



Red Hat Enterprise Linux OpenStack Platform 7 Dell EqualLogic Back End Guide

A Guide to Using Dell EqualLogic Storage in a RHEL OpenStack Platform Environment

OpenStack Team

Red Hat Enterprise Linux OpenStack Platform 7 Dell EqualLogic Back End Guide

A Guide to Using Dell EqualLogic Storage in a RHEL OpenStack Platform Environment

OpenStack Team
rhos-docs@redhat.com

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to configure OpenStack to use one or more Dell EqualLogic back ends.

Table of Contents

CHAPTER 1. INTRODUCTION	3
CHAPTER 2. USE A SINGLE DELL EQUALLOGIC BACK END FOR OPENSTACK BLOCK STORAGE	4
2.1. DEFINE THE BACK END	4
2.2. LOAD THE NECESSARY ADMINISTRATOR CREDENTIALS	5
2.3. CREATE A VOLUME TYPE AND RESTART BLOCK STORAGE SERVICE	6
2.4. TEST YOUR CONFIGURATION	6
CHAPTER 3. USE MULTIPLE DELL EQUALLOGIC BACK ENDS FOR OPENSTACK BLOCK STORAGE ..	7
3.1. DEFINE EACH BACK END	7
3.2. LOAD THE NECESSARY ADMINISTRATOR CREDENTIALS	8
3.3. CONFIGURE THE VOLUME SERVICE	9
3.4. TEST YOUR CONFIGURATION	10
CHAPTER 4. ADDRESS VOLUME SIZE DISCREPANCIES WITH DELL EQUALLOGIC BACK ENDS	11
4.1. EXAMPLE	11
4.2. WORKAROUND	12

CHAPTER 1. INTRODUCTION

This document describes how to configure OpenStack to use one or more Dell EqualLogic back ends. It also includes instructions on addressing volume size discrepancies between Dell EqualLogic and the OpenStack Block Storage service.

The following chapters assume that:

- ✧ OpenStack has already been deployed with a properly-configured Block Storage service
- ✧ A Dell EqualLogic Group is already deployed and accessible through SSH
- ✧ No other back end is configured aside from the Dell EqualLogic devices
- ✧ The target back ends require CHAP authentication
- ✧ You have the necessary credentials for connecting to the Group manager of the available Dell EqualLogic Group (namely, CHAP and Group manager credentials)
- ✧ You have the username and password of an admin account for the OpenStack deployment (see "User and Role Management" in the *Users and Identity Management Guide*, or [Creating additional OpenStack admin users](#) for more information)

CHAPTER 2. USE A SINGLE DELL EQUALLOGIC BACK END FOR OPENSTACK BLOCK STORAGE

This section describes how to configure OpenStack to use a single Dell EqualLogic device as a Block Storage back end.

2.1. DEFINE THE BACK END

In a deployment that uses a single back end, the settings for the back end are defined in the **[DEFAULT]** section. The following snippet displays the different settings required for using a Dell EqualLogic SAN device (in this case, the device is named **eqlsan**):

```
[DEFAULT]
volume_driver=cinder.volume.drivers.eq1x.DellEQ1SanISCSIDriver 1
volume_backend_name=eqlsan
san_thin_provision=true 2
san_ip=10.1.1.1 3
san_login=grpadmin 4
san_password=password 5
eq1x_group_name=group-0 6
eq1x_pool=default 7
eq1x_use_chap=true 8
eq1x_chap_login=chapadmin 9
eq1x_chap_password=8871a772209e70be7222 10
```

1

volume_driver: The volume driver required for Dell EqualLogic back ends (namely, **cinder.volume.drivers.eq1x.DellEQ1SanISCSIDriver**).

2

san_thin_provision: Sets whether thin provisioning for SAN volumes is enabled (**true**), as is required for this setup.

3

san_ip: The IP address used to reach the Dell EqualLogic Group through SSH. This field has no default value.

4

san_login: The user name to login to the Group manager via SSH at the **san_ip**. The default user name is **grpadmin**.

5

san_password: The corresponding password of **san_login**. Default password is **password**.

6

eqlx_group_name: The group to be used for a pool where the Block Storage service will create volumes and snapshots. Default group is **group-0**.

7

eqlx_pool: The pool where the Block Storage service will create volumes and snapshots. Default pool is **default**. This option cannot be used for multiple pools utilized by the Block Storage service on a single Dell EqualLogic Group.

8

eq_l_use_chap: Sets whether CHAP authentication is enabled (**true**).

9

eqlx_chap_login: The CHAP login account for each volume in a pool. Default account name is **chapadmin**.

10

eqlx_chap_password: The corresponding password of **eqlx_chap_login**. The default password is randomly generated in hexadecimal, so you must set this password manually.



Note

See the [Configuration Reference Guide](#) for all supported optional settings for the Dell EqualLogic volume driver.

2.2. LOAD THE NECESSARY ADMINISTRATOR CREDENTIALS

As you will be performing administrative functions from this point onwards, you will need to load the necessary environment variables to facilitate authentication. To do so, run the following commands:

```
# export OS_USERNAME=ADMIN_USER
# export OS_TENANT_NAME=admin
# export OS_PASSWORD=ADMIN_PW
# export OS_AUTH_URL=http://KEYSTONE_IP:35357/v2.0/
# export PS1='[\u@\h \W(keystone_admin)]\$'
```

Where:

✎ *ADMIN_USER* and *ADMIN_PW* are the username/password of a user account with administrative rights within the OpenStack environment.

✎ *KEYSTONE_IP* is the IP address or hostname of the Identity service.

For more information about OpenStack admin accounts, see [Creating additional OpenStack admin users](#).

2.3. CREATE A VOLUME TYPE AND RESTART BLOCK STORAGE SERVICE

After defining the back end, create a volume type for it. The following commands will create the volume type `lvm` and map it to the back end `eqlsan` (from [Section 2.1, “Define the Back End”](#)):

```
# cinder type-create lvm
# cinder type-key lvm set volume_backend_name=eqlsan
```

Then, restart the Block Storage service:

```
# openstack-service restart cinder
```

2.4. TEST YOUR CONFIGURATION

Verify your configuration by creating a 1GB volume named `test_backend`. To do so:

```
# cinder create --volume_type lvm --display_name test_backend 1
```

CHAPTER 3. USE MULTIPLE DELL EQUALLOGIC BACK ENDS FOR OPENSTACK BLOCK STORAGE

This section describes how to configure OpenStack to use multiple Dell EqualLogic devices as Block Storage back ends.

3.1. DEFINE EACH BACK END

Start by creating a section for each back end in the `/etc/cinder/cinder.conf` file of the node hosting the **openstack-cinder-volume** service. The following snippet defines two back ends, **[backend1]** and **[backend2]**:

```
[backend1]
volume_driver=cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver 1
volume_backend_name=backend1 2
san_ip=10.1.1.1 3
san_login=grpadmin 4
san_password=password 5
eqlx_group_name=group-0 6
eqlx_pool=default 7
eqlx_use_chap=true 8
eqlx_chap_login=chapadmin 9
eqlx_chap_password=2399ad0852787563345b 10

[backend2]
volume_driver=cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver 1
volume_backend_name=backend2 2
san_ip=10.1.1.2 3
san_login=grpadmin 4
san_password=password 5
eqlx_group_name=group-0 6
eqlx_pool=default 7
eqlx_use_chap=true 8
eqlx_chap_login=chapadmin 9
eqlx_chap_password=4bde24fdac187ce557b5 10
```

1 11

volume_driver: The volume driver required for Dell EqualLogic back ends (namely, **cinder.volume.drivers.eqlx.DellEQLSanISCSIDriver**).

2 12

volume_backend_name: Defines each back end's name. Each back end must have a unique name.

3 13

san_ip: The IP address used to reach the Dell EqualLogic Group through SSH. This field has no default value.

4 14

san_login: The user name to login to the Group manager via SSH at the **san_ip**. The default user name is **grpadmin**.

5 15

san_password: The corresponding password of **san_login**. Default password is **password**.

6 16

eqlx_group_name: The group to be used for a pool where the Block Storage service will create volumes and snapshots. Default group is **group-0**.

7 17

eqlx_pool: The pool where the Block Storage service will create volumes and snapshots. Default pool is **default**. This option cannot be used for multiple pools utilized by the Block Storage service on a single Dell EqualLogic Group.

8 18

eql_use_chap: Sets whether CHAP authentication is enabled (**true**).

9 19

eqlx_chap_login: The CHAP login account for each volume in a pool. Default account name is **chapadmin**.

10 20

eqlx_chap_password: The corresponding password of **eqlx_chap_login**. The default password is randomly generated in hexadecimal, so you must set this password manually.

3.2. LOAD THE NECESSARY ADMINISTRATOR CREDENTIALS

As you will be performing administrative functions from this point onwards, you will need to load the

necessary environment variables to facilitate authentication. To do so, run the following commands:

```
# export OS_USERNAME=ADMIN_USER
# export OS_TENANT_NAME=admin
# export OS_PASSWORD=ADMIN_PW
# export OS_AUTH_URL=http://KEYSTONE_IP:35357/v2.0/
# export PS1='[\u@\h \W(keystone_admin)]\$'
```

Where:

- ✎ *ADMIN_USER* and *ADMIN_PW* are the username/password of a user account with administrative rights within the OpenStack environment.
- ✎ *KEYSTONE_IP* is the IP address or hostname of the Identity service.

For more information about OpenStack admin accounts, see [Creating additional OpenStack admin users](#).

3.3. CONFIGURE THE VOLUME SERVICE

1. After defining each back end, configure the volume service to use each of them. To do so, set the defined back ends as a comma-delimited list to the `enabled_backends` setting in the **[DEFAULT]** section of `/etc/cinder/cinder.conf`. For example, to set `backend1` and `backend2` (from [Section 3.1, “Define Each Back End”](#)) as your back ends, run:

```
# openstack-config --set /etc/cinder/cinder.conf DEFAULT
enabled_backends backend1,backend2
```

2. Next, declare a *volume type* for each back end. Later on, when you create a volume, you can use the volume type to specify which back end the Block Storage service should use for creating the volume. The following commands will allow you to create two volume types: `eq1` and `eq2`:

```
# cinder type-create eq1# cinder type-create eq2
```

Likewise, map the `eq2` volume type to `backend2` (also from [Section 3.1, “Define Each Back End”](#)):

3. Configure the Block Storage service to intelligently determine which back end to use for a specific request:

```
# openstack-config --set /etc/cinder/cinder.conf DEFAULT
scheduler_default_filters CapacityFilter
```

With this, the Block Storage service will choose from the configured back ends based on each one's capacity.

4. Enable thin provisioning for SAN volumes:

```
# openstack-config --set /etc/cinder/cinder.conf DEFAULT
san_thin_provision true
```

5. Restart the Block Storage service:

```
# openstack-service restart cinder
```

3.4. TEST YOUR CONFIGURATION

Verify your configuration by creating a 1GB volume backed by **backend2** and named **test_backend**:

```
# cinder create --volume_type eql2 --display_name test_backend 1
```

CHAPTER 4. ADDRESS VOLUME SIZE DISCREPANCIES WITH DELL EQUALLOGIC BACK ENDS

When reporting volume sizes, Dell EqualLogic (EQL) back ends also account for additional storage to be used for internal volume metadata. This size will be slightly larger than the volume size reported by the Block Storage services. However, the volume size reported by an EQL back end is the same one used by the Image service.

As a result, when creating an image-backed volume on an EQL back end, check the size of the image first. If the image was originally volume-backed, then EQL (and, by extension, the Image service) will be reporting a volume size slightly larger than what is reported by the Block Storage service.

If the image size reported by EQL is slightly larger, then you need to take the size discrepancy into consideration when creating volumes backed by this image.

4.1. EXAMPLE

To illustrate, when you create a 1GB volume:

```
# cinder create --display-name vol1 1
```

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-12-19T03:57:47.730359
display_description	None
display_name	vol1
encrypted	False
id	6bdace69-bd41-42fc-a63a-f834fb65a2e4
metadata	{}
size	1
snapshot_id	None
source_vol_id	None
status	creating
volume_type	None

The Block Storage service will report a volume size of 1GB, but on the EQL array the size (VolReserve) will be slightly bigger:

```
eql> volume select volume-6bdace69-bd41-42fc-a63a-f834fb65a2e4
```

```
eql (volume-6bdace69-bd41-42fc-a63a-f834fb65a2e4)> show
```

```

_____ Volume Information _____...
Name: volume-6bdace69-bd41-42fc-a63a-f834fb65a2e4
Size: 1GB
VolReserve: 1.01GB
...
```

When you create a new image from this volume, **cinder** will report a correct volume size of 1GB:

```
# cinder upload-to-image --disk-format raw --container-format bare vol1
image_vol1
```

Property	Value
container_format	bare
disk_format	raw
display_description	None
id	6bdace69-bd41-42fc-a63a-f834fb65a2e4
image_id	c65f7eae-e2c1-44ba-8af1-e33695897559
image_name	image_vol1
size	1
status	uploading
updated_at	2014-12-19T03:57:48.000000
volume_type	None

However, the Image service will report a slightly larger size:

```
# glance image-list
```

Name	Disk Format	Container Format	Size	Status
image_vol1	raw	bare	1085276160	active

The **glance** tool reports an image size of approximately 1.01GB. As a result, creating a new 1GB volume backed by this image will fail:

```
# cinder create --display-name vol2 --image-id c65f7eae-e2c1-44ba-8af1-
e33695897559 1
```

```
ERROR: Invalid input received: Size of specified image 2 is larger than
volume size 1
```

4.2. WORKAROUND

As mentioned earlier, you need to consider the discrepancy between the volume sizes reported by the Image and the Block Storage services when specifying the size of image-backed volumes. This means that when specifying the size of the image-backed volume, use the **next whole number after the image size reported by glance**.

Using the previous example, **glance** reported an image size of 1.01GB. This means that when you create a volume, you need to specify a volume size of 2GB instead of 1GB:

```
# cinder create --display-name vol2 --image-id c65f7eae-e2c1-44ba-8af1-
```


e33695897559 2

Property	Value
attachments	[]
availability_zone	nova
bootable	false
created_at	2014-12-19T04:54:07.036260
display_description	None
display_name	vol2
encrypted	False
id	fcf49715-094d-4bba-9f05-8b7fa6deffce
image_id	c65f7eae-e2c1-44ba-8af1-e33695897559
metadata	{}
size	2
snapshot_id	None
source_volid	None
status	creating
volume_type	None