/datacenters/{datacenter:id}/qoss/{qos:id}`

- **DELETE**
  `/datacenters/{datacenter:id}/qoss/{qos:id}`

- **POST**
  `/datacenters/{datacenter:id}/quotas`
GET /datacenters/{datacenter:id}/quotas
GET /datacenters/{datacenter:id}/quotas/{quota:id}
PUT /datacenters/{datacenter:id}/quotas/{quota:id}
DELETE /datacenters/{datacenter:id}/quotas/{quota:id}
POST /datacenters/{datacenter:id}/quotas/{quota:id}/permissions
GET /datacenters/{datacenter:id}/quotas/{quota:id}/permissions
GET /datacenters/{datacenter:id}/quotas/{quota:id}/permissions/{permission:id}
DELETE /datacenters/{datacenter:id}/quotas/{quota:id}/permissions/{permission:id}
POST /datacenters/{datacenter:id}/quotas/{quota:id}/quotaclusterlimits
GET /datacenters/{datacenter:id}/quotas/{quota:id}/quotaclusterlimits
GET /datacenters/{datacenter:id}/quotas/{quota:id}/quotaclusterlimits/{limit:id}
DELETE /datacenters/{datacenter:id}/quotas/{quota:id}/quotaclusterlimits/{limit:id}
POST /datacenters/{datacenter:id}/quotas/{quota:id}/quotastoragelimits
GET /datacenters/{datacenter:id}/quotas/{quota:id}/quotastoragelimits
GET /datacenters/{datacenter:id}/quotas/{quota:id}/quotastoragelimits/{limit:id}
DELETE /datacenters/{datacenter:id}/quotas/{quota:id}/quotastoragelimits/{limit:id}
POST /datacenters/{datacenter:id}/storagedomains
GET /datacenters/{datacenter:id}/storagedomains
GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}
DELETE /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}
POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/activate
POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/deactivate
POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks
GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks
PUT /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}
GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}
DELETE /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}
POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/copy
POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/export
- POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/move
- POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- DELETE /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/register
- POST /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/sparsify
- GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/statistics
- GET /datacenters/{datacenter:id}/storagedomains/{storagedomain:id}/disks/{disk:id}/statistics/{statistic:id}
- POST /diskprofiles
- GET /diskprofiles
- GET /diskprofiles/{diskprofile:id}
- PUT /diskprofiles/{diskprofile:id}
- DELETE /diskprofiles/{diskprofile:id}
- POST /diskprofiles/{diskprofile:id}/permissions
- GET /diskprofiles/{diskprofile:id}/permissions
- GET /diskprofiles/{diskprofile:id}/permissions/{permission:id}
- DELETE /diskprofiles/{diskprofile:id}/permissions/{permission:id}
- POST /diskprofiles/{diskprofile:id}/permissions/{permission:id}
- POST /disks
- GET /disks
- PUT /disks/{disk:id}
- GET /disks/{disk:id}
- DELETE /disks/{disk:id}
- POST /disks/{disk:id}/copy
- POST /disks/{disk:id}/export
- POST /disks/{disk:id}/move
- POST /disks/{disk:id}/permissions
• GET /disks/{disk:id}/permissions
• GET /disks/{disk:id}/permissions/{permission:id}
• DELETE /disks/{disk:id}/permissions/{permission:id}
• POST /disks/{disk:id}/reduce
• POST /disks/{disk:id}/refreshlun
• POST /disks/{disk:id}/sparsify
• GET /disks/{disk:id}/statistics
• GET /disks/{disk:id}/statistics/{statistic:id}
• GET /domains
• GET /domains/{domain:id}
• GET /domains/{domain:id}/groups
• GET /domains/{domain:id}/groups/{group:id}
• GET /domains/{domain:id}/users
• GET /domains/{domain:id}/users/{user:id}
• POST /events
• GET /events
• POST /events/undelete
• GET /events/{event:id}
• DELETE /events/{event:id}
• POST /externalhostproviders
• GET /externalhostproviders
• GET /externalhostproviders/{provider:id}
• PUT /externalhostproviders/{provider:id}
• DELETE /externalhostproviders/{provider:id}
• GET /externalhostproviders/{provider:id}/certificates
• GET /externalhostproviders/{provider:id}/certificates/{certificate:id}
• GET /externalhostproviders/{provider:id}/computeresources
• GET /externalhostproviders/{provider:id}/computeresources/{resource:id}
• GET /externalhostproviders/{provider:id}/discoveredhosts
GET /externalhostproviders/{provider:id}/discoveredhosts/{host:id}
GET /externalhostproviders/{provider:id}/hostgroups
GET /externalhostproviders/{provider:id}/hostgroups/{group:id}
GET /externalhostproviders/{provider:id}/hosts
GET /externalhostproviders/{provider:id}/hosts/{host:id}
POST /externalhostproviders/{provider:id}/importcertificates
POST /externalhostproviders/{provider:id}/testconnectivity
POST /externalvmimports
POST /groups
GET /groups
GET /groups/{group:id}
DELETE /groups/{group:id}
POST /groups/{group:id}/permissions
GET /groups/{group:id}/permissions
GET /groups/{group:id}/permissions/{permission:id}
DELETE /groups/{group:id}/permissions/{permission:id}
GET /groups/{group:id}/roles
GET /groups/{group:id}/roles/{role:id}
DELETE /groups/{group:id}/roles/{role:id}
PUT /groups/{group:id}/roles/{role:id}
POST /groups/{group:id}/roles/{role:id}/permits
GET /groups/{group:id}/roles/{role:id}/permits
GET /groups/{group:id}/roles/{role:id}/permits/{permit:id}
DELETE /groups/{group:id}/roles/{role:id}/permits/{permit:id}
POST /groups/{group:id}/tags
GET /groups/{group:id}/tags
GET /groups/{group:id}/tags/{tag:id}
DELETE /groups/{group:id}/tags/{tag:id}
POST /hosts
GET /hosts
GET /hosts/{host:id}
PUT /hosts/{host:id}
DELETE /hosts/{host:id}
POST /hosts/{host:id}/activate
POST /hosts/{host:id}/affinitylabels
GET /hosts/{host:id}/affinitylabels
GET /hosts/{host:id}/affinitylabels/{label:id}
DELETE /hosts/{host:id}/affinitylabels/{label:id}
POST /hosts/{host:id}/approve
POST /hosts/{host:id}/commitnetconfig
POST /hosts/{host:id}/deactivate
GET /hosts/{host:id}/devices
GET /hosts/{host:id}/devices/{device:id}
POST /hosts/{host:id}/enrollcertificate
GET /hosts/{host:id}/externalnetworkproviderconfigurations
GET /hosts/{host:id}/externalnetworkproviderconfigurations/{configuration:id}
POST /hosts/{host:id}/fence
POST /hosts/{host:id}/fenceagents
GET /hosts/{host:id}/fenceagents
GET /hosts/{host:id}/fenceagents/{agent:id}
PUT /hosts/{host:id}/fenceagents/{agent:id}
DELETE /hosts/{host:id}/fenceagents/{agent:id}
POST /hosts/{host:id}/forceselectspm
GET /hosts/{host:id}/hooks
GET /hosts/{host:id}/hooks/{hook:id}
POST /hosts/{host:id}/install
POST /hosts/{host:id}/iscsidiscover
POST /hosts/{host:id}/iscsilogin
GET /hosts/{host:id}/katelloerrata
GET /hosts/{host:id}/katelloerrata/{katelloerratum:id}
POST /hosts/{host:id}/networkattachments
GET /hosts/{host:id}/networkattachments
GET /hosts/{host:id}/networkattachments/{attachment:id}
PUT /hosts/{host:id}/networkattachments/{attachment:id}
DELETE /hosts/{host:id}/networkattachments/{attachment:id}
GET /hosts/{host:id}/nics
GET /hosts/{host:id}/nics/{nic:id}
GET /hosts/{host:id}/nics/{nic:id}/linklayerdiscoveryprotocolelements
POST /hosts/{host:id}/nics/{nic:id}/networkattachments
GET /hosts/{host:id}/nics/{nic:id}/networkattachments
GET /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}
PUT /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}
DELETE /hosts/{host:id}/nics/{nic:id}/networkattachments/{attachment:id}
POST /hosts/{host:id}/nics/{nic:id}/networklabels
GET /hosts/{host:id}/nics/{nic:id}/networklabels
GET /hosts/{host:id}/nics/{nic:id}/networklabels/{label:id}
DELETE /hosts/{host:id}/nics/{nic:id}/networklabels/{label:id}
GET /hosts/{host:id}/nics/{nic:id}/statistics
GET /hosts/{host:id}/nics/{nic:id}/statistics/{statistic:id}
POST /hosts/{host:id}/nics/{nic:id}/updatevirtualfunctionsconfiguration
POST /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowedlabels
GET /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowedlabels
GET /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowedlabels/{label:id}
DELETE /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowedlabels/{label:id}
POST /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowednetworks
GET /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowednetworks
GET /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowednetworks/{network:id}
- DELETE /hosts/{host:id}/nics/{nic:id}/virtualfunctionallowednetworks/{network:id}
- GET /hosts/{host:id}/numanodes
- GET /hosts/{host:id}/numanodes/{node:id}
- GET /hosts/{host:id}/numanodes/{node:id}/statistics
- GET /hosts/{host:id}/numanodes/{node:id}/statistics/{statistic:id}
- POST /hosts/{host:id}/permissions
- GET /hosts/{host:id}/permissions
- GET /hosts/{host:id}/permissions/{permission:id}
- DELETE /hosts/{host:id}/permissions/{permission:id}
- POST /hosts/{host:id}/refresh
- POST /hosts/{host:id}/setupnetworks
- GET /hosts/{host:id}/statistics
- GET /hosts/{host:id}/statistics/{statistic:id}
- GET /hosts/{host:id}/storage
- GET /hosts/{host:id}/storage/{storage:id}
- POST /hosts/{host:id}/storageconnectionextensions
- GET /hosts/{host:id}/storageconnectionextensions
- GET /hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- PUT /hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- DELETE /hosts/{host:id}/storageconnectionextensions/{storageconnectionextension:id}
- POST /hosts/{host:id}/syncallnetworks
- POST /hosts/{host:id}/tags
- GET /hosts/{host:id}/tags
- GET /hosts/{host:id}/tags/{tag:id}
- DELETE /hosts/{host:id}/tags/{tag:id}
- GET /hosts/{host:id}/unmanagednetworks
- GET /hosts/{host:id}/unmanagednetworks/{unmanagednetwork:id}
- DELETE /hosts/{host:id}/unmanagednetworks/{unmanagednetwork:id}
- POST /hosts/{host:id}/unregisteredstoragedomainsdiscover
POST /hosts/{host:id}/upgrade
POST /hosts/{host:id}/upgradecheck
GET /icons
GET /icons/{icon:id}
POST /imagetransfers
GET /imagetransfers
GET /imagetransfers/{imagetransfer:id}
POST /imagetransfers/{imagetransfer:id}/cancel
POST /imagetransfers/{imagetransfer:id}/extend
POST /imagetransfers/{imagetransfer:id}/finalize
POST /imagetransfers/{imagetransfer:id}/pause
POST /imagetransfers/{imagetransfer:id}/resume
POST /instancetypes
GET /instancetypes
GET /instancetypes/{instancetype:id}
PUT /instancetypes/{instancetype:id}
DELETE /instancetypes/{instancetype:id}
POST /instancetypes/{instancetype:id}/graphicsconsoles
GET /instancetypes/{instancetype:id}/graphicsconsoles
GET /instancetypes/{instancetype:id}/graphicsconsoles/{console:id}
DELETE /instancetypes/{instancetype:id}/graphicsconsoles/{console:id}
POST /instancetypes/{instancetype:id}/nics
GET /instancetypes/{instancetype:id}/nics
GET /instancetypes/{instancetype:id}/nics/{nic:id}
PUT /instancetypes/{instancetype:id}/nics/{nic:id}
DELETE /instancetypes/{instancetype:id}/nics/{nic:id}
POST /instancetypes/{instancetype:id}/watchdogs
GET /instancetypes/{instancetype:id}/watchdogs
GET /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}
- PUT /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}
- DELETE /instancetypes/{instancetype:id}/watchdogs/{watchdog:id}
- POST /jobs
- GET /jobs
- GET /jobs/{job:id}
- POST /jobs/{job:id}/clear
- POST /jobs/{job:id}/end
- POST /jobs/{job:id}/steps
- GET /jobs/{job:id}/steps
- GET /jobs/{job:id}/steps/{step:id}
- POST /jobs/{job:id}/steps/{step:id}/end
- GET /jobs/{job:id}/steps/{step:id}/statistics
- GET /jobs/{job:id}/steps/{step:id}/statistics/{statistic:id}
- GET /katelloerrata
- GET /katelloerrata/{katelloerratum:id}
- POST /macpools
- GET /macpools
- GET /macpools/{macpool:id}
- PUT /macpools/{macpool:id}
- DELETE /macpools/{macpool:id}
- GET /networkfilters
- GET /networkfilters/{networkfilter:id}
- POST /networks
- GET /networks
- GET /networks/{network:id}
- PUT /networks/{network:id}
- DELETE /networks/{network:id}
- POST /networks/{network:id}/networklabels
- GET /networks/{network:id}/networklabels
- GET /networks/{network:id}/networklabels/{label:id}
- DELETE /networks/{network:id}/networklabels/{label:id}
- POST /networks/{network:id}/permissions
- GET /networks/{network:id}/permissions
- GET /networks/{network:id}/permissions/{permission:id}
- DELETE /networks/{network:id}/permissions/{permission:id}
- POST /networks/{network:id}/vnicprofiles
- GET /networks/{network:id}/vnicprofiles
- GET /networks/{network:id}/vnicprofiles/{profile:id}
- DELETE /networks/{network:id}/vnicprofiles/{profile:id}
- POST /networks/{network:id}/vnicprofiles/{profile:id}/permissions
- GET /networks/{network:id}/vnicprofiles/{profile:id}/permissions
- GET /networks/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}
- DELETE /networks/{network:id}/vnicprofiles/{profile:id}/permissions/{permission:id}
- POST /openstackimageproviders
- GET /openstackimageproviders
- GET /openstackimageproviders/{provider:id}
- PUT /openstackimageproviders/{provider:id}
- DELETE /openstackimageproviders/{provider:id}
- GET /openstackimageproviders/{provider:id}/certificates
- GET /openstackimageproviders/{provider:id}/certificates/{certificate:id}
- GET /openstackimageproviders/{provider:id}/images
- GET /openstackimageproviders/{provider:id}/images/{image:id}
- POST /openstackimageproviders/{provider:id}/images/{image:id}/import
- POST /openstackimageproviders/{provider:id}/importcertificates
- POST /openstackimageproviders/{provider:id}/testconnectivity
- POST /openstacknetworkproviders
- GET /openstacknetworkproviders
- GET /openstacknetworkproviders/{provider:id}
CHAPTER 5. REQUESTS

- PUT /openstacknetworkproviders/{provider:id}
- DELETE /openstacknetworkproviders/{provider:id}
- GET /openstacknetworkproviders/{provider:id}/certificates
- GET /openstacknetworkproviders/{provider:id}/certificates/{certificate:id}
- POST /openstacknetworkproviders/{provider:id}/importcertificates
- GET /openstacknetworkproviders/{provider:id}/networks
- GET /openstacknetworkproviders/{provider:id}/networks/{network:id}
- POST /openstacknetworkproviders/{provider:id}/networks/{network:id}/import
- POST /openstacknetworkproviders/{provider:id}/networks/{network:id}/subnets
- GET /openstacknetworkproviders/{provider:id}/networks/{network:id}/subnets
- GET /openstacknetworkproviders/{provider:id}/networks/{network:id}/subnets/{subnet:id}
- DELETE /openstacknetworkproviders/{provider:id}/networks/{network:id}/subnets/{subnet:id}
- POST /openstacknetworkproviders/{provider:id}/testconnectivity

- POST /openstackvolumeproviders
- GET /openstackvolumeproviders
- PUT /openstackvolumeproviders/{provider:id}
- DELETE /openstackvolumeproviders/{provider:id}
- POST /openstackvolumeproviders/{provider:id}/authenticationkeys
- GET /openstackvolumeproviders/{provider:id}/authenticationkeys
- GET /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}
- PUT /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}
- DELETE /openstackvolumeproviders/{provider:id}/authenticationkeys/{key:id}
- GET /openstackvolumeproviders/{provider:id}/certificates
- GET /openstackvolumeproviders/{provider:id}/certificates/{certificate:id}
- POST /openstackvolumeproviders/{provider:id}/importcertificates
- POST /openstackvolumeproviders/{provider:id}/testconnectivity
- GET /openstackvolumeproviders/{provider:id}/volumetypes
- GET /openstackvolumeproviders/{provider:id}/volumetypes/{type:id}
- GET /operatingsystems
- GET /operatingsystems/{operatingsystem:id}
- GET /options/{option:id}
- POST /permissions
- GET /permissions
- GET /permissions/{permission:id}
- DELETE /permissions/{permission:id}
- POST /roles
- GET /roles
- GET /roles/{role:id}
- DELETE /roles/{role:id}
- PUT /roles/{role:id}
- POST /roles/{role:id}/permits
- GET /roles/{role:id}/permits
- GET /roles/{role:id}/permits/{permit:id}
- DELETE /roles/{role:id}/permits/{permit:id}
- POST /schedulingpolicies
- GET /schedulingpolicies
- GET /schedulingpolicies/{policy:id}
- PUT /schedulingpolicies/{policy:id}
- DELETE /schedulingpolicies/{policy:id}
- POST /schedulingpolicies/{policy:id}/balances
- GET /schedulingpolicies/{policy:id}/balances
- GET /schedulingpolicies/{policy:id}/balances/{balance:id}
- DELETE /schedulingpolicies/{policy:id}/balances/{balance:id}
- POST /schedulingpolicies/{policy:id}/filters
- GET /schedulingpolicies/{policy:id}/filters
- GET /schedulingpolicies/{policy:id}/filters/{filter:id}
- DELETE /schedulingpolicies/{policy:id}/filters/{filter:id}
POST /schedulingpolicies/{policy:id}/weights
GET /schedulingpolicies/{policy:id}/weights
GET /schedulingpolicies/{policy:id}/weights/{weight:id}
DELETE /schedulingpolicies/{policy:id}/weights/{weight:id}
GET /schedulingpolicyunits
GET /schedulingpolicyunits/{unit:id}
DELETE /schedulingpolicyunits/{unit:id}
POST /storageconnections
GET /storageconnections
GET /storageconnections/{storageconnection:id}
PUT /storageconnections/{storageconnection:id}
DELETE /storageconnections/{storageconnection:id}
POST /storagedomains
GET /storagedomains
GET /storagedomains/{storagedomain:id}
PUT /storagedomains/{storagedomain:id}
DELETE /storagedomains/{storagedomain:id}
POST /storagedomains/{storagedomain:id}/diskprofiles
GET /storagedomains/{storagedomain:id}/diskprofiles
GET /storagedomains/{storagedomain:id}/diskprofiles/{profile:id}
DELETE /storagedomains/{storagedomain:id}/diskprofiles/{profile:id}
POST /storagedomains/{storagedomain:id}/disks
GET /storagedomains/{storagedomain:id}/disks
GET /storagedomains/{storagedomain:id}/disks/{disk:id}
PUT /storagedomains/{storagedomain:id}/disks/{disk:id}
GET /storagedomains/{storagedomain:id}/disks/{disk:id}
DELETE /storagedomains/{storagedomain:id}/disks/{disk:id}
POST /storagedomains/{storagedomain:id}/disks/{disk:id}/copy
POST /storagedomains/{storagedomain:id}/disks/{disk:id}/export
POST /storagedomains/{storagedomain:id}/disks/{disk:id}/move
- POST /storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- GET /storagedomains/{storagedomain:id}/disks/{disk:id}/permissions
- GET /storagedomains/{storagedomain:id}/disks/{disk:id}/permissions/{permission:id}
- DELETE /storagedomains/{storagedomain:id}/disks/{disk:id}/permissions/{permission:id}
- POST /storagedomains/{storagedomain:id}/disks/{disk:id}/reduce
- POST /storagedomains/{storagedomain:id}/disks/{disk:id}/sparsify
- GET /storagedomains/{storagedomain:id}/disks/{disk:id}/statistics
- GET /storagedomains/{storagedomain:id}/disks/{disk:id}/statistics/{statistic:id}
- GET /storagedomains/{storagedomain:id}/disksnapshots
- GET /storagedomains/{storagedomain:id}/disksnapshots/{snapshot:id}
- DELETE /storagedomains/{storagedomain:id}/disksnapshots/{snapshot:id}
- GET /storagedomains/{storagedomain:id}/files
- GET /storagedomains/{storagedomain:id}/files/{file:id}
- GET /storagedomains/{storagedomain:id}/images
- GET /storagedomains/{storagedomain:id}/images/{image:id}
- POST /storagedomains/{storagedomain:id}/images/{image:id}/import
- POST /storagedomains/{storagedomain:id}/isattached
- POST /storagedomains/{storagedomain:id}/permissions
- GET /storagedomains/{storagedomain:id}/permissions
- GET /storagedomains/{storagedomain:id}/permissions/{permission:id}
- DELETE /storagedomains/{storagedomain:id}/permissions/{permission:id}
- POST /storagedomains/{storagedomain:id}/reduceluns
- POST /storagedomains/{storagedomain:id}/refreshluns
- POST /storagedomains/{storagedomain:id}/storageconnections
- GET /storagedomains/{storagedomain:id}/storageconnections
- GET /storagedomains/{storagedomain:id}/storageconnections/{connection:id}
- DELETE /storagedomains/{storagedomain:id}/storageconnections/{connection:id}
- GET /storagedomains/{storagedomain:id}/templates
- GET /storagedomains/{storagedomain:id}/templates/{template:id}
CHAPTER 5. REQUESTS

- DELETE /storagedomains/{storagedomain.id}/templates/{template.id}
- GET /storagedomains/{storagedomain.id}/templates/{template.id}/disks
- GET /storagedomains/{storagedomain.id}/templates/{template.id}/disks/{disk.id}
- POST /storagedomains/{storagedomain.id}/templates/{template.id}/import
- POST /storagedomains/{storagedomain.id}/templates/{template.id}/register
- POST /storagedomains/{storagedomain.id}/updateovfstore
- GET /storagedomains/{storagedomain.id}/vms
- GET /storagedomains/{storagedomain.id}/vms/{vm.id}
- DELETE /storagedomains/{storagedomain.id}/vms/{vm.id}
- GET /storagedomains/{storagedomain.id}/vms/{vm.id}/diskattachments
- GET /storagedomains/{storagedomain.id}/vms/{vm.id}/diskattachments/{attachment.id}
- GET /storagedomains/{storagedomain.id}/vms/{vm.id}/disks
- GET /storagedomains/{storagedomain.id}/vms/{vm.id}/disks/{disk.id}
- POST /storagedomains/{storagedomain.id}/vms/{vm.id}/import
- POST /storagedomains/{storagedomain.id}/vms/{vm.id}/register
- POST /tags
- GET /tags
- GET /tags/{tag.id}
- PUT /tags/{tag.id}
- DELETE /tags/{tag.id}
- POST /tags/{tag.id}
- GET /templates
- GET /templates
- GET /templates/{template.id}
- PUT /templates/{template.id}
- DELETE /templates/{template.id}
- GET /templates/{template.id}/cdroms
- GET /templates/{template.id}/cdroms/{cdrom.id}
- GET /templates/{template.id}/diskattachments
- GET /templates/{template.id}/diskattachments/{attachment.id}
- DELETE /templates/{template:id}/diskattachments/{attachment:id}
- POST /templates/{template:id}/export
- POST /templates/{template:id}/graphicsconsoles
- GET /templates/{template:id}/graphicsconsoles
- GET /templates/{template:id}/graphicsconsoles/{console:id}
- DELETE /templates/{template:id}/graphicsconsoles/{console:id}
- POST /templates/{template:id}/nics
- GET /templates/{template:id}/nics
- GET /templates/{template:id}/nics/{nic:id}
- PUT /templates/{template:id}/nics/{nic:id}
- DELETE /templates/{template:id}/nics/{nic:id}
- POST /templates/{template:id}/permissions
- GET /templates/{template:id}/permissions
- GET /templates/{template:id}/permissions/{permission:id}
- DELETE /templates/{template:id}/permissions/{permission:id}
- POST /templates/{template:id}/tags
- GET /templates/{template:id}/tags
- GET /templates/{template:id}/tags/{tag:id}
- DELETE /templates/{template:id}/tags/{tag:id}
- POST /templates/{template:id}/watchdogs
- GET /templates/{template:id}/watchdogs
- GET /templates/{template:id}/watchdogs/{watchdog:id}
- PUT /templates/{template:id}/watchdogs/{watchdog:id}
- DELETE /templates/{template:id}/watchdogs/{watchdog:id}
- POST /users
- GET /users
- GET /users/{user:id}
- DELETE /users/{user:id}
- GET /users/{user:id}/groups
• POST /users/{user:id}/permissions

• GET /users/{user:id}/permissions

• GET /users/{user:id}/permissions/{permission:id}

• DELETE /users/{user:id}/permissions/{permission:id}

• GET /users/{user:id}/roles

• GET /users/{user:id}/roles/{role:id}

• DELETE /users/{user:id}/roles/{role:id}

• PUT /users/{user:id}/roles/{role:id}

• POST /users/{user:id}/roles/{role:id}/permits

• GET /users/{user:id}/roles/{role:id}/permits

• GET /users/{user:id}/roles/{role:id}/permits/{permit:id}

• DELETE /users/{user:id}/roles/{role:id}/permits/{permit:id}

• POST /users/{user:id}/sshpublickeys

• GET /users/{user:id}/sshpublickeys

• GET /users/{user:id}/sshpublickeys/{key:id}

• PUT /users/{user:id}/sshpublickeys/{key:id}

• DELETE /users/{user:id}/sshpublickeys/{key:id}

• POST /users/{user:id}/tags

• GET /users/{user:id}/tags

• GET /users/{user:id}/tags/{tag:id}

• DELETE /users/{user:id}/tags/{tag:id}

• POST /vmpools

• GET /vmpools

• GET /vmpools/{pool:id}

• PUT /vmpools/{pool:id}

• DELETE /vmpools/{pool:id}

• POST /vmpools/{pool:id}/allocatevm

• POST /vmpools/{pool:id}/permissions

• GET /vmpools/{pool:id}/permissions
- GET /vmpools/{pool:id}/permissions/{permission.id}
- DELETE /vmpools/{pool:id}/permissions/{permission.id}
- POST /vms
- GET /vms
- GET /vms/{vm:id}
- PUT /vms/{vm:id}
- DELETE /vms/{vm:id}
- POST /vms/{vm:id}/affinitylabels
- GET /vms/{vm:id}/affinitylabels
- GET /vms/{vm:id}/affinitylabels/{label:id}
- DELETE /vms/{vm:id}/affinitylabels/{label:id}
- GET /vms/{vm:id}/applications
- GET /vms/{vm:id}/applications/{application.id}
- POST /vms/{vm:id}/cancelmigration
- POST /vms/{vm:id}/cdroms
- GET /vms/{vm:id}/cdroms
- GET /vms/{vm:id}/cdroms/{cdrom:id}
- PUT /vms/{vm:id}/cdroms/{cdrom:id}
- POST /vms/{vm:id}/clone
- POST /vms/{vm:id}/commitsnapshot
- POST /vms/{vm:id}/detach
- POST /vms/{vm:id}/diskattachments
- GET /vms/{vm:id}/diskattachments
- GET /vms/{vm:id}/diskattachments/{attachment.id}
- DELETE /vms/{vm:id}/diskattachments/{attachment.id}
- PUT /vms/{vm:id}/diskattachments/{attachment.id}
- POST /vms/{vm:id}/export
- POST /vms/{vm:id}/freezefilesystems
- POST /vms/{vm:id}/graphicsconsoles
- GET /vms/{vm:id}/graphicsconsoles
- GET /vms/{vm:id}/graphicsconsoles/{console:id}
- DELETE /vms/{vm:id}/graphicsconsoles/{console:id}
- POST /vms/{vm:id}/graphicsconsoles/{console:id}/proxyticket
- POST /vms/{vm:id}/graphicsconsoles/{console:id}/remoteviewerconnectionfile
- POST /vms/{vm:id}/graphicsconsoles/{console:id}/ticket
- POST /vms/{vm:id}/hostdevices
- GET /vms/{vm:id}/hostdevices
- GET /vms/{vm:id}/hostdevices/{device:id}
- DELETE /vms/{vm:id}/hostdevices/{device:id}
- GET /vms/{vm:id}/katelloerrata
- GET /vms/{vm:id}/katelloerrata/{katelloerratum:id}
- POST /vms/{vm:id}/logon
- POST /vms/{vm:id}/maintenance
- POST /vms/{vm:id}/migrate
- POST /vms/{vm:id}/nics
- GET /vms/{vm:id}/nics
- GET /vms/{vm:id}/nics/{nic:id}
- PUT /vms/{vm:id}/nics/{nic:id}
- DELETE /vms/{vm:id}/nics/{nic:id}
- POST /vms/{vm:id}/nics/{nic:id}/activate
- POST /vms/{vm:id}/nics/{nic:id}/deactivate
- GET /vms/{vm:id}/nics/{nic:id}/networkfilterparameters
- POST /vms/{vm:id}/nics/{nic:id}/networkfilterparameters
- GET /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}
- PUT /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}
- DELETE /vms/{vm:id}/nics/{nic:id}/networkfilterparameters/{parameter:id}
- GET /vms/{vm:id}/nics/{nic:id}/reporteddevices
- GET /vms/{vm:id}/nics/{nic:id}/reporteddevices/{reporteddevice:id}
- GET /vms/{vm:id}/nics/{nic:id}/statistics
- GET /vms/{vm:id}/nics/{nic:id}/statistics/{statistic:id}
- POST /vms/{vm:id}/humanodes
- GET /vms/{vm:id}/humanodes
- GET /vms/{vm:id}/humanodes/{node:id}
- PUT /vms/{vm:id}/humanodes/{node:id}
- DELETE /vms/{vm:id}/humanodes/{node:id}
- POST /vms/{vm:id}/permissions
- GET /vms/{vm:id}/permissions
- GET /vms/{vm:id}/permissions/{permission:id}
- DELETE /vms/{vm:id}/permissions/{permission:id}
- POST /vms/{vm:id}/previewsnapshot
- POST /vms/{vm:id}/reboot
- POST /vms/{vm:id}/reordermacaddresses
- GET /vms/{vm:id}/reporteddevices
- GET /vms/{vm:id}/reporteddevices/{reporteddevice:id}
- GET /vms/{vm:id}/sessions
- GET /vms/{vm:id}/sessions/{session:id}
- POST /vms/{vm:id}/shutdown
- POST /vms/{vm:id}/snapshots
- GET /vms/{vm:id}/snapshots
- GET /vms/{vm:id}/snapshots/{snapshot:id}
- DELETE /vms/{vm:id}/snapshots/{snapshot:id}
- GET /vms/{vm:id}/snapshots/{snapshot:id}/cdroms
- GET /vms/{vm:id}/snapshots/{snapshot:id}/cdroms/{cdrom:id}
- GET /vms/{vm:id}/snapshots/{snapshot:id}/disks
- GET /vms/{vm:id}/snapshots/{snapshot:id}/disks/{disk:id}
- GET /vms/{vm:id}/snapshots/{snapshot:id}/nics
- GET /vms/{vm:id}/snapshots/{snapshot:id}/nics/{nic:id}
- POST /vms/{vm:id}/snapshots/{snapshot:id}/restore
- POST /vms/{vm:id}/start
- GET /vms/{vm:id}/statistics
- GET /vms/{vm:id}/statistics/{statistic:id}
- POST /vms/{vm:id}/stop
- POST /vms/{vm:id}/suspend
- POST /vms/{vm:id}/tags
- GET /vms/{vm:id}/tags
- GET /vms/{vm:id}/tags/{tag:id}
- DELETE /vms/{vm:id}/tags/{tag:id}
- POST /vms/{vm:id}/thawfilesystems
- POST /vms/{vm:id}/ticket
- POST /vms/{vm:id}/undosnapshot
- POST /vms/{vm:id}/watchdogs
- GET /vms/{vm:id}/watchdogs
- GET /vms/{vm:id}/watchdogs/{watchdog:id}
- PUT /vms/{vm:id}/watchdogs/{watchdog:id}
- DELETE /vms/{vm:id}/watchdogs/{watchdog:id}
- POST /vnicprofiles
- GET /vnicprofiles
- GET /vnicprofiles/{profile:id}
- PUT /vnicprofiles/{profile:id}
- DELETE /vnicprofiles/{profile:id}
- POST /vnicprofiles/{profile:id}/permissions
- GET /vnicprofiles/{profile:id}/permissions
- GET /vnicprofiles/{profile:id}/permissions/{permission:id}
- DELETE /vnicprofiles/{profile:id}/permissions/{permission:id}
CHAPTER 6. SERVICES

This section enumerates all the services that are available in the API.

6.1. AFFINITYGROUP

This service manages a single affinity group.

Table 6.1. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieve the affinity group details.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the affinity group.</td>
</tr>
<tr>
<td>update</td>
<td>Update the affinity group.</td>
</tr>
</tbody>
</table>

6.1.1. get GET

Retrieve the affinity group details.

```
<affinity_group id="00000000-0000-0000-0000-000000000000"
    <name>AF_GROUP_001</name>
    <cluster id="00000000-0000-0000-0000-000000000000"/>
    <positive>true</positive>
    <enforcing>true</enforcing>
</affinity_group>
```

Table 6.2. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>group</td>
<td>AffinityGroup</td>
<td>Out</td>
<td>The affinity group.</td>
</tr>
</tbody>
</table>

6.1.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.1.2. remove DELETE

Remove the affinity group.

```
DELETE /ovirt-engine/api/clusters/000-000/affinitygroups/123-456
```

Table 6.3. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the removal should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.1.3. update PUT
Update the affinity group.

**Table 6.4. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>AffinityGroup</td>
<td>In/Out</td>
<td>The affinity group.</td>
</tr>
</tbody>
</table>

### 6.2. AFFINITYGROUPVM
This service manages a single virtual machine to affinity group assignment.

**Table 6.5. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>remove</td>
<td>Remove this virtual machine from the affinity group.</td>
</tr>
</tbody>
</table>

### 6.2.1. remove DELETE
Remove this virtual machine from the affinity group.

**Table 6.6. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the removal should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.3. AFFINITYGROUPVMS
This service manages a collection of all the virtual machines assigned to an affinity group.

**Table 6.7. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
</table>
6.3.1. add POST

Add a virtual machine to the affinity group.

For example, to add the virtual machine 789 to the affinity group 456 of cluster 123, send a request like this:

```
POST /ovirt-engine/api/clusters/123/affinitygroups/456/vms
```

With the following body:

```
<vm id="789"/>
```

Table 6.8. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.3.2. list GET

List all virtual machines assigned to this affinity group.

The order of the returned virtual machines isn't guaranteed.

Table 6.9. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of virtual machines to return.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.3.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.3.2.2. max
Sets the maximum number of virtual machines to return. If not specified, all the virtual machines are returned.

### 6.4. AFFINITYGROUPS

The affinity groups service manages virtual machine relationships and dependencies.

**Table 6.10. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Create a new affinity group.</td>
</tr>
<tr>
<td>list</td>
<td>List existing affinity groups.</td>
</tr>
</tbody>
</table>

#### 6.4.1. add POST

Create a new affinity group.

Post a request like in the example below to create a new affinity group:

```
POST /ovirt-engine/api/clusters/000-000/affinitygroups
```

And use the following example in its body:

```
<affinity_group>
  <name>AF_GROUP_001</name>
  <hosts_rule>
    <enforcing>true</enforcing>
    <positive>true</positive>
  </hosts_rule>
  <vms_rule>
    <enabled>false</enabled>
  </vms_rule>
</affinity_group>
```

**Table 6.11. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>AffinityGroup</td>
<td>In/Out</td>
<td>The affinity group object to create.</td>
</tr>
</tbody>
</table>

#### 6.4.2. list GET

List existing affinity groups.

The order of the affinity groups results isn’t guaranteed.

**Table 6.12. Parameters summary**
6.4.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.4.2.2. max

Sets the maximum number of affinity groups to return. If not specified all the affinity groups are returned.

6.5. AFFINITYLABEL

The details of a single affinity label.

Table 6.13. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the details of a label.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a label from the system and clears all assignments of the removed label.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a label.</td>
</tr>
</tbody>
</table>

6.5.1. get GET

Retrieves the details of a label.

Table 6.14. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>label</td>
<td>AffinityLabel</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.5.2. remove DELETE

Removes a label from the system and clears all assignments of the removed label.

### 6.5.3. update PUT

Updates a label. This call will update all metadata, such as the name or description.

#### Table 6.15. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>AffinityLabel</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.6. AFFINITYLABELHOST

This service represents a host that has a specific label when accessed through the affinitylabels/hosts subcollection.

#### Table 6.16. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves details about a host that has this label assigned.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove a label from a host.</td>
</tr>
</tbody>
</table>

### 6.6.1. get GET

Retrieves details about a host that has this label assigned.

#### Table 6.17. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.6.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.6.2. remove DELETE

Remove a label from a host.
6.7. AFFINITYLABELHOSTS

This service represents list of hosts that have a specific label when accessed through the affinitylabels-hosts subcollection.

Table 6.18. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a label to a host.</td>
</tr>
<tr>
<td>list</td>
<td>List all hosts with the label.</td>
</tr>
</tbody>
</table>

6.7.1. add POST

Add a label to a host.

Table 6.19. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.7.2. list GET

List all hosts with the label.

The order of the returned hosts isn’t guaranteed.

Table 6.20. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>hosts</td>
<td>Host[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.7.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.8. AFFINITYLABELVM

This service represents a vm that has a specific label when accessed through the affinitylabels/vms subcollection.

Table 6.21. Methods summary
### Table 6.22. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.8.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.8.2. remove DELETE

Remove a label from a vm.

### 6.9. AFFINITYLABELVMS

This service represents list of vms that have a specific label when accessed through the affinitylabels/vms subcollection.

#### Table 6.23. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a label to a vm.</td>
</tr>
<tr>
<td>list</td>
<td>List all virtual machines with the label.</td>
</tr>
</tbody>
</table>

#### 6.9.1. add POST

Add a label to a vm.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vm</code></td>
<td>Vm</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.9.2. list GET

List all virtual machines with the label.

The order of the returned virtual machines isn’t guaranteed.

#### Table 6.25. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>follow</code></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <code>followed</code>.</td>
</tr>
<tr>
<td><code>vms</code></td>
<td>Vm[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.9.2.1. follow

Indicates which inner links should be `followed`. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.10. AFFINITYLABELS

Manages the affinity labels available in the system.

#### Table 6.26. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>add</code></td>
<td>Creates a new label.</td>
</tr>
<tr>
<td><code>list</code></td>
<td>Lists all labels present in the system.</td>
</tr>
</tbody>
</table>

#### 6.10.1. add POST

Creates a new label. The label is automatically attached to all entities mentioned in the vms or hosts lists.

#### Table 6.27. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>label</code></td>
<td>AffinityLabel</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.10.2. list GET

Lists all labels present in the system.
The order of the returned labels isn’t guaranteed.

### Table 6.28. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>labels</td>
<td>AffinityLabel[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of labels to return.</td>
</tr>
</tbody>
</table>

#### 6.10.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.10.2.2. max

Sets the maximum number of labels to return. If not specified all the labels are returned.

### 6.11. AREA

This annotation is intended to specify what oVirt area is the annotated concept related to. Currently the following areas are in use, and they are closely related to the oVirt teams, but not necessarily the same:

- Infrastructure
- Network
- SLA
- Storage
- Virtualization

A concept may be associated to more than one area, or to no area.

The value of this annotation is intended for reporting only, and it doesn’t affect at all the generated code or the validity of the model.

### 6.12. ASSIGNEDAFFINITYLABEL

This service represents one label to entity assignment when accessed using the entities/affinitylabels subcollection.

### Table 6.29. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves details about the attached label.</td>
</tr>
</tbody>
</table>
6.12.1. get GET

Retrieves details about the attached label.

Table 6.30. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>label</td>
<td>AffinityLabel</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.12.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.12.2. remove DELETE

Removes the label from an entity. Does not touch the label itself.

6.13. ASSIGNEDAFFINITYLABELS

This service is used to list and manipulate affinity labels that are assigned to supported entities when accessed using entities/affinitylabels.

Table 6.31. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Attaches a label to an entity.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all labels that are attached to an entity.</td>
</tr>
</tbody>
</table>

6.13.1. add POST

Attaches a label to an entity.

Table 6.32. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>AffinityLabel</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>
6.13.2. list GET

Lists all labels that are attached to an entity.

The order of the returned entities isn’t guaranteed.

Table 6.33. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>label</td>
<td>AffinityLabel[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.13.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.14. ASSIGNEDCPUPROFILE

Table 6.34. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.14.1. get GET

Table 6.35. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>profile</td>
<td>CpuProfile</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.14.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.14.2. remove DELETE

Table 6.36. Parameters summary
### 6.15. ASSIGNEDCPUProfiles

**Table 6.37. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new cpu profile for the cluster.</td>
</tr>
<tr>
<td>list</td>
<td>List the CPU profiles assigned to the cluster.</td>
</tr>
</tbody>
</table>

#### 6.15.1. add POST

Add a new cpu profile for the cluster.

**Table 6.38. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>CpuProfile</td>
<td>In/Out</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.15.2. list GET

List the CPU profiles assigned to the cluster.

The order of the returned CPU profiles isn't guaranteed.

**Table 6.39. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of profiles to return.</td>
</tr>
<tr>
<td>profiles</td>
<td>CpuProfile[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.15.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.15.2.2. max
6.16. ASSIGNEDDISKPROFILE

Table 6.40. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.16.1. get GET

Table 6.41. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.16.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.16.2. remove DELETE

Table 6.42. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.17. ASSIGNEDDISKPROFILES

Table 6.43. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new disk profile for the storage domain.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of disk profiles assigned to the storage domain.</td>
</tr>
</tbody>
</table>
6.17.1. add POST

Add a new disk profile for the storage domain.

Table 6.44. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>DiskProfile</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.17.2. list GET

Returns the list of disk profiles assigned to the storage domain.

The order of the returned disk profiles isn’t guaranteed.

Table 6.45. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of profiles to return.</td>
</tr>
<tr>
<td>profiles</td>
<td>DiskProfile[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.17.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.17.2.2. max

Sets the maximum number of profiles to return. If not specified all the profiles are returned.

6.18. ASSIGNEDPERMISSIONS

Represents a permission sub-collection, scoped by user, group or some entity type.

Table 6.46. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Assign a new permission to a user or group for specific entity.</td>
</tr>
<tr>
<td>list</td>
<td>List all the permissions of the specific entity.</td>
</tr>
</tbody>
</table>

6.18.1. add POST

Assign a new permission to a user or group for specific entity.
For example, to assign the **UserVmManager** role to the virtual machine with id **123** to the user with id **456** send a request like this:

```
POST /ovirt-engine/api/vms/123/permissions
```

With a request body like this:

```
<permission>
  <role>
    <name>UserVmManager</name>
  </role>
  <user id="456"/>
</permission>
```

To assign the **SuperUser** role to the system to the user with id **456** send a request like this:

```
POST /ovirt-engine/api/permissions
```

With a request body like this:

```
<permission>
  <role>
    <name>SuperUser</name>
  </role>
  <user id="456"/>
</permission>
```

If you want to assign permission to the group instead of the user please replace the `user` element with the `group` element with proper `id` of the group. For example to assign the **UserRole** role to the cluster with id **123** to the group with id **789** send a request like this:

```
POST /ovirt-engine/api/clusters/123/permissions
```

With a request body like this:

```
<permission>
  <role>
    <name>UserRole</name>
  </role>
  <group id="789"/>
</permission>
```

### Table 6.47. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permission</td>
<td>Permission</td>
<td>In/Out</td>
<td>The permission.</td>
</tr>
</tbody>
</table>

**6.18.2. list GET**
List all the permissions of the specific entity.

For example to list all the permissions of the cluster with id **123** send a request like this:

```plaintext
GET /ovirt-engine/api/clusters/123/permissions
```

The order of the returned permissions isn’t guaranteed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Out</td>
<td>The list of permissions.</td>
</tr>
</tbody>
</table>

**6.18.2.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

**6.19. ASSIGNEDROLES**

Represents a roles sub-collection, for example scoped by user.

**Table 6.49. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the roles assigned to the permission.</td>
</tr>
</tbody>
</table>

**6.19.1. list GET**

Returns the roles assigned to the permission.

The order of the returned roles isn’t guaranteed.
### Table 6.50. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of roles to return.</td>
</tr>
<tr>
<td>roles</td>
<td>Role[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.19.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.19.1.2. max

Sets the maximum number of roles to return. If not specified all the roles are returned.

### 6.20. ASSIGNEDTAG

A service to manage assignment of specific tag to specific entities in system.

### Table 6.51. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the information about the assigned tag.</td>
</tr>
<tr>
<td>remove</td>
<td>Unassign tag from specific entity in the system.</td>
</tr>
</tbody>
</table>

### 6.20.1. get GET

Gets the information about the assigned tag.

For example to retrieve the information about the tag with the id **456** which is assigned to virtual machine with id **123** send a request like this:

```
GET /ovirt-engine/api/vms/123/tags/456
```

```
<tag href="/ovirt-engine/api/tags/456" id="456">
  <name>root</name>
  <description>root</description>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</tag>
```

### Table 6.52. Parameters summary
### 6.20.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.20.2. remove DELETE

Unassign tag from specific entity in the system.

For example to unassign the tag with id **456** from virtual machine with id **123** send a request like this:

```
DELETE /ovirt-engine/api/vms/123/tags/456
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.21. ASSIGNEDTAGS

A service to manage collection of assignment of tags to specific entities in system.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Assign tag to specific entity in the system.</td>
</tr>
<tr>
<td>list</td>
<td>List all tags assigned to the specific entity.</td>
</tr>
</tbody>
</table>

### 6.21.1. add POST

Assign tag to specific entity in the system.

For example to assign tag **mytag** to virtual machine with the id **123** send a request like this:

```
POST /ovirt-engine/api/vms/123/tags
```

With a request body like this:
Table 6.55. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>Tag</td>
<td>In/Out</td>
<td>The assigned tag.</td>
</tr>
</tbody>
</table>

6.21.2. list GET

List all tags assigned to the specific entity.

For example to list all the tags of the virtual machine with id 123 send a request like this:

```bash
GET /ovirt-engine/api/vms/123/tags
```

The order of the returned tags isn’t guaranteed.

Table 6.56. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of tags to return.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>Out</td>
<td>The list of assigned tags.</td>
</tr>
</tbody>
</table>

6.21.2.1. follow

 Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.21.2.2. max

Sets the maximum number of tags to return. If not specified all the tags are returned.

6.22. ASSIGNEDVNICPROFILE

Table 6.57. Methods summary
6.22.1. get GET

Table 6.58. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>profile</td>
<td>VnicProfile</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.22.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.22.2. remove DELETE

Table 6.59. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.23. ASSIGNEDVNICPROFILES

Table 6.60. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new virtual network interface card profile for the network.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of VNIC profiles assigned to the network.</td>
</tr>
</tbody>
</table>

6.23.1. add POST

Add a new virtual network interface card profile for the network.
6.23.2. list GET

Returns the list of VNIC profiles assigned to the network.

The order of the returned VNIC profiles isn't guaranteed.

Table 6.62. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of profiles to return.</td>
</tr>
<tr>
<td>profiles</td>
<td>VnicProfile[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.23.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.23.2.2. max

Sets the maximum number of profiles to return. If not specified all the profiles are returned.

6.24. ATTACHEDSTORAGEDOMAIN

Table 6.63. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate</td>
<td>This operation activates an attached storage domain.</td>
</tr>
<tr>
<td>deactivate</td>
<td>This operation deactivates an attached storage domain.</td>
</tr>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.24.1. activate POST

This operation activates an attached storage domain. Once the storage domain is activated it is ready for use with the data center.
POST /ovirt-engine/api/datacenters/123/storagedomains/456/activate

The activate action does not take any action specific parameters, so the request body should contain an empty action:

\<action/>\n
Table 6.64. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the activation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.24.2. deactivate POST

This operation deactivates an attached storage domain. Once the storage domain is deactivated it will not be used with the data center. For example, to deactivate storage domain 456, send the following request:

POST /ovirt-engine/api/datacenters/123/storagedomains/456/deactivate

With a request body like this:

\<action/>\n
If the force parameter is true then the operation will succeed, even if the OVF update which takes place before the deactivation of the storage domain failed. If the force parameter is false and the OVF update failed, the deactivation of the storage domain will also fail.

Table 6.65. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the deactivation should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the operation should succeed and the storage domain should be moved to a deactivated state, even if the OVF update for the storage domain failed.</td>
</tr>
</tbody>
</table>

6.24.2.1. force

Indicates if the operation should succeed and the storage domain should be moved to a deactivated state, even if the OVF update for the storage domain failed. For example, to deactivate storage domain 456 using force flag, send the following request:

POST /ovirt-engine/api/datacenters/123/storagedomains/456/deactivate
With a request body like this:

```xml
<action>
<force>true</force>
<action>
```

This parameter is optional, and the default value is `false`.

### 6.24.3. get GET

**Table 6.66. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.24.3.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.24.4. remove DELETE

**Table 6.67. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.25. ATTACHEDSTORAGEDOMAINDISK

Manages a single disk available in a storage domain attached to a data center.

**IMPORTANT**

Since version 4.2 of the engine this service is intended only to list disks available in the storage domain, and to register unregistered disks. All the other operations, like copying a disk, moving a disk, etc, have been deprecated and will be removed in the future. To perform those operations use the service that manages all the disks of the system, or the service that manages an specific disk.

**Table 6.68. Methods summary**
### copy POST

Copies a disk to the specified storage domain.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To copy a disk use the `copy` operation of the service that manages that disk.

#### Table 6.69. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In</td>
<td>Description of the resulting disk.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td>The storage domain where the new disk will be created.</td>
</tr>
</tbody>
</table>

### export POST

Exports a disk to an export storage domain.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To export a disk use the `export` operation of the service that manages that disk.

#### Table 6.70. Parameters summary
### 6.25.3. get GET

Retrieves the description of the disk.

**Table 6.71. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td>The description of the disk.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.25.3.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.25.4. move POST

Moves a disk to another storage domain.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To move a disk use the move operation of the service that manages that disk.

**Table 6.72. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the move should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td>The storage domain where the disk will be moved to.</td>
</tr>
</tbody>
</table>

### 6.25.5. register POST

Registers an unregistered disk.
6.25.6. remove DELETE

Removes a disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To remove a disk use the `remove` operation of the service that manages that disk.

6.25.7. sparsify POST

Sparsify the disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To remove a disk use the `remove` operation of the service that manages that disk.

6.25.8. update PUT

Updates the disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To update a disk use the `update` operation of the service that manages that disk.

### Table 6.73. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The update to apply to the disk.</td>
</tr>
</tbody>
</table>

6.26. ATTACHEDSTORAGEDOMAINDISKS

Manages the collection of disks available inside an storage domain that is attached to a data center.

### Table 6.74. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds or registers a disk.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieve the list of disks that are available in the storage domain.</td>
</tr>
</tbody>
</table>

6.26.1. add POST
Adds or registers a disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To add a new disk use the add operation of the service that manages the disks of the system. To register an unregistered disk use the register operation of the service that manages that disk.

### Table 6.75. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The disk to add or register.</td>
</tr>
<tr>
<td>unregistered</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if a new disk should be added or if an existing unregistered disk should be registered.</td>
</tr>
</tbody>
</table>

**6.26.1.1. unregistered**

Indicates if a new disk should be added or if an existing unregistered disk should be registered. If the value is `true` then the identifier of the disk to register needs to be provided. For example, to register the disk with id 456 send a request like this:

```plaintext
POST /ovirt-engine/api/storagedomains/123/disks?unregistered=true
```

With a request body like this:

```xml
<disk id="456"/>
```

If the value is `false` then a new disk will be created in the storage domain. In that case the `provisioned_size`, `format` and `name` attributes are mandatory. For example, to create a new `copy on write` disk of 1 GiB, send a request like this:

```plaintext
POST /ovirt-engine/api/storagedomains/123/disks
```

With a request body like this:

```xml
<disk>
  <name>mydisk</name>
  <format>cow</format>
  <provisioned_size>1073741824</provisioned_size>
</disk>
```

The default value is `false`.

**6.26.2. list GET**

Retrieve the list of disks that are available in the storage domain.

### Table 6.76. Parameters summary
### 6.26.2.1. disks

List of retrieved disks.

The order of the returned disks isn’t guaranteed.

#### 6.26.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.26.2.3. max

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.27. ATTACHEDSTORAGEDOMAINS

Manages the storage domains attached to a data center.

#### Table 6.77. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Attaches an existing storage domain to the data center.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of storage domains attached to the data center.</td>
</tr>
</tbody>
</table>

#### 6.27.1. add POST

Attaches an existing storage domain to the data center.

#### Table 6.78. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In/Out</td>
<td>The storage domain to attach to the data center.</td>
</tr>
</tbody>
</table>

#### 6.27.2. list GET

Returns the list of storage domains attached to the data center.
The order of the returned storage domains isn't guaranteed.

### Table 6.79. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of storage domains to return.</td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td>Out</td>
<td>A list of storage domains that are attached to the data center.</td>
</tr>
</tbody>
</table>

#### 6.27.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.27.2.2. max

Sets the maximum number of storage domains to return. If not specified all the storage domains are returned.

### 6.28. BALANCE

#### Table 6.80. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.28.1. get GET

#### Table 6.81. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance</td>
<td>Balance</td>
<td>Out</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.28.1.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.28.2. remove DELETE

Table 6.82. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.29. BALANCES

Table 6.83. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a balance module to a specified user defined scheduling policy.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of balance modules used by the scheduling policy.</td>
</tr>
</tbody>
</table>

6.29.1. add POST

Add a balance module to a specified user defined scheduling policy.

Table 6.84. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>balance</td>
<td>Balance</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.29.2. list GET

Returns the list of balance modules used by the scheduling policy.

The order of the returned balance modules isn’t guaranteed.

Table 6.85. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>balances</td>
<td>Balance[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.29.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.29.2.2. max

Sets the maximum number of balances to return. If not specified all the balances are returned.

6.30. BOOKMARK

A service to manage a bookmark.

Table 6.86. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get a bookmark.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove a bookmark.</td>
</tr>
<tr>
<td>update</td>
<td>Update a bookmark.</td>
</tr>
</tbody>
</table>

6.30.1. get GET

Get a bookmark.

An example for getting a bookmark:

```
GET /ovirt-engine/api/bookmarks/123
```

```
<bookmark href="/ovirt-engine/api/bookmarks/123" id="123">
  <name>example_vm</name>
  <value>vm: name=example*</value>
</bookmark>
```

Table 6.87. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmark</td>
<td>Bookmark</td>
<td>Out</td>
<td>The requested bookmark.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.30.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.30.2. remove DELETE

Remove a bookmark.

An example for removing a bookmark:

```
DELETE /ovirt-engine/api/bookmarks/123
```

Table 6.88. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.30.3. update PUT

Update a bookmark.

An example for updating a bookmark:

```
PUT /ovirt-engine/api/bookmarks/123
```

With the request body:

```
<bookmark>
  <name>new_example_vm</name>
  <value>vm: name=new_example*</value>
</bookmark>
```

Table 6.89. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>bookmark</td>
<td>Bookmark</td>
<td>In/Out</td>
<td>The updated bookmark.</td>
</tr>
</tbody>
</table>

6.31. BOOKMARKS

A service to manage bookmarks.

Table 6.90. Methods summary
### 6.31.1. add POST

Adding a new bookmark.

Example of adding a bookmark:

```plaintext
POST /ovirt-engine/api/bookmarks

<bookmark>
  <name>new_example_vm</name>
  <value>vm: name=new_example*</value>
</bookmark>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmark</td>
<td>Bookmark</td>
<td>In/Out</td>
<td>The added bookmark.</td>
</tr>
</tbody>
</table>

### 6.31.2. list GET

Listing all the available bookmarks.

Example of listing bookmarks:

```plaintext
GET /ovirt-engine/api/bookmarks

<bookmarks>
  <bookmark href="/ovirt-engine/api/bookmarks/123" id="123">
    <name>database</name>
    <value>vm: name=database*</value>
  </bookmark>
  <bookmark href="/ovirt-engine/api/bookmarks/456" id="456">
    <name>example</name>
    <value>vm: name=example*</value>
  </bookmark>
</bookmarks>
```

The order of the returned bookmarks isn’t guaranteed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookmark</td>
<td>Bookmark</td>
<td>In/Out</td>
<td>The added bookmark.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>bookmarks</td>
<td>Bookmark[]</td>
<td>Out</td>
<td>The list of available bookmarks.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of bookmarks to return.</td>
</tr>
</tbody>
</table>

### 6.31.2.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.31.2.2. max
Sets the maximum number of bookmarks to return. If not specified all the bookmarks are returned.

### 6.32. CLUSTER
A service to manage a specific cluster.

#### Table 6.93. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets information about the cluster.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the cluster from the system.</td>
</tr>
<tr>
<td>resetemulate dmachine</td>
<td>Synchronizes all networks on the cluster.</td>
</tr>
<tr>
<td>syncallnetworks</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Updates information about the cluster.</td>
</tr>
</tbody>
</table>

### 6.32.1. get GET
Gets information about the cluster.

An example of getting a cluster:

```
GET /ovirt-engine/api/clusters/123
```

```xml
<cluster href="/ovirt-engine/api/clusters/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/clusters/123/resetemulatedmachine"
```
rel="resetemulatedmachine"/>
</actions>

<name>Default</name>
<description>The default server cluster</description>
<link href="/ovirt-engine/api/clusters/123/networks" rel="networks"/>
<link href="/ovirt-engine/api/clusters/123/permissions" rel="permissions"/>
<link href="/ovirt-engine/api/clusters/123/glustervolumes" rel="glustervolumes"/>
<link href="/ovirt-engine/api/clusters/123/glusterhooks" rel="glusterhooks"/>
<link href="/ovirt-engine/api/clusters/123/affinitygroups" rel="affinitygroups"/>
<link href="/ovirt-engine/api/clusters/123/cpuprofiles" rel="cpuprofiles"/>

<ballooning_enabled>false</ballooning_enabled>
<cpu>
  <architecture>x86_64</architecture>
  <type>Intel Penryn Family</type>
</cpu>
<error_handling>
  <on_error>migrate</on_error>
</error_handling>
<fencing_policy>
  <enabled>true</enabled>
  <skip_if_connectivity_broken>
    <enabled>false</enabled>
    <threshold>50</threshold>
  </skip_if_connectivity_broken>
  <skip_if_sd_active>
    <enabled>false</enabled>
  </skip_if_sd_active>
</fencing_policy>
<gluster_service>false</gluster_service>
<ha_reservation>false</ha_reservation>
<ksm>
  <enabled>true</enabled>
  <merge_across_nodes>true</merge_across_nodes>
</ksm>
<maintenance_reason_required>false</maintenance_reason_required>
<memory_policy>
  <over_commit>
    <percent>100</percent>
  </over_commit>
  <transparent_hugepages>
    <enabled>true</enabled>
  </transparent_hugepages>
</memory_policy>
<migration>
  <auto_converge>inherit</auto_converge>
  <bandwidth>
    <assignment_method>auto</assignment_method>
  </bandwidth>
  <compressed>inherit</compressed>
</migration>
<optional_reason>false</optional_reason>
<required_rng_sources>
  <required_rng_source>random</required_rng_source>
</required_rng_sources>
<scheduling_policy href="/ovirt-engine/api/schedulingpolicies/456" id="456"/>
<threads_as_cores>false</threads_as_cores>
<trusted_service>false</trusted_service>
<tunnel_migration>false</tunnel_migration>
</version>
<virt_service>true</virt_service>
<data_center href="/ovirt-engine/api/datacenters/111" id="111"/>
</cluster>

Table 6.94. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.32.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.32.2. remove DELETE

Removes the cluster from the system.

DELETE /ovirt-engine/api/clusters/00000000-0000-0000-0000-000000000000

Table 6.95. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.32.3. resetemulatedmachine POST

Table 6.96. Parameters summary
Table 6.97. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.32.4. syscallnetworks POST

Synchronizes all networks on the cluster.

POST /ovirt-engine/api/clusters/123/syscallnetworks

With a request body like this:

```xml
<action/>
```

Table 6.97. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.32.5. update PUT

Updates information about the cluster.

Only the specified fields are updated; others remain unchanged.

For example, to update the cluster’s CPU:

PUT /ovirt-engine/api/clusters/123

With a request body like this:

```xml
<cluster>
  <cpu>
    <type>Intel Haswell-noTSX Family</type>
  </cpu>
</cluster>
```

Table 6.98. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>
6.33. CLUSTERENABLEDFEATURE

Represents a feature enabled for the cluster.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Provides the information about the cluster feature enabled.</td>
</tr>
<tr>
<td>remove</td>
<td>Disables a cluster feature.</td>
</tr>
</tbody>
</table>

6.33.1. get GET

Provides the information about the cluster feature enabled.

For example, to find details of the enabled feature 456 for cluster 123, send a request like this:

GET /ovirt-engine/api/clusters/123/enabledfeatures/456

That will return a ClusterFeature object containing the name:

```xml
<cluster_feature id="456">
  <name>libgfapi_supported</name>
</cluster_feature>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature</td>
<td>ClusterFeature</td>
<td>Out</td>
<td>Retrieved cluster feature that's enabled.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.33.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.33.2. remove DELETE

Disables a cluster feature.

For example, to disable the feature 456 of cluster 123 send a request like this:

DELETE /ovirt-engine/api/clusters/123/enabledfeatures/456

6.34. CLUSTERENABLEDFEATURES
Provides information about the additional features that are enabled for this cluster. The features that are enabled are the available features for the cluster level.

Table 6.101. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Enable an additional feature for a cluster.</td>
</tr>
<tr>
<td>list</td>
<td>Lists the additional features enabled for the cluster.</td>
</tr>
</tbody>
</table>

6.34.1. add POST

Enable an additional feature for a cluster.

For example, to enable a feature 456 on cluster 123, send a request like this:

```
POST /ovirt-engine/api/clusters/123/enabledfeatures
```

The request body should look like this:

```
<cluster_feature id="456"/>
```

Table 6.102. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature</td>
<td>ClusterFeature</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.34.2. list GET

Lists the additional features enabled for the cluster.

For example, to get the features enabled for cluster 123 send a request like this:

```
GET /ovirt-engine/api/clusters/123/enabledfeatures
```

This will return a list of features:

```
<enabled_features>
  <cluster_feature id="123">
    <name>test_feature</name>
  </cluster_feature>
  ...
</enabled_features>
```

Table 6.103. Parameters summary
### 6.34. CLUSTERFEATURE

**features**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>features</td>
<td>ClusterFeature[]</td>
<td>Out</td>
<td>Retrieved features.</td>
</tr>
</tbody>
</table>

**follow**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.34.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.35. CLUSTEREXTERNALPROVIDERS

This service lists external providers.

**Table 6.104. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of external providers.</td>
</tr>
</tbody>
</table>

#### 6.35.1. list GET

Returns the list of external providers.

The order of the returned list of providers is not guaranteed.

**Table 6.105. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>providers</td>
<td>ExternalProvider[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.35.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.36. CLUSTERFEATURE

Represents a feature enabled for the cluster level

**Table 6.106. Methods summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Provides the information about the a cluster feature supported by a cluster level.</td>
</tr>
</tbody>
</table>

### 6.36.1. get GET

Provides the information about the a cluster feature supported by a cluster level.

For example, to find details of the cluster feature 456 for cluster level 4.1, send a request like this:

```
GET /ovirt-engine/api/clusterlevels/4.1/clusterfeatures/456
```

That will return a `ClusterFeature` object containing the name:

```
<cluster_feature id="456">
  <name>libgfapi_supported</name>
</cluster_feature>
```

**Table 6.107. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature</td>
<td>ClusterFeature</td>
<td>Out</td>
<td>Retrieved cluster feature.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.36.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.37. CLUSTERFEATURES

Provides information about the cluster features that are supported by a cluster level.

**Table 6.108. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Lists the cluster features supported by the cluster level.</td>
</tr>
</tbody>
</table>

#### 6.37.1. list GET

Lists the cluster features supported by the cluster level.

```
GET /ovirt-engine/api/clusterlevels/4.1/clusterfeatures
```

This will return a list of cluster features supported by the cluster level:
Table 6.109. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>features</td>
<td>ClusterFeature[]</td>
<td>Out</td>
<td>Retrieved features.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

### 6.37.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.38. CLUSTERLEVEL

Provides information about a specific cluster level. See the ClusterLevels service for more information.

Table 6.110. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Provides the information about the capabilities of the specific cluster level managed by this service.</td>
</tr>
</tbody>
</table>

### 6.38.1. get GET

Provides the information about the capabilities of the specific cluster level managed by this service.

For example, to find what CPU types are supported by level 3.6 you can send a request like this:

```
GET /ovirt-engine/api/clusterlevels/3.6
```

That will return a ClusterLevel object containing the supported CPU types, and other information which describes the cluster level:

```
<cluster_level id="3.6">
  <cpu_types>
    <cpu_type>
      <name>Intel Conroe Family</name>
      <level>3</level>
      <architecture>x86_64</architecture>
    </cpu_type>
    ...
  </cpu_types>
</cluster_level>
```
Table 6.111. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>level</td>
<td>ClusterLevel</td>
<td>Out</td>
<td>Retrieved cluster level.</td>
</tr>
</tbody>
</table>

6.38.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.39. CLUSTERLEVELS

Provides information about the capabilities of different cluster levels supported by the engine. Version 4.0 of the engine supports levels 4.0 and 3.6. Each of these levels support different sets of CPU types, for example. This service provides that information.

Table 6.112. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Lists the cluster levels supported by the system.</td>
</tr>
</tbody>
</table>

6.39.1. list GET

Lists the cluster levels supported by the system.

GET /ovirt-engine/api/clusterlevels

This will return a list of available cluster levels.

The order of the returned cluster levels isn't guaranteed.
Table 6.113. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>levels</td>
<td>ClusterLevel[ ]</td>
<td>Out</td>
<td>Retrieved cluster levels.</td>
</tr>
</tbody>
</table>

6.39.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.40. CLUSTERNETWORK

A service to manage a specific cluster network.

Table 6.114. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the cluster network details.</td>
</tr>
<tr>
<td>remove</td>
<td>Unassigns the network from a cluster.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the network in the cluster.</td>
</tr>
</tbody>
</table>

6.40.1. get GET

Retrieves the cluster network details.

Table 6.115. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>Out</td>
<td>The cluster network.</td>
</tr>
</tbody>
</table>

6.40.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.40.2. remove DELETE

Unassigns the network from a cluster.
6.40.3. update PUT

Updates the network in the cluster.

Table 6.116. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>In/Out</td>
<td>The cluster network.</td>
</tr>
</tbody>
</table>

6.41. CLUSTERNETWORKS

A service to manage cluster networks.

Table 6.117. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Assigns the network to a cluster.</td>
</tr>
<tr>
<td>list</td>
<td>Lists the networks that are assigned to the cluster.</td>
</tr>
</tbody>
</table>

6.41.1. add POST

Assigns the network to a cluster.

Post a request like in the example below to assign the network to a cluster:

```
POST /ovirt-engine/api/clusters/123/networks
```

Use the following example in its body:

```
<network id="123" />
```

Table 6.118. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>In/Out</td>
<td>The network object to be assigned to the cluster.</td>
</tr>
</tbody>
</table>

6.41.2. list GET

Lists the networks that are assigned to the cluster.

The order of the returned clusters isn’t guaranteed.

Table 6.119. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>networks</td>
<td>Network[]</td>
<td>Out</td>
<td>The list of networks that are assigned to the cluster.</td>
</tr>
</tbody>
</table>

**6.41.2.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.41.2.2. max**

Sets the maximum number of networks to return. If not specified, all the networks are returned.

### 6.42. CLUSTERS

A service to manage clusters.

#### Table 6.120. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new cluster.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of clusters of the system.</td>
</tr>
</tbody>
</table>

#### 6.42.1. add POST

Creates a new cluster.

This requires the *name*, *cpu.type*, and *data_center* attributes. Identify the data center with either the *id* or *name* attribute.

```
POST /ovirt-engine/api/clusters
```

With a request body like this:

```
<cluster>
  <name>mycluster</name>
  <cpu>
    <type>Intel Penryn Family</type>
  </cpu>
  <data_center id="123"/>
</cluster>
```
To create a cluster with an external network provider to be deployed on every host that is added to the cluster, send a request like this:

```
POST /ovirt-engine/api/clusters
```

With a request body containing a reference to the desired provider:

```xml
<cluster>
  <name>mycluster</name>
  <cpu>
    <type>Intel Penryn Family</type>
  </cpu>
  <data_center id="123"/>
  <external_network_providers>
    <external_provider name="ovirt-provider-ovn"/>
  </external_network_providers>
</cluster>
```

### Table 6.121. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.42.2. list GET

Returns the list of clusters of the system.

The order of the returned clusters is guaranteed only if the `sortby` clause is included in the `search` parameter.

### Table 6.122. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search should be performed taking case into account.</td>
</tr>
<tr>
<td>clusters</td>
<td>Cluster[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of clusters to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned clusters.</td>
</tr>
</tbody>
</table>

### 6.42.2.1. case_sensitive
Indicates if the search should be performed taking case into account. The default value is `true`, which means that case is taken into account. To search ignoring case, set it to `false`.

### 6.42.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.42.2.3. max

Sets the maximum number of clusters to return. If not specified, all the clusters are returned.

### 6.43. COPYABLE

#### Table 6.123. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.43.1. copy POST

#### Table 6.124. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the copy should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.44. CPUPROFILE

#### Table 6.125. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update the specified cpu profile in the system.</td>
</tr>
</tbody>
</table>

#### 6.44.1. get GET

#### Table 6.126. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.44.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.44.2. remove DELETE

Table 6.127. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.44.3. update PUT

Update the specified cpu profile in the system.

Table 6.128. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>profile</td>
<td>CpuProfile</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.45. CPUPROFILES

Table 6.129. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new cpu profile to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of CPU profiles of the system.</td>
</tr>
</tbody>
</table>

6.45.1. add POST

Add a new cpu profile to the system.

Table 6.130. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>CpuProfile</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.45.2. list GET

Returns the list of CPU profiles of the system.

The order of the returned list of CPU profiles isn’t guaranteed.

<table>
<thead>
<tr>
<th>Table 6.131. Parameters summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>follow</td>
</tr>
<tr>
<td>max</td>
</tr>
<tr>
<td>profile</td>
</tr>
</tbody>
</table>

#### 6.45.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.45.2.2. max

Sets the maximum number of profiles to return. If not specified all the profiles are returned.

### 6.46. DATACENTER

A service to manage a data center.

<table>
<thead>
<tr>
<th>Table 6.132. Methods summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>get</td>
</tr>
<tr>
<td>remove</td>
</tr>
<tr>
<td>update</td>
</tr>
</tbody>
</table>

#### 6.46.1. get GET

Get a data center.

An example of getting a data center:
GET /ovirt-engine/api/datacenters/123

Table 6.133. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

6.46.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.46.2. remove DELETE

Removes the data center.
DELETE /ovirt-engine/api/datacenters/123

Without any special parameters, the storage domains attached to the data center are detached and then removed from the storage. If something fails when performing this operation, for example if there is no host available to remove the storage domains from the storage, the complete operation will fail.

If the `force` parameter is `true` then the operation will always succeed, even if something fails while removing one storage domain, for example. The failure is just ignored and the data center is removed from the database anyway.

<table>
<thead>
<tr>
<th>Table 6.134. Parameters summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>async</td>
</tr>
<tr>
<td>force</td>
</tr>
</tbody>
</table>

### 6.46.2.1. `force`

Indicates if the operation should succeed, and the storage domain removed from the database, even if something fails during the operation.

This parameter is optional, and the default value is `false`.

### 6.46.3. `update` `PUT`

Updates the data center.

The `name`, `description`, `storage_type`, `version`, `storage_format` and `mac_pool` elements are updatable post-creation. For example, to change the name and description of data center 123 send a request like this:

PUT /ovirt-engine/api/datacenters/123

With a request body like this:

```xml
<data_center>
  <name>myupdatedname</name>
  <description>An updated description for the data center</description>
</data_center>
```

<table>
<thead>
<tr>
<th>Table 6.135. Parameters summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>async</td>
</tr>
</tbody>
</table>
### 6.47. DATACENTERNETWORK

A service to manage a specific data center network.

#### Table 6.136. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the data center network details.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the network.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the network in the data center.</td>
</tr>
</tbody>
</table>

#### 6.47.1. get GET

Retrieves the data center network details.

#### Table 6.137. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>Out</td>
<td>The data center network.</td>
</tr>
</tbody>
</table>

#### 6.47.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.47.2. remove DELETE

Removes the network.

#### 6.47.3. update PUT

Updates the network in the data center.

#### Table 6.138. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>In/Out</td>
<td>The data center that is being updated.</td>
</tr>
</tbody>
</table>
Table 6.139. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Create a new network in a data center.</td>
</tr>
<tr>
<td>list</td>
<td>Lists networks in the data center.</td>
</tr>
</tbody>
</table>

6.48.1. add POST

Create a new network in a data center.

Post a request like in the example below to create a new network in a data center with an ID of 123.

```
POST /ovirt-engine/api/datacenters/123(networks

Use the following example in its body:

```
<network>
  <name>mynetwork</name>
</network>
```

Table 6.140. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>In/Out</td>
<td>The network object to be created in the data center.</td>
</tr>
</tbody>
</table>

6.48.2. list GET

Lists networks in the data center.

The order of the returned list of networks isn't guaranteed.

Table 6.141. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.48.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.48.2.2. max

Sets the maximum number of networks to return. If not specified, all the networks are returned.

### 6.49. DATACENTERS

A service to manage data centers.

#### Table 6.142. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong></td>
<td>Creates a new data center.</td>
</tr>
<tr>
<td><strong>list</strong></td>
<td>Lists the data centers.</td>
</tr>
</tbody>
</table>

#### 6.49.1. add POST

Creates a new data center.

Creation of a new data center requires the **name** and **local** elements. For example, to create a data center named **mydc** that uses shared storage (NFS, iSCSI or fibre channel) send a request like this:

```
POST /ovirt-engine/api/datacenters
```

With a request body like this:

```xml
<data_center>
    <name>mydc</name>
    <local>false</local>
</data_center>
```

#### Table 6.143. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>data_center</strong></td>
<td>DataCenter</td>
<td>In/Out</td>
<td>The data center that is being added.</td>
</tr>
</tbody>
</table>
6.49.2. list GET

Lists the data centers.

The following request retrieves a representation of the data centers:

```
GET /ovirt-engine/api/datacenters
```

The above request performed with `curl`:

```
curl \
   --request GET \
   --cacert /etc/pki/ovirt-engine/ca.pem \
   --header "Version: 4" \
   --header "Accept: application/xml" \
   --user "admin@internal:mypassword" \
   https://myengine.example.com/ovirt-engine/api/datacenters
```

This is what an example response could look like:

```
<data_center href="/ovirt-engine/api/datacenters/123" id="123">
   <name>Default</name>
   <description>The default Data Center</description>
   <link href="/ovirt-engine/api/datacenters/123/networks" rel="networks"/>
   <link href="/ovirt-engine/api/datacenters/123/storagedomains" rel="storagedomains"/>
   <link href="/ovirt-engine/api/datacenters/123/permissions" rel="permissions"/>
   <link href="/ovirt-engine/api/datacenters/123/clusters" rel="clusters"/>
   <link href="/ovirt-engine/api/datacenters/123/qoss" rel="qoss"/>
   <link href="/ovirt-engine/api/datacenters/123/iscsibonds" rel="iscsibonds"/>
   <link href="/ovirt-engine/api/datacenters/123/quotas" rel="quotas"/>
   <local>false</local>
   <quota_mode>disabled</quota_mode>
   <status>up</status>
   <supported_versions>
      <version>
         <major>4</major>
         <minor>0</minor>
      </version>
   </supported_versions>
</data_center>
```

Note the `id` code of your Default data center. This code identifies this data center in relation to other resources of your virtual environment.

The data center also contains a link to the storage domains collection. The data center uses this collection to attach storage domains from the storage domains main collection.
The order of the returned list of data centers is guaranteed only if the `sortby` clause is included in the `search` parameter.

**Table 6.144. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>case_sensitive</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <code>search</code> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td><code>data_centers</code></td>
<td>DataCenter[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td><code>filter</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td><code>follow</code></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td><code>max</code></td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of data centers to return.</td>
</tr>
<tr>
<td><code>search</code></td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned data centers.</td>
</tr>
</tbody>
</table>

**6.49.2.1. `case_sensitive`**

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

**6.49.2.2. `follow`**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.49.2.3. `max`**

Sets the maximum number of data centers to return. If not specified all the data centers are returned.

**6.50. DISK**

Manages a single disk.

**Table 6.145. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>copy</code></td>
<td>This operation copies a disk to the specified storage domain.</td>
</tr>
<tr>
<td><code>export</code></td>
<td>Exports a disk to an export storage domain.</td>
</tr>
</tbody>
</table>
### 6.50.1. copy POST

This operation copies a disk to the specified storage domain.

For example, copy of a disk can be facilitated using the following request:

```
POST /ovirt-engine/api/disks/123/copy
```

With a request body like this:

```
<action>
  <storage_domain id="456"/>
  <disk>
    <name>mydisk</name>
  </disk>
</action>
```

If the disk profile or the quota used currently by the disk aren’t defined for the new storage domain, then they can be explicitly specified. If they aren’t then the first available disk profile and the default quota are used.

For example, to explicitly use disk profile 987 and quota 753 send a request body like this:

```
<action>
  <storage_domain id="456"/>
  <disk_profile id="987"/>
  <quota id="753"/>
</action>
```

---

### Table 6.146. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the description of the disk.</td>
</tr>
<tr>
<td>move</td>
<td>Moves a disk to another storage domain.</td>
</tr>
<tr>
<td>reduce</td>
<td>Reduces the size of the disk image.</td>
</tr>
<tr>
<td>refreshlun</td>
<td>Refreshes a direct LUN disk with up-to-date information from the storage.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a disk.</td>
</tr>
<tr>
<td>sparsify</td>
<td>Sparsify the disk.</td>
</tr>
<tr>
<td>update</td>
<td>This operation updates the disk with the appropriate parameters.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>async</td>
<td>Boolean</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
</tr>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
</tr>
</tbody>
</table>

### 6.50.1.1. disk_profile

Disk profile for the disk in the new storage domain.

Disk profiles are defined for storage domains, so the old disk profile will not exist in the new storage domain. If this parameter is not used, the first disk profile from the new storage domain to which the user has permissions will be assigned to the disk.

### 6.50.1.2. quota

Quota for the disk in the new storage domain.

This optional parameter can be used to specify new quota for the disk, because the current quota may not be defined for the new storage domain. If this parameter is not used and the old quota is not defined for the new storage domain, the default (unlimited) quota will be assigned to the disk.

### 6.50.1.3. storage_domain

The storage domain where the new disk will be created. Can be specified using the **id** or **name** attributes. For example, to copy a disk to the storage domain named **mydata**, send a request like this:

```
POST /ovirt-engine/api/storagedomains/123/disks/789
```

With a request body like this:

```
<action>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
</action>
```
6.50.2. export POST

Exports a disk to an export storage domain.

Table 6.147. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the export should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.50.3. get GET

Retrieves the description of the disk.

Table 6.148. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_content</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all of the attributes of the disk should be included in the response.</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td>The description of the disk.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.50.3.1. all_content

Indicates if all of the attributes of the disk should be included in the response.

By default the following disk attributes are excluded:

- vms

For example, to retrieve the complete representation of disk '123':

```
GET /ovirt-engine/api/disks/123?all_content=true
```

6.50.3.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.50.4. move POST
Moves a disk to another storage domain.

For example, to move the disk with identifier **123** to a storage domain with identifier **456** send the following request:

```plaintext
POST /ovirt-engine/api/disks/123/move
```

With the following request body:

```xml
<action>
  <storage_domain id="456"/>
</action>
```

If the disk profile or the quota used currently by the disk aren’t defined for the new storage domain, then they can be explicitly specified. If they aren’t then the first available disk profile and the default quota are used.

For example, to explicitly use disk profile **987** and quota **753** send a request body like this:

```xml
<action>
  <storage_domain id="456"/>
  <disk_profile id="987"/>
  <quota id="753"/>
</action>
```

### Table 6.149. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the move should be performed asynchronously.</td>
</tr>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
<td>In</td>
<td>Disk profile for the disk in the new storage domain.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>In</td>
<td>Quota for the disk in the new storage domain.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.50.4.1. disk_profile

Disk profile for the disk in the new storage domain.

Disk profiles are defined for storage domains, so the old disk profile will not exist in the new storage domain. If this parameter is not used, the first disk profile from the new storage domain to which the user has permissions will be assigned to the disk.
6.50.4.2. quota

Quota for the disk in the new storage domain.

This optional parameter can be used to specify new quota for the disk, because the current quota may not be defined for the new storage domain. If this parameter is not used and the old quota is not defined for the new storage domain, the default (unlimited) quota will be assigned to the disk.

6.50.5. reduce POST

Reduces the size of the disk image.

Invokes `reduce` on the logical volume (i.e. this is only applicable for block storage domains). This is applicable for floating disks and disks attached to non-running virtual machines. There is no need to specify the size as the optimal size is calculated automatically.

Table 6.150. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.50.6. refreshlun POST

Refreshes a direct LUN disk with up-to-date information from the storage.

Refreshing a direct LUN disk is useful when:

- The LUN was added using the API without the host parameter, and therefore does not contain any information from the storage (see `DisksService::add`).
- New information about the LUN is available on the storage and you want to update the LUN with it.

To refresh direct LUN disk 123 using host 456, send the following request:

```
POST /ovirt-engine/api/disks/123/refreshlun
```

With the following request body:

```
<action>
  <host id='456'/>
</action>
```

Table 6.151. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>In</td>
<td>The host that will be used to refresh the direct LUN disk.</td>
</tr>
</tbody>
</table>
6.50.7. remove DELETE

Removes a disk.

Table 6.152. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.50.8. sparsify POST

Sparsify the disk.

Sparsification frees space in the disk image that is not used by its filesystem. As a result, the image will occupy less space on the storage.

Currently sparsification works only on disks without snapshots. Disks having derived disks are also not allowed.

6.50.9. update PUT

This operation updates the disk with the appropriate parameters. The only field that can be updated is `qcow_version`.

For example, update disk can be facilitated using the following request:

```
PUT /ovirt-engine/api/disks/123
```

With a request body like this:

```
<disk>
  <qcow_version>qcow2_v3</qcow_version>
</disk>
```

Since the backend operation is asynchronous the disk element which will be returned to the user might not be synced with the changed properties.

Table 6.153. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The update to apply to the disk.</td>
</tr>
</tbody>
</table>

6.51. DISKATTACHMENT

This service manages the attachment of a disk to a virtual machine.

Table 6.154. Methods summary
### Name | Summary
--- | ---
get | Returns the details of the attachment, including the bootable flag and link to the disk.
remove | Removes the disk attachment.
update | Update the disk attachment and the disk properties within it.

#### 6.51.1. get GET

Returns the details of the attachment, including the bootable flag and link to the disk.

An example of getting a disk attachment:

```plaintext
GET /ovirt-engine/api/vms/123/diskattachments/456
```

```
<disk_attachment href="/ovirt-engine/api/vms/123/diskattachments/456" id="456">
  <active>true</active>
  <bootable>true</bootable>
  <interface>virtio</interface>
  <disk href="/ovirt-engine/api/disks/456" id="456"/>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</disk_attachment>
```

**Table 6.155. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachmen t</td>
<td>DiskAttachm ent</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.51.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.51.2. remove DELETE

Removes the disk attachment.

This will only detach the disk from the virtual machine, but won’t remove it from the system, unless the detach_only parameter is false.

An example of removing a disk attachment:

```plaintext
DELETE /ovirt-engine/api/vms/123/diskattachments/456?detach_only=true
```
Table 6.156. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>detach_only</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the disk should only be detached from the virtual machine, but not removed from the system.</td>
</tr>
</tbody>
</table>

6.51.2.1. detach_only

Indicates if the disk should only be detached from the virtual machine, but not removed from the system. The default value is `true`, which won’t remove the disk from the system.

6.51.3. update PUT

Update the disk attachment and the disk properties within it.

```
PUT /vms/{vm:id}/disksattachments/{attachment:id}
<disk_attachment>
  <bootable>true</bootable>
  <interface>ide</interface>
  <active>true</active>
  <disk>
    <name>mydisk</name>
    <provisioned_size>1024</provisioned_size>
    ...
  </disk>
</disk_attachment>
```

Table 6.157. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_attachment</td>
<td>DiskAttachment</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.52. DISKATTACHMENTS

This service manages the set of disks attached to a virtual machine. Each attached disk is represented by a `DiskAttachment`, containing the bootable flag, the disk interface and the reference to the disk.

Table 6.158. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new disk attachment to the virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>List the disk that are attached to the virtual machine.</td>
</tr>
</tbody>
</table>

6.52.1. add POST
Adds a new disk attachment to the virtual machine. The attachment parameter can contain just a reference, if the disk already exists:

```xml
<disk_attachment>
  <bootable>true</bootable>
  <pass_discard>true</pass_discard>
  <interface>ide</interface>
  <active>true</active>
  <disk id="123"/>
</disk_attachment>
```

Or it can contain the complete representation of the disk, if the disk doesn’t exist yet:

```xml
<disk_attachment>
  <bootable>true</bootable>
  <pass_discard>true</pass_discard>
  <interface>ide</interface>
  <active>true</active>
  <disk>
    <name>mydisk</name>
    <provisioned_size>1024</provisioned_size>
  </disk>
</disk_attachment>
```

In this case the disk will be created and then attached to the virtual machine.

In both cases, use the following URL for a virtual machine with an id 345:

```
POST /ovirt-engine/api/vms/345/diskattachments
```

**IMPORTANT**

The server accepts requests that don’t contain the active attribute, but the effect is undefined. In some cases the disk will be automatically activated and in other cases it won’t. To avoid issues it is strongly recommended to always include the active attribute with the desired value.

**Table 6.159. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>DiskAttachment</td>
<td>In/Out</td>
<td>The disk attachment to add to the virtual machine.</td>
</tr>
</tbody>
</table>

**6.52.2. list GET**

List the disk that are attached to the virtual machine.

The order of the returned list of disks attachments isn't guaranteed.

**Table 6.160. Parameters summary**
### 6.52.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.53. DISKPROFILE

#### Table 6.161. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachmen ts</td>
<td>DiskAttachm ent[]</td>
<td>Out</td>
<td>A list of disk attachments that are attached to the virtual machine.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

#### 6.53.1. get GET

#### Table 6.162. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>profile</td>
<td>DiskProfile</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.53.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.53.2. remove DELETE

#### Table 6.163. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.53.3. update PUT

Update the specified disk profile in the system.

Table 6.164. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>profile</td>
<td>DiskProfile</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.54. DISKPROFILES

Table 6.165. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new disk profile to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of disk profiles of the system.</td>
</tr>
</tbody>
</table>

6.54.1. add POST

Add a new disk profile to the system.

Table 6.166. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>DiskProfile</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.54.2. list GET

Returns the list of disk profiles of the system.

The order of the returned list of disk profiles isn’t guaranteed.

Table 6.167. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of profiles to return.</td>
</tr>
<tr>
<td>profile</td>
<td>DiskProfile[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
6.54.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.54.2.2. max

Sets the maximum number of profiles to return. If not specified all the profiles are returned.

6.55. DISKSNAPSHOT

Table 6.168. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.55.1. get GET

Table 6.169. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>snapshot</td>
<td>DiskSnapshot</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.55.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.55.2. remove DELETE

Table 6.170. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.56. DISKSNAPSHOTS

Manages the collection of disk snapshots available in a storage domain.

Table 6.171. Methods summary
### 6.56.1. list GET

Returns the list of disk snapshots of the storage domain.

The order of the returned list of disk snapshots isn’t guaranteed.

#### Table 6.172. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of snapshots to return.</td>
</tr>
<tr>
<td>snapshots</td>
<td>DiskSnapshot[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.56.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.56.1.2. max

Sets the maximum number of snapshots to return. If not specified all the snapshots are returned.

### 6.57. DISKS

Manages the collection of disks available in the system.

#### Table 6.173. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new floating disk.</td>
</tr>
<tr>
<td>list</td>
<td>Get list of disks.</td>
</tr>
</tbody>
</table>

#### 6.57.1. add POST

Adds a new floating disk.

There are three types of disks that can be added - disk image, direct LUN and **Cinder** disk.

**Adding a new image disk:**
When creating a new floating image Disk, the API requires the storage_domain, provisioned_size and format attributes.

Note that block storage domains (i.e., storage domains with the storage type of iSCSI or FCP) don’t support the combination of the raw format with sparse=true, so sparse=false must be stated explicitly.

To create a new floating image disk with specified provisioned_size, format and name on a storage domain with an id 123, send a request as follows:

POST /ovirt-engine/api/disks

With a request body as follows:

```xml
<disk>
  <storage_domains>
    <storage_domain id="123"/>
  </storage_domains>
  <name>mydisk</name>
  <provisioned_size>1048576</provisioned_size>
  <format>cow</format>
</disk>
```

Adding a new direct LUN disk:

When adding a new floating direct LUN via the API, there are two flavors that can be used:

1. With a host element - in this case, the host is used for sanity checks (e.g., that the LUN is visible) and to retrieve basic information about the LUN (e.g., size and serial).

2. Without a host element - in this case, the operation is a database-only operation, and the storage is never accessed.

To create a new floating direct LUN disk with a host element with an id 123, specified alias, type and logical_unit with an id 456 (that has the attributes address, port and target), send a request as follows:

POST /ovirt-engine/api/disks

With a request body as follows:

```xml
<disk>
  <alias>mylun</alias>
  <lun_storage>
    <host id="123"/>
    <type>iSCSI</type>
    <logical_units>
      <logical_unit id="456">
        <address>10.35.10.20</address>
        <port>3260</port>
        <target>iqn.2017-01.com.myhost:444</target>
      </logical_unit>
    </logical_units>
  </lun_storage>
</disk>
```
To create a new floating direct LUN disk without using a host, remove the **host** element.

**Adding a new Cinder disk:**

To create a new floating Cinder disk, send a request as follows:

```bash
POST /ovirt-engine/api/disks
```

With a request body as follows:

```xml
<disk>
  <openstack_volume_type>
    <name>myceph</name>
  </openstack_volume_type>
  <storage_domains>
    <storage_domain>
      <name>cinderDomain</name>
    </storage_domain>
  </storage_domains>
  <provisioned_size>1073741824</provisioned_size>
  <interface>virtio</interface>
  <format>raw</format>
</disk>
```

**Adding a floating disks in order to upload disk snapshots:**

Since version 4.2 of the engine it is possible to upload disks with snapshots. This request should be used to create the base image of the images chain (The consecutive disk snapshots (images), should be created using **disk-attachments** element when creating a snapshot).

The disk has to be created with the same disk identifier and image identifier of the uploaded image. I.e. the identifiers should be saved as part of the backup process. The image identifier can be also fetched using the `qemu-img info` command. For example, if the disk image is stored into a file named `b7a4c6c5-443b-47c5-967f-6abc79675e8b/myimage.img`:

```bash
$ qemu-img info b7a4c6c5-443b-47c5-967f-6abc79675e8b/myimage.img
image: b548366b-fb51-4b41-97be-733c887fe305
  file format: qcow2
  virtual size: 1.0G (1073741824 bytes)
  disk size: 196K
  backing file: ad58716a-1fe9-481f-815e-664de1df04eb
  backing file format: raw
```

To create a disk with with the disk identifier and image identifier obtained with the `qemu-img info` command shown above, send a request like this:

```bash
POST /ovirt-engine/api/disks
```

With a request body as follows:

```xml
<disk id="b7a4c6c5-443b-47c5-967f-6abc79675e8b">
  <image_id>b548366b-fb51-4b41-97be-733c887fe305</image_id>
</disk>
```
Table 6.174. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The disk.</td>
</tr>
</tbody>
</table>

6.57.2. list GET

Get list of disks.

GET /ovirt-engine/api/disks

You will get a XML response which will look like this one:

```xml
<disks>
  <disk id="123">
    <actions>...</actions>
    <name>MyDisk</name>
    <description>MyDisk description</description>
    <link href="/ovirt-engine/api/disks/123/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/disks/123/statistics" rel="statistics"/>
    <actual_size>5345845248</actual_size>
    <alias>MyDisk alias</alias>
    ...
    <status>ok</status>
    <storage_type>image</storage_type>
    <wipe_after_delete>false</wipe_after_delete>
    <disk_profile_id="123"/>
    <quota id="123"/>
  </disk>
  ...
</disks>
```

The order of the returned list of disks is guaranteed only if the `sortby` clause is included in the `search` parameter.

Table 6.175. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 6.57.2.1. case_sensitive

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

### 6.57.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.57.2.3. max

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.58. DOMAIN

A service to view details of an authentication domain in the system.

#### Table 6.176. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the authentication domain information.</td>
</tr>
</tbody>
</table>

### 6.58.1. get GET

Gets the authentication domain information.

Usage:

```
GET /ovirt-engine/api/domains/5678
```

Will return the domain information:
<domain href="/ovirt-engine/api/domains/5678" id="5678">
  <name>internal-authz</name>
  <link href="/ovirt-engine/api/domains/5678/users" rel="users"/>
  <link href="/ovirt-engine/api/domains/5678/groups" rel="groups"/>
  <link href="/ovirt-engine/api/domains/5678/users?search={query}" rel="users/search"/>
  <link href="/ovirt-engine/api/domains/5678/groups?search={query}" rel="groups/search"/>
</domain>

### Table 6.177. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Out</td>
<td>The authentication domain.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

#### 6.58.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.59. DOMAINGROUP

#### Table 6.178. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.59.1. get GET

#### Table 6.179. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>get</td>
<td>Group</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.59.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.60. DOMAINGROUPS

#### Table 6.180. Methods summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td></td>
<td></td>
<td>Returns the list of groups.</td>
</tr>
</tbody>
</table>

### 6.60.1. list GET

Returns the list of groups.

The order of the returned list of groups isn’t guaranteed.

#### Table 6.181. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>groups</td>
<td>Group[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of groups to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned groups.</td>
</tr>
</tbody>
</table>

#### 6.60.1.1. case_sensitive

Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.

#### 6.60.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.60.1.3. max

Sets the maximum number of groups to return. If not specified all the groups are returned.

### 6.61. DOMAINUSER

A service to view a domain user in the system.

#### Table 6.182. Methods summary
### Table 6.183. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td>Out</td>
<td>The domain user.</td>
</tr>
</tbody>
</table>

#### 6.61.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.62. DOMAINUSERGROUPS

A service that shows a user’s group membership in the AAA extension.

#### Table 6.184. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of groups that the user is a member of.</td>
</tr>
</tbody>
</table>

#### 6.62.1. list GET
Returns the list of groups that the user is a member of.

### Table 6.185. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>groups</td>
<td>Group[]</td>
<td>Out</td>
<td>The list of groups that the user is a member of.</td>
</tr>
</tbody>
</table>

#### 6.62.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.63. DOMAINUSERS

A service to list all domain users in the system.

### Table 6.186. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>List all the users in the domain.</td>
</tr>
</tbody>
</table>

#### 6.63.1. list GET

List all the users in the domain.

Usage:

```
GET /ovirt-engine/api/domains/5678/users
```

Will return the list of users in the domain:

```xml
<users>
  <user href="/ovirt-engine/api/domains/5678/users/1234" id="1234">
    <name>admin</name>
    <namespace>*</namespace>
    <principal>admin</principal>
    <user_name>admin@internal-authz</user_name>
    <domain href="/ovirt-engine/api/domains/5678" id="5678">
      <name>internal-authz</name>
    </domain>
    <groups/>
  </user>
</users>
```

The order of the returned list of users isn’t guaranteed.

### Table 6.187. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of users to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned users.</td>
</tr>
<tr>
<td>users</td>
<td>User[]</td>
<td>Out</td>
<td>The list of users in the domain.</td>
</tr>
</tbody>
</table>

### 6.63.1.1. case_sensitive

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

### 6.63.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.63.1.3. max

Sets the maximum number of users to return. If not specified all the users are returned.

### 6.64. DOMAINS

A service to list all authentication domains in the system.

#### Table 6.188. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>List all the authentication domains in the system.</td>
</tr>
</tbody>
</table>

### 6.64.1. list GET

List all the authentication domains in the system.

**Usage:**

```
GET /ovirt-engine/api/domains
```

Will return the list of domains:
<domains>
  <domain href="/ovirt-engine/api/domains/5678" id="5678">
    <name>internal-authz</name>
    <link href="/ovirt-engine/api/domains/5678/users" rel="users"/>
    <link href="/ovirt-engine/api/domains/5678/groups" rel="groups"/>
    <link href="/ovirt-engine/api/domains/5678/users?search={query}" rel="users/search"/>
    <link href="/ovirt-engine/api/domains/5678/groups?search={query}" rel="groups/search"/>
  </domain>
</domains>

The order of the returned list of domains isn’t guaranteed.

Table 6.189. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>domains</td>
<td>Domain[]</td>
<td>Out</td>
<td>The list of domains.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of domains to return.</td>
</tr>
</tbody>
</table>

6.64.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.64.1.2. max

Sets the maximum number of domains to return. If not specified all the domains are returned.

6.65. ENGINEKATELLOERRATA

A service to manage Katello errata assigned to the engine. The information is retrieved from Katello.

Table 6.190. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Retrieves the representation of the Katello errata.</td>
</tr>
</tbody>
</table>

6.65.1. list GET

Retrieves the representation of the Katello errata.

GET /ovirt-engine/api/katelloerrata

You will receive response in XML like this one:
The order of the returned list of erratum isn’t guaranteed.

Table 6.191. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>errata</td>
<td>KatelloErratum[]</td>
<td>Out</td>
<td>A representation of Katello errata.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of errata to return.</td>
</tr>
</tbody>
</table>

6.65.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.65.1.2. max

Sets the maximum number of errata to return. If not specified all the errata are returned.

6.66. EVENT

A service to manage an event in the system.

Table 6.192. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get an event.</td>
</tr>
</tbody>
</table>
remove

Removes an event from internal audit log.

### 6.66.1. get GET

Get an event.

An example of getting an event:

```
GET /ovirt-engine/api/events/123
```

Note that the number of fields changes according to the information that resides on the event. For example, for storage domain related events you will get the storage domain reference, as well as the reference for the data center this storage domain resides in.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Event</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.66.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.66.2. remove DELETE

Removes an event from internal audit log.

An event can be removed by sending following request

```
DELETE /ovirt-engine/api/events/123
```
Table 6.194. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.67. EVENTS

A service to manage events in the system.

Table 6.195. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds an external event to the internal audit log.</td>
</tr>
<tr>
<td>list</td>
<td>Get list of events.</td>
</tr>
<tr>
<td>undelete</td>
<td></td>
</tr>
</tbody>
</table>

6.67.1. add POST

Adds an external event to the internal audit log.

This is intended for integration with external systems that detect or produce events relevant for the administrator of the system. For example, an external monitoring tool may be able to detect that a file system is full inside the guest operating system of a virtual machine. This event can be added to the internal audit log sending a request like this:

```
POST /ovirt-engine/api/events
<event>
  <description>File system /home is full</description>
  <severity>alert</severity>
  <origin>mymonitor</origin>
  <custom_id>1467879754</custom_id>
</event>
```

Events can also be linked to specific objects. For example, the above event could be linked to the specific virtual machine where it happened, using the `vm` link:

```
POST /ovirt-engine/api/events
<event>
  <description>File system /home is full</description>
  <severity>alert</severity>
  <origin>mymonitor</origin>
  <custom_id>1467879754</custom_id>
  <vm id="aae98225-5b73-490d-a252-899209af17e9"/>
</event>
```
NOTE

When using links, like the `vm` in the previous example, only the `id` attribute is accepted. The `name` attribute, if provided, is simply ignored.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>Event</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.67.2. list GET

Get list of events.

```
GET /ovirt-engine/api/events
```

To the above request we get following response:

```
<events>
  <event href="/ovirt-engine/api/events/2" id="2">
    <description>User admin@internal-authz logged out.</description>
    <code>31</code>
    <correlation_id>1e892ea9</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T12:14:34.541+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
  <event href="/ovirt-engine/api/events/1" id="1">
    <description>User admin logged in.</description>
    <code>30</code>
    <correlation_id>1fbd81f4</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:54:35.229+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
</events>
```

The following events occur:

- `id="1"` - The API logs in the admin user account.
- `id="2"` - The API logs out of the admin user account.

The order of the returned list of events is always guaranteed. If the `sortby` clause is included in the
**search** parameter, then the events will be ordered according to that clause. If the **sortby** clause isn’t included, then the events will be sorted by the numeric value of the **id** attribute, starting with the highest value. This, combined with the **max** parameter, simplifies obtaining the most recent event:

```
GET /ovirt-engine/api/events?max=1
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>case_sensitive</strong></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <strong>search</strong> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td><strong>events</strong></td>
<td>Event[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td><strong>follow</strong></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td><strong>from</strong></td>
<td>Integer</td>
<td>In</td>
<td>Indicates the event index after which events should be returned.</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of events to return.</td>
</tr>
<tr>
<td><strong>search</strong></td>
<td>String</td>
<td>In</td>
<td>The events service provides search queries similar to other resource services.</td>
</tr>
</tbody>
</table>

### 6.67.2.1. **case_sensitive**

Indicates if the search performed using the **search** parameter should be performed taking case into account. The default value is **true**, which means that case is taken into account. If you want to search ignoring case set it to **false**.

### 6.67.2.2. **follow**

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.67.2.3. **from**

Indicates the event index after which events should be returned. The indexes of events are strictly increasing, so when this parameter is used only the events with greater indexes will be returned. For example, the following request will return only the events with indexes greater than **123**:

```
GET /ovirt-engine/api/events?from=123
```

This parameter is optional, and if not specified then the first event returned will be most recently generated.

### 6.67.2.4. **max**

Sets the maximum number of events to return. If not specified all the events are returned.
6.67.2.5. search

The events service provides search queries similar to other resource services.

We can search by providing specific severity.

```
GET /ovirt-engine/api/events?search=severity%3Dnormal
```

To the above request we get a list of events which severity is equal to normal:

```
<events>
  <event href="/ovirt-engine/api/events/2" id="2">
    <description>User admin@internal-authz logged out.</description>
    <code>31</code>
    <correlation_id>1fbd81f4</correlation_id>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:54:35.229+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
  <event href="/ovirt-engine/api/events/1" id="1">
    <description>Affinity Rules Enforcement Manager started.</description>
    <code>10780</code>
    <custom_id>-1</custom_id>
    <flood_rate>30</flood_rate>
    <origin>oVirt</origin>
    <severity>normal</severity>
    <time>2016-09-14T11:52:18.861+02:00</time>
    <user href="/ovirt-engine/api/users/57d91d48-00da-0137-0138-000000000244" id="57d91d48-00da-0137-0138-000000000244"/>
  </event>
</events>
```

A virtualization environment generates a large amount of events after a period of time. However, the API only displays a default number of events for one search query. To display more than the default, the API separates results into pages with the page command in a search query. The following search query tells the API to paginate results using a page value in combination with the sortby clause:

```
sortby time asc page 1
```

Below example paginates event resources. The URL-encoded request is:

```
GET /ovirt-engine/api/events?search=sortby%20time%20asc%20page%201
```

Increase the page value to view the next page of results.

```
GET /ovirt-engine/api/events?search=sortby%20time%20asc%20page%202
```

6.67.3. undelete POST

Table 6.198. Parameters summary
**Sync**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the un-delete should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.68. EXTERNALCOMPUTERESOURCE

Manages a single external compute resource.

Compute resource is a term of host external provider. The external provider also needs to know to where the provisioned host needs to register. The login details of the engine are saved as a compute resource in the external provider side.

#### Table 6.199. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves external compute resource details.</td>
</tr>
</tbody>
</table>

#### 6.68.1. get GET

Retrieves external compute resource details.

For example, to get the details of compute resource **234** of provider **123**, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123/computeresources/234
```

It will return a response like this:

```
<external_compute_resource href="/ovirt-engine/api/externalhostproviders/123/computeresources/234" id="234">
  <name>hostname</name>
  <provider>Ovirt</provider>
  <url>https://hostname/api</url>
  <user>admin@internal</user>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_compute_resource>
```

#### Table 6.200. Parameters summary

| Name    | Type                  | Direction | Summary                                                        |
|---------|-----------------------|-----------|                                                               |
| follow  | String                | In        | Indicates which inner links should be followed.               |
| resource| ExternalComputeResource | Out       | External compute resource information                          |
6.68.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.69. EXTERNALCOMPUTERESOURCES

Manages a collection of external compute resources.

Compute resource is a term of host external provider. The external provider also needs to know to where the provisioned host needs to register. The login details of the engine is saved as a compute resource in the external provider side.

Table 6.201. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Retrieves a list of external compute resources.</td>
</tr>
</tbody>
</table>

6.69.1. list GET

Retrieves a list of external compute resources.

For example, to retrieve the compute resources of external host provider 123, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123/computeresources
```

It will return a response like this:

```
<external_compute_resources>
  <external_compute_resource href="/ovirt-engine/api/externalhostproviders/123/computeresources/234" id="234">
    <name>hostname</name>
    <provider>oVirt</provider>
    <url>https://address/api</url>
    <user>admin@internal</user>
    <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
  </external_compute_resource>
  ...
</external_compute_resources>
```

The order of the returned list of compute resources isn't guaranteed.

Table 6.202. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of resources to return.</td>
</tr>
</tbody>
</table>
### 6.69. resources

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resources</td>
<td>ExternalComputerResource[]</td>
<td>Out</td>
<td>List of external computer resources.</td>
</tr>
</tbody>
</table>

#### 6.69.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.69.1.2. max

Sets the maximum number of resources to return. If not specified all the resources are returned.

### 6.70. EXTERNALDISCOVEREDHOST

This service manages a single discovered host.

#### Table 6.203. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get discovered host info.</td>
</tr>
</tbody>
</table>

#### 6.70.1. get GET

Get discovered host info.

Retrieves information about an host that is managed in external provider management system, such as Foreman. The information includes hostname, address, subnet, base image and more.

For example, to get the details of host 234 from provider 123, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123/discoveredhosts/234
```

The result will be like this:

```xml
<external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/234" id="234">
  <name>mac001a4ad0404040</name>
  <ip>10.34.67.43</ip>
  <last_report>2017-04-24 11:05:41 UTC</last_report>
  <mac>00:1a:4a:d0:40:40</mac>
  <subnet_name>sat0</subnet_name>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_discovered_host>
```
Table 6.204. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>host</td>
<td>ExternalDisccoveredHost</td>
<td>Out</td>
<td>Host's hardware and config information.</td>
</tr>
</tbody>
</table>

6.70.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.71. EXTERNALDISCOVEREDHOSTS

This service manages external discovered hosts.

Table 6.205. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Get list of discovered hosts' information.</td>
</tr>
</tbody>
</table>

6.71.1. list GET

Get list of discovered hosts' information.

Discovered hosts are fetched from third-party providers such as Foreman.

To list all discovered hosts for provider 123 send the following:

```plaintext
GET /ovirt-engine/api/externalhostproviders/123/discoveredhost
```

```xml
<external_discovered_hosts>
  <external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/456" id="456">
    <name>mac001a4ad04031</name>
    <ip>10.34.67.42</ip>
    <last_report>2017-04-24 11:05:41 UTC</last_report>
    <mac>00:1a:4a:d0:40:31</mac>
    <subnet_name>sat0</subnet_name>
  </external_discovered_host>
  <external_discovered_host href="/ovirt-engine/api/externalhostproviders/123/discoveredhosts/789" id="789">
    <name>mac001a4ad04040</name>
    <ip>10.34.67.43</ip>
    <last_report>2017-04-24 11:05:41 UTC</last_report>
    <mac>00:1a:4a:d0:40:40</mac>
    <subnet_name>sat0</subnet_name>
</external_discovered_hosts>
```
The order of the returned list of hosts isn’t guaranteed.

**Table 6.206. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>hosts</td>
<td>ExternalDiscHost[]</td>
<td>Out</td>
<td>List of discovered hosts</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of hosts to return.</td>
</tr>
</tbody>
</table>

**6.71.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

**6.71.1.2. max**

Sets the maximum number of hosts to return. If not specified all the hosts are returned.

**6.72. EXTERNALHOST**

**Table 6.207. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

**6.72.1. get GET**

**Table 6.208. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>host</td>
<td>ExternalHost</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.72.1.1. follow**
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.73. EXTERNALHOSTGROUP

This service manages a single host group information.

Host group is a term of host provider - the host group includes provision details that are applied to new discovered host. Information such as subnet, operating system, domain, etc.

Table 6.209. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get host group information.</td>
</tr>
</tbody>
</table>

6.73.1. get GET

Get host group information.

For example, to get the details of hostgroup 234 of provider 123, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123/hostgroups/234
```

It will return a response like this:

```
<external_host_group href="/ovirt-engine/api/externalhostproviders/123/hostgroups/234" id="234">
  <name>rhel7</name>
  <architecture_name>x86_64</architecture_name>
  <domain_name>s.com</domain_name>
  <operating_system_name>RedHat 7.3</operating_system_name>
  <subnet_name>sat0</subnet_name>
  <external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123"/>
</external_host_group>
```

Table 6.210. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>group</td>
<td>ExternalHost Group</td>
<td>Out</td>
<td>Host group information.</td>
</tr>
</tbody>
</table>

6.73.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.74. EXTERNALHOSTGROUPS

This service manages hostgroups.

Table 6.211. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Get host groups list from external host provider.</td>
</tr>
</tbody>
</table>

6.74.1. list GET

Get host groups list from external host provider.

Host group is a term of host providers - the host group includes provision details. This API returns all possible hostgroups exposed by the external provider.

For example, to get the details of all host groups of provider 123, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123/hostgroups
```

The response will be like this:

```
<external_host_groups>
  <external_host_group href="/ovirt-engine/api/externalhostproviders/123/hostgroups/234" id="234">
    <name>rhel7</name>
    <architecture_name>x86_64</architecture_name>
    <domain_name>example.com</domain_name>
    <operating_system_name>RedHat 7.3</operating_system_name>
    <subnet_name>sat0</subnet_name>
  </external_host_group>
  ...
</external_host_groups>
```

The order of the returned list of host groups isn’t guaranteed.

Table 6.212. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>groups</td>
<td>ExternalHostGroup[]</td>
<td>Out</td>
<td>List of all hostgroups available for external host provider</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of groups to return.</td>
</tr>
</tbody>
</table>

6.74.1.1. follow
Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.74.1.2. max

Sets the maximum number of groups to return. If not specified all the groups are returned.

### 6.75. EXTERNALHOSTPROVIDER

Represents an external host provider, such as Foreman or Satellite.


#### Table 6.213. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get external host provider information</td>
</tr>
<tr>
<td></td>
<td>Host provider, Foreman or Satellite, can be set as an external provider in ovirt.</td>
</tr>
<tr>
<td>importcertificates</td>
<td>Import the SSL certificates of the external host provider.</td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>testconnectivity</td>
<td>In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.</td>
</tr>
<tr>
<td>update</td>
<td>Update the specified external host provider in the system.</td>
</tr>
</tbody>
</table>

### 6.75.1. get GET

Get external host provider information

Host provider, Foreman or Satellite, can be set as an external provider in ovirt. To see details about specific host providers attached to ovirt use this API.

For example, to get the details of host provider 123, send a request like this:

```
GET /ovirt-engine/api/externalhostproviders/123
```

The response will be like this:

```
<external_host_provider href="/ovirt-engine/api/externalhostproviders/123" id="123">
  <name>mysatellite</name>
  <requires_authentication>true</requires_authentication>
  <url>https://mysatellite.example.com</url>
  <username>admin</username>
</external_host_provider>
```
Table 6.214. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>provider</td>
<td>ExternalHost Provider</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.75.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.75.2. importcertificates POST

Import the SSL certificates of the external host provider.

Table 6.215. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.75.3. remove DELETE

Table 6.216. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.75.4. testconnectivity POST

In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

Table 6.217. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the test should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.75.5. update PUT

Update the specified external host provider in the system.

Table 6.218. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>provider</td>
<td>ExternalHost Provider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.76. EXTERNALHOSTPROVIDERS

Table 6.219. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new external host provider to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of external host providers.</td>
</tr>
</tbody>
</table>

6.76.1. add POST

Adds a new external host provider to the system.

Table 6.220. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider</td>
<td>ExternalHost Provider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.76.2. list GET

Returns the list of external host providers.

The order of the returned list of host providers is not guaranteed.

Table 6.221. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of providers to return.</td>
</tr>
</tbody>
</table>
### 6.76.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.76.2.2. max

Sets the maximum number of providers to return. If not specified, all the providers are returned.

### 6.77. EXTERNALHOSTS

#### Table 6.222. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Return the list of external hosts.</td>
</tr>
</tbody>
</table>

#### 6.77.1. list GET

Return the list of external hosts.

The order of the returned list of hosts isn’t guaranteed.

#### Table 6.223. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>hosts</td>
<td>ExternalHost Provider[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of hosts to return.</td>
</tr>
</tbody>
</table>

#### 6.77.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.77.1.2. max
Sets the maximum number of hosts to return. If not specified all the hosts are returned.

6.78. EXTERNALNETWORKPROVIDERCONFIGURATION
Describes how an external network provider is provisioned by the system on the host.

Table 6.224. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the information about an external network provider on the host.</td>
</tr>
</tbody>
</table>

6.78.1. get GET
Returns the information about an external network provider on the host.

Table 6.225. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>configuration</td>
<td>ExternalNetworkProviderConfiguration</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.78.1.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.79. EXTERNALNETWORKPROVIDERCONFIGURATIONS
A service to list all external network providers provisioned by the system on the host.

Table 6.226. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of all external network providers on the host.</td>
</tr>
</tbody>
</table>

6.79.1. list GET
Returns the list of all external network providers on the host.

The order of the returned list of networks is not guaranteed.

Table 6.227. Parameters summary
6.79.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.80. EXTERNALPROVIDER

Provides capability to manage external providers.

#### Table 6.228. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>importcertificates</td>
<td>Import the SSL certificates of the external host provider.</td>
</tr>
<tr>
<td>testconnectivity</td>
<td>In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.</td>
</tr>
</tbody>
</table>

#### 6.80.1. importcertificates POST

Import the SSL certificates of the external host provider.

#### Table 6.229. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.80.2. testconnectivity POST

In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

#### Table 6.230. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the test should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.81. EXTERNALPROVIDERCERTIFICATE

A service to view specific certificate for external provider.

**Table 6.231. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get specific certificate.</td>
</tr>
</tbody>
</table>

#### 6.81.1. get GET

Get specific certificate.

GET `/ovirt-engine/api/externalhostproviders/123/certificate/0`

And here is sample response:

```xml
<certificate id="0">
  <organization>provider.example.com</organization>
  <subject>CN=provider.example.com</subject>
  <content>...</content>
</certificate>
```

**Table 6.232. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificate</td>
<td>Certificate</td>
<td>Out</td>
<td>The details of the certificate.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.81.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.82. EXTERNALPROVIDERCERTIFICATES

A service to view certificates for external provider.

**Table 6.233. Methods summary**

---

210
### 6.82. list GET

Returns the chain of certificates presented by the external provider.

```
GET /ovirt-engine/api/externalhostproviders/123/certificates
```

And here is sample response:

```
<certificates>
  <certificate id="789">...</certificate>
  ...
</certificates>
```

The order of the returned certificates is always guaranteed to be the sign order: the first is the certificate of the server itself, the second the certificate of the CA that signs the first, so on.

#### Table 6.234. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>Out</td>
<td>List containing certificate details.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of certificates to return.</td>
</tr>
</tbody>
</table>

#### 6.82.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.82.1.2. max

Sets the maximum number of certificates to return. If not specified all the certificates are returned.

### 6.83. EXTERNALVMIMPORTS

Provides capability to import external virtual machines.

#### Table 6.235. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 6.83.1. add POST

This operation is used to import a virtual machine from external hypervisor, such as KVM, XEN or VMware.

For example import of a virtual machine from VMware can be facilitated using the following request:

```
POST /externalvmimports
```

With request body of type `ExternalVmImport`, for example:

```
<external_vm_import>
  <vm>
    <name>my_vm</name>
  </vm>
  <cluster id="360014051136c20574f743b6bd28177fd" />
  <storage_domain id="8b59ae65-e988-4000-8b93-dbfc6717fe50" />
  <name>vm_name_as_is_in_vmware</name>
  <sparse>true</sparse>
  <username>vmware_user</username>
  <password>123456</password>
  <provider>VMWARE</provider>
  <url>vpx://vmware_user@vcenter-host/DataCenter/Cluster/esxi-host?
    no_verify=1</url>
  <drivers_iso id="virtio-win-1.6.7.iso" />
</external_vm_import>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>This operation is used to import a virtual machine from external hypervisor, such as KVM, XEN or VMware.</td>
</tr>
</tbody>
</table>

### 6.84. FENCEAGENT

A service to manage fence agent for a specific host.

#### Table 6.237. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets details of this fence agent.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a fence agent for a specific host.</td>
</tr>
</tbody>
</table>
## 6.84.1. get GET

Gets details of this fence agent.

```plaintext
GET /ovirt-engine/api/hosts/123/fenceagents/0
```

And here is sample response:

```xml
<agent id="0">
  <type>apc</type>
  <order>1</order>
  <ip>192.168.1.101</ip>
  <user>user</user>
  <password>xxx</password>
  <port>9</port>
  <options>name1=value1, name2=value2</options>
</agent>
```

### Table 6.238. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Agent</td>
<td>Out</td>
<td>Fence agent details.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

### 6.84.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

## 6.84.2. remove DELETE

Removes a fence agent for a specific host.

```plaintext
DELETE /ovirt-engine/api/hosts/123/fenceagents/0
```

### Table 6.239. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.84.3. update PUT

Update a fencing-agent.

Table 6.240. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Agent</td>
<td>In/Out</td>
<td>Fence agent details.</td>
</tr>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.85. FENCEAGENTS

A service to manage fence agents for a specific host.

Table 6.241. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new fencing-agent to the host.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of fencing agents configured for the host.</td>
</tr>
</tbody>
</table>

6.85.1. add POST

Add a new fencing-agent to the host.

Table 6.242. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent</td>
<td>Agent</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.85.2. list GET

Returns the list of fencing agents configured for the host.

GET /ovirt-engine/api/hosts/123/fenceagents

And here is sample response:

```xml
<agents>
  <agent id="0">
    <type>aapc</type>
    <order>1</order>
    <ip>192.168.1.101</ip>
    <user>user</user>
    <password>xxx</password>
  </agent>
</agents>
```
The order of the returned list of fencing agents isn’t guaranteed.

### Table 6.243. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>agents</td>
<td>Agent[]</td>
<td>Out</td>
<td>List of fence agent details.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of agents to return.</td>
</tr>
</tbody>
</table>

#### 6.85.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.85.2.2. max

Sets the maximum number of agents to return. If not specified all the agents are returned.

### 6.86. FILE

#### Table 6.244. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.86.1. get GET

#### Table 6.245. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>File</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.86.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.
6.87. FILES

Provides a way for clients to list available files.

This service is specifically targeted to ISO storage domains, which contain ISO images and virtual floppy disks (VFDs) that an administrator uploads.

The addition of a CD-ROM device to a virtual machine requires an ISO image from the files of an ISO storage domain.

Table 6.246. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of ISO images and virtual floppy disks available in the storage domain.</td>
</tr>
</tbody>
</table>

6.87.1. list GET

Returns the list of ISO images and virtual floppy disks available in the storage domain. The order of the returned list is not guaranteed.

If the `refresh` parameter is `false`, the returned list may not reflect recent changes to the storage domain; for example, it may not contain a new ISO file that was recently added. This is because the server caches the list of files to improve performance. To get the very latest results, set the `refresh` parameter to `true`.

The default value of the `refresh` parameter is `true`, but it can be changed using the configuration value `ForceRefreshDomainFilesByDefault`:

```shell
# engine-config -s ForceRefreshDomainFilesByDefault=false
```

**IMPORTANT**

Setting the value of the `refresh` parameter to `true` has an impact on the performance of the server. Use it only if necessary.

Table 6.247. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>case_sensitive</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <code>search</code> parameter should take case into account.</td>
</tr>
<tr>
<td><code>file</code></td>
<td>File[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td><code>follow</code></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td><code>max</code></td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of files to return.</td>
</tr>
</tbody>
</table>
### 6.87. refresh

Indicates whether the list of files should be refreshed from the storage domain, rather than showing cached results that are updated at certain intervals.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether the list of files should be refreshed from the storage domain, rather than showing cached results that are updated at certain intervals.</td>
</tr>
</tbody>
</table>

### 6.87.1. search

A query string used to restrict the returned files.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned files.</td>
</tr>
</tbody>
</table>

#### 6.87.1.1. case_sensitive

Indicates if the search performed using the search parameter should take case into account. The default value is true.

#### 6.87.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.87.1.3. max

Sets the maximum number of files to return. If not specified, all the files are returned.

### 6.88. FILTER

Table 6.248. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.88.1. get GET

Table 6.249. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>result</td>
<td>Filter</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
6.88.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.88.2. remove DELETE

Table 6.250. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.89. FILTERS

Manages the filters used by an scheduling policy.

Table 6.251. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a filter to a specified user defined scheduling policy.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of filters used by the scheduling policy.</td>
</tr>
</tbody>
</table>

6.89.1. add POST

Add a filter to a specified user defined scheduling policy.

Table 6.252. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Filter</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.89.2. list GET

Returns the list of filters used by the scheduling policy.

The order of the returned list of filters isn’t guaranteed.

Table 6.253. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-----------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>filters</td>
<td>Filter[]</td>
<td>Out</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of filters to return.</td>
</tr>
</tbody>
</table>

6.89.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.89.2.2. max

Sets the maximum number of filters to return. If not specified all the filters are returned.

6.90. FOLLOW

6.91. GLUSTERBRICK

This service manages a single gluster brick.

Table 6.254. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get details of a brick.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a brick.</td>
</tr>
<tr>
<td>replace</td>
<td>Replaces this brick with a new one.</td>
</tr>
</tbody>
</table>

6.91.1. get GET

Get details of a brick.

Retrieves status details of brick from underlying gluster volume with header All-Content set to true. This is the equivalent of running `gluster volume status <volumename> <brickname> detail`.

For example, to get the details of brick 234 of gluster volume 123, send a request like this:

```
GET /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/234
```

Which will return a response body like this:

```
<brick id="234">
  <name>host1:/rhgs/data/brick1</name>
```
6.91.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.91.2. remove  DELETE

Removes a brick.

Removes a brick from the underlying gluster volume and deletes entries from database. This can be used only when removing a single brick without data migration. To remove multiple bricks and with data migration, use `migrate` instead.

For example, to delete brick 234 from gluster volume 123, send a request like this:
DELETE /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/234

Table 6.256. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.91.3. replace POST

Replaces this brick with a new one.

**IMPORTANT**

This operation has been deprecated since version 3.5 of the engine and will be removed in the future. Use add brick(s) and migrate brick(s) instead.

Table 6.257. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the replacement should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.92. GLUSTERBRICKS

This service manages the gluster bricks in a gluster volume

Table 6.258. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate</td>
<td>Activate the bricks post data migration of remove brick operation.</td>
</tr>
<tr>
<td>add</td>
<td>Adds a list of bricks to gluster volume.</td>
</tr>
<tr>
<td>list</td>
<td>Lists the bricks of a gluster volume.</td>
</tr>
<tr>
<td>migrate</td>
<td>Start migration of data prior to removing bricks.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes bricks from gluster volume.</td>
</tr>
<tr>
<td>stoptmigrate</td>
<td>Stops migration of data from bricks for a remove brick operation.</td>
</tr>
</tbody>
</table>
### 6.92.1. activate POST

Activate the bricks post data migration of remove brick operation.

Used to activate brick(s) once the data migration from bricks is complete but user no longer wishes to remove bricks. The bricks that were previously marked for removal will now be used as normal bricks.

For example, to retain the bricks that on glustervolume 123 from which data was migrated, send a request like this:

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/activate
```

With a request body like this:

```
<action>
  <bricks>
    <brick>
      <name>host1:/rhgs/brick1</name>
    </brick>
  </bricks>
</action>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the activation should be performed asynchronously.</td>
</tr>
<tr>
<td>bricks</td>
<td>GlusterBrick[]</td>
<td>In</td>
<td>The list of bricks that need to be re-activated.</td>
</tr>
</tbody>
</table>

### 6.92.2. add POST

Adds a list of bricks to gluster volume.

Used to expand a gluster volume by adding bricks. For replicated volume types, the parameter `replica_count` needs to be passed. In case the replica count is being increased, then the number of bricks needs to be equivalent to the number of replica sets.

For example, to add bricks to gluster volume 123, send a request like this:

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

With a request body like this:

```
<bricks>
  <brick>
    <server_id>111</server_id>
    <brick_dir>/export/data/brick3</brick_dir>
  </brick>
</bricks>
```
Table 6.260. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricks</td>
<td>GlusterBrick[ ]</td>
<td>In/Out</td>
<td>The list of bricks to be added to the volume</td>
</tr>
<tr>
<td>replica_count</td>
<td>Integer</td>
<td>In</td>
<td>Replica count of volume post add operation.</td>
</tr>
<tr>
<td>stripe_count</td>
<td>Integer</td>
<td>In</td>
<td>Stripe count of volume post add operation.</td>
</tr>
</tbody>
</table>

6.92.3. list GET

Lists the bricks of a gluster volume.

For example, to list bricks of gluster volume 123, send a request like this:

```plaintext
GET /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

Provides an output as below:

```xml
<bricks>
  <brick id="234">
    <name>host1:/rhgs/data/brick1</name>
    <brick_dir>/rhgs/data/brick1</brick_dir>
    <server_id>111</server_id>
    <status>up</status>
  </brick>
  <brick id="233">
    <name>host2:/rhgs/data/brick1</name>
    <brick_dir>/rhgs/data/brick1</brick_dir>
    <server_id>222</server_id>
    <status>up</status>
  </brick>
</bricks>
```

The order of the returned list is based on the brick order provided at gluster volume creation.

Table 6.261. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricks</td>
<td>GlusterBrick[ ]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of bricks to return.</td>
</tr>
</tbody>
</table>
6.92.3.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.92.3.2. max

Sets the maximum number of bricks to return. If not specified all the bricks are returned.

6.92.4. migrate POST

Start migration of data prior to removing bricks.

Removing bricks is a two-step process, where the data on bricks to be removed, is first migrated to remaining bricks. Once migration is completed the removal of bricks is confirmed via the API remove. If at any point, the action needs to be cancelled stopmigrate has to be called.

For instance, to delete a brick from a gluster volume with id 123, send a request:

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/migrate
```

With a request body like this:

```
<action>
   <bricks>
      <brick>
         <name>host1:/rhgs/brick1</name>
      </brick>
   </bricks>
</action>
```

The migration process can be tracked from the job id returned from the API using job and steps in job using step.

Table 6.262. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the migration should be performed asynchronously.</td>
</tr>
<tr>
<td>bricks</td>
<td>GlusterBrick[ ]</td>
<td>In</td>
<td>List of bricks for which data migration needs to be started.</td>
</tr>
</tbody>
</table>

6.92.5. remove DELETE

Removes bricks from gluster volume.

The recommended way to remove bricks without data loss is to first migrate the data using stopmigrate and then removing them. If migrate was not called on bricks prior to remove, the bricks are removed without data migration which may lead to data loss.
For example, to delete the bricks from gluster volume 123, send a request like this:

```
DELETE /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks
```

With a request body like this:

```
<bricks>
  <brick>
    <name>host:brick_directory</name>
  </brick>
</bricks>
```

Table 6.263. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
<tr>
<td>bricks</td>
<td>GlusterBrick[]</td>
<td>In</td>
<td>The list of bricks to be removed</td>
</tr>
<tr>
<td>replica_count</td>
<td>Integer</td>
<td>In</td>
<td>Replica count of volume post add operation.</td>
</tr>
</tbody>
</table>

6.92.6. stopmigrate POST

Stops migration of data from bricks for a remove brick operation.

To cancel data migration that was started as part of the 2-step remove brick process in case the user wishes to continue using the bricks. The bricks that were marked for removal will function as normal bricks post this operation.

For example, to stop migration of data from the bricks of gluster volume 123, send a request like this:

```
POST /ovirt-engine/api/clusters/567/glustervolumes/123/glusterbricks/stopmigrate
```

With a request body like this:

```
<bricks>
  <brick>
    <name>host:brick_directory</name>
  </brick>
</bricks>
```

Table 6.264. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 6.92.6.1. `bricks`

List of bricks for which data migration needs to be stopped. This list should match the arguments passed to `migrate`.

### 6.93. GLUSTERHOOK

**Table 6.265. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>disable</code></td>
<td>Resolves status conflict of hook among servers in cluster by disabling Gluster hook in all servers of the cluster.</td>
</tr>
<tr>
<td><code>enable</code></td>
<td>Resolves status conflict of hook among servers in cluster by disabling Gluster hook in all servers of the cluster.</td>
</tr>
<tr>
<td><code>get</code></td>
<td></td>
</tr>
<tr>
<td><code>remove</code></td>
<td>Removes the this Gluster hook from all servers in cluster and deletes it from the database.</td>
</tr>
<tr>
<td><code>resolve</code></td>
<td>Resolves missing hook conflict depending on the resolution type.</td>
</tr>
</tbody>
</table>

**6.93.1. `disable` POST**

Resolves status conflict of hook among servers in cluster by disabling Gluster hook in all servers of the cluster. This updates the hook status to `DISABLED` in database.

**Table 6.266. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>async</code></td>
<td><code>Boolean</code></td>
<td><code>In</code></td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

**6.93.2. `enable` POST**

Resolves status conflict of hook among servers in cluster by disabling Gluster hook in all servers of the cluster. This updates the hook status to `DISABLED` in database.
Table 6.267. Parameters summary

| Name | Type      | Direction | Summary                                                        |
|------|-----------|-----------|                                                               |
| async | Boolean   | In        | Indicates if the action should be performed asynchronously.    |

6.93.3. get

Table 6.268. Parameters summary

| Name | Type      | Direction | Summary                                                      |
|------|-----------|-----------|                                                              |
| follow | String   | In        | Indicates which inner links should be *followed*.            |
| hook  | GlusterHook | Out      |                                                               |

6.93.3.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.93.4. remove

Removes the this Gluster hook from all servers in cluster and deletes it from the database.

Table 6.269. Parameters summary

| Name | Type    | Direction | Summary                                                        |
|------|---------|-----------|                                                               |
| async | Boolean | In        | Indicates if the remove should be performed asynchronously.    |

6.93.5. resolve

Resolves missing hook conflict depending on the resolution type.

For **ADD** resolves by copying hook stored in engine database to all servers where the hook is missing. The engine maintains a list of all servers where hook is missing.

For **COPY** resolves conflict in hook content by copying hook stored in engine database to all servers where the hook is missing. The engine maintains a list of all servers where the content is conflicting. If a host id is passed as parameter, the hook content from the server is used as the master to copy to other servers in cluster.

Table 6.270. Parameters summary
### GLUSTERHOOKS

#### Table 6.271. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of hooks.</td>
</tr>
</tbody>
</table>

#### Table 6.272. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>hooks</td>
<td>GlusterHook[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of hooks to return.</td>
</tr>
</tbody>
</table>

### 6.94.1. list GET

Returns the list of hooks.

The order of the returned list of hooks isn’t guaranteed.

#### 6.94.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.94.1.2. max

Sets the maximum number of hooks to return. If not specified all the hooks are returned.

### 6.95. GLUSTERVOLUME

This service manages a single gluster volume.
Table 6.273. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get the gluster volume details.</td>
</tr>
<tr>
<td>getprofilestatistics</td>
<td>Get gluster volume profile statistics.</td>
</tr>
<tr>
<td>rebalance</td>
<td>Rebalance the gluster volume.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the gluster volume.</td>
</tr>
<tr>
<td>resetalloptions</td>
<td>Resets all the options set in the gluster volume.</td>
</tr>
<tr>
<td>resetoption</td>
<td>Resets a particular option in the gluster volume.</td>
</tr>
<tr>
<td>setoption</td>
<td>Sets a particular option in the gluster volume.</td>
</tr>
<tr>
<td>start</td>
<td>Starts the gluster volume.</td>
</tr>
<tr>
<td>startprofile</td>
<td>Start profiling the gluster volume.</td>
</tr>
<tr>
<td>stop</td>
<td>Stops the gluster volume.</td>
</tr>
<tr>
<td>stopprofile</td>
<td>Stop profiling the gluster volume.</td>
</tr>
<tr>
<td>stoprebalance</td>
<td>Stop rebalancing the gluster volume.</td>
</tr>
</tbody>
</table>

6.95.1. get GET

Get the gluster volume details.

For example, to get details of a gluster volume with identifier 123 in cluster 456, send a request like this:

```
GET /ovirt-engine/api/clusters/456/glustervolumes/123
```

This GET request will return the following output:

```
<gluster_volume id="123">
  <name>data</name>
  <link href="/ovirt-engine/api/clusters/456/glustervolumes/123/glusterbricks" rel="glusterbricks"/>
  <disperse_count>0</disperse_count>
  <options>
    <option>
      <name>storage.owner-gid</name>
```
Table 6.274. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>volume</td>
<td>GlusterVolume</td>
<td>Out</td>
<td>Representation of the gluster volume.</td>
</tr>
</tbody>
</table>

6.95.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.95.2. getprofilestatistics POST

Get gluster volume profile statistics.

For example, to get profile statistics for a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/getprofilestatistics
```

Table 6.275. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>details</td>
<td>GlusterVolume</td>
<td>Out</td>
<td>Gluster volume profiling information returned from the action.</td>
</tr>
</tbody>
</table>
6.95.3. rebalance POST

Rebalance the gluster volume.

Rebalancing a gluster volume helps to distribute the data evenly across all the bricks. After expanding or shrinking a gluster volume (without migrating data), we need to rebalance the data among the bricks. In a non-replicated volume, all bricks should be online to perform the rebalance operation. In a replicated volume, at least one of the bricks in the replica should be online.

For example, to rebalance a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/rebalance
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the rebalance should be performed asynchronously.</td>
</tr>
<tr>
<td>fix_layout</td>
<td>Boolean</td>
<td>In</td>
<td>If set to true, rebalance will only fix the layout so that new data added to the volume is distributed across all the hosts.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the rebalance should be force started.</td>
</tr>
</tbody>
</table>

6.95.3.1. fix_layout

If set to true, rebalance will only fix the layout so that new data added to the volume is distributed across all the hosts. But it will not migrate/rebalance the existing data. Default is false.

6.95.3.2. force

Indicates if the rebalance should be force started. The rebalance command can be executed with the force option even when the older clients are connected to the cluster. However, this could lead to a data loss situation. Default is false.

6.95.4. remove DELETE

Removes the gluster volume.

For example, to remove a volume with identifier 123 in cluster 456, send a request like this:

```
DELETE /ovirt-engine/api/clusters/456/glustervolumes/123
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.95.5. resetalloptions POST

Resets all the options set in the gluster volume.

For example, to reset all options in a gluster volume with identifier 123 in cluster 456, send a request like this:

```plaintext
POST /ovirt-engine/api/clusters/456/glustervolumes/123/resetalloptions
```

Table 6.278. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the reset should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.95.6. resetoption POST

Resets a particular option in the gluster volume.

For example, to reset a particular option option1 in a gluster volume with identifier 123 in cluster 456, send a request like this:

```plaintext
POST /ovirt-engine/api/clusters/456/glustervolumes/123/resetoption
```

With the following request body:

```xml
<action>
<option name="option1"/>
</action>
```

Table 6.279. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the reset should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>option</td>
<td>Option</td>
<td>In</td>
<td>Option to reset.</td>
</tr>
</tbody>
</table>

6.95.7. setoption POST

Sets a particular option in the gluster volume.

For example, to set option1 with value value1 in a gluster volume with identifier 123 in cluster 456, send a request like this:

```plaintext
POST /ovirt-engine/api/clusters/456/glustervolumes/123/setoption
```
With the following request body:

```xml
<action>
  <option name="option1" value="value1"/>
</action>
```

Table 6.280. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>option</td>
<td>Option</td>
<td>In</td>
<td>Option to set.</td>
</tr>
</tbody>
</table>

6.95.8. start POST

Starts the gluster volume.

A Gluster Volume should be started to read/write data. For example, to start a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/start
```

Table 6.281. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the volume should be force started.</td>
</tr>
</tbody>
</table>

6.95.8.1. force

Indicates if the volume should be force started. If a gluster volume is started already but few/all bricks are down then force start can be used to bring all the bricks up. Default is false.

6.95.9. startprofile POST

Start profiling the gluster volume.

For example, to start profiling a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/startprofile
```

Table 6.282. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>option</td>
<td>option</td>
<td></td>
<td>Option to set.</td>
</tr>
<tr>
<td>name</td>
<td>option</td>
<td></td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>option</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 6.95.10. stop POST

Stops the gluster volume.

Stopping a volume will make its data inaccessible.

For example, to stop a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stop
```

**Table 6.283. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.95.11. stopprofile POST

Stop profiling the gluster volume.

For example, to stop profiling a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stopprofile
```

**Table 6.284. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.95.12. stoprebalance POST

Stop rebalancing the gluster volume.

For example, to stop rebalancing a gluster volume with identifier 123 in cluster 456, send a request like this:

```
POST /ovirt-engine/api/clusters/456/glustervolumes/123/stoprebalance
```
Table 6.285. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.96. GLUSTERVERS

This service manages a collection of gluster volumes available in a cluster.

Table 6.286. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new gluster volume.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all gluster volumes in the cluster.</td>
</tr>
</tbody>
</table>

6.96.1. add POST

Creates a new gluster volume.

The volume is created based on properties of the `volume` parameter. The properties `name`, `volume_type` and `bricks` are required.

For example, to add a volume with name `myvolume` to the cluster 123, send the following request:

```
POST /ovirt-engine/api/clusters/123/glustervolumes
```

With the following request body:

```
<gluster_volume>
  <name>myvolume</name>
  <volume_type>replicate</volume_type>
  <replica_count>3</replica_count>
  <bricks>
    <brick>
      <server_id>server1</server_id>
      <brick_dir>/exp1</brick_dir>
    </brick>
    <brick>
      <server_id>server2</server_id>
      <brick_dir>/exp1</brick_dir>
    </brick>
    <brick>
      <server_id>server3</server_id>
      <brick_dir>/exp1</brick_dir>
    </brick>
  </bricks>
</gluster_volume>
```
Table 6.287. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td>GlusterVolume</td>
<td>In/Out</td>
<td>The gluster volume definition from which to create the volume is passed as input and the newly created volume is returned.</td>
</tr>
</tbody>
</table>

6.96.2. list GET

Lists all gluster volumes in the cluster.

For example, to list all Gluster Volumes in cluster 456, send a request like this:

```bash
GET /ovirt-engine/api/clusters/456/glustervolumes
```

The order of the returned list of volumes isn’t guaranteed.

Table 6.288. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of volumes to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned volumes.</td>
</tr>
<tr>
<td>volumes</td>
<td>GlusterVolume[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.96.2.1. case_sensitive

Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.

6.96.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.96.2.3. max

Sets the maximum number of volumes to return. If not specified all the volumes are returned.
6.97. GROUP

Manages a group of users. Use this service to either get groups details or remove groups. In order to add new groups please use service that manages the collection of groups.

Table 6.289. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the system group information.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the system group.</td>
</tr>
</tbody>
</table>

6.97.1. get GET

Gets the system group information.

Usage:

```
GET /ovirt-engine/api/groups/123
```

Will return the group information:

```xml
<group href="/ovirt-engine/api/groups/123" id="123">
  <name>mygroup</name>
  <link href="/ovirt-engine/api/groups/123/roles" rel="roles"/>
  <link href="/ovirt-engine/api/groups/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/groups/123/tags" rel="tags"/>
  <domain_entry_id>476652557A382F67696B6D2B32762B37796E46476D513D3D</domain_entry_id>
  <namespace>DC=example,DC=com</namespace>
  <domain href="/ovirt-engine/api/domains/ABCDEF" id="ABCDEF">
    <name>myextension-authz</name>
  </domain>
</group>
```

Table 6.290. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>get</td>
<td>Group</td>
<td>Out</td>
<td>The system group.</td>
</tr>
</tbody>
</table>

6.97.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.97.2. remove DELETE

Removes the system group.

Usage:

DELETE /ovirt-engine/api/groups/123

Table 6.291. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.98. GROUPS

Manages the collection of groups of users.

Table 6.292. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add group from a directory service.</td>
</tr>
<tr>
<td>list</td>
<td>List all the groups in the system.</td>
</tr>
</tbody>
</table>

6.98.1. add POST

Add group from a directory service. Please note that domain name is name of the authorization provider.

For example, to add the Developers group from the internal-authz authorization provider send a request like this:

POST /ovirt-engine/api/groups

With a request body like this:

```
<group>
  <name>Developers</name>
  <domain>
    <name>internal-authz</name>
  </domain>
</group>
```

Table 6.293. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>Group</td>
<td>In/Out</td>
<td>The group to be added.</td>
</tr>
</tbody>
</table>
6.98.2. list GET

List all the groups in the system.

Usage:

```
GET /ovirt-engine/api/groups
```

Will return the list of groups:

```xml
<groups>
  <group href="/ovirt-engine/api/groups/123" id="123">
    <name>mygroup</name>
    <link href="/ovirt-engine/api/groups/123/roles" rel="roles"/>
    <link href="/ovirt-engine/api/groups/123/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/groups/123/tags" rel="tags"/>
  </group>
  ...
</groups>
```

The order of the returned list of groups isn’t guaranteed.

Table 6.294. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>groups</td>
<td>Group[]</td>
<td>Out</td>
<td>The list of groups.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of groups to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned groups.</td>
</tr>
</tbody>
</table>

6.98.2.1. case_sensitive

Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.
6.98.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.98.2.3. max

Sets the maximum number of groups to return. If not specified all the groups are returned.

6.99. HOST

A service to manage a host.

Table 6.295. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate</td>
<td>Activates the host for use, for example to run virtual machines.</td>
</tr>
<tr>
<td>approve</td>
<td>Approve a pre-installed Hypervisor host for usage in the virtualization environment.</td>
</tr>
<tr>
<td>commitnetconfig</td>
<td>Marks the network configuration as good and persists it inside the host.</td>
</tr>
<tr>
<td>deactivate</td>
<td>Deactivates the host to perform maintenance tasks.</td>
</tr>
<tr>
<td>enrollcertificate</td>
<td>Enrolls the certificate of the host.</td>
</tr>
<tr>
<td>fence</td>
<td>Controls the host's power management device.</td>
</tr>
<tr>
<td>force selects pm</td>
<td>To manually set a host as the storage pool manager (SPM).</td>
</tr>
<tr>
<td>get</td>
<td>Gets the host details.</td>
</tr>
<tr>
<td>install</td>
<td>Installs the latest version of VDSM and related software on the host.</td>
</tr>
<tr>
<td>iscsidiscove r</td>
<td>Discovers iSCSI targets on the host, using the initiator details.</td>
</tr>
<tr>
<td>iscsilogin</td>
<td>Login to iSCSI targets on the host, using the target details.</td>
</tr>
<tr>
<td>refresh</td>
<td>Refresh the host devices and capabilities.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the host from the system.</td>
</tr>
<tr>
<td>setupnetworks</td>
<td>This method is used to change the configuration of the network interfaces of a host.</td>
</tr>
</tbody>
</table>
To synchronize all networks on the host, send a request like this:

[source] ---- POST /ovirt-engine/api/hosts/123/syncallnetworks ----

With a request body like this:

[source,xml] ---- <action/> ----

Discovers the block Storage Domains which are candidates to be imported to the setup.

Update the host properties.

Upgrades VDSM and selected software on the host.

Check if there are upgrades available for the host.

### 6.99.1. activate POST

Activates the host for use, for example to run virtual machines.

**Table 6.296. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the activation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.99.2. approve POST

Approve a pre-installed Hypervisor host for usage in the virtualization environment.

This action also accepts an optional cluster element to define the target cluster for this host.

**Table 6.297. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the approval should be performed asynchronously.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td>The cluster where the host will be added after it is approved.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>In</td>
<td>The host to approve.</td>
</tr>
</tbody>
</table>

### 6.99.3. commitnetconfig POST
Marks the network configuration as good and persists it inside the host.

An API user commits the network configuration to persist a host network interface attachment or detachment, or persist the creation and deletion of a bonded interface.

**IMPORTANT**

Networking configuration is only committed after the engine has established that host connectivity is not lost as a result of the configuration changes. If host connectivity is lost, the host requires a reboot and automatically reverts to the previous networking configuration.

For example, to commit the network configuration of host with id 123 send a request like this:

```
POST /ovirt-engine/api/hosts/123/commitnetconfig
```

With a request body like this:

```
<action/>
```

Table 6.298. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.99.4. deactivate POST

Deactivates the host to perform maintenance tasks.

Table 6.299. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the deactivation should be performed asynchronously.</td>
</tr>
<tr>
<td>reason</td>
<td>String</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>stop_gluster_service</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the gluster service should be stopped as part of deactivating the host.</td>
</tr>
</tbody>
</table>

6.99.4.1. stop_gluster_service

Indicates if the gluster service should be stopped as part of deactivating the host. It can be used while performing maintenance operations on the gluster host. Default value for this variable is **false**.

6.99.5. enrollcertificate POST
Enrolls the certificate of the host. Useful in case you get a warning that it is about to expire or has already expired.

**Table 6.300. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the enrollment should be performed asynchronously.</td>
</tr>
</tbody>
</table>

**6.99.6. fence POST**

Controls the host’s power management device.

For example, to start the host. This can be done via:

```bash
#!/bin/sh -x
url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="...

curl ""--verbose "
--cacert /etc/pki/ovirt-engine/ca.pem "
--user "$\{user\}:\$\{password\}" "
--request POST "
--header "Version: 4" "
--header "Content-Type: application/xml" "
--header "Accept: application/xml" "
--data ' "
<action>
  <fence_type>start</fence_type>
</action>
' "
"$\{url\}/hosts/123/fence"
```

**Table 6.301. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the fencing should be performed asynchronously.</td>
</tr>
<tr>
<td>fence_type</td>
<td>String</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>power_management</td>
<td>PowerManagement</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.99.7. forceselectspm POST**
To manually set a host as the storage pool manager (SPM).

```
POST /ovirt-engine/api/hosts/123/forceselectspm
```

With a request body like this:

```xml
?action/>
```

### Table 6.302. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.99.8. get GET

Gets the host details.

```
GET /ovirt-engine/api/hosts/123
```

### Table 6.303. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_content</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all of the attributes of the host should be included in the response.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Out</td>
<td>The queried host.</td>
</tr>
</tbody>
</table>

#### 6.99.8.1. all_content

Indicates if all of the attributes of the host should be included in the response.

By default the following attributes are excluded:

- **hosted_engine**

For example, to retrieve the complete representation of host '123':

```
GET /ovirt-engine/api/hosts/123?all_content=true
```
NOTE

These attributes are not included by default because retrieving them impacts performance. They are seldom used and require additional queries to the database. Use this parameter with caution and only when specifically required.

6.99.8.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.99.9. install POST

Installs the latest version of VDSM and related software on the host.

The action also performs every configuration steps on the host which is done during adding host to the engine: kdump configuration, hosted-engine deploy, kernel options changes, etc.

The host type defines additional parameters for the action.

Example of installing a host, using curl and JSON, plain:

```
curl
   --verbose
   --cacert /etc/pki/ovirt-engine/ca.pem
   --request PUT
   --header "Content-Type: application/json"
   --header "Accept: application/json"
   --header "Version: 4"
   --user "admin@internal:...
   --data '
   "root_password": "myrootpassword"
   }
' 
"https://engine.example.com/ovirt-engine/api/hosts/123"
```

Example of installing a host, using curl and JSON, with hosted engine components:

```
curl
   curl
   --verbose
   --cacert /etc/pki/ovirt-engine/ca.pem
   --request PUT
   --header "Content-Type: application/json"
   --header "Accept: application/json"
   --header "Version: 4"
   --user "admin@internal:...
   --data '
   {"root_password": "myrootpassword"
   }
' 
"https://engine.example.com/ovirt-engine/api/hosts/123?
deploy_hosted_engine=true"
```
IMPORTANT

Since version 4.1.2 of the engine when a host is reinstalled we override the host firewall definitions by default.

Table 6.304. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the installation should be performed asynchronously.</td>
</tr>
<tr>
<td>deploy_hosted_engine</td>
<td>Boolean</td>
<td>In</td>
<td>When set to true it means this host should also deploy the self-hosted engine components.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>In</td>
<td>The override_iptables property is used to indicate if the firewall configuration should be replaced by the default one.</td>
</tr>
<tr>
<td>image</td>
<td>String</td>
<td>In</td>
<td>When installing Red Hat Virtualization Host, an ISO image file is required.</td>
</tr>
<tr>
<td>root_pass</td>
<td>String</td>
<td>In</td>
<td>The password of the root user, used to connect to the host via SSH.</td>
</tr>
<tr>
<td>ssh</td>
<td>Ssh</td>
<td>In</td>
<td>The SSH details used to connect to the host.</td>
</tr>
<tr>
<td>undeploy_hosted_engine</td>
<td>Boolean</td>
<td>In</td>
<td>When set to true it means this host should un-deploy the self-hosted engine components and this host will not function as part of the High Availability cluster.</td>
</tr>
</tbody>
</table>

6.99.9.1. deploy_hosted_engine

When set to true it means this host should also deploy the self-hosted engine components. A missing value is treated as true i.e deploy. Omitting this parameter means false and will perform no operation in the self-hosted engine area.

6.99.9.2. undeploy_hosted_engine

When set to true it means this host should un-deploy the self-hosted engine components and this host will not function as part of the High Availability cluster. A missing value is treated as true i.e un-deploy Omitting this parameter means false and will perform no operation in the self-hosted engine area.

6.99.10. iscsidiscover POST

Discovers iSCSI targets on the host, using the initiator details.

For example, to discover iSCSI targets available in myiscsi.example.com, from host 123, send a request like this:
POST /ovirt-engine/api/hosts/123/iscsidiscover

With a request body like this:

```xml
<action>
  <iscsi>
    <address>myiscsi.example.com</address>
  </iscsi>
</action>
```

The result will be like this:

```xml
<discovered_targets>
  <iscsi_details>
    <address>10.35.1.72</address>
    <port>3260</port>
    <portal>10.35.1.72:3260,1</portal>
    <target>iqn.2015-08.com.tgt:444</target>
  </iscsi_details>
</discovered_targets>
```

Table 6.305. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the discovery should be performed asynchronously.</td>
</tr>
<tr>
<td>discovered_targets</td>
<td>IscsiDetails[]</td>
<td>Out</td>
<td>The discovered targets including all connection information.</td>
</tr>
<tr>
<td>iscsi</td>
<td>IscsiDetails</td>
<td>In</td>
<td>The target iSCSI device.</td>
</tr>
<tr>
<td>iscsi_targets</td>
<td>String[]</td>
<td>Out</td>
<td>The iSCSI targets.</td>
</tr>
</tbody>
</table>

6.99.10.1. iscsi_targets

The iSCSI targets.

Since version 4.2 of the engine, this parameter is deprecated, use discovered_targets instead.

6.99.11. iscsiLogin POST

Login to iSCSI targets on the host, using the target details.

Table 6.306. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>

247
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the login should be performed asynchronously.</td>
</tr>
<tr>
<td>iscsi</td>
<td>IscsiDetails</td>
<td>In</td>
<td>The target iSCSI device.</td>
</tr>
</tbody>
</table>

### 6.99.12. refresh POST

Refresh the host devices and capabilities.

#### Table 6.307. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the refresh should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.99.13. remove DELETE

Remove the host from the system.

```sh
#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="...

curl 
  --verbose 
  --cacert /etc/pki/ovirt-engine/ca.pem 
  --user "$user:$password" 
  --request DELETE 
  --header "Version: 4" 
  "$url/hosts/1ff7a191-2f3b-4eff-812b-9f91a30c3acc"
```

#### Table 6.308. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.99.14. setupnetworks POST

This method is used to change the configuration of the network interfaces of a host.

For example, if you have a host with three network interfaces eth0, eth1 and eth2 and you want to configure a new bond using eth0 and eth1, and put a VLAN on top of it. Using a simple shell script and the curl command line HTTP client that can be done as follows:
#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="...

curl \
  --verbose \n  --cacert /etc/pki/ovirt-engine/ca.pem \n  --user "$user":$password" \n  --request POST \n  --header "Version: 4" \n  --header "Content-Type: application/xml" \n  --header "Accept: application/xml" \n  --data ' \
    <action> 
      <modified_bonds> 
        <host_nic> 
          <name>bond0</name> 
          <bonding> 
            <options> 
              <option> 
                <name>mode</name> 
                <value>4</value> 
              </option> 
              <option> 
                <name>miimon</name> 
                <value>100</value> 
              </option> 
            </options> 
            <slaves> 
              <host_nic> 
                <name>eth1</name> 
              </host_nic> 
              <host_nic> 
                <name>eth2</name> 
              </host_nic> 
            </slaves> 
          </bonding> 
        </host_nic> 
      </modified_bonds> 
      <modified_network_attachments> 
        <network_attachment> 
          <network> 
            <name>myvlan</name> 
          </network> 
          <host_nic> 
            <name>bond0</name> 
          </host_nic> 
          <ip_address_assignments> 
            <assignment_method>static</assignment_method> 
            <ip_address_assignment> 
              <ip> 
                <address>192.168.122.10</address> 
                <netmask>255.255.255.0</netmask> 
              </ip> 
            </ip_address_assignment> 
          </ip_address_assignments> 
        </network_attachment> 
      </modified_network_attachments> 
    </action>
NOTE

This is valid for version 4 of the API. In previous versions some elements were represented as XML attributes instead of XML elements. In particular the options and ip elements were represented as follows:

```xml
<options name="mode" value="4"/>
<options name="miimon" value="100"/>
<ip address="192.168.122.10" netmask="255.255.255.0"/>
```

Using the Python SDK the same can be done with the following code:

```python
# Find the service that manages the collection of hosts:
hosts_service = connection.system_service().hosts_service()

# Find the host:
host = hosts_service.list(search='name=myhost')[0]

# Find the service that manages the host:
host_service = hosts_service.host_service(host.id)

# Configure the network adding a bond with two slaves and attaching it to a
# network with an static IP address:
host_service.setup_networks(
    modified_bonds=[
        types.HostNic(  
            name='bond0',  
            bonding=types.Bonding(  
                options=[  
                    types.Option(  
                        name='mode',  
                        value='4',  
                    ),  
                    types.Option(  
                        name='miimon',  
                        value='100',  
                    ),  
                ],  
            slaves=[
        ],
```

```
IMPORTANT

To make sure that the network configuration has been saved in the host, and that it will be applied when the host is rebooted, remember to call `commitnetconfig`.

Table 6.309. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>check_connectivity</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>connectivity_timeout</td>
<td>Integer</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>modified_bonds</td>
<td>HostNic[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>modified_labels</td>
<td>NetworkLabel[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>modified_network_attachments</td>
<td>NetworkAttachment[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>removed_bonds</td>
<td>HostNic[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>removed_labels</td>
<td>NetworkLabel[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>removed_network_attachments</td>
<td>NetworkAttachment[]</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>synchronized_network_attachments</td>
<td>NetworkAttachment[]</td>
<td>In</td>
<td>A list of network attachments that will be synchronized.</td>
</tr>
</tbody>
</table>

### 6.99.15. syncallnetworks POST

To synchronize all networks on the host, send a request like this:

```
POST /ovirt-engine/api/hosts/123/syncallnetworks
```

With a request body like this:

```
<action/>
```

Table 6.310. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.99.16. unregisteredstoragedomainsdiscover POST

Discovers the block Storage Domains which are candidates to be imported to the setup. For FCP no arguments are required.

Table 6.311. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the discovery should be performed asynchronously.</td>
</tr>
<tr>
<td>iscsi</td>
<td>IscsiDetails</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.99.17. update PUT

Update the host properties.

For example, to update a the kernel command line of a host send a request like this:

```
PUT /ovirt-engine/api/hosts/123
```

With request body like this:

```
<host>
 <os>
  <custom_kernel_cmdline>vfio_iommu_type1.allow_unsafe_interrupts=1</custom_kernel_cmdline>
 </os>
</host>
```

Table 6.312. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>
6.99.18. upgrade POST

Upgrades VDSM and selected software on the host.

Table 6.313. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the upgrade should be performed asynchronously.</td>
</tr>
<tr>
<td>image</td>
<td>String</td>
<td>In</td>
<td>The image parameter specifies path to image, which is used for upgrade.</td>
</tr>
<tr>
<td>reboot</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the host should be rebooted after upgrade.</td>
</tr>
</tbody>
</table>

6.99.18.1. image

The image parameter specifies path to image, which is used for upgrade. This parameter is used only to upgrade Vintage Node hosts and it is not relevant for other hosts types.

6.99.18.2. reboot

Indicates if the host should be rebooted after upgrade. By default the host is rebooted.

NOTE

This parameter is ignored for Red Hat Virtualization Host, which is always rebooted after upgrade.

6.99.19. upgradecheck POST

Check if there are upgrades available for the host. If there are upgrades available an icon will be displayed next to host status icon in the Administration Portal. Audit log messages are also added to indicate the availability of upgrades. The upgrade can be started from the webadmin or by using the upgrade host action.

6.100. HOSTDEVICE

A service to access a particular device of a host.

Table 6.314. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieve information about a particular host’s device.</td>
</tr>
</tbody>
</table>

6.100.1. get GET

Retrieve information about a particular host’s device.
An example of getting a host device:

```
GET /ovirt-engine/api/hosts/123/devices/456
```

```xml
<host_device href="/ovirt-engine/api/hosts/123/devices/456" id="456">
  <name>usb_1_9_1_1_0</name>
  <capability>usb</capability>
</host_device>
```

### Table 6.315. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
<td>HostDevice</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.100.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.101. HOSTDEVICES

A service to access host devices.

#### Table 6.316. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>List the devices of a host.</td>
</tr>
</tbody>
</table>

#### 6.101.1. list GET

List the devices of a host.

The order of the returned list of devices isn’t guaranteed.

### Table 6.317. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>devices</td>
<td>HostDevice[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.101.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.101.2. max

Sets the maximum number of devices to return. If not specified all the devices are returned.

### 6.102. HOSTHOOK

#### Table 6.318. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>get</strong></td>
<td></td>
</tr>
</tbody>
</table>

#### 6.102.1. get GET

#### Table 6.319. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>follow</strong></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td><strong>hook</strong></td>
<td>Hook</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.102.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.103. HOSTHOOKS

#### Table 6.320. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>list</strong></td>
<td>Returns the list of hooks configured for the host.</td>
</tr>
</tbody>
</table>

#### 6.103.1. list GET

Returns the list of hooks configured for the host.
The order of the returned list of hooks isn’t guaranteed.

Table 6.321. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>hooks</td>
<td>Hook[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of hooks to return.</td>
</tr>
</tbody>
</table>

6.103.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.103.1.2. max

Sets the maximum number of hooks to return. If not specified all the hooks are returned.

6.104. HOSTNIC

A service to manage a network interface of a host.

Table 6.322. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>updatevirtualfunctionconfig</td>
<td>The action updates virtual function configuration in case the current resource represents an SR-IOV enabled NIC.</td>
</tr>
</tbody>
</table>

6.104.1. get GET

Table 6.323. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>nic</td>
<td>HostNic</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.104.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.104.2. updatevirtualfunctionsconfiguration POST

The action updates virtual function configuration in case the current resource represents an SR-IOV enabled NIC. The input should be consisted of at least one of the following properties:

- allNetworksAllowed
- numberOfVirtualFunctions

Please see the `HostNicVirtualFunctionsConfiguration` type for the meaning of the properties.

Table 6.324. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>virtual_functions_configuration</td>
<td>HostNicVirtualFunctionsConfiguration</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.105. HOSTNICS

A service to manage the network interfaces of a host.

Table 6.325. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of network interfaces of the host.</td>
</tr>
</tbody>
</table>

6.105.1. list GET

Returns the list of network interfaces of the host.

The order of the returned list of network interfaces isn't guaranteed.

Table 6.326. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of NICs to return.</td>
</tr>
<tr>
<td>nics</td>
<td>HostNic[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.105.1.2. max

Sets the maximum number of NICs to return. If not specified all the NICs are returned.

6.106. HOSTNUMANODE

Table 6.327. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

6.106.1. get GET

Table 6.328. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>node</td>
<td>NumaNode</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.106.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.107. HOSTNUMANODES

Table 6.329. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of NUMA nodes of the host.</td>
</tr>
</tbody>
</table>

6.107.1. list GET

Returns the list of NUMA nodes of the host.

The order of the returned list of NUMA nodes isn’t guaranteed.

Table 6.330. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of nodes to return.</td>
</tr>
<tr>
<td>nodes</td>
<td>NumaNode[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.107.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.107.1.2. max

Sets the maximum number of nodes to return. If not specified all the nodes are returned.

### 6.108. HOSTSTORAGE

A service to manage host storages.

#### Table 6.331. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Get list of storages.</td>
</tr>
</tbody>
</table>

#### 6.108.1. list GET

Get list of storages.

```
GET /ovirt-engine/api/hosts/123/storage
```

The XML response you get will be like this one:

```xml
<host_storages>
  <host_storage id="123">
    ...
  </host_storage>
  ...
</host_storages>
```

The order of the returned list of storages isn’t guaranteed.

#### Table 6.332. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>report_status</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the status of the LUNs in the storage should be checked.</td>
</tr>
<tr>
<td>storages</td>
<td>HostStorage[]</td>
<td>Out</td>
<td>Retrieved list of storages.</td>
</tr>
</tbody>
</table>

### 6.108.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.108.1.2. report_status

Indicates if the status of the LUNs in the storage should be checked. Checking the status of the LUN is an heavy weight operation and this data is not always needed by the user. This parameter will give the option to not perform the status check of the LUNs.

The default is `true` for backward compatibility.

Here an example with the LUN status:

```xml
<host_storage id="123">
  <logical_units>
    <logical_unit id="123">
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
      <serial>123</serial>
      <size>10737418240</size>
      <status>used</status>
      <vendor_id>LIO-ORG</vendor_id>
      <volume_group_id>123</volume_group_id>
    </logical_unit>
  </logical_units>
  <type>iscsi</type>
  <host id="123"/>
</host_storage>
```

Here an example without the LUN status:

```xml
<host_storage id="123">
  <logical_units>
    <logical_unit id="123">
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
      <serial>123</serial>
      <size>10737418240</size>
      <vendor_id>LIO-ORG</vendor_id>
      <volume_group_id>123</volume_group_id>
    </logical_unit>
  </logical_units>
  <type>iscsi</type>
  <host id="123"/>
</host_storage>
```
6.109. HOSTS

A service that manages hosts.

Table 6.333. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new host.</td>
</tr>
<tr>
<td>list</td>
<td>Get a list of all available hosts.</td>
</tr>
</tbody>
</table>

6.109.1. add POST

Creates a new host.

The host is created based on the attributes of the host parameter. The name, address and root_password properties are required.

For example, to add a host send the following request:

```
POST /ovirt-engine/api/hosts
```

With the following request body:

```
<host>
  <name>myhost</name>
  <address>myhost.example.com</address>
  <root_password>myrootpassword</root_password>
</host>
```

NOTE

The root_password element is only included in the client-provided initial representation and is not exposed in the representations returned from subsequent requests.

IMPORTANT

Since version 4.1.2 of the engine when a host is newly added we override the host firewall definitions by default.

To add a hosted engine host, use the optional deploy_hosted_engine parameter:

```
POST /ovirt-engine/api/hosts?deploy_hosted_engine=true
```
If the cluster has a default external network provider which is supported for automatic deployment, the external network provider is deployed when adding the host. Only external network providers for OVN are supported for the automatic deployment. To deploy an external network provider that differs to what is defined in the clusters, overwrite the external network provider when adding hosts by sending a request like this:

```
POST /ovirt-engine/api/hosts
```

With a request body that contains a reference to the desired provider in the `external_network_provider_configuration`:

```
<host>
  <name>myhost</name>
  <address>myhost.example.com</address>
  <root_password>123456</root_password>
  <external_network_provider_configurations>
    <external_network_provider_configuration>
      <external_network_provider name="ovirt-provider-ovn"/>
    </external_network_provider_configuration>
  </external_network_provider_configurations>
</host>
```

Table 6.334. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>deploy_hosted_engine</td>
<td>Boolean</td>
<td>In</td>
<td>When set to true it means this host should deploy also hosted engine components.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>In/Out</td>
<td>The host definition from which to create the new host is passed as parameter, and the newly created host is returned.</td>
</tr>
<tr>
<td>undeploy_hosted_engine</td>
<td>Boolean</td>
<td>In</td>
<td>When set to true it means this host should un-deploy hosted engine components and this host will not function as part of the High Availability cluster.</td>
</tr>
</tbody>
</table>

6.109.1.1. deploy_hosted_engine

When set to true it means this host should deploy also hosted engine components. Missing value is treated as true i.e deploy. Omitting this parameter means false and will perform no operation in hosted engine area.

6.109.1.2. undeploy_hosted_engine

When set to true it means this host should un-deploy hosted engine components and this host will not function as part of the High Availability cluster. Missing value is treated as true i.e un-deploy. Omitting this parameter means false and will perform no operation in hosted engine area.

6.109.2. list GET
Get a list of all available hosts.

For example, to list the hosts send the following request:

```
GET /ovirt-engine/api/hosts
```

The response body will be something like this:

```xml
<hosts>
  <host href="/ovirt-engine/api/hosts/123" id="123">
    ...
  </host>
  <host href="/ovirt-engine/api/hosts/456" id="456">
    ...
  </host>
  ...
</hosts>
```

The order of the returned list of hosts is guaranteed only if the `sortby` clause is included in the `search` parameter.

### Table 6.335. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>all_content</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all of the attributes of the hosts should be included in the response.</td>
</tr>
<tr>
<td><code>case_sensitive</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <code>search</code> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td><code>filter</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td><code>follow</code></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td><code>hosts</code></td>
<td>Host[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td><code>max</code></td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of hosts to return.</td>
</tr>
<tr>
<td><code>search</code></td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned hosts.</td>
</tr>
</tbody>
</table>

#### 6.109.2.1. all_content

Indicates if all of the attributes of the hosts should be included in the response.

By default the following host attributes are excluded:

- `hosted_engine`

For example, to retrieve the complete representation of the hosts:
GET /ovirt-engine/api/hosts?all_content=true

NOTE
These attributes are not included by default because retrieving them impacts performance. They are seldom used and require additional queries to the database. Use this parameter with caution and only when specifically required.

6.109.2.2. case_sensitive
Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.

6.109.2.3. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.109.2.4. max
Sets the maximum number of hosts to return. If not specified all the hosts are returned.

6.110. ICON
A service to manage an icon (read-only).

Table 6.336. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get an icon.</td>
</tr>
</tbody>
</table>

6.110.1. get GET
Get an icon.

GET /ovirt-engine/api/icons/123

You will get a XML response like this one:

```xml
<icon id="123">
  <data>Some binary data here</data>
  <media_type>image/png</media_type>
</icon>
```

Table 6.337. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>icon</td>
<td>Icon</td>
<td>Out</td>
<td>Retrieved icon.</td>
</tr>
</tbody>
</table>

### 6.110.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.111. ICONS

A service to manage icons.

**Table 6.338. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Get a list of icons.</td>
</tr>
</tbody>
</table>

#### 6.111.1. list GET

Get a list of icons.

```
GET /ovirt-engine/api/icons
```

You will get a XML response which is similar to this one:

```xml
<icons>
  <icon id="123">
    <data>...</data>
    <media_type>image/png</media_type>
  </icon>
  ...
</icons>
```

The order of the returned list of icons isn’t guaranteed.

**Table 6.339. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>icons</td>
<td>Icon[]</td>
<td>Out</td>
<td>Retrieved list of icons.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of icons to return.</td>
</tr>
</tbody>
</table>
6.111.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.111.1.2. max

Sets the maximum number of icons to return. If not specified all the icons are returned.

6.112. IMAGE

Table 6.340. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>Imports an image.</td>
</tr>
</tbody>
</table>

6.112.1. get GET

Table 6.341. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>image</td>
<td>Image</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.112.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.112.2. import POST

Imports an image.

If the import_as_template parameter is true then the image will be imported as a template, otherwise it will be imported as a disk.

When imported as a template, the name of the template can be specified by the optional template.name parameter. If that parameter is not specified, then the name of the template will be automatically assigned by the engine as GlanceTemplate-x (where x will be seven random hexadecimal characters).

When imported as a disk, the name of the disk can be specified by the optional disk.name parameter. If that parameter is not specified, then the name of the disk will be automatically assigned by the engine as GlanceDisk-x (where x will be the seven hexadecimal characters of the image identifier).
It is recommended to always explicitly specify the template or disk name, to avoid these automatic names generated by the engine.

### Table 6.342. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the import should be performed asynchronously.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td>The cluster to which the image should be imported if the \texttt{import_as_template} parameter is set to \texttt{true}.</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In</td>
<td>The disk to import.</td>
</tr>
<tr>
<td>import_as_template</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if a template should be created from the imported disk.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDom_ain</td>
<td>In</td>
<td>The storage domain to which the disk should be imported.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In</td>
<td>The name of the template being created if the \texttt{import_as_template} parameter is set to \texttt{true}.</td>
</tr>
</tbody>
</table>

### 6.113. IMAGETRANSFER

This service provides a mechanism to control an image transfer. The client will have to create a transfer by using \texttt{add} of the Section 6.114, "ImageTransfers" service, stating the image to transfer data to/from.

After doing that, the transfer is managed by this service.

**Using oVirt's Python's SDK:**

Uploading a \texttt{disk} with id 123 (on a random host in the data center):

```python
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(types.ImageTransfer(
    disk=types.Disk(id='123'))
)
```

Uploading a \texttt{disk} with id 123 on \texttt{host} id 456:

```python
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(types.ImageTransfer(
    disk=types.Disk(id='123'))
)
If the user wishes to download a disk rather than upload, he/she should specify *download* as the *direction* attribute of the transfer. This will grant a read permission from the image, instead of a write permission.

E.g:

```python
transfers_service = system_service.image_transfers_service()
transfer = transfers_service.add(
    types.ImageTransfer(
        disk=types.Disk(
            id='123'
        ),
        direction=types.ImageTransferDirection.DOWNLOAD
    )
)
```

Transfers have phases, which govern the flow of the upload/download. A client implementing such a flow should poll/check the transfer's phase and act accordingly. All the possible phases can be found in `ImageTransferPhase`.

After adding a new transfer, its phase will be *initializing*. The client will have to poll on the transfer's phase until it changes. When the phase becomes *transferring*, the session is ready to start the transfer.

For example:

```python
transfer_service = transfers_service.image_transfer_service(transfer.id)
while transfer.phase == types.ImageTransferPhase.INITIALIZING:
    time.sleep(3)
    transfer = transfer_service.get()
```

At that stage, if the transfer’s phase is *paused_system*, then the session was not successfully established. One possible reason for that is that the ovirt-imageio-daemon is not running in the host that was selected for transfer. The transfer can be resumed by calling `resume` of the service that manages it.

If the session was successfully established - the returned transfer entity will contain the *proxy_url* and *signed_ticket* attributes, which the client needs to use in order to transfer the required data. The client can choose whatever technique and tool for sending the HTTPS request with the image’s data.

- *proxy_url* is the address of a proxy server to the image, to do I/O to.
- *signed_ticket* is the content that needs to be added to the *Authentication* header in the HTTPS request, in order to perform a trusted communication.

For example, Python’s HTTPSConnection can be used in order to perform a transfer, so an *transfer_headers* dict is set for the upcoming transfer:
Using Python's `HTTPSConnection`, a new connection is established:

```python
transfer_headers = {
    'Authorization': transfer.signed_ticket,
}
```

For upload, the specific content range being sent must be noted in the `Content-Range` HTTPS header. This can be used in order to split the transfer into several requests for a more flexible process.

For doing that, the client will have to repeatedly extend the transfer session to keep the channel open. Otherwise, the session will terminate and the transfer will get into `paused_system` phase, and HTTPS requests to the server will be rejected.

E.g., the client can iterate on chunks of the file, and send them to the proxy server while asking the service to extend the session:

```python
path = '/path/to/image'
MB_per_request = 32
with open(path, 'rb') as disk:
    size = os.path.getsize(path)
    chunk_size = 1024 * 1024 * MB_per_request
    pos = 0
    while (pos < size):
        transfer_service.extend()
        transfer_headers['Content-Range'] = 'bytes %d-%d/%d' % (pos, min(pos + chunk_size, size) - 1, size)
        proxy_connection.request('PUT', url.path, disk.read(chunk_size), headers=transfer_headers)
        r = proxy_connection.getresponse()
        print r.status, r.reason, "Completed", "{:0%}".format(pos / float(size))
        pos += chunk_size
```

Similarly, for a download transfer, a `Range` header must be sent, making the download process more easily managed by downloading the disk in chunks.

E.g., the client will again iterate on chunks of the disk image, but this time he/she will download it to a local file, rather than uploading its own file to the image:

```python
output_file = '/home/user/downloaded_image'
MiB_per_request = 32
```
When finishing the transfer, the user should call `finalize`. This will make the final adjustments and verifications for finishing the transfer process.

For example:

```python
chunk_size = 1024*1024*MiB_per_request
total = disk_size

with open(output_file, "wb") as disk:
    pos = 0
    while pos < total:
        transfer_service.extend()
        transfer_headers["Range"] = "bytes=%d-%d" % (pos, min(total, pos + chunk_size) - 1)
        proxy_connection.request('GET', proxy_url.path, headers=transfer_headers)
        r = proxy_connection.getresponse()
        disk.write(r.read())
        print "Completed", "{:0%}".format(pos/ float(total))
        pos += chunk_size
```

When finishing the transfer, the user should call `finalize`. This will make the final adjustments and verifications for finishing the transfer process.

For example:

```python
transfer_service.finalize()
```

In case of an error, the transfer's phase will be changed to `finished_failure`, and the disk's status will be changed to `Illegal`. Otherwise it will be changed to `finished_success`, and the disk will be ready to be used. In both cases, the transfer entity will be removed shortly after.

**Using HTTP and cURL calls:**

- For upload, create a new disk first:
  - Specify `initial_size` and `provisioned_size` in bytes.
  - `initial_size` must be bigger or the same as the size of the uploaded data.

```python
POST /ovirt-engine/api/disks
```

With a request body as follows:

```xml
<disk>
    <storage_domains>
        <storage_domain id="123"/>
    </storage_domains>
    <alias>mydisk</alias>
    <initial_size>1073741824</initial_size>
    <provisioned_size>1073741824</provisioned_size>
    <format>raw</format>
</disk>
```

- Create a new image transfer for downloading/uploading a `disk` with id 456:

```python
POST /ovirt-engine/api/imagetransfers
```

With a request body as follows:

```xml
```
Will respond:

```
<image_transfer id="123">
  <direction>download|upload</direction>
  <phase>initializing|transferring</phase>
  <proxy_url>https://proxy_fqdn:54323/images/41c732d4-2210-4e7b-9e5c-4e2805baadbb</proxy_url>
  <transfer_url>https://daemon_fqdn:54322/images/41c732d4-2210-4e7b-9e5c-4e2805baadbb</transfer_url>
...</image_transfer>
```

Note: If the phase is ‘initializing’, poll the `image_transfer` till its phase changes to ‘transferring’.

- Use the ‘transfer_url’ or ‘proxy_url’ to invoke a curl command:

  - use ‘transfer_url’ for transferring directly from/to ovirt-imageio-daemon, or, use ‘proxy_url’ for transferring from/to ovirt-imageio-proxy. Note: using the proxy would mitigate scenarios where there’s no direct connectivity to the daemon machine, e.g. vdsm machines are on a different network than the engine.

— Download:

```
$ curl --cacert /etc/pki/ovirt-engine/ca.pem https://daemon_fqdn:54322/images/41c732d4-2210-4e7b-9e5c-4e2805baadbb -o <output_file>
```

— Upload:

```
$ curl --cacert /etc/pki/ovirt-engine/ca.pem --upload-file <file_to_upload> -X PUT https://daemon_fqdn:54322/images/41c732d4-2210-4e7b-9e5c-4e2805baadbb
```

- Finalize the image transfer by invoking the action:

```
POST /ovirt-engine/api/imagetransfers/123/finalize
```

With a request body as follows:

```
<action />
```

### Table 6.343. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancel</td>
<td>Cancel the image transfer session.</td>
</tr>
</tbody>
</table>
6.113.1. cancel POST

Cancel the image transfer session. This terminates the transfer operation and removes the partial image.

6.113.2. extend POST

Extend the image transfer session.

6.113.3. finalize POST

After finishing to transfer the data, finalize the transfer.

This will make sure that the data being transferred is valid and fits the image entity that was targeted in the transfer. Specifically, will verify that if the image entity is a QCOW disk, the data uploaded is indeed a QCOW file, and that the image doesn’t have a backing file.

6.113.4. get GET

Get the image transfer entity.

Table 6.344. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>image_transfer</td>
<td>ImageTransfer</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.113.4.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.113.5. pause POST

Pause the image transfer session.
6.113.6. resume POST

Resume the image transfer session. The client will need to poll the transfer’s phase until it is different than `resuming`. For example:

```python
transfer_service = transfers_service.image_transfer_service(transfer.id)
transfer_service.resume()
transfer = transfer_service.get()

while transfer.phase == types.ImageTransferPhase.RESUMING:
    time.sleep(1)
    transfer = transfer_service.get()
```

6.114. IMAGETRANSFERS

This service manages image transfers, for performing Image I/O API in Red Hat Virtualization. Please refer to [image transfer](https://ovirt-engine.readthedocs.io/en/4.2/ovirt-engine/api/index.html#image-transfer) for further documentation.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new image transfer.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieves the list of image transfers that are currently being performed.</td>
</tr>
</tbody>
</table>

6.114.1. add POST

Add a new image transfer. An image, disk or disk snapshot needs to be specified in order to make a new transfer.

**IMPORTANT**

The `image` attribute is deprecated since version 4.2 of the engine. Use the `disk` or `snapshot` attributes instead.

Creating a new image transfer for downloading or uploading a disk:

To create an image transfer to download or upload a disk with id 123, send the following request:

```xml
POST /ovirt-engine/api/imagetransfers
```

With a request body like this:

```xml
<image_transfer>
  <disk id="123"/>
  <direction>upload|download</direction>
</image_transfer>
```

Creating a new image transfer for downloading or uploading a disk_snapshot:
To create an image transfer to download or upload a disk snapshot with id 456, send the following request:

```plaintext
POST /ovirt-engine/api/imagetransfers
```

With a request body like this:

```xml
<image_transfer>
  <snapshot id="456"/>
  <direction>download|upload</direction>
</image_transfer>
```

### Table 6.346. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>image_transfer</td>
<td>ImageTransfer</td>
<td>In/Out</td>
<td>The image transfer to add.</td>
</tr>
</tbody>
</table>

### 6.114.2. list GET

Retrieves the list of image transfers that are currently being performed.

The order of the returned list of image transfers is not guaranteed.

### Table 6.347. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>image_transfer</td>
<td>ImageTransfer[]</td>
<td>Out</td>
<td>A list of image transfers that are currently being performed.</td>
</tr>
</tbody>
</table>

### 6.114.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.115. IMAGES

Manages the set of images available in an storage domain or in an OpenStack image provider.

### Table 6.348. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of images available in the storage domain or provider.</td>
</tr>
</tbody>
</table>
6.115.1. list GET

Returns the list of images available in the storage domain or provider.

The order of the returned list of images isn’t guaranteed.

Table 6.349. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>images</td>
<td>Image[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of images to return.</td>
</tr>
</tbody>
</table>

6.115.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.115.1.2. max

Sets the maximum number of images to return. If not specified all the images are returned.

6.116. INSTANCETYPE

Table 6.350. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get a specific instance type and it's attributes.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a specific instance type from the system.</td>
</tr>
<tr>
<td>update</td>
<td>Update a specific instance type and it's attributes.</td>
</tr>
</tbody>
</table>

6.116.1. get GET

Get a specific instance type and it's attributes.

GET /ovirt-engine/api/instancetypes/123

Table 6.351. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.116.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.116.2. remove DELETE

Removes a specific instance type from the system.

If a virtual machine was created using an instance type X after removal of the instance type the virtual machine’s instance type will be set to custom.

DELETE /ovirt-engine/api/instancetypes/123

### Table 6.352. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.116.3. update PUT

Update a specific instance type and its attributes.

All the attributes are editable after creation. If a virtual machine was created using an instance type X and some configuration in instance type X was updated, the virtual machine’s configuration will be updated automatically by the engine.

PUT /ovirt-engine/api/instancetypes/123

For example, to update the memory of instance type 123 to 1 GiB and set the cpu topology to 2 sockets and 1 core, send a request like this:

```xml
<instance_type>
  <memory>1073741824</memory>
  <cpu>
    <topology>
      <cores>1</cores>
      <sockets>2</sockets>
      <threads>1</threads>
    </topology>
  </cpu>
</instance_type>
```
Table 6.353. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.117. INSTANCETYPEGRAPHICSCONSOLE

Table 6.354. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets graphics console configuration of the instance type.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the graphics console from the instance type.</td>
</tr>
</tbody>
</table>

6.117.1. get GET

Gets graphics console configuration of the instance type.

Table 6.355. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>GraphicsConsole</td>
<td>Out</td>
<td>The information about the graphics console of the instance type.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed. The objects referenced</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>by these links will be fetched as part of the current request.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See here for details.</td>
</tr>
</tbody>
</table>

6.117.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.117.2. remove DELETE

Remove the graphics console from the instance type.

Table 6.356. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>
### 6.118. INSTANCETYPEGRAPHICSCONSOLES

#### Table 6.357. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add new graphics console to the instance type.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all the configured graphics consoles of the instance type.</td>
</tr>
</tbody>
</table>

#### 6.118.1. add POST

Add new graphics console to the instance type.

#### Table 6.358. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>GraphicsConsole</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.118.2. list GET

Lists all the configured graphics consoles of the instance type.

The order of the returned list of graphics consoles isn’t guaranteed.

#### Table 6.359. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>consoles</td>
<td>GraphicsConsole[]</td>
<td>Out</td>
<td>The list of graphics consoles of the instance type.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of consoles to return.</td>
</tr>
</tbody>
</table>

#### 6.118.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.118.2.2. max

Sets the maximum number of consoles to return. If not specified all the consoles are returned.

### 6.119. INSTANCETYPEPENIC
Table 6.360. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets network interface configuration of the instance type.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the network interface from the instance type.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the network interface configuration of the instance type.</td>
</tr>
</tbody>
</table>

6.119.1. get GET

Gets network interface configuration of the instance type.

Table 6.361. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>nic</td>
<td>Nic</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.119.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.119.2. remove DELETE

Remove the network interface from the instance type.

Table 6.362. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.119.3. update PUT

Updates the network interface configuration of the instance type.

Table 6.363. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
</tbody>
</table>
### 6.120. INSTANCETYPENICS

#### Table 6.364. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add new network interface to the instance type.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all the configured network interface of the instance type.</td>
</tr>
</tbody>
</table>

#### 6.120.1. add POST

Add new network interface to the instance type.

#### Table 6.365. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>nic</td>
<td>Nic</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.120.2. list GET

Lists all the configured network interface of the instance type.

The order of the returned list of network interfaces isn't guaranteed.

#### Table 6.366. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of NICs to return.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>Out</td>
<td>A query string used to restrict the returned templates.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.120.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.120.2.2. max

Sets the maximum number of NICs to return. If not specified all the NICs are returned.

6.121. INSTANCETYPEWATCHDOG

Table 6.367. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets watchdog configuration of the instance type.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove a watchdog from the instance type.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the watchdog configuration of the instance type.</td>
</tr>
</tbody>
</table>

6.121.1. get GET

Gets watchdog configuration of the instance type.

Table 6.368. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.121.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.121.2. remove DELETE

Remove a watchdog from the instance type.

Table 6.369. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.121.3. update PUT

Updates the watchdog configuration of the instance type.

Table 6.370. Parameters summary
### 6.122. INSTANCETYPEWATCHDOGS

#### Table 6.371. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add new watchdog to the instance type.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all the configured watchdogs of the instance type.</td>
</tr>
</tbody>
</table>

#### 6.122.1. add POST

Add new watchdog to the instance type.

#### Table 6.372. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.122.2. list GET

Lists all the configured watchdogs of the instance type.

The order of the returned list of watchdogs isn’t guaranteed.

#### Table 6.373. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of watchdogs to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned templates.</td>
</tr>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.122.2.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.122.2.2. max

Sets the maximum number of watchdogs to return. If not specified all the watchdogs are returned.

6.123. INSTANCETYPES

Table 6.374. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new instance type.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all existing instance types in the system.</td>
</tr>
</tbody>
</table>

6.123.1. add POST

Creates a new instance type.

This requires only a name attribute and can include all hardware configurations of the virtual machine.

POST /ovirt-engine/api/instancetypes

With a request body like this:

```xml
<instance_type>
  <name>myinstancetype</name>
</template>
```

Creating an instance type with all hardware configurations with a request body like this:

```xml
<instance_type>
  <name>myinstancetype</name>
  <console>
    <enabled>true</enabled>
  </console>
  <cpu>
    <topology>
      <cores>2</cores>
      <sockets>2</sockets>
      <threads>1</threads>
    </topology>
  </cpu>
  <custom_cpu_model>AMD Opteron_G2</custom_cpu_model>
  <custom_emulated_machine>q35</custom_emulated_machine>
  <display>
    <monitors>1</monitors>
    <single_qxl_pci>true</single_qxl_pci>
    <smartcard_enabled>true</smartcard_enabled>
    <type>spice</type>
</instance_type>
```
Table 6.375. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.123.2. list GET

Lists all existing instance types in the system.
The order of the returned list of instance types isn’t guaranteed.

### Table 6.376. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <code>search</code> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of instance types to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned templates.</td>
</tr>
</tbody>
</table>

#### 6.123.2.1. `case_sensitive`

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

#### 6.123.2.2. `follow`

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.123.2.3. `max`

Sets the maximum number of instance types to return. If not specified all the instance types are returned.

### 6.124. ISCSIBOND

#### Table 6.377. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes of an existing iSCSI bond.</td>
</tr>
<tr>
<td>update</td>
<td>Updates an iSCSI bond.</td>
</tr>
</tbody>
</table>

#### 6.124.1. `get` GET
### Table 6.378. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond</td>
<td>IscsiBond</td>
<td>Out</td>
<td>The iSCSI bond.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.124.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.124.2. remove DELETE

Removes of an existing iSCSI bond.

For example, to remove the iSCSI bond 456 send a request like this:

```
DELETE /ovirt-engine/api/datacenters/123/iscsibonds/456
```

### Table 6.379. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.124.3. update PUT

Updates an iSCSI bond.

Updating of an iSCSI bond can be done on the *name* and the *description* attributes only. For example, to update the iSCSI bond 456 of data center 123, send a request like this:

```
PUT /ovirt-engine/api/datacenters/123/iscsibonds/1234
```

The request body should look like this:

```
<iscsi_bond>
   <name>mybond</name>
   <description>My iSCSI bond</description>
</iscsi_bond>
```

### Table 6.380. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>-----------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>bond</td>
<td>IscsiBond</td>
<td>In/Out</td>
<td>The iSCSI bond to update.</td>
</tr>
</tbody>
</table>

### 6.125. ISCSIBONDS

#### Table 6.381. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Create a new iSCSI bond on a data center.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of iSCSI bonds configured in the data center.</td>
</tr>
</tbody>
</table>

#### 6.125.1. add POST

Create a new iSCSI bond on a data center.

For example, to create a new iSCSI bond on data center 123 using storage connections 456 and 789, send a request like this:

```
POST /ovirt-engine/api/datacenters/123/iscsibonds
```

The request body should look like this:

```xml
<iscsi_bond>
  <name>mybond</name>
  <storage_connections>
    <storage_connection id="456"/>
    <storage_connection id="789"/>
  </storage_connections>
  <networks>
    <network id="abc"/>
  </networks>
</iscsi_bond>
```

#### Table 6.382. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bond</td>
<td>IscsiBond</td>
<td>In/Out</td>
</tr>
</tbody>
</table>

#### 6.125.2. list GET

Returns the list of iSCSI bonds configured in the data center.

The order of the returned list of iSCSI bonds isn’t guaranteed.

#### Table 6.383. Parameters summary
### 6.125.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.125.2.2. max

Sets the maximum number of bonds to return. If not specified all the bonds are returned.

### 6.126. JOB

A service to manage a job.

**Table 6.384. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>Set an external job execution to be cleared by the system.</td>
</tr>
<tr>
<td>end</td>
<td>Marks an external job execution as ended.</td>
</tr>
<tr>
<td>get</td>
<td>Retrieves a job.</td>
</tr>
</tbody>
</table>

**6.126.1. clear POST**

Set an external job execution to be cleared by the system.

For example, to set a job with identifier 123 send the following request:

```
POST /ovirt-engine/api/jobs/clear
```

With the following request body:

```
<action/>
```

**Table 6.385. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bonds</td>
<td>IscsiBond[]</td>
<td>Out</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of bonds to return.</td>
</tr>
</tbody>
</table>
### 6.126.2. end POST

Marks an external job execution as ended.

For example, to terminate a job with identifier **123** send the following request:

```
POST /ovirt-engine/api/jobs/end
```

With the following request body:

```
<action>
  <force>true</force>
  <status>finished</status>
</action>
```

### Table 6.38. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the job should be forcibly terminated.</td>
</tr>
<tr>
<td>succeeded</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the job should be marked as successfully finished or as failed.</td>
</tr>
</tbody>
</table>

#### 6.126.2.1. succeeded

Indicates if the job should be marked as successfully finished or as failed.

This parameter is optional, and the default value is **true**.

### 6.126.3. get GET

Retrieves a job.

```
GET /ovirt-engine/api/jobs/123
```

You will receive response in XML like this one:

```
<job href="/ovirt-engine/api/jobs/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
    <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
  </actions>
</job>
```
6.126.3.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.127. JOBS

A service to manage jobs.

Table 6.388. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add an external job.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieves the representation of the jobs.</td>
</tr>
</tbody>
</table>

6.127.1. add POST

Add an external job.

For example, to add a job with the following request:

```plaintext
POST /ovirt-engine/api/jobs
```

With the following request body:

```xml
<job>
  <description>Doing some work</description>
  <auto_cleared>true</auto_cleared>
</job>
```
The response should look like:

```xml
<job href="/ovirt-engine/api/jobs/123" id="123">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
    <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
  </actions>
  <description>Doing some work</description>
  <auto_cleared>true</auto_cleared>
  <external>true</external>
  <last_updated>2016-12-13T02:15:42.130+02:00</last_updated>
  <start_time>2016-12-13T02:15:42.130+02:00</start_time>
  <status>started</status>
  <owner href="/ovirt-engine/api/users/456" id="456"/>
</job>
```

Table 6.389. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>job</td>
<td>Job</td>
<td>In/Out</td>
<td>Job that will be added.</td>
</tr>
</tbody>
</table>

6.127.2. list GET

Retrieves the representation of the jobs.

```
GET /ovirt-engine/api/jobs
```

You will receive response in XML like this one:

```xml
<jobs>
  <job href="/ovirt-engine/api/jobs/123" id="123">
    <actions>
      <link href="/ovirt-engine/api/jobs/123/clear" rel="clear"/>
      <link href="/ovirt-engine/api/jobs/123/end" rel="end"/>
    </actions>
    <description>Adding Disk</description>
    <auto_cleared>true</auto_cleared>
    <end_time>2016-12-12T23:07:29.758+02:00</end_time>
    <external>false</external>
    <last_updated>2016-12-12T23:07:29.758+02:00</last_updated>
    <start_time>2016-12-12T23:07:26.593+02:00</start_time>
    <status>failed</status>
    <owner href="/ovirt-engine/api/users/456" id="456"/>
  </job>
  ...
</jobs>
```

The order of the returned list of jobs isn’t guaranteed.

Table 6.390. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>jobs</td>
<td>Job[]</td>
<td>Out</td>
<td>A representation of jobs.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of jobs to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned jobs.</td>
</tr>
</tbody>
</table>

### 6.127.2.1. case_sensitive

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

### 6.127.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.127.2.3. max

Sets the maximum number of jobs to return. If not specified all the jobs are returned.

### 6.128. KATELLOERRATA

A service to manage Katello errata. The information is retrieved from Katello.

#### Table 6.391. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Retrieves the representation of the Katello errata.</td>
</tr>
</tbody>
</table>

### 6.128.1. list GET

Retrieves the representation of the Katello errata.

```plaintext
GET /ovirt-engine/api/katelloerrata
```

You will receive response in XML like this one:

```xml
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/123" id="123">
```

293
Table 6.392. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>errata</td>
<td>KatelloErratum[]</td>
<td>Out</td>
<td>A representation of Katello errata.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of errata to return.</td>
</tr>
</tbody>
</table>

6.128.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.128.1.2. max

Sets the maximum number of errata to return. If not specified all the errata are returned.

6.129. KATELLOERRATUM

A service to manage a Katello erratum.

Table 6.393. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves a Katello erratum.</td>
</tr>
</tbody>
</table>

6.129.1. get GET
Retrieves a Katello erratum.

```bash
GET /ovirt-engine/api/katelloerrata/123
```

You will receive response in XML like this one:

```xml
<katello_erratum href="/ovirt-engine/api/katelloerrata/123" id="123">
  <name>RHBA-2013:XYZ</name>
  <description>The description of the erratum</description>
  <title>some bug fix update</title>
  <type>bugfix</type>
  <issued>2013-11-20T02:00:00.000+02:00</issued>
  <solution>Few guidelines regarding the solution</solution>
  <summary>Updated packages that fix one bug are now available for XYZ</summary>
  <packages>
    <package>
      <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
    </package>
    ...
  </packages>
</katello_erratum>
```

Table 6.394. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>erratum</td>
<td>KatelloErratum</td>
<td>Out</td>
<td>Retrieves the representation of the Katello erratum.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.129.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.130. LINKLAYERDISCOVERYPROTOCOL

A service to fetch information elements received by Link Layer Discovery Protocol (LLDP).

Table 6.395. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Fetches information elements received by LLDP.</td>
</tr>
</tbody>
</table>

6.130.1. list GET

Fetches information elements received by LLDP.
Table 6.396. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>elements</td>
<td>LinkLayerDiscoveryProtocolElement[]</td>
<td>Out</td>
<td>Retrieves a list of information elements received by LLDP.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.130.1.1. elements

Retrieves a list of information elements received by LLDP.

For example, to retrieve the information elements received on the NIC 321 on host 123, send a request like this:

```plaintext
GET ovirt-engine/api/hosts/123/nics/321/linklayerdiscoveryprotocolelements
```

It will return a response like this:

```xml
<link_layer_discovery_protocol_elements>
  ...
  <link_layer_discovery_protocol_element>
    <name>Port Description</name>
    <properties>
      <property>
        <name>port description</name>
        <value>Summit300-48-Port 1001</value>
      </property>
    </properties>
    <type>4</type>
  </link_layer_discovery_protocol_element>
  ...
</link_layer_discovery_protocol_elements>
```

6.130.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.131. MACPOOL

Table 6.397. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes a MAC address pool.</td>
</tr>
</tbody>
</table>
Table 6.398. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>pool</td>
<td>MacPool</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.131.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.131.2. remove DELETE

Removes a MAC address pool.

For example, to remove the MAC address pool having id 123 send a request like this:

```
DELETE /ovirt-engine/api/macpools/123
```

Table 6.399. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.131.3. update PUT

Updates a MAC address pool.

The *name*, *description*, *allow_duplicates*, and *ranges* attributes can be updated.

For example, to update the MAC address pool of id 123 send a request like this:

```
PUT /ovirt-engine/api/macpools/123
```

With a request body like this:

```
<mac_pool>
  <name>UpdatedMACPool</name>
  <description>An updated MAC address pool</description>
</mac_pool>
```
Table 6.400. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>pool</td>
<td>MacPool</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.132. MACPOOLS

Table 6.401. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new MAC address pool.</td>
</tr>
<tr>
<td>list</td>
<td>Return the list of MAC address pools of the system.</td>
</tr>
</tbody>
</table>

6.132.1. add POST

Creates a new MAC address pool.

Creation of a MAC address pool requires values for the name and ranges attributes.

For example, to create MAC address pool send a request like this:

```
POST /ovirt-engine/api/macpools
```

With a request body like this:

```
<mac_pool>
  <name>MACPool</name>
  <description>A MAC address pool</description>
  <allow_duplicates>true</allow_duplicates>
  <default_pool>false</default_pool>
  <ranges>
    <range>
      <from>00:1A:4A:16:01:51</from>
      <to>00:1A:4A:16:01:e6</to>
    </range>
    <range>
      <from>02:1A:4A:01:00:00</from>
      <to>02:1A:4A:FF:FF:FF</to>
    </range>
  </ranges>
</mac_pool>
```
Table 6.402. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool</td>
<td>MacPool</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.132.2. list GET

Return the list of MAC address pools of the system.

The returned list of MAC address pools isn’t guaranteed.

Table 6.403. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of pools to return.</td>
</tr>
<tr>
<td>pools</td>
<td>MacPool[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.132.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.132.2.2. max

Sets the maximum number of pools to return. If not specified all the pools are returned.

6.133. MEASURABLE

6.134. MOVEABLE

Table 6.404. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>move</td>
<td></td>
</tr>
</tbody>
</table>

6.134.1. move POST
### Table 6.405. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the move should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.135. NETWORK

A service managing a network

### Table 6.406. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets a logical network.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a logical network, or the association of a logical network to a data center.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a logical network.</td>
</tr>
</tbody>
</table>

### 6.135.1. get GET

Gets a logical network.

For example:

```
GET /ovirt-engine/api/networks/123
```

Will respond:

```xml
<network href="/ovirt-engine/api/networks/123" id="123">
  <name>ovirtmgmt</name>
  <description>Default Management Network</description>
  <link href="/ovirt-engine/api/networks/123/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/networks/123/vnicprofiles" rel="vnicprofiles"/>
  <link href="/ovirt-engine/api/networks/123/networklabels" rel="networklabels"/>
  <mtu>0</mtu>
  <stp>false</stp>
  <usages>
    <usage>vm</usage>
  </usages>
  <data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
</network>
```
### Name | Type | Direction  | Summary
--- | --- | --- | ---
follow | String | In | Indicates which inner links should be *followed*.  

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.135.1. follow
Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.135.2. remove DELETE
Removes a logical network, or the association of a logical network to a data center.

For example, to remove the logical network **123** send a request like this:

```plaintext
DELETE /ovirt-engine/api/networks/123
```

Each network is bound exactly to one data center. So if we disassociate network with data center it has the same result as if we would just remove that network. However it might be more specific to say we’re removing network **456** of data center **123**.

For example, to remove the association of network **456** to data center **123** send a request like this:

```plaintext
DELETE /ovirt-engine/api/datacenters/123/networks/456
```

**NOTE**
To remove an external logical network, the network has to be removed directly from its provider by [OpenStack Networking API](#). The entity representing the external network inside Red Hat Virtualization is removed automatically, if `auto_sync` is enabled for the provider, otherwise the entity has to be removed using this method.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.135.3. update PUT
Updates a logical network.

The `name`, `description`, `ip`, `vlan`, `stp` and `display` attributes can be updated.

For example, to update the description of the logical network **123** send a request like this:

```plaintext
PUT /ovirt-engine/api/networks/123
```
With a request body like this:

```xml
<network>
  <description>My updated description</description>
</network>
```

The maximum transmission unit of a network is set using a PUT request to specify the integer value of the `mtu` attribute.

For example, to set the maximum transmission unit send a request like this:

```
PUT /ovirt-engine/api/datacenters/123/networks/456
```

With a request body like this:

```xml
<network>
  <mtu>1500</mtu>
</network>
```

---

**Table 6.409. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

---

**6.136. NETWORKATTACHMENT**

**Table 6.410. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update the specified network attachment on the host.</td>
</tr>
</tbody>
</table>

---

**6.136.1. get GET**

**Table 6.411. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>NetworkAttachment</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
### 6.136.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.136.2. remove DELETE

**Table 6.412. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.136.3. update PUT

Update the specified network attachment on the host.

**Table 6.413. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>attachment</td>
<td>NetworkAttachment</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.137. NETWORKATTACHMENTS

Manages the set of network attachments of a host or host NIC.

**Table 6.414. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new network attachment to the network interface.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of network attachments of the host or host NIC.</td>
</tr>
</tbody>
</table>
### Table 6.415. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>NetworkAttachment</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.137.2. list GET

Returns the list of network attachments of the host or host NIC.

The order of the returned list of network attachments isn’t guaranteed.

### Table 6.416. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachments</td>
<td>NetworkAttachment[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of attachments to return.</td>
</tr>
</tbody>
</table>

#### 6.137.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.137.2.2. max

Sets the maximum number of attachments to return. If not specified all the attachments are returned.

### 6.138. NETWORKFILTER

Manages a network filter.

```xml
<network_filter id="00000019-0019-0019-0019-00000000026b">
  <name>example-network-filter-b</name>
  <version>
    <major>4</major>
    <minor>0</minor>
    <build>-1</build>
    <revision>-1</revision>
  </version>
</network_filter>
```

Please note that version is referring to the minimal support version for the specific filter.
### Table 6.418. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>network_filter</td>
<td>NetworkFilter</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.138.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.139. NETWORKFILTERS

Represents a readonly network filters sub-collection.

The network filter enables to filter packets send to/from the VM's nic according to defined rules. For more information please refer to NetworkFilter service documentation

Network filters are supported in different versions, starting from version 3.0.

A network filter is defined for each vnic profile.

A vnic profile is defined for a specific network.

A network can be assigned to several different clusters. In the future, each network will be defined in cluster level.

Currently, each network is being defined at data center level. Potential network filters for each network are determined by the network's data center compatibility version V. V must be >= the network filter version in order to configure this network filter for a specific network. Please note, that if a network is assigned to cluster with a version supporting a network filter, the filter may not be available due to the data center version being smaller then the network filter's version.

Example of listing all of the supported network filters for a specific cluster:

```
GET http://localhost:8080/ovirt-engine/api/clusters/{cluster:id}/networkfilters
```

Output:

```
<network_filters>
```
Table 6.419. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Retrieves the representations of the network filters.</td>
</tr>
</tbody>
</table>

6.139.1. list GET

Retrieves the representations of the network filters.

The order of the returned list of network filters isn't guaranteed.

Table 6.420. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filters</td>
<td>NetworkFilter[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.139.1.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.140. NETWORKLABEL

Table 6.421. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes a label from a logical network.</td>
</tr>
</tbody>
</table>

6.140.1. get GET

Table 6.422. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>label</td>
<td>NetworkLabel</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.140.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.140.2. remove DELETE

Removes a label from a logical network.

For example, to remove the label exemplary from a logical network having id 123 send the following request:

```
DELETE /ovirt-engine/api/networks/123/labels/exemplary
```

Table 6.423. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.141. NETWORKLABELS

Manages the set of labels attached to a network or to a host NIC.
Table 6.424. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Attaches label to logical network.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of labels attached to the network or host NIC.</td>
</tr>
</tbody>
</table>

6.141.1. add POST

Attaches label to logical network.

You can attach labels to a logical network to automate the association of that logical network with physical host network interfaces to which the same label has been attached.

For example, to attach the label `mylabel` to a logical network having id `123` send a request like this:

```
POST /ovirt-engine/api/networks/123/labels
```

With a request body like this:

```
<label id="mylabel"/>
```

Table 6.425. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>NetworkLabel</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.141.2. list GET

Returns the list of labels attached to the network or host NIC.

The order of the returned list of labels isn’t guaranteed.

Table 6.426. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>labels</td>
<td>NetworkLabel[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of labels to return.</td>
</tr>
</tbody>
</table>

6.141.2.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.141.2.2. max

Sets the maximum number of labels to return. If not specified all the labels are returned.

6.142. NETWORKS

Manages logical networks.

The engine creates a default ovirtmgmt network on installation. This network acts as the management network for access to hypervisor hosts. This network is associated with the Default cluster and is a member of the Default data center.

Table 6.427. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new logical network, or associates an existing network with a data center.</td>
</tr>
<tr>
<td>list</td>
<td>List logical networks.</td>
</tr>
</tbody>
</table>

6.142.1. add POST

Creates a new logical network, or associates an existing network with a data center.

Creation of a new network requires the name and data_center elements.

For example, to create a network named mynetwork for data center 123 send a request like this:

```
POST /ovirt-engine/api/networks
```

With a request body like this:

```
<network>
  <name>mynetwork</name>
  <data_center id="123"/>
</network>
```

To associate the existing network 456 with the data center 123 send a request like this:

```
POST /ovirt-engine/api/datacenters/123/networks
```

With a request body like this:

```
<network>
  <name>ovirtmgmt</name>
</network>
```
To create a network named `exnetwork` on top of an external OpenStack network provider `456` send a request like this:

```plaintext
POST /ovirt-engine/api/networks

<network>
  <name>exnetwork</name>
  <external_provider id="456"/>
  <data_center id="123"/>
</network>
```

Table 6.428. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.142.2. list GET

List logical networks.

For example:

```plaintext
GET /ovirt-engine/api/networks
```

Will respond:

```xml
<networks>
  <network href="/ovirt-engine/api/networks/123" id="123">
    <name>ovirtmgmt</name>
    <description>Default Management Network</description>
    <link href="/ovirt-engine/api/networks/123/permissions" rel="permissions"/>
    <link href="/ovirt-engine/api/networks/123/vnicprofiles" rel="vnicprofiles"/>
    <link href="/ovirt-engine/api/networks/123/networklabels" rel="networklabels"/>
    <mtu>0</mtu>
    <stp>false</stp>
    <usages>
      <usage>vm</usage>
    </usages>
    <data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
  </network>
  ...
</networks>
```

The order of the returned list of networks is guaranteed only if the `sortby` clause is included in the `search` parameter.

Table 6.429. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>networks</td>
<td>Network[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned networks.</td>
</tr>
</tbody>
</table>

### 6.142.2.1. case_sensitive

Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.

### 6.142.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.142.2.3. max

Sets the maximum number of networks to return. If not specified all the networks are returned.

### 6.143. NICNETWORKFILTERPARAMETER

This service manages a parameter for a network filter.

#### Table 6.430. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves a representation of the network filter parameter.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the filter parameter.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the network filter parameter.</td>
</tr>
</tbody>
</table>

#### 6.143.1. get GET

Retrieves a representation of the network filter parameter.
### Table 6.431. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>parameter</td>
<td>NetworkFilter Parameter</td>
<td>Out</td>
<td>The representation of the network filter parameter.</td>
</tr>
</tbody>
</table>

#### 6.143.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.143.2. remove DELETE

Removes the filter parameter.

For example, to remove the filter parameter with id **123** on NIC **456** of virtual machine **789** send a request like this:

```plaintext
DELETE /ovirt-engine/api/vms/789/nics/456/networkfilterparameters/123
```

#### 6.143.3. update PUT

Updates the network filter parameter.

For example, to update the network filter parameter having with with id **123** on NIC **456** of virtual machine **789** send a request like this:

```plaintext
PUT /ovirt-engine/api/vms/789/nics/456/networkfilterparameters/123
```

With a request body like this:

```xml
<network_filter_parameter>
  <name>updatedName</name>
  <value>updatedValue</value>
</network_filter_parameter>
```

### Table 6.432. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter</td>
<td>NetworkFilter Parameter</td>
<td>In/Out</td>
<td>The network filter parameter that is being updated.</td>
</tr>
</tbody>
</table>

### 6.144. NICNETWORKFILTERPARAMETERS

This service manages a collection of parameters for network filters.
Table 6.433. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a network filter parameter.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieves the representations of the network filter parameters.</td>
</tr>
</tbody>
</table>

6.144.1. add POST

Add a network filter parameter.

For example, to add the parameter for the network filter on NIC 456 of virtual machine 789 send a request like this:

```
POST /ovirt-engine/api/vms/789/nics/456/networkfilterparameters
```

With a request body like this:

```
<network_filter_parameter>
  <name>IP</name>
  <value>10.0.1.2</value>
</network_filter_parameter>
```

Table 6.434. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter</td>
<td>NetworkFilter Parameter</td>
<td>In/Out</td>
<td>The network filter parameter that is being added.</td>
</tr>
</tbody>
</table>

6.144.2. list GET

Retrieves the representations of the network filter parameters.

The order of the returned list of network filters isn't guaranteed.

Table 6.435. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>parameter</td>
<td>NetworkFilter Parameter[]</td>
<td>Out</td>
<td>The list of the network filter parameters.</td>
</tr>
</tbody>
</table>

6.144.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.145. OPENSTACKIMAGE

Table 6.436. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>Imports a virtual machine from a Glance image storage domain.</td>
</tr>
</tbody>
</table>

6.145.1. get GET

Table 6.437. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>image</td>
<td>OpenStackImage</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.145.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.145.2. import POST

Imports a virtual machine from a Glance image storage domain.

For example, to import the image with identifier 456 from the storage domain with identifier 123 send a request like this:

```
POST /ovirt-engine/api/openstackimageproviders/123/images/456/import
```

With a request body like this:

```
<action>
  <storage_domain>
    <name>images0</name>
  </storage_domain>
  <cluster>
    <name>images0</name>
  </cluster>
</action>
```

Table 6.438. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the import should be performed asynchronously.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td>This parameter is mandatory in case of using import_as_template and indicates which cluster should be used for import glance image as template.</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>import_as_template</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether the image should be imported as a template.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.146. OPENSTACKIMAGEPROVIDER

Table 6.439. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>importcertificates</td>
<td>Import the SSL certificates of the external host provider.</td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>testconnectivity</td>
<td>In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.</td>
</tr>
<tr>
<td>update</td>
<td>Update the specified OpenStack image provider in the system.</td>
</tr>
</tbody>
</table>

### 6.146.1. get GET

Table 6.440. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.146.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.146.2. importcertificates POST

Import the SSL certificates of the external host provider.

Table 6.441. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.146.3. remove DELETE

Table 6.442. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.146.4. testconnectivity POST

In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

Table 6.443. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the test should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.146.5. update PUT
Update the specified OpenStack image provider in the system.

Table 6.444. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>provider</td>
<td>OpenStackImageProvider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.147. OPENSTACKIMAGEPROVIDERS

Table 6.445. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new OpenStack image provider to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of providers.</td>
</tr>
</tbody>
</table>

6.147.1. add POST

Adds a new OpenStack image provider to the system.

Table 6.446. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider</td>
<td>OpenStackImageProvider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.147.2. list GET

Returns the list of providers.

The order of the returned list of providers is not guaranteed.

Table 6.447. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of providers to return.</td>
</tr>
</tbody>
</table>
### 6.147.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.147.2.2. max

Sets the maximum number of providers to return. If not specified, all the providers are returned.

### 6.148. OPENSTACKIMAGES

**Table 6.448. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Lists the images of a Glance image storage domain.</td>
</tr>
</tbody>
</table>

**Table 6.449. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>images</td>
<td>OpenStackImage[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of images to return.</td>
</tr>
</tbody>
</table>

#### 6.148.1. list GET

Lists the images of a Glance image storage domain.

The order of the returned list of images isn't guaranteed.

#### 6.148.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.
6.148.2. max

Sets the maximum number of images to return. If not specified all the images are returned.

6.149. OPENSTACKNETWORK

Table 6.450. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>This operation imports an external network into Red Hat Virtualization.</td>
</tr>
</tbody>
</table>

6.149.1. get GET

Table 6.451. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>network</td>
<td>OpenStackNetwork</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.149.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.149.2. import POST

This operation imports an external network into Red Hat Virtualization. The network will be added to the specified data center.

Table 6.452. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the import should be performed asynchronously.</td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>In</td>
<td>The data center into which the network is to be imported.</td>
</tr>
</tbody>
</table>

6.149.2.1. data_center

The data center into which the network is to be imported. Data center is mandatory, and can be specified using the id or name attributes. The rest of the attributes will be ignored.
NOTE

If auto_sync is enabled for the provider, the network might be imported automatically. To prevent this, automatic import can be disabled by setting the auto_sync to false, and enabling it again after importing the network.

6.150. OPENSTACKNETWORKPROVIDER

This service manages the OpenStack network provider.

Table 6.453. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the representation of the object managed by this service.</td>
</tr>
<tr>
<td>importcertificates</td>
<td>Import the SSL certificates of the external host provider.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the provider.</td>
</tr>
<tr>
<td>testconnectivity</td>
<td>In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the provider.</td>
</tr>
</tbody>
</table>

6.150.1. get GET

Returns the representation of the object managed by this service.

For example, to get the OpenStack network provider with identifier 1234, send a request like this:

```
GET /ovirt-engine/api/openstacknetworkproviders/1234
```

Table 6.454. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>provider</td>
<td>OpenStackNetworkProvider</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.150.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.150.2. importcertificates POST
Import the SSL certificates of the external host provider.

### Table 6.455. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.150.3. remove DELETE

Removes the provider.

For example, to remove the OpenStack network provider with identifier **1234**, send a request like this:

```
DELETE /ovirt-engine/api/openstacknetworkproviders/1234
```

### Table 6.456. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.150.4. testconnectivity POST

In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.

```
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

### Table 6.457. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the test should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.150.5. update PUT

Updates the provider.

For example, to update **provider_name**, **requires_authentication**, **url**, **tenant_name** and **type** properties, for the OpenStack network provider with identifier **1234**, send a request like this:

```
PUT /ovirt-engine/api/openstacknetworkproviders/1234
```

With a request body like this:

```
<openstack_network_provider>
```
Table 6.458. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>provider</td>
<td>OpenStackNetworkProvider</td>
<td>In/Out</td>
<td>The provider to update.</td>
</tr>
</tbody>
</table>

6.151. OPENSTACKNETWORKPROVIDERS

This service manages OpenStack network providers.

Table 6.459. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new network provider to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of providers.</td>
</tr>
</tbody>
</table>

6.151.1. add POST

Adds a new network provider to the system. If the type property is not present, a default value of NEUTRON will be used.

Table 6.460. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>provider</td>
<td>OpenStackNetworkProvider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.151.2. list GET

Returns the list of providers.

The order of the returned list of providers is not guaranteed.

Table 6.461. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of providers to return.</td>
</tr>
<tr>
<td>providers</td>
<td>OpenStackNetworkProvider[]</td>
<td>Out</td>
<td>A query string used to restrict the returned OpenStack network providers.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned OpenStack network providers.</td>
</tr>
</tbody>
</table>

### 6.151.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.151.2.2. max

Sets the maximum number of providers to return. If not specified, all the providers are returned.

### 6.152. OPENSTACKNETWORKS

#### Table 6.462. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of networks.</td>
</tr>
</tbody>
</table>

#### 6.152.1. list GET

Returns the list of networks.

The order of the returned list of networks isn’t guaranteed.

#### Table 6.463. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>networks</td>
<td>OpenStackNetworkProvider[]</td>
<td>Out</td>
<td>A query string used to restrict the returned OpenStack network providers.</td>
</tr>
</tbody>
</table>

### 6.152.1.1. follow
Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.152.1.2. max

Sets the maximum number of networks to return. If not specified all the networks are returned.

### 6.153. OPENSTACKSUBNET

**Table 6.464. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.153.1. get GET

**Table 6.465. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>subnet</td>
<td>OpenStackSubnet</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.153.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.153.2. remove DELETE

**Table 6.466. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.154. OPENSTACKSUBNETS

**Table 6.467. Methods summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of sub-networks.</td>
</tr>
</tbody>
</table>

### 6.154.1. add POST
Table 6.468. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>subnet</td>
<td>OpenStackSubnet</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.154.2. list GET
Returns the list of sub-networks.

The order of the returned list of sub-networks isn’t guaranteed.

Table 6.469. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of sub-networks to return.</td>
</tr>
<tr>
<td>subnets</td>
<td>OpenStackSubnet[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.154.2.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.154.2.2. max
Sets the maximum number of sub-networks to return. If not specified all the sub-networks are returned.

### 6.155. OPENSTACKVOLUMEAUTHENTICATIONKEY
Table 6.470. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update the specified authentication key.</td>
</tr>
</tbody>
</table>

### 6.155.1. get GET

#### Table 6.471. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>key</td>
<td>OpenstackVolumeAuthenticationTokenKey</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.155.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.155.2. remove DELETE

#### Table 6.472. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.155.3. update PUT

Update the specified authentication key.

#### Table 6.473. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>OpenstackVolumeAuthenticationTokenKey</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.156. OPENSTACKVOLUMEAUTHENTICATIONKEYS

#### Table 6.474. Methods summary
### 6.156.1. add POST

Add a new authentication key to the OpenStack volume provider.

#### Table 6.475. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>OpenstackVolumeAuthenticationKey</td>
<td>In/Out</td>
<td>Add a new authentication key to the OpenStack volume provider.</td>
</tr>
</tbody>
</table>

### 6.156.2. list GET

Returns the list of authentication keys.

The order of the returned list of authentication keys isn’t guaranteed.

#### Table 6.476. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>keys</td>
<td>OpenstackVolumeAuthenticationKey[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of keys to return.</td>
</tr>
</tbody>
</table>

#### 6.156.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.156.2.2. max

Sets the maximum number of keys to return. If not specified all the keys are returned.

### 6.157. OPENSTACKVOLUMEPROVIDER

Table 6.477. Methods summary
### Table 6.478. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>provider</td>
<td>OpenStackVolumeProvider</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.157.1. get GET

Table 6.478. Parameters summary

- **follow**: Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.157.2. importcertificates POST

Import the SSL certificates of the external host provider.

Table 6.479. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.157.3. remove DELETE

Table 6.480. Parameters summary
### 6.157.3.1. force

Indicates if the operation should succeed, and the provider removed from the database, even if something fails during the operation.

This parameter is optional, and the default value is `false`.

### 6.157.4. testconnectivity POST

In order to test connectivity for external provider we need to run following request where 123 is an id of a provider.

```plaintext
POST /ovirt-engine/api/externalhostproviders/123/testconnectivity
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the test should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.157.5. update PUT

Update the specified OpenStack volume provider in the system.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>provider</td>
<td>OpenStackVolumeProvider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.158. OPENSTACKVOLUMEPROVIDERS

Table 6.483. Methods summary
Name | Summary
---|---
**add** | Adds a new volume provider.
**list** | Retrieves the list of volume providers.

### 6.158.1. add POST

Adds a new volume provider.

For example:

```bash
POST /ovirt-engine/api/openstackvolumeproviders
```

With a request body like this:

```xml
<openstack_volume_provider>
    <name>mycinder</name>
    <url>https://mycinder.example.com:8776</url>
    <data_center>
        <name>mydc</name>
    </data_center>
    <requires_authentication>true</requires_authentication>
    <username>admin</username>
    <password>mypassword</password>
    <tenant_name>mytenant</tenant_name>
</openstack_volume_provider>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>provider</strong></td>
<td>OpenStackVolumeProvider</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.158.2. list GET

Retrieves the list of volume providers.

The order of the returned list of volume providers is not guaranteed.

**Table 6.485. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>follow</strong></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td><strong>max</strong></td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of providers to return.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>providers</td>
<td>OpenStackVolumeProvider[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned volume providers.</td>
</tr>
</tbody>
</table>

### 6.158.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.158.2.2. max

Sets the maximum number of providers to return. If not specified, all the providers are returned.

### 6.159. OPENSTACKVOLUMETYPE

Table 6.486. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.159.1. get GET

Table 6.487. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>type</td>
<td>OpenStackVolumeType</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.159.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.160. OPENSTACKVOLUMETYPES

Table 6.488. Methods summary
## 6.160. list GET

Returns the list of volume types.

The order of the returned list of volume types isn’t guaranteed.

### Table 6.489. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of volume types to return.</td>
</tr>
<tr>
<td>types</td>
<td>OpenStackVolumeType[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.160.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.160.1.2. max

Sets the maximum number of volume types to return. If not specified all the volume types are returned.

## 6.161. OPERATINGSYSTEM

### Table 6.490. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

### 6.161.1. get GET

### Table 6.491. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.161.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.162. OPERATINGSYSTEMS

Manages the set of types of operating systems available in the system.

#### Table 6.492. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of types of operating system available in the system.</td>
</tr>
</tbody>
</table>

#### 6.162.1. list GET

Returns the list of types of operating system available in the system.

The order of the returned list of operating systems isn’t guaranteed.

#### Table 6.493. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>operating_system</td>
<td>OperatingSystemInfo[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.162.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.162.1.2. max

Sets the maximum number of networks to return. If not specified all the networks are returned.

### 6.163. PERMISSION
### 6.163. get GET

#### Table 6.495. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>permission</td>
<td>Permission</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.163.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.163.2. remove DELETE

#### Table 6.496. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.164. PERMIT

A service to manage a specific permit of the role.

#### Table 6.497. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the information about the permit of the role.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the permit from the role.</td>
</tr>
</tbody>
</table>

#### 6.164.1. get GET

Gets the information about the permit of the role.

For example to retrieve the information about the permit with the id 456 of the role with the id 123 send
a request like this:

```bash
GET /ovirt-engine/api/roles/123/permits/456
```

```xml
<permit href="/ovirt-engine/api/roles/123/permits/456" id="456">
  <name>change_vm_cd</name>
  <administrative>false</administrative>
  <role href="/ovirt-engine/api/roles/123" id="123"/>
</permit>
```

Table 6.498. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>permit</td>
<td>Permit</td>
<td>Out</td>
<td>The permit of the role.</td>
</tr>
</tbody>
</table>

6.164.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.164.2. remove DELETE

Removes the permit from the role.

For example to remove the permit with id 456 from the role with id 123 send a request like this:

```bash
DELETE /ovirt-engine/api/roles/123/permits/456
```

Table 6.499. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.165. PERMITS

Represents a permits sub-collection of the specific role.

Table 6.500. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a permit to the role.</td>
</tr>
<tr>
<td>list</td>
<td>List the permits of the role.</td>
</tr>
</tbody>
</table>
6.165.1. add POST

Adds a permit to the role. The permit name can be retrieved from the Section 6.39, "ClusterLevels" service.

For example to assign a permit create_vm to the role with id 123 send a request like this:

```
POST /ovirt-engine/api/roles/123/permits
```

With a request body like this:

```
<permit>
  <name>create_vm</name>
</permit>
```

Table 6.501. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permit</td>
<td>Permit</td>
<td>In/Out</td>
<td>The permit to add.</td>
</tr>
</tbody>
</table>

6.165.2. list GET

List the permits of the role.

For example to list the permits of the role with the id 123 send a request like this:

```
GET /ovirt-engine/api/roles/123/permits
```

```
<permits>
  <permit href="/ovirt-engine/api/roles/123/permits/5" id="5">
    <name>change_vm_cd</name>
    <administrative>false</administrative>
    <role href="/ovirt-engine/api/roles/123" id="123"/>
  </permit>
  <permit href="/ovirt-engine/api/roles/123/permits/7" id="7">
    <name>connect_to_vm</name>
    <administrative>false</administrative>
    <role href="/ovirt-engine/api/roles/123" id="123"/>
  </permit>
</permits>
```

The order of the returned list of permits isn’t guaranteed.

Table 6.502. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of permits to return.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>permits</td>
<td>Permit[]</td>
<td>Out</td>
<td>List of permits.</td>
</tr>
</tbody>
</table>

### 6.165.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.165.2.2. max

Sets the maximum number of permits to return. If not specified all the permits are returned.

### 6.166. QOS

#### Table 6.503. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get specified QoS in the data center.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove specified QoS from datacenter.</td>
</tr>
<tr>
<td>update</td>
<td>Update the specified QoS in the dataCenter.</td>
</tr>
</tbody>
</table>

#### 6.166.1. get GET

Get specified QoS in the data center.

```
GET /ovirt-engine/api/datacenters/123/qoss/123
```

You will get response like this one below:

```xml
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>123</name>
  <description>123</description>
  <max_iops>1</max_iops>
  <max_throughput>1</max_throughput>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

#### Table 6.504. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>
6.166.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.166.2. remove DELETE

Remove specified QoS from datacenter.

DELETE /ovirt-engine/api/datacenters/123/qoss/123

Table 6.505. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.166.3. update PUT

Update the specified QoS in the dataCenter.

PUT /ovirt-engine/api/datacenters/123/qoss/123

For example with curl:

```
curl -u admin@internal:123456 -X PUT -H "content-type: application/xml" -d "
  <qos>
    <name>321</name>
    <description>321</description>
    <max_iops>10</max_iops>
  </qos>
" https://engine/ovirt-engine/api/datacenters/123/qoss/123
```

You will receive response like this:

```
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>321</name>
  <description>321</description>
  <max_iops>10</max_iops>
  <max_throughput>1</max_throughput>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

Table 6.506. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
### 6.167. QOSS

Manages the set of quality of service configurations available in a data center.

#### Table 6.507. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new QoS to the dataCenter.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of quality of service configurations available in the data center.</td>
</tr>
</tbody>
</table>

#### 6.167.1. add POST

Add a new QoS to the dataCenter.

```
POST /ovirt-engine/api/datacenters/123/qoss
```

The response will look as follows:

```
<qos href="/ovirt-engine/api/datacenters/123/qoss/123" id="123">
  <name>123</name>
  <description>123</description>
  <max_iops>10</max_iops>
  <type>storage</type>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
</qos>
```

#### Table 6.508. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>qos</td>
<td>Qos</td>
<td>In/Out</td>
<td>Added QoS object.</td>
</tr>
</tbody>
</table>

#### 6.167.2. list GET

Returns the list of quality of service configurations available in the data center.

```
GET /ovirt-engine/api/datacenter/123/qoss
```

You will get response which will look like this:

```
- 
```
The returned list of quality of service configurations isn't guaranteed.

Table 6.509. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of QoS descriptors to return.</td>
</tr>
<tr>
<td>qoss</td>
<td>Qos[]</td>
<td>Out</td>
<td>List of queried QoS objects.</td>
</tr>
</tbody>
</table>

6.167.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.167.2.2. max

Sets the maximum number of QoS descriptors to return. If not specified all the descriptors are returned.

6.168. QUOTA

Table 6.510. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves a quota.</td>
</tr>
<tr>
<td>remove</td>
<td>Delete a quota.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a quota.</td>
</tr>
</tbody>
</table>

6.168.1. get GET

Retrieves a quota.

An example of retrieving a quota:

```
GET /ovirt-engine/api/datacenters/123/quotas/456
```

```
<quota id="456">
  <name>myquota</name>
</quota>
```
Table 6.511. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.168.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.168.2. remove DELETE

Delete a quota.

An example of deleting a quota:

```
DELETE /ovirt-engine/api/datacenters/123-456/quotas/654-321-0472718ab224 HTTP/1.1
Accept: application/xml
Content-type: application/xml
```

Table 6.512. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.168.3. update PUT

Updates a quota.

An example of updating a quota:

```
PUT /ovirt-engine/api/datacenters/123/quotas/456

<quota>
  <cluster_hard_limit_pct>30</cluster_hard_limit_pct>
  <cluster_soft_limit_pct>70</cluster_soft_limit_pct>
</quota>
```
Table 6.513. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.169. QUOTA_CLUSTERS

Table 6.514. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.169.1. get GET

Table 6.515. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>limit</td>
<td>QuotaCluster Limit</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.169.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.169.2. remove DELETE

Table 6.516. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.170. QUOTACLUSTERLIMITS

Manages the set of quota limits configured for a cluster.

Table 6.517. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a cluster limit to a specified Quota.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the set of quota limits configured for the cluster.</td>
</tr>
</tbody>
</table>

6.170.1. add POST

Add a cluster limit to a specified Quota.

Table 6.518. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td>QuotaClusterLimit</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.170.2. list GET

Returns the set of quota limits configured for the cluster.

The returned list of quota limits isn’t guaranteed.

Table 6.519. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>limits</td>
<td>QuotaClusterLimit[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of limits to return.</td>
</tr>
</tbody>
</table>

6.170.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

6.170.2.2. max

Sets the maximum number of limits to return. If not specified all the limits are returned.
6.171. QUOTASTORAGELIMIT

Table 6.520. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.171.1. get GET

Table 6.521. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>limit</td>
<td>QuotaStorageLimit</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.171.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.171.2. remove DELETE

Table 6.522. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.172. QUOTASTORAGELIMITS

Manages the set of storage limits configured for a quota.

Table 6.523. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a storage limit to a specified quota.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of storage limits configured for the quota.</td>
</tr>
</tbody>
</table>

6.172.1. add POST
Adds a storage limit to a specified quota.

To create a 100GiB storage limit for all storage domains in a data center, send a request like this:

```plaintext
POST /ovirt-engine/api/datacenters/123/quotas/456/quotastoragelimits
```

With a request body like this:

```xml
<quota_storage_limit>
  <limit>100</limit>
</quota_storage_limit>
```

To create a 50GiB storage limit for a storage domain with the ID 000, send a request like this:

```plaintext
POST /ovirt-engine/api/datacenters/123/quotas/456/quotastoragelimits
```

With a request body like this:

```xml
<quota_storage_limit>
  <limit>50</limit>
  <storage_domain id="000"/>
</quota_storage_limit>
```

**Table 6.524. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td>QuotaStorageLimit</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.172.2. list GET**

Returns the list of storage limits configured for the quota.

The order of the returned list of storage limits is not guaranteed.

**Table 6.525. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>limits</td>
<td>QuotaStorageLimit[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of limits to return.</td>
</tr>
</tbody>
</table>

**6.172.2.1. follow**

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.172.2.2. max
Sets the maximum number of limits to return. If not specified, all the limits are returned.

6.173. QUOTAS
Manages the set of quotas configured for a data center.

Table 6.526. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new quota.</td>
</tr>
<tr>
<td>list</td>
<td>Lists quotas of a data center.</td>
</tr>
</tbody>
</table>

6.173.1. add POST
Creates a new quota.

An example of creating a new quota:

```
POST /ovirt-engine/api/datacenters/123/quotas

<quota>
  <name>myquota</name>
  <description>My new quota for virtual machines</description>
</quota>
```

Table 6.527. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>quota</td>
<td>Quota</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.173.2. list GET
Lists quotas of a data center.

The order of the returned list of quotas isn’t guaranteed.

Table 6.528. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of quota descriptors to return.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>quotas</td>
<td>Quota[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.173.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.173.2.2. max

Sets the maximum number of quota descriptors to return. If not specified all the descriptors are returned.

### 6.174. ROLE

**Table 6.529. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get the role.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the role.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a role.</td>
</tr>
</tbody>
</table>

#### 6.174.1. get GET

Get the role.

```
GET /ovirt-engine/api/roles/123
```

You will receive XML response like this one:

```
<role id="123">
  <name>MyRole</name>
  <description>MyRole description</description>
  <link href="/ovirt-engine/api/roles/123/permits" rel="permits"/>
  <administrative>true</administrative>
  <mutable>false</mutable>
</role>
```

**Table 6.530. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>role</td>
<td>Role</td>
<td>Out</td>
<td>Retrieved role.</td>
</tr>
</tbody>
</table>
6.174.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.174.2. remove DELETE

Removes the role.

To remove the role you need to know its id, then send request like this:

```
DELETE /ovirt-engine/api/roles/{role_id}
```

Table 6.531. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.174.3. update PUT

Updates a role. You are allowed to update name, description and administrative attributes after role is created. Within this endpoint you can't add or remove roles permits you need to use service that manages permits of role.

For example to update role's name, description and administrative attributes send a request like this:

```
PUT /ovirt-engine/api/roles/123
```

With a request body like this:

```
<role>
  <name>MyNewRoleName</name>
  <description>My new description of the role</description>
  <administrative>true</administrative>
</role>
```

Table 6.532. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>role</td>
<td>Role</td>
<td>In/Out</td>
<td>Updated role.</td>
</tr>
</tbody>
</table>

6.175. ROLES

Provides read-only access to the global set of roles
Table 6.533. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Create a new role.</td>
</tr>
<tr>
<td>list</td>
<td>List roles.</td>
</tr>
</tbody>
</table>

### 6.175.1. add POST

Create a new role. The role can be administrative or non-administrative and can have different permits.

For example, to add the **MyRole** non-administrative role with permits to login and create virtual machines send a request like this (note that you have to pass permit id):

```
POST /ovirt-engine/api/roles
```

With a request body like this:

```
<role>
    <name>MyRole</name>
    <description>My custom role to create virtual machines</description>
    <administrative>false</administrative>
    <permits>
        <permit id="1"/>
        <permit id="1300"/>
    </permits>
</role>
```

Table 6.534. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>Role</td>
<td>In/Out</td>
<td>Role that will be added.</td>
</tr>
</tbody>
</table>

### 6.175.2. list GET

List roles.

```
GET /ovirt-engine/api/roles
```

You will receive response in XML like this one:

```
<roles>
    <role id="123">
       <name>SuperUser</name>
       <description>Roles management administrator</description>
       <link href="/ovirt-engine/api/roles/123/permits" rel="permits"/>
       <administrative>true</administrative>
       <mutable>false</mutable>
    </role>
</roles>
```
The order of the returned list of roles isn’t guaranteed.

### Table 6.535. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of roles to return.</td>
</tr>
<tr>
<td>roles</td>
<td>Role[]</td>
<td>Out</td>
<td>Retrieved list of roles.</td>
</tr>
</tbody>
</table>

#### 6.175.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.175.2.2. max

Sets the maximum number of roles to return. If not specified all the roles are returned.

### 6.176. SCHEDULINGPOLICIES

Manages the set of scheduling policies available in the system.

#### Table 6.536. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new scheduling policy to the system.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of scheduling policies available in the system.</td>
</tr>
</tbody>
</table>

#### 6.176.1. add POST

Add a new scheduling policy to the system.

#### Table 6.537. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>SchedulingPolicy</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.176.2. list GET
Returns the list of scheduling policies available in the system.

The order of the returned list of scheduling policies isn’t guaranteed.

**Table 6.538. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of policies to return.</td>
</tr>
<tr>
<td>policies</td>
<td>SchedulingPolicy[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.176.2.1. follow**

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.176.2.2. max**

Sets the maximum number of policies to return. If not specified all the policies are returned.

**6.177. SCHEDULINGPOLICY**

**Table 6.539. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update the specified user defined scheduling policy in the system.</td>
</tr>
</tbody>
</table>

**6.177.1. get GET**

**Table 6.540. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>policy</td>
<td>SchedulingPolicy</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.177.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.177.2. remove DELETE

**Table 6.541. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.177.3. update PUT

Update the specified user defined scheduling policy in the system.

**Table 6.542. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>policy</td>
<td>SchedulingPolicy</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.178. SCHEDULINGPOLICYUNIT

**Table 6.543. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

### 6.178.1. get GET

**Table 6.544. Parameters summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>unit</td>
<td>SchedulingPolicyUnit</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.178.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.178.2. remove DELETE

Table 6.545. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.179. SCHEDULINGPOLICYUNITS

Manages the set of scheduling policy units available in the system.

Table 6.546. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of scheduling policy units available in the system.</td>
</tr>
</tbody>
</table>

6.179.1. list GET

Returns the list of scheduling policy units available in the system.

The order of the returned list of scheduling policy units isn’t guaranteed.

Table 6.547. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of policy units to return.</td>
</tr>
<tr>
<td>units</td>
<td>SchedulingPolicyUnit[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.179.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

**6.179.1.2. max**

Sets the maximum number of policy units to return. If not specified all the policy units are returned.

---

**6.180. SNAPSHOT**

**Table 6.548. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>restore</td>
<td>Restores a virtual machine snapshot.</td>
</tr>
</tbody>
</table>

**6.180.1. get GET**

**Table 6.549. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

**6.180.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

**6.180.2. remove DELETE**

**Table 6.550. Parameters summary**
6.180.2.1. all_content

Indicates if all the attributes of the virtual machine snapshot should be included in the response.

By default the attribute `initialization.configuration.data` is excluded.

For example, to retrieve the complete representation of the snapshot with id 456 of the virtual machine with id 123 send a request like this:

```
GET /ovirt-engine/api/vms/123/snapshots/456?all_content=true
```

6.180.3. restore POST

Restores a virtual machine snapshot.

For example, to restore the snapshot with identifier 456 of virtual machine with identifier 123 send a request like this:

```
POST /ovirt-engine/api/vms/123/snapshots/456/restore
```

With an empty `action` in the body:

```
<action/>
```

Table 6.551. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>async</strong></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the restore should be performed asynchronously.</td>
</tr>
<tr>
<td><strong>disks</strong></td>
<td>Disk[]</td>
<td>In</td>
<td>Specify the disks included in the snapshot’s restore.</td>
</tr>
<tr>
<td><strong>restore_memory</strong></td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.180.3.1. disks

Specify the disks included in the snapshot’s restore.

For each disk parameter, it is also required to specify its `image_id`. 
For example, to restore a snapshot with an identifier 456 of a virtual machine with identifier 123, including a disk with identifier 111 and image_id of 222, send a request like this:

```
POST /ovirt-engine/api/vms/123/snapshots/456/restore
```

Request body:

```
<action>
  <disks>
    <disk id="111">
      <image_id>222</image_id>
    </disk>
  </disks>
</action>
```

6.181. SNAPSHOTCDROM

Table 6.552. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

6.181.1. get GET

Table 6.553. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Cdrom</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.181.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.182. SNAPSHOTCDROMS

Manages the set of CD-ROM devices of a virtual machine snapshot.

Table 6.554. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of CD-ROM devices of the snapshot.</td>
</tr>
</tbody>
</table>
6.182.1. list GET

Returns the list of CD-ROM devices of the snapshot.

The order of the returned list of CD-ROM devices isn’t guaranteed.

**Table 6.555. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of CDROMS to return.</td>
</tr>
</tbody>
</table>

6.182.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

6.182.1.2. max

Sets the maximum number of CDROMS to return. If not specified all the CDROMS are returned.

6.183. SNAPSHOTTDISK

**Table 6.556. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

6.183.1. get GET

**Table 6.557. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

6.183.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

6.184. SNAPSHOTTDISKS
Manages the set of disks of an snapshot.

### Table 6.558. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of disks of the snapshot.</td>
</tr>
</tbody>
</table>

### 6.184.1. list GET

Returns the list of disks of the snapshot.

The order of the returned list of disks isn’t guaranteed.

### Table 6.559. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of disks to return.</td>
</tr>
</tbody>
</table>

### 6.184.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.184.1.2. max

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.185. SNAPSHOTNIC

### Table 6.560. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

### 6.185.1. get GET

### Table 6.561. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.185.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.186. SNAPSHOTNICS

Manages the set of NICs of an snapshot.

**Table 6.562. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of NICs of the snapshot.</td>
</tr>
</tbody>
</table>

#### 6.186.1. list GET

Returns the list of NICs of the snapshot.

The order of the returned list of NICs isn’t guaranteed.

**Table 6.563. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of NICs to return.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.186.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.186.1.2. max

Sets the maximum number of NICs to return. If not specified all the NICs are returned.

### 6.187. SNAPSHOTS

Manages the set of snapshots of a storage domain or virtual machine.

**Table 6.564. Methods summary**
### Name | Summary
--- | ---
**add** | Creates a virtual machine snapshot.

**list** | Returns the list of snapshots of the storage domain or virtual machine.

#### 6.187.1. add POST

Creates a virtual machine snapshot.

For example, to create a new snapshot for virtual machine **123** send a request like this:

```
POST /ovirt-engine/api/vms/123/snapshots
```

With a request body like this:

```
<snapshot>
  <description>My snapshot</description>
</snapshot>
```

For including only a sub-set of disks in the snapshots, add **disk_attachments** element to the request body. Note that disks which are not specified in **disk_attachments** element will not be a part of the snapshot. If an empty **disk_attachments** element is passed, the snapshot will include only the virtual machine configuration. If no **disk_attachments** element is passed, then all the disks will be included in the snapshot.

For each disk, **image_id** element can be specified for setting the new active image id. This is used in order to restore a chain of images from backup. I.e. when restoring a disk with snapshots, the relevant **image_id** should be specified for each snapshot (so the identifiers of the disk snapshots are identical to the backup).

```
<snapshot>
  <description>My snapshot</description>
  <disk_attachments>
    <disk_attachment>
      <disk id="123">
        <image_id>456</image_id>
      </disk>
    </disk_attachment>
  </disk_attachments>
</snapshot>
```
IMPORTANT

When a snapshot is created the default value for the `persist_memorystate` attribute is `true`. That means that the content of the memory of the virtual machine will be included in the snapshot, and it also means that the virtual machine will be paused for a longer time. That can negatively affect applications that are very sensitive to timing (NTP servers, for example). In those cases make sure that you set the attribute to `false`:

```
<snapshot>
  <description>My snapshot</description>
  <persist_memorystate>false</persist_memorystate>
</snapshot>
```

Table 6.565. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.187.2. list GET

Returns the list of snapshots of the storage domain or virtual machine.

The order of the returned list of snapshots isn’t guaranteed.

Table 6.566. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_content</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all the attributes of the virtual machine snapshot should be included in the response.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of snapshots to return.</td>
</tr>
<tr>
<td>snapshots</td>
<td>Snapshot[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.187.2.1. all_content

Indicates if all the attributes of the virtual machine snapshot should be included in the response.

By default the attribute `initialization.configuration.data` is excluded.

For example, to retrieve the complete representation of the virtual machine with id 123 snapshots send a request like this:

```
GET /ovirt-engine/api/vms/123/snapshots?all_content=true
```

6.187.2.2. follow
Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.187.2.3. max

Sets the maximum number of snapshots to return. If not specified all the snapshots are returned.

### 6.188. SSHPUBLICKEY

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.188.1. get GET

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>key</td>
<td>SshPublicKey</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.188.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.188.2. remove DELETE

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.188.3. update PUT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>

Red Hat Virtualization 4.2 REST API Guide
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>key</td>
<td>SshPublicKey</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.189. SSHPUBLICKEYS

#### Table 6.571. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Returns a list of SSH public keys of the user.</td>
</tr>
</tbody>
</table>

#### 6.189.1. add POST

#### Table 6.572. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>SshPublicKey</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.189.2. list GET

Returns a list of SSH public keys of the user.

For example, to retrieve the list of SSH keys of user with identifier **123**, send a request like this:

```
GET /ovirt-engine/api/users/123/sshpublickeys
```

The result will be the following XML document:

```xml
<ssh_public_keys>
  <ssh_public_key href="/ovirt-engine/api/users/123/sshpublickeys/456" id="456">
    <content>ssh-rsa ...
    </content>
  </ssh_public_key>
</ssh_public_keys>
```

Or the following JSON object

```json
{
  "ssh_public_key": [
```
The order of the returned list of keys is not guaranteed.

### Table 6.573. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>keys</td>
<td>SshPublicKey[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of keys to return.</td>
</tr>
</tbody>
</table>

#### 6.189.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.189.2.2. max

Sets the maximum number of keys to return. If not specified all the keys are returned.

### 6.190. STATISTIC

#### Table 6.574. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.190.1. get GET

#### Table 6.575. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.190.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.191. STATISTICS

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistic</td>
<td>Statistic</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.576. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Retrieves a list of statistics.</td>
</tr>
</tbody>
</table>

6.191.1. list GET

Retrieves a list of statistics.

For example, to retrieve the statistics for virtual machine 123 send a request like this:

```
GET /ovirt-engine/api/vms/123/statistics
```

The result will be like this:

```
<statistics>
  <statistic href="/ovirt-engine/api/vms/123/statistics/456" id="456">
    <name>memory.installed</name>
    <description>Total memory configured</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>1073741824</datum>
      </value>
    </values>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </statistic>
  ...
</statistics>
```

Just a single part of the statistics can be retrieved by specifying its id at the end of the URI. That means:

```
GET /ovirt-engine/api/vms/123/statistics/456
```

Outputs:
The order of the returned list of statistics isn’t guaranteed.

Table 6.577. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <strong>followed</strong>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of statistics to return.</td>
</tr>
<tr>
<td>statistic</td>
<td>Statistic[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.191.1.1. **follow**

Indicates which inner links should be **followed**. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

6.191.1.2. **max**

Sets the maximum number of statistics to return. If not specified all the statistics are returned.

6.192. **STEP**

A service to manage a step.

Table 6.578. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>end</td>
<td>Marks an external step execution as ended.</td>
</tr>
<tr>
<td>get</td>
<td>Retrieves a step.</td>
</tr>
</tbody>
</table>

6.192.1. **end** POST

Marks an external step execution as ended.
For example, to terminate a step with identifier 456 which belongs to a job with identifier 123 send the following request:

```
POST /ovirt-engine/api/jobs/123/steps/456/end
```

With the following request body:

```
<action>
  <force>true</force>
  <succeeded>true</succeeded>
</action>
```

### Table 6.579. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the step should be forcibly terminated.</td>
</tr>
<tr>
<td>succeeded</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the step should be marked as successfully finished or as failed.</td>
</tr>
</tbody>
</table>

#### 6.192.1. succeeded

Indicates if the step should be marked as successfully finished or as failed.

This parameter is optional, and the default value is `true`.

#### 6.192.2. get GET

Retrieves a step.

```
GET /ovirt-engine/api/jobs/123/steps/456
```

You will receive response in XML like this one:

```
<step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
  </actions>
  <description>Validating</description>
  <end_time>2016-12-12T23:07:26.627+02:00</end_time>
  <external>false</external>
  <number>0</number>
  <start_time>2016-12-12T23:07:26.605+02:00</start_time>
  <status>finished</status>
  <type>validating</type>
  <job href="/ovirt-engine/api/jobs/123" id="123"/>
</step>
```
Table 6.580. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>step</td>
<td>Step</td>
<td>Out</td>
<td>Retrieves the representation of the step.</td>
</tr>
</tbody>
</table>

6.192.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.193. STEPS

A service to manage steps.

Table 6.581. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add an external step to an existing job or to an existing step.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieves the representation of the steps.</td>
</tr>
</tbody>
</table>

6.193.1. add POST

Add an external step to an existing job or to an existing step.

For example, to add a step to job with identifier 123 send the following request:

```
POST /ovirt-engine/api/jobs/123/steps
```

With the following request body:

```
<step>
  <description>Validating</description>
  <start_time>2016-12-12T23:07:26.605+02:00</start_time>
  <status>started</status>
  <type>validating</type>
</step>
```

The response should look like:

```
<step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
  <actions>
    <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
  </actions>
  <description>Validating</description>
  <link href="/ovirt-engine/api/jobs/123/steps/456/statistics" rel="statistics"/>
</step>
```
Table 6.582. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>step</td>
<td>Step</td>
<td>In/Out</td>
<td>Step that will be added.</td>
</tr>
</tbody>
</table>

6.193.2. list GET

Retrieves the representation of the steps.

GET /ovirt-engine/api/job/123/steps

You will receive response in XML like this one:

```xml
<steps>
  <step href="/ovirt-engine/api/jobs/123/steps/456" id="456">
    <actions>
      <link href="/ovirt-engine/api/jobs/123/steps/456/end" rel="end"/>
    </actions>
    <description>Validating</description>
    <link href="/ovirt-engine/api/jobs/123/steps/456/statistics" rel="statistics"/>
    <external>true</external>
    <number>2</number>
    <start_time>2016-12-13T01:06:15.380+02:00</start_time>
    <status>started</status>
    <type>validating</type>
    <job href="/ovirt-engine/api/jobs/123" id="123"/>
  </step>
...
</steps>
```

The order of the returned list of steps isn’t guaranteed.

Table 6.583. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of steps to return.</td>
</tr>
<tr>
<td>steps</td>
<td>Step[]</td>
<td>Out</td>
<td>A representation of steps.</td>
</tr>
</tbody>
</table>
6.193.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.193.2.2. max

Sets the maximum number of steps to return. If not specified all the steps are returned.

6.194. STORAGE

Table 6.584. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

6.194.1. get GET

Table 6.585. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>report_status</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the status of the LUNs in the storage should be checked.</td>
</tr>
<tr>
<td>storage</td>
<td>HostStorage</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.194.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.194.1.2. report_status

Indicates if the status of the LUNs in the storage should be checked. Checking the status of the LUN is an heavy weight operation and this data is not always needed by the user. This parameter will give the option to not perform the status check of the LUNs.

The default is true for backward compatibility.

Here an example with the LUN status :

```xml
<host_storage id="360014051136c20574f743b0d28177f">  
  <logical_units>
  
    <logical_unit id="360014051136c20574f743b0d28177f">  
      <lun_mapping>0</lun_mapping>
      <paths>1</paths>
      <product_id>lun0</product_id>
    
```
6.195. STORAGEDOMAIN

Table 6.586. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the description of the storage domain.</td>
</tr>
<tr>
<td>isattached</td>
<td>Used for querying if the storage domain is already attached to a data center using the is_attached boolean field, which is part of the storage server.</td>
</tr>
<tr>
<td>reduceluns</td>
<td>This operation reduces logical units from the storage domain.</td>
</tr>
<tr>
<td>refreshluns</td>
<td>This operation refreshes the LUN size.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the storage domain.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a storage domain.</td>
</tr>
<tr>
<td>updateovfstore</td>
<td>This operation forces the update of the <strong>OVF_STORE</strong> of this storage domain.</td>
</tr>
</tbody>
</table>
### 6.195.1. get GET

Retrieves the description of the storage domain.

#### Table 6.587. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Out</td>
<td>The description of the storage domain.</td>
</tr>
</tbody>
</table>

#### 6.195.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.195.2. isattached POST

Used for querying if the storage domain is already attached to a data center using the is_attached boolean field, which is part of the storage server. IMPORTANT: Executing this API will cause the host to disconnect from the storage domain.

#### Table 6.588. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>In</td>
<td>Indicates the data center’s host.</td>
</tr>
<tr>
<td>is_attached</td>
<td>Boolean</td>
<td>Out</td>
<td>Indicates whether the storage domain is attached to the data center.</td>
</tr>
</tbody>
</table>

### 6.195.3. reduceluns POST

This operation reduces logical units from the storage domain.
In order to do so the data stored on the provided logical units will be moved to other logical units of the storage domain and only then they will be reduced from the storage domain.

For example, in order to reduce two logical units from a storage domain send a request like this:

```
POST /ovirt-engine/api/storageDomains/123/reduceluns
```

With a request body like this:

```
<action>
  <logical_units>
    <logical_unit id="1IET_00010001"/>
    <logical_unit id="1IET_00010002"/>
  </logical_units>
</action>
```

Note that this operation is only applicable to block storage domains (i.e., storage domains with the types/storage_type, storage type> of iSCSI or FCP).

**Table 6.589. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>logical_units</td>
<td>LogicalUnit[]</td>
<td>In</td>
<td>The logical units that need to be reduced from the storage domain.</td>
</tr>
</tbody>
</table>

**6.195.4. refreshluns POST**

This operation refreshes the LUN size.

After increasing the size of the underlying LUN on the storage server, the user can refresh the LUN size. This action forces a rescan of the provided LUNs and updates the database with the new size, if required.

For example, in order to refresh the size of two LUNs send a request like this:

```
POST /ovirt-engine/api/storageDomains/262b056b-aede-40f1-9666-b883eff59d40/refreshluns
```

With a request body like this:

```
<action>
  <logical_units>
    <logical_unit id="1IET_00010001"/>
    <logical_unit id="1IET_00010002"/>
  </logical_units>
</action>
```

**Table 6.590. Parameters summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the refresh should be performed asynchronously.</td>
</tr>
<tr>
<td>logical_units</td>
<td>LogicalUnit[]</td>
<td>In</td>
<td>The LUNs that need to be refreshed.</td>
</tr>
</tbody>
</table>

### 6.195.5. remove DELETE

Removes the storage domain.

Without any special parameters, the storage domain is detached from the system and removed from the database. The storage domain can then be imported to the same or to a different setup, with all the data on it. If the storage is not accessible the operation will fail.

If the `destroy` parameter is `true` then the operation will always succeed, even if the storage is not accessible, the failure is just ignored and the storage domain is removed from the database anyway.

If the `format` parameter is `true` then the actual storage is formatted, and the metadata is removed from the LUN or directory, so it can no longer be imported to the same or to a different setup.

Table 6.591. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
<tr>
<td>destroy</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the operation should succeed, and the storage domain removed from the database, even if the storage is not accessible.</td>
</tr>
<tr>
<td>format</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the actual storage should be formatted, removing all the metadata from the underlying LUN or directory:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[source] ---- DELETE /ovirt-engine/api/storageDomains/123?format=true ----</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This parameter is optional, and the default value is <code>false</code>.</td>
</tr>
<tr>
<td>host</td>
<td>String</td>
<td>In</td>
<td>Indicates which host should be used to remove the storage domain.</td>
</tr>
</tbody>
</table>

### 6.195.5.1. destroy

Indicates if the operation should succeed, and the storage domain removed from the database, even if the storage is not accessible.
DELETE /ovirt-engine/api/storageDomains/123?destroy=true

This parameter is optional, and the default value is \textit{false}. When the value of \textit{destroy} is \textit{true} the \textit{host} parameter will be ignored.

\subsection*{6.195.5.2. host}

Indicates which host should be used to remove the storage domain.

This parameter is mandatory, except if the \textit{destroy} parameter is included and its value is \textit{true}, in that case the \textit{host} parameter will be ignored.

The value should contain the name or the identifier of the host. For example, to use the host named \textit{myhost} to remove the storage domain with identifier \textit{123} send a request like this:

DELETE /ovirt-engine/api/storageDomains/123?host=myhost

\subsection*{6.195.6. update PUT}

Updates a storage domain.

Not all of the \textit{StorageDomain}'s attributes are updatable after creation. Those that can be updated are: \textit{name}, \textit{description}, \textit{comment}, \textit{warning_low_space_indicator}, \textit{critical_space_action_blocker} and \textit{wipe_after_delete}. (Note that changing the \textit{wipe_after_delete} attribute will not change the wipe after delete property of disks that already exist).

To update the \textit{name} and \textit{wipe_after_delete} attributes of a storage domain with an identifier \textit{123}, send a request as follows:

PUT /ovirt-engine/api/storageDomains/123

With a request body as follows:

\begin{verbatim}
<storage_domain>
  <name>data2</name>
  <wipe_after_delete>true</wipe_after_delete>
</storage_domain>
\end{verbatim}

\begin{table}[h]
\centering
\begin{tabular}{|l|l|l|l|}
\hline
\textbf{Name} & \textbf{Type} & \textbf{Direction} & \textbf{Summary} \\
\hline
async & Boolean & In & Indicates if the update should be performed asynchronously. \\
\hline
storage_domain & StorageDomain & In/Out & The updated storage domain. \\
\hline
\end{tabular}
\caption{Parameters summary}
\end{table}

\subsection*{6.195.7. updateovfstore POST}

This operation forces the update of the \textit{OVF_STORE} of this storage domain.
The **OVF_STORE** is a disk image that contains the metadata of virtual machines and disks that reside in the storage domain. This metadata is used in case the domain is imported or exported to or from a different data center or a different installation.

By default the **OVF_STORE** is updated periodically (set by default to 60 minutes) but users might want to force an update after an important change, or when they believe the **OVF_STORE** is corrupt.

When initiated by the user, **OVF_STORE** update will be performed whether an update is needed or not.

### Table 6.593. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the <strong>OVF_STORE</strong> update should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.196. STORAGEDOMAINCONTENTDISK

### Table 6.594. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.196.1. get GET

### Table 6.595. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.196.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.197. STORAGEDOMAINCONTENTDISKS

Manages the set of disks available in a storage domain.

### Table 6.596. Methods summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td></td>
<td></td>
<td>Returns the list of disks available in the storage domain.</td>
</tr>
</tbody>
</table>

### 6.197.1. list GET

Returns the list of disks available in the storage domain.

The order of the returned list of disks is guaranteed only if the \texttt{sortby} clause is included in the \texttt{search} parameter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the \texttt{search} parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of disks to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned disks.</td>
</tr>
</tbody>
</table>

### 6.197.1.1. case_sensitive

Indicates if the search performed using the \texttt{search} parameter should be performed taking case into account. The default value is \texttt{true}, which means that case is taken into account. If you want to search ignoring case set it to \texttt{false}.

### 6.197.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.197.1.3. max

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.198. STORAGEDOMAINDISK

Manages a single disk available in a storage domain.
IMPORTANT
Since version 4.2 of the engine this service is intended only to list disks available in the storage domain, and to register unregistered disks. All the other operations, like copying a disk, moving a disk, etc, have been deprecated and will be removed in the future. To perform those operations use the service that manages all the disks of the system, or the service that manages an specific disk.

Table 6.598. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>Copies a disk to the specified storage domain.</td>
</tr>
<tr>
<td>export</td>
<td>Exports a disk to an export storage domain.</td>
</tr>
<tr>
<td>get</td>
<td>Retrieves the description of the disk.</td>
</tr>
<tr>
<td>move</td>
<td>Moves a disk to another storage domain.</td>
</tr>
<tr>
<td>reduce</td>
<td>Reduces the size of the disk image.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a disk.</td>
</tr>
<tr>
<td>sparsify</td>
<td>Sparsify the disk.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the disk.</td>
</tr>
</tbody>
</table>

6.198.1. copy POST
Copies a disk to the specified storage domain.

IMPORTANT
Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To copy a disk use the copy operation of the service that manages that disk.

Table 6.599. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In</td>
<td>Description of the resulting disk.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td>The storage domain where the new disk will be created.</td>
</tr>
</tbody>
</table>

6.198.2. export POST
Exports a disk to an export storage domain.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To export a disk use the export operation of the service that manages that disk.

**Table 6.600. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td>The export storage domain where the disk should be exported to.</td>
</tr>
</tbody>
</table>

**6.198.3. get GET**

Retrieves the description of the disk.

**Table 6.601. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td>The description of the disk.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

**6.198.3.1. follow**

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.198.4. move POST**

Moves a disk to another storage domain.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To move a disk use the move operation of the service that manages that disk.

**Table 6.602. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the move should be performed asynchronously.</td>
</tr>
</tbody>
</table>
### 6.198.5. reduce POST

Reduces the size of the disk image.

Invokes reduce on the logical volume (i.e. this is only applicable for block storage domains). This is applicable for floating disks and disks attached to non-running virtual machines. There is no need to specify the size as the optimal size is calculated automatically.

**Table 6.603. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.198.6. remove DELETE

Removes a disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To remove a disk use the remove operation of the service that manages that disk.

### 6.198.7. sparsify POST

Sparsify the disk.

**IMPORTANT**

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To remove a disk use the remove operation of the service that manages that disk.

### 6.198.8. update PUT

Updates the disk.
IMPORTANT

Since version 4.2 of the engine this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To update a disk use the update operation of the service that manages that disk.

Table 6.604. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The update to apply to the disk.</td>
</tr>
</tbody>
</table>

6.199. STORAGEDOMAINDISKS

Manages the collection of disks available inside a specific storage domain.

Table 6.605. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds or registers a disk.</td>
</tr>
<tr>
<td>list</td>
<td>Retrieves the list of disks that are available in the storage domain.</td>
</tr>
</tbody>
</table>

6.199.1. add POST

Adds or registers a disk.

IMPORTANT

Since version 4.2 of the Red Hat Virtualization Manager this operation is deprecated, and preserved only for backwards compatibility. It will be removed in the future. To add a new disk use the add operation of the service that manages the disks of the system. To register an unregistered disk use the register operation of the service that manages that disk.

Table 6.606. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td>The disk to add or register.</td>
</tr>
<tr>
<td>unregistered</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if a new disk should be added or if an existing unregistered disk should be registered.</td>
</tr>
</tbody>
</table>

6.199.1.1. unregistered
Indicates if a new disk should be added or if an existing unregistered disk should be registered. If the value is `true` then the identifier of the disk to register needs to be provided. For example, to register the disk with ID `456` send a request like this:

```markdown
POST /ovirt-engine/api/storagedomains/123/disks?unregistered=true
```

With a request body like this:

```xml
<d\id="456"/>
```

If the value is `false` then a new disk will be created in the storage domain. In that case the `provisioned_size`, `format`, and `name` attributes are mandatory. For example, to create a new `copy on write` disk of 1 GiB, send a request like this:

```markdown
POST /ovirt-engine/api/storagedomains/123/disks
```

With a request body like this:

```xml
<d
  <name>mydisk</name>
  <format>cow</format>
  <provisioned_size>1073741824</provisioned_size>
</d>
```

The default value is `false`.

This parameter has been deprecated since version 4.2 of the Red Hat Virtualization Manager.

### 6.199.2. list GET

Retrieves the list of disks that are available in the storage domain.

The order of the returned list of disks is not guaranteed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>Out</td>
<td>The list of retrieved disks.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of disks to return.</td>
</tr>
<tr>
<td>unregistered</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether to retrieve a list of registered or unregistered disks in the storage domain.</td>
</tr>
</tbody>
</table>

#### 6.199.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.
6.199.2.2. max

Sets the maximum number of disks to return. If not specified, all the disks are returned.

6.199.2.3. unregistered

Indicates whether to retrieve a list of registered or unregistered disks in the storage domain. To get a list of unregistered disks in the storage domain the call should indicate the unregistered flag. For example, to get a list of unregistered disks the REST API call should look like this:

```
GET /ovirt-engine/api/storagedomains/123/disks?unregistered=true
```

The default value of the unregistered flag is `false`. The request only applies to storage domains that are attached.

6.200. STORAGEDOMAINSERVERCONNECTION

Table 6.608. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Detaches a storage connection from storage.</td>
</tr>
</tbody>
</table>

6.200.1. get GET

Table 6.609. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection</td>
<td>StorageConnection</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.200.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.200.2. remove DELETE

Detaches a storage connection from storage.

Table 6.610. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
async | Boolean | In | Indicates if the action should be performed asynchronously.

6.201. STORAGEDOMAINSERVERCONNECTIONS

Manages the set of connections to storage servers that exist in a storage domain.

Table 6.611. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of connections to storage servers that exist in the storage domain.</td>
</tr>
</tbody>
</table>

6.201.1. add POST

Table 6.612. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection</td>
<td>StorageConnection</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.201.2. list GET

Returns the list of connections to storage servers that exist in the storage domain.

The order of the returned list of connections isn’t guaranteed.

Table 6.613. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>connections</td>
<td>StorageConnection[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of connections to return.</td>
</tr>
</tbody>
</table>

6.201.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.201.2.2. max
Sets the maximum number of connections to return. If not specified all the connections are returned.

6.202. STORAGEDOMAINTEMPLATE

Table 6.614. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>Action to import a template from an export storage domain.</td>
</tr>
<tr>
<td>register</td>
<td>Register the Template means importing the Template from the data domain by inserting the configuration of the Template and disks into the database without the copy process.</td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.202.1. get GET

Table 6.615. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.202.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.202.2. import POST

Action to import a template from an export storage domain.

For example, to import the template 456 from the storage domain 123 send the following request:

```
POST /ovirt-engine/api/storagedomains/123/templates/456/import
```

With the following request body:

```
<action>
   <storage_domain>
      <name>myexport</name>
   </storage_domain>
   <cluster>
```
If you register an entity without specifying the cluster ID or name, the cluster name from the entity’s OVF will be used (unless the register request also includes the cluster mapping).

Table 6.616. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the import should be performed asynchronously.</td>
</tr>
<tr>
<td>clone</td>
<td>Boolean</td>
<td>In</td>
<td>Use the optional clone parameter to generate new UUIDs for the imported template and its entities.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>exclusive</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

6.202.2.1. clone

Use the optional clone parameter to generate new UUIDs for the imported template and its entities.

You can import a template with the clone parameter set to false when importing a template from an export domain, with templates that were exported by a different Red Hat Virtualization environment.

6.202.3. register POST

Register the Template means importing the Template from the data domain by inserting the configuration of the Template and disks into the database without the copy process.

Table 6.617. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_partial_import</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether a template is allowed to be registered with only some of its disks.</td>
</tr>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the registration should be performed asynchronously.</td>
</tr>
</tbody>
</table>
### 6.202.3.1. allow_partial_import

Indicates whether a template is allowed to be registered with only some of its disks.

If this flag is `true`, the system will not fail in the validation process if an image is not found, but instead it will allow the template to be registered without the missing disks. This is mainly used during registration of a template when some of the storage domains are not available. The default value is `false`.

### 6.202.3.2. registration_configuration

This parameter describes how the template should be registered.

This parameter is optional. If the parameter is not specified, the template will be registered with the same configuration that it had in the original environment where it was created.

### 6.202.3.3. vnic_profile_mappings

Deprecated attribute describing mapping rules for virtual NIC profiles that will be applied during the import/register process.

---

**WARNING**

Please note that this attribute has been deprecated since version 4.2.1 of the engine, and preserved only for backward compatibility. It will be removed in the future. To specify `vnic_profile_mappings` use the `vnic_profile_mappings` attribute inside the `RegistrationConfiguration` type.

---

### 6.202.4. remove DELETE

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>exclusive</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>registration_configuration</td>
<td>RegistrationConfiguration</td>
<td>In</td>
<td>This parameter describes how the template should be registered.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>vnic_profile_mappings</td>
<td>VnicProfileMapping[]</td>
<td>In</td>
<td>Deprecated attribute describing mapping rules for virtual NIC profiles that will be applied during the import/register process.</td>
</tr>
</tbody>
</table>
Table 6.618. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.203. STORAGEDOMAINTEMPLATES

Manages the set of templates available in a storage domain.

Table 6.619. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of templates available in the storage domain.</td>
</tr>
</tbody>
</table>

6.203.1. list GET

Returns the list of templates available in the storage domain.

The order of the returned list of templates isn't guaranteed.

Table 6.620. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of templates to return.</td>
</tr>
<tr>
<td>templates</td>
<td>Template[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>unregistered</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether to retrieve a list of registered or unregistered templates which contain disks on the storage domain.</td>
</tr>
</tbody>
</table>

6.203.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.203.1.2. max

Sets the maximum number of templates to return. If not specified all the templates are returned.

6.203.1.3. unregistered
Indicates whether to retrieve a list of registered or unregistered templates which contain disks on the storage domain. To get a list of unregistered templates the call should indicate the unregistered flag. For example, to get a list of unregistered templates the REST API call should look like this:

GET /ovirt-engine/api/storagedomains/123/templates?unregistered=true

The default value of the unregistered flag is false. The request only apply to storage domains that are attached.

6.204. STORAGEDOMAINVM

Table 6.621. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>import</td>
<td>Imports a virtual machine from an export storage domain.</td>
</tr>
<tr>
<td>register</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Deletes a virtual machine from an export storage domain.</td>
</tr>
</tbody>
</table>

6.204.1. get GET

Table 6.622. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.204.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.204.2. import POST

Imports a virtual machine from an export storage domain.

For example, send a request like this:

POST /ovirt-engine/api/storagedomains/123/vms/456/import

With a request body like this:

<action>
To import a virtual machine as a new entity add the **clone** parameter:

```xml
<storage_domain>
  <name>mydata</name>
</storage_domain>
<cluster>
  <name>mycluster</name>
</cluster>
<clone>true</clone>
<vm>
  <name>myvm</name>
</vm>
</action>
```

Include an optional **disks** parameter to choose which disks to import. For example, to import the disks of the template that have the identifiers **123** and **456** send the following request body:

```xml
<action>
  <storage_domain>
    <name>mydata</name>
  </storage_domain>
  <cluster>
    <name>mycluster</name>
  </cluster>
  <clone>true</clone>
  <vm>
    <name>myvm</name>
  </vm>
  <disks>
    <disk id="123"/>
    <disk id="456"/>
  </disks>
</action>
```

If you register an entity without specifying the cluster ID or name, the cluster name from the entity’s OVF will be used (unless the register request also includes the cluster mapping).

### Table 6.623. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the import should be performed asynchronously.</td>
</tr>
<tr>
<td>clone</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the identifiers of the imported virtual machine should be regenerated.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>
6.204.2.1. clone

Indicates if the identifiers of the imported virtual machine should be regenerated.

By default when a virtual machine is imported the identifiers are preserved. This means that the same virtual machine can’t be imported multiple times, as that identifiers needs to be unique. To allow importing the same machine multiple times set this parameter to true, as the default is false.

6.204.2.2. collapse_snapshots

Indicates of the snapshots of the virtual machine that is imported should be collapsed, so that the result will be a virtual machine without snapshots.

This parameter is optional, and if it isn’t explicitly specified the default value is false.

6.204.3. register POST

Table 6.624. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_partial_import</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether a virtual machine is allowed to be registered with only some of its disks.</td>
</tr>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the registration should be performed asynchronously.</td>
</tr>
<tr>
<td>clone</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>reassign_bad_macs</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the problematic MAC addresses should be re-assigned during the import process by the engine.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>registration_configuration</td>
<td>Registration Configuration</td>
<td>In</td>
<td>This parameter describes how the virtual machine should be registered.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In</td>
<td>Deprecated attribute describing mapping rules for virtual NIC profiles that will be applied during the import/register process.</td>
</tr>
<tr>
<td>vnic_profile_mappings</td>
<td>VnicProfileMapping[]</td>
<td>In</td>
<td>Deprecated attribute describing mapping rules for virtual NIC profiles that will be applied during the import/register process.</td>
</tr>
</tbody>
</table>

6.204.3.1. allow_partial_import

Indicates whether a virtual machine is allowed to be registered with only some of its disks.

If this flag is true, the engine will not fail in the validation process if an image is not found, but instead it will allow the virtual machine to be registered without the missing disks. This is mainly used during registration of a virtual machine when some of the storage domains are not available. The default value is false.

6.204.3.2. reassign_bad_macs

Indicates if the problematic MAC addresses should be re-assigned during the import process by the engine.

A MAC address would be considered as a problematic one if one of the following is true:

- It conflicts with a MAC address that is already allocated to a virtual machine in the target environment.
- It's out of the range of the target MAC address pool.

6.204.3.3. registration_configuration

This parameter describes how the virtual machine should be registered.

This parameter is optional. If the parameter is not specified, the virtual machine will be registered with the same configuration that it had in the original environment where it was created.

6.204.3.4. vnic_profile_mappings

Deprecated attribute describing mapping rules for virtual NIC profiles that will be applied during the import/register process.
6.204.4. remove DELETE

Deletes a virtual machine from an export storage domain.

For example, to delete the virtual machine 456 from the storage domain 123, send a request like this:

```
DELETE /ovirt-engine/api/storagedomains/123/vms/456
```

Table 6.625. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.205. STORAGEDOMAINVMDISKATTACHMENT

Returns the details of the disks attached to a virtual machine in the export domain.

Table 6.626. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the details of the attachment with all its properties and a link to the disk.</td>
</tr>
</tbody>
</table>

6.205.1. get GET

Returns the details of the attachment with all its properties and a link to the disk.

Table 6.627. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachmen t</td>
<td>DiskAttachment</td>
<td>Out</td>
<td>The disk attachment.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.205.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.206. STORAGEDOMAINVMDISKATTACHMENTS

Returns the details of a disk attached to a virtual machine in the export domain.

**Table 6.628. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>List the disks that are attached to the virtual machine.</td>
</tr>
</tbody>
</table>

**6.206.1. list GET**

List the disks that are attached to the virtual machine.

The order of the returned list of disk attachments isn’t guaranteed.

**Table 6.629. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachmen ts</td>
<td>DiskAttachm ent[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

**6.206.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.207. STORAGEDOMAINVMS

Lists the virtual machines of an export storage domain.

For example, to retrieve the virtual machines that are available in the storage domain with identifier 123 send the following request:

```
GET /ovirt-engine/api/storagedomains/123/vms
```

This will return the following response body:

```
<vms>
  <vm id="456" href="/api/storagedomains/123/vms/456">
    <name>vm1</name>
    ...
    <storage_domain id="123" href="/api/storagedomains/123"/>
  </vm>
</vms>
```
Virtual machines and templates in these collections have a similar representation to their counterparts in the top-level Vm and Template collections, except they also contain a StorageDomain reference and an import action.

### Table 6.630. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of virtual machines of the export storage domain.</td>
</tr>
</tbody>
</table>

#### 6.207.1. list GET

Returns the list of virtual machines of the export storage domain.

The order of the returned list of virtual machines isn’t guaranteed.

### Table 6.631. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of virtual machines to return.</td>
</tr>
<tr>
<td>unregistered</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates whether to retrieve a list of registered or unregistered virtual machines which contain disks on the storage domain.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.207.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.207.1.2. max

Sets the maximum number of virtual machines to return. If not specified all the virtual machines are returned.

#### 6.207.1.3. unregistered

Indicates whether to retrieve a list of registered or unregistered virtual machines which contain disks on the storage domain. To get a list of unregistered virtual machines the call should indicate the
unregistered flag. For example, to get a list of unregistered virtual machines the REST API call should look like this:

```
GET /ovirt-engine/api/storagedomains/123/vms?unregistered=true
```

The default value of the unregisterd flag is `false`. The request only apply to storage domains that are attached.

## 6.208. STORAGEDOMAINS

Manages the set of storage domains in the system.

### Table 6.632. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a new storage domain.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of storage domains in the system.</td>
</tr>
</tbody>
</table>

### 6.208.1. add POST

Adds a new storage domain.

Creation of a new `StorageDomain` requires the `name`, `type`, `host`, and `storage` attributes. Identify the `host` attribute with the `id` or `name` attributes. In Red Hat Virtualization 3.6 and later you can enable the wipe after delete option by default on the storage domain. To configure this, specify `wipe_after_delete` in the POST request. This option can be edited after the domain is created, but doing so will not change the wipe after delete property of disks that already exist.

To add a new storage domain with specified `name`, `type`, `storage.type`, `storage.address`, and `storage.path`, and using a host with an id `123`, send a request like this:

```
POST /ovirt-engine/api/storageDomains
```

With a request body like this:

```
<storage_domain>
  <name>mydata</name>
  <type>data</type>
  <storage>
    <type>nfs</type>
    <address>mynfs.example.com</address>
    <path>/exports/mydata</path>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
```

To create a new NFS ISO storage domain send a request like this:
To create a new iSCSI storage domain send a request like this:

```xml
<storage_domain>
  <name>myiscsi</name>
  <type>data</type>
  <storage>
    <type>iscsi</type>
    <logical_units>
      <logical_unit id="3600144f09dbd050000004eedbd340001"/>
      <logical_unit id="3600144f09dbd050000004eedbd340002"/>
    </logical_units>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
```

To create a new iSCSI storage domain send a request like this:

```xml
<storage_domain>
  <name>myiscsi</name>
  <type>data</type>
  <storage>
    <type>iscsi</type>
    <logical_units>
      <logical_unit id="3600144f09dbd050000004eedbd340001"/>
      <logical_unit id="3600144f09dbd050000004eedbd340002"/>
    </logical_units>
  </storage>
  <host>
    <name>myhost</name>
  </host>
</storage_domain>
```

**Table 6.633. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In/Out</td>
<td>The storage domain to add.</td>
</tr>
</tbody>
</table>

**6.208.2. list GET**

Returns the list of storage domains in the system.

The order of the returned list of storage domains is guaranteed only if the `sortby` clause is included in the `search` parameter.

**Table 6.634. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search should be performed taking case into account.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of storage domains to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned storage domains.</td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td>Out</td>
<td>A list of the storage domains in the system.</td>
</tr>
</tbody>
</table>

6.208.2.1. case_sensitive

Indicates if the search should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case, set it to false.

6.208.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.208.2.3. max

Sets the maximum number of storage domains to return. If not specified, all the storage domains are returned.

6.209. STORAGESERVERCONNECTION

Table 6.635. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes a storage connection.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the storage connection.</td>
</tr>
</tbody>
</table>

6.209.1. get GET

Table 6.636. Parameters summary
### 6.209.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.209.2. remove DELETE

Removes a storage connection.

A storage connection can only be deleted if neither storage domain nor LUN disks reference it. The host name or id is optional; providing it disconnects (unmounts) the connection from that host.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
<tr>
<td>host</td>
<td>String</td>
<td>In</td>
<td>The name or identifier of the host from which the connection would be unmounted (disconnected).</td>
</tr>
</tbody>
</table>

#### 6.209.2.1. host

The name or identifier of the host from which the connection would be unmounted (disconnected). If not provided, no host will be disconnected.

For example, to use the host with identifier 456 to delete the storage connection with identifier 123 send a request like this:

```
DELETE /ovirt-engine/api/storageconnections/123?host=456
```

#### 6.209.3. update PUT

Updates the storage connection.

For example, to change the address of an NFS storage server, send a request like this:

```
PUT /ovirt-engine/api/storageconnections/123
```

With a request body like this:
To change the connection of an iSCSI storage server, send a request like this:

```
PUT /ovirt-engine/api/storageconnections/123
```

With a request body like this:

```
<storage_connection>
  <address>mynewnfs.example.com</address>
</storage_connection>
```

Table 6.638. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>connection</td>
<td>StorageConnection</td>
<td>In/Out</td>
<td></td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the operation should succeed regardless to the relevant storage domain’s status (i.e. updating is also applicable when storage domain’s status is not maintenance).</td>
</tr>
</tbody>
</table>

6.209.3.1. force

Indicates if the operation should succeed regardless to the relevant storage domain’s status (i.e. updating is also applicable when storage domain’s status is not maintenance).

This parameter is optional, and the default value is **false**.

6.210. STORAGESERVERCONNECTIONEXTENSION

Table 6.639. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update a storage server connection extension for the given host.</td>
</tr>
</tbody>
</table>

6.210.1. get GET
### Table 6.640. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>extension</td>
<td>StorageConnectionExtensi on</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

**6.210.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.210.2. remove DELETE**

**Table 6.641. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

**6.210.3. update PUT**

Update a storage server connection extension for the given host.

To update the storage connection 456 of host 123 send a request like this:

```
PUT /ovirt-engine/api/hosts/123/storageconnectionextensions/456
```

With a request body like this:

```
<storage_connection_extension>
  <target>iqn.2016-01.com.example:mytarget</target>
  <username>myuser</username>
  <password>mypassword</password>
</storage_connection_extension>
```

**Table 6.642. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>extension</td>
<td>StorageConnectionExtensi on</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>
### 6.211. STORAGESERVERCONNECTIONEXTENSIONS

#### Table 6.643. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong></td>
<td>Creates a new storage server connection extension for the given host.</td>
</tr>
<tr>
<td><strong>list</strong></td>
<td>Returns the list of storage connection extensions.</td>
</tr>
</tbody>
</table>

#### 6.211.1. add POST

**Summary:**

Creates a new storage server connection extension for the given host.

The extension lets the user define credentials for an iSCSI target for a specific host. For example, to use `myuser` and `mypassword` as the credentials when connecting to the iSCSI target from host `123`, send a request like this:

```
POST /ovirt-engine/api/hosts/123/storageconnectionextensions
```

With a request body like this:

```
<storage_connection_extension>
  <target>iqn.2016-01.com.example:mytarget</target>
  <username>myuser</username>
  <password>mypassword</password>
</storage_connection_extension>
```

#### Table 6.644. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>extension</strong></td>
<td>StorageConnectionExtension</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.211.2. list GET

**Summary:**

Returns the list of storage connection extensions.

The order of the returned list of storage connections isn’t guaranteed.

#### Table 6.645. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>extensions</strong></td>
<td>StorageConnectionExtension[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
### 6.211.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.211.2.2. max

Sets the maximum number of extensions to return. If not specified all the extensions are returned.

### 6.212. STORAGESERVERCONNECTIONS

#### Table 6.646. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new storage connection.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of storage connections.</td>
</tr>
</tbody>
</table>

#### 6.212.1. add POST

Creates a new storage connection.

For example, to create a new storage connection for the NFS server `mynfs.example.com` and NFS share `/export/mydata` send a request like this:

```
POST /ovirt-engine/api/storageconnections
```

With a request body like this:

```
<storage_connection>
  <type>nfs</type>
  <address>mynfs.example.com</address>
  <path>/export/mydata</path>
  <host>
    <name>myhost</name>
  </host>
</storage_connection>
```

#### Table 6.647. Parameters summary
### Name | Type | Direction | Summary
--- | --- | --- | ---
connection | StorageConnection | In/Out |

#### 6.212.2. list GET

Returns the list of storage connections.

The order of the returned list of connections isn't guaranteed.

**Table 6.648. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>connections</td>
<td>StorageConnection[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of connections to return.</td>
</tr>
</tbody>
</table>

#### 6.212.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.212.2.2. max

Sets the maximum number of connections to return. If not specified all the connections are returned.

### 6.213. SYSTEM

**Table 6.649. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns basic information describing the API, like the product name, the version number and a summary of the number of relevant objects.</td>
</tr>
<tr>
<td>reloadconfigurations</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.213.1. get GET

Returns basic information describing the API, like the product name, the version number and a summary of the number of relevant objects.
GET /ovirt-engine/api

We get following response:

```xml
<api>
  <link rel="capabilities" href="/api/capabilities"/>
  <link rel="clusters" href="/api/clusters"/>
  <link rel="clusters/search" href="/api/clusters?search={query}"/>
  <link rel="datacenters" href="/api/datacenters"/>
  <link rel="datacenters/search" href="/api/datacenters?search={query}"/>
  <link rel="events" href="/api/events"/>
  <link rel="events/search" href="/api/events?search={query}"/>
  <link rel="hosts" href="/api/hosts"/>
  <link rel="hosts/search" href="/api/hosts?search={query}"/>
  <link rel="networks" href="/api/networks"/>
  <link rel="roles" href="/api/roles"/>
  <link rel="storagedomains" href="/api/storagedomains"/>
  <link rel="storagedomains/search" href="/api/storagedomains?search={query}"/>
  <link rel="tags" href="/api/tags"/>
  <link rel="templates" href="/api/templates"/>
  <link rel="templates/search" href="/api/templates?search={query}"/>
  <link rel="users" href="/api/users"/>
  <link rel="groups" href="/api/groups"/>
  <link rel="domains" href="/api/domains"/>
  <link rel="vmpools" href="/api/vmpools"/>
  <link rel="vmpools/search" href="/api/vmpools?search={query}"/>
  <link rel="vms" href="/api/vms"/>
  <link rel="vms/search" href="/api/vms?search={query}"/>
</api>

<product_info>
  <name>oVirt Engine</name>
  <vendor>ovirt.org</vendor>
  <version>
    <build>4</build>
    <full_version>4.0.4</full_version>
    <major>4</major>
    <minor>0</minor>
    <revision>0</revision>
  </version>
</product_info>

<special_objects>
  <blank_template href="/ovirt-engine/api/templates/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000"/>
  <root_tag href="/ovirt-engine/api/tags/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000"/>
</special_objects>

<summary>
  <hosts>
    <active>0</active>
    <total>0</total>
  </hosts>
  <storage_domains>
    <active>0</active>
    <total>1</total>
  </storage_domains>
  <users>
```
The entry point provides a user with links to the collections in a virtualization environment. The `rel` attribute of each collection link provides a reference point for each link.

The entry point also contains other data such as `product_info`, `special_objects` and `summary`.

**Table 6.650. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>api</td>
<td>Api</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

**6.213.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

**6.213.2. reloadconfigurations POST**

**Table 6.651. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the reload should be performed asynchronously.</td>
</tr>
</tbody>
</table>

**6.214. SYSTEMOPTION**

A service that provides values of specific configuration option of the system.

**Table 6.652. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Get the values of specific configuration option.</td>
</tr>
</tbody>
</table>

**6.214.1. get GET**
Get the values of specific configuration option.

For example to retrieve the values of configuration option `MigrationPoliciesSupported` send a request like this:

```
GET /ovirt-engine/api/options/MigrationPoliciesSupported
```

The response to that request will be the following:

```
<system_option href="/ovirt-engine/api/options/MigrationPoliciesSupported" id="MigrationPoliciesSupported">
  <name>MigrationPoliciesSupported</name>
  <values>
    <system_option_value>
      <value>true</value>
      <version>4.0</version>
    </system_option_value>
    <system_option_value>
      <value>true</value>
      <version>4.1</version>
    </system_option_value>
    <system_option_value>
      <value>true</value>
      <version>4.2</version>
    </system_option_value>
    <system_option_value>
      <value>false</value>
      <version>3.6</version>
    </system_option_value>
  </values>
</system_option>
```

**NOTE**

The appropriate permissions are required to query configuration options. Some options can be queried only by users with administrator permissions.

**IMPORTANT**

There is NO backward compatibility and no guarantee about the names or values of the options. Options may be removed and their meaning can be changed at any point.

We strongly discourage the use of this service for applications other than the ones that are released simultaneously with the engine. Usage by other applications is not supported. Therefore there will be no documentation listing accessible configuration options.

**Table 6.653. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
<td>SystemOption</td>
<td>Out</td>
<td>The returned configuration option of the system.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>In</td>
<td>Optional version parameter that specifies that only particular version of the configuration option should be returned.</td>
</tr>
</tbody>
</table>

### 6.214.1. version

Optional version parameter that specifies that only particular version of the configuration option should be returned. If this parameter isn’t used then all the versions will be returned.

For example, to get the value of the MigrationPoliciesSupported option but only for version 4.2 send a request like this:

```bash
GET /ovirt-engine/api/options/MigrationPoliciesSupported?version=4.2
```

The response to that request will be like this:

```xml
<system_option href="/ovirt-engine/api/options/MigrationPoliciesSupported" id="MigrationPoliciesSupported">
  <name>MigrationPoliciesSupported</name>
  <values>
    <system_option_value>
      <value>true</value>
      <version>4.2</version>
    </system_option_value>
  </values>
</system_option>
```

### 6.215. SYSTEMOPTIONS

Service that provides values of configuration options of the system.

### 6.216. SYSTEMPERMISSIONS

This service doesn’t add any new methods, it is just a placeholder for the annotation that specifies the path of the resource that manages the permissions assigned to the system object.

**Table 6.654. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Assign a new permission to a user or group for specific entity.</td>
</tr>
<tr>
<td>list</td>
<td>List all the permissions of the specific entity.</td>
</tr>
</tbody>
</table>

### 6.216.1. add POST
Assign a new permission to a user or group for specific entity.

For example, to assign the **UserVmManager** role to the virtual machine with id **123** to the user with id **456** send a request like this:

```
POST /ovirt-engine/api/vms/123/permissions
```

With a request body like this:

```xml
<permission>
  <role>
    <name>UserVmManager</name>
  </role>
  <user id="456"/>
</permission>
```

To assign the **SuperUser** role to the system to the user with id **456** send a request like this:

```
POST /ovirt-engine/api/permissions
```

With a request body like this:

```xml
<permission>
  <role>
    <name>SuperUser</name>
  </role>
  <user id="456"/>
</permission>
```

If you want to assign permission to the group instead of the user please replace the **user** element with the **group** element with proper id of the group. For example to assign the **UserRole** role to the cluster with id **123** to the group with id **789** send a request like this:

```
POST /ovirt-engine/api/clusters/123/permissions
```

With a request body like this:

```xml
<permission>
  <role>
    <name>UserRole</name>
  </role>
  <group id="789"/>
</permission>
```

### Table 6.655. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permission</td>
<td>Permission</td>
<td>In/Out</td>
<td>The permission.</td>
</tr>
</tbody>
</table>
6.216.2. list GET

List all the permissions of the specific entity.

For example to list all the permissions of the cluster with id 123 send a request like this:

```
GET /ovirt-engine/api/clusters/123/permissions
```

The order of the returned permissions isn’t guaranteed.

### Table 6.656. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Out</td>
<td>The list of permissions.</td>
</tr>
</tbody>
</table>

6.216.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.217. TAG

A service to manage a specific tag in the system.

### Table 6.657. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the information about the tag.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the tag from the system.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the tag entity.</td>
</tr>
</tbody>
</table>
6.217.1. get GET

Gets the information about the tag.

For example to retrieve the information about the tag with the id 123 send a request like this:

GET /ovirt-engine/api/tags/123

Table 6.658. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>tag</td>
<td>Tag</td>
<td>Out</td>
<td>The tag.</td>
</tr>
</tbody>
</table>

6.217.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.217.2. remove DELETE

Removes the tag from the system.

For example to remove the tag with id 123 send a request like this:

DELETE /ovirt-engine/api/tags/123

Table 6.659. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.217.3. update PUT

Updates the tag entity.

For example to update parent tag to tag with id 456 of the tag with id 123 send a request like this:

PUT /ovirt-engine/api/tags/123

With request body like:
You may also specify a tag name instead of id. For example to update parent tag to tag with name `mytag` of the tag with id `123` send a request like this:

```xml
<tag>
    <parent id="456"/>
</tag>
```

Table 6.660. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>tag</td>
<td>Tag</td>
<td>In/Out</td>
<td>The updated tag.</td>
</tr>
</tbody>
</table>

6.218. TAGS

Represents a service to manage collection of the tags in the system.

Table 6.661. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new tag to the system.</td>
</tr>
<tr>
<td>list</td>
<td>List the tags in the system.</td>
</tr>
</tbody>
</table>

6.218.1. add POST

Add a new tag to the system.

For example, to add new tag with name `mytag` to the system send a request like this:

```xml
POST /ovirt-engine/api/tags
```

With a request body like this:

```xml
<tag>
    <name>mytag</name>
</tag>
```
NOTE
The root tag is a special pseudo-tag assumed as the default parent tag if no parent tag is specified. The root tag cannot be deleted nor assigned a parent tag.

To create new tag with specific parent tag send a request body like this:

```xml
<tag>
  <name>mytag</name>
  <parent>
    <name>myparenttag</name>
  </parent>
</tag>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>tag</td>
<td>Tag</td>
<td>In/Out</td>
<td>The added tag</td>
</tr>
</tbody>
</table>

### 6.218.2. list GET

List the tags in the system.

For example to list the full hierarchy of the tags in the system send a request like this:

```
GET /ovirt-engine/api/tags
```

```xml
<tags>
  <tag href="/ovirt-engine/api/tags/222" id="222">
    <name>root2</name>
    <description>root2</description>
    <parent href="/ovirt-engine/api/tags/111" id="111"/>
  </tag>
  <tag href="/ovirt-engine/api/tags/333" id="333">
    <name>root3</name>
    <description>root3</description>
    <parent href="/ovirt-engine/api/tags/222" id="222"/>
  </tag>
  <tag href="/ovirt-engine/api/tags/111" id="111">
    <name>root</name>
    <description>root</description>
  </tag>
</tags>
```

In the previous XML output you can see the following hierarchy of the tags:

root: (id: 111)
  - root2 (id: 222)
  - root3 (id: 333)

The order of the returned list of tags isn't guaranteed.
### Table 6.663. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of tags to return.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>Out</td>
<td>List of all tags in the system.</td>
</tr>
</tbody>
</table>

### 6.218.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.218.2.2. max

Sets the maximum number of tags to return. If not specified all the tags are returned.

### 6.219. TEMPLATE

Manages the virtual machine template and template versions.

### Table 6.664. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>export</td>
<td>Exports a template to the data center export domain.</td>
</tr>
<tr>
<td>get</td>
<td>Returns the information about this template or template version.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a virtual machine template.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the template.</td>
</tr>
</tbody>
</table>

### 6.219.1. export POST

Exports a template to the data center export domain.

For example, send the following request:

```bash
POST /ovirt-engine/api/templates/123/export
```

With a request body like this:

```xml
<action>
  <storage_domain id="456"/>
  <exclusive>true<exclusive/>
</action>
```
Table 6.665. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>exclusive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the existing templates with the same name should be overwritten.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td>Specifies the destination export storage domain.</td>
</tr>
</tbody>
</table>

6.219.1.1. exclusive

Indicates if the existing templates with the same name should be overwritten.

The export action reports a failed action if a template of the same name exists in the destination domain. Set this parameter to true to change this behavior and overwrite any existing template.

6.219.2. get GET

Returns the information about this template or template version.

Table 6.666. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Out</td>
<td>The information about the template or template version.</td>
</tr>
</tbody>
</table>

6.219.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.219.3. remove DELETE

Removes a virtual machine template.

DELETE /ovirt-engine/api/templates/123

Table 6.667. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the removal should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.219.4. update PUT

Updates the template.

The name, description, type, memory, cpu, topology, os, high_availability, display, stateless, usb, and timezone elements can be updated after a template has been created.

For example, to update a template so that it has 1 GiB of memory send a request like this:

```plaintext
PUT /ovirt-engine/api/templates/123
```

With the following request body:

```xml
<template>
  <memory>1073741824</memory>
</template>
```

The version_name name attribute is the only one that can be updated within the version attribute used for template versions:

```xml
<template>
  <version>
    <version_name>mytemplate_2</version_name>
  </version>
</template>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.220. TEMPLATECDROM

A service managing a CD-ROM device on templates.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the information about this CD-ROM device.</td>
</tr>
</tbody>
</table>

6.220.1. get GET

Returns the information about this CD-ROM device.

For example, to get information about the CD-ROM device of template 123 send a request like:
GET /ovirt-engine/api/templates/123/cdroms/

Table 6.670. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Cdrom</td>
<td>Out</td>
<td>The information about the CD-ROM device.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.220.1.1. cdrom

The information about the CD-ROM device.

The information consists of cdrom attribute containing reference to the CD-ROM device, the template, and optionally the inserted disk.

If there is a disk inserted then the file attribute will contain a reference to the ISO image:

```xml
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <template href="/ovirt-engine/api/templates/123" id="123"/>
  <file id="mycd.iso"/>
</cdrom>
```

If there is no disk inserted then the file attribute won’t be reported:

```xml
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <template href="/ovirt-engine/api/templates/123" id="123"/>
</cdrom>
```

6.220.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.221. TEMPLATECDROMS

Lists the CD-ROM devices of a template.

Table 6.671. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of CD-ROM devices of the template.</td>
</tr>
</tbody>
</table>

6.221.1. list GET

Returns the list of CD-ROM devices of the template.

The order of the returned list of CD-ROM devices isn’t guaranteed.
Table 6.672. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>Out</td>
<td>The list of CD-ROM devices of the template.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of CD-ROMs to return.</td>
</tr>
</tbody>
</table>

6.221.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.221.1.2. max

Sets the maximum number of CD-ROMs to return. If not specified all the CD-ROMs are returned.

6.222. TEMPLATEDISK

Table 6.673. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>Copy the specified disk attached to the template to a specific storage domain.</td>
</tr>
<tr>
<td>export</td>
<td></td>
</tr>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.222.1. copy POST

Copy the specified disk attached to the template to a specific storage domain.

Table 6.674. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the copy should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
</tbody>
</table>
### 6.222.2. export POST

Table 6.675. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the export should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.222.3. get GET

Table 6.676. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

### 6.222.3.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.222.4. remove DELETE

Table 6.677. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.223. TEMPLATEDISKATTACHMENT

This service manages the attachment of a disk to a template.
Table 6.678. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the details of the attachment.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the disk from the template.</td>
</tr>
</tbody>
</table>

6.223.1. get GET

Returns the details of the attachment.

Table 6.679. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachment</td>
<td>DiskAttachment</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

6.223.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.223.2. remove DELETE

Removes the disk from the template. The disk will only be removed if there are other existing copies of the disk on other storage domains.

A storage domain has to be specified to determine which of the copies should be removed (template disks can have copies on multiple storage domains).

```
DELETE /ovirt-engine/api/templates/{template:id}/diskattachments/{attachment:id}?storage_domain=072fbaa1-08f3-4a40-9f34-a5ca22dd1d74
```

Table 6.680. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>storage_domain</td>
<td>String</td>
<td>In</td>
<td>Specifies the identifier of the storage domain the image to be removed resides on.</td>
</tr>
</tbody>
</table>
This service manages the set of disks attached to a template. Each attached disk is represented by a DiskAttachment.

Table 6.681. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>List the disks that are attached to the template.</td>
</tr>
</tbody>
</table>

6.224.1. list GET

List the disks that are attached to the template.

The order of the returned list of attachments isn’t guaranteed.

Table 6.682. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>attachments</td>
<td>DiskAttachment[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.224.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.225. TEMPLATEDISKS

Table 6.683. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of disks of the template.</td>
</tr>
</tbody>
</table>

6.225.1. list GET

Returns the list of disks of the template.

The order of the returned list of disks isn’t guaranteed.

Table 6.684. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.226. TEMPLATEGRAPHICSCONSOLE

#### Table 6.685. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets graphics console configuration of the template.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the graphics console from the template.</td>
</tr>
</tbody>
</table>

#### 6.226.1. get GET

Gets graphics console configuration of the template.

#### Table 6.686. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>GraphicsConsole</td>
<td>Out</td>
<td>The information about the graphics console of the template.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.226.2. remove DELETE

Remove the graphics console from the template.

### Table 6.687. Parameters summary
### 6.227. TEMPLATEGRAPHICSCONSOLES

Table 6.688. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add new graphics console to the template.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all the configured graphics consoles of the template.</td>
</tr>
</tbody>
</table>

#### 6.227.1. add POST

Add new graphics console to the template.

Table 6.689. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>GraphicsConsole</td>
<td>In/Out</td>
<td>The list of graphics consoles of the template.</td>
</tr>
</tbody>
</table>

#### 6.227.2. list GET

Lists all the configured graphics consoles of the template.

The order of the returned list of graphics consoles isn’t guaranteed.

Table 6.690. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>consoles</td>
<td>GraphicsConsole[]</td>
<td>Out</td>
<td>The list of graphics consoles of the template.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of consoles to return.</td>
</tr>
</tbody>
</table>

#### 6.227.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.227.2.2. max

Sets the maximum number of consoles to return. If not specified all the consoles are returned.

6.228. TEMPLATENIC

Table 6.691. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
</table>
get | |
remove | |
update | Update the specified network interface card attached to the template. |

6.228.1. get GET

Table 6.692. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
follow | String | In | Indicates which inner links should be followed. |
nic | Nic | Out | |

6.228.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.228.2. remove DELETE

Table 6.693. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
async | Boolean | In | Indicates if the remove should be performed asynchronously. |

6.228.3. update PUT

Update the specified network interface card attached to the template.

Table 6.694. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>nic</td>
<td>Nic</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.229. TEMPLATENICS

#### Table 6.695. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a new network interface card to the template.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of NICs of the template.</td>
</tr>
</tbody>
</table>

#### 6.229.1. add POST

Add a new network interface card to the template.

#### Table 6.696. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>nic</td>
<td>Nic</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.229.2. list GET

Returns the list of NICs of the template.

The order of the returned list of NICs isn't guaranteed.

#### Table 6.697. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of NICs to return.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.229.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.229.2.2. max
Sets the maximum number of NICs to return. If not specified all the NICs are returned.

6.230. TEMPLATEWATCHDOG

Table 6.698. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
<tr>
<td>update</td>
<td>Update the watchdog for the template identified by the given id.</td>
</tr>
</tbody>
</table>

6.230.1. get GET

Table 6.699. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.230.1.1. follow
Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.230.2. remove DELETE

Table 6.700. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.230.3. update PUT
Update the watchdog for the template identified by the given id.

Table 6.701. Parameters summary
### 6.231. TEMPLATEWATCHDOGS

#### Table 6.702. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a watchdog to the template identified by the given id.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of watchdogs.</td>
</tr>
</tbody>
</table>

#### 6.231.1. add POST

Add a watchdog to the template identified by the given id.

#### Table 6.703. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.231.2. list GET

Returns the list of watchdogs.

The order of the returned list of watchdogs isn’t guaranteed.

#### Table 6.704. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of watchdogs to return.</td>
</tr>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.231.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.
6.231.2.2. max

Sets the maximum number of watchdogs to return. If not specified all the watchdogs are returned.

6.232. TEMPLATES

This service manages the virtual machine templates available in the system.

Table 6.705. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new template.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of virtual machine templates.</td>
</tr>
</tbody>
</table>

6.232.1. add POST

Creates a new template.

This requires the name and vm elements. To identify the virtual machine use the vm.id or vm.name attributes. For example, to create a template from a virtual machine with the identifier 123 send a request like this:

```
POST /ovirt-engine/api/templates
```

With a request body like this:

```
<template>
  <name>mytemplate</name>
  <vm id="123"/>
</template>
```

The disks of the template can be customized, making some of their characteristics different from the disks of the original virtual machine. To do so use the vm.disk_attachments attribute, specifying the identifier of the disk of the original virtual machine and the characteristics that you want to change. For example, if the original virtual machine has a disk with the identifier 456, and, for that disk, you want to change the name to mydisk the format to Copy On Write and make it sparse, send a request body like this:

```
<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <name>mydisk</name>
          <format>cow</format>
          <sparse>true</sparse>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>
```
The template can be created as a sub-version of an existing template. This requires the name and vm attributes for the new template, and the base_template and version_name attributes for the new template version. The base_template and version_name attributes must be specified within a version section enclosed in the template section. Identify the virtual machine with the id or name attributes.

```
<template>
  <name>mytemplate</name>
  <vm id="123"/>
  <version>
    <base_template id="456"/>
    <version_name>mytemplate_001</version_name>
  </version>
</template>
```

The destination storage domain of the template can be customized, in one of two ways:

1. Globally, at the request level. The request must list the desired disk attachments to be created on the storage domain. If the disk attachments are not listed, the global storage domain parameter will be ignored.

```
<template>
  <name>mytemplate</name>
  <vm id="456">
    <disk_attachments>
      <disk_attachment>
        <disk id="789">
          <format>cow</format>
          <sparse>true</sparse>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>
```

2. Per each disk attachment. Specify the desired storage domain for each disk attachment. Specifying the global storage definition will override the storage domain per disk attachment specification.

```
<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <format>cow</format>
          <sparse>true</sparse>
          <storage_domains>
            <storage_domain id="789"/>
          </storage_domains>
        </disk>
      </disk_attachment>
    </disk_attachments>
  </vm>
</template>
```
Table 6.706. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone_permissions</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if the permissions of the virtual machine should be copied to the template.</td>
</tr>
<tr>
<td>seal</td>
<td>Boolean</td>
<td>In</td>
<td>Seals the template.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>In/Out</td>
<td>The information about the template or template version.</td>
</tr>
</tbody>
</table>

6.232.1.1. clone_permissions

Specifies if the permissions of the virtual machine should be copied to the template.

If this optional parameter is provided, and its value is `true`, then the permissions of the virtual machine (only the direct ones, not the inherited ones) will be copied to the created template. For example, to create a template from the `myvm` virtual machine copying its permissions, send a request like this:

```plaintext
POST /ovirt-engine/api/templates?clone_permissions=true
```

With a request body like this:

```xml
<template>
  <name>mytemplate</name>
  <vm>
    <name>myvm</name>
  </vm>
</template>
```

6.232.1.2. seal

Seals the template.

If this optional parameter is provided and its value is `true`, then the template is sealed after creation.

Sealing erases all host-specific configuration from the filesystem: SSH keys, UDEV rules, MAC addresses, system ID, hostname, and so on, thus making it easier to use the template to create multiple virtual machines without manual intervention.

Currently, sealing is supported only for Linux operating systems.

6.232.2. list GET

Returns the list of virtual machine templates.

For example:
GET /ovirt-engine/api/templates

Will return the list of virtual machines and virtual machine templates.

The order of the returned list of templates is not guaranteed.

Table 6.707. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <strong>search</strong> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <strong>followed</strong>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of templates to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned templates.</td>
</tr>
<tr>
<td>templates</td>
<td>Template[]</td>
<td>Out</td>
<td>The list of virtual machine templates.</td>
</tr>
</tbody>
</table>

6.232.2.1. case_sensitive

Indicates if the search performed using the **search** parameter should be performed taking case into account. The default value is **true**, which means that case is taken into account. If you want to search ignoring case set it to **false**.

6.232.2.2. follow

Indicates which inner links should be **followed**. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

6.232.2.3. max

Sets the maximum number of templates to return. If not specified, all the templates are returned.

6.233. UNMANAGEDNETWORK

Table 6.708. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>
6.233.1. get GET

Table 6.709. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>network</td>
<td>Unmanaged Network</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.233.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.233.2. remove DELETE

Table 6.710. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.234. UNMANAGEDNETWORKS

Table 6.711. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of unmanaged networks of the host.</td>
</tr>
</tbody>
</table>

6.234.1. list GET

Returns the list of unmanaged networks of the host.

The order of the returned list of networks isn’t guaranteed.

Table 6.712. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>networks</td>
<td>Unmanaged Network[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
**6.234.1.1. follow**

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

**6.234.1.2. max**

Sets the maximum number of networks to return. If not specified all the networks are returned.

**6.235. USER**

A service to manage a user in the system. Use this service to either get users details or remove users. In order to add new users please use Section 6.236, “Users”.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Gets the system user information.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the system user.</td>
</tr>
</tbody>
</table>

**6.235.1. get GET**

Gets the system user information.

**Usage:**

```shell
GET /ovirt-engine/api/users/1234
```

Will return the user information:

```xml
<user href="/ovirt-engine/api/users/1234" id="1234">
  <name>admin</name>
  <link href="/ovirt-engine/api/users/1234/sshpublickeys" rel="sshpublickeys"/>
  <link href="/ovirt-engine/api/users/1234/roles" rel="roles"/>
  <link href="/ovirt-engine/api/users/1234/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/users/1234/tags" rel="tags"/>
  <department/>
  <domain_entry_id>23456</domain_entry_id>
  <email>user1@domain.com</email>
  <last_name>Lastname</last_name>
  <namespace>*</namespace>
  <principal>user1</principal>
</user>
```
Table 6.714. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td>Out</td>
<td>The system user.</td>
</tr>
</tbody>
</table>

6.235.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.235.2. remove DELETE

Removes the system user.

Usage:

```
DELETE /ovirt-engine/api/users/1234
```

Table 6.715. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.236. USERS

A service to manage the users in the system.

Table 6.716. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add user from a directory service.</td>
</tr>
<tr>
<td>list</td>
<td>List all the users in the system.</td>
</tr>
</tbody>
</table>

6.236.1. add POST

Add user from a directory service.
For example, to add the **myuser** user from the **myextension-authz** authorization provider send a request like this:

```
POST /ovirt-engine/api/users
```

With a request body like this:

```
<user>
  <user_name>myuser@myextension-authz</user_name>
  <domain>
    <name>myextension-authz</name>
  </domain>
</user>
```

In case you are working with Active Directory you have to pass user principal name (UPN) as **username**, followed by authorization provider name. Due to bug 1147900 you need to provide also **principal** parameter set to UPN of the user.

For example, to add the user with UPN **myuser@mysubdomain.mydomain.com** from the **myextension-authz** authorization provider send a request body like this:

```
<user>
  <principal>myuser@mysubdomain.mydomain.com</principal>
  <user_name>myuser@mysubdomain.mydomain.com@myextension-authz</user_name>
  <domain>
    <name>myextension-authz</name>
  </domain>
</user>
```

### Table 6.717. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>User</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.236.2. list GET

List all the users in the system.

Usage:

```
GET /ovirt-engine/api/users
```

Will return the list of users:

```
<users>
  <user href="/ovirt-engine/api/users/1234" id="1234">
    <name>admin</name>
    <link href="/ovirt-engine/api/users/1234/sshpublickeys" rel="sshpublickeys"/>
    <link href="/ovirt-engine/api/users/1234/roles" rel="roles"/>
    <link href="/ovirt-engine/api/users/1234/permissions"/>
  </user>
</users>
```
The order of the returned list of users isn’t guaranteed.

### Table 6.718. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <code>search</code> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of users to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned users.</td>
</tr>
<tr>
<td>users</td>
<td>User[]</td>
<td>Out</td>
<td>The list of users.</td>
</tr>
</tbody>
</table>

#### 6.236.2.1. case_sensitive

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

#### 6.236.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.236.2.3. max

Sets the maximum number of users to return. If not specified all the users are returned.

### 6.237. VIRTUALFUNCTIONALLOWEDNETWORK

#### Table 6.719. Methods summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

### 6.237.1. get GET

Table 6.720. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.237.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.237.2. remove DELETE

Table 6.721. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.238. VIRTUALFUNCTIONALLOWEDNETWORKS

Table 6.722. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of networks.</td>
</tr>
</tbody>
</table>

#### 6.238.1. add POST

Table 6.723. Parameters summary
### 6.238.2. list GET

Returns the list of networks.

The order of the returned list of networks isn’t guaranteed.

#### Table 6.724. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of networks to return.</td>
</tr>
<tr>
<td>networks</td>
<td>Network[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.238.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.238.2.2. max

Sets the maximum number of networks to return. If not specified all the networks are returned.

### 6.239. VM

#### Table 6.725. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancelmigrat...</td>
<td>This operation stops any migration of a virtual machine to another physical host.</td>
</tr>
<tr>
<td>clone</td>
<td></td>
</tr>
<tr>
<td>commitsnapsho...</td>
<td>Permanently restores the virtual machine to the state of the previewed snapshot.</td>
</tr>
<tr>
<td>detach</td>
<td>Detaches a virtual machine from a pool.</td>
</tr>
<tr>
<td>export</td>
<td>Exports the virtual machine.</td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>freezefilesystems</td>
<td>Freezes virtual machine file systems.</td>
</tr>
<tr>
<td>get</td>
<td>Retrieves the description of the virtual machine.</td>
</tr>
<tr>
<td>logon</td>
<td>Initiates the automatic user logon to access a virtual machine from an external console.</td>
</tr>
<tr>
<td>maintenance</td>
<td>Sets the global maintenance mode on the hosted engine virtual machine.</td>
</tr>
<tr>
<td>migrate</td>
<td>Migrates a virtual machine to another physical host.</td>
</tr>
<tr>
<td>previewsnaps hot</td>
<td>Temporarily restores the virtual machine to the state of a snapshot.</td>
</tr>
<tr>
<td>reboot</td>
<td>Sends a reboot request to a virtual machine.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the virtual machine, including the virtual disks attached to it.</td>
</tr>
<tr>
<td>reordermacadresses</td>
<td></td>
</tr>
<tr>
<td>shutdown</td>
<td>This operation sends a shutdown request to a virtual machine.</td>
</tr>
<tr>
<td>start</td>
<td>Starts the virtual machine.</td>
</tr>
<tr>
<td>stop</td>
<td>This operation forces a virtual machine to power-off.</td>
</tr>
<tr>
<td>suspend</td>
<td>This operation saves the virtual machine state to disk and stops it.</td>
</tr>
<tr>
<td>thawfilesystems</td>
<td>Thaws virtual machine file systems.</td>
</tr>
<tr>
<td>ticket</td>
<td>Generates a time-sensitive authentication token for accessing a virtual machine’s display.</td>
</tr>
<tr>
<td>undosnapshot</td>
<td>Restores the virtual machine to the state it had before previewing the snapshot.</td>
</tr>
<tr>
<td>update</td>
<td>Update the virtual machine in the system for the given virtual machine id.</td>
</tr>
</tbody>
</table>

### 6.239.1. cancelmigration POST

This operation stops any migration of a virtual machine to another physical host.

```
POST /ovirt-engine/api/vms/123/cancelmigration
```
The cancel migration action does not take any action specific parameters; therefore, the request body should contain an empty action:

```xml
<action/>
```

### Table 6.726. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the migration should cancelled asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.239.2. clone POST

### Table 6.727. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the clone should be performed asynchronously.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.239.3. commitsnapshot POST

Permanently restores the virtual machine to the state of the previewed snapshot.

See the `preview_snapshot` operation for details.

### Table 6.728. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the snapshots should be committed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.239.4. detach POST

Detaches a virtual machine from a pool.

```plaintext
POST /ovirt-engine/api/vms/123/detach
```

The detach action does not take any action specific parameters; therefore, the request body should contain an empty action:

```xml
<action/>
```

### Table 6.729. Parameters summary
### 6.239.5. export POST

Exports the virtual machine.

A virtual machine can be exported to an export domain. For example, to export virtual machine **123** to the export domain **myexport**:

```
POST /ovirt-engine/api/vms/123/export
```

With a request body like this:

```
<action>
  <storage_domain>
    <name>myexport</name>
  </storage_domain>
  <exclusive>true</exclusive>
  <discard_snapshots>true</discard_snapshots>
</action>
```

Since version 4.2 of the engine it is also possible to export a virtual machine as a virtual appliance (OVA). For example, to export virtual machine **123** as an OVA file named **myvm.ova** that is placed in the directory `/home/ovirt/` on host **myhost**:

```
POST /ovirt-engine/api/vms/123/export
```

With a request body like this:

```
<action>
  <host>
    <name>myhost</name>
  </host>
  <directory>/home/ovirt</directory>
  <filename>myvm.ova</filename>
</action>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the detach action should be performed asynchronously.</td>
</tr>
<tr>
<td>discard_snapshots</td>
<td>Boolean</td>
<td>In</td>
<td>Use the <code>discard_snapshots</code> parameter when the virtual machine should be exported with all of its snapshots collapsed.</td>
</tr>
</tbody>
</table>
Use the `exclusive` parameter when the virtual machine should be exported even if another copy of it already exists in the export domain (override).

The (export) storage domain to export the virtual machine to.

### 6.239.6. `freezefilesystems` POST

Freezes virtual machine file systems.

This operation freezes a virtual machine’s file systems using the QEMU guest agent when taking a live snapshot of a running virtual machine. Normally, this is done automatically by the manager, but this must be executed manually with the API for virtual machines using OpenStack Volume (Cinder) disks.

Example:

```
POST /ovirt-engine/api/vms/123/freezefilesystems
```

### Table 6.731. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>async</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the freeze should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.239.7. `get` GET

Retrieves the description of the virtual machine.

### Table 6.732. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>all_content</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all of the attributes of the virtual machine should be included in the response.</td>
</tr>
<tr>
<td><code>filter</code></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td><code>follow</code></td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Direction</td>
<td>Summary</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>next_run</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the returned result describes the virtual machine as it is currently running or if describes the virtual machine with the modifications that have already been performed but that will only come into effect when the virtual machine is restarted.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Out</td>
<td>Description of the virtual machine.</td>
</tr>
</tbody>
</table>

6.239.7.1. all_content

Indicates if all of the attributes of the virtual machine should be included in the response.

By default the following attributes are excluded:

- console
- `initialization.configuration.data` - The OVF document describing the virtual machine.
- rng_source
- soundcard
- virtio_scsi

For example, to retrieve the complete representation of the virtual machine '123':

```
GET /ovirt-engine/api/vms/123?all_content=true
```

**NOTE**

These attributes are not included by default as they reduce performance. These attributes are seldom used and require additional queries to the database. Only use this parameter when required as it will reduce performance.

6.239.7.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.239.7.3. next_run

Indicates if the returned result describes the virtual machine as it is currently running or if describes the virtual machine with the modifications that have already been performed but that will only come into effect when the virtual machine is restarted. By default the value is `false`.

If the parameter is included in the request, but without a value, it is assumed that the value is `true`. The following request:
GET /vms/{vm:id};next_run

Is equivalent to using the value **true**:

GET /vms/{vm:id};next_run=true

### 6.239.8. logon POST

Initiates the automatic user logon to access a virtual machine from an external console.

This action requires the **ovirt-guest-agent-gdm-plugin** and the **ovirt-guest-agent-pam-module** packages to be installed and the **ovirt-guest-agent** service to be running on the virtual machine.

Users require the appropriate user permissions for the virtual machine in order to access the virtual machine from an external console.

For example:

POST /ovirt-engine/api/vms/123/logon

Request body:

```xml
<action/>
```

**Table 6.733. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the logon should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.239.9. maintenance POST

Sets the global maintenance mode on the hosted engine virtual machine.

This action has no effect on other virtual machines.

Example:

POST /ovirt-engine/api/vms/123/maintenance

```xml
<action>
  <maintenance_enabled>true</maintenance_enabled>
</action>
```

**Table 6.734. Parameters summary**
### 6.239.10. migrate POST

Migrates a virtual machine to another physical host.

Example:

```
POST /ovirt-engine/api/vms/123/migrate
```

To specify a specific host to migrate the virtual machine to:

```
<action>
  <host id="2ab5e1da-b726-4274-bbf7-0a42b16a0fc3"/>
</action>
```

### Table 6.735. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>async</strong></td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the migration should be performed asynchronously.</td>
</tr>
<tr>
<td><strong>cluster</strong></td>
<td>Cluster</td>
<td>In</td>
<td>Specifies the cluster the virtual machine should migrate to.</td>
</tr>
<tr>
<td><strong>force</strong></td>
<td>Boolean</td>
<td>In</td>
<td>Specifies that the virtual machine should migrate even if the virtual machine is defined as non-migratable.</td>
</tr>
<tr>
<td><strong>host</strong></td>
<td>Host</td>
<td>In</td>
<td>Specifies a specific host that the virtual machine should migrate to.</td>
</tr>
</tbody>
</table>

### 6.239.10.1. cluster

Specifies the cluster the virtual machine should migrate to. This is an optional parameter. By default, the virtual machine is migrated to another host within the same cluster.
Live migration to another cluster is not supported. Strongly consider the target cluster’s hardware architecture and network architecture before attempting a migration.

### 6.239.10.2. force

Specifies that the virtual machine should migrate even if the virtual machine is defined as non-migratable. This is an optional parameter. By default, it is set to false.

### 6.239.10.3. host

Specifies a specific host that the virtual machine should migrate to. This is an optional parameter. By default, the Red Hat Virtualization Manager automatically selects a default host for migration within the same cluster. If an API user requires a specific host, the user can specify the host with either an id or name parameter.

### 6.239.11. previewsnapshot POST

Temporarily restores the virtual machine to the state of a snapshot.

The snapshot is indicated with the snapshot.id parameter. It is restored temporarily, so that the content can be inspected. Once that inspection is finished, the state of the virtual machine can be made permanent, using the commit_snapshot method, or discarded using the undo_snapshot method.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the preview should be performed asynchronously.</td>
</tr>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>In</td>
<td>Specify the disks included in the snapshot’s preview.</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>In</td>
<td>Specify the lease storage domain ID to use in the preview of the snapshot.</td>
</tr>
<tr>
<td>restore_memory</td>
<td>Boolean</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

### 6.239.11.1. disks
Specify the disks included in the snapshot’s preview.

For each disk parameter, it is also required to specify its **image_id**.

For example, to preview a snapshot with identifier **456** which includes a disk with identifier **111** and its **image_id** as **222**, send a request like this:

```plaintext
POST /ovirt-engine/api/vms/123/previewsnapshot
```

Request body:

```xml
<action>
  <disks>
    <disk id="111">
      <image_id>222</image_id>
    </disk>
  </disks>
  <snapshot id="456"/>
</action>
```

### 6.239.11.2. lease

Specify the lease storage domain ID to use in the preview of the snapshot. If lease parameter is not passed, then the previewed snapshot lease storage domain will be used. If lease parameter is passed with empty storage domain parameter, then no lease will be used for the snapshot preview. If lease parameter is passed with storage domain parameter then the storage domain ID can be only one of the leases domain IDs that belongs to one of the virtual machine snapshots. This is an optional parameter, set by default to **null**.

### 6.239.12. reboot POST

Sends a reboot request to a virtual machine.

For example:

```plaintext
POST /ovirt-engine/api/vms/123/reboot
```

The reboot action does not take any action specific parameters; therefore, the request body should contain an empty **action**:

```xml
<action/>
```

**Table 6.737. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the reboot should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.239.13. remove DELETE

Removes the virtual machine, including the virtual disks attached to it.
For example, to remove the virtual machine with identifier 123:

```
DELETE /ovirt-engine/api/vms/123
```

Table 6.738. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
<tr>
<td>detach_only</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the attached virtual disks should be detached first and preserved instead of being removed.</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the virtual machine should be forcibly removed.</td>
</tr>
</tbody>
</table>

6.239.13.1. force

Indicates if the virtual machine should be forcibly removed.

Locked virtual machines and virtual machines with locked disk images cannot be removed without this flag set to true.

6.239.14. reordermacaddresses POST

Table 6.739. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.239.15. shutdown POST

This operation sends a shutdown request to a virtual machine.

For example:

```
POST /ovirt-engine/api/vms/123/shutdown
```

The shutdown action does not take any action specific parameters; therefore, the request body should contain an empty action:

```
&action/>
```

Table 6.740. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the shutdown should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.239.16. start POST

Starts the virtual machine.

If the virtual environment is complete and the virtual machine contains all necessary components to function, it can be started.

This example starts the virtual machine:

```
POST /ovirt-engine/api/vms/123/start
```

With a request body:

```
<action/>
```

**Table 6.741. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the start action should be performed asynchronously.</td>
</tr>
<tr>
<td>authorize</td>
<td>AuthorizedKey</td>
<td>In</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>pause</td>
<td>Boolean</td>
<td>In</td>
<td>If set to <code>true</code>, start the virtual machine in paused mode.</td>
</tr>
<tr>
<td>use_cloud</td>
<td>Boolean</td>
<td>In</td>
<td>If set to <code>true</code>, the initialization type is set to <code>cloud-init</code>.</td>
</tr>
<tr>
<td>_init</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>use_sysprep</td>
<td>Boolean</td>
<td>In</td>
<td>If set to <code>true</code>, the initialization type is set to <code>Sysprep</code>.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In</td>
<td>The definition of the virtual machine for this specific run.</td>
</tr>
<tr>
<td>volatile</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates that this run configuration will be discarded even in the case of guest-initiated reboot.</td>
</tr>
</tbody>
</table>

### 6.239.16.1. pause
If set to `true`, start the virtual machine in paused mode. The default is `false`.

### 6.239.16.2. use_cloud_init

If set to `true`, the initialization type is set to `cloud-init`. The default value is `false`. See [this](#) for details.

### 6.239.16.3. use_sysprep

If set to `true`, the initialization type is set to `Sysprep`. The default value is `false`. See [this](#) for details.

### 6.239.16.4. vm

The definition of the virtual machine for this specific run.

For example:

```xml
<action>
  <vm>
    <os>
      <boot>
        <devices>
          <device>cdrom</device>
        </devices>
      </boot>
    </os>
  </vm>
</action>
```

This will set the boot device to the CDROM only for this specific start. After the virtual machine is powered off, this definition will be reverted.

### 6.239.16.5. volatile

Indicates that this run configuration will be discarded even in the case of guest-initiated reboot. The default value is `false`.

### 6.239.17. stop POST

This operation forces a virtual machine to power-off.

For example:

```bash
POST /ovirt-engine/api/vms/123/stop
```

The stop action does not take any action specific parameters; therefore, the request body should contain an empty `action`:

```xml
<action/>
```

**Table 6.742. Parameters summary**
6.239.18. suspend POST

This operation saves the virtual machine state to disk and stops it. Start a suspended virtual machine and restore the virtual machine state with the start action.

For example:

POST /ovirt-engine/api/vms/123/suspend

The suspend action does not take any action specific parameters; therefore, the request body should contain an empty action:

<action/>

Table 6.743. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the suspend action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.239.19. thawfilesystems POST

Thaws virtual machine file systems.

This operation thaws a virtual machine’s file systems using the QEMU guest agent when taking a live snapshot of a running virtual machine. Normally, this is done automatically by the manager, but this must be executed manually with the API for virtual machines using OpenStack Volume (Cinder) disks.

Example:

POST /api/vms/123/thawfilesystems

<action/>

Table 6.744. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the thaw file systems action should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.239.20. ticket POST
Generates a time-sensitive authentication token for accessing a virtual machine’s display.

For example:

```
POST /ovirt-engine/api/vms/123/ticket
```

The client-provided action optionally includes a desired ticket value and/or an expiry time in seconds.

The response specifies the actual ticket value and expiry used.

```
<action>
  <ticket>
    <value>abcd12345</value>
    <expiry>120</expiry>
  </ticket>
</action>
```

**IMPORTANT**

If the virtual machine is configured to support only one graphics protocol then the generated authentication token will be valid for that protocol. But if the virtual machine is configured to support multiple protocols, VNC and SPICE, then the authentication token will only be valid for the SPICE protocol.

In order to obtain an authentication token for a specific protocol, for example for VNC, use the `ticket` method of the `service`, which manages the graphics consoles of the virtual machine, by sending a request:

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/ticket
```

### Table 6.745. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the generation of the ticket should be performed asynchronously.</td>
</tr>
<tr>
<td>ticket</td>
<td>Ticket</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.239.21. undosnapshot POST

Restores the virtual machine to the state it had before previewing the snapshot.

See the `preview_snapshot` operation for details.

### Table 6.746. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the undo snapshot action should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.239.22. update PUT

Update the virtual machine in the system for the given virtual machine id.

Table 6.747. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>next_run</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be applied to the virtual machine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>immediately or if it should be applied only when the virtual machine</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>is restarted.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.239.22.1. next_run

Indicates if the update should be applied to the virtual machine immediately or if it should be applied only when the virtual machine is restarted. The default value is false, so by default changes are applied immediately.

6.240. VMAPPLICATION

A service that provides information about an application installed in a virtual machine.

Table 6.748. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the information about the application.</td>
</tr>
</tbody>
</table>

6.240.1. get GET

Returns the information about the application.

Table 6.749. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>application</td>
<td>Application</td>
<td>Out</td>
<td>The information about the application.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.240.1.1. application

The information about the application.

The information consists of `name` attribute containing the name of the application (which is an arbitrary string that may also contain additional information such as version) and `vm` attribute identifying the virtual machine.

For example, a request like this:

```plaintext
GET /ovirt-engine/api/vms/123/applications/789
```

May return information like this:

```xml
<application href="/ovirt-engine/api/vms/123/applications/789" id="789">
  <name>ovirt-guest-agent-common-1.0.12-3.el7</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
```

6.240.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.241. VMAPPLICATIONS

A service that provides information about applications installed in a virtual machine.

Table 6.750. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns a list of applications installed in the virtual machine.</td>
</tr>
</tbody>
</table>

6.241.1. list GET

Returns a list of applications installed in the virtual machine.

The order of the returned list of applications isn’t guaranteed.

Table 6.751. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>applicati ons</td>
<td>Application[]</td>
<td>Out</td>
<td>A list of applications installed in the virtual machine.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.241.1.1. applications

A list of applications installed in the virtual machine.

For example, a request like this:

```
GET /ovirt-engine/api/vms/123/applications/
```

May return a list like this:

```
<applications>
  <application href="/ovirt-engine/api/vms/123/applications/456" id="456">
    <name>kernel-3.10.0-327.36.1.el7</name>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </application>
  <application href="/ovirt-engine/api/vms/123/applications/789" id="789">
    <name>ovirt-guest-agent-common-1.0.12-3.el7</name>
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
  </application>
</applications>
```

6.241.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.241.1.3. max

Sets the maximum number of applications to return. If not specified all the applications are returned.

6.242. VMCDROM

Manages a CDROM device of a virtual machine.

Changing and ejecting the disk is done using always the update method, to change the value of the file attribute.

### Table 6.752. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the information about this CDROM device.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the information about this CDROM device.</td>
</tr>
</tbody>
</table>
6.242.1. **get** GET

Returns the information about this CDROM device.

The information consists of `cdrom` attribute containing reference to the CDROM device, the virtual machine, and optionally the inserted disk.

If there is a disk inserted then the `file` attribute will contain a reference to the ISO image:

```
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <file id="mycd.iso"/>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</cdrom>
```

If there is no disk inserted then the `file` attribute won’t be reported:

```
<cdrom href="..." id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</cdrom>
```

### Table 6.753. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Cdrom</td>
<td>Out</td>
<td>The information about the CDROM device.</td>
</tr>
<tr>
<td>current</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the operation should return the information for the currently running virtual machine.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

#### 6.242.1.1. current

Indicates if the operation should return the information for the currently running virtual machine. This parameter is optional, and the default value is `false`.

#### 6.242.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.242.2. update PUT

Updates the information about this CDROM device.

It allows to change or eject the disk by changing the value of the `file` attribute. For example, to insert or change the disk send a request like this:

```
PUT /ovirt-engine/api/vms/123/cdroms/00000000-0000-0000-0000-000000000000
```

The body should contain the new value for the `file` attribute:
The value of the id attribute, mycd.iso in this example, should correspond to a file available in an attached ISO storage domain.

To eject the disk use a file with an empty id:

```
<cdrom>
  <file id=""/>
</cdrom>
```

By default the above operations change permanently the disk that will be visible to the virtual machine after the next boot, but they don't have any effect on the currently running virtual machine. If you want to change the disk that is visible to the current running virtual machine, add the current=true parameter. For example, to eject the current disk send a request like this:

```
PUT /ovirt-engine/api/vms/123/cdroms/00000000-0000-0000-0000-000000000000?current=true
```

With a request body like this:

```
<cdrom>
  <file id=""/>
</cdrom>
```

**IMPORTANT**

The changes made with the current=true parameter are never persisted, so they won't have any effect after the virtual machine is rebooted.

### Table 6.754. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Cdrom</td>
<td>In/Out</td>
<td>The information about the CDROM device.</td>
</tr>
<tr>
<td>current</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should apply to the currently running virtual machine, or to the virtual machine after the next boot.</td>
</tr>
</tbody>
</table>

#### 6.242.2.1. current

Indicates if the update should apply to the currently running virtual machine, or to the virtual machine after the next boot. This parameter is optional, and the default value is false, which means that by default the update will have effect only after the next boot.

#### 6.243. VMCDROMS

Manages the CDROM devices of a virtual machine.
Currently virtual machines have exactly one CDROM device. No new devices can be added, and the existing one can’t be removed, thus there are no add or remove methods. Changing and ejecting CDROM disks is done with the update method of the service that manages the CDROM device.

### Table 6.755. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a cdrom to a virtual machine identified by the given id.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of CDROM devices of the virtual machine.</td>
</tr>
</tbody>
</table>

#### 6.243.1. add POST

Add a cdrom to a virtual machine identified by the given id.

### Table 6.756. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Cdrom</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.243.2. list GET

Returns the list of CDROM devices of the virtual machine.

The order of the returned list of CD-ROM devices isn’t guaranteed.

### Table 6.757. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>Out</td>
<td>The list of CDROM devices of the virtual machine.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of CDROMs to return.</td>
</tr>
</tbody>
</table>

#### 6.243.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.243.2.2. max

Sets the maximum number of CDROMs to return. If not specified all the CDROMs are returned.

### 6.244. VMDISK

### Table 6.758. Methods summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate</td>
<td></td>
</tr>
<tr>
<td>deactivate</td>
<td></td>
</tr>
<tr>
<td>export</td>
<td></td>
</tr>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>move</td>
<td></td>
</tr>
<tr>
<td>reduce</td>
<td>Reduces the size of the disk image.</td>
</tr>
<tr>
<td>remove</td>
<td>Detach the disk from the virtual machine.</td>
</tr>
<tr>
<td>update</td>
<td></td>
</tr>
</tbody>
</table>

### 6.244.1. activate POST

#### Table 6.759. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the activation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.244.2. deactivate POST

#### Table 6.760. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the deactivation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.244.3. export POST

#### Table 6.761. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the export should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
</tbody>
</table>
6.244.4. get GET

Table 6.762. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>

6.244.4.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.244.5. move POST

Table 6.763. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the move should be performed asynchronously.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
</tbody>
</table>

6.244.6. reduce POST

Reduces the size of the disk image.

Invokes *reduce* on the logical volume (i.e. this is only applicable for block storage domains). This is applicable for floating disks and disks attached to non-running virtual machines. There is no need to specify the size as the optimal size is calculated automatically.

Table 6.764. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.244.7. remove DELETE
Detach the disk from the virtual machine.

**NOTE**

In version 3 of the API this used to also remove the disk completely from the system, but starting with version 4 it doesn't. If you need to remove it completely use the remove method of the top level disk service.

### Table 6.765. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.244.8. update PUT

**Table 6.766. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.245. VMDISKS

**Table 6.767. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of disks of the virtual machine.</td>
</tr>
</tbody>
</table>

### 6.245.1. add POST

**Table 6.768. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.245.2. list GET

Returns the list of disks of the virtual machine.
The order of the returned list of disks isn’t guaranteed.

### Table 6.769. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td>Out</td>
<td></td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of disks to return.</td>
</tr>
</tbody>
</table>

#### 6.245.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.245.2.2. max

Sets the maximum number of disks to return. If not specified all the disks are returned.

### 6.246. VMGRAPHICSCONSOLE

#### Table 6.770. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves the graphics console configuration of the virtual machine.</td>
</tr>
<tr>
<td>proxysite</td>
<td></td>
</tr>
<tr>
<td>remoteviewer</td>
<td>Generates the file which is compatible with remote-viewer client.</td>
</tr>
<tr>
<td>connectionfile</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Remove the graphics console from the virtual machine.</td>
</tr>
<tr>
<td>ticket</td>
<td>Generates a time-sensitive authentication token for accessing this virtual machine's console.</td>
</tr>
</tbody>
</table>

#### 6.246.1. get GET

Retrieves the graphics console configuration of the virtual machine.
IMPORTANT

By default, when the current parameter is not specified, the data returned corresponds to the next execution of the virtual machine. In the current implementation of the system this means that the address and port attributes will not be populated because the system does not know what address and port will be used for the next execution. Since in most cases those attributes are needed, it is strongly advised to always explicitly include the current parameter with the value true.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>console</td>
<td>GraphicsConsole</td>
<td>Out</td>
<td>The information about the graphics console of the virtual machine.</td>
</tr>
<tr>
<td>current</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if the data returned should correspond to the next execution of the virtual machine, or to the current execution.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>

6.246.1.1. current

Specifies if the data returned should correspond to the next execution of the virtual machine, or to the current execution.

IMPORTANT

The address and port attributes will not be populated unless the value is true.

For example, to get data for the current execution of the virtual machine, including the address and port attributes, send a request like this:

```
GET /ovit-engine/api/vms/123/graphicsconsoles/456?current=true
```

The default value is false.

6.246.1.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.246.2. proxyticket POST

Table 6.772. Parameters summary
async | Boolean | In | Indicates if the generation of the ticket should be performed asynchronously.

proxy_ticket | ProxyTicket | Out |

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the generation of the ticket should be performed asynchronously.</td>
</tr>
<tr>
<td>proxy_ticket</td>
<td>ProxyTicket</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.246.3. remoteviewerconnectionfile POST

Generates the file which is compatible with `remote-viewer` client.

Use the following request to generate remote viewer connection file of the graphics console. Note that this action generates the file only if virtual machine is running.

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/remoteviewerconnectionfile
```

The `remoteviewerconnectionfile` action does not take any action specific parameters, so the request body should contain an empty `action`:

```
<action/>
```

The response contains the file, which can be used with `remote-viewer` client.

```
<action>
  <remote_viewer_connection_file>
    [virt-viewer]
    type=spice
    host=192.168.1.101
    port=-1
    password=123456789
    delete-this-file=1
    fullscreen=0
    toggle-fullscreen=shift+f11
    release-cursor=shift+f12
    secure-attention=ctrl+alt+end
    tls-port=5900
    enable-smartcard=0
    enable-usb-autoshare=0
    usb-filter=null
    tls-ciphers=DEFAULT
    host-subject=O=local,CN=example.com
    ca=...
  </remote_viewer_connection_file>
</action>
```

E.g., to fetch the content of remote viewer connection file and save it into temporary file, user can use oVirt Python SDK as follows:

```
# Find the virtual machine:
vm = vms_service.list(search='name=myvm')[0]
```
When you create the remote viewer connection file, then you can connect to virtual machine graphic console, as follows:

```python
# Locate the service that manages the virtual machine, as that is where # the locators are defined:
vm_service = vms_service.vm_service(vm.id)

# Find the graphic console of the virtual machine:
graphics_consoles_service = vm_service.graphics_consoles_service()
graphics_console = graphics_consoles_service.list()[0]

# Generate the remote viewer connection file:
console_service =
graphics_consoles_service.console_service(graphics_console.id)
remote_viewer_connection_file =
console_service.remote_viewer_connection_file()

# Write the content to file "/tmp/remote_viewer_connection_file.vv"
path = "/tmp/remote_viewer_connection_file.vv"
with open(path, "w") as f:
    f.write(remote_viewer_connection_file)
```

When you create the remote viewer connection file, then you can connect to virtual machine graphic console, as follows:

```bash
#!/bin/sh -ex
remote-viewer --ovirt-ca-file=/etc/pki/ovirt-engine/ca.pem
/tmp/remote_viewer_connection_file.vv
```

### Table 6.773. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>remote_viewer_connection_file</td>
<td>String</td>
<td>Out</td>
<td>Contains the file which is compatible with remote-viewer client.</td>
</tr>
</tbody>
</table>

### 6.246.3.1. remote_viewer_connection_file

Contains the file which is compatible with remote-viewer client.

User can use the content of this attribute to create a file, which can be passed to remote-viewer client to connect to virtual machine graphic console.

### 6.246.4. remove DELETE

Remove the graphics console from the virtual machine.

### Table 6.774. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 6.246.5. ticket POST

Generates a time-sensitive authentication token for accessing this virtual machine’s console.

```
POST /ovirt-engine/api/vms/123/graphicsconsoles/456/ticket
```

The client-provided action optionally includes a desired ticket value and/or an expiry time in seconds.

In any case, the response specifies the actual ticket value and expiry used.

```
<action>
  <ticket>
    <value>abcd12345</value>
    <expiry>120</expiry>
  </ticket>
</action>
```

#### Table 6.775. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ticket</td>
<td>Ticket</td>
<td>In/Out</td>
<td>The generated ticket that can be used to access this console.</td>
</tr>
</tbody>
</table>

### 6.247. VMGRAPHICSCONSOLES

#### Table 6.776. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add new graphics console to the virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>Lists all the configured graphics consoles of the virtual machine.</td>
</tr>
</tbody>
</table>

#### 6.247.1. add POST

Add new graphics console to the virtual machine.

#### Table 6.777. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
</table>
6.247.2. list GET

Lists all the configured graphics consoles of the virtual machine.

**IMPORTANT**

By default, when the `current` parameter is not specified, the data returned corresponds to the next execution of the virtual machine. In the current implementation of the system this means that the `address` and `port` attributes will not be populated because the system does not know what address and port will be used for the next execution. Since in most cases those attributes are needed, it is strongly advised to always explicitly include the `current` parameter with the value `true`.

The order of the returned list of graphics consoles is not guaranteed.

### Table 6.778. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>consoles</td>
<td>GraphicsConsole</td>
<td>Out</td>
<td>The list of graphics consoles of the virtual machine.</td>
</tr>
<tr>
<td>current</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if the data returned should correspond to the next execution of the virtual machine, or to the current execution.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of consoles to return.</td>
</tr>
</tbody>
</table>

#### 6.247.2.1. current

Specifies if the data returned should correspond to the next execution of the virtual machine, or to the current execution.

**IMPORTANT**

The `address` and `port` attributes will not be populated unless the value is `true`.

For example, to get data for the current execution of the virtual machine, including the `address` and `port` attributes, send a request like this:

```bash
GET /ovirt-engine/api/vms/123/graphicsconsoles?current=true
```
The default value is `false`.

### 6.247.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.247.2.3. max

Sets the maximum number of consoles to return. If not specified all the consoles are returned.

### 6.248. VMHOSTDEVICE

A service to manage individual host device attached to a virtual machine.

#### Table 6.779. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieve information about particular host device attached to given virtual machine.</td>
</tr>
<tr>
<td>remove</td>
<td>Remove the attachment of this host device from given virtual machine.</td>
</tr>
</tbody>
</table>

#### 6.248.1. get GET

Retrieve information about particular host device attached to given virtual machine.

Example:

```plaintext
GET /ovirt-engine/api/vms/123/hostdevices/456
```

```xml
<host_device href="/ovirt-engine/api/hosts/543/devices/456" id="456">
  <name>pci_0000_04_00_0</name>
  <capability>pci</capability>
  <iommu_group>30</iommu_group>
  <placeholder>true</placeholder>
  <product id="0x13ba">
    <name>GM107GL [Quadro K2200]</name>
  </product>
  <vendor id="0x10de">
    <name>NVIDIA Corporation</name>
  </vendor>
  <parent_device href="/ovirt-engine/api/hosts/543" id="543"/>
  <parent_device href="/ovirt-engine/api/hosts/543/devices/456" id="456">
    <name>pci_0000_00_03_0</name>
  </parent_device>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</host_device>
```

#### Table 6.780. Parameters summary
### Table 6.781. Parameters summary

| Name  | Type      | Direction | Summary                                                        |
|-------|-----------|-----------|                                                               |
| async | Boolean   | In        | Indicates if the remove should be performed asynchronously.   |

### 6.249. VMHOSTDEVICES

A service to manage host devices attached to a virtual machine.

#### Table 6.782. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Attach target device to given virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>List the host devices assigned to given virtual machine.</td>
</tr>
</tbody>
</table>

#### 6.249.1. add POST

Attach target device to given virtual machine.
Example:

POST /ovirt-engine/api/vms/123/hostdevices

With request body of type HostDevice, for example

```xml
<host_device id="123" />
```

**NOTE**
A necessary precondition for a successful host device attachment is that the virtual machine must be pinned to **exactly** one host. The device ID is then taken relative to this host.

**NOTE**
Attachment of a PCI device that is part of a bigger IOMMU group will result in attachment of the remaining devices from that IOMMU group as "placeholders". These devices are then identified using the placeholder attribute of the HostDevice type set to true.

In case you want attach a device that already serves as an IOMMU placeholder, simply issue an explicit Add operation for it, and its placeholder flag will be cleared, and the device will be accessible to the virtual machine.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
<td>HostDevice</td>
<td>In/Out</td>
<td>The host device to be attached to given virtual machine.</td>
</tr>
</tbody>
</table>

### 6.249.2. list GET

List the host devices assigned to given virtual machine.

The order of the returned list of devices isn’t guaranteed.

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
</tr>
</tbody>
</table>

#### 6.249.2.1. follow

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
</tr>
<tr>
<td>max</td>
</tr>
</tbody>
</table>
Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.249.2.2. max

Sets the maximum number of devices to return. If not specified all the devices are returned.

### 6.250. VMNIC

Table 6.785. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activate</td>
<td></td>
</tr>
<tr>
<td>deactivate</td>
<td></td>
</tr>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes the NIC.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the NIC.</td>
</tr>
</tbody>
</table>

### 6.250.1. activate POST

Table 6.786. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the activation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.250.2. deactivate POST

Table 6.787. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the deactivation should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.250.3. get GET

Table 6.788. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
</tbody>
</table>
#### 6.250.3.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.250.4. remove DELETE

Removes the NIC.

For example, to remove the NIC with id 456 from the virtual machine with id 123 send a request like this:

```plaintext
DELETE /ovirt-engine/api/vms/123/nics/456
```

**IMPORTANT**

The hotplugging feature only supports virtual machine operating systems with hotplugging operations. Example operating systems include:

- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 5
- Windows Server 2008 and
- Windows Server 2003

#### Table 6.789. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

#### 6.250.5. update PUT

Updates the NIC.

For example, to update the NIC having with 456 belonging to virtual the machine with id 123 send a request like this:

```plaintext
PUT /ovirt-engine/api/vms/123/nics/456
```

With a request body like this:
The hotplugging feature only supports virtual machine operating systems with hotplugging operations. Example operating systems include:

- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 5
- Windows Server 2008 and
- Windows Server 2003

### Table 6.790. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>nic</td>
<td>Nic</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.251. VMNICS

### Table 6.791. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds a NIC to the virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of NICs of the virtual machine.</td>
</tr>
</tbody>
</table>

### 6.251.1. add POST

Adds a NIC to the virtual machine.

The following example adds to the virtual machine 123 a network interface named mynic using virtio and the NIC profile 456.

```
POST /ovirt-engine/api/vms/123/nics

<nic>
    <name>mynic</name>
    <interface>virtio</interface>
</nic>
```
The following example sends that request using `curl`:

```
curl \
--request POST \
--header "Version: 4" \
--header "Content-Type: application/xml" \
--header "Accept: application/xml" \
--user "admin@internal:mypassword" \
--cacert /etc/pki/ovirt-engine/ca.pem \
--data ' 
  <nic>
    <name>mynic</name>
    <interface>virtio</interface>
    <vnic_profile id="456"/>
  </nic>
', \
https://myengine.example.com/ovirt-engine/api/vms/123/nics
```

### IMPORTANT

The hotplugging feature only supports virtual machine operating systems with hotplugging operations. Example operating systems include:

- Red Hat Enterprise Linux 6
- Red Hat Enterprise Linux 5
- Windows Server 2008 and
- Windows Server 2003

### Table 6.792. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>nic</td>
<td>Nic</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.251.2. list GET

Returns the list of NICs of the virtual machine.

The order of the returned list of NICs isn't guaranteed.

### Table 6.793. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
### 6.251.2. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.251.2.2. max

Sets the maximum number of NICs to return. If not specified all the NICs are returned.

### 6.252. VMNUMANODE

#### Table 6.794. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td>Removes a virtual NUMA node.</td>
</tr>
<tr>
<td>update</td>
<td>Updates a virtual NUMA node.</td>
</tr>
</tbody>
</table>

#### 6.252.1. get GET

#### Table 6.795. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>node</td>
<td>VirtualNuma Node</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.252.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.252.2. remove DELETE

Removes a virtual NUMA node.

An example of removing a virtual NUMA node:
DELETE /ovirt-engine/api/vms/123/numanodes/456

NOTE

It’s required to remove the numa nodes from the highest index first.

Table 6.796. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.252.3. update PUT

Updates a virtual NUMA node.

An example of pinning a virtual NUMA node to a physical NUMA node on the host:

PUT /ovirt-engine/api/vms/123/numanodes/456

The request body should contain the following:

```
<vm_numa_node>
  <numa_node_pins>
    <numa_node_pin>
      <index>0</index>
    </numa_node_pin>
  </numa_node_pins>
</vm_numa_node>
```

Table 6.797. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>node</td>
<td>VirtualNuma Node</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.253. VMNUMANODES

Table 6.798. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new virtual NUMA node for the virtual machine.</td>
</tr>
</tbody>
</table>
### 6.253.1. add POST

Creates a new virtual NUMA node for the virtual machine.

An example of creating a NUMA node:

```plaintext
POST /ovirt-engine/api/vms/c7ecd2dc/numanodes
Accept: application/xml
Content-type: application/xml
```

The request body can contain the following:

```xml
<vm_numa_node>
  <cpu>
    <cores>
      <core>
        <index>0</index>
      </core>
    </cores>
  </cpu>
  <index>0</index>
  <memory>1024</memory>
</vm_numa_node>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>node</td>
<td>VirtualNuma Node</td>
<td>In/Out</td>
<td>Lists virtual NUMA nodes of a virtual machine.</td>
</tr>
</tbody>
</table>

### 6.253.2. list GET

Lists virtual NUMA nodes of a virtual machine.

The order of the returned list of NUMA nodes isn’t guaranteed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of nodes to return.</td>
</tr>
</tbody>
</table>
6.253.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.253.2.2. max

Sets the maximum number of nodes to return. If not specified all the nodes are returned.

6.254. VMPOOL

A service to manage a virtual machines pool.

Table 6.801. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocatevm</td>
<td>This operation allocates a virtual machine in the virtual machine pool.</td>
</tr>
<tr>
<td>get</td>
<td>Get the virtual machine pool.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes a virtual machine pool.</td>
</tr>
<tr>
<td>update</td>
<td>Update the virtual machine pool.</td>
</tr>
</tbody>
</table>

6.254.1. allocatevm POST

This operation allocates a virtual machine in the virtual machine pool.

```markdown
POST /ovirt-engine/api/vmpools/123/allocatevm
```

The allocate virtual machine action does not take any action specific parameters, so the request body should contain an empty `action`:

```xml
<action/>
```

Table 6.802. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the allocation should be performed asynchronously.</td>
</tr>
</tbody>
</table>
6.254.2. get GET

Get the virtual machine pool.

GET /ovirt-engine/api/vmpools/123

You will get a XML response like that one:

```
<vm_pool id="123">
  <actions>...</actions>
  <name>MyVmPool</name>
  <description>MyVmPool description</description>
  <link href="/ovirt-engine/api/vmpools/123/permissions" rel="permissions"/>
  <max_user_vms>1</max_user_vms>
  <prestarted_vms>0</prestarted_vms>
  <size>100</size>
  <stateful>false</stateful>
  <type>manual</type>
  <use_latest_template_version>false</use_latest_template_version>
  <cluster id="123"/>
  <template id="123"/>
  <vm id="123">...</vm>
  ...
</vm_pool>
```

Table 6.803. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>pool</td>
<td>VmPool</td>
<td>Out</td>
<td>Retrieved virtual machines pool.</td>
</tr>
</tbody>
</table>

6.254.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.254.3. remove DELETE

Removes a virtual machine pool.

DELETE /ovirt-engine/api/vmpools/123

Table 6.804. Parameters summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.254.4. update PUT

Update the virtual machine pool.

PUT /ovirt-engine/api/vmpools/123

The `name`, `description`, `size`, `prestarted_vms` and `max_user_vms` attributes can be updated after the virtual machine pool has been created.

```xml
<vmpool>
  <name>VM_Pool_B</name>
  <description>Virtual Machine Pool B</description>
  <size>3</size>
  <prestarted_vms>1</prestarted_vms>
  <max_user_vms>2</max_user_vms>
</vmpool>
```

Table 6.805. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>pool</td>
<td>VmPool</td>
<td>In/Out</td>
<td>The virtual machine pool that is being updated.</td>
</tr>
</tbody>
</table>

### 6.255. VMPOOLS

Provides read-write access to virtual machines pools.

Table 6.806. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new virtual machine pool.</td>
</tr>
<tr>
<td>list</td>
<td>Get a list of available virtual machines pools.</td>
</tr>
</tbody>
</table>

### 6.255.1. add POST

Creates a new virtual machine pool.
A new pool requires the **name**, **cluster** and **template** attributes. Identify the cluster and template with the **id** or **name** nested attributes:

```
POST /ovirt-engine/api/vmpools
```

With the following body:

```
<vmpool>
  <name>mypool</name>
  <cluster id="123"/>
  <template id="456"/>
</vmpool>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool</td>
<td>VmPool</td>
<td>In/Out</td>
<td>Pool to add.</td>
</tr>
</tbody>
</table>

### 6.255.2. list GET

Get a list of available virtual machines pools.

```
GET /ovirt-engine/api/vmpools
```

You will receive the following response:

```
<vm_pools>
  <vm_pool id="123">...
    ...
  </vm_pool>
  ...
</vm_pools>
```

The order of the returned list of pools is guaranteed only if the **sortby** clause is included in the **search** parameter.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the <strong>search</strong> parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
max | Integer | In | Sets the maximum number of pools to return.
--- | --- | --- | ---
pools | VmPool[] | Out | Retrieved pools.
search | String | In | A query string used to restrict the returned pools.

### 6.255.2.1. case_sensitive

Indicates if the search performed using the `search` parameter should be performed taking case into account. The default value is `true`, which means that case is taken into account. If you want to search ignoring case set it to `false`.

### 6.255.2.2. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.255.2.3. max

Sets the maximum number of pools to return. If this value is not specified, all of the pools are returned.

### 6.256. VMREPORTEDDEVICE

#### Table 6.809. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.256.1. get GET

#### Table 6.810. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>reported_device</td>
<td>ReportedDevice</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

#### 6.256.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.257. VMREPORTEDDEVICES
Table 6.811. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Returns the list of reported devices of the virtual machine.</td>
</tr>
</tbody>
</table>

### 6.257.1. list GET

Returns the list of reported devices of the virtual machine.

The order of the returned list of devices isn’t guaranteed.

Table 6.812. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of devices to return.</td>
</tr>
<tr>
<td>reported_device</td>
<td>ReportedDevice[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.257.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

### 6.257.1.2. max

Sets the maximum number of devices to return. If not specified all the devices are returned.

### 6.258. VMSESSION

Table 6.813. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
</tbody>
</table>

### 6.258.1. get GET

Table 6.814. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
</tbody>
</table>
6.258.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.259. VMSESSIONS

Provides information about virtual machine user sessions.

Table 6.815. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>Lists all user sessions for this virtual machine.</td>
</tr>
</tbody>
</table>

6.259.1. list GET

Lists all user sessions for this virtual machine.

For example, to retrieve the session information for virtual machine 123 send a request like this:

```
GET /ovirt-engine/api/vms/123/sessions
```

The response body will contain something like this:

```
<sessions>
  <session href="/ovirt-engine/api/vms/123/sessions/456" id="456">
    <console_user>true</console_user>
    <ip>
      <address>192.168.122.1</address>
    </ip>
  </session>

  ...
  ...
  ...
</sessions>
```

The order of the returned list of sessions isn’t guaranteed.

Table 6.816. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of sessions to return.</td>
</tr>
</tbody>
</table>
### 6.259.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

### 6.259.1.2. max

Sets the maximum number of sessions to return. If not specified all the sessions are returned.

### 6.260. VMWATCHDOG

A service managing a watchdog on virtual machines.

#### Table 6.817. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Returns the information about the watchdog.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the watchdog from the virtual machine.</td>
</tr>
<tr>
<td>update</td>
<td>Updates the information about the watchdog.</td>
</tr>
</tbody>
</table>

#### 6.260.1. get GET

Returns the information about the watchdog.

#### Table 6.818. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>Out</td>
<td>The information about the watchdog.</td>
</tr>
</tbody>
</table>

#### 6.260.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See [here](#) for details.

#### 6.260.1.2. watchdog

The information about the watchdog.
The information consists of `model` element, `action` element and the reference to the virtual machine. It may look like this:

```xml
<watchdogs>
  <watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
    <action>poweroff</action>
    <model>i6300esb</model>
  </watchdog>
</watchdogs>
```

### 6.260.2. remove DELETE

Removes the watchdog from the virtual machine.

For example, to remove a watchdog from a virtual machine, send a request like this:

```
DELETE /ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

### 6.260.3. update PUT

Updates the information about the watchdog.

You can update the information using `action` and `model` elements.

For example, to update a watchdog, send a request like this:

```
PUT /ovirt-engine/api/vms/123/watchdogs
<watchdog>
  <action>reset</action>
</watchdog>
```

with response body:

```xml
<watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <action>reset</action>
  <model>i6300esb</model>
</watchdog>
```

### Table 6.820. Parameters summary
### Name Type Direction Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>In/Out</td>
<td>The information about the watchdog.</td>
</tr>
</tbody>
</table>

#### 6.260.3.1. watchdog

The information about the watchdog.

The request data must contain at least one of `model` and `action` elements. The response data contains complete information about the updated watchdog.

#### 6.261. VMWATCHDOGS

Lists the watchdogs of a virtual machine.

**Table 6.821. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Adds new watchdog to the virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>The list of watchdogs of the virtual machine.</td>
</tr>
</tbody>
</table>

**6.261.1. add POST**

Adds new watchdog to the virtual machine.

For example, to add a watchdog to a virtual machine, send a request like this:

```
POST /ovirt-engine/api/vms/123/watchdogs
<watchdog>
    <action>poweroff</action>
    <model>i6300esb</model>
</watchdog>
```

with response body:

```
<watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
    <action>poweroff</action>
    <model>i6300esb</model>
</watchdog>
```

**Table 6.822. Parameters summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>watchdog</td>
<td>Watchdog</td>
<td>In/Out</td>
<td>The information about the watchdog.</td>
</tr>
</tbody>
</table>

### 6.261.1. watchdog

The information about the watchdog.

The request data must contain `model` element (such as `i6300esb`) and `action` element (one of `none`, `reset`, `poweroff`, `dump`, `pause`). The response data additionally contains references to the added watchdog and to the virtual machine.

### 6.261.2. list GET

The list of watchdogs of the virtual machine.

The order of the returned list of watchdogs isn’t guaranteed.

#### Table 6.823. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of watchdogs to return.</td>
</tr>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>Out</td>
<td>The information about the watchdog.</td>
</tr>
</tbody>
</table>

#### 6.261.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

#### 6.261.2.2. max

Sets the maximum number of watchdogs to return. If not specified all the watchdogs are returned.

#### 6.261.2.3. watchdogs

The information about the watchdog.

The information consists of `model` element, `action` element and the reference to the virtual machine. It may look like this:

```xml
<watchdogs>
  <watchdog href="/ovirt-engine/api/vms/123/watchdogs/00000000-0000-0000-0000-000000000000" id="00000000-0000-0000-0000-000000000000">
    <vm href="/ovirt-engine/api/vms/123" id="123"/>
    <action>poweroff</action>
  </watchdog>
</watchdogs>
```
6.262. VMS

Table 6.824. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Creates a new virtual machine.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of virtual machines of the system.</td>
</tr>
</tbody>
</table>

6.262.1. add POST

Creates a new virtual machine.

The virtual machine can be created in different ways:

- From a template. In this case the identifier or name of the template must be provided. For example, using a plain shell script and XML:

```
#!/bin/sh -ex

url="https://engine.example.com/ovirt-engine/api"
user="admin@internal"
password="..."
curl \
  --verbose \n  --cacert /etc/pki/ovirt-engine/ca.pem \n  --user "$user":$password" \n  --request POST \n  --header "Version: 4" \n  --header "Content-Type: application/xml" \n  --header "Accept: application/xml" \n  --data '<vm>
    <name>myvm</name>
    <template>
      <name>Blank</name>
    </template>
    <cluster>
      <name>mycluster</name>
    </cluster>
  </vm>' \n  "$url/vms"
```

- From a snapshot. In this case the identifier of the snapshot has to be provided. For example, using a plain shell script and XML:

```
#!/bin/sh -ex
```

```
When creating a virtual machine from a template or from a snapshot it is usually useful to explicitly indicate in what storage domain to create the disks for the virtual machine. If the virtual machine is created from a template then this is achieved passing a set of `disk_attachment` elements that indicate the mapping:

```xml
<disk_attachments>
  <disk_attachment>
    <disk id="8d4bd566-6c86-4592-a4a7-912dbf93c298">
      <storage_domains>
        <storage_domain id="9cb6cb0a-cf1d-41c2-92ca-5a6d665649c9"/>
      </storage_domains>
    </disk>
  </disk_attachment>
</disk_attachments>
```

When the virtual machine is created from a snapshot this set of disks is slightly different, it uses the `image_id` attribute instead of `id`.

```xml
<disk_attachments>
  <disk_attachment>
    <disk>
      <image_id>8d4bd566-6c86-4592-a4a7-912dbf93c298</image_id>
      <storage_domains>
        <storage_domain id="9cb6cb0a-cf1d-41c2-92ca-5a6d665649c9"/>
      </storage_domains>
    </disk>
  </disk_attachment>
</disk_attachments>
```
It is possible to specify additional virtual machine parameters in the XML description, e.g. a virtual machine of \texttt{desktop} type, with 2 GiB of RAM and additional description can be added sending a request body like the following:

\begin{verbatim}
<vm>
  <name>myvm</name>
  <description>My Desktop Virtual Machine</description>
  <type>desktop</type>
  <memory>2147483648</memory>
  ...
</vm>
\end{verbatim}

A bootable CDROM device can be set like this:

\begin{verbatim}
<vm>
  ...
  <os>
    <boot dev="cdrom"/>
  </os>
</vm>
\end{verbatim}

In order to boot from CDROM, you first need to insert a disk, as described in the \texttt{CDROM service}. Then booting from that CDROM can be specified using the \texttt{os.boot.devices} attribute:

\begin{verbatim}
<vm>
  ...
  <os>
    <boot>
      <devices>
        <device>cdrom</device>
      </devices>
    </boot>
  </os>
</vm>
\end{verbatim}

In all cases the name or identifier of the cluster where the virtual machine will be created is mandatory.

\textbf{Table 6.825. Parameters summary}

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if the virtual machine should be independent of the template.</td>
</tr>
<tr>
<td>clone_permiss</td>
<td>Boolean</td>
<td>In</td>
<td>Specifies if the permissions of the template should be copied to the virtual machine.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>
6.262.1.1. clone

Specifies if the virtual machine should be independent of the template.

When a virtual machine is created from a template by default the disks of the virtual machine depend on the disks of the template, they are using the copy on write mechanism so that only the differences from the template take up real storage space. If this parameter is specified and the value is true then the disks of the created virtual machine will be cloned, and independent of the template. For example, to create an independent virtual machine, send a request like this:

```
POST /ovirt-engine/vms?clone=true
```

With a request body like this:

```
<vm>
  <name>myvm</name>
  <template>
    <name>mytemplate</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>
```

**NOTE**

When this parameter is true the permissions of the template will also be copied, as when using clone_permissions=true.

6.262.1.2. clone_permissions

Specifies if the permissions of the template should be copied to the virtual machine.

If this optional parameter is provided, and its values is true then the permissions of the template (only the direct ones, not the inherited ones) will be copied to the created virtual machine. For example, to create a virtual machine from the mytemplate template copying its permissions, send a request like this:

```
POST /ovirt-engine/api/vms?clone_permissions=true
```

With a request body like this:

```
<vm>
  <name>myvm</name>
  <template>
    <name>mytemplate</name>
  </template>
  <cluster>
    <name>mycluster</name>
  </cluster>
</vm>
```

6.262.2. list GET
Returns the list of virtual machines of the system.

The order of the returned list of virtual machines is guaranteed only if the sortby clause is included in the search parameter.

### Table 6.826. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_content</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if all the attributes of the virtual machines should be included in the response.</td>
</tr>
<tr>
<td>case_sensitive</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the search performed using the search parameter should be performed taking case into account.</td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>The maximum number of results to return.</td>
</tr>
<tr>
<td>search</td>
<td>String</td>
<td>In</td>
<td>A query string used to restrict the returned virtual machines.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

### 6.262.2.1. all_content

Indicates if all the attributes of the virtual machines should be included in the response.

By default the following attributes are excluded:

- console
- \texttt{initialization.configuration.data} - The OVF document describing the virtual machine.
- rng_source
- soundcard
- virtio_scsi

For example, to retrieve the complete representation of the virtual machines send a request like this:

```
GET /ovirt-engine/api/vms?all_content=true
```
NOTE

The reason for not including these attributes is performance: they are seldom used and they require additional queries to the database. So try to use the this parameter only when it is really needed.

6.262.2.2. case_sensitive

Indicates if the search performed using the search parameter should be performed taking case into account. The default value is true, which means that case is taken into account. If you want to search ignoring case set it to false.

6.262.2.3. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.263. VNICPROFILE

This service manages a vNIC profile.

Table 6.827. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td>Retrieves details about a vNIC profile.</td>
</tr>
<tr>
<td>remove</td>
<td>Removes the vNIC profile.</td>
</tr>
<tr>
<td>update</td>
<td>Updates details of a vNIC profile.</td>
</tr>
</tbody>
</table>

6.263.1. get GET

Retrieves details about a vNIC profile.

Table 6.828. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>profile</td>
<td>VnicProfile</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.263.1.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.263.2. remove DELETE
Removes the vNIC profile.

**Table 6.829. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

**6.263.3. update PUT**

Updates details of a vNIC profile.

**Table 6.830. Parameters summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the update should be performed asynchronously.</td>
</tr>
<tr>
<td>profile</td>
<td>VnicProfile</td>
<td>In/Out</td>
<td>The vNIC profile that is being updated.</td>
</tr>
</tbody>
</table>

**6.264. VNICPROFILES**

This service manages the collection of all vNIC profiles.

**Table 6.831. Methods summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a vNIC profile.</td>
</tr>
<tr>
<td>list</td>
<td>List all vNIC profiles.</td>
</tr>
</tbody>
</table>

**6.264.1. add POST**

Add a vNIC profile.

For example to add vNIC profile **123** to network **456** send a request to:

```
POST /ovirt-engine/api/networks/456/vnicprofiles
```

With the following body:

```
<vnic_profile id="123">
  <name>new_vNIC_name</name>
  <pass_through>
    <mode>disabled</mode>
  </pass_through>
</vnic_profile>
```
Please note that there is a default network filter to each VNIC profile. For more details of how the default network filter is calculated please refer to the documentation in NetworkFilters.

NOTE

The automatically created vNIC profile for the external network will be without network filter.

The output of creating a new VNIC profile depends in the body arguments that were given. In case no network filter was given, the default network filter will be configured. For example:

```xml
<vnic_profile href="/ovirt-engine/api/vnicprofiles/123" id="123">
    <name>new_vNIC_name</name>
    <link href="/ovirt-engine/api/vnicprofiles/123/permissions" rel="permissions"/>
    <pass_through/>
    <port_mirroring>false</port_mirroring>
    <network href="/ovirt-engine/api/networks/456" id="456"/>
    <network_filter href="/ovirt-engine/api/networkfilters/789" id="789"/>
</vnic_profile>
```

In case an empty network filter was given, no network filter will be configured for the specific VNIC profile regardless of the VNIC profile’s default network filter. For example:

```xml
<vnic_profile>
    <name>no_network_filter</name>
    <network_filter/>
</vnic_profile>
```

In case that a specific valid network filter id was given, the VNIC profile will be configured with the given network filter regardless of the VNIC profiles’s default network filter. For example:

```xml
<vnic_profile>
    <name>user_choice_network_filter</name>
    <network_filter id="0000001b-001b-001b-001b-0000000001d5"/>
</vnic_profile>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile</td>
<td>VnicProfile</td>
<td>In/Out</td>
<td>The vNIC profile that is being added.</td>
</tr>
</tbody>
</table>

**Table 6.832. Parameters summary**

**6.264.2. list GET**

List all vNIC profiles.
The order of the returned list of vNIC profiles isn’t guaranteed.

Table 6.833. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of profiles to return.</td>
</tr>
<tr>
<td>profiles</td>
<td>VnicProfile[]</td>
<td>Out</td>
<td>The list of all vNIC profiles.</td>
</tr>
</tbody>
</table>

6.264.2.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.264.2.2. max

Sets the maximum number of profiles to return. If not specified all the profiles are returned.

6.265. WEIGHT

Table 6.834. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>get</td>
<td></td>
</tr>
<tr>
<td>remove</td>
<td></td>
</tr>
</tbody>
</table>

6.265.1. get GET

Table 6.835. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be <em>followed</em>.</td>
</tr>
<tr>
<td>weight</td>
<td>Weight</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>

6.265.1.1. follow

Indicates which inner links should be *followed*. The objects referenced by these links will be fetched as part of the current request. See here for details.
6.265.2. remove DELETE

Table 6.836. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>async</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the remove should be performed asynchronously.</td>
</tr>
</tbody>
</table>

6.266. WEIGHTS

Table 6.837. Methods summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add a weight to a specified user defined scheduling policy.</td>
</tr>
<tr>
<td>list</td>
<td>Returns the list of weights.</td>
</tr>
</tbody>
</table>

6.266.1. add POST

Add a weight to a specified user defined scheduling policy.

Table 6.838. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight</td>
<td>Weight</td>
<td>In/Out</td>
<td></td>
</tr>
</tbody>
</table>

6.266.2. list GET

Returns the list of weights.

The order of the returned list of weights isn’t guaranteed.

Table 6.839. Parameters summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Direction</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>Boolean</td>
<td>In</td>
<td>Indicates if the results should be filtered according to the permissions of the user.</td>
</tr>
<tr>
<td>follow</td>
<td>String</td>
<td>In</td>
<td>Indicates which inner links should be followed.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>In</td>
<td>Sets the maximum number of weights to return.</td>
</tr>
<tr>
<td>weights</td>
<td>Weight[]</td>
<td>Out</td>
<td></td>
</tr>
</tbody>
</table>
6.266.2.1. follow

Indicates which inner links should be followed. The objects referenced by these links will be fetched as part of the current request. See here for details.

6.266.2.2. max

Sets the maximum number of weights to return. If not specified all the weights are returned.
CHAPTER 7. TYPES

This section enumerates all the data types that are available in the API.

7.1. ACCESSPROTOCOL ENUM

Represents the access protocols supported by Gluster volumes. `gluster` and `nfs` are enabled by default.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cifs</td>
<td>CIFS access protocol.</td>
</tr>
<tr>
<td>gluster</td>
<td>Gluster access protocol.</td>
</tr>
<tr>
<td>nfs</td>
<td>NFS access protocol.</td>
</tr>
</tbody>
</table>

7.2. ACTION STRUCT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_partial_import</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>async</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>attachment</td>
<td>DiskAttachment</td>
<td></td>
</tr>
<tr>
<td>authorized_key</td>
<td>AuthorizedKey</td>
<td></td>
</tr>
<tr>
<td>bricks</td>
<td>GlusterBrick[]</td>
<td></td>
</tr>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td></td>
</tr>
<tr>
<td>check_connectivity</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>clone</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>clone_permissions</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>collapse_snapshots</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>connection</td>
<td>StorageConnection</td>
<td></td>
</tr>
<tr>
<td>connectivity_timeout</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td></td>
</tr>
<tr>
<td>deploy_hosted_engine</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>details</td>
<td>GlusterVolumeProfileDetails</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>discard_snapshots</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>discovered_targets</td>
<td>IscsiDetails[]</td>
<td></td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td></td>
</tr>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
<td></td>
</tr>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td></td>
</tr>
<tr>
<td>exclusive</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>fault</td>
<td>Fault</td>
<td></td>
</tr>
<tr>
<td>fence_type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>filename</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>filter</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>fix_layout</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>force</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>grace_period</td>
<td>GracePeriod</td>
<td></td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>image</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>image_transfer</td>
<td>ImageTransfer</td>
<td></td>
</tr>
<tr>
<td>import_as_template</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>is_attached</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>iscsi</td>
<td>IscsiDetails</td>
<td></td>
</tr>
<tr>
<td>iscsi_targets</td>
<td>String[]</td>
<td></td>
</tr>
<tr>
<td>job</td>
<td>Job</td>
<td></td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td></td>
</tr>
<tr>
<td>logical_units</td>
<td>LogicalUnit[]</td>
<td></td>
</tr>
<tr>
<td>maintenance_enabled</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>modified_bonds</td>
<td>HostNic[]</td>
<td></td>
</tr>
<tr>
<td>modified_labels</td>
<td>NetworkLabel[]</td>
<td></td>
</tr>
<tr>
<td>modified_network_attachments</td>
<td>NetworkAttachment[]</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>option</td>
<td>Option</td>
<td></td>
</tr>
<tr>
<td>pause</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>permission</td>
<td>Permission</td>
<td></td>
</tr>
<tr>
<td>power_management</td>
<td>PowerManagement</td>
<td></td>
</tr>
<tr>
<td>proxy_ticket</td>
<td>ProxyTicket</td>
<td></td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td></td>
</tr>
<tr>
<td>reason</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>reassign_bad_macs</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>reboot</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>registration_configuration</td>
<td>RegistrationConfiguration</td>
<td></td>
</tr>
<tr>
<td>remote_viewer_connection_file</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>removed_bonds</td>
<td>HostNic[]</td>
<td></td>
</tr>
<tr>
<td>removed_labels</td>
<td>NetworkLabel[]</td>
<td></td>
</tr>
<tr>
<td>removed_network_attatchments</td>
<td>NetworkAttachment[]</td>
<td></td>
</tr>
<tr>
<td>resolution_type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>restore_memory</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>root_password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>seal</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td></td>
</tr>
<tr>
<td>ssh</td>
<td>Ssh</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>stop_cluster_service</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td></td>
</tr>
<tr>
<td>succeeded</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>synchronized_network_attachments</td>
<td>NetworkAttachment[]</td>
<td></td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td></td>
</tr>
<tr>
<td>ticket</td>
<td>Ticket</td>
<td></td>
</tr>
<tr>
<td>undeploy_hosted_engine</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>use_cloud_init</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>use_sysprep</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>virtual_functions_config</td>
<td>HostNicVirtualFunctionsConfiguration</td>
<td></td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td></td>
</tr>
<tr>
<td>vnic_profile_mappings</td>
<td>VnicProfileMapping[]</td>
<td></td>
</tr>
<tr>
<td>volatile</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>
7.3. AFFINITYGROUP STRUCT

An affinity group represents a group of virtual machines with a defined relationship.

Table 7.3. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>enforcing</td>
<td>Boolean</td>
<td>Specifies whether the affinity group uses hard or soft enforcement of the affinity applied to virtual machines that are members of that affinity group.</td>
</tr>
<tr>
<td>hosts_rule</td>
<td>AffinityRule</td>
<td>Specifies the affinity rule applied between virtual machines and hosts that are members of this affinity group.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>positive</td>
<td>Boolean</td>
<td>Specifies whether the affinity group applies positive affinity or negative affinity to virtual machines that are members of that affinity group.</td>
</tr>
<tr>
<td>vms_rule</td>
<td>AffinityRule</td>
<td>Specifies the affinity rule applied to virtual machines that are members of this affinity group.</td>
</tr>
</tbody>
</table>

7.3.1. enforcing

Specifies whether the affinity group uses hard or soft enforcement of the affinity applied to virtual machines that are members of that affinity group.

WARNING

Please note that this attribute has been deprecated since version 4.1 of the engine, and will be removed in the future. Use the vms_rule attribute from now on.

7.3.2. positive

Specifies whether the affinity group applies positive affinity or negative affinity to virtual machines that are members of that affinity group.
WARNING
Please note that this attribute has been deprecated since version 4.1 of the engine, and will be removed in the future. Use the vms_rule attribute from now on.

Table 7.4. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>A reference to the cluster to which the affinity group applies.</td>
</tr>
<tr>
<td>hosts</td>
<td>Host[]</td>
<td>A list of all hosts assigned to this affinity group.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>A list of all virtual machines assigned to this affinity group.</td>
</tr>
</tbody>
</table>

7.4. AFFINITYLABEL STRUCT

The affinity label can influence virtual machine scheduling. It is most frequently used to create a sub-cluster from the available hosts.

Table 7.5. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>read_only</td>
<td>Boolean</td>
<td>The read_only property marks a label that can not be modified.</td>
</tr>
</tbody>
</table>

7.4.1. read_only

The read_only property marks a label that can not be modified. This is usually the case when listing internally-generated labels.

Table 7.6. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts</td>
<td>Host[]</td>
<td>A list of hosts that were labeled using this scheduling label.</td>
</tr>
</tbody>
</table>
7.5. AFFINITYRULE STRUCT

Generic rule definition for affinity group. Each supported resource type (virtual machine, host) is controlled by a separate rule. This allows expressing of rules like: no affinity between defined virtual machines, but hard affinity between defined virtual machines and virtual hosts.

Table 7.7. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>A list of virtual machines that were labeled using this scheduling label.</td>
</tr>
</tbody>
</table>

7.5.1. enabled

Specifies whether the affinity group uses this rule or not. This attribute is optional during creation and is considered to be true when it is not provided. In case this attribute is not provided to the update operation, it is considered to be true if AffinityGroup positive attribute is set as well. The backend enabled value will be preserved when both enabled and positive attributes are missing.

7.5.2. enforcing

Specifies whether the affinity group uses hard or soft enforcement of the affinity applied to the resources that are controlled by this rule. This argument is mandatory if the rule is enabled and is ignored when the rule is disabled.

7.5.3. positive

Specifies whether the affinity group applies positive affinity or negative affinity to the resources that are controlled by this rule. This argument is mandatory if the rule is enabled and is ignored when the rule is disabled.

7.6. AGENT STRUCT

Type representing a fence agent.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>Fence agent address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>concurrent</td>
<td>Boolean</td>
<td>Specifies whether the agent should be used concurrently or sequentially</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>encrypt_options</td>
<td>Boolean</td>
<td>Specifies whether the options should be encrypted.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>options</td>
<td>Option[]</td>
<td>Fence agent options (comma-delimited list of key-value pairs).</td>
</tr>
<tr>
<td>order</td>
<td>Integer</td>
<td>The order of this agent if used with other agents.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Fence agent password.</td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td>Fence agent port.</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Fence agent type.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Fence agent user name.</td>
</tr>
</tbody>
</table>

Table 7.9. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host service.</td>
</tr>
</tbody>
</table>

7.6.1. host

Reference to the host service. Each fence agent belongs to a single host.

7.7. AGENTCONFIGURATION STRUCT

Table 7.10. Attributes summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>broker_type</td>
<td>MessageBrokerType</td>
<td></td>
</tr>
<tr>
<td>network_mappings</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.8. API STRUCT

This type contains the information returned by the root service of the API.

To get that information send a request like this:

```
GET /ovirt-engine/api
```

The result will be like this:

```xml
<api>
    <link rel="hosts" href="/ovirt-engine/api/hosts"/>
    <link rel="vms" href="/ovirt-engine/api/vms"/>
    ...
    <product_info>
        <name>oVirt Engine</name>
        <vendor>ovirt.org</vendor>
        <version>
            <build>0</build>
            <full_version>4.1.0_master</full_version>
            <major>4</major>
            <minor>1</minor>
            <revision>0</revision>
        </version>
    </product_info>
    <special_objects>
        <link rel="templates/blank" href="..."/>
        <link rel="tags/root" href="..."/>
    </special_objects>
    <summary>
        <vms>
            <total>10</total>
            <active>3</active>
        </vms>
        <hosts>
            <total>2</total>
        </hosts>
    </summary>
</api>
```
Table 7.11. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>product_info</td>
<td>ProductInfo</td>
<td>Information about the product, such as its name, the name of the vendor, and the version.</td>
</tr>
<tr>
<td>special_objects</td>
<td>SpecialObjects</td>
<td>References to special objects, such as the blank template and the root of the hierarchy of tags.</td>
</tr>
<tr>
<td>summary</td>
<td>ApiSummary</td>
<td>A summary containing the total number of relevant objects, such as virtual machines, hosts, and storage domains.</td>
</tr>
<tr>
<td>time</td>
<td>Date</td>
<td>The date and time when this information was generated.</td>
</tr>
</tbody>
</table>

Table 7.12. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authenticate_user</td>
<td>User</td>
<td>Reference to the authenticated user.</td>
</tr>
<tr>
<td>effective_user</td>
<td>User</td>
<td>Reference to the effective user.</td>
</tr>
</tbody>
</table>

7.8.1. authenticated_user

Reference to the authenticated user.

The authenticated user is the user whose credentials were verified in order to accept the current request. In the current version of the system the authenticated user and the effective user are always the same. In the future, when support for user impersonation is introduced, they will be potentially different.

7.8.2. effective_user

Reference to the effective user.
The effective user is the user whose permissions apply during the current request. In the current version of the system the authenticated user and the effective user are always the same. In the future, when support for user impersonation is introduced, they will be potentially different.

### 7.9. APISUMMARY STRUCT

A summary containing the total number of relevant objects, such as virtual machines, hosts, and storage domains.

#### Table 7.13. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts</td>
<td>ApiSummaryItem</td>
<td>The summary of hosts.</td>
</tr>
<tr>
<td>storage_domains</td>
<td>ApiSummaryItem</td>
<td>The summary of storage domains.</td>
</tr>
<tr>
<td>users</td>
<td>ApiSummaryItem</td>
<td>The summary of users.</td>
</tr>
<tr>
<td>vms</td>
<td>ApiSummaryItem</td>
<td>The summary of virtual machines.</td>
</tr>
</tbody>
</table>

### 7.10. APISUMMARYITEM STRUCT

This type contains an item of the API summary. Each item contains the total and active number of some kind of object.

#### Table 7.14. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Integer</td>
<td>The total number of active objects.</td>
</tr>
<tr>
<td>total</td>
<td>Integer</td>
<td>The total number of objects.</td>
</tr>
</tbody>
</table>

### 7.11. APPLICATION STRUCT

Represents an application installed on a virtual machine. Applications are reported by the guest agent, if you deploy one on the virtual machine operating system.

To get that information send a request like this:

```
GET /ovirt-engine/api/vms/123/applications/456
```

The result will be like this:

```
<application href="/ovirt-engine/api/vms/123/applications/456" id="456">
  <name>application-test-1.0.0-0.el7</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
</application>
```
### Table 7.15. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### Table 7.16. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm</td>
<td>Vm</td>
<td>A reference to the virtual machine the application is installed on.</td>
</tr>
</tbody>
</table>

### 7.12. ARCHITECTURE ENUM

#### Table 7.17. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ppc64</td>
<td></td>
</tr>
<tr>
<td>s390x</td>
<td>IBM S390X CPU architecture.</td>
</tr>
<tr>
<td>undefined</td>
<td></td>
</tr>
<tr>
<td>x86_64</td>
<td></td>
</tr>
</tbody>
</table>

#### 7.12.1. s390x

IBM S390X CPU architecture.

Needs to be specified for virtual machines and clusters running on the S390X architecture.

Note that S390 is often used in an ambiguous way to describe either the general machine architecture as such or its 31-bit variant. S390X is used specifically for the 64-bit architecture, which is in line with the other architectures, like X86_64 or PPC64.

### 7.13. AUTHORIZEDKEY STRUCT

#### Table 7.18. Attributes summary
### 7.14. AUTONUMASTATUS ENUM

#### Table 7.20. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disable</td>
<td></td>
</tr>
<tr>
<td>enable</td>
<td></td>
</tr>
<tr>
<td>unknown</td>
<td></td>
</tr>
</tbody>
</table>

### 7.15. BALANCE STRUCT

#### Table 7.21. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>
### 7.16. BIOS STRUCT

Table 7.23. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_menu</td>
<td>BootMenu</td>
<td></td>
</tr>
</tbody>
</table>

### 7.17. BLOCKSTATISTIC STRUCT

Table 7.24. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>statistics</td>
<td>Statistic[</td>
<td></td>
</tr>
</tbody>
</table>

### 7.18. BONDING STRUCT

Represents a network interfaces bond.

Table 7.25. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad_partner_mac</td>
<td>Mac</td>
<td>The ad_partner_mac property of the partner bond in mode 4.</td>
</tr>
<tr>
<td>options</td>
<td>Option[]</td>
<td>A list of option elements for a bonded interface.</td>
</tr>
<tr>
<td>slaves</td>
<td>HostNic[]</td>
<td>A list of slave NICs for a bonded interface.</td>
</tr>
</tbody>
</table>

#### 7.18.1. ad_partner_mac

The ad_partner_mac property of the partner bond in mode 4. Bond mode 4 is the 802.3ad standard, which is also called dynamic link aggregation. See Wikipedia and Presentation for more information. ad_partner_mac is the MAC address of the system (switch) at the other end of a bond. This parameter is read-only. Setting it will have no effect on the bond. It is retrieved from /sys/class/net/bondX/bonding/ad_partner_mac file on the system where the bond is located.
7.18.2. options
A list of option elements for a bonded interface. Each option contains property name and value attributes. Only required when adding bonded interfaces.

7.18.3. slaves
A list of slave NICs for a bonded interface. Only required when adding bonded interfaces.

Table 7.26. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active_slave</td>
<td>HostNic</td>
<td>The active_slave property of the bond in modes that support it (active-backup, balance-alb and balance-tlb).</td>
</tr>
</tbody>
</table>

7.18.4. active_slave
The active_slave property of the bond in modes that support it (active-backup, balance-alb and balance-tlb). See Linux documentation for further details. This parameter is read-only. Setting it will have no effect on the bond. It is retrieved from /sys/class/net/bondX/bonding/active_slave file on the system where the bond is located.

For example:

```
GET /ovirt-engine/api/hosts/123/nics/321
```

Will respond:

```
<host_nic href="/ovirt-engine/api/hosts/123/nics/321" id="321">
  ...
  <bonding>
    <slaves>
      <host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456" />
      ...                  
    </slaves>
    <active_slave href="/ovirt-engine/api/hosts/123/nics/456" id="456" />
  </bonding>
  ...
</host_nic>
```

7.19. BOOKMARK STRUCT
Represents a bookmark in the system.

Table 7.27. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
</tbody>
</table>
7.20. **BOOT STRUCT**

Configuration of the boot sequence of a virtual machine.

Table 7.28. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>devices</td>
<td>BootDevice[]</td>
<td>Ordered list of boot devices.</td>
</tr>
</tbody>
</table>

7.20.1. **devices**

Ordered list of boot devices. The virtual machine will try to boot from the given boot devices, in the given order.

7.21. **BOOTDEVICE ENUM**

Represents the kinds of devices that a virtual machine can boot from.

Table 7.29. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td>Boot from CD-ROM.</td>
</tr>
<tr>
<td>hd</td>
<td>Boot from the hard drive.</td>
</tr>
<tr>
<td>network</td>
<td>Boot from the network, using PXE.</td>
</tr>
</tbody>
</table>

7.21.1. **cdrom**

Boot from CD-ROM. The CD-ROM can be chosen from the list of ISO files available in an ISO domain attached to the ata center that the virtual machine belongs to.

7.21.2. **network**

Boot from the network, using PXE. It is necessary to have PXE configured on the network that the virtual machine is connected to.
7.22. BOOTMENU STRUCT

Represents boot menu configuration for virtual machines and templates.

Table 7.30. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Whether the boot menu is enabled for this virtual machine (or template), or not.</td>
</tr>
</tbody>
</table>

7.23. BOOTPROTOCOL ENUM

Defines the options of the IP address assignment method to a NIC.

Table 7.31. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoconf</td>
<td>Stateless address auto-configuration.</td>
</tr>
<tr>
<td>dhcp</td>
<td>Dynamic host configuration protocol.</td>
</tr>
<tr>
<td>none</td>
<td>No address configuration.</td>
</tr>
<tr>
<td>static</td>
<td>Statically-defined address, mask and gateway.</td>
</tr>
<tr>
<td>poly_dhcp_autoconf</td>
<td>DHCP alongside Stateless address auto-configuration (SLAAC)</td>
</tr>
<tr>
<td></td>
<td>The SLAAC mechanism is defined by <a href="http://tools">http://tools</a>.</td>
</tr>
</tbody>
</table>

7.23.1. autoconf

Stateless address auto-configuration.

The mechanism is defined by RFC 4862. Please refer to [this wikipedia article](https://en.wikipedia.org/wiki/Stateless_address_auto-configuration) for more information.

**NOTE**

The value is valid for IPv6 addresses only.

7.23.2. dhcp

Dynamic host configuration protocol.

Please refer to [this wikipedia article](https://en.wikipedia.org/wiki/Dynamic_host_configuration_protocol) for more information.

7.23.3. poly_dhcp_autoconf

DHCP alongside Stateless address auto-configuration (SLAAC)
The SLAAC mechanism is defined by RFC 4862. Please refer to the Stateless address auto-configuration article and the DHCP article for more information.

NOTE
The value is valid for IPv6 addresses only.

### 7.24. BRICKPROFILEDETAIL STRUCT

**Table 7.32. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile_details</td>
<td>ProfileDetail[]</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.33. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>brick</td>
<td>GlusterBrick</td>
<td></td>
</tr>
</tbody>
</table>

### 7.25. CDROM STRUCT

**Table 7.34. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>file</td>
<td>File</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

**Table 7.35. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
</tbody>
</table>
### 7.25.1. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.26. CERTIFICATE STRUCT

Table 7.36. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>organization</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>subject</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.27. CLOUDINIT STRUCT

Deprecated type to specify cloud-init configuration.

This type has been deprecated and replaced by alternative attributes inside the Initialization type. See the cloud_init attribute documentation for details.

Table 7.37. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authorized_keys</td>
<td>AuthorizedKey[]</td>
<td></td>
</tr>
<tr>
<td>files</td>
<td>File[]</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>network_configuration</td>
<td>NetworkConfiguration</td>
<td></td>
</tr>
<tr>
<td>regenerate_ssh_keys</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>timezone</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>users</td>
<td>User[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.28. CLUSTER STRUCT

Type representation of a cluster.

A JSON representation of a cluster:

```json
{
  "cluster" : [ {
    "ballooning_enabled" : "false",
    "cpu" : { 
      "architecture" : "x86_64",
      "type" : "Intel SandyBridge Family"
    },
    "custom_scheduling_policy_properties" : { 
      "property" : [ {
        "name" : "HighUtilization",
        "value" : "80"
      }, {
        "name" : "CpuOverCommitDurationMinutes",
        "value" : "2"
      } ]
    },
    "error_handling" : { 
      "on_error" : "migrate"
    },
    "fencing_policy" : { 
      "enabled" : "true",
      "skip_if_connectivity_broken" : { 
        "enabled" : "false",
        "threshold" : "50"
      },
      "skip_if_gluster_bricks_up" : "false",
      "skip_if_gluster_quorum_not_met" : "false",
      "skip_if_sd_active" : { 
        "enabled" : "false"
      }
    },
    "gluster_service" : "false",
  }
}
"firewall_type" : "iptables",
"ha_reservation" : "false",
"ksm" : {
   "enabled" : "true",
   "merge_across_nodes" : "true"
},
"maintenance_reason_required" : "false",
"memory_policy" : {
   "over_commit" : {
      "percent" : "100"
   },
   "transparent_hugepages" : {
      "enabled" : "true"
   }
},
"migration" : {
   "auto_converge" : "inherit",
   "bandwidth" : {
      "assignment_method" : "auto"
   },
   "compressed" : "inherit",
   "policy" : {
      "id" : "00000000-0000-0000-0000-000000000000"
   }
},
"optional_reason" : "false",
"required_rng_sources" : {
   "required_rng_source" : [ "random" ]
},
"switch_type" : "legacy",
"threads_as_cores" : "false",
"trusted_service" : "false",
"tunnel_migration" : "false",
"version" : {
   "major" : "4",
   "minor" : "1"
},
"virt_service" : "true",
"data_center" : {
   "href" : "/ovirt-engine/api/datacenters/123",
   "id" : "123"
},
"mac_pool" : {
   "href" : "/ovirt-engine/api/macpools/456",
   "id" : "456"
},
"scheduling_policy" : {
   "href" : "/ovirt-engine/api/schedulingpolicies/789",
   "id" : "789"
},
"actions" : {
   "link" : [ {
      "href" : "/ovirt-engine/api/clusters/234/resetemulatedmachine",
      "rel" : "resetemulatedmachine"
   } ]
}
Table 7.38. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ballooning_enabled</td>
<td>Boolean</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>Custom scheduling policy properties of the cluster.</td>
</tr>
<tr>
<td>custom_scheduling_policy_properties</td>
<td>Property[]</td>
<td>Custom scheduling policy properties of the cluster.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>error_handling</td>
<td>ErrorHandling</td>
<td></td>
</tr>
<tr>
<td>fencing_policy</td>
<td>FencingPolicy</td>
<td>A custom fencing policy can be defined for a cluster.</td>
</tr>
<tr>
<td>firewall_type</td>
<td>FirewallType</td>
<td>The type of firewall to be used on hosts in this cluster.</td>
</tr>
<tr>
<td>gluster_service</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>gluster_tuned_profile</td>
<td>String</td>
<td>The name of the <a href="https://fedorahosted">https://fedorahosted</a>.</td>
</tr>
<tr>
<td>ha_reservation</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ksm</td>
<td>Ksm</td>
<td></td>
</tr>
<tr>
<td>maintenance_reason_required</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td></td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>optional_reason</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>required_rng_sources</td>
<td>RngSource[]</td>
<td>Set of random number generator (RNG) sources required from each host in the cluster.</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td></td>
</tr>
<tr>
<td>supported_versions</td>
<td>Version[]</td>
<td></td>
</tr>
<tr>
<td>switch_type</td>
<td>SwitchType</td>
<td>The type of switch to be used by all networks in given cluster.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>threads_as_cores</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>trusted_service</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>tunnel_migration</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>Version</td>
<td>The compatibility version of the cluster.</td>
</tr>
<tr>
<td>virt_service</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>

### 7.28.1. custom_scheduling_policy_properties

Custom scheduling policy properties of the cluster. These optional properties override the properties of the scheduling policy specified by the `scheduling_policy` link, and apply only for this specific cluster.

For example, to update the custom properties of the cluster, send a request:

```plaintext
PUT /ovirt-engine/api/clusters/123
```

With a request body:

```xml
<cluster>
  <custom_scheduling_policy_properties>
    <property>
      <name>HighUtilization</name>
      <value>70</value>
    </property>
  </custom_scheduling_policy_properties>
</cluster>
```

Update operations using the `custom_scheduling_policy_properties` attribute will not update the properties of the scheduling policy specified by the `scheduling_policy` link, they will only be reflected on this specific cluster.

### 7.28.2. fencing_policy

A custom fencing policy can be defined for a cluster.

For example:

```plaintext
PUT /ovirt-engine/api/cluster/123
```

With request body like this:

```xml
<cluster>
</cluster>
```
7.28.3. firewall_type

The type of firewall to be used on hosts in this cluster.

Up to version 4.1, it was always **iptables**. Since version 4.2, you can choose between **iptables** and **firewalld**. For clusters with a compatibility version of 4.2 and higher, the default firewall type is **firewalld**.

7.28.4. gluster_tuned_profile

The name of the **tuned** profile to set on all the hosts in the cluster. This is not mandatory and relevant only for clusters with Gluster service.

7.28.5. required_rng_sources

Set of random number generator (RNG) sources required from each host in the cluster.

When read, it returns the implicit **urandom** (for cluster version 4.1 and higher) or **random** (for cluster version 4.0 and lower) plus additional selected RNG sources. When written, the implicit **urandom** and **random** RNG sources cannot be removed.

**IMPORTANT**

Before version 4.1 of the engine, the set of required random number generators was completely controllable by the administrator; any source could be added or removed, including the **random** source. But starting with version 4.1, the **urandom** and **random** sources will always be part of the set, and can’t be removed.

**IMPORTANT**

Engine version 4.1 introduces a new RNG source **urandom** that replaces **random** RNG source in clusters with compatibility version 4.1 or higher.

7.28.6. version

The compatibility version of the cluster.

All hosts in this cluster must support at least this compatibility version.

For example:

```xml
<cluster>
  <fencing_policy>
    <enabled>true</enabled>
    <skip_if_sd_active>
      <enabled>false</enabled>
    </skip_if_sd_active>
    <skip_if_connectivity_broken>
      <enabled>false</enabled>
      <threshold>50</threshold>
    </skip_if_connectivity_broken>
  </fencing_policy>
</cluster>
```
GET /ovirt-engine/api/clusters/123

Will respond with:

```xml
<cluster>
  ...
  <version>
    <major>4</major>
    <minor>0</minor>
  </version>
  ...
</cluster>
```

To update the compatibility version, use:

PUT /ovirt-engine/api/clusters/123

With a request body like this:

```xml
<cluster>
  <version>
    <major>4</major>
    <minor>1</minor>
  </version>
</cluster>
```

In order to update the cluster compatibility version, all hosts in the cluster must support the new compatibility version.

**Table 7.39. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>affinity_groups</code></td>
<td>AffinityGroup[]</td>
<td></td>
</tr>
<tr>
<td><code>cpu_profiles</code></td>
<td>CpuProfile[]</td>
<td></td>
</tr>
<tr>
<td><code>data_center</code></td>
<td>DataCenter</td>
<td></td>
</tr>
<tr>
<td><code>enabled_features</code></td>
<td>ClusterFeature[]</td>
<td>Custom features that are enabled for the cluster.</td>
</tr>
<tr>
<td><code>external_network_providers</code></td>
<td>ExternalProvider[]</td>
<td>A reference to the external network provider available in the cluster.</td>
</tr>
<tr>
<td><code>cluster_hooks</code></td>
<td>ClusterHook[]</td>
<td></td>
</tr>
</tbody>
</table>
### 7.28.7. `external_network_providers`

A reference to the external network provider available in the cluster.

If the automatic deployment of the external network provider is supported, the networks of the referenced network provider are available on every host in the cluster. External network providers of a cluster can only be set during adding the cluster. This value may be overwritten for individual hosts during adding the host.

### 7.28.8. `scheduling_policy`

Reference to the default scheduling policy used by this cluster.

**NOTE**

The scheduling policy properties are taken by default from the referenced scheduling policy, but they are overridden by the properties specified in the `custom_scheduling_policy_properties` attribute for this cluster.

### 7.29. CLUSTERFEATURE STRUCT

Type represents an additional feature that is available at a cluster level.

Table 7.40. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>comment</code></td>
<td><code>String</code></td>
<td>Free text containing comments about this object.</td>
</tr>
</tbody>
</table>
### Table 7.41. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_level</td>
<td>ClusterLevel</td>
<td>Reference to the cluster level.</td>
</tr>
</tbody>
</table>

### 7.30. CLUSTERLEVEL STRUCT

Describes the capabilities supported by a specific cluster level.

### Table 7.42. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>cpu_types</td>
<td>CpuType[]</td>
<td>The CPU types supported by this cluster level.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>permits</td>
<td>Permit[]</td>
<td>The permits supported by this cluster level.</td>
</tr>
</tbody>
</table>

### Table 7.43. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_features</td>
<td>ClusterFeature[]</td>
<td>The additional features supported by this cluster level.</td>
</tr>
</tbody>
</table>

### 7.31. CONFIGURATION STRUCT

### Table 7.44. Attributes summary
### 7.31.1. data

The document describing the virtual machine.

Example of the OVF document:

```xml
<?xml version='1.0' encoding='UTF-8'?>
<ovf:Envelope xmlns:ovf="http://schemas.dmtf.org/ovf/envelope/1/
   xmlns:rasd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_ResourceAllocationSettingData"
   xmlns:vssd="http://schemas.dmtf.org/wbem/wscim/1/cim-schema/2/CIM_VirtualSystemSettingData"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   ovf:version="3.5.0.0">
   <References/>
   <Section xsi:type="ovf:NetworkSection_Type">
     <Info>List of networks</Info>
     <Network ovf:name="Network 1"/>
   </Section>
   <Section xsi:type="ovf:DiskSection_Type">
     <Info>List of Virtual Disks</Info>
   </Section>
   <Content ovf:id="out" xsi:type="ovf:VirtualSystem_Type">
     <CreationDate>2014/12/03 04:25:45</CreationDate>
     <ExportDate>2015/02/09 14:12:24</ExportDate>
     <DeleteProtected>false</DeleteProtected>
     <SsoMethod>guest_agent</SsoMethod>
     <IsSmartcardEnabled>false</IsSmartcardEnabled>
     <TimeZone>Etc/GMT</TimeZone>
     <default_boot_sequence>0</default_boot_sequence>
     <Generation>1</Generation>
     <VmType>1</VmType>
     <MinAllocatedMem>1024</MinAllocatedMem>
     <IsStateless>false</IsStateless>
     <IsRunAndPause>false</IsRunAndPause>
     <AutoStartup>false</AutoStartup>
     <Priority>1</Priority>
     <CreatedByUserId>fdfc627c-d875-11e0-90f0-83df133b58cc</CreatedByUserId>
     <IsBootMenuEnabled>false</IsBootMenuEnabled>
     <IsSpiceFileTransferEnabled>true</IsSpiceFileTransferEnabled>
     <IsSpiceCopyPasteEnabled>true</IsSpiceCopyPasteEnabled>
     <Name>VM_export</Name>
     <TemplateId>00000000-0000-0000-0000-000000000000</TemplateId>
     <TemplateName>Blank</TemplateName>
     <IsInitialized>false</IsInitialized>
     <Origin>3</Origin>
     <DefaultDisplayType>1</DefaultDisplayType>
   </Content>
</ovf:Envelope>
```
### 7.32. CONFIGURATIONTYPE ENUM

Configuration format types.

**Table 7.45. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ova</td>
<td>ConfigurationType of type standard OVF.</td>
</tr>
<tr>
<td>ovf</td>
<td>ConfigurationType of type oVirt-compatible OVF.</td>
</tr>
</tbody>
</table>
7.32.1. ova

ConfigurationType of type standard OVF.

The provided virtual machine configuration conforms with the Open Virtualization Format (OVF) standard. This value should be used for an OVF configuration that is extracted from an Open Virtual Appliance (OVA) that was generated by oVirt or by other vendors. See here for the OVF specification.

7.32.2. ovf

ConfigurationType of type oVirt-compatible OVF.

The provided virtual machine configuration conforms with the oVirt-compatible form of the Open Virtualization Format (OVF). Note that the oVirt-compatible form of the OVF may differ from the OVF standard that is used by other vendors. This value should be used for an OVF configuration that is taken from a storage domain.

7.33. CONSOLE STRUCT

Representation for serial console device.

Table 7.46. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Enable/disable the serial console device.</td>
</tr>
</tbody>
</table>

7.34. CORE STRUCT

Table 7.47. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>socket</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

7.35. CPU STRUCT

Table 7.48. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>Architecture</td>
<td></td>
</tr>
<tr>
<td>cores</td>
<td>Core[]</td>
<td></td>
</tr>
<tr>
<td>cpu_tune</td>
<td>CpuTune</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>level</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>mode</td>
<td>CpuMode</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>speed</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>topology</td>
<td>CpuTopology</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

**7.36. CPUMODE ENUM**

Table 7.49. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom</td>
<td></td>
</tr>
<tr>
<td>host_model</td>
<td></td>
</tr>
<tr>
<td>host_passthrough</td>
<td></td>
</tr>
</tbody>
</table>

**7.37. CPUPROFILE STRUCT**

Table 7.50. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

Table 7.51. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td></td>
</tr>
</tbody>
</table>

### 7.38. CPUTOPOLOGY STRUCT

Table 7.52. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cores</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>sockets</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>threads</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

### 7.39. CPUTUNE STRUCT

Table 7.53. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>vcpu_pins</td>
<td>VcpuPin[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.40. CPUTYPE STRUCT

Describes a supported CPU type.

Table 7.54. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>Architecture</td>
<td>The architecture of the CPU.</td>
</tr>
<tr>
<td>level</td>
<td>Integer</td>
<td>The level of the CPU type.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the CPU type, for example Intel Conroe Family.</td>
</tr>
</tbody>
</table>

### 7.41. CREATIONSTATUS ENUM

Table 7.55. Values summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>complete</td>
<td></td>
</tr>
<tr>
<td>failed</td>
<td></td>
</tr>
<tr>
<td>in_progress</td>
<td></td>
</tr>
<tr>
<td>pending</td>
<td></td>
</tr>
</tbody>
</table>

### 7.42. CUSTOMPROPERTY STRUCT

Custom property representation.

#### Table 7.56. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Property name.</td>
</tr>
<tr>
<td>regexp</td>
<td>String</td>
<td>A regular expression defining the available values a custom property can get.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>Property value.</td>
</tr>
</tbody>
</table>

### 7.43. DATACENTER STRUCT

#### Table 7.57. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>local</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>quota_mode</td>
<td>QuotaModeType</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>DataCenterStatus</td>
<td></td>
</tr>
<tr>
<td>storage_form</td>
<td>StorageFormat</td>
<td></td>
</tr>
</tbody>
</table>
supported_versions | Version[] |
|------------------|---------|
| version          | Version | The compatibility version of the data center.

### 7.43.1. version

The compatibility version of the data center.

All clusters in this data center must already be set to at least this compatibility version.

For example:

```
GET /ovirt-engine/api/datacenters/123
```

Will respond:

```
<data_center>
  ...
  <version>
    <major>4</major>
    <minor>0</minor>
  </version>
  ...
</data_center>
```

To update the compatibility version, use:

```
PUT /ovirt-engine/api/datacenters/123
```

With a request body:

```
<data_center>
  <version>
    <major>4</major>
    <minor>1</minor>
  </version>
</data_center>
```

### Table 7.58. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>clusters</td>
<td>Cluster[]</td>
<td>Reference to clusters inside this data center.</td>
</tr>
</tbody>
</table>
## 7.44. DATACENTERSTATUS ENUM

### Table 7.59. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>contend</td>
<td></td>
</tr>
<tr>
<td>maintenance</td>
<td></td>
</tr>
<tr>
<td>not_operational</td>
<td></td>
</tr>
<tr>
<td>problematic</td>
<td></td>
</tr>
<tr>
<td>uninitialized</td>
<td></td>
</tr>
<tr>
<td>up</td>
<td></td>
</tr>
</tbody>
</table>

## 7.45. DEVICE STRUCT

A device wraps links to potential parents of a device.

### Table 7.60. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
</tbody>
</table>
### Table 7.61. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use vms instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

### 7.45.1. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.46. DISK STRUCT

Represents a virtual disk device.

### Table 7.62. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Boolean</td>
<td>Indicates if the disk is visible to the virtual machine.</td>
</tr>
<tr>
<td>actual_size</td>
<td>Integer</td>
<td>The actual size of the disk, in bytes.</td>
</tr>
<tr>
<td>alias</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>bootable</td>
<td>Boolean</td>
<td>Indicates if the disk is marked as bootable.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content_type</td>
<td>DiskContentType</td>
<td>Indicates the actual content residing on the disk.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>format</td>
<td>DiskFormat</td>
<td>The underlying storage format.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>image_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>initial_size</td>
<td>Integer</td>
<td>The initial size of a sparse image disk created on block storage, in bytes.</td>
</tr>
<tr>
<td>interface</td>
<td>DiskInterface</td>
<td>The type of interface driver used to connect the disk device to the virtual machine.</td>
</tr>
<tr>
<td>logical_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>lun_storage</td>
<td>HostStorage</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>propagate_errors</td>
<td>Boolean</td>
<td>Indicates if disk errors should cause virtual machine to be paused or if disk errors should be propagated to the the guest operating system instead.</td>
</tr>
<tr>
<td>provisioned_size</td>
<td>Integer</td>
<td>The virtual size of the disk, in bytes.</td>
</tr>
<tr>
<td>qcow_version</td>
<td>QcowVersion</td>
<td>The underlying QCOW version of a QCOW volume.</td>
</tr>
<tr>
<td>read_only</td>
<td>Boolean</td>
<td>Indicates if the disk is in read-only mode.</td>
</tr>
<tr>
<td>sgio</td>
<td>ScsiGenericIO</td>
<td></td>
</tr>
<tr>
<td>shareable</td>
<td>Boolean</td>
<td>Indicates if the disk can be attached to multiple virtual machines.</td>
</tr>
<tr>
<td>sparse</td>
<td>Boolean</td>
<td>Indicates if the physical storage for the disk should not be preallocated.</td>
</tr>
<tr>
<td>status</td>
<td>DiskStatus</td>
<td>The status of the disk device.</td>
</tr>
<tr>
<td>storage_type</td>
<td>DiskStorageType</td>
<td></td>
</tr>
<tr>
<td>total_size</td>
<td>Integer</td>
<td>The total size of the disk including all of its snapshots, in bytes.</td>
</tr>
<tr>
<td>uses_scsi_reservation</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wipe_after_delete</td>
<td>Boolean</td>
<td>Indicates if the disk’s blocks will be read back as zeros after it is deleted:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- On block storage, the disk will be zeroed and only then deleted.</td>
</tr>
</tbody>
</table>

### 7.46.1. active

Indicates if the disk is visible to the virtual machine.

**IMPORTANT**

When adding a disk attachment to a virtual machine, if the server accepts requests that do not contain this attribute the result is undefined. In some cases the disk will be automatically activated and in other cases it will not. To avoid issues it is strongly recommended to always include the this attribute with the desired value.

### 7.46.2. actual_size

The actual size of the disk, in bytes.

The actual size is the number of bytes actually used by the disk. It will be smaller than the provisioned size for disks that use the cow format.

### 7.46.3. bootable

Indicates if the disk is marked as bootable.

**IMPORTANT**

This attribute only makes sense for disks that are actually connected to virtual machines, and in version 4 of the API it has been moved to the DiskAttachment type. It is preserved here only for backwards compatibility, and it will be removed in the future.

### 7.46.4. initial_size

The initial size of a sparse image disk created on block storage, in bytes.

The initial size is the number of bytes a sparse disk is initially allocated with when created on block storage. The initial size will be smaller than the provisioned size. If not specified the default initial size used by the system will be allocated.

### 7.46.5. interface

The type of interface driver used to connect the disk device to the virtual machine.

**IMPORTANT**

This attribute only makes sense for disks that are actually connected to virtual machines, and in version 4 of the API it has been moved to the DiskAttachment type. It is preserved here only for backwards compatibility, and it will be removed in the future.
7.46.6. **provisioned_size**

The virtual size of the disk, in bytes.

This attribute is mandatory when creating a new disk.

7.46.7. **qcow_version**

The underlying QCOW version of a QCOW volume. The QCOW version specifies to the qemu which qemu version the volume supports. This field can be updated using the update API and will be reported only for QCOW volumes. It is determined by the version of the storage domain that the disk is created on. Storage domains with a version lower than V4 support QCOW2 volumes. V4 storage domains also support QCOW2v3. For more information about features of the different QCOW versions, see here.

7.46.8. **read_only**

Indicates if the disk is in read-only mode.

Since version 4.0 this attribute is not shown in the API and was moved to DiskAttachment.

Since version 4.1.2 of Red Hat Virtualization Manager this attribute is deprecated, and it will be removed in the future. In order to attach a disk in read only mode use the **read_only** attribute of the DiskAttachment type. For example:

```
POST /ovirt-engine/api/vms/123/diskattachments

<disk_attachment>
  <read_only>true</read_only>
  ...
</disk_attachment>
```

7.46.9. **shareable**

Indicates if the disk can be attached to multiple virtual machines.

**IMPORTANT**

When a disk is attached to multiple virtual machines it is the responsibility of the guest operating systems of those virtual machines to coordinate access to it, to avoid corruption of the data, for example using a shared file system like GlusterFS or GFS.

7.46.10. **total_size**

The total size of the disk including all of its snapshots, in bytes.

The total size is the number of bytes actually used by the disk plus the size of its snapshots. It won't be populated for direct LUN and Cinder disks. For disks without snapshots the total size is equal to the actual size.

7.46.11. **wipe_after_delete**

Indicates if the disk's blocks will be read back as zeros after it is deleted:
- On block storage, the disk will be zeroed and only then deleted.
- On file storage, since the file system already guarantees that previously removed blocks are read back as zeros, the disk will be deleted immediately.

Table 7.63. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
<td></td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>openstack_volume_type</td>
<td>OpenStackVolume Type</td>
<td></td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td></td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Statistics exposed by the disk.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td>The storage domains associated with this disk.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use vms instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

7.46.12. statistics

Statistics exposed by the disk. For example:

```xml
<statistics>
  <statistic href="/ovirt-engine/api/api/disks/123/statistics/456" id="456">
    <name>data.current.read</name>
    <description>Read data rate</description>
    <kind>gauge</kind>
    <type>decimal</type>
    <unit>bytes_per_second</unit>
    <values>
      <value>
        <datum>1052</datum>
      </value>
    </values>
  </statistic>
</statistics>
```
These statistics are not directly included when the disk is retrieved, only a link. To obtain the statistics follow the included link:

```plaintext
GET /ovirt-engine/api/disks/123/statistics
```

### 7.46.13. storage_domains

The storage domains associated with this disk.

**NOTE**

Only required when the first disk is being added to a virtual machine that was not itself created from a template.

### 7.46.14. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk my be used simultaneously by two or more virtual machines.

### 7.47. DISKATTACHMENT STRUCT

Describes how a disk is attached to a virtual machine.

**Table 7.64. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Boolean</td>
<td>Defines whether the disk is active in the virtual machine it’s attached to.</td>
</tr>
<tr>
<td>bootable</td>
<td>Boolean</td>
<td>Defines whether the disk is bootable.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>interface</td>
<td>DiskInterface</td>
<td>The type of interface driver used to connect the disk device to the virtual machine.</td>
</tr>
<tr>
<td>logical_name</td>
<td>String</td>
<td>The logical name of the virtual machine’s disk, as seen from inside the virtual machine.</td>
</tr>
</tbody>
</table>
### Name | Type | Summary
---|---|---
name | String | A human-readable name in plain text.
pass_discard | Boolean | Defines whether the virtual machine passes discard commands to the storage.
read_only | Boolean | Indicates whether the disk is connected to the virtual machine as read only.
uses_scsi_reservation | Boolean | Defines whether SCSI reservation is enabled for this disk.

### 7.47.1. active
Defines whether the disk is active in the virtual machine it’s attached to.

A disk attached to a virtual machine in an active status is connected to the virtual machine at run time and can be used.

### 7.47.2. logical_name
The logical name of the virtual machine’s disk, as seen from inside the virtual machine.

The logical name of a disk is reported only when the guest agent is installed and running inside the virtual machine.

For example, if the guest operating system is Linux and the disk is connected via a VirtIO interface, the logical name will be reported as `/dev/vda`:

```
<disk_attachment>
  ...
  <logical_name>/dev/vda</logical_name>
</disk_attachment>
```

If the guest operating system is Windows, the logical name will be reported as `\PHYSICALDRIVE0`.

### 7.47.3. read_only
Indicates whether the disk is connected to the virtual machine as read only.

When adding a new disk attachment the default value is `false`.

```
<disk_attachment>
  ...
  <read_only>true</read_only>
</disk_attachment>
```

### 7.47.4. uses_scsi_reservation
Defines whether SCSI reservation is enabled for this disk.

Virtual machines with VIRTIO-SCSI passthrough enabled can set persistent SCSI reservations on disks. If they set persistent SCSI reservations, those virtual machines cannot be migrated to a different host because they would lose access to the disk, because SCSI reservations are specific to SCSI initiators, and therefore hosts. This scenario cannot be automatically detected. To avoid migrating these virtual machines, the user can set this attribute to `true`, to indicate the virtual machine is using SCSI reservations.

### Table 7.65. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>The reference to the disk.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>The reference to the template.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>The reference to the virtual machine.</td>
</tr>
</tbody>
</table>

### 7.48. DISKCONTENTTYPE ENUM

The actual content residing on the disk.

### Table 7.66. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>The disk contains data.</td>
</tr>
<tr>
<td>iso</td>
<td>The disk contains an ISO image to be used a CDROM device.</td>
</tr>
<tr>
<td>memory_dump_volume</td>
<td>The disk contains a memory dump from a live snapshot.</td>
</tr>
<tr>
<td>memory_metad_volume</td>
<td>The disk contains memory metadata from a live snapshot.</td>
</tr>
<tr>
<td>ovf_store</td>
<td>The disk is an OVF store.</td>
</tr>
</tbody>
</table>

### 7.49. DISKFORMAT ENUM

The underlying storage format of disks.

### Table 7.67. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cow</td>
<td>The <em>Copy On Write</em> format allows snapshots, with a small performance overhead.</td>
</tr>
</tbody>
</table>
### 7.50. DISKINTERFACE ENUM

The underlying storage interface of disks communication with controller.

**Table 7.68. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw</td>
<td>The raw format does not allow snapshots, but offers improved performance.</td>
</tr>
</tbody>
</table>

### 7.50.1. ide

Legacy controller device. Works with almost all guest operating systems, so it is good for compatibility. Performance is lower than with the other alternatives.

### 7.50.2. virtio

Virtualization interface where just the guest’s device driver knows it is running in a virtual environment. Enables guests to get high performance disk operations.

### 7.50.3. virtio_scsi

Para-virtualized SCSI controller device. Fast interface with the guest via direct physical storage device address, using the SCSI protocol.

### 7.51. DISKPROFILE STRUCT

**Table 7.69. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>
### Name | Type | Summary
--- | --- | ---
name | String | A human-readable name in plain text.

#### Table 7.70. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td></td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
</tbody>
</table>

#### 7.52. DISKSNAPSHOT STRUCT

#### Table 7.71. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Boolean</td>
<td>Indicates if the disk is visible to the virtual machine.</td>
</tr>
<tr>
<td>actual_size</td>
<td>Integer</td>
<td>The actual size of the disk, in bytes.</td>
</tr>
<tr>
<td>alias</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>bootable</td>
<td>Boolean</td>
<td>Indicates if the disk is marked as bootable.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content_type</td>
<td>DiskContentType</td>
<td>Indicates the actual content residing on the disk.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>format</td>
<td>DiskFormat</td>
<td>The underlying storage format.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>image_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>initial_size</td>
<td>Integer</td>
<td>The initial size of a sparse image disk created on block storage, in bytes.</td>
</tr>
<tr>
<td>interface</td>
<td>DiskInterface</td>
<td>The type of interface driver used to connect the disk device to the virtual machine.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>logical_name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>lun_storage</td>
<td>HostStorage</td>
<td>Indicates if disk errors should cause virtual machine to be paused or if disk errors should be propagated to the guest operating system instead.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>The virtual size of the disk, in bytes.</td>
</tr>
<tr>
<td>propagate_errors</td>
<td>Boolean</td>
<td>The underlying QCOW version of a QCOW volume.</td>
</tr>
<tr>
<td>read_only</td>
<td>Boolean</td>
<td>Indicates if the disk is in read-only mode.</td>
</tr>
<tr>
<td>sgio</td>
<td>ScsiGenericIO</td>
<td>Indicates if the disk can be attached to multiple virtual machines.</td>
</tr>
<tr>
<td>shareable</td>
<td>Boolean</td>
<td>Indicates if the physical storage for the disk should not be preallocated.</td>
</tr>
<tr>
<td>status</td>
<td>DiskStatus</td>
<td>The status of the disk device.</td>
</tr>
<tr>
<td>storage_type</td>
<td>DiskStorageType</td>
<td>The total size of the disk including all of its snapshots, in bytes.</td>
</tr>
<tr>
<td>uses_scsi_reservation</td>
<td>Boolean</td>
<td>Indicates if the disk's blocks will be read back as zeros after it is deleted:</td>
</tr>
<tr>
<td>wipe_after_delete</td>
<td>Boolean</td>
<td>- On block storage, the disk will be zeroed and only then deleted.</td>
</tr>
</tbody>
</table>

7.52.1. active

Indicates if the disk is visible to the virtual machine.
When adding a disk attachment to a virtual machine, if the server accepts requests that do not contain this attribute the result is undefined. In some cases the disk will be automatically activated and in other cases it will not. To avoid issues it is strongly recommended to always include the this attribute with the desired value.

7.52.2. actual_size

The actual size of the disk, in bytes.

The actual size is the number of bytes actually used by the disk. It will be smaller than the provisioned size for disks that use the cow format.

7.52.3. bootable

Indicates if the disk is marked as bootable.

This attribute only makes sense for disks that are actually connected to virtual machines, and in version 4 of the API it has been moved to the DiskAttachment type. It is preserved here only for backwards compatibility, and it will be removed in the future.

7.52.4. initial_size

The initial size of a sparse image disk created on block storage, in bytes.

The initial size is the number of bytes a sparse disk is initially allocated with when created on block storage. The initial size will be smaller than the provisioned size. If not specified the default initial size used by the system will be allocated.

7.52.5. interface

The type of interface driver used to connect the disk device to the virtual machine.

This attribute only makes sense for disks that are actually connected to virtual machines, and in version 4 of the API it has been moved to the DiskAttachment type. It is preserved here only for backwards compatibility, and it will be removed in the future.

7.52.6. provisioned_size

The virtual size of the disk, in bytes.

This attribute is mandatory when creating a new disk.

7.52.7. qcow_version

The underlying QCOW version of a QCOW volume. The QCOW version specifies to the qemu which qemu version the volume supports. This field can be updated using the update API and will be reported only for QCOW volumes. It is determined by the version of the storage domain that the disk is created
on. Storage domains with a version lower than V4 support QCOW2 volumes. V4 storage domains also support QCOW2v3. For more information about features of the different QCOW versions, see here.

### 7.52.8. read_only

Indicates if the disk is in read-only mode.

Since version 4.0 this attribute is not shown in the API and was moved to DiskAttachment.

Since version 4.1.2 of Red Hat Virtualization Manager this attribute is deprecated, and it will be removed in the future. In order to attach a disk in read only mode use the read_only attribute of the DiskAttachment type. For example:

```xml
POST /ovirt-engine/api/vms/123/diskattachments

<disk_attachment>
    <read_only>true</read_only>
    ...
</disk_attachment>
```

### 7.52.9. shareable

Indicates if the disk can be attached to multiple virtual machines.

**IMPORTANT**

When a disk is attached to multiple virtual machines it is the responsibility of the guest operating systems of those virtual machines to coordinate access to it, to avoid corruption of the data, for example using a shared file system like GlusterFS or GFS.

### 7.52.10. total_size

The total size of the disk including all of its snapshots, in bytes.

The total size is the number of bytes actually used by the disk plus the size of its snapshots. It won’t be populated for direct LUN and Cinder disks. For disks without snapshots the total size is equal to the actual size.

### 7.52.11. wipe_after_delete

Indicates if the disk’s blocks will be read back as zeros after it is deleted:

- On block storage, the disk will be zeroed and only then deleted.
- On file storage, since the file system already guarantees that previously removed blocks are read back as zeros, the disk will be deleted immediately.

**Table 7.72. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>disk_profile</td>
<td>DiskProfile</td>
<td></td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>openstack_volume_type</td>
<td>OpenStackVolumeType</td>
<td></td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td></td>
</tr>
<tr>
<td>snapshot</td>
<td>Snapshot</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Statistics exposed by the disk.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
<tr>
<td>storage_domains</td>
<td>StorageDomain[]</td>
<td>The storage domains associated with this disk.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use vms instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

### 7.52.12. statistics

Statistics exposed by the disk. For example:

```xml
<statistics>
  <statistic href="/ovirt-engine/api/disks/123/statistics/456" id="456">
    <name>data.current.read</name>
    <description>Read data rate</description>
    <kind>gauge</kind>
    <type>decimal</type>
    <unit>bytes_per_second</unit>
    <values>
      <value>
        <datum>1052</datum>
      </value>
    </values>
  </statistic>
</statistics>
```
These statistics are not directly included when the disk is retrieved, only a link. To obtain the statistics follow the included link:

```
GET /ovirt-engine/api/disks/123/statistics
```

### 7.52.13. storage_domains

The storage domains associated with this disk.

**NOTE**

Only required when the first disk is being added to a virtual machine that was not itself created from a template.

### 7.52.14. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.53. DISKSTATUS ENUM

Current status representation for disk.

**Table 7.73. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>illegal</td>
<td>Disk cannot be accessed by the virtual machine, and the user needs to take action to resolve the issue.</td>
</tr>
<tr>
<td>locked</td>
<td>The disk is being used by the system, therefore it cannot be accessed by virtual machines at this point.</td>
</tr>
<tr>
<td>ok</td>
<td>The disk status is normal and can be accessed by the virtual machine.</td>
</tr>
</tbody>
</table>

#### 7.53.1. locked

The disk is being used by the system, therefore it cannot be accessed by virtual machines at this point. This is usually a temporary status, until the disk is freed.

### 7.54. DISKSTORAGETYPE ENUM

**Table 7.74. Values summary**
### 7.55. DISKTYPE ENUM

#### Table 7.75. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td></td>
</tr>
<tr>
<td>system</td>
<td></td>
</tr>
</tbody>
</table>

### 7.56. DISPLAY STRUCT

Represents a graphic console configuration.

#### Table 7.76. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>The IP address of the guest to connect the graphic console client to.</td>
</tr>
<tr>
<td>allow_override</td>
<td>Boolean</td>
<td>Indicates if to override the display address per host.</td>
</tr>
<tr>
<td>certificate</td>
<td>Certificate</td>
<td>The TLS certificate in case of a TLS connection.</td>
</tr>
<tr>
<td>copy_paste_enabled</td>
<td>Boolean</td>
<td>Indicates whether a user is able to copy and paste content from an external host into the graphic console.</td>
</tr>
<tr>
<td>disconnect_action</td>
<td>String</td>
<td>Returns the action that will take place when the graphic console is disconnected.</td>
</tr>
<tr>
<td>file_transfer_enabled</td>
<td>Boolean</td>
<td>Indicates if a user is able to drag and drop files from an external host into the graphic console.</td>
</tr>
<tr>
<td>keyboard_layout</td>
<td>String</td>
<td>The keyboard layout to use with this graphic console.</td>
</tr>
<tr>
<td>monitors</td>
<td>Integer</td>
<td>The number of monitors opened for this graphic console.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td>The port address on the guest to connect the graphic console client to.</td>
</tr>
<tr>
<td>proxy</td>
<td>String</td>
<td>The proxy IP which will be used by the graphic console client to connect to the guest.</td>
</tr>
<tr>
<td>secure_port</td>
<td>Integer</td>
<td>The secured port address on the guest, in case of using TLS, to connect the graphic console client to.</td>
</tr>
<tr>
<td>single_qxl_pci</td>
<td>Boolean</td>
<td>Indicates if to use one PCI slot for each monitor or to use a single PCI channel for all multiple monitors.</td>
</tr>
<tr>
<td>smartcard_enabled</td>
<td>Boolean</td>
<td>Indicates if to use smart card authentication.</td>
</tr>
<tr>
<td>type</td>
<td>DisplayType</td>
<td>The graphic console protocol type.</td>
</tr>
</tbody>
</table>

7.56.1. allow_override

Indicates if to override the display address per host. Relevant only for the **Host.display** attribute. If set, the graphical console address of a virtual machine will be overridden by the host specified display address. If not set, the graphical console address of a virtual machine will not be overridden.

7.56.2. certificate

The TLS certificate in case of a TLS connection. If TLS isn’t enabled then it won’t be reported.

7.56.3. copy_paste_enabled

Indicates whether a user is able to copy and paste content from an external host into the graphic console. This option is only available for the SPICE console type.

7.56.4. disconnect_action

Returns the action that will take place when the graphic console is disconnected. The options are:

- **none**
  
  No action is taken.

- **lock_screen**
  
  Locks the currently active user session.

- **logout**
  
  Logs out the currently active user session.

- **reboot**
  
  Initiates a graceful virtual machine reboot.

- **shutdown**
  
  Initiates a graceful virtual machine shutdown.
This option is only available for the SPICE console type.

### 7.56.5. file_transfer_enabled

Indicates if a user is able to drag and drop files from an external host into the graphic console. This option is only available for the SPICE console type.

### 7.56.6. keyboard_layout

The keyboard layout to use with this graphic console. This option is only available for the VNC console type. If no keyboard is enabled then it won’t be reported.

### 7.56.7. monitors

The number of monitors opened for this graphic console. This option is only available for the SPICE console type. Possible values are 1, 2 or 4.

### 7.56.8. proxy

The proxy IP which will be used by the graphic console client to connect to the guest. It is useful when the client is outside the guest’s network. This option is only available for the SPICE console type. This proxy can be set in global configuration, cluster level, virtual machine pool level or disabled per virtual machine. If the proxy is set in any of this mentioned places and not disabled for the virtual machine, it will be returned by this method. If the proxy is not set, nothing will be reported.

### 7.56.9. secure_port

The secured port address on the guest, in case of using TLS, to connect the graphic console client to. If TLS isn’t enabled then it won’t be reported.

### 7.56.10. single_qxl_pci

Indicates if to use one PCI slot for each monitor or to use a single PCI channel for all multiple monitors. This option is only available for the SPICE console type and only for connecting a guest Linux based OS.

### 7.56.11. smartcard_enabled

Indicates if to use smart card authentication. This option is only available for the SPICE console type.

### 7.57. DISPLAYTYPE ENUM

Represents an enumeration of the protocol used to connect to the graphic console of the virtual machine.

#### Table 7.77. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>spice</td>
<td>Display of type SPICE.</td>
</tr>
<tr>
<td>vnc</td>
<td>Display of type VNC.</td>
</tr>
</tbody>
</table>
7.57.1. spice
Display of type SPICE. See https://www.spice-space.org for more details.

7.57.2. vnc
Display of type VNC. VNC stands for Virtual Network Computing, and it is a graphical desktop sharing system that uses RFB (Remote Frame Buffer) protocol to remotely control another machine.

7.58. DNS STRUCT
Represents the DNS resolver configuration.

Table 7.78. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>search_domains</td>
<td>Host[]</td>
<td>Array of hosts serving as search domains.</td>
</tr>
<tr>
<td>servers</td>
<td>Host[]</td>
<td>Array of hosts serving as DNS servers.</td>
</tr>
</tbody>
</table>

7.59. DNSRESOLVERCONFIGURATION STRUCT
Represents the DNS resolver configuration.

Table 7.79. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name_servers</td>
<td>String[]</td>
<td>Array of addresses of name servers.</td>
</tr>
</tbody>
</table>

7.59.1. name_servers
Array of addresses of name servers. Either IPv4 or IPv6 addresses may be specified.

7.60. DOMAIN STRUCT
This type represents a directory service domain.

Table 7.80. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>
### Table 7.81. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>groups</td>
<td>Group[]</td>
<td>A reference to all groups in the directory service.</td>
</tr>
<tr>
<td>users</td>
<td>User[]</td>
<td>A reference to a list of all users in the directory service.</td>
</tr>
</tbody>
</table>

#### 7.60.1. users

A reference to a list of all users in the directory service. This information is used to add new users to the Red Hat Virtualization environment.

#### 7.61. ENTITYEXTERNALSTATUS ENUM

Type representing an external entity status.

### Table 7.82. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>The external entity status is erroneous.</td>
</tr>
<tr>
<td>failure</td>
<td>The external entity has an issue that causes failures.</td>
</tr>
<tr>
<td>info</td>
<td>There external entity status is okay but with some information that might be relevant.</td>
</tr>
<tr>
<td>ok</td>
<td>The external entity status is okay.</td>
</tr>
<tr>
<td>warning</td>
<td>The external entity status is okay but with an issue that might require attention.</td>
</tr>
</tbody>
</table>

#### 7.61.1. error

The external entity status is erroneous. This might require a moderate attention.

#### 7.61.2. failure

The external entity has an issue that causes failures. This might require immediate attention.

#### 7.62. ENTITYPROFILEDETAIL STRUCT
### 7.63. ERRORHANDLING STRUCT

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>on_error</td>
<td>MigrateOnError</td>
<td></td>
</tr>
</tbody>
</table>

### 7.64. EVENT STRUCT

Type representing an event.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>Integer</td>
<td>The event code.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>correlation_id</td>
<td>String</td>
<td>The event correlation identifier.</td>
</tr>
<tr>
<td>custom_data</td>
<td>String</td>
<td>Free text representing custom event data.</td>
</tr>
<tr>
<td>custom_id</td>
<td>Integer</td>
<td>A custom event identifier.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>flood_rate</td>
<td>Integer</td>
<td>Defines the flood rate.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>index</td>
<td>Integer</td>
<td>The numeric index of this event.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>Free text identifying the origin of the event.</td>
</tr>
<tr>
<td>severity</td>
<td>LogSeverity</td>
<td>The event severity.</td>
</tr>
</tbody>
</table>
7.64.1. correlation_id

The event correlation identifier. Used in order to correlate several events together.

7.64.2. flood_rate

Defines the flood rate. This prevents flooding in case an event appeared more than once in the defined rate. Defaults is 30 seconds.

7.64.3. index

The numeric index of this event. The indexes of events are always increasing, so events with higher indexes are guaranteed to be older than events with lower indexes.

**IMPORTANT**

In the current implementation of the engine, the id attribute has the same value as this index attribute. That is an implementation detail that the user of the API should not rely on. In the future the id attribute may be changed to an arbitrary string, containing non numeric characters and no implicit order. On the other hand this index attribute is guaranteed to stay as integer and ordered.

**Table 7.86. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to the cluster service.</td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>Reference to the data center service.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host service.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to the storage domain service.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Reference to the template service.</td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td>Reference to the user service.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Reference to the virtual machine service.</td>
</tr>
</tbody>
</table>

7.64.4. cluster

Reference to the cluster service. Event can be associated with a cluster.
7.64.5. data_center
Reference to the data center service. Event can be associated with a data center.

7.64.6. host
Reference to the host service. Event can be associated with a host.

7.64.7. storage_domain
Reference to the storage domain service. Event can be associated with a storage domain.

7.64.8. template
Reference to the template service. Event can be associated with a template.

7.64.9. user
Reference to the user service. Event can be associated with a user.

7.64.10. vm
Reference to the virtual machine service. Event can be associated with a virtual machine.

7.65. EXTERNALCOMPUTERESOURCE STRUCT

Table 7.87. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>provider</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.88. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td></td>
</tr>
</tbody>
</table>

### 7.66. EXTERNALDISCOVEREDHOST STRUCT

#### Table 7.89. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ip</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>last_report</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>mac</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>subnet_name</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 7.90. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td></td>
</tr>
</tbody>
</table>

### 7.67. EXTERNALHOST STRUCT

Represents a host provisioned by a host provider (such as Foreman/Satellite).


#### Table 7.91. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>The address of the host, either IP address of FQDN (Fully Qualified Domain Name).</td>
</tr>
</tbody>
</table>
Table 7.92. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>external_host_provider</strong></td>
<td><strong>ExternalHostProvider</strong></td>
<td>A reference to the external host provider that the host is managed by.</td>
</tr>
</tbody>
</table>

### 7.68. EXTERNALHOSTGROUP STRUCT

Table 7.93. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>architecture_name</strong></td>
<td>String</td>
<td></td>
</tr>
<tr>
<td><strong>comment</strong></td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td><strong>description</strong></td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td><strong>domain_name</strong></td>
<td>String</td>
<td></td>
</tr>
<tr>
<td><strong>id</strong></td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td><strong>name</strong></td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td><strong>operating_system_name</strong></td>
<td>String</td>
<td></td>
</tr>
<tr>
<td><strong>subnet_name</strong></td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.94. Links summary
7.69. EXTERNALHOSTPROVIDER STRUCT

Represents an external host provider, such as Foreman or Satellite.


<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.95. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

7.69.1. requires_authentication

Defines whether provider authentication is required or not.

If authentication is required, both **username** and **password** attributes will be used during authentication.

Table 7.96. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>A reference to the certificates the engine supports for this provider.</td>
</tr>
<tr>
<td>compute_resources</td>
<td>ExternalComputeResource[]</td>
<td>A reference to the compute resource as represented in the host provider.</td>
</tr>
<tr>
<td>discovered_hosts</td>
<td>ExternalDiscoveredHost[]</td>
<td>A reference to the discovered hosts in the host provider.</td>
</tr>
<tr>
<td>host_groups</td>
<td>ExternalHostGroup[]</td>
<td>A reference to the host groups in the host provider.</td>
</tr>
<tr>
<td>hosts</td>
<td>Host[]</td>
<td>A reference to the hosts provisioned by the host provider.</td>
</tr>
</tbody>
</table>

### 7.69.2. compute_resources

A reference to the compute resource as represented in the host provider. Each host provider optionally has the engine defined as a compute resource, which allows to create virtual machines in the engine. This compute resource details are used in the Bare-Metal provisioning use-case, in order to deploy the hypervisor.

### 7.69.3. discovered_hosts

A reference to the discovered hosts in the host provider. Discovered hosts are hosts that were not provisioned yet.

### 7.69.4. host_groups

A reference to the host groups in the host provider. Host group contains different properties that the host provider applies on all hosts that are member of this group. Such as installed software, system definitions, passwords and more.

### 7.70. EXTERNALNETWORKPROVIDERCONFIGURATION STRUCT

Describes how an external network provider is provisioned on a host.

#### Table 7.97. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>
### Table 7.98. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>external_network_provider</td>
<td>ExternalProvider</td>
<td>Link to the external network provider.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Link to the host.</td>
</tr>
</tbody>
</table>

### 7.71. EXTERNALPROVIDER STRUCT

Represents an external provider.

### Table 7.99. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

### 7.71.1. requires_authentication

Defines whether provider authentication is required or not.
If authentication is required, both username and password attributes will be used during authentication.

### 7.72. EXTERNALSTATUS ENUM

Represents an external status. This status is currently used for hosts and storage domains, and allows an external system to update status of objects it is aware of.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>error</td>
<td>Error status.</td>
</tr>
<tr>
<td>failure</td>
<td>Failure status.</td>
</tr>
<tr>
<td>info</td>
<td>Info status.</td>
</tr>
<tr>
<td>ok</td>
<td>OK status.</td>
</tr>
<tr>
<td>warning</td>
<td>Warning status.</td>
</tr>
</tbody>
</table>

#### 7.72.1. error

Error status. There is some kind of error in the relevant object.

#### 7.72.2. failure

Failure status. The relevant object is failing.

#### 7.72.3. info

Info status. The relevant object is in OK status, but there is an information available that might be relevant for the administrator.

#### 7.72.4. ok

OK status. The relevant object is working well.

#### 7.72.5. warning

Warning status. The relevant object is working well, but there is some warning that might be relevant for the administrator.

### 7.73. EXTERNALSYSTEMTYPE ENUM

Represents the type of the external system that is associated with the step.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.101. Values summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the virtual machine to be imported, as is defined within the external system.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>The password to authenticate against the external hypervisor system.</td>
</tr>
<tr>
<td>provider</td>
<td>ExternalVmProvider</td>
<td>The type of external virtual machine provider.</td>
</tr>
<tr>
<td>sparse</td>
<td>Boolean</td>
<td>Specifies the disk allocation policy of the resulting virtual machine: true for sparse, false for preallocated.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>The URL to be passed to the virt-v2v tool for conversion.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>The username to authenticate against the external hypervisor system.</td>
</tr>
</tbody>
</table>

### 7.74.1. url

The URL to be passed to the virt-v2v tool for conversion.

Example:

```
vpx://vmware_user@vcenter-host/DataCenter/Cluster/esxi-host?no_verify=1
```

More examples can be found at [http://libguestfs.org/virt-v2v.1.html](http://libguestfs.org/virt-v2v.1.html).

#### 7.74. EXTERNALVMIMPORT STRUCT

Describes the parameters for the virtual machine import operation from an external system.

##### Table 7.102. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the virtual machine to be imported, as is defined within the external system.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>The password to authenticate against the external hypervisor system.</td>
</tr>
<tr>
<td>provider</td>
<td>ExternalVmProvider</td>
<td>The type of external virtual machine provider.</td>
</tr>
<tr>
<td>sparse</td>
<td>Boolean</td>
<td>Specifies the disk allocation policy of the resulting virtual machine: true for sparse, false for preallocated.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>The URL to be passed to the virt-v2v tool for conversion.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>The username to authenticate against the external hypervisor system.</td>
</tr>
</tbody>
</table>

### 7.74.1. url

The URL to be passed to the virt-v2v tool for conversion.

Example:

```
vpx://vmware_user@vcenter-host/DataCenter/Cluster/esxi-host?no_verify=1
```

More examples can be found at [http://libguestfs.org/virt-v2v.1.html](http://libguestfs.org/virt-v2v.1.html).

#### Table 7.103. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Specifies the target cluster for the resulting virtual machine.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Optional.</td>
</tr>
</tbody>
</table>
7.74.2. cpu_profile
Optional. Specifies the CPU profile of the resulting virtual machine.

7.74.3. drivers_iso
Optional. The name of the ISO containing drivers that can be used during the `virt-v2v` conversion process.

7.74.4. host
Optional. Specifies the host (using host's ID) to be used for the conversion process. If not specified, one is selected automatically.

7.74.5. quota
Optional. Specifies the quota that will be applied to the resulting virtual machine.

7.74.6. vm
The virtual machine entity used to specify a name for the newly created virtual machine.

If a name is not specified, the source virtual machine name will be used.

7.75. EXTERNALVMPROVIDERTYPE ENUM
Describes the type of external hypervisor system.

Table 7.104. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>kvm</td>
<td></td>
</tr>
<tr>
<td>vmware</td>
<td></td>
</tr>
</tbody>
</table>
### 7.76. FAULT STRUCT

Table 7.105. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>detail</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>reason</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.77. FENCETYPE ENUM

Type representing the type of the fence operation.

Table 7.106. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>manual</td>
<td>Manual host fencing via power management.</td>
</tr>
<tr>
<td>restart</td>
<td>Restart the host via power management.</td>
</tr>
<tr>
<td>start</td>
<td>Start the host via power management.</td>
</tr>
<tr>
<td>status</td>
<td>Check the host power status via power management.</td>
</tr>
<tr>
<td>stop</td>
<td>Stop the host via power management.</td>
</tr>
</tbody>
</table>

### 7.78. FENCINGPOLICY STRUCT

Type representing a cluster fencing policy.

Table 7.107. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Enable or disable fencing on this cluster.</td>
</tr>
<tr>
<td>skip_if_connectivity_broken</td>
<td>SkipIfConnectivityBroken</td>
<td>If enabled, we will not fence a host in case more than a configurable percentage of hosts in the cluster lost connectivity as well.</td>
</tr>
</tbody>
</table>
### 7.78.1. skip_if_connectivity_broken

If enabled, we will not fence a host in case more than a configurable percentage of hosts in the cluster lost connectivity as well. This comes to prevent fencing *storm* in cases where there is a global networking issue in the cluster.

### 7.78.2. skip_if_gluster_bricks_up

A flag indicating if fencing should be skipped if Gluster bricks are up and running in the host being fenced. This flag is optional, and the default value is `false`.

### 7.78.3. skip_if_gluster_quorum_not_met

A flag indicating if fencing should be skipped if Gluster bricks are up and running and Gluster quorum will not be met without those bricks. This flag is optional, and the default value is `false`.

### 7.78.4. skip_if_sd_active

If enabled, we will skip fencing in case the host maintains its lease in the storage. It means that if the host still has storage access then it won’t get fenced.

### 7.79. FILE STRUCT

Table 7.108. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>
Table 7.109. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
</tbody>
</table>

### 7.80. FILTER STRUCT

Table 7.110. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>position</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.111. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>scheduling_policy_unit</td>
<td>SchedulingPolicyUnit</td>
<td></td>
</tr>
</tbody>
</table>

### 7.81. FIREWALLTYPE ENUM

Describes all firewall types supported by the system.

Table 7.112. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>firewalld</td>
<td>FirewallD firewall type.</td>
</tr>
</tbody>
</table>
7.81.1. firewalld

FirewallD firewall type.

When a cluster has the firewall type set to `firewalld`, the firewalls of all hosts in the cluster will be configured using `firewalld`. FirewallD replaced IPTables in version 4.2. It simplifies configuration using a command line program and dynamic configuration.

7.81.2. iptables

IPTables firewall type.

When a cluster has the firewall type set to `iptables`, the firewalls of all hosts in the cluster will be configured using `iptables`. `iptables` adds firewall rules to `/etc/sysconfig/iptables` using a special `iptables` syntax. For more information, see the `iptables` manual page.

`iptables` is deprecated in cluster version 4.2 and will be removed in cluster version 4.3.

7.82. FLOPPY STRUCT

The underlying representation of a floppy file.

Table 7.113. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>file</td>
<td>File</td>
<td>File object that represent the Floppy device’s content and its type.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

Table 7.114. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use vms instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

### 7.82.1. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.83. FOPSTATISTIC STRUCT

Table 7.115. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.84. GLUSTERBRICK STRUCT

Table 7.116. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>brick_dir</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>device</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>fs_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>gluster_clients</td>
<td>GlusterClient[]</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>memory_pools</td>
<td>GlusterMemoryPool[]</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7.117. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>gluster_volume</td>
<td>GlusterVolume</td>
<td>Optiony references to a gluster volume the device is used by.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally reference to an instance type the device is used by.</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Optiony references to a template the device is used by.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>References to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don't use this element, use <code>vms</code> instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

#### 7.84.1. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.85. GLUSTERBRICKADVANCEDDETAILS STRUCT

#### Table 7.118. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
</tbody>
</table>
### Table 7.119. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use <code>vms</code> instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

#### 7.85.1. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.86. GLUSTERBRICKMEMORYINFO STRUCT

#### Table 7.120. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>memory_pools</td>
<td>GlusterMemoryPool[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.87. GLUSTERBRICKSTATUS ENUM

Table 7.121. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>Brick is in <strong>down</strong> state, the data cannot be stored or retrieved from it.</td>
</tr>
<tr>
<td>unknown</td>
<td>When the status cannot be determined due to host being non-responsive.</td>
</tr>
<tr>
<td>up</td>
<td>Brick is in <strong>up</strong> state, the data can be stored or retrieved from it.</td>
</tr>
</tbody>
</table>

### 7.88. GLUSTERCLIENT STRUCT

Table 7.122. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes_read</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>bytes_written</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>client_port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>host_name</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.89. GLUSTERHOOK STRUCT

Table 7.123. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>conflict_status</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>conflicts</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
Table 7.124. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>server_hooks</td>
<td>GlusterServerHook</td>
<td>[]</td>
</tr>
</tbody>
</table>

7.90. GLUSTERHOOKSTATUS ENUM

Table 7.125. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>Hook is disabled in the cluster.</td>
</tr>
<tr>
<td>enabled</td>
<td>Hook is enabled in the cluster.</td>
</tr>
<tr>
<td>missing</td>
<td>Unknown/missing hook status.</td>
</tr>
</tbody>
</table>

7.91. GLUSTERMEMORYPOOL STRUCT

Table 7.126. Attributes summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>alloc_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>cold_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>hot_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>max_alloc</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>max_stdalloc</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>padded_size</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>pool_misses</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.92. GLUSTERSERVERHOOK STRUCT

Table 7.127. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>checksum</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content_type</td>
<td>HookContentType</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>status</td>
<td>GlusterHookStatus</td>
<td></td>
</tr>
</tbody>
</table>
Table 7.128. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
</tbody>
</table>

7.93. GLUSTERSTATE ENUM

Table 7.129. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td></td>
</tr>
<tr>
<td>unknown</td>
<td></td>
</tr>
<tr>
<td>up</td>
<td></td>
</tr>
</tbody>
</table>

7.94. GLUSTERVERVOLUME STRUCT

Table 7.130. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>disperse_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>options</td>
<td>Option[]</td>
<td></td>
</tr>
<tr>
<td>redundancy_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>replica_count</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>GlusterVolumeStatus</td>
<td></td>
</tr>
<tr>
<td>stripe_count</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>
### Table 7.131. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bricks</td>
<td>GlusterBrick[]</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.95. GLUSTERVOLUMEPROFILEDETAILS STRUCT

### Table 7.132. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>brick_profile_details</td>
<td>BrickProfileDetail[]</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>nfs_profile_details</td>
<td>NfsProfileDetail[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.96. GLUSTERVOLUMESTATUS ENUM

### Table 7.133. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>Volume needs to be started, for clients to be able to mount and use it.</td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>unknown</td>
<td>When the status cannot be determined due to host being non-responsive.</td>
</tr>
<tr>
<td>up</td>
<td>Volume is started, and can be mounted and used by clients.</td>
</tr>
</tbody>
</table>

### 7.97. GLUSTERVOLUMETYPE ENUM

Type representing the type of Gluster Volume.

#### Table 7.134. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disperse</td>
<td>Dispersed volumes are based on erasure codes, providing space-efficient protection against disk or server failures.</td>
</tr>
<tr>
<td>distribute</td>
<td>Distributed volumes distributes files throughout the bricks in the volume.</td>
</tr>
<tr>
<td>distributed_disperse</td>
<td>Distributed dispersed volumes distribute files across dispersed subvolumes.</td>
</tr>
<tr>
<td>distributed_replicate</td>
<td>Distributed replicated volumes distributes files across replicated bricks in the volume.</td>
</tr>
<tr>
<td>distributed_stripe</td>
<td>Distributed striped volumes stripe data across two or more nodes in the cluster.</td>
</tr>
<tr>
<td>distributed_stripped_replicate</td>
<td>Distributed striped replicated volumes distributes striped data across replicated bricks in the cluster.</td>
</tr>
<tr>
<td>replicate</td>
<td>Replicated volumes replicates files across bricks in the volume.</td>
</tr>
<tr>
<td>stripe</td>
<td>Striped volumes stripes data across bricks in the volume.</td>
</tr>
<tr>
<td>striped_replicate</td>
<td>Striped replicated volumes stripes data across replicated bricks in the cluster.</td>
</tr>
</tbody>
</table>

#### 7.97.1. disperse

Dispersed volumes are based on erasure codes, providing space-efficient protection against disk or server failures.

Dispersed volumes an encoded fragment of the original file to each brick in a way that only a subset of the fragments is needed to recover the original file. The number of bricks that can be missing without losing access to data is configured by the administrator on volume creation time.

#### 7.97.2. distribute

Red Hat Virtualization 4.2 REST API Guide
Distributed volumes distributes files throughout the bricks in the volume.

Distributed volumes can be used where the requirement is to scale storage and the redundancy is either not important or is provided by other hardware/software layers.

**7.97.3. distributed_disperse**

Distributed dispersed volumes distribute files across dispersed subvolumes.

This has the same advantages of distribute replicate volumes, but using disperse to store the data into the bricks.

**7.97.4. distributed_replicate**

Distributed replicated volumes distributes files across replicated bricks in the volume.

Distributed replicated volumes can be used in environments where the requirement is to scale storage and high-reliability is critical. Distributed replicated volumes also offer improved read performance in most environments.

**7.97.5. distributed_stripe**

Distributed striped volumes stripe data across two or more nodes in the cluster.

Distributed striped volumes should be used where the requirement is to scale storage and in high concurrency environments accessing very large files is critical.

Note: With the introduction of Sharding in Glusterfs 3.7 releases, striped volumes are not recommended and it will be removed in future release.

**7.97.6. distributed_striped_replicate**

Distributed striped replicated volumes distributes striped data across replicated bricks in the cluster.

For best results, distributed striped replicated volumes should be used in highly concurrent environments where parallel access of very large files and performance is critical.

Note: With the introduction of Sharding in Glusterfs 3.7 releases, striped volumes are not recommended and it will be removed in future release.

**7.97.7. replicate**

Replicated volumes replicates files across bricks in the volume.

Replicated volumes can be used in environments where high-availability and high-reliability are critical.

**7.97.8. stripe**

Striped volumes stripes data across bricks in the volume.

For best results, striped volumes should only in high concurrency environments accessing very large files.

Note: With the introduction of Sharding in Glusterfs 3.7 releases, striped volumes are not recommended and it will be removed in future release.
7.97.9. striped_replicate

Striped replicated volumes stripes data across replicated bricks in the cluster.

For best results, striped replicated volumes should be used in highly concurrent environments where there is parallel access of very large files and performance is critical.

Note: With the introduction of Sharding in Glusterfs 3.7 releases, striped volumes are not recommended and it will be removed in future release.

7.98. GRACEPERIOD STRUCT

Table 7.135. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>expiry</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

7.99. GRAPHICSConSOLE STRUCT

Table 7.136. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>protocol</td>
<td>GraphicsType</td>
<td></td>
</tr>
<tr>
<td>tls_port</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.137. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td></td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td></td>
</tr>
</tbody>
</table>
### 7.100. GRAPHICSTYPE ENUM

The graphics protocol used to connect to the graphic console.

**Table 7.138. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>spice</td>
<td>Graphics protocol of type SPICE.</td>
</tr>
<tr>
<td>vnc</td>
<td>Graphics protocol of type VNC.</td>
</tr>
</tbody>
</table>

### 7.100.1. spice

Graphics protocol of type SPICE. See [https://www.spice-space.org](https://www.spice-space.org) for more details.

### 7.100.2. vnc

Graphics protocol of type VNC. VNC stands for Virtual Network Computing, and it is a graphical desktop sharing system that uses RFB (Remote Frame Buffer) protocol to remotely control another machine.

### 7.101. GROUP STRUCT

This type represents all groups in the directory service.

**Table 7.139. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>domain_entry_id</td>
<td>String</td>
<td>The containing directory service domain id.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>namespace</td>
<td>String</td>
<td>Namespace where group resides.</td>
</tr>
</tbody>
</table>

**Table 7.140. Links summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>Domain</td>
<td>A link to the domain containing this group.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>A link to the permissions sub-collection for permissions attached to this group.</td>
</tr>
<tr>
<td>roles</td>
<td>Role[]</td>
<td>A link to the roles sub-collection for roles attached to this group.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>A link to the tags sub-collection for tags attached to this group.</td>
</tr>
</tbody>
</table>

### 7.101.1. roles

A link to the roles sub-collection for roles attached to this group.

Used only to represent the initial role assignments for a new group; thereafter, modification of role assignments is only supported via the `roles` sub-collection.

### 7.102. GUESTOPERATINGSYSTEM STRUCT

Represents an operating system installed on the virtual machine.

To get that information send a request like this:

```plaintext
GET /ovirt-engine/api/vms/123
```

The result will be like this:

```xml
<vm href="/ovirt-engine/api/vms/123" id="123">
  ...
  <guest_operating_system>
    <architecture>x86_64</architecture>
    <codename>Maipo</codename>
    <distribution>Red Hat Enterprise Linux Server</distribution>
    <family>Linux</family>
    <kernel>
      <version>
        <build>0</build>
        <full_version>3.10.0-514.10.2.el7.x86_64</full_version>
        <major>3</major>
        <minor>10</minor>
        <revision>514</revision>
      </version>
    </kernel>
    <version>
      <full_version>7.3</full_version>
      <major>7</major>
      <minor>3</minor>
    </version>
  </guest_operating_system>
</vm>
```
Table 7.141. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>String</td>
<td>The architecture of the operating system, such as x86_64.</td>
</tr>
<tr>
<td>codename</td>
<td>String</td>
<td>Code name of the operating system, such as Maipo.</td>
</tr>
<tr>
<td>distribution</td>
<td>String</td>
<td>Full name of operating system distribution.</td>
</tr>
<tr>
<td>family</td>
<td>String</td>
<td>Family of operating system, such as Linux.</td>
</tr>
<tr>
<td>kernel</td>
<td>Kernel</td>
<td>Kernel version of the operating system.</td>
</tr>
<tr>
<td>version</td>
<td>Version</td>
<td>Version of the installed operating system.</td>
</tr>
</tbody>
</table>

7.103. HARDWAREINFORMATION STRUCT

Represents hardware information of host.

To get that information send a request like this:

```
GET /ovirt-engine/api/hosts/123
```

The result will be like this:

```xml
<host href="/ovirt-engine/api/hosts/123" id="123">
  ...
  <hardware_information>
    <family>Red Hat Enterprise Linux</family>
    <manufacturer>Red Hat</manufacturer>
    <product_name>RHEV Hypervisor</product_name>
    <serial_number>01234567-89AB-CDEF-0123-456789ABCDEF</serial_number>
    <supported_rng_sources>
      <supported_rng_source>random</supported_rng_source>
    </supported_rng_sources>
    <uuid>12345678-9ABC-DEF0-1234-56789ABCDEF</uuid>
    <version>1.2-34.5.el7ev</version>
  </hardware_information>
  ...
</host>
```

Table 7.142. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>family</td>
<td>String</td>
<td>Type of host's CPU.</td>
</tr>
<tr>
<td>manufacturer</td>
<td>String</td>
<td>Manufacturer of the host's machine and hardware vendor.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>product_name</td>
<td>String</td>
<td>Host's product name (for example RHEV Hypervisor).</td>
</tr>
<tr>
<td>serial_number</td>
<td>String</td>
<td>Unique ID for host's chassis.</td>
</tr>
<tr>
<td>supported_rng_sources</td>
<td>RngSource[]</td>
<td>Supported sources of random number generator.</td>
</tr>
<tr>
<td>uuid</td>
<td>String</td>
<td>Unique ID for each host.</td>
</tr>
<tr>
<td>version</td>
<td>String</td>
<td>Unique name for each of the manufacturer.</td>
</tr>
</tbody>
</table>

### 7.104. HIGHAVAILABILITY STRUCT

Type representing high availability of a virtual machine.

#### Table 7.143. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Define if the virtual machine should be consider highly available.</td>
</tr>
<tr>
<td>priority</td>
<td>Integer</td>
<td>Indicates the priority of the virtual machine inside the run and migration queues.</td>
</tr>
</tbody>
</table>

#### 7.104.1. priority

Indicates the priority of the virtual machine inside the run and migration queues.

Virtual machines with higher priorities will be started and migrated before virtual machines with lower priorities.

The value is an integer between 0 and 100. The higher the value, the higher the priority.

The graphical user interface (GUI) does not allow specifying all the possible values, instead it only allows you to select Low, Medium or High. When the value is set using the API, the GUI will set the label as follows:

<table>
<thead>
<tr>
<th>API Value</th>
<th>GUI Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 25</td>
<td>Low</td>
</tr>
<tr>
<td>26 - 74</td>
<td>Medium</td>
</tr>
<tr>
<td>75 - 100</td>
<td>High</td>
</tr>
</tbody>
</table>
When the label is selected using the GUI, the value in the API will be set as follows:

<table>
<thead>
<tr>
<th>GUI Label</th>
<th>API Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>1</td>
</tr>
<tr>
<td>Medium</td>
<td>50</td>
</tr>
<tr>
<td>High</td>
<td>100</td>
</tr>
</tbody>
</table>

### 7.105. HOOK STRUCT

Represents a hook.

**Table 7.144. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>event_name</td>
<td>String</td>
<td>Name of the event to execute the hook on.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>md5</td>
<td>String</td>
<td>Checksum of the hook.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

**Table 7.145. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host the hook belongs to.</td>
</tr>
</tbody>
</table>

### 7.106. HOOKCONTENTTYPE ENUM

Represents content type of hook script.

**Table 7.146. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary</td>
<td>Binary content type of the hook.</td>
</tr>
<tr>
<td>text</td>
<td>Text content type of the hook.</td>
</tr>
</tbody>
</table>
7.107. HOOKSTAGE ENUM

Type represents a stage of volume event at which hook executes.

Table 7.147. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>post</td>
<td>Stage after start of volume.</td>
</tr>
<tr>
<td>pre</td>
<td>Stage before start of volume.</td>
</tr>
</tbody>
</table>

7.108. HOOKSTATUS ENUM

Type represents the status of a hook.

Table 7.148. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>Hook is disabled.</td>
</tr>
<tr>
<td>enabled</td>
<td>Hook is enabled.</td>
</tr>
<tr>
<td>missing</td>
<td>Hook is missing.</td>
</tr>
</tbody>
</table>

7.109. HOST STRUCT

Type representing a host.

Table 7.149. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>The host address (FQDN/IP).</td>
</tr>
<tr>
<td>auto_numa_status</td>
<td>AutoNumaStatus</td>
<td>The host auto non uniform memory access (NUMA) status.</td>
</tr>
<tr>
<td>certificate</td>
<td>Certificate</td>
<td>The host certificate.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The CPU type of this host.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>device_passthrough</td>
<td>HostDevicePassThrough</td>
<td>Specifies whether host device passthrough is enabled on this host.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>Optionally specify the display address of this host explicitly.</td>
</tr>
<tr>
<td>external_status</td>
<td>ExternalStatus</td>
<td>The host external status.</td>
</tr>
<tr>
<td>hardware_information</td>
<td>HardwareInformation</td>
<td>The host hardware information.</td>
</tr>
<tr>
<td>hosted_engine</td>
<td>HostedEngine</td>
<td>The self-hosted engine status of this host.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>iscsi</td>
<td>IscsiDetails</td>
<td>The host iSCSI details.</td>
</tr>
<tr>
<td>kdump_status</td>
<td>KdumpStatus</td>
<td>The host KDUMP status.</td>
</tr>
<tr>
<td>ksm</td>
<td>Ksm</td>
<td>Kernel SamePage Merging (KSM) reduces references to memory pages from multiple identical pages to a single page reference.</td>
</tr>
<tr>
<td>libvirt_version</td>
<td>Version</td>
<td>The host libvirt version.</td>
</tr>
<tr>
<td>max_scheduling_memory</td>
<td>Integer</td>
<td>The max scheduling memory on this host in bytes.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The amount of physical memory on this host in bytes.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>network_operation_in_progress</td>
<td>Boolean</td>
<td>Specifies whether a network-related operation, such as 'setup networks' or 'sync networks', is currently being executed on this host.</td>
</tr>
<tr>
<td>numa_supported</td>
<td>Boolean</td>
<td>Specifies whether non uniform memory access (NUMA) is supported on this host.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>The operating system on this host.</td>
</tr>
<tr>
<td>override_iptables</td>
<td>Boolean</td>
<td>Specifies whether we should override firewall definitions.</td>
</tr>
</tbody>
</table>
### Name | Type | Summary
--- | --- | ---
| **port** | Integer | The host port. |
| **power_management** | PowerManagement | The host power management definitions. |
| **protocol** | HostProtocol | The protocol that the engine uses to communicate with the host. |
| **root_password** | String | When creating a new host, a root password is required if the password authentication method is chosen, but this is not subsequently included in the representation. |
| **se_linux** | SeLinux | The host SElinux status. |
| **spm** | Spm | The host *storage pool manager* (SPM) status and definition. |
| **ssh** | Ssh | The SSH definitions. |
| **status** | HostStatus | The host status. |
| **status_detail** | String | The host status details. |
| **summary** | VmSummary | The virtual machine summary - how many are active, migrating and total. |
| **transparent_huge_pages** | TransparentHugePages | Transparent huge page support expands the size of memory pages beyond the standard 4 KiB limit. |
| **type** | HostType | Indicates if the host contains a full installation of the operating system or a scaled-down version intended only to host virtual machines. |
| **update_available** | Boolean | Specifies whether there is an oVirt-related update on this host. |
| **version** | Version | The version of VDSM. |

#### 7.109.1. external_status

The host external status. This can be used by third-party software to change the host external status in case of an issue. This has no effect on the host lifecycle, unless a third-party software checks for this status and acts accordingly.

#### 7.109.2. hosted_engine

The self-hosted engine status of this host.
7.109.3. kdump_status

The host KDUMP status. KDUMP happens when the host kernel has crashed and it is now going through memory dumping.

7.109.4. ksm

Kernel SamePage Merging (KSM) reduces references to memory pages from multiple identical pages to a single page reference. This helps with optimization for memory density.

For example, to enable KSM for host 123, send a request like this:

```
PUT /ovirt-engine/api/hosts/123
```

With a request body like this:

```
<host>
  <ksm>
    <enabled>true</enabled>
  </ksm>
</host>
```

7.109.5. libvirt_version

The host libvirt version. For more information on libvirt please go to libvirt.

7.109.6. override_iptables

Specifies whether we should override firewall definitions. This applies only when the host is installed or re-installed.

7.109.7. protocol

The protocol that the engine uses to communicate with the host.

---

**WARNING**

Since version 4.1 of the engine the protocol is always set to stomp since xml was removed.

7.109.8. se_linux
The host SElinux status. *Security-Enhanced Linux (SELinux)* is a component in the Linux kernel that provides a mechanism for supporting access control security policies.

### 7.109.9. spm

The host *storage pool manager* (SPM) status and definition. Use it to set the SPM priority of this host, and to see whether this is the current SPM or not.

### 7.109.10. status_detail

The host status details. Relevant for Gluster hosts.

### 7.109.11. transparent_huge_pages

Transparent huge page support expands the size of memory pages beyond the standard 4 KiB limit. This reduces memory consumption and increases host performance.

For example, to enable transparent huge page support for host 123, send a request like this:

```
PUT /ovirt-engine/api/hosts/123
```

With a request body like this:

```
<host>
  <transparent_hugepages>
    <enabled>true</enabled>
  </transparent_hugepages>
</host>
```

### 7.109.12. version

The version of VDSM.

For example:

```
GET /ovirt-engine/api/hosts/123
```

This `GET` request will return the following output:

```
<host>
  ...
  <version>
    <build>999</build>
    <full_version>vdsd-4.18.999-419.gitcf06367.el7</full_version>
    <major>4</major>
    <minor>18</minor>
    <revision>0</revision>
  </version>
  ...
</host>
```

**Table 7.150. Links summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity_labels</td>
<td>AffinityLabel[]</td>
<td></td>
</tr>
<tr>
<td>agents</td>
<td>Agent[]</td>
<td></td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>devices</td>
<td>Device[]</td>
<td></td>
</tr>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td>External network providers provisioned on the host.</td>
</tr>
<tr>
<td>external_network_provider_configurations</td>
<td>ExternalNetworkProviderConfiguration[]</td>
<td></td>
</tr>
<tr>
<td>hooks</td>
<td>Hook[]</td>
<td></td>
</tr>
<tr>
<td>katello_errata</td>
<td>KatelloErratum[]</td>
<td>Lists all the Katello errata assigned to the host.</td>
</tr>
<tr>
<td>network_attachments</td>
<td>NetworkAttachment[]</td>
<td></td>
</tr>
<tr>
<td>nics</td>
<td>HostNic[]</td>
<td></td>
</tr>
<tr>
<td>numa_nodes</td>
<td>NumaNode[]</td>
<td></td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Each host resource exposes a statistics sub-collection for host-specific statistics.</td>
</tr>
<tr>
<td>storage_connection_extensions</td>
<td>StorageConnectionExtension[]</td>
<td></td>
</tr>
<tr>
<td>storages</td>
<td>HostStorage[]</td>
<td></td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td></td>
</tr>
<tr>
<td>unmanaged_networks</td>
<td>UnmanagedNetwork[]</td>
<td></td>
</tr>
</tbody>
</table>

7.109.13. external_network_provider_configurations
External network providers provisioned on the host.

External network providers on the host can be controlled when adding the host.

7.109.14. katello_errata

Lists all the Katello errata assigned to the host.

GET /ovirt-engine/api/hosts/123/katelloerrata

You will receive response in XML like this one:

```xml
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

7.109.15. statistics

Each host resource exposes a statistics sub-collection for host-specific statistics.

An example of an XML representation:

```xml
<statistics>
  <statistic href="/ovirt-engine/api/hosts/123/statistics/456" id="456">
    <name>memory.total</name>
    <description>Total memory</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>25165824000</datum>
      </value>
    </values>
  </statistic>
  ...
</statistics>
```
NOTE
This statistics sub-collection is read-only.

The following list shows the statistic types for hosts:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory.total</td>
<td>Total memory in bytes on the host.</td>
</tr>
<tr>
<td>memory.used</td>
<td>Memory in bytes used on the host.</td>
</tr>
<tr>
<td>memory.free</td>
<td>Memory in bytes free on the host.</td>
</tr>
<tr>
<td>memory.shared</td>
<td>Memory in bytes shared on the host.</td>
</tr>
<tr>
<td>memory.buffers</td>
<td>I/O buffers in bytes.</td>
</tr>
<tr>
<td>memory.cached</td>
<td>OS caches in bytes.</td>
</tr>
<tr>
<td>swap.total</td>
<td>Total swap memory in bytes on the host.</td>
</tr>
<tr>
<td>swap.free</td>
<td>Swap memory in bytes free on the host.</td>
</tr>
<tr>
<td>swap.used</td>
<td>Swap memory in bytes used on the host.</td>
</tr>
<tr>
<td>swap.cached</td>
<td>Swap memory in bytes also cached in host's memory.</td>
</tr>
<tr>
<td>ksm.cpu.current</td>
<td>Percentage of CPU usage for Kernel SamePage Merging.</td>
</tr>
<tr>
<td>cpu.current.user</td>
<td>Percentage of CPU usage for user slice.</td>
</tr>
<tr>
<td>cpu.current.system</td>
<td>Percentage of CPU usage for system.</td>
</tr>
<tr>
<td>cpu.current.idle</td>
<td>Percentage of idle CPU usage.</td>
</tr>
<tr>
<td>cpu.load.avg.5m</td>
<td>CPU load average per five minutes.</td>
</tr>
<tr>
<td>boot.time</td>
<td>Boot time of the machine.</td>
</tr>
</tbody>
</table>

7.110. HOSTDEVICE STRUCT

Table 7.151. Attributes summary
### 7.110.1. driver

The name of the driver this device is bound to.

For example: `pcieport` or `uhci_hcd`.

#### Table 7.152. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>parent_device</td>
<td>HostDevice</td>
<td></td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td></td>
</tr>
</tbody>
</table>

### 7.111. HOSTDEVICEPASSTHROUGH STRUCT
Table 7.153. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>

7.112. HOSTNIC STRUCT

Represents a host NIC.

For example, the XML representation of a host NIC looks like this:

```xml
<host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456">
  <name>eth0</name>
  <boot_protocol>static</boot_protocol>
  <bridged>true</bridged>
  <custom_configuration>true</custom_configuration>
  <ip>
    <address>192.168.122.39</address>
    <gateway>192.168.122.1</gateway>
    <netmask>255.255.255.0</netmask>
    <version>v4</version>
  </ip>
  <ipv6>
    <gateway>::</gateway>
    <version>v6</version>
  </ipv6>
  <ipv6_boot_protocol>none</ipv6_boot_protocol>
  <mac>
    <address>52:54:00:0c:79:1d</address>
  </mac>
  <mtu>1500</mtu>
  <status>up</status>
</host_nic>
```

A bonded interface is represented as a HostNic object containing the bonding and slaves attributes.

For example, the XML representation of a bonded host NIC looks like this:

```xml
<host_nic href="/ovirt-engine/api/hosts/123/nics/456" id="456">
  <name>bond0</name>
  <mac_address="00:00:00:00:00:00"/>
  <ip>
    <address>192.168.122.39</address>
    <gateway>192.168.122.1</gateway>
    <netmask>255.255.255.0</netmask>
    <version>v4</version>
  </ip>
  <boot_protocol>dhcp</boot_protocol>
  <bonding>
    <options>
      <option>
        <name>mode</name>
        <value>4</value>
      </option>
    </options>
  </bonding>
</host_nic>
```
Table 7.154. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad_aggregator_id</td>
<td>Integer</td>
<td>The <code>ad_aggregator_id</code> property of a bond or bond slave, for bonds in mode 4.</td>
</tr>
<tr>
<td>base_interface</td>
<td>String</td>
<td>The base interface of the NIC.</td>
</tr>
<tr>
<td>bonding</td>
<td>Bonding</td>
<td>The bonding parameters of the NIC.</td>
</tr>
<tr>
<td>boot_protocol</td>
<td>BootProtocol</td>
<td>The IPv4 boot protocol configuration of the NIC.</td>
</tr>
<tr>
<td>bridged</td>
<td>Boolean</td>
<td>Defines the bridged network status.</td>
</tr>
<tr>
<td>check_connectivity</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>custom_configuration</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ip</td>
<td>Ip</td>
<td>The IPv4 address of the NIC.</td>
</tr>
<tr>
<td>ipv6</td>
<td>Ip</td>
<td>The IPv6 address of the NIC.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ipv6_boot_protocol</td>
<td>BootProtocol</td>
<td>The IPv6 boot protocol configuration of the NIC.</td>
</tr>
<tr>
<td>mac</td>
<td>Mac</td>
<td>The MAC address of the NIC.</td>
</tr>
<tr>
<td>mtu</td>
<td>Integer</td>
<td>The maximum transmission unit for the interface.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>network_labels</td>
<td>NetworkLabel[]</td>
<td>The labels that are applied to this NIC.</td>
</tr>
<tr>
<td>override_configuration</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td></td>
</tr>
<tr>
<td>speed</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>A link to the statistics of the NIC.</td>
</tr>
<tr>
<td>status</td>
<td>NicStatus</td>
<td></td>
</tr>
<tr>
<td>virtual_functions_configuration</td>
<td>HostNicVirtualFunctionsConfiguration</td>
<td>Describes the virtual functions configuration of a physical function NIC.</td>
</tr>
<tr>
<td>vlan</td>
<td>Vlan</td>
<td></td>
</tr>
</tbody>
</table>

### 7.112.1. ad_aggregator_id

The ad_aggregator_id property of a bond or bond slave, for bonds in mode 4. Bond mode 4 is the 802.3ad standard, also called dynamic link aggregation. (See Wikipedia and Presentation for more information). This is only valid for bonds in mode 4, or NICs which are part of a bond. It is not present for bonds in other modes, or NICs which are not part of a bond in mode 4. The ad_aggregator_id property indicates which of the bond slaves are active. The value of the ad_aggregator_id of an active slave is the same as the value of the ad_aggregator_id property of the bond. This parameter is read only. Setting it will have no effect on the bond/NIC. It is retrieved from the /sys/class/net/bondX/bonding/ad_aggregator file for a bond, and the /sys/class/net/ensX/bonding_slave/ad_aggregator_id file for a NIC.

### 7.112.2. bridged

Defines the bridged network status. Set to true for a bridged network and false for a bridgeless network.
7.112.3. statistics

A link to the statistics of the NIC.

The data types for HostNic statistical values:

- `data.current.rx` - The rate in bytes per second of data received.
- `data.current.tx` - The rate in bytes per second of data transmitted.
- `data.current.rx.bps` - The rate in bits per second of data received (since version 4.2).
- `data.current.tx.bps` - The rate in bits per second of data transmitted (since version 4.2).
- `data.total.rx` - Total received data.
- `data.total.tx` - Total transmitted data.
- `errors.total.rx` - Total errors from receiving data.
- `errors.total.tx` - Total errors from transmitting data.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>A reference to the network to which the interface should be connected.</td>
</tr>
<tr>
<td>physical_function</td>
<td>HostNic</td>
<td>A reference to the physical function NIC of a SR-IOV virtual function NIC.</td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td>A link to the quality-of-service configuration of the interface.</td>
</tr>
</tbody>
</table>

7.112.4. network

A reference to the network to which the interface should be connected. A blank network ID is allowed.

7.113. HOSTNICVIRTUALFUNCTIONSCONFIGURATION STRUCT

Describes the virtual functions configuration of an SR-IOV-enabled physical function NIC.

Table 7.156. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>all_networks_allowed</td>
<td>Boolean</td>
<td>Defines whether all networks are allowed to be defined on the related virtual functions, or specified ones only.</td>
</tr>
</tbody>
</table>
### 7.113.1. max_number_of_virtual_functions

The maximum number of virtual functions the NIC supports. This property is read-only.

### 7.113.2. number_of_virtual_functions

The number of virtual functions currently defined. A user-defined value between 0 and max_number_of_virtual_functions.

### 7.114. HOSTPROTOCOL ENUM

The protocol used by the engine to communicate with a host.

**WARNING**

Since version 4.1 of the engine the protocol is always set to **stomp** since **xml** was removed.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>stomp</strong></td>
<td>JSON-RPC protocol on top of STOMP.</td>
</tr>
<tr>
<td><strong>xml</strong></td>
<td>XML-RPC protocol.</td>
</tr>
</tbody>
</table>

### 7.115. HOSTSTATUS ENUM

Type representing a host status.

**Table 7.157. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.158. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>connecting</td>
<td>The engine cannot communicate with the host for a specific threshold so it is now trying to connect before going through fencing.</td>
</tr>
<tr>
<td>down</td>
<td>The host is down.</td>
</tr>
<tr>
<td>error</td>
<td>The host is in error status.</td>
</tr>
<tr>
<td>initializing</td>
<td>The host is initializing.</td>
</tr>
<tr>
<td>install_failed</td>
<td>The host installation failed.</td>
</tr>
<tr>
<td>installing</td>
<td>The host is being installed.</td>
</tr>
<tr>
<td>installing_os</td>
<td>The host operating system is now installing.</td>
</tr>
<tr>
<td>kdumping</td>
<td>The host kernel has crashed and it is now going through memory dumping.</td>
</tr>
<tr>
<td>maintenance</td>
<td>The host is in maintenance status.</td>
</tr>
<tr>
<td>non_operational</td>
<td>The host is non operational.</td>
</tr>
<tr>
<td>non_responsive</td>
<td>The host is not responsive.</td>
</tr>
<tr>
<td>pending_approval</td>
<td>The host is pending administrator approval.</td>
</tr>
<tr>
<td>preparing_for_maintenance</td>
<td>The host is preparing for maintenance.</td>
</tr>
<tr>
<td>reboot</td>
<td>The host is being rebooted.</td>
</tr>
<tr>
<td>unassigned</td>
<td>The host is in activation process.</td>
</tr>
<tr>
<td>up</td>
<td>The host is up.</td>
</tr>
</tbody>
</table>

### 7.115.1. error

The host is in error status. This will happen if we will try to run a virtual machine several times and it will fail.

### 7.115.2. initializing
The host is initializing. This is an intermediate step before moving the host to 'up' status.

7.115.3. install_failed

The host installation failed. In such cases look at the event log to understand what failed the installation, and issue a re-install.

7.115.4. installing_os

The host operating system is now installing. This status is relevant when using a Satellite/Foreman provider, and issuing a bare-metal provisioning (discovered host provisioning).

7.115.5. maintenance

The host is in maintenance status. When a host is in maintenance it cannot run virtual machines.

7.115.6. non_operational

The host is non operational. This can happen due to various reasons, such as not having a connection with the storage, not supporting a mandatory network, not supporting the cluster level, and more.

7.115.7. non_responsive

The host is not responsive. This means that the engine is not able to communicate with the host.

7.115.8. pending_approval

The host is pending administrator approval. This is relevant only for vintage ovirt-node / RHV-H.

7.115.9. preparing_for_maintenance

The host is preparing for maintenance. During this time the engine makes sure to live migrate all the virtual machines from this host to other hosts. Once all migrations have been completed the host will move to 'maintenance' status.

7.116. HOSTSTORAGE STRUCT

Table 7.159. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
</tbody>
</table>
### Name | Type | Summary
--- | --- | ---
logical_units | LogicalUnit[] |  
mount_options | String |  
name | String | A human-readable name in plain text.  
nfs_retrans | Integer | The number of times to retry a request before attempting further recovery actions.  
nfs_timeo | Integer | The time in tenths of a second to wait for a response before retrying NFS requests.  
nfs_version | NfsVersion |  
override_luns | Boolean |  
password | String |  
path | String |  
port | Integer |  
portal | String |  
target | String |  
type | StorageType |  
username | String |  
vfs_type | String |  
volume_group | VolumeGroup |  

#### 7.116.1. nfs_retrans

The number of times to retry a request before attempting further recovery actions. The value must be in the range of 0 to 65535. For more details see the description of the `retrans` mount option in the `nfs` man page.

#### 7.116.2. nfs_timeo
The time in tenths of a second to wait for a response before retrying NFS requests. The value must be in the range of 0 to 65535. For more details see the description of the `timeo` mount option in the `nfs` man page.

### Table 7.160. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
</tr>
</tbody>
</table>

#### 7.117. HOSTTYPE ENUM

This enumerated type is used to determine which type of operating system is used by the host.

### Table 7.161. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ovirt_node</td>
<td>The host contains Red Hat Virtualization Host (RHVH): a new implementation of Red Hat Enterprise Virtualization Hypervisor (RHEV-H) which uses the same installer as Red Hat Enterprise Linux, CentOS, or Fedora.</td>
</tr>
<tr>
<td>rhel</td>
<td>The host contains a full Red Hat Enterprise Linux, CentOS, or Fedora installation.</td>
</tr>
<tr>
<td>rhev_h</td>
<td>The host contains Red Hat Enterprise Virtualization Hypervisor (RHEV-H), a small-scaled version of Red Hat Enterprise Linux, CentOS, or Fedora, used solely to host virtual machines.</td>
</tr>
</tbody>
</table>

#### 7.117.1. ovirt_node

The host contains Red Hat Virtualization Host (RHVH): a new implementation of Red Hat Enterprise Virtualization Hypervisor (RHEV-H) which uses the same installer as Red Hat Enterprise Linux, CentOS, or Fedora. The main difference between RHVH and legacy RHEV-H is that RHVH has a writeable file system and will handle its own installation instead of having RPMs pushed to it by the Manager like in legacy RHEV-H.

#### 7.118. HOSTEDENGINE STRUCT

### Table 7.162. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>configured</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>global_maint</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>enance</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>
### 7.119. ICON STRUCT

Icon of virtual machine or template.

**Table 7.163. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>data</td>
<td>String</td>
<td>Base64 encode content of the icon file.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>media_type</td>
<td>String</td>
<td>Format of icon file.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

#### 7.119.1. media_type

Format of icon file.

One of:

- image/jpeg
- image/png
- image/gif

### 7.120. IDENTIFIED STRUCT

This interface is the base model for all types that represent objects with an identifier.

**Table 7.164. Attributes summary**
### 7.121. IMAGE STRUCT

Represents an image entity.

**Table 7.165. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>size</td>
<td>Integer</td>
<td>The size of the image file.</td>
</tr>
<tr>
<td>type</td>
<td>ImageFileType</td>
<td>The type of the image file.</td>
</tr>
</tbody>
</table>

**Table 7.166. Links summary**

| Name         | Type                   | Summary                                                      |
|--------------|                       |                                                              |
| storage_doma | StorageDomain         | The storage domain associated with this image.              |

### 7.122. IMAGEFILETYPE ENUM

Represents the file type of an image.

**Table 7.167. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>The image is a disk format that can be used as a virtual machine’s disk.</td>
</tr>
</tbody>
</table>
The image is a floppy disk that can be attached to a virtual machine, for example to install the VirtIO drivers in Windows.

The image is a `.iso` file that can be used as a CD-ROM to boot and install a virtual machine.

**7.123. IMAGETRANSFER STRUCT**

This type contains information regarding an image transfer being performed.

**Table 7.168. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Boolean</td>
<td>Indicates whether there’s at least one active session for this transfer, i.e. there’s at least one live transfer session between the client and the daemon.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>direction</td>
<td>ImageTransferDirection</td>
<td>The direction indicates whether the transfer is sending image data (upload) or receiving image data (download).</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>inactivity_timeout</td>
<td>Integer</td>
<td>The timeout in seconds of client inactivity, after which the transfer is aborted by the Red Hat Virtualization Manager.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>phase</td>
<td>ImageTransferPhase</td>
<td>The current phase of the image transfer in progress.</td>
</tr>
<tr>
<td>proxy_url</td>
<td>String</td>
<td>The URL of the proxy server that the user inputs or outputs to.</td>
</tr>
<tr>
<td>signed_ticket</td>
<td>String</td>
<td>The signed ticket that should be attached as an Authentication header in the HTTPS request for the proxy server to input or output to (See the proxy_url attribute).</td>
</tr>
<tr>
<td>transfer_url</td>
<td>String</td>
<td>The URL of the daemon server that the user can input or output to directly.</td>
</tr>
</tbody>
</table>
transferred

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>transferred</td>
<td>Integer</td>
<td>Indicates the amount of transferred bytes.</td>
</tr>
</tbody>
</table>

### 7.123.1. direction

The direction indicates whether the transfer is sending image data (**upload**) or receiving image data (**download**).

If a direction is not set during an addition of a new transfer, the default direction for the transfer will be **upload**.

### 7.123.2. inactivity_timeout

The timeout in seconds of client inactivity, after which the transfer is aborted by the Red Hat Virtualization Manager. To disable the inactivity timeout specify ‘0’. If not specified, the value is defaulted to the **engine-config** value: TransferImageClientInactivityTimeoutInSeconds.

### 7.123.3. phase

The current phase of the image transfer in progress. Each transfer needs a managed session, which must be opened for the user to input or output an image. Please refer to **image transfer** for further documentation.

### 7.123.4. proxy_url

The URL of the proxy server that the user inputs or outputs to. This attribute is available only if the image transfer is in the **transferring** phase. See **phase** for details.

### 7.123.5. transfer_url

The URL of the daemon server that the user can input or output to directly.

This is as an alternative to the **proxy_url**. I.e. if the client has access to the host machine, it could bypass the proxy and transfer directly to the host, potentially improving the throughput performance. This attribute is available only if the image transfer is in the **transferring** phase. See **phase** for details.

#### Table 7.169. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk</td>
<td>Disk</td>
<td>The disk which is targeted for input or output.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>The host which will be used to write to the image which is targeted for input or output.</td>
</tr>
<tr>
<td>image</td>
<td>Image</td>
<td>The image which is targeted for input or output.</td>
</tr>
<tr>
<td>snapshot</td>
<td>DiskSnapshot</td>
<td>The disk snapshot which is targeted for input or output.</td>
</tr>
</tbody>
</table>
7.123.6. host

The host which will be used to write to the image which is targeted for input or output. If not specified, an active host will be randomly selected from the data center.

7.123.7. image

The image which is targeted for input or output.

**IMPORTANT**

This attribute is deprecated since version 4.2 of the engine. Use the `disk` or `snapshot` attributes instead.

7.124. IMAGETRANSFERDIRECTION ENUM

The image transfer direction for a transfer.

When adding a new transfer, the user can choose whether the transfer will be to an image, choosing `upload`, or to transfer from an image- choosing `download` as an ImageTransferDirection.

Please refer to image transfer for further documentation.

**Table 7.170. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>download</td>
<td>The user must choose <code>download</code> when he/she wants to stream data from an image.</td>
</tr>
<tr>
<td>upload</td>
<td>The user can choose <code>upload</code> when he/she wants to stream data to an image.</td>
</tr>
</tbody>
</table>

7.125. IMAGETRANSFERPHASE ENUM

A list of possible phases for an image transfer entity. Each of these values defines a specific point in a transfer flow.

Please refer to image transfer for more information.

**Table 7.171. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cancelled</td>
<td>This phase will be set as a result of the user cancelling the transfer.</td>
</tr>
<tr>
<td>finalizing_failure</td>
<td>This phase can only be set in the Administration Portal, and indicates that there was an error during the transfer, and it is being finalized with a failure.</td>
</tr>
<tr>
<td>finalizing_success</td>
<td>This phase will be set when the user calls <code>finalize</code>.</td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>finished_failure</td>
<td>Indicates that the targeted image failed the verification, and cannot be used.</td>
</tr>
<tr>
<td>finished_success</td>
<td>Indicates that the transfer session was successfully closed, and the targeted image was verified and ready to be used.</td>
</tr>
<tr>
<td>initializing</td>
<td>The initial phase of an image transfer.</td>
</tr>
<tr>
<td>paused_system</td>
<td>This phase means the session timed out, or some other error occurred with this transfer; for example ovirt-imageio-daemon is not running in the selected host.</td>
</tr>
<tr>
<td>paused_user</td>
<td>This phase is a result of a pause call by the user, using <code>pause</code>.</td>
</tr>
<tr>
<td>resuming</td>
<td>The phase where the transfer has been resumed by the client calling <code>resume</code>.</td>
</tr>
<tr>
<td>transferring</td>
<td>The phase where the transfer session is open, and the client can input or output the desired image using the preferred tools.</td>
</tr>
<tr>
<td>unknown</td>
<td>An unknown phase.</td>
</tr>
</tbody>
</table>

### 7.125.1. cancelled

This phase will be set as a result of the user cancelling the transfer. The cancellation can only be performed in the Administration Portal.

### 7.125.2. finalizing_success

This phase will be set when the user calls `finalize`. Calling finalize is essential to finish the transfer session, and finish using the targeted image. After finalizing, the phase will be changed to `finished_success` or `finished_failure`.

Refer to image transfer for more information.

### 7.125.3. finished_failure

Indicates that the targeted image failed the verification, and cannot be used. After reaching this phase, the image transfer entity will be deleted, and the targeted image will be set to illegal.

### 7.125.4. finished_success

Indicates that the transfer session was successfully closed, and the targeted image was verified and ready to be used. After reaching this phase, the image transfer entity will be deleted.

### 7.125.5. initializing

The initial phase of an image transfer. It is set while the transfer session is establishing. Once the session is established, the phase will be changed to `transferring`
7.125.6. paused_system

This phase means the session timed out, or some other error occurred with this transfer; for example ovirt-imageio-daemon is not running in the selected host. To resume the session, the client should call resume. After resuming, the phase will change to resuming.

7.125.7. resuming

The phase where the transfer has been resumed by the client calling resume. Resuming starts a new session, and after calling it, the phase will be changed to transferring, or paused_system in case of a failure.

7.125.8. unknown

An unknown phase. This will only be set in cases of unpredictable errors.

7.126. INHERITABLEBOOLEAN ENUM

Enum representing the boolean value that can be either set, or inherited from a higher level. The inheritance order is virtual machine → cluster → engine-config.

Table 7.172. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>Set the value to false on this level.</td>
</tr>
<tr>
<td>inherit</td>
<td>Inherit the value from higher level.</td>
</tr>
<tr>
<td>true</td>
<td>Set the value to true on this level.</td>
</tr>
</tbody>
</table>

7.127. INITIALIZATION STRUCT

Table 7.173. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active_directory_ou</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>authorized_ssh_keys</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>cloud_init</td>
<td>CloudInit</td>
<td>Deprecated attribute to specify cloud-init configuration.</td>
</tr>
<tr>
<td>configuration</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>custom_script</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>dns_search</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>dns_servers</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>host_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>input_locale</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>nic_configurations</td>
<td>NicConfiguration[]</td>
<td></td>
</tr>
<tr>
<td>org_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>regenerate_ids</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>regenerate_ssh_keys</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>root_password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>system_locale</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>timezone</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>ui_language</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>user_locale</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>user_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>windows_license_key</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.127.1. cloud_init

Deprecated attribute to specify `cloud-init` configuration.

This attribute, and the CloudInit type have been deprecated and will be removed in the future. To specify the `cloud-init` configuration, use the attributes inside the `Initialization` type. The mapping between the attributes of these two types are as follows:
CloudInit | Initialization
---|---
authorized_keys | authorized_ssh_keys

dns.search_domains | dns_search

dns.servers | dns_servers

files | custom_script

host | host_name

network_configuration.nics | nic_configurations

regenerate_ssh_keys | regenerate_ssh_keys

timezone | timezone

users | user_name & root_password

For more details on how to use `cloud-init` see the examples in Python, Ruby and Java.

7.128. INSTANCETYPE STRUCT

Describes the hardware configuration of virtual machines.

For example medium instance type includes 1 virtual CPU and 4 GiB of memory. It is a top-level entity (e.g. not bound to any data center or cluster). The attributes that are used for instance types and are common to virtual machine and template types are:

- console
- cpu
- custom_cpu_model
- custom_emulated_machine
- display
- high_availability
- io
- memory
- memory_policy
- migration
- migration_downtime
When creating a virtual machine from both an instance type and a template, the virtual machine will inherit the hardware configurations from the instance type.

**NOTE**

An instance type inherits its attributes from the template entity although most template attributes are not used in instance types.

### Table 7.174. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios</td>
<td>Bios</td>
<td>Reference to virtual machine’s BIOS configuration.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console</td>
<td>Console</td>
<td>Console configured for this virtual machine.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The configuration of the virtual machine CPU.</td>
</tr>
<tr>
<td>cpu_shares</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>creation_time</td>
<td>Date</td>
<td>The virtual machine creation date.</td>
</tr>
<tr>
<td>custom_compatibility_version</td>
<td>Version</td>
<td>Virtual machine custom compatibility version.</td>
</tr>
<tr>
<td>custom_cpu_model</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_emulated_machine</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Properties sent to VDSM to configure various hooks.</td>
</tr>
<tr>
<td>delete_protected</td>
<td>Boolean</td>
<td>If true, the virtual machine cannot be deleted.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The virtual machine display configuration.</td>
</tr>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Domain configured for this virtual machine.</td>
</tr>
<tr>
<td>high_availability</td>
<td>HighAvailability</td>
<td>The virtual machine high availability configuration.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialization</td>
<td>Reference to the virtual machine’s initialization configuration.</td>
</tr>
<tr>
<td>io</td>
<td>Io</td>
<td>For performance tuning of IO threading.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Virtual machine’s large icon.</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>Reference to the storage domain this virtual machine/template lease reside on.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The virtual machine’s memory, in bytes.</td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td>Reference to virtual machine’s memory management configuration.</td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td>Reference to configuration of migration of running virtual machine to another host.</td>
</tr>
<tr>
<td>migration_downtime</td>
<td>Integer</td>
<td>Maximum time the virtual machine can be non responsive during its live migration to another host in ms.</td>
</tr>
<tr>
<td>multi_queues_enabled</td>
<td>Boolean</td>
<td>If true, each virtual interface will get the optimal number of queues, depending on the available virtual CPUs.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>The origin of this virtual machine.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>Operating system type installed on the virtual machine.</td>
</tr>
<tr>
<td>placement_policy</td>
<td>VmPlacementPolicy</td>
<td>The configuration of the virtual machine’s placement policy.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>Random Number Generator device configuration for this virtual machine.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td>Virtual machine’s serial number in a cluster.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Virtual machine’s small icon.</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>If true, the sound card is added to the virtual machine.</td>
</tr>
<tr>
<td>sso</td>
<td>Sso</td>
<td>Reference to the Single Sign On configuration this virtual machine is configured for.</td>
</tr>
<tr>
<td>start_paused</td>
<td>Boolean</td>
<td>If true, the virtual machine will be initially in 'paused' state after start.</td>
</tr>
<tr>
<td>stateless</td>
<td>Boolean</td>
<td>If true, the virtual machine is stateless - it’s state (disks) are rolled-back after shutdown.</td>
</tr>
<tr>
<td>status</td>
<td>TemplateStatus</td>
<td>The status of the template.</td>
</tr>
<tr>
<td>storage_error_resume_behaviour</td>
<td>VmStorageErrorResumeBehaviour</td>
<td>Determines how the virtual machine will be resumed after storage error.</td>
</tr>
<tr>
<td>time_zone</td>
<td>TimeZone</td>
<td>The virtual machine’s time zone set by oVirt.</td>
</tr>
<tr>
<td>tunnel_migation</td>
<td>Boolean</td>
<td>If true, the network data transfer will be encrypted during virtual machine live migration.</td>
</tr>
<tr>
<td>type</td>
<td>VmType</td>
<td>Determines whether the virtual machine is optimized for desktop or server.</td>
</tr>
<tr>
<td>usb</td>
<td>Usb</td>
<td>Configuration of USB devices for this virtual machine (count, type).</td>
</tr>
<tr>
<td>version</td>
<td>TemplateVersion</td>
<td>Indicates whether this is a base version or a sub version of another template.</td>
</tr>
<tr>
<td>virtio_scsi</td>
<td>VirtioScsi</td>
<td>Reference to VirtIO SCSI configuration.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>The virtual machine configuration associated with this template.</td>
</tr>
</tbody>
</table>

### 7.128.1. cpu

The configuration of the virtual machine CPU.

The socket configuration can be updated without rebooting the virtual machine. The cores and the threads require a reboot.
For example, to change the number of sockets to 4 immediately, and the number of cores and threads to 2 after reboot, send the following request:

```plaintext
PUT /ovirt-engine/api/vms/123
```

With a request body:

```xml
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

### 7.128.2. custom_compatibility_version

Virtual machine custom compatibility version.

Enables a virtual machine to be customized to its own compatibility version. If `custom_compatibility_version` is set, it overrides the cluster’s compatibility version for this particular virtual machine.

The compatibility version of a virtual machine is limited by the data center the virtual machine resides in, and is checked against capabilities of the host the virtual machine is planned to run on.

### 7.128.3. high_availability

The virtual machine high availability configuration. If set, the virtual machine will be automatically restarted when it unexpectedly goes down.

### 7.128.4. initialization

Reference to the virtual machine’s initialization configuration.

**NOTE**

Since Red Hat Virtualization 4.1.8 this property can be cleared by sending an empty tag.

For example, to clear the `initialization` attribute send a request like this:

```plaintext
PUT /ovirt-engine/api/vms/123
```

With a request body like this:

```xml
<vm>
  <initialization/>
</vm>
```
The response to such a request, and requests with the header `All-Content: true` will still contain this attribute.

### 7.128.5. large_icon

Virtual machine's large icon. Either set by user or refers to image set according to operating system.

### 7.128.6. lease

Reference to the storage domain this virtual machine/template lease reside on.

A virtual machine running with a lease requires checking while running that the lease is not taken by another host, preventing another instance of this virtual machine from running on another host. This provides protection against split-brain in highly available virtual machines. A template can also have a storage domain defined for a lease in order to have the virtual machines created from this template to be preconfigured with this storage domain as the location of the leases.

### 7.128.7. memory

The virtual machine's memory, in bytes.

For example, to update a virtual machine to contain 1 Gibibyte (GiB) of memory, send the following request:

PUT /ovirt-engine/api/vms/123

With the following request body:

```xml
<vm>
  <memory>1073741824</memory>
</vm>
```

Memory hot plug is supported from Red Hat Virtualization 3.6 onwards. You can use the example above to increase memory while the virtual machine is in state `up`. The size increment must be dividable by the value of the `HotPlugMemoryBlockSizeMb` configuration value (256 MiB by default). If the memory size increment is not dividable by this value, the memory size change is only stored to next run configuration. Each successful memory hot plug operation creates one or two new memory devices.

Memory hot unplug is supported since Red Hat Virtualization 4.2 onwards. Memory hot unplug can only be performed when the virtual machine is in state `up`. Only previously hot plugged memory devices can be removed by the hot unplug operation. The requested memory decrement is rounded down to match sizes of a combination of previously hot plugged memory devices. The requested memory value is stored to next run configuration without rounding.

**NOTE**

Memory in the example is converted to bytes using the following formula:

1 GiB = $2^{30}$ bytes = 1073741824 bytes.

**NOTE**

Red Hat Virtualization Manager internally rounds values down to whole MiBs ($1\text{MiB} = 2^{20}$ bytes).
7.128.8. migration_downtime

Maximum time the virtual machine can be non responsive during its live migration to another host in ms.

Set either explicitly for the virtual machine or by `engine-config -s DefaultMaximumMigrationDowntime=[value]

7.128.9. origin

The origin of this virtual machine.

Possible values:

- ovirt
- rhev
- vmware
- xen
- external
- hosted_engine
- managed_hosted_engine
- kvm
- physical_machine
- hyperv

7.128.10. placement_policy

The configuration of the virtual machine’s placement policy.

This configuration can be updated to pin a virtual machine to one or more hosts.

**NOTE**

Virtual machines that are pinned to multiple hosts cannot be live migrated, but in the event of a host failure, any virtual machine configured to be highly available is automatically restarted on one of the other hosts to which the virtual machine is pinned.

For example, to pin a virtual machine to two hosts, send the following request:

```
PUT /api/vms/123
```

With a request body like this:

```
<vm>
  <high_availability>
    <enabled>true</enabled>
  </high_availability>
</vm>
```
7.128.11. **small_icon**

Virtual machine’s small icon. Either set by user or refers to image set according to operating system.

7.128.12. **sso**

Reference to the Single Sign On configuration this virtual machine is configured for. The user can be automatically signed in the virtual machine’s operating system when console is opened.

**Table 7.175. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>References to the CD-ROM devices attached to the template.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster the virtual machine belongs to.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Reference to CPU profile used by this virtual machine.</td>
</tr>
<tr>
<td>disk_attachments</td>
<td>DiskAttachment[]</td>
<td>References to the disks attached to the template.</td>
</tr>
<tr>
<td>graphics_consoles</td>
<td>GraphicsConsole[]</td>
<td>References to the graphic consoles attached to the template.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>References to the network interfaces attached to the template.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>References to the user permissions attached to the template.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Reference to quota configuration set for this virtual machine.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to storage domain the virtual machine belongs to.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>References to the tags attached to the template.</td>
</tr>
</tbody>
</table>
### 7.129. IO STRUCT

**Table 7.176. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>References to the watchdog devices attached to the template.</td>
</tr>
</tbody>
</table>

### 7.130. IP STRUCT

Represents the IP configuration of a network interface.

**Table 7.177. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>The text representation of the IP address.</td>
</tr>
<tr>
<td>gateway</td>
<td>String</td>
<td>The address of the default gateway.</td>
</tr>
<tr>
<td>netmask</td>
<td>String</td>
<td>The network mask.</td>
</tr>
<tr>
<td>version</td>
<td>IpVersion</td>
<td>The version of the IP protocol.</td>
</tr>
</tbody>
</table>

#### 7.130.1. address

The text representation of the IP address.

For example, an IPv4 address will be represented as follows:

```xml
<ip>
  <address>192.168.0.1</address>
  ...
</ip>
```

An IPv6 address will be represented as follows:

```xml
<ip>
  <address>2620:52:0:20f0:4216:7eff:feaa:1b50</address>
  ...
</ip>
```

#### 7.130.2. netmask
The network mask.

For IPv6 addresses the value is an integer in the range of 0-128, which represents the subnet prefix.

### 7.130.3. version

The version of the IP protocol.

**NOTE**

From version 4.1 of the Manager this attribute will be optional, and when a value is not provided, it will be inferred from the value of the address attribute.

### 7.131. IPAddressAssignment Struct

Represents an IP address assignment for a network device.

For a static boot protocol assignment, subnet mask and IP address (and optionally default gateway) must be provided in the IP configuration.

**Table 7.178. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignment_method</td>
<td>BootProtocol</td>
<td>Sets the boot protocol used to assign the IP configuration for a network device.</td>
</tr>
<tr>
<td>ip</td>
<td>Ip</td>
<td>Sets the IP configuration for a network device.</td>
</tr>
</tbody>
</table>

### 7.132. IPVersion Enum

Defines the values for the IP protocol version.

**Table 7.179. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>v4</td>
<td>IPv4.</td>
</tr>
<tr>
<td>v6</td>
<td>IPv6.</td>
</tr>
</tbody>
</table>

### 7.133. iSCSiBond Struct

**Table 7.180. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
</tbody>
</table>
### Table 7.181. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td></td>
</tr>
<tr>
<td>networks</td>
<td>Network[]</td>
<td></td>
</tr>
<tr>
<td>storage_connections</td>
<td>StorageConnection[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.134. ISCSIDETAILS STRUCT

#### Table 7.182. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>disk_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>initiator</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>lun_mapping</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>paths</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>portal</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>product_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>serial</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>
### 7.135. JOB STRUCT

Represents a job, which monitors execution of a flow in the system. A job can contain multiple steps in a hierarchic structure. The steps can be processed in parallel, depends on the implementation of the flow.

#### Table 7.183. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_cleared</td>
<td>Boolean</td>
<td>Indicates if the job should be cleared automatically after it was completed by the system.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>end_time</td>
<td>Date</td>
<td>The end time of the job.</td>
</tr>
<tr>
<td>external</td>
<td>Boolean</td>
<td>Indicates if the job is originated by an external system.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>last_updated</td>
<td>Date</td>
<td>The last update date of the job.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>start_time</td>
<td>Date</td>
<td>The start time of the job.</td>
</tr>
<tr>
<td>status</td>
<td>JobStatus</td>
<td>The status of the job.</td>
</tr>
</tbody>
</table>

#### 7.135.1. external
Indicates if the job is originated by an external system. External jobs are managed externally, by the creator of the job.

**Table 7.184. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>owner</td>
<td>User</td>
<td>The user who is the owner of the job.</td>
</tr>
<tr>
<td>steps</td>
<td>Step[]</td>
<td>The steps of the job.</td>
</tr>
</tbody>
</table>

## 7.136. JOBSTATUS ENUM

Represents the status of the job.

**Table 7.185. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>aborted</td>
<td>The aborted job status.</td>
</tr>
<tr>
<td>failed</td>
<td>The failed job status.</td>
</tr>
<tr>
<td>finished</td>
<td>The finished job status.</td>
</tr>
<tr>
<td>started</td>
<td>The started job status.</td>
</tr>
<tr>
<td>unknown</td>
<td>The unknown job status.</td>
</tr>
</tbody>
</table>

### 7.136.1. aborted

The aborted job status. This status is applicable for an external job that was forcibly aborted.

### 7.136.2. finished

The finished job status. This status describes a completed job execution.

### 7.136.3. started

The started job status. This status represents a job which is currently being executed.

### 7.136.4. unknown

The unknown job status. This status represents jobs which their resolution is not known, i.e. jobs that were executed before the system was unexpectedly restarted.

## 7.137. KATELLOERRATUM STRUCT

Type representing a Katello erratum.
**Table 7.186. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>issued</td>
<td>Date</td>
<td>The date when the Katello erratum was issued.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>packages</td>
<td>Package[]</td>
<td>The list of packages which solve the issue reported by the Katello erratum.</td>
</tr>
<tr>
<td>severity</td>
<td>String</td>
<td>The severity of the Katello erratum.</td>
</tr>
<tr>
<td>solution</td>
<td>String</td>
<td>The solution for the issue described by the Katello erratum.</td>
</tr>
<tr>
<td>summary</td>
<td>String</td>
<td>The summary of the Katello erratum.</td>
</tr>
<tr>
<td>title</td>
<td>String</td>
<td>The title of the Katello erratum.</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>The type of the Katello erratum.</td>
</tr>
</tbody>
</table>

**7.137.1. severity**

The severity of the Katello erratum.

The supported severities are `moderate`, `important` or `critical`.

**7.137.2. type**

The type of the Katello erratum.

The supported types are `bugfix`, `enhancement` or `security`.

**Table 7.187. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host that the Katello erratum is assigned to.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Reference to the virtual machine that the Katello erratum is assigned to.</td>
</tr>
</tbody>
</table>
7.138. KDUMPSTATUS ENUM

Table 7.188. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td></td>
</tr>
<tr>
<td>enabled</td>
<td></td>
</tr>
<tr>
<td>unknown</td>
<td></td>
</tr>
</tbody>
</table>

7.139. KERNEL STRUCT

Table 7.189. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>Version</td>
<td></td>
</tr>
</tbody>
</table>

7.140. KSM STRUCT

Table 7.190. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>merge_across_nodes</td>
<td>Boolean</td>
<td></td>
</tr>
</tbody>
</table>

7.141. LINKLAYERDISCOVERYPROTOCOLELEMENT STRUCT

Represents an information element received by Link Layer Discovery Protocol (LLDP). IEEE 802.1AB defines type, length, value (TLV) as a "short, variable length encoding of an information element". This type represents such an information element.

The attribute name is a human-readable string used to describe what the value is about, and may not be unique. The name is redundant, because it could be created from type and the optional oui and subtype. The purpose of name is to simplify the reading of the information element. The name of a property is exactly the same string which is used in IEEE 802.1AB chapter 8.

Organizationally-specific information elements have the type of 127 and the attributes oui and subtype.

For example, the XML representation of an information element may look like this:

```xml
<link_layer_discovery_protocol_element>
  <name>Port VLAN Id</name>
</link_layer_discovery_protocol_element>
```
Table 7.191. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>oui</td>
<td>Integer</td>
<td>The organizationally-unique identifier (OUI) encoded as an integer.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Represents structured data transported by the information element as a list of name/value pairs.</td>
</tr>
<tr>
<td>subtype</td>
<td>Integer</td>
<td>The organizationally-defined subtype encoded as an integer.</td>
</tr>
<tr>
<td>type</td>
<td>Integer</td>
<td>The type of the LinkLayerDiscoveryProtocolElement encoded as an integer.</td>
</tr>
</tbody>
</table>

7.141.1. oui

The organizationally-unique identifier (OUI) encoded as an integer. Only available if **type** is 127.

7.141.2. subtype

The organizationally-defined subtype encoded as an integer. Only available if **type** is 127.

7.142. LOGSEVERITY ENUM

Enum representing a severity of an event.

Table 7.192. Values summary
### 7.142. alert
Alert severity. Used to specify a condition that requires an immediate attention.

### 7.142.2. error
Error severity. Used to specify that there is an error that needs to be examined.

### 7.142.3. normal
Normal severity. Used for information events.

### 7.142.4. warning
Warning severity. Used to warn something might be wrong.

### 7.143. LOGICALUNIT STRUCT

Table 7.193. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>discard_max_size</td>
<td>Integer</td>
<td>The maximum number of bytes that can be discarded by the logical unit's underlying storage in a single operation.</td>
</tr>
<tr>
<td>discard_zeroes_data</td>
<td>Boolean</td>
<td>True, if previously discarded blocks in the logical unit's underlying storage are read back as zeros.</td>
</tr>
<tr>
<td>disk_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>lun_mapping</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>paths</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>portal</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>product_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>serial</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>LunStatus</td>
<td></td>
</tr>
<tr>
<td>storage_domain_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>vendor_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>volume_group_id</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### 7.143.1. `discard_max_size`

The maximum number of bytes that can be discarded by the logical unit’s underlying storage in a single operation. A value of 0 means that the device does not support discard functionality.

**NOTE**

This is the software limit, and not the hardware limit, as noted in the documentation of `queue-sysfs` for `discard_max_bytes`.

### 7.143.2. `discard_zeroes_data`

True, if previously discarded blocks in the logical unit’s underlying storage are read back as zeros. For more information please see the documentation of `queue-sysfs` for `discard_zeroes_data`.

**IMPORTANT**

Since version 4.2.1 of the system, the support for this attribute has been removed as the sysfs file, `discard_zeroes_data`, was deprecated in the kernel. It is preserved for backwards compatibility, but the value will always be `false`. 
7.144. LUNSTATUS ENUM

Table 7.194. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>free</td>
<td></td>
</tr>
<tr>
<td>unusable</td>
<td></td>
</tr>
<tr>
<td>used</td>
<td></td>
</tr>
</tbody>
</table>

7.145. MAC STRUCT

Represents a MAC address of a virtual network interface.

Table 7.195. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>MAC address.</td>
</tr>
</tbody>
</table>

7.146. MACPOOL STRUCT

Represents a MAC address pool.

Example of an XML representation of a MAC address pool:

```xml
<mac_pool href="/ovirt-engine/api/macpools/123" id="123">
  <name>Default</name>
  <description>Default MAC pool</description>
  <allow_duplicates>false</allow_duplicates>
  <default_pool>true</default_pool>
  <ranges>
    <range>
      <from>00:1A:4A:16:01:51</from>
      <to>00:1A:4A:16:01:E6</to>
    </range>
  </ranges>
</mac_pool>
```

Table 7.196. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow_duplicates</td>
<td>Boolean</td>
<td>Defines whether duplicate MAC addresses are permitted in the pool.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>default_pool</td>
<td>Boolean</td>
<td>Defines whether this is the default pool.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>ranges</td>
<td>Range[]</td>
<td>Defines the range of MAC addresses for the pool.</td>
</tr>
</tbody>
</table>

7.146.1. allow_duplicates
Defines whether duplicate MAC addresses are permitted in the pool. If not specified, defaults to false.

7.146.2. default_pool
Defines whether this is the default pool. If not specified, defaults to false.

7.146.3. ranges
Defines the range of MAC addresses for the pool. Multiple ranges can be defined.

### 7.147. MEMORYOVERCOMMIT STRUCT

Table 7.197. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>percent</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

### 7.148. MEMORYPOLICY STRUCT

Logical grouping of memory-related properties of virtual machine-like entities.

Table 7.198. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ballooning</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>guaranteed</td>
<td>Integer</td>
<td>The amount of memory, in bytes, that is guaranteed to not be drained by the balloon mechanism.</td>
</tr>
<tr>
<td>max</td>
<td>Integer</td>
<td>Maximum virtual machine memory, in bytes.</td>
</tr>
</tbody>
</table>
### 7.148. guaranteed

The amount of memory, in bytes, that is guaranteed to not be drained by the balloon mechanism.

The Red Hat Virtualization Manager internally rounds this value down to whole MiB (1MiB = \(2^{20}\) bytes).

**NOTE**

It can be updated while the virtual machine is running since Red Hat Virtualization 4.2 onwards, provided memory is updated in the same request as well, and the virtual machine is in state up.

### 7.148.2. max

Maximum virtual machine memory, in bytes.

The user provides the value in bytes, and the Red Hat Virtualization Manager rounds the value down to the nearest lower MiB value.

For example, if the user enters a value of 1073741825 (1 GiB + 1 byte), then the Red Hat Virtualization Manager will truncate that value to the nearest lower MiB boundary: in this case 1073741824 (1 GiB).

### 7.149. MESSAGEBROKERTYPE ENUM

Table 7.199. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>qpid</td>
<td></td>
</tr>
<tr>
<td>rabbit_mq</td>
<td></td>
</tr>
</tbody>
</table>

### 7.150. METHOD STRUCT

Table 7.200. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>SsoMethod</td>
<td></td>
</tr>
</tbody>
</table>
7.151. MIGRATEONERROR ENUM

Table 7.201. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>do_not_migrate</td>
<td></td>
</tr>
<tr>
<td>migrate</td>
<td></td>
</tr>
<tr>
<td>migrate_highly_available</td>
<td></td>
</tr>
</tbody>
</table>

7.152. MIGRATIONBANDWIDTH STRUCT

Defines the bandwidth used by migration.

Table 7.202. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignment_method</td>
<td>MigrationBandwidthAssignmentMethod</td>
<td>The method used to assign the bandwidth.</td>
</tr>
<tr>
<td>custom_value</td>
<td>Integer</td>
<td>Custom bandwidth in Mbps.</td>
</tr>
</tbody>
</table>

7.152.1. custom_value

Custom bandwidth in Mbps. Will be applied only if the assignmentMethod attribute is custom.

7.153. MIGRATIONBANDWIDTHASSIGNMENTMETHOD ENUM

Defines how the migration bandwidth is assigned.

Table 7.203. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td>Takes the bandwidth from the Quality of Service if the Quality of Service is defined.</td>
</tr>
<tr>
<td>custom</td>
<td>Custom defined bandwidth in Mbit/s.</td>
</tr>
<tr>
<td>hypervisor_default</td>
<td>Takes the value as configured on the hypervisor.</td>
</tr>
</tbody>
</table>
Takes the bandwidth from the Quality of Service if the Quality of Service is defined. If the Quality of Service is not defined the bandwidth is taken from the detected link speed being used. If nothing is detected, bandwidth falls back to the hypervisor_default value.

### 7.154. MIGRATIONOPTIONS STRUCT

The type for migration options.

#### Table 7.204. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_converge</td>
<td>InheritableBoolean</td>
<td></td>
</tr>
<tr>
<td>bandwidth</td>
<td>MigrationBandwidth</td>
<td>The bandwidth that is allowed to be used by the migration.</td>
</tr>
<tr>
<td>compressed</td>
<td>InheritableBoolean</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 7.205. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>MigrationPolicy</td>
<td>A reference to the migration policy, as defined using engine-config.</td>
</tr>
</tbody>
</table>

### 7.155. MIGRATIONPOLICY STRUCT

A policy describing how the migration is treated, such as convergence or how many parallel migrations are allowed.

#### Table 7.206. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### 7.156. NETWORK STRUCT

The type for a logical network.

An example of the JSON representation of a logical network:
An example of the XML representation of the same logical network:

```xml
<network href="/ovirt-engine/api/networks/456" id="456">
  <name>ovirtmgmt</name>
  <description>Management Network</description>
  <link href="/ovirt-engine/api/networks/456/permissions" rel="permissions"/>
  <link href="/ovirt-engine/api/networks/456/vnicprofiles" rel="vnicprofiles"/>
  <link href="/ovirt-engine/api/networks/456/labels" rel="labels"/>
  <data_center href="/ovirt-engine/api/datacenters/123" id="123"/>
  <stp>false</stp>
  <mtu>0</mtu>
  <usages>
    <usage>vm</usage>
  </usages>
</network>
```

Table 7.207. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>display</td>
<td>Boolean</td>
<td>Deprecated, 'usages' should be used to define network as a display network.</td>
</tr>
<tr>
<td>dns_resolver_configuration</td>
<td>DnsResolverConfiguration</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ip</td>
<td>Ip</td>
<td>Deprecated, not in use.</td>
</tr>
<tr>
<td>mtu</td>
<td>Integer</td>
<td>Specifies the maximum transmission unit for the network.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>profile_required</td>
<td>Boolean</td>
<td>Specifies whether upon creation of the network a virtual network interface profile should automatically be created.</td>
</tr>
<tr>
<td>required</td>
<td>Boolean</td>
<td>Defines whether the network is mandatory for all the hosts in the cluster.</td>
</tr>
<tr>
<td>status</td>
<td>NetworkStatus</td>
<td>The status of the network.</td>
</tr>
<tr>
<td>stp</td>
<td>Boolean</td>
<td>Specifies whether the spanning tree protocol is enabled for the network.</td>
</tr>
<tr>
<td>usages</td>
<td>NetworkUsage[]</td>
<td>Defines a set of usage elements for the network.</td>
</tr>
<tr>
<td>vlan</td>
<td>Vlan</td>
<td>A VLAN tag.</td>
</tr>
</tbody>
</table>

**7.156.1. required**

Defines whether the network is mandatory for all the hosts in the cluster. In case a 'required' operational network is omitted from a host, the host will be marked as non_operational.

**7.156.2. status**

The status of the network. non_operational if the network defined as 'required' and omitted from any active cluster host. operational otherwise.

**7.156.3. usages**

Defines a set of usage elements for the network.

For example, users can specify that the network is to be used for virtual machine traffic and also for display traffic with the vm and display values.
Table 7.208. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>A reference to the cluster this network is attached to.</td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>A reference to the data center that the network is a member of.</td>
</tr>
<tr>
<td>external_provider</td>
<td>OpenStackNetworkProvider</td>
<td>An optional reference to the OpenStack network provider on which the network is created.</td>
</tr>
<tr>
<td>external_provider_physical_network</td>
<td>Network</td>
<td>An optional reference to a network that should be used for physical network access.</td>
</tr>
<tr>
<td>network_labels</td>
<td>NetworkLabel[]</td>
<td>A reference to the labels assigned to the network.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>A reference to the permissions of the network.</td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td>Reference to quality of service.</td>
</tr>
<tr>
<td>vnic_profiles</td>
<td>VnicProfile[]</td>
<td>A reference to the profiles of the network.</td>
</tr>
</tbody>
</table>

7.156.4. cluster
A reference to the cluster this network is attached to. Will be filled only if the network is accessed from the cluster level.

7.156.5. external_provider
An optional reference to the OpenStack network provider on which the network is created.

If it is specified when a network is created, a matching OpenStack network will be also created.

7.156.6. external_provider_physical_network
An optional reference to a network that should be used for physical network access. Valid only if external_provider is specified.

7.157. NETWORKATTACHMENT STRUCT
Describes how a host connects to a network.

An XML representation of a network attachment on a host:

```xml
<network_attachment href="/ovirt-engine/api/hosts/123/nics/456/networkattachments/789" id="789"/>
<network href="/ovirt-engine/api/networks/234" id="234"/>
<host_nic href="/ovirt-engine/api/hosts/123/nics/123" id="123"/>
```
The network element, with either a name or an id, is required in order to attach a network to a network interface card (NIC).

For example, to attach a network to a host network interface card, send a request like this:

```plaintext
POST /ovirt-engine/api/hosts/123/nics/456/networkattachments
```

With a request body like this:

```xml
<networkattachment>
  <network id="234"/>
</networkattachment>
```

To attach a network to a host, send a request like this:

```plaintext
POST /ovirt-engine/api/hosts/123/networkattachments
```

With a request body like this:

```xml
<network_attachment>
  <network id="234"/>
  <host_nic id="456"/>
</network_attachment>
```

The ip_address_assignments and properties elements are updatable post-creation.

```
<in_sync>true</in_sync>
<ip_address_assignments>
  <ip_address_assignment>
    <assignment_method>static</assignment_method>
    <ip>
      <address>192.168.122.39</address>
      <gateway>192.168.122.1</gateway>
      <netmask>255.255.255.0</netmask>
      <version>v4</version>
    </ip>
  </ip_address_assignment>
</ip_address_assignments>
<reported_configurations>
  <reported_configuration>
    <name>mtu</name>
    <expected_value>1500</expected_value>
    <actual_value>1500</actual_value>
    <in_sync>true</in_sync>
  </reported_configuration>
  <reported_configuration>
    <name>bridged</name>
    <expected_value>true</expected_value>
    <actual_value>true</actual_value>
    <in_sync>true</in_sync>
  </reported_configuration>
  ...
</reported_configurations>
</network_attachment>
```
For example, to update a network attachment, send a request like this:

```plaintext
PUT /ovirt-engine/api/hosts/123/nics/456/networkattachments/789
```

With a request body like this:

```xml
<network_attachment>
  <ip_address_assignments>
    <ip_address_assignment>
      <assignment_method>static</assignment_method>
      <ip>
        <address>7.1.1.1</address>
        <gateway>7.1.1.2</gateway>
        <netmask>255.255.255.0</netmask>
        <version>v4</version>
      </ip>
    </ip_address_assignment>
  </ip_address_assignments>
</network_attachment>
```

To detach a network from the network interface card send a request like this:

```plaintext
DELETE /ovirt-engine/api/hosts/123/nics/456/networkattachments/789
```

**IMPORTANT**

Changes to network attachment configuration must be explicitly committed.

An XML representation of a network attachment's `properties` sub-collection:

```xml
<network_attachment>
  <properties>
    <property>
      <name>bridge_opts</name>
      <value>
        forward_delay=1500 group_fwd_mask=0x0 multicast_snooping=1
      </value>
    </property>
    ...
  </properties>
</network_attachment>
```

**Table 7.209. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
</tbody>
</table>
DNS resolver configuration will be reported when retrieving the network attachment using GET. It is optional when creating a new network attachment or updating an existing one.

### 7.157.2. properties

Defines custom properties for the network configuration.

Bridge options have the set name of `bridge_opts`. Separate multiple entries with a whitespace character. The following keys are valid for `bridge_opts`:

<table>
<thead>
<tr>
<th>Name</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward_delay</td>
<td>1500</td>
</tr>
<tr>
<td>gc_timer</td>
<td>3765</td>
</tr>
<tr>
<td>group_addr</td>
<td>1:80:c2:0:0:0:0</td>
</tr>
<tr>
<td>group_fwd_mask</td>
<td>0x0</td>
</tr>
<tr>
<td>hash_elasticity</td>
<td>4</td>
</tr>
<tr>
<td>hash_max</td>
<td>512</td>
</tr>
<tr>
<td>Name</td>
<td>Default value</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>hello_time</td>
<td>200</td>
</tr>
<tr>
<td>hello_timer</td>
<td>70</td>
</tr>
<tr>
<td>max_age</td>
<td>2000</td>
</tr>
<tr>
<td>multicast_last_member_count</td>
<td>2</td>
</tr>
<tr>
<td>multicast_last_member_interval</td>
<td>100</td>
</tr>
<tr>
<td>multicast_membership_interval</td>
<td>26000</td>
</tr>
<tr>
<td>multicast_querier</td>
<td>0</td>
</tr>
<tr>
<td>multicast_querier_interval</td>
<td>25500</td>
</tr>
<tr>
<td>multicast_query_interval</td>
<td>13000</td>
</tr>
<tr>
<td>multicast_query_response_interval</td>
<td>1000</td>
</tr>
<tr>
<td>multicast_query_use_ifaddr</td>
<td>0</td>
</tr>
<tr>
<td>multicast_router</td>
<td>1</td>
</tr>
<tr>
<td>multicast_snooping</td>
<td>1</td>
</tr>
<tr>
<td>multicast_startup_query_count</td>
<td>2</td>
</tr>
<tr>
<td>multicast_startup_query_interval</td>
<td>3125</td>
</tr>
</tbody>
</table>

Table 7.210. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>host_nic</td>
<td>HostNic</td>
<td>A reference to the host network interface.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>A reference to the network that the interface is attached to.</td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td></td>
</tr>
</tbody>
</table>

7.158. NETWORKCONFIGURATION STRUCT
### 7.159. NETWORKFILTER STRUCT

Network filters filter packets sent to and from the virtual machine’s NIC according to defined rules.

There are several types of network filters supported based on libvirt. For more details about the different network filters see [here](#).

In addition to libvirt’s network filters, there are two additional network filters: The first is called `vdsm-no-mac-spoofing` and is composed of `no-mac-spoofing` and `no-arp-mac-spoofing`. The second is called `ovirt-no-filter` and is used when no network filter is to be defined for the virtual machine’s NIC. The `ovirt-no-filter` network filter is only used for internal implementation, and does not exist on the NICs.

This is an example of the XML representation:

```xml
<network_filter id="00000019-0019-0019-0019-000000026C">
    <name>example-filter</name>
    <version>
        <major>4</major>
        <minor>0</minor>
        <build>-1</build>
        <revision>-1</revision>
    </version>
</network_filter>
```

If any part of the version is not present, it is represented by -1.
The minimum supported version of a specific NetworkFilter. This is the version that the NetworkFilter was first introduced in.

### 7.160. NETWORKFILTERPARAMETER STRUCT

Parameter for the network filter.

See Libvirt-Filters for further details. This is an example of the XML representation:

```xml
<network_filter_parameter id="123">
  <name>IP</name>
  <value>10.0.1.2</value>
</network_filter_parameter>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>Represents the value of the parameter.</td>
</tr>
</tbody>
</table>

### 7.161. NETWORKLABEL STRUCT

Represents a label which can be added to a host network interface and to a network. The label binds the network to the host network interface by the label `id`.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

**Table 7.216. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_nic</td>
<td>HostNic</td>
<td>A reference to the host network interface which contains this label.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>A reference to the network which contains this label.</td>
</tr>
</tbody>
</table>

### 7.162. NETWORKPLUGINTYPE ENUM

Network plug-in type.

Specifies the provider driver implementation on the host.

Since version 4.2 of the Red Hat Virtualization Manager, this type has been deprecated in favour of the `external_plugin_type` attribute of the `OpenStackNetworkProvider` type.

**Table 7.217. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>open_vswitch</td>
<td>Open vSwitch.</td>
</tr>
</tbody>
</table>

#### 7.162.1. open_vswitch

Open vSwitch.

Specifies that Open vSwitch based driver implementation should be used for this provider.

Since version 4.2 of the Red Hat Virtualization Manager, this value has been deprecated. Use the string `open_vswitch` in the `OpenStackNetworkProvider.external_plugin_type` attribute instead.

### 7.163. NETWORKSTATUS ENUM

**Table 7.218. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>non_operational</td>
<td></td>
</tr>
<tr>
<td>operational</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 7. TYPES

7.164. NETWORKUSAGE ENUM

This type indicates the purpose that the network is used for in the cluster.

Table 7.219. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>default_route</td>
<td>The default gateway and the DNS resolver configuration of the host will be taken from this network.</td>
</tr>
<tr>
<td>display</td>
<td>The network will be used for SPICE and VNC traffic.</td>
</tr>
<tr>
<td>gluster</td>
<td>The network will be used for Gluster (bricks) data traffic.</td>
</tr>
<tr>
<td>management</td>
<td>The network will be used for communication between the Red Hat Virtualization Manager and the nodes.</td>
</tr>
<tr>
<td>migration</td>
<td>The network will be used for virtual machine migration.</td>
</tr>
<tr>
<td>vm</td>
<td></td>
</tr>
</tbody>
</table>

7.164.1. default_route

The default gateway and the DNS resolver configuration of the host will be taken from this network.

If this network is attached to the host, then the DNS resolver configuration will be taken from the dns_resolver_configuration attribute of the network attachment. If there is no dns_resolver_configuration attribute in this network attachment, then they will be taken from the dns_resolver_configuration of the network itself. If dns_resolver_configuration attribute isn’t present even there, DNS resolver configuration won’t be set.

If you set this flag on a network, then the default gateway for the host will be taken from the gateway attribute of the ip_address_assignment of the network attachment.

7.164.2. management

The network will be used for communication between the Red Hat Virtualization Manager and the nodes. This is the network where the ovirtmgmt bridge will be created.

7.165. NFSPROFILEDETAIL STRUCT

Table 7.220. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfs_server_ip</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
### 7.166. NFSVERSION ENUM

Table 7.221. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto</td>
<td></td>
</tr>
<tr>
<td>v3</td>
<td></td>
</tr>
<tr>
<td>v4</td>
<td></td>
</tr>
<tr>
<td>v4_1</td>
<td></td>
</tr>
<tr>
<td>v4_2</td>
<td>NFS 4.</td>
</tr>
</tbody>
</table>

#### 7.166.1. v4_2

NFS 4.2.

### 7.167. NIC STRUCT

Represents a virtual machine NIC.

For example, the XML representation of a NIC will look like this:

```xml
<nic href="/ovirt-engine/api/vms/123/nics/456" id="456">
  <name>nic1</name>
  <vm href="/ovirt-engine/api/vms/123" id="123"/>
  <interface>virtio</interface>
  <linked>true</linked>
  <mac>
    <address>02:00:00:00:00:00</address>
  </mac>
  <plugged>true</plugged>
  <vnic_profile href="/ovirt-engine/api/vnicprofiles/789" id="789"/>
</nic>
```

Table 7.222. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_protocol</td>
<td>BootProtocol</td>
<td>Defines how an IP address is assigned to the NIC.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>interface</td>
<td>NicInterface</td>
<td>The type of driver used for the NIC.</td>
</tr>
<tr>
<td>linked</td>
<td>Boolean</td>
<td>Defines if the NIC is linked to the virtual machine.</td>
</tr>
<tr>
<td>mac</td>
<td>Mac</td>
<td>The MAC address of the interface.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>on_boot</td>
<td>Boolean</td>
<td>Defines if the network interface should be activated upon operation system startup.</td>
</tr>
<tr>
<td>plugged</td>
<td>Boolean</td>
<td>Defines if the NIC is plugged in to the virtual machine.</td>
</tr>
</tbody>
</table>

Table 7.223. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>network</td>
<td>Network</td>
<td>A reference to the network that the interface should be connected to.</td>
</tr>
<tr>
<td>network_attachments</td>
<td>NetworkAttachment[]</td>
<td>A link to a collection of network attachments that are associated with the host NIC.</td>
</tr>
<tr>
<td>network_filter_parameters</td>
<td>NetworkFilterParameter[]</td>
<td>A link to the network filter parameters.</td>
</tr>
<tr>
<td>network_labels</td>
<td>NetworkLabel[]</td>
<td>A link to a collection of network labels that are associated with the host NIC.</td>
</tr>
<tr>
<td>reported_devices</td>
<td>ReportedDevice[]</td>
<td>A link to a collection of reported devices that are associated with the virtual network interface.</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>A link to the statistics for the NIC.</td>
</tr>
</tbody>
</table>
### 7.167.1. network

A reference to the network that the interface should be connected to. A blank network ID is allowed.

Usage of this element for creating or updating a NIC is deprecated; use `vnic_profile` instead. It is preserved because it is still in use by the `initialization` element, as a holder for IP addresses and other network details.

### 7.167.2. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk my be used simultaneously by two or more virtual machines.

### 7.168. NICCONFIGURATION STRUCT

The type describes the configuration of a virtual network interface.

#### Table 7.224. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot_protocol</td>
<td>BootProtocol</td>
<td>IPv4 boot protocol.</td>
</tr>
<tr>
<td>ip</td>
<td>Ip</td>
<td>IPv4 address details.</td>
</tr>
<tr>
<td>ipv6</td>
<td>Ip</td>
<td>IPv6 address details.</td>
</tr>
<tr>
<td>ipv6_boot_protocol</td>
<td>BootProtocol</td>
<td>IPv6 boot protocol.</td>
</tr>
</tbody>
</table>
7.169. NICINTERFACE ENUM

Defines the options for an emulated virtual network interface device model.

Table 7.225. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>e1000</td>
<td>e1000.</td>
</tr>
<tr>
<td>pci_passthrough</td>
<td>PCI Passthrough.</td>
</tr>
<tr>
<td>rtl8139</td>
<td>rtl8139.</td>
</tr>
<tr>
<td>rtl8139_virtio</td>
<td>Dual mode rtl8139, VirtIO.</td>
</tr>
<tr>
<td>spapr_vlan</td>
<td>sPAPR VLAN.</td>
</tr>
<tr>
<td>virtio</td>
<td>VirtIO.</td>
</tr>
</tbody>
</table>

7.170. NICSTATUS ENUM

Network interface card status.

Table 7.226. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>The NIC is down and cannot be accessed.</td>
</tr>
<tr>
<td>up</td>
<td>The NIC is up and can be accessed.</td>
</tr>
</tbody>
</table>

7.171. NUMANODE STRUCT

Represents a physical NUMA node.

Example XML representation:

```xml
<host_numa_node href="/ovirt-engine/api/hosts/0923f1ea/numanodes/007cf1ab"`
```
Table 7.227. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>index</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>Memory of the NUMA node in MB.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>node_distance</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.228. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Each host NUMA node resource exposes a statistics sub-collection for host NUMA node specific statistics.</td>
</tr>
</tbody>
</table>

7.171.1. statistics

Each host NUMA node resource exposes a statistics sub-collection for host NUMA node specific statistics.

An example of an XML representation:
This statistics sub-collection is read-only.

The following list shows the statistic types for a host NUMA node:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory.total</td>
<td>Total memory in bytes on the NUMA node.</td>
</tr>
<tr>
<td>memory.used</td>
<td>Memory in bytes used on the NUMA node.</td>
</tr>
<tr>
<td>memory.free</td>
<td>Memory in bytes free on the NUMA node.</td>
</tr>
<tr>
<td>cpu.current.user</td>
<td>Percentage of CPU usage for user slice.</td>
</tr>
<tr>
<td>cpu.current.system</td>
<td>Percentage of CPU usage for system.</td>
</tr>
<tr>
<td>cpu.current.idle</td>
<td>Percentage of idle CPU usage.</td>
</tr>
</tbody>
</table>

### 7.172. NUMANODEPIN STRUCT

Represents the pinning of a virtual NUMA node to a physical NUMA node.

#### Table 7.229. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host_numa_node</td>
<td>NumaNode</td>
<td>Deprecated.</td>
</tr>
</tbody>
</table>
### 7.172.1. host_numa_node

Deprecated. Has no function.

### 7.172.2. pinned

Deprecated. Should always be `true`.

### 7.173. NUMATUNEMODE ENUM

Table 7.230. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>interleave</td>
<td></td>
</tr>
<tr>
<td>preferred</td>
<td></td>
</tr>
<tr>
<td>strict</td>
<td></td>
</tr>
</tbody>
</table>

### 7.174. OPENSTACKIMAGE STRUCT

Table 7.231. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

Table 7.232. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack_image_provider</td>
<td>OpenStackImageProvider</td>
<td></td>
</tr>
</tbody>
</table>

### 7.175. OPENSTACKIMAGEPROVIDER STRUCT

#### Table 7.233. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>tenant_name</td>
<td>String</td>
<td>Defines the tenant name for OpenStack Identity API v2.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

#### 7.175.1. requires_authentication

Defines whether provider authentication is required or not.

If authentication is required, both `username` and `password` attributes will be used during authentication.

#### 7.175.2. tenant_name

Defines the tenant name for OpenStack Identity API v2.0.

#### Table 7.234. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td></td>
</tr>
<tr>
<td>images</td>
<td>OpenStackImage[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.176. OPENSTACKNETWORK STRUCT

#### Table 7.235. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

#### Table 7.236. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack_network_provider</td>
<td>OpenStackNetworkProvider</td>
<td></td>
</tr>
</tbody>
</table>

### 7.177. OPENSTACKNETWORKPROVIDER STRUCT

#### Table 7.237. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent_configuration</td>
<td>AgentConfiguration</td>
<td>Agent configuration settings.</td>
</tr>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>auto_sync</td>
<td>Boolean</td>
<td>Indicates if the networks of this provider are automatically synchronized.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>external_plugin_type</td>
<td>String</td>
<td>Network plug-in type.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>plugin_type</td>
<td>NetworkPluginType</td>
<td>Network plug-in type.</td>
</tr>
<tr>
<td>project_domain_name</td>
<td>String</td>
<td>Defines the project's domain name for OpenStack Identity API v3.</td>
</tr>
<tr>
<td>project_name</td>
<td>String</td>
<td>Defines the project name for OpenStack Identity API v3.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>read_only</td>
<td>Boolean</td>
<td>Indicates whether the provider is read-only.</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>tenant_name</td>
<td>String</td>
<td>Defines the tenant name for OpenStack Identity API v2.</td>
</tr>
<tr>
<td>type</td>
<td>OpenStackNetworkProviderType</td>
<td>The type of provider.</td>
</tr>
<tr>
<td>unmanaged</td>
<td>Boolean</td>
<td>Indicates whether the provider is unmanaged by Red Hat Virtualization.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>user_domain_name</td>
<td>String</td>
<td>Defines the domain name of the username in ExternalProvider for OpenStack Identity API v3.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

7.177.1. auto_sync

Indicates if the networks of this provider are automatically synchronized.

If true, the networks of this provider are automatically and cyclically synchronized to Red Hat Virtualization in the background. This means that all new networks of this provider are imported, and all discarded networks are removed from all clusters that have this external provider as the default provider.
If the name of a network is changed on the provider, the change is synchronized to the network entity in Red Hat Virtualization. Furthermore, if a new cluster that has the provider as the default provider is added, already imported networks are attached to this new cluster during synchronization.

The automatically initiated import triggers the following steps:

- The networks of the external provider will be imported to every data center in the data centers of the clusters that have that external provider as the default provider.
- A vNIC profile will be created for each involved data center and network.
- The networks will be assigned to each cluster that has that external provider as the default provider.

All users are allowed to use the new vNIC Profile.

The default is false for backwards compatibility.

7.177.2. external_plugin_type

Network plug-in type.

This attribute allows you to choose the correct provider driver on the host when an external NIC is added or modified. If automated installation of the driver is supported (only available for some predefined implementations, for example ovirt-provider-ovn), this attribute will also allow the system to decide which driver implementation to install on newly added hosts.

7.177.3. plugin_type

Network plug-in type.

Since version 4.2 of the Red Hat Virtualization Manager, this attribute has been deprecated in favour of external_plugin_type. This attribute is only valid for providers of type open_vswitch, and will only be returned when the value of the external_plugin_type attribute value is equal to open_vswitch.

If both plugin_type and external_plugin_type are specified during an update, the value of plugin_type will be ignored.

For external providers this value will not be shown and will be ignored during update requests.

7.177.4. read_only

Indicates whether the provider is read-only.

A read-only provider does not allow adding, modifying, or deleting of networks or subnets. Port-related operations are allowed, as they are required for the provisioning of virtual NICs.

7.177.5. requires_authentication

Defines whether provider authentication is required or not.

If authentication is required, both username and password attributes will be used during authentication.

7.177.6. tenant_name
Defines the tenant name for OpenStack Identity API v2.0.

### 7.177. unmanaged

Indicates whether the provider is unmanaged by Red Hat Virtualization.

If true, authentication and subnet control are entirely left to the external provider and are unmanaged by Red Hat Virtualization.

The default is false for backwards compatibility.

#### Table 7.238. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td>Reference to the certificates list.</td>
</tr>
<tr>
<td>networks</td>
<td>OpenStackNetwork[]</td>
<td>Reference to the OpenStack networks list.</td>
</tr>
<tr>
<td>subnets</td>
<td>OpenStackSubnet[]</td>
<td>Reference to the OpenStack networks subnets list.</td>
</tr>
</tbody>
</table>

### 7.178. OPENSTACKNETWORKPROVIDERTYPE ENUM

The OpenStack network provider can either be implemented by OpenStack Neutron, in which case the Neutron agent is automatically installed on the hosts, or it can be an external provider implementing the OpenStack API, in which case the virtual interface driver is a custom solution installed manually.

#### Table 7.239. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>external</td>
<td>Indicates that the provider is an external one, implementing the OpenStack Neutron API.</td>
</tr>
<tr>
<td>neutron</td>
<td>Indicates that the provider is OpenStack Neutron.</td>
</tr>
</tbody>
</table>

#### 7.178.1. external

Indicates that the provider is an external one, implementing the OpenStack Neutron API. The virtual interface driver in this case is implemented by the external provider.

#### 7.178.2. neutron

Indicates that the provider is OpenStack Neutron. The standard OpenStack Neutron agent is used as the virtual interface driver.

### 7.179. OPENSTACKPROVIDER STRUCT
Table 7.240. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>tenant_name</td>
<td>String</td>
<td>Defines the tenant name for OpenStack Identity API v2.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

7.179.1. requires_authentication

Defines whether provider authentication is required or not.

If authentication is required, both **username** and **password** attributes will be used during authentication.

7.179.2. tenant_name

Defines the tenant name for OpenStack Identity API v2.0.

7.180. OPENSTACKSUBNET STRUCT

Table 7.241. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cidr</td>
<td>String</td>
<td>Defines network CIDR.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>dns_servers</td>
<td>String[]</td>
<td>Defines a list of DNS servers.</td>
</tr>
<tr>
<td>gateway</td>
<td>String</td>
<td>Defines IP gateway.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ip_version</td>
<td>String</td>
<td>Defines IP version.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### 7.180.1. ip_version

Defines IP version.

Values can be `v4` for IPv4 or `v6` for IPv6.

#### Table 7.242. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack_network</td>
<td>OpenStackNetwork</td>
<td>Reference to the service managing the OpenStack network.</td>
</tr>
</tbody>
</table>

### 7.181. OPENSTACKVOLUMEPROVIDER STRUCT

#### Table 7.243. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_url</td>
<td>String</td>
<td>Defines the external provider authentication URL address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>Defines password for the user during the authentication process.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td>Array of provider name/value properties.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>requires_authentication</td>
<td>Boolean</td>
<td>Defines whether provider authentication is required or not.</td>
</tr>
<tr>
<td>tenant_name</td>
<td>String</td>
<td>Defines the tenant name for OpenStack Identity API v2.</td>
</tr>
<tr>
<td>url</td>
<td>String</td>
<td>Defines URL address of the external provider.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>Defines user name to be used during authentication process.</td>
</tr>
</tbody>
</table>

### 7.181.1. requires_authentication

Defines whether provider authentication is required or not.

If authentication is required, both **username** and **password** attributes will be used during authentication.

### 7.181.2. tenant_name

Defines the tenant name for OpenStack Identity API v2.0.

**Table 7.244. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_keys</td>
<td>OpenstackVolume AuthenticationKey[]</td>
<td></td>
</tr>
<tr>
<td>certificates</td>
<td>Certificate[]</td>
<td></td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td></td>
</tr>
<tr>
<td>volume_types</td>
<td>OpenStackVolume Type[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.182. OPENSTACKVOLUMETYPE STRUCT

**Table 7.245. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.246. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack_volume_provider</td>
<td>OpenStackVolumeProvider</td>
<td></td>
</tr>
</tbody>
</table>

7.183. OPENSTACKVOLUMEAUTHENTICATIONKEY STRUCT

Table 7.247. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>creation_date</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>usage_type</td>
<td>OpenstackVolume AuthenticationKey UsageType</td>
<td></td>
</tr>
<tr>
<td>uuid</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.248. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>openstack_volume_provider</td>
<td>OpenStackVolumeProvider</td>
<td></td>
</tr>
</tbody>
</table>
7.184. OPENSTACKVOLUMEAUTHENTICATIONKEYUSAGETYPE ENUM

Table 7.249. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ceph</td>
<td></td>
</tr>
</tbody>
</table>

7.185. OPERATINGSYSTEM STRUCT

Information describing the operating system. This is used for both virtual machines and hosts.

Table 7.250. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boot</td>
<td>Boot</td>
<td>Configuration of the boot sequence.</td>
</tr>
<tr>
<td>cmdline</td>
<td>String</td>
<td>Custom kernel parameters for start the virtual machine with if Linux operating system is used.</td>
</tr>
<tr>
<td>custom_kernel_cmdline</td>
<td>String</td>
<td>A custom part of the host kernel command line.</td>
</tr>
<tr>
<td>initrd</td>
<td>String</td>
<td>Path to custom initial ramdisk on ISO storage domain if Linux operating system is used.</td>
</tr>
<tr>
<td>kernel</td>
<td>String</td>
<td>Path to custom kernel on ISO storage domain if Linux operating system is used.</td>
</tr>
<tr>
<td>reported_kernel_cmdline</td>
<td>String</td>
<td>The host kernel command line as reported by a running host.</td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td>Operating system name in human readable form.</td>
</tr>
<tr>
<td>version</td>
<td>Version</td>
<td></td>
</tr>
</tbody>
</table>

7.185.1. boot

Configuration of the boot sequence.

**NOTE**

Not used for hosts.

7.185.2. cmdline

Custom kernel parameters for start the virtual machine with if Linux operating system is used.
7.185.3. custom_kernel_cmdline

A custom part of the host kernel command line. This will be merged with the existing kernel command line.

You must reinstall and then reboot the host to apply the changes implemented by this attribute.

During each host deploy procedure, kernel parameters that were added in the previous host deploy procedure are removed using `grubby --update-kernel DEFAULT --remove-args <previous_custom_params>`, and the current kernel command line customization is applied using `grubby --update-kernel DEFAULT --args <custom_params>`. The Manager internally keeps track of the last-applied kernel parameters customization.

7.185.4. initrd

Path to custom initial ramdisk on ISO storage domain if Linux operating system is used.

For example `iso://initramfs-3.10.0-514.6.1.el7.x86_64.img`.

7.185.5. kernel

Path to custom kernel on ISO storage domain if Linux operating system is used.

For example `iso://vmlinuz-3.10.0-514.6.1.el7.x86_64`.

7.185.6. reported_kernel_cmdline

The host kernel command line as reported by a running host.

This is a read-only attribute. Attempts to change this attribute are silently ignored.

7.185.7. type
Operating system name in human readable form.

For example **Fedora** or **RHEL**. In general one of the names returned by the operating system service.

**NOTE**

Read only for hosts.

### 7.186. OPERATINGSYSTEMINFO STRUCT

Represents a guest operating system.

**Table 7.251. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>architecture</td>
<td>Architecture</td>
<td>Operating system architecture.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Large icon of the guest operating system.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Small icon of the guest operating system.</td>
</tr>
</tbody>
</table>

**7.186.1. large_icon**

Large icon of the guest operating system. Maximum dimensions: width 150px, height 120px.

**7.186.2. small_icon**

Small icon of the guest operating system. Maximum dimensions: width 43px, height 43px.

### 7.187. OPTION STRUCT

**Table 7.252. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
7.188. OSTYPE ENUM

Type representing kind of operating system.

WARNING
This type has been deprecated with the introduction of the OperatingSystemInfo type. Operating systems are available as a top-level collection in the API: operating_systems

The end-user declares the type of the operating system installed in the virtual machine (guest operating system) by selecting one of these values. This declaration enables the system to tune the virtual machine configuration for better user experience. For example, the system chooses devices that are most suitable for the operating system. Note that the system rely on user’s selection and does not verify it by inspecting the actual guest operating system installed.

Table 7.253. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>other</td>
<td>Other type of operating system, not specified by the other values.</td>
</tr>
<tr>
<td>other_linux</td>
<td>Distribution of Linux other than those specified by the other values.</td>
</tr>
<tr>
<td>rhel_3</td>
<td>Red Hat Enterprise Linux 3 32-bit.</td>
</tr>
<tr>
<td>rhel_3x64</td>
<td>Red Hat Enterprise Linux 3 64-bit.</td>
</tr>
<tr>
<td>rhel_4</td>
<td>Red Hat Enterprise Linux 4 32-bit.</td>
</tr>
<tr>
<td>rhel_4x64</td>
<td>Red Hat Enterprise Linux 4 64-bit.</td>
</tr>
<tr>
<td>rhel_5</td>
<td>Red Hat Enterprise Linux 5 32-bit.</td>
</tr>
<tr>
<td>rhel_5x64</td>
<td>Red Hat Enterprise Linux 5 64-bit.</td>
</tr>
<tr>
<td>rhel_6</td>
<td>Red Hat Enterprise Linux 6 32-bit.</td>
</tr>
<tr>
<td>rhel_6x64</td>
<td>Red Hat Enterprise Linux 6 64-bit.</td>
</tr>
<tr>
<td>unassigned</td>
<td>This value is mapped to other.</td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>windows_2003 x64</td>
<td>Windows 2003 64-bit.</td>
</tr>
<tr>
<td>windows_2008 r2x64</td>
<td>Windows 2008 R2 64-bit.</td>
</tr>
<tr>
<td>windows_2008 x64</td>
<td>Windows 2008 64-bit.</td>
</tr>
<tr>
<td>windows_2012 x64</td>
<td>Windows 2012 64-bit.</td>
</tr>
<tr>
<td>windows_7</td>
<td>Windows 7 32-bit.</td>
</tr>
<tr>
<td>windows_7x64</td>
<td>Windows 7 64-bit.</td>
</tr>
<tr>
<td>windows_8</td>
<td>Windows 8 32-bit.</td>
</tr>
<tr>
<td>windows_8x64</td>
<td>Windows 8 64-bit.</td>
</tr>
<tr>
<td>windows_xp</td>
<td>Windows XP.</td>
</tr>
</tbody>
</table>

### 7.189. PACKAGE STRUCT

Type representing a package.

This is an example of the package element:

```xml
<package>
  <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
</package>
```

Table 7.254. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name of the package.</td>
</tr>
</tbody>
</table>

### 7.190. PAYLOAD STRUCT

Table 7.255. Attributes summary
Table 7.256. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>base64</td>
<td></td>
</tr>
<tr>
<td>plaintext</td>
<td></td>
</tr>
</tbody>
</table>

7.192. PERMISSION STRUCT

Type represents a permission.

Table 7.257. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

Table 7.258. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster.</td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>Reference to data center.</td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>Reference to disk.</td>
</tr>
<tr>
<td>group</td>
<td>Group</td>
<td>Reference to group.</td>
</tr>
</tbody>
</table>
### Table 7.259. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>administrative</code></td>
<td>Boolean</td>
<td>Specifies whether permit is administrative or not.</td>
</tr>
<tr>
<td><code>comment</code></td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td><code>description</code></td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td><code>id</code></td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td><code>name</code></td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### Table 7.260. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>role</td>
<td>Role</td>
<td>Reference to the role the permit belongs to.</td>
</tr>
</tbody>
</table>

### 7.194. PMPROXY STRUCT

Table 7.261. Attributes summary
### 7.195. PMPROXYTYPE ENUM

Table 7.262. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>The fence proxy is selected from the same cluster as the fenced host.</td>
</tr>
<tr>
<td>dc</td>
<td>The fence proxy is selected from the same data center as the fenced host.</td>
</tr>
<tr>
<td>other_dc</td>
<td>The fence proxy is selected from a different data center than the fenced host.</td>
</tr>
</tbody>
</table>

### 7.196. POLICYUNITTYPE ENUM

Holds the types of all internal policy unit types.

Table 7.263. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td></td>
</tr>
<tr>
<td>load_balancing</td>
<td></td>
</tr>
<tr>
<td>weight</td>
<td></td>
</tr>
</tbody>
</table>

### 7.197. PORTMIRRORING STRUCT

### 7.198. POWERMANAGEMENT STRUCT

Table 7.264. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>The host name or IP address of the host.</td>
</tr>
<tr>
<td>agents</td>
<td>Agent[]</td>
<td>Specifies fence agent options when multiple fences are used.</td>
</tr>
<tr>
<td>automatic_pm_enabled</td>
<td>Boolean</td>
<td>Toggles the automated power control of the host in order to save energy.</td>
</tr>
</tbody>
</table>
### enabled

**Type:** Boolean

Indicates whether power management configuration is enabled or disabled.

### kdump_detection

**Type:** Boolean

Toggles whether to determine if kdump is running on the host before it is shut down.

### options

**Type:** Option[]

Fencing options for the selected type= specified with the option name="" and value="" strings.

### password

**Type:** String

A valid, robust password for power management.

### pm_proxies

**Type:** PmProxy[]

Determines the power management proxy.

### status

**Type:** PowerManagementStatus

Determines the power status of the host.

### type

**Type:** String

Fencing device code.

### username

**Type:** String

A valid user name for power management.

---

#### 7.198.1. agents

Specifies fence agent options when multiple fences are used.

Use the order sub-element to prioritize the fence agents. Agents are run sequentially according to their order until the fence action succeeds. When two or more fence agents have the same order, they are run concurrently. Other sub-elements include type, ip, user, password, and options.

#### 7.198.2. automatic_pm_enabled

Toggles the automated power control of the host in order to save energy. When set to true, the host will be automatically powered down if the cluster’s load is low, and powered on again when required. This is set to true when a host is created, unless disabled by the user.

#### 7.198.3. kdump_detection

Toggles whether to determine if kdump is running on the host before it is shut down. When set to true, the host will not shut down during a kdump process. This is set to true when a host has power management enabled, unless disabled by the user.

#### 7.198.4. type

Fencing device code.

A list of valid fencing device codes are available in the capabilities collection.
7.199. POWERMANAGEMENTSTATUS ENUM

Table 7.265. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>Host is OFF.</td>
</tr>
<tr>
<td>on</td>
<td>Host is ON.</td>
</tr>
<tr>
<td>unknown</td>
<td>Unknown status.</td>
</tr>
</tbody>
</table>

7.200. PRODUCT STRUCT

Table 7.266. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

7.201. PRODUCTINFO STRUCT

Product information.

The entry point contains a `product_info` element to help an API user determine the legitimacy of the Red Hat Virtualization environment. This includes the name of the product, the vendor and the version.

Verify a genuine Red Hat Virtualization environment

The follow elements identify a genuine Red Hat Virtualization environment:

```xml
<api>
  ...
  <product_info>
    <name>oVirt Engine</name>
    <vendor>ovirt.org</vendor>
    <version>
      <build>0</build>
      <full_version>4.1.0_master</full_version>
      <major>4</major>
      <minor>1</minor>
      <revision>0</revision>
    </version>
  </product_info>
</api>
```
7.201.1. vendor

The name of the vendor, for example ovirt.org.

7.202. PROFILEDETAIL STRUCT

Table 7.268. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_statistics</td>
<td>BlockStatistic[]</td>
<td></td>
</tr>
<tr>
<td>duration</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>fop_statistics</td>
<td>FopStatistic[]</td>
<td></td>
</tr>
<tr>
<td>profile_type</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td></td>
</tr>
</tbody>
</table>

7.203. PROPERTY STRUCT

Table 7.269. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

7.204. PROXYTICKET STRUCT
7.205. QCOWVERSION ENUM

The QCOW version specifies to the qemu which qemu version the volume supports.

This field can be updated using the update API and will be reported only for QCOW volumes, it is determined by the storage domain’s version which the disk is created on. Storage domains with version lower than V4 support QCOW2 version 2 volumes, while V4 storage domains also support QCOW2 version 3. For more information about features of the different QCOW versions, see here.

Table 7.271. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>qcow2_v2</td>
<td>The Copy On Write default compatibility version It means that every QEMU can use it.</td>
</tr>
<tr>
<td>qcow2_v3</td>
<td>The Copy On Write compatibility version which was introduced in QEMU 1.</td>
</tr>
</tbody>
</table>

7.205.1. qcow2_v3

The Copy On Write compatibility version which was introduced in QEMU 1.1 It means that the new format is in use.

7.206. QOS STRUCT

This type represents the attributes to define Quality of service (QoS).

For storage the type is storage, the attributes max_throughput, max_read_throughput, max_write_throughput, max_iops, max_read_iops and max_write_iops are relevant.

For resources with computing capabilities the type is cpu, the attribute cpu_limit is relevant.

For virtual machines networks the type is network, the attributes inbound_average, inbound_peak, inbound_burst, outbound_average, outbound_peak and outbound_burst are relevant.

For host networks the type is hostnetwork, the attributes outbound_average_linkshare, outbound_average_upperlimit and outbound_average_realtime are relevant.

Table 7.272. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>cpu_limit</td>
<td>Integer</td>
<td>The maximum processing capability in %.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>inbound_average</td>
<td>Integer</td>
<td>The desired average inbound bit rate in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>inbound_burst</td>
<td>Integer</td>
<td>The amount of data that can be delivered in a single burst, in MB.</td>
</tr>
<tr>
<td>inbound_peak</td>
<td>Integer</td>
<td>The maximum inbound rate in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>max_iops</td>
<td>Integer</td>
<td>Maximum permitted number of input and output operations per second.</td>
</tr>
<tr>
<td>max_read_iops</td>
<td>Integer</td>
<td>Maximum permitted number of input operations per second.</td>
</tr>
<tr>
<td>max_read_throughput</td>
<td>Integer</td>
<td>Maximum permitted throughput for read operations.</td>
</tr>
<tr>
<td>max_throughput</td>
<td>Integer</td>
<td>Maximum permitted total throughput.</td>
</tr>
<tr>
<td>max_write_iops</td>
<td>Integer</td>
<td>Maximum permitted number of output operations per second.</td>
</tr>
<tr>
<td>max_write_throughput</td>
<td>Integer</td>
<td>Maximum permitted throughput for write operations.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>outbound_average</td>
<td>Integer</td>
<td>The desired average outbound bit rate in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>outbound_average_linkshare</td>
<td>Integer</td>
<td>Weighted share.</td>
</tr>
<tr>
<td>outbound_average_realtime</td>
<td>Integer</td>
<td>The committed rate in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>outbound_average_upperlimit</td>
<td>Integer</td>
<td>The maximum bandwidth to be used by a network in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>outbound_burst</td>
<td>Integer</td>
<td>The amount of data that can be sent in a single burst, in MB.</td>
</tr>
<tr>
<td>outbound_peak</td>
<td>Integer</td>
<td>The maximum outbound rate in Mbps (Megabits per sec).</td>
</tr>
<tr>
<td>type</td>
<td>QosType</td>
<td>The kind of resources this entry can be assigned.</td>
</tr>
</tbody>
</table>

### 7.206.1. cpu_limit

The maximum processing capability in %.

Used to configure computing resources.

### 7.206.2. inbound_average

The desired average inbound bit rate in Mbps (Megabits per sec).

Used to configure virtual machines networks. If defined, `inbound_peak` and `inbound_burst` also has to be set.

See Libvirt-QOS for further details.

### 7.206.3. inbound_burst

The amount of data that can be delivered in a single burst, in MB.

Used to configure virtual machine networks. If defined, `inbound_average` and `inbound_peak` must also be set.

See Libvirt-QOS for further details.

### 7.206.4. inbound_peak

The maximum inbound rate in Mbps (Megabits per sec).

Used to configure virtual machines networks. If defined, `inbound_average` and `inbound_burst` also has to be set.

See Libvirt-QOS for further details.

### 7.206.5. max_iops

Maximum permitted number of input and output operations per second.

Used to configure storage. Must not be set if `max_read_iops` or `max_write_iops` is set.

### 7.206.6. max_read_iops
Maximum permitted number of input operations per second.
Used to configure storage. Must not be set if `max_iops` is set.

### 7.206.7. max_read_throughput
Maximum permitted throughput for read operations.
Used to configure storage. Must not be set if `max_throughput` is set.

### 7.206.8. max_throughput
Maximum permitted total throughput.
Used to configure storage. Must not be set if `max_read_throughput` or `max_write_throughput` is set.

### 7.206.9. max_write_iops
Maximum permitted number of output operations per second.
Used to configure storage. Must not be set if `max_iops` is set.

### 7.206.10. max_write_throughput
Maximum permitted throughput for write operations.
Used to configure storage. Must not be set if `max_throughput` is set.

### 7.206.11. outbound_average
The desired average outbound bit rate in Mbps (Megabits per sec).
Used to configure virtual machines networks. If defined, `outbound_peak` and `outbound_burst` also has to be set.

See [Libvirt-QOS](https://www.redhat.com/en) for further details.

### 7.206.12. outbound_average_linkshare
Weighted share.
Used to configure host networks. Signifies how much of the logical link’s capacity a specific network should be allocated, relative to the other networks attached to the same logical link. The exact share depends on the sum of shares of all networks on that link. By default this is a number in the range 1-100.

### 7.206.13. outbound_average_realtime
The committed rate in Mbps (Megabits per sec).
Used to configure host networks. The minimum bandwidth required by a network. The committed rate requested is not guaranteed and will vary depending on the network infrastructure and the committed rate requested by other networks on the same logical link.
7.206.14. outbound_average_upperlimit
The maximum bandwidth to be used by a network in Mbps (Megabits per sec).

Used to configure host networks. If outboundAverageUpperlimit and outbound_average_realtime are provided, the outbound_averageUpperlimit must not be lower than the outbound_average_realtime.

See Libvirt-QOS for further details.

7.206.15. outbound_burst
The amount of data that can be sent in a single burst, in MB.

Used to configure virtual machine networks. If defined, outbound_average and outbound_peak must also be set.

See Libvirt-QOS for further details.

7.206.16. outbound_peak
The maximum outbound rate in Mbps (Megabits per sec).

Used to configure virtual machines networks. If defined, outbound_average and outbound_burst also has to be set.

See Libvirt-QOS for further details.

Table 7.273. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>The data center the QoS is assiciated to.</td>
</tr>
</tbody>
</table>

7.207. QOSTYPE ENUM
This type represents the kind of resource the Quality of service (QoS) can be assigned to.

Table 7.274. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>The Quality of service (QoS) can be assigned to resources with computing capabilities.</td>
</tr>
<tr>
<td>hostnetwork</td>
<td>The Quality of service (QoS) can be assigned to host networks.</td>
</tr>
<tr>
<td>network</td>
<td>The Quality of service (QoS) can be assigned to virtual machines networks.</td>
</tr>
<tr>
<td>storage</td>
<td>The Quality of service (QoS) can be assigned to storage.</td>
</tr>
</tbody>
</table>

7.208. QUOTA STRUCT
Represents a quota object.

An example XML representation of a quota:

```xml
<quota href="/ovirt-engine/api/datacenters/7044934e/quotas/dcad5ddc"
      id="dcad5ddc">
  <name>My Quota</name>
  <description>A quota for my oVirt environment</description>
  <cluster_hard_limit_pct>0</cluster_hard_limit_pct>
  <cluster_soft_limit_pct>0</cluster_soft_limit_pct>
  <data_center href="/ovirt-engine/api/datacenters/7044934e"
               id="7044934e">
    <storage_hard_limit_pct>0</storage_hard_limit_pct>
    <storage_soft_limit_pct>0</storage_soft_limit_pct>
  </data_center>
  <storage_hard_limit_pct>0</storage_hard_limit_pct>
  <storage_soft_limit_pct>0</storage_soft_limit_pct>
</quota>
```

Table 7.275. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster_hard_limit_pct</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>cluster_soft_limit_pct</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>disks</td>
<td>Disk[]</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>storage_hard_limit_pct</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>storage_soft_limit_pct</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>users</td>
<td>User[]</td>
<td></td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.276. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>quota_cluster_limits</td>
<td>QuotaClusterLimit[]</td>
<td></td>
</tr>
<tr>
<td>quota_storage_limits</td>
<td>QuotaStorageLimit[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.209. QUOTA_CLUSTERLIMIT STRUCT

#### Table 7.277. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>memory_limit</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>memory_usage</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>vcpu_limit</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>vcpu_usage</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

#### Table 7.278. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td></td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td></td>
</tr>
</tbody>
</table>

### 7.210. QUOTA_MODETYPE ENUM

#### Table 7.279. Values summary
### 7.211. QUOTASTORAGELIMIT STRUCT

**Table 7.280. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>limit</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>usage</td>
<td>Decimal</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.281. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>quota</td>
<td>Quota</td>
<td></td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td></td>
</tr>
</tbody>
</table>

### 7.212. RANGE STRUCT

**Table 7.282. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>to</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
7.213. RATE STRUCT

Determines maximum speed of consumption of bytes from random number generator device.

Table 7.283. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes</td>
<td>Integer</td>
<td>Number of bytes allowed to consume per period.</td>
</tr>
<tr>
<td>period</td>
<td>Integer</td>
<td>Duration of one period in milliseconds.</td>
</tr>
</tbody>
</table>

7.214. REGISTRATIONAFFINITYGROUPMAPPING STRUCT

This type describes how to map affinity groups as part of the object registration. An object can be a virtual machine, template, etc.

An example of an XML representation using this mapping:

```xml
<action>
  <registration_configuration>
    <affinity_group_mappings>
      <registration_affinity_group_mapping>
        <from>
          <name>affinity</name>
        </from>
        <to>
          <name>affinity2</name>
        </to>
      </registration_affinity_group_mapping>
    </affinity_group_mappings>
  </registration_configuration>
</action>
```

Table 7.284. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>AffinityGroup</td>
<td>Reference to the original affinity group.</td>
</tr>
<tr>
<td>to</td>
<td>AffinityGroup</td>
<td>Reference to the destination affinity group.</td>
</tr>
</tbody>
</table>

7.214.1. from

Reference to the original affinity group. It can be specified using name.

7.215. REGISTRATIONAFFINITYLABELMAPPING STRUCT

This type describes how to map affinity labels as part of the object registration. An object can be a virtual machine, template, etc.
An example of an XML representation using mapping:

```
<action>
  <registration_configuration>
    <affinity_label_mappings>
      <registration_affinity_label_mapping>
        <from>
          <name>affinity_label</name>
        </from>
        <to>
          <name>affinity_label2</name>
        </to>
      </registration_affinity_label_mapping>
    </affinity_label_mappings>
  </registration_configuration>
</action>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>AffinityLabel</td>
<td>Reference to the original affinity label.</td>
</tr>
<tr>
<td>to</td>
<td>AffinityLabel</td>
<td>Reference to the destination affinity label.</td>
</tr>
</tbody>
</table>

### 7.215.1. from

Reference to the original affinity label. It can be specified using **name**.

### 7.216. REGISTRATIONCLUSTERMAPPING STRUCT

This type describes how to map clusters as part of the object registration. An object can be a virtual machine, template, etc.

An example of an XML representation using this mapping:

```
<action>
  <registration_configuration>
    <cluster_mappings>
      <registration_cluster_mapping>
        <from>
          <name>myoriginalcluster</name>
        </from>
        <to>
          <name>mynewcluster</name>
        </to>
      </registration_cluster_mapping>
    </cluster_mappings>
  </registration_configuration>
</action>
```

Table 7.286. Links summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>Cluster</td>
<td>Reference to the original cluster.</td>
</tr>
<tr>
<td>to</td>
<td>Cluster</td>
<td>Reference to the destination cluster.</td>
</tr>
</tbody>
</table>

### 7.216.1. from

Reference to the original cluster. It can be specified using the **id** or the **name**.

### 7.216.2. to

Reference to the destination cluster. It can be specified using the **id** or the **name**.

### 7.217. REGISTRATIONCONFIGURATION STRUCT

This type describes how an object (virtual machine, template, etc) is registered, and is used for the implementation of disaster recovery solutions.

Each mapping contained in this type can be used to map objects in the original system to corresponding objects in the system where the virtual machine or template is being registered. For example, there could be a primary setup with a virtual machine configured on cluster A, and an active secondary setup with cluster B. Cluster B is compatible with that virtual machine, and in case of a disaster recovery scenario the storage domain can be imported to the secondary setup, and the user can register the virtual machine to cluster B.

In that case, we can automate the recovery process by defining a cluster mapping. After the entity is registered, its OVF will indicate it belongs to cluster A, but the mapping will indicate that cluster A will be replaced with cluster B. Red Hat Virtualization Manager should do the switch and register the virtual machine to cluster B in the secondary site.

Cluster mapping is just one example, there are different types of mappings:

- Cluster mapping.
- LUN mapping.
- Role mapping.
- Domain mapping.
- Permissions mapping.
- Affinity Group mapping.
- Affinity Label mapping.
- Virtual NIC profile mapping.

Each mapping will be used for its specific OVF’s data once the register operation takes place in the Red Hat Virtualization Manager.

An example of an XML representation using the mapping:
<action>
  <registration_configuration>
    <cluster_mappings>
      <registration_cluster_mapping>
        <from>
          <name>myoriginalcluster</name>
        </from>
        <to>
          <name>mynewcluster</name>
        </to>
      </registration_cluster_mapping>
    </cluster_mappings>
    <role_mappings>
      <registration_role_mapping>
        <from>
          <name>SuperUser</name>
        </from>
        <to>
          <name>UserVmRunTimeManager</name>
        </to>
      </registration_role_mapping>
    </role_mappings>
    <domain_mappings>
      <registration_domain_mapping>
        <from>
          <name>redhat</name>
        </from>
        <to>
          <name>internal</name>
        </to>
      </registration_domain_mapping>
    </domain_mappings>
    <lun_mappings>
      <registration_lun_mapping>
        <from id="111"/>
        <to id="222">
          <alias>weTestLun</alias>
          <lun_storage>
            <type>iscsi</type>
            <logical_units>
              <logical_unit id="36001405fb1db491e44078f1fffcecf">
                <address>44.33.11.22</address>
                <port>3260</port>
                <portal>1</portal>
              </logical_unit>
            </logical_units>
          </lun_storage>
        </to>
      </registration_lun_mapping>
    </lun_mappings>
    <affinity_group_mappings>
      <registration_affinity_group_mapping>
        <from>
          <name>affinity</name>
        </from>
      </registration_affinity_group_mapping>
    </affinity_group_mappings>
  </registration_configuration>
</action>
Table 7.287. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity_group_mappings</td>
<td>RegistrationAffinityGroupMapping[]</td>
<td>Describes how the affinity groups are mapped.</td>
</tr>
<tr>
<td>affinity_label_mappings</td>
<td>RegistrationAffinityLabelMapping[]</td>
<td>Describes how the affinity labels are mapped.</td>
</tr>
</tbody>
</table>
### cluster_mappings

- **Type**: RegistrationClusterMapping[
- **Summary**: Describes how the clusters that the object references are mapped.

### domain_mappings

- **Type**: RegistrationDomainMapping[
- **Summary**: Describes how the users' domains are mapped.

### lun_mappings

- **Type**: RegistrationLunMapping[
- **Summary**: Describes how the LUNs are mapped.

### role_mappings

- **Type**: RegistrationRoleMapping[
- **Summary**: Describes how the roles are mapped.

### vnic_profile_mappings

- **Type**: RegistrationVnicProfileMapping[
- **Summary**: Mapping rules for virtual NIC profiles that will be applied during the register process.

### 7.218. REGISTRATIONDOMAINMAPPING STRUCT

This type describes how to map the users' domain as part of the object registration. An object can be a virtual machine, template, etc. NOTE: This is based on the assumption that user names will be the same, and that only the domain name will be changed.

An example of an XML representation using this mapping:

```xml
<action>
  <registration_configuration>
    <domain_mappings>
      <registration_domain_mapping>
        <from>
          <name>redhat</name>
        </from>
        <to>
          <name>internal</name>
        </to>
      </registration_domain_mapping>
    </domain_mappings>
  </registration_configuration>
</action>
```

### Table 7.288. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>Domain</td>
<td>Reference to the original domain.</td>
</tr>
<tr>
<td>to</td>
<td>Domain</td>
<td>Reference to the destination domain.</td>
</tr>
</tbody>
</table>
7.218.1. from
Reference to the original domain. It can be specified using name.

7.219. REGISTRATIONLUNMAPPING STRUCT
This type describes how to map LUNs as part of the object registration. An object can be a virtual machine, template, etc.

An external LUN disk is an entity which does not reside on a storage domain. It must be specified because it doesn’t need to exist in the environment where the object is registered. An example of an XML representation using this mapping:

```xml
<action>
  <registration_configuration>
    <lun_mappings>
      <registration_lun_mapping>
        <from id="111"/>
        <to id="222">
          <alias>weTestLun</alias>
          <lun_storage>
            <type>iscsi</type>
            <logical_units>
              <logical_unit id="36001405fb1dddb91e44078f1fffcfef">
                <address>44.33.11.22</address>
                <port>3260</port>
                <portal>1</portal>
              </logical_unit>
            </logical_units>
          </lun_storage>
        </to>
      </registration_lun_mapping>
    </lun_mappings>
  </registration_configuration>
</action>
```

Table 7.289. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>Disk</td>
<td>Reference to the original LUN.</td>
</tr>
<tr>
<td>to</td>
<td>Disk</td>
<td>Reference to the LUN which is to be added to the virtual machine.</td>
</tr>
</tbody>
</table>

7.219.1. from
Reference to the original LUN. This must be specified using the id attribute.
7.220. REGISTRATIONROLEMAPPING STRUCT

This type describes how to map roles as part of the object registration. An object can be a virtual machine, template, etc.

A role mapping is intended to map correlating roles between the primary site and the secondary site. For example, there may be permissions with role `UserVmRunTimeManager` for the virtual machine that is being registered. Therefore we can send a mapping that will register the virtual machine in the secondary setup using the `SuperUser` role instead of `UserVmRunTimeManager`. An example of an XML representation using this mapping:

```xml
<action>
  <registration_configuration>
    <role_mappings>
      <registration_role_mapping>
        <from>
          <name>SuperUser</name>
        </from>
        <to>
          <name>UserVmRunTimeManager</name>
        </to>
      </registration_role_mapping>
    </role_mappings>
  </registration_configuration>
</action>
```

### Table 7.290. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>Role</td>
<td>Reference to the original role.</td>
</tr>
<tr>
<td>to</td>
<td>Role</td>
<td>Reference to the destination role.</td>
</tr>
</tbody>
</table>

#### 7.220.1. from

Reference to the original role. It can be specified using `name`.

7.221. REGISTRATIONVNICPROFILEMAPPING STRUCT

Maps an external virtual NIC profile to one that exists in the Red Hat Virtualization Manager. The target may be specified as a profile ID or a pair of profile name and network name.

If, for example, the desired virtual NIC profile mapping includes the following lines:

<table>
<thead>
<tr>
<th>Source network name</th>
<th>Source network profile name</th>
<th>Target virtual NIC profile ID\names</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>gold</td>
<td>738dd914-8ec8-4a8b-8628-34672a5d449b</td>
</tr>
<tr>
<td>Source network name</td>
<td>Source network profile name</td>
<td>Target virtual NIC profile ID\names</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>&lt;empty&gt; (no network name)</td>
<td>&lt;empty&gt; (no network profile name)</td>
<td>892a12ec-2028-4451-80aa-ff3bf55d6bac</td>
</tr>
<tr>
<td>blue</td>
<td>silver</td>
<td>orange\copper</td>
</tr>
<tr>
<td>yellow</td>
<td>platinum</td>
<td>&lt;empty&gt; (no profile)</td>
</tr>
<tr>
<td>green</td>
<td>bronze</td>
<td></td>
</tr>
</tbody>
</table>

Then the following snippet should be added to `RegistrationConfiguration`:

```xml
<vnic_profile_mappings>
  <registration_vnic_profile_mapping>
    <from>
      <name>gold</name>
      <network>
        <name>red</name>
      </network>
    </from>
    <to id="738dd914-8ec8-4a8b-8628-34672a5d449b"/>
  </registration_vnic_profile_mapping>
  <registration_vnic_profile_mapping>
    <from>
      <name>silver</name>
      <network>
        <name>blue</name>
      </network>
    </from>
    <to id="892a12ec-2028-4451-80aa-ff3bf55d6bac"/>
  </registration_vnic_profile_mapping>
  <registration_vnic_profile_mapping>
    <from>
      <name>platinum</name>
      <network>
        <name>yellow</name>
      </network>
    </from>
    <to>
      <name>copper</name>
      <network>
        <name>orange</name>
      </network>
    </to>
  </registration_vnic_profile_mapping>
</vnic_profile_mappings>
```
Table 7.291. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>from</td>
<td>VnicProfile</td>
<td>References to the external network and the external network profile.</td>
</tr>
<tr>
<td>to</td>
<td>VnicProfile</td>
<td>Reference to to an existing virtual NIC profile.</td>
</tr>
</tbody>
</table>

7.221.1. from

References to the external network and the external network profile. Both should be specified using their name.

7.221.2. to

Reference to to an existing virtual NIC profile. It should be specified using its name or id. Either name or id should be specified but not both.

7.222. REPORTEDCONFIGURATION STRUCT

Table 7.292. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>actual_value</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>expected_value</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>in_sync</td>
<td>Boolean</td>
<td><strong>false</strong> when the network attachment contains uncommitted network configuration.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
7.223. REPORTEDDEVICE STRUCT

Table 7.293. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ips</td>
<td>Ip[]</td>
<td></td>
</tr>
<tr>
<td>mac</td>
<td>Mac</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>type</td>
<td>ReportedDeviceType</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.294. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>vm</td>
<td>Vm</td>
<td></td>
</tr>
</tbody>
</table>

7.224. REPORTEDDEVICETYPE ENUM

Table 7.295. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td></td>
</tr>
</tbody>
</table>

7.225. RESOLUTIONTYPE ENUM

Table 7.296. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td></td>
</tr>
<tr>
<td>copy</td>
<td></td>
</tr>
</tbody>
</table>

7.226. RNGDEVICE STRUCT
Random number generator (RNG) device model.

Table 7.297. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>rate</td>
<td>Rate</td>
<td>Determines maximum speed of consumption of bytes from random number generator device.</td>
</tr>
<tr>
<td>source</td>
<td>RngSource</td>
<td>Backend of the random number generator device.</td>
</tr>
</tbody>
</table>

7.227. RNGSOURCE ENUM

Representing the random generator backend types.

Table 7.298. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>hwrng</td>
<td>Obtains random data from the /dev/hwrng (usually specialized HW generator) device.</td>
</tr>
<tr>
<td>random</td>
<td>Obtains random data from the /dev/random device.</td>
</tr>
<tr>
<td>urandom</td>
<td>Obtains random data from the /dev/urandom device.</td>
</tr>
</tbody>
</table>

7.227.1. urandom

Obtains random data from the /dev/urandom device.

This RNG source is meant to replace random RNG source for non-cluster-aware entities (i.e. Blank template and instance types) and entities associated with clusters with compatibility version 4.1 or higher.

7.228. ROLE STRUCT

Represents a system role.

Table 7.299. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>administrative</td>
<td>Boolean</td>
<td>Defines the role as administrative-only or not.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
</tbody>
</table>
### 7.228. `mutable`

Defines the ability to update or delete the role.

Roles with `mutable` set to `false` are predefined roles.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>mutable</td>
<td>Boolean</td>
<td>Defines the ability to update or delete the role.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

#### Table 7.300. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>permits</td>
<td>Permit[]</td>
<td>A link to the permits sub-collection for role permits.</td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td></td>
</tr>
</tbody>
</table>

### 7.229. `ROLETYPE` **ENUM**

Type representing whether a role is administrative or not. A user which was granted at least one administrative role is considered an administrator.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Administrative role.</td>
</tr>
<tr>
<td>user</td>
<td>User role.</td>
</tr>
</tbody>
</table>

### 7.230. `SCHEDULINGPOLICY` **STRUCT**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>default_policy</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
</tbody>
</table>
## 7.231. SCHEDULINGPOLICYUNIT STRUCT

### Table 7.304. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>internal</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>properties</td>
<td>Property[]</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>PolicyUnitType</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7.303. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>balances</td>
<td>Balance[]</td>
<td></td>
</tr>
<tr>
<td>filters</td>
<td>Filter[]</td>
<td></td>
</tr>
<tr>
<td>weight</td>
<td>Weight[]</td>
<td></td>
</tr>
</tbody>
</table>

## 7.232. SCSIGENERICIO ENUM

### Table 7.305. Values summary
### Table 7.306. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>SeLinuxMode</td>
<td>SELinux current mode.</td>
</tr>
</tbody>
</table>

### Table 7.307. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>SELinux is disabled in the kernel.</td>
</tr>
<tr>
<td>enforcing</td>
<td>SELinux is running and enforcing permissions.</td>
</tr>
<tr>
<td>permissive</td>
<td>SELinux is running and logging but not enforcing permissions.</td>
</tr>
</tbody>
</table>

### Table 7.308. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>policy</td>
<td>SerialNumberPolicy</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7.309. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
</table>

### 7.233. SELINUX STRUCT

Represents SELinux in the system.

### 7.234. SELINUXMODE ENUM

Represents an SELinux enforcement mode.

### 7.235. SERIALNUMBER STRUCT

### 7.236. SERIALNUMBERPOLICY ENUM
### 7.237. SESSION STRUCT

Describes a user session to a virtual machine.

**Table 7.310. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console_user</td>
<td>Boolean</td>
<td>Indicates if this is a console session.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>ip</td>
<td>Ip</td>
<td>The IP address the user is connected from.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>protocol</td>
<td>String</td>
<td>The protocol used by the session.</td>
</tr>
</tbody>
</table>

#### 7.237.1. console_user

Indicates if this is a console session.

The value will be **true** for console users (SPICE or VNC), and **false** for others (such as RDP or SSH).

#### 7.237.2. ip

The IP address the user is connected from.

Currently only available for console users.

#### 7.237.3. protocol

The protocol used by the session.

Currently not used. Intended for info about how the user is connected: through SPICE, VNC, SSH, or RDP.
Table 7.311. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>User</td>
<td>The user related to this session.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>A link to the virtual machine related to this session.</td>
</tr>
</tbody>
</table>

7.237.4. user

The user related to this session.

If the user is a console user, this is a link to the real Red Hat Virtualization user. Otherwise, only the user name is provided.

7.238. SKIPIFCONNECTIVITYBROKEN STRUCT

Table 7.312. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>If enabled, we will not fence a host in case more than a configurable percentage of hosts in the cluster lost connectivity as well.</td>
</tr>
<tr>
<td>threshold</td>
<td>Integer</td>
<td>Threshold for connectivity testing.</td>
</tr>
</tbody>
</table>

7.238.1. enabled

If enabled, we will not fence a host in case more than a configurable percentage of hosts in the cluster lost connectivity as well. This comes to prevent fencing storm in cases where there is a global networking issue in the cluster.

7.238.2. threshold

Threshold for connectivity testing. If at least the threshold percentage of hosts in the cluster lost connectivity then fencing will not take place.

7.239. SKIPIFSDACTIVE STRUCT

This type represents the storage related configuration in the fencing policy.

Table 7.313. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>If enabled, we will skip fencing in case the host maintains its lease in the storage.</td>
</tr>
</tbody>
</table>
7.239.1. enabled
If enabled, we will skip fencing in case the host maintains its lease in the storage. It means that if the host still has storage access then it won’t get fenced.

7.240. SNAPSHOT STRUCT
Represents a snapshot object.

Example XML representation:

```
<snapshot id="456" href="/ovirt-engine/api/vms/123/snapshots/456">
  <actions>
    <link rel="restore" href="/ovirt-engine/api/vms/123/snapshots/456/restore"/>
  </actions>
  <vm id="123" href="/ovirt-engine/api/vms/123"/>
  <description>Virtual Machine 1 - Snapshot A</description>
  <type>active</type>
  <date>2010-08-16T14:24:29</date>
  <persist_memorystate>false</persist_memorystate>
</snapshot>
```

Table 7.314. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios</td>
<td>Bios</td>
<td>Reference to virtual machine’s BIOS configuration.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console</td>
<td>Console</td>
<td>Console configured for this virtual machine.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The configuration of the virtual machine CPU.</td>
</tr>
<tr>
<td>cpu_shares</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>creation_time</td>
<td>Date</td>
<td>The virtual machine creation date.</td>
</tr>
<tr>
<td>custom_compatibility_version</td>
<td>Version</td>
<td>Virtual machine custom compatibility version.</td>
</tr>
<tr>
<td>custom_cpu_model</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_emulated_machine</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Properties sent to VDSM to configure various hooks.</td>
</tr>
<tr>
<td>date</td>
<td>Date</td>
<td>The date when this snapshot has been created.</td>
</tr>
<tr>
<td>delete_protected</td>
<td>Boolean</td>
<td>If true, the virtual machine cannot be deleted.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The virtual machine display configuration.</td>
</tr>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Domain configured for this virtual machine.</td>
</tr>
<tr>
<td>fqdn</td>
<td>String</td>
<td>Fully qualified domain name of the virtual machine.</td>
</tr>
<tr>
<td>guestOperating_system</td>
<td>GuestOperatingSystem</td>
<td>What operating system is installed on the virtual machine.</td>
</tr>
<tr>
<td>guest_time_zone</td>
<td>TimeZone</td>
<td>What time zone is used by the virtual machine (as returned by guest agent).</td>
</tr>
<tr>
<td>has_illegal_images</td>
<td>Boolean</td>
<td>Indicates whether the virtual machine has snapshots with disks in ILLEGAL state.</td>
</tr>
<tr>
<td>high_availability</td>
<td>HighAvailability</td>
<td>The virtual machine high availability configuration.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialization</td>
<td>Reference to the virtual machine’s initialization configuration.</td>
</tr>
<tr>
<td>io</td>
<td>Io</td>
<td>For performance tuning of IO threading.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Virtual machine’s large icon.</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>Reference to the storage domain this virtual machine/template lease reside on.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The virtual machine’s memory, in bytes.</td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td>Reference to virtual machine’s memory management configuration.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td>Reference to configuration of migration of running virtual machine to another host.</td>
</tr>
<tr>
<td>migration_down_time</td>
<td>Integer</td>
<td>Maximum time the virtual machine can be non responsive during its live migration to another host in ms.</td>
</tr>
<tr>
<td>multi_queues_enabled</td>
<td>Boolean</td>
<td>If true, each virtual interface will get the optimal number of queues, depending on the available virtual Cpus.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>next_run_configuration_exists</td>
<td>Boolean</td>
<td>Virtual machine configuration has been changed and requires restart of the virtual machine.</td>
</tr>
<tr>
<td>numa_tune_mode</td>
<td>NumaTuneMode</td>
<td>How the NUMA topology is applied.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>The origin of this virtual machine.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>Operating system type installed on the virtual machine.</td>
</tr>
<tr>
<td>payloads</td>
<td>Payload[]</td>
<td>Optional payloads of the virtual machine, used for ISOs to configure it.</td>
</tr>
<tr>
<td>persist_memorystate</td>
<td>Boolean</td>
<td>Indicates if the content of the memory of the virtual machine is included in the snapshot.</td>
</tr>
<tr>
<td>placement_policy</td>
<td>VmPlacementPolicy</td>
<td>The configuration of the virtual machine’s placement policy.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>Random Number Generator device configuration for this virtual machine.</td>
</tr>
<tr>
<td>run_once</td>
<td>Boolean</td>
<td>If true, the virtual machine has been started using the run once command, meaning it’s configuration might differ from the stored one for the purpose of this single run.</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td>Virtual machine’s serial number in a cluster.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Virtual machine’s small icon.</td>
</tr>
<tr>
<td>snapshot_status</td>
<td>SnapshotStatus</td>
<td>Status of the snapshot.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>snapshot_type</td>
<td>SnapshotType</td>
<td>Type of the snapshot.</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>If true, the sound card is added to the virtual machine.</td>
</tr>
<tr>
<td>sso</td>
<td>Sso</td>
<td>Reference to the Single Sign On configuration this virtual</td>
</tr>
<tr>
<td></td>
<td></td>
<td>machine is configured for.</td>
</tr>
<tr>
<td>start_puased</td>
<td>Boolean</td>
<td>If true, the virtual machine will be initially in 'paused'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>state after start.</td>
</tr>
<tr>
<td>start_time</td>
<td>Date</td>
<td>The date in which the virtual machine was started.</td>
</tr>
<tr>
<td>stateless</td>
<td>Boolean</td>
<td>If true, the virtual machine is stateless - it's state (disks)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>are rolled-back after shutdown.</td>
</tr>
<tr>
<td>status</td>
<td>VmStatus</td>
<td>The current status of the virtual machine.</td>
</tr>
<tr>
<td>status_detail</td>
<td>String</td>
<td>Human readable detail of current status.</td>
</tr>
<tr>
<td>stop_reason</td>
<td>String</td>
<td>The reason the virtual machine was stopped.</td>
</tr>
<tr>
<td>stop_time</td>
<td>Date</td>
<td>The date in which the virtual machine was stopped.</td>
</tr>
<tr>
<td>storage_error_resume_behaviour</td>
<td>VmStorageErrorResumeBehaviour</td>
<td>Determines how the virtual machine will be resumed after</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage error.</td>
</tr>
<tr>
<td>time_zone</td>
<td>TimeZone</td>
<td>The virtual machine's time zone set by oVirt.</td>
</tr>
<tr>
<td>tunnel Migration</td>
<td>Boolean</td>
<td>If true, the network data transfer will be encrypted during</td>
</tr>
<tr>
<td></td>
<td></td>
<td>virtual machine live migration.</td>
</tr>
<tr>
<td>type</td>
<td>VmType</td>
<td>Determines whether the virtual machine is optimized for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>desktop or server.</td>
</tr>
<tr>
<td>usb</td>
<td>Usb</td>
<td>Configuration of USB devices for this virtual machine (count,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>type).</td>
</tr>
<tr>
<td>use_latest_template_version</td>
<td>Boolean</td>
<td>If true, the virtual machine is reconfigured to the latest</td>
</tr>
<tr>
<td></td>
<td></td>
<td>version of it's template when it is started.</td>
</tr>
<tr>
<td>virtio_scsi</td>
<td>VirtioScsi</td>
<td>Reference to VirtIO SCSI configuration.</td>
</tr>
</tbody>
</table>
7.240.1. cpu

The configuration of the virtual machine CPU.

The socket configuration can be updated without rebooting the virtual machine. The cores and the threads require a reboot.

For example, to change the number of sockets to 4 immediately, and the number of cores and threads to 2 after reboot, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With a request body:

```xml
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

7.240.2. custom_compatibility_version

Virtual machine custom compatibility version.

Enables a virtual machine to be customized to its own compatibility version. If `custom_compatibility_version` is set, it overrides the cluster’s compatibility version for this particular virtual machine.

The compatibility version of a virtual machine is limited by the data center the virtual machine resides in, and is checked against capabilities of the host the virtual machine is planned to run on.

7.240.3. high_availability

The virtual machine high availability configuration. If set, the virtual machine will be automatically restarted when it unexpectedly goes down.

7.240.4. initialization

Reference to the virtual machine’s initialization configuration.

**NOTE**

Since Red Hat Virtualization 4.1.8 this property can be cleared by sending an empty tag.

For example, to clear the `initialization` attribute send a request like this:

```
PUT /ovirt-engine/api/vms/123
```

With a request body like this:
The response to such a request, and requests with the header `All-Content: true` will still contain this attribute.

### 7.240.5. large_icon

Virtual machine's large icon. Either set by user or refers to image set according to operating system.

### 7.240.6. lease

Reference to the storage domain this virtual machine/template lease reside on.

A virtual machine running with a lease requires checking while running that the lease is not taken by another host, preventing another instance of this virtual machine from running on another host. This provides protection against split-brain in highly available virtual machines. A template can also have a storage domain defined for a lease in order to have the virtual machines created from this template to be preconfigured with this storage domain as the location of the leases.

### 7.240.7. memory

The virtual machine's memory, in bytes.

For example, to update a virtual machine to contain 1 Gibibyte (GiB) of memory, send the following request:

```plaintext
PUT /ovirt-engine/api/vms/123
```

With the following request body:

```xml
<vm>
  <initialization/>
</vm>
```

Memory hot plug is supported from Red Hat Virtualization 3.6 onwards. You can use the example above to increase memory while the virtual machine is in state `up`. The size increment must be dividable by the value of the `HotPlugMemoryBlockSizeMb` configuration value (256 MiB by default). If the memory size increment is not dividable by this value, the memory size change is only stored to next run configuration. Each successful memory hot plug operation creates one or two new memory devices.

Memory hot unplug is supported since Red Hat Virtualization 4.2 onwards. Memory hot unplug can only be performed when the virtual machine is in state `up`. Only previously hot plugged memory devices can be removed by the hot unplug operation. The requested memory decrement is rounded down to match sizes of a combination of previously hot plugged memory devices. The requested memory value is stored to next run configuration without rounding.

**NOTE**

Memory in the example is converted to bytes using the following formula:

\[ 1 \text{ GiB} = 2^{30} \text{ bytes} = 1073741824 \text{ bytes}. \]
NOTE

Red Hat Virtualization Manager internally rounds values down to whole MiBs (1MiB = $2^{20}$ bytes)

7.240.8. migration_downtime

Maximum time the virtual machine can be non responsive during its live migration to another host in ms.

Set either explicitly for the virtual machine or by `engine-config -s DefaultMaximumMigrationDowntime=[value]`

7.240.9. next_run_configuration_exists

Virtual machine configuration has been changed and requires restart of the virtual machine. Changed configuration is applied at processing the virtual machine’s `shut down`.

7.240.10. origin

The origin of this virtual machine.

Possible values:

- ovirt
- rhev
- vmware
- xen
- external
- hosted_engine
- managed_hosted_engine
- kvm
- physical_machine
- hyperv

7.240.11. persist_memorystate

Indicates if the content of the memory of the virtual machine is included in the snapshot.

When a snapshot is created the default value is true.

7.240.12. placement_policy

The configuration of the virtual machine’s placement policy.

This configuration can be updated to pin a virtual machine to one or more hosts.
**NOTE**

Virtual machines that are pinned to multiple hosts cannot be live migrated, but in the event of a host failure, any virtual machine configured to be highly available is automatically restarted on one of the other hosts to which the virtual machine is pinned.

For example, to pin a virtual machine to two hosts, send the following request:

```plaintext
PUT /api/vms/123
```

With a request body like this:

```xml
<vm>
  <high_availability>
    <enabled>true</enabled>
    <priority>1</priority>
  </high_availability>
  <placement_policy>
    <hosts>
      <host>
        <name>Host1</name>
      </host>
      <host>
        <name>Host2</name>
      </host>
    </hosts>
    <affinity>pinned</affinity>
  </placement_policy>
</vm>
```

**7.240.13. small_icon**

Virtual machine’s small icon. Either set by user or refers to image set according to operating system.

**7.240.14. sso**

Reference to the Single Sign On configuration this virtual machine is configured for. The user can be automatically signed in the virtual machine’s operating system when console is opened.

**7.240.15. stop_reason**

The reason the virtual machine was stopped. Optionally set by user when shutting down the virtual machine.

**Table 7.315. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity_labels</td>
<td>AffinityLabel[]</td>
<td>Optional.</td>
</tr>
<tr>
<td>applications</td>
<td>Application[]</td>
<td>List of applications installed on the virtual machine.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>Reference to the ISO mounted to the CDROM.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster the virtual machine belongs to.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Reference to CPU profile used by this virtual machine.</td>
</tr>
<tr>
<td>disk_attachments</td>
<td>DiskAttachment[]</td>
<td>References the disks attached to the virtual machine.</td>
</tr>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td></td>
</tr>
<tr>
<td>floppies</td>
<td>Floppy[]</td>
<td>Reference to the ISO mounted to the floppy.</td>
</tr>
<tr>
<td>graphics_consoles</td>
<td>GraphicsConsole[]</td>
<td>List of graphics consoles configured for this virtual machine.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host the virtual machine is running on.</td>
</tr>
<tr>
<td>host_devices</td>
<td>HostDevice[]</td>
<td>References devices associated to this virtual machine.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>The virtual machine configuration can be optionally predefined via one of the instance types.</td>
</tr>
<tr>
<td>katello_errata</td>
<td>KatelloErratum[]</td>
<td>Lists all the Katello errata assigned to the virtual machine.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>References the list of network interface devices on the virtual machine.</td>
</tr>
<tr>
<td>numa_nodes</td>
<td>NumaNode[]</td>
<td>Refers to the NUMA Nodes configuration used by this virtual machine.</td>
</tr>
<tr>
<td>original_template</td>
<td>Template</td>
<td>References the original template used to create the virtual machine.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Permissions set for this virtual machine.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Reference to quota configuration set for this virtual machine.</td>
</tr>
<tr>
<td>reported_devices</td>
<td>ReportedDevice[]</td>
<td></td>
</tr>
<tr>
<td>sessions</td>
<td>Session[]</td>
<td>List of user sessions opened for this virtual machine.</td>
</tr>
<tr>
<td>snapshots</td>
<td>Snapshot[]</td>
<td>Refers to all snapshots taken from the virtual machine.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Statistics data collected from this virtual machine.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to storage domain the virtual machine belongs to.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td></td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Reference to the template the virtual machine is based on.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>The virtual machine this snapshot has been taken for.</td>
</tr>
<tr>
<td>vm_pool</td>
<td>VmPool</td>
<td>Reference to the pool the virtual machine is optionally member of.</td>
</tr>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>Refers to the Watchdog configuration.</td>
</tr>
</tbody>
</table>

### 7.240.16. affinity_labels

Optional. Used for labeling of sub-clusters.

### 7.240.17. katello_errata

Lists all the Katello errata assigned to the virtual machine.

GET /ovirt-engine/api/vms/123/katelloerrata

You will receive response in XML like this one:

```xml
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>Few guidelines regarding the solution</solution>
    <summary>Updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```
7.240.18. original_template

References the original template used to create the virtual machine.

If the virtual machine is cloned from a template or another virtual machine, the template links to the Blank template, and the original_template is used to track history.

Otherwise the template and original_template are the same.

7.240.19. statistics

Statistics data collected from this virtual machine.

Note that some statistics, notably memory.buffered and memory.cached are available only when oVirt guest agent is installed in the virtual machine.

7.241. SNAPSHOTSTATUS ENUM

Represents the current status of the snapshot.

Table 7.316. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>in_preview</td>
<td>The snapshot is being previewed.</td>
</tr>
<tr>
<td>locked</td>
<td>The snapshot is locked.</td>
</tr>
<tr>
<td>ok</td>
<td>The snapshot is OK.</td>
</tr>
</tbody>
</table>

7.241.1. locked

The snapshot is locked.

The snapshot is locked when it is in process of being created, deleted, restored or previewed.

7.242. SNAPSHOTTYPE ENUM

Represents the type of the snapshot.

Table 7.317. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>active</td>
<td>Reference to the current configuration of the virtual machines.</td>
</tr>
<tr>
<td>preview</td>
<td>The active snapshot will become preview if some snapshot is being previewed.</td>
</tr>
<tr>
<td>regular</td>
<td>Snapshot created by user.</td>
</tr>
</tbody>
</table>
stateless
Snapshot created internally for stateless virtual machines.

7.242.1. preview
The active snapshot will become preview if some snapshot is being previewed.
In other words, this is the active snapshot before preview.

7.242.2. stateless
Snapshot created internally for stateless virtual machines.
This snapshot is created when the virtual machine is started and it is restored when the virtual machine is shut down.

7.243. SPECIALOBJECTS STRUCT
This type contains references to special objects, such as blank templates and the root of a hierarchy of tags.

Table 7.318. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>blank_template</td>
<td>Template</td>
<td>A reference to a blank template.</td>
</tr>
<tr>
<td>root_tag</td>
<td>Tag</td>
<td>A reference to the root of a hierarchy of tags.</td>
</tr>
</tbody>
</table>

7.244. SPM STRUCT

Table 7.319. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>priority</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>SpmStatus</td>
<td></td>
</tr>
</tbody>
</table>

7.245. SPMSTATUS ENUM

Table 7.320. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td>contending</td>
<td></td>
</tr>
<tr>
<td>none</td>
<td></td>
</tr>
<tr>
<td>spm</td>
<td></td>
</tr>
</tbody>
</table>

### 7.246. SSH STRUCT

#### Table 7.321. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>authentication_method</td>
<td>SshAuthenticationMethod</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>fingerprint</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td></td>
</tr>
</tbody>
</table>

### 7.247. SSHPUBLICKEY METHOD ENUM

#### Table 7.322. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>password</td>
<td></td>
</tr>
<tr>
<td>publickey</td>
<td></td>
</tr>
</tbody>
</table>

### 7.248. SSHPUBLICKEY STRUCT

#### Table 7.323. Attributes summary
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>content</td>
<td>String</td>
<td>Contains a saved SSH key.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### Table 7.324. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>user</td>
<td>User</td>
<td></td>
</tr>
</tbody>
</table>

### 7.249. SSO STRUCT

### Table 7.325. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>methods</td>
<td>Method[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.250. SSOMETHOD ENUM

### Table 7.326. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>guest_agent</td>
<td></td>
</tr>
</tbody>
</table>

### 7.251. STATISTIC STRUCT

A generic type used for all kinds of statistics.

Statistic contains the statistics values for various entities. The following object contain statistics:

- Disk
- Host
- HostNic
- NumaNode
An example of a XML representation:

```
<statistics>
  <statistic id="1234" href="/ovirt-engine/api/hosts/1234/nics/1234/statistics/1234">
    <name>data.current.rx</name>
    <description>Receive data rate</description>
    <values type="DECIMAL">
      <value>
        <datum>0</datum>
      </value>
    </values>
    <type>GAUGE</type>
    <unit>BYTES_PER_SECOND</unit>
    <host_nic id="1234" href="/ovirt-engine/api/hosts/1234/nics/1234"/>
  </statistic>
  ...
</statistics>
```

**NOTE**

This statistics sub-collection is read-only.

### Table 7.327. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>kind</td>
<td>StatisticKind</td>
<td>The type of statistic measures.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>type</td>
<td>ValueType</td>
<td>The data type for the statistical values that follow.</td>
</tr>
<tr>
<td>unit</td>
<td>StatisticUnit</td>
<td>The unit or rate to measure of the statistical values.</td>
</tr>
</tbody>
</table>
Table 7.328. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>brick</td>
<td>GlusterBrick</td>
<td></td>
</tr>
<tr>
<td>disk</td>
<td>Disk</td>
<td>A relationship to the containing disk resource.</td>
</tr>
<tr>
<td>gluster_volume</td>
<td>GlusterVolume</td>
<td></td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>host_nic</td>
<td>HostNic</td>
<td>A reference to the host NIC.</td>
</tr>
<tr>
<td>host_numa_node</td>
<td>NumaNode</td>
<td></td>
</tr>
<tr>
<td>nic</td>
<td>Nic</td>
<td></td>
</tr>
<tr>
<td>step</td>
<td>Step</td>
<td></td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td></td>
</tr>
</tbody>
</table>

7.252. STATISTICKIND ENUM

Table 7.329. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>counter</td>
<td></td>
</tr>
<tr>
<td>gauge</td>
<td></td>
</tr>
</tbody>
</table>

7.253. STATISTICUNIT ENUM

Table 7.330. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bits_per_second</td>
<td></td>
</tr>
</tbody>
</table>
### 7.254. STEP STRUCT

Represents a step, which is part of job execution. Step is used to describe and track a specific execution unit which is part of a wider sequence. Some steps support reporting their progress.

#### Table 7.331. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>end_time</td>
<td>Date</td>
<td>The end time of the step.</td>
</tr>
<tr>
<td>external</td>
<td>Boolean</td>
<td>Indicates if the step is originated by an external system.</td>
</tr>
<tr>
<td>external_type</td>
<td>ExternalSystemType</td>
<td>The external system which is referenced by the step.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>number</td>
<td>Integer</td>
<td>The order of the step in current hierarchy level.</td>
</tr>
<tr>
<td>progress</td>
<td>Integer</td>
<td>The step progress (if reported) in percentages.</td>
</tr>
<tr>
<td>start_time</td>
<td>Date</td>
<td>The start time of the step.</td>
</tr>
<tr>
<td>status</td>
<td>StepStatus</td>
<td>The status of the step.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>type</td>
<td>StepEnum</td>
<td>The type of the step.</td>
</tr>
</tbody>
</table>

### 7.254.1. external

Indicates if the step is originated by an external system. External steps are managed externally, by the creator of the step.

#### Table 7.332. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>execution_host</td>
<td>Host</td>
<td>The host used for the step execution (optional).</td>
</tr>
<tr>
<td>job</td>
<td>Job</td>
<td>References the job which is the top of the current step hierarchy.</td>
</tr>
<tr>
<td>parent_step</td>
<td>Step</td>
<td>References the parent step of the current step in the hierarchy.</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td></td>
</tr>
</tbody>
</table>

### 7.255. STEPENUM ENUM

Type representing a step type.

#### Table 7.333. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>executing</td>
<td>The executing step type.</td>
</tr>
<tr>
<td>finalizing</td>
<td>The finalizing step type.</td>
</tr>
<tr>
<td>rebalancing_volume</td>
<td>The rebalancing volume step type.</td>
</tr>
<tr>
<td>removing_bricks</td>
<td>The removing bricks step type.</td>
</tr>
<tr>
<td>unknown</td>
<td>The unknown step type.</td>
</tr>
<tr>
<td>validating</td>
<td>The validation step type.</td>
</tr>
</tbody>
</table>

### 7.255.1. executing
The executing step type. Used to track the main execution block of the job. Usually it will be a parent step of several sub-steps which describe portions of the execution step.

7.255.2. finalizing

The finalizing step type. Describes the post-execution steps requires to complete the job.

7.255.3. rebalancing_volume

The rebalancing volume step type. Describes a step type which is part of Gluster flow.

7.255.4. removing_bricks

The removing bricks step type. Describes a step type which is part of Gluster flow.

7.255.5. unknown

The unknown step type. Describes a step type which its origin is unknown.

7.255.6. validating

The validation step type. Used to verify the correctness of parameters and the validity of the parameters prior to the execution.

7.256. STEPSTATUS ENUM

Represents the status of the step.

Table 7.334. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>aborted</td>
<td>The aborted step status.</td>
</tr>
<tr>
<td>failed</td>
<td>The failed step status.</td>
</tr>
<tr>
<td>finished</td>
<td>The finished step status.</td>
</tr>
<tr>
<td>started</td>
<td>The started step status.</td>
</tr>
<tr>
<td>unknown</td>
<td>The unknown step status.</td>
</tr>
</tbody>
</table>

7.256.1. aborted

The aborted step status. This status is applicable for an external step that was forcibly aborted.

7.256.2. finished

The finished step status. This status describes a completed step execution.
The started step status. This status represents a step which is currently being executed.

The unknown step status. This status represents steps which their resolution is not known, i.e. steps that were executed before the system was unexpectedly restarted.

**7.257. STORAGECONNECTION STRUCT**

Represents a storage server connection.

Example XML representation:

```xml
<storage_connection id="123">
  <address>mynfs.example.com</address>
  <type>nfs</type>
  <path>/exports/mydata</path>
</storage_connection>
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>String</td>
<td>A storage server connection’s address.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>mount_options</td>
<td>String</td>
<td>The mount options of an NFS storage server connection.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>nfs_retrans</td>
<td>Integer</td>
<td>The NFS retrans value of an NFS storage server connection.</td>
</tr>
<tr>
<td>nfs_timeo</td>
<td>Integer</td>
<td>The NFS timeo value of an NFS storage server connection.</td>
</tr>
<tr>
<td>nfs_version</td>
<td>NfsVersion</td>
<td>The NFS version of an NFS storage server connection.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td>The password of an iSCSI storage server connection.</td>
</tr>
<tr>
<td>path</td>
<td>String</td>
<td>The path of an NFS storage server connection.</td>
</tr>
<tr>
<td>port</td>
<td>Integer</td>
<td>The port of an iSCSI storage server connection.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>portal</td>
<td>String</td>
<td>The portal of an iSCSI storage server connection.</td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td>The target of an iSCSI storage server connection.</td>
</tr>
<tr>
<td>type</td>
<td>StorageType</td>
<td>A storage server connection’s type.</td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td>The user name of an iSCSI storage server connection.</td>
</tr>
<tr>
<td>vfs_type</td>
<td>String</td>
<td>The VFS type of an NFS storage server connection.</td>
</tr>
</tbody>
</table>

**Table 7.336. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>gluster_volume</td>
<td>GlusterVolume</td>
<td>Link to the gluster volume, used by that storage domain.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
</tbody>
</table>

### 7.258. STORAGECONNECTIONEXTENSION STRUCT

**Table 7.337. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>target</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>username</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

**Table 7.338. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
</tbody>
</table>
7.259. STORAGEDOMAIN STRUCT

Storage domain.

An XML representation of a NFS storage domain with identifier 123:

```xml
<storage_domain href="/ovirt-engine/api/storagedomains/123" id="123">
  <name>mydata</name>
  <description>My data</description>
  <available>38654705664</available>
  <committed>1073741824</committed>
  <critical_space_action_blocker>5</critical_space_action_blocker>
  <external_status>ok</external_status>
  <master>true</master>
  <storage>
    <address>mynfs.example.com</address>
    <nfs_version>v3</nfs_version>
    <path>/exports/mydata</path>
    <type>nfs</type>
  </storage>
  <storage_format>v3</storage_format>
  <type>data</type>
  <used>13958643712</used>
  <warning_low_space_indicator>10</warning_low_space_indicator>
  <wipe_after_delete>false</wipe_after_delete>
  <data_centers>
    <data_center href="/ovirt-engine/api/datacenters/456" id="456"/>
  </data_centers>
</storage_domain>
```

Table 7.339. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>available</td>
<td>Integer</td>
<td>This attribute indicates whether a data storage domain is used as backup domain or not.</td>
</tr>
<tr>
<td>backup</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>committed</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>critical_space_action_blocker</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>discard_after_delete</td>
<td>Boolean</td>
<td>Indicates whether disks' blocks on block storage domains will be discarded right before they are deleted.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>external_status</td>
<td>ExternalStatus</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>import</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>master</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>status</td>
<td>StorageDomainStatus</td>
<td></td>
</tr>
<tr>
<td>storage</td>
<td>HostStorage</td>
<td></td>
</tr>
<tr>
<td>storage_format</td>
<td>StorageFormat</td>
<td></td>
</tr>
<tr>
<td>supports_discard</td>
<td>Boolean</td>
<td>Indicates whether a block storage domain supports discard operations.</td>
</tr>
<tr>
<td>supports_discard_zeroes_data</td>
<td>Boolean</td>
<td>Indicates whether a block storage domain supports the property that discard zeroes the data.</td>
</tr>
<tr>
<td>type</td>
<td>StorageDomainType</td>
<td></td>
</tr>
<tr>
<td>used</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>warning_low_space_indicator</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>wipe_after_delete</td>
<td>Boolean</td>
<td>Serves as the default value of <code>wipe_after_delete</code> for disks on this storage domain.</td>
</tr>
</tbody>
</table>

### 7.259.1. backup

This attribute indicates whether a data storage domain is used as backup domain or not. If the domain is set to backup then it will be used to store virtual machines and templates for disaster recovery purposes in the same way we use export storage domain. This attribute is only available with data storage domain and not with ISO domain or export storage domain. User can use this functionality while creating a data storage domain or importing a data storage domain.
7.259.2. discard_after_delete

Indicates whether disks' blocks on block storage domains will be discarded right before they are deleted. If true, and a disk on this storage domain has its \texttt{wipe_after_delete} value enabled, then when the disk is deleted:

1. It is first wiped.
2. Then its blocks are discarded.
3. Finally it is deleted.

Note that:

- Discard after delete will always be \texttt{false} for non block storage types.
- Discard after delete can be set to \texttt{true} only if the storage domain supports discard.

7.259.3. supports_discard

Indicates whether a block storage domain supports discard operations. A storage domain only supports discard if all of the logical units that it is built from support discard; that is, if each logical unit's \texttt{discard_max_size} value is greater than 0. This is one of the conditions necessary for a virtual disk in this storage domain to have its \texttt{pass_discard} attribute enabled.

7.259.4. supports_discard_zeroes_data

Indicates whether a block storage domain supports the property that discard zeroes the data. A storage domain only supports the property that discard zeroes the data if all of the logical units that it is built from support it; that is, if each logical unit's \texttt{discard_zeroes_data} value is true.

\textbf{IMPORTANT}

Since version 4.2.1 of the system, the support for this attribute has been removed as the sysfs file, \texttt{discard_zeroes_data}, was deprecated in the kernel. It is preserved for backwards compatibility, but the value will always be \texttt{false}.

7.259.5. wipe_after_delete

Serves as the default value of \texttt{wipe_after_delete} for disks on this storage domain. That is, newly created disks will get their \texttt{wipe_after_delete} value from their storage domains by default. Note that the configuration value \texttt{SANWipeAfterDelete} serves as the default value of block storage domains' \texttt{wipe_after_delete} value.

Table 7.340. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data_center</td>
<td>DataCenter</td>
<td>A link to the data center that the storage domain is attached to.</td>
</tr>
</tbody>
</table>
### 7.259.6. data_center

A link to the data center that the storage domain is attached to. This is preserved for backwards compatibility only, as the storage domain may be attached to multiple data centers (if it is an ISO domain). Use the `dataCenters` element instead.

### 7.260. STORAGEDOMAINLEASE STRUCT

Represents a lease residing on a storage domain.

A lease is a Sanlock resource residing on a special volume on the storage domain, this Sanlock resource is used to provide storage base locking.

#### Table 7.341. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to the storage domain on which the lock resides on.</td>
</tr>
</tbody>
</table>
7.261. STORAGEDOMAINSTATUS ENUM

Table 7.342. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>activating</td>
<td></td>
</tr>
<tr>
<td>active</td>
<td></td>
</tr>
<tr>
<td>detaching</td>
<td></td>
</tr>
<tr>
<td>inactive</td>
<td></td>
</tr>
<tr>
<td>locked</td>
<td></td>
</tr>
<tr>
<td>maintenance</td>
<td></td>
</tr>
<tr>
<td>mixed</td>
<td></td>
</tr>
<tr>
<td>preparing_for_mainte</td>
<td></td>
</tr>
<tr>
<td>maintenance</td>
<td></td>
</tr>
<tr>
<td>unattached</td>
<td></td>
</tr>
<tr>
<td>unknown</td>
<td></td>
</tr>
</tbody>
</table>

7.262. STORAGEDOMAINTYPE ENUM

Indicates the kind of data managed by a storage domain.

Table 7.343. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Data domains are used to store the disks and snapshots of the virtual machines and templates in the system.</td>
</tr>
<tr>
<td>export</td>
<td>Export domains are temporary storage repositories used to copy and move virtual machines and templates between data centers and Red Hat Virtualization environments.</td>
</tr>
<tr>
<td>image</td>
<td>Image domain store images that can be imported into from an external system.</td>
</tr>
<tr>
<td>iso</td>
<td>ISO domains store ISO files (or logical CDs) used to install and boot operating systems and applications for the virtual machines.</td>
</tr>
<tr>
<td>volume</td>
<td>Volume domains store logical volumes that can be used as disks for virtual machines.</td>
</tr>
</tbody>
</table>
7.262.1. data
Data domains are used to store the disks and snapshots of the virtual machines and templates in the system. In addition, snapshots of the disks are also stored in data domains. Data domains cannot be shared across data centers.

7.262.2. export
Export domains are temporary storage repositories used to copy and move virtual machines and templates between data centers and Red Hat Virtualization environments. Export domains can also be used to backup virtual machines. An export domain can be moved between data centers but it can only be active in one data center at a time.

7.262.3. image
Image domain store images that can be imported into from an external system. For example, images from an OpenStack Glance image repository.

7.262.4. iso
ISO domains store ISO files (or logical CDs) used to install and boot operating systems and applications for the virtual machines. ISO domains remove the data center’s need for physical media. An ISO domain can be shared across different data centers.

7.262.5. volume
Volume domains store logical volumes that can be used as disks for virtual machines. For example, volumes from an OpenStack Cinder block storage service.

7.263. STORAGEFORMAT ENUM
Type which represents a format of storage domain.

Table 7.263.1. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1</td>
<td>Version 1 of the storage domain format is applicable to NFS, iSCSI and FC storage domains.</td>
</tr>
<tr>
<td>v2</td>
<td>Version 2 of the storage domain format is applicable to iSCSI and FC storage domains.</td>
</tr>
<tr>
<td>v3</td>
<td>Version 3 of the storage domain format is applicable to NFS, POSIX, iSCSI and FC storage domains.</td>
</tr>
<tr>
<td>v4</td>
<td>Version 4 of the storage domain format.</td>
</tr>
</tbody>
</table>

7.263.1. v1
Version 1 of the storage domain format is applicable to NFS, iSCSI and FC storage domains.

Each storage domain contains metadata describing its own structure, and all of the names of physical
volumes that are used to back virtual machine disk images. Master domains additionally contain metadata for all the domains and physical volume names in the storage pool. The total size of this metadata is limited to 2 KiB, limiting the number of storage domains that can be in a pool. Template and virtual machine base images are read only.

**7.263.2. v2**

Version 2 of the storage domain format is applicable to iSCSI and FC storage domains.

All storage domain and pool metadata is stored as logical volume tags rather than written to a logical volume. Metadata about virtual machine disk volumes is still stored in a logical volume on the domains. Physical volume names are no longer included in the metadata. Template and virtual machine base images are read only.

**7.263.3. v3**

Version 3 of the storage domain format is applicable to NFS, POSIX, iSCSI and FC storage domains.

All storage domain and pool metadata is stored as logical volume tags rather than written to a logical volume. Metadata about virtual machine disk volumes is still stored in a logical volume on the domains. Virtual machine and template base images are no longer read only. This change enables live snapshots, live storage migration, and clone from snapshot. Support for Unicode metadata is added, for non-English volume names.

**7.264. STORAGETYPE ENUM**

Type representing a storage domain type.

**Table 7.345. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cinder</td>
<td>Cinder storage domain.</td>
</tr>
<tr>
<td>fcp</td>
<td>Fibre-Channel storage domain.</td>
</tr>
<tr>
<td>glance</td>
<td>Glance storage domain.</td>
</tr>
<tr>
<td>glusterfs</td>
<td>Gluster-FS storage domain.</td>
</tr>
<tr>
<td>iscsi</td>
<td>iSCSI storage domain.</td>
</tr>
<tr>
<td>localfs</td>
<td>Storage domain on Local storage.</td>
</tr>
<tr>
<td>nfs</td>
<td>NFS storage domain.</td>
</tr>
<tr>
<td>posixfs</td>
<td>POSIX-FS storage domain.</td>
</tr>
</tbody>
</table>

**7.264.1. cinder**

Cinder storage domain. For more details on Cinder please go to [Cinder](#).
7.264.2. glance
Glance storage domain. For more details on Glance please go to Glance.

7.264.3. glusterfs
Gluster-FS storage domain. For more details on Gluster please go to Gluster.

7.265. SWITCHTYPE ENUM
Describes all switch types supported by the Manager.

Table 7.346. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>The native switch type.</td>
</tr>
<tr>
<td>ovs</td>
<td>The Open vSwitch type.</td>
</tr>
</tbody>
</table>

7.266. SYSTEMOPTION STRUCT
Type representing a configuration option of the system.

Table 7.347. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>values</td>
<td>SystemOptionValue[]</td>
<td>Values of the option for various system versions.</td>
</tr>
</tbody>
</table>

7.267. SYSTEMOPTIONVALUE STRUCT
Type representing a pair of value and version of a configuration option.

Table 7.348. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
<td>Configuration option’s value for specific version.</td>
</tr>
</tbody>
</table>
### 7.268. TAG STRUCT

Represents a tag in the system.

#### Table 7.349. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

#### Table 7.350. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>group</td>
<td>Group</td>
<td>Reference to the group which has this tag assigned.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host which has this tag assigned.</td>
</tr>
<tr>
<td>parent</td>
<td>Tag</td>
<td>Reference to the parent tag of this tag.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Reference to the template which has this tag assigned.</td>
</tr>
<tr>
<td>user</td>
<td>User</td>
<td>Reference to the user who has this tag assigned.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Reference to the virtual machine which has this tag assigned.</td>
</tr>
</tbody>
</table>

### 7.269. TEMPLATE STRUCT

Type representing a virtual machine template. This allows a rapid instanstiation of virtual machines with common configuration and disk states.

#### Table 7.351. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios</td>
<td>Bios</td>
<td>Reference to virtual machine’s BIOS configuration.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console</td>
<td>Console</td>
<td>Console configured for this virtual machine.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The configuration of the virtual machine CPU.</td>
</tr>
<tr>
<td>cpu_shares</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>creation_time</td>
<td>Date</td>
<td>The virtual machine creation date.</td>
</tr>
<tr>
<td>custom_compatibility_version</td>
<td>Version</td>
<td>Virtual machine custom compatibility version.</td>
</tr>
<tr>
<td>custom_cpu_model</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_emulated_machine</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Properties sent to VDSM to configure various hooks.</td>
</tr>
<tr>
<td>delete_protected</td>
<td>Boolean</td>
<td>If true, the virtual machine cannot be deleted.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The virtual machine display configuration.</td>
</tr>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Domain configured for this virtual machine.</td>
</tr>
<tr>
<td>high_availability</td>
<td>HighAvailability</td>
<td>The virtual machine high availability configuration.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialization</td>
<td>Reference to the virtual machine’s initialization configuration.</td>
</tr>
<tr>
<td>io</td>
<td>Io</td>
<td>For performance tuning of IO threading.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Virtual machine’s large icon.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>Reference to the storage domain this virtual machine/template lease reside on.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The virtual machine’s memory, in bytes.</td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td>Reference to virtual machine’s memory management configuration.</td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td>Reference to configuration of migration of running virtual machine to another host.</td>
</tr>
<tr>
<td>migration_downtime</td>
<td>Integer</td>
<td>Maximum time the virtual machine can be non responsive during its live migration to another host in ms.</td>
</tr>
<tr>
<td>multi_queues_enabled</td>
<td>Boolean</td>
<td>If true, each virtual interface will get the optimal number of queues, depending on the available virtual Cpus.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>The origin of this virtual machine.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>Operating system type installed on the virtual machine.</td>
</tr>
<tr>
<td>placement_policy</td>
<td>VmPlacementPolicy</td>
<td>The configuration of the virtual machine’s placement policy.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>Random Number Generator device configuration for this virtual machine.</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td>Virtual machine’s serial number in a cluster.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Virtual machine’s small icon.</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>If true, the sound card is added to the virtual machine.</td>
</tr>
<tr>
<td>sso</td>
<td>Sso</td>
<td>Reference to the Single Sign On configuration this virtual machine is configured for.</td>
</tr>
<tr>
<td>start_paused</td>
<td>Boolean</td>
<td>If true, the virtual machine will be initially in ‘paused’ state after start.</td>
</tr>
<tr>
<td>stateless</td>
<td>Boolean</td>
<td>If true, the virtual machine is stateless - it’s state (disks) are rolled-back after shutdown.</td>
</tr>
</tbody>
</table>
### Name | Type | Summary
---|---|---
**status** | TemplateStatus | The status of the template.

**storage_error_resume_behaviour** | VmStorageErrorResumeBehaviour | Determines how the virtual machine will be resumed after storage error.

**time_zone** | TimeZone | The virtual machine’s time zone set by oVirt.

**tunnel_migration** | Boolean | If true, the network data transfer will be encrypted during virtual machine live migration.

**type** | VmType | Determines whether the virtual machine is optimized for desktop or server.

**usb** | Usb | Configuration of USB devices for this virtual machine (count, type).

**version** | TemplateVersion | Indicates whether this is a base version or a sub version of another template.

**virtio_scsi** | VirtioScsi | Reference to VirtIO SCSI configuration.

**vm** | Vm | The virtual machine configuration associated with this template.

---

### 7.269.1. cpu

The configuration of the virtual machine CPU.

The socket configuration can be updated without rebooting the virtual machine. The cores and the threads require a reboot.

For example, to change the number of sockets to 4 immediately, and the number of cores and threads to 2 after reboot, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With a request body:

```xml
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```
7.269.2. custom_compatibility_version

Virtual machine custom compatibility version.

Enables a virtual machine to be customized to its own compatibility version. If custom_compatibility_version is set, it overrides the cluster’s compatibility version for this particular virtual machine.

The compatibility version of a virtual machine is limited by the data center the virtual machine resides in, and is checked against capabilities of the host the virtual machine is planned to run on.

7.269.3. high_availability

The virtual machine high availability configuration. If set, the virtual machine will be automatically restarted when it unexpectedly goes down.

7.269.4. initialization

Reference to the virtual machine’s initialization configuration.

NOTE

Since Red Hat Virtualization 4.1.8 this property can be cleared by sending an empty tag.

For example, to clear the initialization attribute send a request like this:

PUT /ovirt-engine/api/vms/123

With a request body like this:

<vm>
  <initialization/>
</vm>

The response to such a request, and requests with the header All-Content: true will still contain this attribute.

7.269.5. large_icon

Virtual machine’s large icon. Either set by user or refers to image set according to operating system.

7.269.6. lease

Reference to the storage domain this virtual machine/template lease reside on.

A virtual machine running with a lease requires checking while running that the lease is not taken by another host, preventing another instance of this virtual machine from running on another host. This provides protection against split-brain in highly available virtual machines. A template can also have a storage domain defined for a lease in order to have the virtual machines created from this template to be preconfigured with this storage domain as the location of the leases.

7.269.7. memory
The virtual machine’s memory, in bytes.

For example, to update a virtual machine to contain 1 Gibibyte (GiB) of memory, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With the following request body:

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Memory hot plug is supported from Red Hat Virtualization 3.6 onwards. You can use the example above to increase memory while the virtual machine is in state `up`. The size increment must be dividable by the value of the `HotPlugMemoryBlockSizeMb` configuration value (256 MiB by default). If the memory size increment is not dividable by this value, the memory size change is only stored to next run configuration. Each successful memory hot plug operation creates one or two new memory devices.

Memory hot unplug is supported since Red Hat Virtualization 4.2 onwards. Memory hot unplug can only be performed when the virtual machine is in state `up`. Only previously hot plugged memory devices can be removed by the hot unplug operation. The requested memory decrement is rounded down to match sizes of a combination of previously hot plugged memory devices. The requested memory value is stored to next run configuration without rounding.

**NOTE**

Memory in the example is converted to bytes using the following formula:

\[
1 \text{ GiB} = 2^{30} \text{ bytes} = 1073741824 \text{ bytes}.
\]

**NOTE**

Red Hat Virtualization Manager internally rounds values down to whole MiBs (1MiB = \(2^{20}\) bytes)

**7.269.8. migration_downtime**

Maximum time the virtual machine can be non responsive during its live migration to another host in ms.

Set either explicitly for the virtual machine or by `engine-config -s DefaultMaximumMigrationDowntime=[value]`

**7.269.9. origin**

The origin of this virtual machine.

Possible values:

- `ovirt`
- `rhev`
- `vmware`
* xen
* external
* hosted_engine
* managed_hosted_engine
* kvm
* physical_machine
* hyperv

**7.269.10. placement_policy**

The configuration of the virtual machine’s placement policy.

This configuration can be updated to pin a virtual machine to one or more hosts.

**NOTE**

Virtual machines that are pinned to multiple hosts cannot be live migrated, but in the event of a host failure, any virtual machine configured to be highly available is automatically restarted on one of the other hosts to which the virtual machine is pinned.

For example, to pin a virtual machine to two hosts, send the following request:

```text
PUT /api/vms/123
```

With a request body like this:

```xml
<vm>
  <high_availability>
    <enabled>true</enabled>
    <priority>1</priority>
  </high_availability>
  <placement_policy>
    <hosts>
      <host>
        <name>Host1</name>
      </host>
      <host>
        <name>Host2</name>
      </host>
    </hosts>
    <affinity>pinned</affinity>
  </placement_policy>
</vm>
```

**7.269.11. small_icon**

Virtual machine’s small icon. Either set by user or refers to image set according to operating system.
7.269.12. sso
Reference to the Single Sign On configuration this virtual machine is configured for. The user can be automatically signed in the virtual machine’s operating system when console is opened.

Table 7.352. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>References to the CD-ROM devices attached to the template.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster the virtual machine belongs to.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Reference to CPU profile used by this virtual machine.</td>
</tr>
<tr>
<td>disk_attachments</td>
<td>DiskAttachment[]</td>
<td>References to the disks attached to the template.</td>
</tr>
<tr>
<td>graphics_consoles</td>
<td>GraphicsConsole[]</td>
<td>References to the graphic consoles attached to the template.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>References to the network interfaces attached to the template.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>References to the user permissions attached to the template.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Reference to quota configuration set for this virtual machine.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to storage domain the virtual machine belongs to.</td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>References to the tags attached to the template.</td>
</tr>
<tr>
<td>watchdogs</td>
<td>Watchdog[]</td>
<td>References to the watchdog devices attached to the template.</td>
</tr>
</tbody>
</table>

7.270. TEMPLATESTATUS ENUM
Type representing a status of a virtual machine template.

Table 7.353. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>illegal</td>
<td>This status indicates that at least one of the disks of the template is illegal.</td>
</tr>
<tr>
<td>locked</td>
<td>This status indicates that some operation that prevents other operations with the template is being executed.</td>
</tr>
<tr>
<td>ok</td>
<td>This status indicates that the template is valid and ready for use.</td>
</tr>
</tbody>
</table>
7.271. TEMPLATEVERSION STRUCT
Type representing a version of a virtual machine template.

Table 7.354. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>version_name</td>
<td>String</td>
<td>The name of this version.</td>
</tr>
<tr>
<td>version_number</td>
<td>Integer</td>
<td>The index of this version in the versions hierarchy of the template.</td>
</tr>
</tbody>
</table>

7.271.1. version_number
The index of this version in the versions hierarchy of the template. The index 1 represents the original version of a template that is also called base version.

Table 7.355. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>base_template</td>
<td>Template</td>
<td>References the template that this version is associated with.</td>
</tr>
</tbody>
</table>

7.272. TICKET STRUCT
Type representing a ticket that allows virtual machine access.

Table 7.356. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>expiry</td>
<td>Integer</td>
<td>Time to live for the ticket in seconds.</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>The virtual machine access ticket.</td>
</tr>
</tbody>
</table>

7.273. TIMEZONE STRUCT
Time zone representation.

Table 7.357. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the time zone.</td>
</tr>
<tr>
<td>utc_offset</td>
<td>String</td>
<td>Offset from <a href="https://en">https://en</a>.</td>
</tr>
</tbody>
</table>
7.273.1. utc_offset
Offset from UTC.

7.274. TRANSPARENTHUGEPPAGES STRUCT
Type representing a transparent huge pages (THP) support.

Table 7.358. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Enable THP support.</td>
</tr>
</tbody>
</table>

7.275. TRANSPORTTYPE ENUM
Protocol used to access a Gluster volume.

Table 7.359. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdma</td>
<td>Remote direct memory access.</td>
</tr>
<tr>
<td>tcp</td>
<td>TCP.</td>
</tr>
</tbody>
</table>

7.276. UNMANAGEDNETWORK STRUCT

Table 7.360. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

Table 7.361. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>host_nic</td>
<td>HostNic</td>
<td></td>
</tr>
</tbody>
</table>
7.277. USB STRUCT

Configuration of the USB device of a virtual machine.

### Table 7.362. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Determines whether the USB device should be included or not.</td>
</tr>
<tr>
<td>type</td>
<td>UsbType</td>
<td>USB type, currently only <strong>native</strong> is supported.</td>
</tr>
</tbody>
</table>

7.278. USBTYPE ENUM

Type of USB device redirection.

### Table 7.363. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>legacy</td>
<td>Legacy USB redirection.</td>
</tr>
<tr>
<td>native</td>
<td>Native USB redirection.</td>
</tr>
</tbody>
</table>

7.278.1. legacy

Legacy USB redirection.

This USB type has been deprecated since version 3.6 of the engine, and has been completely removed in version 4.1. It is preserved only to avoid syntax errors in existing scripts. If it is used it will be automatically replaced by **native**.

7.278.2. native

Native USB redirection.

Native USB redirection allows KVM/SPICE USB redirection for Linux and Windows virtual machines. Virtual (guest) machines require no guest-installed agents or drivers for native USB. On Linux clients, all packages required for USB redirection are provided by the **virt-viewer** package. On Windows clients, you must also install the **usbdk** package.

7.279. USER STRUCT

Represents a user in the system.

### Table 7.364. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
<td>----------------------------------------------------------------</td>
</tr>
<tr>
<td>department</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>domain_entry_id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>email</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>last_name</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>logged_in</td>
<td>Boolean</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>namespace</td>
<td>String</td>
<td>Namespace where the user resides.</td>
</tr>
<tr>
<td>password</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>principal</td>
<td>String</td>
<td>Similar to user_name.</td>
</tr>
<tr>
<td>user_name</td>
<td>String</td>
<td>The user’s username.</td>
</tr>
</tbody>
</table>

### 7.279.1. namespace

Namespace where the user resides. When using the authorization provider that stores users in the LDAP server, this attribute equals the naming context of the LDAP server. See [https://github.com/oVirt/ovirt-engine-extension-aaa-ldap](https://github.com/oVirt/ovirt-engine-extension-aaa-ldap) for more information. When using the built-in authorization provider that stores users in the database this attribute is ignored. See [https://github.com/oVirt/ovirt-engine-extension-aaa-jdbc](https://github.com/oVirt/ovirt-engine-extension-aaa-jdbc) for more information.

### 7.279.2. principal

Similar to user_name. The format depends on the LDAP provider. With most LDAP providers it is the value of the `uid` LDAP attribute. In the case of Active Directory it is the User Principal Name (UPN).

### 7.279.3. user_name

The user’s username. The format depends on authorization provider type. In most LDAP providers it is the value of the `uid` LDAP attribute. In Active Directory it is the User Principal Name (UPN). UPN or `uid` must be followed by the authorization provider name. For example, in the case of LDAP’s `uid` attribute it is: `myuser@myextension-authz`. In the case of Active Directory using UPN it is: `myuser@mysubdomain.mydomain.com@myextension-authz`. This attribute is a required parameter when adding a new user.
Table 7.365. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>Domain</td>
<td></td>
</tr>
<tr>
<td>groups</td>
<td>Group[]</td>
<td></td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td></td>
</tr>
<tr>
<td>roles</td>
<td>Role[]</td>
<td>A link to the roles sub-collection for user resources.</td>
</tr>
<tr>
<td>ssh_public_keys</td>
<td>SshPublicKey[]</td>
<td></td>
</tr>
<tr>
<td>tags</td>
<td>Tag[]</td>
<td>A link to the tags sub-collection for user resources.</td>
</tr>
</tbody>
</table>

7.280. VALUE STRUCT

Table 7.366. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>datum</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>detail</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

7.281. VALUETYPE ENUM

Table 7.367. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>decimal</td>
<td></td>
</tr>
<tr>
<td>integer</td>
<td></td>
</tr>
<tr>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>

7.282. VCPUPIN STRUCT

Table 7.368. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu_set</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vcpu</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

### 7.283. VENDOR STRUCT

**Table 7.369. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

### 7.284. VERSION STRUCT

**Table 7.370. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>build</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>full_version</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>major</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>minor</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>revision</td>
<td>Integer</td>
<td></td>
</tr>
</tbody>
</table>

### 7.285. VIRTIOSCSI STRUCT

Type representing the support of virtio-SCSI. If it supported we use virtio driver for SCSI guest device.
Table 7.371. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabled</td>
<td>Boolean</td>
<td>Enable Virtio SCSI support.</td>
</tr>
</tbody>
</table>

### 7.286. VIRTUALNUMANODE STRUCT

Represents the virtual NUMA node.

An example XML representation:

```xml
<vm_numa_node href="/ovirt-engine/api/vms/123/numanodes/456" id="456">
  <cpu>
    <cores>
      <core>
        <index>0</index>
      </core>
    </cores>
  </cpu>
  <index>0</index>
  <memory>1024</memory>
  <numa_node_pins>
    <numa_node_pin>
      <index>0</index>
    </numa_node_pin>
  </numa_node_pins>
  <vm href="/ovirt-engine/api/vms/123" id="123" />
</vm_numa_node>
```

Table 7.372. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>index</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>Memory of the NUMA node in MB.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>node_distance</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>
Table 7.373. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>host</td>
<td>Host</td>
<td>Each host NUMA node resource exposes a statistics sub-collection for host NUMA node specific statistics.</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Each host NUMA node resource exposes a statistics sub-collection for host NUMA node specific statistics.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td></td>
</tr>
</tbody>
</table>

7.286.1. statistics

Each host NUMA node resource exposes a statistics sub-collection for host NUMA node specific statistics.

An example of an XML representation:

```xml
<statistics>
  <statistic href="/ovirt-engine/api/hosts/123/numanodes/456/statistics/789" id="789">
    <name>memory.total</name>
    <description>Total memory</description>
    <kind>gauge</kind>
    <type>integer</type>
    <unit>bytes</unit>
    <values>
      <value>
        <datum>25165824000</datum>
      </value>
    </values>
  </statistic>
  ...
</statistics>
```

**NOTE**

This statistics sub-collection is read-only.

The following list shows the statistic types for a host NUMA node:
### Name Description

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory.total</td>
<td>Total memory in bytes on the NUMA node.</td>
</tr>
<tr>
<td>memory.used</td>
<td>Memory in bytes used on the NUMA node.</td>
</tr>
<tr>
<td>memory.free</td>
<td>Memory in bytes free on the NUMA node.</td>
</tr>
<tr>
<td>cpu.current.user</td>
<td>Percentage of CPU usage for user slice.</td>
</tr>
<tr>
<td>cpu.current.system</td>
<td>Percentage of CPU usage for system.</td>
</tr>
<tr>
<td>cpu.current.idle</td>
<td>Percentage of idle CPU usage.</td>
</tr>
</tbody>
</table>

#### 7.287. VLAN STRUCT

Type representing a Virtual LAN (VLAN) type.

**Table 7.374. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>Integer</td>
<td>Virtual LAN ID.</td>
</tr>
</tbody>
</table>

#### 7.288. VM STRUCT

Represents a virtual machine.

**Table 7.375. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios</td>
<td>Bios</td>
<td>Reference to virtual machine’s BIOS configuration.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console</td>
<td>Console</td>
<td>Console configured for this virtual machine.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The configuration of the virtual machine CPU.</td>
</tr>
<tr>
<td>cpu_shares</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>creation_time</td>
<td>Date</td>
<td>The virtual machine creation date.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>custom_compatibility_version</td>
<td>Version</td>
<td>Virtual machine custom compatibility version.</td>
</tr>
<tr>
<td>custom_cpu_model</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_emulated_machine</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Properties sent to VDSM to configure various hooks.</td>
</tr>
<tr>
<td>delete_protected</td>
<td>Boolean</td>
<td>If true, the virtual machine cannot be deleted.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The virtual machine display configuration.</td>
</tr>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Domain configured for this virtual machine.</td>
</tr>
<tr>
<td>fqdn</td>
<td>String</td>
<td>Fully qualified domain name of the virtual machine.</td>
</tr>
<tr>
<td>guest_operating_system</td>
<td>GuestOperatingSystem</td>
<td>What operating system is installed on the virtual machine.</td>
</tr>
<tr>
<td>guest_timezone</td>
<td>TimeZone</td>
<td>What time zone is used by the virtual machine (as returned by guest agent).</td>
</tr>
<tr>
<td>has_illegal_images</td>
<td>Boolean</td>
<td>Indicates whether the virtual machine has snapshots with disks in ILLEGAL state.</td>
</tr>
<tr>
<td>high_availability</td>
<td>HighAvailability</td>
<td>The virtual machine high availability configuration.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialization</td>
<td>Reference to the virtual machine’s initialization configuration.</td>
</tr>
<tr>
<td>io</td>
<td>Io</td>
<td>For performance tuning of IO threading.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Virtual machine’s large icon.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>Reference to the storage domain this virtual machine/template lease reside on.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The virtual machine’s memory, in bytes.</td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td>Reference to virtual machine’s memory management configuration.</td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td>Reference to configuration of migration of running virtual machine to another host.</td>
</tr>
<tr>
<td>migration_downtime</td>
<td>Integer</td>
<td>Maximum time the virtual machine can be non responsive during its live migration to another host in ms.</td>
</tr>
<tr>
<td>multi_queues_enabled</td>
<td>Boolean</td>
<td>If true, each virtual interface will get the optimal number of queues, depending on the available virtual Cpus.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>next_run_configuration_exists</td>
<td>Boolean</td>
<td>Virtual machine configuration has been changed and requires restart of the virtual machine.</td>
</tr>
<tr>
<td>numa_tune_mode</td>
<td>NumaTuneMode</td>
<td>How the NUMA topology is applied.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>The origin of this virtual machine.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>Operating system type installed on the virtual machine.</td>
</tr>
<tr>
<td>payloads</td>
<td>Payload[]</td>
<td>Optional payloads of the virtual machine, used for ISOS to configure it.</td>
</tr>
<tr>
<td>placement_policy</td>
<td>VmPlacementPolicy</td>
<td>The configuration of the virtual machine's placement policy.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>Random Number Generator device configuration for this virtual machine.</td>
</tr>
<tr>
<td>run_once</td>
<td>Boolean</td>
<td>If true, the virtual machine has been started using the run once command, meaning it's configuration might differ from the stored one for the purpose of this single run.</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td>Virtual machine's serial number in a cluster.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Virtual machine’s small icon.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>If <code>true</code>, the sound card is added to the virtual machine.</td>
</tr>
<tr>
<td>sso</td>
<td>Sso</td>
<td>Reference to the Single Sign On configuration this virtual machine is configured for.</td>
</tr>
<tr>
<td>start_paused</td>
<td>Boolean</td>
<td>If <code>true</code>, the virtual machine will be initially in 'paused' state after start.</td>
</tr>
<tr>
<td>start_time</td>
<td>Date</td>
<td>The date in which the virtual machine was started.</td>
</tr>
<tr>
<td>stateless</td>
<td>Boolean</td>
<td>If <code>true</code>, the virtual machine is stateless - it’s state (disks) are rolled-back after shutdown.</td>
</tr>
<tr>
<td>status</td>
<td>VmStatus</td>
<td>The current status of the virtual machine.</td>
</tr>
<tr>
<td>status_detail</td>
<td>String</td>
<td>Human readable detail of current status.</td>
</tr>
<tr>
<td>stop_reason</td>
<td>String</td>
<td>The reason the virtual machine was stopped.</td>
</tr>
<tr>
<td>stop_time</td>
<td>Date</td>
<td>The date in which the virtual machine was stopped.</td>
</tr>
<tr>
<td>storage_error_resume_behaviour</td>
<td>VmStorageErrorResumeBehaviour</td>
<td>Determines how the virtual machine will be resumed after storage error.</td>
</tr>
<tr>
<td>time_zone</td>
<td>TimeZone</td>
<td>The virtual machine’s time zone set by oVirt.</td>
</tr>
<tr>
<td>tunnel_migration</td>
<td>Boolean</td>
<td>If <code>true</code>, the network data transfer will be encrypted during virtual machine live migration.</td>
</tr>
<tr>
<td>type</td>
<td>VmType</td>
<td>Determines whether the virtual machine is optimized for desktop or server.</td>
</tr>
<tr>
<td>usb</td>
<td>Usb</td>
<td>Configuration of USB devices for this virtual machine (count, type).</td>
</tr>
<tr>
<td>use_latest_template_version</td>
<td>Boolean</td>
<td>If <code>true</code>, the virtual machine is reconfigured to the latest version of it’s template when it is started.</td>
</tr>
<tr>
<td>virtio_scsi</td>
<td>VirtioScsi</td>
<td>Reference to VirtIO SCSI configuration.</td>
</tr>
</tbody>
</table>

7.288.1. cpu
The configuration of the virtual machine CPU.
The socket configuration can be updated without rebooting the virtual machine. The cores and the 
threads require a reboot.

For example, to change the number of sockets to 4 immediately, and the number of cores and threads to 
2 after reboot, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With a request body:

```
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

### 7.288.2. custom_compatibility_version

Virtual machine custom compatibility version.

Enables a virtual machine to be customized to its own compatibility version. If `custom_compatibility_version` is set, it overrides the cluster’s compatibility version for this particular virtual machine.

The compatibility version of a virtual machine is limited by the data center the virtual machine resides in, and is checked against capabilities of the host the virtual machine is planned to run on.

### 7.288.3. high_availability

The virtual machine high availability configuration. If set, the virtual machine will be automatically restarted when it unexpectedly goes down.

### 7.288.4. initialization

Reference to the virtual machine’s initialization configuration.

**NOTE**

Since Red Hat Virtualization 4.1.8 this property can be cleared by sending an empty tag.

For example, to clear the `initialization` attribute send a request like this:

```
PUT /ovirt-engine/api/vms/123
```

With a request body like this:

```
<vm>
  <initialization/>
</vm>
```
The response to such a request, and requests with the header \texttt{All-Content: true} will still contain this attribute.

### 7.288.5. large_icon

Virtual machine's large icon. Either set by user or refers to image set according to operating system.

### 7.288.6. lease

Reference to the storage domain this virtual machine/template lease reside on.

A virtual machine running with a lease requires checking while running that the lease is not taken by another host, preventing another instance of this virtual machine from running on another host. This provides protection against split-brain in highly available virtual machines. A template can also have a storage domain defined for a lease in order to have the virtual machines created from this template to be preconfigured with this storage domain as the location of the leases.

### 7.288.7. memory

The virtual machine’s memory, in bytes.

For example, to update a virtual machine to contain 1 Gibibyte (GiB) of memory, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With the following request body:

```
<vm>
  <memory>1073741824</memory>
</vm>
```

Memory hot plug is supported from Red Hat Virtualization 3.6 onwards. You can use the example above to increase memory while the virtual machine is in state \texttt{up}. The size increment must be dividable by the value of the \texttt{HotPlugMemoryBlockSizeMb} configuration value (256 MiB by default). If the memory size increment is not dividable by this value, the memory size change is only stored to next run configuration. Each successful memory hot plug operation creates one or two new memory devices.

Memory hot unplug is supported since Red Hat Virtualization 4.2 onwards. Memory hot unplug can only be performed when the virtual machine is in state \texttt{up}. Only previously hot plugged memory devices can be removed by the hot unplug operation. The requested memory decrement is rounded down to match sizes of a combination of previously hot plugged memory devices. The requested memory value is stored to next run configuration without rounding.

**NOTE**

Memory in the example is converted to bytes using the following formula:

\[ 1 \text{ GiB} = 2^{30} \text{ bytes} = 1073741824 \text{ bytes}. \]
NOTE

Red Hat Virtualization Manager internally rounds values down to whole MiBs (1MiB = 2^{20} bytes)

7.288.8. migration_downtime

Maximum time the virtual machine can be non responsive during its live migration to another host in ms.

Set either explicitly for the virtual machine or by `engine-config -s
Default MaximumMigrationDowntime=[value]`

7.288.9. next_run_configuration_exists

Virtual machine configuration has been changed and requires restart of the virtual machine. Changed configuration is applied at processing the virtual machine’s shut down.

7.288.10. origin

The origin of this virtual machine.

Possible values:

- ovirt
- rhev
- vmware
- xen
- external
- hosted_engine
- managed_hosted_engine
- kvm
- physical_machine
- hyperv

7.288.11. placement_policy

The configuration of the virtual machine’s placement policy.

This configuration can be updated to pin a virtual machine to one or more hosts.

NOTE

Virtual machines that are pinned to multiple hosts cannot be live migrated, but in the event of a host failure, any virtual machine configured to be highly available is automatically restarted on one of the other hosts to which the virtual machine is pinned.
For example, to pin a virtual machine to two hosts, send the following request:

```
PUT /api/vms/123
```

With a request body like this:

```
<vm>
  <high_availability>
    <enabled>true</enabled>
    <priority>1</priority>
  </high_availability>
  <placement_policy>
    <hosts>
      <host>
        <name>Host1</name>
      </host>
      <host>
        <name>Host2</name>
      </host>
    </hosts>
    <affinity>pinned</affinity>
  </placement_policy>
</vm>
```

### 7.288.12. small_icon

Virtual machine’s small icon. Either set by user or refers to image set according to operating system.

### 7.288.13. sso

Reference to the Single Sign On configuration this virtual machine is configured for. The user can be automatically signed in the virtual machine’s operating system when console is opened.

### 7.288.14. stop_reason

The reason the virtual machine was stopped. Optionally set by user when shutting down the virtual machine.

### Table 7.376. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity_labels</td>
<td>AffinityLabel[]</td>
<td>Optional.</td>
</tr>
<tr>
<td>applications</td>
<td>Application[]</td>
<td>List of applications installed on the virtual machine.</td>
</tr>
<tr>
<td>cdroms</td>
<td>Cdrom[]</td>
<td>Reference to the ISO mounted to the CDROM.</td>
</tr>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster the virtual machine belongs to.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Reference to CPU profile used by this virtual machine.</td>
</tr>
</tbody>
</table>
# Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disk_attachments</td>
<td>DiskAttachment[]</td>
<td>References the disks attached to the virtual machine.</td>
</tr>
<tr>
<td>external_host_provider</td>
<td>ExternalHostProvider</td>
<td></td>
</tr>
<tr>
<td>floppies</td>
<td>Floppy[]</td>
<td>Reference to the ISO mounted to the floppy.</td>
</tr>
<tr>
<td>graphics_consoles</td>
<td>GraphicsConsole[]</td>
<td>List of graphics consoles configured for this virtual machine.</td>
</tr>
<tr>
<td>host</td>
<td>Host</td>
<td>Reference to the host the virtual machine is running on.</td>
</tr>
<tr>
<td>host_devices</td>
<td>HostDevice[]</td>
<td>References devices associated to this virtual machine.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>The virtual machine configuration can be optionally predefined via one of the instance types.</td>
</tr>
<tr>
<td>katello_errata</td>
<td>KatelloErratum[]</td>
<td>Lists all the Katello errata assigned to the virtual machine.</td>
</tr>
<tr>
<td>nics</td>
<td>Nic[]</td>
<td>References the list of network interface devices on the virtual machine.</td>
</tr>
<tr>
<td>numa_nodes</td>
<td>NumaNode[]</td>
<td>Refers to the NUMA Nodes configuration used by this virtual machine.</td>
</tr>
<tr>
<td>original_template</td>
<td>Template</td>
<td>References the original template used to create the virtual machine.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Permissions set for this virtual machine.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Reference to quota configuration set for this virtual machine.</td>
</tr>
<tr>
<td>reported_devices</td>
<td>ReportedDevice[]</td>
<td></td>
</tr>
<tr>
<td>sessions</td>
<td>Session[]</td>
<td>List of user sessions opened for this virtual machine.</td>
</tr>
<tr>
<td>snapshots</td>
<td>Snapshot[]</td>
<td>Refers to all snapshots taken from the virtual machine.</td>
</tr>
<tr>
<td>statistics</td>
<td>Statistic[]</td>
<td>Statistics data collected from this virtual machine.</td>
</tr>
</tbody>
</table>
### 7.288.15. affinity_labels

Optional. Used for labeling of sub-clusters.

### 7.288.16. katello_errata

Lists all the Katello errata assigned to the virtual machine.

```
GET /ovirt-engine/api/vms/123/katelloerrata
```

You will receive response in XML like this one:

```
<katello_errata>
  <katello_erratum href="/ovirt-engine/api/katelloerrata/456" id="456">
    <name>RHBA-2013:XYZ</name>
    <description>The description of the erratum</description>
    <title>some bug fix update</title>
    <type>bugfix</type>
    <issued>2013-11-20T02:00:00.000+02:00</issued>
    <solution>few guidelines regarding the solution</solution>
    <summary>updated packages that fix one bug are now available for XYZ</summary>
    <packages>
      <package>
        <name>libipa_hbac-1.9.2-82.11.el6_4.i686</name>
      </package>
      ...
    </packages>
  </katello_erratum>
  ...
</katello_errata>
```

### 7.288.17. original_template

References the original template used to create the virtual machine.
If the virtual machine is cloned from a template or another virtual machine, the template links to the Blank template, and the original_template is used to track history.

Otherwise the template and original_template are the same.

### 7.288.18. statistics

Statistics data collected from this virtual machine.

Note that some statistics, notably memory.buffered and memory.cached are available only when oVirt guest agent is installed in the virtual machine.

### 7.289. VMAFFINITY ENUM

Table 7.377. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>migratable</td>
<td></td>
</tr>
<tr>
<td>pinned</td>
<td></td>
</tr>
<tr>
<td>user_migratable</td>
<td></td>
</tr>
</tbody>
</table>

### 7.290. VMBASE STRUCT

Represents basic virtual machine configuration. This is used by virtual machines, templates and instance types.

Table 7.378. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>bios</td>
<td>Bios</td>
<td>Reference to virtual machine’s BIOS configuration.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>console</td>
<td>Console</td>
<td>Console configured for this virtual machine.</td>
</tr>
<tr>
<td>cpu</td>
<td>Cpu</td>
<td>The configuration of the virtual machine CPU.</td>
</tr>
<tr>
<td>cpu_shares</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>creation_time</td>
<td>Date</td>
<td>The virtual machine creation date.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>custom_compatibility_version</td>
<td>Version</td>
<td>Virtual machine custom compatibility version.</td>
</tr>
<tr>
<td>custom_cpu_model</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_emulated_machine</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Properties sent to VDSM to configure various hooks.</td>
</tr>
<tr>
<td>delete_protected</td>
<td>Boolean</td>
<td>If true, the virtual machine cannot be deleted.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The virtual machine display configuration.</td>
</tr>
<tr>
<td>domain</td>
<td>Domain</td>
<td>Domain configured for this virtual machine.</td>
</tr>
<tr>
<td>high_availability</td>
<td>HighAvailability</td>
<td>The virtual machine high availability configuration.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialization</td>
<td>Reference to the virtual machine’s initialization configuration.</td>
</tr>
<tr>
<td>io</td>
<td>Io</td>
<td>For performance tuning of IO threading.</td>
</tr>
<tr>
<td>large_icon</td>
<td>Icon</td>
<td>Virtual machine’s large icon.</td>
</tr>
<tr>
<td>lease</td>
<td>StorageDomainLease</td>
<td>Reference to the storage domain this virtual machine/template lease reside on.</td>
</tr>
<tr>
<td>memory</td>
<td>Integer</td>
<td>The virtual machine’s memory, in bytes.</td>
</tr>
<tr>
<td>memory_policy</td>
<td>MemoryPolicy</td>
<td>Reference to virtual machine’s memory management configuration.</td>
</tr>
<tr>
<td>migration</td>
<td>MigrationOptions</td>
<td>Reference to configuration of migration of running virtual machine to another host.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>migration_downtime</td>
<td>Integer</td>
<td>Maximum time the virtual machine can be non responsive during its live migration to another host in ms.</td>
</tr>
<tr>
<td>multi_queues_enabled</td>
<td>Boolean</td>
<td>If true, each virtual interface will get the optimal number of queues, depending on the available virtual Cpus.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>origin</td>
<td>String</td>
<td>The origin of this virtual machine.</td>
</tr>
<tr>
<td>os</td>
<td>OperatingSystem</td>
<td>Operating system type installed on the virtual machine.</td>
</tr>
<tr>
<td>placement_policy</td>
<td>VmPlacementPolicy</td>
<td>The configuration of the virtual machine’s placement policy.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>Random Number Generator device configuration for this virtual machine.</td>
</tr>
<tr>
<td>serial_number</td>
<td>SerialNumber</td>
<td>Virtual machine’s serial number in a cluster.</td>
</tr>
<tr>
<td>small_icon</td>
<td>Icon</td>
<td>Virtual machine’s small icon.</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>If true, the sound card is added to the virtual machine.</td>
</tr>
<tr>
<td>sso</td>
<td>Sso</td>
<td>Reference to the Single Sign On configuration this virtual machine is configured for.</td>
</tr>
<tr>
<td>start_paused</td>
<td>Boolean</td>
<td>If true, the virtual machine will be initially in 'paused' state after start.</td>
</tr>
<tr>
<td>stateless</td>
<td>Boolean</td>
<td>If true, the virtual machine is stateless - it’s state (disks) are rolled-back after shutdown.</td>
</tr>
<tr>
<td>storage_error_resume_behaviour</td>
<td>VmStorageErrorResumeBehaviour</td>
<td>Determines how the virtual machine will be resumed after storage error.</td>
</tr>
<tr>
<td>time_zone</td>
<td>TimeZone</td>
<td>The virtual machine’s time zone set by oVirt.</td>
</tr>
<tr>
<td>tunnel_migration</td>
<td>Boolean</td>
<td>If true, the network data transfer will be encrypted during virtual machine live migration.</td>
</tr>
<tr>
<td>type</td>
<td>VmType</td>
<td>Determines whether the virtual machine is optimized for desktop or server.</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Summary</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>usb</td>
<td>Usb</td>
<td>Configuration of USB devices for this virtual machine (count, type).</td>
</tr>
<tr>
<td>virtio_scsi</td>
<td>VirtioScsi</td>
<td>Reference to VirtIO SCSI configuration.</td>
</tr>
</tbody>
</table>

### 7.290.1. cpu

The configuration of the virtual machine CPU.

The socket configuration can be updated without rebooting the virtual machine. The cores and the threads require a reboot.

For example, to change the number of sockets to 4 immediately, and the number of cores and threads to 2 after reboot, send the following request:

```
PUT /ovirt-engine/api/vms/123
```

With a request body:

```xml
<vm>
  <cpu>
    <topology>
      <sockets>4</sockets>
      <cores>2</cores>
      <threads>2</threads>
    </topology>
  </cpu>
</vm>
```

### 7.290.2. custom_compatibility_version

Virtual machine custom compatibility version.

Enables a virtual machine to be customized to its own compatibility version. If `custom_compatibility_version` is set, it overrides the cluster’s compatibility version for this particular virtual machine.

The compatibility version of a virtual machine is limited by the data center the virtual machine resides in, and is checked against capabilities of the host the virtual machine is planned to run on.

### 7.290.3. high_availability

The virtual machine high availability configuration. If set, the virtual machine will be automatically restarted when it unexpectedly goes down.

### 7.290.4. initialization

Reference to the virtual machine's initialization configuration.
NOTE

Since Red Hat Virtualization 4.1.8 this property can be cleared by sending an empty tag.

For example, to clear the `initialization` attribute send a request like this:

```plaintext
PUT /ovirt-engine/api/vms/123
```

With a request body like this:

```xml
<vm>
  <initialization/>
</vm>
```

The response to such a request, and requests with the header `All-Content: true` will still contain this attribute.

### 7.290.5. large_icon

Virtual machine’s large icon. Either set by user or refers to image set according to operating system.

### 7.290.6. lease

Reference to the storage domain this virtual machine/template lease reside on.

A virtual machine running with a lease requires checking while running that the lease is not taken by another host, preventing another instance of this virtual machine from running on another host. This provides protection against split-brain in highly available virtual machines. A template can also have a storage domain defined for a lease in order to have the virtual machines created from this template to be preconfigured with this storage domain as the location of the leases.

### 7.290.7. memory

The virtual machine’s memory, in bytes.

For example, to update a virtual machine to contain 1 Gibibyte (GiB) of memory, send the following request:

```plaintext
PUT /ovirt-engine/api/vms/123
```

With the following request body:

```xml
<vm>
  <memory>1073741824</memory>
</vm>
```

Memory hot plug is supported from Red Hat Virtualization 3.6 onwards. You can use the example above to increase memory while the virtual machine is in state `up`. The size increment must be dividable by the value of the `HotPlugMemoryBlockSizeMb` configuration value (256 MiB by default). If the memory size increment is not dividable by this value, the memory size change is only stored to next run configuration. Each successful memory hot plug operation creates one or two new memory devices.

Memory hot unplug is supported since Red Hat Virtualization 4.2 onwards. Memory hot unplug can only
be performed when the virtual machine is in state **up**. Only previously hot plugged memory devices can be removed by the hot unplug operation. The requested memory decrement is rounded down to match sizes of a combination of previously hot plugged memory devices. The requested memory value is stored to next run configuration without rounding.

**NOTE**

Memory in the example is converted to bytes using the following formula:

\[ 1 \text{ GiB} = 2^{30} \text{ bytes} = 1073741824 \text{ bytes}. \]

**NOTE**

Red Hat Virtualization Manager internally rounds values down to whole MiBs \((1 \text{ MiB} = 2^{20} \text{ bytes})\)

### 7.290.8. migration_downtime

Maximum time the virtual machine can be non responsive during its live migration to another host in ms.

Set either explicitly for the virtual machine or by `engine-config -s` 

```
DefaultMaximumMigrationDowntime=[value]
```

### 7.290.9. origin

The origin of this virtual machine.

Possible values:

- `ovirt`
- `rhev`
- `vmware`
- `xen`
- `external`
- `hosted_engine`
- `managed_hosted_engine`
- `kvm`
- `physical_machine`
- `hyperv`

### 7.290.10. placement_policy

The configuration of the virtual machine’s placement policy.

This configuration can be updated to pin a virtual machine to one or more hosts.
NOTE

Virtual machines that are pinned to multiple hosts cannot be live migrated, but in the event of a host failure, any virtual machine configured to be highly available is automatically restarted on one of the other hosts to which the virtual machine is pinned.

For example, to pin a virtual machine to two hosts, send the following request:

```plaintext
PUT /api/vms/123
```

With a request body like this:

```xml
<vm>
  <high_availability>
    <enabled>true</enabled>
    <priority>1</priority>
  </high_availability>
  <placement_policy>
    <hosts>
      <host>
        <name>Host1</name>
      </host>
      <host>
        <name>Host2</name>
      </host>
    </hosts>
    <affinity>pinned</affinity>
  </placement_policy>
</vm>
```

### 7.290.11. small_icon

Virtual machine’s small icon. Either set by user or refers to image set according to operating system.

### 7.290.12. sso

Reference to the Single Sign On configuration this virtual machine is configured for. The user can be automatically signed in the virtual machine’s operating system when console is opened.

#### Table 7.379. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to cluster the virtual machine belongs to.</td>
</tr>
<tr>
<td>cpu_profile</td>
<td>CpuProfile</td>
<td>Reference to CPU profile used by this virtual machine.</td>
</tr>
<tr>
<td>quota</td>
<td>Quota</td>
<td>Reference to quota configuration set for this virtual machine.</td>
</tr>
<tr>
<td>storage_domain</td>
<td>StorageDomain</td>
<td>Reference to storage domain the virtual machine belongs to.</td>
</tr>
</tbody>
</table>
7.291. VMDEVICETYPE ENUM

Table 7.380. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdrom</td>
<td></td>
</tr>
<tr>
<td>floppy</td>
<td></td>
</tr>
</tbody>
</table>

7.292. VMPLACEMENTPOLICY STRUCT

Table 7.381. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>affinity</td>
<td>VmAffinity</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.382. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>hosts</td>
<td>Host[]</td>
<td></td>
</tr>
</tbody>
</table>

7.293. VMPOOL STRUCT

Type representing a virtual machines pool.

Table 7.383. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_storage_select</td>
<td>Boolean</td>
<td>Indicates if the pool should automatically distribute the disks of the virtual machines across the multiple storage domains where the template is copied.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>display</td>
<td>Display</td>
<td>The display settings configured for virtual machines in the pool.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>max_user_vms</td>
<td>Integer</td>
<td>The maximum number of virtual machines in the pool that could be assigned to a particular user.</td>
</tr>
</tbody>
</table>
### Table 7.384. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>prestarted_vms</td>
<td>Integer</td>
<td>The system attempts to prestart the specified number of virtual machines from the pool.</td>
</tr>
<tr>
<td>rng_device</td>
<td>RngDevice</td>
<td>The random number generator device configured for virtual machines in the pool.</td>
</tr>
<tr>
<td>size</td>
<td>Integer</td>
<td>The number of virtual machines in the pool.</td>
</tr>
<tr>
<td>soundcard_enabled</td>
<td>Boolean</td>
<td>Indicates if sound card should be configured for virtual machines in the pool.</td>
</tr>
<tr>
<td>stateful</td>
<td>Boolean</td>
<td>Virtual machine pool’s stateful flag.</td>
</tr>
<tr>
<td>type</td>
<td>VmPoolType</td>
<td>The deallocation policy of virtual machines in the pool.</td>
</tr>
<tr>
<td>use_latest_template_version</td>
<td>Boolean</td>
<td>Indicates if virtual machines in the pool are updated to newer versions of the template the pool is based on.</td>
</tr>
</tbody>
</table>

#### 7.293.1. auto_storage_select

Indicates if the pool should automatically distribute the disks of the virtual machines across the multiple storage domains where the template is copied.

When the template used by the pool is present in multiple storage domains, the disks of the virtual machines of the pool will be created in one of those storage domains. By default, or when the value of this attribute is `false`, that storage domain is selected when the pool is created, and all virtual machines will use the same. If this attribute is `true`, then, when a virtual machine is added to the pool, the storage domain that has more free space is selected.

#### 7.293.2. prestarted_vms

The system attempts to prestart the specified number of virtual machines from the pool.

These virtual machines are started without being attached to any user. That way, users can acquire virtual machines from the pool faster.

#### 7.293.3. stateful

Virtual machine pool’s stateful flag.

Virtual machines from a stateful virtual machine pool are always started in stateful mode (stateless snapshot is not created). The state of the virtual machine is preserved even when the virtual machine is passed to a different user.
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>Cluster</td>
<td>Reference to the cluster the pool resides in.</td>
</tr>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Reference to the instance type on which this pool is based.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Permissions set for this virtual machine pool.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Reference to the template the pool is based on.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Reference to an arbitrary virtual machine that is part of the pool.</td>
</tr>
</tbody>
</table>

### 7.293.4. instance_type
Reference to the instance type on which this pool is based. It can be set only on pool creation and cannot be edited.

### 7.293.5. vm
Reference to an arbitrary virtual machine that is part of the pool.

Note that this virtual machine may not be based to the latest version of the pool’s template.

### 7.294. VMPOOLTYPE ENUM
Type representing the deallocation policy of virtual machines in a virtual machines pool.

**Table 7.385. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>automatic</td>
<td>This policy indicates that virtual machines in the pool are automatically deallocated by the system.</td>
</tr>
<tr>
<td>manual</td>
<td>This policy indicates that virtual machines in the pool are deallocated manually by the administrator.</td>
</tr>
</tbody>
</table>

#### 7.294.1. automatic
This policy indicates that virtual machines in the pool are automatically deallocated by the system.

With this policy, when a virtual machine that is part of the pool and is assigned to a user is shut-down, it is detached from the user, its state is restored to the pool’s default state, and the virtual machine returns to pool (i.e., the virtual machine can then be assigned to another user).

#### 7.294.2. manual
This policy indicates that virtual machines in the pool are deallocated manually by the administrator.
With this policy, a virtual machine that is part of the pool remains assigned to its user and preserves its state on shut-down. In order to return the virtual machine back to the pool, the administrator needs to deallocate it explicitly by removing the user’s permissions on that virtual machine.

### 7.295. VMSTATUS ENUM

Type representing a status of a virtual machine.

#### Table 7.386. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>down</strong></td>
<td>This status indicates that the virtual machine process is not running.</td>
</tr>
<tr>
<td><strong>image_locked</strong></td>
<td>This status indicates that the virtual machine process is not running and there is some operation on the disks of the virtual machine that prevents it from being started.</td>
</tr>
<tr>
<td><strong>migrating</strong></td>
<td>This status indicates that the virtual machine process is running and the virtual machine is being migrated from one host to another.</td>
</tr>
<tr>
<td><strong>not_responding</strong></td>
<td>This status indicates that the hypervisor detected that the virtual machine is not responding.</td>
</tr>
<tr>
<td><strong>paused</strong></td>
<td>This status indicates that the virtual machine process is running and the virtual machine is paused.</td>
</tr>
<tr>
<td><strong>powering_down</strong></td>
<td>This status indicates that the virtual machine process is running and it is about to stop running.</td>
</tr>
<tr>
<td><strong>powering_up</strong></td>
<td>This status indicates that the virtual machine process is running and the guest operating system is being loaded.</td>
</tr>
<tr>
<td><strong>reboot_in_progress</strong></td>
<td>This status indicates that the virtual machine process is running and the guest operating system is being rebooted.</td>
</tr>
<tr>
<td><strong>restoring_state</strong></td>
<td>This status indicates that the virtual machine process is about to run and the virtual machine is going to awake from hibernation.</td>
</tr>
<tr>
<td><strong>saving_state</strong></td>
<td>This status indicates that the virtual machine process is running and the virtual machine is being hibernated.</td>
</tr>
<tr>
<td><strong>suspended</strong></td>
<td>This status indicates that the virtual machine process is not running and a running state of the virtual machine was saved.</td>
</tr>
<tr>
<td><strong>unassigned</strong></td>
<td>This status is set when an invalid status is received.</td>
</tr>
<tr>
<td><strong>unknown</strong></td>
<td>This status indicates that the system failed to determine the status of the virtual machine.</td>
</tr>
</tbody>
</table>
### Table: Statuses of Virtual Machine Processes

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>up</strong></td>
<td>This status indicates that the virtual machine process is running and the guest operating system is loaded.</td>
</tr>
<tr>
<td><strong>wait_for_launch</strong></td>
<td>This status indicates that the virtual machine process is about to run.</td>
</tr>
</tbody>
</table>

#### 7.295.1. paused

This status indicates that the virtual machine process is running and the virtual machine is paused. This may happen in two cases: when running a virtual machine is paused mode and when the virtual machine is being automatically paused due to an error.

#### 7.295.2. powering_up

This status indicates that the virtual machine process is running and the guest operating system is being loaded. Note that if no guest-agent is installed, this status is set for a predefined period of time, that is by default 60 seconds, when running a virtual machine.

#### 7.295.3. restoring_state

This status indicates that the virtual machine process is about to run and the virtual machine is going to awake from hibernation. In this status, the running state of the virtual machine is being restored.

#### 7.295.4. saving_state

This status indicates that the virtual machine process is running and the virtual machine is being hibernated. In this status, the running state of the virtual machine is being saved. Note that this status does not mean that the guest operating system is being hibernated.

#### 7.295.5. suspended

This status indicates that the virtual machine process is not running and a running state of the virtual machine was saved. This status is similar to Down, but when the VM is started in this status its saved running state is restored instead of being booted using the normal procedure.

#### 7.295.6. unknown

This status indicates that the system failed to determine the status of the virtual machine. The virtual machine process may be running or not running in this status. For instance, when host becomes non-responsive the virtual machines that ran on it are set with this status.

#### 7.295.7. up

This status indicates that the virtual machine process is running and the guest operating system is loaded. Note that if no guest-agent is installed, this status is set after a predefined period of time, that is by default 60 seconds, when running a virtual machine.

#### 7.295.8. wait_for_launch
This status indicates that the virtual machine process is about to run. This status is set when a request to run a virtual machine arrives to the host. It is possible that the virtual machine process will fail to run.

**7.296. VMSTORAGEERRORRESUMEBEHAVIOUR ENUM**

If the storage, on which this virtual machine has some disks gets unresponsive, the virtual machine gets paused.

This are the possible options, what should happen with the virtual machine in the moment the storage gets available again.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_resume</td>
<td>The virtual machine gets resumed automatically in the moment the storage is available again.</td>
</tr>
<tr>
<td>kill</td>
<td>The virtual machine will be killed after a timeout (configurable on the hypervisor).</td>
</tr>
<tr>
<td>leave_paused</td>
<td>Do nothing with the virtual machine.</td>
</tr>
</tbody>
</table>

**7.296.1. auto_resume**

The virtual machine gets resumed automatically in the moment the storage is available again.

This is the only behavior available before 4.2.

**7.296.2. kill**

The virtual machine will be killed after a timeout (configurable on the hypervisor).

This is the only option supported for highly available virtual machines with leases. The reason is that the highly available virtual machine is restarted using the infrastructure and any kind of resume risks split brains.

**7.296.3. leave_paused**

Do nothing with the virtual machine.

Useful if there is a custom failover implemented and the user does not want the virtual machine to get resumed.

**7.297. VMSUMMARY STRUCT**

Type containing information related to virtual machines on a particular host.

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
</table>
### active
- **Type:** Integer
- **Summary:** The number of virtual machines active on the host.

### migrating
- **Type:** Integer
- **Summary:** The number of virtual machines migrating to or from the host.

### total
- **Type:** Integer
- **Summary:** The number of virtual machines present on the host.

---

### 7.298. VMTYPE ENUM

Type representing what the virtual machine is optimized for.

### Table 7.389. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>desktop</td>
<td>The virtual machine is intended to be used as a desktop.</td>
</tr>
<tr>
<td>high_performance</td>
<td>The virtual machine is intended to be used as a high performance virtual machine.</td>
</tr>
<tr>
<td>server</td>
<td>The virtual machine is intended to be used as a server.</td>
</tr>
</tbody>
</table>

#### 7.298.1. desktop

The virtual machine is intended to be used as a desktop.

Currently, its implication is that a sound device will automatically be added to the virtual machine.

#### 7.298.2. high_performance

The virtual machine is intended to be used as a high performance virtual machine.

Currently, its implication is that the virtual machine configuration will automatically be set for running with the highest possible performance, and with performance metrics as close to bare metal as possible.

Some of the recommended configuration settings for the highest possible performance cannot be set automatically; manually setting them before running the virtual machine is recommended.

The following configuration changes are set automatically:

- Enable headless mode.
- Enable serial console.
- Enable pass-through host CPU.
- Enable I/O threads.
- Enable I/O threads pinning and set the pinning topology.
- Enable the paravirtualized random number generator PCI (virtio-rng) device.
- Disable all USB devices.
- Disable the soundcard device.
- Disable the smartcard device.
- Disable the memory balloon device.
- Disable the watchdog device.
- Disable migration.
- Disable high availability.

The following recommended configuration changes have to be set manually by the user:

- Enable CPU pinning topology.
- Enable non-uniform memory access (NUMA) pinning topology.
- Enable and set huge pages configuration.
- Disable kernel same-page merging (KSM).

### 7.298.3. server

The virtual machine is intended to be used as a server.

Currently, its implication is that a sound device will not automatically be added to the virtual machine.

### 7.299. VNICPASSTHROUGH STRUCT

**Table 7.390. Attributes summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>VnicPassThrough Mode</td>
<td>Defines whether the vNIC will be implemented as a virtual device, or as a pass-through to a host device.</td>
</tr>
</tbody>
</table>

### 7.300. VNICPASSTHROUGHMODE ENUM

Describes whether the vNIC is to be implemented as a pass-through device or a virtual one.

**Table 7.391. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>disabled</td>
<td>To be implemented as a virtual device.</td>
</tr>
<tr>
<td>enabled</td>
<td>To be implemented as a pass-through device.</td>
</tr>
</tbody>
</table>
7.301. VNICPROFILE STRUCT

A vNIC profile is a collection of settings that can be applied to individual NIC.

Table 7.392. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>custom_properties</td>
<td>CustomProperty[]</td>
<td>Custom properties applied to the vNIC profile.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>migratable</td>
<td>Boolean</td>
<td>Marks whether pass_through NIC is migratable or not.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
<tr>
<td>pass_through</td>
<td>VnicPassThrough</td>
<td>Enables passthrough to an SR-IOV-enabled host NIC.</td>
</tr>
<tr>
<td>port_mirroring</td>
<td>Boolean</td>
<td>Enables port mirroring.</td>
</tr>
</tbody>
</table>

7.301.1. migratable

Marks whether pass_through NIC is migratable or not.

If pass_through.mode is set to disabled this option has no meaning, and it will be considered to be true. If you omit this option from a request, by default, this will be set to true.

When migrating a virtual machine, this virtual machine will be migrated only if all pass_through NICs are flagged as migratable.

7.301.2. pass_through

Enables passthrough to an SR-IOV-enabled host NIC.

A vNIC profile enables a NIC to be directly connected to a virtual function (VF) of an SR-IOV-enabled host NIC, if passthrough is enabled. The NIC will then bypass the software network virtualization and connect directly to the VF for direct device assignment.

Passthrough cannot be enabled if the vNIC profile is already attached to a NIC. If a vNIC profile has passthrough enabled, qos and port_mirroring are disabled for the vNIC profile.

7.301.3. port_mirroring

Enables port mirroring.
Port mirroring copies layer 3 network traffic on a given logical network and host to a NIC on a virtual machine. This virtual machine can be used for network debugging and tuning, intrusion detection, and monitoring the behavior of other virtual machines on the same host and logical network. The only traffic copied is internal to one logical network on one host. There is no increase in traffic on the network external to the host; however a virtual machine with port mirroring enabled uses more host CPU and RAM than other virtual machines.

Port mirroring has the following limitations:

- Hot plugging a NIC with a vNIC profile that has port mirroring enabled is not supported.
- Port mirroring cannot be altered when the vNIC profile is attached to a virtual machine.

Given the above limitations, it is recommended that you enable port mirroring on an additional, dedicated vNIC profile.

**IMPORTANT**

Enabling port mirroring reduces the privacy of other network users.

### Table 7.393. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>Network</td>
<td>Reference to the network that the vNIC profile is applied to.</td>
</tr>
<tr>
<td>network_filter</td>
<td>NetworkFilter</td>
<td>Reference to the top-level network filter that applies to the NICs that use this profile.</td>
</tr>
<tr>
<td>permissions</td>
<td>Permission[]</td>
<td>Permissions to allow usage of the vNIC profile.</td>
</tr>
<tr>
<td>qos</td>
<td>Qos</td>
<td>Reference to the quality of service attributes to apply to the vNIC profile.</td>
</tr>
</tbody>
</table>

#### 7.301.4. network_filter

Reference to the top-level network filter that applies to the NICs that use this profile.

Network filters enhance the ability to manage the network packets traffic to and from virtual machines. The network filter may either contain a reference to other filters, rules for traffic filtering, or a combination of both.

#### 7.301.5. qos

Reference to the quality of service attributes to apply to the vNIC profile.

Quality of Service attributes regulate inbound and outbound network traffic of the NIC.

#### 7.302. VNICPROFILEMAPPING STRUCT

Deprecated type that maps an external virtual NIC profile to one that exists in the Red Hat Virtualization Manager.
If, for example, the desired virtual NIC profile’s mapping includes the following two lines:

<table>
<thead>
<tr>
<th>Source network name</th>
<th>Source network profile name</th>
<th>Target virtual NIC profile ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>red</td>
<td>gold</td>
<td>738dd914-8ec8-4a8b-8628-34672a5d449b</td>
</tr>
<tr>
<td>blue</td>
<td>silver</td>
<td>892a12ec-2028-4451-80aa-ff3bf55d6bac</td>
</tr>
</tbody>
</table>

The following form is deprecated since 4.2.1 and will be removed in the future:

```xml
<vnic_profile_mappings>
  <vnic_profile_mapping>
    <source_network_name>red</source_network_name>
    <source_network_profile_name>gold</source_network_profile_name>
    <target_vnic_profile_id>738dd914-8ec8-4a8b-8628-34672a5d449b</target_vnic_profile_id>
  </vnic_profile_mapping>
  <vnic_profile_mapping>
    <source_network_name>blue</source_network_name>
    <source_network_profile_name>silver</source_network_profile_name>
    <target_vnic_profile_id>892a12ec-2028-4451-80aa-ff3bf55d6bac</target_vnic_profile_id>
  </vnic_profile_mapping>
</vnic_profile_mappings>
```

### Table 7.394. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_network_name</td>
<td>String</td>
<td>Deprecated attribute describing the name of the external network.</td>
</tr>
<tr>
<td>source_network_profile_name</td>
<td>String</td>
<td>Deprecated attribute describing the name of the external network profile.</td>
</tr>
</tbody>
</table>

#### 7.302.1. source_network_name

Deprecated attribute describing the name of the external network.

**WARNING**

Please note that this attribute has been deprecated since version 4.2.1 of the engine, and preserved only for backward compatibility. It will be removed in the future.
7.302.2. source_network_profile_name

Deprecated attribute describing the name of the external network profile.

**WARNING**
Please note that this attribute has been deprecated since version 4.2.1 of the engine, and preserved only for backward compatibility. It will be removed in the future.

Table 7.395. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>target_vnic_profile</td>
<td>VnicProfile</td>
<td>Deprecated attribute describing an existing virtual NIC profile.</td>
</tr>
</tbody>
</table>

7.302.3. target_vnic_profile

Deprecated attribute describing an existing virtual NIC profile.

**WARNING**
Please note that this attribute has been deprecated since version 4.2.1 of the engine, and preserved only for backward compatibility. It will be removed in the future.

7.303. VOLUMEGROUP STRUCT

Table 7.396. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>logical_units</td>
<td>LogicalUnit[]</td>
<td></td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

7.304. WATCHDOG STRUCT
This type represents a watchdog configuration.

### Table 7.397. Attributes summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>WatchdogAction</td>
<td>Watchdog action to be performed when watchdog is triggered.</td>
</tr>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>model</td>
<td>WatchdogModel</td>
<td>Model of watchdog device.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

#### 7.304.1. model

Model of watchdog device. Currently supported only i6300ESB.

### Table 7.398. Links summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>instance_type</td>
<td>InstanceType</td>
<td>Optionally references to an instance type the device is used by.</td>
</tr>
<tr>
<td>template</td>
<td>Template</td>
<td>Optionally references to a template the device is used by.</td>
</tr>
<tr>
<td>vm</td>
<td>Vm</td>
<td>Don’t use this element, use vms instead.</td>
</tr>
<tr>
<td>vms</td>
<td>Vm[]</td>
<td>References to the virtual machines that are using this device.</td>
</tr>
</tbody>
</table>

#### 7.304.2. vms

References to the virtual machines that are using this device. A device may be used by several virtual machines; for example, a shared disk may be used simultaneously by two or more virtual machines.

### 7.305. WATCHDOGACTION ENUM

This type describes available watchdog actions.

### Table 7.399. Values summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>dump</td>
<td>Virtual machine process will get core dumped to the default path on the host.</td>
</tr>
<tr>
<td>Name</td>
<td>Summary</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>none</td>
<td>No action will be performed when watchdog action is triggered.</td>
</tr>
<tr>
<td>pause</td>
<td>Virtual machine will be paused when watchdog action is triggered.</td>
</tr>
<tr>
<td>poweroff</td>
<td>Virtual machine will be powered off when watchdog action is triggered.</td>
</tr>
<tr>
<td>reset</td>
<td>Virtual machine will be rebooted when watchdog action is triggered.</td>
</tr>
</tbody>
</table>

**7.305. none**

No action will be performed when watchdog action is triggered. However log message will still be generated.

**7.306. WATCHDOGMODEL ENUM**

This type represents the watchdog model.

**Table 7.400. Values summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>diag288</td>
<td>The watchdog model for S390X machines.</td>
</tr>
<tr>
<td>i6300esb</td>
<td>PCI based watchdog model.</td>
</tr>
</tbody>
</table>

**7.306.1. diag288**

The watchdog model for S390X machines.

S390X has an integrated watchdog facility that is controlled via the DIAG288 instruction. Use this model for S390X virtual machines.

**7.306.2. i6300esb**

PCI based watchdog model.

Use the I6300ESB watchdog for x86_64 and PPC64 virtual machines.

**7.307. WEIGHT STRUCT**

**Table 7.401. Attributes summary**
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>comment</td>
<td>String</td>
<td>Free text containing comments about this object.</td>
</tr>
<tr>
<td>description</td>
<td>String</td>
<td>A human-readable description in plain text.</td>
</tr>
<tr>
<td>factor</td>
<td>Integer</td>
<td></td>
</tr>
<tr>
<td>id</td>
<td>String</td>
<td>A unique identifier.</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>A human-readable name in plain text.</td>
</tr>
</tbody>
</table>

**Table 7.402. Links summary**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>scheduling_policy</td>
<td>SchedulingPolicy</td>
<td></td>
</tr>
<tr>
<td>scheduling_policy_unit</td>
<td>SchedulingPolicyUnit</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A. PRIMITIVE TYPES

This section describes the primitive data types supported by the API.

A.1. STRING PRIMITIVE

A finite sequence of Unicode characters.

A.2. BOOLEAN PRIMITIVE

Represents the false and true concepts used in mathematical logic.

The valid values are the strings false and true.

Case is ignored by the engine, so for example False and FALSE also valid values. However the server will always return lower case values.

For backwards compatibility with older versions of the engine, the values 0 and 1 are also accepted. The value 0 has the same meaning than false, and 1 has the same meaning than true. Try to avoid using these values, as support for them may be removed in the future.

A.3. INTEGER PRIMITIVE

Represents the mathematical concept of integer number.

The valid values are finite sequences of decimal digits.

Currently the engine implements this type using a signed 32 bit integer, so the minimum value is $-2^{31}$ (-2147483648) and the maximum value is $2^{31}-1$ (2147483647).

However, there are some attributes in the system where the range of values possible with 32 bit isn’t enough. In those exceptional cases the engine uses 64 bit integers, in particular for the following attributes:

- Disk.actual_size
- Disk.provisioned_size
- GlusterClient.bytes_read
- GlusterClient.bytes_written
- Host.max_scheduling_memory
- Host.memory
- HostNic.speed
- LogicalUnit.size
- MemoryPolicy.guaranteed
- NumaNode.memory
- QuotaStorageLimit.limit
- StorageDomain.available
- StorageDomain.used
- StorageDomain.committed
- VmBase.memory

For these exception cases the minimum value is \(-2^{63}\) (-9223372036854775808) and the maximum value is \(2^{63} - 1\) (9223372036854775807).

**NOTE**
In the future the integer type will be implemented using unlimited precision integers, so the above limitations and exceptions will eventually disappear.

### A.4. DECIMAL PRIMITIVE

Represents the mathematical concept of real number.

Currently the engine implements this type using 32 bit IEEE 754 single precision floating point numbers.

For some attributes this isn't enough precision. In those exceptional cases the engine uses 64 bit double precision floating point numbers, in particular for the following attributes:

- QuotaStorageLimit.usage
- QuotaStorageLimit.memory_limit
- QuotaStorageLimit.memory_usage

**NOTE**
In the future the decimal type will be implemented using unlimited precision decimal numbers, so the above limitations and exceptions will eventually disappear.

### A.5. DATE PRIMITIVE

Represents a date and time.

The format returned by the engine is the one described in the XML Schema specification when requesting XML. For example, if you send a request like this to retrieve the XML representation of a virtual machine:

```plaintext
GET /ovirt-engine/api/vms/123
Accept: application/xml
```

The response body will contain the following XML document:

```xml
<vm id="123" href="/ovirt-engine/api/vms/123">
  ...
  <creation_time>2016-09-08T09:53:35.138+02:00</creation_time>
</vm>
```
When requesting the JSON representation the engine uses a different, format: an integer containing the number of seconds since Jan 1\textsuperscript{st} 1970, also know as \textit{epoch time}. For example, if you send a request like this to retrieve the JSON representation of a virtual machine:

\begin{verbatim}
GET /ovirt-engine/api/vms/123
Accept: application/json
\end{verbatim}

The response body will contain the following JSON document:

\begin{verbatim}
{
  "id": "123",
  "href": "/ovirt-engine/api/vms/123",
  ...
  "creation_time": 1472564909990,
  ...
}
\end{verbatim}

\textbf{NOTE}

In both cases, the dates returned by the engine use the time zone configured in the server where it is running, in the above examples it is UTC+2.
APPENDIX B. CHANGES IN VERSION 4 OF THE API

This section enumerates the backwards compatibility breaking changes that have been introduced in version 4 of the API.

B.1. REMOVED YAML SUPPORT

The support for YAML has been completely removed.

B.2. RENAMED COMPLEX TYPES

The following XML schema complex types have been renamed:

<table>
<thead>
<tr>
<th>Version 3</th>
<th>Version 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Api</td>
</tr>
<tr>
<td>CPU</td>
<td>Cpu</td>
</tr>
<tr>
<td>CPUs</td>
<td>Cpus</td>
</tr>
<tr>
<td>CdRom</td>
<td>Cdrom</td>
</tr>
<tr>
<td>CdRoms</td>
<td>Cdroms</td>
</tr>
<tr>
<td>DNS</td>
<td>Dns</td>
</tr>
<tr>
<td>GuestNicConfiguration</td>
<td>NicConfiguration</td>
</tr>
<tr>
<td>GuestNicsConfiguration</td>
<td>NicConfigurations</td>
</tr>
<tr>
<td>HostNICStates</td>
<td>HostNicStates</td>
</tr>
<tr>
<td>HostNIC</td>
<td>HostNic</td>
</tr>
<tr>
<td>HostStorage</td>
<td>HostStorages</td>
</tr>
<tr>
<td>IO</td>
<td>Io</td>
</tr>
<tr>
<td>IP</td>
<td>Ip</td>
</tr>
<tr>
<td>IPs</td>
<td>Ips</td>
</tr>
<tr>
<td>KSM</td>
<td>Ksm</td>
</tr>
<tr>
<td>MAC</td>
<td>Mac</td>
</tr>
<tr>
<td>Version 3</td>
<td>Version 4</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>NIC</td>
<td>Nic</td>
</tr>
<tr>
<td>PreviewVMs</td>
<td>PreviewVms</td>
</tr>
<tr>
<td>QoS</td>
<td>Qos</td>
</tr>
<tr>
<td>QoSs</td>
<td>Qoss</td>
</tr>
<tr>
<td>RSDL</td>
<td>Rsd1</td>
</tr>
<tr>
<td>SELinux</td>
<td>Selinux</td>
</tr>
<tr>
<td>SPM</td>
<td>Smp</td>
</tr>
<tr>
<td>SSHPublicKey</td>
<td>SshPublicKey</td>
</tr>
<tr>
<td>SSHPublicKeys</td>
<td>SshPublicKeys</td>
</tr>
<tr>
<td>SSH</td>
<td>Ssh</td>
</tr>
<tr>
<td>SkipIfSDActive</td>
<td>SkipIfSdActive</td>
</tr>
<tr>
<td>Slaves</td>
<td>HostNics</td>
</tr>
<tr>
<td>Storage</td>
<td>HostStorage</td>
</tr>
<tr>
<td>SupportedVersions</td>
<td>Versions</td>
</tr>
<tr>
<td>VcpuPin</td>
<td>VcpuPin</td>
</tr>
<tr>
<td>VLAN</td>
<td>Vlan</td>
</tr>
<tr>
<td>VM</td>
<td>Vm</td>
</tr>
<tr>
<td>VMs</td>
<td>Vms</td>
</tr>
<tr>
<td>VirtIO_SCSI</td>
<td>VirtioScsi</td>
</tr>
<tr>
<td>WatchDog</td>
<td>Watchdog</td>
</tr>
<tr>
<td>WatchDogs</td>
<td>Watchdogs</td>
</tr>
</tbody>
</table>

B.3. REPLACED THE STATUS TYPE WITH ENUM TYPES
Currently the status of different objects is reported using the Status type, which contains a state string describing the status and another detail string for additional details. For example, the status of a virtual machine that is paused due to an IO error is currently reported as follows:

```xml
<vm>
  ...
  <status>
    <state>paused</state>
    <detail>eio</detail>
  </status>
  ...
</vm>
```

In version 4 of the API this Status type has been removed and replaced by enum types. When the additional detail string is needed it has been replaced with an additional status_detail attribute. So, for example, the status of the same virtual machine will now be reported as follows:

```xml
<vm>
  ...
  <status>paused</status>
  <status_detail>eio</status_detail>
  ...
</vm>
```

### B.4. REMOVE THE NIC NETWORK AND PORT_MIRRORING PROPERTIES

The NIC network and port_mirroring elements have been replaced by the vnic_profile element, so when creating or updating a NIC instead of specifying the network and port mirroring configuration, these are previously specified creating a vNIC profile:

```xml
POST /ovirt-engine/api/vnicprofiles

<vnic_profile>
  <name>myprofile</name>
  <network_id="..."/>
  <port_mirroring>true</port_mirroring>
</vnic_profile>
```

And then the NIC is created or referencing the existing vNIC profile:

```xml
PUT /ovirt-engine/api/vms/123/nics/456

<nic>
  <vnic_profile_id="/vnicprofiles/..."/>
</nic>
```

The old elements and their meaning were preserved for backwards compatibility, but they have now been completely removed.

Note that the network element hasn’t been removed from the XML schema because it is still used by the initialization element, but it will be completely ignored if provided when creating or updating a NIC.
B.5. REMOVE THE NIC active PROPERTY

The NIC active property was replaced by plugged some time ago. It has been completely removed now.

B.6. REMOVE THE DISK type PROPERTY

The type property of disks has been removed, but kept in the XML schema and ignored. It has been completely removed now.

B.7. REMOVE THE DISK size PROPERTY

The disk size property has been replaced by provisioned_size long ago. It has been completely removed now.

B.8. REMOVED SUPPORT FOR PINNING A VM TO A SINGLE HOST

Before version 3.6 the API had the possibility to pin a VM to a single host, using the placement_policy element of the VM entity:

```
PUT /ovirt-engine/api/vms/123

<vm>
  <placement_policy>
    <host id="456"/>
  </placement_policy>
<vm>
```

In version 3.6 this capability was enhanced to support multiple hosts, and to do so a new hosts element was added:

```
PUT /ovirt-engine/api/vms/123

<vm>
  <placement_policy>
    <hosts>
      <host id="456"/>
      <host id="789"/>
    </hosts>
    ...
  </placement_policy>
<vm>
```

To preserve backwards compatibility the single host element was preserved. In 4.0 this has been removed, so applications will need to use the hosts element even if when pinning to a single host.

B.9. REMOVED THE capabilities.permits ELEMENT
The list of permits is potentially different for each cluster level, and it has been added to the `version` element long ago, but it has been kept into the `capabilities` element as well, just for backwards compatibility.

In 4.0 the `capabilities` service has been completely removed, and replaced by the new `clusterlevels` service. To find the permits supported by cluster level 4.0 a request like this should be used:

```
GET /ovirt-engine/api/clusterlevels/4.0
```

The result will be a document containing the information specific to that cluster level, in particular the set of supported permits:

```
<cluster_level id="4.0" href="/clusterlevels/4.0">
...
<permits>
 <permit id="1">
  <name>create_vm</name>
  <administrative>false</administrative>
 </permit>
...
</permits>
</cluster_level>
```

### B.10. REMOVED THE STORAGE_MANAGER ELEMENT

The `storage_manager` element was replaced by the `spm` element some time ago. The old one was kept for backwards compatibility, but it has been completely removed now.

### B.11. REMOVED THE DATA CENTER STORAGE_TYPE ELEMENT

Data centers used to be associated to a specific storage type (NFS, Fiber Channel, iSCSI, etc) but they have been changed some time so that there are only two types: with local storage and with shared storage. A new `local` element was introduced to indicate this, and the old `storage_type` was element was preserved for backwards compatibility. This old element has now been completely removed.

### B.12. REMOVE THE TIMEZONE ELEMENT

The VM resource used to contain a `timezone` element to represent the time zone. This element only allowed a string:

```
<vm>
 <timezone>Europe/Madrid</timezone>
</vm>
```

This doesn't allow extension, and as a it was necessary to add the UTC offset, it was replaced with a new structured `time_zone` element:

```
<vm>
 <time_zone>
   <name>Europe/Madrid</name>
 </time_zone>
</vm>
```
The old timezone element was preserved, but it has been completely removed now.

**B.13. REMOVED THE GUEST_INFO ELEMENT**

The guest_info element was used to hold information gathered by the guest agent, like the IP addresses and the fully qualified host name. This information is also available in other places. For example, the IP addresses are available within VM resource:

```
GET /ovirt-engine/api/vms/123
```

```
<vm>
  <guest_info>
    <ips>
      <ip address="192.168.122.30"/>
    </ips>
    <fqdn>myvm.example.com</fqdn>
  </guest_info>
</vm>
```

And also within the NIC resource, using the newer reported_devices element:

```
GET /ovirt-engine/api/vms/{vm:id}/nics/{nic:id}
```

```
<nic>
  <reported_devices>
    <reported_device>
      <name>eth0</name>
      <mac address="00:1a:4a:b5:4c:94"/>
      <ips>
        <ip address="192.168.1.115" version="v4"/>
        <ip address="fe80::21a:4aff:feb5:4c94" version="v6"/>
        <ip address="::1:21a:4aff:feb5:4c94" version="v6"/>
      </ips>
    </reported_device>
  </reported_devices>
</nic>
```

In addition this newer reported_devices element provides more complete information, like multiple IP addresses, MAC addresses, etc.

To remove this duplication the guest_info element has been removed.

To support the fully qualified domain name a new fqdn element has been added to the VM resource:

```
GET /ovirt-engine/api/vms/123
```

```
<vm>
  <fqdn>myvm.example.com</fqdn>
</vm>
```
This will contain the same information that guest_info fqdn used to contain.

B.14. REPLACED CPU ID ATTRIBUTE WITH TYPE ELEMENT

The cpu element used to have an id attribute that indicates the type of CPU:

```xml
<cpu id="Intel Conroe Family">
  <architecture>X86_64</architecture>
  ...
</cpu>
```

This is in contradiction with the rest of the elements of the API model, where the id attribute is used for opaque identifiers. This id attribute has been replaced with a new type element:

```xml
<cpu>
  <type>Intel Conroe Family</type>
  <architecture>X86_64</architecture>
</cpu>
```

B.15. USE ELEMENTS INSTEAD OF ATTRIBUTES IN CPU TOPOLOGY

In the past the CPU topology element used attributes for its properties:

```xml
<cpu>
  <topology sockets="1" cores="1" threads="1"/>
  ...
</cpu>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<cpu>
  <topology>
    <sockets>1</sockets>
    <cores>1</cores>
    <threads>1</threads>
  </topology>
  ...
</cpu>
```

B.16. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VCPU PIN

In the past the VCPU pin element used attributes for its properties:

```xml
<cpu_tune>
  <vcpu_pin vcpu="0" cpu_set="0"/>
</cpu_tune>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<cpu_tune>
```
B.17. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VCPU PIN

In the past the version element used attributes for its properties:

```xml
<version major="3" minor="5" />
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<version>
  <major>3</major>
  <minor>5</minor>
  ...
</version>
```

B.18. USE ELEMENTS INSTEAD OF ATTRIBUTES IN MEMORY OVERCOMMIT

In the past the overcommit element used attributes for its properties:

```xml
<memory_policy>
  <overcommit percent="100"/>
  ...
</memory_policy>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<memory_policy>
  <overcommit>
    <percent>100</percent>
  </overcommit>
  ...
</memory_policy>
```

B.19. USE ELEMENTS INSTEAD OF ATTRIBUTES IN CONSOLE

In the past the console element used attributes for its properties:

```xml
<console enabled="true"/>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<console>
  <enabled>true</enabled>
</console>
```
B.20. USE ELEMENTS INSTEAD OF ATTRIBUTES IN VIRTIO SCSI

In the past the VIRTIO ISCSI element used attributes for its properties:

```xml
<virtio_scsi enabled="true"/>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<virtio_scsi>
    <enabled>true</enabled>
</virtio_scsi>
```

B.21. USE ELEMENT INSTEAD OF ATTRIBUTE FOR POWER MANAGEMENT AGENT TYPE

The power management type property was represented as an attribute:

```xml
<agent type="apc">
    <username>myuser</username>
    ...
</agent>
```

This is contrary to the common practice in the API. It has been replaced with an inner element:

```xml
<agent>
    <type>apc</type>
    <username>myuser</username>
    ...
</agent>
```

B.22. USE ELEMENTS INSTEAD OF ATTRIBUTES IN POWER MANAGEMENT AGENT OPTIONS

In the past the power management agent options element used attributes for its properties:

```xml
<options>
    <option name="port" value="22"/>
    <option name="slot" value="5"/>
    ...
</options>
```

This is contrary to the common practice in the API. They have been replaced with inner elements:

```xml
<options>
    <option>
        <name>port</name>
        <value>22</value>
    </option>
    <option>
        <name>slot</name>
        <value>5</value>
    </option>
</options>
```
B.23. USE ELEMENTS INSTEAD OF ATTRIBUTES IN IP ADDRESS:

In the past the IP address element used attributes for its properties:

```xml
<ip address="192.168.122.1" netmask="255.255.255.0"/>
```

This is contrary to the common practice in the API. They have been replaced with inner elements:

```xml
<ip>
  <address>192.168.122.1</address>
  <netmask>255.255.255.0</netmask>
</ip>
```

B.24. USE ELEMENTS INSTEAD OF ATTRIBUTES IN MAC ADDRESS:

In the past the MAC address element used attributes for its properties:

```xml
<mac address="66:f2:c5:5f:bb:8d"/>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<mac>
  <address>66:f2:c5:5f:bb:8d</address>
</mac>
```

B.25. USE ELEMENTS INSTEAD OF ATTRIBUTES IN BOOT DEVICE:

In the past the boot device element used attributes for its properties:

```xml
<boot dev="cdrom"/>
```

This is contrary to the common practice in the API. They have been replaced by inner elements:

```xml
<boot>
  <dev>cdrom</dev>
</boot>
```

B.26. USE ELEMENT INSTEAD OF ATTRIBUTE FOR OPERATING SYSTEM TYPE

The operating system `type` property was represented as an attribute:

```xml
<os type="other">
  ...
</os>
```
This is contrary to the common practice in the API. It has been replaced with an inner element:

```xml
<os>
  <type>other</type>
  ...
</os>
```

### B.27. REMOVED THE **FORCE** PARAMETER FROM THE REQUEST TO RETRIEVE A HOST

The request to retrieve a host used to support a *force* matrix parameter to indicate that the data of the host should be refreshed (calling VDSM to reload host capabilities and devices) before retrieving it from the database:

```
GET /ovirt-engine/api/hosts/123;force
```

This *force* parameter has been superseded by the host *refresh* action, but kept for backwards compatibility. It has been completely removed now. Applications that require this functionality should perform two requests, first one to refresh the host:

```
POST /ovirt-engine/api/hosts/123/refresh

<action/>
```

And then one to retrieve it, without the *force* parameter:

```
GET /ovirt-engine/api/hosts/123
```

### B.28. REMOVED DEPRECATED HOST POWER MANAGEMENT CONFIGURATION

The host power management configuration used to be part of the host resource, using embedded configuration elements:

```xml
<power_management type="apc">
  <enabled>true</enabled>
  <address>myaddress</address>
  <username>myaddress</username>
  <options>
    <option name="port" value="22/>
    <option name="slot" value="5/>
  </options>
  ...
</power_management>
```

This has been changed some time ago, in order to support multiple power management agents, introducing a new `/hosts/123/fenceagents` collection.

The old *type* attribute, the old *address, username* and *password* elements, and the inner *agents* element directly inside *power_management* were preserved for backwards compatibility. All these elements have been completely removed, so the only way to query or modify the power management...
agents is now the /hosts/123/fenceagents sub-collection.

B.29. USE MULTIPLE boot.devices.device INSTEAD OF MULTIPLE boot

In the past the way to specify the boot sequence when starting a virtual machine was to use multiple boot elements, each containing a dev element. For example, to specify that the virtual machine should first try to boot from CDROM and then from hard disk the following request was used:

```
POST /ovirt-engine/api/vms/123/start

<action>
  <vm>
    ...
    <boot>
      <dev>cdrom</dev>
    </boot>
    <boot>
      <dev>hd</dev>
    </boot>
  </vm>
</action>
```

The common practice in other parts of the API is to represent arrays with a wrapper element. In that case that wrapper element could be named boots, but that doesn’t make much sense, as what can have multiple values here is the boot device, not the boot sequence. To fix this inconsistency this has been replaced with a single boot element that can contain multiple devices:

```
POST /ovirt-engine/api/vms/123/start

<action>
  <vm>
    ...
    <boot>
      <devices>
        <device>cdrom</device>
        <device>hd</device>
      </devices>
    </boot>
  </vm>
</action>
```

B.30. REMOVED THE disks.clone AND disks.detach_only ELEMENTS

These elements aren’t really part of the representation of disks, but parameters of the operations to add and remove virtual machines.

The disks.clone element was used to indicate that the disks of a new virtual machine have to be cloned:

```
POST /ovirt-engine/api/vms

<vm>
```
This has been now removed, and replaced by a new `clone` query parameter:

```
POST /ovirt-engine/api/vms?clone=true
```

The `disks.detach_only` element was used to indicate that when removing a virtual machine the disks don't have to be removed, but just detached from the virtual machine:

```
DELETE /ovirt-engine/api/vms/123
```

This has been now removed, and replaced by a new `detach_only` query parameter:

```
DELETE /ovirt-engine/api/vms/123?detach_only=true
```

### B.31. RENAME ELEMENT VMPOOL TO VM_POOL

The names of the elements that represent pools of virtual machines used to be `vmpool` and `vmpools`. They have been renamed to `vm_pool` and `vm_pools` in order to have a consistent correspondence between names of complex types (`VmPool` and `VmPools` in this case) and elements.

### B.32. USE LOGICAL_UNITS INSTEAD OF MULTIPLE LOGICAL_UNIT

The logical units that are part of a volume group used to be reported as an unbounded number of `logical_unit` elements. For example, when reporting the details of a storage domain:
This is contrary to the usual practice in the API, as list of elements are always wrapped with an element. This has been fixed now, so the list of logical units will be wrapped with the `logical_units` element:

```
GET /ovirt-engine/api/storagedomains/123
```

```
<storage_domain>
 ...
 <storage>
  ...
  <volume_group>
   <logical_units>
    <logical_unit>
      <!-- First LU -->
    </logical_unit>
    <logical_unit>
      <!-- Second LU -->
    </logical_unit>
    ...
   </logical_units>
  </volume_group>
 </storage>
</storage_domain>
```

**B.33. REMOVED THE `snapshots.collapse_snapshots` ELEMENT**

This element isn't really part of the representation of snapshots, but a parameter of the operation that imports a virtual machine from an export storage domain:

```
POST /ovirt-engine/api/storagedomains/123/vms/456/import
```

```
<action>
 <vm>
  <snapshots>
   <collapse_snapshots>true</collapse_snapshots>
  </snapshots>
 </vm>
</action>
```

This has been now removed, and replaced by a new `collapse_snapshots` query parameter:

```
POST /ovirt-engine/api/storagedomains/123/vms/456/import?
collapse_snapshots=true
```
B.34. RENAMED **STORAGE** AND **HOST_STORAGE** ELEMENTS

The host storage collection used the `storage` and `host_storage` elements and the `Storage` and `HostStorage` complex types to report the storage associated to a host:

```xml
GET /ovirt-engine/api/hosts/123/storage
```

```xml
<host_storage>
  <storage>...
  </storage>
  ...
  <storage>...
  </storage>
  ...
</host_storage>
```

This doesn’t follow the pattern used in the rest of the API, where the outer element is a plural name and the inner element is the same name but in singular. This has now been changed to use `host_storages` as the outer element and `host_storage` as the inner element:

```xml
GET /ovirt-engine/api/hosts/123/storage
```

```xml
<host_storages>
  <host_storage>...
  ...
  <host_storage>...
  ...
</host_storage>
```

B.35. REMOVED THE `PERMISSIONS.CLONE` ELEMENT

This element isn’t really part of the representation of permissions, but a parameter of the operations to create virtual machines or templates:

```xml
POST /ovirt-engine/api/vms
```

```xml
<vm>
  <template id="...">
    <permissions>
      <clone>true</clone>
    </permissions>
  </template>
</vm>
```
POST /ovirt-engine/api/templates

This has been now removed, and replaced by a new `clone_permissions` query parameter:

POST /ovirt-engine/api/vms?clone_permissions=true

POST /ovirt-engine/api/templates?clone_permissions=true

B.36. RENAMED THE RANDOM NUMBER GENERATOR SOURCE ELEMENTS

The random number generator sources used to be reported using a collection of `source` elements wrapped by an element with a name reflecting its use. For example, the required random number generator sources of a cluster used to be reported as follows:

GET /ovirt-engine/api/clusters/123

```
<cluster>
  ...
  <required_rng_sources>
    <source>random</source>
  </required_rng_sources>
  ...
</cluster>
```

And the random number generator sources suported by a host used to be reported as follows:

GET /ovirt-engine/api/hosts/123

```
<host>
  ...
  <hardware_information>
    <supported_rng_sources>
      <source>random</source>
    </supported_rng_sources>
  </hardware_information>
```
This isn’t consistent with the rest of the API, where collections are wrapped by a name in plural and elements by the same name in singular. This has been now fixed. The required random number generator sources will now be reported as follows:

```xml
GET /ovirt-engine/api/clusters/123

<cluster>
  <required_rng_sources>
    <required_rng_sources>random</required_rng_source>
  </required_rng_sources>
  ...
</cluster>
```

And the random number generator sources supported by a host will be reported as follows:

```xml
GET /ovirt-engine/api/hosts/123

<host>
  ...
  <hardware_information>
    <supported_rng_sources>
      <supported_rng_source>random</supported_rng_source>
    </supported_rng_sources>
  </hardware_information>
  ...
</host>
```

Note the use of `required_rng_source` and `supported_rng_source` instead of just `source`.

**B.37. REMOVED THE INTERMEDIATE `TAG.PARENT` ELEMENT**

The relationship between a tag and its parent tag used to be represented using an intermedite `parent` tag, that in turn contains another `tag` element:

```xml
<tag>
  <name>mytag</name>
  <parent>
    <tag_id="..." href="..."/>
  </parent>
</tag>
```

This structure has been simplified so that only one `parent` element is used now:

```xml
<tag>
  <name>mytag</name>
  <parent id="..." href="..."/>
</tag>
```
B.38. REMOVE SCHEDULING BUILT-IN NAMES AND THRESHOLDS

In the past the specification of scheduling policies for clusters was based in built-in names and thresholds. For example a cluster that used the evenly distributed scheduling policy was represented as follows:

```xml
<cluster>
  <name>mycluster</name>
  <scheduling_policy>
    <policy>evenly_distributed</policy>
    <thresholds high="80" duration="120"/>
  </scheduling_policy>
  ...
</cluster>
```

This mechanism was replaced with a top level /schedulingpolicies collection where scheduling policies can be defined with arbitrary names and properties. For example, the same scheduling policy is represented as follows in that top level collection:

```xml
<scheduling_policy>
  <name>evenly_distributed</name>
  <properties>
    <property>
      <name>CpuOverCommitDurationMinutes</name>
      <value>2</value>
    </property>
    <property>
      <name>HighUtilization</name>
      <value>80</value>
    </property>
  </properties>
</scheduling_policy>
```

The representation of the cluster references the scheduling policy with its identifier:

```xml
<cluster>
  <name>mycluster</name>
  <scheduling_policy id="..."/>
  ...
</cluster>
```

To preserve backwards compatibility the old policy and thresholds elements were preserved. The scheduling policy representation embedded within the cluster was also preserved. All these things have been completely removed now, so the only way to reference a scheduling policy when retrieving, creating or updating a cluster is to reference an existing one using its identifier. For example, when retrieving a cluster only the id (and href) will be populated:

```xml
GET /ovirt-engine/api/clusters/123

<cluster>
  ...
  <scheduling_policy id="..." href="/...">
    ...
  </scheduling_policy>
  ...
</cluster>
```
When creating or updating a cluster only the id will be accepted.

B.39. REMOVED THE **bricks.replica_count** AND **bricks.stripe_count** ELEMENTS

These elements aren’t really part of the representation of a collection of bricks, but parameters of the operations to add and remove bricks. They have now been removed, and replaced by new **replica_count** and **stripe_count** parameters:

```
POST .../bricks?replica_count=3&stripe_count=2
DELETE .../bricks?replica_count=3
```

B.40. RENAMED THE STATISTICS **type** PROPERTY TO **kind**

The statistics used to be represented using a **type** element that indicates the kind of statistic (gauge, counter, etc) and also a **type** attribute that indicates the type of the values (integer, string, etc):

```
<statistic>
  <type>GUAGE</type>
  <values type="INTEGER">
    <value>...</value>
    <value>...</value>
    ...
  </values>
</statistic>
```

To avoid the use of the **type** concept for both things the first has been replaced by **kind**, and both **kind** and **type** are now elements:

```
<statistic>
  <kind>gauge</kind>
  <type>integer</type>
  <values>
    <value>...</value>
    <value>...</value>
    ...
  </values>
</statistic>
```

B.41. USE MULTIPLE **vcpu_pins.vcpu_pin** INSTEAD OF MULTIPLE **vcpu_pin**

In the past the way to specify the virtual to physical CPU pinning of a virtual machine was to use multiple **vcpu_pin** elements:

```
<vm>
  <cpu>
    <cpu_tune>
      <vcpu_pin>...</vcpu_pin>
      <vcpu_pin>...</vcpu_pin>
```
In order to conform to the common practice in other parts of the API this has been changed to use a wrapper element, in this case `vcpu_pins`:

```xml
<vm>
  <cpu>
    <cpu_tune>
      <vcpu_pins>
        <vcpu_pin>...</vcpu_pin>
        <vcpu_pin>...</vcpu_pin>
        ...
      </vcpu_pins>
    </cpu_tune>
  </cpu>
</vm>
```

B.42. USE `force` PARAMETER TO FORCE REMOVE A DATA CENTER

The operation that removes a data center supports a `force` parameter. In order to use it the `DELETE` operation used to support an optional action parameter:

```bash
DELETE /ovirt-engine/api/datacenters/123
```

```xml
<action>
  <force>true</force>
</action>
```

This optional action parameter has been replaced with an optional parameter:

```bash
DELETE /ovirt-engine/api/datacenters/123?force=true
```

B.43. USE `force` PARAMETER TO FORCE REMOVE A HOST

The operation that removes a host supports a `force` parameter. In order to use it the `DELETE` operation used to support an optional action parameter:

```bash
DELETE /ovirt-engine/api/host/123
```

```xml
<action>
  <force>true</force>
</action>
```

This optional action parameter has been replaced with an optional parameter:

```bash
DELETE /ovirt-engine/api/host/123?force=true
```
B.44. USE PARAMETERS FOR FORCE REMOVE STORAGE DOMAIN

The operation that removes a storage domain supports the force, destroy and host parameters. These parameters were passed to the DELETE method using the representation of the storage domain as the body:

```
DELETE /ovirt-engine/api/storagedomains/123

<storage_domain>
  <force>...</force>
  <destroy>...</destroy>
  <host id="...">
    <name></name>
  </host>
</storage_domain>
```

This was problematic, as the HTTP DELETE parameters shouldn’t have a body, and the representation of the storage domain shouldn’t include things that aren’t attributes of the storage domain, rather parameters of the operation.

The force, delete and host attributes have been replaced by equivalent parameters, and the operation doesn’t now accept a body. For example, now the correct way to delete a storage domain with the force parameter is the following:

```
DELETE /ovirt-engine/api/storagedomain/123?host=myhost&force=true
```

To delete with the destroy parameter:

```
DELETE /ovirt-engine/api/storagedomain/123?host=myhost&destroy=true
```

B.45. USE HOST PARAMETER TO REMOVE STORAGE SERVER CONNECTION

The operation that removes a storage server connection supports a host parameter. In order to use it the DELETE method used to support an optional action parameter:

```
DELETE /ovirt-engine/api/storageconnections/123

<action>
  <host id="...">
    <name></name>
  </host>
</action>
```

This optional action parameter has been replaced with an optional parameter:

```
DELETE /ovirt-engine/api/storageconnections/123?host=myhost
```
B.46. USE **force** AND **storage_domain** PARAMETERS TO REMOVE TEMPLATE DISKS

The operation that removes a template disk supports the **force** and **storage_domain** parameters. In order to use them the **DELETE** method used to support an optional action parameter:

```
DELETE /ovirt-engine/api/templates/123/disks/456
```

In version 4 of the API this operation has been moved to the new **diskattachments** collection, and the request body has been replaced with the query parameters **force** and **storage_domain**:

```
DELETE /ovirt-engine/api/templates/123/disksattachments/456?force=true
DELETE /ovirt-engine/api/templates/123/disksattachments/456?
storage_domain=123
```

B.47. DON’T REMOVE DISKS VIA THE VM DISK API

Removing an entity by deleting **/vms/123/disks/456** means removing the relationship between the VM and the disk - i.e., this operation should just detach the disk from the VM. This operation is no longer able to remove disks completely from the system, which was prone to user errors and had irreversible consequences. To remove a disk, instead use the **/disk/456** API:

```
DELETE /ovirt-engine/api/disks/456
```

B.48. USE **force** QUERY PARAMETER TO FORCE REMOVE A VIRTUAL MACHINE

The operation that removes a virtual machine supports a **force** parameter. In order to use it the **DELETE** method used to support an optional action parameter:

```
DELETE /ovirt-engine/api/vms/123
```

```
?action
 <force>true</force>
 </action>
```

This optional action parameter has been replaced with an optional query parameter:

```
DELETE /ovirt-engine/api/vms/123?force=true
```

B.49. USE **post** INSTEAD OF **delete** TO REMOVE MULTIPLE BRICKS
The operation that removes multiple Gluster bricks was implemented using the `DELETE` method and passing the list of bricks as the body of the request:

```plaintext
DELETE /ovirt-engine/api/clusters/123/glustervolumes/456/bricks
```

This is problematic because the `DELETE` method shouldn’t have a body, so it has been replaced with a new `remove` action that uses the `POST` method:

```plaintext
POST /ovirt-engine/api/clusters/123/glustervolumes/456/bricks/remove
```

---

**B.50. REMOVED THE `SCHEDULING_POLICY.POLICY` ELEMENT**

The element was kept for backward compatibility. Use `scheduling_policy.name` instead.

```plaintext
POST /ovirt-engine/api/schedulingpolicies
```

```xml
<scheduling_policy>
  ...
  <name>policy_name</name>
  ...
</scheduling_policy>
```

```plaintext
PUT /ovirt-engine/api/schedulingpolicies/123
```

```xml
<scheduling_policy>
  ...
  <name>policy_name</name>
  ...
</scheduling_policy>
```

---

**B.51. ADDED `SNAPSHOT.SNAPSHOT_TYPE`**

 Enums are being gradually introduced to the API. Some fields which were string until now, are replaced with an appropriate enum. One such field is `vm.type`. But this field is inherited by snapshot, and snapshot type is different than `vm` type. So a new field has been added to snapshot entity: `snapshot.snapshot_type`.

```xml
<n snapshot>
```
B.52. REMOVED MOVE ACTION FROM VM

The deprecated move action of the VM entity has been removed. Instead, you can move individual disks.

B.53. MOVED reported_configurations.in_sync TO network_attachment

In version 3 of the API the XML schema type ReportedConfigurations had a in_sync property:

```
<network_attachment>
  <reported_configurations>
    <in_sync>true</in_sync>
    <reported_configuration>
      ...
    </reported_configuration>
    ...
  </reported_configurations>
</network_attachment>
```

In the specification mechanism used by version 4 of the API this can't be expressed, because list types (the list of reported configurations) can't have attributes. To be able to represent it the attribute has been moved to the enclosing network_attachment:

```
<network_attachment>
  <in_sync>true</in_sync>
  <reported_configurations>
    <reported_configuration>
      ...
    </reported_configuration>
    ...
  </reported_configurations>
</network_attachment>
```

B.54. REPLACED capabilities WITH clusterlevels

The top level capabilities collection has been replaced by the new clusterlevels collection. This new collection will contain the information that isn't available in the model, like the list of CPU types available for each cluster level:

```
GET /ovirt-engine/api/clusterlevels
```

This will return a list of ClusterLevel objects containing the details for all the cluster levels supported by the system:

```
<cluster_levels>
  <cluster_level id="3.6" href="/clusterlevels/3.6">
    <cpu_types>
      <cpu_type>
```

...
Each specific cluster level has its own subresource, identified by the version itself:

```
GET /ovirt-engine/api/clusterlevels/3.6
```

This will return the details of that version:

```
<cluster_level id="3.6" href="/clusterlevels/3.6">
  <cpu_types>
    <cpu_type>
      <name>Intel Conroe Family</name>
      <level>2</level>
      <architecture>x86_64</architecture>
    </cpu_type>
    ...
  </cpu_types>
  ...
</cluster_level>
```

**B.55. REPLACED DISKS WITH DISKATTACHMENTS**

In version 3 of the API virtual machines and templates had a **disks** collection containing all the information of the disks attached to them. In version 4 of the API these **disks** collections have been removed and replaced with a new **diskattachments** collection that will contain only the references to the disk and the attributes that are specific of the relationship between disks and the virtual machine or template that they are attached to: **interface** and **bootable**.

To find what disks are attached to a virtual machine, for example, send a request like this:

```
GET /ovirt-engine/api/vms/123/diskattachments
```

That will return a response like this:

```
<disk_attachments>
  <disk_attachment href="/vms/123/diskattachments/456" id="456">
    <bootable>false</bootable>
    <interface>virtio</interface>
    <disk href="/disks/456" id="456"/>
    <vm href="/vms/123" id="123"/>
  </disk_attachment>
  ...
</disk_attachments>
```

To find the rest of the details of the disk, follow the link provided.
Adding disks to a virtual machine or template uses the new `disk_attachment` element as well: request like this:

```
POST /ovirt-engine/api/vms/123/diskattachments
```

With the following body if the disk doesn’t exist and you want to create it:

```
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <disk>
    <description>My disk</description>
    <format>cow</format>
    <name>mydisk</name>
    <provisioned_size>1048576</provisioned_size>
    <storage_domains>
      <storage_domain>
        <name>mydata</name>
      </storage_domain>
    </storage_domains>
  </disk>
</disk_attachment>
```

Or with the following body if the disk already exists, and you just want to attach it to the virtual machine:

```
<disk_attachment>
  <bootable>false</bootable>
  <interface>virtio</interface>
  <disk id="456"/>
</disk_attachment>
```

Take into account that the `vm.disks` and `template.disks` attribtes have `disk_attachments` for all usages. For example, when creating a template the `vm.disks` element was used to indicate in which storage domain to create the disks of the template. This usage has also been replaced by `vm.disk_attachments`, so the request to create a template with disks in specific storage domains will now look like this:

```
<template>
  <name>mytemplate</name>
  <vm id="123">
    <disk_attachments>
      <disk_attachment>
        <disk id="456">
          <storage_domains>
            <storage_domain id="789"></storage_domain>
          </storage_domains>
        </disk>
      </disk_attachment>
      ...
    </disk_attachments>
  </vm>
</template>
```
B.56. USE iscsi_targets ELEMENT TO DISCOVER UNREGISTERED STORAGE

In version 3 of the API the operation to discover unregistered storage domains used to receive a list of iSCSI targets, using multiple iscsi_target elements:

```
POST /ovirt-engine/api/hosts/123/unregisteredstoragedomaindiscover

<action>
  <iscsi>
    <address>myiscsiserver</address>
  </iscsi>
  <iscsi_target>iqn.2016-07.com.example:mytarget1</iscsi_target>
  <iscsi_target>iqn.2016-07.com.example:mytarget2</iscsi_target>
</action>
```

In version 4 of the API all repeating elements, like iscsi_target in this case, are wrapped with another element, iscsi_targets in case. So the same request should now look like this:

```
POST /ovirt-engine/api/hosts/123/unregisteredstoragedomaindiscover

<action>
  <iscsi>
    <address>myiscsiserver</address>
  </iscsi>
  <iscsi_targets>
    <iscsi_target>iqn.2016-07.com.example:mytarget1</iscsi_target>
    <iscsi_target>iqn.2016-07.com.example:mytarget2</iscsi_target>
  </iscsi_targets>
</action>
```