

=====

README TO INSTALL/CONFIGURE CINDER HNAS DRIVER

=====

This OpenStack Block Storage volume driver provides iSCSI and NFS support for Hitachi NAS Platform Models 3080, 3090, 4040, 4060, 4080 and 4100.

Supported Operations

=====

The NFS and iSCSI drivers support these operations:

- * Create, delete, attach, and detach volumes.
- * Create, list, and delete volume snapshots.
- * Create a volume from a snapshot.
- * Copy an image to a volume.
- * Copy a volume to an image.
- * Clone a volume.
- * Extend a volume.
- * Get volume statistics.
- * Manage and unmanage a volume.

Service labels

The HNAS driver supports differentiated types of service using the service labels. It is possible to create up to four types of them, for example, gold, platinum, silver and ssd.

Host Requirements

=====

This package is supported in SUSE OpenStack Cloud and RedHat OpenStack Platform (RHOSP) environments:

1.4.0_2015.1 (Kilo)

- RHOSP 7

1.4.0_2014.2 (Juno)

MK-92ADPTR093-01

- RHOSP 6

- SUSE OpenStack Cloud 5

Before installing this package, you must have an OpenStack environment. Please Refer to SUSE and RHOSP documentation to see how to configure OpenStack and Cinder services.

The following packages must be installed in the Cinder host and compute nodes:

- * nfs-utils for RH

- * nfs-client for SUSE

- * if you are not using SSH communication (see how to configure in SSH Configuration) install the HNAS SSC command to communicate with an HNAS array. This utility package is available in the physical media distributed with the HNAS hardware or it can be downloaded in HDS support portal.

HNAS Storage Requirements

=====

This driver is supported in HNAS/SMU software version 12.2 or later. Before using iSCSI and NFS services, make sure you check your system version. Then, use the HNAS Configuration and Management GUI or SSC CLI to create storage pool(s), file system(s), and assign an EVS. Make sure that the file system used by Openstack Cinder driver is not created as replication targets. Additionally:

For NFS:

Create NFS exports, choose a path for them (it must be different from "/") and set the 'Show snapshots' option to hide and disable access. Also, in the "Access Configuration" set the option `norootsquash`, for example:

`"* (rw, norootsquash)"`, so HNAS Cinder driver can change the permissions of its volumes.

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. On RedHat, if the system is configured to use SELinux, you need to set

`'virt_use_nfs = on'` for NFS driver works properly.

`::`

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

For iSCSI: You need to set an iSCSI domain.

Package installation

=====

To install the driver follow the steps below:

1 - Install the OS dependencies on Compute and Cinder volume nodes:

In RHOSP:

```
::  
  
# yum install nfs-utils nfs-utils-lib
```

Or in SUSE:

```
::  
  
# zypper install nfs-client
```

2 - If Cinder volume services are running in the host, stop all Cinder services (api, volume, scheduler and backup).

3 - Uninstall any previous versions of the package:

In RHOSP::

```
::  
  
# rpm -e openstack-cinder-hnas
```

Or in SUSE:

```
::  
  
# zypper remove openstack-cinder-hnas
```

4 - Install HNAS Cinder driver package:

In RHOSP 7:

```
::  
  
# rpm -i openstack-cinder-hnas-1.4.0_2015.1-1.x86_64.rpm
```

In RHOSP 6:

```
::
```

```
# rpm -i openstack-cinder-hnas-1.4.0_2014.2-1.x86_64.rpm
```

Or in Suse Cloud 5:

```
# zypper install openstack-cinder-hnas-1.4.0_2014.2-1.x86_64.rpm
```

5 - Configure the driver as described in the 'Driver Configuration' section

6 - Restart all Cinder services (api, volume, scheduler and backup).

Driver configuration

=====

The HDS driver supports the concept of differentiated services (also referred to as quality of service) by mapping volume types to services provided through HNAS. HNAS supports a variety of storage options and file system capabilities, which are selected through the definition of volume types and the use of multiple back ends. The driver maps up to four volume types into separated exports or file systems. If you want to support more than 4 type of services, you need to configure multiple backends.

The configuration for the driver is read from an XML-formatted file (one per back end), which you need to create and set its path in the `cinder.conf` configuration file.

Below are the configurations needed in the `cinder.conf` configuration file:

```
::
```

```
[DEFAULT]
```

```
enabled_backends = hnas-iscsil, hnas-nfs1
```

For HNAS iSCSI driver create this section:

```
::
```

```
[hnas-iscsil]
```

```
volume_driver=cinder.volume.drivers.hitachi.enterprise.hnas_iscsi.HDSISCSIDriver
```

```
hds_hnas_iscsi_config_file=/path/to/config/hnas_config_file.xml
```

```
volume_backend_name=HNAS-ISCSI
```

For HNAS NFS driver create this section:

```
::
```

```
[hnas-nfs1]
```

```
volume_driver=cinder.volume.drivers.hitachi.enterprise.hnas_nfs.HDSNFSDriver
```

```
hds_hnas_nfs_config_file=/path/to/config/hnas_config_file.xml
```

```
volume_backend_name=HNAS-NFS
```

The XML file has the following format:

```
::
```

```
<?xml version="1.0" encoding="UTF-8" ?>

<config>

  <mgmt_ip0>172.24.44.15</mgmt_ip0>

  <hnas_cmd>ssc</hnas_cmd>

  <chap_enabled>False</chap_enabled>

  <ssh_enabled>False</ssh_enabled>

  <cluster_admin_ip0>10.1.1.1</cluster_admin_ip0>

  <username>supervisor</username>

  <password>supervisor</password>

  <svc_0>

    <volume_type>default</volume_type>

    <iscsi_ip>172.24.44.20</iscsi_ip>

    <hdp>fs01-husvm</hdp>

  </svc_0>

  <svc_1>

    <volume_type>platinum</volume_type>

    <iscsi_ip>172.24.44.20</iscsi_ip>

    <hdp>fs01-platinum</hdp>

  </svc_1>

</config>
```

HNAS volume driver XML configuration options

An Openstack Block Storage node using HNAS drivers can have up to 4 services. Each service is defined by a `svc_n` tag, `svc_0`, `svc_1`, `svc_2` or `svc_3`, for example. These are the configuration options available for each service label: `volume_type`: (required, default='default') When a create volume call with a certain volume type happens, the volume type will try to be matched up with this tag. In each configuration file you must define the default volume type in the service labels and, if no volume type is specified, the default is used. Other labels are case sensitive and should match exactly. If no configured volume types match the incoming requested type, an error occurs in the volume creation.

`hdp`: (required)

For iSCSI driver: virtual file system label associated with the service.

For NFS driver: path to the volume (`<ip_address>:/<path>`) associated with the service.

Additionally, this entry must be added in the file used to list available NFS shares. This file is located, by default, in `/etc/cinder/nfs_shares` or you can specify the location in the `nfs_shares_config` option in the `cinder.conf` configuration file.

`iscsi_ip`: (required for iSCSI only)

An iSCSI IP address dedicated to the service

These are the configuration options available to the 'config' section of the XML config file:

`mgmt_ip0`: (required)

Management Port 0 IP address. Should be the IP address of the 'Admin' EVS.

`hnas_cmd`: (optional, default='ssc')

`hnas_cmd` is a command to communicate to HNAS array.

`chap_enabled`: (optional, for iSCSI only, default='True')

`chap_enabled` is a boolean tag used to enable CHAP authentication protocol.

`cluster_admin_ip0`: (optional when `ssh_enable` is True)

The address of HNAS cluster admin.

`username`: (required)

Username is always required on HNAS.

`password`: (required if `ssh_enabled`='False')

Password on HNAS.

svc_0, svc_1, svc_2, svc_3: (at least 1 label must be defined)

Service labels: these 4 predefined names help four different sets of configuration options. Each can specify HDP and a unique volume type.

ssh_enabled: (optional, default='False')

Enables ssh authentication between Cinder host and the SMU

ssh_private_key: (required if ssh_enabled='True')

Path to the SSH private key used to authenticate in HNAS SMU. The public key must be uploaded to HNAS SMU using ssh-register-public-key (this is an SSH subcommand). Note that copying the public key HNAS using ssh-copy-id doesn't work properly as the SMU periodically wipe out those keys.

Service labels

~~~~~

HNAS driver supports differentiated types of service using the service labels. It is possible to create up to four types of them, as gold, platinum, silver and ssd, for example.

After creating the services in the XML configuration file, you must configure one volume\_type per service. Each volume\_type must have the metadata service\_label with the same name configured in the <volume\_type> section of that service. If this is not set, OpenStack Block Storage will schedule the volume creation to the pool with largest available free space or other criteria configured in volume filters.

::

```
$ cinder type-create 'platinum-tier'

$ cinder type-key 'platinum' set service_label='platinum'

$ cinder type-create 'default'

$ cinder type-key 'default' set service_label='default'
```

## Multi-backend configuration

-----

If you use multiple backends and intend to enable the creation of a volume in a specific backend, you must configure volume types to set the volume\_backend\_name option to the appropriate back end. Then, create volume\_type configurations with the same volume\_backend\_name:

::

```
$ cinder type-create 'iscsi'

$ cinder type-key 'iscsi' set volume_backend_name='HNAS-ISCSI'

$ cinder type-create nfs
```

```
$ cinder type-key 'nfs' set volume_backend_name='HNAS-NFS'
```

You can deploy multiple OpenStack HNAS drivers instances that each control a separate HNAS array. Each service (svc\_0, svc\_1, svc\_2, svc\_3) on the instances need to have a volume\_type and service\_label metadata associated with it. If no metadata is associated with a pool, OpenStack Block Storage filtering algorithm selects the pool with the largest available free space.

## SSH configuration

-----

Instead of using SSC on the Block Storage host and store its credential on the XML configuration file, HNAS driver supports SSH authentication. To configure that:

1 - If you don't have a pair of public keys already generated, create them in the Block Storage host (leave the passphrase empty):

```
::
```

```
$ mkdir -p /opt/hds/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):

```
::
```

```
# chown -R cinder.cinder /opt/hds/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/hds/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 Block Storage host:

::

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

<cluster\_admin\_ip0> is 'localhost' for single node deployments. This should return a list of available file systems on HNAS.

\* Editing the XML config file:

- set the 'username'

- enable SSH adding the line "<ssh\_enabled>True</ssh\_enabled>" under 'config' section

- set the private key path:

```
"<ssh_private_key>/opt/hds/ssh/hnaskey</ssh_private_key>" under
```

'config' section

- if the HNAS is in a multi-cluster configuration set <cluster\_admin\_ip0>

to the cluster node admin IP. In a single node HNAS, leave it empty.

\* restart Cinder services

Note that copying the public key HNAS using ssh-copy-id doesn't work properly as the SMU periodically wipe out those keys.

Manage and unmanage extensions

-----

Manage and unmanage are two new API extensions and add some new features to the driver. The manage action on an existing volume is very similar to a volume creation. It creates a volume entry on Cinder DB but instead of create a new volume in the backend, it only adds a 'link' to an existing volume. Volume name, description, volume\_type, metadata and availability\_zone are supported as in a normal volume creation.

The unmanage action on an existing volume removes the volume from Cinder DB but keep the actual volume in the backend. For all intents from a Cinder perspective the volume would be deleted, but the volume would still exist outside of OpenStack/Cinder.

How to manage:

MK-92ADPTR093-01

~~~~~

On Dashboard:

+++++

For NFS:

1. Under the tab System -> Volumes choose the option

[+ Manage Volume]

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

a) **Identifier:** ip:/type/volume_name

Example: 172.24.44.34:/silver/volume-test

b) **Host:** host@backend-name#pool_name

Example: c-node@hnas-nfs#test_silver

c) **Volume Name:** volume_name Example: volume-test

d) **Volume Type:** choose a type of volume Example: silver

For iSCSI:

1. Under the tab System -> Volumes choose the option

[+ Manage Volume]

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

a) **Identifier:** filesystem-name/volume-name

Example: filesystem-test/volume-test

b) **Host:** c-node@backend-name#pool_name

Example: root@hnas-iscsi#test_silver

c) **Volume Name:** volume_name Example: volume-test

d) **Volume Type:** choose a type of volume Example: silver

On SUSE Cloud, the manage and unmanage functions are only available through CLI.

By CLI:

+++++

This is the command to manage a volume:

::

```
$ cinder --os-volume-api-version 2 manage [--id-type <id-type>] \  
[--name <name>][--description <description>][--volume-type <volume-type>] \  
[--availability-zone <availability-zone>][--metadata [<key=value> \  
[<key=value> ...]]] [--bootable]<host> <identifier>
```

Example:

For NFS:

::

```
$ cinder --os-volume-api-version 2 manage --name <volume_test> \  
--volume-type <silver> \  
--source-name <evs_ip:/export_name/volume_test> \  
<c-node@hnas-nfs#test_silver>
```

For iSCSI:

::

```
$ cinder --os-volume-api-version 2 manage --name <volume-test> \  
--volume-type <silver> \  
--source-name <filesystem-test/volume-test> \  
<c-node@hnas-iscsi#test_silver>
```

How to unmanage:

~~~~~

On Dashboard:

+++++

1. Under the tab [ System -> Volumes ] choose a volume
2. On the volume options, choose \*[ +Unmanage Volume ]\*
3. Check the data and Confirm.

By CLI:

+++++

This is the command to unmanage a volume:

::

```
$ cinder --os-volume-api-version 2 unmanage <volume>
```

Example:

::

```
$ cinder --os-volume-api-version 2 unmanage <volume-test>
```

\* It is not possible to manage a volume if there is a slash ( '/' ) on the volume name.

Additional Notes

-----

\* After changing the configuration on the storage, all Cinder services must be restarted!

\* In HNAS iSCSI driver, allows a total of 1024 attached volumes per EVS.

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

## Package Upgrading

-----

RPM Upgrading (`rpm -U`) of packages is not supported. If you need to upgrade, first remove the older version and then install the new one. Note also that if you are using a previous version of this driver or the upstream version (that comes with your distro), the `volume_driver` path on `cinder.conf` has changed.

