Red Hat Storage 3
Console Command Line Shell Guide

A guide for installing and using the Command Line Shell for Red Hat Storage Console.

Pavithra Srinivasan

Red Hat Storage 3 Console Command Line Shell Guide

A guide for installing and using the Command Line Shell for Red Hat Storage Console.

Pavithra Srinivasan
Red Hat Engineering Content Services
psriniva@redhat.com

Legal Notice

Abstract

This following guide contains information for installing and using the Red Hat Storage Console Command Line Shell. Red Hat Storage Console Command Line Interface is under Technology Preview. The features are not fully supported under Red Hat subscription level agreements (SLAs), may not be functionally complete, and are not intended for production use. However, these features provide early access to upcoming product innovations, enabling customers to test functionality and provide feedback during the development process. As Red Hat considers making future iterations of Technology Preview features generally available, we will provide commercially reasonable efforts to resolve any reported issues that customers experience when using these features.

Table of Contents

Chapter 1. About the Command Line Interface

Introduction to the Command Line Interface

The Red Hat Storage Console features a command line interface (CLI). This CLI provides users with a means to connect to Red Hat Storage Console other than the standard web interface. The CLI also contains a scripting system that helps system administrators perform periodic maintenance or repetitive tasks in their environment with client machines.

Important

Red Hat Storage Console Command Line Interface is under Technology Preview. Hence, its features are not fully supported under Red Hat subscription level agreements (SLAs), may not be functionally complete, and are not intended for production use. However, these features provide early access to upcoming product innovations, enabling customers to test functionality and provide feedback during the development process. As Red Hat considers making future iterations of Technology Preview features generally available, we will provide commercially reasonable efforts to resolve any reported issues that customers experience when using these features.

Chapter 2. Using the CLI

2.1. Installing the CLI

Procedure 2.1. Installing the Red Hat Storage Console CLI on a Client Machine

1. Log into the client machine as the **root** user.
2. Subscribe to the Red Hat Storage Console Channels:

   Red Hat Satellite Server instructions

   ```
   # rhn_register
   # rhn-channel --add --channel=rhel-x86_64-server-6-rhs-rhsc-3
   ```

   Red Hat Content Delivery Network instructions:

   ```
   # subscription-manager list --available | grep -A8 "Red Hat Storage
   Console"
   # subscription-manager subscribe --pool=POOLID
   # yum-config-manager --enablerepo=rhsc-3-for-rhel-6-server-rpms
   ```

   Note

   See *Red Hat Storage Console Release Notes* for channel names specific to your system.

3. Install the CLI package and dependencies:

   ```
   # yum install rhsc-cli
   ```

2.2. TLS/SSL Certification

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

Important

Obtain your certificate from the Red Hat Storage Console using a secure network connection.

Procedure 2.2. Obtaining a Certificate

This process helps a user obtain a certificate from the Red Hat Storage Console and transfer it to the client machine. A user achieves this using one of three methods:

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

    a. If using **cURL**:

    ```
    curl -o rhsc.cer http://[rhsc-server]/ca.crt
    ```

    b. If using **Wget**:

    ```
    wget -O rhsc.cer http://[rhsc-server]/ca.crt
    ```

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

    ```
    http://[rhsc-server]/ca.crt
    ```

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

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

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

Each of the above three methods creates a certificate file named **rhsc.cer** on your client machine. An API user imports this file into the certificate store of the client.

2.3. Running the CLI

Start the CLI application with the following command:

```
# rhsc-shell
```

The **rhsc-shell** application is an interactive shell for Red Hat Storage Console.

Users connect automatically to Red Hat Storage Console using the following additional options.

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

Replace the following values:

- ***server*** - The hostname or IP address of the Red Hat Storage Console. The CLI connects to the Red Hat Storage Console via the REST API.
- ***user@domain*** - The user name and directory service domain for the user logging into Red Hat Storage Console.
- ***certificate*** - The path 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 Storage Console.

Note

The certificate is the only mandatory option; the other options used in this example will be prompted by the shell. Also, instead of specifying the certificate, you can use the **`--insecure`** option to connect without a certificate. However, this is not recommended as it may allow man-in-the-middle (MITM) attackers to spoof the identity of the server.

Table 2.1. Options for rhsc-shell

Command Options	Description
-h, --help	Show help for rhsc-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.
-I, --insecure	Allow the CLI to connect via SSL without certification. This option should be used with caution, as it may 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.
-f FILE, --file=FILE	Read commands from *FILE* instead of **`stdin`**.

Note

Although users can use the **`--file`** option to call a file that includes their password information, this practice is not recommended for security reasons. Users with a non-interactive shell can connect to the Red Hat Storage Console from within the shell, where the **`--password`** option can be used.

2.4. Interacting with the CLI

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

Example 2.1. Entering a shell command

```
[RHSC shell (connected)]# show cluster --name desktop_clusters
```

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 2.2. Listing and automatic completion of commands and parameters

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

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

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

```
show TAB TAB
brick      glustervolume      network      role
cluster    group              nic          statistic
domain     hook               permission   tag
event      host               permit       user
```

Pressing **TAB** twice also completes commands and parameters.

```
[RHSC shell (connected)]# show cluster TAB TAB
id          name
[RHSC shell (connected)]# show cluster naTAB TAB
[RHSC shell (connected)]# show cluster --name
```

Note that pressing **TAB** twice also automatically formats **na** to the **--name** parameter, including the prefix. If the incomplete parameter matches multiple parameters, pressing **TAB** twice lists all those parameters.

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

Example 2.3. Running Linux shell commands

Use the **shell** command:

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

Or use the bang (!) character:

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

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

Example 2.4. Piping CLI commands

Pipe CLI data to a Linux shell command:

```
[RHSC shell (connected)]# list clusters --show-all | grep "Example"
name                          : Example1
name                          : Example2
name                          : ExampleEngineering
description                   : An Example description
name                          : BestExampleCluster
```

Pipe CLI data to a file:

```
[RHSC shell (connected)]# list clusters --show-all > cluster_List.txt
```

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

Example 2.5. Using online help for the show command

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

2.5. Collections

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

```
[RHSC 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.

Chapter 3. Commands

3.1. Connecting to RHSC

3.1.1. Connect to RHSC via `connect`

The **`connect`** command connects to Red Hat Storage Console Application as follows:

```
# connect [options]
```

Table 3.1. Options for connect

Option	Required	Description
`--ca-file`	Yes, unless ***`--insecure`*** is used.	The certificate authority file to connect via SSL.
`--cert-file`	No	The certificate file to connect via SSL.
`--filter`	No	Enable filtering based on user permissions.
`--insecure`	Yes, but only if no certificate authority is provided.	Allow the CLI to connect via SSL without certification. Use this option with caution, as it may allow man-in-the-middle (MITM) attackers to spoof the identity of the server.
`--key-file`	No	The key file to connect via SSL.
`--password`	Yes	The password for the user attempting access to the Red Hat Storage Console Application.
`--port`	No	The port number to connect to the REST API, if not specified as part of the ***`--url`***.
`--session-timeout`	No	The timeout period for the session.
`--timeout`	-	The timeout period for connection.
`--url`	Yes	The URL to the Red Hat Storage Console's REST API. This takes the form of **`https://`*****`[server]`*****`/api`**.
`--user`	Yes	The user name and directory service domain of the user attempting access to the Red Hat Storage Console Application. This takes the form of *[username]@[domain]*.

Example 3.1. Example for connect

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

 ===========================================
 >>> connected to RHSC manager 3.3.0.0 <<<
 ===========================================

[RHSC shell (connected)]#
```

Note

Use the ***--insecure*** option to connect without certification. However, this is not recommended because it allows man-in-the-middle (MITM) attackers to spoof the server's identity.

3.1.2. Disconnect from RHSC Using `disconnect`

The **disconnect** command disconnects from Red Hat Storage Console Application as follows:

```
# disconnect
```

Example 3.2. Example for disconnect

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

  =======================================
  >>> disconnected from RHSC manager <<<
  =======================================

[RHSC shell (disconnected)]#
```

3.2. Resources

3.2.1. List Resources in a Collection using `list`

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

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

Table 3.2. Options for list

Option	Description
--show-all	Displays all non-empty properties for each listed resource. Without this option, only the **id**, **name** and **description** properties display.
--query *[QUERY]*	Filters the list using a server-side query based on Red Hat Storage Console's query language. For more information about search query language, refer to *Appendix E.1. Search* of the *Red Hat Storage Console Administration Guide*.
--kwargs *[QUERY]*	Filters the list using a client-side query.
--case_sensitive *true|false*	Toggle case sensitive searches for search queries.
--max	Maximum number of results to display.

Note

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

Example 3.3. Examples for list

List clusters:

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

List clusters with all properties listed:

```
[RHSC shell (connected)]# list clusters --show-all
```

List clusters which are named default:

```
[RHSC shell (connected)]# list clusters --query name=Default
```

3.2.2. Show a Resource Using show

Use the **show** command to display resource properties as follows:

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

Table 3.3. show standard options

Option	Description
--id *[UUID]*	Identify resource with the resource's UUID value.
--name *[NAME]*	Identify resource with the **name** value.

Note

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

Example 3.4. Examples for show

Show clusters based on **id**:

```
[RHSC shell (connected)]# show cluster --id fcadfd5f-9a12-4a1e-bb9b-
2b9d5c2e04c3
```

Show clusters based on **name**:

```
[RHSC shell (connected)]# show cluster --name TestCluster
```

3.2.3. Update a Resource Using `update`

Use the **update** command to modify an existing resource as follows:

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

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

Example 3.5. Examples for update

Update a cluster:

```
[RHSC shell (connected)]# update cluster TestCluster --name Cluster1
```

3.2.4. Remove a Resource Using `remove`

Use the **remove** command to remove a resource as follows:

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

Table 3.4. Options for remove

Option	Description
--async	Perform an asynchronous removal of the resource.
--force	Forcefully remove the resource. This removes all database entries and associations for a particular resource.

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

Example 3.6. Examples for remove

Remove a cluster:

```
[RHSC shell (connected)]# remove cluster TestCluster
```

Asynchronous removal of a cluster:

```
[RHSC shell (connected)]# remove cluster TestCluster --async
```

Force remove host:

```
[RHSC shell (connected)]# remove host Node1 --force true
```

See Also:

- Section 3.1.1, “Connect to RHSC via **connect**”

3.2.5. Perform Action on a Resource Using `action`

Certain actions can be performed on resources with CLI commands.

Example 3.7. Examples of Performing Actions on a Resource

Activate a host:

```
[RHSC shell (connected)]# action host Node1 activate
```

Deactivate a host:

```
[RHSC shell (connected)]# action host Node1 deactivate
```

3.2.6. Using Sub-Resources with `--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 used to target its sub-resource.

Example 3.8. Examples of Creating Sub-resources with add

To create a distributed volume:

```
RHSC shell (connected)]# add glustervolume --cluster-name TestCluster -
-name vol1 --volume_type DISTRIBUTE --bricks-brick
"brick.server_id=35cfcaa5-1b1e-4be7-
b87d-5a018ea98d98,brick.brick_dir=/home/brickInfo/b10" --bricks-brick
"brick.server_id=35cfcaa5-1b1e-4be7b87d-
5a018ea98d98,brick.brick_dir=/home/brickInfo/b20"
```

Note the use of the ***--cluster-name*** option. This adds a Red Hat Storage volume to the cluster called **TestCluster**.

3.3. Other Commands

3.3.1. `.rhscshellrc` Configuration

The **.rhscshellrc** configuration file contains options for connecting to the rhsc-shell. The **.rhscshellrc** file is available in the **/home/[user name]/** directory.

Section 3.1.1, “Connect to RHSC via **connect**”

Table 3.5. .rhscshellrc Parameters

Name	Type	Description
url	string	The address of the Red Hat Storage Console Application.
username	string	User name to be used to log in.
password	string	Password to be used for user name.
insecure	boolean	Toggles CA certificate requirement. The status is either **True** or **False**.
autopage	boolean	Toggles pagination in the shell. The status is either **True** or **False**.
filter	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**.
--session-timeout	integer	The timeout period for connection.
timeout	integer	Requests timeout. The default is **-1**.

See Also:

- Section 3.2.5, “Perform Action on a Resource Using **action**”
- Section 3.2.3, “Update a Resource Using **update**”

3.3.2. Clear the Screen Using clear

Use the **clear** command to clear the CLI screen as follows:

```
# clear
```

Example 3.9. Example for clear

Clear the screen:

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

3.3.3. Print Input Using echo

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

```
# echo [input]
```

Example 3.10. Example for echo

Print input:

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

Print last output:

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

3.3.4. Show Last Status Using `status`

Use the **`status`** command to display the most recently run command status as follows:

```
# status
```

Example 3.11. Example for status

View the last status:

```
[RHSC shell (connected)]# status

last command status: 0 (OK)
```

3.3.5. Show CLI Information Using `info`

Use the **`info`** command to display environment connection details and version information as follows:

```
# info
```

Example 3.12. Example for info

View CLI information:

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

backend version: 3.3
sdk version    : 2.1
cli version    : 2.1
python version : 2.6.6.final.0

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

3.3.6. Test Connection Using `ping`

Use the **`ping`** command to test the connection to your Red Hat Storage Console Application. The command retrieves a remote resource and ensures the URL, username and password for the connection are correct.

```
# ping
```

Example 3.13. Example for ping

Test your connection:

```
[RHSC shell (connected)]# ping

success: RHSC manager could be reached OK.
```

3.3.7. Exit from the CLI Using `exit`

Use the **exit** command to leave the CLI as follows:

```
# exit
```

Example 3.14. Example for exit

Leave the CLI:

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

3.3.8. End of File Using `EOF`

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

```
# EOF
```

Example 3.15. Example for EOF

Leave the CLI shell:

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

3.3.9. Show Help Using `help`

Use the **help** command displays help for CLI command and resource combinations as follows:

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

Example 3.16. Examples for help

Get CLI help:

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

Get help for the **add** command:

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

> Get help for the **`add`** command on the **`cluster`** resource type:
>
> ```
> [RHSC shell (connected)]# help add cluster
> ```

Chapter 4. Resource Types

4.1. brick

Important

The information in this section is provided as a technical preview only.

The **brick** resource type groups all Red Hat Storage bricks in a Red Hat Storage Console Application.

Table 4.1.

Name	Type	Description	Required	User Creatable	User Updatable
--brick_dir	string	The brick's directory on the Red Hat Storage server.	Yes	Yes	No
--server-id	string	The address of the Red Hat Storage server.	Yes	Yes	No

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

Table 4.2. glustervolume 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.
--all_content	Displays the advance options of a brick.

Example 4.1. Add a brick

```
[RHSC shell (connected)]# add brick --cluster-identifier 99408929-82cf-
4dc7-a532-9d998063fa95 --glustervolume-identifier 89999345-45fh-q234-
78f7793n --brick "brick.server_id=35cfcaa5-1b1e-4be7-b87d-
5a018ea98d98,brick.brick_dir=/home/brickInfo/b15"
```

Example 4.2. View the advance options of a brick

```
[RHSC shell (connected)]# list bricks --glustervolume-identifier
TestVol  --cluster-identifier TestCluster --show-all --all_content True
```

4.2. glustervolume

Important

The information in this section is provided as a technical preview only.

The **glustervolume** resource type groups all Red Hat Storage storage volume resources in the Red Hat Storage Console Application.

Table 4.3.

Name	Type	Description	Required	User Creatable	User Updatable
--name	string	The name of the Red Hat Storage volume.	Yes	Yes	No
--replica_count	int	Defines the file replication count for a replicated volume.	No	Yes	No
--stripe_count	int	Defines the stripe count for a striped volume.	No	Yes	No
--transport_types	collection	A reference to available transport methods for the Red Hat Storage volume. See below for collection details.	No	Yes	No
--options-option	collection	A reference to options for the Red Hat Storage volume. See below for collection details	No	Yes	No

Name	Type	Description	Required	User Creatable	User Updatable
--bricks-brick	collection	A new Red Hat Storage volume requires a set of Red Hat Storage 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
--volume_type	string	Defines the Red Hat Storage volume type. Choose from **DISTRIBUTE**, **REPLICATE**, **DISTRIBUTED_REPLICATE**, **STRIPE** or **DISTRIBUTED_STRIPE**.	Yes	Yes	No

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

Table 4.4. bricks-brick parameters

Name	Type	Description
brick.server_id	string	The address of the Red Hat Storage server.
brick.brick_dir	string	The brick's directory on the Red Hat Storage server.

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

Table 4.5. transport_types parameters

Name	Type	Description
transport_type	string	Defines a transport type to use. Specify multiple **transport_type** 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 4.6. options-option parameters

Name	Type	Description
option.name	string	The Red Hat Storage option name.
option.value	string	The Red Hat Storage option value.

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

Table 4.7. Additional command options

Option	Description
--cluster-identifier	Adds the Red Hat Storage volume to a cluster as a sub-resource.

Example 4.3. Creating a Red Hat Storage volume with two bricks

```
[RHSC 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 4.4. Deleting a Red Hat Storage volume

```
[RHSC shell (connected)]# remove glustervolume Vol1 --cluster-
identifier Cluster_bigbend
```

The following table lists actions for a Red Hat Storage volume resource.

Table 4.8. Red Hat Storage volume actions

Action	Description
start	Makes a Red Hat Storage volume available for use.
stop	Deactivates a Red Hat Storage volume.
setOption	Sets a Red Hat Storage volume option.
resetOption	Resets a Red Hat Storage volume option to the default.

Action	Description
resetAllOptions	Resets all Red Hat Storage volume options to defaults.
rebalance	moves the data across the brick upon shrinking or expanding a volume.
stopRebalance	Stops the currently executing rebalance process.
migrate	Pertains to starting a remove-brick operation.
stopMigrate	Pertains to stopping a remove-brick operation.
activate	Pertains to committing the data migration operation.

4.3. networks

The **network** resource type groups all logical network resources in a Red Hat Storage Console Application.

Table 4.9. Network parameters

Name	Type	Description	Required	User Creatable	User Updatable
--data_center-id|name	string	A reference to the data center for a logical network.	Yes	Yes	No
--name	string	A plain text name for the logical network.	Yes	Yes	No
--description	String	A description for the logical network.	No	Yes	Yes
--vlan-id	String	A VLAN tag.	No	Yes	Yes
--ip-address	String	The IP address for the logical network's bridge.	No	Yes	Yes
--ip-gateway	String	The gateway for the logical network's bridge.	No	Yes	Yes
--ip-netmask	String	The netmask for the logical network's bridge.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
`--display`	boolean	Signifies if a logical network is used for display communication usage. Set to either **`true`** or **`false`**.	No	Yes	Yes
`--stp`	boolean	Set to **`true`** if Spanning Tree Protocol is enabled on this network.	No	Yes	Yes
`--mtu`	int	Sets a user-defined value for the maximum transmission unit of the logical network.	No	Yes	Yes
`--usages-usage`	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 4.10. usages-usage parameters

Name	Type	Description
Usage	string	Usage types for the network.

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

Table 4.11. Additional command options

Option	Description
`--cluster-identifier`	Adds the network to a **`cluster`** as a sub-resource.

Example 4.5. Creating a new network

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

Example 4.6. Updating a network

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

Example 4.7. Deleting a network

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

4.4. permissions

The **`permission`** resource type groups all permission resources in a Red Hat Storage Console Application.

Table 4.12. Permission parameters

Name	Type	Description	Required	User Createable	User Updatable
`--role-id`	string	A reference to a role to assign for the permission.	Yes	Yes	No
`--user-id|group-id`	string	A reference to the user or group using the permission.	Yes	Yes	No

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

Table 4.13. Additional command options

Option	Description
`--cluster-identifier`	Adds the permission to a cluster.
`--group-identifier`	Adds the permission to a group.
`--host-identifier`	Adds the permission to a host.
`--network-identifier`	Adds the permission to a network.
`--user-identifier`	Adds the permission to a user.

Example 4.8. List permissions for a host

```
[RHSC shell (connected)]# list permissions --host-identifier Node1 --
show-all
```

4.5. roles

The **`role`** resource type groups all individual roles in a Red Hat Storage Console Application.

Table 4.14. Role parameters

Name	Type	Description	Required	User Createable	User Updatable
`--name`	string	The name of the role.	Yes	Yes	Yes
`--description`	string	A description for the role.	No	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
--permits-permit	collection	A list of permits for initial inclusion with the role. Additional permits included with the permit resource type.	Yes	Yes	No
--administrative	Boolean	Set to **true** 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 4.15. --permits-permit parameters

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

Example 4.9. Creating a new role

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

4.6. tags

The **tag** resource type groups all tags in a Red Hat Storage Console Application.

Table 4.16. Tag parameters

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

Example 4.10. Creating a new tag

```
[RHSC shell (connected)]# add tag --name MyTag --description "A TAG"
```

Example 4.11. Updating a new tag

```
[RHSC shell (connected)]# update tag MyTag --name Tag1
```

Example 4.12. Removing a tag

```
[RHSC shell (connected)]# remove tag Tag1
```

4.7. clusters

The **cluster** resource type groups all host cluster resources in the Red Hat Storage Console Application.

Note

Only the default data center is supported in Red Hat Storage Console Application.

Table 4.17. Cluster parameters

Name	Type	Description	Required	User Creatable	User Updatable
--name	string	The name of a host cluster.	Yes	Yes	Yes
--description	string	A description for the host cluster.	No	Yes	Yes
--datacenter-id | name	string	A reference to the data center for a host cluster.	Yes	Yes	No
--version-major	int	The major version number of the compatible version of cluster. For example, for Red Hat Storage Console 2.1, the major version is 2.	Yes	Yes	Yes
--version-minor	int	The minor version number of the compatible version of cluster. For example, for Red Hat Storage Console 2.1, the major version is 1.	Yes	Yes	Yes

Example 4.13. Creating a new cluster

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

Example 4.14. Updating a cluster

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

Example 4.15. Deleting a cluster

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

4.8. groups

The **group** resource type defines all identity service groups for a Red Hat Storage Console Application.

Table 4.18. User parameters

Name	Type	Description	Required	User Creatable	User Updatable
--name	string	The username from the directory service.	Yes	Yes	No

Example 4.16. List all the groups

```
[RHSC shell (connected)]#list groups --show-all
```

4.9. hosts

The **host** resource type groups all host resources in a Red Hat Storage Console.

Table 4.19.

Name	Type	Description	Required	User Creatable	User Updatable
--name	string	The name of the host.	Yes	Yes	Yes
--address	string	The IP address or hostname for the host.	Yes	Yes	Yes
--cluster-id | name	string	Defines the cluster that includes the host.	Yes	Yes	Yes

Name	Type	Description	Required	User Creatable	User Updatable
--port	int	The port for communication with the VDSM daemon running on the host.	No	Yes	Yes

Example 4.17. Creating a new host

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

Example 4.18. Updating a host

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

Example 4.19. Deleting a host

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

The following table lists actions for a host resource.

Table 4.20. Host actions

Action	Description
activate	Activate a host.
deactivate	Deactivate a host.
approve	Approve a host.
install	Install VDSM on a host.
commitnetconfig	Save the network configuration.

4.10. nic

The **nic** resource type groups network interface resources in a Red Hat Storage Console Application. This section contains two tables with parameters for each.

Table 4.21. Host network interface parameters

Name	Type	Description	Required	User Creatable	User Updatable
--id	string	A reference to the network, if any, that the interface is attached.	Yes	Yes	Yes
--name	string	The name of the host network interface, e.g. **eth0**.	Yes	Yes	Yes

Table 4.22. Additional command options

Option	Description
`--host-identifier`	Lists the NIC of a **`host`**.

Example 4.20. List all the associated network interfaces of a host

```
[RHSC shell (connected)]# list  nics --host-identifier Node1 --show-all
```

4.11. permit

The **`permit`** resource type groups all individual permits for roles in a Red Hat Storage Console Application.

Table 4.23. Permission parameters

Name	Type	Description	Required	User Creatable	User Updatable
`--id`	string	A reference to the permit to add.	Yes	Yes	No

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

Table 4.24. Additional command options

Option	Description
`--role-identifier`	Adds the permit to a role.

Example 4.21. List all permissions

```
[RHSC shell (connected)]# list permits --role-identifier ClusterAdmin -
-show-all
```

4.12. statistics

The **`Statistics`** resource type provides all the memory related parameters of a resource present in the shell.

Table 4.25. Statistics parameters

Name	Type	Description	Required	User Creatable	User Updatable
`--id`	string	A reference to the statistic	No	No	No
`--name`	string	The name of the statistic	No	No	No

Name	Type	Description	Required	User Creatable	User Updatable
--description	string	A description of the statistic	No	No	No

Example 4.22. List statistics of a host

```
[RHSC shell (connected)]# list statistics --host-identifier Node1
```

Example 4.23. To list all the non empty statistics of a host

```
[RHSC shell (connected)]# list statistics --host-identifier Node1  --
show -all
```

4.13. user

The **user** resource type groups all user resources in a Red Hat Storage Console Application.

Table 4.26. Permission parameters

Name	Type	Description	Required	User Creatable	User Updatable
--user_name	string	The username from the directory service.	Yes	Yes	No
--domain-id | name	string	A reference to the domain.	Yes	Yes	Yes

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

Example 4.24. List all users in a domain

```
[RHSC shell (connected)]# list users --domain-identifier internal --
show-all
```

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 Storage Console search query language:

Table 5.1.

Collections	Criteria	Result
hosts	**host.status=up**	Displays a list of all hosts running clusters that are **up**.
events	**severity>normal sortby time**	Displays the list of all **events** with severity higher than **normal** and sorted by the **time** element values.
events	**severity>normal sortby time desc**	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=cluster*

```
[RHSC shell (connected)]# list cluster --query "name=cluster*"
```

This query would result in all clusters with names beginning with cluster, such as **cluster1**, **cluster2**, **clustera** or **cluster-webserver**.

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

```
[RHSC shell (connected)]# list cluster --query "name=c*1"
```

This query would result in all clusters with names beginning with c and ending with **1**, such as **ccluster1**, **cclusters1** or **cclusterszoneone1**.

Revision History

Revision 3-11	**Mon Sep 22 2014**	**Shalaka Harne**

Version for 3.0 GA release.