



Red Hat Virtualization 4.1

RHEVM Shell Guide

Installing and Using the Command Line Shell for Red Hat Virtualization

Red Hat Virtualization 4.1 RHEVM Shell Guide

Installing and Using the Command Line Shell for Red Hat Virtualization

Red Hat Virtualization Documentation Team

Red Hat Customer Content Services

rhev-docs@redhat.com

Legal Notice

Copyright © 2018 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). 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.

Abstract

This guide contains information for installing and using the Red Hat Virtualization Manager Command Line Shell.

Table of Contents

CHAPTER 1. USING THE CLI	3
1.1. INSTALLING THE CLI	3
1.2. TLS/SSL CERTIFICATION	3
1.3. .OVIRTSHELLRC CONFIGURATION	5
1.4. RUNNING THE CLI	6
1.5. INTERACTING WITH THE CLI	8
1.6. COLLECTIONS	10
CHAPTER 2. QUICK START EXAMPLE	12
2.1. CREATING A BASIC VIRTUALIZATION ENVIRONMENT WITH THE CLI	12
CHAPTER 3. COMMANDS	17
3.1. CONNECTING TO RHEVM	17
3.2. RESOURCES	18
3.3. OTHER COMMANDS	23
CHAPTER 4. RESOURCE TYPES	29
4.1. BRICK	29
4.2. CDROM	29
4.3. CLUSTER	30
4.4. DATACENTER	32
4.5. DISK	34
4.6. GLUSTERVOLUME	36
4.7. GROUP	38
4.8. HOST	38
4.9. NETWORK	41
4.10. NIC	43
4.11. PERMISSION	46
4.12. PERMIT	48
4.13. QUOTAS	48
4.14. ROLE	49
4.15. SNAPSHOT	50
4.16. STATISTIC	50
4.17. STORAGECONNECTION	51
4.18. STORAGEDOMAIN	53
4.19. TAG	56
4.20. TEMPLATE	57
4.21. USER	64
4.22. VM	64
4.23. VMPOOL	73
4.24. VNICPROFILE	73
CHAPTER 5. CLI QUERIES	76
5.1. QUERY SYNTAX	76
5.2. WILDCARDS	76

CHAPTER 1. USING THE CLI

The Red Hat Virtualization suite features a command line interface (CLI). This CLI provides users with a means to connect to Red Hat Virtualization Manager outside of the standard web interface. The CLI also contains a scripting system, which helps system administrators perform periodic maintenance or repetitive tasks on their virtualization environment via client machines.



IMPORTANT

Version 4 of the Red Hat Virtualization Manager includes a build of version 3.6 of the CLI for Red Hat Enterprise Linux 7. This version of the CLI uses version 3.6 of the Python SDK, and the version 3 compatibility mode of the REST API.

1.1. INSTALLING THE CLI

Install the Red Hat Virtualization CLI to a client machine:

1. Log into the client machine as the **root** user.
2. Register your system with the Content Delivery Network, entering your Customer Portal user name and password when prompted:

```
# subscription-manager register
```

3. Find the **Red Hat Virtualization** subscription pool and note down the pool ID.

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

4. Use the pool identifiers located in the previous step to attach the **Red Hat Virtualization** entitlement to the system:

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

5. Enable the required repository:

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

6. Install the CLI package and dependencies:

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

1.2. TLS/SSL CERTIFICATION

The Red Hat Virtualization Manager API requires Hypertext Transfer Protocol Secure (HTTPS) ^[1] for secure interaction with client software, such as the Manager's SDK and CLI components. This involves a process of obtaining a certificate from the Red Hat Virtualization Manager and importing it into the certificate store of your client.



IMPORTANT

Obtain your certificate from the Red Hat Virtualization Manager using a secure network connection.

Procedure 1.1. Obtaining a Certificate

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

1. **Method 1** - Use a command line tool to download the certificate from the Manager. Examples of command line tools include **cURL** and **Wget**, both of which are available on multiple platforms.

- a. If using **cURL**:

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

- b. If using **Wget**:

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

2. **Method 2** - Use a web browser to navigate to the certificate located at:

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

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

- a. If the browser downloads the certificate: save the file as **rhvm.cer**.

If the browser imports the certificate: export it from the browser's certification options and save it as **rhvm.cer**.

3. **Method 3** - Log in to the Manager, export the certificate from the truststore and copy it to your client machine.

- a. Log in to the Manager as the **root** user.

- b. Export the certificate from the truststore using the Java **keytool** management utility:

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

This creates a certificate file called **rhvm.cer**.

- c. Copy the certificate to the client machine using the **scp** command:

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


Each of these methods results in a certificate file named `rhvm.cer` on your client machine. An API user imports this file into the certificate store of the client.

Procedure 1.2. Importing a Certificate to a Client

- Importing a certificate to a client relies on how the client itself stores and interprets certificates. This guide contains some examples on importing certificates. For clients not using Network Security Services (NSS) or Java KeyStore (JKS), see your client documentation for more information on importing a certificate.

1.3. .OVIRTSHHELLRC CONFIGURATION

The `.ovirtshellrc` file is a configuration file that is automatically created and populated when the user first connects to the `ovirt-shell`. It allows users to configure options for connecting to the Red Hat Virtualization environment. The `.ovirtshellrc` file is located by default in `/home/[user name]/.ovirtshellrc`.

The configuration information of the `.ovirtshellrc` file falls under two section headings, `[cli]` and `[ovirt-shell]`. These headings are necessary for the configuration file to be parsed.

Table 1.1. `[cli]` Parameters

Name	Type	Description
<code>autoconnect</code>	boolean	Toggles whether to automatically connect to an <code>ovirt-shell</code> session. The status is either True or False .
<code>autopage</code>	boolean	Toggles pagination in the shell. The status is either True or False .

Table 1.2. `[ovirt-shell]` Parameters

Name	Type	Description
<code>username</code>	string	User name to be used to log in.
<code>timeout</code>	integer	Specifies timeout for requests. The default is -1.
<code>extended_prompt</code>	boolean	Toggles the extended prompt option, which displays the hostname in the shell command prompt.
<code>url</code>	string	The address of the Red Hat Virtualization environment.
<code>insecure</code>	boolean	Toggles CA certificate requirement. The status is either True or False .
<code>renew_session</code>	boolean	Toggles automatic renewal of the session upon expiration. The status is either True or False .
<code>filter</code>	boolean	Toggles object filtering. Object filtering allows users to fetch objects according to their permissions. Only admin roles can toggle filtering off. The status is either True or False .

Name	Type	Description
<code>session_timeout</code>	integer	Specifies timeout (in minutes) for authentication session. Must be a positive number.
<code>ca_file</code>	string	Specifies the server CA certificate to use.
<code>dont_validate_cert_chain</code>	boolean	Toggles validation of server CA certificate. The status is either True or False .
<code>key_file</code>	string	Specifies client PEM key-file.
<code>password</code>	string	Password to be used for user name.
<code>cert_file</code>	string	Specifies client PEM cert-file.

1.4. RUNNING THE CLI

Start the CLI application with the following command:

```
# ovirt-shell
```

This `ovirt-shell` application is an interactive shell for Red Hat Virtualization environments.

The URL, user name, certificate authority file, and password for connecting to the Red Hat Virtualization Manager can be configured in the `.ovirtshellrc` file. The `ovirt-shell` command uses the parameters in this file to connect to the Manager, so that the user does not need to specify options each time.

Alternatively, users can connect automatically to Red Hat Virtualization Manager using the following additional options.

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

Ensure to replace the following values:

- `server` - The hostname or IP Address of the Red Hat Virtualization Manager. The CLI connects to the Red Hat Virtualization Manager via the REST API.
- `user@domain` - The user name and directory service domain for the user logging into Red Hat Virtualization Manager.
- `certificate` - The path name of the Certificate Authority file.

The shell will prompt you for the password, and, if not already provided, the username and the URL for the Red Hat Virtualization Manager.

**NOTE**

You do not need to specify additional options if you have configured your user name, password, URL, and certificate authority file in the `.ovirtshellrc` file.

**NOTE**

The certificate is the only obligatory option as the others used in this example will be prompted by the shell. Instead of specifying the certificate you can use the `'--insecure'` option to connect without certification, however this is not recommended as it may allow man-in-the-middle (MITM) attackers to spoof the identity of the server.

Options for ovirt-shell**-h, --help**

Show help for `ovirt-shell`.

-d, --debug

Enables debugging.

-l URL, --url=URL

Specifies the API entry point URL.

-u USERNAME, --username=USERNAME

Connect as this user.

-K KEY_FILE, --key-file=KEY_FILE

Specify key file.

-C CERT_FILE, --cert-file=CERT_FILE

Specify certificate file.

-A CA_FILE, --ca-file=CA_FILE

Specify server Certificate Authority file.

-l, --insecure

Allow the CLI to connect via SSL without certification. Use this option with caution because it can allow man-in-the-middle (MITM) attackers to spoof the identity of the server.

-F, --filter

Enable filtering based upon user permissions.

-P PORT, --port=PORT

Specify port.

-T TIMEOUT, --timeout=TIMEOUT

Specify timeout.

-c, --connect

Automatically connect.

-e, --extended-prompt

Enables the extended prompt option for the shell. This option displays the hostname of the environment in the command prompt. Default is 'false'.

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

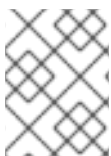
Connects to the Manager to execute only the given commands, in the form of "*command resource;command resource*" and prints the output to STDIO.

-f FILE, --file=FILE

Read commands from FILE instead of stdin.

--kerberos

Use a valid Kerberos ticket to authenticate connection to the shell.

**NOTE**

Users with a non-interactive shell are able to connect to the Red Hat Virtualization Manager from within the shell, where the `--password` option can be used.

1.5. INTERACTING WITH THE CLI

The CLI is an interactive shell for controlling your Red Hat Virtualization environment from the command line. Type the required command and any additional parameters.

Example 1.1. Entering a shell command

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

To support the construction of command and parameter combinations, the CLI includes the functionality to list and automatically complete commands and parameters by pressing the **TAB** key twice, similar to the `bash` shell.

Example 1.2. Listing and automatic completion of commands and parameters

Press double **TAB** at a blank prompt to list all available commands.

```
[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
```

Choose a command and press double **TAB** to view the next set of available parameters for the command. For the `add` command, this lists all resources.

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

Double **TAB** also completes commands and parameters.

```
[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
```

Note that the double **TAB** also automatically formats **na** to the **--name** parameter, including the prefix.

If the incomplete parameter matches multiple parameters, double **TAB** lists them.

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

The CLI provides functions to run Linux commands using either the **shell** command or the bang (!) character.

Example 1.3. Running Linux shell commands

Use the **shell** command:

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

Or use the bang (!) character:

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

Similar to the Linux shell, the CLI can pipe data to other commands and sources.

Example 1.4. Piping CLI commands

Pipe CLI data to a Linux shell command:

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

```

name           : ExampleEngineering
description    : An Example description
name           : BestExampleVM

```

Pipe CLI data to a file:

```

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

```

The CLI also contains an online help system to provide descriptions and syntax for each command via the `help` command.

Example 1.5. Using online help for the `show` command

```

[RHEVM shell (connected)]# help show

```

You can also connect to the Manager from the Linux shell to execute specific commands, in the form of "*command resource*" and print them to STDOUT

Example 1.6. Connecting to the Manager to execute specific commands

Use the `--execute` or `-E` parameter to connect to the Manager to execute the specific commands.

```

# 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-9491-59faff89b113
name         : Gluster

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

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

```

1.6. COLLECTIONS

Some command parameters require a collection. A collection is a set of sub-parameter data. Collections are defined using the following syntax.

-

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

Sub-parameters for collections are listed after resource parameter listings on each resource page.

[1] HTTPS is described in [RFC 2818 HTTP Over TLS](#).

CHAPTER 2. QUICK START EXAMPLE

2.1. CREATING A BASIC VIRTUALIZATION ENVIRONMENT WITH THE CLI

This chapter provides an example to demonstrate the CLI's ability to add a virtual machine within a basic Red Hat Virtualization environment. This example uses the following prerequisites:

- A networked and configured Red Hat Enterprise Linux host for use as a hypervisor;
- A networked and configured NFS storage server with two shares:
 - `/exports/data` - The data storage domain; and
 - `/exports/iso` - The ISO storage domain.
- A networked and configured Red Hat Virtualization Manager;
- An installation of the CLI on either the Red Hat Virtualization Manager or a client machine; and,
- An ISO file containing a desired virtual machine operating system to install. This chapter uses Red Hat Enterprise Linux Server 6 for our installation ISO example.



NOTE

Red Hat Virtualization Manager generates a globally unique identifier (GUID) for each resource. Identifier codes in this example might appear different to the identifier codes in your Red Hat Virtualization environment.

Procedure 2.1. Quick Start Example

1. Load the CLI shell and connect to your Red Hat Virtualization Manager.

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

2. List all data centers in the environment. This example uses the **Default** data center.

```
[RHEVM shell (connected)]# list datacenters  
  
id           : 5e3b55d8-c585-11e1-a7df-001a4a400e0d  
name        : Default  
description: The default Data Center
```

3. List all host clusters and note down the relevant cluster ID or cluster name, which will be required when adding the host and for creating a virtual machine. This example uses the **Default** cluster to group resources in your Red Hat Virtualization environment.

```
[RHEVM shell (connected)]# list clusters  
  
id           : 99408929-82cf-4dc7-a532-9d998063fa95  
name        : Default  
description: The default server cluster
```


4. List all CPU profiles and note down the relevant CPU profile ID, which will be required when creating a virtual machine. This example uses the `Default` CPU profile.

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

id           : 0000001a-001a-001a-001a-00000000035e
name        : Default
```

5. List all logical networks with the `show-all` option to view the details of the logical networks in the environment. Red Hat Virtualization Manager creates a default logical network called `ovirtmgmt` for management traffic. This example uses the `ovirtmgmt` logical network on the `Default` data center.

```
[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
```

Note the `data_center-id` value matches the `id` for the `Default` data center.

6. Add the Red Hat Enterprise Linux host to the virtualization environment as a new hypervisor. The host is activated automatically.

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

7. Add an NFS share as a data storage domain by creating, attaching, and activating the NFS share. An NFS data storage domain is an exported NFS share attached to a data center. It provides storage for virtual machines. Ensure to substitute `storage-address` and `storage-path` with the correct values for the NFS server.

- a. Create a data storage domain.

```
[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. Verify that the created storage domain is available. The creation process might take several minutes. Once the `status-state` is `unattached`, you can proceed to the next step.

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

id           : xxxx
name        : DataStorage
```

```

master          : False
status-state    : unattached
...

```

- c. Attach the data storage domain to the data center. The storage domain is activated automatically.

```

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

```



NOTE

If the storage domain is not activated, activate it manually using the following command:

```

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

```

8. Add an NFS share as the ISO storage domain by creating, attaching, and activating the NFS share. An NFS ISO storage domain is an exported NFS share attached to a data center. It provides storage for DVD/CD-ROM ISO and virtual floppy disk (VFD) image files. Ensure to substitute **storage-address** and **storage-path** with the correct values for the NFS server.

- a. Create an ISO storage domain.

```

[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. Verify that the created storage domain is available. The creation process might take a while. Once the **status-state** is **unattached**, you can proceed to the next step.

```

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

```

- c. Attach the ISO storage domain to the data center. The storage domain is activated automatically.

```

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

```

9. Create a new virtual machine.

```

[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

```

- Use the `add nic` command to add a new network interface. Add the `vm-identifier` option to attach the interface as a sub-resource of `MyVM` and a `network-name` option to connect to the `ovirtmgmt` network.

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

- Use the `add disk` command to add a new virtual hard disk. Add the `vm-identifier` option to attach the disk as a sub-resource of `MyVM`.

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

- On the Manager, upload ISO images to the `ISOStorage` domain for the virtual machines to use. Red Hat Virtualization Manager provides an ISO uploader tool that ensures images are uploaded into the correct directory path with the correct user permissions.

```
# 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):
```

- In the CLI shell, use the `list files` command to list the available ISO files in the storage domain.

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

- Add a virtual CD-ROM drive for your installation media. Add the `vm-identifier` option to attach the CD-ROM as a sub-resource of `MyVM`.

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

- Start the virtual machine. The virtual environment is complete and the virtual machine contains all necessary components to function.

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

Note the use of the `vm-os-boot` option. This changes the boot device to `cdrom` for this initial boot session. After installation, the virtual machine restarts and restores the boot device back to `hd`.

- Use the `list events` with an additional `query` option to display specific event types. The `start` action for the virtual machine adds several entries in the `events` collection.

```
[RHEVM shell (connected)]# list events --query "type=153"
id          : 105
description: MyVM was started by admin (Host: MyHost).
```

-

The "**type=153**" query refers to events where a user starts a virtual machine.

17. Use the **show event** command to display comprehensive details of an event. This command can be used to show events by **type**, **name**, and **id**.

```
[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. Access your virtual machine with the **console** command.

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



IMPORTANT

Ensure your client machine has a console application installed to match the virtual machine's **display-type**. Protocols available include **SPICE** (default) and **VNC**.

CHAPTER 3. COMMANDS

3.1. CONNECTING TO RHEVM

3.1.1. Connect to RHEVM (connect)

The `connect` command connects to Red Hat Virtualization Manager. The URL, user name, certificate authority file, and password for connecting to the Red Hat Virtualization Manager can be configured in the `.ovirtshellrc` file. The `connect` command uses the parameters in this file to connect to the Manager, so that the user does not need to specify options each time.

Syntax

```
connect [options]
```



NOTE

You do not need to specify additional options if you have configured your user name, password, URL, and certificate authority file in the `.ovirtshellrc` file.

Table 3.1. Options for connect

Option	Description	Required
<code>--url</code>	The URL to the Red Hat Virtualization Manager's REST API. This takes the form of <code>https://[server]/ovirt-engine/api</code> .	Yes
<code>--username</code>	The user name and directory service domain of the user attempting access to the Red Hat Virtualization Manager. This takes the form of <code>[username]@[domain]</code> .	Yes
<code>--password</code>	The password for the user attempting access to the Red Hat Virtualization Manager.	Yes
<code>--key-file</code>	The key file for connection via SSL.	No
<code>--cert-file</code>	The certificate file for connection via SSL.	No
<code>--ca-file</code>	The certificate authority file for connection via SSL.	Yes, unless <code>--insecure</code> is used
<code>--insecure</code>	Allow the CLI to connect via SSL without certification. Use this option with caution because it can allow man-in-the-middle (MITM) attackers to spoof the identity of the server.	Yes, but only if no certificate authority is provided

Option	Description	Required
<code>--filter</code>	Enable filtering based upon user permissions.	No
<code>--port</code>	The port number for connection to the REST API, if not specified as part of the <code>--url</code> .	No
<code>--timeout</code>	The timeout period for connection.	No

Example 3.1. Example for connect when `.ovirtshellrc` is not configured

```
[RHEVM shell (disconnected)]# connect --url
"https://rhev.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)]#
```



NOTE

Instead of specifying the certificate you can use the `'--insecure'` option to connect without certification, however this is not recommended as it may allow man-in-the-middle (MITM) attackers to spoof the identity of the server.

3.1.2. Disconnect from RHEVM (disconnect)

The `disconnect` command disconnects from Red Hat Virtualization Manager.

Syntax

`disconnect`

Example 3.2. Example for disconnect

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

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

[RHEVM shell (disconnected)]#
```

3.2. RESOURCES

3.2.1. List Resources in a Collection (list)

Use the `list` command to display all resources of a specific type. Lists also include optional search queries to filter results.

Syntax

```
list [collection] [options]
```

Table 3.2. `list` standard options

Option	Description
<code>--show-all</code>	Displays all non-empty properties for each listed resource. Without this option, only the <code>id</code> , <code>name</code> and <code>description</code> properties display.
<code>--query [QUERY]</code>	Filters the list using a server-side query based upon Red Hat Virtualization Manager query language.
<code>--kwargs [QUERY]</code>	Filters the list using a client-side query.
<code>--case_sensitive true false</code>	Match search queries using case sensitivity.
<code>--max</code>	Maximum number of results for display.



NOTE

Options specific to resource types are listed in the definition pages for each resource type.

Example 3.3. Examples for `list`

List virtual machines:

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

List virtual machines with all properties listed:

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

List virtual machines with a status of 'up':

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

List users that match the specified user name across all domains with the use of a wildcard:

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

Get help with `list` search syntax:

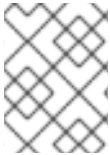
```
[RHEVM shell (connected)]# list --help
```

3.2.2. Show a Resource (show)

Use the `show` command to display resource properties.

Syntax

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



NOTE

Options specific to resource types are listed in the definition pages for each resource type.

Example 3.4. Examples for show

Show virtual machines based upon `id`:

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

Show virtual machines based upon `name`:

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

3.2.3. Add a Resource (add)

Use the `add` command to add a new resource.

Syntax

```
add [resource] [options]
```



NOTE

Options specific to resource types are listed in the definition pages for each resource type.

Example 3.5. Examples for add

Create a virtual machine:

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

Create a user:

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


The `add` command can be made synchronous (if supported) by using the `expect` option:

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

3.2.4. Update a Resource (update)

Use the `update` command to modify an existing resource.

Syntax

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



NOTE

Options specific to resource types are listed in the definition pages for each resource type.

Example 3.6. Examples for update

Update a virtual machine:

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

3.2.5. Remove a Resource (remove)

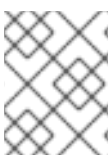
Use the `remove` command to remove a resource.

Syntax

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

Table 3.3. remove standard options

Option	Description
<code>--async</code>	Perform an asynchronous removal of the resource.
<code>--force</code>	Perform a force remove of the resource. This removes all database entries and associations for a particular resource. This action applies only to datacenter and vm resources.



NOTE

Options specific to resource types are listed in the definition pages for each resource type.

Example 3.7. Examples for remove

Remove a virtual machine:

```
■
```

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

Asynchronous removal of a virtual machine:

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

Force remove virtual machine:

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

3.2.6. Perform Action on a Resource (action)

Use the `action` command to perform a special function relevant to resource type.

Syntax

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



NOTE

Options specific to resource actions are listed in the definition pages for each resource type.

Example 3.8. Examples for action

Start a virtual machine

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

Stop a virtual machine:

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

The `action` command can be made synchronous (if supported) by using the `async` option:

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

3.2.7. Using Sub-Resources (--RESOURCE-identifier)

Certain resources act as sub-resources of other resources. This means there is a dependent relationship between the sub-resource and its parent resource. Use the `--RESOURCE-identifier` [*name*] option, where *RESOURCE* is the parent resource type, to target a sub-resource part of a parent resource.

Example 3.9. Examples for creating sub-resources with add

Create a NIC on a virtual machine:

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

Note the use of the `--vm-identifier RHEL6-Server` option. This adds a NIC on the RHEL6-Server virtual machine.

Create a storage disk on a virtual machine:

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

Note the use of the `--vm-identifier RHEL6-Server` option. This adds a storage disk on the RHEL6-Server virtual machine.

3.3. OTHER COMMANDS

3.3.1. End of File (EOF)

Use the EOF command to leave the CLI shell using a `Ctrl+D` sequence.

Syntax

EOF

Example 3.10. Example for EOF

Leave the CLI shell:

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

3.3.2. List System Capabilities (capabilities)

Use the `capabilities --features` command to list all version capabilities and new features of the current version.

Syntax

`capabilities --features`

Example 3.11. Example for capabilities

List system capabilities of the current version:

```
[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. Clear the Screen (clear)

Use the `clear` command to clear the CLI screen.

Syntax

`clear`

Example 3.12. Example for clear

Clear the screen:

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

3.3.4. Connect to VM (console)

Use the `console` command to open a graphical console to a virtual machine. This command opens either an external VNC or SPICE client based upon the virtual machine's `display-type` parameter.

Syntax

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

Example 3.13. Example for console

Open graphical console to a virtual machines:

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

3.3.5. Print Input (echo)

Use the `echo` command to print input to the screen. Use the `$out` variable to print the last shell command output.

Syntax

`echo [input]`

Example 3.14. Example for echo

Print input:

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

Print last output:

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

3.3.6. Exit from the CLI (exit)

Use the `exit` command to leave a CLI.

Syntax

```
exit
```

Example 3.15. Example for exit

Leave the CLI:

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

3.3.7. Run a Script (file)

Use the `file` command to run a CLI script file. A script is a plain text file that contains a list of commands for execution.

Syntax

```
file [file-location]
```

Example 3.16. Example for file

Run a script file:

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

3.3.8. Show Help (help)

Use the `help` command displays help for CLI command and resource combinations.

Syntax

```
help [command] [resource] [options]
```

Example 3.17. Examples for help

Get CLI help:

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

Get help for the `add` command:

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

Get help for the `add` command on the `vm` resource type:

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

3.3.9. Display the User Command History (`history`)

Use the `history` command to display the history of user commands for the CLI shell.

Syntax

`history`

Example 3.18. Example for `history`

Display the user command history:

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

Example 3.19. Example for `history --first`

Display the first specified entries in the user command history with the `--first n` parameter:

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

Example 3.20. Example for `history --last`

Display the last specified entries in the user command history with the `--last n` parameter:

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

3.3.10. Show CLI Information (`info`)

Use the `info` command to display environment connection details and version information.

Syntax

`info`

Example 3.21. Example for `info`

View CLI information:

```
[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. Test Connection (ping)

Use the `ping` command tests the connection to your Red Hat Virtualization Manager. The command retrieves a remote resource and ensures the URL, user name and password for the connection are correct.

Syntax

`ping`

Example 3.22. Example for ping

Test your connection:

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

3.3.12. Run a Shell Command (shell)

Use the `shell` command to run a command from the Linux shell. This command helps with performing file management tasks in conjunction with the Red Hat Virtualization Manager shell.

Syntax

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

Example 3.23. Examples for shell

List files in current working directory:

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

Create a file:

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

Copy a file:

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

**NOTE**

The CLI offers an alternative to the `shell` using the bang (!) character. For example:

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

3.3.13. Show Last Status (`status`)

Use the `status` command to display the most recently run command status.

Syntax

`status`

Example 3.24. Example for `status`

View the last status:

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

3.3.14. Show System Summary (`summary`)

Use the `summary` command to display a summary of the system status.

Syntax

`summary`

Example 3.25. Example for `summary`

Display system status:

```
[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
```


CHAPTER 4. RESOURCE TYPES

4.1. BRICK

The **brick** resource type groups all Gluster bricks in a Red Hat Virtualization environment.

Table 4.1. Gluster brick parameters

Name	Type	Description	Required	User Creatable	User Updatable
--server-id	string	The address of the Gluster server.	Yes	Yes	No
--brick-dir	string	The brick's directory on the Gluster server.	Yes	Yes	No
--replica-count	integer	Defines the file replication count for a replicated volume.	No	Yes	No
--stripe-count	Integer	Defines the stripe count for a striped volume	No	Yes	No

The following table lists additional **glustervolume** options for resource-based commands.

Table 4.2. Additional command options

Option	Description
--cluster-identifier	Reference to the cluster that contains a glustervolume sub-resource.
--glustervolume-identifier	Adds the brick to a glustervolume as a sub-resource.

Example 4.1. Creating a bricks

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

4.2. CDROM

The **cdrom** resource type groups all virtual CD-ROM drive resources in a Red Hat Virtualization environment.

Table 4.3. CD-ROM parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--file-id</code>	string	Defines the file name of the ISO that resides on an ISO storage domain.	Yes	Yes	Yes

Example 4.2. Creating a new CD-ROM

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

Example 4.3. Updating a CD-ROM

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

Example 4.4. Deleting a CD-ROM

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

4.3. CLUSTER

The `cluster` resource type groups all host cluster resources in a Red Hat Virtualization environment.

Table 4.4. Cluster parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--cpu-id</code>	string	A server CPU reference that defines the CPU type all hosts must support in the cluster.	Yes	Yes	Yes
<code>--data-center-id name</code>	string	A reference to the data center for a host cluster.	Yes	Yes	No
<code>--name</code>	string	The name of a host cluster.	Yes	Yes	Yes
<code>--version-major</code>	int	The major version number of the cluster. For example, for Red Hat Virtualization 4.1, the major version is 4.	Yes	Yes	Yes

Name	Type	Description	Required	User Crea table	User Upda table
<code>--version-minor</code>	int	The minor version number of the cluster. For example, for Red Hat Virtualization 4.1, the minor version is 0.	Yes	Yes	Yes
<code>--description</code>	string	A description for the host cluster.	No	Yes	Yes
<code>--error_handling-on_error</code>	string	Defines virtual machine handling when a host within a cluster becomes non-operational, including <code>migrate</code> , <code>do_not_migrate</code> and <code>migrate_highly_available</code> .	No	Yes	Yes
<code>--gluster_service</code>	Boolean	The status is either <code>true</code> or <code>false</code> .	No	Yes	Yes
<code>--memory_policy-overcommit-percent</code>	double	The percentage of host memory allowed in use before a host can no longer run any more virtual machines. Virtual machines can use more than the available host memory due to memory sharing under KSM. Recommended values include 100 (None), 150 (Server Load) and 200 (Desktop Load).	No	Yes	Yes
<code>--memory_policy-transparent_hugepages-enabled</code>	Boolean	Defines the enabled status of Transparent Hugepages. The status is either <code>true</code> or <code>false</code> .	No	Yes	Yes
<code>--scheduling_policy-policy</code>	string	The VM scheduling mode for hosts in the cluster, such as <code>evenly_distributed</code> , <code>power_saving</code> or blank for none.	No	Yes	Yes
<code>--scheduling_policy-thresholds-duration</code>	int	The number of seconds the host can be overloaded before the scheduler starts and moves the load to another host.	No	Yes	Yes
<code>--scheduling_policy-thresholds-high</code>	int	Controls the highest CPU usage percentage the host can have before being considered overloaded.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--scheduling_policy_thresholds_low</code>	int	Controls the lowest CPU usage percentage the host can have before being considered underutilized.	No	Yes	Yes
<code>--threads_as_cores</code>	Boolean	Hosts treat threads as cores, allowing hosts to run virtual machines with a total number of processor cores greater than the number of cores in the host. The status is either true or false .	No	No	No
<code>--trusted_service</code>	Boolean	Defines whether an OpenAttestation server is used to verify hosts.	No	Yes	Yes
<code>--virt_service</code>	Boolean	The status is either true or false .	No	Yes	Yes
<code>--expect</code>	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No
<code>--correlation_id</code>	string	A tagging identifier of an action for cross-system logging. If the client does not define the identifier, one will be generated.	No	Yes	No

Example 4.5. Creating a new cluster

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

Example 4.6. Updating a cluster

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

Example 4.7. Deleting a cluster

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

4.4. DATACENTER

The `datacenter` resource type groups all data center resources in a Red Hat Virtualization environment.

Table 4.5. Data Center Parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the data center.	Yes	Yes	Yes
<code>--storage-type</code>	string	The type of storage for the data center, including <code>iscsi</code> , <code>fcp</code> , <code>nfs</code> , <code>localfs</code> or <code>posixfs</code> .	Yes	Yes	Yes
<code>--version-major</code>	int	The major version number of the data center. For example, for Red Hat Virtualization 4.1, the major version is 4.	Yes	Yes	Yes
<code>--version-minor</code>	int	The minor version number of the data center. For example, for Red Hat Virtualization 4.1, the minor version is 0.	Yes	Yes	Yes
<code>--description</code>	string	A description for the data center.	No	Yes	Yes
<code>--storage-format</code>	string	The metadata format for the data center, including <code>v1</code> , <code>v2</code> or <code>v3</code> .	No	Yes	Yes
<code>--expect</code>	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No
<code>--correlation_id</code>	string	A tagging identifier of an action for cross-system logging. If the client does not define the identifier, one will be generated.	No	Yes	No

Example 4.8. Creating a new data center

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

Example 4.9. Updating a data center

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

Example 4.10. Deleting a data center

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

4.5. DISK

The **disk** resource type groups all virtual hard disk resources in a Red Hat Virtualization environment.

Table 4.6. Disk parameters

Name	Type	Description	Required	User Creatable	User Updatable
--provisioned_size	int	The reserved storage space for the disk. This space is preallocated for the disk to use, even if the disk size is less than the provisioned_size	Yes	Yes	Yes
--interface	string	The interface type of the disk. Either ide or virtio .	Yes	Yes	Yes
--format	string	The underlying storage format. Copy On Write (cow) allows snapshots, with a small performance overhead. Raw (raw) does not allow snapshots, but offers improved performance.	Yes	Yes	Yes
--size	int	The actual size of the disk.	No	Yes	Yes
--sparse	Boolean	true if the physical storage for the disk should not be preallocated.	No	Yes	Yes
--bootable	Boolean	true if this disk is to be marked as bootable.	No	Yes	Yes
--shareable	Boolean	true if this disk is shareable.	No	Yes	Yes
--allow_snapshot	Boolean	true if this disk allows snapshots.	No	Yes	Yes
--propagate_errors	Boolean	true if disk errors should not cause virtual machine to be paused and, instead, disk errors should be propagated to the guest OS.	No	Yes	Yes
--wipe_after_delete	boolean	true if the underlying physical storage for the disk should be zeroed when the disk is deleted. This increases security but is a more intensive operation and may prolong delete times.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--storage_domains-storage_domain</code>	collection	Defines a specific storage domain for the disk.	No	Yes	No

The `--storage_domains-storage_domain` parameter is a collection that uses the sub-parameters in the following table.

Table 4.7. --storage_domains-storage_domain parameters

Name	Type	Description
<code>storage_domain.id name</code>	string	A reference to a storage domain for the disk.

The following table lists additional disk options for resource-based commands.

Table 4.8. Additional command options

Option	Description
<code>--vm-identifier</code>	Adds the disk to a VM as a sub-resource.
<code>--alias</code>	Identifies a disk name when using a <code>show</code> command. Use <code>--alias</code> instead of the <code>--name</code> parameter for disk-specific queries.

Example 4.11. Creating a new disk

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

Example 4.12. Updating a storage domain

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

Example 4.13. Deleting a storage domain

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

The following table lists actions for a virtual disk resource.

Table 4.9. Virtual disk actions

Action	Description
<code>activate</code>	Activate a disk on a virtual machine.
<code>deactivate</code>	Deactivate a disk on a virtual machine.

4.6. GLUSTERVOLUME

The `glustervolume` resource type groups all Gluster storage volume resources in a Red Hat Virtualization environment.

Table 4.10. Gluster volume parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the Gluster volume.	Yes	Yes	No
<code>--volume_type</code>	string	Defines the Gluster volume type. Choose from DISTRIBUTE , REPLICATE , DISTRIBUTED_REPLICATE , STRIPE or DISTRIBUTED_STRIPE .	Yes	Yes	No
<code>--bricks-brick</code>	collection	A new Gluster volume requires a set of Gluster bricks to add and manage. This parameter references a collection of brick details. Specify at least one brick but list multiple bricks-brick parameters for multiple bricks. See below for collection details.	Yes	Yes	No
<code>--transport_types</code>	collection	A reference to available transport methods for the Gluster volume. See below for collection details.	No	Yes	No
<code>--replica_count</code>	integer	Defines the file replication count for a replicated volume.	No	Yes	No
<code>--stripe_count</code>	Integer	Defines the stripe count for a striped volume	No	Yes	No
<code>--options-option</code>	collection	A reference to options for the Gluster volume. See below for collection details.	No	Yes	No

The `--bricks-brick` parameter is a collection that uses the sub-parameters in the following table.

Table 4.11. bricks-brick parameters

Name	Type	Description
<code>brick.server_id</code>	string	The address of the Gluster server.
<code>brick.brick_dir</code>	string	The brick's directory on the Gluster server.

The `--transport_types` parameter is a collection that uses the sub-parameters in the following table.

Table 4.12. transport_types parameters

Name	Type	Description
<code>transport_type</code>	string	Defines a transport type to use. Specify multiple <code>transport_type</code> parameters for more than one type. Choose from TCP and RDMA .

The `--options-option` parameter is a collection that uses the sub-parameters in the following table.

Table 4.13. options-option parameters

Name	Type	Description
<code>option.name</code>	string	The Gluster option name.
<code>option.value</code>	string	The Gluster option value.

The following table lists additional `glustervolume` options for resource-based commands.

Table 4.14. Additional command options

Option	Description
<code>--cluster-identifier</code>	Adds the Gluster volume to a cluster as a sub-resource.

Example 4.14. Creating a Gluster volume with two bricks

```
[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"
```

Example 4.15. Deleting a Gluster volume

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

The following table lists actions for a Gluster volume resource.

Table 4.15. Gluster volume actions

Action	Description
start	Makes a Gluster volume available for use.
stop	Deactivates a Gluster volume.
setOption	Sets a Gluster volume option.
resetOption	Resets a Gluster volume option to the default.
resetAllOptions	Resets all Gluster volume options to defaults.

4.7. GROUP

The group resource type defines all identity service groups for a Red Hat Virtualization environment.

Table 4.16. Group parameters

Name	Type	Description	Required	User Creatable	User Updatable
--name	string	The name of the group, usually the full group path within the identity directory service.	No	No	No

Example 4.16. Creating a group

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

4.8. HOST

The host resource type groups all host resources in a Red Hat Virtualization environment.

Table 4.17. Host parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the host.	Yes	Yes	Yes
<code>--address</code>	string	The IP address or hostname for the host.	Yes	Yes	Yes
<code>--root_password</code>	string	The password for the host's root user.	Yes	Yes	Yes
<code>--cluster-id name</code>	string	Defines the cluster that includes the host.	Yes	Yes	Yes
<code>--port</code>	int	The port for communication with the VDSM daemon running on the host.	No	Yes	Yes
<code>--storage_manager_priority</code>	int	Sets the priority of host order for storage pool manager (SPM).	No	Yes	Yes
<code>--power_management-type</code>	string	The type of power management device in the host.	No	Yes	Yes
<code>--power_management-enabled</code>	boolean	Indicates whether power management configuration is enabled or disabled.	No	Yes	Yes
<code>--power_management-address</code>	string	The host name or IP address of the power management device.	No	Yes	Yes
<code>--power_management-user_name</code>	string	A valid user name for power management.	No	Yes	Yes
<code>--power_management-password</code>	string	A valid, robust password for power management.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--power_management-options-option</code>	collection	Fencing options for the selected <code>power_management - type</code> .	No	Yes	Yes
<code>--reboot_after_installation</code>	boolean	Defines if the host reboots after VDSM installation.	No	Yes	No

The `--power_management-options-option` parameter is a collection that uses the sub-parameters in the following table.

Table 4.18. `--power_management-options-option` parameters

Name	Type	Description
<code>option.name</code>	string	Power management option name.
<code>option.value</code>	string	Power management option value.

Example 4.17. Creating a new host

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

Example 4.18. Updating a host

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

Example 4.19. Deleting a host

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

The following table lists actions for a host resource.

Table 4.19. Host actions

Action	Description
--------	-------------

Action	Description
activate	Activate a host.
approve	Approve a host.
commitnetconfig	Save the network configuration.
deactivate	Deactivate a host.
fence	Fence a host.
forceselectspm	Select the host to be the Storage Pool Manager.
install	Install VDSM on a host.
iscsidiscover	Perform an iSCSI discover command.
iscsilogin	Perform an iSCSI login command.

The following table lists additional options for the **fence** action.

Table 4.20. Fencing options

Option	Description
manual	Manually fence the host. Use this action to confirm to the Manager that the host became non-responsive and was manually rebooted.
restart	Restart the host, implemented as stop, wait, status, start, wait, status.
start	Power on the host.
stop	Power off the host.
status	Check the operational status of the host.

Example 4.20. Confirming a host is rebooted

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

4.9. NETWORK

The **network** resource type groups all logical network resources in a Red Hat Virtualization environment.

Table 4.21. Logical network parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--data-center-id name</code>	string	A reference to the data center for a logical network.	Yes	Yes	No
<code>--name</code>	string	A plain text name for the logical network.	Yes	Yes	No
<code>--description</code>	string	A description for the logical network.	No	Yes	Yes
<code>--vlan-id</code>	string	A VLAN tag.	No	Yes	Yes
<code>--ip-address</code>	string	The IP address for the logical network's bridge.	No	Yes	Yes
<code>--ip-gateway</code>	string	The gateway for the logical network's bridge.	No	Yes	Yes
<code>--ip-netmask</code>	string	The netmask for the logical network's bridge.	No	Yes	Yes
<code>--display</code>	boolean	Signifies if a logical network is used for display communication usage. Set to either true or false .	No	Yes	Yes
<code>--stp</code>	boolean	Set to true if Spanning Tree Protocol is enabled on this network.	No	Yes	Yes
<code>--mtu</code>	int	Sets a user-defined value for the maximum transmission unit of the logical network.	No	Yes	Yes
<code>--usages-usage</code>	collection	Defines usage parameters for the logical network.	No	No	Yes

The `--usages-usage` parameter is a collection that uses the sub-parameters in the following table.

Table 4.22. usages-usage parameters

Name	Type	Description
<code>usage</code>	string	Usage types for the network. Options include VM and DISPLAY .

The following table lists additional `network` options for resource-based commands.

Table 4.23. Additional command options

Option	Description
<code>--cluster-identifier</code>	Adds the network to a cluster as a sub-resource.

Example 4.21. Creating a new network

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

Example 4.22. Attaching an existing network to a cluster

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

Example 4.23. Updating a network

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

Example 4.24. Deleting a network

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

4.10. NIC

The `nic` resource type groups network interface resources in a Red Hat Virtualization environment. These resources act as sub-resources for both `host` and `vm` resources but are defined differently for each. This section contains two tables with parameters for each.

Table 4.24. Host network interface parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--network-id name</code>	string	A reference to the network, if any, that the interface is attached.	Yes	Yes	Yes
<code>--name</code>	string	The name of the host network interface, e.g. <code>eth0</code> .	Yes	Yes	Yes
<code>--bonding-slaves-host_nic</code>	collection	A collection of slave network interfaces that form a bonded interface.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
--bonding-options-option	collection	A list of options for a bonded interface. Each option contains property name and value attributes.	No	Yes	Yes
--ip-gateway	string	The IP address for the network's gateway.	No	Yes	Yes
--boot-protocol	string	The protocol for IP address assignment when the host is booting, such as dhcp or static .	No	Yes	Yes
--mac	string	The MAC address of the interface.	No	Yes	Yes
--ip-address	string	The IP address of the interface.	No	Yes	Yes
--ip-netmask	string	The netmask for the interface's IP address.	No	Yes	Yes
--ip-mtu	int	The maximum transmission unit for the interface.	No	No	Yes

Table 4.25. Virtual Machine network interface parameters

Name	Type	Description	Required	User Creatable	User Updatable
--network-id name	string	A reference to the network, if any, that the interface is attached.	Yes	Yes	Yes
--name	string	The name of the interface, e.g. eth0 .	Yes	Yes	Yes
--mac-address	string	The MAC address of the interface.	No	Yes	Yes
--interface	string	Defines the interface type, such as e1000 , virtio , rtl8139 and rtl8139_virtio .	No	Yes	Yes
--port-mirroring-networks-network	collection	Defines a set of networks to copy (mirror) network data from the network interface.	No	Yes	Yes

The `--bonding-slaves-host_nic` parameter is a collection that uses the sub-parameters in the following table.

Table 4.26. --bonding-slaves-host_nic

Name	Type	Description
<code>host_nic.id name</code>	string	A reference to another host NIC to bond.

The `--bonding-options-option` parameter is a collection that uses the sub-parameters in the following table.

Table 4.27. --bonding-options-option

Name	Type	Description
<code>option.name</code>	string	The bonding option name.
<code>option.value</code>	string	The bonding option value.
<code>type</code>	string	The bonding option type.

The `--port_mirroring-networks-network` parameter is a collection that uses the sub-parameters in the following table.

Table 4.28. --port_mirroring-networks-network

Name	Type	Description
<code>network.id</code>	string	A reference to the network to mirror.

The following table lists additional NIC options for resource-based commands.

Table 4.29. Additional command options

Option	Description
<code>--host-identifier</code>	Adds the NIC to a host as a sub-resource.
<code>--vm-identifier</code>	Adds the NIC to a vm as a sub-resource.

Example 4.25. Creating a new network interface on a virtual machine

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

Example 4.26. Updating a network interface on a virtual machine

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

Example 4.27. Deleting a network interface on a virtual machine

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

Example 4.28. Configuring network bonding on a host

```
[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
```

Example 4.29. Assigning a logical network to a host network interface

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

The following table lists actions for a host NIC resource.

Table 4.30. Host NIC actions

Action	Description
attach	Attach a NIC to a host.
detach	Detach a NIC from a host.

The following table lists actions for a virtual machine NIC resource.

Table 4.31. Virtual machine NIC actions

Action	Description
activate	Activate a NIC on a virtual machine.
deactivate	Deactivate a NIC on a virtual machine.

4.11. PERMISSION

The `permission` resource type groups all permission resources in a Red Hat Virtualization environment.

Table 4.32. Permission parameters

Name	Type	Description	Required	User Creatable	User Updatable
--user-id, --group-id	string	A reference to the user or group using the permission.	Yes	Yes	No
--role-id	string	A reference to a role to assign for the permission.	Yes	Yes	No
--expect	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No

The following table lists additional **permission** options for resource-based commands.

Table 4.33. Additional command options

Option	Description
--cluster-identifier	Adds the permission to a cluster.
--correlation-id	A tagging identifier for cross-system logging.
--cpuprofile-identifier	Adds the permission to a CPU profile.
--datacenter-identifier	Adds the permission to a data center.
--disk-identifier	Adds the permission to a disk.
--diskprofile-identifier	Adds the permission to a disk profile.
--host-identifier	Adds the permission to a host.
--iscsibond-identifier	Adds the permission to an iSCSI bond.
--network-identifier	Adds the permission to a network.
--storagedomain-identifier	Adds the permission to a storage domain.
--template-identifier	Adds the permission to a template.
--vm-identifier	Adds the permission to a virtual machine.

Option	Description
<code>--vmpool-identifier</code>	Adds the permission to a virtual machine pool.
<code>--vnicprofile-identifier</code>	Adds the permission to a VNIC profile.

Example 4.30. Creating a new permission

```
[RHEVM shell (connected)]# add permission --role-id 00000000-0000-0000-0000-000000000001 --user-id 8b9456ae-e2c8-426e-922d-b01bb8a805fb
```

4.12. PERMIT

The `permit` resource type groups all individual permits for roles in a Red Hat Virtualization environment.

Table 4.34. Permission parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--id</code>	string	A reference to the permit to add.	Yes	Yes	No

The following table lists additional `permit` options for resource-based commands.

Table 4.35. Additional command options

Option	Description
<code>--role-identifier</code>	Adds the permit to a role.

Example 4.31. Creating a new permission

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

4.13. QUOTAS

The `quota` resource type groups all datacenter quotas in a Red Hat Virtualization environment.

Table 4.36. Quota parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the quota.	Yes	Yes	Yes
<code>--description</code>	string	A description for the quota.	Yes	Yes	Yes

4.14. ROLE

The `role` resource type groups all individual roles in a Red Hat Virtualization environment.

Table 4.37. Role parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the role.	Yes	Yes	Yes
<code>--permits-permit</code>	collection	A list of permits for initial inclusion with the role. Additional permits included with the permit resource type.	Yes	Yes	No
<code>--description</code>	string	A description for the role.	No	Yes	Yes
<code>--administrative</code>	Boolean	Set to <code>true</code> if this is an administrative role.	No	Yes	Yes

The `--permits-permit` parameter is a collection that uses the sub-parameters in the following table.

Table 4.38. --permits-permit parameters

Name	Type	Description
<code>permit.id</code>	string	A reference to a permit to add to the role's permits.

Example 4.32. Creating a new role

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

4.15. SNAPSHOT

The `snapshot` resource type groups all virtual machine snapshot resources in a Red Hat Virtualization environment.

Table 4.39. Snapshot parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--description</code>	string	A description for the snapshot.	Yes	Yes	No

The following table lists additional snapshot options for resource-based commands.

Table 4.40. Additional command options

Option	Description
<code>--vm-identifier</code>	Adds the disk to a VM as a sub-resource.

Example 4.33. Creating a new snapshot

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

Example 4.34. Deleting a storage domain

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

The following table lists actions for a virtual machine snapshot resource.

Table 4.41. Virtual machine snapshot actions

Action	Description
<code>restore</code>	Restore a snapshot.

4.16. STATISTIC

The `statistic` resource type groups statistics for resources in a Red Hat Virtualization environment. Resource statistics are listed based on their resource identifier.

Table 4.42. statistic resource identifiers

Option	Description
<code>--brick-identifier</code>	The resource identifier to view statistics for the specified brick.
<code>--cluster-identifier</code>	The resource identifier to view statistics for the specified cluster.
<code>--datacenter-identifier</code>	The resource identifier to view statistics for the specified data center.
<code>--disk-identifier</code>	The resource identifier to view statistics for the specified virtual disk.
<code>--glustervolume-identifier</code>	The resource identifier to view statistics for the specified gluster volume.
<code>--host-identifier</code>	The resource identifier to view statistics for the specified host.
<code>--job-identifier</code>	The resource identifier to view statistics for the specified job.
<code>--nic-identifier</code>	The resource identifier to view statistics for the specified NIC.
<code>--numanode-identifier</code>	The resource identifier to view statistics for the specified NUMA node.
<code>--step-identifier</code>	The resource identifier to view statistics for the specified step.
<code>--storagedomain-identifier</code>	The resource identifier to view statistics for the specified storage domain.
<code>--vm-identifier</code>	The resource identifier to view statistics for the specified virtual machine.

View the collection of statistics for each resource by using the `list` command and the relevant resource identifier. The following example provides a list of the available statistics for the specified host:

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

The `name` or `id` of the provided statistics can be used with the `show` command and the resource identifier to view further information on the specified statistic. The following example shows the details of the specified statistic for the host:

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

4.17. STORAGECONNECTION

The `storageconnection` resource type allows you to add, edit, and delete storage connections.

Table 4.43. Storage connection parameters

Name	Type	Description	Required	User Creatable	User Updatable
--address	string	The hostname or IP address of the storage domain.	Yes (NFS and iSCSI only)	Yes	Yes
--correlation_id	string	A tagging identifier for the storage connection.	No	No	Yes
--expect	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No
--iqn	string	The target IQN for the storage device.	Yes (iSCSI only)	Yes	Yes
--mount_options	string	The options for mounting the PosixFS share.	No	Yes	Yes
--nfs_retrans	integer	The number of retransmissions the NFS client will attempt to complete a request.	No	Yes	Yes
--nfs_timeo	integer	The amount of time, in deciseconds, the NFS client will wait for a request to complete.	No	Yes	Yes
--nfs_version	string	The version of NFS used.	No	Yes	Yes
--password	string	A CHAP password for logging into a target of an iSCSI storage domain.	No	Yes	Yes
--path	string	The mounted file path of the storage domain. The path cannot be updated to one already used by a storage connection.	Yes (NFS, local, and PosixFS only)	Yes	Yes
--port	integer	The TCP port used for the iSCSI storage domain.	Yes (iSCSI only)	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--storagedomain-identifier</code>	string	A reference to a storage domain for the disk.	No	No	No
<code>--type</code>	string	The type of storage domain.	Yes	Yes	No
<code>--username</code>	string	A CHAP user name for logging into a target of an iSCSI storage domain.	No	Yes	Yes
<code>--vfs_type</code>	string	The Linux-supported file system type of the PosixFS share.	Yes (PosixFS only)	Yes	Yes

Example 4.35. Creating a new storage connection

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

4.18. STORAGE DOMAIN

The `storagedomain` resource type groups all storage domain resources in a Red Hat Virtualization environment.

Table 4.44. Storage domain parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the storage domain.	No	Yes	Yes
<code>--format</code>	Boolean	The metadata format for the data center, including v1 , v2 or v3 .	Yes	Yes	No
<code>--host-id name</code>	string	A reference to the host from which this storage domain should be initialized. The only restriction on this host is that it should have access to the physical storage specified.	Yes	Yes	No

Name	Type	Description	Required	User Creatable	User Updatable
<code>--storage-address</code>	string	The IP address or hostname of the storage device.	Yes	Yes	No
<code>--storage-logical_unit</code>	collection	The logical unit information of the storage device.	See below	Yes	No
<code>--storage-mount_options</code>	string	The options for mounting the storage domain.	See below	Yes	No
<code>--storage-override_luns</code>	Boolean	Defines whether to override the logical unit number. The status is either true or false .	See below	Yes	No
<code>--storage-path</code>	string	The path on the storage device to use for the storage domain.	See below	Yes	No
<code>--storage-type</code>	string	The type of storage for the data center, including iscsi , fc , nfs , glusterfs , localfs or posixfs .	Yes	Yes	No
<code>--storage-vfs_type</code>	string	Defines the file system type of the storage domain.	See below	Yes	No
<code>--type</code>	string	The type of storage domain, including data , iso and export .	Yes	Yes	No

The `--storage-logical_unit` parameter is a collection that requires all sub-parameters in the following table.

Table 4.45. storage-logical_unit parameters

Name	Type	Description
<code>logical_unit.address</code>	string	The address of the server containing the storage device.
<code>logical_unit.port</code>	integer	The port number of the server.
<code>logical_unit.target</code>	string	The target IQN for the storage device.
<code>logical_unit.username</code>	string	A CHAP user name for logging into a target.

Name	Type	Description
<code>logical_unit.password</code>	string	A CHAP password for logging into a target.
<code>logical_unit.serial</code>	string	The serial ID for the target.
<code>logical_unit.vendor_id</code>	string	The vendor name for the target.
<code>logical_unit.product_id</code>	string	The product code for the target.
<code>logical_unit.lun_mapping</code>	integer	The Logical Unit Number device mapping for the target.
<code>logical_unit.portal</code>	string	The logical unit portal.
<code>logical_unit.paths</code>	integer	The logical unit paths.
<code>logical_unit.id</code>	string	A reference to the logical unit ID.

Use the following parameters depending on `storage-type`.

Table 4.46. Storage type parameters

Type	Parameters
<code>nfs</code>	<code>--storage-address, --storage-path</code>
<code>iscsi</code> or <code>fc</code>	<code>--storage-address, --storage-logical_unit, --storage-override_luns</code>
<code>glusterfs</code>	<code>--storage-address, --storage-path, --storage-vfs_type</code>
<code>local</code>	<code>--storage-path</code>
<code>posixfs</code>	<code>--storage-path, --storage-vfs_type, --storage-address, --storage-mount_options</code>

The following table lists additional `storagedomain` options for resource-based commands.

Table 4.47. Additional command options

Option	Description
--datacenter-identifier	Adds the storage domain to a datacenter as a sub-resource.

Example 4.36. Creating a new storage domain

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

Example 4.37. Adding a gluster storage domain

```
[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
```

Example 4.38. Updating a storage domain

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

Example 4.39. Deleting a storage domain

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

The following table lists actions for a storage domain resource.

Table 4.48. Storage domain actions

Action	Description
activate	Activate a storage domain on a data center.
deactivate	Deactivate a storage domain on a data center.

4.19. TAG

The **tag** resource type groups all tags in a Red Hat Virtualization environment.

Table 4.49. Tag parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the tag.	Yes	Yes	Yes
<code>--description</code>	string	A description for the string.	Yes	Yes	Yes
<code>--parent-name</code>	string	A reference to the parent tag that the tag is attached.	Yes	Yes	Yes

Example 4.40. Creating a new tag

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

4.20. TEMPLATE


The `template` resource type groups all virtual machine templates in a Red Hat Virtualization environment. Only `--vm-id|name` and `--name` are required parameters. If the optional parameters are not specified, the template will inherit the settings from the virtual machine used to make the template.

Table 4.50. Template parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--vm-id name</code>	string	A reference to the virtual machine used as the basis for the template.	Yes	Yes	No
<code>--name</code>	string	The name of the virtual machine template.	Yes	Yes	Yes
<code>--memory</code>	long	The amount of memory for the virtual machine template in bytes.	No	Yes	Yes
<code>--cpu-topology-cores</code>	int	The number of CPU cores available to the virtual machine template.	No	Yes	Yes
<code>--high-availability-enabled</code>	Boolean	Set to <code>true</code> to enable high availability for the virtual machine template.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--os-cmdline</code>	string	A kernel command line parameter string to be used with the defined kernel. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes
<code>--origin</code>	string	The virtual machine template's origin. Specify rhev , vmware , or xen .	No	Yes	Yes
<code>--high-availability-priority</code>	int	Sets the priority value (i.e. boot order) of each virtual machine template's high availability.	No	Yes	Yes
<code>--timezone</code>	string	The Sysprep timezone setting for a Windows virtual machine template. Specify formats such as GMT+00:00 .	No	Yes	Yes
<code>--domain-name</code>	string	The domain name of the virtual machine template.	No	Yes	Yes
<code>--type</code>	string	Defines the virtual machine type. Specify either desktop or server .	No	Yes	Yes
<code>--stateless</code>	boolean	Set to true if the resulting virtual machines are stateless. A stateless virtual machine contains a snapshot of its disk image taken at boot and removed at shutdown. This means state changes do not persist after a reboot.	No	Yes	Yes
<code>--delete-protected</code>	boolean	Set to true to make it impossible to delete a virtual machine created from this template.	No	Yes	Yes
<code>--sso-methods-method</code>	collection	Defines the single sign-on method used. For example, <code>--sso-methods-method method.id=GUEST_AGENT</code> .	No	Yes	Yes
<code>--rng-device-rate-bytes</code>	int	Specifies how many bytes are permitted to be consumed per period.	No	Yes	Yes
<code>--rng-device-rate-period</code>	int	Specifies the duration of a period in milliseconds. If specified, <code>--rng-device-rate-bytes</code> must be specified as well.	No	Yes	Yes

Name	Type	Description	Required	User Crea table	User Upda table
<code>--rng_device-source</code>	string	The source of the random number generator. Specify either random or hw rng .	No	Yes	Yes
<code>--console-enabled</code>	boolean	Set to true to enable the VirtIO console device feature.	No	Yes	Yes
<code>--placement_policy-affinity</code>	string	The migration affinity for each virtual machine created from the template. Specify migratable , user_migratable , or pinned .	No	Yes	Yes
<code>--description</code>	string	A description for the virtual machine template.	No	Yes	Yes
<code>--comment</code>	string	A comment for the virtual machine template.	No	Yes	Yes
<code>--custom_properties-custom_property</code>	collection	A set of user-defined environment variables passed as parameters to custom scripts.	No	Yes	Yes
<code>--os-type</code>	string	The operating system type for the virtual machine template.	No	Yes	Yes
<code>--os-boot</code>	collection	The boot device for the virtual machine template. Specify cdrom , hd , or network . For example, <code>--os-boot boot.dev=hd</code> .	No	Yes	Yes
<code>--cpu-topology-sockets</code>	int	The number of CPU sockets available to the virtual machine template.	No	Yes	Yes
<code>--cpu_shares</code>	int	The level of CPU resources a virtual machine can demand relative to other virtual machines. For example, 512 for low priority virtual machines, 1024 for medium priority virtual machines, and 2048 for high priority virtual machines.	No	Yes	Yes
<code>--cpu-architecture</code>	string	Defines the CPU architecture. Specify x86_64 , ppc64 , or undefined .	No	Yes	Yes
<code>--os-kernel</code>	string	A path to a kernel image the resulting virtual machines are configured to boot. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--display-type</code>	string	Defines the display type. Specify either spice or vnc .	No	Yes	Yes
<code>--display-monitors</code>	int	Defines the number of displays available.	No	Yes	Yes
<code>--display-single_qxl_pci</code>	boolean	Set to true to drive multiple monitors using a single virtual PCI device.	No	Yes	Yes
<code>--display-allow_override</code>	boolean	Set to true to allow override of the template console settings.	No	Yes	Yes
<code>--display-smartcard_enabled</code>	boolean	Set to true to enable the Smart card feature for virtual machines.	No	Yes	Yes
<code>--display-file_transfer_enabled</code>	boolean	Set to true to enable SPICE file transfer.	No	Yes	Yes
<code>--display-copy_paste_enabled</code>	boolean	Set to true to enable SPICE clipboard copy and paste.	No	Yes	Yes
<code>--display-keyboard_layout</code>	string	Defines the keyboard layout for the virtual machine. This option is only available when using the VNC protocol. Specify formats such as en-US .	No	Yes	Yes
<code>--os-initRd</code>	string	A path to an initrd image to be used with a specified kernel. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes
<code>--usb-enabled</code>	Boolean	Set to true to enable USB support on the virtual machine. This option is only available for virtual machines using the SPICE protocol.	No	Yes	Yes
<code>--usb-type</code>	string	Defines the USB type if USB support is enabled. Specify either Legacy or Native .  IMPORTANT The Legacy USB option has been deprecated and will be removed in Red Hat Virtualization 4.1.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--tunnel-migration</code>	boolean	Set to true to enable data transport over a libvirt daemon. A tunneled transport uses a stronger encryption algorithm but increases the data load during transport.	No	Yes	Yes
<code>--migration-downtime</code>	int	Defines the maximum number of milliseconds that the virtual machine can be down during live migration.	No	Yes	Yes
<code>--virtio-scsi-enabled</code>	boolean	Set to true to allow attaching a VirtIO console device to the virtual machine.	No	Yes	Yes
<code>--soundcard-enabled:</code>	boolean	Set to true to enable sound cards.	No	Yes	Yes
<code>--vm-disks-disk</code>	collection	References to disks attached to the template.	No	Yes	No
<code>--id</code>	string	The ID of the virtual machine template.	No	Yes	Yes
<code>--permissions-clone</code>	boolean	Set to true to copy the permissions of the source virtual machine to the template.	No	Yes	Yes
<code>--version-version-name</code>	string	Used with the <code>--version-base-template-id</code> parameter. Defines the name for the sub template.	No	Yes	Yes
<code>--version-base-template-id</code>	string	Defines the template ID to be used as the root template. Used if you want to create this template as a sub template of a root template.	No	Yes	Yes
<code>--cpu-cpu-tune-vcpu-pin</code>	collection	Defines which virtual CPUs of a virtual machine to pin to the physical CPUs of a host.	No	Yes	Yes
<code>--serial-number-policy</code>	string	Defines the serial number policy for the virtual machine template. Specify host , vm , or custom . If custom is used, also define the serial number value using <code>--serial-number-value</code> .	No	Yes	Yes
<code>--serial-number-value</code>	string	Defines the serial number for the virtual machine template.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--bios-boot-menu-enabled</code>	boolean	Set to true to enable boot menu.	No	Yes	Yes
<code>--cluster-id</code>	string	Defines the cluster to use by specifying the cluster ID.	No	Yes	Yes
<code>--cluster-name</code>	string	Defines the cluster to use by specifying the cluster name.	No	Yes	Yes
<code>--cpu-profile-id</code>	string	Defines the CPU profile to use. Use the list cpuprofiles command to retrieve a full list of CPU profile IDs.	No	Yes	Yes
<code>--expect</code>	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	Yes	Yes
<code>--correlation-id</code>	string	A tagging identifier of an action for cross-system logging. If the client does not define the identifier, one will be generated.	No	Yes	Yes

The `--sso-methods-method` parameter is a collection that uses the sub-parameters in the following table.

Table 4.51. --sso-methods-method parameters

Name	Type	Description
<code>method.id</code>	string	The single sign-on method used: GUEST_AGENT .

The `--custom-properties-custom-property` parameter is a collection that uses the sub-parameters in the following table.

Table 4.52. --custom-properties-custom-property parameters

Name	Type	Description
<code>custom-property.name</code>	string	The custom property name.
<code>custom-property.value</code>	string	The custom property value.

The `--os-boot` parameter is a collection that uses the sub-parameters in the following table.

Table 4.53. --os-boot parameters

Name	Type	Description
<code>boot.dev</code>	string	The boot device for the virtual machine template. Specify <code>cdrom</code> , <code>hd</code> , or <code>network</code> .

The `--vm-disks-disk` parameter is a collection that uses the sub-parameters in the following table.

Table 4.54. --vm-disks-disk parameters

Name	Type	Description
<code>disk.id</code>	string	A reference to a virtual disk.
<code>storage_domain.s.storage_domain</code>	collection	Defines a set of sub-parameters for the disk's storage domain.

The `--cpu-cpu_tune-vcpu_pin` parameter is a collection that uses the sub-parameters in the following table.

Table 4.55. --cpu-cpu_tune-vcpu_pin parameters

Name	Type	Description
<code>vcpu_pin.vcpu</code>	int	The virtual CPU to assign.
<code>vcpu_pin.cpu_set</code>	string	The physical CPUs on the host.

Example 4.41. Creating a new template

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

Example 4.42. Updating a template

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

Example 4.43. Deleting a template

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

The following table lists actions for a virtual machine template resource.

Table 4.56. Virtual machine template actions

Action	Description
<code>export</code>	Export a template to an export storage domain.

4.21. USER

The `user` resource type groups all users in a Red Hat Virtualization environment.

Table 4.57. User parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--user_name</code>	string	The user name from the directory service.	Yes	Yes	No
<code>--domain-id name</code>	string	A reference to the directory service domain.	Yes	Yes	No

Example 4.44. Creating a new user

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

4.22. VM

The `vm` resource type groups all virtual machine resources in a Red Hat Virtualization environment.

Table 4.58. Virtual machine parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the virtual machine	Yes	Yes	Yes
<code>--template-id name</code>	string	A reference to the template used as the basis for the virtual machine.	Yes	Yes	No
<code>--cluster-id name</code>	string	A reference to the cluster that includes this VM.	Yes	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--instance-type-id name</code>	string	Defines the instance type. Specify custom , large , medium , small , tiny , or xlarge .	No	Yes	Yes
<code>--quota-id</code>	string	A reference to the quota usage for the virtual machine.	No	Yes	No
<code>--timezone</code>	string	The Sysprep time zone setting for a Windows virtual machine.	No	Yes	Yes
<code>--os-boot</code>	collection	The boot device for the virtual machine. Specify cdrom , hd , or network .	No	Yes	Yes
<code>--custom-properties-custom-property</code>	collection	A set of user-defined environment variables passed as parameters to custom scripts.	No	Yes	Yes
<code>--os-type</code>	string	The operating system type for this virtual machine.	No	Yes	Yes
<code>--usb-enabled</code>	boolean	Defines the USB policy for a virtual machine. Set to true to enable USB on the virtual machine.	No	Yes	Yes
<code>--usb-type</code>	string	Defines the USB type if enabled.	No	Yes	Yes
<code>--type</code>	string	Defines the virtual machine type. Specify either desktop or server .	No	Yes	Yes
<code>--os-initRd</code>	string	A path to an initrd image to be used with a specified kernel. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes
<code>--display-monitors</code>	int	Defines the number of displays available.	No	Yes	Yes
<code>--display-single_qxl_pci</code>	boolean	Set to true to drive multiple monitors using a single virtual PCI device.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--display-type</code>	string	Defines the display type. Specify either spice or vnc .	No	Yes	Yes
<code>--display-allow-override</code>	boolean	Set to true to allow override of the virtual machine console settings.	No	Yes	Yes
<code>--display-smartcard-enabled</code>	boolean	Set to true to enable the Smart card feature.	No	Yes	Yes
<code>--display-file-transfer-enabled</code>	boolean	Set to true to enable SPICE file transfer.	No	Yes	Yes
<code>--display-copy-paste-enabled</code>	boolean	Set to true to enable SPICE clipboard copy and paste.	No	Yes	Yes
<code>--display-keyboard-layout</code>	string	Defines the keyboard layout for the virtual machine. This option is only available when using the VNC protocol. Specify formats such as en-US .	No	Yes	Yes
<code>--os-cmdline</code>	string	A kernel command line parameter string to be used with the defined kernel. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes
<code>--cpu-topology-cores</code>	int	The number of CPU cores available to the virtual machine.	No	Yes	Yes
<code>--cpu-architecture</code>	string	Defines the CPU architecture. Specify x86_64 , ppc64 , or undefined .	No	Yes	Yes
<code>--memory</code>	long	The amount of memory for the virtual machine in bytes.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
--memory-policy-guaranteed	long	The minimum amount of memory, in bytes, guaranteed on a host in order for the virtual machine to run.	No	Yes	Yes
--memory-policy-balloonin	boolean	Set to true to enable memory balloon device.	No	Yes	Yes
--high-availability-priority	int	Sets the priority value (migration and restart order) of each virtual machine using high availability.	No	Yes	Yes
--high-availability-enabled	boolean	Defines whether high availability is enabled for the virtual machine.	No	Yes	Yes
--domain-name	string	The domain name of the virtual machine.	No	Yes	Yes
--description	string	A description of the virtual machine.	No	Yes	Yes
--comment	string	A comment for the virtual machine.	No	Yes	Yes
--stateless	boolean	Set to true if the virtual machine is stateless. A stateless virtual machine contains a snapshot of its disk image taken at boot and removed at shutdown. This means state changes do not persist after a reboot.	No	Yes	Yes
--permissions-clone	boolean	Set to true to copy the permissions of the source virtual machine to the template.	No	Yes	Yes
--delete-protected	boolean	Set to true to make it impossible to delete a virtual machine created from this template.	No	Yes	Yes
--sso-methods-method	collection	Defines the single sign-on method used. For example, --sso-methods-method method.id=GUEST_AGENT .	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--rng_device-rate-bytes</code>	int	Specifies how many bytes are permitted to be consumed per period.	No	Yes	Yes
<code>--rng_device-rate-period</code>	int	Specifies the duration of a period in milliseconds. If specified, <code>--rng_device-rate-bytes</code> must be specified as well.	No	Yes	Yes
<code>--rng_device-source</code>	string	The source of the random number generator. Specify either <code>random</code> or <code>hwrng</code> .	No	Yes	Yes
<code>--console-enabled</code>	boolean	Set to <code>true</code> to enable the VirtIO console device feature.	No	Yes	Yes
<code>--cpu-mode</code>	string	Defines the CPU mode. Specify <code>custom</code> , <code>host_model</code> , or <code>host_passthrough</code> .	No	Yes	Yes
<code>--cpu-topology-sockets</code>	int	The number of CPU sockets available to the virtual machine.	No	Yes	Yes
<code>--cpu_shares</code>	int	The level of CPU resources a virtual machine can demand relative to other virtual machines. For example, <code>512</code> for low priority virtual machines, <code>1024</code> for medium priority virtual machines, and <code>2048</code> for high priority virtual machines.	No	Yes	Yes
<code>--placement_policy-affinity</code>	string	The migration affinity for each virtual machine. Specify <code>migratable</code> , <code>user_migratable</code> , or <code>pinned</code> .	No	Yes	Yes
<code>--placement_policy-host-id name</code>	string	A reference to the preferred host for migration affinity.	No	Yes	Yes
<code>--origin</code>	string	The virtual machine's origin. Specify <code>rhev</code> , <code>vmware</code> , or <code>xen</code> .	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--os-kernel</code>	string	A path to a kernel image the virtual machine is configured to boot. This option supports booting a Linux kernel directly rather than through the BIOS bootloader.	No	Yes	Yes
<code>--disks-clone</code>	boolean	Defines whether to clone the disk from the defined <code>template</code> .	No	Yes	No
<code>--disks-disk</code>	collection	References to disks attached to the virtual machine.	No	Yes	Yes
<code>--tunnel-migration</code>	boolean	Set to <code>true</code> to enable data transport over a <code>libvirt</code> daemon. A tunneled transport uses a stronger encryption algorithm but increases the data load during transport.	No	Yes	Yes
<code>--migration-downtime</code>	int	Defines the maximum number of milliseconds that the virtual machine can be down during live migration.	No	Yes	Yes
<code>--virtio-serial-enabled</code>	boolean	Set to <code>true</code> to allow attaching a VirtIO console device to the virtual machine.	No	Yes	Yes
<code>--soundcard-enabled:</code>	boolean	Set to <code>true</code> to enable sound cards.	No	Yes	Yes
<code>--payloads-payload</code>	collection	Defines content to send to the virtual machine upon booting.	No	Yes	Yes
<code>--initialization-configuration-type</code>	string	Defines the virtual machine format. Accepts only <code>ovf</code> .	No	Yes	Yes
<code>--initialization-configuration-data</code>	string	This parameter must match the <code>--initialization-configuration-type</code> parameter. Accepts only <code>ovf</code> .	No	Yes	Yes
<code>--cpu-cpu-tune-vcpu-pin</code>	collection	Defines which virtual CPUs of a virtual machine to pin to the physical CPUs of a host.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
<code>--serial-number-policy</code>	string	Defines the serial number policy for the virtual machine template. Specify host , vm , or custom . If custom is used, also define the serial number value using <code>--serial-number-value</code> .	No	Yes	Yes
<code>--serial-number-value</code>	string	Defines the serial number for the virtual machine template.	No	Yes	Yes
<code>--bios-boot-menu-enabled</code>	boolean	Set to true to enable boot menu.	No	Yes	Yes
<code>--numa-tune-mode</code>	string	Defines how to allocate memory for the domain process on a NUMA host. Specify interleave , strict , or preferred . If no value is given, the parameter defaults to strict .	No	Yes	Yes
<code>--cpu-profile-id</code>	string	Defines the CPU profile to use. Use the list cpuprofiles command to retrieve a full list of CPU profile IDs.	No	Yes	Yes
<code>--expect</code>	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No
<code>--correlation-id</code>	string	A tagging identifier of an action for cross-system logging. If the client does not define the identifier, one will be generated.	No	Yes	No

The `--os-boot` parameter is a collection that uses the sub-parameters in the following table.

Table 4.59. --os-boot parameters

Name	Type	Description
<code>boot.dev</code>	string	The boot device for the virtual machine template. Specify cdrom , hd , or network .

The `--custom-properties-custom-property` parameter is a collection that uses the sub-parameters in the following table.

Table 4.60. --custom-properties-custom-property parameters

Name	Type	Description
<code>custom_property.name</code>	string	The custom property name.
<code>custom_property.value</code>	string	The custom property value.

The `--sso-methods-method` parameter is a collection that uses the sub-parameters in the following table.

Table 4.61. --sso-methods-method parameters

Name	Type	Description
<code>method.id</code>	string	The single sign-on method used: GUEST_AGENT .

The `--disks-disk` parameter is a collection that uses the sub-parameters in the following table.

Table 4.62. --disks-disk parameters

Name	Type	Description
<code>disk.id</code>	string	A reference to a virtual disk.
<code>storage_domains.storage_domain</code>	collection	Defines a set of sub-parameters for the disk's storage domain.

The `--payloads-payload` parameter is a collection that uses the sub-parameters in the following table.

Table 4.63. --payloads-payload parameters

Name	Type	Description
<code>payload.type</code>	string	Payload delivery type. Specify either cdrom or floppy .
<code>payload.file.name</code>	string	The payload file name and location on the root file system of the virtual machine.
<code>payload.file.content</code>	string	The content to deliver to the file.

The `--cpu-cpu_tune-vcpu_pin` parameter is a collection that uses the sub-parameters in the following table.

Table 4.64. --cpu-cpu_tune-vcpu_pin

Name	Type	Description
<code>vcpu_pin.vcpu</code>	int	The virtual CPU to assign.
<code>vcpu_pin.cpu_set</code>	string	The physical CPUs on the host.

Example 4.45. Creating a new virtual machine

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

Example 4.46. Updating a virtual machine

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

Example 4.47. Deleting a virtual machine

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

The following table lists actions for a virtual machine resource.

Table 4.65. Virtual machine actions

Action	Description
<code>start</code>	Launch a virtual machine.
<code>stop</code>	Stop a virtual machine.
<code>shutdown</code>	Shut down a virtual machine.
<code>suspend</code>	Suspend a virtual machine.
<code>detach</code>	Detach a virtual machine from a pool.
<code>migrate</code>	Migrate a virtual machine to another host.
<code>cancelmigration</code>	Stop migration in progress.
<code>export</code>	Export a virtual machine to an export storage domain.
<code>move</code>	Move virtual disks to another storage domain.

Action	Description
<code>ticket</code>	Create a ticket for console access.
<code>logon</code>	Enable user logon for console access using third-party applications.

4.23. VMPOOL

The `vmpool` resource type groups all virtual machine pool resources in a Red Hat Virtualization environment.

Table 4.66. Virtual machine pool parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--cluster-id name</code>	string	A reference to the cluster for the virtual machine pool.	Yes	Yes	Yes
<code>--template-id name</code>	string	A reference to the template for the virtual machine pool.	Yes	Yes	Yes
<code>--name</code>	string	The name of the virtual machine pool.	Yes	Yes	Yes
<code>--size</code>	integer	The number of the virtual machines in the pool.	Yes	Yes	Yes

Example 4.48. Creating a new virtual machine pool

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

Example 4.49. Updating a virtual machine pool

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

Example 4.50. Deleting a virtual machine pool

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

4.24. VNICPROFILE

The `vniprofile` resource type groups all VNIC (virtual network interface controller) profiles, also referred to as VM (virtual machine) interface profiles, in a Red Hat Virtualization environment.

Table 4.67. Virtual Network Interface Controller Profile Parameters

Name	Type	Description	Required	User Creatable	User Updatable
<code>--name</code>	string	The name of the VNIC profile.	Yes	Yes	Yes
<code>--network-id</code>	string	A reference to the logical network to which the profile will be applied.	Yes	No	No
<code>--correlation_id</code>	string	A tagging identifier of an action for cross-system logging. If the client does not define the identifier, one will be generated.	No	Yes	No
<code>--description</code>	string	A description for the profile.	No	Yes	Yes
<code>--expect</code>	'201-created'	Request becomes asynchronous until the expected HTTP header is returned. Useful for long-running tasks that would otherwise return as successful before the task is completed.	No	No	No
<code>--custom_properties-custom_property</code>	collection	A set of user-defined environment variables passed as parameters to custom scripts.	No	Yes	Yes
<code>--port_mirroring</code>	Boolean	Toggles whether port mirroring is used for the profile. The status is either True or False . Default is False .	No	No	No

The `--custom_properties-custom_property` parameter is a collection that uses the sub-parameters in the following table.

Table 4.68. --custom_properties-custom_property parameters

Name	Type	Description
<code>custom_property.name</code>	string	The custom property name.
<code>custom_property.value</code>	string	The custom property value.

Example 4.51. Creating a new vnic profile

■

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

Example 4.52. Updating a vnic profile

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

Example 4.53. Deleting a vnic profile

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

CHAPTER 5. CLI QUERIES

5.1. QUERY SYNTAX

The CLI `list` command uses the `--query` attribute to perform server-side queries, which uses the same format as Red Hat Virtualization Manager search query language:

Table 5.1. Example search queries

Collection	Criteria	Result
<code>hosts</code>	<code>vms.status=up</code>	Displays a list of all hosts running virtual machines that are up .
<code>vms</code>	<code>domain=qa.company.com</code>	Displays a list of all virtual machines running on the specified domain.
<code>vms</code>	<code>users.name=mary</code>	Displays a list of all virtual machines belonging to users with the user name mary .
<code>events</code>	<code>severity>normal sortby time</code>	Displays the list of all events with severity higher than normal and sorted by the time element values.
<code>events</code>	<code>severity>normal sortby time desc</code>	Displays the list of all events with severity higher than normal and sorted by the time element values in descending order.

5.2. WILDCARDS

Search queries substitute part of a value with an asterisk as a wildcard.

Example 5.1. Wildcard search query for name=vm*

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

This query would result in all virtual machines with names beginning with `vm`, such as `vm1`, `vm2`, `vma` or `vm-webserver`.

Example 5.2. Wildcard search query for name=v*1

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


This query would result in all virtual machines with names beginning with **v** and ending with **1**, such as **vm1**, **vr1** or **virtualmachine1**.