



Red Hat Virtualization

4.1

RHEVM Shell 指南

安装并使用 Red Hat Virtualization 命令行 shell

Red Hat Virtualization 文档团队

安装并使用 Red Hat Virtualization 命令行 shell

Red Hat Virtualization 文档团队

Red Hat 出版部

rhev-docs@redhat.com

法律通告

Copyright © 2016 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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.

摘要

本指南包含安装及使用 Red Hat Virtualization Manager 命令行 Shell 的信息。

目录

第 1 章 使用 CLI	2
1.1. 安装 CLI	2
1.2. TLS/SSL 认证	2
1.3. .ovirtshellrc 配置	4
1.4. 运行 CLI	4
1.5. 使用 CLI	6
1.6. 集合	9
第 2 章 快速入门实例	10
2.1. 使用 CLI 创建一个基本的虚拟化环境	10
第 3 章 命令	15
3.1. 连接到 RHEVM	15
3.2. 资源	16
3.3. 其它命令	20
第 4 章 资源类型	27
4.1. brick	27
4.2. cdrom	27
4.3. cluster	28
4.4. datacenter	30
4.5. disk	31
4.6. glustervolume	32
4.7. group	34
4.8. host	34
4.9. network	36
4.10. nic	38
4.11. permission	40
4.12. permit	41
4.13. quotas	42
4.14. role	42
4.15. snapshot	42
4.16. statistic	43
4.17. storageconnection	44
4.18. storagedomain	45
4.19. tag	48
4.20. template	48
4.21. user	53
4.22. vm	53
4.23. vmpool	59
4.24. vnicprofile	60
第 5 章 CLI 查询	62
5.1. 查询语法	62
5.2. 通配符	62

第 1 章 使用 CLI

Red Hat Virtualization 包括了一个命令行接口（CLI - Command Line Interface），用户可以在不使用标准网站接口的情况下，使用 CLI 连接到 Red Hat Virtualization Manager。CLI 还包括了一个脚本运行系统，系统管理员可以使用脚本来从客户端上直接执行一些需要定期执行的系统维护操作。



重要

Red Hat Virtualization Manager 版本 4 包括了 Red Hat Enterprise Linux 7 的 CLI 版本 3.6。这个版本的 CLI 使用 Python SDK 版本 3.6 和 REST API 版本 3 的兼容模式。

1.1. 安装 CLI

在客户端上安装 Red Hat Virtualization CLI：

1. 以 **root** 用户身份登录到客户端系统。
2. 在 Content Delivery Network 中注册您的系统，提示时输入您的客户门户网站（Customer Portal）的用户名和密码：

```
# subscription-manager register
```

3. 找到 **Red Hat Virtualization** 订阅池并记录下池 ID。

```
# subscription-manager list --available
```

4. 使用上一步中找到的池标识符在该系统中添加 **Red Hat Virtualization** 权利。

```
# subscription-manager attach --pool=pool_id
```

5. 启用所需存储库：

```
# subscription-manager repos --enable=rhel-7-server-rhv-4.1-rpms
```

6. 安装 CLI 软件包和依赖的软件包：

```
# yum install ovirt-engine-cli
```

1.2. TLS/SSL 认证

Red Hat Virtualization Manager API 需要使用 Hypertext Transfer Protocol Secure (HTTPS) ^[1] 来和客户端软件（如 Manager 的 SDK 和 CLI）进行安全的通信。这需从 Red Hat Virtualization Manager 中获得一个证书，并把它导入到客户端系统的证书存储系统中。



重要

请使用一个安全的网络来从 Red Hat Virtualization Manager 获得您的证书。

过程 1.1. 获得一个证书

可以使用以下 3 种方法中的一个来从 Red Hat Virtualization Manager 获得证书，并把证书传输到客户端系统上：

1. **方法 1** - 使用一个命令行工具从 Manager 上下载证书。例如使用 **cURL** 或 **Wget**。多个平台都提供了这两个工具。

- a. 使用 **cURL**：

```
$ curl -o rhvm.cer http://[manager-fqdn]/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA
```

- b. 使用 **Wget**：

```
$ wget -O rhvm.cer http://[manager-fqdn]/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA
```

2. **方法 2** - 使用一个网络浏览器来访问提供证书的位置：

```
http://[manager-fqdn]/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA
```

根据您所使用的浏览器，证书可能会被下载，也可能被直接导入到浏览器的密钥库（keystore）中。

- a. 如果浏览器需要下载这个证书：把文件保存为 **rhvm.cer**。

如果浏览器导入了这个证书：把它从浏览器的证书选项中导出，并保存为 **rhvm.cer**。

3. **方法 3** - 登录到 Manager 所在的系统，把证书从 truststore 中导出，并复制到您的客户端系统上。

- a. 以 **root** 用户身份登录到 Manager 所在的机器上。
- b. 使用 Java **keytool** 工具把证书从 truststore 中导出：

```
$ keytool -exportcert -keystore /etc/pki/ovirt-engine/.truststore -alias cacert -storepass mypass -file rhvm.cer
```

它会创建一个名为 **rhvm.cer** 的证书文件。

- c. 使用 **scp** 命令把证书复制到客户端系统上：

```
$ scp rhvm.cer [username]@[client-machine]:[directory]
```

以上 3 个方法都会您的客户端系统上创建一个名为 **rhvm.cer** 的证书文件。API 用户可以把这个文件导入到客户端系统的证书库中。

过程 1.2. 把证书导入到一个客户端系统上

- ✱ 把证书导入到客户端系统的方法取决于客户端系统如何保存和解析证书。本指南包括了一些导入证书的示例。如果没有使用 Network Security Services (NSS) 或 Java KeyStore (JKS) 的客户端系统，请参阅相关文档来了解如何导入证书的方法。

1.3. .ovirtshellrc 配置

.ovirtshellrc 是用户首次连接到 **ovirt-shell** 时自动创建和填充的配置文件。它允许用户配置用来连接到 Red Hat Virtualization 环境的选项。默认情况下，**.ovirtshellrc** 文件位于 **/home/[username]/.ovirtshellrc**。

.ovirtshellrc 文件配置信息包括在两个部分中，即 **[cli]** 和 **[ovirt-shell]**。解析该配置文件需要这两个部分。

表 1.1. [cli] 参数

名称	类型	描述
autoconnect	布尔值	是否自动连接到 ovirt-shell 部分。该状态可以是 True 或者 False 。
autopage	布尔值	指定在 shell 中是否分页。它的值是 True 或 False 。

表 1.2. [ovirt-shell] 参数

名称	类型	描述
username	字符串	登录用户的用户名。
timeout	整数	指定请求超时的时间。默认值为 -1。
extended_prompt	布尔值	切换扩展的提示选项，使用该选项可在 shell 命令提示符后显示主机名。
url	字符串	Red Hat Virtualization 环境的地址。
insecure	布尔值	指定是否需要 CA 证书。它的值是 True 或 False 。
renew_session	布尔值	会话过期后是否自动续订。它的值可以是 True 或者 False 。
filter	布尔值	指定是否需要对象过滤。对象过滤允许用户根据对象的权限获取相关的对象。只有具有 admin 角色的用户才可以设置这个选项。它的值是 True 或 False 。
session_timeout	整数	以分钟为单位指定认证会话的超时时间。必须是一个正数。
ca_file	字符串	指定要使用的 CA 证书。
dont_validate_certificate_chain	布尔值	是否验证服务器 CA 证书。它的值可以是 True 或 False 。
key_file	字符串	指定客户端 PEM key-file。
password	字符串	登录用户所使用的密码。
cert_file	字符串	指定客户端 PEM cert-file。

1.4. 运行 CLI

使用以下命令启动 CLI 应用程序：

```
# ovirt-shell
```

ovirt-shell 应用程序是一个用来和 Red Hat Virtualization 环境进行交流的、交换式的 shell。

用来和 Red Hat Virtualization Manager 进行连接的 URL、用户名、证书颁发机构文件和密码可以在 **.ovirtshellrc** 文件中进行配置。**ovirt-shell** 命令使用这个文件中的参数来与 Manager 进行连接，因此用户无需每次都指定这些选项。

另外，用户也可以使用以下选项来自动连接到 Red Hat Virtualization Manager：

```
# ovirt-shell -c -l "https://[server]/ovirt-engine/api" -P [port] -u
"[user@domain]" -A "[certificate]"
```

包括在方括号中的值需要根据您的实际情况进行替换：

- ✧ **server** - Red Hat Virtualization Manager 的主机名或 IP 地址。CLI 使用 REST API 连接到 Red Hat Virtualization Manager。
- ✧ **user@domain** - 登录到 Red Hat Virtualization Manager 的用户的用户名和所在的目录服务域名。
- ✧ **certificate** - 证书授权文件的路径名。

shell 会提示您输入密码。如果没有提供用户名和 Red Hat Virtualization Manager 的 URL，它还会要求您输入它们。



注意

如果您已经在 **.ovirtshellrc** 文件中配置了用户名、密码、URL 和证书颁发文件，您将不需要指定额外的选项。



注意

以上示例中的选项只有“certificate”是必须指定的，其它的选项都可以在 shell 的提示下输入。您可以使用 ‘--insecure’ 选项而不再需要指定“certificate”，这将可以在不使用证书的情况下进行连接，但是，这会使“中间人（man-in-the-middle - MITM）”袭击有机可乘，因此我们不推荐使用这个方法。

ovirt-shell 的选项

-h, --help

显示 **ovirt-shell** 的帮助信息。

-d, --debug

启用调试模式

-l URL, --url=URL

指定 API 进入点 URL。

-u USERNAME, --username=USERNAME

以指定的用户帐号连接。

-K KEY_FILE, --key-file=KEY_FILE

指定密钥文件。

-C CERT_FILE, --cert-file= CERT_FILE

指定证书文件。

-A CA_FILE, --ca-file= CA_FILE

指定服务器证书授权文件。

-I, --insecure

允许 CLI 不使用证书来通过 SSL 进行连接。因为这个方法可能会遭到“中间人（man-in-the-middle - MITM）”袭击，因此在使用时应格外小心。

-F, --filter

根据用户的权限启用过滤功能。

-P PORT, --port= PORT

指定端口。

-T TIMEOUT, --timeout= TIMEOUT

指定超时时间。

-c, --connect

自动连接。

-e, --extended-prompt

启用扩展的 shell 提示符。使用这个选项后，shell 的提示符将包括环境的主机名。它的默认值是 'false'。

-E "command resource", --execute-command= "command resource"

连接到 Manager 后只执行指定的命令（指定命令的格式是 "command resource;command resource"），并把命令的输出打印到 STDIO。

-f FILE, --file= FILE

从指定的文件而不是 stdin 中读取命令。

--kerberos

使用有效 Kerberos ticket 验证与 shell 的连接。



注意

使用非交互式 shell 的用户可以通过在 shell 中使用 --password 选项来与 Red Hat Virtualization Manager 进行连接。

1.5. 使用 CLI

CLI 是一个交互式的 shell，您可以使用它来通过命令行对 Red Hat Virtualization 环境进行控制。您可以在 shell 中输入所需的命令和选项。

例 1.1. 输入一个 shell 命令

```
[RHEVM shell (connected)]# show vm RHEL6-Server
```

CLI 包括了“列出和自动完成命令和参数”的功能，连续按 **TAB** 键两次可以根据您已经输入的内容自动列出并完成剩下的输入。这个功能和 **bash** shell 的功能相似。

例 1.2. 列出并自动完成命令和参数

在还没有在提示符后面输入任何内容时，双击 **TAB** 键会列出所有可用的命令。

```
[RHEVM shell (connected)]# TAB TAB
EOF          clear          echo          history      remove
summary
action       connect       exit          info         shell
update
add          console      file          list         show
capabilities disconnect    help          ping         status
```

选择一个命令后双击 **TAB** 键会列出这个命令可用的参数。假设选择了 **add** 命令，它会列出所有资源。

```
[RHEVM shell (connected)]# add TAB TAB
affinitygroup  datacenter  event        group        nic
quota          label       template     vmppool      cdrom
disk           filter      host         permission   role
storagedomain  user        cluster      qos          glustervolume
network        permit     snapshot     tag          vm
```

双击 **TAB** 键会完成命令和参数输入。

```
[RHEVM shell (connected)]# add vm TAB TAB
comment        console-enabled  cpu-architecture  delete_protected
description     disks-clone      display-type       io-threads
memory         name             os-type            rng_device-source
stateless       soundcard_enabled  start_paused       timezone
[RHEVM shell (connected)]# add vm naTAB TAB
[RHEVM shell (connected)]# add vm --name
```

双击 **TAB** 键可以自动把 **na** 变为 **--name** 参数（包括 '--' 前缀）。

如果未完成的参数和多个参数匹配，双击 **TAB** 将会列出所有匹配的参数。

```
[RHEVM shell (connected)]# add vTAB TAB
vmppool        vm
```

CLI 提供了使用 **shell** 命令或 **!** 来运行 Linux 命令的功能。

例 1.3. 运行 Linux shell 命令

使用 **shell** 命令：

```
[RHEVM shell (connected)]# shell ls -la
```

或使用 ! :

```
[RHEVM shell (connected)]# !ls -la
```

和 Linux shell 相似，CLI 可以使用管道（pipe）把数据传递给其它命令或资源。

例 1.4. 在 CLI 中使用管道（pipe）

把 CLI 数据传递给 Linux shell 命令：

```
[RHEVM shell (connected)]# list vms --show-all | grep "Example"
name           : Example1
name           : Example2
name           : ExampleEngineering
description    : An Example description
name           : BestExampleVM
```

把 CLI 数据传递给一个文件：

```
[RHEVM shell (connected)]# list vms --show-all > list vms --show-all >
VM_List.txt
```

CLI 还包括了在线帮助信息，使用 **help** 命令可以显示相关命令的介绍和使用语法。

例 1.5. 使用 show 命令的在线帮助信息

```
[RHEVM shell (connected)]# help show
```

您也可以在 Linux shell 连接到 Manager 后执行特定的命令（命令格式是 "*command resource*"），并把命令输出打印到 STDIO

例 1.6. 连接到 Manager 来执行特定的命令

使用 **--execute** 或 **-E** 参数连接到 Manager 来执行特定命令。

```
# ovirt-shell -c -l "https://[server]/ovirt-engine/api" -P [port] -u
"[user@domain]" -A "[certificate]" -E "list vms;list hosts"
[RHEVM shell (connected)]# list vms

id           : 9e6977f4-4351-4feb-bba0-dc7c22adec30
name         : desktop-01

id           : 60b12e28-7965-4296-86bf-c991aa32c2d5
name         : server-01

[RHEVM shell (connected)]# list hosts
```

```
id      : 3598cdb9-d21b-49bd-9491-59fa9b89b113
name    : Gluster

id      : a0c384f9-0940-4562-9c42-4ceaadf8f1f1
name    : Host-01

id      : 593ec966-c3ea-4bdc-84ad-5dc3f9fe64c7
name    : Host-03
```

1.6. 集合

一些命令参数需要一个集合（collection）。集合就是一组子参数数据，它以以下的形式定义。

```
[RHEVM shell (connected)]# command --param-collection
{subparam1=value1;subparam2=value2;subparam3=value3;...},
{subparam1=value1;subparam2=value2;subparam3=value3;...},...
```

在每个资源页中，集合的子参数会在资源参数后列出。

[1] [RFC 2818 HTTP Over TLS](#) 包括了 HTTPS 的信息。

第 2 章 快速入门实例

2.1. 使用 CLI 创建一个基本的虚拟化环境

本章提供了一个使用 CLI 把虚拟机加入到 Red Hat Virtualization 环境中的示例。这个示例需要具备以下先决条件：

- ✧ 已被配置并连接到网络中的、作为 hypervisor 使用的 Red Hat Enterprise Linux 主机
- ✧ 带有以下两个共享的、已经被配置并连接到网络中的 NFS 存储服务器：
 - `/exports/data` - 数据存储域
 - `/exports/iso` - ISO 存储域
- ✧ 一个已经被配置并连接到网络中的 Red Hat Virtualization Manager
- ✧ 在 Red Hat Virtualization Manager 或客户端系统上已经安装了 CLI
- ✧ 包括了需要安装的虚拟机操作系统的 ISO 文件。在这个示例中我们使用 Red Hat Enterprise Linux Server 6 作为虚拟机的操作系统。



注意

Red Hat Virtualization Manager 会为每个资源生成全局唯一标识符（GUID）。这个示例中使用的标识符代码可能会与您在 Red Hat Virtualization 环境中使用的标识符代码不同。

过程 2.1. 快速入门示例

1. 启动 CLI shell 并连接到 Red Hat Virtualization Manager。

```
# ovirt-shell -c --url https://[rhev-host]/ovirt-engine/api --
username [user]@[domain] --ca-file certificate/authority/path/name
```

2. 列出该环境中的所有数据中心。这个示例使用 **Default** 数据中心。

```
[RHEVM shell (connected)]# list datacenters

id           : 5e3b55d8-c585-11e1-a7df-001a4a400e0d
name         : Default
description: The default Data Center
```

3. 列出所有主机集群并记录下相关集群 ID 或集群名称，添加主机或创建虚拟机时需要此信息。这个示例使用 **Default** 集群来对 Red Hat Virtualization 环境中的资源进行分组。

```
[RHEVM shell (connected)]# list clusters

id           : 99408929-82cf-4dc7-a532-9d998063fa95
name         : Default
description: The default server cluster
```

4. 列出所有 CPU 配置及并记录相关 CPU 配置 ID，创建虚拟机时需要此信息。这个示例使用 **Default** CPU 配置。

```
[RHEVM shell (connected)]# list cpuprofiles

id           : 0000001a-001a-001a-001a-000000000035e
name         : Default
```

5. 使用 **show-all** 选项列出所有逻辑网络，查看该环境中的逻辑网络详情。Red Hat Virtualization Manager 创建名为 **ovirtmgmt** 的逻辑网络管理流量。这个示例使用 **Default** 数据中心的 **ovirtmgmt** 逻辑网络。

```
[RHEVM shell (connected)]# list networks --show-all

id           : 00000000-0000-0000-0000-000000000009
name         : ovirtmgmt
description  : Management Network
data_center-id: 5e3b55d8-c585-11e1-a7df-001a4a400e0d
mtu          : 0
required     : True
status-state : operational
stp          : False
usages-usage : VM
```

请注意，**data_center-id** 的值与 **Default** 数据中心的 **id** 相匹配。

6. 在虚拟化环境中添加 Red Hat Enterprise Linux 主机作为新的 hypervisor。该主机会自动激活。

```
[RHEVM shell (connected)]# add host --name MyHost --address
host.example.com --cluster-name Default --root_password p@55w0rd!
```

7. 通过创建、附加并激活 NFS 共享来添加 NFS 共享作为数据存储域。NFS 数据存储域是附加到数据中心的导出 NFS 共享。它为虚拟机提供存储。使用该 NFS 服务器的实际确值替换 **storage-address** 和 **storage-path**。

- a. 创建数据存储域。

```
[RHEVM shell (connected)]# add storagedomain --host-name MyHost
--type data --storage-type nfs --storage_format v3 --storage-
address x.x.x.x --storage-path /exports/data --name DataStorage
```

- b. 验证所创建存储域是否可用。创建过程可能需要几分中。**status-state** 处于 **unattached** 状态后即可进行下一步操作。

```
[RHEVM shell (connected)]# show storagedomain DataStorage

id           : xxxx
name         : DataStorage
master       : False
status-state : unattached
...
```

- c. 将数据存储域附加到数据中心。该存储域会自动激活。

```
[RHEVM shell (connected)] # add storagedomain --datacenter-
identifier Default --name DataStorage
```



注意

如果该存储域未被激活，则可使用下面的命令手动将其激活：

```
[RHEVM shell (connected)]# action storagedomain DataStorage
--datacenter-identifier Default activate
```

8. 通过创建、附加并激活 NFS 共享添加该 NFS 共享作为 ISO 存储域使用。NFS ISO 存储域是一个附加到数据中心的导出 NFS 共享。它为 DVD/CD-ROM ISO 以及虚拟软盘（VFD）镜像文件提供存储。使用 NFS 服务器的实际值替换 **storage-address** 和 **storage-path**。

- a. 创建 ISO 存储域。

```
[RHEVM shell (connected)]# add storagedomain --host-name MyHost
--type iso --storage-type nfs --storage_format v3 --storage-
address x.x.x.x --storage-path /exports/iso --name ISOStorage
```

- b. 验证所创建存储域是否可用。创建过程可能需要几分钟。**status-state** 处于 **unattached** 状态后即可进行下一步操作。

```
[RHEVM shell (connected)]# show storagedomain --name ISOStorage
id                : xxxx
name              : ISOStorage
master            : False
status-state      : unattached
...
```

- c. 将 ISO 存储域附加到数据中心。该存储域会自动激活。

```
[RHEVM shell (connected)] # add storagedomain --datacenter-
identifier Default --name ISOStorage
```

9. 创建新虚拟机。

```
[RHEVM shell (connected)]# add vm --name MyVM --cluster-name Default -
-template-name Blank --memory 536870912 --os-boot boot.dev=hd --
cpu_profile-id 0000001a-001a-001a-001a-000000000035e
```

10. 使用 **add nic** 命令添加新网络接口。其中的 **vm-identifier** 选项将接口附加为 **MyVM** 的子资源；**network-name** 选项将其连接到 **rhev** 网络。

```
[RHEVM shell (connected)]# add nic --vm-identifier MyVM --name nic1 --
network-name ovirtmgmt --bootable true
```

11. 使用 **add disk** 命令来添加一个新虚拟硬盘。其中的 **vm-identifier** 选项把磁盘附加为 **MyVM** 的子资源。


```
[RHEVM shell (connected)]# add disk --vm-identifier MyVM --
provisioned_size 8589934592 --interface virtio --format cow --
storage_domains-storage_domain storage_domain.name=DataStorage
```

12. 在管理器中将该 ISO 镜像上传到 **ISOStorage** 域中以供虚拟机使用。Red Hat Virtualization Manager 提供了一个上传工具，以保证使用正确用户权限将 ISO 镜像上传到正确的目录中。

```
# engine-iso-uploader --iso-domain=ISOStorage upload rhel-server-6.6-
x86_64-dvd.iso
Please provide the REST API password for the admin@internal oVirt
Engine user (CTRL+D to abort):
```

13. 在 CLI shell 中使用 **list files** 命令列出该存储域中的所有可用 ISO 文件。

```
[RHEVM shell (connected)]# list files --storagedomain-identifier
ISOStorage
```

14. 为安装介质添加新的虚拟 CD-ROM 驱动器。**vm-identifier** 选项将 CD-ROM 附加为 **MyVM** 的子资源。

```
[RHEVM shell (connected)]# add cdrom --vm-identifier MyVM --file-id
rhel-server-6.6-x86_64-dvd.iso
```

15. 启动该虚拟机。创建虚拟环境的过程已完成，虚拟机中包含了所有必须的功能组件。

```
[RHEVM shell (connected)]# action vm MyVM start --vm-os-boot
boot.dev=cdrom
```

请注意以上命令中的 **vm-os-boot** 选项，它把系统初始引导的引导设备设为 **cdrom**。在安装完成后，虚拟机会重新启动，系统的引导设备被重新恢复为 **hd**。

16. 使用 **list events** 命令并附带 **query** 选项显示具体事件类型。虚拟机的 **start** 操作可在 **events** 集合中添加几个条目。

```
[RHEVM shell (connected)]# list events --query "type=153"

id          : 105
description: MyVM was started by admin (Host: MyHost).
```

其中的 **"type=153"** 查询代表了用户启动一个虚拟机的事件。

17. 使用 **show event** 命令显示事件的完整信息。可使用这个命令根据 **type**、**name** 或 **id** 显示事件。

```
[RHEVM shell (connected)]# show event '60'

id          : 60
description  : New Tag foo was created by admin@internal.
code        : 432
correlation_id: 3e4d4350
custom_id   : -1
flood_rate  : 30
```

```
origin      : oVirt
severity    : normal
time        : 2013-07-03 10:57:43.257000+03:00
user-id     : fdfc627c-d875-11e0-90f0-83df133b58cc
```

18. 使用 **console** 命令访问您的虚拟机。

```
[RHEVM shell (connected)]# console MyVM
```



重要

请确认您的客户端系统上已经安装了和虚拟机的 **display-type** 相匹配的控制台程序。有效的协议包括 **SPICE**（默认协议）和 **VNC**。

第 3 章 命令

3.1. 连接到 RHEVM

3.1.1. 连接到 RHEVM (connect)

使用 **connect** 命令连接到 Red Hat Virtualization Manager，连接时使用的 URL、用户名、证书授权文件和密码可以在 **.ovirtshellrc** 文件中进行配置。**connect** 命令使用这个文件中的参数与 Manager 进行连接，而用户不需要在每次连接时都指定参数。

语法

connect *[options]*



注意

如果您已经在 **.ovirtshellrc** 文件中配置了用户名、密码、URL 和证书颁发文件，您将不需要指定额外的选项。

表 3.1. connect 命令的选项

选项	描述	必需的选项
--url	Red Hat Virtualization Manager 的 REST API 的 URL。它的格式是 https://[server]/ovirt-engine/api 。	是
--username	访问 Red Hat Virtualization Manager 的用户的用户名和所在的目录服务域。它的格式是 [username]@[domain] 。	是
--password	访问 the Red Hat Virtualization Manager 的用户的密码。	是
--key-file	使用 SSL 进行连接时使用的密钥文件。	不是
--cert-file	使用 SSL 进行连接时使用的证书文件。	不是
--ca-file	使用 SSL 进行连接时使用的证书授权文件。	是（除非使用了 --insecure 选项）
--insecure	允许 CLI 不使用证书来通过 SSL 进行连接。因为这个方法可能会遭到“中间人（man-in-the-middle -MITM）”袭击，因此在使用时应格外小心。	是（在没有提供证书授权的情况下）
--filter	根据用户的权限启用过滤功能。	不是
--port	连接到 REST API 所使用的端口号（适用于在 --url 中没有指定的情况）。	不是
--timeout	连接的超时时间。	不是

例 3.1. connect 示例（没有配置 .ovirtshellrc）

```
[RHEVM shell (disconnected)]# connect --url
"https://rhevm.example.com/ovirt-engine/api" --username
"admin@exampleids.com" --password "p@55w0rd!" --ca-file
"/home/user/ca.crt"
```

```
=====
>>> connected to RHEVM manager 4.0.0.0 <<<
=====

[RHEVM shell (connected)]#
```



注意

您可以使用 '--insecure' 选项而不需要指定“certificate”，这将可以在不使用证书的情况下进行连接，但是，这将使“中间人（man-in-the-middle - MITM）”袭击有机可乘，因此我们不推荐使用这个方法。

3.1.2. 从 RHEVM 断开连接（disconnect）

使用 **disconnect** 命令断开到 Red Hat Virtualization Manager 的连接。

语法

disconnect

例 3.2. disconnect 命令示例

```
[RHEVM shell (connected)]# disconnect

=====
>>> disconnected from RHEVM manager <<<
=====

[RHEVM shell (disconnected)]#
```

3.2. 资源

3.2.1. 列出一个集合中的资源（list）

使用 **list** 命令列出一个特定类型中的所有资源。它还可以包括一个可以对结果进行过滤的查询。

语法

list [*collection*] [*options*]

表 3.2. list 标准选项

选项	描述
--show-all	显示列出的每个资源的所有非空属性。如果没有使用这个选项，列出的资源只会显示 id 、 name 和 description 属性。
--query [<i>QUERY</i>]	使用基于 Red Hat Virtualization Manager 查询语言的服务器端查询来过滤列表。
--kwargs [<i>QUERY</i>]	使用客户端的查询来过滤结果。

选项	描述
<code>--case_sensitive true false</code>	在进行查询时区分大小写。
<code>--max</code>	可以显示的最多结果数。



注意

只适用于特定资源的选项会在相应资源的定义页中列出。

例 3.3. list 示例

列出虚拟机：

```
[RHEVM shell (connected)]# list vms
```

列出虚拟机（包括所有属性）：

```
[RHEVM shell (connected)]# list vms --show-all
```

列出状态为 'up' 的虚拟机：

```
[RHEVM shell (connected)]# list vms --query "status=up"
```

使用通配符列出所有域中的、满足一定条件的用户：

```
[RHEVM shell (connected)]# list users --query "username=jsmith*" --
case_sensitive false
```

显示 list 的帮助信息：

```
[RHEVM shell (connected)]# list --help
```

3.2.2. 显示一个资源 (show)

使用 **show** 命令显示资源属性。

语法

```
show [resource] [id|name] [options]
```



注意

只适用于特定资源的选项会在相应资源的定义页中列出。

例 3.4. show 命令示例

显示带有指定 **id** 的虚拟机:

```
[RHEVM shell (connected)]# show vm fcadfd5f-9a12-4a1e-bb9b-2b9d5c2e04c3
```

显示带有指定**名称**的虚拟机 :

```
[RHEVM shell (connected)]# show vm RHEL6-Server
```

3.2.3. 添加一个资源 (add)

使用 **add** 命令添加一个新资源。

语法

add *[resource] [options]*



注意

只适用于特定资源的选项会在相应资源的定义页中列出。

例 3.5. add 命令示例

创建一个虚拟机 :

```
[RHEVM shell (connected)]# add vm [vm-options]
```

创建一个用户 :

```
[RHEVM shell (connected)]# add user [user-options]
```

使用 **expect** 选项可以使 **add** 命令同步（在以下示例中，**add** 命令会等待 https/http 返回 '201'）：

```
[RHEVM shell (connected)]# add vm [vm-options] --expect '201-created'
```

3.2.4. 更新一个资源 (update)

使用 **update** 命令修改一个存在的资源。

语法

update *[resource] [id|name] [options]*



注意

只适用于特定资源的选项会在相应资源的定义页中列出。

例 3.6. update 命令示例

更新一个虚拟机：

```
[RHEVM shell (connected)]# update vm RHEL6-Server [vm-options]
```

3.2.5. 删除一个资源 (remove)

使用 **remove** 命令删除一个资源。

语法

```
remove [resource] [id|name] [options]
```

表 3.3. remove 标准选项

选项	描述
--async	异步删除资源。
--force	强制删除资源。这会同时删除相关资源在数据库中的所有记录，并只对 datacenter 和 vm 资源有效。

**注意**

只适用于特定资源的选项会在相应资源的定义页中列出。

例 3.7. remove 命令示例

删除一个虚拟机：

```
[RHEVM shell (connected)]# remove vm RHEL6-Server
```

异步删除一个虚拟机：

```
[RHEVM shell (connected)]# remove vm RHEL6-Server --async true
```

强制删除虚拟机：

```
[RHEVM shell (connected)]# remove vm RHEL6-Server --force
```

3.2.6. 在资源上执行特定操作 (action)

使用 **action** 命令执行一个与资源类型相关的特殊操作。

语法

```
action [resource] [id|name] [action] [options]
```



注意

只适用于特定资源的操作会在相应资源的定义页中列出。

例 3.8. action 命令示例

启动一个虚拟机

```
[RHEVM shell (connected)]# action vm RHEL6-Server start
```

停止一个虚拟机：

```
[RHEVM shell (connected)]# action vm RHEL6-Server stop
```

async 选项可以使用 **action** 命令以同步形式进行：

```
[RHEVM shell (connected)]# action vm [vm-options] --async false
```

3.2.7. 使用子资源 (--RESOURCE-identifier)

一些资源是以其它资源的子资源的形式出现的，子资源和它们的上级资源存在一个依赖关系。使用 **-RESOURCE-identifier [name]** 选项（其中的 *RESOURCE* 是上级资源的类型）来把一个资源标记为另一个资源的子资源。

例 3.9. 使用 add 创建子资源的示例

在虚拟机上创建一个网卡（NIC）

```
[RHEVM shell (connected)]# add nic --vm-identifier RHEL6-Server [nic-options]
```

以上命令中的 **--vm-identifier RHEL6-Server** 选项指定了把 NIC 添加到运行 RHEL6 Server 的虚拟机上。

在虚拟机上创建一个存储磁盘：

```
[RHEVM shell (connected)]# add disk --vm-identifier RHEL6-Server [user-options]
```

以上命令中的 **--vm-identifier RHEL6-Server** 选项指定了把 一个存储磁盘添加到运行 RHEL6 Server 的虚拟机上。

3.3. 其它命令

3.3.1. 退出 CLI shell (EOF)

使用 **EOF** 命令退出 CLI shell（或按 **Ctrl+D** 键）。

语法

EOF

例 3.10. EOF 使用示例

退出 CLI shell：

```
[RHEVM shell (connected)]# EOF
```

3.3.2. 列出系统的能力（capabilities）

使用 **capabilities --features** 命令列出所有版本的能力，以及当前版本的新功能。

语法

capabilities --features

例 3.11. capabilities 示例

列出当前版本的系统能力：

```
[RHEVM shell (connected)]# capabilities --features

name                : Search - Case Sensitivity
description          : Ability to specify whether a search
query should ignore case, by providing a URL parameter
url-parameters_set-parameter-name  : case_sensitive
url-parameters_set-parameter-context: matrix
url-parameters_set-parameter-type   : boolean
:
```

3.3.3. 清屏（clear）

使用 **clear** 命令清空 CLI 屏幕。

语法

clear

例 3.12. clear 使用示例

清空屏幕

```
[RHEVM shell (connected)]# clear
```

3.3.4. 连接到 VM (console)

使用 **console** 命令打开到一个虚拟机的图形控制台。这个命令会根据虚拟机的 **display-type** 参数设置使用 VNC 或 SPICE。

语法

console [*vm-id*|*vm-name*]

例 3.13. console 命令使用示例

打开到虚拟机的图形控制台：

```
[RHEVM shell (connected)]# console RHEL6-Server
```

3.3.5. 显示输入的信息 (echo)

使用 **echo** 命令在平面上显示输入的信息。使用 **\$out** 来显示最后一个 shell 命令的输出。

语法

echo [*input*]

例 3.14. 使用 echo 示例

在屏幕上显示输入的信息：

```
[RHEVM shell (connected)]# echo "Example text!"
```

显示最后一个 shell 命令的输出：

```
[RHEVM shell (connected)]# echo $out
```

3.3.6. 退出 CLI (exit)

使用 **exit** 命令退出 CLI。

语法

exit

例 3.15. exit 命令示例

退出 CLI：

```
[RHEVM shell (connected)]# exit
```

3.3.7. 运行一个脚本 (file)

使用 **file** 命令运行一个 CLI 脚本文件。脚本文件包括了一组可执行命令的文本文件。

语法

file [*file-location*]

例 3.16. file 命令示例

运行一个脚本文件：

```
[RHEVM shell (connected)]# file /example/example-script
```

3.3.8. 显示帮助信息 (help)

使用 **help** 命令显示 CLI 命令的帮助信息。

语法

help [*command*] [*resource*] [*options*]

例 3.17. help 命令的帮助示例

显示 CLI 帮助信息：

```
[RHEVM shell (connected)]# help
```

显示 **add** 命令的帮助信息：

```
[RHEVM shell (connected)]# help add
```

显示针对资源类型为 **vm** 的 **add** 命令的帮助信息。

```
[RHEVM shell (connected)]# help add vm
```

3.3.9. 显示用户命令的历史记录 (history)

使用 **history** 命令显示在 CLI shell 中所运行命令的历史记录。

语法

history

例 3.18. history 命令使用示例

显示用户命令的历史记录：

```
[RHEVM shell (connected)]# history
```

例 3.19. history --first 示例

使用 **--first** *n* 参数显示历史记录中开始的 *n* 条信息：

```
[RHEVM shell (connected)]# history --first 5
```

例 3.20. history --last 示例

使用 **--last** *n* 参数显示历史记录中最后的 *n* 条信息：

```
[RHEVM shell (connected)]# history --last 5
```

3.3.10. 显示 CLI 信息 (info)

使用 **info** 命令显示所在环境的连接详情和版本信息。

语法

info

例 3.21. info 命令使用示例

查看 CLI 信息：

```
[RHEVM shell (connected)]# info

backend version: 3.1
sdk version      : 3.1.0.4
cli version      : 3.1.0.6
python version   : 2.7.3.final.0

entry point      : https://www.example.com:8443/ovirt-engine/api
```

3.3.11. 测试连接 (ping)

使用 **ping** 命令测试到 Red Hat Virtualization Manager 的连接。如果这个命令收到了从远程系统发回的响应，说明连接所使用的 URL、用户名和密码是正确的。

语法

ping

例 3.22. 使用 ping 命令示例

测试您的连接：

```
[RHEVM shell (connected)]# ping
success: RHEVM manager could be reached OK.
```

3.3.12. 运行一个 Shell 命令 (shell)

使用 **shell** 命令从 Linux shell 中运行一个命令。这个命令可以和 Red Hat Virtualization Manager shell 一起被用来执行文件管理任务。

语法

shell [*vm-id*|*vm-name*]

例 3.23. 使用 shell 命令示例

列出当前工作目录中的文件：

```
[RHEVM shell (connected)]# shell ls
```

创建一个文件：

```
[RHEVM shell (connected)]# shell touch example.txt
```

复制一个文件：

```
[RHEVM shell (connected)]# shell cp example.txt /example-dir/.
```



注意

在 CLI 中，使用 **!** 和使用 **shell** 命令的功能是相同的。例如：

```
[RHEVM shell (connected)]# !touch example.txt
```

3.3.13. 显示最后运行的命令的状态 (status)

使用 **status** 命令显示最新运行命令的状态。

语法

status

例 3.24. status 命令使用示例

查看最新运行命令的状态：

```
[RHEVM shell (connected)]# status  
  
last command status: 0 (OK)
```

3.3.14. 显示系统的概况（summary）

使用 **summary** 命令显示系统状态的概况。

语法

summary

例 3.25. summary 命令使用示例

显示系统状态：

```
[RHEVM shell (connected)]# summary  
  
hosts-active           : 1  
hosts-total            : 2  
storage_domains-active: 2  
storage_domains-total  : 3  
users-active           : 1  
users-total            : 1  
vms-active             : 1  
vms-total              : 1
```

第 4 章 资源类型

4.1. brick

brick 资源类型是所有 Gluster 数据块在 Red Hat Virtualization 环境中的组合。

表 4.1. Gluster 数据块参数

名称	类型	描述	必需的	用户可创建	用户可更新
<code>--server_id</code>	字符串	Gluster 服务器的地址。	是	是	否
<code>--brick_dir</code>	字符串	数据块在 Gluster 服务器上所在的目录。	是	是	否
<code>--replica_count</code>	整数	指定一个重复的卷所具有的副本数量。	否	是	否
<code>--stripe_count</code>	整数	指定一个条带卷的条带数量	否	是	否

下表列出了基于资源的命令中的 **glustervolume** 选项。

表 4.2. 命令选项

选项	描述
<code>--cluster-identifier</code>	包括一个 glustervolume 子资源的 集群 的标识。
<code>--glustervolume-identifier</code>	把数据块添加为一个 glustervolume 的子资源。

例 4.1. 创建一个数据块

```
[RHEVM shell (connected)]# add brick --cluster-identifier Default --
glustervolume-identifier GlusterVol1 --server_id="server1" --
brick_dir="/exp1"
```

4.2. cdrom

cdrom 资源类型是所有虚拟 CD-ROM 资源在 Red Hat Virtualization 环境中的组合。

表 4.3. CD-ROM 参数

名称	类型	描述	必需的	用户可创建	用户可更新
<code>--file-id</code>	字符串	包括在 ISO 存储域中的一个 ISO 的文件名。	是	是	是

例 4.2. 创建一个新的 CD-ROM

```
[RHEVM shell (connected)]# add cdrom --vm-identifier MyVM --file-id rhel-server-6.2-x86_64-dvd.iso.iso
```

例 4.3. 更新一个 CD-ROM

```
[RHEVM shell (connected)]# update cdrom --vm-identifier MyVM --file-id rhel-server-6.3-x86_64-dvd.iso.iso
```

例 4.4. 删除一个 CD-ROM

```
[RHEVM shell (connected)]# remove cdrom --vm-identifier MyVM rhel-server-6.3-x86_64-dvd.iso.iso
```

4.3. cluster

cluster 资源类型是所有主机集群资源在 Red Hat Virtualization 环境中的组合。

表 4.4. 集群参数

名称	类型	描述	必需的	用户可创建	用户可更新
--cpu-id	字符串	用来定义集群中的所有主机都必须支持的 CPU 类型的服务器 CPU 标识。	是	是	是
--data_center-id name	字符串	主机集群所在数据中心的标识。	是	是	否
--name	字符串	主机集群的名称。	是	是	是
--version-major	整数	集群的主版本号。例如，Red Hat Virtualization 4.1 的主版本号是 4。	是	是	是
--version-minor	整数	集群的次版本号。例如，Red Hat Virtualization 4.1 的次版本值是 1。	是	是	是
--description	字符串	主机集群的描述。	否	是	是
--error_handling-on_error	字符串	指定当集群中的某个主机无法正常工作时需要进行的操作，可能的值包括 migrate 、 do_not_migrate 和 migrate_highly_available 。	否	是	是
--gluster_service	布尔值	值是 true 或 false 。	否	是	是
--memory_policy-overcommit-percent	double	指定主机内存的百分比，当所使用的内存超过这个百分比时，主机将无法运行更多的虚拟机。因为使用了 KSM 内存共享技术，虚拟机可以使用比主机物理内存更多的内存。推荐的值包括 100 (None)、 150 (Server Load) 和 200 (Desktop Load)。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--memory_policy-transparent-hugepages-enabled	布尔值	指定 Transparent Hugepages 的状态，它的值是 true 或 false 。	否	是	是
--scheduling_policy-policy	字符串	集群中的主机上的虚拟机调度模式，如 evenly_distributed 、 power_saving 或 blank （无调度模式）。	否	是	是
--scheduling_policy-thresholds-duration	整数	主机在负载超载的状态下运行多长时间后，调度程序才开始把负载移到其它主机上。	否	是	是
--scheduling_policy-thresholds-high	整数	指定 CPU 使用的百分比，当主机 CPU 的使用超过这个值时被认为主机处于负载超载状态。	否	是	是
--scheduling_policy-thresholds-low	整数	指定 CPU 使用的百分比，当主机 CPU 的使用低于这个值时被认为主机处于没有被充分利用的状态。	否	是	是
--threads_as_cores	布尔值	主机把线程看作为内核，这将允许主机上所运行的处理器核的数量大于主机本身的处理器核的数量。它的值是 true 或 false 。	否	否	否
--trusted_service	布尔值	指定是否使用一个 OpenAttestation 服务器来验证主机。	否	是	是
--virt_service	布尔值	值是 true 或 false 。	否	是	是
--expect	'201-create'd'	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否
--correlation_id	字符串	一个跨系统日志操作的标识。如果客户端没有定义这个标识，系统将会自动产生一个。	否	是	否

例 4.5. 创建一个新集群

```
[RHEVM shell (connected)]# add cluster --name Engineering --cpu-id "Intel Penryn Family" --datacenter-name Default --version-major 3 --version-minor 2
```

例 4.6. 更新一个集群

```
[RHEVM shell (connected)]# update cluster Engineering --name Finance
```

例 4.7. 删除一个集群

```
[RHEVM shell (connected)]# remove cluster Engineering
```

4.4. datacenter

datacenter 资源类型是所有数据中心资源在 Red Hat Virtualization 环境中的组合。

表 4.5. 数据中心参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	数据中心的名称。	是	是	是
--storage-type	字符串	数据中心的存储类型，它的值包括 iscsi 、 fc 、 nfs 、 localfs 和 posixfs 。	是	是	是
--version-major	int	数据中心的主版本号。例如，Red Hat Virtualization 4.1 的主版本号是 4。	是	是	是
--version-minor	int	数据中心的次版本号。例如，Red Hat Virtualization 4.1 的次版本号是 1。	是	是	是
--description	字符串	数据中心的描述。	否	是	是
--storage-format	字符串	数据中心的元数据格式，它包括 v1 、 v2 或 v3 。	否	是	是
--expect	'201- create '	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否
--correlation-id	字符串	一个跨系统日志操作的标识。如果客户端没有定义这个标识，系统将会自动产生一个。	否	是	否

例 4.8. 创建一个新的数据中心

```
[RHEVM shell (connected)]# add datacenter --name Boston --storage-type nfs
--version-major 3 --version-minor 2
```

例 4.9. 更新一个数据中心

```
[RHEVM shell (connected)]# update datacenter Boston --name India
```

例 4.10. 删除一个数据中心

```
[RHEVM shell (connected)]# remove datacenter Boston
```

4.5. disk

disk 资源类型是所有虚拟磁盘资源在 Red Hat Virtualization 环境中的组合。

表 4.6. 磁盘参数

名称	类型	描述	必需的	用户可创建	用户可更新
--provisioned-size	int	为磁盘保留的存储空间。这个空间被预先分配给磁盘使用，即使磁盘的大小小于 provisioned_size 的值。	是	是	是
--interface	字符串	磁盘的接口类型，可以是 ide 或 virtio 。	是	是	是
--format	字符串	底层的存储格式。使用 Copy On Write (cow) 可以创建快照，但对系统性能会有一些影响； Raw (raw) 不能创建快照，但可以提供更好的性能。	是	是	是
--size	int	磁盘实际的大小。	否	是	是
--sparse	布尔值	true ：磁盘所使用的物理存储无法被预先分配。	否	是	是
--bootable	布尔值	true ：磁盘被设为可引导。	否	是	是
--shareable	布尔值	true ：磁盘可以被共享。	否	是	是
--allow_snapshot	布尔值	true ：磁盘可以进行快照。	否	是	是
--propagate_errors	布尔值	true ：当磁盘出现错误时，不会导致虚拟机暂停，而是把磁盘错误传递给虚拟机的 OS。	否	是	是
--wipe_after_delete	布尔值	如果它的值是 true ，在磁盘被删除后，这个磁盘的底层物理存储会被清零。此选项会增加数据的安全性，但同时会消耗大量的操作资源，并会增加删除过程所需的时间。	否	是	是
--storage_domains-storage_domain	集合	为磁盘指定一个存储域。	否	是	否

--storage_domains-storage_domain 参数是一个使用下表中的子参数的集合。

表 4.7. --storage_domains-storage_domain parameters

名称	类型	描述
storage_domain.id name	字符串	磁盘的存储域的标识。

下表列出了基于资源的命令中的磁盘选项。

表 4.8. 格外的命令选项

选项	描述
--vm-identifier	把磁盘添加到一个 虚拟机 作为一个子资源。
--alias	在使用 show 命令时指定一个磁盘名。在针对于磁盘的查询中使用 --alias 来替代 --name 参数。

例 4.11. 创建一个新磁盘

```
[RHEVM shell (connected)]# add disk --name MyDisk --provisioned_size 8589934592 --interface virtio --format cow
```

例 4.12. 更新一个存储域

```
[RHEVM shell (connected)]# update disk MyDisk --shareable true
```

例 4.13. 删除一个存储域

```
[RHEVM shell (connected)]# remove disk MyDisk
```

下表列出了针对一个虚拟磁盘资源的操作。

表 4.9. 虚拟磁盘操作

操作	描述
activate	在虚拟机上激活一个磁盘。
deactivate	在虚拟机上取消激活一个磁盘。

4.6. glustervolume

glustervolume 资源类型是所有 Gluster 存储卷资源在 Red Hat Virtualization 环境中的组合。

表 4.10. Gluster 卷参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	Gluster 卷的名称。	是	是	否
--volume_type	字符串	指定 Gluster 卷的类型。有效值包括 DISTRIBUTE 、 REPLICATE 、 DISTRIBUTED_REPLICATE 、 STRIPE 和 DISTRIBUTED_STRIPE 。	是	是	否
--bricks-brick	集合	一个新的 Gluster 卷需要一组 Gluster 数据块 (brick) 的集合，这个参数指定了一个数据块的详情，使用多个 bricks-brick 参数可以指定多个数据块。下一个表介绍了数据块集合的详细信息。	是	是	否
--transport_types	集合	指定 Gluster 卷的传输类型。后面介绍了传输类型集合的详细信息。	否	是	否

名称	类型	描述	必需的	用户可创建	用户可更新
-- replica_count	整数	指定一个重复的卷所具有的副本数量。	否	是	否
-- stripe_count	整数	指定一个条带卷的条带数量	否	是	否
--options-option	集合	指定 Gluster 卷的选项。下一个表介绍了数据块集合的详细信息。	否	是	否

--bricks-brick 参数是一个使用下表中的子参数的集合。

表 4.11. bricks-brick 参数

名称	类型	描述
brick.server_id	字符串	Gluster 服务器的地址。
brick.brick_dir	字符串	数据块在 Gluster 服务器上的目录。

--transport_types 参数是一个使用下表中的子参数的集合。

表 4.12. transport_types 参数

名称	类型	描述
transport_type	字符串	指定要使用的传输类型（重复使用 transport_type 参数可以指定多个类型）。有效的值包括 TCP 和 RDMA。

--options-option 参数是一个使用下表中的子参数的集合。

表 4.13. options-option 参数

名称	类型	描述
option.name	字符串	Gluster 选项名。
option.value	字符串	Gluster 选项值。

下表列出了基于资源的命令中的 **glustervolume** 选项。

表 4.14. 格外的命令选项

选项	描述
--cluster-identifier	把 Gluster 卷添加到一个集群作为一个子资源。

例 4.14. 使用两个数据块创建一个 Gluster 卷

```
[RHEVM shell (connected)]# add glustervolume --cluster-identifier Default
--name GlusterVol1 --volume-type DISTRIBUTE --bricks-brick
"brick.server_id=UUID,brick.brick_dir=filepath"--bricks-brick
"brick.server_id=UUID,brick.brick_dir=filepath"
```

例 4.15. 删除一个 Gluster 卷

```
[RHEVM shell (connected)]# remove glustervolume --cluster-identifier
Default --name GlusterVol1
```

下表列出了针对 Gluster 卷资源的操作。

表 4.15. Gluster 卷操作

操作	描述
start	使一个 Gluster 卷可以被使用。
stop	停用一个 Gluster 卷。
setOption	设置一个 Gluster 卷选项。
resetOption	把一个 Gluster 卷选项重置为默认的值。
resetAllOptions	把所有 Gluster 卷选项重置为默认的值。

4.7. group

group 资源类型是所有数据中心资源在 Red Hat Virtualization 环境中的组合。

表 4.16. 组参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	组的名称（一般是在身份目录服务中的、完全的组路径）	否	否	否

例 4.16. 创建一个组

```
[RHEVM shell (connected)]# add group --name
www.example.com/accounts/groups/mygroup --domain-name example.com
```

4.8. host

host 资源类型是所有主机资源在 Red Hat Virtualization 环境中的组合。

表 4.17. 主机参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	主机的名称。	是	是	是
--address	字符串	主机的 IP 地址或主机名。	是	是	是
--root_password	字符串	主机 root 用户的密码。	是	是	是
--cluster-id name	字符串	指定主机所在的集群。	是	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
<code>--port</code>	整数	与运行在主机上的 VDSM 守护进程进行通信所使用的端口。	否	是	是
<code>--storage-manager-priority</code>	整数	主机作为 SPM (storage pool manager - 存储池管理器) 的优先级。	否	是	是
<code>--power-management-type</code>	字符串	主机电源管理设备的类型。	否	是	是
<code>--power-management-enabled</code>	布尔值	指定是否启用电源管理配置。	否	是	是
<code>--power-management-address</code>	字符串	电源管理设备的主机名或 IP 地址。	否	是	是
<code>--power-management-user-name</code>	字符串	电源管理的用户名。	否	是	是
<code>--power-management-password</code>	字符串	电源管理的密码。	否	是	是
<code>--power-management-options-option</code>	集合	<code>power_management-type</code> 参数所指定的隔离 (fence) 选项。	否	是	是
<code>--reboot_after_installation</code>	布尔值	指定在 VDSM 安装完成后主机是否重启。	否	是	否

`--power_management-options-option` 参数是一个使用下表中的子参数的集合。

表 4.18. `--power_management-options-option` 参数

名称	类型	描述
<code>option.name</code>	字符串	电源管理选项名。
<code>option.value</code>	字符串	电源管理选项值。

例 4.17. 创建一个新主机

```
[RHEVM shell (connected)]# add host --name Host1 --address
host1.example.com --root_password p@55w0rd! --cluster-name Default
```

例 4.18. 更新一个新主机

```
[RHEVM shell (connected)]# update host Host1 --name Host2
```

例 4.19. 删除一个主机

```
[RHEVM shell (connected)]# remove host Host1
```

下表列出了针对主机资源的操作。

表 4.19. 主机操作

操作	描述
activate	激活一个主机。
approve	批准一个主机。
commitnetconfig	保存网络配置。
deactivate	取消激活一个主机。
fence	隔离一个主机。
forceselectspm	把主机设为 SPM。
install	在主机上安装 VDSM。
iscsidiscover	执行 iSCSI discover 命令。
iscsilogin	执行 iSCSI login 命令。

下表列出了针对于 **fence** 操作的选项。

表 4.20. 隔离选项

选项	描述
manual	手工隔离主机。使用这个选项通知 Manager 主机处于无响应状态，并已经被手动重启。
restart	重启主机（将执行 stop、wait、status、start、wait、status 操作）。
start	开启主机。
stop	关闭主机。
status	检查主机的状态。

例 4.20. 确认重启的主机

```
[RHEVM shell (connected)]# action host Host1 fence --fence_type manual
```

4.9. network

network 资源类型是所有逻辑网络资源在 Red Hat Virtualization 环境中的组合。

表 4.21. 逻辑网络参数

名称	类型	描述	必需的	用户可创建	用户可更新
--data-center-id name	字符串	逻辑网络所在的数据中心的标识。	是	是	否
--name	字符串	逻辑网络的名称。	是	是	否
--description	字符串	逻辑网络的描述信息。	否	是	是
--vlan-id	字符串	VLAN 标签 (tag)。	否	是	是
--ip-address	字符串	逻辑网络的网桥的 IP 地址。	否	是	是
--ip-gateway	字符串	逻辑网络的网桥的网关。	否	是	是
--ip-netmask	字符串	逻辑网络的网桥的网络掩码。	否	是	是
--display	布尔值	指定一个逻辑网络是否只作为显示使用。它的值包括 true 或 false 。	否	是	是
--stp	布尔值	当把这个参数设为 true 时, Spanning Tree Protocol 将在这个网络中被启用。	否	是	是
--mtu	整数	指定逻辑网络中的、用户定义的 mtu (maximum transmission unit - 最大传输单元) 值。	否	是	是
--usages-usage	集合	为逻辑网络定义使用参数。	否	否	是

--usages-usage 参数是一个使用下表中的子参数的集合。

表 4.22. usages-usage 参数

名称	类型	描述
usage	字符串	网络的使用类型。可选的值是 VM 和 DISPLAY 。

下表列出了基于资源的命令中的**网络**选项。

表 4.23. 命令选项

选项	描述
--cluster-identifier	把网络添加到一个 集群 作为一个子资源。

例 4.21. 创建一个新网络

```
[RHEVM shell (connected)]# add network --name WebNetwork --datacenter-name Default
```

例 4.22. 为集群附加一个存在的网络

```
[RHEVM shell (connected)]# add network --name WebNetwork --datacenter-name Default --cluster-identifier Default
```

例 4.23. 更新一个网络

```
[RHEVM shell (connected)]# update network WebNetwork --name DataNetwork
```

例 4.24. 删除一个网络

```
[RHEVM shell (connected)]# remove network WebNetwork
```

4.10. nic

nic 资源类型是所有逻辑网络资源在 Red Hat Virtualization 环境中的组合。它是 **host** 和 **vm** 的子资源，但相关的定义会有所不同。下面的两个表格分别提供了 **nic** 资源在这两个资源中使用时的参数。

表 4.24. 主机网络接口参数

名称	类型	描述	必需的	用户可创建	用户可更新
--network-id name	字符串	指定网卡所连接的网络。	是	是	是
--name	字符串	主机网络接口名，如 eth0 。	是	是	是
--bonding-slaves-host_nic	集合	组成绑定接口（bond）的从网卡的集合。	否	是	是
--bonding-options-option	集合	绑定接口的选项列表。每个选项包括 name 和 value 属性。	否	是	是
--ip-gateway	字符串	网络网关的 IP 地址。	否	是	是
--boot_protocol	字符串	在主机启动时分配 IP 地址的协议，如 dhcp 或 static 。	否	是	是
--mac	字符串	接口的 MAC 地址。	否	是	是
--ip-address	字符串	接口的 IP 地址。	否	是	是
--ip-netmask	字符串	接口 IP 地址的子网掩码。	否	是	是
--ip-mtu	int	网络接口的最大传输单位。	否	否	是

表 4.25. 虚拟机网络接口参数

名称	类型	描述	必需的	用户可创建	用户可更新
--network-id name	字符串	指定网卡所连接的网络。	是	是	是
--name	字符串	网络接口名，如 eth0 。	是	是	是
--mac-address	字符串	接口的 MAC 地址。	否	是	是
--interface	字符串	定义网络接口类型，如 e1000 、 virtio 、 rtl8139 和 rtl8139_virtio 。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
<code>--port-mirroring-networks-network</code>	集合	指定一组被用来对网络接口进行复制（镜像）的网络。	否	是	是

`--bonding-slaves-host_nic` 参数是一个使用下表中的子参数的集合。

表 4.26. `--bonding-slaves-host_nic`

名称	类型	描述
<code>host_nic.id name</code>	字符串	指定另外一个主机 NIC 作为绑定。

`--bonding-options-option` 参数是一个使用下表中的子参数的集合。

表 4.27. `--bonding-options-option`

名称	类型	描述
<code>option.name</code>	字符串	绑定选项名。
<code>option.value</code>	字符串	绑定选项值。
<code>type</code>	字符串	绑定选项类型。

`--port-mirroring-networks-network` 参数是一个使用下表中的子参数的集合。

表 4.28. `--port-mirroring-networks-network`

名称	类型	描述
<code>network.id</code>	字符串	指定作为网络镜像的网络。

下表列出了基于资源的命令中的 NIC 选项。

表 4.29. 命令选项

选项	描述
<code>--host-identifier</code>	把 NIC 添加到一个主机作为一个子资源。
<code>--vm-identifier</code>	把 NIC 添加到一个虚拟机作为一个子资源。

例 4.25. 在虚拟机上创建一个新网络接口

```
[RHEVM shell (connected)]# add nic --vm-identifier MyVM1 --name eth0 --
network-name MyNetwork
```

例 4.26. 更新虚拟机中的网络接口

```
[RHEVM shell (connected)]# update nic eth0 --vm-identifier MyVM1 --ip-
address 10.5.68.123
```

例 4.27. 删除虚拟机中的网络接口

```
[RHEVM shell (connected)]# remove nic eth0 --vm-identifier MyVM1
```

例 4.28. 在主机中配置网络绑定

```
[RHEVM shell (connected)]# add nic --host-identifier MyHost1 --name bond1
--network-name MyNetwork --bonding-slaves-host_nic host_nic.name=eth0 --
bonding-slaves-host_nic host_nic.name=eth1
```

例 4.29. 为主机网络接口分配逻辑网络

```
[RHEVM shell (connected)]# action nic eth0 attach --host-identifier
MyHost1 --network-name MyNetwork
```

下表列出了针对主机 NIC 资源的操作。

表 4.30. 主机 NIC 操作

操作	描述
attach	把 NIC 添加到一个主机。
detach	把一个 NIC 从主机上删除。

下表列出了针对一个虚拟机 NIC 资源的操作。

表 4.31. 虚拟机 NIC 操作

操作	描述
activate	在虚拟机上激活一个 NIC
deactivate	在虚拟机上取消激活一个 NIC。

4.11. permission

permission资源类型是所有权限资源在 Red Hat Virtualization 环境中的组合。

表 4.32. 权限参数

名称	类型	描述	必需的	用户可创建	用户可更新
--user-id、--group-id	字符串	用户或组的权限。	是	是	否
--role-id	字符串	指定用于分配权限的角色。	是	是	否
--expect	'201-create d'	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否

下表列出了基于资源的命令中的 **permission** 选项。

表 4.33. 格外的命令选项

选项	描述
--cluster-identifier	为一个集群添加权限。
--correlation-id	跨系统日志的标签标识
--cpuprofile-identifier	为一个 CPU 配置档案添加权限。
--datacenter-identifier	为数据中心添加权限。
--disk-identifier	为磁盘添加权限。
--diskprofile-identifier	为磁盘配置档案添加权限。
--host-identifier	为主机添加权限。
--iscsibond-identifier	为一个 iSCSI 绑定添加权限。
--network-identifier	为网络添加权限。
--storagedomain-identifier	为存储域添加权限。
--template-identifier	为模板添加权限。
--vm-identifier	为虚拟机添加权限。
--vmpool-identifier	为虚拟机池添加权限。
--vnicprofile-identifier	为一个 VNIC 配置档案添加权限。

例 4.30. 创建一个新权限

```
[RHEVM shell (connected)]# add permission --role-id 00000000-0000-0000-0000-0000000000001 --user-id 8b9456ae-e2c8-426e-922d-b01bb8a805fb
```

4.12. permit

permit 资源类型是组成用户角色的权利资源在 Red Hat Virtualization 环境中的组合。

表 4.34. 权利参数

名称	类型	描述	必需的	用户可创建	用户可更新
--id	字符串	添加权利的标识。	是	是	否

下表列出了基于资源的命令中的 **permit** 选项。

表 4.35. 格外的命令选项

选项	描述
--role-identifier	为一个用户角色添加权利。

例 4.31. 创建一个新权限

```
[RHEVM shell (connected)]# add permit --role-identifier MyRole --id 1
```

4.13. quotas

quota 资源类型是所有数据中心配额资源在 Red Hat Virtualization 环境中的组合。

表 4.36. Quota 参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	配额名。	是	是	是
--description	字符串	配额的描述信息。	是	是	是

4.14. role

role 资源类型是所有用户角色资源在 Red Hat Virtualization 环境中的组合。

表 4.37. 角色参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	角色名。	是	是	是
--permits-permit	集合	角色开始所包括的权利列表。	是	是	否
--description	字符串	角色的描述。	否	是	是
--administrative	布尔值	如果这个角色是管理角色，把这个参数设为 true 。	否	是	是

--permits-permit 参数是一个使用下表中的子参数的集合。

表 4.38. --permits-permit 参数

名称	类型	描述
permit.id	字符串	添加到角色中的权利。

例 4.32. 创建一个新角色

```
[RHEVM shell (connected)]# add role --name MyRole --permits-permit {permit.id: 1;},{permit.id: 2;}
```

4.15. snapshot

snapshot 资源类型是所有虚拟快照资源在 Red Hat Virtualization 环境中的组合。

表 4.39. 快照参数

名称	类型	描述	必需的	用户可创建	用户可更新
--description	字符串	快照描述。	是	是	否

下表列出了基于资源的命令中的快照选项。

表 4.40. 额外的命令选项

选项	描述
--vm-identifier	把磁盘添加到一个 虚拟机 作为一个子资源。

例 4.33. 创建一个新快照

```
[RHEVM shell (connected)]# add snapshot --vm-identifier MyVM --description 'My Snapshot'
```

例 4.34. 删除一个快照

```
[RHEVM shell (connected)]# remove snapshot [snapshot_id]
```

下表列出了针对一个虚拟机快照资源的操作。

表 4.41. 虚拟机快照操作

操作	描述
restore	恢复一个快照。

4.16. statistic

statistic资源类型是 Red Hat Virtualization 环境中资源统计的组合。资源统计数据会根据它们的标识符列出。

表 4.42. statistic 资源标识符

选项	描述
--brick-identifier	查看具体块统计内容的资源标识符。
--cluster-identifier	查看指定集群统计数据的资源标识符。
--datacenter-identifier	查看指定数据中心统计数据的资源标识符。
--disk-identifier	查看指定虚拟磁盘统计数据的资源标识符。
--glustervolume-identifier	查看指定 gluster 卷统计数据的资源标识符。
--host-identifier	查看指定主机统计数据的资源标识符。
--job-identifier	查看指定任务统计数据的资源标识符。
--nic-identifier	查看指定 NIC 统计数据的资源标识符。
--numanode-identifier	查看指定 NUMA 节点统计数据的资源标识符。

选项	描述
--step-identifier	查看指定步骤统计数据的资源标识符。
--storagedomain-identifier	查看指定存储域统计数据的资源标识符。
--vm-identifier	查看指定虚拟机统计数据的资源标识符。

使用 **list** 命令及其相关资源标识符查看每个资源的统计数据集合。以下示例提供了指定主机的可用统计数据列表：

```
[RHEVM shell (connected)]# list statistics --host-identifier Host_name|id
```

所提供统计数据中的 **name** 或者 **id** 可与 **show** 命令及资源标识符一同使用，以便查看所指定统计数据中的进一步信息。以下示例显示该主机指定统计数据中的详细资料：

```
[RHEVM shell (connected)]# show statistic statistic_name|id --host-identifier Host name|id
```

4.17. storageconnection

storageconnection 资源允许您添加、编辑、删除存储连接。

表 4.43. 存储连接参数

名称	类型	描述	必需的	用户可创建	用户可更新
--address	字符串	存储域的主机名或 IP 地址。	是（仅限于 NFS 和 iSCSI）	是	是
--correlation_id	字符串	存储连接的标签标识	否	否	是
--expect	'201-creatd'	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否
--iqn	字符串	存储设备的目标 IQN。	是（只适用于 iSCSI）	是	是
--mount_options	字符串	挂载 PosixFS 共享的选项。	否	是	是
--nfs_retrans	整数	NFS 客户端为了完成一个请求尝试重新传输的次数。	否	是	是
--nfs_timeo	整数	NFS 客户端为了完成一个请求可以等待的时间（以 0.1 秒为单位）。	否	是	是
--nfs_version	字符串	NFS 使用的版本。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--password	字符串	登录到 iSCSI 存储域目标的 CHAP 密码。	否	是	是
--path	字符串	存储域的文件挂载路径。已经被存储连接使用的路径无法被更新。	是（只适用于 NFS、本地和 Posix FS）	是	是
--port	整数	iSCSI 存储域使用的 TCP 端口。	是（只适用于 iSCSI）	是	是
--storagedomain-identifier	字符串	作为磁盘的存储域。	否	否	否
--type	字符串	存储域类型。	是	是	否
--username	字符串	登录到 iSCSI 存储域目标的 CHAP 用户名。	否	是	是
--vfs_type	字符串	Linux 支持的 PosixFS 共享文件系统类型。	是（只适用于 Posix FS）	是	是

例 4.35. 创建一个新存储连接

```
[RHEVM shell (connected)]# add storageconnection --address
storage.example.com --path /storage/nfs --type nfs
```

4.18. storagedomain

storagedomain 资源类型是所有存储域资源在 Red Hat Virtualization 环境中的组合。

表 4.44. 存储域参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	存储域的名称。	否	是	是
--format	布尔值	数据中心的元数据格式，包括 v1 、 v2 或 v3 。	是	是	否
--host-id name	字符串	存储域进行初始化所在的主机。这个主机需要可以访问到指定的物理存储设备。	是	是	否
--storage-address	字符串	存储设备的 IP 地址和主机名。	是	是	否
--storage-logical_unit	集合	存储设备的逻辑单元信息。	参阅后面的信息	是	否
--storage-mount_options	字符串	挂载存储域的选项。	参阅后面的信息	是	否

名称	类型	描述	必需的	用户可创建	用户可更新
<code>--storage-override_luns</code>	布尔值	指定是否覆盖逻辑单元号（logical unit number）。它的值只能是 true 或 false 。	参阅后面的信息	是	否
<code>--storage-path</code>	字符串	存储设备上作为存储域使用的路径。	参阅后面的信息	是	否
<code>--storage-type</code>	字符串	用于数据中心的存储类型，其中包括 iscsi 、 fcp 、 nfs 、 glusterfs 、 localfs 和 posixfs 。	是	是	否
<code>--storage-vfs_type</code>	字符串	定义存储域的文件系统类型。	参阅后面的信息	是	否
<code>--type</code>	字符串	存储域的类型，有效值包括 data 、 iso 和 export 。	是	是	否

`--storage-logical_unit` 参数是一个使用下表中的子参数的集合。

表 4.45. storage-logical_unit 参数

名称	类型	描述
<code>logical_unit.address</code>	字符串	包括存储设备的服务器的地址。
<code>logical_unit.port</code>	整数	服务器的端口号。
<code>logical_unit.target</code>	字符串	存储设备的目标 IQN。
<code>logical_unit.username</code>	字符串	登录到一个目标的 CHAP 用户名。
<code>logical_unit.password</code>	字符串	登录到一个目标的 CHAP 密码。
<code>logical_unit.serial</code>	字符串	目标的序列 ID。
<code>logical_unit.vendor_id</code>	字符串	目标的厂商名。
<code>logical_unit.product_id</code>	字符串	目标的产品代码。
<code>logical_unit.lun_mapping</code>	整数	映射为目标的 LUN（Logical Unit Number）设备。
<code>logical_unit.portal</code>	字符串	逻辑单元门户。
<code>logical_unit.paths</code>	整数	逻辑单元路径。
<code>logical_unit.id</code>	字符串	逻辑单元 ID 的标识。

根据 `storage-type` 的值使用以下参数。

表 4.46. 存储类型参数

类型	参数
nfs	--storage-address、--storage-path
iscsi 或 fcp	--storage-address、--storage-logical_unit、--storage-override_luns
glusterfs	--storage-address、--storage-path、--storage-vfs_type
local	--storage-path
posixfs	--storage-path、--storage-vfs_type、--storage-address、--storage-mount_options

下表列出了基于资源的命令中的 **storagedomain** 选项。

表 4.47. 额外的命令选项

选项	描述
--datacenter-identifier	把存储域添加到一个 datacenter 作为一个子资源。

例 4.36. 创建一个新的存储域

```
[RHEVM shell (connected)]# add storagedomain --name DataStorage --
datacenter-name Default -type data
```

例 4.37. 添加 gluster 存储域

```
[RHEVM shell (connected)]# add storagedomain --type data --storage-type
glusterfs --name RHS_01 --storage-address 192.0.2.0 --storage-path Vol_ONE
--storage-vfs_type glusterfs
```

例 4.38. 更新一个存储域

```
[RHEVM shell (connected)]# update storagedomain DataStorage --name
DataStorageOld
```

例 4.39. 删除一个存储域

```
[RHEVM shell (connected)]# remove storagedomain DataStorage
```

下表列出了针对存储域资源的操作。

表 4.48. 存储域操作

操作	描述
activate	在数据中心中激活一个存储域。
deactivate	在数据中心中取消激活存储域。

4.19. tag

tag 资源类型是所有标签资源在 Red Hat Virtualization 环境中的组合。

表 4.49. 标签参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	标签名称。	是	是	是
--description	字符串	标签描述。	是	是	是
--parent-name	字符串	标签所附加到的上一级标签。	是	是	是

例 4.40. 创建一个新标签

```
[RHEVM shell (connected)]# add tag --name MyTag --description "A virtual machine tag" --parent MyParentTag
```

4.20. template

template 资源类型是所有模板资源在 Red Hat Virtualization 环境中的组合。只需要 --vm-id|name 和 --name 参数。如果未指定自选参数，则模板将继承用来生成该模板的虚拟机所使用的设置。

表 4.50. 模板参数

名称	类型	描述	必需的	用户可创建	用户可更新
--vm-id name	字符串	作为基本模板使用的虚拟机的 ID 或名称。	是	是	否
--name	字符串	虚拟机模板名称。	是	是	是
--memory	长整数	虚拟机模板的内存大小（以字节为单位）	否	是	是
--cpu-topology-cores	整数	虚拟机模板可用的 CPU 内核数量。	否	是	是
--high_availability-enabled	布尔值	设为 true 以便为虚拟机启用高可用性。	否	是	是
--os-cmdline	字符串	内核需要使用的内核命令行参数。这个选项支持直接引导一个 Linux 内核，而不需要使用 BIOS bootloader。	否	是	是
--origin	字符串	虚拟机模板的原形。请指定 rhev 、 vmware 或者 xen 。	否	是	是
--high_availability-priority	整数	设置虚拟机模板高可用性的优先级（如引导顺序）。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--timezone	字符串	Windows 虚拟机模板在 Sysprep 中的时区设置。请指定类似 GMT+00:00 的格式。	否	是	是
--domain-name	字符串	虚拟机模板的域名。	否	是	是
--type	字符串	定义虚拟机的类型，请指定 desktop 或者 server 。	否	是	是
--stateless	布尔值	如果虚拟机处于无状态（stateless），则将这个参数设为 true 。无状态的虚拟机会在系统启动时提取磁盘快照，并在关机时删除对系统所做的所有改变。这意味着每次重启后将丢失对系统所做改动。	否	是	是
--delete-protected	布尔值	设为 true 以便无法删除使用此模板创建的虚拟机。	否	是	是
--sso-methods-method	集合	定义要使用的单点登录方法。例如： --sso-methods-method method.id=GUEST_AGENT 。	否	是	是
--rng-device-rate-bytes	整数	指定每个周期中允许消耗的字节数。	否	是	是
--rng-device-rate-period	整数	以毫秒为单位指定周期时间。如果指定该参数，就必须要指定 --rng-device-rate-bytes 。	否	是	是
--rng-device-source	字符串	随机数产生器的资源。请指定 random 或者 hwrng 。	否	是	是
--console-enabled	布尔值	设为 true 以便启用 VirtIO 控制台设备功能。	否	是	是
--placement-policy-affinity	字符串	使用该模板创建的虚拟机的迁移关联（migration affinity），请指定 migratable 、 user_migratable 或者 pinned 。	否	是	是
--description	字符串	虚拟机模板描述。	否	是	是
--comment	字符串	虚拟机模板注释。	否	是	是
--custom-properties-custom-property	集合	传递给脚本的一组用户自定义的环境变量。	否	是	是
--os-type	字符串	虚拟机模板的操作系统类型。	否	是	是
--os-boot	集合	虚拟机模板的引导设备。请指定 cdrom 、 hd 或者 network 。例如： --os-boot boot.dev=hd 。	否	是	是
--cpu-topology-sockets	整数	虚拟机模板可用的 CPU 插槽数量。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--cpu_shares	整数	相比其他虚拟机可要求的 CPU 资源水平。例如：低优先级虚拟机使用 512 ，中等优先级虚拟机使用 1024 ，而高优先级虚拟机使用 2048 。	否	是	是
--cpu-architecture	字符串	定义 CPU 架构。请指定 x86_64 、 ppc64 或者 undefined 。	否	是	是
--os-kernel	字符串	到基于模板所创建的虚拟机的引导内核镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	否	是	是
--display-type	字符串	定义显示类型，请指定 spice 或 vnc 。	否	是	是
--display-monitors	整数	定义可用的显示数量。	否	是	是
--display-single_qxl_pci	布尔值	设为 true 以便使用单一虚拟 PCI 设备驱动多台显示器。	否	是	是
--display-allow_override	布尔值	设为 true 以便可以覆盖模板控制台设置。	否	是	是
--display-smartcard_enabled	布尔值	设为 true 以便为虚拟机启用智能卡功能。	否	是	是
--display-file_transfer_enabled	布尔值	设为 true 以便启用 SPICE 文件传输。	否	是	是
--display-copy_paste_enabled	布尔值	设为 true 以便启用 SPICE 剪切版复制和粘贴。	否	是	是
--display-keyboard_layout	字符串	定义虚拟机键盘类型。只有使用 VNC 协议时这个选项才有效。请指定类似 en-US 的格式。	否	是	是
--os-initRd	字符串	指定到一个指定内核所使用的 initrd 镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	否	是	是
--usb-enabled	布尔值	设定为 true 以便在该虚拟机中启用 USB 支持。这个选项只可用于使用 SPICE 协议的虚拟机。	否	是	是
--usb-type	字符串	如果已启用 USB 支持，则定义 USB 类型。请指定 Legacy 或者 Native 。	否	是	是
<div>  重要 Legacy USB 选项已过时，将不会出现在 Red Hat Virtualization 4.1 中。 </div>					
--tunnel_migration	布尔值	设为 true 启用使用 libvirt 守护进程进行数据传输的功能。隧道传输使用更强大的加密算法，但会增加传输过程中的数据负载。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--migration-downtime	整数	以毫秒为单位定义进行实时迁移时虚拟机可以停机的最长时间。	否	是	是
--virtio-scsi-enabled	布尔值	设为 true 以便在该虚拟机中添加 VirtIO 控制台设备。	否	是	是
--soundcard-enabled:	布尔值	设为 true 以便启用声卡。	否	是	是
--vm-disks-disk	集合	附加到模板中的磁盘。	否	是	否
--id	字符串	虚拟机模板 ID。	否	是	是
--permissions-clone	布尔值	设为 true 以便将源虚拟机权限复制到模板中。	否	是	是
--version-version-name	字符串	与 --version-base-template-id 参数一同使用。定义子模板名称。	否	是	是
--version-base-template-id	字符串	定义作为 root 模板使用的模板 ID。如果要创建这个模板作为 root 模板的子模板使用，则请使用此参数。	否	是	是
--cpu-cpu-tune-vcpu-pin	集合	定义虚拟机中哪个虚拟 CPU 要与主机物理 CPU 固定 (pin)。	否	是	是
--serial-number-policy	字符串	定义虚拟机模板的序列号策略。请指定 host 、 vm 或者 custom 。如果使用 custom ，则还要使用 --serial-number-value 定义序列号。	否	是	是
--serial-number-value	字符串	定义虚拟机模板的序列号。	否	是	是
--bios-boot-menu-enabled	布尔值	设为 true 以便启用引导菜单。	否	是	是
--cluster-id	字符串	通过指定集群 ID 定义要使用的集群。	否	是	是
--cluster-name	字符串	通过指定集群名称定义要使用的集群。	否	是	是
--cpu-profile-id	字符串	定义要使用的 CPU 配置文件。请使用 list cpuprofiles 命令查询 CPU 配置文件 ID 完整列表。	否	是	是
--expect	'201-created'	返回预期 HTTP 标头前，请求以异步的方式进行处理。它对长线任务非常有用，可以防止在任务完成前返回成功信号。	否	是	是
--correlation-id	字符串	跨系统日志操作标识符。如果客户端没有定义，系统将会自动生成该标识符。	否	是	是

--vm-disks-disk 参数是使用下表中子参数的集合。

表 4.51. --sso-methods-method parameters

名称	类型	描述
<code>method.id</code>	字符串	要使用的单点登录方法。例如： <code>GUEST_AGENT</code> 。

`--custom_properties-custom_property` 参数是一个使用下表中的子参数的集合。

表 4.52. --custom_properties-custom_property 参数

名称	类型	描述
<code>custom_property.name</code>	字符串	自定义属性名。
<code>custom_property.value</code>	字符串	自定义属性的值。

`--os-boot` 参数是使用下表中子参数的集合。

表 4.53. --os-boot parameters

名称	类型	描述
<code>boot.dev</code>	字符串	虚拟机模板的引导设备。请指定 <code>cdrom</code> 、 <code>hd</code> 或者 <code>network</code> 。

`--vm-disks-disk` 参数是一个使用下表中的子参数的集合。

表 4.54. --vm-disks-disk 参数

名称	类型	描述
<code>disk.id</code>	字符串	虚拟磁盘的标识。
<code>storage_domains.storage_domain</code>	集合	为磁盘存储域定义一组子参数。

`--cpu-cpu_tune-vcpu_pin` 参数是使用下表中的子参数的集合。

表 4.55. --cpu-cpu_tune-vcpu_pin parameters

名称	类型	描述
<code>vcpu_pin.vcpu</code>	整数	分配的虚拟 CPU。
<code>vcpu_pin.cpu_set</code>	字符串	主机中的物理 CPU。

例 4.41. 创建一个新模板

```
[RHEVM shell (connected)]# add template --name MyTemplate1 --vm-name MyVM1
```

例 4.42. 更新一个新模板

```
[RHEVM shell (connected)]# update template MyTemplate1 --memory 1073741824
```

例 4.43. 删除一个模板


```
[RHEVM shell (connected)]# remove template MyTemplate1
```

下表列出了针对一个虚拟机模板资源的操作。

表 4.56. 虚拟机模板操作

操作	描述
export	把模板导出到一个导出存储域中。

4.21. user

user 资源类型是所有用户资源在 Red Hat Virtualization 环境中的组合。

表 4.57. 用户参数

名称	类型	描述	必需的	用户可创建	用户可更新
--user_name	字符串	目录服务所提供的用户名。	是	是	否
--domain-id name	字符串	目录服务域的标识。	是	是	否

例 4.44. 创建新用户

```
[RHEVM shell (connected)]# add user --user_name jsmith --domain-name example.com
```

4.22. vm

vm 资源类型是所有虚拟机资源在 Red Hat Virtualization 环境中的组合。

表 4.58. 虚拟机参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	虚拟机名	是	是	是
--template-id name	字符串	虚拟机所基于的模板标识。	是	是	否
--cluster-id name	字符串	指定主机所在的集群。	是	是	是
--instance_type-id name	字符串	定义实例类型。指定 custom 、 large 、 medium 、 small 、 tiny 或者 xlarge 。	否	是	是
--quota-id	字符串	虚拟机配额使用的标识。	否	是	否

名称	类型	描述	必需的	用户可创建	用户可更新
--timezone	字符串	用于 Windows 虚拟机的 Sysprep 时区设置。	否	是	是
--os-boot	集合	虚拟机的引导设备。请指定 cdrom 、 hd 或者 network 。	否	是	是
--custom_properties-custom_property	集合	用来自定义脚本的一组用户定义环境变量参数。	否	是	是
--os-type	字符串	虚拟机的操作系统类型。	否	是	是
--usb-enabled	布尔值	定义虚拟机的 USB 策略。如果设为 true ，将在虚拟机上启用 USB 功能。	否	是	是
--usb-type	字符串	指定 USB 的类型（在 USB 已被启用的情况下）。	否	是	是
--type	字符串	指定虚拟机的类型，请指定 desktop 或者 server 。	否	是	是
--os-initrd	字符串	到一个指定内核所使用的 initrd 镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	否	是	是
--display-monitors	整数	定义可用的显示数量。	否	是	是
--display-single_qxl_pci	布尔值	设定为 true 以便使用单一虚拟 PCI 设备驱动多台显示器。	否	是	是
--display-type	字符串	定义显示类型，请指定 spice 或 vnc 。	否	是	是
--display-allow_override	布尔值	设为 true 以便可以覆盖虚拟机控制台设置。	否	是	是
--display-smartcard_enabled	布尔值	设为 true 以便启用智能卡功能。	否	是	是
--display-file_transfer_enabled	布尔值	设为 true 以便启用 SPICE 文件传输。	否	是	是
--display-copy_paste_enabled	布尔值	设为 true 以便启用 SPICE 剪切版复制和粘贴。	否	是	是
--display-keyboard_layout	字符串	定义虚拟机键盘类型。只有使用 VNC 协议时这个选项才有效。请指定类似 en-US 的格式。	否	是	是
--os-cmdline	字符串	内核需要使用的内核命令行参数。这个选项支持直接引导一个 Linux 内核，而不需要使用 BIOS bootloader。	否	是	是
--cpu-topology-cores	整数	虚拟机可用的 CPU 内核数量。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--cpu-architecture	字符串	定义 CPU 架构。请指定 x86_64 、 ppc64 或者 undefined 。	否	是	是
--memory	长整数	虚拟机的内存大小（以字节为单位）	否	是	是
--memory-policy-guaranteed	长整数	为了保证虚拟机的运行，在主机上最少需要保证具有的内存数量（以字节为单位）。	否	是	是
--memory-policy-ballooning	布尔值	设为 true 以便启用内存布尔设备。	否	是	是
--high-availability-priority	整数	设置虚拟机高可用性的优先级（如迁移和重启的顺序）。	否	是	是
--high-availability-enabled	布尔值	定义是否为该虚拟机启用高可用性。	否	是	是
--domain-name	字符串	虚拟机的域名。	否	是	是
--description	字符串	虚拟机的描述。	否	是	是
--comment	字符串	虚拟机的注释。	否	是	是
--stateless	布尔值	如果虚拟机处于无状态的（stateless），则将这个参数设为 true 。无状态的虚拟机会在系统启动时对磁盘做一个快照，并在关机时删除对系统所做的所有改动。这意味着重启后会丢失对系统所做的改动。	否	是	是
--permissions-clone	布尔值	设为 true 以便将源虚拟机权限复制到模板中。	否	是	是
--delete-protected	布尔值	设为 true 以便无法删除使用此模板创建的虚拟机。	否	是	是
--sso-methods-method	集合	定义要使用的单点登录方法。例如： --sso-methods-method method.id=GUEST_AGENT 。	否	是	是
--rng-device-rate-bytes	整数	指定每个周期中允许消耗的字节数。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
--rng-device-rate-period	整数	以毫秒为单位指定周期时间。如果指定这个选项，则必须同时指定 --rng-device-rate-bytes。	否	是	是
--rng-device-source	字符串	随机数产生器的资源。请指定 random 或者 hwrng 。	否	是	是
--console-enabled	布尔值	设为 true 以便启用 VirtIO 控制台设备功能。	否	是	是
--cpu-mode	字符串	定义 CPU 模式。请指定 custom 、 host_model 或者 host_passthrough 。	否	是	是
--cpu-topology-sockets	整数	虚拟机可用的 CPU 插槽数量。	否	是	是
--cpu-shares	整数	相比其他虚拟机可要求的 CPU 资源水平。例如：低优先级虚拟机使用 512 ，中等优先级虚拟机使用 1024 ，而高优先级虚拟机使用 2048 。	否	是	是
--placement-policy-affinity	字符串	每台虚拟机的迁移关联（migration affinity），请指定 migratable 、 user_migratable 或者 pinned 。	否	是	是
--placement-policy-host-id name	字符串	指定迁移关联（migration affinity）中优先选择的主机。	否	是	是
--origin	字符串	虚拟机原形。请指定 rhev 、 vmware 或者 xen 。	否	是	是
--os-kernel	字符串	到虚拟机引导内核镜像的路径。这个选项支持直接引导 Linux 内核，而不需要 BIOS bootloader。	否	是	是
--disks-clone	布尔值	指定是否从 template 中克隆磁盘。	否	是	否
--disks-disk	集合	请参考附加到该虚拟机的磁盘。	否	是	是
--tunnel-migration	布尔值	设为 true 启用使用 libvirt 守护进程进行数据传输的功能。隧道传输使用更强大的加密算法，但会增加传输过程中的数据负载。	否	是	是
--migration-downtime	整数	以毫秒为单位定义在实时迁移过程中虚拟机可以停机的最长时间。	否	是	是
--virtio-scsi-enabled	布尔值	设为 true 以便在该虚拟机中添加 VirtIO 控制台设备。	否	是	是

名称	类型	描述	必需的	用户可创建	用户可更新
-- soundcard_ enabled:	布尔值	设为 true 以便启用声卡。	否	是	是
-- payloads- payload	集合	指定在虚拟机引导时发送到虚拟机的内容。	否	是	是
-- initializa tion- configurat ion-type	字符串	定义虚拟机格式。只接受 ovf 。	否	是	是
-- initializa tion- configurat ion-data	字符串	这个参数必须与 --initialization-configuration-type 参数匹配。只接受 ovf 。	否	是	是
--cpu- cpu_tune- vcpu_pin	集合	定义将虚拟机的哪个虚拟 CPU 与主机的物理 CPU 固定 (pin) 。	否	是	是
-- serial_num ber-policy	字符串	定义虚拟机模板的序列号策略。请指定 host 、 vm 或者 custom 。如果使用 custom ，则还要使用 --serial_number-value 定义序列号。	否	是	是
-- serial_num ber-value	字符串	定义虚拟机模板的序列号。	否	是	是
--bios- boot_menu- enabled	布尔值	设为 true 以便启用引导菜单。	否	是	是
-- numa_tune_ mode	字符串	定义如何在 NUMA 主机中为域进程分配内存。请指定 interleave 、 strict 或者 preferred 。如果没有给定任何值，则该参数默认使用 strict 。	否	是	是
-- cpu_profil e-id	字符串	定义要使用的 CPU 配置文件。请使用 list cpuprofiles 命令查询 CPU 配置文件 ID 的完整列表。	否	是	是
--expect	'201-created'	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否
-- correlatio n_id	字符串	一个跨系统日志操作的标识。如果客户端没有定义这个标识，系统将会自动产生一个。	否	是	否

--os-boot 参数是使用下表中子参数的集合。

表 4.59. --os-boot parameters

名称	类型	描述
<code>boot.dev</code>	字符串	虚拟机模板的引导设备。请指定 cdrom 、 hd 或者 network 。

`--custom_properties-custom_property` 参数是一个使用下表中的子参数的集合。

表 4.60. `--custom_properties-custom_property` 参数

名称	类型	描述
<code>custom_property.name</code>	字符串	自定义属性名。
<code>custom_property.value</code>	字符串	自定义属性的值。

`--vm-disks-disk` 参数是使用下表中子参数的集合。

表 4.61. `--sso-methods-method parameters`

名称	类型	描述
<code>method.id</code>	字符串	要使用的单点登录方法。例如： GUEST_AGENT 。

`--disks-disk` 参数是使用下表中子参数的集合。

表 4.62. `--disks-disk parameters`

名称	类型	描述
<code>disk.id</code>	字符串	虚拟磁盘的标识。
<code>storage_domains.storage_domain</code>	集合	为磁盘存储域定义一组子参数。

`--payloads-payload` 参数是一个使用下表中的子参数的集合。

表 4.63. `--payloads-payload` 参数

名称	类型	描述
<code>payload.type</code>	字符串	payload 的分发类型。请指定 cdrom 或 floppy 。
<code>payload.file.name</code>	字符串	payload 文件在虚拟机引导文件系统上的文件名和位置。
<code>payload.file.content</code>	字符串	payload 文件的内容。

`--cpu-cpu_tune-vcpu_pin` 参数是一个使用下表中的子参数的集合。

表 4.64. `--cpu-cpu_tune-vcpu_pin`

名称	类型	描述
<code>vcpu_pin.vcpu</code>	整数	分配的虚拟 CPU。
<code>vcpu_pin.cpu_set</code>	字符串	主机上的物理 CPU。

例 4.45. 创建一个新虚拟机

```
[RHEVM shell (connected)]# add vm --name MyVM --template-name Blank --
cluster-name Default --memory 536870912
```

例 4.46. 更新一个虚拟机。

```
[RHEVM shell (connected)]# update vm MyVM --memory 1073741824
```

例 4.47. 删除一个虚拟机

```
[RHEVM shell (connected)]# remove vm MyVM
```

下表列出了针对一个虚拟机资源的操作。

表 4.65. 虚拟机操作

操作	描述
start	启动一个虚拟机。
stop	停止一个虚拟机。
shutdown	关闭虚拟机。
suspend	挂起一个虚拟机。
detach	把虚拟机从虚拟机池中移出。
migrate	将虚拟机迁移到另一台主机。
cancelmigration	停止正在进行的迁移。
export	将虚拟机导出到导出存储域
move	把虚拟磁盘移到另一个存储域中。
ticket	为控制台访问创建一个 ticket。
logon	使用第三方应用程序启用用户登录以便访问控制台。

4.23. vmpool

vmpool1 资源类型是所有虚拟机池资源在 Red Hat Virtualization 环境中的组合。

表 4.66. 虚拟机池参数

名称	类型	描述	必需的	用户可创建	用户可更新
--cluster-id name	字符串	虚拟机池所在集群的标识。	是	是	是
--template-id name	字符串	虚拟机池中的虚拟机所基于的模板标识。	是	是	是
--name	字符串	虚拟机池的名称。	是	是	是
--size	整数	虚拟机池中的虚拟机数量。	是	是	是

例 4.48. 创建一个新虚拟机池

```
[RHEVM shell (connected)]# add vmpool --cluster-name MyCluster --template-name MyTemplate --name MyPool --size 3
```

例 4.49. 更新一个虚拟机池

```
[RHEVM shell (connected)]# update vmpool MyPool --size 4
```

例 4.50. 删除一个虚拟机池

```
[RHEVM shell (connected)]# remove vmpool MyPool
```

4.24. vnicprofile

vnicprofile 资源类型是所有 VNIC（virtual network interface controller - 虚拟网络接口控制器）配置集资源在 Red Hat Virtualization 环境中的组合。

表 4.67. VNIC 配置集参数

名称	类型	描述	必需的	用户可创建	用户可更新
--name	字符串	VNIC 配置集的名称。	是	是	是
--network-id	字符串	使用这个配置集的逻辑网络的标识。	是	否	否
--correlation-id	字符串	一个跨系统日志操作的标识。如果客户端没有定义这个标识，系统将会自动产生一个。	否	是	否
--description	字符串	配置集的描述。	否	是	是
--expect	'201- create '	在所期望的 HTTP 头返回前，请求以异步的方式进行处理。它对需要长时间来执行的任务非常有用，可以防止在任务完成前返回成功信号。	否	否	否
--custom-properties-custom-property	集合	传递给脚本的一组用户自定义的环境变量。	否	是	是
--port-mirroring	布尔值	指定这个配置集是否使用端口镜像功能。它的有效值包括 True 和 False ，默认值是 False 。	否	否	否

--custom-properties-custom-property 参数是一个使用下表中的子参数的集合。

表 4.68. --custom-properties-custom-property 参数

名称	类型	描述
custom_property.name	字符串	自定义属性名。

名称	类型	描述
<code>custom_property.value</code>	字符串	自定义属性的值。

例 4.51. 创建一个新的 VNIC 配置集

```
[RHEVM shell (connected)]# add vnicprofile --name Gold --network-id  
08305a2f-6952-4999-9646-c16137dc6d42
```

例 4.52. 更新一个 vnic 配置集

```
[RHEVM shell (connected)]# update vnicprofile Gold --port_mirroring true
```

例 4.53. 删除一个 vnic 配置集

```
[RHEVM shell (connected)]# remove vnicprofile Gold
```

第 5 章 CLI 查询

5.1. 查询语法

CLI 的 `list` 命令使用 `--query` 参数来执行服务器端的查询，这个查询的语法规则和 Red Hat Virtualization Manager 搜索查询语言的语法相同：

表 5.1. 搜索查询示例

集合	搜索条件	结果
hosts	<code>vms.status=up</code>	显示所有正在运行虚拟机的主机列表，这些虚拟机的状态需要为 <code>up</code> 。
vms	<code>domain=qa.company.com</code>	显示在指定域中运行的所有虚拟机列表。
vms	<code>users.name=mary</code>	显示属于用户 <code>mary</code> 的所有虚拟机列表。
events	<code>severity>normal sortby time</code>	显示所有 <code>severity</code> 的值高于 <code>normal</code> 的事件，并以时间顺序排序。
events	<code>severity>normal sortby time desc</code>	显示所有 <code>severity</code> 的值高于 <code>normal</code> 的事件，并以时间进行倒序排序。

5.2. 通配符

在搜索查询中可以使用星号作为通配符。

例 5.1. 在搜索查询中使用通配符

```
[RHEVM shell (connected)]# list vms --query "name=vm*"
```

这个查询的结果将包括所有名称以 `vm` 开头的虚拟机，如 `vm1`、`vm2`、`vma` 和 `vm-webserver`。

例 5.2. 在搜索查询中使用通配符

```
[RHEVM shell (connected)]# list vms --query "name=v*1"
```

这个查询的结果将包括所有名称以 `v` 开头并以 `1` 结束的虚拟机，如 `vm1`、`vr1` 和 `virtualmachine1`。