Hewlett Packard
Enterprise

OpenStack HPE StoreVirtual Block Storage Driver Configuration Best Practices

OpenStack Liberty update

Technical white paper

Contents

Revision history

REV	DATE	DESCRIPTION
1.0	20-Apr-2014	Initial release to support the OpenStack® Icehouse release
2.0	16-Oct-2014	Update for OpenStack Juno release • Admin Horizon UI now supports adding extra-specs settings • Added documentation for Volume retype (driver support was added in Icehouse)
3.0	30-Apr-2015	Update for OpenStack Kilo release • Added cinder volume manage and unmanage support • Block Storage scheduler configuration with driver filter and weigher
4.0	16-Oct-2015	Update for OpenStack Liberty Release • Support added for configuring the over subscription ratio and reserved percentage for thin provisioned volumes • Enabled support for consistency groups • The **legacy mode** version of the driver was marked as deprecated and will be removed in a future release • Added support for ClusterHQ Flocker open source technology

NOTE
This document should be used for all OpenStack releases up to and including the Liberty release. A new document will be created for the OpenStack Mitaka release which will be the first release that will contain the HPE re-branded Cinder drivers.

Executive summary

HPE brings the power of OpenStack to the enterprise. HPE's commitment to the OpenStack community brings new and enhanced offerings that enable enterprises to increase agility, speed innovation, and lower costs.

Since the Grizzly release, HPE has been a top contributor to the advancement of the OpenStack Project.[1] HPE's contributions have focused on continuous integration and quality assurance, which has supported the development of a reliable and scalable cloud platform that is equipped to handle production workloads.

To support the need that many larger organizations and service providers have for enterprise-class storage, HPE has developed the HPE StoreVirtual Block Storage Driver, which supports the OpenStack technology using the iSCSI protocol. This provides the flexibility and cost-effectiveness of a cloud-based open-source platform to customers with mission-critical environments and high resiliency requirements.

Figure 1 shows the high-level components of a basic cloud architecture.

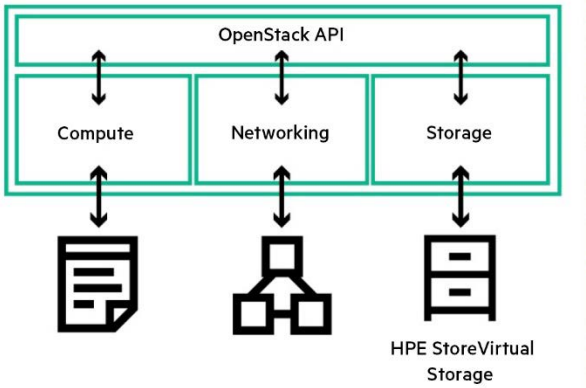

Figure 1. OpenStack cloud architecture

[1] Stackalytics.com, "OpenStack Liberty Analysis," September 2015. stackalytics.com/?release=liberty&metric=commits&project_type=openstack&metric=commits

Introduction

This document provides information about the new best practice features in the OpenStack Kilo release. These include configuring and using volume types, extra specs, consistency groups, over subscription and multiple backend support with the HPE StoreVirtual Block Storage Driver.

The HPE StoreVirtualiSCSI Driver is based on the Block Storage (Cinder) plug-in architecture. The driver executes the volume operations by communicating with the HPE StoreVirtual storage system over HTTPS and Secure Shell (SSH) connections. The HTTPS communication uses the HPE LeftHand Client, which is part of the Python Package Index (PyPi).

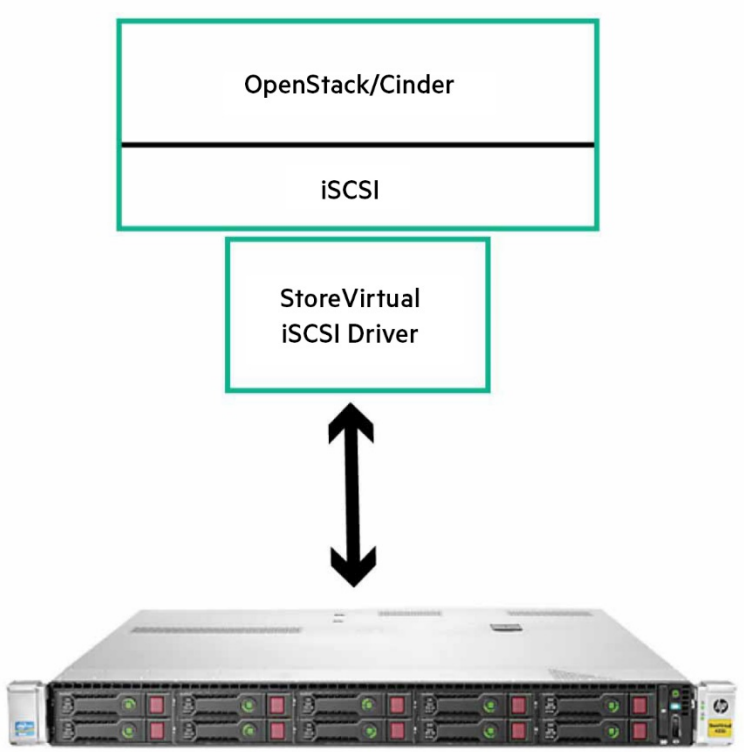

Figure 2. HPE StoreVirtualiSCSI Driver for OpenStack/Cinder

HPE StoreVirtual Storage

HPE StoreVirtual Storage uses a single architecture, shown in figure 3, to deliver primary storage platforms for midrange, enterprise, and optimized all-flash arrays.

HPE StoreVirtual Storage, based on the HPE LeftHand operating system, is a scale-out storage platform that is designed to meet the fluctuating needs of virtualized environments. Intuitive, common management, and storage federation meet the need for simplicity and flexibility in today's virtual data centers.

It allows data mobility across tiers, locations, and between physical and virtual storage. HPE StoreVirtual is the most versatile storage platform on the market today.

Its software-defined storage VSA software and ProLiant rack and BladeSystem-based hardware models provide options to fit any infrastructure and budget. Enterprise-class storage software functionality and leading virtualization software integration are built-in.

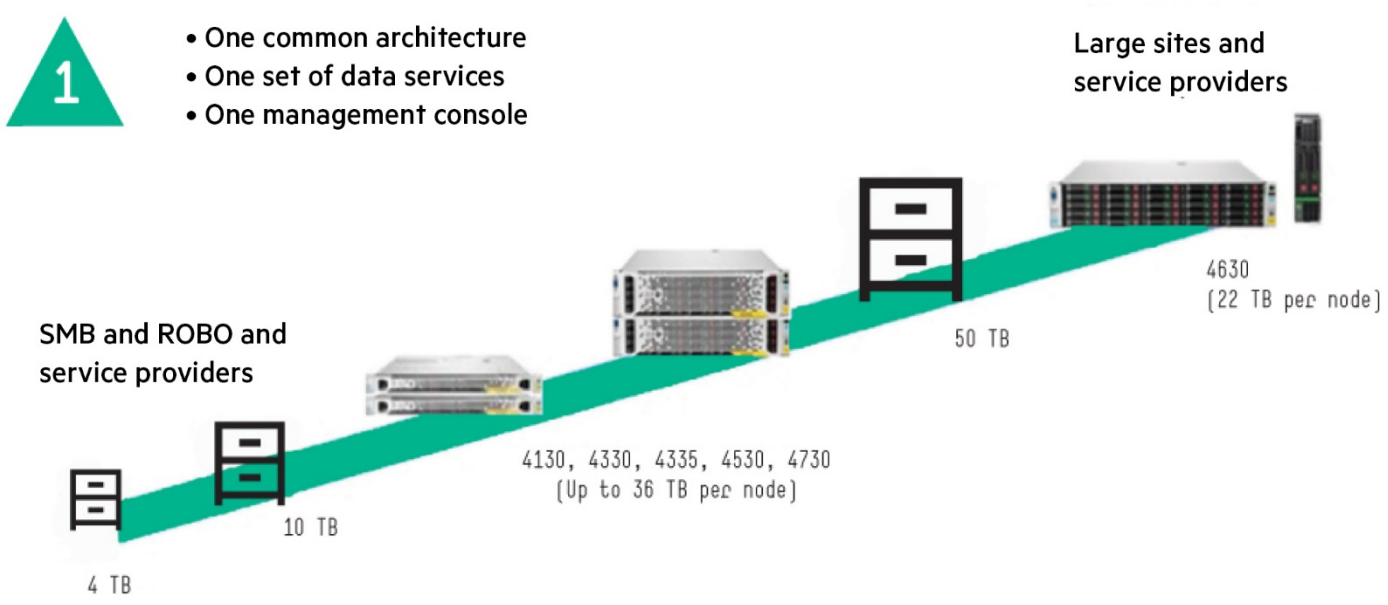

Figure 3. HPE StoreVirtual Storage

Configuration

The HPE StoreVirtual Block Storage Driver for iSCSI was introduced in the OpenStack Icehouse release.

The HPE StoreVirtual driver can be configured to run in one of two possible modes; **legacy mode** which uses SSH/CLIQ to communicate with the HPE StoreVirtual Storage, or **standard mode**, which uses a new REST client to communicate with the array. No new functionality has been, or will be, supported in legacy mode. For performance improvements and new functionality, the following conditions must be met:

- The driver must be configured for standard mode.
- The HPE LeftHand Client must be downloaded.
- The array requires the HPE LeftHand/StoreVirtual Operating System software version 11.5 or higher.

Kilo

- The array requires the HPE LeftHand/StoreVirtual Operating System software version 12.0 or higher only if volume manage and unmanage is going to be used.

Liberty

- Cinder's **cinder.conf** settings of `max_over_subscription_ratio` is used for over subscription of thin provisioned volumes and reserved_percentage to prevent over provisioning on the StoreVirtual.
- The **legacy mode** version of the driver which uses SSH/CLIQ was marked as deprecated in the Liberty release.

Volume types creation

Block storage volume types are a type or label that can be selected at volume create time in OpenStack. These types can be created either in the Admin Horizon UI or using the command line, as shown. This capability is only available when the driver is configured to execute in **standard mode**.

```
$cinder --os-username admin --os-tenant-name admin type-create <name>
```

The **<name>** is the name of the new volume type. After the volume type names have been created, you can assign extra specs or capabilities to these types. This will affect the block storage driver's behavior, depending on which extra specs you assign to each type you created.

This example illustrates how to create three new volume type names with the names gold, silver, and bronze:

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

$cinder --os-username admin --os-tenant-name admin type-create silver

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

Setting extra_specs or capabilities

The extra_specs or capabilities must be set/unset for a volume type. The extra_specs are set or unset either in the Admin Horizon UI (new in Juno) or using the command line, as shown:

```
$cinder --os-username admin --os-tenant-name admin type-key <vtype><action>
     [<key=value> [<key=value> ...]]
```

The argument **<vtype>** is the name or ID of the volume type. The argument **<action>** must be one of the actions: set or unset. The optional argument <key=value> is the extra_specs to set/unset the key. Only the key is necessary on unset. When extra_specs are set on a volume type, you must add the following **key value pair** to connect it to a particular block storage driver.

The <volume backend name> must match the same value that is in the **cinder.conf** file for a particular block storage driver. Each volume type must also have a volume_backend_name assigned to it.

```
volume_backend_name=<volume backend name>
```

"Set" examples

```
$cinder type-key gold set hplh:data_pl=r-6 hplh:provisioning=full hplh:ao=true

$cinder type-key silver set hplh:data_pl=r-5 hplh:provisioning=full

$cinder type-key bronze set hplh:data_pl=r-0 hplh:provisioning=thin
```

"Unset" examples

```
$cinder type-key gold unset hplh:ao
```

When you want to unset a particular key value pair from a volume type, the value is not required—only the key.

The current StoreVirtual extra_specs that can be specified in the Icehouse release require a scoping of the LeftHand keys. The filter scheduler uses the extra-specs data to determine capabilities and the backend. It also enforces strict checking.

Therefore, the scheduler requires scoping or adding, **"hplh:"** to the LeftHand keys. The current list of supported LeftHand keys include:

- hplh:provisioning—The valid values are **thin** and **full**. Default **thin**.
- hplh:ao—The valid values are **true** and **false**. Default **true**.
- hplh:data_pl—The valid values are: Default **r-0**.
  - r-0 Network RAID-0 (None)
  - r-5 Network RAID-5 (Single Parity)
  - r-10-2 Network RAID-10 (2-Way Mirror)
  - r-10-3 Network RAID-10 (3-Way Mirror)
  - r-10-4 Network RAID-10 (4-Way Mirror)
  - r-6 Network RAID-6 (Dual Parity)

Any or all of the above capabilities can be set on a volume type. These are additional capabilities that the HPE StoreVirtual Storage array offers. Use the following command to list all the volume types and extra_specs currently configured:

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

Multiple storage backend support and block storage configuration

Multiple backend support was added in the Grizzly release to OpenStack. Detailed instructions on setting up multiple backends can be found in the OpenStack Cloud Administrator Guide.

The multi-backend configuration is done in the **cinder.conf** file. The enabled_backends flag has to be set up. This flag defines the names (separated by a comma) of the configuration groups for the different backends. One name is associated to one configuration group for a backend (e.g., [lefthand-1]). Each group must have a full set of the driver-required configuration options. The list shown in figure 4 shows the entries in **cinder.conf** for two different HPE StoreVirtual Storage array backends, with one driver configured in legacy mode (Deprecated in the Liberty release) and one driver configured in standard mode.

Note

Currently, when the StoreVirtual Driver is configured to execute in standard mode, it communicates with the HPE StoreVirtual Storage array over **HTTPS**. When the driver is configured to execute in legacy mode, it communicates with the HPE StoreVirtual array over **SSH/CLIQ**. This means that each backend configuration uses different configuration entries in the **cinder.conf** file.

```
# List of backends that will be served by this node

enabled_backends=lefthand-rest-1,lefthand-rest-2,lefthand-cliq

# [lefthand-rest-1]
volume_driver=cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver

volume_backend_name=lefthand-rest
hplefthand_api_url=https://10.10.22.241:8081/lhos

hplefthand_username=<username>

hplefthand_password=<password>

hplefthand_clustername=lefthandSmallCluster

max_over_subscription_ratio=10.0

reserved_percentage=15

# [lefthand-rest-2]
volume_driver=cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver

volume_backend_name=lefthand-rest
hplefthand_api_url=https://10.10.22.244:8081/lhos

hplefthand_username=<username>

hplefthand_password=<password>

hplefthand_clustername=lefthandLargeCluster

# [lefthand-cliq]
volume_driver=cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver

volume_backend_name=lefthand-cliq

san_ip=10.10.22.6

san_login=<san_username>

san_password=<san_password>

san_ssh_port=16022

san_clustername=ClusterVSA440A0
```

Figure 4. Sample cinder.conf file

In this configuration, both the lefthand-rest-1 and lefthand-rest-2 have the same volume_backend_name. When a volume request comes in with the LeftHand backend name, the scheduler must choose which one is most suitable. This is done with the capacity filter scheduler detailed in the section below. This example also includes a single, legacy-configured LeftHand Cinder driver with a different volume_backend_name.

The StoreVirtual driver configured for legacy mode (Deprecated in the Liberty release) provides the ability to communicate with the HPE StoreVirtual Storage using the common SSH/CLIQ communication method. This is handy if the HPE StoreVirtual Storage array is running a LeftHand OS version lower than 11.5. However, the performance improvements and newer functionality will not be available.

The over subscription feature adds two additional parameters that be set at the global level to apply to all drivers or under each driver instance in Liberty. The first new parameters, `max_over_subscription_ratio`, the ratio of over subscription when thin provisioned volumes are involved. The default ratio is 20.0; this means that a provisioned capacity can be 20 times of the total physical capacity. The second is `reserved_percentage`; this option represents the percentage of reserved backend capacity. The default percentage is 0. The capacity filter scheduler will take these values into account during its decision for a backend to host a particular volume request.

Block storage scheduler configuration with multi-backend

Multi-backend **must be** used with `filter_scheduler` enabled. Filter scheduler acts in two steps:

1. Filter scheduler filters the available backends. By default, AvailabilityZoneFilter, CapacityFilter, and CapabilitiesFilter are enabled.
2. Filter scheduler weighs the previously filtered backends. By default, CapacityWeigher is enabled. The CapacityWeigher attributes high scores to backends with the most available space.

According to the filtering and weighing, the scheduler will be able to pick “the best” backend to handle the request. In that way, filter scheduler achieves the goal of explicitly creating volumes on specific backends using volume types.

Block storage scheduler configuration with driver filter and weigher

The driver filter and weigher for the Block storage scheduler is a feature (new in Kilo) that, when enabled, allows for a filter and goodness function to be defined in your **cinder.conf** file. The two functions are used during volume creation time by the Block storage scheduler to determine which backend is the ideal for the volume. The filter function is used to filter out backend choices that should not be considered at all. The goodness function is used to rank the filtered backends from 0 to 100. This feature should be used when the default Block storage scheduling does not provide enough control for where volumes are being created.

Enable the usage of the driver filter for the scheduler by adding DriverFilter to the `scheduler_default_filters` property in your **cinder.conf** file. Enabling the driver weigher is similar. Add GoodnessWeigher to the `scheduler_default_weighers property` in your **cinder.conf** file. If you wish to include other OpenStack filters and weighers in your setup make sure to add those to the `scheduler_default_filters` and `scheduler_default_weighers` properties as well.

Note
You can choose to have only the `DriverFilter` or `GoodnessWeigher` enabled in your **cinder.conf** file depending on how much customization you want.

OpenStack supports various math operations that can be used in the filter and goodness functions. The currently supported list of math operations can be seen in table 1.

Table 1. Supported math operations for filter and goodness functions

OPERATIONS	TYPE
+, -, *, /, ^	standard math
not, and, or, &, |, !	logic
>, >=, <, <=, ==, <>, !=	equality
+, -	sign
x ? a : b	ternary
abs(x), max(x,y), min(x,y)	math helper functions

Several driver specific properties are available for use in the filter and goodness functions for an HPE StoreVirtual backend. The currently supported list of HPE StoreVirtual specific properties include:

- `capacity_utilization`—Percent of total space used on the HPE StoreVirtual cluster.
- `total_volumes`—The total number of volumes on the HPE StoreVirtual cluster.

Additional generic volume properties are available from OpenStack for use in the filter and goodness functions. These properties can be seen in the OpenStack Cloud Administrator Guide.

Note
Access the HPE LeftHand specific properties by using the following format in your filter or goodness functions: `capabilities.<property>`

The sample **cinder.conf** file in figure 5 shows an example of how several HPE StoreVirtualbackends could be configured to use the driver filter and weigher from the Block storage scheduler.

```
[default]
scheduler_default_filters = DriverFilter
scheduler_default_weighers = GoodnessWeigher
enabled_backends = lefthand-rest-1, lefthand-rest-2, lefthand-rest-3
[lefthand-rest-1]
hplefthand_api_url = <api_url>
hplefthand_username = <username>
hplefthand_password = <password>
hplefthand_clustername = cluster-1
volume_backend_name = lefthand
volume_driver =
cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver
filter_function = “capabilities.total_volumes< 10”
goodness_function = “(capabilities.capacity_utilization< 75)? 90 : 50”
```

```
[lefthand-rest-2]
hplefthand_api_url = <api_url>
hplefthand_username = <username>
hplefthand_password = <password>
hplefthand_clustername = cluster-2
volume_backend_name = lefthand
volume_driver =
cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver
filter_function = "capabilities.total_volumes< 10"
goodness_function = "(capabilities.capacity_utilization< 50)? 95 : 45"
[lefthand-rest-3]
hplefthand_api_url = <api_url>
hplefthand_username = <username>
hplefthand_password = <password>
hplefthand_clustername = cluster-3
volume_backend_name = lefthand
volume_driver =
cinder.volume.drivers.san.hp.hp_lefthand_iscsi.HPLeftHandISCSIDriver
filter_function = "capabilities.total_volumes< 20"
goodness_function = "(capabilities.capacity_utilization< 90)? 75 : 40"
```

Figure 5. Sample **cinder.conf** file showing driver filter and weigher usage

In figure 5 there are three HPE StoreVirtual backends enabled in the **cinder.conf** file. The sample shows how you can use HPE StoreVirtual specific properties to distribute volumes with more control than the default Block storage scheduler.

Note

Remember that you can combine the HPE StoreVirtual specific properties with the generic volume properties provided by OpenStack. Also the values used in the above sample are only for examples. In your own environment you have full control over the filter and goodness functions that you create. Refer to the OpenStack Cloud Administrator Guide for more details and examples.

.

Volume types assignment

Use the following command or the Admin Horizon UI (new in Juno) to specify a `volume_backend_name` for each volume type you create. This links the volume type to a backend name.

```
$ cinder --os-username admin --os-tenant-name admin type-key gold set
volume_backend_name=lefthand_gold
```

The second volume type could be for an iSCSI driver volume type named silver.

```
$ cinder --os-username admin --os-tenant-name admin type-key silver set
volume_backend_name=lefthand_silver
```

Multiple key value pairs can be specified when running the above command. For example, the following command is used to create a volume type named “gold”, with a data protection level of “r-6”, Network RAID-6 (Dual Parity), and with “full” provisioning.

```
$ cinder --os-username admin --os-tenant-name admin type-key gold set
volume_backend_name=lefthand_gold hplh:data_pl=r-6 hplh:provisioning=full
```

Volume manage and unmanage

Starting in the Kilo release, HPE StoreVirtual volumes can be managed and unmanaged. This allows for importing non-OpenStack volumes already on an HPE StoreVirtual Storage array into OpenStack/Cinder or exporting, which would remove them from the OpenStack/Cinder perspective. However, the volume on the HPE StoreVirtual Storage array would be left intact. Using the command line, you can see the available driver instances represented as “hosts” within the **cinder.conf** file. This host is where the HPE StoreVirtual volume that you would like to manage resides. Use the following command:

```
$ cinder-manage host list

mystack

mystack@lefthand

mystack@lefthand-1

mystack@lefthand-2
```

To manage volumes that exist on the HPE LeftHand but are not already managed by OpenStack/Cinder, use the command:

```
$ cinder manage --name <cinder name><host>#<pool><source-name>
```

Where <source-name> represents the name of the volume to manage and `<cinder name>` is optional but represents the OpenStack name and `<host>` represents the driver instance. The `<pool>` value is required. For the HPE StoreVirtual drivers this is just a repeat of the driver backend name. The manage volume command will also accept an optional `<--volume-type>` parameter that will perform a retype of the virtual volume after being managed. The following is an example of the manage command:

```
$ cinder manage --name vol-lhmystack@lefthand#lefthand volume12345
```

Note
Cinder manage will rename the volume on the HPE StoreVirtual Storage array to a name that starts with “volume-” followed by a UUID as this is required for OpenStack/Cinder to locate the volume under its management.

To unmanage a volume from OpenStack/Cinder and leave the volume intact on the HPE StoreVirtual Storage array, use the command:

```
$ cinder unmanage <volume_id>
```

Where `<volume_id>` is the ID of the OpenStack/Cinder volume to unmanage. The following is an example of the unmanage command:

```
$ cinder unmanage 647bd7d5-e8e1-4701-9038-7b4de14fda18
```

Note
Cinder `unmanage` will remove the OpenStack/Cinder volume from OpenStack but the volume will remain intact on the HPE StoreVirtual Storage array. The volume name will have “unm-” prefixed to it, followed by an encoded UUID. This is required because the HPE StoreVirtual has name length and character limitations.

Consistency groups

Prior to consistency groups, every operation in cinder happened at the volume level. Grouping like volumes allows for improved data protection, paves way to maintaining consistency of data across multiple different volumes, and allows for operations to be performed on groups of volumes. The fundamental supported operations include creating a consistency group, deleting a consistency group (and all volumes inside of it), adding volumes to a consistency group, removing volumes from a consistency group, and snapshotting a consistency group.

Note
The LeftHand Cinder driver does not currently support creating a consistency group from a source consistency group or a source consistency group snapshot.

Consistency group CLI support is defaulted to off. In order to access the consistency group related CLI commands, **/etc/cinder/policy.conf** needs to be modified by removing group: nobody from the following lines as such:

```
”consistencygroup:create” : ””,
”consistencygroup:delete”: ””,
”consistencygroup:update”: ””,
”consistencygroup:get”: ””,
”consistencygroup:get_all”: ””,
”consistencygroup:create_cgsnapshot” : ””,
”consistencygroup:delete_cgsnapshot”: ””,
”consistencygroup:get_cgsnapshot”: ””,
”consistencygroup:get_all_cgsnapshots”: ””,
```

Creating a consistency group

Once the **policy.conf** file is correctly modified, we can create a consistency group:

```
$cinder consisgroup-create [--name <name>] [--description <description>] <volume-type>
```

Where <volume-type> is the OpenStack/Cinder volume type name and --name and --description are optional:

```
$cinder consisgroup-create --name "MyCG" --description "lefthand cg" lefthand
```

To view the newly created consistency group:

```
$cinder consisgroup-list
```

```
| 831b2099-d5ba-4b92-a097-8c08f9a8404f | available | MyCG |
```

Deleting a consistency group

An empty consistency group can be deleted by issuing the following command:

```
$cinder consisgroup-delete <consisgroup-id>
```

Where <consisgroup-id> represents to consistency group ID:

```
$cinder consisgroup-delete 831b2099-d5ba-4b92-a097-8c08f9a8404f
```

If the group has volumes, we can add the force flag (NOTE: This will fully delete all volumes in the group):

```
$cinder consisgroup-delete <consisgroup-id> --force
```

Where <consisgroup-id> represents to consistency group ID:

```
$cinder consisgroup-delete 831b2099-d5ba-4b92-a097-8c08f9a8404f --force
```

Adding volumes

In order to add volumes to the group:

```
$cinder consisgroup-update <consisgroup-id> --add-volumes <uuid1,uuid2,......>
```

Where <consisgroup-id> represents the consistency group ID and <uuid1, uuid2,.....> is a comma separated list of OpenStack/Cinder volume IDs:

```
$cinder consisgroup-update 831b2099-d5ba-4b92-a097-8c08f9a8404f --add-volumes 87ac88d4-e360-4bdb-b888-
2208fbe282dd,9ce09c09-0a20-4bcd-bd1a-0eaa95dbe0cd,a4563466-61d4-4018-b586-f07f84c4010c
```

Removing volumes

To delete volumes from a consistency group:

```
$cinder consisgroup-update <consisgroup-id> --remove-volumes <uuid1,uuid2,......>
```

Where <consisgroup-id> represents the consistency group ID and <uuid1, uuid2,.....> is a comma separated list of OpenStack/Cinder volume IDs:

```
$cinder consisgroup-update 831b2099-d5ba-4b92-a097-8c08f9a8404f --remove-volumes 87ac88d4-e360-4bdb-
b888-2208fbe282dd
```

Note

This does not delete the volume, it only removes it from a group.

Creating a cgsnapshot

Snapshotting a consistency group can be accomplished with:

```
$cinder cgsnapshot-create [--name <name>] [--description <description>] <consisgroup-id>
```

Where <consisgroup-id> represents the consistency group ID and --name and --description are optional:

```
$cinder cgsnapshot-create --name "MyCGSnap" --description "Snapshot of MyCg" 831b2099-d5ba-4b92-a097-8c08f9a8404f
```

To view the newly created cgsnapshot:

```
$cinder cgsnapshot-list

| 70c266bb-8255-4f2b-83cf-87f79d54dfb4 | creating | MyCGSnap |
```

Deleting a cgsnapshot

To delete a cgsnapshot:

```
$cinder cgsnapshot-delete <cgsnapshot-id>
```

Where <cgsnapshot-id> is the consistency group snapshot ID:

```
$cinder cgsnapshot-delete 70c266bb-8255-4f2b-83cf-87f79d54dfb4
```

Multiple backend requirements

The `hplefthand_clustername` is required and is case-sensitive.

Backend assisted volume migration

Backend assisted volume migration is another new capability when the driver is configured to execute in standard mode. When a volume is migrated using backend assisted volume migration, both source and destination clusters must be in the same HPE StoreVirtual management group. The HPE StoreVirtual Storage array uses native LeftHand APIs to migrate the volume. The volume cannot be attached or have snapshots to migrate.

Volume retype

Volume retype was available in the Icehouse release. The retype will only work if the volume is on the same HPE StoreVirtual Storage array. This allows the volume retype say from a "silver" volume type to a "gold" volume type. The HPE StoreVirtual OpenStack drivers will modify the volume's data protection level, provisioning type, and adaptive optimization settings, as needed, to make the volume behave appropriately for the new volume type. Use caution when using the optional "—migration-policy on-demand" as this falls back to copying the entire volume (using dd over the network) to the cinder node and then to the destination HPE StoreVirtual storage array. The cinder node also must have enough space available to store the entire volumes during the migration. We recommend that you use the default "—migration-policy never" when retype is used.

Note

The `volume_backend_name` in **cinder.conf** must be the same between the source and destination volume types when "–migration-policy" is set to "never" the default and recommend retype method.

Support for Containerized Stateful Services

The HPE LeftHand drivers are supported by the ClusterHQFlocker open source technology. Details on the setup and usage of Flocker can be found at docs.clusterhq.com/en/latest/.

Summary

HPE is a Platinum member of The OpenStack Foundation. HPE has integrated OpenStack open source cloud platform technology into its enterprise solutions to enable customers and partners to build enterprise-grade private, public, and hybrid clouds.

The Liberty release continues HPE's contributions to the Cinder project, enhancing core Cinder capabilities; consistency groups, and over subscription of thin provisioned volumes in the HPE StoreVirtual Block Storage Driver. The focus continues to be on adding enterprise functionality, such as volume manage/unmanage and enhanced Block Storage scheduling based on filtering and goodness functions from the Driver. The HPE StoreVirtual Block Storage Drivers support the OpenStack technology across the iSCSI protocol.

Resources

HPE Cloud
HPE Helion

HPE HelionOpenStack Community

HPE Helion Hybrid Cloud

HPE Helion Cloud News

OpenStack
OpenStack Website

OpenStack Documentation

OpenStack Cloud Administrator Guide

HPE LeftHand StoreVirtual Array
HPE StoreVirtual Storage Family

HPE StoreVirtual iSCSI Driver

To help us improve our documents, provide feedback at hp.com/solutions/feedback.

Learn more at

hp.com/go/OpenStack

Sign up for updates

★ Rate this document

4AA5-1966ENW, January 2016, Rev. 5

Hewlett Packard Enterprise