HITACHI

Hitachi Vantara

Deploy OpenStack Stretched Clusters with Hitachi Virtual Storage Platform and Global-Active Device

Reference Architecture Guide

MK-SL-465-00
May 2026

The open source content used in Hitachi Vantara products may be found within the Product documentation or you may request a copy of such information (including source code and/or modifications to the extent the license for any open source requires Hitachi make it available) by sending an email to OSS_licensing@hitachivantara.com.

Feedback

Hitachi Vantara welcomes your feedback. Please share your thoughts by sending an email message to Docs-Feedback@hitachivantara.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.

Thank you!

Revision history

Changes	Date
Initial release	May 2026

Contents

Reference Architecture Guide

A high availability stretched cluster reference architecture for OpenStack

OpenStack continues to be a strategic platform for organizations that need to run virtual machine workloads with flexibility, operational control, and infrastructure consistency across private and hybrid cloud environments. As customers place more business-critical applications on OpenStack, the availability requirements of the underlying platform increase: the environment must tolerate host, storage, and site-level failures while preserving data integrity and enabling predictable recovery. This reference architecture addresses those requirements by combining OpenStack with Hitachi Virtual Storage Platform (VSP), Global-Active Device (GAD), and the Hitachi Block Storage Driver for OpenStack (HBSD) integrated with OpenStack Cinder.

This paper provides a validated reference design for deploying a stretched OpenStack cluster across two data center sites using Hitachi block storage with synchronous active-active replication. In this solution, HBSD integrates with Cinder to provision GAD-enabled volumes, while GAD maintains synchronized copies of those volumes across both sites. The result is a resilient storage foundation for OpenStack virtual machine workloads that require zero data loss protection and controlled recovery behavior across a metro or campus-style deployment.

The stretched solution described in this paper is based on the capabilities of GAD and can be implemented using the supported GAD system configurations appropriate to customer requirements. GAD supports different deployment models, including single-server, server-cluster, and cross-path configurations, enabling customers to align the architecture to their preferred balance of availability, operational complexity, and connectivity design. This flexibility allows the reference architecture to serve as a general design pattern for stretched OpenStack environments on Hitachi storage, while specific topology choices can be adapted to customer operational and infrastructure constraints.

The solution also combines the infrastructure and orchestration elements needed for end-to-end availability. OpenStack services are deployed in a highly available control-plane model, Masakari is used for compute host failure recovery, multipath I/O provides Fibre Channel path redundancy, and GAD supplies synchronous replication and storage-side arbitration. Together, these capabilities create a complete platform pattern for resilient OpenStack operations across two sites.

Overview

This reference architecture describes a stretched OpenStack environment deployed across two data center sites and integrated with Hitachi Virtual Storage Platform (VSP) block storage. The solution uses Global-Active Device (GAD) to provide synchronous replication between storage systems and the Hitachi Block Storage Driver for OpenStack (HBSD) to expose that capability through OpenStack Cinder.

At a high level, the architecture combines a highly available OpenStack control plane, compute nodes distributed across sites, and GAD-protected block volumes for virtual machine workloads. When a GAD-enabled Cinder volume type is used, HBSD provisions and manages the corresponding replicated volume pair across the participating storage systems.

The stretched solution can be implemented using the supported GAD deployment models that correspond to customer requirements. GAD supports multiple system configurations, including single-server, server-cluster, and cross-path, allowing customers to align the design to their preferred balance of availability, connectivity, and operational complexity.

The sections that follow describe the solution components, the validated implementation used in the lab, and the recovery behavior observed during host, storage, and site-related failure scenarios. Together, these sections show how Hitachi VSP, GAD, HBSD, and OpenStack services such as Cinder, Nova, Masakari, and multipath I/O can be combined to deliver a resilient two-site OpenStack platform.

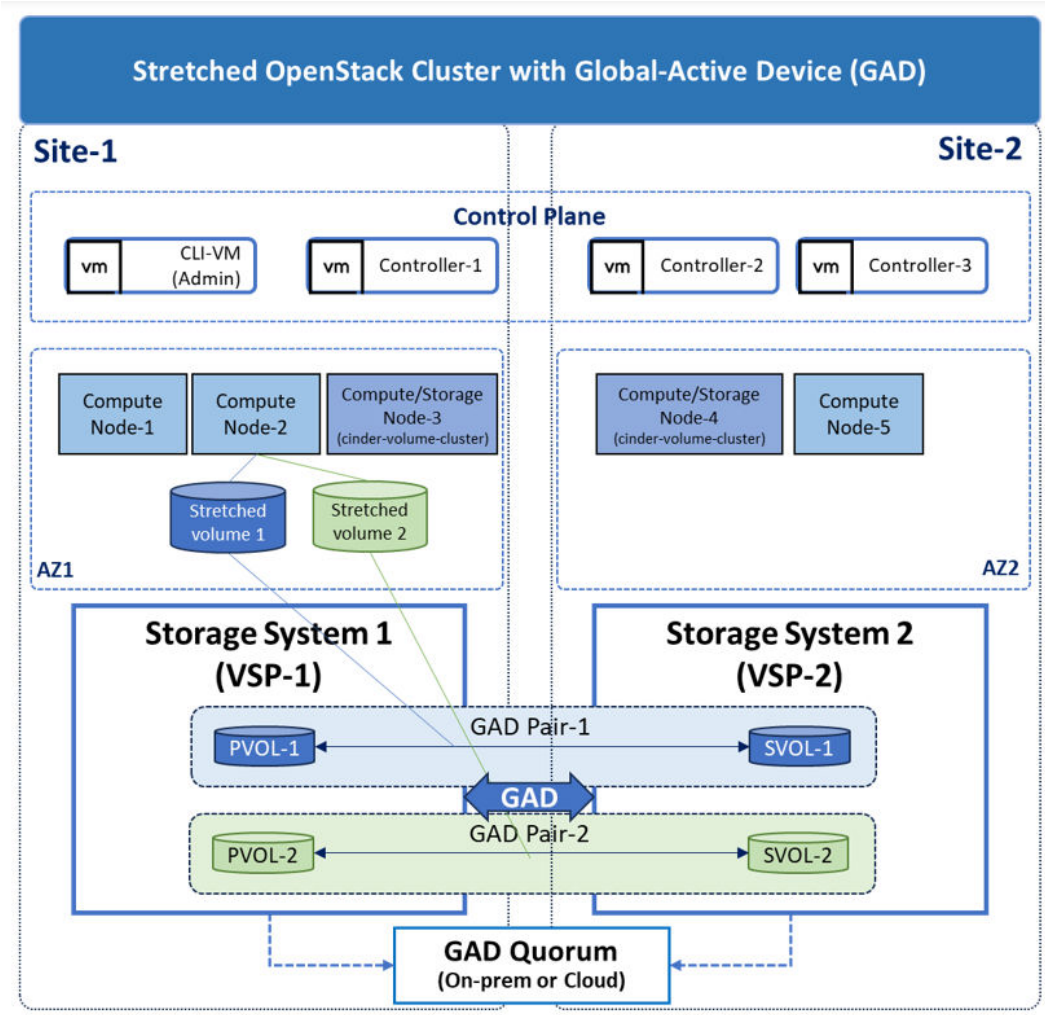
Intended audience

This guide is intended for architects, consultants, administrators, and solution engineering teams who are designing or validating a stretched OpenStack infrastructure on Hitachi VSP storage.

It documents the architecture, validated use cases, and observed recovery behavior from the lab, including both single stretched AZ recovery and cross-AZ/site failover scenarios. Follow the sections in this guide to understand how to design and deploy a stretched OpenStack environment using Hitachi VSP, GAD, and HBSD, and how the platform behaves during the failure scenarios most relevant to enterprise production environments.

Solution design and components

The stretched OpenStack cluster architecture consists of the OpenStack control plane and compute services distributed across two on-premises data center sites, each equipped with Hitachi VSP block storage systems. A server-cluster GAD configuration (non-uniform) is used, meaning compute nodes at each site have Fibre Channel connectivity to their local storage system, with GAD providing synchronous replication between the two storage systems.



The following components are deployed.

Control plane

The OpenStack control plane is deployed in a highly available configuration using three controller nodes. Pacemaker and HAProxy provide clustering and load balancing for OpenStack services including Keystone, Nova API, Neutron, Cinder API, Cinder Scheduler, Horizon, RabbitMQ, MariaDB (Galera), and Masakari. The controllers can be distributed across sites or placed centrally, depending on the deployment model.

For this validation, virtual machines were used for all of the controller nodes.

Compute nodes

Compute nodes run the Nova compute service and host the virtual machine workloads. In this reference architecture, compute nodes are distributed across both sites and organized into Availability Zones (AZs) that align with their physical data center location. Each compute node is equipped with dual Fibre Channel HBAs for storage connectivity.

Storage layer

Each site hosts a Hitachi VSP storage system (for example, VSP 5600 at Site-1 and VSP 5500 at Site-2). The storage systems are interconnected by remote copy paths for GAD replication. A quorum disk (which can be hosted on a third site or cloud-based GAD Cloud Quorum from AWS Marketplace) is used to arbitrate in split-brain scenarios.

Cinder integration with HBSD

The Hitachi Block Storage Driver for OpenStack (HBSD) is the Cinder volume driver that manages volume lifecycle operations against Hitachi VSP storage. For GAD configurations, HBSD is configured with both primary and secondary (mirror) storage system parameters. When a volume is created using a GAD-enabled volume type, HBSD automatically provisions a GAD pair, creating a synchronously replicated volume accessible from both sites.

The cinder-volume service runs in an active/standby configuration across storage nodes at each site, using Cinder volume clustering for high availability.

Global-Active Device overview

Global-Active Device (GAD) is a Hitachi storage replication feature that enables simultaneous read/write access to paired volumes across sites. It provides synchronous, active-active data replication between two Hitachi VSP storage systems. With GAD, a single logical volume is presented simultaneously from both storage systems, enabling hosts connected to either system to perform read and write operations. All writes are synchronously committed to both copies before being acknowledged to the host, ensuring zero data loss (RPO = 0).

Global-active device provides a robust solution for maintaining high availability and disaster recovery in enterprise storage environments. By enabling active-active storage replication, Global-Active Device ensures that data is always up-to-date and accessible across multiple data centers. This capability allows for seamless failover and failback operations, minimizing downtime and ensuring business continuity. Global-Active Device enables you to create and maintain synchronous, remote copies of data volumes.

Key features of Global-Active Device include:

- Synchronous Replication: Ensures real-time data mirroring between primary and secondary storage systems, maintaining data consistency and integrity.
- High Availability: Provides continuous server I/O operations even during failures, ensuring that applications remain operational.
- Disaster Recovery: Facilitates rapid recovery from unexpected failures by enabling seamless failover and failback without impacting storage.
- Load Balancing: Allows for the migration of virtual storage machines without impacting storage, optimizing resource utilization.
- Active-Active Design: Enables production workloads on two systems simultaneously, ensuring full data consistency and protection.
- Zero Recovery Time Objectives: Offers zero downtime and no data loss for applications that require continuous operations.

- Global Storage Virtualization: Allows read and write copies of the same data across two systems or geographic locations.
- Simplified Operations: Automates high availability and simplifies distributed system design.
- Fault-Tolerant Infrastructure: Provides failover clustering and server load balancing without impacting storage.

GAD components

A typical Global-Active Device system consists of storage systems, paired volumes, a consistency group, a quorum disk, a virtual storage machine, paths and ports, alternate path software, and cluster software.

Storage system

The supported combination of storage systems at the primary and secondary sites differs, depending on the models and microcode versions of the storage systems.. An external storage system or iSCSI-attached or cloud-based server, which is connected to the primary and secondary storage systems using Universal Volume Manager, is required for the quorum disk. See the Open-Systems Host Attachment Guide for Virtual Storage Platform Family on the Product Documentation portal (docs.hitachivantara.com) for more information.

Paired volumes

A Global-Active Device pair consists of a P-Vol in the primary storage system and an S-Vol in the secondary storage system.

Consistency group

A consistency group consists of multiple Global-Active Device pairs. By registering global active device pairs to a consistency group, you can resynchronize or suspend the global-active device pairs by consistency group.

Quorum disk

The quorum disk is used to determine the storage system on which server I/O should continue when a storage system or path failure occurs. The quorum disk is virtualized from an external storage system that is connected to both the primary and secondary storage systems. Alternatively, a disk in an iSCSI server, including cloud- based virtual servers (for example, running in Azure), can be used as a quorum disk if the server is supported by Universal Volume Manager.

Virtual storage machine

A virtual storage machine (VSM) is configured in the secondary storage system with the same model and serial number as the (actual) primary storage system. The servers treat the virtual storage machine and the storage system at the primary site as one virtual storage machine.

You can create Global-Active Device pairs using volumes in virtual storage machines. When you want to create a Global-Active Device pair using volumes in VSMs, the VSM for the volume in the secondary site must have the same model and serial number as the VSM for the volume in the primary site.

Paths and ports

Global-Active Device operations are carried out between hosts and primary and secondary storage systems that are connected by data paths composed of one or more physical links.

The data path, also referred to as the remote connection, connects ports on the primary storage system to ports on the secondary storage system. Both Fibre Channel and iSCSI remote copy connections are supported.

Cluster software

Cluster software is used to configure a system with multiple servers and to switch operations to another server when a server failure occurs. Cluster software is required when two servers are in a Global-Active Device server-cluster system configuration.

System configurations for GAD solutions

Single-server configuration

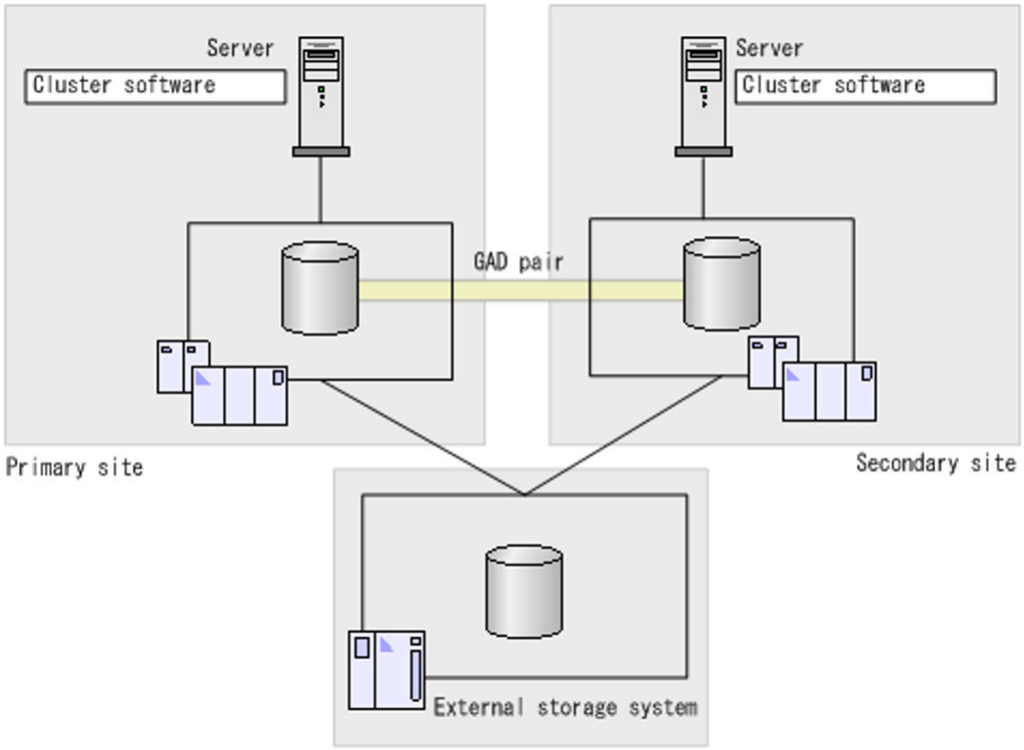
In a single-server configuration, the primary and secondary storage systems connect to the host server at the primary site. If a failure occurs in one storage system, you can use alternate path software to switch server I/O to the other site.

Server-cluster configuration

In a server-cluster configuration that corresponds to a non-uniform topology, servers are located at both the primary and secondary sites. The primary storage system connects to the primary-site server, and the secondary storage system connects to the secondary-site server. The cluster software is used for failover and failback. When I/O on the virtual machine of one server increases, you can migrate the virtual machine to the paired server to balance the load.

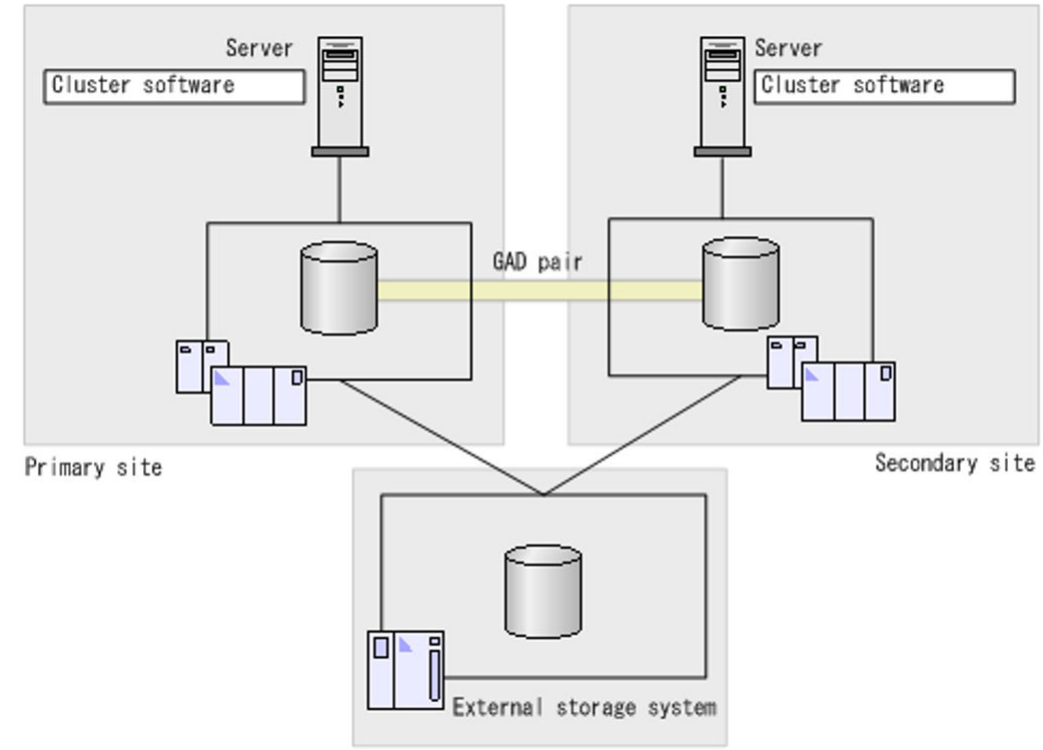
This differs from a uniform (cross-connected) configuration where every host has paths to both storage systems.

This non-uniform topology was chosen for this reference architecture because it simplifies the SAN fabric design (no cross-site FC connectivity required between compute hosts and remote storage) while still providing full data protection through GAD replication. Recovery from a site failure requires VM evacuation to the surviving site, which is managed by Masakari or manual/scripted operational procedures.



Cross-path configuration

In a cross-path configuration that corresponds to a uniform topology, primary-site and secondary-site servers are connected to both the primary and secondary storage systems. If a failure occurs in one storage system, alternate path software is used to switch server I/O to the paired site. The cluster software is used for failover and failback.



GAD failure behavior

When a failure occurs, GAD responds according to the type and scope of the failure:

- Storage system failure: If one storage system goes offline, the surviving system continues serving I/O. The P-VOL or S-VOL enters Block I/O mode on the failed system, and the surviving copy switches to Local I/O mode.
- Inter-site link failure: If the remote copy path between storage systems is disrupted, GAD pairs transition to a suspended state (PSUE). Each host retains access to its local copy. After the link is restored, pairs can be resynchronized.
- Quorum failure: The quorum disk is used for arbitration. If the quorum becomes unreachable along with other failures, GAD follows defined fencing rules to determine which copy remains active.

See *System configurations for GAD solutions* in the *Global Active Device for VSP One Block* guide on the Product Documentation portal (docs.hitachivantara.com) for more information.

System setup and lab environment

System setup

The stretched OpenStack cluster architecture consists of the OpenStack control plane and compute services distributed across two on-premises data center sites, each equipped with Hitachi VSP block storage systems. A server-cluster GAD configuration (non-uniform) is used, meaning compute nodes at each site have Fibre Channel connectivity to their local storage system, with GAD providing synchronous replication between the two storage systems.

The objective of this validation is to evaluate the resiliency of a stretched OpenStack cluster with GAD-backed Cinder volumes against common failure scenarios, including compute host failure, storage path failure, and inter-site link failure. The validation covers both single Availability Zone and multi-AZ deployment models.

Lab environment

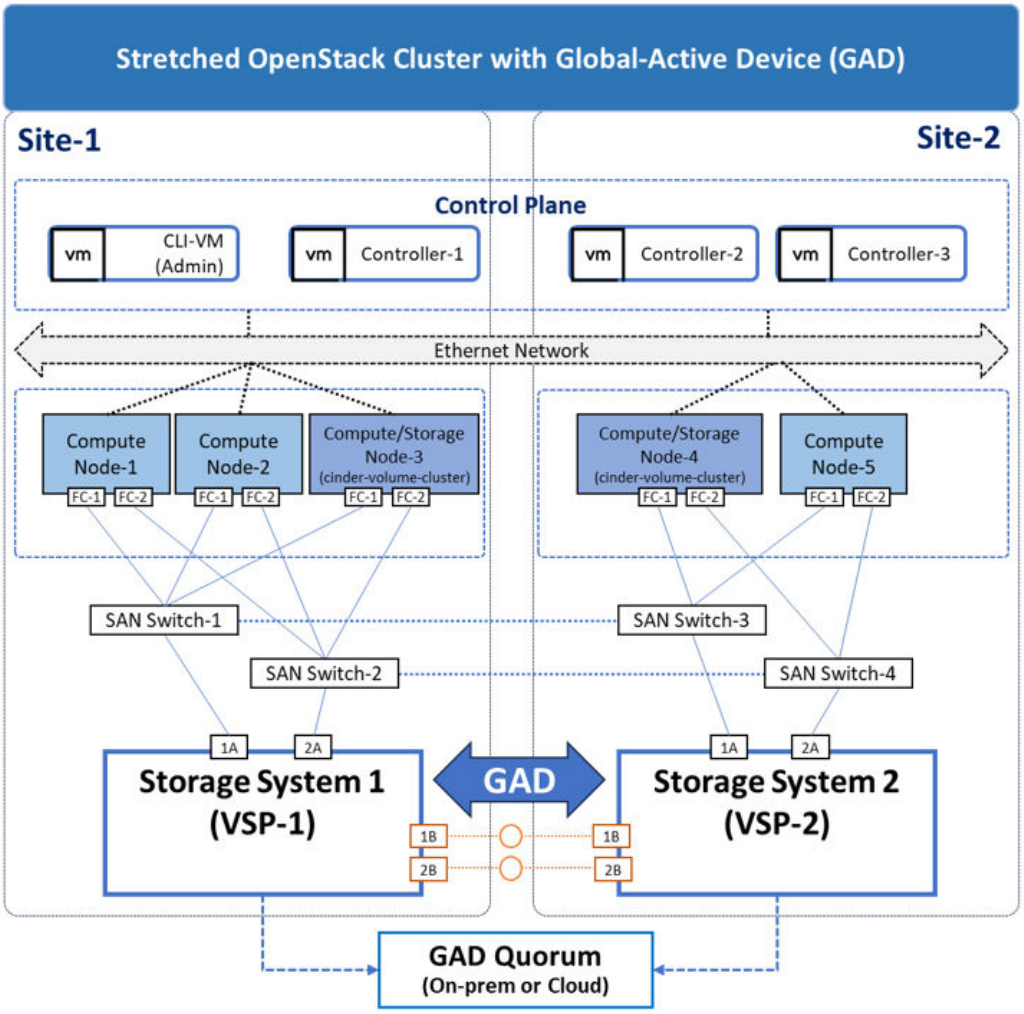
The lab environment consists of the following components:

- OpenStack Version and OS
  - OpenStack 2024.1 (Caracal)/Ubuntu 22.04 LTS
  - OpenStack 2025.1 (Epoxy)/Ubuntu 24.04 LTS
- Control Plane
  - 3 controller nodes (Controller1, Controller2, Controller3) running OpenStack services in HA with Pacemaker
  - Services: Keystone, Nova API, Neutron, Cinder API, Cinder Scheduler, Horizon, RabbitMQ, MariaDB (Galera), Masakari
  - 1 CLI/Admin VM for operational management
- Compute Nodes
  - 5 compute nodes (Hitachi HA810 G3 servers) distributed across two sites:
    - Site-1: Compute Node1, Compute Node2, Compute Node3
    - Site-2: Compute Node4, Compute Node5
  - Each compute node has dual FC HBAs for storage connectivity

- Storage Systems
  - Site-1: Hitachi VSP 5600
  - Site-2: Hitachi VSP 5500
  - GAD replication configured between the two arrays
    - Remote copy paths using FC ports (1B/2B on VSP 5600, 1B/2B on VSP 5500)
  - Quorum disk for GAD arbitration
    - Quorum disk for GAD arbitration
  - VSP E1090 used for GAD quorum

    For detailed storage configuration see Appendix B: Storage System Configuration using GAD (on page 41).

- Storage Nodes
  - Storage-node1 at Site-1:
    - Configured on Compute node-3
    - Active cinder-volume service
  - Storage-node2 at Site-2:
    - Configured on Compute node-4
    - Standby cinder-volume service



The following illustration shows the reference architecture setup.

- Network/SAN Connectivity
  - An Ethernet/IP network (10/25 GbE) connects all OpenStack nodes (controllers, compute, storage nodes) across both sites.
  - Fibre Channel SAN fabrics provide storage data paths. Each site has a pair of FC switches. Compute nodes connect using dual HBAs to the local FC switches.
  - Remote copy paths between the two VSP storage systems use dedicated FC ports for GAD replication traffic.
  - A quorum path connects both storage systems to the quorum disk or GAD Cloud Quorum service.

Infrastructure components summary

Component	Site-1	Site-2
Controller Nodes	Controller1, Controller2	Controller3
Compute Nodes	Node1, Node2, Node3 (HA810 G3)	Node4, Node5 (HA810 G3)
Storage System	VSP 5600 (S/N 400XY)	VSP 5500 (S/N 305AB)
FC Switches	FC Switch-1, FC Switch-2	FC Switch-3, FC Switch-4
FC Data Ports	CL1-A, CL2-A	CL1-A, CL2-A
FC Remote Copy Ports	CL1-B, CL2-B	CL1-B, CL2-B

Storage system configuration for GAD

The configuration for a GAD-enabled OpenStack cluster requires the following.

Virtual Storage Machine (VSM)

Create a VSM on each storage system with the same virtual serial number and model type. The VSM provides a common identity for both storage systems, which is required for GAD pair management. This is configured using the Hitachi Configuration Manager REST API or Storage Navigator.

User Accounts and Resource Groups

Create a dedicated user group with Storage Administrator (Provisioning), Storage Administrator (Remote Copy), and Storage Administrator (View Only) roles. Create a user assigned to this group for Cinder to authenticate against the storage REST API. The user group should have access only to the resource group associated with the VSM.

DP Pools

Create a Dynamic Provisioning (DP) pool on each storage system from pool volumes assigned to the VSM resource group. Ensure the secondary pool has at least the same capacity as the primary pool. The pool IDs are referenced in the cinder.conf configuration (`hitachi_pools and hitachi_mirror_pool`).

LDEV Allocation

Assign unused LDEV IDs to the VSM resource group on both storage systems. Set the virtual LDEV ID to unassigned (65534) for LDEVs that will be managed by Cinder. Ensure virtual LDEV IDs are set for LDEVs in the hitachi_ldev_range on the primary system, because these are required for GAD pair creation.

Host Groups

For each FC port used for data paths, assign unused host group IDs to the VSM resource group. The number of host group IDs must be at least the total number of compute hosts plus one. Alternatively, set `hitachi_group_create = True` in `cinder.conf` to allow HBSD to create host groups automatically.

Remote Copy Paths

Create remote copy paths between the primary and secondary storage systems. The same path group ID must be used on both sides. These paths carry synchronous replication traffic for GAD. Both Fibre Channel and iSCSI are supported for remote copy paths.

Quorum Disk

Create a quorum disk for GAD arbitration. The same quorum disk ID must be used on both storage systems. The quorum can be provisioned using the GAD Cloud Quorum managed service on AWS Marketplace or on a third storage system. The quorum disk ID is referenced in `cinder.conf` as `hitachi_quorum_disk_id`.

For detailed step-by-step storage configuration using the REST API, see the storage system configuration in Appendix B: Storage System Configuration using GAD (on page 41).

HBSD configuration for GAD

The Hitachi Block Storage Driver (HBSD) for OpenStack Cinder must be configured with parameters for both the primary and secondary (mirror) storage systems. The following is a sample cinder.conf backend section for a Fibre Channel GAD configuration:

```
## GAD backend
[hitachi_vsp_5k_gad]
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
volume_backend_name = hitachi_vsp_5k_gad

# Primary storage system (Site-1)
san_ip = <primary_storage_mgmt_ip>
san_login = <username>
san_password = <password>
hitachi_storage_id = <primary_serial_12_digits>
```

```
hitachi_pools = <primary_pool_id>
hitachi_ldev_range = <ldev_range>
hitachi_target_ports = CL1-A,CL2-A
hitachi_group_create = True

# GAD / Mirror configuration (Site-2)
hitachi_mirror_rest_api_ip = <secondary_storage_mgmt_ip>
hitachi_mirror_rest_user = <username>
hitachi_mirror_rest_password = <password>
hitachi_mirror_storage_id = <secondary_serial_12_digits>
hitachi_mirror_pool = <secondary_pool_id>
hitachi_mirror_ldev_range = <ldev_range>
hitachi_mirror_target_ports = CL1-A,CL2-A
hitachi_mirror_compute_target_ports = CL1-A,CL2-A

# GAD-specific parameters
hitachi_path_group_id = <path_group_id>
hitachi_quorum_disk_id = <quorum_disk_id>
hitachi_mirror_ssl_cert_verify = false
hitachi_mirror_rest_api_port = 443
hitachi_replication_copy_speed = 15

# Pair target for internal replication management
hitachi_pair_target_number = 0
hitachi_mirror_pair_target_number = 0
```

See https://docs.openstack.org/cinder/latest/configuration/block-storage/drivers/hitachi-vsp-driver.html#configuration-options to verify default and maximum values.

Volume Type for GAD

To create volumes with GAD replication, you must create a dedicated volume type with the correct extra-spec:

```
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
```

All volumes created using this volume type will automatically be provisioned as GAD pairs with synchronous replication between the primary and secondary storage systems.

Storage Node High Availability

The cinder-volume service should be deployed in an active/standby configuration using Cinder volume clustering. A storage node runs at each site, and the cinder-volume service is configured with the same cluster name on both nodes. If the active storage node fails, the standby node takes over volume management operations. In the reference example, Compute-node-3 (acting as a storage node) at Site-1 runs as the active instance and Compute-node-4 (acting as a storage node) at Site-2 as the standby.

Important Notes

- Do not create volumes whose volume types do not have hbsd:topology=active_active_mirror_volume extra-spec on a GAD-configured backend.
- Virtual LDEV IDs must be assigned for every LDEV in the hitachi_ldev_range on the primary storage system, as virtual LDEV IDs are required for GAD pair creation.
- When using the same storage system for both GAD and non-GAD backends, ensure that different ports are configured for each backend to avoid conflicts.
- Certain Cinder operations (such as storage-assisted migration) have limitations in GAD configurations. Refer to the HBSD driver documentation for the full list of supported operations.

See Appendix A: OpenStack configurations (on page 37) for details.

Enhancing resiliency

Resiliency to Compute Host Failure

To provide automatic recovery from compute host failures, the OpenStack Masakari service is deployed. Masakari monitors the health of compute nodes and, upon detecting a host failure, triggers automatic VM evacuation to a surviving compute node.

Masakari monitors are deployed on each compute host to detect node-level failures. When a host becomes unreachable, Masakari triggers a host-failure recovery workflow that evacuates all VMs from the failed host. Evacuated VMs are restarted on available compute nodes, leveraging the GAD-replicated volumes accessible at the surviving site.

Note: For cross-AZ configuration where VMs are deployed with explicit availability-zone request, automation scripts were developed to orchestrate the recovery in the secondary datacenter. Details are provided in the following sections.

Resiliency to Storage Path Failure

Each compute node is configured with multipath I/O (multipathd) to provide redundant Fibre Channel data paths to the storage systems. With dual HBAs and dual FC switches per site, each volume is accessible through multiple paths. If a path fails, multipathd automatically redirects I/O to a surviving path without any application disruption.

Resiliency to Storage System Failure

Global-Active Device ensures that if one storage system fails entirely, the surviving system continues serving I/O seamlessly. In a non-uniform configuration, hosts connected to the failed storage system lose their local paths but can regain access to volumes after being evacuated (or restarted) on a compute node at the surviving site.

Resiliency to Inter-Site Link Failure

If the remote copy path between the two storage systems is disrupted, GAD pairs transition to a suspended state (PSUE — Pair Suspended due to Error). In this state, each host retains access to its local volume copy and VMs continue running. When the link is restored, GAD pairs can be resynchronized to restore full active-active replication.

Validated use cases and results

To validate the resiliency of the stretched OpenStack architecture with Global-Active Device (GAD) and the Hitachi Block Storage Driver for OpenStack (HBSD), a set of resiliency and recovery scenarios was executed in the lab environment. These tests were designed to measure how the platform responds to failures affecting compute, storage, and inter-site connectivity, as well as how workloads can be recovered across sites when Availability Zone boundaries are introduced, specifically when the deployed VMs are pinned to an availability zone.

The validation covered the following scenarios:

- Single Availability Zone:
  - Compute host failure recovery (on page 21)
  - Storage failure recovery (on page 27)
  - Inter-data-center link failure (on page 27)
- Multi-Availability Zone:
  - OpenStack behavior across availability zones (on page 27)
  - Cross-AZ automated failover/failback at scale (on page 28)

Each failure was intentionally triggered, and the resulting platform behavior was evaluated in terms of workload continuity, recovery workflow, failover behavior, and operational complexity. The sections that follow summarize the results observed in the single-AZ stretched deployment and the multi-AZ recovery model.

Single availability zone — resilience and recovery

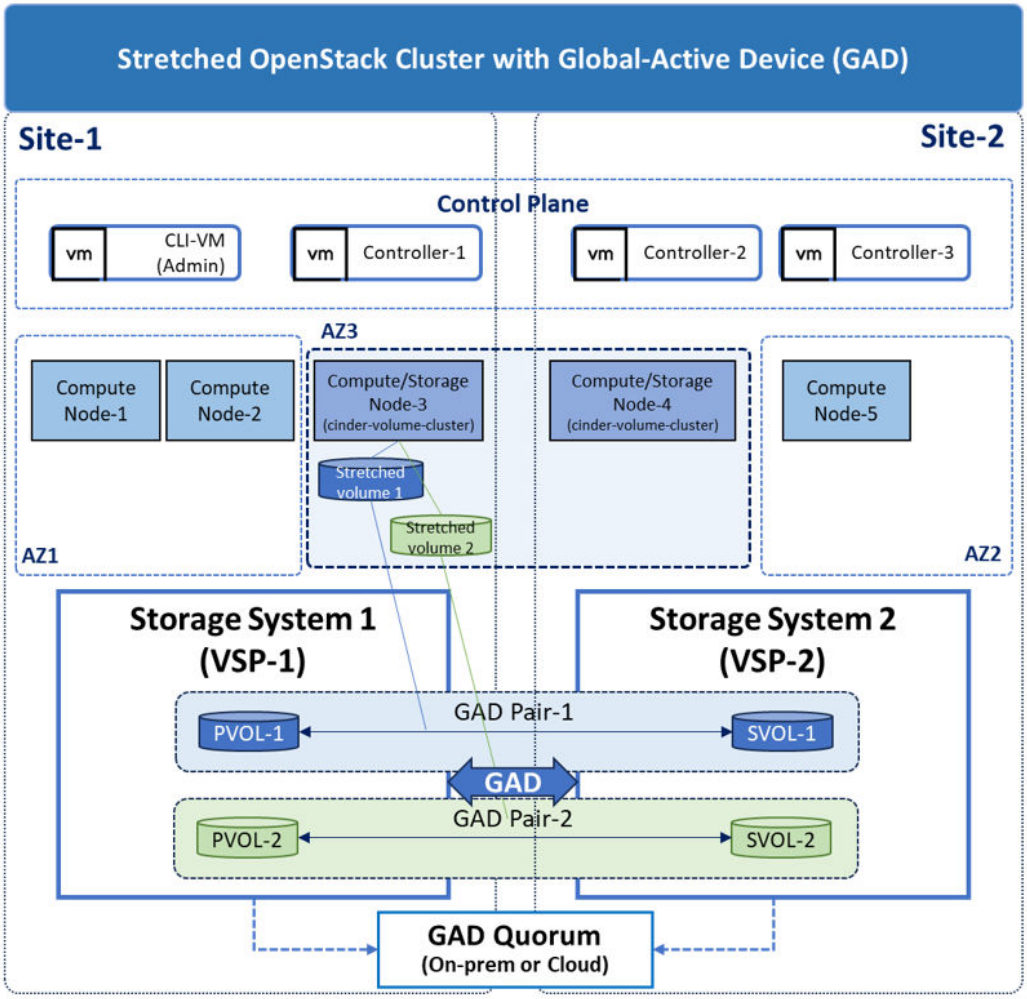
The first series of validation tests focused on resilience within a single Availability Zone, using the stretched architecture and GAD replication across data centers. The objective was to demonstrate that the platform can maintain service continuity and enable recovery under common failure conditions.

Compute host failure recovery

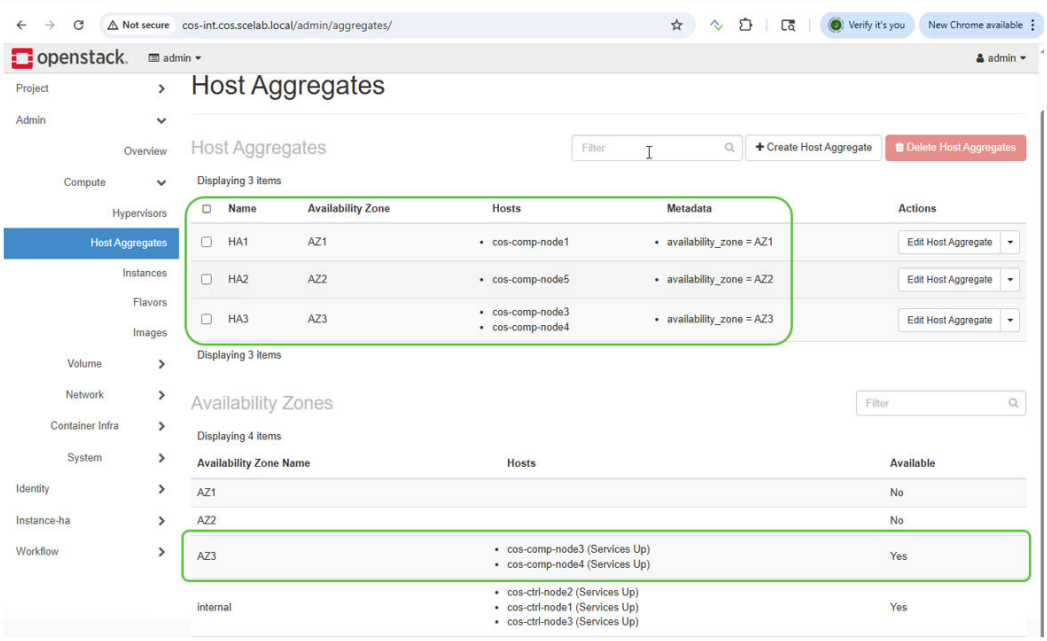
A compute host failure was simulated by powering off a compute node hosting active VMs. The Masakari HA service detected the failure and automatically triggered evacuation of the affected VMs to surviving compute nodes at the secondary site.

Procedure

1. To test the automated failover in the same AZ (across DCs), the cluster was configured with only one active node (node-3) in AZ3 in Site-1 and one active node (node-4) in AZ3 in Site-2, as shown.



2. On the OpenStack dashboard we can see that only node3 and node4 are active in the same AZ3 that spans across Site-1 and Site-2.

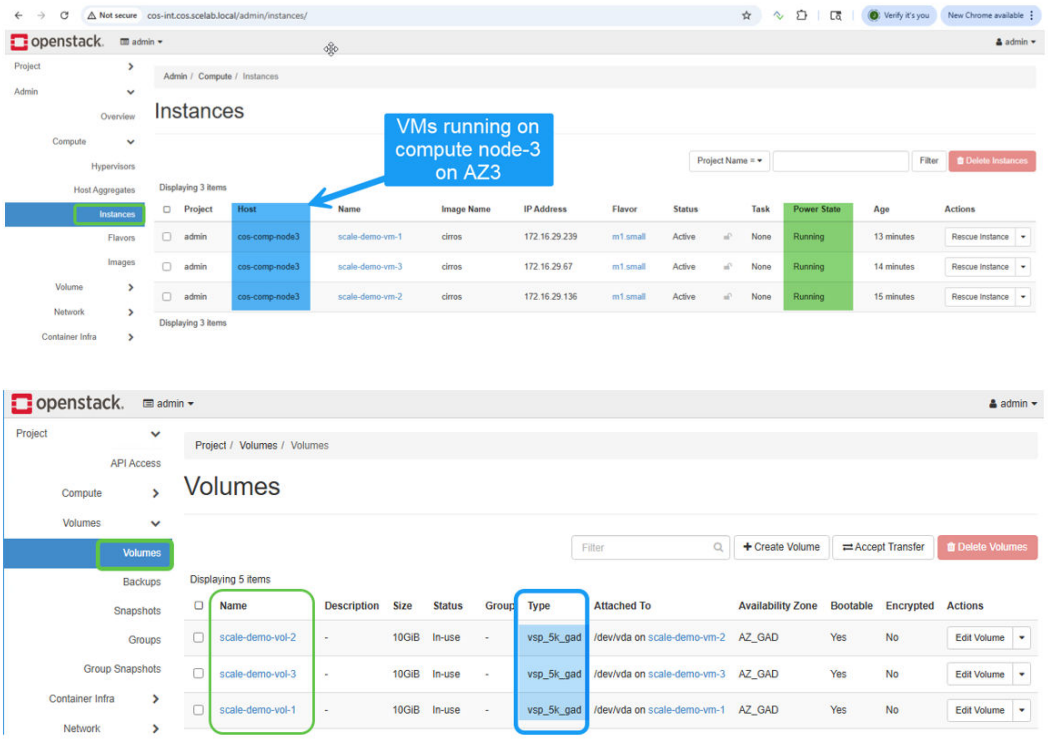


From the CLI we can see the same details about the active compute nodes in AZ3 and the status of the cinder volume service.

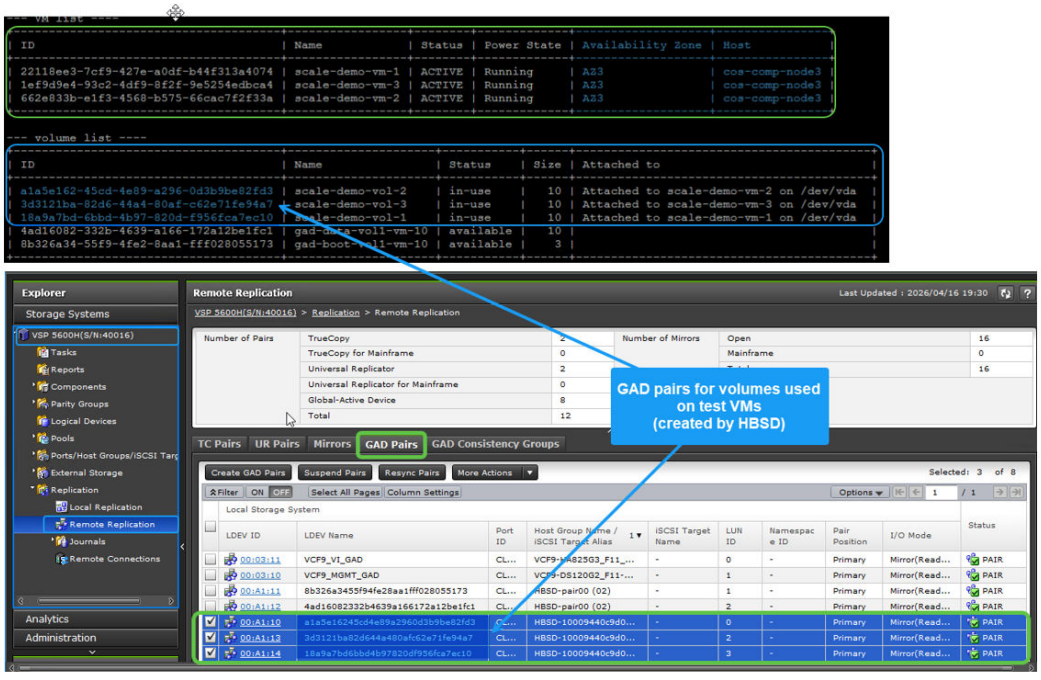
```
$ openstack compute service list
+--------------------------------------+----------------+---------------+----------+----------+-------+
| ID                                   | Binary         | Host          | Zone     | Status   | State |
+--------------------------------------+----------------+---------------+----------+----------+-------+
| f15a3e03-f458-4213-8654-89ee98e956f1 | nova-scheduler | cos-ctrl-node1 | internal | enabled  | up    |
| 53c8b0c0-db73-4072-b4f0-ac89b6737c80 | nova-scheduler | cos-ctrl-node2 | internal | enabled  | up    |
| c41ccf40-ce9b-4df7-929f-1459684e0272 | nova-scheduler | cos-ctrl-node3 | internal | enabled  | up    |
| 98f1f222-65fd-4074-b02f-a2296f538b89 | nova-conductor | cos-ctrl-node2 | internal | enabled  | up    |
| 36ddd9f3-954e-4e85-baaf-c29d4a0ad7da | nova-conductor | cos-ctrl-node3 | internal | enabled  | up    |
| ecc97239-dfcd-4bb3-8e32-6bdb9745700b | nova-conductor | cos-ctrl-node1 | internal | enabled  | up    |
| a31f8d54-3190-4bab-9149-14949fa4bb5d | nova-compute   | cos-comp-node4 | AZ3      | enabled  | up    |
| 2637978f-bd9f-4a34-b72f-1ea92277bc05 | nova-compute   | cos-comp-node1 | AZ1      | disabled | up    |
| 29be0ae7-7bc7-4c9e-adf5-205c0ef06720 | nova-compute   | cos-comp-node5 | AZ2      | disabled | up    |
| 7556f893-2580-4422-b1f8-f6a043553834 | nova-compute   | cos-comp-node3 | AZ3      | enabled  | up    |
+--------------------------------------+----------------+---------------+----------+----------+-------+
```

```
$ openstack volume service list
+------------------+-----------------------------------------+--------+---------+-------+
| Binary           | Host                                    | Zone   | Status  | State |
+------------------+-----------------------------------------+--------+---------+-------+
| cinder-scheduler | cos-ctrl-node1                          | nova   | enabled | up    |
| cinder-scheduler | cos-ctrl-node2                          | nova   | enabled | up    |
| cinder-scheduler | cos-ctrl-node3                          | nova   | enabled | up    |
| cinder-volume    | cos-comp-node1@hitachi_vsp_1090         | AZ1    | enabled | up    |
| cinder-volume    | cos-comp-node1@hitachi_vsp_5600         | AZ1    | enabled | up    |
| cinder-volume    | cos-cinder-ha@hitachi_vsp_5k_gad        | AZ_GAD | enabled | up    |
| cinder-volume    | cos-comp-node5@hitachi_vsp_5500         | AZ2    | enabled | up    |
+------------------+-----------------------------------------+--------+---------+-------+
```

3. VMs are deployed and running on compute node-3 in AZ3, using volumes provisioned with a GAD volume type.

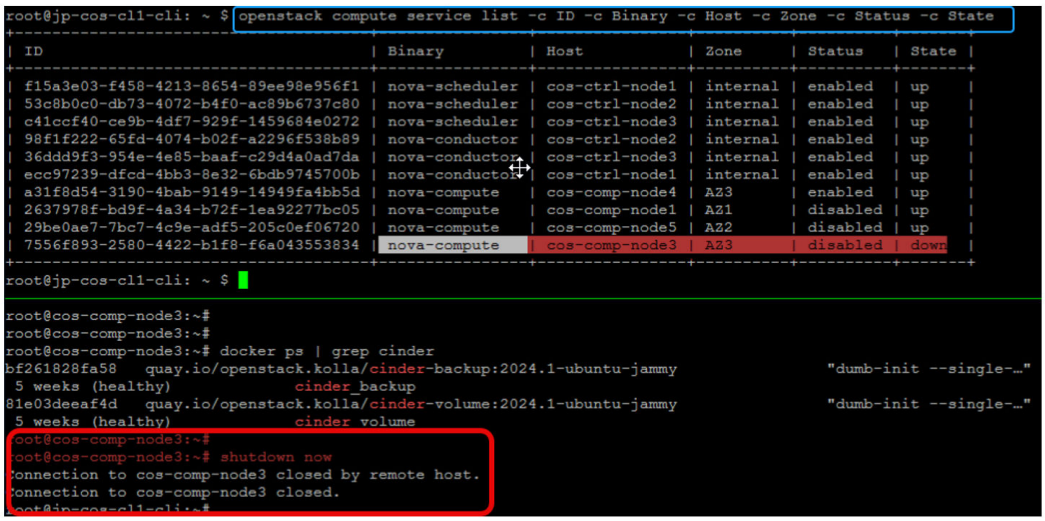


The following figure shows the corresponding GAD pairs for each of the disks (volumes) of the test VMs. HBSD provisions all the volumes and corresponding GAD pairs automatically.



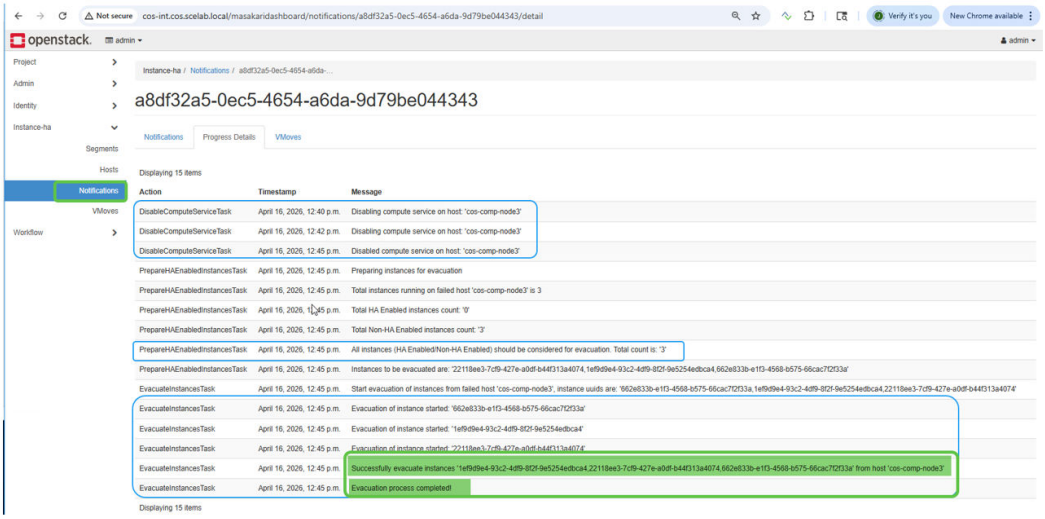
4. Considering that a single compute node is active and available on Site-1, a failure on this node is similar to a complete DC failure. To test this, we did an ungraceful shutdown of the only active node (node-3) that was running the VMs in Site-1/AZ3.

The following screen capture shows compute node-3 being shut down, immediately after the nova-compute service is marked as disabled and in the “down” state.

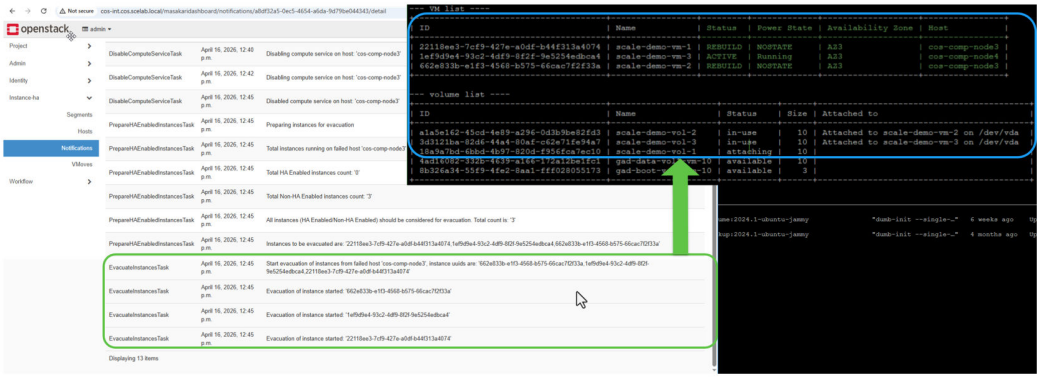


5. Masakari monitors detect the failure and notify the masakari-api service, which records the compute node as disabled in its database.

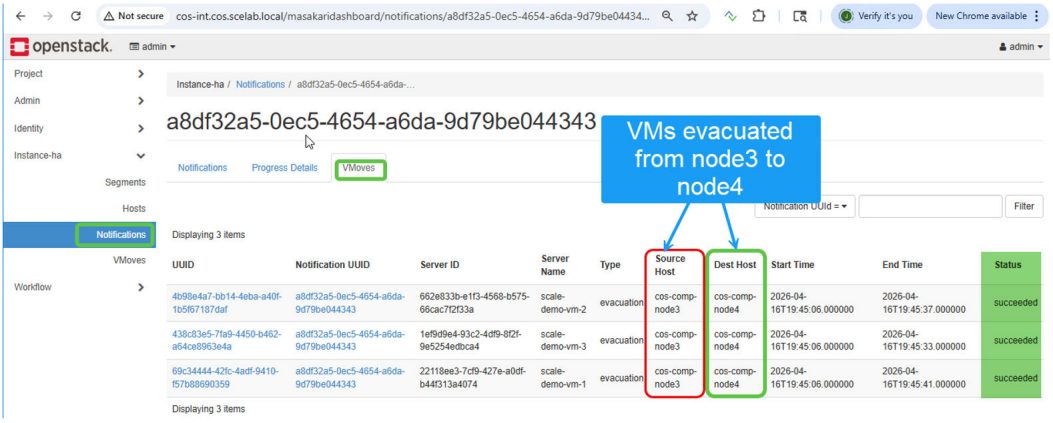
   The masakari-engine then initiates the recovery workflow as shown:



Here we can see how the VMs are being evacuated from compute node-3 to the available compute node (node-4) on the same AZ3, located on Site-2. Masakari calls the Nova evacuate API, and Nova’s scheduler then selects the destination host within the same availability zone (AZ3 in this case). Because the volumes are GAD volumes, they are already accessible to compute node-4 from the Hitachi VSP-2 on Site-2, so evacuation is essentially a re-spawn.

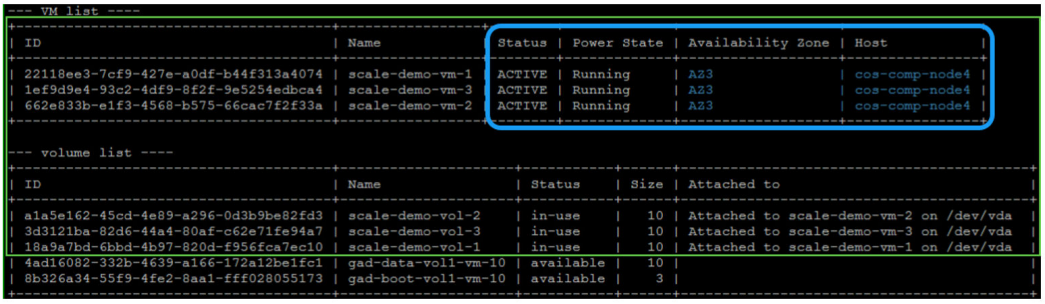


The following figure shows details for each of the evacuated VMs:



The engine processes notifications sequentially per host, so a single host failure won't race against itself, and simultaneous multi-host failures are addressed in parallel.

Here is another screenshot where we can see the VMs running in compute node-4 in AZ3 (Site-2).



--- VM list ----

ID	Name	Status	Power State	Availability Zone	Host
22118ee3-7cf9-427e-a0df-b44f313a4074	scale-demo-vm-1	ACTIVE	Running	AZ3	cos-comp-node4
1ef9d9e4-93c2-4df9-8f2f-9e5254edbca4	scale-demo-vm-3	ACTIVE	Running	AZ3	cos-comp-node4
662e833b-e1f3-4568-b575-66cac7f2f33a	scale-demo-vm-2	ACTIVE	Running	AZ3	cos-comp-node4

--- volume list ----

ID	Name	Status	Size	Attached to
a1a5e162-45cd-4e89-a296-0d3b9be82fd3	scale-demo-vol-2	in-use	10	Attached to scale-demo-vm-2 on /dev/vda
3d3121ba-82d6-44a4-80af-c62e71fe94a7	scale-demo-vol-3	in-use	10	Attached to scale-demo-vm-3 on /dev/vda
18a9a7bd-6bbd-4b97-820d-f956fca7ec10	scale-demo-vol-1	in-use	10	Attached to scale-demo-vm-1 on /dev/vda
4ad16082-332b-4639-a166-172a12be1fc1	gad-data-vol1-vm-10	available	10	
8b326a34-55f9-4fe2-8aa1-fff028055173	gad-boot-vol1-vm-10	available	3	

Result

- VMs were brought back online on surviving compute resources with minimal disruption. Because the Cinder volumes are backed by GAD pairs, the evacuated VMs could attach to the same volumes from the secondary storage system at Site-2.
- The automatic recovery time depends on the number of VMs being evacuated, target host capacity, and boot times, but typically takes a few minutes. In this case, we can see it took 5 minutes to get the VMs up and running on Site-2.

Storage failure recovery

A storage path failure was simulated by disabling FC switch ports connecting compute hosts to the local storage system. In a non-uniform GAD configuration, this causes all VMs on the affected compute nodes to lose I/O access, and VMs enter a paused state because of I/O errors.

Result

VMs could be recovered by restarting or unpausing them on compute nodes at the secondary site, where the GAD secondary volumes (S-VOLs) remained fully accessible. A simple VM restart/unpause operation restores service. GAD contributes to data availability by maintaining the active-active volume copies across sites.

Inter-data-center link failure

The inter-site link between the two storage systems was disrupted to simulate a connectivity failure between data centers.

Result

- VMs continued running without interruption. GAD pairs transitioned to PSUE (Pair Suspended due to Error) state, and each host retained access to its local volume. After the link was restored, GAD pairs were resynchronized to resume active-active replication. No VM downtime occurred during the link failure itself.
- In a single-AZ model, the stretched design successfully tolerated and recovered from host, storage, and inter-site link failures using GAD-backed storage.

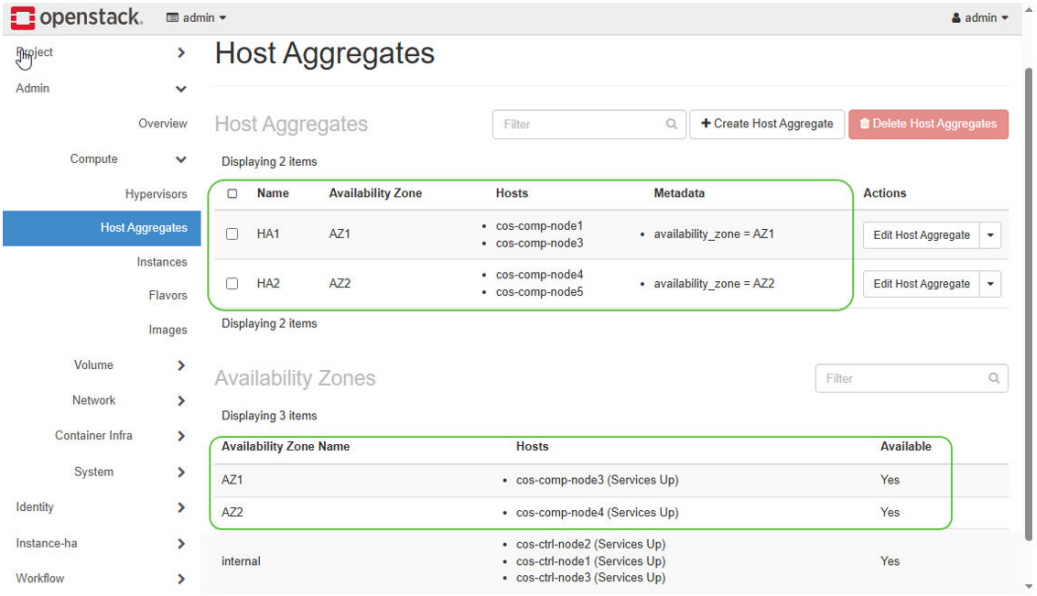
Multi-AZ/host aggregates — cross-AZ recovery

The second series of tests addressed a more advanced deployment model using multiple Availability Zones aligned to host aggregates at each data center site. This model is for cross-AZ recovery scenarios where the deployed VMs are pinned to an AZ.

OpenStack behavior across availability zones

OpenStack Masakari provides automatic recovery for VM instances affected by compute-host failure by invoking Nova evacuation. Whether evacuation remains within the original Availability Zone or can place the instance in a different AZ depends primarily on Nova scheduling constraints and the deployment's Masakari failover-segment design. If a VM was created without an explicit availability-zone request, its RequestSpec does not carry an AZ pin, Nova is free to place the evacuated instance on any eligible destination host, including a host in another AZ.

If a VM was created with an explicit Availability Zone, Nova treats that as a placement constraint and continues to restrict move operations to that AZ. In that case, if no healthy destination hosts remain in the original AZ, evacuation cannot find an eligible target in another AZ automatically. Re-homing such a VM across AZs therefore requires operational steps or external orchestration to re-home workloads to a different AZ/site.



Cross-AZ automated failover/failback at scale

To address the need for automated recovery of multiple VMs across Availability Zones and data centers, where VMs have been deployed with explicit availability zone, Hitachi developed the Hitachi Site Failover Orchestrator for OpenStack.

This automation tool enables organizations to orchestrate disaster recovery operations at scale, including:

- Automated failover of virtual machines across AZs and between sites
- Automated failback to the original AZ or site after stabilization, using the same DR failover workflow

Note: Partners and customers can contact the Hitachi PM team or CoE to schedule a demo or to learn more about the Hitachi Site Failover Orchestrator for OpenStack.

Resilience to Compute Host Failure

Procedure

1. To test the automated failover across DCs/AZs, the cluster was configured with only one active node (node-3) in AZ1 and one active node (node-4) in AZ2, as shown.

```
$ openstack compute service list
+--------------------------------------+----------------+----------------+----------+----------+-------+
| ID                                   | Binary         | Host           | Zone     | Status   | State |
+--------------------------------------+----------------+----------------+----------+----------+-------+
| f15a3e03-f458-4213-8654-89ee98e956f1 | nova-scheduler | cos-ctrl-node1 | internal | enabled  | up    |
| 53c8b0c0-db73-4072-b4f0-ac89b6737c80 | nova-scheduler | cos-ctrl-node2 | internal | enabled  | up    |
| c41ccf40-ce9b-4df7-929f-1459684e0272 | nova-scheduler | cos-ctrl-node3 | internal | enabled  | up    |
| 98f1f222-65fd-4074-b02f-a2296f538b89 | nova-conductor | cos-ctrl-node2 | internal | enabled  | up    |
| 36ddd9f3-954e-4e85-baaf-c29d4a0ad7da | nova-conductor | cos-ctrl-node3 | internal | enabled  | up    |
| ecc97239-dfcd-4bb3-8e32-6bdb9745700b | nova-conductor | cos-ctrl-node1 | internal | enabled  | up    |
| a31f8d54-3190-4bab-9149-14949fa4bb5d | nova-compute   | cos-comp-node4 | AZ2      | enabled  | up    |
| 2637978f-bd9f-4a34-b72f-1ea92277bc05 | nova-compute   | cos-comp-node1 | AZ1      | disabled | up    |
| 29be0ae7-7bc7-4c9e-adf5-205c0ef06720 | nova-compute   | cos-comp-node5 | AZ2      | disabled | up    |
| 7556f893-2580-4422-b1f8-f6a043553834 | nova-compute   | cos-comp-node3 | AZ1      | enabled  | up    |
+--------------------------------------+----------------+----------------+----------+----------+-------+
```

The following command shows details of the GAD volume type used to test the deployment of VMs/disks.

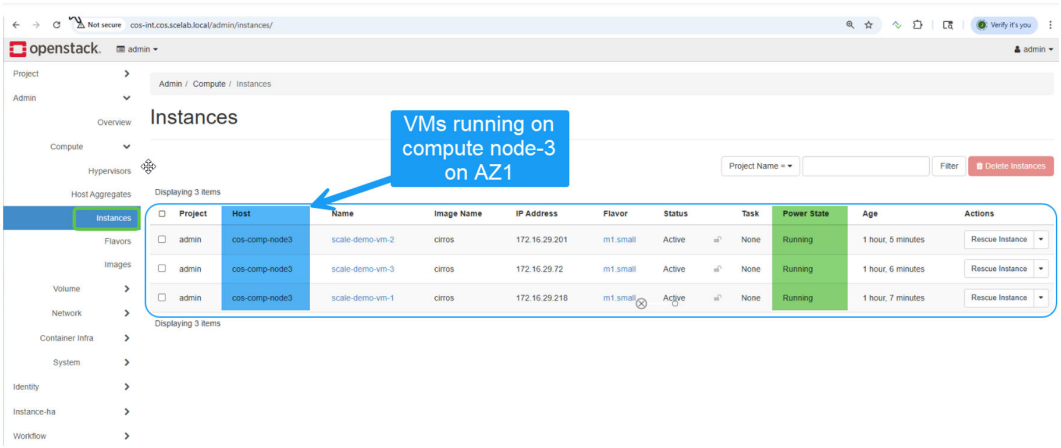
```
$ openstack volume type list
+--------------------------------------+---------------+-----------+
| ID                                   | Name          | Is Public |
+--------------------------------------+---------------+-----------+
| dcd1e6e7-75dc-4220-aa5d-dd7cd86f61af | vsp_5k_gad    | True      |
| 778031d7-6622-4855-ad5e-fb4128cddf37 | vsp_5500      | True      |
| 23a6bf80-a5d9-485a-8dc6-e5091088c5de | vsp_5600      | True      |
| 531d7d12-b07d-4b5d-a101-47ab71e0dbcd | vsp_1090      | True      |
| fb081ded-c995-486c-ba45-c9fa786c40ba | __DEFAULT__   | True      |
+--------------------------------------+---------------+-----------+

$ openstack volume type show vsp_5k_gad
+--------------------+---------------------------------------------------------------------------------------------+
| Field              | Value                                                                                       |
+--------------------+---------------------------------------------------------------------------------------------+
| access_project_ids | None                                                                                        |
| description        | None                                                                                        |
| id                 | dcd1e6e7-75dc-4220-aa5d-dd7cd86f61af                                                        |
| is_public          | True                                                                                        |
| name               | vsp_5k_gad                                                                                  |
| properties         | hbsd:topology='active_active_mirror_volume', volume_backend_name='hitachi_vsp_5k_gad'       |
| qos_specs_id       | None                                                                                        |
+--------------------+---------------------------------------------------------------------------------------------+
```

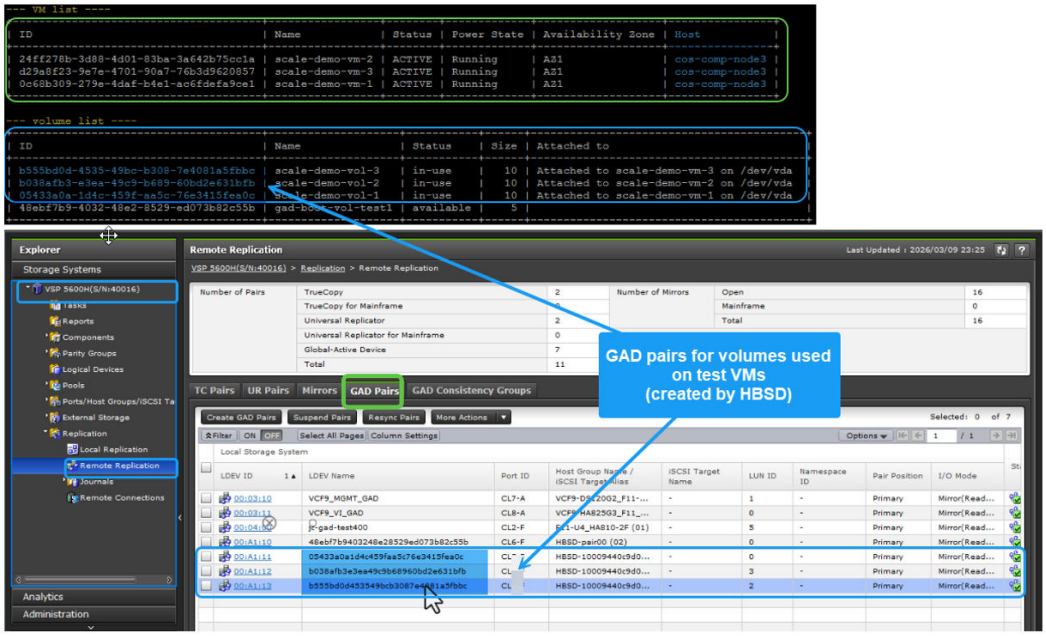
This command shows the cinder volume service enabled and running:

```
$ openstack volume service list
+------------------+-----------------------------------------+--------+---------+-------+
| Binary           | Host                                    | Zone   | Status  | State |
+------------------+-----------------------------------------+--------+---------+-------+
| cinder-scheduler | cos-ctrl-node1                          | nova   | enabled | up    |
| cinder-scheduler | cos-ctrl-node2                          | nova   | enabled | up    |
| cinder-scheduler | cos-ctrl-node3                          | nova   | enabled | up    |
| cinder-volume    | cos-comp-node1@hitachi_vsp_1090         | AZ1    | enabled | up    |
| cinder-volume    | cos-comp-node1@hitachi_vsp_5600         | AZ1    | enabled | up    |
| cinder-volume    | cos-cinder-ha@hitachi_vsp_5k_gad        | AZ_GAD | enabled | up    |
| cinder-volume    | cos-comp-node5@hitachi_vsp_5500         | AZ2    | enabled | up    |
+------------------+-----------------------------------------+--------+---------+-------+
```

2. VMs are deployed and running on compute node-3 in AZ1, using volumes provisioned on GAD volume type. GAD pairs are automatically created by HBSD, as shown.



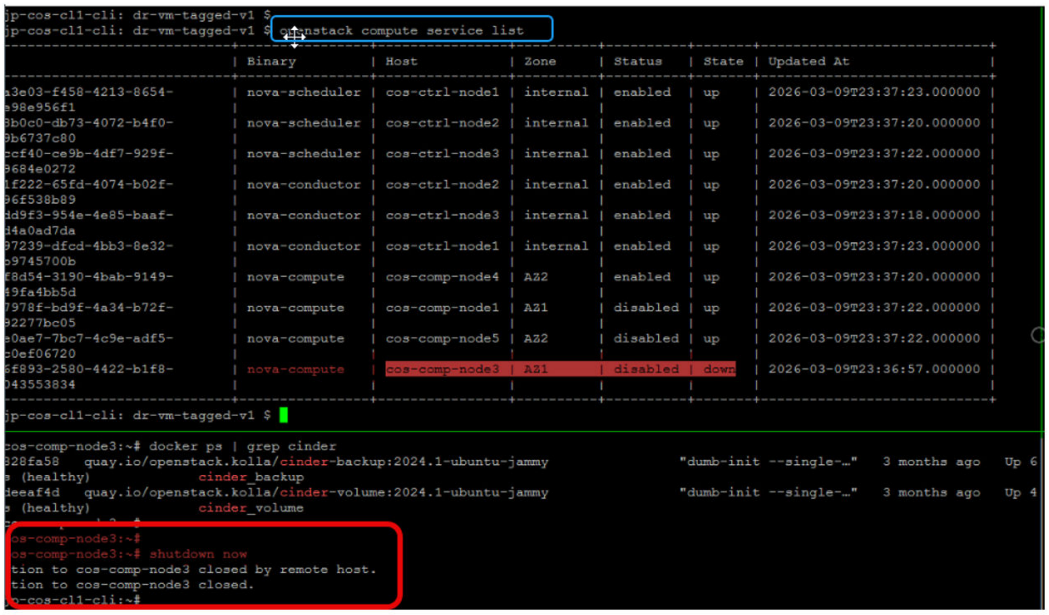
The following figure shows the corresponding GAD pairs for each of the disks (volumes) of the test VMs. HBSD provisions all the volumes and corresponding GAD pairs automatically.



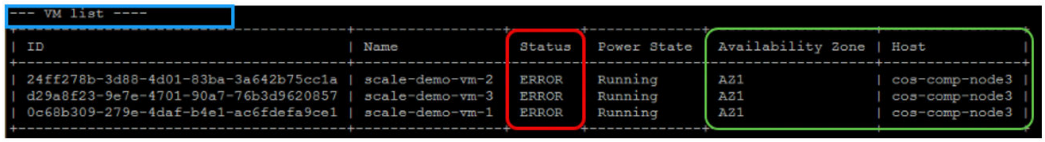
3. To simulate a DC failure, we did an ungraceful shutdown of the only active node (node-3) that was running the VMs in Site-1/AZ1.

> **Note:** Before the shutdown, some preparation tasks were completed:
>
> - VMs were tagged with the target AZ and the VM was labeled `dr-enabled`
> - A protection group of VMs was created to be recovered after AZ1 (node-3 in this case) goes down.

The following screenshot shows compute node-3 shut down, immediately after the nova-compute service is marked as disabled and in the “down” state.

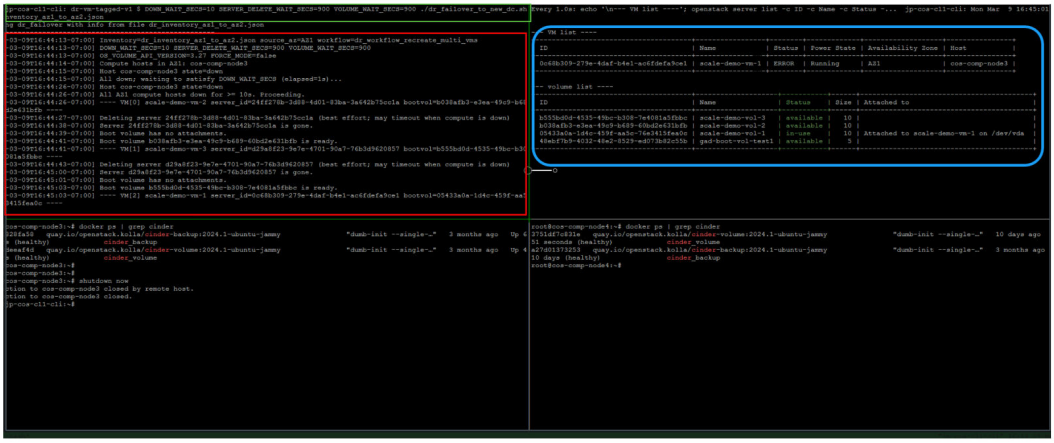


It could take up to 5 minutes for OpenStack to indicate that VMs are in the “ERROR” state after host failure.



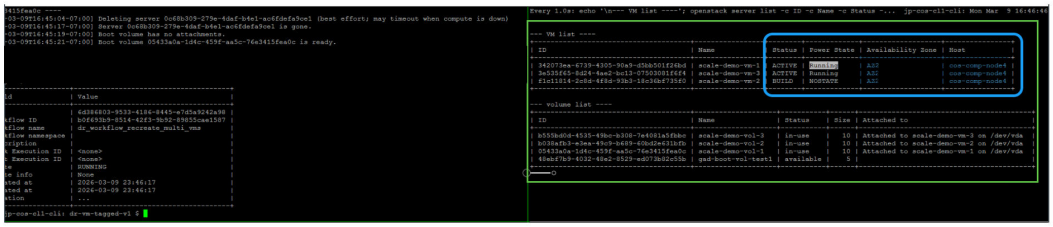
4. If Masakari service is running on the cluster, by default it tries to evacuate VMs, but because the only available nodes are in a different AZ, it will fail. Because of this, we use Hitachi’s developed orchestration to re-home/recover VMs on AZ2.
5. Run dr_failover to SITE-2/AZ2. This process uses the protection group (list of dr_enabled VMs) created in previous steps. The dr_failover process does the following:
   - Ensures all servers in the source availability zone (AZ1) are down.
   - Unregister dr_enabled VMs (after a VM has been unregistered, the volumes become “available”).

   This process can be seen in the following screen capture: dr_failover on the left side, and VMs getting unregistered and becoming available on the right side.

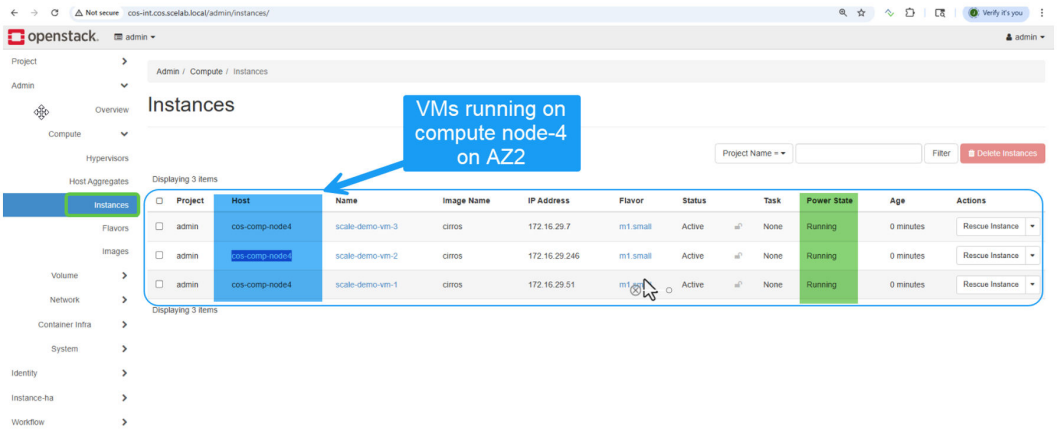


6. The dr_failover process automatically triggers the recovery of VMs on the availability zone AZ2 on SITE-2.

   The following screen capture shows VMs getting rebuilt and becoming active on compute node-4 on AZ2.



   This recovery process takes a few seconds/minutes depending on the number of VMs being recovered. In the OpenStack GUI we can also see the status of the recovered VMs. In this case we see VMs in the running state on compute node-4 in availability zone AZ2 on SITE-2.



Result

- The automation approach worked reliably and was demonstrated to be repeatable and suitable for larger environments. The failover process was successfully run for multiple VMs simultaneously.
- While not documented as part of this document, a failback works in the same way as a planned failover using the same orchestrated scripts:
  - Stop VMs
  - Shutdown compute nodes
  - Complete the `dr_failover` process
- Even when cross-AZ recovery is not natively automated by OpenStack, both manual recovery and scripted automation were validated as viable approaches for scalable failover and failback across AZs and data centers.
- To refine this cross-AZ recovery process into a customer's more prescriptive needs, Hitachi Vantara Professional Services are available.

Results summary

The following table summarizes the tested failure scenarios and their outcomes:

Failure Scenario	Behavior	Outcome
Compute host failure (single AZ)	Masakari evacuates VMs to surviving hosts	Minimal downtime (minutes); VMs recover automatically
Storage path failure	VMs pause because of I/O errors; can restart on secondary site	Recovery with VM restart/ unpause on surviving site
Inter-DC link failure	GAD pairs enter PSUE; hosts retain local access	No VM downtime; resync after link restoration
Site failure (single AZ)	All resources at one site lost; VMs evacuated to surviving site	Minimal downtime; GAD provides data continuity
Cross-AZ manual recovery	Manual evacuation to alternate AZ/site	Feasible with operational procedures
Cross-AZ automated failover	Scripted failover of multiple VMs across AZs	Reliable, repeatable, scalable

Product descriptions

The products described in this section are part of the solution.

Hitachi Virtual Storage Platform One Block

Hitachi Virtual Storage Platform One Block simplifies data management while enabling enterprises to reliability and sustainability meet initiatives to fuel growth and innovation.

Dynamic Carbon Reduction optimizes energy usage by switching CPUs to ECO mode during low activity. Adaptive Data Reduction (ADR) is always on, enhancing efficiency and reducing the overall CO2 footprint.

Thin Image Advanced (TIA) integrates with major snapshot ecosystems, supporting business continuity and defending against threats. CyberArk Privileged Access Manager plugins enhance block storage system security by prioritizing data confidentiality, ensuring compliance, and actively defending against security threats.

Hitachi Virtual Storage Platform One Block includes the following dedicated models:

- VSP One Block 24 – 256 GB Cache + SW Advanced Data Reduction (ADR) + 24 cores
- VSP One Block 26 – 768 GB Cache + 2x Compression Accelerator Module (CAM) + 24 cores

- VSP One Block 28 – 1 TB Cache + 4x CAM + 64 cores
- VSP One Block High End

  – 1 TB Cache + 6x CAM + 64 cores + data reduction with 4:1 ADR guaranteed

All models support Fibre Channel, iSCSI, and NVMe TCP connectivity. The new capabilities reduce complexity: data reduction is always on, Dynamic Drive Protection removes complicated RAID setup, and Dynamic Carbon Reduction delivers real-world reduction in power consumption. In addition, the models are FIPS compliant.

In short, Hitachi Virtual Storage Platform One Block combines simplicity, sustainability, and robust security features to optimize system management, energy efficiency, and data protection.

See https://www.hitachivantara.com/en-us/products/storage-platforms/block-storage/midrange/vsp-one-block for more information.

Hitachi Virtual Storage Platform One Block High End

Hitachi Virtual Storage Platform One Block High End (VSP One Block High End or VSP One B85) all-flash NVMe block storage systems deliver ultra-fast, highly reliable, and scalable data access for the most demanding enterprise applications.

Powered by Hitachi-engineered technology, they offer eight nines of availability, next-generation connectivity, comprehensive easy-to-use management, and seamless workload consolidation across open systems and mainframes. With end-to-end NVMe and best-in-class data reduction (4:1 ADR guaranteed), VSP One Block High End greatly enhances performance, scale and resilience in a simple, secure, sustainable way.

VSP One Block High End supports 2, 4, or 6-node configurations, and scales from 4 to 12 tightly coupled controllers with up to 288 NVMe SSDs.

See the *VSP One Block Specification Sheet* on the company website (http://www.hitachivantara.com) for more information.

Hitachi Virtual Storage Platform 5000 series

This enterprise-class, flash array evolution storage, Hitachi Virtual Storage Platform 5000 series has an innovative, scale-out design optimized for NVMe and storage class memory. It achieves the following:

- **Agility using NVMe:** Speed, massive scaling with no performance slowdowns, intelligent tiering, and efficiency.
- **Resilience:** Superior application availability and flash resilience. Your data is always available, mitigating business risk.
- **Storage simplified:** Do more with less, integrate AI (artificial intelligence) and ML (machine learning), simplify management, and save money and time with consolidation.

See https://www.hitachivantara.com/en-us/products/storage-platforms/block-storage/enterprise for more information.

Hitachi Advanced Server portfolio

The Hitachi Advanced Server portfolio delivers enterprise-class performance with flexible configurations to meet diverse deployment requirements. The portfolio includes multiple server families supporting both 2-socket and 4-socket configurations:

- HA8x0 G3, HA8x5 G3, HA8x0 G6 and DSx20 G6 series: Enterprise-grade 2-socket servers optimized for performance and efficiency with flexible memory and storage options.
- DS7000 series: Modular architecture supporting 2-socket and 4-socket configurations with exceptional scalability for complex workloads.

All server models provide the reliability, performance, and I/O flexibility required for demanding environments. Each series offers flexible expansion options and enterprise-grade features while allowing customers to select the optimal platform for their specific requirements.

Cisco Nexus switches

The Cisco Nexus switch product line offers a range of solutions that simplify the connection and management of disparate data center resources through software-defined networking (SDN). Leveraging the Cisco Unified Fabric, which unifies storage, data, and networking (Ethernet/IP) services, the Nexus switches create an open, programmable network foundation built to support a virtualized data center environment.

Brocade switches from Broadcom

Brocade and Hitachi Vantara have partnered to deliver storage networking and data center solutions. These solutions reduce complexity and cost, as well as enable virtualization and cloud computing to increase business agility.

Brocade Fibre Channel switches deliver industry-leading performance with seventh and eighth generation Fibre Channel interfaces, simplifying scale-out network architectures. Get the high-performance, availability, ease of management, and support for the next generation of Hitachi Virtual Storage Platform storage systems on a solid storage network foundation that can grow as your need grows.

See https://www.broadcom.com/products/fibre-channel-networking/switches for more information.

Conclusion

This paper presented a validated reference architecture for deploying a stretched OpenStack cluster with Hitachi Virtual Storage Platform (VSP) block storage, Global-Active Device (GAD) synchronous replication, and the Hitachi Block Storage Driver for OpenStack (HBSD) integrated with Cinder. Together, these components provide a resilient two-site infrastructure for OpenStack virtual machine workloads, combining synchronous data protection with operational recovery mechanisms at the compute, storage, and site levels.

The validation demonstrated that the architecture can recover from compute host failures through Masakari-driven instance evacuation, tolerate storage path failures through host multipathing, maintain data continuity during storage-system failures through GAD replication, and continue operating through inter-site link failures with controlled GAD pair suspension and later resynchronization. These results confirm that the design can deliver strong resilience for workloads deployed within a stretched single-AZ model while preserving the zero-data-loss characteristics associated with synchronous metro replication.

The paper also showed that a multi-AZ / host aggregate model is feasible for customers who require explicit site-aligned placement and recovery domains. In that model, OpenStack does not natively automate cross-AZ recovery in the same way that it supports recovery within a single AZ, but both manual operational recovery and scripted failover/failback workflows were validated as practical approaches for larger-scale environments. This remains a key architectural takeaway: GAD provides the storage continuity layer, but end-to-end recovery behavior across sites depends on how OpenStack placement, Masakari policy, and external orchestration are designed.

An important way to understand this architecture is to compare it with Hitachi GAD-based vMSC designs on VMware vSphere and VMware Cloud Foundation. From a storage architecture perspective, the pattern is very similar: two sites share a metro-synchronous storage foundation built on Hitachi GAD, with quorum-based arbitration and continuous access to protected storage objects across failure scenarios. Hitachi documents this pattern for both vSphere Metro Storage Cluster (vMSC) implementations and VCF stretched-cluster / metro-datastore designs using GAD-backed storage.

The primary difference is at the virtualization and orchestration layer. In vSphere / VCF, VM availability and placement are managed through VMware constructs such as the vMSC operational model, clustered datastores, and VMware HA–driven VM recovery behavior. In OpenStack, the same class of storage resilience is exposed through Cinder with HBSD, while VM placement and recovery are managed through Nova, Masakari, and Availability Zone / host-aggregate policy. In other words, the storage resiliency pattern is familiar to VMware administrators, but the control plane and recovery workflow are implemented using OpenStack-native services rather than VMware-native clustering constructs.

This same principle also aligns with Hitachi's stretched OpenShift reference architectures: the underlying GAD-based storage design remains fundamentally consistent, while the orchestration layer changes according to the platform. In OpenShift, the integration point is CSI-based persistent storage and Kubernetes operators; in OpenStack, it is HBSD, Cinder, Nova, and Masakari; and in vSphere / VCF, it is the vMSC / metro-datastore model with VMware-native VM lifecycle management. That consistency across platforms is one of the key strengths of Hitachi GAD as a metro-availability foundation.

In summary, this reference architecture demonstrates that a stretched OpenStack deployment with GAD and HBSD can provide a practical, repeatable, and scalable foundation for infrastructure resilience across two data centers. For readers familiar with VMware vMSC / VCF metro-storage architectures, this OpenStack solution can be viewed as the OpenStack-native equivalent of the same metro-resilient storage pattern: the same storage protection model, but integrated with OpenStack services and recovery workflows. That makes the architecture highly relevant for enterprises that want VMware-class metro-storage resilience while standardizing on OpenStack as their virtual infrastructure platform.

Appendix A: OpenStack configurations

The following are some of the configuration examples used for cinder/GAD-enabled backends and other configurations used on the OpenStack cluster. Replace IP addresses, serial numbers, users/passwords, ports, LDEV ranges, and other required settings with values that match your environment.

HBSD/Cinder example

The following shows a complete configuration example for Cinder, using a single baseline and host-specific overrides. Each file/section with annotations showing how each parameter maps to the storage resources configured in this paper. Change values according to your environment.

Single cinder.conf baseline, this contains all backend stanzas (GAD + non-GAD)

File name: …/config/cinder.conf

```
[DEFAULT]
enabled_backends=hitachi_vsp_5k_gad,hitachi_vsp_5600,hitachi_vsp_1090

## GAD backend
[hitachi_vsp_5k_gad]
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
volume_backend_name = hitachi_vsp_5k_gad

# Primary storage system (Site-1)
san_ip = <primary_storage_mgmt_ip>
san_login = <username>
san_password = <password>
hitachi_storage_id = <primary_serial_12_digits>
hitachi_pools = <primary_pool_id>
hitachi_ldev_range = 00:A1:10-00:A1:DB
hitachi_target_ports = CL1-A,CL2-A
hitachi_group_create = True

# GAD / Mirror configuration (Site-2)
hitachi_mirror_rest_api_ip = <secondary_storage_mgmt_ip>
hitachi_mirror_rest_user = <username>
hitachi_mirror_rest_password = <password>
hitachi_mirror_storage_id = <secondary_serial_12_digits>
hitachi_mirror_pool = <secondary_pool_id>
hitachi_mirror_ldev_range = 00:A1:10-00:A1:DB
hitachi_mirror_target_ports = CL1-A,CL2-A
hitachi_mirror_compute_target_ports = CL1-A,CL2-A

# GAD-specific parameters
hitachi_path_group_id = <path_group_id>
hitachi_quorum_disk_id = <quorum_disk_id>
hitachi_mirror_ssl_cert_verify = false
```

```
hitachi_mirror_rest_api_port = 443
hitachi_replication_copy_speed = 15

# Pair target for internal replication management
hitachi_pair_target_number = 0
hitachi_mirror_pair_target_number = 0

##Non-GAD backends

#FC-backend
[hitachi_vsp_5600]
volume_driver = cinder.volume.drivers.hitachi.hbsd_fc.HBSDFCDriver
volume_backend_name = hitachi_vsp_5600
san_ip = <storage_mgmt_ip>
san_login = <user>
san_password = <user-password>
hitachi_storage_id = 900000040ABC
hitachi_pools = 1
hitachi_target_ports = CL5-F,CL6-F
hitachi_group_create = True

#iSCSI-backend:
[hitachi_vsp_1090]
volume_driver = cinder.volume.drivers.hitachi.hbsd_iscsi.HBSDISCSIDriver
volume_backend_name = hitachi_vsp_5500_iscsi
san_ip = <storage_mgmt_ip>
san_login = <user>
san_password = <user-password>
hitachi_storage_id = 938000030XYZ
hitachi_pools = 2
hitachi_target_ports = CL2-E,CL3-C
hitachi_group_create = True
```

Host-specific overrides, this can be used to set enabled_backends for each cinder-volume node. In this case we are using this to enable the cinder volume active/passive cluster for GAD:

File name: …/config/cinder/cos-comp-node3/cinder.conf

```
[DEFAULT]
host = cos-cinder-ha
enabled_backends = hitachi_vsp_5k_gad
storage_availability_zone=AZ_GAD
```

File name: …/config/cinder/cos-comp-node4/cinder.conf

```
[DEFAULT]
host = cos-cinder-ha
enabled_backends = hitachi_vsp_5k_gad
storage_availability_zone=AZ_GAD
```

Volume type creation

The following commands can be used to create non-GAD volume types. This example uses backends for VSP E1090 and VSP 5600:

```
openstack volume type create vsp_1090
openstack volume type set --property volume_backend_name=hitachi_vsp_1090 vsp_1090
openstack volume type create vsp_5600
openstack volume type set --property volume_backend_name=hitachi_vsp_5600 vsp_5600
```

This example shows how to create a volume type for GAD volumes:

```
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
```

Use these commands to list and show details for specific volume types.

```
openstack volume type list
+--------------------------------------+----------------+-----------+
| ID                                   | Name           | Is Public |
+--------------------------------------+----------------+-----------+
| dcd1e6e7-75dc-4220-aa5d-dd7cd86f61af | vsp_5k_gad     | True      |
| 778031d7-6622-4855-ad5e-fb4128cddf37 | vsp_5500       | True      |
| 23a6bf80-a5d9-485a-8dc6-e5091088c5de | vsp_5600       | True      |
| 531d7d12-b07d-4b5d-a101-47ab71e0dbcd | vsp_1090       | True      |
| fb081ded-c995-486c-ba45-c9fa786c40ba | __DEFAULT__    | True      |
+--------------------------------------+----------------+-----------+

openstack volume type show vsp_5k_gad
+--------------------+-------------------------------------------------+
| Field              | Value                                           |
+--------------------+-------------------------------------------------+
| access_project_ids | None                                            |
| description        | None                                            |
| id                 | dcd1e6e7-75dc-4220-aa5d-dd7cd86f61af            |
| is_public          | True                                            |
| name               | vsp_5k_gad                                      |
| properties         | hbsd:topology='active_active_mirror_volume',    |
|                      volume_backend_name='hitachi_vsp_5k_gad'      |
| qos_specs_id       | None                                            |
+--------------------+-------------------------------------------------+
```

GAD volume creation and verification

To create a GAD boot volume, ensure that an image is available. In this case an image called `cirros` is used:

```
openstack volume create --size 10 --type vsp_5k_gad --availability-zone AZ_GAD --
image cirros gad-boot-vol1
```

This example shows how to create a standard/non-boot volume on GAD:

```
openstack volume create --size 10 --type vsp_5k_gad --availability-zone AZ_GAD test-
gad-volume
openstack volume show test-gad-volume
```

Upon successful creation, the volume should appear as 'available' and the HBSD driver will have automatically created a GAD pair with a P-VOL on the primary storage system and an S-VOL on the secondary storage system. You can verify the GAD pair status using Storage Navigator on either storage system.

Create a VM

This example shows how to create a VM with a GAD boot volume:

```
openstack server create --flavor m1.small --network demo-net --volume gad-boot-vol1 --
availability-zone AZ1 --wait test_vm_gad_1
```

Nova configurations

These settings will improve volume creation timeout:

File name: `nova.conf`

```
[DEFAULT]
block_device_allocate_retries = 300
block_device_allocate_retries_interval = 6
default_schedule_zone=nova
max_concurrent_builds=20
```

Masakari configurations

These parameters can be configured to control how VMs are evacuated by Masakari during a host failure:

`evacuate_all_instances` (boolean, default: `True`): Controls the scope of VM evacuation when a host failure is detected. When True, all instances on the failed host are evacuated, prioritizing those flagged with the HA metadata key before evacuating the rest. When False, only instances explicitly marked with the HA metadata key are evacuated, and unmarked instances are not evacuated.

`ha_enabled_instance_metadata_key` (string, default: `HA_Enabled`): Defines the instance metadata key used to identify which VMs are HA-eligible for evacuation. This key determines which instances receive priority (or exclusive) evacuation depending on the `evacuate_all_instances` setting. The default name can be overridden to differentiate behavior between host-failure and instance-failure scenarios.

For example:

File name: `masakari.conf`

```
[host_failure]
#ha_enabled_instance_metadata_key = HA_Enabled
evacuate_all_instances = True
```

Appendix B: Storage System Configuration using GAD

This appendix provides some of the storage system configuration steps required to prepare VSP storage systems for use with Global-Active Device (GAD) for OpenStack Cinder. The host-side Cinder/HBSD configuration is covered in the main body of this document and Appendix A: OpenStack configurations (on page 37).

In this example, we configure a VSP 5600 (primary, Site-1) and a VSP 5500 (secondary, Site-2) so that they can be used to create GAD-replicated Cinder volumes. The REST API servers for the primary and secondary storage systems are running at their respective management IP addresses.

The following procedures use the REST API. You can also use Storage Navigator or RAIDCOM for equivalent operations.

Note: The following procedures must be completed on both the primary and secondary storage systems unless otherwise stated. Replace IP addresses, serial numbers, session tokens, and resource IDs with values that match your environment.

Complete the following procedures to configure components in this solution:

- Create a Virtual Storage Machine (on page 42)
- Create a User Group and a User (on page 42)
- Create a pool (on page 44)
- Assign LDEVs to the VSM (on page 44)
- Assign Ports to the VSM (on page 45)
- Assign Host Group IDs (on page 45)
- Create remote copy paths (on page 46)
- Create a quorum disk (on page 47)

Create a Virtual Storage Machine

Create a VSM on each storage system with the same serial number and model type. The VSM provides a common virtual identity across both systems, which is required for GAD pair management. See Global Active Device for VSP One Block on the doc portal to verify the supported model types depending on the storage systems being used.

Primary storage system

The following REST API is an example of how to create a VSM, model type "VSP 5200H, 5600H " and a resource group named openstack_gad in the primary storage system.

```
curl --location \
  'https://<primary_mgmt_ip>/ConfigurationManager/v1/objects/virtual-storages' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "virtualSerialNumber": "40016",
    "virtualModel": "VSP 5200H, 5600H",
    "resourceGroupName": "openstack_gad"
  }'
```

Secondary storage system

The following REST API is an example how to create a VSM, model type VSP E1090, and resource group named openstack_gad in the secondary storage system.

```
curl --location \
  'https://<secondary_mgmt_ip>/ConfigurationManager/v1/objects/virtual-storages' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "virtualSerialNumber": "40316",
    "virtualModel": " VSP 5200H, 5600H ",
    "resourceGroupName": "openstack_gad"
  }'
```

Parameters

- `virtualSerialNumber`: Must be identical on both primary and secondary storage systems.
- `virtualModel`: Must match between both systems. Specify any storage system model supported by GAD.
- `resourceGroupName`: Name of the resource group associated with the VSM. Use the same name on both systems for simplified tracking.

Create a User Group and a User

Create a user group that has access permissions only for the resource group of the created VSM. Then create a user assigned to that user group. This user account will be used by Cinder (HBSD driver) to authenticate against the storage REST API.

Create a user group (primary storage system)

```
curl --location \
  'https://<primary_mgmt_ip>/ConfigurationManager/v1/objects/user-groups' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "userGroupId": "openstack_gad",
    "roleNames": [
      "Storage Administrator (Provisioning)",
      "Storage Administrator (Remote Copy)",
      "Storage Administrator (View Only)"
    ],
    "resourceGroupIds": [<resource_group_id>],
    "hasAllResourceGroup": false
  }'
```

Repeat on the secondary storage system with the corresponding resource group ID.

Parameters

- `roleNames`: The minimum set of roles required for GAD operations: Provisioning, Remote Copy, and View Only.
- `resourceGroupIds`: The ID of the resource group created for the VSM. Retrieve this using the REST API to get VSM details.
- `hasAllResourceGroup`: Must always be false to limit access to only the designated resource group.

Create a user (primary storage system)

```
curl --location \
  'https://<primary_mgmt_ip>/ConfigurationManager/v1/objects/users' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "userId": "<openstack_gad_user>",
    "authentication": "local",
    "userPassword": "<secure_password>",
    "userGroupNames": [
      "openstack_gad"
    ]
  }'
```

Repeat on the secondary storage system. Use the same username and password on both systems.

Parameters

- `userId`: The username for Cinder authentication. Corresponds to `san_login`/`hitachi_mirror_rest_user` in cinder.conf.
- `authentication`: Set to 'local' for storage-local authentication.

- `userPassword`: The password. Corresponds to `san_password`/ `hitachi_mirror_rest_password` in cinder.conf.
- `userGroupNames`: Must contain only the user group created above.

Create a pool

Create a Dynamic Provisioning (DP) pool from the pool volumes in the created virtual storage machines.

Note the following:

- Determine the size of the pool depending on how you use the stretched PV.
- Ensure that the capacity of the pool in the secondary storage system is at least the same size as the pool in the primary storage system.
- If the storage system already has unallocated pool volumes, you can use them to create a pool or you can create new pool volumes for the pool depending on your operation policy.

Assign LDEVs to the VSM

Assign the required number of unused LDEV IDs to the virtual storage machines on both storage systems. For the assigned LDEVs, change the virtual LDEV ID to unassigned (65534).

We confirmed that LDEV IDs from 41232 to 41432 are unused. Depending on your needs, you might have to assign more unused LDEV IDs to the VSM. The following REST APIs illustrates the operation for 41232. You will have to repeat this for other IDs as well.

These LDEVs will be used by HBSD to dynamically provision Cinder volumes.

Procedure

1. Delete the virtual LDEV ID of unused LDEVs

   Deleting the virtual LDEV ID sets it to 65534 (FF:FE in hexadecimal), which is required before assigning the LDEV to a VSM resource group.

```
curl --location \
  'https://<mgmt_ip>/ConfigurationManager/v1/objects/ldevs/41232/actions/\
    unassign-virtual-ldevid/invoke' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "parameters": {
      "virtualLdevId": 41232
    }
  }'
```

   Repeat for each unused LDEV ID that needs to be assigned. On the lab setup, a script was used to iterate across multiple virtualLdevIds.

2. Assign unused LDEV IDs to the VSM resource group.

```
'https://<mgmt_ip>/ConfigurationManager/v1/objects/resource-groups/\
    <resource_group_id>/actions/add-resource/invoke' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "parameters": {
      "ldevIds": [<ldev_id_1>, <ldev_id_2>, ...]
    }
  }'
```

Note: If you are using the `hitachi_ldev_range` parameter in `cinder.conf`, ensure that virtual LDEV IDs are set for every LDEV in the specified range on the primary storage system, because virtual LDEV IDs are required for GAD pair creation.

Assign Ports to the VSM

Add each of the Fibre Channel ports used for data paths to the VSM resource group.

The following REST API assigns the host group to the resource group of the VSP created in the primary storage system.

```
curl --location \
  'https://<primary_mgmt_ip>/ConfigurationManager/v1/objects/resource-groups/
<resource_group_id>/actions/add-resource/invoke' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
    "parameters": {
      "portIds": [
        "CL1-A",
        "CL2-A"
      ]
    }
  }'
```

Repeat on the secondary storage system with the corresponding port IDs.

Parameter

- `portIds`: A list of ports in the format `CLx-y`.

Assign Host Group IDs

For each Fibre Channel port used for data paths, assign the required number of unused host group IDs to the VSM resource group. The number of host group IDs must be at least the total number of compute hosts plus one.

Note: If you set `hitachi_group_create = True` in cinder.conf, the HBSD driver will create host groups automatically as needed. However, the host group IDs must still be available in the VSM resource group.

The following REST API assigns the host group to the resource group of the VSP created in the primary storage system.

```
curl --location \
  'https://<primary_mgmt_ip>/ConfigurationManager/v1/objects/resource-groups/
<resource_group_id>/actions/add-resource/invoke' \
  --header 'Authorization: Session <session_token>' \
  --header 'Content-Type: application/json' \
  --data '{
  "parameters": {
    "hostGroupIds": [
      "CL1-A,1",
      "CL1-A,2",
      "CL1-A,3",
      "CL2-A,1",
      "CL2-A,2",
      "CL2-A,3"
    ]
  }
}'
```

Repeat on the secondary storage system with the corresponding port IDs and host group IDs.

Parameter

- `hostGroupIds`: A list of port and host group ID pairs in the format `CLx-y,n`. The number of host group IDs per port must be at least (number of compute hosts + 1).

Create remote copy paths

Create remote copy paths between the primary and secondary storage systems. The same path group ID must be used on both sides. Remote copy paths carry the synchronous GAD replication traffic.

Both Fibre Channel and IP (iSCSI) networks are supported for remote copy paths:

- Fibre Channel: Connect dedicated FC ports on each storage system (preferably multiple for redundancy) to the SAN switches in their respective sites.
- IP Network (iSCSI): If inter-site Fibre Channel connections are unavailable, use iSCSI ports connected to the LAN switches in their respective sites.

In the lab, FC ports 1B and 2B on the VSP 5600 and ports 1B and 2B on the VSP 5500 were used to create a remote copy path group using Storage Navigator.

Note: The path group ID configured here must match the `hitachi_path_group_id` parameter in `cinder.conf`.

Create a quorum disk

Create a quorum disk for GAD arbitration. The same quorum disk ID must be used on both the primary and secondary storage systems. The quorum disk is used by GAD to determine which volume copy remains active in split-brain failure scenarios.

Options for quorum disk provisioning

- GAD Cloud Quorum (AWS Marketplace): A managed service that provides quorum functionality without requiring a third physical storage system. This is the recommended approach for two-site deployments. Deploy from the AWS Marketplace and configure both storage systems to use the cloud quorum endpoint.
- Third storage system: A quorum disk can also be provisioned on a third Hitachi VSP storage system at a separate site. This provides hardware-based arbitration.

Note: The quorum disk ID configured here must match the `hitachi_quorum_disk_id` parameter in `cinder.conf`.

Reference documents

See the following reference documents for more information:

- On the Hitachi Vantara Product Documentation portal (docs.hitachivantara.com)
  - *Global Active Device for VSP One Block*
  - *Virtual Storage Platform Block Storage REST API Reference*
  - *Virtual Storage Platform Best Practices for OpenStack Environments*
- On the OpenStack website (docs.openstack.org/cinder/latest/)
  - *Hitachi block storage driver*

Hitachi Vantara

Corporate Headquarters
2535 Augustine Drive
Santa Clara, CA 95054 USA

HitachiVantara.com/contact