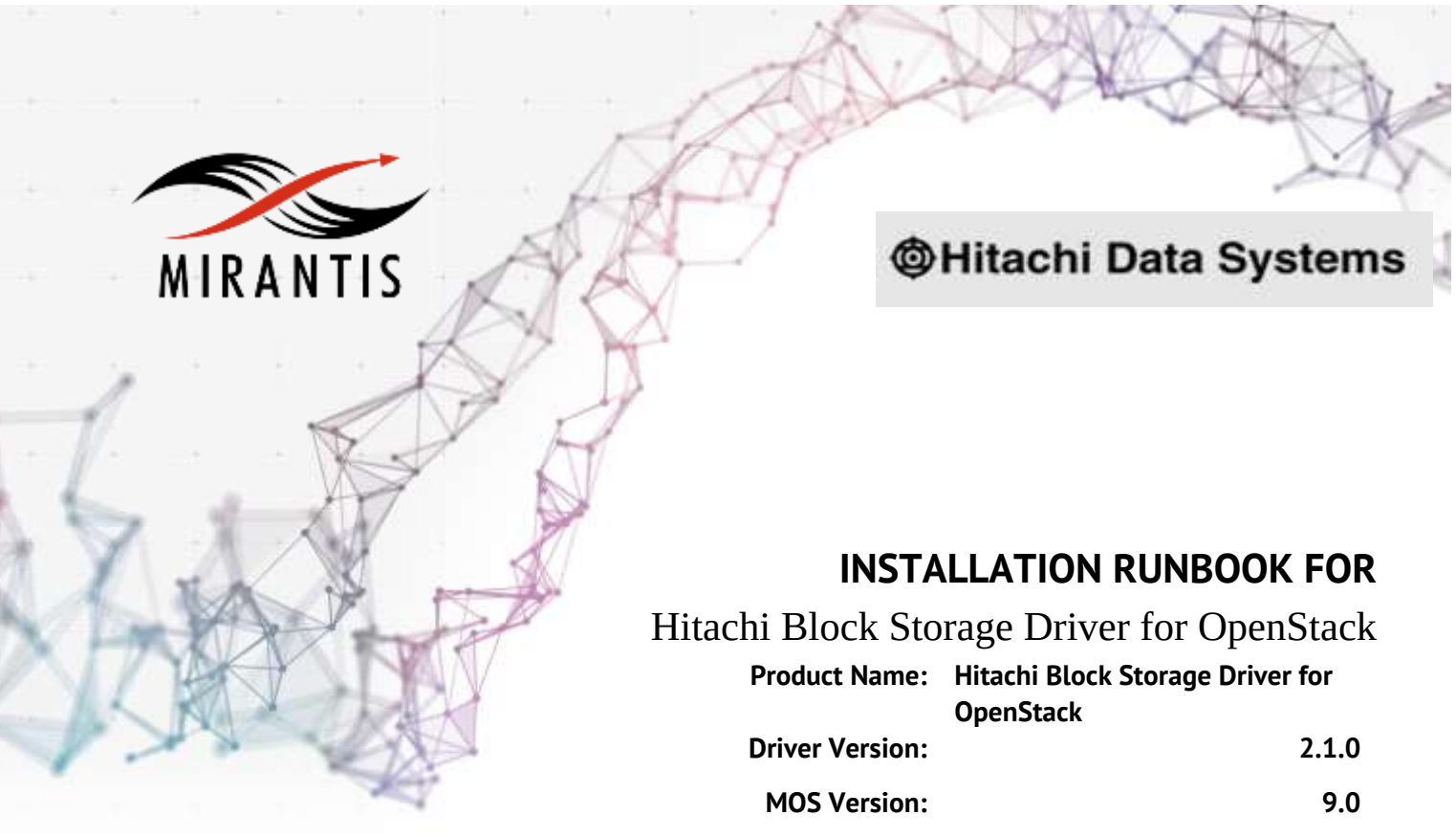


INSTALLATION RUNBOOK FOR

Hitachi Block Storage Driver for OpenStack

Product Name:	**Hitachi Block Storage Driver for OpenStack**
Driver Version:	**2.1.0**
MOS Version:	**9.0**
OpenStack Version:	**Mitaka**
Product Type:	**Storage Driver for Cinder**

Document History

Version	Revision Date	Description
1.0	12-08-2016	Initial Version

1. Introduction

This document serves as a runbook for deploying the Hitachi Block Storage Driver for OpenStack within Mirantis OpenStack deployment. Integrating Hitachi Block Storage Driver for OpenStack into Mirantis deployment allows high-performance and high-reliability features for Hitachi storage managed by Cinder.

The objective of Mirantis OpenStack certification is to provide Mirantis program partners with a Consistent and unified approach for acceptance of their solution into the Mirantis Technology Partner Program.

Certification is designed within the context of Mirantis OpenStack infrastructure, including **Mirantis Fuel deployment tool and supported cloud reference architectures.**

1.1 Target Audience

OpenStack administrators, Storage administrators, Network administrators who are familiar with Mirantis OpenStack, Fuel and Hitachi Block Storage Driver for OpenStack.

2. Product Overview

Hitachi Block Storage Driver for OpenStack (abbreviated as HBSD hereafter) is a driver for Cinder, which is a block storage management component, in OpenStack environments. HBSD allows you to use high-performance and high-reliability features for Hitachi storage managed by Cinder. Both Mirantis OpenStack and HBSD can be configured to provide services in a variety of ways. To ensure that the best possible end result is achieved, the guidelines and best practices for Mirantis OpenStack should be followed to configure OpenStack.

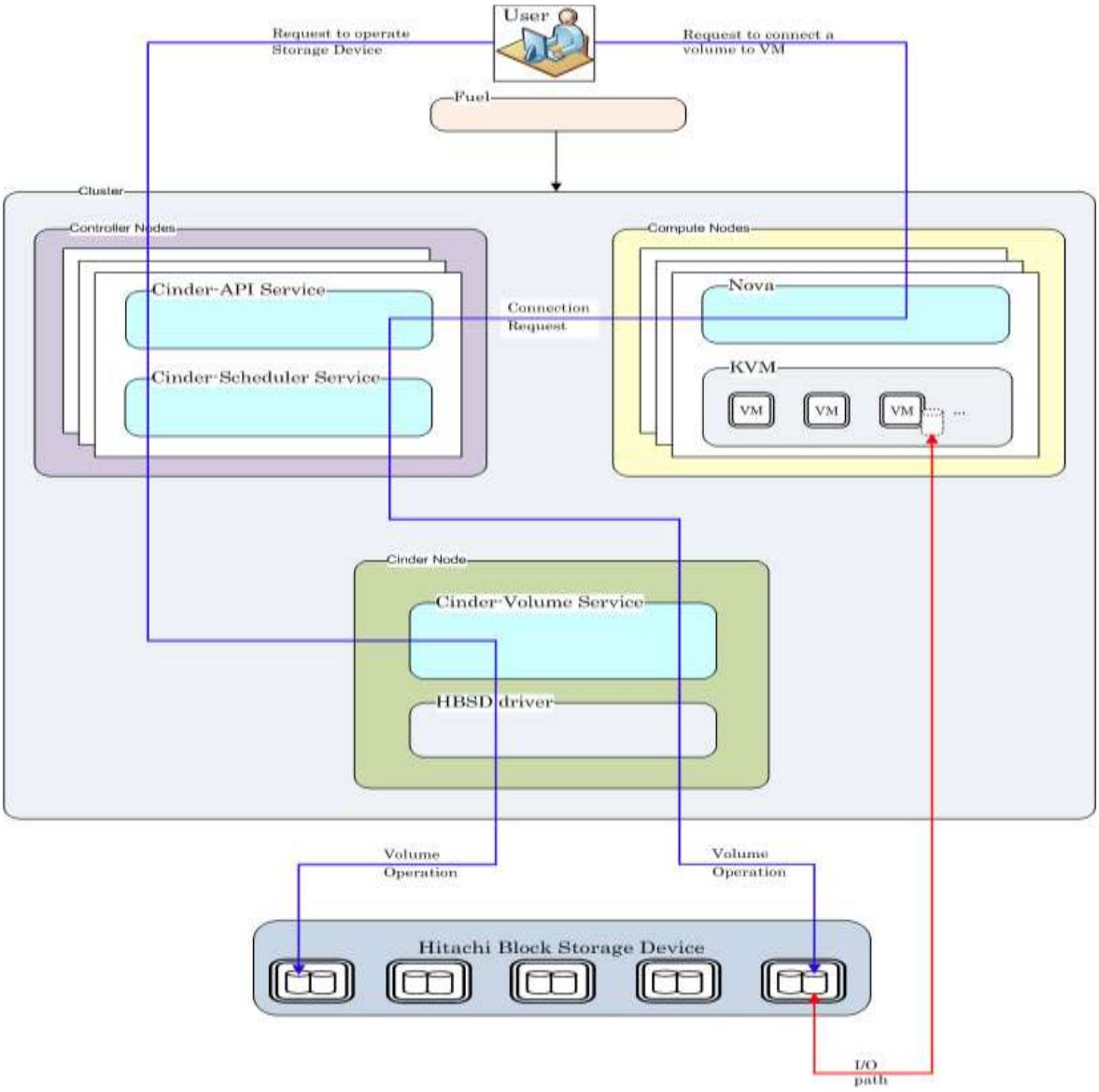
For HBSD best practices it is suggested that the administrator follow the guidelines outlined in the HBSD User Manual to configure a compute and controller node.

3. Joint Reference Architecture

Overview:

This reference architecture describes how to integrate Mirantis OpenStack 9.0 (using OpenStack Mitaka) with HBSD 2.1.0, utilizing HBSD as backend storage.

- Hitachi Block Storage Device - To use high-performance and high-reliability features.
- Controller nodes - Servers running OpenStack controller elements.
- Compute nodes - Servers running OpenStack compute elements.
- Cinder node - Server running OpenStack cinder elements.
- Fuel - Infrastructure running OpenStack deployment and management tool.

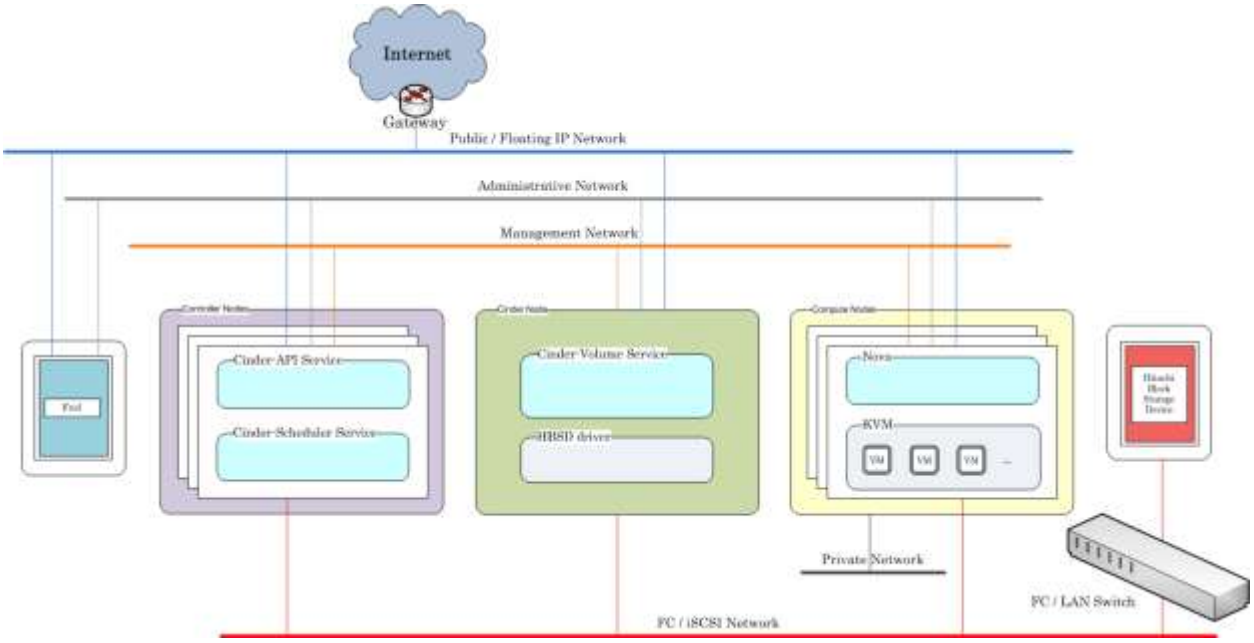


Node configuration:

When HBSD is used in an environment that is managed by Mirantis OpenStack 9.0, deployment of all nodes and OS configuration must be performed through Mirantis Fuel. HBSD supports deployment using Ubuntu 14.04 LTS when using OpenStack Mitaka based releases.
Note: This document assumes Ubuntu is being used when referencing command-line utilities and/or OS level configuration files and tools.

High Availability configurations for cinder-volume service:
HBSD supports both HA [Active / Standby] mode and 'non-HA' mode.

4. Physical and Logical Network Topology



Fuel operates with a set of logical networks. In this scheme, these logical networks are mapped With such example as follows:

- Administrative (Fuel) network: untagged on this scheme.
- Public Network: network: untagged on this scheme
- Floating Network: network: untagged on this scheme
- Management Network: VLAN 101.
- FC/iSCSI Network [Physical LAN or FC] is created manually to connect the control and compute nodes with Hitachi Storage.

  Note: Fuel uses a separate network to connect Hitachi Storage directly.
- Private Network: VLANs 200-210

5. Installation and Configuration

Overview:
When HBSD is used as a backend storage solution for OpenStack, the guidelines and best practices published for Mirantis apply.
The deployment of Mirantis OpenStack should be done through FUEL, and the deployment should pass all automated health checks.

After installing an OpenStack environment using Mirantis Fuel a number of configuration changes are required to use HBSD as backend storage for Cinder.

Prerequisites:
This guide assumes that the following base requirements are satisfied:

- HBSD 2.1.0 is installed and configured on supported hardware.
- Mirantis OpenStack 9.0 is used and Mirantis FUEL is used to deploy/manage servers.
- Technically, this document is specific to Mirantis OpenStack 9.0 and Mitaka.
- The environment is running on Ubuntu 14.04 LTS.

5.1 Environment Preparation

Please follow the Mirantis OpenStack deployment guide for getting the Fuel master node up and the controller, compute nodes discovered.

Details available in the following links:
http://docs.openstack.org/developer/fuel-docs/userdocs/fuel-install-guide.html
https://docs.mirantis.com/openstack/fuel/fuel-9.0/pdf/Mirantis-OpenStack-9.0-QuickStartGuide.pdf

After completing Fuel setup, the Fuel UI screen shows all your Slave nodes as "Unallocated nodes". You can now create, configure, and deploy your first OpenStack environment. One Fuel Master can deploy and manage multiple OpenStack environments but you must create each environment separately.
During the certification and functional verification of HBSD and Mirantis OpenStack 9.0 the following configuration was used:

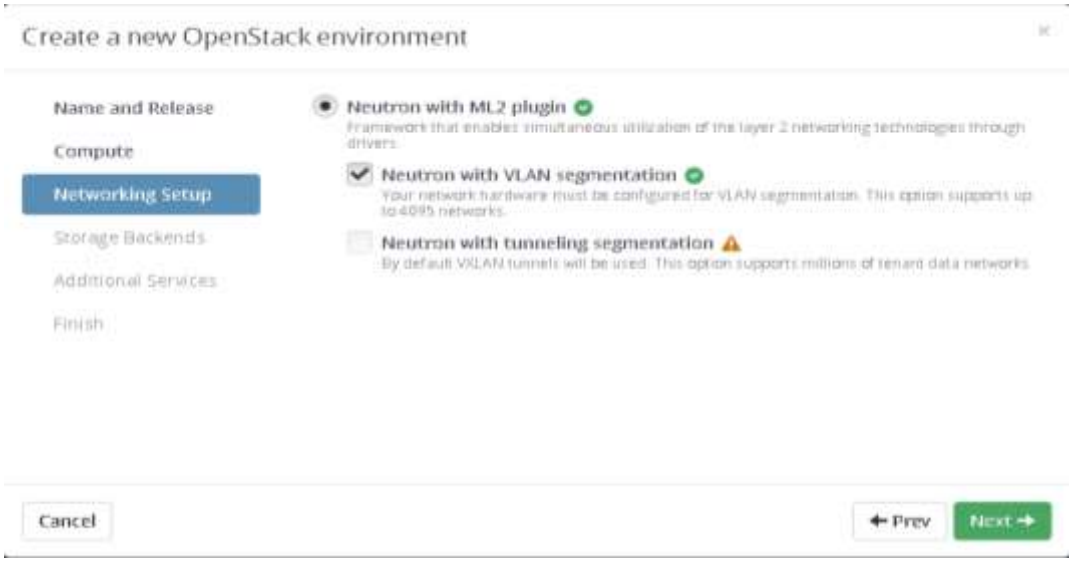
- One Mirantis Fuel Master Node.
- One server used for Controller node and Cinder node
- Two OpenStack Compute nodes.

Creation of OpenStack environment:

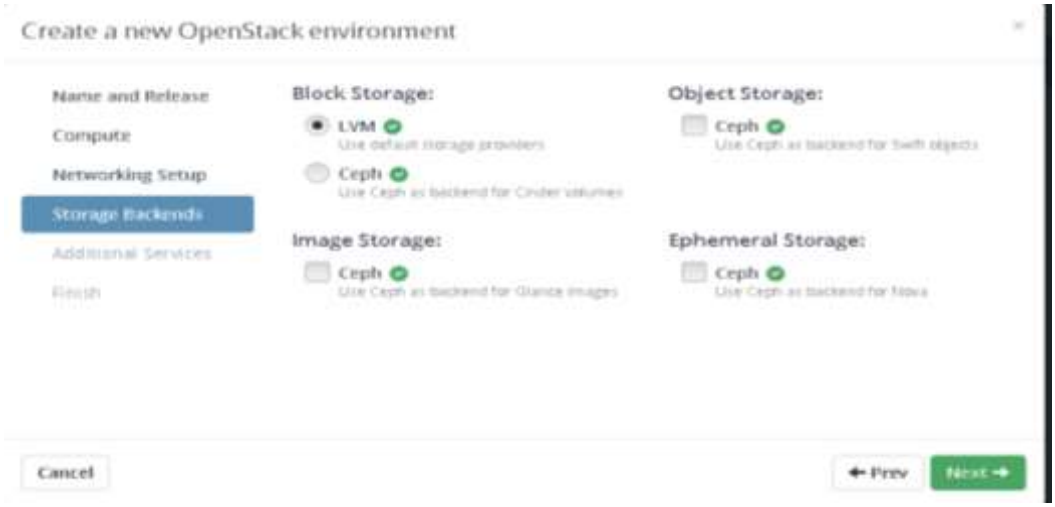
- Launch Wizard to Create New Environment.
- Click on the "New OpenStack environment" icon to launch the wizard that creates a new OpenStack environment.
- Give the environment a name and select the Linux distribution from the drop-down list As Mitaka on Ubuntu 14.04
- The operating system Ubuntu 14.04 will be installed on the target nodes in the environment.
  - On the Fuel UI, click on “New OpenStack Environment”.
  - When the wizard opens, enter the name and the desired OpenStack Release.

- Select the Compute for the Environment

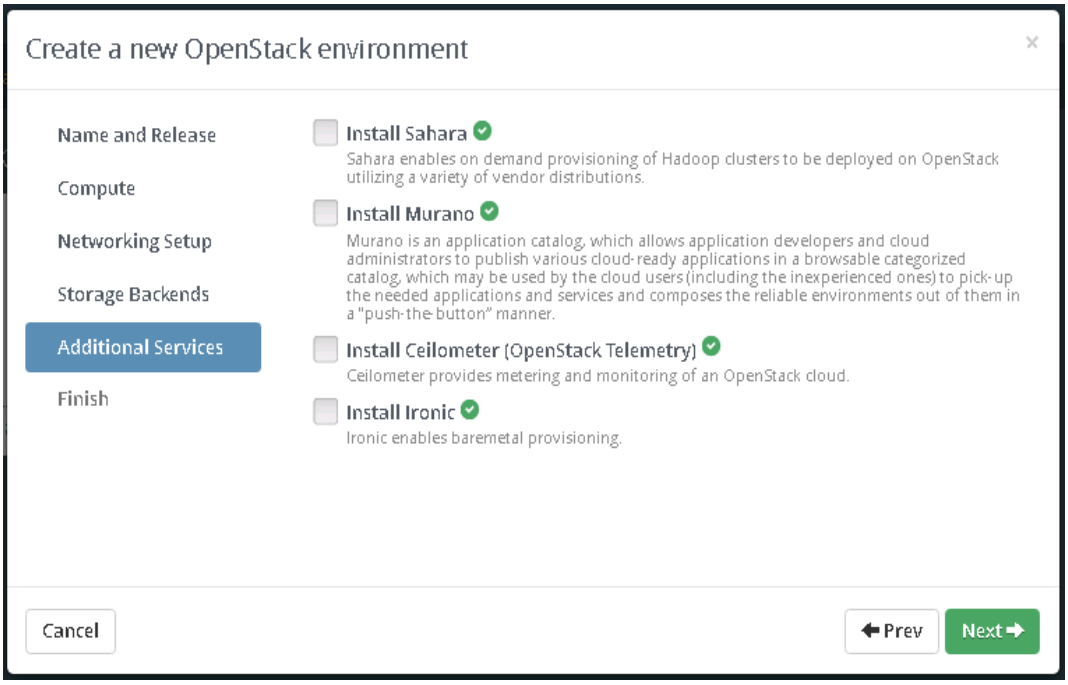
- Select the network setup option ‘Neutron with VLAN segmentation’.



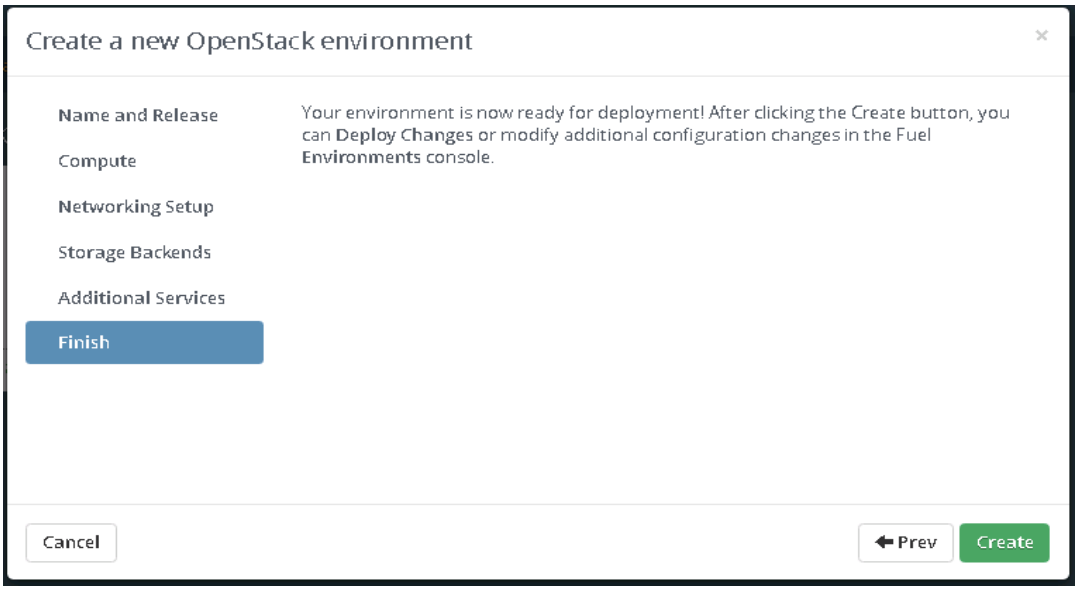
- Under Storage Backend, select the Option “LVM” of Block Storage. Hitachi Volume driver can be installed after the OpenStack is deployed.



- Select the additional services and click on next.



- Click Create to start deploy the OpenStack.



5.2 MOS Installation

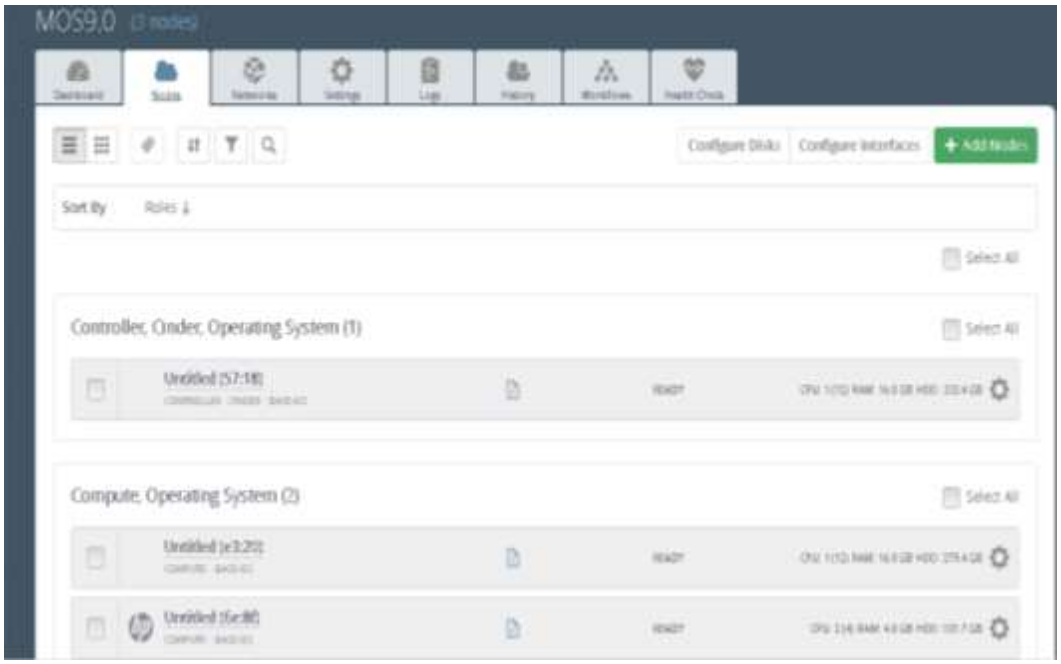
The MOS deployment will consist of,

- One Fuel server.
- At least one MOS controller (preferred 3 MOS controllers in HA configuration).
- Neutron VLAN based configuration is recommended.
- Storage backend as default providers [Cinder LVM over iSCSI for volumes] is mandatory.

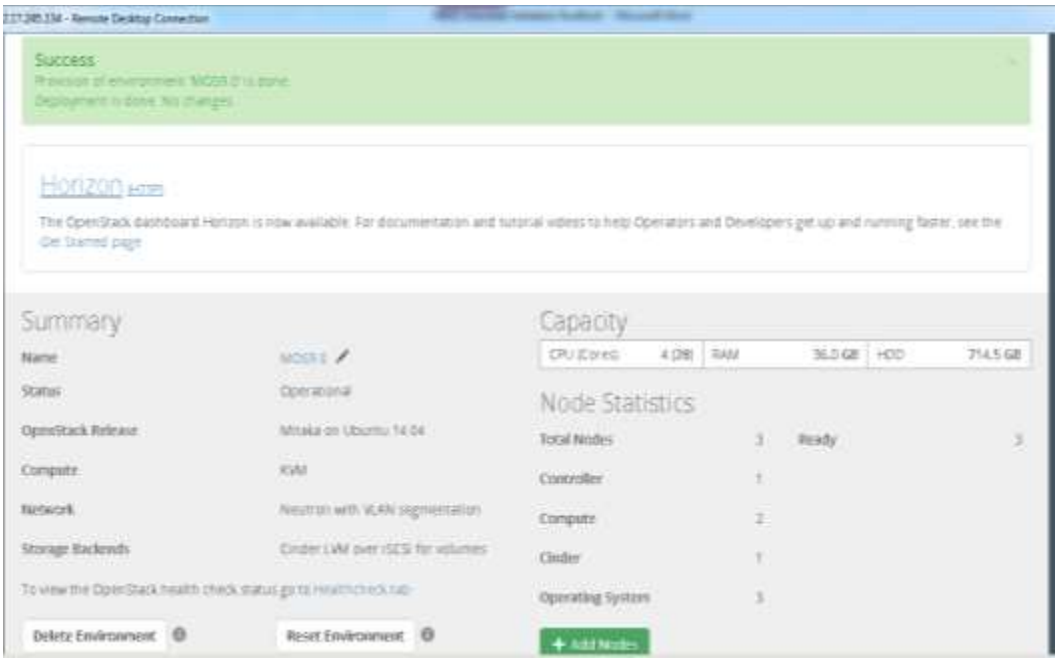
Please follow Mirantis documentation on bringing up a fuel node and discovering nodes on which OpenStack controller/ compute services shall run.

- Add nodes to the environment.
- Assign a role or roles to each node server.
- Do the required Network settings.
- Mapping logical networks to physical interfaces on servers [if required].
- Verify Networks

• The network verification check should get succeeded in order to ensure that deployment is not failed due to network settings.
• Deploy Changes.



- Status after Deployment



5.2.1 Health Check Results

Validating the installation:

• After the configuration has been completed, it should be validated using the automated health check capabilities of Mirantis Fuel.

• Doing this will catch most errors before trying to deploy production workloads.

• All Cinder related tests should pass with no errors.

• The Health Check is initiated from the Mirantis Fuel console (within the context of the relevant OpenStack cloud).

• All of the Sanity Tests should pass, and it is important that the “Create Volume” related Functional Tests also pass.

• If any of these basic tests fail, the cause should be determined and corrected before proceeding to deploy a workload on these systems.

In Health Check - Functional test, we have skipped the step “Check network connectivity from instance via floating IP ".

The reason for skipping this step in Health Check - Functional test are mentioned below,

• Target component: Neutron - This component testing is not required cinder certification.

• Scenario used to “Check network connectivity from instance via floating IP” includes,

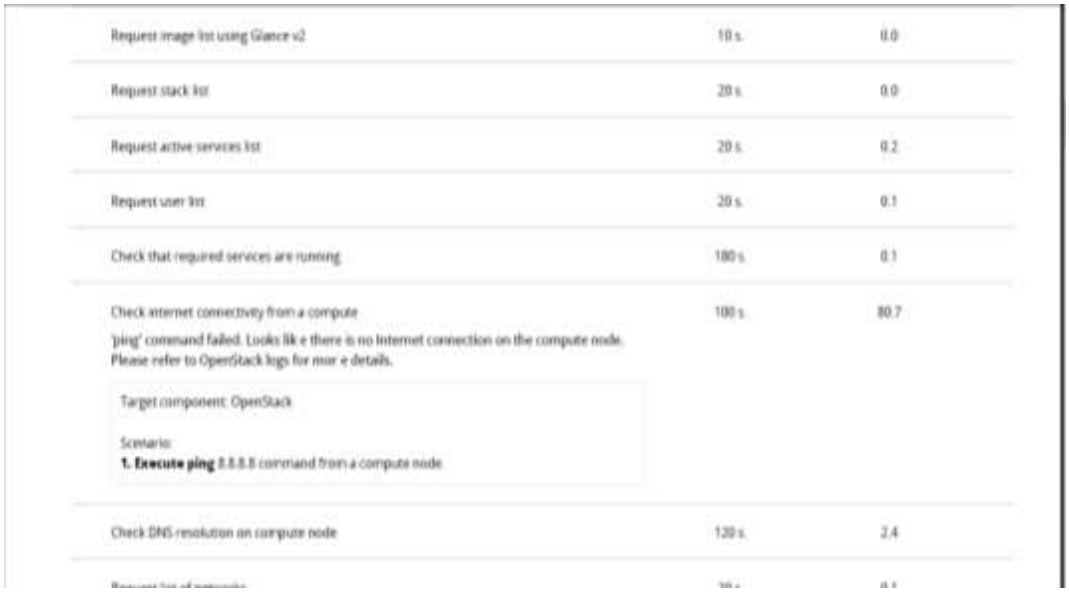
1. Create a new security group (if it doesn`t exist yet).
2. Create router, Create network and Create subnet.

3. Uplink subnet to router.
4. Create an instance using the new security group with created subnet.
5. Create a new floating IP.
6. Assign the new floating IP to the instance.
7. Check connectivity to the floating IP using ping command.
8. Check that public IP 8.8.8.8 can be pinged from instance.
9. Disassociate server floating IP.
10. Delete floating IP.
11. Delete server.
12. Remove router, Remove subnet and Remove network.

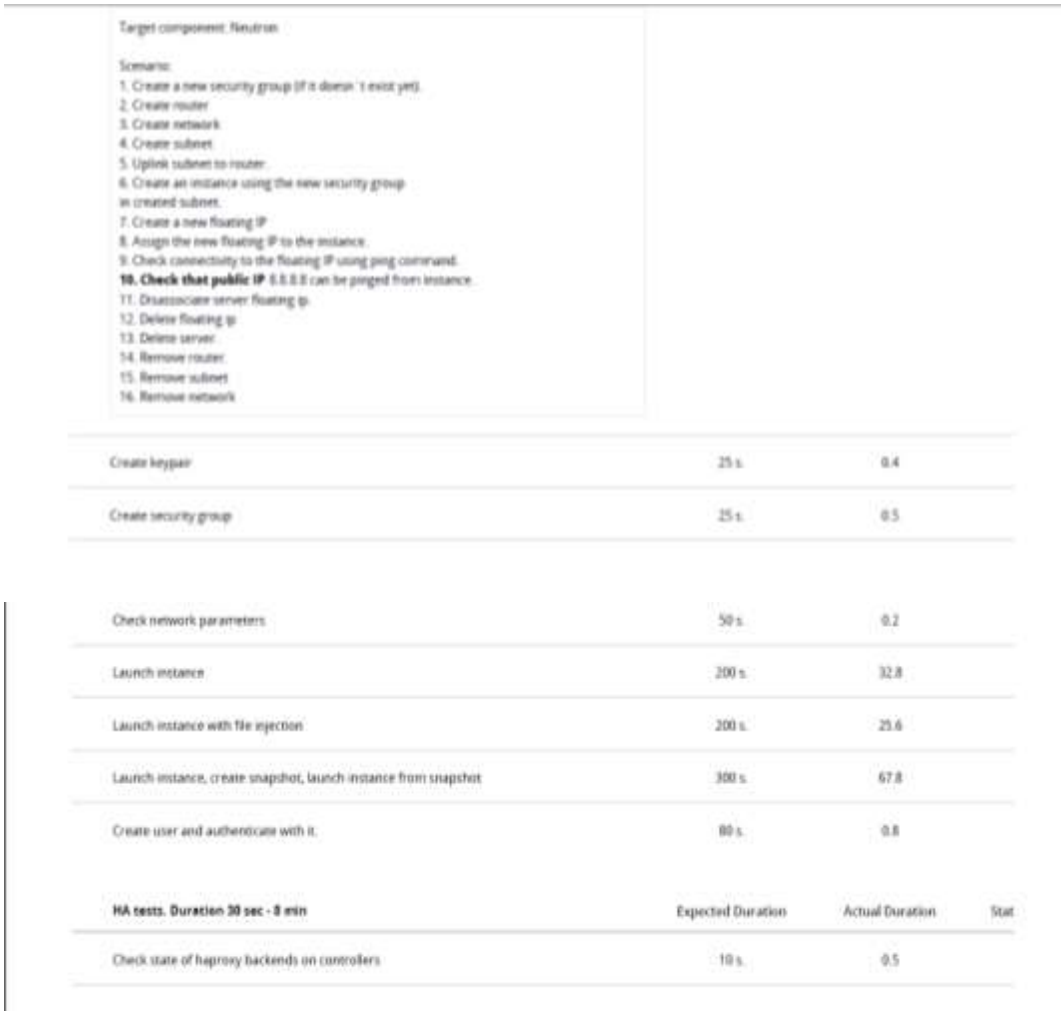
In the above mentioned scenario, step #8 will check whether public IP 8.8.8.8 can be pinged or not. As the environment built for this certification does not contain 8.8.8.8 in DNS list [Available in "Mirantis OpenStack Environment - Settings tab - Host OS DNS Servers"], the pinging will not happen. Hence this step has been skipped.

Note: Instead we have used a proxy server IP to establish connectivity between instance and public connectivity. This is non-HA setup, so we have skipped the HA tests.


OpenStack Health Check
Select All
Provide credentials
Run Te
Sanity tests. Duration 30 sec - 2 min
Expected Duration
Actual Duration
Stat
Request flavor list
20 s.
0.2
Request image list using Nova
20 s.
0.2
Request instance list
20 s.
0.1
Request absolute limits list
20 s.
0.0
Request snapshot list
20 s.
0.2
Request volume list
20 s.
0.3
Request image list using Glance v1
10 s.
0.0


Request image list using Glance v2
10 s.
0.0
Request stack list
20 s.
0.0
Request active services list
20 s.
0.2
Request user list
20 s.
0.1
Check that required services are running
180 s.
0.1
Check internet connectivity from a compute
100 s.
80.7
'ping' command failed. Looks lik e there is no Internet connection on the compute node.
Please refer to OpenStack logs for mor e details.
Target component: OpenStack
Scenario:
1. Execute ping 8.8.8.8 command from a compute node
Check DNS resolution on compute node
120 s.
2.4

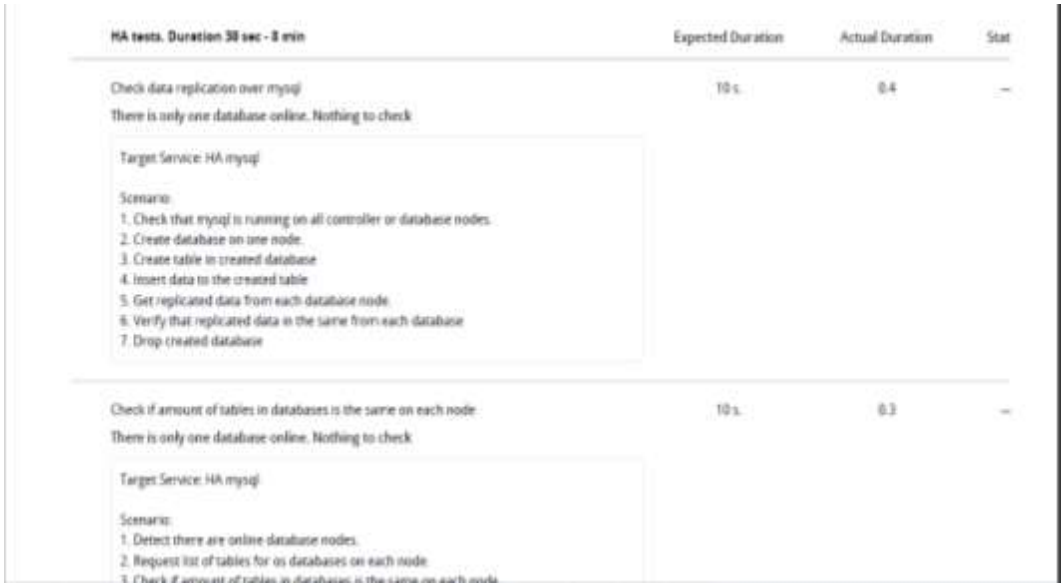
Request list of networks	20 s.	0.1	
Functional tests. Duration 3 min - 14 min	Expected Duration	Actual Duration	Stat
https://172.17.14.67:8443/#cluster/2/healthcheck			1/4
11/24/2016 Fuel Dashboard - MOS9.0			
Create instance flavor	30 s.	0.4	
Check create, update and delete image actions using Glance v2	70 s.	1.9	
Create volume and boot instance from it	350 s.	57.6	
Create volume and attach it to instance	350 s.	77.4	
Check network connectivity from instance via floating IP	300 s.	616.1	
Time limit exceeded while waiting for public connectivity checking fr om VM to finish. Please refer to OpenStack logs for mor e details.			



Target component: Neutron

Scenario:
1. Create a new security group (if it doesn`t exist yet).
2. Create router
3. Create network
4. Create subnet
5. Uplink subnet to router
6. Create an instance using the new security group
in created subnet.
7. Create a new floating IP
8. Assign the new floating IP to the instance.
9. Check connectivity to the floating IP using ping command.
10. Check that public IP 8.8.8.8 can be pinged from instance.
11. Disassociate server floating ip.
12. Delete floating ip
13. Delete server
14. Remove router
15. Remove subnet
16. Remove network

Create keypair	25 s.	0.4	
Create security group	25 s.	0.5	
Check network parameters	50 s.	0.2	
Launch instance	200 s.	32.8	
Launch instance with file injection	200 s.	25.6	
Launch instance, create snapshot, launch instance from snapshot	300 s.	67.8	
Create user and authenticate with it.	80 s.	0.8	
HA tests. Duration 30 sec - 8 min	Expected Duration	Actual Duration	Stat
Check state of haproxy backends on controllers	10 s.	0.5	



HA tests. Duration 30 sec - 8 min	Expected Duration	Actual Duration	Stat
Check data replication over mysql There is only one database online. Nothing to check Target Service: HA mysql Scenario: 1. Check that mysql is running on all controller or database nodes. 2. Create database on one node. 3. Create table in created database 4. Insert data to the created table 5. Get replicated data from each database node. 6. Verify that replicated data in the same from each database 7. Drop created database	10 s.	0.4	–
Check if amount of tables in databases is the same on each node There is only one database online. Nothing to check Target Service: HA mysql Scenario: 1. Detect there are online database nodes. 2. Request list of tables for os databases on each node 3. Check if amount of tables in databases is the same on each node	10 s.	0.3	–

Check galera environment state There is only one database online. Nothing to check Target Service: HA mysql Scenario: 1. Detect there are online database nodes. 2. Ssh on each node containing database and request state of galera node 3. For each node check cluster size 4. For each node check status is ready 5. For each node check that node is connected to cluster	10 s.	0.3	–
Check pacemaker status	10 s.	0.7	
RabbitMQ availability There is only one RabbitMQ node online. Nothing to check Scenario: 1. Retrieve cluster status for each controller. 2. Check that numbers of rabbit nodes is the same in Hiera DB and in actual cluster	100 s.	0.7	–

3. Check crm status for rabbit
4. List channels

RabbitMQ replication	100 s.	0.6	–

There is only one RabbitMQ node online. Nothing to check

Scenario:
1. Check rabbitmq connections.
2. Create queue.
3. Publish test message in created queue.
4. Request created queue and message.
5. Delete queue.

Platform services functional tests. Duration 3 min - 60 min	Expected Duration	Actual Duration	Stat
Typical stack actions: create, delete, show details, etc.	720 s.	38.3	

Advanced stack actions: suspend, resume and check	900 s.	79.3	
Check stack rollback	470 s.	96.5	
Update stack actions: inplace, replace and update whole template	1300 s.	104.6	
Check creation of stack with Wait Condition/Handle resources	820 s.	78.8	

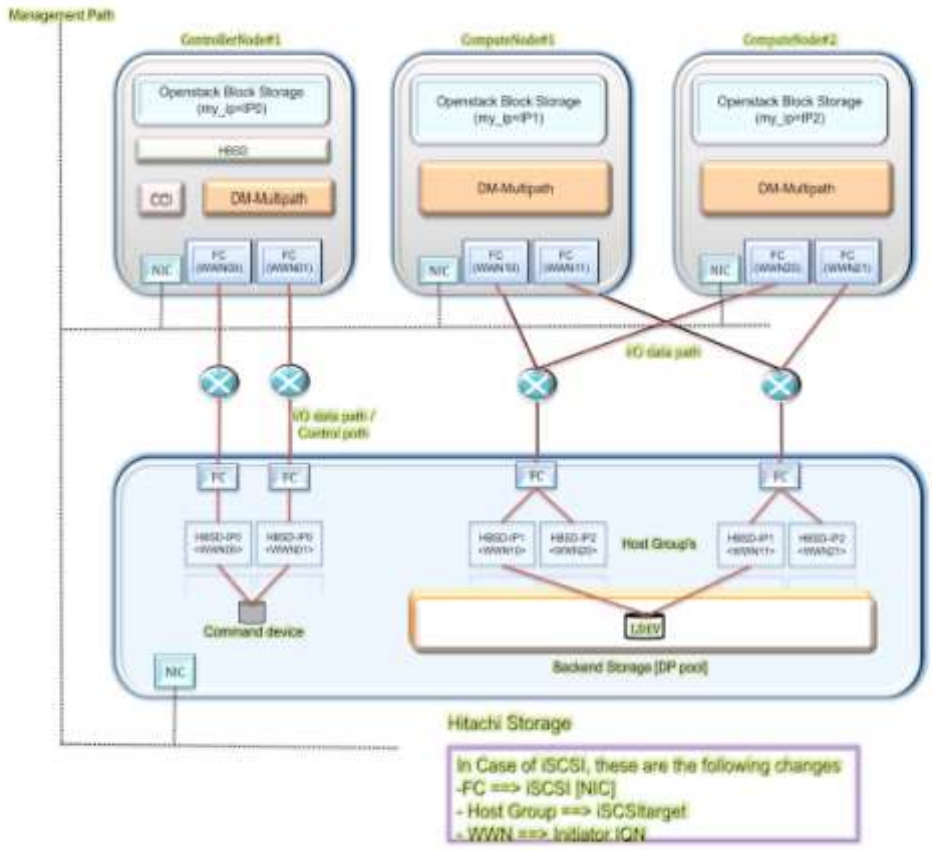
Cloud validation tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Stat
Check disk space outage on controller and compute nodes	20 s.	1.0	
Check log rotation configuration on all nodes	20 s.	0.9	

Configuration tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Stat
Check usage of default credentials on master node	20 s.	0.2	

Default credentials for ssh on master node were not changed. Please refer to OpenStack logs for more details.

5.3 HBSD Installation Procedure

The information described in this section about storage resource setting, installation and configuration of storage management software is all belong to Hitachi Storage Administrators. And, they will be responsible for doing the necessary configuration in order to use Hitachi storage as mentioned below.



Note: This is an example connection configuration for VSP G1000/VSP G200, G400, G600, G800/VSP/HUS VM with FC and the same can be used in case of VSP G200, G400, G600, and G800 with iSCSI also.
For more detailed information on Storage resource setting, installation and configuration of management software, kindly refer the support documents from https://support.hds.com/en_us/documents.html

1. Setting contents for each node:

Table mentioned below shows the setting contents for each node.

Node type	Items	Contents
Cinder node	my_ip for cinder service (/etc/cinder/cinder.conf)	Specify IPv4 address for management LAN of the node. The IPv4 address must be a unique value among other nodes. (less than 15 characters)
	Initiator IQN (/etc/iscsi/initiatorname.iscsi)	Specify Initiator IQN which must be a unique value among other nodes.
Compute node	my_ip for nova compute service (/etc/nova/nova.conf)	Specify IPv4 address for management LAN of the node. The IPv4 address must be a unique value among other nodes. (less than 15 characters)
	Initiator IQN (/etc/iscsi/initiatorname.iscsi)	Specify Initiator IQN which must be a unique value among other nodes.

How to identify "my_ip" and where to include:

"my_ip" is the IPv4 address for management LAN of the specific Node. The IPv4 address must be a unique value among other nodes. During configuration, /etc/nova/nova.conf file must be populated with the value of "my_ip"

Execute the command "ifconfig -a" to find out "my_ip" of the each specific node.

Here is an example:

```
root@node-6:~# ifconfig -a
br-ex     Link encap:Ethernet  HWaddr 00:15:60:53:6e:8e
          inet addr:172.17.26.34  Bcast:172.17.27.255  Mask:255.255.254.0
          inet6 addr: fe80::215:60ff:fe53:6e8e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12502269 errors:0 dropped:600991 overruns:0 frame:0
          TX packets:23037 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1181838159 (1.1 GB)  TX bytes:1091645 (1.0 MB)
```

The /etc/nova/nova.conf file of Compute node should be populated with the IPv4 address for management LAN of Compute node as below.

```
# cat /etc/nova/nova.conf
# Enables or disables logging values of all registered options when starting a
# service (at DEBUG level). (boolean value)
#log_options = true

# Specify a timeout after which a gracefully shutdown server will exit. Zero
# value means endless wait. (integer value)
#graceful_shutdown_timeout = 60
```

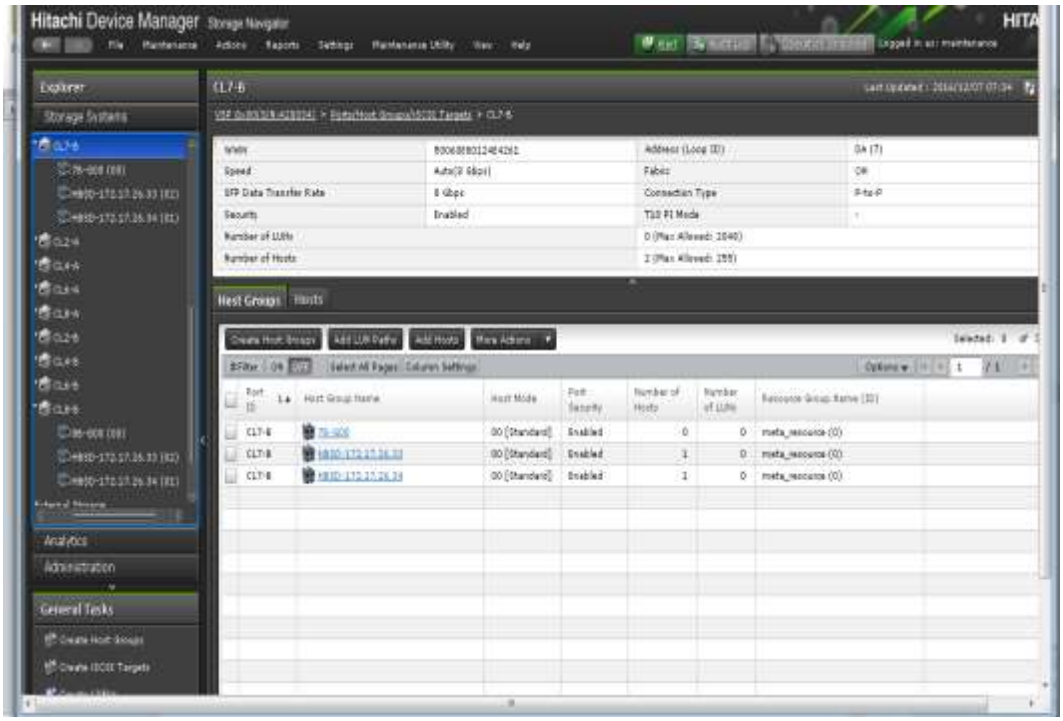
force_raw_images=True
notify_api_faults=False
resume_guests_state_on_host_boot=False
block_device_allocate_retries_interval=3
compute_manager=nova.compute.manager.ComputeManager
network_device_mtu=65000
state_path=/var/lib/nova
report_interval=60
remove_unused_original_minimum_age_seconds=86400
image_service=nova.image.glance.GlanceImageService
use_cow_images=True
heal_instance_info_cache_interval=60
notify_on_state_change=vm_and_task_state
instance_usage_audit=True
block_device_allocate_retries=300
reserved_host_memory_mb=512
config_drive_format=vfat
service_down_time=180
use_syslog_rfc_format=True
notification_topics=notifications
instance_usage_audit_period=hour
auth_strategy=keystone
compute_driver=libvirt.LibvirtDriver
rootwrap_config=/etc/nova/rootwrap.conf
force_config_drive=True
allow_resize_to_same_host=True
connection_type=libvirt
use_neutron=True
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver
security_group_api=neutron
force_snat_range=0.0.0.0/0
linuxnet_ovs_integration_bridge=br-int
#my_ip=192.168.0.13
my_ip=172.17.26.34 ----------→ “my_ip”
firewall_driver=nova.virt.firewall.NoopFirewallDriver
vif_plugging_is_fatal=True

Configuring the storage device with “my_ip”.

The “my_ip” value is used to configure Hitachi storage devices for Hitachi Block Storage Driver. During Hitachi storage provisioning, users requires to create a host group.

The host group must to be named as “HBSD-<my_ip>”. For example: “HBSD-172.17.26.34”. my_ip must be the same value as the setting for the service (cinder or nova compute) in each node.

Here is the screenshot of Hitachi Device Manger for Storage Hitachi VSP200.



In above screenshot, the CL7-B is the fiber Channel port of the Hitachi Storage. It is connected with both the Compute node through fiber channel switch.
"HBSD-172.17.26.33" and "HBSD-172.17.26.34" are the Host groups of both Compute Nodes having IP address "172.17.26.33" and "172.17.26.34" respectively.

2. Resource setting of the storage:

For the target storage device, set the resources to allow HBSD to use each FC or iSCSI connection.

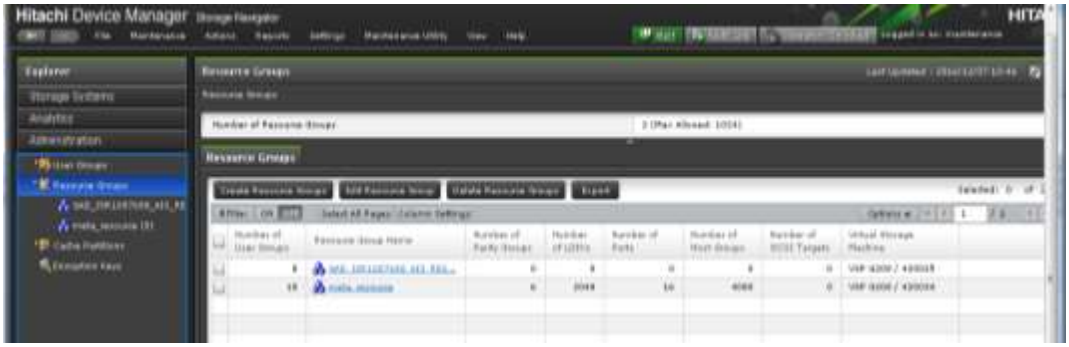
Configure storage resources (Fibre Channel connectivity)

All storage resources, such as DP pools and host groups, must have a name so that HBSD can use them (name fields cannot be left blank).

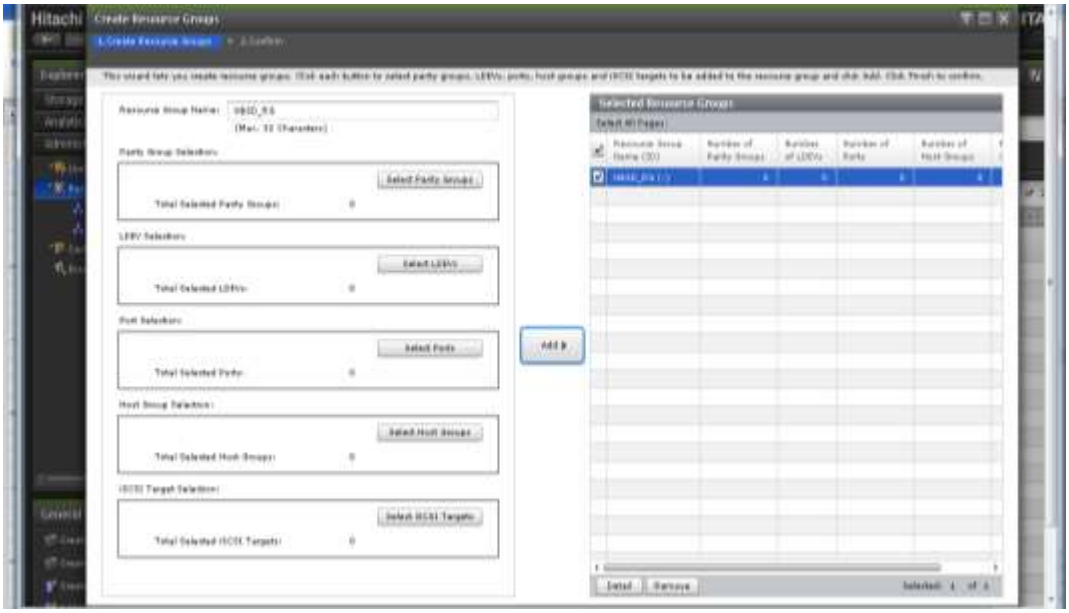
(1) Creating Resource Group

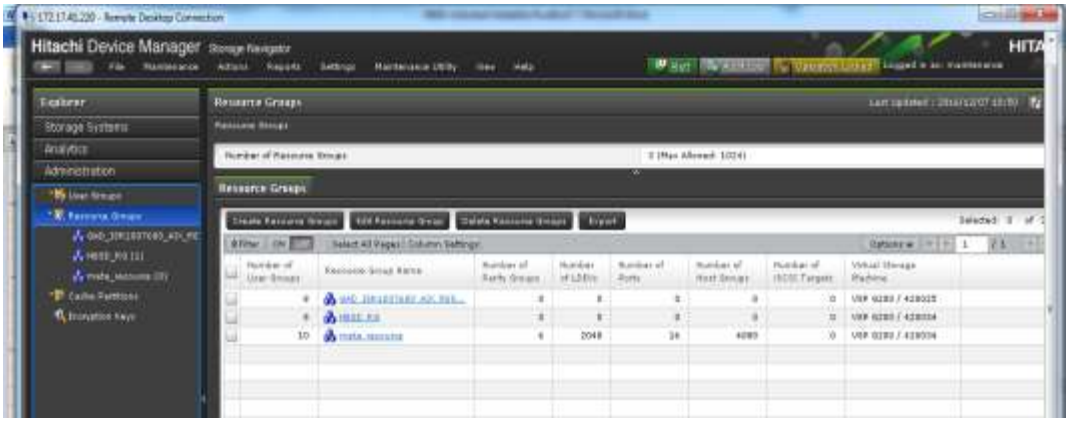
Existing resource group of the storage can be used for HBSD configuration. Also, a new resource group can be used exclusively for an OpenStack system. To create new resource group, refer following section

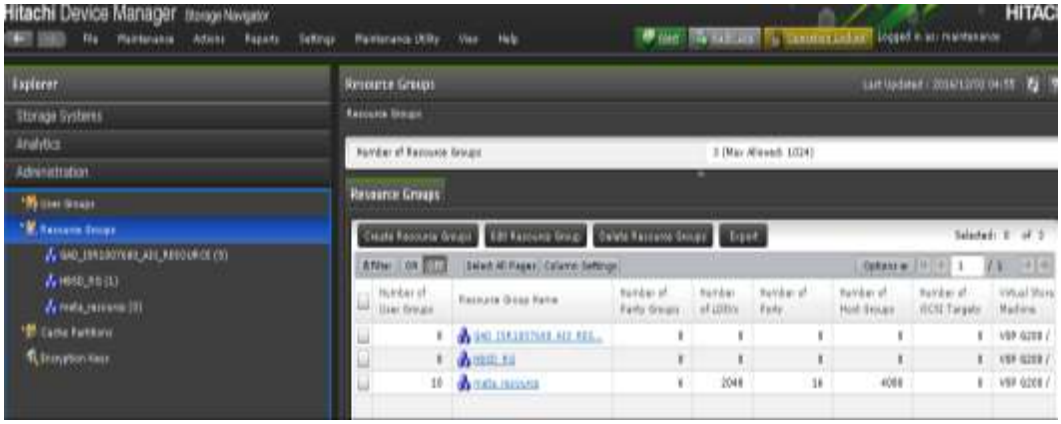
(a) Open the Hitachi Device manager, click on Administration -> Resource Groups -> **Create Resource Groups** tab



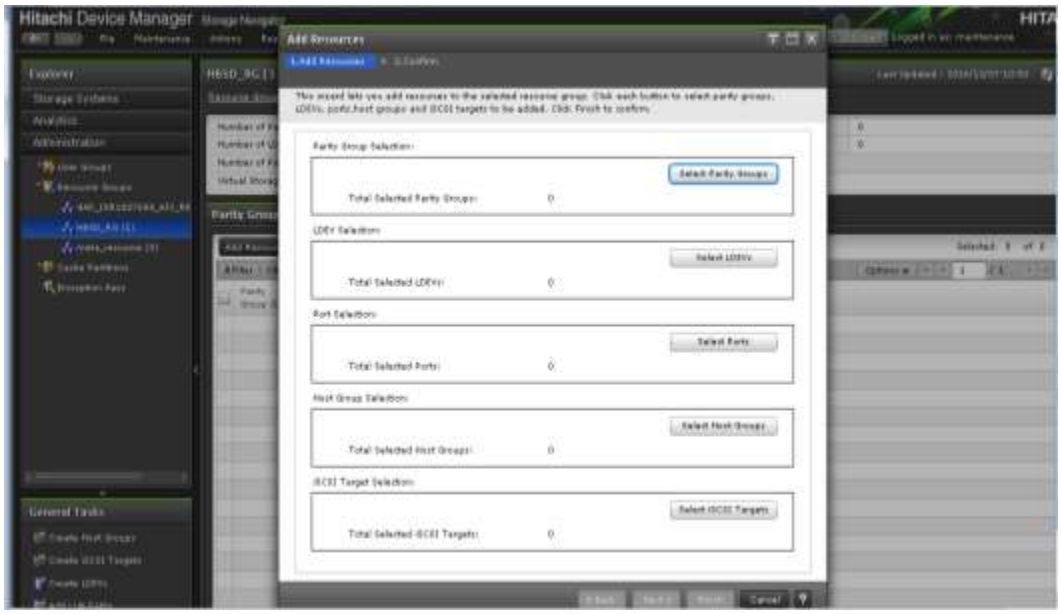
(b) Next, enter the "Resource Group Name" in Resource Group dialog box and create the resource group as shown below.







(c) After creation of new resource group, resources like Host Group, ldev can be added to the new resource Group.
Click on Administration -> Resource Groups -> Resource Group Name "HBSD_RG" -> Add Resource.
Select the resources like "Host group", "LDEV" and click on the tab "Finish" to add resources to the resource Group.
Here is the screenshot of the operation.



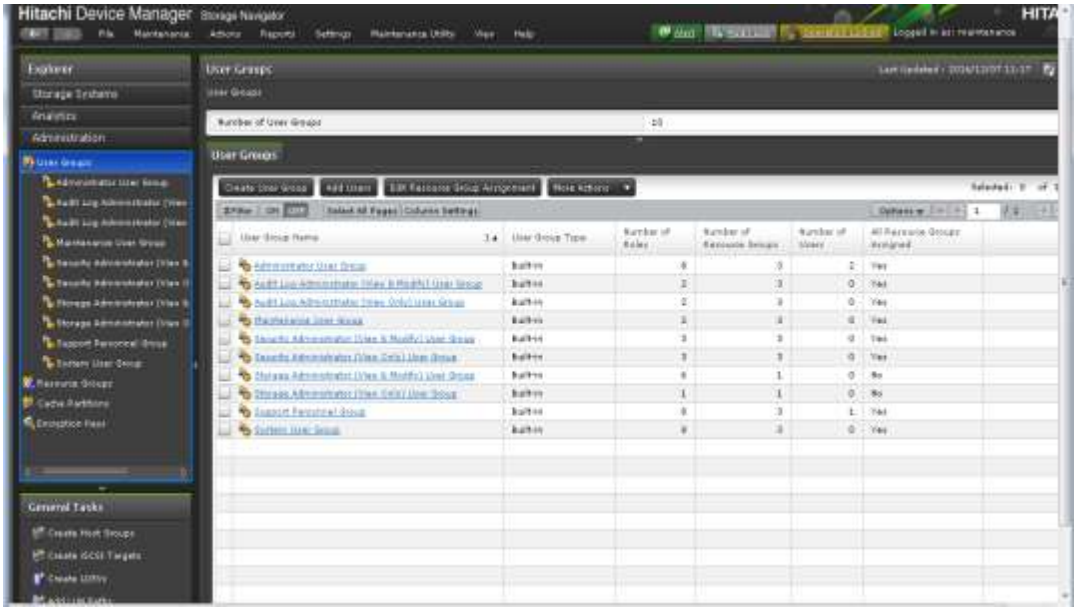
(2) Creating User accounts

Existing user groups can be used for HBSD configuration. Also, new user groups can be used exclusively for an OpenStack system. Create an account and assign the account to the following user groups:

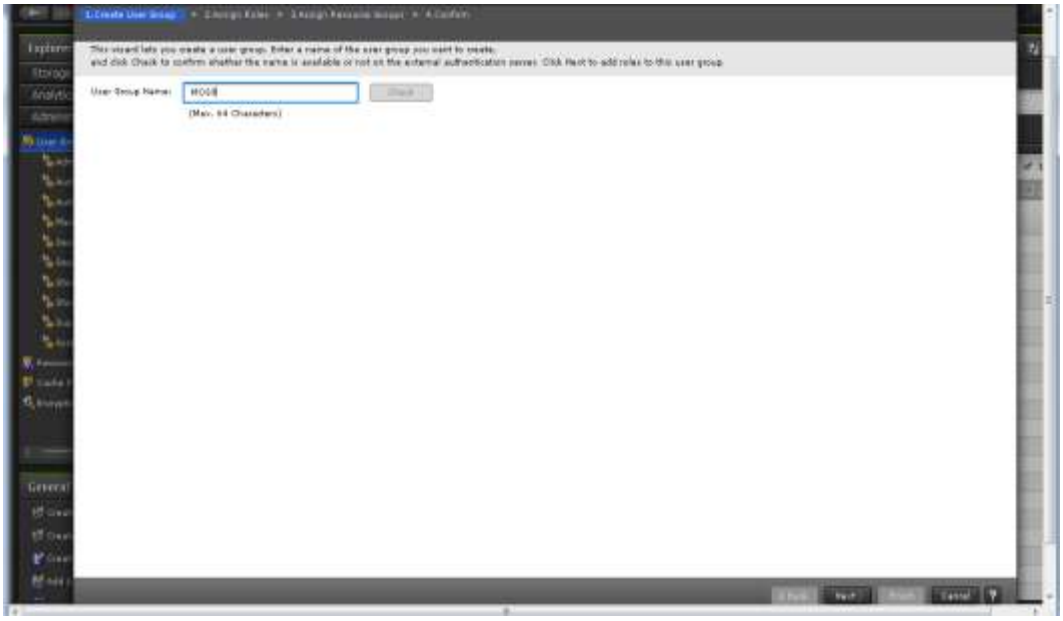
- Storage Administrator (View Only)
- Storage Administrator (Provisioning)
- Storage Administrator (Local Copy)
- Storage Administrator (Performance Management)

Note: These user groups must have management privileges for the created Resource-Group.

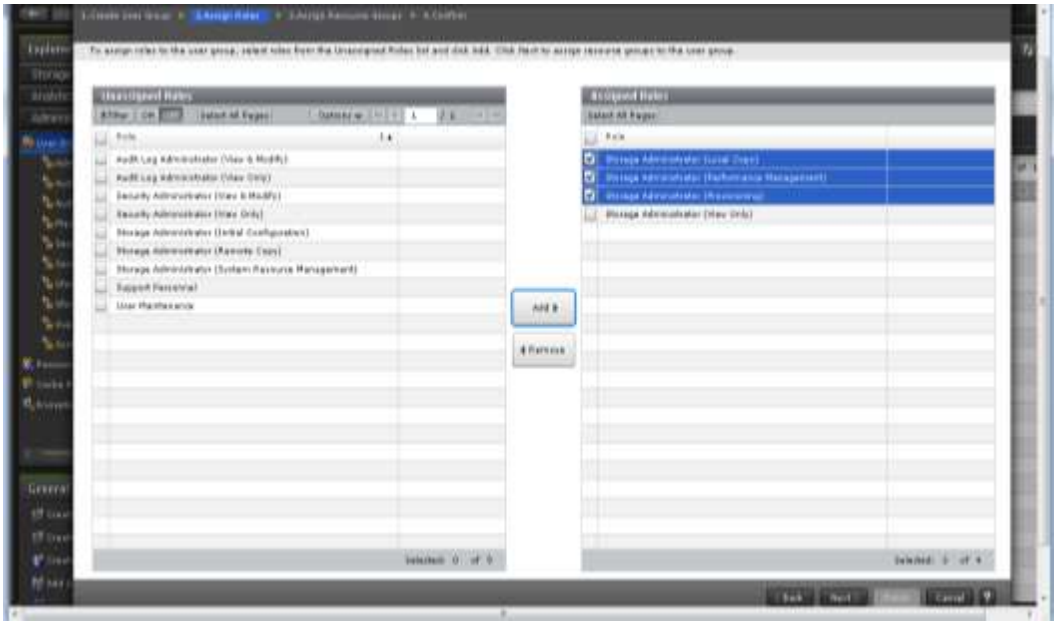
(a) User Groups can be created from Hitachi Device manager.
Open the Hitachi Device manager, click on Administration -> User Groups -> Create User Group tab.



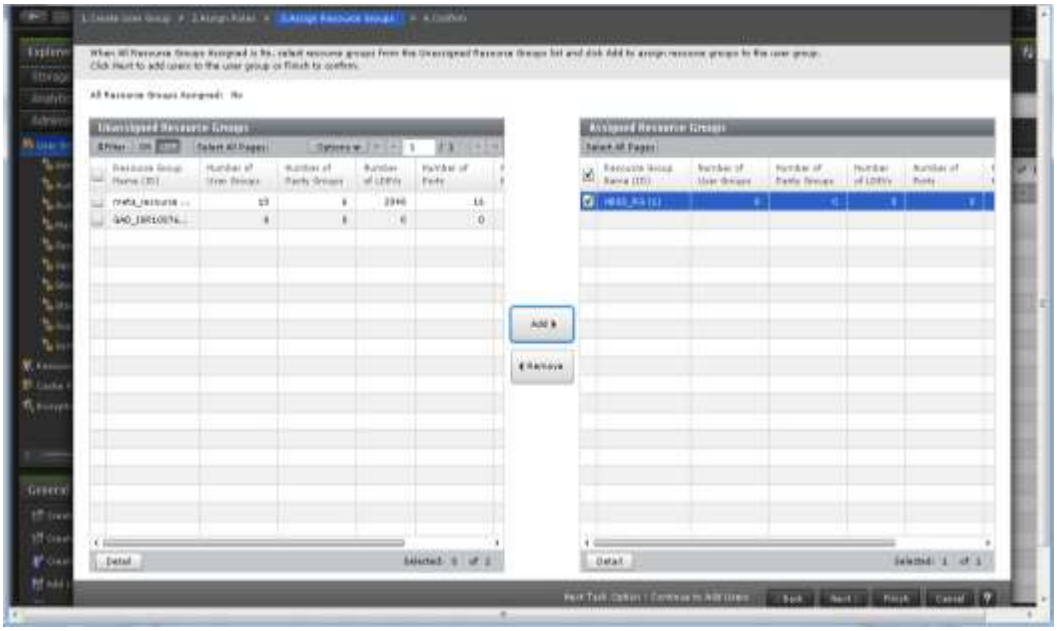
(b) After that, click on “Create User Group” tab in ”user Group name” section, Provide the name of the user group and click on next.

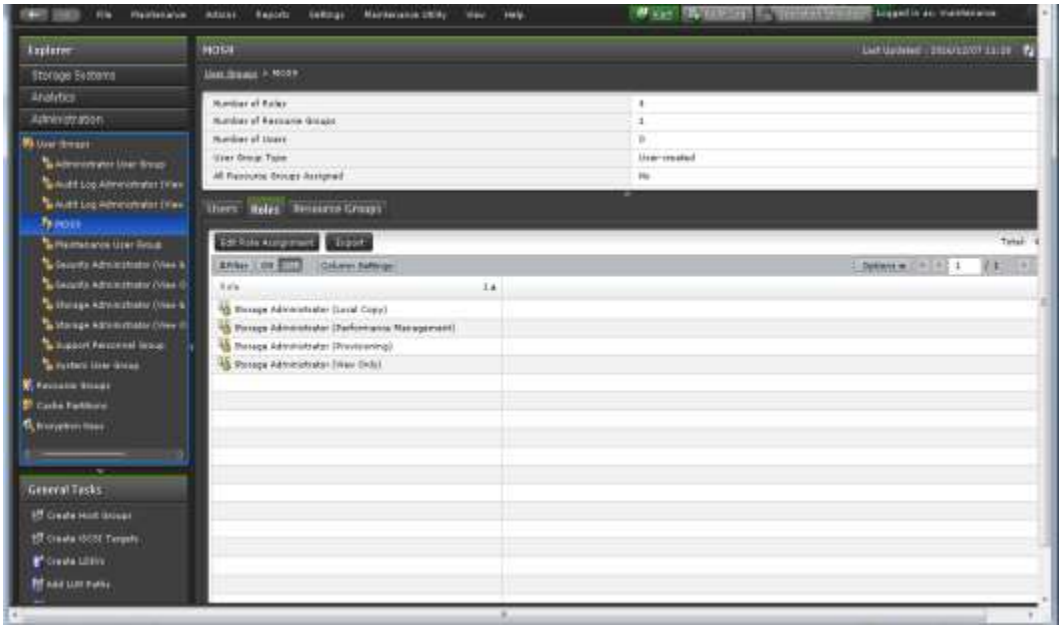


(c) Then, assign the role to the new user group and click on next.



(d) Next, assign the resource group to the User group and click on finish.





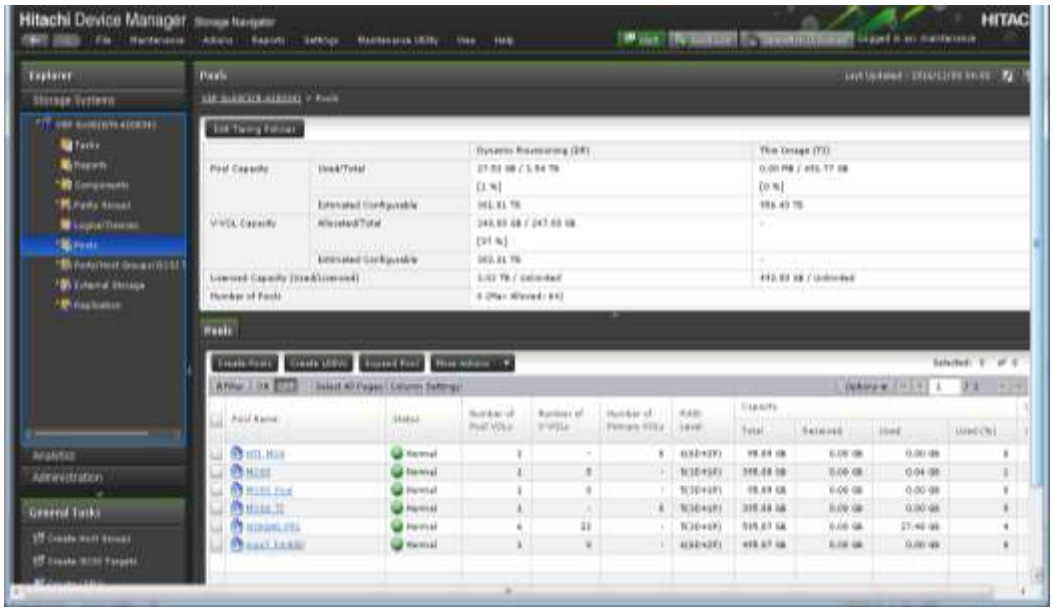
(3) **Create Dynamic Provisioning pool.**

Create a DP pool that is used by HBSD backend. HBSD manages it by a virtual capacity (the capacity reserved for the overprovisioning of the actual capacity of the DP pool), thus set the overprovisioning based on an operation policy. If the overprovisioning is set to 100%, space for the actual capacity is guaranteed.

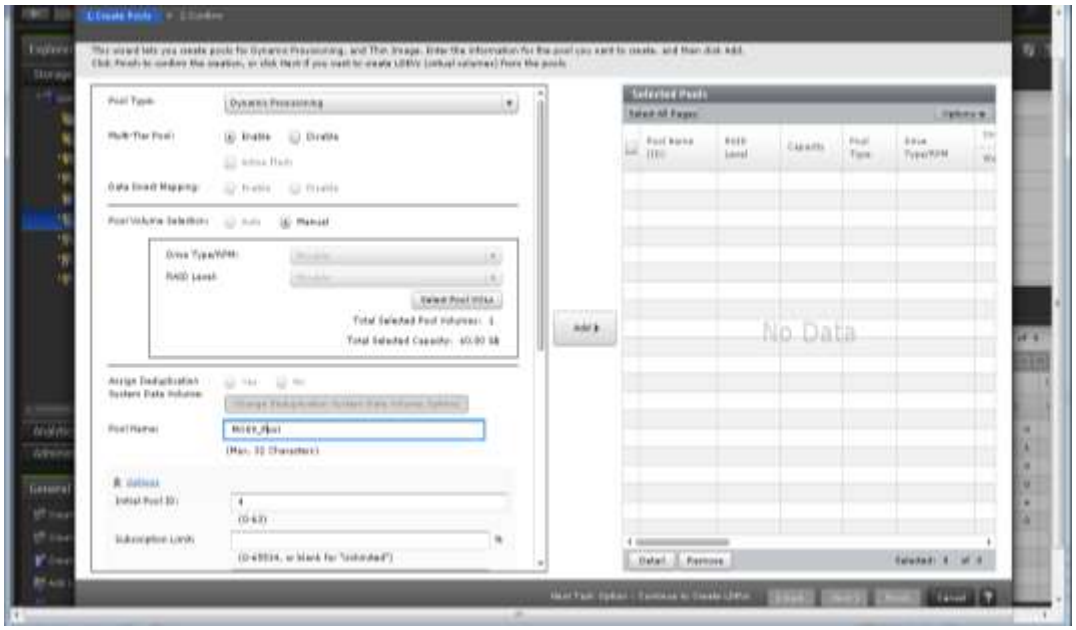
If using Thin Image, create a pool for Thin Image.

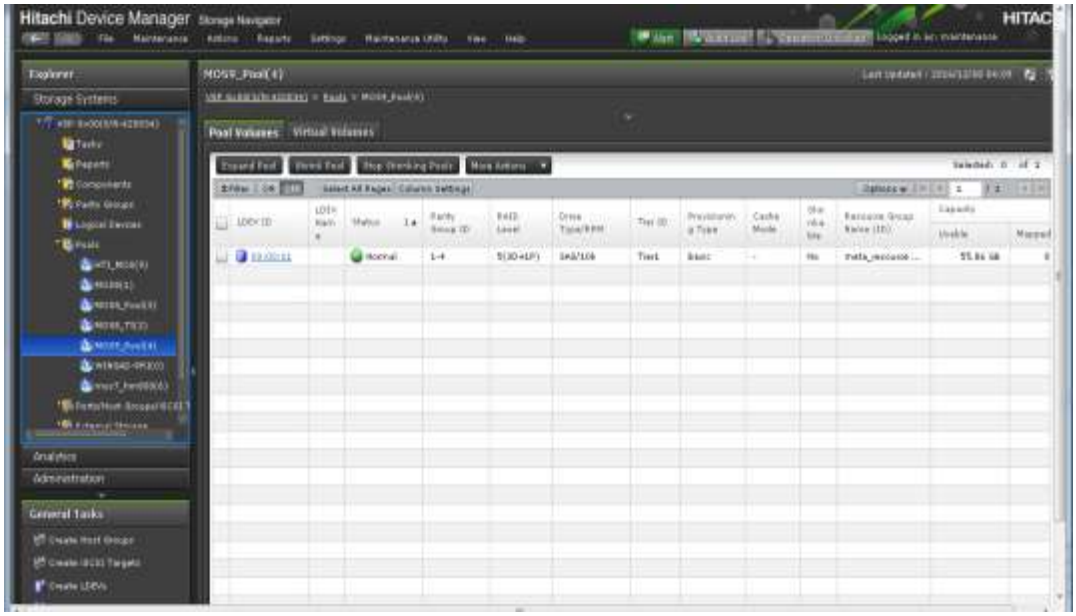
To create a new DP, please refer following section

(a) Open the Hitachi Device Manager, click on Pools -> Create Pools tab



(b) Next, select Pool Type as "Dynamic Provisioning" in the dialog box shown below,
Provide the Name of the DP Pool in the dialog box "Pool Name",
Select the LDEV as per requirement in Pool Volume option, and then click on "Finish"





(4) Setting Fibre Channel zoning

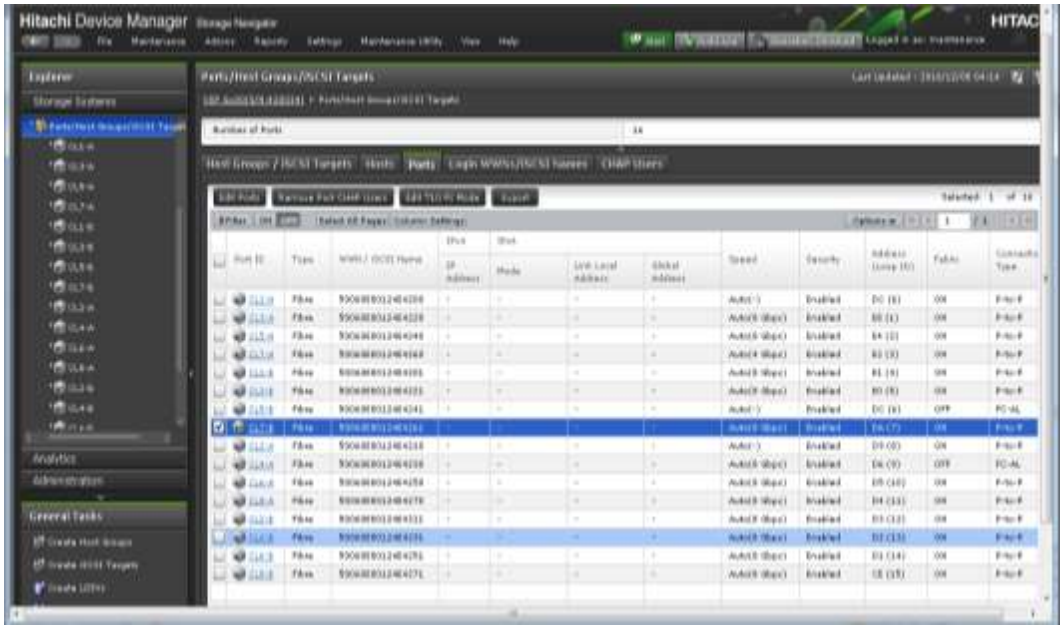
Manually configure the zoning in FC switch as per switch manufacturer documentation for connecting nodes with the storage devices using FC switch.

(5) Storage Port Setting

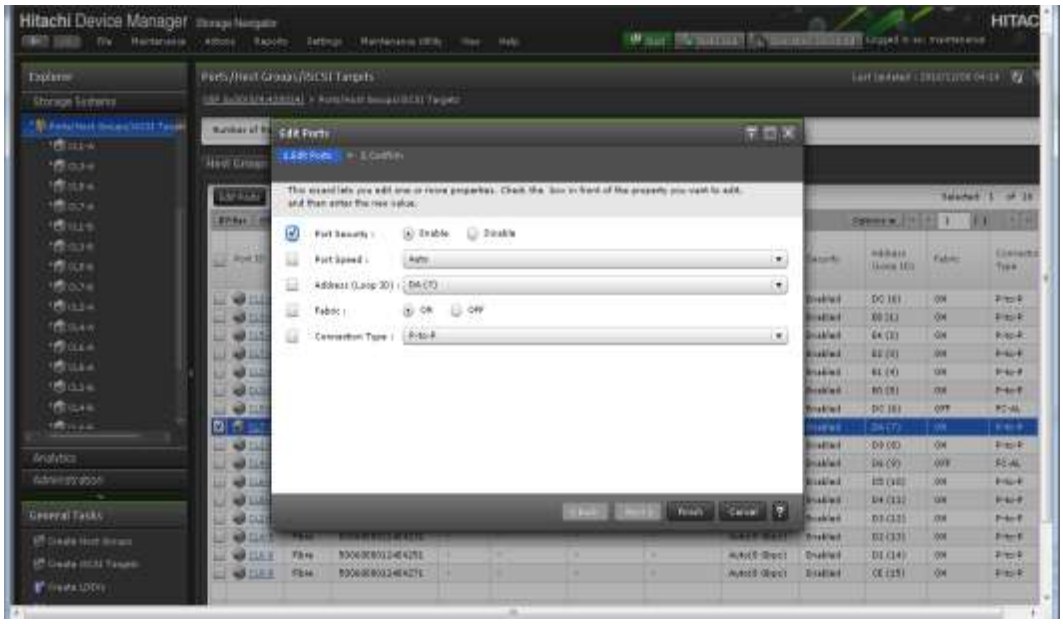
Enable Port Security for the ports used by HBSD. If you change the port configuration used by HBSD, restart the service "openstack-cinder-volume". Failing to restart this service will impact attach or detach volume operation.

Please refer following section to enable the port security of the port

(a) Open the Hitachi Device manager, click on Ports/Host Groups/iSCSI Targets -> select the port -> Click on "Edit Ports" tab



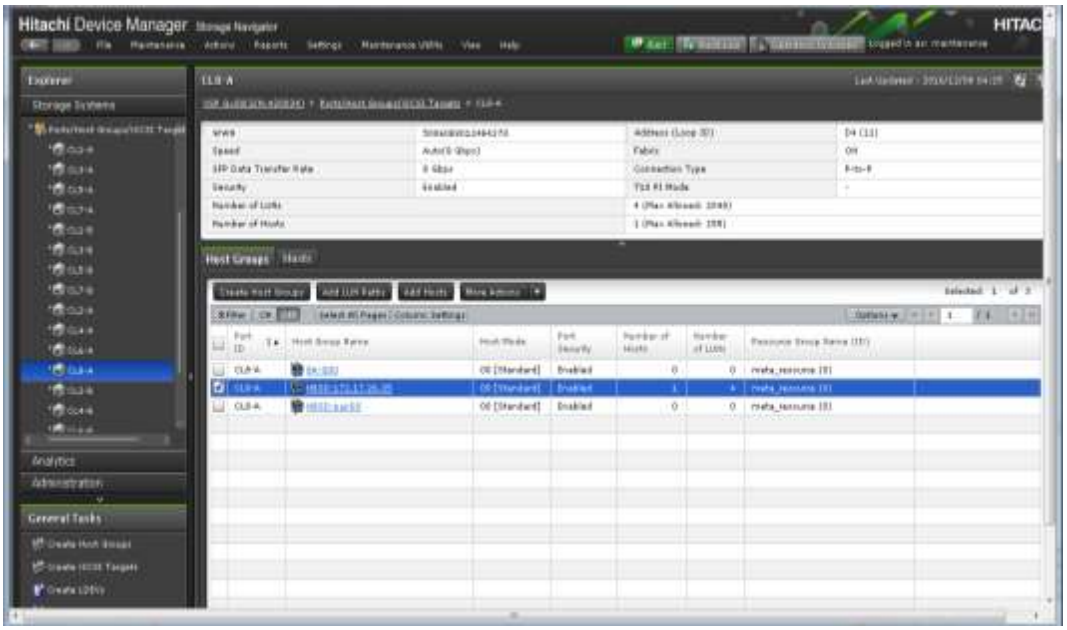
(b) Then select “enable” the option “Port Security” and click on “Finish”.



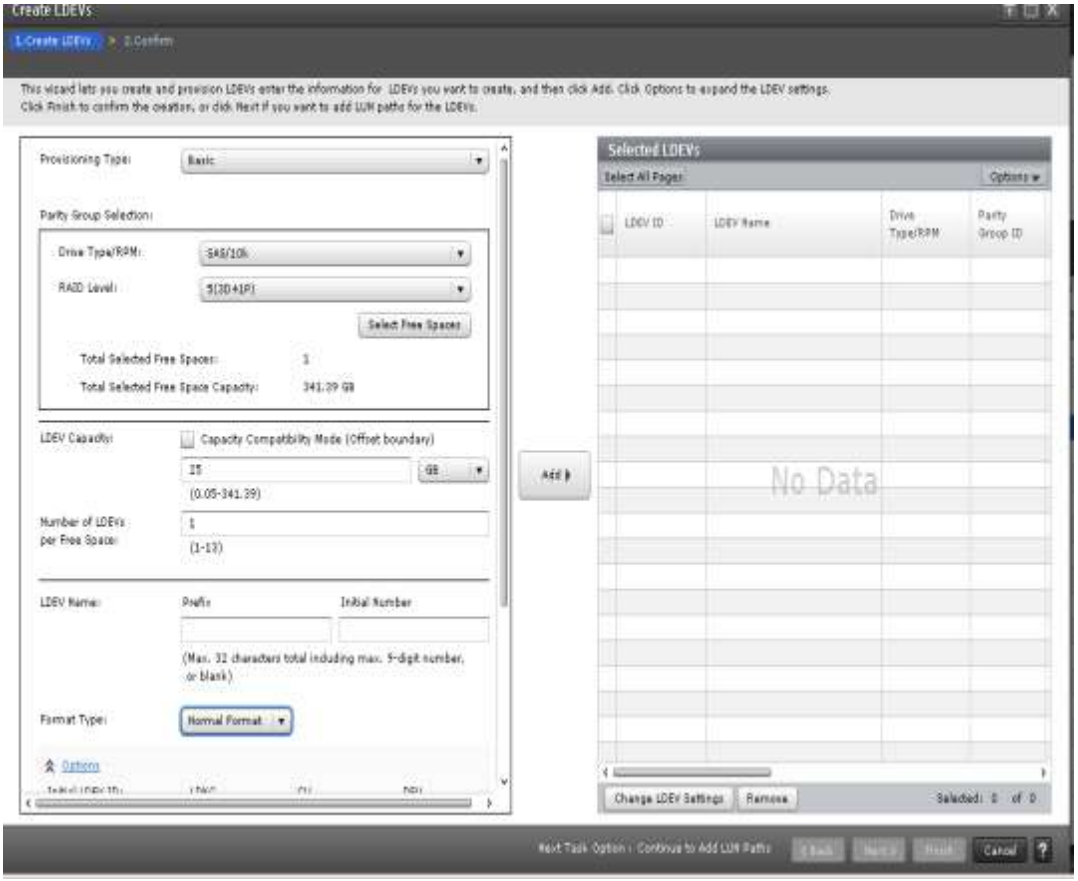
(6) Creating Host group for storage control path

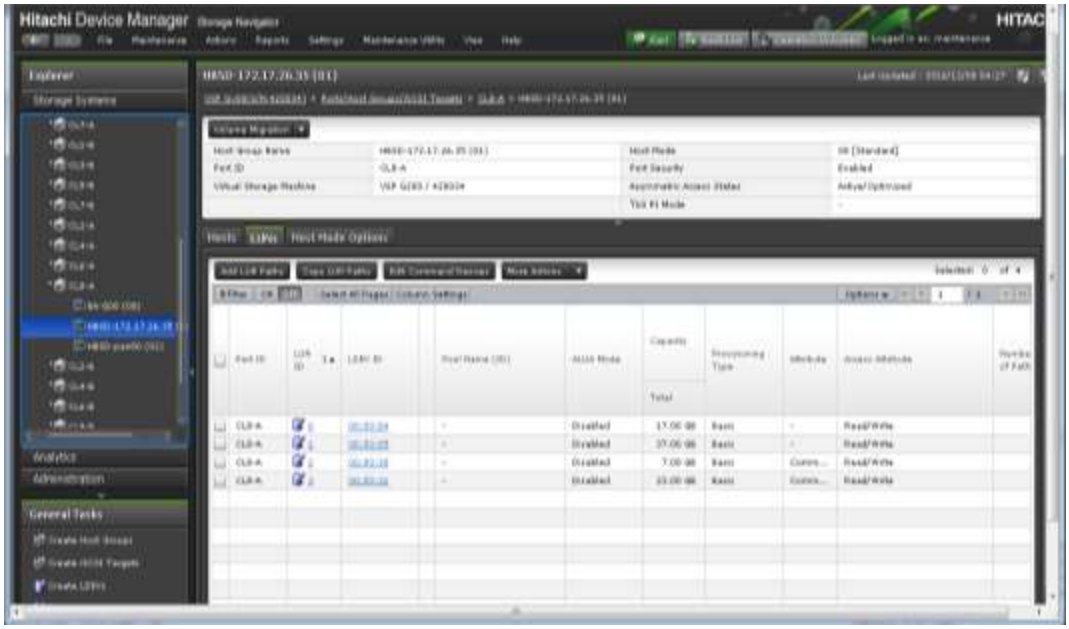
Set the host group for the Controller node, so that the Controller node can operate with the target storage device via the command device (In-Band). Please execute the following steps.

(a) Manually create a new host group in the port used for storage control path. Configure fiber channel switch zoning as per the switch vendor documentation. Manually select WWN of the HBA available in the storage.

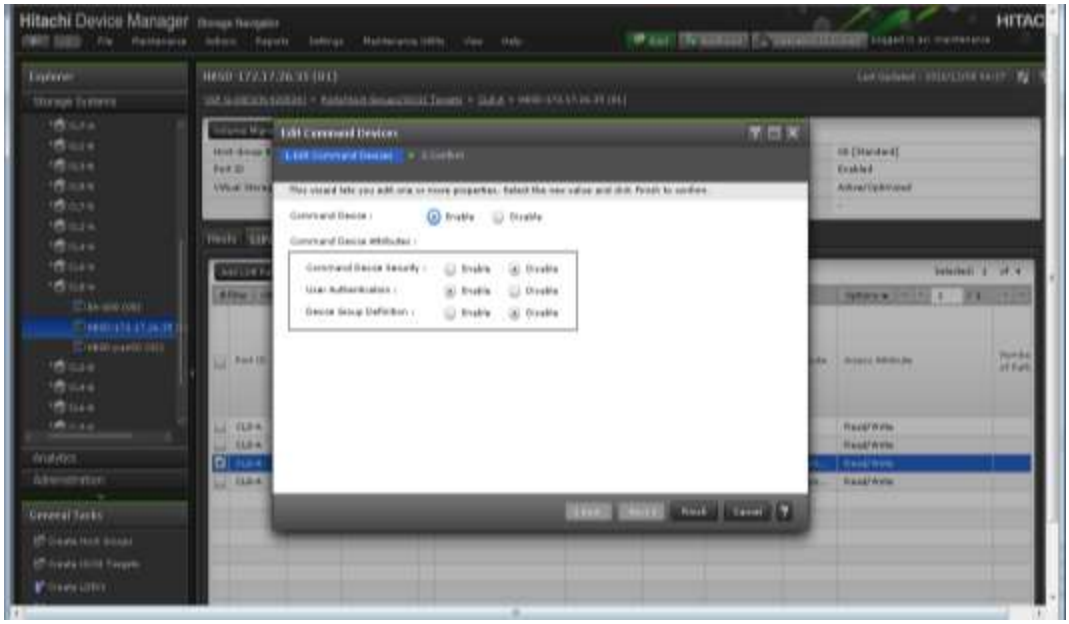


(b) Manually create a LDEV and map the LDEV to the newly created host group for the controller node.





(c) Open the Hitachi Device manager -> select the Host Group -> click on "Edit Command Device" -> click on "enable" for "Command device"->select "disable" for "Command device Security ", "enable " for "User Authentication" and "disable" for "Device Group Definition".



(7) Creating Host group for I/O data path

Create host group for all the storage ports that connects to the compute nodes. In multipath configuration, create host group for all connected ports.

The host group must to be named as "HBSD-<my_ip>". For example: "HBSD-172.17.26.34". my_ip must be the same value as the setting for the service (cinder or nova compute) in each node

Register the WWN of the connected node in the newly created host group.

(8) System reboots

Finally system reboot is required to complete the configuration and for making storage available to the controller node and compute node.

Configuring storage resources for iSCSI connectivity

All storage resources, such as DP pools and host groups, must have a name so that HBSD can use them (name fields cannot be left blank).

(1) Creating Resource Group

Please refer Resource Group section of FC connectivity in this document.

(2) Creating User accounts

Please refer User accounts section of FC connectivity in this document.

(3) Creating Dynamic Provisioning pool.

Please refer DP pool section of FC connectivity in this document.

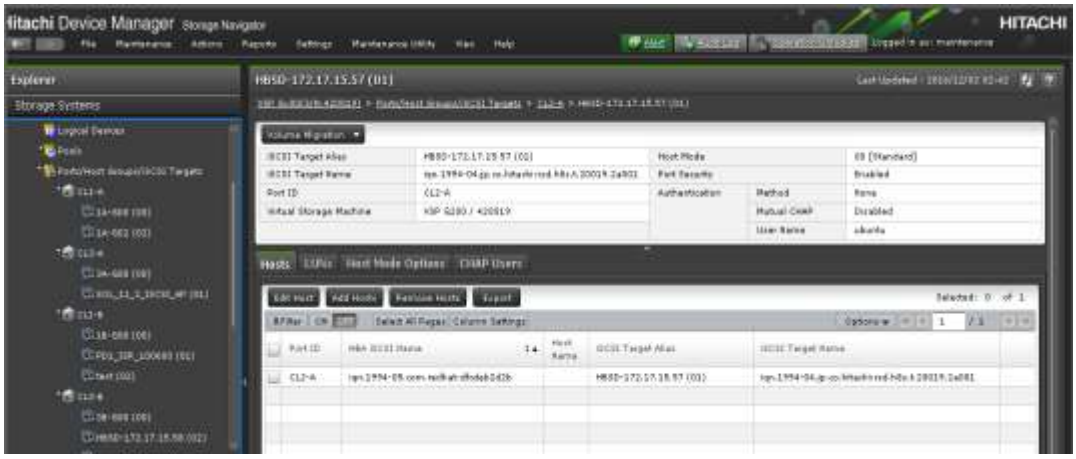
(4) Storage Port Setting

Please refer Port Setting section of FC connectivity in this document.

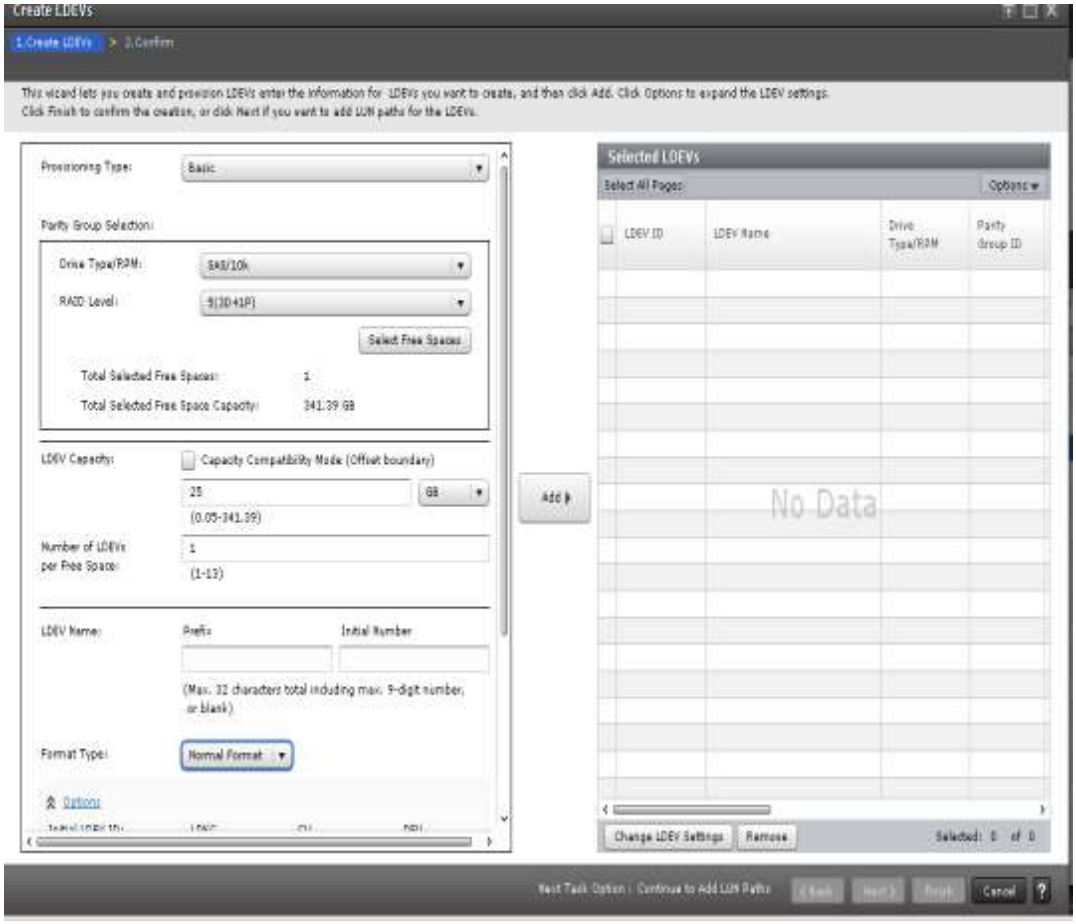
(5) Creating iSCSI target for storage control path

Set the iSCSI target for the Controller node, so that the Controller node can operate with the target storage device via the command device (In-Band). Please execute the following steps.

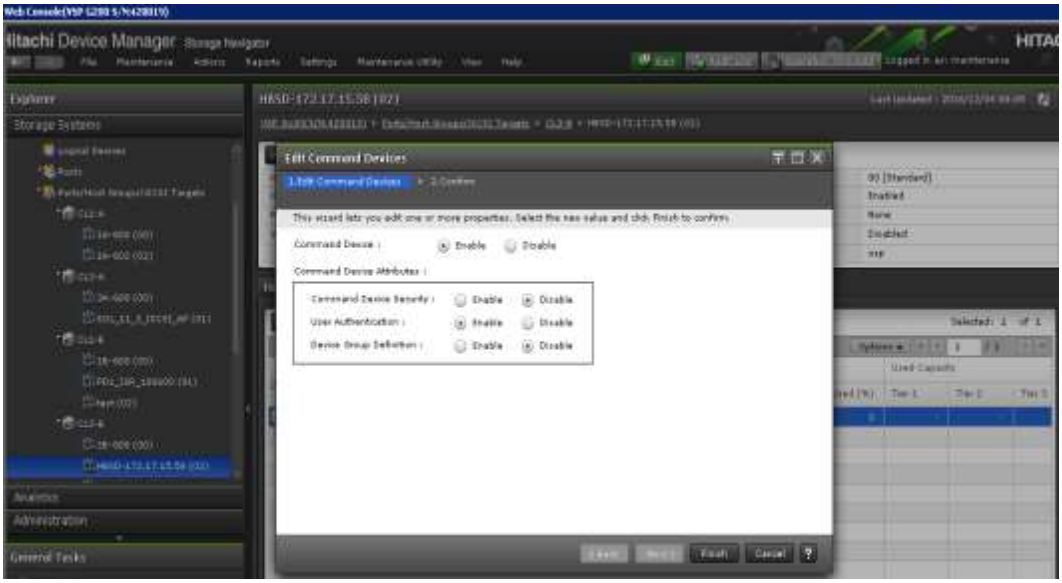
(a) Manually create a new iSCSI target in the port used for storage control path. Configure iSCSI Initiator IQN and ports depending on your environment. The ports used for storage control path cannot be used for I/O data path and separate port must be used for each path.



(b) Manually create a LDEV and map the LDEV to the newly created iSCSI target for the controller node.



(c) Open the Hitachi Device manager -> select the iSCSI target -> click on "Edit Command Device" -> click on "enable" for "Command device"->select "disable" for "Command device Security ", "enable " for "User Authentication" and "disable" for "Device Group Definition".



(6) Configuring iSCSI target for I/O data path

Manual configuration:

Create an iSCSI target for all the storage ports that connects to the compute nodes. In multipath configuration, create iSCSI target for all connected ports.

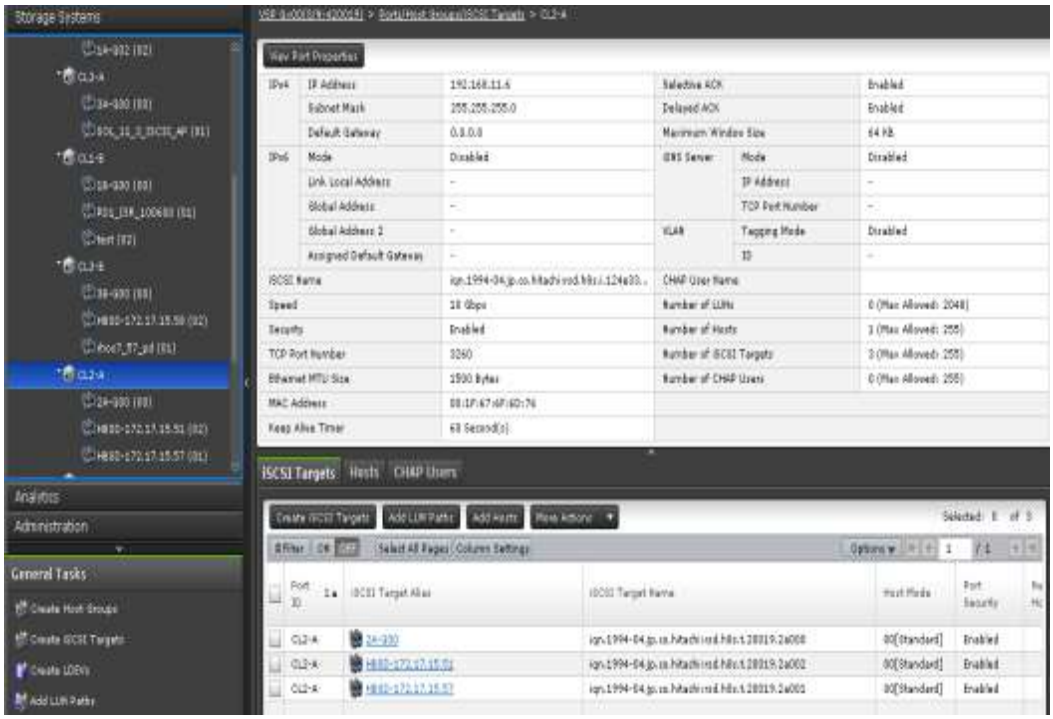
The iSCSI target must to be named as "HBSD-<my_ip>". For example : "HBSD-172.17.26.33". my_ip must be the same value as the setting for the service (cinder or nova compute) in each node

Register the Initiator IQN of the connected node in the newly created iSCSI target.

Register the Target IQN in the newly created iSCSI target. In the multipath configuration, register the same target IQN in all iSCSI targets connected to a node.

In the multipath configuration, set HostModeOption=83 on the created iSCSI targets.

If CHAP authentication is used, register the CHAP user name and password with the iSCSI target. Use the same CHAP user and password for all iSCSI targets.



3. Install and Configure the storage management software:

Designated management software must be configured on the Controller node for each target storage device.

- Setting of CCI for VSP G1000/ VSP G200, G400, G600, G800/VSP/HUS VM
  - Install CCI to the Controller node.
  - At the command device (In-Band),
    - Confirm that there is a connection to a command device.
    - Create the configuration file for horcm instance

4. HBSD Installation:

Follow the procedure given below to install HBSD package.

- Use the dpkg command to install HBSD.

- You must log in as a super user (root) on the Controller node where you want to install HBSD.
  - o Before installing HBSD, stop the cinder-volume service.

/usr/sbin/service cinder-volume stop

o If you use the cinder-backup service, stop that service also.
/usr/sbin/service cinder-backup stop

o Perform the installation.
dpkg -i hbsd_ 2.1.0-0-8.0_all.deb

Note: The HBSD package will be available from Hitachi Data Systems support team. Kindly contact Hitachi Data Systems in order to get and use this package.

==========

5. Initial Settings:

Mirantis OpenStack needs HBSD configuration along with cinder, edit the configuration file (/etc/cinder/cinder.conf) on the Cinder node by manually.

• Associating volume type and backend.

/usr/bin/cinder type-create <volume type name>

/usr/bin/cinder type-key <volume type name> set volume_backend_name=<volume backend name>

• Adding the configuration of HBSD.

According to the using of OpenStack configuration installer, add the configuration of HBSD to the editing target. After this, adding it to the configuration file (/etc/cinder/cinder.conf) provided by the OpenStack-cinder package is explained.

In DEFAULT section:

o Enable backend list: VSPG200, HUS100(shown in the cinder.conf sample below)
o logging format: Thread information is add to default format to log analysis.

In VSPG200 section:

o Backend definition section: VSPG200 (any string)
o Backend name registered with the volume type using the cinder type-key command: hbsd_backend
o Volume driver: cinder.volume.drivers.hitachi.hbsd.hbsd_fc.HBSDFCDriver
o Storage device serial number: 12345
o DP pool ID: 0
o TI pool ID for Thin Image: 1
o Login user name to the target storage: user
o Login password to the target storage: password
o Storage controller port names which Controller node uses: CL1-A, CL2-A
o Storage controller port names which Compute nodes use : CL1-B, CL2-B

#The following table provides a sample for cinder.conf file

```
######################
# cinder.conf sample #
######################
[DEFAULT]
 : (Omitted)
enabled_backends=VSPG200
logging_context_format_string=%(asctime)s.%(msecs)03d %( process)d%(thread)s
%(levelname)s %(name)s [%(request_id)s %(user_identity)s]
%(instance)s%(message)s
 : (Omitted)
[VSPG200]
volume_driver=cinder.volume.drivers.hitachi.hbsd.hbsd_fc.HBSDFCDriver
volume_backend_name=hbsd_backend1
hitachi_storage_cli=HORCM
hitachi_storage_id=12345
hitachi_pool=0
hitachi_thin_pool=1
hitachi_horcm_user=user
hitachi_horcm_password=password
hitachi_target_ports=CL1-A,CL-2A
hitachi_compute_target_ports=CL-1B,CL2-B
```

6. Syntax of Hitachi Block Storage Driver for OpenStack:

Specify "parameter=value" pair per line. The table shown below describes the HBSD specific parameters that has to be defined in HBSD settings in the configuration file (/etc/cinder/cinder.conf) provided by the OpenStack cinder package.

Name	Description
hitachi_storage_cli	Specify the CLI type to operate the storage device.
hitachi_storage_id	Specify the chassis ID of the storage device to operate.
hitachi_pool	Specify the ID of the DP pool (integer) or pool name that stores LDEVs for volumes (or snapshots).
hitachi_horcm_user	Specify the user name that the instance used by CCI uses to login to the storage.
hitachi_horcm_password	Specify the password that the horcm instance used by CCI uses to log in to the storage.
hitachi_target_ports	Specify the controller port name to search host groups(iSCSI targets) when attaching volumes.

Note: The above mentioned details are specific to Hitachi Storage and will be available with Hitachi Storage Administrator or User who has configured this Storage Device. Therefore, HBSD user has to get this information from them.

7. Restart the Cinder service:

- start the cinder-volume service
  #/usr/sbin/service cinder-volume start
  cinder-volume start/running, process <Process ID>

- If you use the cinder-backup service, start that service also.
  #/usr/sbin/service cinder-backup start
  cinder-backup start/running, process <Process ID>

8. Operation check:

- Pre-operation check by the storage operation software (ex. CCI or SNM2 CLI).
- Confirm that HBSD is being used.
- Confirm Create Volume
- Confirm Attach Volume
- Confirm Detach Volume
- Confirm Create Snapshot
- Confirm Create Volume from Snapshot
- Confirm Delete Volume
- Confirm Delete Snapshot
- Confirm Delete Volume

5.4 Limitations

OS	Mode	HV	Network	Storage
			VLAN	Ceph
Ubuntu	Standalone and HA [1*]	KVM	✓	x

[1*] - HA configuration cannot be done for "cinder-volume" service.

Note: Create a new OpenStack environment for MOS deployment with following limitations; HBSD does not support liberty on CentOS 6.5.
HBSD administrator requires storage backend with default providers [Cinder LVM over iSCSI for volumes] as this configuration setting is used to update Hitachi storage details with cinder-volume service.

5.5 Testing

5.5.1 Test cases

In addition to functional tests that are a part of the Fuel Health Check:
Verify instances connected to Hitachi Storage via HBSD with below mentioned functional testing.

#	Category	Function	Description
1	Provisioning	Create volume	Create new volume (DP-VOL)
2		Create cloned volume	Create new volume from existing volume using Shadow Image or Thin Image
3		Delete volume	Delete a volume
4	Snapshot	Create snapshot	Create a snapshot from a volume using Shadow Image or Thin Image
5		Create volume from snapshot	Create new volume from a snapshot using Shadow Image or Thin Image
6		Delete snapshot	Delete a snapshot
7	Attach / Detach	Initialize connection	Map the specified volume to a host group or iSCSI target
8		Terminate connection	Un map the specified volume to a host group or iSCSI target
9	Image creation	Copy image to volume	Copy OS image to the specified volume using dd
10		Copy volume to image	Copy the specified volume as OS image data using dd
11	Mange / Unmanage	Manage volumes	LDEV which Cinder of other OpenStack made is added under management of target Cinder.
12		Unmanage volume	The volume which Cinder made is removed from the Cinder management

5.5.2 Test results

#	Category	Function	Test Results
1	Provisioning	Create volume	Success
2		Create cloned volume	Success

3		Delete volume	Success
4	Snapshot	Create snapshot	Success
5		Create volume from snapshot	Success
6		Delete snapshot	Success
7	Attach / Detach	Initialize connection	Success
8		Terminate connection	Success
9	Image creation	Copy image to volume	Success
10		Copy volume to image	Success
11	Mange / Unmanage	Manage volumes	Success
12		Unmanage volume	Success

6. Troubleshooting

This section explains how to perform troubleshooting for HBSD.
Service cinder-volume does not start:

- An error message for HBSD is output to "/var/log/cinder/cinder-volume.log". Kindly check and take necessary action to resolve the cause.

- If no error message is logged for HBSD in var/log/cinder/cinder-volume.log", then check "/var/log/hbsd/debug.log" file and takes necessary action to resolve the cause.

- Similarly, do troubleshoot all issues related to HBSD functionalities. [Ex: Create Volume, Create snapshot, etc.]

7. Conventions: Abbreviations for product names

- HBSD: Hitachi Block Storage Driver for OpenStack
- HUS 1xx: Hitachi Unified Storage Family
- HUS VM: Hitachi Unified Storage VM
- VSP: Hitachi Virtual Storage Platform
- VSP G1000: Hitachi Virtual Storage Platform G1000
- VSP G200: Hitachi Virtual Storage Platform G200
- VSP G400: Hitachi Virtual Storage Platform G400
- VSP G600: Hitachi Virtual Storage Platform G600
- VSP G800: Hitachi Virtual Storage Platform G800
- SNM2: Hitachi Storage Navigator Modular 2
- CCI: Command Control Interface