

HITACHI

Hitachi Virtual Storage Platform Best Practices for OpenStack Environments

Best Practices Guide

MK-SL-444-00

April 2026

Hitachi Vantara

© Hitachi Vantara LLC 2026. All Rights Reserved.

Table of Contents

- Hitachi Block Storage Driver Best Practices for OpenStack..... 5**
 - Introduction5
 - Overview5
 - OpenStack architecture 6
 - OpenStack core services 6
 - Supported VSP storage systems7
 - HBSD installation.....7
 - Set up the Hitachi storage volume driver and volume operations.....7
- Cinder architecture 9**
- HBSD multi-backend 9**
 - Benefits of multi-backends 9
 - Multi-backend and Cinder scheduler 10
- Cinder implementation 11**
 - How Cinder scheduler works:..... 11
- Adaptive QoS..... 12**
- Cinder operations using Global-Active Device 15**
 - Overview and configuration..... 15
 - Creating a global-active device environment16
 - Volume operations in a GAD configuration 17
 - Creating volumes 19
 - Unavailable Cinder functions 19
- General requirements 20**
- Hitachi Virtual Storage Platform system configuration 20**

Set up Hitachi storage.....	20
Configuring Host Mode and Host Mode Options.....	21
Volume management	21
Supported storage protocols.....	22
Fibre Channel.....	22
iSCSI	23
Host configuration	24
HBA and WWN	24
Multipathing	25
NTP configuration	27

Notices and Disclaimer

© 2026 Hitachi Vantara LLC. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or stored in a database or retrieval system for commercial purposes without the express written permission of Hitachi, Ltd., or Hitachi Vantara (collectively, "Hitachi"). Licensee may make copies of the Materials provided that any such copy is: (i) created as an essential step in utilization of the Software as licensed and is used in no other manner; or (ii) used for archival purposes. Licensee may not make any other copies of the Materials. "Materials" mean text, data, photographs, graphics, audio, video and documents.

Hitachi reserves the right to make changes to this Material at any time without notice and assumes no responsibility for its use. The Materials contain the most current information available at the time of publication.

Some of the features described in the Materials might not be currently available. Refer to the most recent product announcement for information about feature and product availability, or contact Hitachi Vantara at https://support.HitachiVantara.com/en_us/contact-us.html.

Notice: Hitachi products and services can be ordered only under the terms and conditions of the applicable Hitachi agreements. The use of Hitachi products is governed by the terms of your agreements with Hitachi Vantara.

By using this software, you agree that you are responsible for:

- 1) Acquiring the relevant consents as may be required under local privacy laws or otherwise from authorized employees and other individuals to access relevant data; and
- 2) Verifying that data continues to be held, retrieved, deleted, or otherwise processed in accordance with relevant laws.

Notice on Export Controls. The technical data and technology inherent in this Document may be subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Reader agrees to comply strictly with all such regulations and acknowledges that Reader has the responsibility to obtain licenses to export, re-export, or import the Document and any Compliant Products.

EXPORT CONTROLS - Licensee will comply fully with all applicable export laws and regulations of the United States and other countries, and Licensee shall not export, or allow the export or re-export of, the Software, API, or Materials in violation of any such laws or regulations. By downloading or using the Software, API, or Materials, Licensee agrees to the foregoing and represents and warrants that Licensee is not located in, under the control of, or a national or resident of any embargoed or restricted country.

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries.

AIX, AS/400e, DB2, Domino, DS6000, DS8000, Enterprise Storage Server, eServer, FICON, Flash Copy, IBM, Lotus, MVS, OS/390, PowerPC, RS6000, S/390, System z9, System z10, Tivoli, z/OS, z9, z10, z13, z/VM, BCPii™ and z/VSE are registered trademarks or trademarks of International Business Machines Corporation.

Active Directory, ActiveX, Bing, Excel, Hyper-V, Internet Explorer, the Internet Explorer logo, Microsoft, the Microsoft Corporate Logo, MS-DOS, Outlook, PowerPoint, SharePoint, Silverlight, SmartScreen, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, the Windows logo, Windows Azure, Windows PowerShell, Windows Server, the Windows start button, and Windows Vista are registered trademarks or trademarks of Microsoft Corporation. Microsoft product screen shots are reprinted with permission from Microsoft Corporation.

All other trademarks, service marks, and company names in this document or web site are properties of their respective owners.

Intended Audience

This document is intended for Hitachi Vantara staff and IT professionals of Hitachi Vantara customers and partners who are responsible for planning and deploying such configurations.

Document Revisions

Revision Number	Date	Details
v1.0	April 2026	Initial release

Comments

Send any comments on this document to Docs-Feedback@hitachivantara.com. Include the document title, including the revision level, and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Hitachi Block Storage Driver Best Practices for OpenStack

Introduction

This document outlines the best practices for configuring the Hitachi Block Storage Driver (HBSD) and Hitachi Virtual Storage Platform (VSP) storage systems within OpenStack environments.

Overview

Hitachi Vantara LLC, a subsidiary of Hitachi Ltd., provides various datacenter infrastructure components to enable IT environments to support the OpenStack ecosystem. This includes mid-range and enterprise storage systems, converged, and hyperconverged infrastructure, as well as a suite of software and software integrations to enable a robust, automated operational environment.

This document outlines most of the best practices to implement an OpenStack environment with VSP storage systems using the HBSD Cinder driver for OpenStack.

These best practices cover the Hitachi products listed in the following tables.

Hardware	Product
Storage Platforms	Hitachi Virtual Storage Platform One Block (all models) Hitachi Virtual Storage Platform 5000 series Hitachi Virtual Storage Platform E series

Software	Product
Hitachi Plugins for OpenStack Ecosystem	Hitachi Block Storage Driver for OpenStack (HBSD)
Hitachi Storage Software	Hitachi Storage Virtualization Operating System RF (SVOS RF)

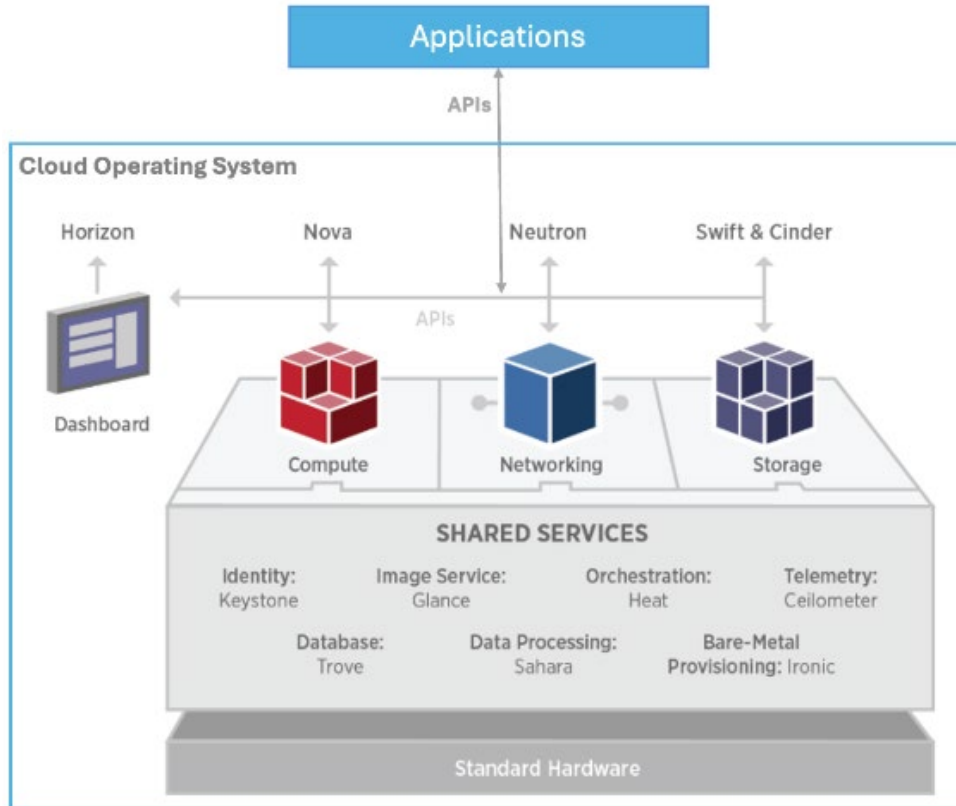
The following are HBSD best practice objectives:

- Accelerate Volume Operations**
 Reduce port mappings per operation by grouping ports into smaller sets assigned to distinct backends to speed volume creation and attachment.
- Maximize Bandwidth Utilization**
 Spread volumes across all ports through multiple backends to maintain full bandwidth and redundancy.
- Maintain Configuration Simplicity**
 Use standard Cinder multi-backend features and HBSD driver options to keep the configuration straightforward without added complexity.
- Preserve Existing Policies**
 Do not alter compute host multipathing policies or zoning. Focus on backend optimization and scheduler-driven load distribution.

OpenStack architecture

OpenStack is an open-source software platform designed for building and operating cloud infrastructure. It is primarily used to deliver Infrastructure-as-a-Service (IaaS) in both private and public cloud environments. It is a cloud computing platform that efficiently manages extensive pools of computing, storage, and networking resources within a data center.

The following illustration presents a high-level overview of OpenStack components.

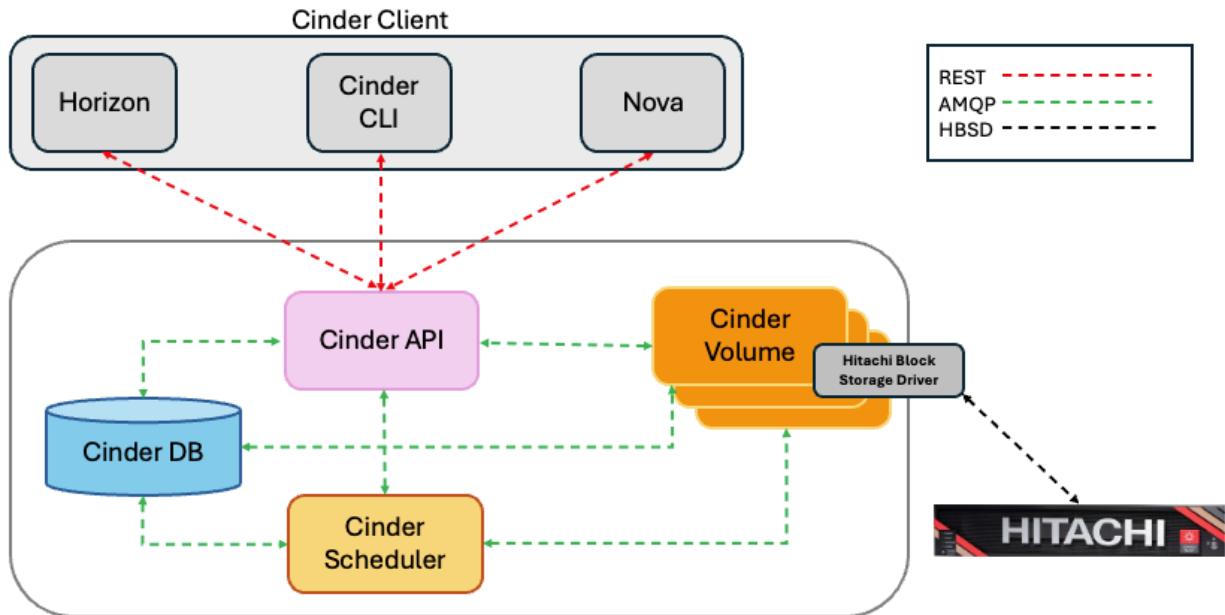


OpenStack core services

The following table highlights the key OpenStack services along with their respective project names.

Services	Project	Description
Compute	Nova	Manage virtual machine instances, scheduling, and lifecycle.
Networking	Neutron	Provide network connectivity as a service, managing virtual interfaces, routers, and firewalls.
Block Storage	Cinder	Manage persistent block-level storage devices for use with compute instances.
Object Storage	Swift	Provide a distributed API-accessible storage system for large amounts of data.
Image	Glance	Stores, discovers, and retrieves virtual machine images.
Dashboard	Horizon	A web-based interface for managing OpenStack services.

Identity	Keystone	Manages user authentication, authorization, and the service catalog.
Bare metal	Ironic	Provisioning physical, bare metal servers.
Database	Trove	Provides database-as-a-service for relational and non-relational engines.
Telemetry	Ceilometer	Provides metering, usage data, and alarming.
Orchestration	Heat	Manages the application lifecycle using templates, supporting infrastructure-as-code.



To ensure optimal performance and reliability, follow the guidelines outlined in the *Block Storage Driver for OpenStack User Guide* on the Product Documentation portal (docs.hitachivantara.com) when configuring compute and controller nodes.

Supported VSP storage systems

For a list of supported Hitachi VSP models that support HBSD, see the *Hitachi Block Storage Driver for OpenStack User Guide* at <https://docs.openstack.org/cinder/2025.2/configuration/block-storage/drivers/hitachi-vsp-driver.html>

HBSD installation

The Hitachi Cinder driver (HBSD) is readily available as a standard component within the Cinder volume service.

Set up the Hitachi storage volume driver and volume operations

Configure the volume driver to be a HBSD by setting the `volume_driver` option in the `cinder.conf` file as follows:

When using Fibre Channel:

```
[hitachi_vsp]
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
san_ip = 1.2.3.4
```

```
san_login = hitachiuser
san_password = password
hitachi_storage_id = 1234566789012
hitachi_pools = pool0
hitachi_target_ports = CL5-A,CL6-A
hitachi_group_create = True
```

When using iSCSI:

```
[hitachi_vsp]
volume_driver = cinder.volume.drivers.hitachi.hbsd_iscsi.HBSDISCSIDriver
san_ip = 1.2.3.4
san_login = hitachiuser
san_password = password
hitachi_storage_id = 1234566789012
hitachi_pools = pool0, pool1
hitachi_target_ports = CL1-E,CL3-E
hitachi_group_create = True
```

Required options:**san_ip**

IP address of SAN controller

- **san_login**
Username for SAN controller
- **san_password**
Password for SAN controller
- **hitachi_storage_id**
Product number of the storage system
- **hitachi_pools**
Pool numbers or pool names of the DP Pool
- **hitachi_target_ports**
hitachi_group_create

For additional parameter information, see the *Hitachi Block Storage Driver for Red Hat OpenStack Services on OpenShift Installation Guide* on the Product Documentation portal (docs.hitachivantara.com).

To ensure the correct installation of HBSD, configuration options operation, and for additional features, review the installation guide specific to your OpenStack deployment or see the OpenStack website at the following link:

<https://docs.openstack.org/cinder/2025.2/configuration/block-storage/drivers/hitachi-vsp-driver.html>

Cinder architecture

Cinder, the block storage provider for OpenStack includes various independent components known as OpenStack services. Cinder operates through three main components: Cinder API, Cinder scheduler, and Cinder volume.

These components communicate using the Advanced Message Queuing Protocol (AMQP). Other OpenStack projects interact with Cinder using REST protocols, and Cinder responds with its own REST protocol.

The Cinder-API service provides a RESTful API for essential volume operations such as expanding volumes and creating, deleting, and taking snapshots. It transmits requests from AMQP to the scheduler service, which verifies the availability of sufficient resources before forwarding the requests to the Volume service.

Each Cinder backend configured in the environment is paired with a corresponding Volume service instance. For a Hitachi backend storage system, the Cinder-Volume instance loads the Hitachi Cinder driver to manage and communicate with the backend storage system.

HBSD multi-backend

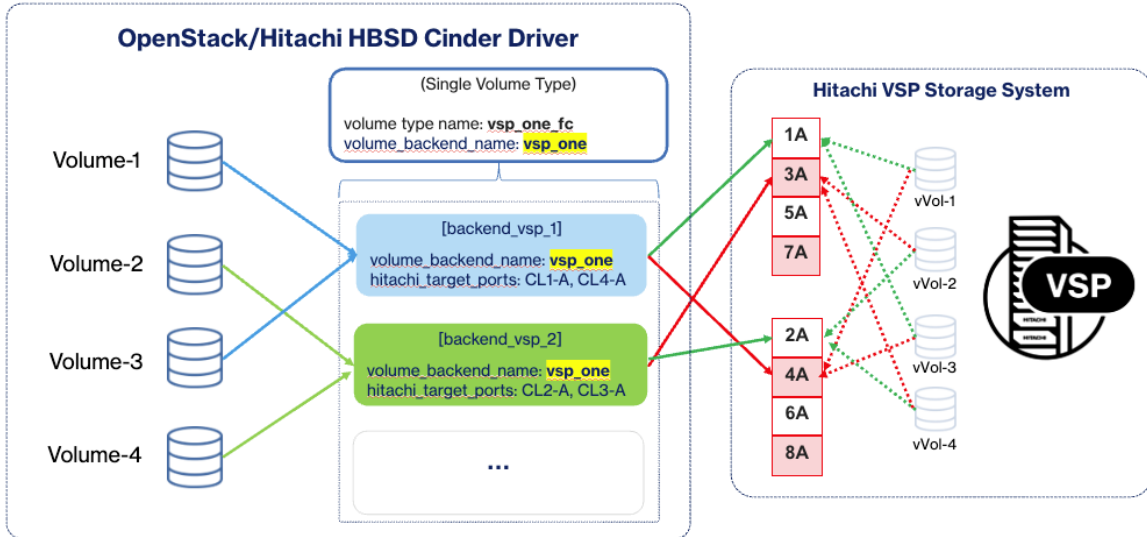
Benefits of multi-backends

- Faster provisioning
Multi-backend port grouping speeds up provisioning by reducing host-group and LU-path operations per volume. Restricting backends to two ports reduces steps and improves responsiveness during provisioning.
- Enhanced performance and scalability
Workloads are distributed across multiple-ports, maintaining high performance and scalability.
- Simplified tenant interaction
A single volume type is exposed to tenants, reducing complexity.

These concepts are shown in the following illustration:

- Multi-backends
 - Different storage ports per backend
 - Creates host group on its own small port set
 - Storage ports are connected to different fabric for HA
- Single volume type
This reflects the Cinder multi-backend pattern where several backends share the same `volume_backend_name` and the cinder scheduler selects one per create action.
- HBSD cinder restricts mapping to the exact ports on the backend using `hitachi_target_port` and `hitachi_compute_target_ports`.
- In combination, load is spread across all ports via multiple small port groups while minimizing per-volume mapping work.

Note: A similar process is supported with Fibre Channel or iSCSI protocols.



The following table shows a configuration with four storage ports.

Component	Configuration
VSP Ports	CL1-A, CL2-A, CL3-A, CL4-A
Backend Groups	Backend_VSP_1: CL1-A/CL4-A Backend_VSP_2, CL2-A/CL3-A
Shared Volume Backend Name	vsp_one
Volume Type	volume_backend_name=vsp_one_fc

The following table shows a configuration with eight storage ports.

Component	Configuration
VSP Ports	CL1-A, CL2-A, CL3-A, CL4-A, CL5-A, CL6-A, CL7-A, CL8-A
Backend Groups	Backend_VSP_1: CL1-A/CL4-A Backend_VSP_2, CL2-A/CL3-A Backend_VSP_3, CL5-A/CL8-A Backend_VSP_4, CL6-A/CL7-A
Shared Volume Backend Name	vsp_one
Volume Type	volume_backend_name=vsp_one_fc

Multi-backend and Cinder scheduler

The following are multi-backend and Cinder scheduler concepts:

- Port group configuration**
 Multiple backends represent sets of port groups on the VSP storage system, dividing ports into manageable sets.
- Backend volume mapping**
 Volumes map only to ports defined for the selected backend, ensuring precise and efficient allocation.

- **Filter mechanism**
Cinder's filter scheduler uses filters such as `AvailabilityZoneFilter` and `CapacityFilter` to select backends.
- **Capacity-based weighing**
`CapacityWeigher` assigns weights based on free capacity to balance workload across storage backends.
- **Optional randomness**
`ChanceWeigher` introduces randomness to avoid predictable scheduling patterns and improve distribution.
- **Efficient provisioning**
The default scheduling logic ensures fast, balanced provisioning without needing custom schedulers. It leverages all physical ports without extra overhead.

Cinder implementation

How Cinder scheduler works:

- The scheduler uses filters and weights to pick the best backend to take care of the request.
- The scheduler uses volume types to explicitly create volumes on specific backends.
Default scheduler: filter scheduler (not round-robin). By default Cinder filters and then weighs candidate backends, but it can be configured differently.
- Stage 1 – Filter: Narrow the candidates using the following default filters:
 - `AvailabilityZoneFilter`
 - `CapacityFilter`
 - `CapabilitiesFilter` (defaults)
- Stage 2 – Weigh: Rank the filtered candidates with weighers:
Default “`CapacityWeigher`” favors backends with more free/virtual capacity; its default behavior is to spread volumes across hosts.
 - Selection: The highest-weighted backend is chosen for the request.
If multiple candidates tie, Cinder uses consistent internal ordering (not round-robin). The defaults are defined in the scheduler's `HostManager`.
 - You can always add or change filters and weighers:
`ChanceWeigher` to randomize ties
Driver filter and goodness weigher for driver-aware placement
 - With identical backends (the same capabilities, similar free capacity) behind one `volume_backend_name`, the default `CapacityWeigher` spreads the load, often alternating `A→B→A→B...`. This creates the appearance of round-robin, but it is a result of weighing, not a round-robin algorithm.
 - If you remove filters but keep weighers, all candidates remain, and the weigher still decides the winner (still not round-robin). Defaults are set in Cinder's scheduler configuration.
 - If you remove weighers, selection becomes deterministic by internal order when weights are equal – again not round-robin. (Cinder still runs the filter scheduler.)
 - To influence distribution style:
 - Keep the default for an even spread across equals.
 - Add `ChanceWeigher` to introduce randomness among equals.
 - Use driver-aware (driver filter and goodness weigher) to system-specific logic.

The following table is an example of a multi-backend Cinder configuration.

Cinder.conf	Volume type and volume
<pre>[DEFAULT] enabled_backends=backend_vsp_1,backend_vsp_2 [backend_vsp_1] volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver volume_backend_name = vsp_one san_ip = <SVP IP> san_login = <user> san_password = <password> hitachi_storage_id = <storage_id> hitachi_pools = 2 hitachi_target_ports = CL1-A, CL4-A hitachi_group_create = True [backend_vsp_2] volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver volume_backend_name = vsp_one san_ip = <SVP IP> san_login = <user> san_password = <password> hitachi_storage_id = <storage_id> hitachi_pools = 2 hitachi_target_ports = CL2-A, CL3-A hitachi_group_create = True</pre>	<pre>#Create volume type openstack volume type create vsp_one_fc openstack volume type set --property volume_backend_name=vsp_one vsp_one_fc #Create a bootable volume openstack volume create --size 4 --type vsp_one_fc --availability-zone nova -- image cirros 1090-boot-vol1</pre>

Adaptive QoS

By configuring Quality of Service (QoS) settings, you can restrict the I/O processing of each volume, thereby maintaining the required performance and quality levels.

In HBSD, you can configure the following settings for each volume. However, you cannot configure these settings for journal volumes.

- Enable Host Mode option (HMO) 122
- This is the following control mechanism that issues SCSI CHECK condition KEY SENSE “TASK FULL” which notifies the host that the storage is busy. This will temporarily notify the host to halt I/O so that the storage does not hold dirty I/O.

Host Mode Option: HMO 122	Task Set Full response after reaching QoS upper limit	Host Mode: 01 [Deprecated] VMware] 21 [VMware Extension] 0C [(Deprecated) Windows] 2C [Windows Extension] 00 [Standard]	Use this HMO when a Windows/Linux/VMware host is connected, and when the QoS upper limit is reached. If you want to return a TASK SET FULL response to the host in order not to retain an I/O inside the storage system. Note: If this option is set for a host other than a Windows/Linux/VMware host, an I/O might not be issued from the host
--	---	--	---

To know more about Host Mode and Host Mode Options, review Open-System Host Attachment Guide for Virtual Storage Platform family on the following link:
<https://docs.hitachivantara.com/r/en-us/svos/a3-04-0x/mk-90rd7037/host-modes-and-host-mode-options/host-modes-and-host-mode-options-for-vsp-one-block-20>

To know more about QoS upper-limit control review Quality of Service user Guide for VSP One Block on the following link:
<https://docs.hitachivantara.com/r/en-us/svos/10.4.x/mk-23vsp1b013>

- Throughput (IOPS, amount of data transferred in MB/s)
Set the upper and lower limits of throughput. If an upper limit is exceeded, I/O is suppressed. If a lower limit is not met, I/O is adjusted so that the lower limit is met.
- Priority level of the I/O processing
Set priority levels for the I/O processing of multiple volumes. I/O is adjusted for faster I/O response, starting with high-priority volumes.
- Adaptive QoS settings are configured the same way that normal QoS settings are configured.
- The currently supported Adaptive QoS setting is `upperIOPSPerGB`.
 - The purpose is to set a QoS value per GB of storage.
 - If the volume is extended, the QoS settings are adjusted on the volume.
- The Adaptive QoS `upperIOPSPerGB` setting can be mixed with normal QoS settings:
 - `UpperTransferRate`, `lowerTransferRate`, and `responsePriority` behave as normal.
 - If `upperIOPS` is configured, `upperIOPS` will be the maximum number of IOPS regardless of the Adaptive calculation.
 - If `lowerIOPS` is configured, `lowerIOPS+1` will be the minimum number of IOPS regardless of the Adaptive calculation.

Setting this property is the same as setting any other QoS property and uses the following procedure:

1. Create a QoS object.

```
Openstack volume qos create -consumer back_end -property
upperIOPSPerGB=1000 upper-iops-gb-1k
```

This creates a QoS object set to 1000 upper IOPS for each GB of volume. The name of the QoS object is `upper-iops-gb-1k`, and it is a backend QoS setting.

2. Create a volume type and associate the QoS object with it.

```
Openstack volume type create qos_voltype
Openstack volume qos associate upper-iops-gb-1k qos_voltype
```

This creates a volume type named `qos_voltype`, and then associates the QoS object named `upper-iops-gk-1k` with it.

3. Use the `type` option when creating a volume.

```
Openstack volume create -size 1 -type qos_voltype my-volume
```

Zoning and host connectivity

The following are recommendations for zoning and host connectivity:

- Assume any compute can access the VM: Nova's scheduler filters and weigh all eligible compute nodes for each request, so a VM can land on any host. Every compute must be able to see the storage ports used by Cinder backends.
- Zone all computes to the exact port groups your backends use because a multi-backend design exposes small port sets per backend (CL1-A/CL4-A, CL2-A/CL3-A). Ensure every compute WWPN is zoned to the ports on each backend so attaches never fail regardless of which backend the scheduler picks behind the shared `volume_backend_name`.
- Dual-fabric, redundant paths: Standard SAN practice – present the same backend ports through at least two fabrics and enable multipath on computes to avoid single-path failures.

Masakari is an OpenStack project that provides Virtual Machine High Availability (VMHA) by automatically detecting and recovering from failures, such as compute node, instance, or process failures.

Why zoning all computes to all backend port groups is crucial:

- Masakari can restart or evacuate VMs to any healthy compute in the segment.
- Masakari detects the failed compute and triggers a Nova evacuate action for its instances. Nova then schedules them on another compute (subject to policy).
- Transparent to HA services: A multi-backend design (several Cinder backends sharing one `volume_backend_name`) is fully transparent to Masakari and other HA automations because Masakari operates at the Nova VM recovery level. The volumes are reattached on the destination host from the Cinder os-brick as long as fabric access exists. Cinder explicitly supports scheduling among multiple backends behind the same name.
- Best practice: dual fabrics + multipath enabled on all computes – when Masakari/Nova relocates a VM, the destination host already has redundant paths to the VSP ports used by that backend.

- Masakari relies on Nova's evacuate scheduler and does not need information about port-grouping or backends. The only requirement is end-to-end reachability from every compute to the storage ports that any backend might use.

Cinder operations using Global-Active Device

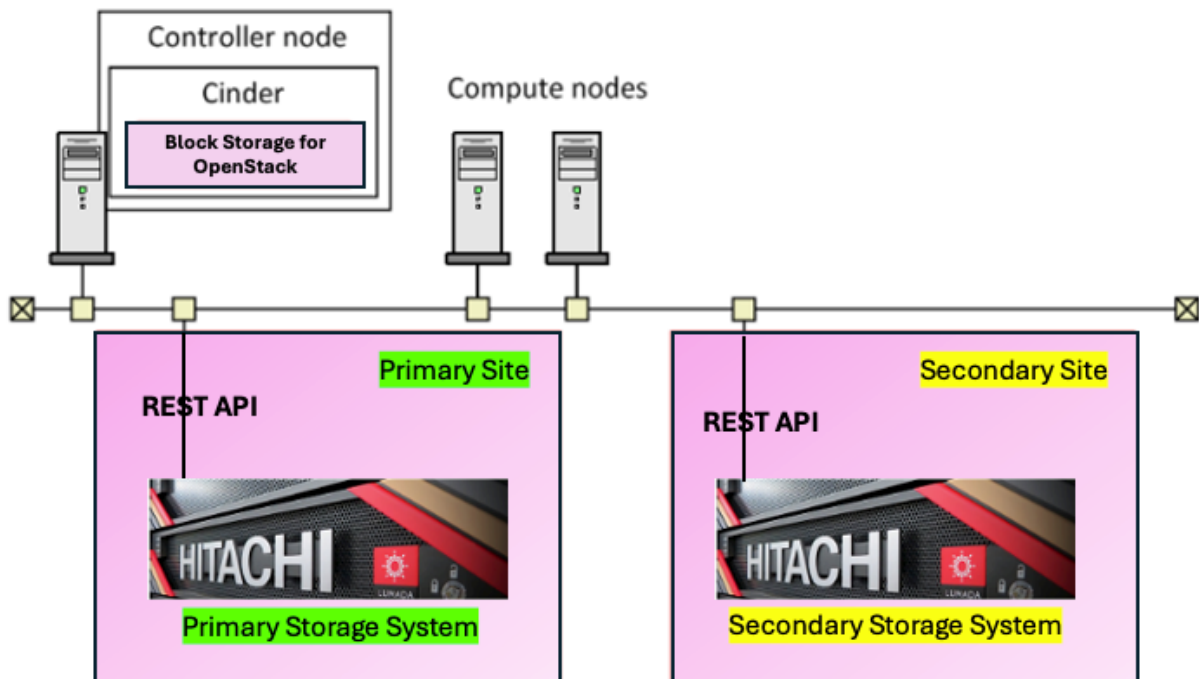
Overview and configuration

Global-Active Device (GAD) is a high-availability capability of VSP storage systems that provides synchronous, active-active volume replication, enabling continuous data access across geographically separated sites.

It allows both sites to read and write data simultaneously, ensuring near-zero RTOs and no data loss during failures by maintaining consistent, mirrored data volumes. Use GAD to make data on individual volumes redundant between two VSP storage systems.

See the *Global Active Device User Guide* on the Product Documentation portal (docs.hitachivantara.com) for more information.

The following illustration shows a configuration example.



A GAD pair is mapped as a Cinder volume, on which you can perform operations from a server in a single-server configuration or in a cross-path configuration. You cannot perform operations from a server in a server-cluster configuration.

Note the following:

- You cannot apply GAD configuration and remote replication configuration to the same backend.
- You cannot use Asymmetric Logical Unit Access (ALUA).

Creating a global-active device environment

Before you can use global-active device (GAD), you must use other storage system management tools to create the prerequisite environment by connecting remote paths, configuring a quorum disk, and creating a virtual storage machine (VSM).

Block Storage Driver for OpenStack supports the following configurations:

- The P-VOL is not registered to a VSM
A VSM is configured on the secondary storage system by using the actual information of the primary storage system, and the P-VOL is not registered to a VSM on the primary storage system.
- The P-VOL is registered to a VSM
A VSM is configured on both the primary and secondary storage systems, and the volumes registered to each VSM are assigned the same virtual LDEV number. The P-VOL is registered to the VSM on the primary storage system.

Block Storage Driver for OpenStack automatically runs some procedures. See *Workflow for creating a GAD environment* in the *Global Active Device User Guide* on the Product Documentation portal (docs.hitachivantara.com) for details.

Complete the following steps to set up the secondary storage system:

1. Set the GAD reserve attribute on the S-VOL.
2. Create a host group.
If the `hitachi_group_request` parameter is `True`, you can skip this procedure.
3. Create the S-VOL.
4. Add an LU path to the S-VOL.
5. Update the CCI configuration definition files.
6. Create the GAD pair.
7. Add an alternate path to the S-VOL.

Note: The user specified for the `hitachi_rest_user` parameter and `hitachi_gad_rest_user` parameter must have following roles:

- Storage Administrator (View and Modify)
- Storage Administrator (Remote Copy)

Reserve unused host group IDs (iSCSI target IDs) in ascending order for the resource groups created on the VSM. The number of IDs is one plus the sum of the number of controller nodes and the number of compute nodes.

The LUNs of the host group (iSCSI targets) of the specific ports on the primary storage system must match the LUNs of the host group (iSCSI targets) of the ports on the secondary storage system. If they do not match, match the LUNs of the primary storage system with those of the secondary storage system.

When the same storage system is used both as the secondary system for GAD configuration and as a backend storage system for general use, the same ports cannot be shared across different backend storage systems. Specify different ports in `hitachi_target_ports` parameter, `hitachi_compute_target_ports` parameter, or `hitachi_rest_pair_target_ports` parameter between different backend storage systems.

The following is an example of a backend storage system for GAD to be configured on `cinder.conf`:

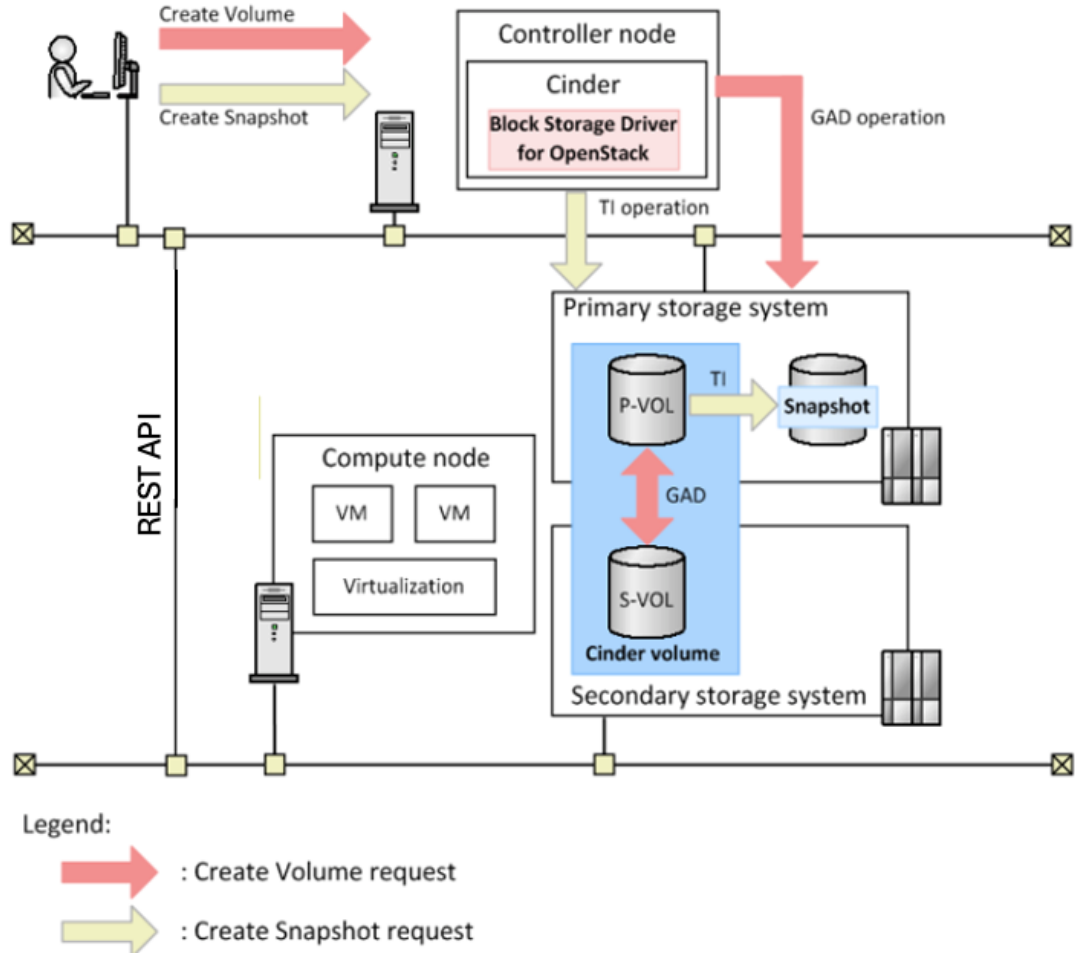
```
[hitachi_vsp_5k_gad]
#vsp-Site1
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
volume_backend_name = hitachi_vsp_5k_gad
san_ip = ip-address
san_login = username
```

```
san_password = password
hitachi_storage_id = vsp-id
hitachi_pools = 5
hitachi_ldev_range = 00:A1:10-00:A1:64
hitachi_target_ports = CL5-F,CL6-F
hitachi_group_create = True
# vsp-Site2
hitachi_mirror_rest_api_ip = ip-address
hitachi_mirror_rest_user = username
hitachi_mirror_rest_password = password
hitachi_mirror_storage_id = vsp-id
hitachi_mirror_pool = 3
hitachi_mirror_ldev_range = 00:A1:10-00:A1:64
hitachi_mirror_target_ports = CL1-B,CL2-B
hitachi_mirror_pair_target_number = 0
hitachi_mirror_rest_api_port = 443
hitachi_mirror_rest_pair_target_ports = CL1-B,CL2-B
hitachi_mirror_ssl_cert_verify = false
hitachi_quorum_disk_id = 2
hitachi_replication_copy_speed = 15
```

Volume operations in a GAD configuration

If you create a Cinder volume in a GAD configuration, each GAD pair is mapped to a Cinder volume. You can then perform operations on Cinder volume without thinking of it as a GAD pair.

The following illustration shows the behavior of the volumes in a GAD pair in two storage systems that have a Cinder volume in the compute node when a `Create Volume` or `Create Snapshot` request is run.



If you run a `Create Volume` request to create a volume with the GAD attribute, a GAD pair is created for a Cinder volume. Using OpenStack, you cannot operate each volume for a GAD pair as a separate Cinder volume.

If you run a `Create Snapshot` request to create a snapshot of a Cinder volume, a snapshot of the P-VOL of the GAD pair is created.

Creating volumes

To create a volume using `Create Volume` with the GAD attribute specified, specify `hbsd:topology=active_active_mirror_volume` for the volume type as follows:

```
# cinder type-create <volume type name>
# cinder type-key <volume type name> set
hbsd:topology=active_active_mirror_volume
# cinder create --volume-type <volume type name>
```

The following is an example of creating a volume type for GAD using the OpenStack CLI:

```
openstack volume type create vsp_5k_gad

openstack volume type set --property
volume_backend_name=hitachi_vsp_5k_gad vsp_5k_gad

openstack volume type set --property
hbsd:topology=active_active_mirror_volume vsp_5k_gad
```

In this case, the following restrictions apply:

- You cannot create a volume for which the deduplication and compression function is enabled.
- When you create a volume running `create volume from volume` or `create volume from snapshot`, you cannot specify `THIN` for the `hitachi_default_copy_method` parameter or the `copy_method` metadata.

If P-VOL registered to a VSM is configured:

- Create volumes with the GAD attribute specified.
- Before performing a volume operation, set a virtual LDEV number for every P-VOL.

Unavailable Cinder functions

If a GAD configuration is used, you cannot use the following Cinder functions:

- Volume migration (storage assisted)
- Manage Volume
- Unmanage Volume

In addition, if P-VOL registered to a VSM is configured, the `backup creation` command cannot be run with the `--snapshot` option or the `--force` option.

See the *Block Storage Driver for OpenStack User Guide* on the Product Documentation portal (docs.hitachivantara.com) for more information about using Cinder with GAD as well as configuring the host mode option.

General requirements

Configuring VSP storage systems in an OpenStack environment involves the following prerequisite foundational steps:

- Configure VSP storage systems
- Use supported operating systems
- Use supported storage protocols
- Select storage protocols to connect hosts to VSP storage systems
- By leveraging one of the following storage protocols or a combination of them, link any host to VSP storage systems.
 - Fibre Channel storage connectivity (required HBA and respective driver)
 - iSCSI storage connectivity
 - Host configuration
 - Multipathing

Hitachi Virtual Storage Platform system configuration

Set up Hitachi storage

Configure the settings for the storage systems as outlined. For a detailed explanation of each setting, refer to the user guide for the storage system.

Common resources:

1. **All resources**
The name of any storage resource, such as a DP pool or a host group. The name cannot contain any whitespace characters or it will be unusable by the driver.
2. **User accounts**
Create a storage device account belonging to the Administrator User Group.
3. **DP Pool**
Create a DP Pool to be used by the driver.
4. **Resource group**
If using a new resource group for exclusive use by an OpenStack system, create a new resource group, and assign the necessary resources, such as LDEVs, port, and host group (iSCSI target) to the created resource.
5. **Ports**
Enable Port Security for the ports used by the driver.

If you use iSCSI:

Ports

Assign an IP address and a TCP port number to the port.

See the *Hitachi block storage driver* system requirements at

<https://docs.openstack.org/cinder/2024.2/configuration/block-storage/drivers/hitachi-vsp-driver.html> for more information.

Configuring Host Mode and Host Mode Options

On VSP storage systems, configure the Host Mode and Host Mode Options (HMOs) specific to operating system (for example Linux) to ensure correct communication and command processing between the host and the storage system.

Identify the correct host mode, review the applicable HMOs, and then apply these settings to the storage system.

Carefully review and determine which host modes you will set when configuring your storage system and observe all cautions and restrictions for the host modes.

The host mode and host mode options must be set on the host-facing port before the host is connected to that port.

Note: If you change the host mode or HMOs after the host is connected, you must restart the host (server) for the new settings to be recognized.

For example, 00 [Standard] is Host mode for Red Hat Linux server hosts, while the HMO's best practice recommendations for Linux are 2, 22, 25, 68.

See *Host mode options for host-facing ports* in the *Provisioning Guide for Open Systems for VSP 5000 Series* on the Product Documentation portal (docs.hitachivantara.com) for more information.

LUN security: To protect mission-critical data in your storage system from illegal access, apply security policies to logical volumes.

Use LUN Manager to enable LUN security on ports to safeguard LUs from illegal access.

If LUN security is enabled on ports, host groups affect which host can access which LUs. Hosts can access only the LUs associated with the host group to which the hosts belong. Hosts cannot access LUs associated with other host groups. For example, hosts in the Linux host group cannot access LUs associated with the windows host group. Also, hosts in the Windows host group cannot access LUs associated with the Linux host group.

See *Applying LUN security on ports* in the *Provisioning Guide for Open Systems for VSP 5000 Series* on the Product Documentation portal (docs.hitachivantara.com) for more information.

Volume management

Configure Port Topology: In the VSP storage system, set the port configuration to Fabric ON with Point-to-Point (P-P) mode for SAN environments that use switches, and to Fabric OFF with Point-to-Point (P-P) mode for direct connections between the HBA and storage ports.

- Fabric ON: For SAN environment includes Fabric switches
- Fabric OFF: Direct connect (HBA-to-Storage ports)

Correct port configuration ensures stable path discovery and optimized throughput between the Linux host and Hitachi storage.

Host Group and LU path configuration: After hosts and the storage system are physically connected through cables, switches, and related components, I/O paths can be established between the hosts and the logical volumes.

LU Paths: The LU paths determine which host has access to which logical volume. Logical volumes accessible to hosts are referred to as logical units (LUs), and the connections between hosts and LUs are referred to as LU paths.

Before LU paths are defined, server hosts must be classified into host groups.

Host group: The host bus adapters (HBAs) of the hosts must be registered as host groups.

A host group can include only hosts connected to the same port. For example, if two hosts are connected to port 1A and three Linux hosts are connected to port 1B, all five Linux hosts cannot be

registered in a single host group. The two Linux hosts on port 1A must be registered in one host group, and the three Linux hosts on port 1B must be registered in another host group.

After server hosts are classified into host groups, you associate the host groups with logical volumes.

Paths can be defined between a single server host and multiple logical units (LUs).

In a Fibre Channel environment, up to 2,048 LU paths can be defined for one host group, and up to 2,048 LU.

See *Configuring the storage system* in the *System Administrator Guide for VSP 5000 Series* on the Product Documentation portal (docs.hitachivantara.com) for more information.

Supported storage protocols

Fibre Channel

Fibre Channel (FC) is a high-speed, low-latency, and lossless network protocol primarily used to connect servers to data storage devices within storage area networks (SANs) and presented from the storage system.

Fibre Channel Zoning

Fibre Channel (FC) zoning is a fabric-based service used to partition a Storage Area Network (SAN) into logical, isolated group of devices (initiators and targets).

Zoning allows specific groups of devices to communicate with each other. It enables access control in a SAN environment.

Zoning terminology

The following are zone member types:

- PWWNs (WWPN) – Port Worldwide name
- Switch/domain-id + physical port/interface
- N_Port_ID (FCID)
- NWWN- Node Worldwide Name
- FC Alias: Contains one or more zone member types

Through zoning, a SAN administrator can configure which HBA WWPNs on a host can connect to which WWPNs on the Hitachi storage processors.

Initiator: The host port in the Fibre Channel HBA is referred to as the initiator.

Target: The storage processor port in the Hitachi storage system is referred to as the target.

Hitachi recommends the following:

- For utmost availability with slightly higher administrative cost, recommends Single Initiator to Single Target (SI-ST) zoning. Brocade Peer Zoning and Cisco Smart Zoning are supported to reduce the administrative burden.
- Each HBA port should only see one instance of each LUN. This is primarily based on years of experience with fabrics and to avoid potential availability issues where host HBA ports can be overrun leading to performance issues and error recovery with fabric path issues (transient or otherwise) is faster and less impactful to hosts.

See *Configuring Fibre Channel ports* in the *Provisioning Guide for Open Systems for VSP 5000 Series* on the Product Documentation portal (docs.hitachivantara.com) for more information.

iSCSI

iSCSI (Internet Small Computer System Interface) is a network protocol that allows computer to access remote storage devices over standard TCP/IP network as if they are locally attached. By encapsulating SCSI command into IP packets, it enables cost-effective, block-level storage access to SANs.

Linux distributions provide support for iSCSI and include a built-in software initiator.

In environments based on RHEL OS, the open-iSCSI package functions as the default initiator.

The iSCSI software and related tools are provided through the `iscsi-initiator-utils` package.

Install the package, if it is not available.

Start and enable iSCSI daemon. iSCSI initiator information can be obtained in Red Hat host in the `/etc/iscsi/initiatorname.iscsi` file.

Hitachi storage iSCSI guidelines

The following guidelines should be implemented on both ends of a connection and in some cases require compatibility through all adjoining switches and links.

- The Linux host must have an Ethernet interface, and preferably dedicated ports to establish reliable communication with the iSCSI ports on the VSP storage system. Using a dedicated network interface helps ensure optimal performance and minimizes potential conflicts with other network traffic.
- It is considered best practice to separate iSCSI traffic from regular network traffic by using dedicated network ports, switches, and supporting infrastructure. If physical network separation is not feasible due to topology constraints, it is recommended to isolate iSCSI traffic using VLAN subnets. Additionally, it is strongly advised to configure iSCSI in a multipath setup to ensure path redundancy and improve fault tolerance.
- Changes to iSCSI port settings usually cause a link down/up event so should be changed with resilience and alternate paths in place and within appropriate maintenance periods or quiet times. The default Maximum Transmission Unit (MTU) size is 1500 which is generally compatible across networks. However, increasing the MTU or using jumbo frames (MTU=9000) can provide a performance benefit (less overhead from IP/TCP headers). You must ensure compatibility throughout the end-to-end switches and links when using larger MTU sizes. The MTU frame size or Maximum Segment Size (MSS payload) may need adjustment across the environment. The iSCSI ports do not support fragment processing (dividing a packet). When the maximum transmission unit (MTU) of a switch is smaller than that of an iSCSI port, packets might be lost, and data cannot be transferred correctly. The MTU value for the switch must be the same as or greater than the MTU value for the iSCSI port. For details of the MTU setting and value, see the user documentation for the switch.
- The default Maximum Window Size is 64 KB and can be adjusted up to 1024 KB. Larger values can improve performance using high delay (long distance) and high bandwidth networks. This is a tuning value which will usually require adjustment, and results may vary between environments and applications. It is recommended to validate and test values in advance of production usage.

Review the instructions in the following documentation, to understand how iSCSI connections are configured.

- *iSCSI port settings* in the *Provisioning Guide for VSP One Block 20 Series* on the Product Documentation portal (docs.hitachivantara.com).
- *Configuring iSCSI on a Linux System with Virtual Storage Platform One Block 20 25G iSCSI Interface* on the Hitachi company website (hitachivantara.com)
- *Overview of iSCSI operations* in the *Provisioning Guide for Open Systems for VSP 5000 Series* on the Product Documentation portal (docs.hitachivantara.com).
- *Creating iSCSI targets and registering hosts in an iSCSI target* in the *Provisioning Guide for VSP E Series* on the Product Documentation portal (docs.hitachivantara.com)

Host configuration

HBA and WWN

Host Bus Adapter (HBA) hardware and the corresponding driver package are required for Fibre channel connectivity to VSP Storage. Host Bus Adapters (HBAs) play a crucial role in Storage Area Network (SAN) environments, enabling high-speed and reliable data transfer between servers and storage systems. For iSCSI or NVMe/TCP, configure the respective initiator software and confirm connectivity to target ports.

Key optimization parameters include tuning the HBA port timeout, queue depth, and other performance-related settings. Depending on the HBA's type and model, some parameters may require modification through the HBA BIOS. For detailed instructions, refer to the vendor's documentation on BIOS-level configuration.

Verify that the Fibre Channel adapters and their device drivers are installed correctly.

After the installation is confirmed, identify the World Wide Name (WWN) for each HBA that will connect to the storage system.

To identify and collect information about Fibre Channel HBAs on a RedHat Linux host, the following commands can be used:

```
cat /sys/class/fc_host/host*/port_name
cat /sys/class/fc_host/host*/node_name
cat /sys/class/fc_host/host*/speed
cat /sys/class/fc_host/host*/port_state
```

These files provide the following details:

- **port_name** – World Wide Port Name (WWPN)
- **node_name** – World Wide Node Name (WWNN)
- **speed** – Current link speed
- **port_state** – Link status

HBA port timeout: In Red Hat Enterprise Linux (RHEL), the HBA Port Down timeout parameter specifies the duration a Linux system waits before terminating a Fibre Channel connection after losing communication with the port.

These settings are crucial for maintaining stable multipathing and avoiding unnecessary path failovers. To mitigate I/O disruptions, configure the HBA port timeouts according to the vendor's recommendations.

For Emulex HBA, the default value is:

```
# cat /sys/module/lpfc/parameters/lpfc_devloss_tmo
30
```

For QLogic HBA, the default value is:

```
# cat /sys/module/qla2xxx/parameters/qlport_down_retry
0
```

Multipathing

Multipathing is a host-level software solution that enables operating systems to utilize multiple physical paths between server and a storage system. While optional, multipathing is highly recommended in production environments or any setup where availability and performance are critical.

It provides key capabilities such as:

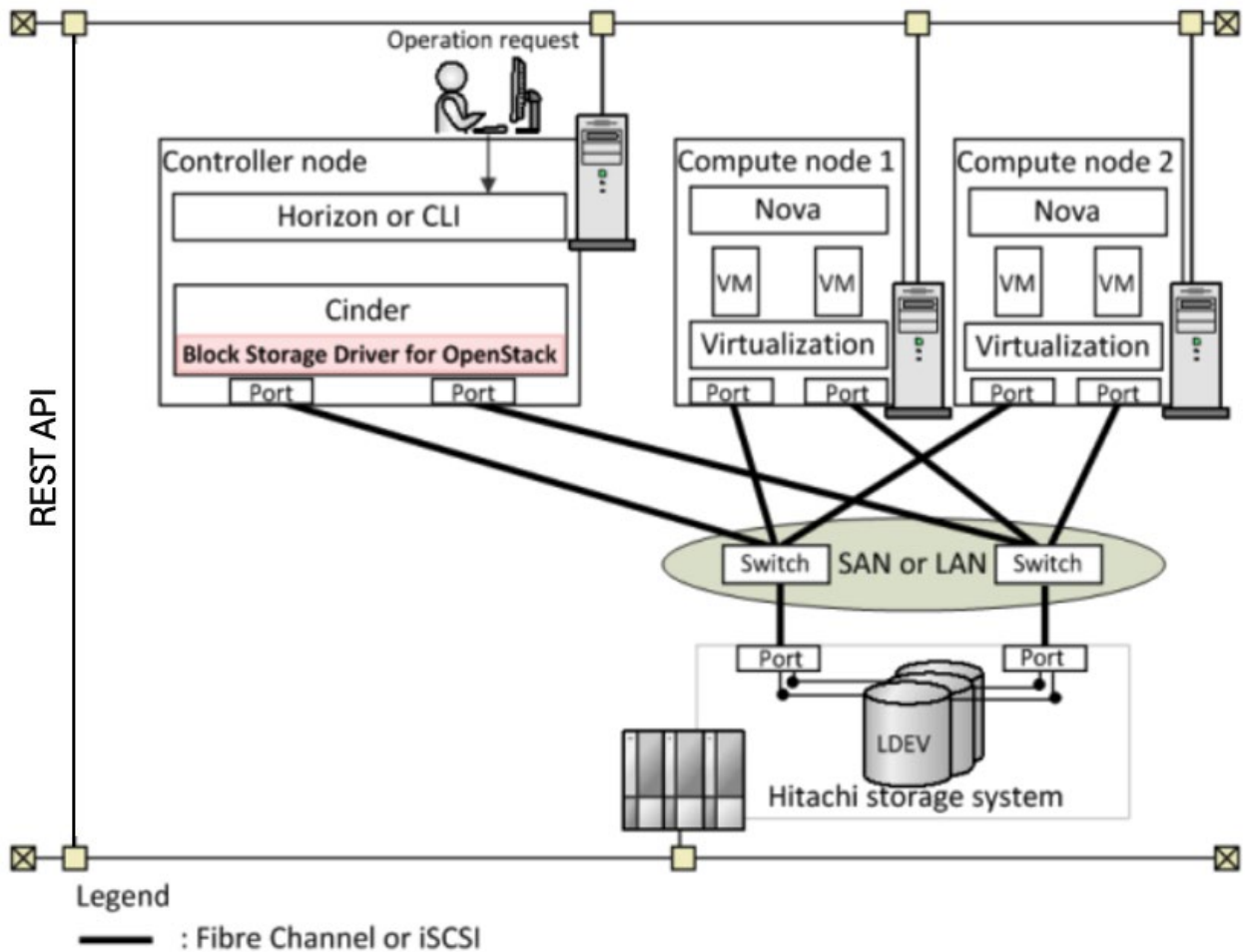
- Path redundancy and fault tolerance: If any component in the SAN (Storage Area Network), such as an adapter, switch, or cable fails, the host can automatically switch to an alternate path. This process, known as path failover, ensures uninterrupted access to storage.
- Load balancing: Multipathing distributes I/O traffic across multiple physical paths, reducing bottlenecks and improving overall performance.
- Improved application availability: Automatic failover and failback mechanisms help maintain consistent access to storage resources.
- Enhanced I/O performance: By leveraging multiple parallel paths, multipathing supports better throughput and responsiveness.
- Simplified administration: Persistent, user-friendly device naming across cluster nodes makes storage management easier.

In a Fibre Channel and iSCSI SAN environments, full fault tolerance typically involves the following:

- Two or more Host Bus Adapters (HBAs) / Network Interface Card (NIC) on each host.
- Redundant SAN / Ethernet switches.
- Dual storage controller on the storage system (e.g., Hitachi), allowing alternate paths to the disk array.
- Redundant I/O data paths to the storage device enable continuous operations when one of the paths fails.

Note: Do not use Hitachi Dynamic Link Manager (HDLM) for multipathing on an OpenStack solution.

Device mapper multipathing (DM-Multipathing) provides path redundancy in OpenStack as shown in the following illustration.



Controller nodes:

1. Set the `use_multipath_for_image_xfer` parameter to `True`.
2. Configure DM-Multipath for the port specified by the Hitachi Block Storage Driver for OpenStack parameter `hitachi_target_ports` and `hitachi_compute_target_ports`.
3. Start multipath daemon after configuring the DM-Multipath setting.

Compute nodes:

1. Set the `volume_use_multipath` parameter to `True`. List this parameter in the `libvirt` section in the Nova configuration file.
2. Increase the default timeouts in the `nova.conf` file to improve block device allocation performance:

```
[DEFAULT]
block_device_allocate_retries = 300
block_device_allocate_retries_interval = 6
```

```
[libvirt]
```

```
volume_use_multipath = True
```

- Start the multipath daemon after configuring the DM-Multipath setting.

Note the following:

- The recommended value for the DM-Multipath varies depending on the storage device. Refer to multipath configuration guidelines of your storage device.
- Attach or detach volume operation of the volume may fail when one of the iSCSI paths fails. However, VMs that the volume is attached to can be used with the other path.

See the *Hitachi Block Storage Driver for OpenStack Train User Guide* on the Product Documentation portal (docs.hitachivantara.com) to know more about multipathing and high availability.

NTP configuration

For optimal performance, configure VSP storage systems, along with all OpenStack controller and compute nodes, to use NTP. Failure to synchronize the storage system and OpenStack nodes can result in error messages in the Cinder volume controller and REST logs, which manifest as 401 errors.