Hedvig

Hedvig Cinder Driver for OpenStack

User Guide

Table of Contents

Hedvig/OpenStack version support matrixes

Here are version support matrixes for the Hedvig Cinder Driver, with respect to Hedvig software releases and OpenStack versions.

Note: Hedvig has code present in the OpenStack repository starting at the Stein release.

Table 1: Hedvig software release version support matrix

Hedvig software release	Supported versions of Hedvig Cinder Driver
2.x	1.0
3.x	1.0

Table 2: OpenStack version support matrix

Module version	OpenStack version codename	Supported versions of Hedvig Cinder Driver
14.z.y	Stein	1.0
15.z.y	Train	1.0
16.z.y	Ussuri	1.0

Changes since last user guide edition

Here are the changes since the last edition of this user guide.

Table 3: User guide changes since last edition

date	change
031120	page 3: updated *Hedvig/OpenStack version support matrixes.*

Hedvig Cinder Driver deployment environment

Organizations rely on a wide range of applications and need a full set of storage capabilities within their OpenStack deployment.

The Hedvig Distributed Storage Platform is an all-software solution that provides:

- ***Unprecedented simplicity and flexibility:*** Block storage protocol is supported via Cinder, and object storage is supported via Swift.

  Note: For more information on the latter, see the *Hedvig Object Storage with OpenStack Swift User Guide.*

- ***A complete, robust set of enterprise storage features:*** Caching, compression, deduplication, snapshotting, and cloning are included.

- ***Programmability that is fully accessible via REST APIs:*** Every aspect of the platform is API accessible, making it ideal for service catalogs.

Hedvig Cinder Driver overview

With the *Hedvig Cinder Driver for OpenStack*, you can:

- ***Integrate public and private clouds***: Build a unified hybrid environment to easily migrate to or from your data center and public clouds.

- ***Set granular virtual disk policies***: Assign enterprise-class features on a per volume basis to best fit your application requirements.

- ***Connect to any compute environment:*** Use with any hypervisor, application, or bare-metal system.

- ***Grow seamlessly with an elastic cluster***: Scale storage performance and capacity on-the-fly with off-the-shelf x86 servers.

- ***Deliver predictable performance***: Receive consistent high-IOPS performance for demanding applications through massive parallelism, dedicated flash, and edge cache configurations.

Hedvig Cinder Driver for OpenStack notes

Here are a few notes when using the Hedvig Cinder Driver for OpenStack:

- Hedvig supports *background delete.* Therefore, a snapshot cannot be deleted from within OpenStack unless the clones of the snapshot have been completely deleted from within Hedvig.

  Use either the Hedvig WebUI or the Hedvig CLI to confirm that the clone has been deleted.

- The resize operation is not allowed on Cinder Volumes created from snapshots.

  A Cinder Volume created from a snapshot should have the same size as that of the base volume.

- Hedvig does not support clone of clones.

- For an image backed by Hedvig, ensure that the image is immutable. No operation should be performed on the base volume backed by the image.

- Hedvig does not take care of quiescing during clone or snapshot.

Installing and configuring the Hedvig Cinder Driver

Installing the Hedvig Cinder Driver

Note: If you are working on OpenStack **Stein** or above, then skip to the next section, *Configuring the Hedvig Cinder Driver on an OpenStack Cinder Node.*

Install the Hedvig Cinder Drive on all OpenStack Compute Host and Controller Nodes that run the Cinder Volume service.

1. Download to the OpenStack Cinder Node:

   ```
   cinder.tar
   ```

2. Untar the file. You should see a directory named:

   ```
   hedvig
   ```

3. Copy the `hedvig` directory to the location of the Cinder directory. For example, on a CentOS machine setup, the location would be:

   ```
   /usr/lib/python2.7/site-packages/cinder/volume/drivers/hedvig
   ```

Configuring the Hedvig Cinder Driver on an OpenStack Cinder Node

1. Choose a backend name, preferably `hedvig-<cluster_name>,` for example, `hedvig-devtest.`

2. Run the following commands on the OpenStack Cinder Node to create a Volume Type for Hedvig:

   ```
   cinder type-create hedvig-devtest
   cinder type-key hedvig-devtest set volume_backend_name=hedvig-devtest
   ```

   Note: See *Creating a Hedvig Cinder Volume with custom attributes (QoS Specs).*

3. Update the `/etc/cinder/cinder.conf` file with this configuration:

```
[DEFAULT]
enabled_backends=hedvig-devtest

[hedvig-devtest]
volume_backend_name=hedvig-devtest
volume_driver=cinder.volume.drivers.hedvig.hedvig_cinder.HedvigISCSIDriver
san_ip=<Comma-separated list of cluster nodes hostname/ip>
san_login=<Username to login to the hedvig cluster UI>
san_password=<Password to login to the hedvig cluster UI>
san_clustername=<Name of the hedvig cluster>cluster>
```

4. **Note**: If you are working on OpenStack **Stein** or above, then skip to step 5.

   Update the `exception.py` file, located in the Cinder directory, with the `#Hedvig Driver` section.

   For example, on a CentOS machine setup, the location of this file would be:

   ```
   /usr/lib/python2.7/site-packages/cinder/exception.py
   ```

```
#Hedvig Driver
class HedvigDriverException(CinderException):
   message = _("Hedvig Cinder driver error" )
class HedvigVolumeException(VolumeDriverException):
   message = _("Hedvig Cinder volume driver error ")
```

5. Add the entries for your Hedvig Storage Cluster Nodes and Hedvig Storage Proxy to:

   ```
   /etc/hosts
   ```

6. Restart the `cinder-volume` service to apply the changes and to initialize the Hedvig Cinder Driver.

When you create a new Cinder Volume using *Horizon* (the OpenStack Dashboard), you should see the `hedvig` option in the `Type` dropdown box.

Configuring the Hedvig Cinder Driver on a Hedvig Storage Cluster

1. Make a list of all Hedvig Storage Proxies running on each OpenStack Compute Host.

2. Run the following command at the Hedvig CLI, one for each OpenStack Compute Host:

   ```
   registertgt -h <OpenStack Compute Hostname>
      -t <Hedvig Storage Proxy hostname>
   ```

   Note: For the `<Hedvig Storage Proxy hostname>`, use the *exact* hostname/IP address that is displayed by the Hedvig CLI command, `showallcontrollers`.

3. Run the following command to register the IQN, one for each of the OpenStack Controller Node and the compute nodes:

   ```
   registeriqn -h <OpenStack Controller Node hostname>
      -i <OpenStack Controller Node IQN>
   ```

Provisioning storage with the Cinder Driver

1. Log into *Horizon* (the OpenStack Dashboard) to create a Cinder Volume with suitable policies and size. This information is passed to the Hedvig Cinder Driver. The driver instructs the Hedvig Storage Cluster to provision a virtual disk based on those policies.
2. The driver presents the virtual disk as a Cinder Volume. Each Cinder Volume is backed by a Hedvig virtual disk — one volume to one disk.
3. When you take a snapshot of a Cinder Volume, you are taking a snapshot of the Hedvig virtual disk. You can then clone a virtual disk from one of the snapshots.

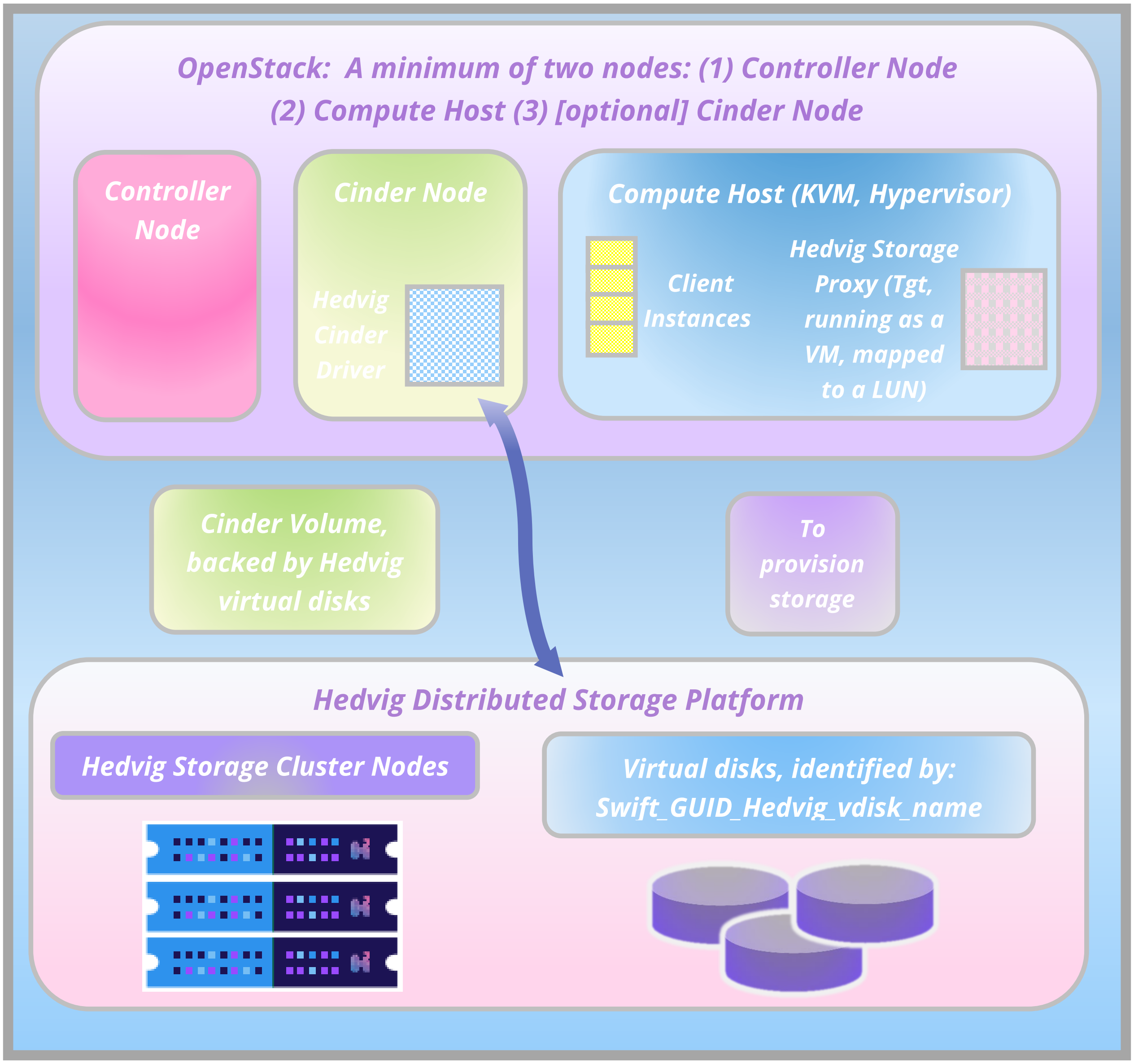

OpenStack: A minimum of two nodes: (1) Controller Node (2) Compute Host (3) [optional] Cinder Node
Controller Node
Cinder Node
Hedvig Cinder Driver
Compute Host (KVM, Hypervisor)
Client Instances
Hedvig Storage Proxy (Tgt, running as a VM, mapped to a LUN)
Cinder Volume, backed by Hedvig virtual disks
To provision storage
Hedvig Distributed Storage Platform
Hedvig Storage Cluster Nodes
Virtual disks, identified by: Swift_GUID_Hedvig_vdisk_name

Figure 1: Provisioning storage with the Hedvig Cinder Driver

Presenting a Hedvig Cinder Volume to an OpenStack Compute Instance

1. You may either:

   a. Attach the Hedvig Cinder Volume to an instance, ***OR***
   b. Directly create a new instance using the provisioned Hedvig Cinder Volume.

2. The Hedvig Cinder Volume is added as a LUN on the iSCSI target that is residing on the same OpenStack Compute Host as that of the OpenStack Compute Instance.

3. The IQN of the OpenStack Compute Host is added to the ACL of the LUN on the iSCSI target.

   The Hedvig Cinder Volume appears on the OpenStack Compute Host when it does an iSCSI login.

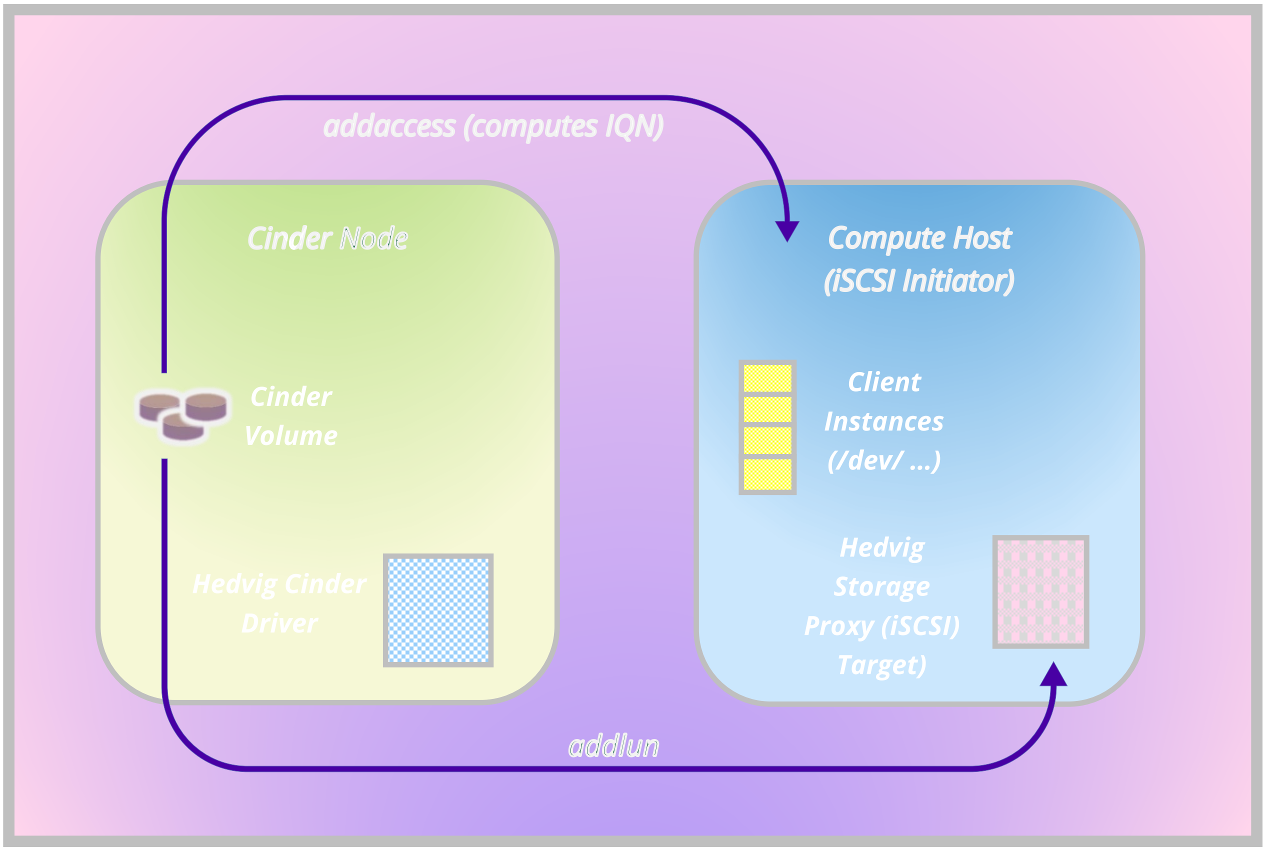
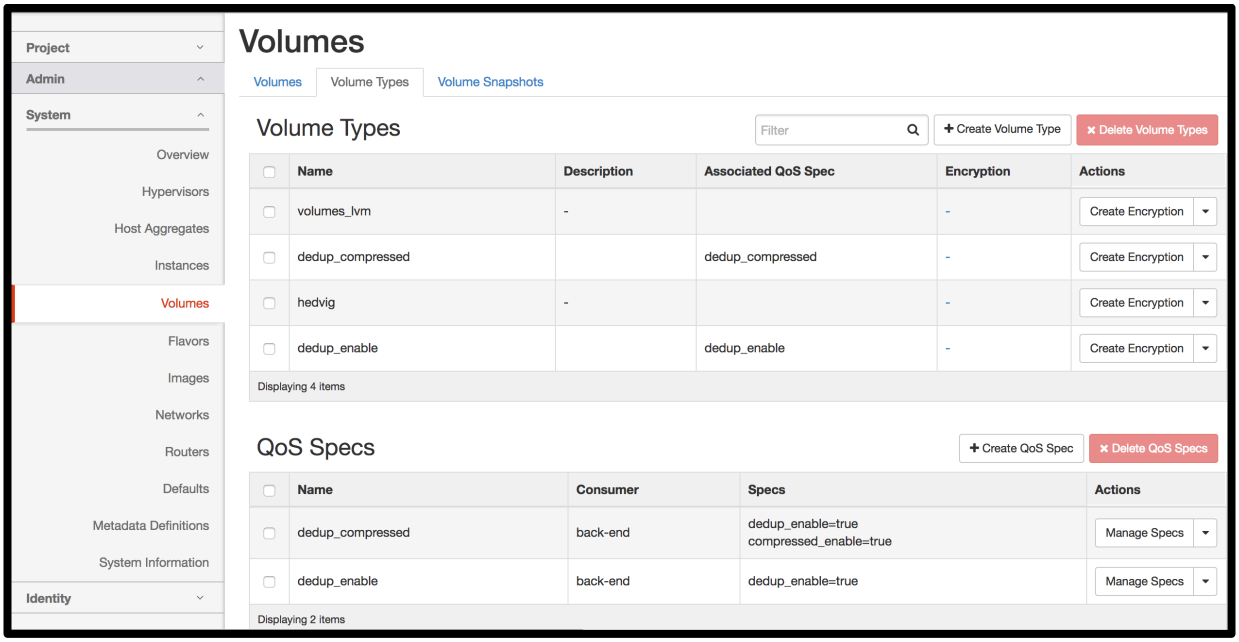

addaccess (computes IQN)
Cinder Node
Cinder Volume
Hedvig Cinder Driver
Compute Host (iSCSI Initiator)
Client Instances (/dev/ ...)
Hedvig Storage Proxy (iSCSI) Target)
addlun

Figure 2: Presenting a Hedvig Cinder Volume to an OpenStack Compute Instance

Creating a Hedvig Cinder Volume with custom attributes (QoS Specs)



Volume Types

☐	Name	Description	Associated QoS Spec	Encryption	Actions
☐	volumes_lvm	-		-	Create Encryption ▾
☐	dedup_compressed		dedup_compressed	-	Create Encryption ▾
☐	hedvig	-		-	Create Encryption ▾
☐	dedup_enable		dedup_enable	-	Create Encryption ▾

Displaying 4 items

QoS Specs

☐	Name	Consumer	Specs	Actions

☐	dedup_compressed	back-end	dedup_enable=true
			compressed_enable=true
				Manage Specs ▾

☐	dedup_enable	back-end	dedup_enable=true	Manage Specs ▾

Displaying 2 items

Figure 3: Volume Types and QoS Specs dialog

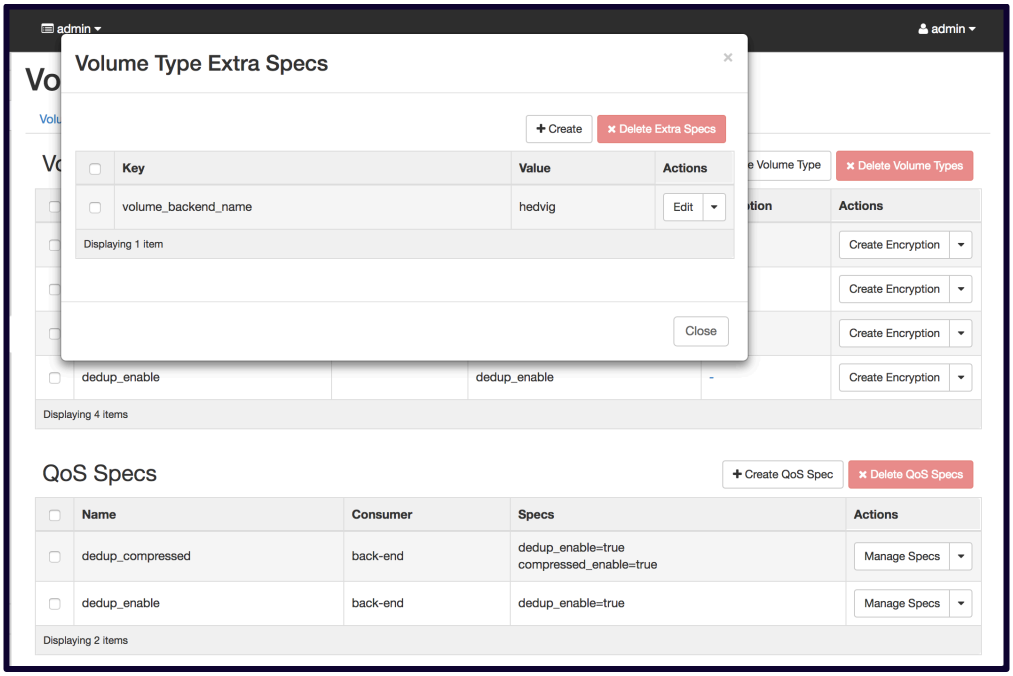
In the figure above, notice the default Volume Type, `hedvig`, which was created when you configured the Hedvig Cinder Driver.

Note: See *Configuring the Hedvig Cinder Driver on an OpenStack Cinder Node.*

1. Create a QoS Spec with the list of attributes that you want to associate with a virtual disk.
2. Create a new volume type and associate this QoS Spec with that volume type, ***OR*** associate the QoS Spec with an existing Hedvig volume type.

   For example, to create a Cinder Volume with deduplication enabled, create a QoS Spec called `dedup_enable` with `dedup_enable=true` in the QoS Specs (as shown above).
3. Associate this QoS Spec with the `hedvig` volume type. Thus, every Cinder Volume that you create of type `hedvig` will have deduplication enabled.
4. You can also create a new volume type (for example `dedup_enable`) with this QoS Spec so that only volumes of this type will have deduplication enabled.

5. If you do create a new volume type, make sure to add the key `volume_backend_name` with value `hedvig` in the **Volume Type Extra Specs** dialog (as shown below) so that OpenStack knows that the Hedvig Cinder Driver handles all requests for this volume.



Volume Type Extra Specs

□	Key	Value	Actions
□	volume_backend_name	hedvig	Edit ▾

Displaying 1 item

□			dedup_enable	ption	Actions
□					Create Encryption ▾
□					Create Encryption ▾
□					Create Encryption ▾
□	dedup_enable		dedup_enable	-	Create Encryption ▾

Displaying 4 items

QoS Specs

□	Name	Consumer	Specs	Actions

□	dedup_compressed	back-end	dedup_enable=true
			compressed_enable=true
				Manage Specs ▾

□	dedup_enable	back-end	dedup_enable=true	Manage Specs ▾

Displaying 2 items

Figure 4: Volume Type Extra Specs dialog

Hedvig QoS Spec parameters and values

- **dedup_enable** – true/false
- **compressed_enable** – true/false
- **cache_enable** – true/false
- **replication_factor** – 1-6
- **replication_policy** – Agnostic/RackAware/DataCenterAware
- **replication_policy_info** – comma-separated list of data center names (applies only to a replication_policy of DataCenterAware)
- **disk_residence** – Flash/HDD
- **encryption** – true/false

Using Glance images backed by Hedvig backend

1. Modify the following parameters in the Glance config file:

   ```
   stores = <existing> , cinder

   show_multiple_locations = True
   ```

1. Modify the following parameters in the Cinder config file:

   ```
   allowed_direct_url_schemes = cinder

   image_upload_use_cinder_backend = True
   ```

2. Create a volume on Hedvig backend, backed by an image, using the Cinder UI.

3. Create an image using:

   ```
   openstack image create --disk-format raw --container-format bare <ImgName>

   glance location-add <image-uuid> --url cinder://<volume-uuid>
   ```

Glossary

This glossary contains definitions of terms used in this document. See also the Hedvig Storage Glossary.

Table 4: Glossary of terms

term	definition
ACL	An *access control list* is a list of permissions attached to an object.
agnostic	*Agnostic*, with respect to Replication Policy, means the replication procedure is independent of rack position and data center. This may also be referred to as rack unaware.
API	An *application programming interface* is a set of routines, protocols, and tools for building software and applications.
Hedvig Storage Cluster	A *Hedvig Storage Cluster* is an elastic cluster, formed by using any type of commodity server(s).
Hedvig Storage Cluster Node	A *Hedvig Storage Cluster Node* is an individual commodity server running *Hedvig Storage Service* software.
Hedvig Storage Proxy	A *Hedvig Storage Proxy* is a lightweight software component that deploys at the application tier as a virtual machine or Docker container, or on bare metal, to provide storage access to any physical host or virtual machine in the application tier. The storage proxy presents block, file, and object (Amazon S3) storage access to app hosts, accelerates read performance with flash caching, drives efficiency with deduplication, and secures data with encryption. *This may also be referred to as an HSP, controller, CVM, target, or tgt.*

term	definition
IOPS	*Input/output operations per second* is a common performance measurement used to benchmark computer storage devices.

IQN	An *iSCSI qualified name* is the most commonly used format for assigning iSCSI names to nodes (targets and initiators) in an iSCSI network. All IQNs follow this pattern:
	`iqn.yyyy-mm.reversed_domain_name:storage_target_name`
	`iqn` = literal for iSCSI Qualified Name
	`yyyy-mm` = year and month that the *naming authority* took ownership of the domain name
	`reversed_domain_name` = reversed domain name of the *naming authority*
	`storage_target_name` = optional string to uniquely identify each IQN under the same domain
	For example:
	`iqn.1991-05.com.microsoft:hyperv-1.corp.hedviginc.com`

iSCSI	*Internet small computer system interface* is an IP-based storage networking standard for linking data storage facilities.
LUN	A *logical unit number* is a number that identifies a logical unit, which is a device addressed by the SCSI protocol or SAN protocols, which encapsulate SCSI, such as Fibre Channel or iSCSI.
OpenStack Cinder Node	An *OpenStack Cinder Node* is a node where your Cinder block storage service is running.
OpenStack Compute Host	An *OpenStack Compute Host* is a node where your Nova services are running.

term	definition
QoS	*Quality of Service* is defined as the ability to guarantee certain network requirements, such as bandwidth, latency, jitter, and reliability, to satisfy a Service Level Agreement (SLA) between an application provider and end users.
QoS Spec	*QoS specifications* apply generic QoS support for volumes. They can be enforced at either the hypervisor (front-end) or the storage subsystem (back-end) or both. QoS specifications are added as standalone objects that can then be associated with Cinder Volume Types.
REST	*Representational state transfer* is the software architectural style of the World Wide Web.
virtual disk	A *virtual disk* is an abstracted logical disk volume presented to a computer or application for read/write use.

Hedvig Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. The information in this publication is provided as is. Hedvig Inc. makes no representations or warranties of any kind with respect to the information in this publication and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any Hedvig Inc. software described in this publication requires an applicable software license. All trademarks are the property of their respective owners. Revision date: 031120.

Software-defined AES-256, FIPS compliant encryption of data in flight and at rest.