HITACHI
Inspire the Next

Hitachi NAS Platform Drivers for OpenStack

User Guide

FastFind Links

Hitachi Data Systems

MK-92ADPTR124-00

Contents

About this guide

This document contains the installation and user guide of the Hitachi NAS Platform (HNAS) Drivers for OpenStack. Although some Cinder operations are mentioned in this guide, describing OpenStack operations is out of the scope of this document.

Who should use this guide

This guide is intended for anyone who installs, configures, and performs Cinder operations. This document assumes that they have basic knowledge of Linux operating system.

Related information and publications

OpenStack documentation

- OpenStack Cloud Administrator Guide
- OpenStack Command-Line interface Reference
- OpenStack Configuration Reference
- Red Hat Enterprise Linux OpenStack Platform Product Manual
- SUSE OpenStack Cloud Product Manual

Hitachi NAS Platform and Virtual Storage Platform Gx00 documentation

- Hitachi NAS Platform 3080 and 3090 G1 Hardware Reference
- Hitachi NAS Platform 3080 and 3090 G2 Hardware Reference
- Hitachi NAS Platform Series 4000 Hardware Reference
- Hitachi NAS Platform System Manager Unit (SMU) Hardware Reference
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Virtual SMU Administration Guide
- Hitachi NAS Platform and Hitachi Unified Storage System Installation Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 System Access Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 File Service Administration Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Server Cluster and Administration Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Storage Subsystem Administration Guide

- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Backup Administration Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 User Administration Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Network Administration Guide
- Hitachi NAS Platform and Hitachi Virtual Storage Platform Gx00 Antivirus Administration Guide

Getting help

The Hitachi Data Systems Support Connect is the destination for technical support of products and solutions sold by Hitachi Data Systems. To contact technical support, log on to Hitachi Data Systems Support Connect for contact information: https://support.hds.com/en_us/contact-us.html.

Hitachi Data Systems Community is a global online community for HDS customers, partners, independent software vendors, employees, and prospects. It is the destination to get answers, discover insights, and make connections. **Join the conversation today!** Go to https://community.hds.com, register, and complete your profile.

Comments

Please send us your comments on this document: doc.comments@hds.com. Include the document title, number, and revision, and refer to specific section(s) and paragraph(s) whenever possible.

Thank you! (All comments become the property of Hitachi Data Systems Corporation.)

Chapter 1. Introduction

In OpenStack Block Storage (Cinder), the driver layer is the one that makes the communication between the commands the end user sends by the UI (Horizon) or CLI and the storage used as backend. The driver receives the commands from Cinder volume manager, sends the proper commands to the storage, returning the results to Cinder.

Figure 1 below shows an example of how the Hitachi NAS Platform (HNAS) drivers work in the overall OpenStack environment:

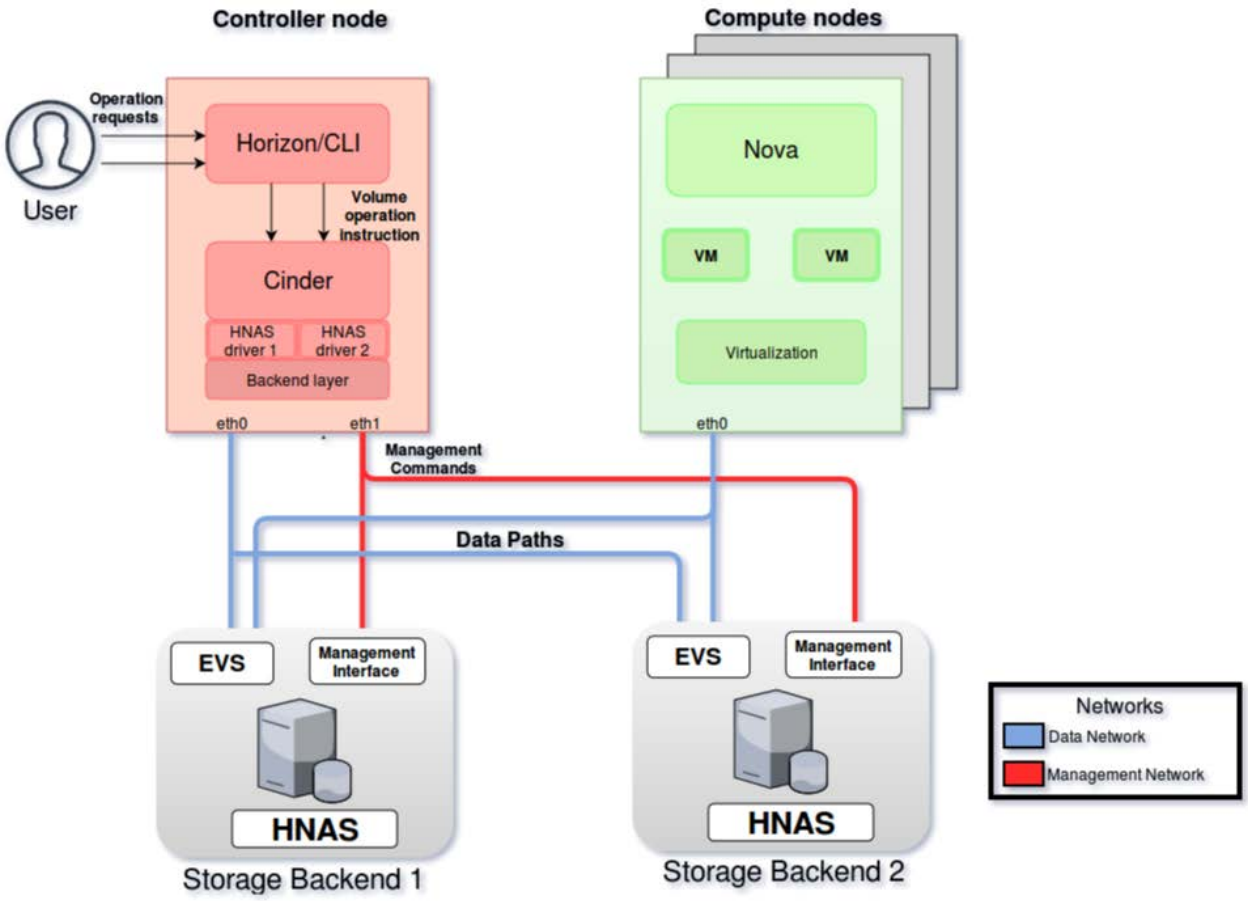

Figure 1 Overview of an OpenStack environment with HNAS

Driver Architecture

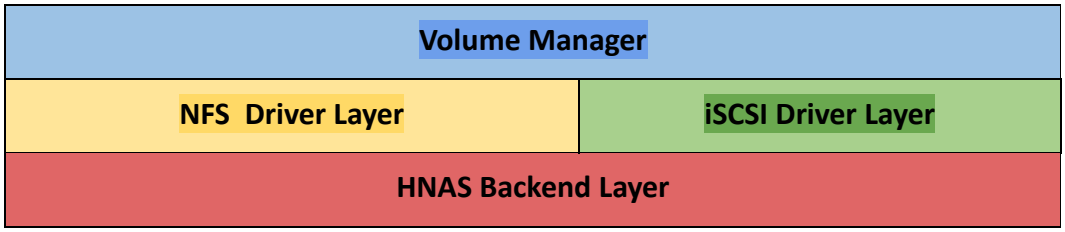

Figure 2 Concept diagram of the driver architecture

The Volume Manager is responsible for sending the commands to the specific driver. Each configured backend is an instance of the Volume Manager and calls the driver's specific behavior.

The NFS Driver Layer provides support to work with the Network File System protocol by using HNAS as a NFS server and the controllers, computes and storage nodes as clients. This layer mounts the NFS export configured on HNAS in those Cinder nodes and uses multiple Linux and HNAS backend commands to handle the Cinder volumes. Similarly, the iSCSI Driver Layer provides support to work with the iSCSI (internet Small Computer Interface) protocol by connecting iSCSI initiators in the controller, computes and storage nodes to iSCSI targets configured on HNAS file systems. The driver handles the operations by using the HNAS backend commands in the backend, including the creation and deletion of targets when needed.

The HNAS Backend Layer is responsible for executing commands in HNAS, parsing and formatting the output, and reporting back to the drivers (NFS or iSCSI), which actually contain all the logic. SSC handles the complexity of the protocol used to communicate with HNAS; it is by default installed in the HNAS system and is used via SSH by the HNAS Backend part of the driver.

The HNAS Driver (NFS or iSCSI) supports up to 4 different storage pools, such as file systems when using iSCSI or exports when using NFS, per backend. Each file system or export can be configured in HNAS to provide different levels of Quality of Service. These pools are selected by the Cinder scheduler according to the volume type associated with the volume being created.

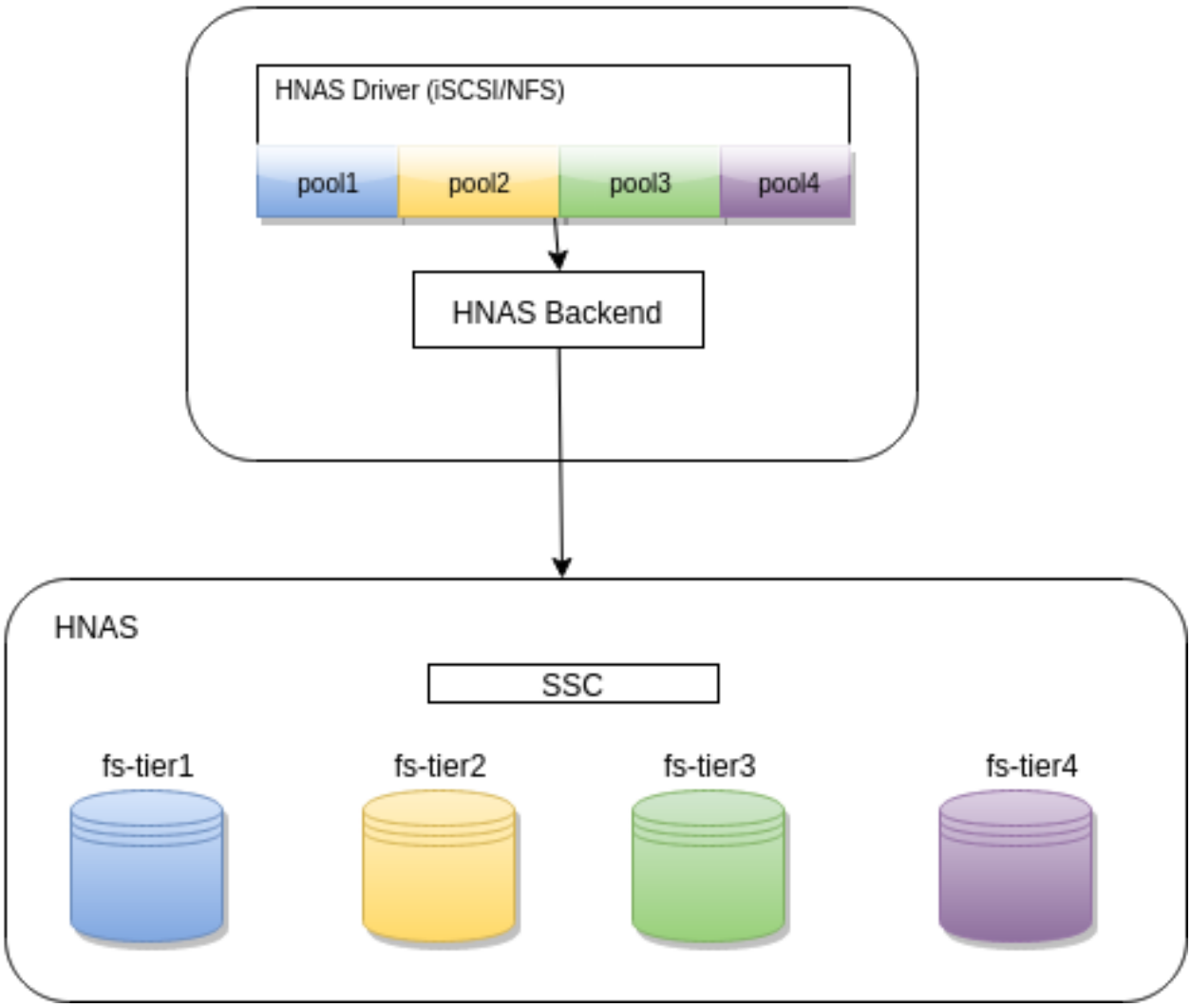

Figure 3 Communication diagram for HNAS driver

General considerations

Starting with the driver version 1.5.0, you can no longer run the driver using a locally installed instance of the SSC utility package. Instead, all communications with the HNAS backend are handled through SSH. This version also deprecates the xml configuration file in favor of having the entire driver configuration in the c*inder.conf* file.

Package nomenclature

The initial version of the driver released to the community had adopted the following version naming convention X.Y.Z, where X shows new features, Y, bug fixes, and Z, non-functional changes. The community version sequence is 1.0 for Juno, 3.0 for Kilo (version 2.0 was an intermediate version between Juno and Kilo), and 4.0 for Liberty.

The version numbering convention used for Hitachi enterprise drivers is different from that of the community drivers, and it is X.Y.Z-W-YYYY.R, where:

- X is the driver major version number;
- Y is the driver minor version number;
- Z id the bug fix version number;
- W is the package build number;
- YYYY.R is the OpenStack release with which this driver is compatible (e.g.: 2015.1 is the kilo release and 2015.2 is the liberty release);

The current enterprise version is v1.5.0-0-2015.2.

Hitachi NAS Platform (HNAS) requirements

The HNAS driver v1.5.0 provides support for HNAS models 3080, 3090, 4040, 4060, 4080 and 4100 with NAS OS 12.2 or higher.

Before using iSCSI and NFS services, use the HNAS configuration and management GUI (SMU) or SSC CLI to configure HNAS to work with the drivers.

1. General requirements:
    a. It's mandatory to have at least 1 storage pool, 1 EVS and 1 file system to be able to run any of the HNAS drivers.
    b. HNAS drivers consider the space allocated to the file systems to provide the reports to Cinder. So, when creating a file system, make sure it has enough space to fit your needs.
    c. The file system used should not be created as a replication target and should be mounted.

d. It's possible to configure HNAS drivers to use distinct EVSs and file systems, but all compute nodes and controllers in the cloud must have access to the EVSs.

2. For NFS:
   a. Create NFS exports, choose a path for them (it must be different from "/") and set the **Show snapshots** option to `hide` and `disable access`.
   b. For each export used, set the option `norootsquash` in the share "Access configuration" so Cinder services can change the permissions of its volumes. For example, "* (`rw, norootsquash`)".
   c. Make sure that all computes and controllers have R/W access to the shares used by Cinder HNAS driver.
   d. In order to use the hardware accelerated features of NFS HNAS, we recommend setting `max-nfs-version` to 3. Refer to HNAS command line reference to see how to configure this option.

3. For iSCSI:
   a. You must set an iSCSI domain to EVS.

OS and platform support

The HNAS driver version 1.5.0 is supported for Red Hat Enterprise Linux OpenStack Platform, SUSE OpenStack Cloud and Ubuntu OpenStack in the versions compatible with the OpenStack Liberty release. Note that these systems bring a community version of HNAS drivers that are not officially supported. To use the enterprise version of the drivers, you must follow the instructions described in Chapter 2. Installation and Chapter 3. Configuration.

Compatibility and requirements

The following packages must be installed in *all compute and controllers/storage nodes:*

- nfs-utils for Red Hat Enterprise Linux OpenStack
- nfs-client for SUSE OpenStack Cloud

- nfs-common, libc6-i386 and cinder-common for Ubuntu OpenStack

The following packages must be installed in *all controllers/storage nodes:*

- cinder-volume for Red Hat Enterprise Linux OpenStack, SUSE OpenStack Cloud and Ubuntu OpenStack Cloud.

Driver restrictions and limitations

- The driver does not manage a volume if the volume name has a slash ('/') or a colon (':') ,
- SSC simultaneous connections limit: In very busy environments, if 2 or more volume hosts are configured to use the same storage, some requests, such as create or delete, can have some attempts failed and retried (5 attempts by default) due to an HNAS connection limitation (max of 5 simultaneous connections).
- Each backend can have up to 4 services (pools).
- File system auto-expansion: Although supported, Hitachi Data Systems does not recommend using file systems with auto-expansion setting enabled because the scheduler uses the file system capacity reported by the driver to determine if new volumes can be created. For instance, in a setup with a file system that can expand to 200GB but is at 100GB capacity, with 10GB free, the scheduler will not allow a 15GB volume to be created. In this case, manual expansion would have to be triggered by an administrator. Hitachi Data Systems recommends always creating the file system at the maximum capacity or periodically expanding the file system manually.
- iSCSI driver limitations: The iSCSI driver has a limit of 1024 volumes attached to instances.
- The `hnas_svcX_volume_type` option must be unique for a given backend.

Cinder supported features

Following are the Cinder operations supported by the HNAS driver:

Create Volume	**Supported**
Delete Volume	**Supported**
Attach Volume	**Supported**
Detach Volume	**Supported**
Extend Volume	**Supported**
Create Snapshot	**Supported**
Delete Snapshot	**Supported**
List Snapshot	**Supported**
Create Volume from Snapshot	**Supported**
Create Volume from Image	**Supported**
Create Volume from Volume (Clone)	**Supported**
Create Image from Volume	**Supported**
Manage Volume	**Supported**
Unmanage Volume	**Supported**
Volume Migrate (host assisted)	**Supported**
Image Caching	**Supported**
Backup attached volumes	**Supported**
QoS	**Not Supported**
Volume Replication	**Not Supported**
Consistency Groups	**Not Supported**

Chapter 2. Installation

Install Packages

The HNAS drivers are distributed in rpm or deb packages. In order to install the drivers, simply download the latest version compatible with your supported Operating System from https://support.hds.com and install it through your OS official package manager.

Installation instructions for Ubuntu (deb)

Open a Linux terminal and execute the following command:

```
$ sudo dpkg -i hnas_1.5.0-0-2015.2_all.deb
```

Installation instructions for Red Hat / Suse (rpm)

Open a Linux terminal and execute the following command:

```
$ sudo rpm -ivh hnas-1.5.0_0_2015.2-1.el7.noarch.rpm
```

Or you can use another tool to automatically resolve dependencies problems if any:

```
$ sudo yum --nogpgcheck localinstall hnas-1.5.0_0_2015.2-1.el7.noarch.rpm
```

NOTE: The OpenStack Cinder HNAS Drivers should be installed in every Cinder node of your OpenStack deployment.

Uninstalling the OpenStack Cinder HNAS Driver:

In order to uninstall the OpenStack Cinder HNAS Drivers, simply use your package manager to remove the HNAS package from your system. This process will not remove your configurations from `/etc/cinder/cinder.conf`.

Uninstall instructions for Ubuntu (deb)

Open a Linux terminal and execute the following command:

```
$ sudo dpkg -r hnas
```

Uninstall instructions for Red Hat / SUSE (rpm)

Open a Linux terminal and execute the following command:

```
$ sudo rpm -e hnas-1.5.0_0_2015.2-1.el7.noarch
```

Chapter 3. Configuration

HNAS supports a variety of storage options and file system capabilities, which are selected through the definition of volume types combined with the use of multiple backends and/or multiple services. Each backend can configure up to 4 service pools, which can be mapped to Cinder volume types.

The configuration for the driver is read from the backend sections of the cinder.conf. Each backend section must have the appropriate configurations to communicate with your HNAS backend, such as the IP address of the HNAS EVS that is hosting your data, HNAS SSH access credentials, the configuration of each of the services in that backend, etc. You can find examples of such configurations in Configuration example.

NOTE 1: The new HNAS drivers still support the XML configuration the same way it was in the older versions, but it is recommended that you configure the new HNAS drivers only through the cinder.conf file, since the XML configuration file from previous versions is being deprecated as of version 1.5.

NOTE 2: It's not recommended to use the same NFS export or file system (iSCSI driver) for different backends. If possible, configure each backend to use a different NFS export or file system.

The table below provides the definition of each configuration option that can be used in a HNAS backend section in the *cinder.conf* file:

Option	Type	Default	Description
volume_backend_name	Optional	Not applicable	
volume_driver	Required	Not applicable	The python module path to the HNAS volume driver python class. When installing through the rpm or deb packages, you should configure this to cinder.volume.drivers.hitachi.hnas.hnas_iscsi.HNASISCSIDriver for the iSCSI backend or cinder.volume.drivers.hitachi.hnas.hnas_nfs.HNASNFSDriver for the NFS backend.
nfs_shares_config	Required (only for NFS)	/etc/cinder/nfs_shares	Path to the nfs_shares file. This is required by the base Cinder generic NFS river and therefore also required by the HNAS NFS driver. This file should list, one per line, every NFS share being used by the backend, i.e., all the values found in the configuration keys hnas_svcX_hdp in the HNAS NFS backend sections. E.g. For a backend configuration: [hnas-nfs] … hnas_svc0_hdp = 172.24.44.112:/export1 hnas_svc1_hdp = 172.24.44.113:/export2 The nfs_shares_config file should have the following content: 172.24.44.112:/export1 172.24.44.113:/export2
hnas_mgmt_ip0	Required	Not applicable	HNAS management IP address. Should be the IP address of the "Admin" EVS. It is also the IP through which you access the web SMU administration frontend of HNAS.
hnas_chap_enabled	Optional (iSCSI only)	True	Boolean tag used to enable CHAP authentication protocol for iSCSI

			driver.
hnas_username	Required	Not applicable	HNAS ssh username
hds_hnas_nfs_config_file and hds_hnas_iscsi_config_file	Optional (deprecated)	/opt/hds/hnas/cinder_[nfs|iscsi]_conf.xml	Path to the deprecated XML configuration file (only required if using the XML file)
hnas_cluster_admin_ip0	Optional (required only for HNAS multi-farm setups)	Not applicable	The IP of the HNAS farm admin. If your SMU controls more than one system or cluster, this option must be set with the IP of the desired node. Note that this is different for HNAS multi-cluster setups, which does not require this option to be set.
hnas_ssh_private_key	Optional	Not applicable	Path to the SSH private key used to authenticate to the HNAS SMU. Only required if you don't want to set hnas_password.
hnas_ssh_port	Optional	22	Port on which HNAS is listening for SSH connections
hnas_password	Required (unless hnas_ssh_private_key is provided)	Not applicable	HNAS password
hnas_svcX_hdp[1]	Required (at least 1)	Not applicable	HDP (export or file system) where the volumes will be created. Use exports paths for the NFS backend or the file system names for the iSCSI backend (note that when using the file system name, it does not contain the IP addresses of the HDP). Examples: NFS: hnas_svc0_hdp = 172.24.44.112:/export1 ISCSI: hnas_svc0_hdp = FS-cinder
hnas_svcX_iscsi_ip[1]	Required (only for iSCSI)	Not applicable	The IP of the EVS that contains the file system specified in hnas_svcX_hdp.
hnas_svcX_volume_type[1]**Error ! Bookmark not defined.**	Required	Not applicable	A **unique string** that is used to refer to this pool within the context of Cinder. You can tell Cinder to put volumes of a specific volume type into this backend, within this pool. See, Service labels and Configuration example for more details.
hnas_enable_trace	Optional	False	When set to True, enables the trace behavior of HNAS Drivers. It will log

[1] Replace X with a number from 0 to 3 (keep the sequence when configuring the driver).

			the driver functions calls, arguments and return values on /var/log/hnas/debug.log file. This option should be added in the **[DEFAULT]** section of cinder.conf.

Service labels

HNAS driver supports different types of service using the service labels. It is possible to create up to 4 types of them for each backend, for example, gold, platinum, silver, and ssd.

After creating the services in the cinder.conf configuration file, you need to configure one Cinder `volume_type` per service. Each `volume_type` *must have* the metadata `service_label` with the same name configured in the `hnas_svcX_volume_type` option of that service. See Configuration example for more details. If the volume type is not set, Cinderservice pool with largest available free space or other criteria configured in scheduler filters.

Multi-backend configuration

You can deploy multiple OpenStack HNAS driver instances (backends) that each controls a separate HNAS or a single HNAS. If you use multiple Cinder backends, remember that each Cinder backend can host up to 4 services. Each backend section must have the appropriate configurations to communicate with your HNAS backend, such as the IP address of the HNAS EVS that is hosting your data, HNAS SSH access credentials, the configuration of each of the services in that backend, etc. You can find examples of such configurations in Configuration example.

If you want the volumes from a `volume_type` to be casted into a specific backend, you must configure an `extra_spec` in the `volume_type` with the value of the `volume_backend_name` option from that backend.

For multiple NFS backends configuration, each backend should have a separated `nfs_shares_config` and, if using different shares for them, separated `nfs_shares` file defined (e.g. `nfs_shares1, nfs_shares2`) with the desired shares listed in separated lines. See the example below:

cinder.conf	Nfs_shares files
[backend-1] **nfs_shares_config = /home/cinder/nfs_shares1**	Path of the file -> */home/cinder/nfs_shares1* Content -> 172.24.44.112:/export1

	172.24.44.112:/export2
[backend-2] **nfs_shares_config = /home/cinder/nfs_shares2**	Path of the file -> /home/cinder/nfs_shares2 Content -> 172.24.44.113:/export3 172.24.44.114:/export4

Managing volumes

If there are some existing volumes on HNAS that you want to import to Cinder, it's possible to use the manage volume feature to do this. The manage action on an existing volume is very similar to a volume creation. It creates a volume entry on Cinder database, but instead of creating a new volume in the backend, it only adds a 'link' to an existing volume. Note that it's an admin-only feature and you have to be logged as an user with admin rights to be able to use this.

- For NFS:

  1. Under the tab Admin -> Volumes choose the option [+ Manage Volume]

  2. Fill the fields Identifier, Host and Volume Type with volume information to be managed:

  Identifier: ip:/type/volume_name (e.g. 172.24.44.34:/silver/volume-test)
  Host: host@backend-name#pool_name (e.g. ubuntu@hnas-nfs#test_silver)
  Volume Name: volume_name (e.g. volume-test)
  Volume Type: choose the type of volume (e.g. silver)

  By CLI:

```
$ cinder manage --name volume-test --volume-type silver 172.24.44.34:/silver/volume-test ubuntu@hnas-nfs#test_silver
```

- For iSCSI:

  1. Under the tab Admin -> Volumes choose the option [+ Manage Volume]
  2. Fill the fields Identifier, Host and Volume Type with volume information to be managed:

  Identifier: filesystem-name/volume-name (e.g. filesystem-test/volume-test)
  Host: host@backend-name#pool_name (e.g. ubuntu@hnas-iscsi#test_silver)
  Volume Name: volume name (e.g. volume-test)
  Volume Type: choose the type of volume (e.g. silver)

  By CLI:

```
$ cinder manage --name volume-test --volume-type silver filesystem-test/volume-test ubuntu@hnas-iscsi#test_silver
```

SSH configuration

You can use username and password to authenticate the Cinder storage node to the HNAS backend. In order to do that, simply configure `hnas_username` and `hnas_password` in your backend section within the cinder.conf file as shown below:

```
[hnas-iscsi]
…
hnas_username = supervisor
hnas_password = supervisor

[hnas-nfs]
…
hnas_username = supervisor
hnas_password = supervisor
```

Alternatively, the HNAS driver also supports SSH authentication through public key. To configure SSH authentication through public key:

1. If you don't have a pair of public keys already generated, create it in the Cinder storage node (leave the pass-phrase empty):

```
$ mkdir -p /opt/hitachi/ssh
$ ssh-keygen -f /opt/hds/ssh/hnaskey
```

2. Change the owner of the key to cinder (or the user the volume service will be run as):

```
# chown -R cinder.cinder /opt/hitachi/ssh
```

3. Create the directory "ssh_keys" in the SMU server:

```
$ ssh [manager|supervisor]@<smu-ip> 'mkdir -p /var/opt/mercury-main/home/[manager|supervisor]/ssh_keys/'
```

4. Copy the public key to the "ssh_keys" directory:

```
$ scp /opt/hitachi/ssh/hnaskey.pub [manager|supervisor]@<smu-ip>:/var/opt/mercury-main/home/[manager|supervisor]/ssh_keys/
```

5. Access the SMU server:

```
$ ssh [manager|supervisor]@<smu-ip>
```

6. Run the command to register the SSH keys:

```
$ ssh-register-public-key -u [manager|supervisor] -f ssh_keys/hnaskey.pub
```

7. Check the communication with HNAS in the Cinder storage node:

   For multi-farm HNAS:

```
$ ssh -i /opt/hitachi/ssh/hnaskey [manager|supervisor]@<smu-ip> 'ssc <cluster_admin_ip0> df -a'
```

   Or, for Single-node/Multi-Cluster:

```
$ ssh -i /opt/hitachi/ssh/hnaskey [manager|supervisor]@<smu-ip> 'ssc localhost df -a'
```

8. Configure your backend section in cinder.conf to use your public key:

```
[hnas-iscsi]
…
hnas_ssh_private_key = /opt/hitachi/ssh/hnaskey

[hnas-nfs]
…
hnas_ssh_private_key = /opt/hitachi/ssh/hnaskey
```

Configuration example

Below are configuration examples for both NFS and iSCSI backends:

1. HNAS NFS Driver

   1.1. For HNAS NFS driver, create this section in your cinder.conf file:

```
[hnas-nfs]
volume_driver = cinder.volume.drivers.hitachi.hnas.hnas_nfs.HNASNFSDriver
nfs_shares_config = /home/cinder/nfs_shares
volume_backend_name = hnas_nfs_backend

hnas_username = supervisor
hnas_password = supervisor
hnas_mgmt_ip0 = 172.24.44.15

hnas_svc0_volume_type = nfs_gold
hnas_svc0_hdp = 172.24.49.21:/gold_export

hnas_svc1_volume_type = nfs_platinum
hnas_svc1_hdp = 172.24.49.21:/silver_platinum

hnas_svc2_volume_type = nfs_silver
hnas_svc2_hdp = 172.24.49.22:/silver_export

hnas_svc3_volume_type = nfs_bronze
hnas_svc3_hdp = 172.24.49.23:/bronze_export
```

1.2. Add it to the enabled_backends list, under the DEFAULT section of your cinder.conf file:

```
[DEFAULT]
enabled_backends = hnas-nfs
```

1.3. Add the configured exports to the nfs_shares file:

```
172.24.49.21:/gold_export
172.24.49.21:/silver_platinum
172.24.49.22:/silver_export
172.24.49.23:/bronze_export
```

1.4. Register a volume type with Cinder and associate it with this backend:

```
$cinder type-create hnas_nfs_gold
$cinder type-key hnas_nfs_gold set \
  volume_backend_name=hnas_nfs_backend \
  service_label=nfs_gold

$cinder type-create hnas_nfs_platinum
$cinder type-key hnas_nfs_platinum set \
  volume_backend_name=hnas_nfs_backend \
  service_label=nfs_platinum

$cinder type-create hnas_nfs_silver
$cinder type-key hnas_nfs_silver set \
  volume_backend_name=hnas_nfs_backend \
  service_label=nfs_silver

$cinder type-create hnas_nfs_bronze
$cinder type-key hnas_nfs_bronze set \
  volume_backend_name=hnas_nfs_backend \
  service_label=nfs_bronze
```

2. OpenStack Cinder HNAS iSCSI Driver

2.1. For HNAS iSCSI driver, create this section in your cinder.conf file:

```
[hnas-iscsi]
volume_driver = cinder.volume.drivers.hitachi.hnas.hnas_iscsi.HNASISCSIDriver
volume_backend_name = hnas_iscsi_backend

hnas_username = supervisor
hnas_password = supervisor
hnas_mgmt_ip0 = 172.24.44.15
hnas_chap_enabled = True

hnas_svc0_volume_type = iscsi_gold
hnas_svc0_hdp = FS-gold
hnas_svc0_iscsi_ip = 172.24.49.21


hnas_svc1_volume_type = iscsi_platinum
hnas_svc1_hdp = FS-platinum
hnas_svc1_iscsi_ip = 172.24.49.21


hnas_svc2_volume_type = iscsi_silver
hnas_svc2_hdp = FS-silver
hnas_svc2_iscsi_ip = 172.24.49.22


hnas_svc3_volume_type = iscsi_bronze
hnas_svc3_hdp = FS-bronze
hnas_svc3_iscsi_ip = 172.24.49.23
```

2.2. Add it to the `enabled_backends` list, under the DEFAULT section of your cinder.conf file:

```
[DEFAULT]
enabled_backends = hnas-nfs, hnas-iscsi
```

2.3. Register a volume type with Cinder and associate it with this backend to complete the configuration:

```
$cinder type-create hnas_iscsi_gold
$cinder type-key hnas_iscsi_gold set \
  volume_backend_name=hnas_iscsi_backend \
  service_label=iscsi_gold

$cinder type-create hnas_iscsi_platinum
$cinder type-key hnas_iscsi_platinum set \
  volume_backend_name=hnas_iscsi_backend \
  service_label=iscsi_platinum

$cinder type-create hnas_iscsi_silver
$cinder type-key hnas_iscsi_silver set \
  volume_backend_name=hnas_iscsi_backend \
  service_label=iscsi_silver

$cinder type-create hnas_iscsi_bronze
$cinder type-key hnas_iscsi_bronze set \
  volume_backend_name=hnas_iscsi_backend \
  service_label=iscsi_bronze
```

Chapter 4. Troubleshooting

Resolving patching errors

During operation, you might encounter this warning message:

“This XML configuration file <FILE_NAME> is deprecated. Please, move all the configurations to the cinder.conf file. If you keep both configuration files, the options set on cinder.conf will be used.”

As the driver moves away from xml configuration files, it’ll emit deprecation warnings whenever it detects that a xml configuration file is in use. If that is your case, please follow the warning recommendations and move your configurations into your cinder.conf file, as described in the configuration section.

hnasgetinfo tool

Upon installation of the HNAS Driver, you will have access to the hnasgetinfo command line tool. It can be used to collect information about your system and current setup, which can in turn be used to help technical support solve any issues that you might have. In order to use the all hnasgetinfo tool features, it's recommended to us it with a user that has access to Cinder services. To do it, go to a directory in which you have write permissions, set the Keystone auth environment variables to use credentials that have access to the Cinder services, and run hnasgetinfo:

```
$export OS_USERNAME='your_openstack_user'
$export OS_PASSWORD='your_password’
$export OS_TENANT_NAME='tenant_of_your_openstack_user'
$export OS_AUTH_URL='http://<keystoneHost>:<keystonePort>/v2.0/'
$hnasgetinfo
```

The hnasgetinfo tool will produce, at the end of its execution, a tar.gz file in the current working directory containing the following data:

- Directories `/etc/cinder/* and /etc/nova/*`
- Directories `/var/log/cinder/* and /var/log/nova/*`
- Directories `/var/log/hnas/*`

- XML configurations files
- nfs_share files
- Cinder commands (e.g. cinder list, cinder type-list, cinder snapshot-list, etc)
- Openstack-status command
- Installed packages rpm or deb
- Network (ifconfig, ethtool, external ping and storage ping)
- iscsiadm -m commands
- showmount, df -h, proc/mounts
- check if the public key exists
- journalctl

Common errors and resolutions

- After changing the configuration on the storage, the Cinder services must be restarted (cinder-api, cinder-scheduler and cinder-volume). Note that in HA environments, the services need to be restarted in all controllers or storage nodes where cinder is running.
- On Red Hat, if the system is configured to use SELinux, you need to set "virt_use_nfs = on" for NFS driver work properly.

```
# setsebool -P virt_use_nfs on
```

- SSH authentication errors: If you are using ssh public key authentication, note that copying your Cinder host public key into HNAS using ssh-copy-id doesn't work properly as the SMU periodically wipes out those keys. Instead, you must use the ssh-register-public-key command to do that as described on *"SSH configuration"* section.
- Cinder fails when trying to manage an existing volume: If you want to manage a volume which is located in a service that **does not** have the option *hnas_svcX_volume_type* set as *'default'*, it will be necessary to include the volume type related to this service in the Cinder manage volume call. Also, it is

not possible to manage a volume if there is a slash ('/') or a colon (':') in the volume name.

Chapter 5. Best practices

Use NFS protocol

The HNAS iSCSI driver has some storage limitations related to the maximum number of volumes that can be attached at the same time. It only allows the creation of 32 targets per EVS with 32 LUs on each (1024 volumes attached to instances). Hitachi Data Systems recommends the HNAS NFS driver over the HNAS iSCSI driver as it does not have that limitation and can take benefit of the HNAS NFS optimizations.

Chapter 6. Known issues

Extend volume limits

Phenomenon: If a user has enough quota, it can extends the volume to a size that is bigger than size of the pool where the volume is placed.

Cause: HNAS driver only supports thin provisioning and therefore, the volumes allocated does not consume space on the storage until data is written to it.

Workaround: Make sure that the sum of tenant's volume_type quotas associated with the pool does is not bigger than what is provisioned by that pool. More details of how to set quotas in Cinder, please refer to *"Openstack Operation and Administration Guides"*.

iSCSI Driver fails when booting multiple instances (booting from volumes creating an image)

Phenomenon: If a user creates several instances at the same time (booting from volumes creating an image), some operations might fail.

Cause: HNAS iSCSI target's limitations. As the target needs to be shared among hosts, the process of adding a LU on a host (compute or controller) might left some dangling device entries on the other hosts.

Workaround: Prefer NFS driver if possible. Otherwise retry the failed operations.

Creating a volume bigger than source

Phenomenon: When creating a volume from a snapshot or other volume, if the created volume is bigger than the source, the request is not completed.

Cause: Cinder API allows this operation but the driver does not support it.

Workaround: If you need the volume to be bigger than the source, create the volume with the same size as the source, than extend it to the desired size.

Chapter 7. General questions

Is there any specification where metadata, such as ldev and copy_method, is kept in volume and snapshot?

No, HNAS uses the volume id provided by Cinder and creates the LUs in the storage using this id. So, for iSCSI, volume '317eb9e9-8912-4039-a601-5747e1ec26b5' can be found in the storage by:

```
# 'ssc localhost console-context --evs 2 iscsi-lu list volume-317eb9e9-8912-4039-a601-5747e1ec26b5'

Name: volume-317eb9e9-8912-4039-a601-5747e1ec26b5
Comment:
Path: /.cinder/volume-317eb9e9-8912-4039-a601-5747e1ec26b5.iscsi
Size: 1 GB
File System: easy-stack
File System Mounted: Yes
Logical Unit Mounted: Yes
```

For NFS the same applies, but the volumes files are created inside the NFS export. Snapshots will have the same id of the snapshot id in Cinder, but the prefix in the storage will be 'snapshot-'

Does the HNAS driver support OpenStack functions (live migration, multipath, FC zoning manager, HA environment)?

Yes. For those operations, HNAS driver has the same capabilities supported by Cinder (Liberty). High Availability (HA), for example, is only supported in Active/Passive mode.

If HNAS driver supports live migration, what settings do I need to use?

Yes. Live migration is a Nova feature and does not relies on any feature provided by the Cinder driver. Refer to Nova user guide: http://docs.openstack.org/admin-guide/compute-configuring-migrations.html

In HA cluster configuration, what settings do I need to use for the HNAS driver?

No particular settings are required for the HNAS driver for HA configuration. As the volume service for Cinder will run in A/P, the passive node running HNAS driver needs to have the same configuration and access keys to HNAS that are needed in the Active controller.

Does the HNAS driver support NAS OS12.6 (or later)?

Yes. HNAS Driver supports OS version 12.2 and higher.

What is the server prerequisite information, such as memory, HDD, and capacity?

The hardware requirements must be defined by the OpenStack distro. Refer to the documentation for the OpenStack distro in use.

What is the specification of log generation management?

The log levels are controlled by the default Cinder logging mechanism and details about this can be found in the official OpenStack documentation. Additionally, you can enable trace functions with detailed driver messages setting hnas_enable_trace to True in the [DEFAULT] section of cinder.conf file.

Hitachi Data Systems

Corporate Headquarters
2845 Lafayette Street
Santa Clara, California 95050-2639
U.S.A.
www.hds.com

Regional Contact Information

Americas
+1 408 970 1000
info@hds.com

Europe, Middle East, and Africa
+44 (0) 1753 618000
info.emea@hds.com

Asia Pacific
+852 3189 7900
hds.marketing.apac@hds.com

Hitachi Data Systems

MK-92ADPTR124-00