Red Hat Enterprise Linux OpenStack Platform 5 Cloud Administrator Guide

Managing and troubleshooting a Red Hat Enterprise Linux OpenStack Platform environment

6 Aug 2014 Red Hat Documentation Team

Red Hat Enterprise Linux OpenStack Platform 5 Cloud Administrator Guide

Managing and troubleshooting a Red Hat Enterprise Linux OpenStack Platform environment

6 Aug 2014

Red Hat Documentation Team

Abstract

This guide covers the software administrators can use to manage and troubleshoot a Red Hat Enterprise Linux OpenStack Platform cloud.

Table of Contents

Chapter 1. Block Storage **2**
1.1. Configure a multiple-storage back-end 2

Enable multi back-end **2**

Configure Block Storage scheduler multi back-end **3**

Volume type **3**

Usage **4**

Chapter 1. Block Storage

1.1. Configure a multiple-storage back-end

With multiple storage back-ends configured, you can create several back-end storage solutions serving the same OpenStack Compute configuration. Basically, multi back-end launches one **cinder-volume** for each back-end or back-end pool.

In a multi back-end configuration, each back-end has a name (**volume_backend_name**). Several back-ends can have the same name. In that case, the scheduler properly decides which back-end the volume has to be created in.

The name of the back-end is declared as an extra-specification of a volume type (such as, **volume_backend_name=LVM_iSCSI**). When a volume is created, the scheduler chooses an appropriate back-end to handle the request, according to the volume type specified by the user.

Enable multi back-end

To enable a multi back-end configuration, you must set the **enabled_backends** flag in the **cinder.conf** file. This flag defines the names (separated by a comma) of the configuration groups for the different back-ends: one name is associated to one configuration group for a back-end (such as, **[lvmdriver-1]**).

Note

The configuration group name is not related to the **volume_backend_name**.

The options for a configuration group must be defined in the group (or default options are used). All the standard Block Storage configuration options (**volume_group**, **volume_driver**, and so on) might be used in a configuration group. Configuration values in the **[DEFAULT]** configuration group are not used.

This example shows three back-ends:

```
enabled_backends=lvmdriver-1,lvmdriver-2,lvmdriver-3
[lvmdriver-1]
volume_group=cinder-volumes-1
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI
[lvmdriver-2]
volume_group=cinder-volumes-2
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI
[lvmdriver-3]
volume_group=cinder-volumes-3
volume_driver=cinder.volume.drivers.lvm.LVMISCSIDriver
volume_backend_name=LVM_iSCSI_b
```

In this configuration, **lvmdriver-1** and **lvmdriver-2** have the same **volume_backend_name**. If a volume creation requests the **LVM_iSCSI** back-end name, the scheduler uses the capacity filter scheduler to choose the most suitable driver, which is either **lvmdriver-1** or **lvmdriver-2**. The capacity filter scheduler is enabled by default. The next section provides more information. In addition, this example presents a **lvmdriver-3** back-end.

Some volume drivers require additional settings to be configured for each back-end. The following example shows the typical configuration for a Block Storage service that uses two Dell EqualLogic back-ends:

```
enabled_backends=backend1,backend2
san_ssh_port=22
ssh_conn_timeout=30

[backend1]
volume_driver=cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
volume_backend_name=backend1
san_ip=IP_EQLX
san_login=SAN_UNAME
san_password=SAN_PW
eqlx_group_name=EQLX_GROUP
eqlx_pool=EQLX_POOL

[backend2]
volume_driver=cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver
volume_backend_name=backend2
san_ip=IP_EQLX
san_login=SAN_UNAME
san_password=SAN_PW
eqlx_group_name=EQLX_GROUP
eqlx_pool=EQLX_POOL
```

In this example:

- Each Dell EqualLogic back-end configuration (**[backend1]** and **[backend2]**) has the same required settings as a single back-end configuration, with the addition of ***volume_backend_name***.
- The ***san_ssh_port*** option is set to its default value, **22**. This option sets the port used for SSH.
- The ***ssh_conn_timeout*** option is also set to its default value, **30**. This option sets the timeout (in seconds) for CLI commands over SSH.

For more information on required and optional settings for Dell EqualLogic back-ends, refer to the *Configuration Reference Guide*:

Dell EqualLogic volume driver

Configure Block Storage scheduler multi back-end

You must enable the **filter_scheduler** option to use multi back-end. Filter scheduler acts in two steps:

1. The filter scheduler filters the available back-ends. By default, **AvailabilityZoneFilter**, **CapacityFilter** and **CapabilitiesFilter** are enabled.
2. The filter scheduler weighs the previously filtered back-ends. By default, **CapacityWeigher** is enabled. The **CapacityWeigher** attributes higher scores to back-ends with the most available capacity.

The scheduler uses the filtering and weighing process to pick the best back-end to handle the request, and explicitly creates volumes on specific back-ends through the use of volume types.

Volume type

Before using it, a volume type has to be declared to Block Storage. This can be done by the following command:

```
$ cinder --os-username admin --os-tenant-name admin type-create lvm
```

Then, an extra-specification has to be created to link the volume type to a back-end name. Run this command:

```
$ cinder --os-username admin --os-tenant-name admin type-key lvm set
volume_backend_name=LVM_iSCSI
```

This example creates a **lvm** volume type with **volume_backend_name=LVM_iSCSI** as extra-specifications.

Create another volume type:

```
$ cinder --os-username admin --os-tenant-name admin type-create lvm_gold
```

```
$ cinder --os-username admin --os-tenant-name admin type-key lvm_gold set
volume_backend_name=LVM_iSCSI_b
```

This second volume type is named **lvm_gold** and has **LVM_iSCSI_b** as back-end name.

Note

To list the extra-specifications, use this command:

```
$ cinder --os-username admin --os-tenant-name admin extra-specs-list
```

Note

If a volume type points to a **volume_backend_name** that does not exist in the Block Storage configuration, the **filter_scheduler** returns an error that it cannot find a valid host with the suitable back-end.

Usage

When you create a volume, you must specify the volume type. The extra-specifications of the volume type are used to determine which back-end has to be used.

```
$ cinder create --volume_type lvm --display_name test_multi_backend 1
```

Considering the **cinder.conf** described previously, the scheduler creates this volume on **lvmdriver-1** or **lvmdriver-2**.

```
$ cinder create --volume_type lvm_gold --display_name test_multi_backend 1
```

This second volume is created on **lvmdriver-3**.