Hitachi Virtual Storage Software

1.12

Block VSP Administrator Guide

This guide describes and provides instructions for performing administration tasks for Hitachi Virtual Storage Software block (VSS block) on multi-tenancy systems.

Contents

Contents

Contents

Preface

This guide describes and provides instructions for performing system administration tasks for Hitachi Virtual Storage Software block (VSS block)mulit-tenancy.

This manual applies to both the virtual machine and bare metal models of VSS block.

- Sections in this manual marked with (Virtual machine) apply to the virtual machine model.

- Sections in this manual marked with (Bare metal) apply to the bare metal model.

Please read this document carefully to understand how to use these products and maintain a copy for reference purposes.

Intended audience

This guide describes and provides instructions for performing administration tasks for Hitachi Virtual Storage Software block (VSS block) on multi-tenancy systems.

This document is intended for VPD administrators, Hitachi Vantara representatives, and authorized service providers who are involved in installing, configuring, and operating multi-tenancy Virtual Storage Software block.

Readers of this document should have at least the following knowledge and experience:

Skills using CLI and REST API

Product version

This document revision applies to Virtual Storage Software block version 1.12 (01.12.0*x.xx*).

The version in this document is described only by [aa.bb], and [aa.bb.cc.dd] is used only when required.

Release notes

Read the release notes before installing and using this product. They may contain requirements or restrictions that are not fully described in this document or updates or corrections to this document. Release notes are available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents.

Changes made in this revision

Newly created to support the multi-tenancy function.

Document conventions

This document uses the following typographic conventions:

Convention	Description
Bold	▪ Indicates text in a window, including window titles, menus, menu options, buttons, fields, and labels. Example: Click **OK**. ▪ Indicates emphasized words in list items.
Italic	▪ Indicates a document title or emphasized words in text. ▪ Indicates a variable, which is a placeholder for actual text provided by the user or for output by the system. Example: `pairdisplay -g group` (For exceptions to this convention for variables, see the entry for angle brackets.)
`Monospace`	Indicates text that is displayed on screen or entered by the user. Example: `pairdisplay -g oradb`
< > angle brackets	Indicates variables in the following scenarios: ▪ Variables are not clearly separated from the surrounding text or from other variables. Example: `Status-<report-name><file-version>.csv` ▪ Variables in headings.
[ ] square brackets	Indicates optional values. Example: [a | b] indicates that you can choose a, b, or nothing.
{ } braces	Indicates required or expected values. Example: { a | b } indicates that you must choose either a or b.
| vertical bar	Indicates that you have a choice between two or more options or arguments. Examples: [a | b] indicates that you can choose a, b, or nothing. { a | b } indicates that you must choose either a or b.

This document uses the following icons to draw attention to information:

Icon	Label	Description
	Note	Calls attention to additional information.
	Tip	Provides helpful information, guidelines, or suggestions for performing tasks more effectively.
	Important	Highlights information that is essential to the completion of a task.
	Caution	Warns the user of adverse conditions and/or consequences (for example, disruptive operations, data loss, or a system crash).
	CAUTION	Warns the user of a hazardous situation that, if not avoided, could result in major or minor injury.
	WARNING	Warns the user of a hazardous situation which, if not avoided, could result in death or serious injury.

Accessing product documentation

Product user documentation is available on the Hitachi Vantara Support Website: https://knowledge.hitachivantara.com/Documents. Check this site for the most current documentation, including important updates that may have been made after the release of the product.

Getting help

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

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

Comments

Please send comments to doc.comments@hitachivantara.com. Include the document title and number, including the revision level (for example, -07), and refer to specific sections and paragraphs whenever possible. All comments become the property of Hitachi Vantara LLC.

Thank you!

Chapter 1: Overview of Hitachi Virtual Storage Software block

About Hitachi Virtual Storage Software block

Virtual Storage Software block is a storage software product that builds and sets up a virtual storage system from multiple general-purpose servers.

The operation and management features of the Virtual Storage Software block system are as follows. The system offers a high-performance, high-capacity block storage service with high reliability. HPEC stores user data on a local drive.

- The initial cost is low because general-purpose servers (x86 servers) are used.

- You can quickly introduce the system, add storage, or reduce storage.

- You can centrally manage multiple storage nodes as a single storage system.

- You can easily determine the operation status of resources, the maximum volume capacity that can be created, and I/O performance at the time of checking.

- The REST APIs for Virtual Storage Software block are compatible with the REST APIs for Hitachi Storage Advisor Embedded. Administrators with experience in managing Hitachi storage products can manage storage using Virtual Storage Software block without extra time on learning Virtual Storage Software block.

System configuration

The system built by Virtual Storage Software block is a separate Software Defined Storage (SDS) that separates the storage node from the compute node.

Virtual Storage Software block's storage system consists of multiple storage nodes. A storage node consists of multiple drives.

Operations require the registration of compute nodes running by applications accessing the storage system .

Also, the controller node from which management operations of the storage system can be performed (by using the REST API or CLI) is installed by system administrators.

Overview of multi tenancy

Multi tenancy is a function to allow resources of a storage in a large storage system to be distributed to and shared by multiple tenants (companies and divisions).

The multi-tenancy function of Virtual Storage Software block is intended for cases where other tenants can be trusted, such as when a company-owned storage system is shared by multiple departments within the company.

The system configuration that can be perceived by a VPS administrator is limited to the area that the VPS administrator belongs to.
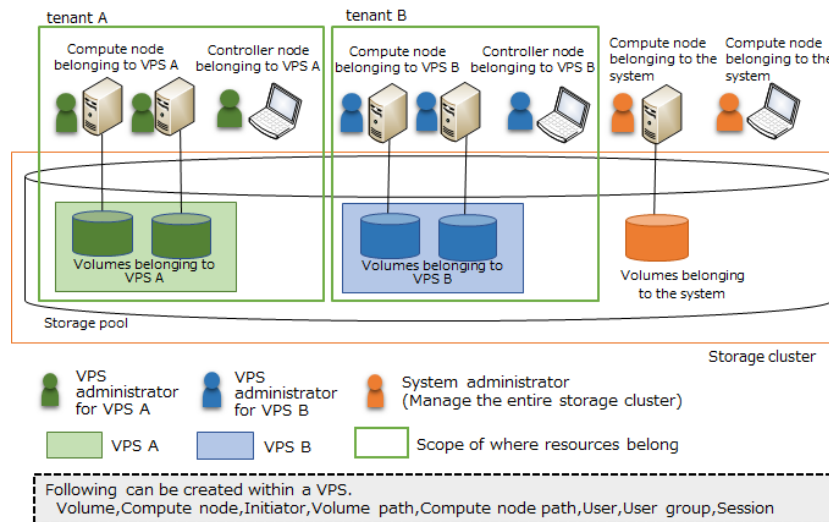
VPS administrator

An administrator who manages a VPS in multi tenancy configuration is called a VPS administrator.

Each VPS is created by a system administrator. VPS administrators can use storage software functions under the defined conditions.

After VPS creation, assign tasks for system administrators and VPS administrators as follows.

Where resources belong	Category	Administrator in charge
System (outside of a VPS)		System administrator
Within a VPS	Normal operation	VPS administrator
	Abnormal operation	System administrator and VPS administrator (acting cooperatively)

> 📄 **Note:**
>
> VPS administrators cannot log in to the GUI. Furthermore, system administrators cannot perform operations with resources that belong to a VPS from the GUI.

Roles for VPS administrators

The roles VpsSecurity, VpsStorage, and VpsMonitor are provided for VPS administrators.

The following table shows the main operations of VPS administrators for each role.

Role	Main operations to be performed
VpsSecurity	Administrators with this role manage users within a VPS they are in charge of. VPS administrators with this role can perform management operations for users, user groups, or sessions as required for a security administrator.
VpsStorage	Administrators with this role manage capacity within a VPS they are in charge of. VPS administrators with this role can perform management operations for the following resources as required for a security administrator. ▪ Volume (including snapshot volumes) ▪ Compute node ▪ Allocating volumes to a compute node (volume path) ▪ Access path for a compute node (compute node path) ▪ Initiator
VpsMonitor	VPS administrators with this role can obtain information about the following resources in a VPS they are in charge of. ▪ Volume (including snapshot volumes) ▪ Compute node ▪ Allocating volumes to a compute node (volume path) ▪ Access path for a compute node (compute node path) ▪ Initiator

Resources that can be created and where they belong

Resources that can be created in a VPS are listed further on in this section. Created resources belong to the same VPS as the user who created the resources. The resources are given the ID and attribute of the VPS to which they belong.

- Volume (including snapshot volumes)

- Compute node

- Allocating volumes to a compute node (volume path)

- Access path for a compute node (compute node path)

- Initiator

- User

- User group

- Session

Scope

A scope is used to specify the range of resources users can operate. A scope is set for user groups, and a user's scope is determined by which user group they belong to.

By default, the scope of user groups for VPS administrators is set with the VPS to which a user who created the VPS belongs.

Chapter 2: Configuring the operating environment

Installing a compute node

A compute node is a node that runs the application of the user and instructs input/output of user data to the storage node. The compute node is connected to the storage system through the compute network.

After the connection is complete, confirm the following requirements and schedule the tasks. Then, register the information about the compute node according to *Registering information about compute nodes* in this manual.

The following table describes the requirements for compute nodes.

For details about how to enable ALUA, see *Appendix A. ALUA configuration guidelines*. For details about other items, see the documentation for your OS.

Item	Requirement
Supporting and enabling ALUA	Asymmetric logical unit access (ALUA) must be supported. In addition, ALUA must be enabled.
SCSI timeout setting	The SCSI timeout setting of the compute node must be 120 seconds or longer to prevent I/O operations from stopping if a node failure occurs.[1]
Multipath setting (Linux DM-Multipath)	Meet all of the following: ▪ The retry count setting (no_path_retry) for the same path is 6 or more. ▪ The path polling interval (polling_interval) setting value is 30 or more. ▪ The setting value of fast_io_fail_tmo is other than off.[2] ▪ The setting value of dev_loss_tmo is the OS maximum value.[3] ▪ The failback setting is "immediate".[4] ▪ The path_checker setting is readsector0.[5]

Item	Requirement
1. The SCSI Timeout setting value is the time until the OS determines that a SCSI command has not responded. When the OS detects no response to the SCSI command, it performs recovery processing for the failed path. If the failed path cannot be recovered by the recovery process, the path is blocked and switched to another path. During the recovery process, I/O operations for the relevant path are stopped. Therefore, in the application layer timeout design, it is necessary to consider the recovery processing time in addition to the SCSI Timeout setting value.	
2. The default value for fast_io_fail_tmo is 5. When set to off, path switching does not occur until dev_loss_tmo seconds have elapsed when a path failure occurs, and it does not operate as expected.	
3. The maximum value depends on your distribution. See your distribution manual for the maximum value.	
4. Set the failback policy ("failback") of the path group to "immediate" (immediate failback). If the failback policy is not set to "immediate", I/O will still be issued to the non-priority path after recovery from a path fault, thus requiring a manual path switch operation.	
5. If a failure occurs in a storage node of Virtual Storage Software block, when a value other than readsector0 is set, paths might be wrongly blocked. For this reason, make sure that you set readsector0.	

Virtual Storage Software block supports the following VMware vStorage APIs for Array Integration (VAAI) primitives.

VAAI primitive	Description	Support status
Block Zeroing	When a virtual machine is created, blocks are formatted (zeroed out). This primitive enables the storage to perform this procedure, speeding up provisioning of virtual machines. When used with the thin provisioning feature, the primitive can free the block areas that are usually reserved when a virtual machine is created, enabling efficient use of disk capacity.	Supported
Hardware Assisted Locking	If multiple virtual machines share one VMFS volume in your environment, SCSI reservation conflicts might occur and performance might degrade when Storage vMotion or a virtual machine is turned on. With this primitive, you can avoid such a problem. The primitive offers efficient locking.	Supported

VAAI primitive	Description	Support status
Full Copy	Traditionally, VMware ESXi copy data between volumes. This primitive offloads the operation to the storage, greatly reducing the time required for virtual machine cloning and Storage vMotion processing.	Not supported

Notes on installing compute nodes

- In the case of iSCSI connection, you can use the iSCSI Data Digest function and iSCSI Header Digest function by enabling the iSCSI Digest settings on the compute node. Using these functions improves data reliability on the communication paths but degrades performance. To what extent the performance degrades depends on the environment of hosts, networks, and other components. Determine whether to use these functions based on your purpose.

- (Virtual machine) In the case of FC connection, if a storage node failure occurs, SCSI commands might return no response in the path to the faulty storage nodes. In this case, in addition to the timeout setting value for switching from the failed path to another path, recovery processing time is required, such as discarding SCSI commands that become unresponsive.

Scheduling the tasks

Virtual Storage Software block narrows the number of passes on the storage side so as not to violate the Windows MultiPath I/O limit.

Introducing tasks to compute nodes (hosts) with Windows is essential to simplify operations against this limitation. For compute nodes with Windows, follow the steps below to register the task scheduler.

The task scheduler registers for initial setup and re-execution by restart. The task scheduler periodically updates the device, allowing it to follow configuration information after failure and capacity balancing.

> ⚠️ **Caution:**
>
> Schedule the task so that the task is run when the compute node is logged out for maintenance, or at login and restart after an unexpected shutdown.

Procedure

1. From the start menu, select [Administrative Tools], and then [Task Scheduler].

   The [Task Scheduler] windows appears.
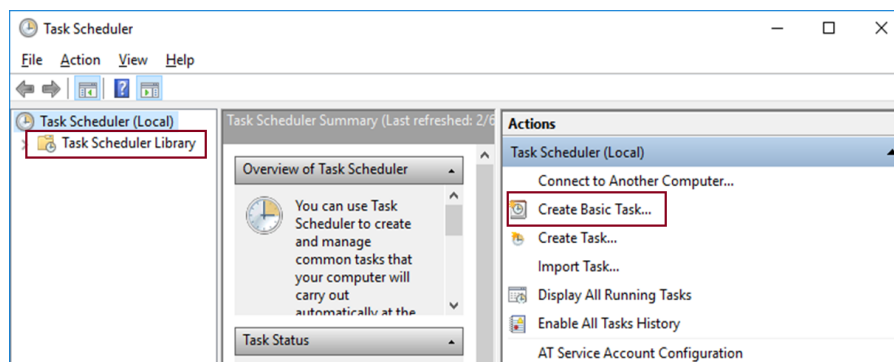
2. Select [Task Scheduler Library] in the left side of the window, and then click [Create Basic Task] on the right side of the window.

The [Create Basic Task] windows appears.
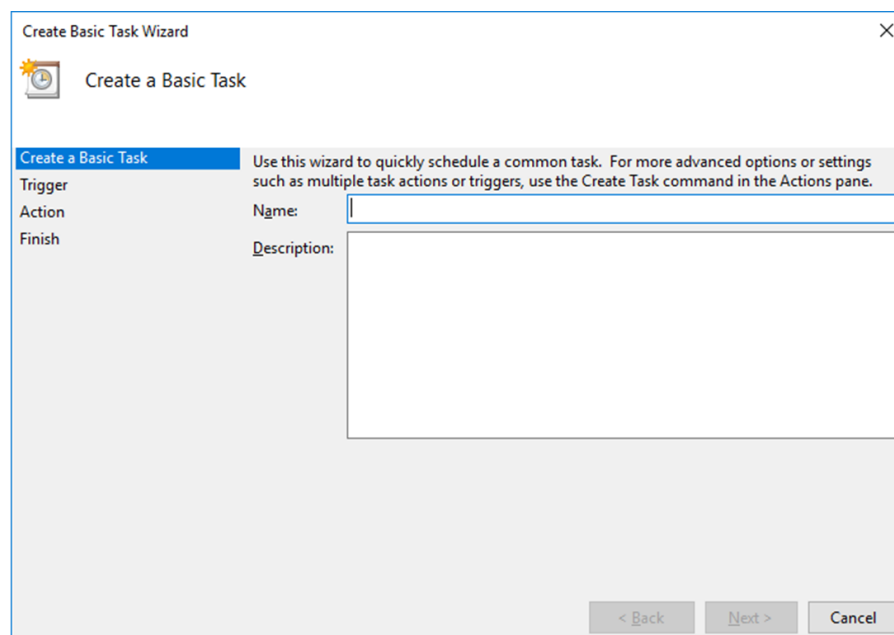
3. Enter a name in [Name] and a description in [Description], and then click [Next].

The [Task Trigger] windows appears.

Create Basic Task Wizard   ✕

    Task Trigger

Create a Basic Task
Trigger
Action
Finish

When do you want the task to start?

○ Daily
○ Weekly
○ Monthly
○ One time
○ When the computer starts
◉ When I log on
○ When a specific event is logged

     < Back   Next >   Cancel
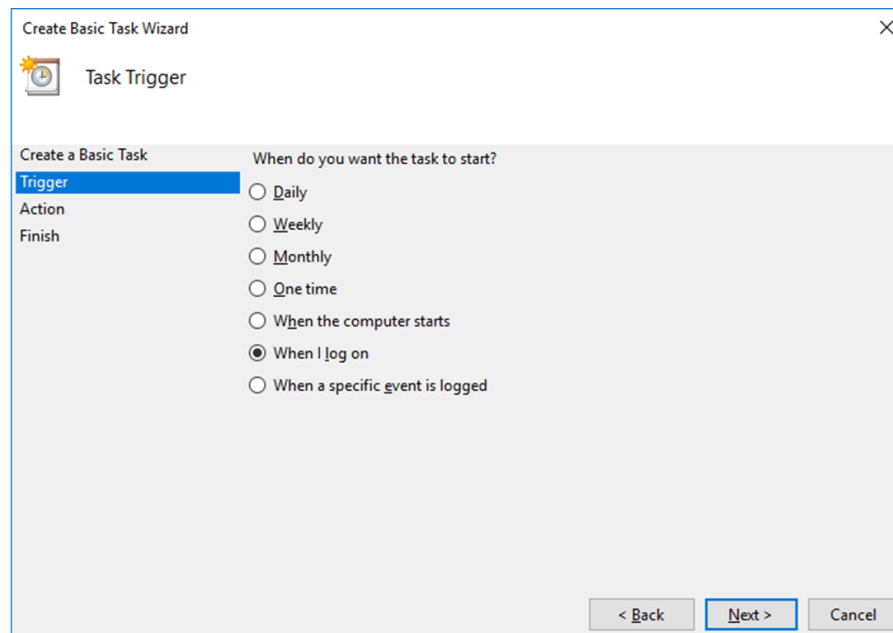
4. Select [When I log on], and then click [Next].

The [Action] windows appears.

Create Basic Task Wizard   ✕

    Action

Create a Basic Task
Trigger
Action
Finish

What action do you want the task to perform?

◉ Start a program
○ Send an e-mail (deprecated)
○ Display a message (deprecated)

     < Back   Next >   Cancel
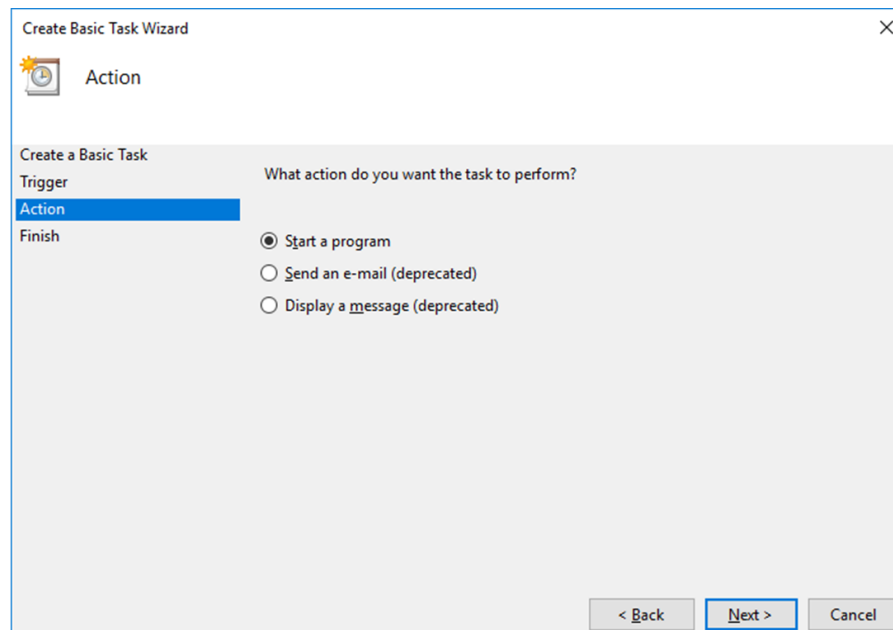
5. Select [Start a program], and then click [Next].

The [Start a Program] windows appears.

Chapter 2: Configuring the operating environment
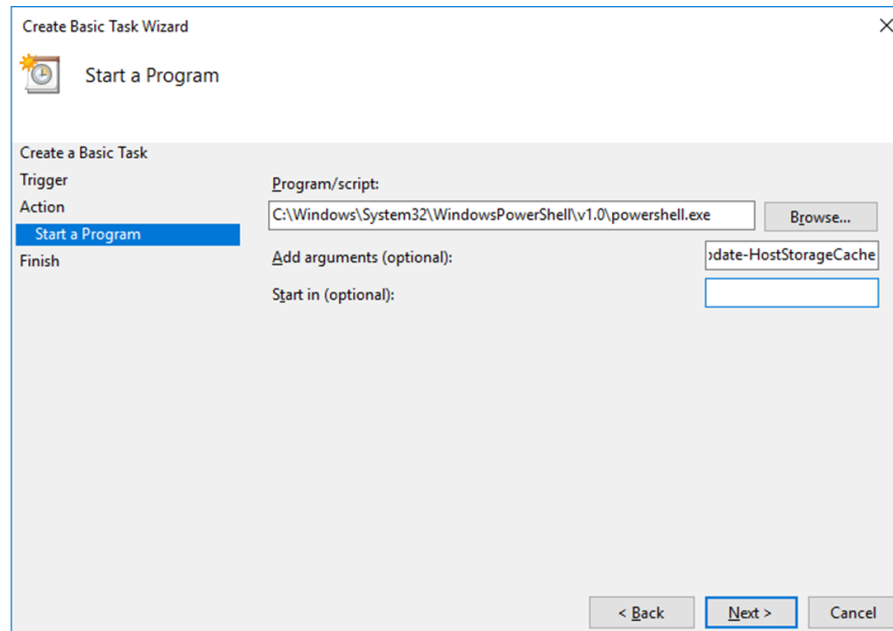
6. Specify the following, and then click [Next].

   ▪ Program/script: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

   ▪ Add arguments: -windowstyle hidden Update-HostStorageCache

   The [Summary] windows appears.
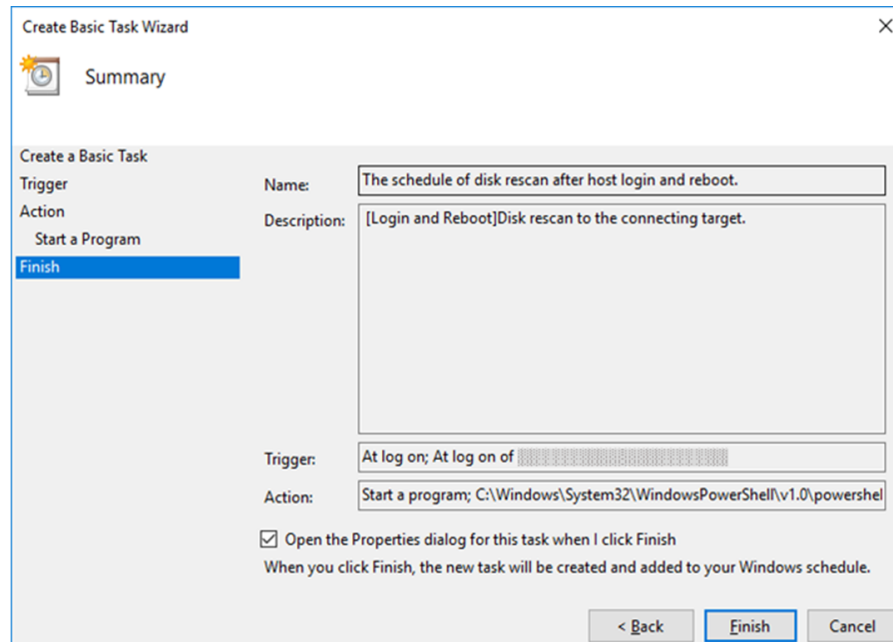
7. Verify the content, select [Open the Properties dialog for this task when I click Finish], and then click [Finish].

   The Properties window for the task appears.

   The property settings should be set according to the operation. As a reference, here's an example of the setting.

Chapter 2: Configuring the operating environment

On the [General] tab, set security options.

Select between [Run only when user is logged on] or [Run whether user is logged on or not], depending on your operation.

If [Run only when user is logged on] is set, you may not be able to follow the storage-side update while logging out, resulting in access lost.

If the previously specified user accesses the volume while logged out, select [Run whether user is logged on or not] and check [Do not store password. The task will only have access to local computer resource.].

[Run with highest privileges] does not require a check. [Hidden] does not need to be checked. Set the default value for "Configuration for:".
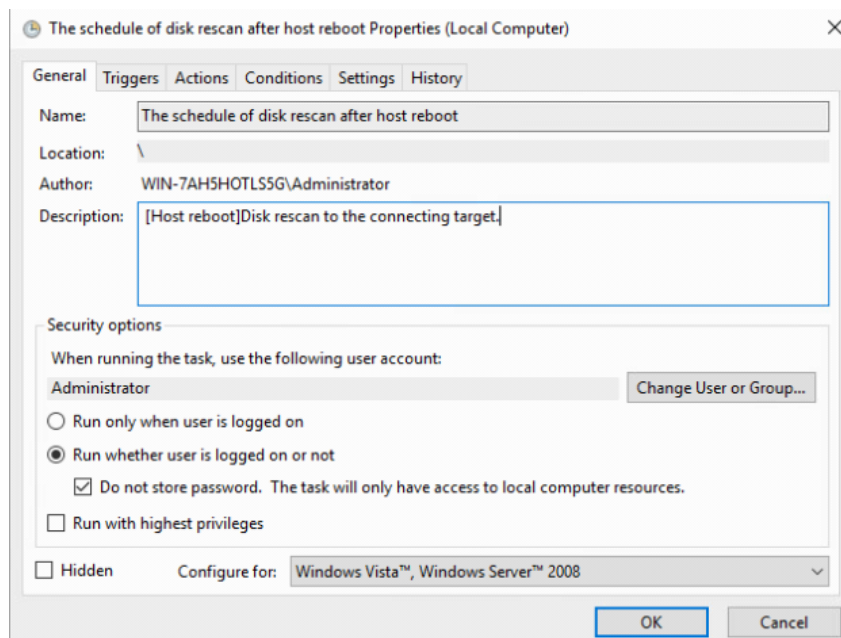
> 📄 **Note:**
>
> As a user specified as a trigger, you can click [Edit], and then specify the login user of the Windows host server (used as the compute node) as required.
>
>
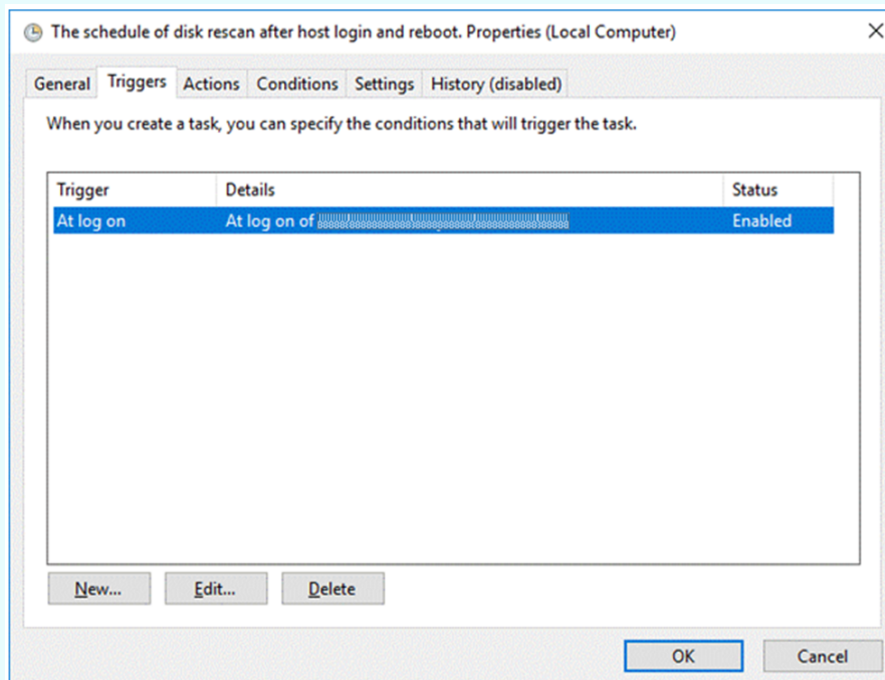
8. Select the [Triggers] tab, select a trigger, and then click [Edit].

   The [Edit Trigger] windows appears.

9. Specify the following, and then click [OK].

    ■   Repeat task every: 5 minutes

    ■   for a duration of: Indefinitely

    Other settings should be set according to the operation.

    The Properties window for the task reappears.

10. Click [OK].

    The [Task Scheduler] windows reappears.

11. Next, create a task to be run initially. Select the created task, and then select [Export] from the right-click menu to save the task.

    The exported task can be saved in any location.
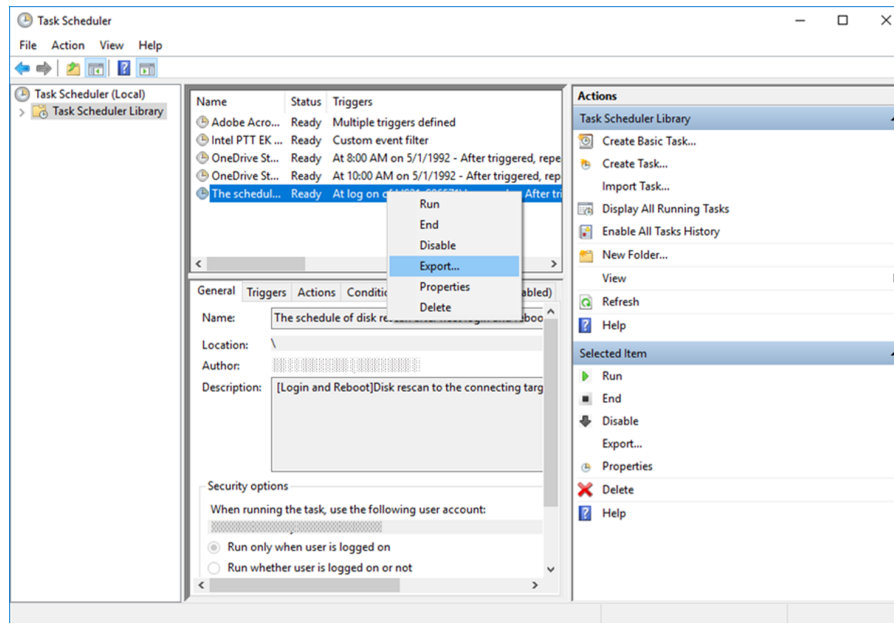
12. Import the exported task. From the [Actions] menu in the [Task Scheduler] window, select [Import Task], and then specify the task saved in step 11.

This scheduler triggers the task to run in the current login state.
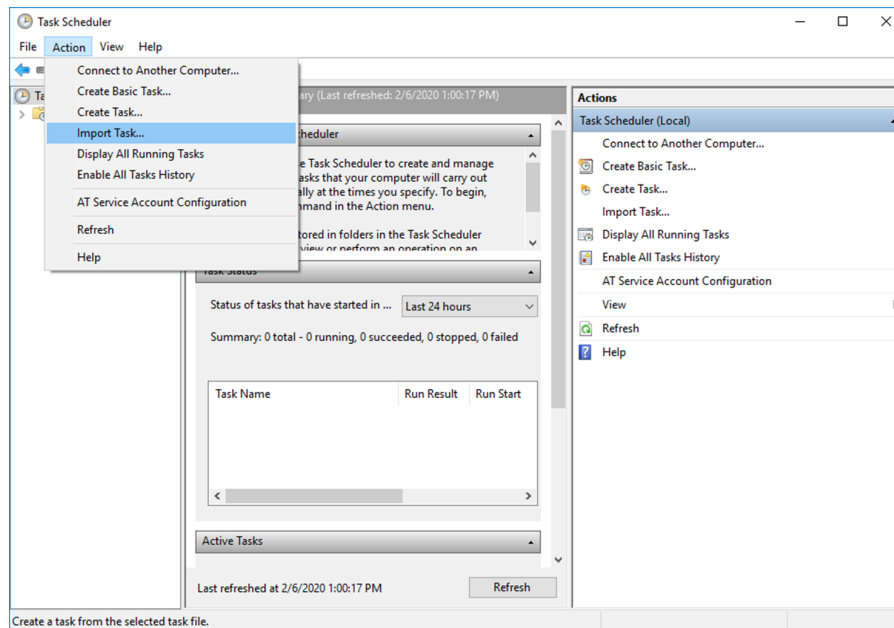
The [Create Task] windows appears.
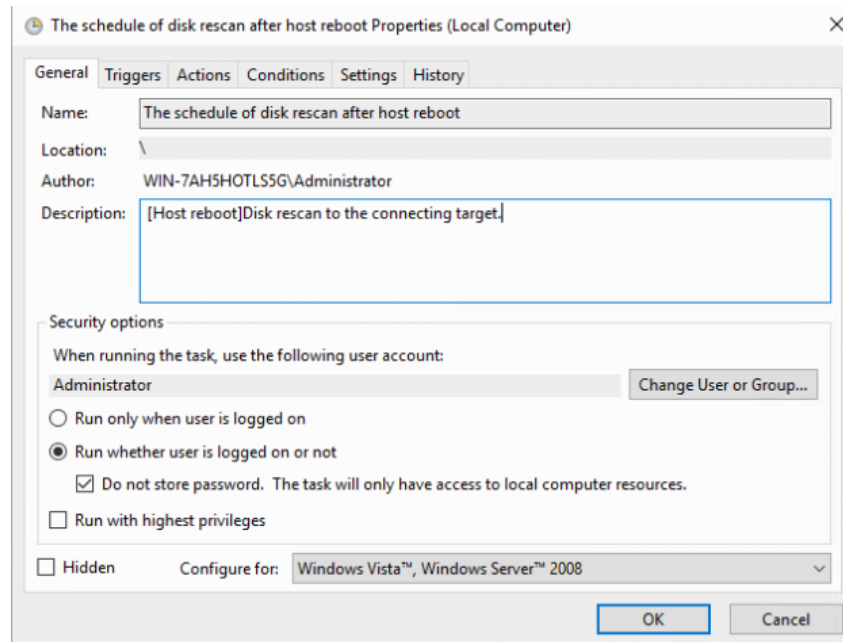
13. In the [General] tab, change the name in [Name] and the description in [Description].
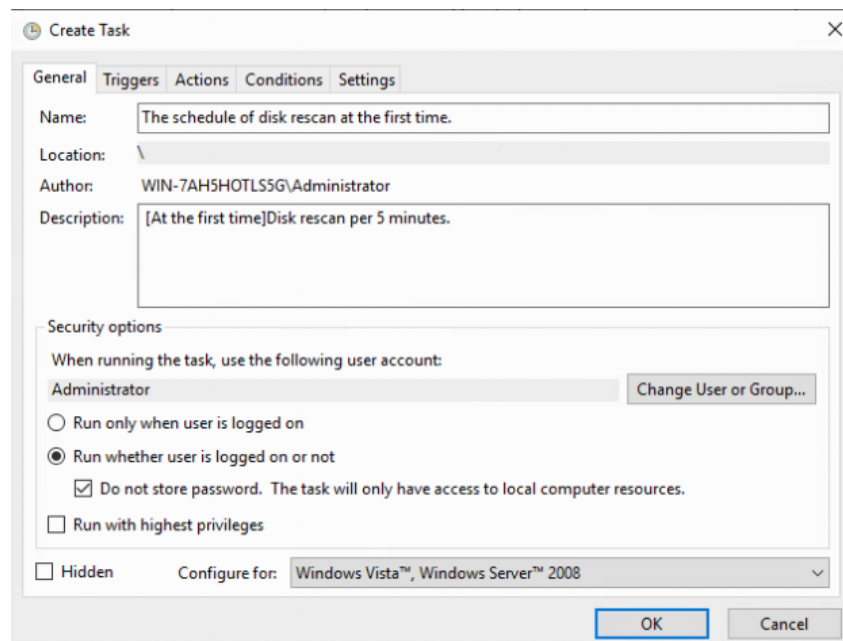
14. Select the [Triggers] tab, select the task you changed in step 13, and then click [Edit].

    The [Edit Trigger] windows appears.
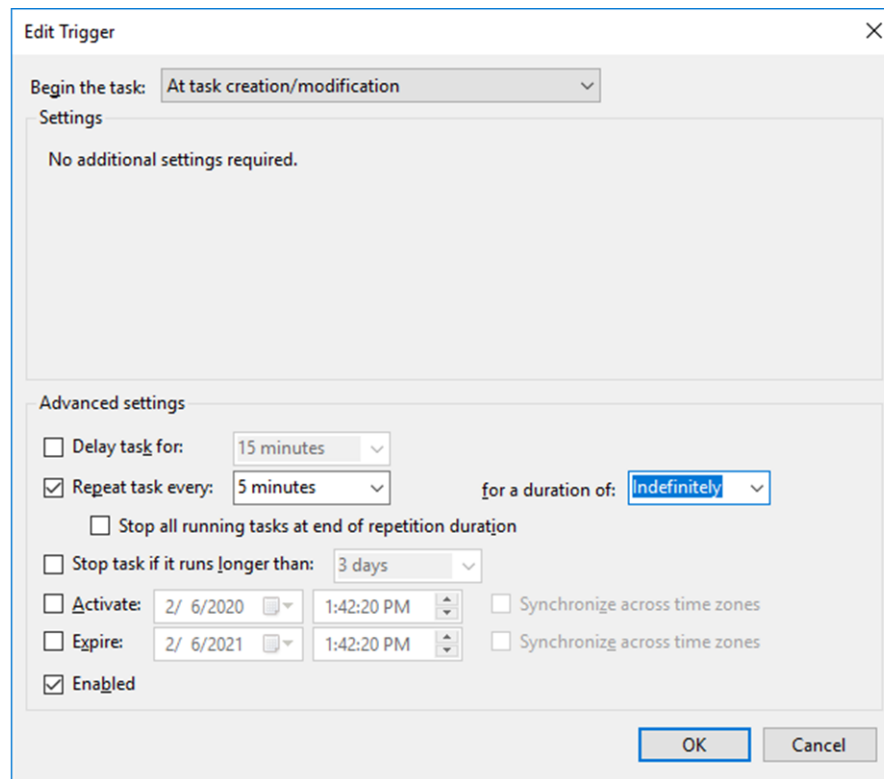
15. Specify the following, and then click [OK].

    ▪ Begin the task: At task creation/modification

    ▪ Repeat task every: 5 minutes

    ▪ for a duration of: Indefinitely

    The [Create Task] window reappears.

16. Click [OK].

    The scheduler to be run initially starts.

    The [Task Scheduler] windows reappears.

    Select [Task Scheduler Library]. Verify that the task you created is displayed and "Status" is "Ready".
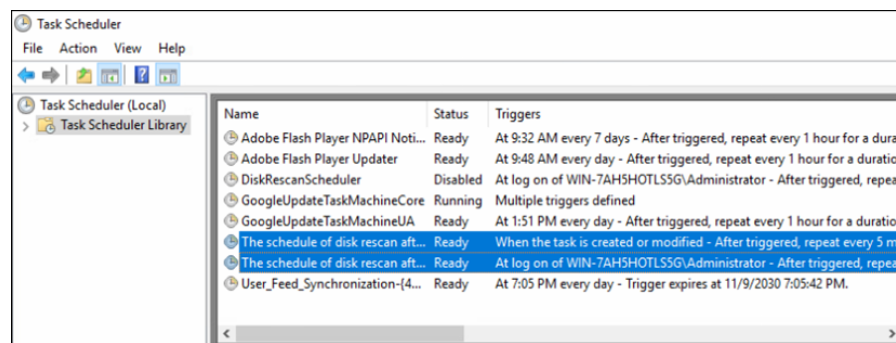
17. Close the [Task Scheduler] window.

> **Note:**
>
> To disconnect the Windows-based compute node from the compute network, delete the created task scheduler. In the [Task Scheduler] window, select the task to be deleted, and then select [Delete] in the right-click menu.

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

Registering information about compute nodes (CLI or REST API)

Register information about compute nodes.

System administrators set the maximum number of compute nodes that can be registered when creating VPS.

Before you begin

Required role: VpsStorage

Procedure

1. Register information about compute nodes.

    Run either of the following commands with the nicknames of the intended compute nodes and the type of the OS running on the compute nodes specified.

    Conventions to be followed when setting a nickname

    - Number of characters: 1 to 229

    - Characters that can be used: Numbers (0 to 9), uppercase alphabet (A to Z), lowercase alphabet (a to z), symbols (\ . : @ _) for the first character. In addition to these, a hyphen (-) can be used for the second and subsequent characters.

    - Each compute node must have a unique nickname.

    REST API: POST /v1/objects/servers

    CLI: server_create

    Verify the job ID which is displayed after the command is run.

2. Verify the state of the job.

    Run either of the following commands with the job ID specified.

    REST API: GET /v1/objects/jobs/<jobID>

    CLI: job_show

    If the job state is "Succeeded", the job is completed.

3. Obtain a list of compute nodes and verify that the information about the intended compute nodes is registered.

   REST API: GET /v1/objects/servers

   CLI: server_list

Obtaining a list of information about compute nodes (CLI or REST API)

The following information can be obtained.

- id: IDs (uuid) of compute nodes

- nickname: Nicknames of compute nodes

- osType: OS types of compute nodes

- totalCapacity: Total capacity of the volumes on the storage pool allocated to the compute node

- usedCapacity: Consumed amount of the volumes on the storage pool allocated to the compute node

- numberOfPaths: Number of registered paths

- vpsId: ID of the VPS to which compute nodes belong

- vpsName: Name of the VPS to which compute nodes belong

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Obtain a list of information about compute nodes.

   REST API: GET /v1/objects/servers

   CLI: server_list

Obtaining information about individual compute nodes (CLI or REST API)

The following information can be obtained for the compute node with the ID specified.

- numberOfVolumes: Number of allocated volumes

- paths: List of information about registered paths (WWN or iSCSI name of the initiator for the intended compute node, list of IDs of compute ports of targets with which the applicable initiator negotiates)

- id: ID (uuid) of the intended compute node

- nickname: Nickname of the intended compute node

- osType: OS type of the intended compute node

- totalCapacity: Total capacity of the volumes on the storage pool allocated to the compute node

- usedCapacity: Consumed amount of the volumes on the storage pool allocated to the compute node

- numberOfPaths: Number of registered paths

- vpsId: ID of the VPS to which the intended compute node belongs

- vpsName: Name of the VPS to which the intended compute node belongs

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the ID of the intended compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Obtain information about the intended compute node.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*

   CLI: server_show

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

Editing information about compute nodes (CLI or REST API)

Edit information about the intended compute node. You can edit the nickname and OS type of the intended compute node.

Each compute node must have a unique nickname.

> ⚠️ **Caution:**
>
> While the information about a compute node is edited, I/O processing to a volume specified by a volume path in the compute node is temporarily stopped.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the intended compute node.

    If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

    REST API: GET /v1/objects/servers

    CLI: server_list

2. Edit information about the intended compute node.

    Run either of the following commands with the compute node ID specified.

    If you use the CLI, you can specify a nickname instead of the compute node ID.

    REST API: PATCH /v1/objects/servers/*<id>*

    CLI: server_set

    Verify the job ID which is displayed after the command is run.

3. Verify the state of the job.

    Run either of the following commands with the job ID specified.

    REST API: GET /v1/objects/jobs/*<jobId>*

    CLI: job_show

    After running the command, if you receive a response indicating "Succeeded" as the state, the job is completed.

4. Obtain a list of compute nodes and verify that the information about the intended compute node is edited.

    REST API: GET /v1/objects/servers

    CLI: server_list

Deleting information about individual compute nodes (CLI or REST API)

Delete information about the intended compute node. Deleting the compute node information also deletes all compute node initiator information, all compute node path information, and all volume path information.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the intended compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Delete information about the intended compute node.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: DELETE /v1/objects/servers/*<id>*

   CLI: server_delete

   Verify the job ID which is displayed after the command is run.

3. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobID>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

4. Obtain a list of compute nodes and verify that the information about the intended compute node is deleted.

   REST API: GET /v1/objects/servers

   CLI: server_list

   If tasks are scheduled according to *Scheduling the tasks* in this manual, go to the next step. If no tasks are scheduled, this completes the procedure.

5. From the Start menu, select [Administrative Tools], and then [Task Scheduler].

   The [Task Scheduler] window appears.

6. Select [Task Scheduler Library] in the left side of the window, select the task created in *Scheduling the tasks*, and then click [Delete] from the right-click menu.

   Delete all the tasks created in *Scheduling the tasks*.

7. Close the [Task Scheduler] window.

Registering information about the initiators for compute nodes (CLI or REST API)

After operating the compute node to investigate the initiator name (iSCSI name or WWN) of the compute node, register the compute node initiator information in Virtual Storage Software block from the controller node.

The maximum number of initiators that can be registered is automatically set.

The procedure differs depending on whether iSCSI connection or FC connection is made between the compute node and the storage node.

In the case of iSCSI connection (CLI or REST API)

> ⚠️ **Caution:**
>
> The initiator name (iSCSI name) of the compute node to be registered with Virtual Storage Software block must be unique in the system. Verify it before registration. It is impossible to see the volume from a compute node whose initiator name (iSCSI name) is the same as that of another compute node.
>
> Also, "iqn" and "eui" at the beginning of the initiator name (iSCSI name) cannot be uppercase. It can only be specified in lower case.

Before you begin

- Required role: VpsStorage

- The information about the applicable compute node must be registered beforehand.

Procedure

1. Verify the initiator name (iSCSI name) of the applicable compute node.

   For details, see the documentation for the OS used on the compute node.

2. Verify that the initiator name (iSCSI name) verified in step 1 is not the same as the initiator name (iSCSI name) of another compute node.

   If they are the same, change the initiator name (iSCSI name).

3. Verify the VPS ID and conditions set for the VPS (upper limit for the number of initiators).

   If you want to specify a VPS by its name in the CLI, verify the VPS name.

   REST API: GET /v1/objects/virtual-private-storages

   CLI: vps_list

4. Verify the ID of the applicable compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

5. Register information about the intended initiator.

   Run either of the following commands with the ID of the compute node, connection protocol for the initiator, and iSCSI name of the initiator.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: POST /v1/objects/servers/<*id*>/hbas

   CLI: hba_create

   Verify the job ID which is displayed after the command is run.

6. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

7. Obtain a list of information about initiators and verify that the information about the intended initiator is registered.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/<*id*>/hbas

   CLI: hba_list

In the case of FC connection (CLI or REST API) (Virtual machine)

Before you begin

- Required role: VpsStorage

- The information about the applicable compute node must be registered beforehand.

Procedure

1. Verify WWN in documentation and so on, from each HBA vendor.

2. Verify the VPS ID and conditions set for the VPS (upper limit for the number of initiators).

   If you want to specify a VPS by its name in the CLI, verify the VPS name.

   REST API: GET /v1/objects/virtual-private-storages

   CLI: vps_list

3. Verify the ID of the applicable compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

4. Register information about the intended initiator.

   Run either of the following commands with the ID of the compute node, connection protocol for the initiator, and the initiator name (WWN) specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: POST /v1/objects/servers/<*id*>/hbas

   CLI: hba_create

   Verify the job ID which is displayed after the command is run.

5. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

6. Obtain a list of information about initiators and verify that the information about the intended initiator is registered.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/<*id*>/hbas

   CLI: hba_list

Obtaining a list of information about initiators for compute nodes (CLI or REST API)

The following information can be obtained.

- id: IDs (uuid) of initiators

- serverId: IDs (uuid) of compute nodes

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

- name: WWN or iSCSI names of initiators

- protocol: Connection protocols for initiators

- portIds: List of IDs (uuid) of compute ports of targets with which initiators negotiate

- vpsId: ID of the VPS to which initiators belong

- vpsName: Name of the VPS to which initiators belong

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the IDs of compute nodes.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Obtain a list of information about initiators.

   Run either of the following commands with a compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/hbas

   CLI: hba_list

Obtaining information about individual initiators for compute nodes (CLI or REST API)

The following information can be obtained for the initiator with the specified ID.

- id: ID (uuid) of the intended initiator

- serverId: ID (uuid) of the applicable compute node

- name: WWN or iSCSI names of initiators

- protocol: Connection protocol for the intended initiator

- portIds: List of IDs (uuid) of compute ports of targets with which the intended initiator negotiates

- vpsId: ID of the VPS to which the intended initiator belongs

- vpsName: Name of the VPS to which the intended initiator belongs

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the ID of the applicable compute node.

    If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

    REST API: GET /v1/objects/servers

    CLI: server_list

2. Verify the ID of the intended initiator.

    If you use the CLI to specify an initiator by WWN or iSCSI name, check the WWN or iSCSI name of the initiator.

    Run either of the following commands with the compute node ID specified.

    If you use the CLI, you can specify a nickname instead of the COMPUTE node ID.

    REST API: GET /v1/objects/servers/*<id>*/hbas

    CLI: hba_list

3. Obtain information about the intended initiator.

    Specify the ID of the applicable compute node and ID of the intended initiator, and run either of the following commands.

    If you use the CLI, you can specify a nickname instead of the compute node ID, or a WWN or iSCSI name instead of the initiator ID.

    REST API: GET /v1/objects/servers/*<id>*/hbas/*<hbaId>*

    CLI: hba_show

Deleting information about the initiators for compute nodes (CLI or REST API)

Delete the initiator information of a compute node. Deleting information about an initiator also deletes the path information of all the related compute nodes.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the applicable compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Verify the ID of the intended initiator.

   If you use the CLI to specify an initiator by WWN or iSCSI name, check the WWN or iSCSI name of the initiator.

   Specify the ID of the applicable compute node and run either of the following commands.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/hbas

   CLI: hba_list

3. Delete information about the intended initiator.

   Specify the ID of the applicable compute node and ID of the intended initiator, and run either of the following commands.

   If you use the CLI, you can specify a nickname instead of the compute node ID, or a WWN or iSCSI name instead of the initiator ID.

   REST API: DELETE /v1/objects/servers/*<id>*/hbas/*<hbaId>*

   CLI: hba_delete

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain a list of information about initiators and verify that the information about the intended initiator is deleted.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/hbas

   CLI: hba_list

Registering path information about compute nodes (CLI or REST API)

The maximum number of registered compute node paths is 4096 per compute node.

Depending on how parameters portId and hbaId are specified, the following are possible:

- Both portId and hbaId are omitted: All the initiators of the intended compute node are allocated to all the compute ports.

- Only hbaId is specified: The specified initiator is allocated to all the compute ports.

> ⚠ **Caution:**
>
> - If you have changed the path information of a compute node, perform a rescan of the storage in that compute node before you start I/O operations.
>
> - If using VMware ESXi as a compute node, set up a path between the compute node and all computer ports. Unless all paths have been set, some volumes might be invisible from a compute node.
>
>   If not using VMware ESXi as a compute node, it is also recommended to set up a path between the compute node and all computer ports to prevent I/O performance from being deteriorated.

Before you begin

- Required role: VpsStorage

- The information about the intended compute node and its initiator must be registered beforehand.

Procedure

1. Verify the ID of the intended compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. When you specify the hbaId parameter, verify the ID of the initiator for the intended compute node.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/hbas

   CLI: hba_list

3. Register compute node path information.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID. You can also specify the compute port of the target behavior to which it is assigned by WWN or iSCSI Name instead of the ID.

   REST API: POST /v1/objects/servers/*<id>*/paths

   CLI: path_create

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobID>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain a list of path information and verify that the intended path information is added.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/paths

   CLI: path_list

Obtaining a list of path information about compute nodes (CLI or REST API)

The following information can be obtained.

- id: Path IDs (character string of initiator ID of the computer node and ID of the compute port for the target operation connected by a comma)

- serverId: IDs (uuid) of compute nodes

- hbaName: WWNs or iSCSI names for compute nodes

- hbaId: IDs (uuid) of initiators for compute nodes

- portId: IDs (uuid) of compute ports of targets with which initiators negotiate

- portName: WWNs or iSCSI names of compute ports of targets with which initiators negotiate

- portNickname: Nicknames of compute ports of targets with which initiators negotiate

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

▪ vpsId: ID of the VPS to which compute node paths belong

▪ vpsName: Name of the VPS to which compute node paths belong

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the IDs of compute nodes.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Obtain a list of path information about compute nodes.

   Run either of the following commands with a compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/paths

   CLI: path_list

Obtaining specific path information about compute nodes (CLI or REST API)

The following information can be obtained for the path information about compute nodes with the specified ID.

▪ id: Path IDs (character string of initiator ID of the computer node and ID of the compute port for the target operation connected by a comma)

▪ serverId: IDs (uuid) of compute nodes

▪ hbaName: WWNs or iSCSI names of initiators for compute nodes

▪ hbaId: IDs (uuid) of initiators for compute nodes

▪ portId: IDs (uuid) of compute ports of targets with which initiators negotiate

▪ portName: WWNs or iSCSI names of compute ports of targets with which initiators negotiate

▪ portNickname: Nicknames of compute ports of targets with which initiators negotiate

▪ vpsId: ID of the VPS to which the compute node path belongs

▪ vpsName: Name of the VPS to which the compute node path belongs

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the ID of the applicable compute node.

    If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

    REST API: GET /v1/objects/servers

    CLI: server_list

2. Verify the ID of the initiator for the applicable compute node and the ID of the compute port.

    If you use the CLI to specify initiators and compute ports by WWN or iSCSI name, check the WWN or iSCSI name of the initiator and compute port.

    Run either of the following commands with the compute node ID specified.

    If you use the CLI, you can specify a nickname instead of the compute node ID.

    REST API: GET /v1/objects/servers/*<id>*/hbas

    CLI: hba_list

3. Obtain path information about the applicable compute node.

    Run either of the following commands with the ID of the compute node, ID of the initiator, and ID of the compute port for the target operation specified.

    If you use the CLI, you can specify a nickname instead of the compute node ID, a WWN or iSCSI name instead of the initiator ID, and a WWN or iSCSI name instead of the compute port ID.

    REST API: GET /v1/objects/servers/*<id>*/paths/*<hbaId>*,*<portId>*

    CLI: path_show

Deleting path information about compute nodes (CLI or REST API)

> ⚠️ **Caution:**
>
> - When compute nodes are clustered and the volumes recognized by the compute nodes are online, set the intended volume offline and then delete path information.
>
> - Before you delete path information from a compute node, verify whether the volumes that can be accessed from the compute node are in SCSI-2 Reserve status or SCSI-3 Persistent Reserve status. If the volumes are in either status, release them from the status, and then delete path information.
>
> - When you change the path information of a compute node, perform a rescan of the storage on that compute node. If the path information already deleted remains on the compute node, it might cause malfunction.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the applicable compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

2. Obtain a list of path information about compute nodes.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/paths

   CLI: path_list

3. Delete path information from the applicable compute node.

   Run either of the following commands with the ID of the compute node, ID of the initiator, and ID of the compute port for the target operation specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID, a WWN or iSCSI name instead of the initiator ID, and a WWN or iSCSI name instead of the compute port ID.

   REST API: DELETE /v1/objects/servers/*<id>*/paths/*<hbaId>*,*<portId>*

   CLI: path_delete

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain a list of path information and verify that the intended path information is deleted.

   Run either of the following commands with the compute node ID specified.

   If you use the CLI, you can specify a nickname instead of the compute node ID.

   REST API: GET /v1/objects/servers/*<id>*/paths

   CLI: path_list

Allocating volumes to compute nodes (CLI or REST API)

Set paths (volume paths) between volumes and compute nodes.

The following two combinations are possible for specifying parameters.

- Combination of volumeId, serverId, and lun: Allocates a volume specified by volumeId to a compute node specified by serverId (lun is optional).

- Combination of volumeIds, serverIds, and startLun: Allocates all the volumes specified with volumeIds to all the compute nodes specified with serverIds (startLun can be omitted).

> ⚠️ **Caution:**
>
> - When using VMware ESXi as a compute node, set up a path between the compute node and all computer ports when registering path information of the compute node. Unless all paths have been set, some volumes might be invisible from a compute node.
>
>   Even when not using VMware ESXi as a compute node, we recommend that you set up a path between the compute node and all computer ports to prevent deterioration in I/O performance.
>
>   For how to set up a path, see *Registering path information about compute nodes*.
>
> - If you have changed the connection information between a volume and compute node, perform a rescan of the storage in that compute node before you start I/O operations.
>
> - If you specify a volume path by omitting lun or startLun, the smallest LUN number that is not used is automatically allocated. However, if multiple volume paths are set without specifying lun at the same time, or if volume paths are set for multiple volumes, LUNs might not be assigned in ascending order they are set. To assign a specific LUN, set a volume path by specifying lun.
>
> - If some volumes cannot be recognized by the OS, the system might behave as follows. Resolve the state where volumes cannot be recognized, and then perform a rescan of the storage on the compute node.
>
>   - During recognition of LUNs in the order from the smallest, if an unrecognizable LUN exists, LUN recognition stops.
>
>   - When LUN=0 cannot be recognized, LUN recognition stops.
>
> - Processing performance when registering volume paths
>
>   The processing time varies depending on the number of volume paths registered on the applicable compute node.
>
>   - When registering the first volume path, the processing completes in one or two seconds.
>
>   - For every 1,000 registered volume paths, the processing time for registering one volume path will increase by 1 to 2 seconds.
>
>   - When registering the 8,192th volume path (upper limit), the processing takes approximately 15 seconds.

Before you begin

- Required role: VpsStorage

- The intended volume must be created and the information about the intended compute node must be registered beforehand.

Procedure

1. Verify the VPS ID and conditions set for the VPS (upper limit for the number of volume paths).

   If you want to specify a VPS by its name in the CLI, verify the VPS name.

   REST API: GET /v1/objects/virtual-private-storages

   CLI: vps_list

2. Verify the ID of the volume to be allocated to the compute node.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

3. Verify the ID of the compute node to which the volume is to be allocated.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

4. Allocate the volume to the compute node.

   REST API: POST /v1/objects/volume-server-connections

   CLI: volume_server_connection_create

   Verify the job ID which is displayed after the command is run.

5. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

6. Obtain a list of information about allocation of volumes to compute nodes, and verify that the intended volume is allocated to the intended compute node.

   REST API: GET /v1/objects/volume-server-connections

   CLI: volume_server_connection_list

Obtaining a list of information about allocation of volumes to compute nodes (CLI or REST API)

The following information can be obtained.

- id: IDs of compute node allocation information (each ID is a string consisting of a volume ID and a compute node ID connected by a Comma (,))

- serverId: IDs (uuid) of compute nodes

- volumeId: IDs (uuid) of volumes

- lun: LUN

- vpsId: ID of the VPS to which volume paths belong

- vpsName: Name of the VPS to which volume paths belong

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the IDs of volumes.

    REST API: GET /v1/objects/volumes

    CLI: volume_list

2. Verify the IDs of compute nodes.

    REST API: GET /v1/objects/servers

    CLI: server_list

3. Obtain a list of information about allocation of volumes to compute nodes.

    Run one of the following commands with the volume ID and compute node ID specified in the query parameter.

    REST API: GET /v1/objects/volume-server-connections

    CLI: volume_server_connection_list

Obtaining information about allocation of individual volumes to individual compute nodes (CLI or REST API)

Obtain the information about allocation of volumes to compute nodes with the specified volume ID and compute node ID.

The following information can be obtained.

- id: ID of compute node allocation information (a string consisting of a volume ID and a compute node ID connected by a Comma (,))

- serverId: ID (uuid) of the intended compute node

- volumeId: ID (uuid) of the intended volume

- lun: LUN

- vpsId: ID of the VPS to which the volume path belongs

- vpsName: Name of the VPS to which the volume path belongs

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the ID of the intended volume.

    If you use the CLI to specify a volume by name, check the name of the volume.

    REST API: GET /v1/objects/volumes

    CLI: volume_list

2. Verify the ID of the intended compute node.

    If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

    REST API: GET /v1/objects/servers

    CLI: server_list

3. Obtain information about allocation of the intended volume to the intended compute node.

    Specify the ID of the intended volume and ID of the intended compute node, and run either of the following commands.

    If you use the CLI, you can specify a name instead of the ID of the volume, but a nickname instead of the ID of the compute node.

    REST API: GET /v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>

    CLI: volume_server_connection_show

Canceling allocation of volumes to compute nodes (CLI or REST API)

Disconnects by removing the path (volume path) between the volume and the compute node. Before you run the command, verify that no I/O operation is being performed between the intended compute node and the intended volume.

> ⚠️ **Caution:**
>
> - When compute nodes are clustered and the volumes recognized by the compute nodes are online, set the intended volume offline and then cancel allocation of the volume.
>
> - Before you cancel allocation of a volume to a compute node, verify whether the volumes that can be accessed from the compute node are in SCSI-2 Reserve status or SCSI-3 Persistent Reserve status. If the volumes are in either status, release them from the status, and then cancel allocation of the intended volume.
>
> - When you change the path information of a compute node, perform a rescan of the storage on that compute node. If the path information already deleted remains on the compute node, it might cause malfunction.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the intended volume.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Verify the ID of the intended compute node.

   If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

   REST API: GET /v1/objects/servers

   CLI: server_list

3. Obtain a list of information about allocation of volumes to compute nodes.

   Run one of the following commands with the volume ID and compute node ID specified in the query parameter.

   If you use the CLI, you can specify a name instead of the ID of the volume, but a nickname instead of the ID of the compute node.

   REST API: GET /v1/objects/volume-server-connections

   CLI: volume_server_connection_list

Chapter 3: Managing compute nodes and allocating volumes to compute nodes

4. Cancel allocation of the volume to the compute node.

   Specify the ID of the intended volume and ID of the intended compute node, and run either of the following commands.

   If you use the CLI, you can specify a name instead of the ID of the volume, but a nickname instead of the ID of the compute node.

   REST API: DELETE /v1/objects/volume-server-connections/<*volumeId*>,<*serverId*>

   CLI: volume_server_connection_delete

   Verify the job ID which is displayed after the command is run.

5. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

6. Obtain a list of information about allocation of volumes to compute nodes and verify that the intended allocation is canceled.

   REST API: GET /v1/objects/volume-server-connections

   CLI: volume_server_connection_list

Releasing multiple connections between the volumes and compute nodes (CLI or REST API)

Releases the connections by removing all specified volumes and the path (volume path) of all specified compute nodes. Before you execute the operation, verify that no I/O operation is being performed between the intended compute node and the intended volume.

> ⚠️ **Caution:**
>
> - When compute nodes are clustered and the volumes recognized by the compute nodes are online, set the intended volume offline, and then cancel allocation of the volume.
>
> - Before you cancel allocation of a volume to a compute node, verify whether the volumes that can be accessed from the compute node are in SCSI-2 Reserve status or SCSI-3 Persistent Reserve status. If the volumes are in either status, release them from the status, and then cancel allocation of the intended volume.
>
> - When you change the path information of a compute node, perform a rescan of the storage on that compute node. If the path information already deleted remains on the compute node, it might cause malfunction.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the IDs of volumes.

    If you use the CLI to specify a volume by name, check the name of the volume.

    REST API: GET /v1/objects/volumes

    CLI: volume_list

2. Verify the ID of the intended compute node.

    If you use the CLI to specify a compute node by nickname, check the nickname of the compute node.

    REST API: GET /v1/objects/servers

    CLI: server_list

3. Cancel allocation of the volumes to the compute nodes.

    Specify the IDs of the volumes and IDs of the compute node (for which you want to cancel allocation), and then run either of the following commands.

    If you use the CLI, you can specify a name instead of the ID of the volume, but a nickname instead of the ID of the compute node.

    REST API: POST /v1/objects/volume-server-connections/actions/release/invoke

    CLI: volume_server_connection_release_connections

    Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

    Run either of the following commands with the job ID specified.

    REST API: GET /v1/objects/jobs/*<jobId>*

    CLI: job_show

    After running the command, if you receive a response indicating "Succeeded" as the state, the job is completed.

5. Obtain a list of information about allocation of volumes to compute nodes and verify that the intended allocation is canceled.

    REST API: GET /v1/objects/volume-server-connections

    CLI: volume_server_connection_list

Chapter 4: Managing compute ports

Obtaining a list of information about compute ports (CLI or REST API)

The following information can be obtained.

- id: IDs (uuid) of compute ports

- protocol: Connection protocols of compute ports

- type: Types of compute ports

- nickname: Nicknames of compute ports

- name: WWNs or iSCSI names of compute ports

- configuredPortSpeed: Link speed setting

- portSpeed:

  (Virtual machine) Always "DependsOnHypervisor"

  (Bare metal) Actual link speed

- portSpeedDuplex:

  (Virtual machine) Link speed only (for FC connection configuration) or always "DependsOnHypervisor" (for iSCSI connection configuration)

  (Bare metal) Actual link speed and duplex settings of the physical port used for communication

- protectionDomainId: IDs (uuid) of protection domains containing compute ports

- storageNodeId: IDs (uuid) of storage nodes containing compute ports

- interfaceName: Interface name

- statusSummary: Summary of statuses of compute ports

- status: Status of each compute port

- fcInformation:

  (Virtual machine) FC port information

  (Bare metal) Null

- iscsiInformation: iSCSI port information

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Obtain a list of information about compute ports.

   REST API: GET /v1/objects/ports

   CLI: port_list

Obtaining information about individual compute ports (CLI or REST API)

The following information can be obtained for the compute port with the ID specified.

- id: ID (uuid) of the intended compute port

- protocol: Connection protocol of the intended compute port

- type: Type of the intended compute port

- nickname: Nickname of the intended compute port

- name: WWN or iSCSI name of the intended compute port

- configuredPortSpeed: Link speed setting

- portSpeed:

  (Virtual machine) Always "DependsOnHypervisor"

  (Bare metal) Actual link speed

- portSpeedDuplex:

  (Virtual machine) Link speed only (for FC connection configuration) or always "DependsOnHypervisor" (for iSCSI connection configuration)

  (Bare metal) Actual link speed and duplex settings of the physical port used for communication

- protectionDomainId: ID (uuid) of the protection domain containing the intended compute port

- storageNodeId: ID (uuid) of the storage node containing the intended compute port

- interfaceName: Interface name

- statusSummary: Summary of statuses of compute ports

- status: Status of each compute port

- fcInformation:

  (Virtual machine) FC port information

  (Bare metal) Null

- iscsiInformation: iSCSI port information

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Verify the ID of the intended compute port.

   If you use the CLI to specify a compute port by WWN or iSCSI name, check the WWN or iSCSI name of the compute port.

   REST API: GET /v1/objects/ports

   CLI: port_list

2. Obtain information about the intended compute port.

   Run either of the following commands with the compute port ID specified.

   If you use the CLI, you can specify the WWN or iSCSI name instead of the compute port ID.

   REST API: GET /v1/objects/ports/*<id>*

   CLI: port_show

Chapter 5: Managing volumes

Overview of managing volumes

Volumes are logical devices and they are allocated to compute nodes.

The specifications of volumes are as follows.

Item	Specifications
Maximum number of simultaneous operations of volume operations and snapshot operations	Approx. 20 If you perform 20 or more operations at the same time, operations might not be performed successfully due to a timeout. Volume operations refer to volume creation, deletion, expansion, and setting edition. Snapshot operations refer to preparation for, taking, deleting, and restoring snapshots. A single operation handling multiple volumes is counted as one operation. For example, a volume creation operation with 20 or more volumes specified is counted as one operation. Completion of the job means completion of the operation. When the job state is "Succeeded", the operation is complete.
Volume name	Number of characters: 1 to 32 Characters that can be used: Numbers (0 to 9), uppercase alphabet (A to Z), lowercase alphabet (a to z), symbols (- , . : @ _)
Volume nickname	Number of characters: 1 to 32 Characters that can be used: Numbers (0 to 9), uppercase alphabet (A to Z), lowercase alphabet (a to z), symbols (- , . : @ _)

About the Quality of Service (QoS) function

QoS setting can be applied to volumes. The QoS function provides various performance levels (such as I/O or the amount of transfer) for each volume. When a storage system is shared by multiple services or when multi-tenancy configuration in which multiple companies (services) coexist is applied in the public cloud, required storage performance levels differ depending on services (applications). In normal storage settings, requests for excessive I/O operations from an application have a tendency to degrade the performance level of other applications because an attempt to perform I/O processing is made in the order of requesting. In such a case, using the QoS function that enables controlling I/O processing in units of volumes can suppress performance interference between applications and provide a certain level of performance and quality.

In detail, the QoS function provides the following:

- Function for controlling the performance upper limit for host I/Os for specific volumes

- Function for monitoring alert threshold values of volumes subject to QoS

VPS administrators can set upper limit performance values for individual volumes on the condition that set values do not exceed the upper limit performance values of volumes in a VPS. QoS alert settings cannot be made.

Controlling the upper limit by using QoS

The function for controlling the upper limit provided by QoS determines the performance upper limit when a storage system performs I/O processing for the services that requested the I/O.

Each volume has its upper limit value. Read and write processing is not differentiated. Therefore, their respective upper limits cannot be separately set. When I/O processing is requested for a volume with the upper limit set, a storage system verifies the average number of I/Os and the average amount of transfer for the volume in the last one second. After the I/O upper limit is reached, the subsequent processing will be suppressed although I/O requests from services are accepted. When the number of I/Os or the amount of transfer (or both of them) become lower than the upper limit, I/O processing is resumed. The QoS function provides methods for controlling the upper limit by the number of I/Os, by the amount of transfer, or by both of them. For the method of controlling the upper limit by both the number of I/Os and the amount of transfer, I/O processing is suppressed when either of them reaches the upper limit.

The QoS function controls the upper limit based on the average number of I/Os or the average amount of transfer per second. However, performance information that can be referenced is collected at five second intervals even for high-resolution information. Therefore, in some cases, control of the upper limit by QoS cannot be observed.

The performance information about a storage system includes I/O processing that occurs within the system due to capacity balancing or volume migration that is performed when a storage node is removed. However, this I/O processing that occurs within the system is excluded from the target of the QoS function. As a result, when you reference the performance information about a storage system, for each volume, the number of I/Os might exceed the QoS upper limit by approximately 80 IOPS and the amount of transfer might exceed the QoS upper limit by approximately 20 MiB/s.

The following shows the range of upper limit values that can be set for QoS. When creating volumes, QoS settings of volumes set for a VPS are automatically set if you omit specification of the QoS settings. You can set upper limit performance values for individual volumes on the condition that set values do not exceed the upper limit performance values for volumes in a VPS.

Item	Number of I/Os (IOPS)	Amount of transfer (MiB/s)
Range of upper limit values that can be set	-1 (upper limit is not set), 100 to 2,147,483,647	-1 (upper limit is not set), 1 to 2,097,151

Creating volumes (CLI or REST API)

📄 **Note:**

The data reduction function is not supported.

Before you begin

Required role: VpsStorage

Procedure

1. Create volumes.

   Run either of the following commands with the parameters for creating volumes specified.

   REST API: POST /v1/objects/volumes

   CLI: volume_create

   Verify the job ID which is displayed after the command is run.

   💡 **Tip:**

   - You can set the name and nickname for a volume. The name must be unique in the storage cluster. The nickname does not have to be unique (can also be used for other volumes). For this reason, a name can be used for identifying each individual volume, and a nickname can be used for create a group of volumes.

   - In creating volumes, even though you can specify the ID of the storage controller that manages them, specification is not required because a storage controller is automatically selected. To specify the ID of the storage controller that manages the volumes you create, contact a system administrator.

2. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

3. Obtain a list of volumes and verify that the volumes are created.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

Deleting volumes (CLI or REST API)

Delete unnecessary volumes as follows.

Volumes cannot be deleted in the following cases.

- If a path is set between a compute node and a volume, the volume cannot be deleted.

  Before you delete a volume, cancel the connection between the volume to be deleted and the associated compute node.

- Volumes with "P-VOL", "P/S-VOL", or "S-VOL" snapshotAttribute cannot be deleted using the procedure described in this section. For details about how to delete these volumes, see *Deleting snapshots*.

- A volume whose status is "ExpansionFailed" cannot be deleted. If you want to delete a volume whose status is "ExpansionFailed", expand the volume, verify that the status becomes "Normal", and then run the volume deletion command again.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the intended volume.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Delete the intended volume.

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: DELETE /v1/objects/volumes/*<id>*

   CLI: volume_delete

   Verify the job ID which is displayed after the command is run.

Chapter 5: Managing volumes

3. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

4. Obtain a list of volumes and verify that the volumes are deleted.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

Expanding volumes (CLI or REST API)

You can only expand a volume whose status is "Normal" or "ExpansionFailed", and "volumeType" is "Normal".

- If you expand a volume whose "status" is "Normal", the volume is expanded by the capacity specified in the parameter "additionalCapacity".

- If you expand a volume whose status is "ExpansionFailed", the volume expansion is performed again.

Before you begin

Required role: VpsStorage

Procedure

1. Check the ID of the volume you want to extend.

   If you use the CLI to specify a volume by name, check the name of the volume.

   It also checks the status and volumeType of the volume to expand the capacity of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

   When "status" of the volume to be expanded is "Normal" or "ExpansionFailed", and "volumeType" is "Normal", go to the next step.

2. Expand the volume capacity.

   Run one of the following commands with the capacity to be added to the volume specified for additionalCapacity. Run the command with a value specified if the status of the volume is "Normal", or with no value specified if the status is "ExpansionFailed".

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: POST /v1/objects/volumes/<*id*>/actions/expand/invoke

   CLI: volume_expand

   Verify the job ID which is displayed after the command is run.

3. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

4. Obtain volume information to verify that the volume capacity is expanded.

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/<*id*>

   CLI: volume_show

Obtaining a list of volumes (CLI or REST API)

The following information can be obtained.

> 📄 **Note:**
>
> The data reduction function is not supported.

- savingEffects: Effect of the data reduction function

- id: IDs (uuid) of volumes

- name: Names of volumes

- nickname: Nicknames of volumes

- volumeNumber: Numbers of volumes

- poolId: IDs (uuid) of storage pools

- poolName: Names of storage pools

- totalCapacity: Total logical capacity

- usedCapacity: Consumed logical capacity

- numberOfConnectingServers: Number of connected compute nodes

- numberOfSnapshots: Number of snapshots

- protectionDomainId: IDs (uuid) of protection domains containing volumes

- fullAllocated: Whether all the area where user data is written is allocated in advance

- volumeType: List of volume types (attributes)

- statusSummary: Summary of statuses of volumes

- status: Status of each volume

- storageControllerId: IDs (uuid) of storage controllers managing volumes

- snapshotAttribute: Attributes of snapshots

- snapshotStatus: Statuses of snapshots

- savingSetting: Setting of the data reduction function

- savingMode: Processing mode of the data reduction function

- dataReductionStatus: Status of the data reduction function

- dataReductionProgressRate: Progress rate of the data reduction function

- vpsId: ID of the VPS to which volumes belong

- vpsName: Name of the VPS to which volumes belong

- qosParam: QoS-related parameter

  - upperLimitForIops: Upper limit for volume performance (IOPS)

  - upperLimitForTransferRate: Upper limit for volume performance (MiB/s)

  - upperAlertAllowableTime: Alert threshold value for the upper limit of volume performance (second)

  - upperAlertTime: The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC)

- naaId: NAA ID of the VPS to which volumes belong

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Obtain a list of volumes.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

Obtaining information about individual volumes (CLI or REST API)

The following information can be obtained for the volume with the ID specified.

> 📄 **Note:**
>
> The data reduction function is not supported.

- reservedCapacity: Reserved logical capacity

- freeCapacity: Free logical capacity

- luns: List of volume LUNs

- snapshotProgressRate: Progress rate of preparation for, deleting, and restoring snapshot volumes

- snapshotTimestamp: For S-VOL or P/S-VOL, time when recording of difference data started for the P-VOL at the time of snapshot creation

- snapshotType: Snapshot type ("Snapshot" for S-VOL or P/S-VOL, null for others)

- savingEffects: Effect of the data reduction function

- snapshotConcordanceRate: Match rate between the target volume and one newer generation of S-VOL or copy-source volume

- isWrittenInSvol: Whether the volume can be written from the controller node

- id: ID (uuid) of the intended volume

- name: Name of the intended volume

- nickname: Nickname of the intended volume

- volumeNumber: Number of the intended volume

- poolId: ID (uuid) of the storage pool

- poolName: Name of the storage pool

- totalCapacity: Total logical capacity

- usedCapacity: Consumed logical capacity

- numberOfConnectingServers: Number of connected compute nodes

- numberOfSnapshots: Number of snapshots

- protectionDomainId: ID (uuid) of the protection domain containing the intended volume

- fullAllocated: Whether all the area where user data is written is allocated in advance

- volumeType: List of volume types (attributes)

- statusSummary: Summary of statuses of volumes

- status: Status of each volume

- storageControllerId: ID (uuid) of the storage controller that manages this volume.

- snapshotAttribute: Attributes of snapshots

- snapshotStatus: Statuses of snapshots

- savingSetting: Setting of the data reduction function

- savingMode: Processing mode of the data reduction function

- dataReductionStatus: Status of the data reduction function

- dataReductionProgressRate: Progress rate of the data reduction function

- vpsId: ID of the VPS to which the intended volume belongs

- vpsName: Name of the VPS to which the intended volume belongs

- qosParam: QoS-related parameter

  - upperLimitForIops: Upper limit for volume performance (IOPS)

  - upperLimitForTransferRate: Upper limit for volume performance (MiB/s)

  - upperAlertAllowableTime: Alert threshold value for the upper limit of volume performance (second)

  - upperAlertTime: The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC)

- naaId: NAA ID of the VPS to which the intended volume belongs

Chapter 5: Managing volumes

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the ID of the intended volume.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain information about the intended volume.

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/*<id>*

   CLI: volume_show

Modifying settings of individual volumes (CLI or REST API)

Edit the settings for a regular or snapshot volume.

You can edit volume names, nicknames, and QoS-related parameters (upperLimitForIops, upperLimitForTransferRate, and upperAlertAllowableTime).

You can only edit settings of volumes whose status is "Normal" or "UpdateFailed". For the volumes with the "UpdateFailed" status, re-edit volume settings without specifying parameters corresponding to the editing target.

When you perform editing, note the following points:

- Name can only be set to a unique value in the storage cluster.

- Nickname can have duplicate values between volumes.

- If you specify multiple parameters in step 2, you can specify them only in the following combinations: If you want to make a configuration change that cannot be combined, run the command again.

  - Combination of name and nickname

  - Combination of upperLimitForIops, upperLimitForTransferRate, and upperAlertAllowableTime

- For snapshot volumes, you can also edit the name and nickname of the volume.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the intended volume.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Change the settings of the intended volume.

   Run either of the following commands with the volume ID and parameters for setting volumes specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: PATCH /v1/objects/volumes/<*id*>

   CLI: volume_set

   Verify the job ID which is displayed after the command is run.

3. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

4. Obtain information about the intended volume and verify that the settings of the volume are modified.

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/<*id*>

   CLI: volume_show

Chapter 6: Backing up and restoring volumes by using snapshots

Overview of snapshots

A duplicate image of a volume at a certain point in time is referred to as a snapshot. If you unintentionally update copy-source volume data, you can restore the data from a previously taken snapshot. The copy-source volume is referred to as the primary volume (P-VOL), and the copy-destination volume is referred to as the secondary volume (S-VOL). When the P-VOL is updated, only the data existing before updating in the update-target area is copied to the storage pool as data in snapshot volumes.

By storing only difference data between P-VOL and S-VOL, this scheme reduces capacity and allows a cost-efficient copy to be created. Furthermore, because data is shared between S-VOLs, less capacity is used even if multiple generations of snapshots are taken.

> ⚠️ **Caution:**
> - You should back up any necessary data to another device (for example, via the host) with the following in mind. Also, monitor the used capacity of the storage pool to secure sufficient free space.
>   - The S-VOL uses the data in the P-VOL. Therefore, if a failure occurs in the P-VOL and the data of the P-VOL becomes unavailable, the data of the S-VOL also becomes unavailable.
>   - Difference data is stored in the storage pool. If the storage pool becomes unavailable due to a failure or the storage pool becoming full, data in snapshot volumes will be lost.
>   - If data in snapshot volumes is lost during restore (that is, during execution of an operation to restore P-VOL data from S-VOL data), data in the restore-destination volume (P-VOL) also becomes unavailable.

A snapshot configuration consisting of the copy source and copy destinations as follows is referred to as a snapshot tree. A different generation of the S-VOL is created by the time at which a snapshot is taken.
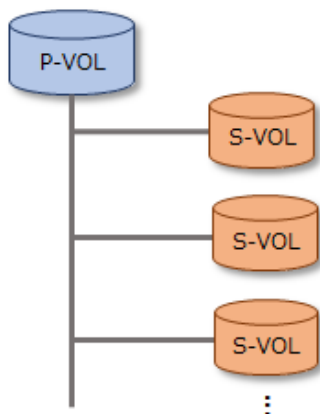
Another snapshot can be created by using an already created snapshot as the copy source. Another snapshot can be created by using an already created snapshot as the copy source. This snapshot configuration is called a cascade configuration. A cascade configuration can comprise of a maximum of 64 layers. In a cascade configuration, a volume can have both the P-VOL and S-VOL attributes. Such a volume is called a P/S-VOL.

A cascade configuration with a maximum of 64 layers can be created under the P-VOL layer. For example, the cascade configuration in the following figure has three layers.
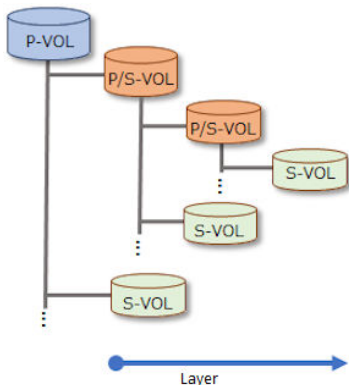
The following table lists the operations on snapshots.

Operation	Description	See:
Prepare to take a snapshot	Create a volume (S-VOL) for taking snapshots. When you prepare to take a snapshot, a volume (S-VOL) will be created. A snapshot cannot be taken without creating a volume (S-VOL).	Preparing to take a snapshot (on page 74)
Take a snapshot	Take a snapshot for the prepared S-VOL.	Taking a snapshot (CLI or REST API) (on page 76)

Operation	Description	See:
Delete snapshots	Delete snapshot volumes that are no longer needed.	Deleting snapshots (CLI or REST API) (on page 77)
Restore P-VOLs from a snapshot	Restore P-VOL data from S-VOL data.	Restoring P-VOLs from a snapshot (CLI or REST API) (on page 79)
Obtain P-VOL information for the volume	Obtain information about the P-VOL from which S-VOL was created.	Obtaining P-VOL information for the volume (CLI or REST API) (on page 83)
Obtain a list of S-VOL information for the volume	Obtain a list of S-VOL information created from P-VOL.	Obtaining a list of S-VOL information for the volume (CLI or REST API) (on page 84)
Modify settings of individual volumes	Edit the settings for the volume.	Modifying settings of individual volumes (CLI or REST API) (on page 63)

Snapshot specifications

The following table lists the specifications for snapshots.

Item	Specifications
Number of S-VOLs (per P-VOL)	1,023
Maximum number of simultaneous operations of volume operations and snapshot operations	Approx. 20 If you perform more than 20 operations at the same time, operations might not be performed successfully due to a timeout. Volume operations refer to volume creation, deletion, expansion, and setting edition. Snapshot operations refer to preparation to take snapshots, taking, deleting and restoring snapshots. A single operation handling multiple volumes is counted as one operation. For example, a volume creation operation with 20 or more volumes specified is counted as one operation. Completion of the job means completion of the operation. When the job state is "Succeeded", the operation is complete.
Snapshot volume name	Number of characters: 1 to 32

Item	Specifications
	Characters that can be used: Numbers (0 to 9), uppercase alphabet (A to Z), lowercase alphabet (a to z), symbols (- , . : @ _)
Maximum number of layers in a cascade configuration	64

Snapshot-specific information obtained from volume information

You can obtain the following snapshot-specific information by obtaining a list of snapshot information.

Type of information	Description
snapshotStatus	Snapshot status indication: Normal, Empty (empty status without snapshot meta data)*, Deleting, Restoring, Preparing (to take snapshots), Prepared, or Error An ordinary volume is indicated by "-".
	* This is a temporary status when creating or deleting S-VOLs. When you create S-VOLs, the status transitions to Preparing after a while, and the applicable volumes are deleted when you delete S-VOLs.
	You can also obtain snapshotStatus from a list of volume information.
snapshotProgressRate	Progress rate of snapshot operation. Indicated only when snapshotStatus is Preparing, Restoring or Deleting. In the case of other statuses, null is output.
	snapshotProgressRate is not displayed at the same time as the match rate (snapshotConcordanceRate).
	You can also obtain snapshotProgressRate from volume information.
snapshotTimestamp	Time when the snapshot is taken.
	This is displayed when snapshotStatus is "Normal". In the case of other statuses, null is output.
	You can also obtain snapshotTimestamp from volume information.
snapshotType	"Snapshot" is displayed for a snapshot volume.

Chapter 6: Backing up and restoring volumes by using snapshots

Type of information	Description
	You can also obtain snapshotType from volume information.
snapshotConcordanceRate	Match rate (%) between this volume and the comparison-target volume.
	When this volume is the latest generation S-VOL, the volume to be compared will be the P-VOL. When this volume is not the latest generation S-VOL, the volume to be compared will be an S-VOL of a newer generation. For example, if the P-VOL is written in a multi-generation configuration (1:N configuration for a P-VOL and S-VOL), only the latest generation of S-VOL will have a lower match rate. If the S-VOL is written, the match rate will decrease only for the S-VOL of the written generation.
	The match rate is an approximate value and might have a large margin of error.
	Differential data indicated by the match rate is managed in units of 256 KiB.
	When restoring and deleting snapshots, the larger the amount of data difference between P-VOL and S-VOL is, the longer the process will take.
	This information can be used to estimate the processing time for snapshot restoration and deletion in advance. *
	You can also obtain snapshotConcordanceRate from volume information.
isWrittenInSvol	This information shows whether the host has written to the S-VOL.
	This is displayed when snapshotStatus is "Normal". In the case of other statuses, null is output.
	You can also obtain isWrittenInSvol from volume information.
* See *Processing time for snapshot operations* in this manual.	

Status transition (snapshot operation and snapshotStatus)

The snapshot Status transits depending on the snapshot operation.

Note:

It might take about one minute for snapshotStatus to transit. To confirm the transition of snapshotStatus, wait about one minute and then check again.
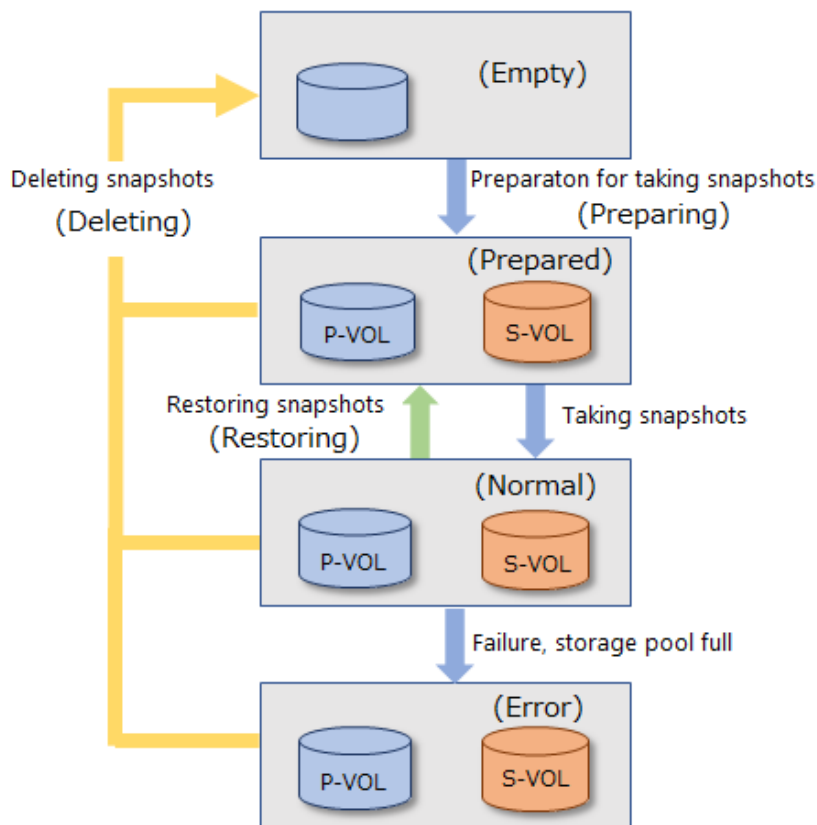
Tip:

Whether the volume is a snapshot volume can be found in snapshotAttribute of volume information. "P-VOL", "P/S-VOL", or "S-VOL" is indicated. An ordinary volume is indicated by "-".

Also, the number of S-VOLs in the P-VOL or P/S-VOL can be found in numberOfSnapshots in volume information.

Whether the P-VOL and S-VOL can be written or read

The P-VOL is readable/writable regardless of snapshotStatus of the copy-destination volume.

The S-VOL is readable/writable only when snapshotStatus is "Normal".

Whether or not the operations can be performed for each snapshotStatus

The snapshot tree configuration and whether or not the operations can be performed for each snapshotStatus are as follows:

📄 **Note:**

It might take about one minute for snapshotStatus to transit. To confirm the transition of snapshotStatus, wait about one minute and then check again.

■ Non-cascade configuration

"Snapshot operation for A" in the table indicates preparation for taking a snapshot of volume A or taking a snapshot of volume A. To prepare for taking a snapshot of volume B under volume A or take a snapshot of volume B under volume A, see "Snapshot operation for B" in "■ Cascade configuration".

snapshot Type	snapshot Status of A	Snapshot operation for A				
		Prepare and take a snapshot (PrepareAndFinalize)	Prepare	Take a snapshot (Finalize)	Restore	Delete
None	None (A does not exist)	○	○	-	-	-
Snapshot	Empty	-	-	×	×	× *
	Preparing	-	-	×	×	×
	Prepared	-	-	○	△	○
	Normal	-	-	△	○	○
	Restoring	-	-	×	×	×
	Error	-	-	×	×	○
	Deleting	-	-	×	×	△

* If removal processing is interrupted because the storage cluster has stopped, the processing can be completed by retrying the operation.

○: Succeeds △: Succeeds but nothing is performed (The job ends normally without performing processing.) ×: Does not succeed -: The operation is not available

■ Cascade configuration

Chapter 6: Backing up and restoring volumes by using snapshots

You can take a snapshot of volume B only if the volume is operable.

snapshot Type	snapshot Status of A	Snapshot operation for B				
		Prepare and take a snapshot (PrepareAndFinalize)	Prepare	Take a snapshot (Finalize)	Restore	Delete
Snapshot	Empty	×	×	-	-	-
	Preparing	×	×	-	-	-
	Prepared	×	○	×	△	○
	Normal	○	○	○	○	○
	Restoring	×	×	×	△	○
	Error	×	×	×	×	○
	Deleting	×	×	-	-	-
○: Succeeds △: Succeeds but nothing is performed (The job ends normally without performing processing.) ×: Does not succeed -: The operation is not available						

You can take a snapshot of volume A only if the volume is operable.

snapshot Type	snapshot Status of B	Snapshot operation for A				
		Prepare and take a snapshot (PrepareAndFinalize)	Prepare	Take a snapshot (Finalize)	Restore	Delete
Snapshot	Empty	-	-	○	○	×
	Preparing	-	-	○	○	×
	Prepared	-	-	○	○	×
	Normal	-	-	△	×	×
	Restoring	-	-	△	×	×
	Error	-	-	△	×	×
	Deleting	-	-	△	×	×

Chapter 6: Backing up and restoring volumes by using snapshots

snapshot Type	snapshot Status of B	Snapshot operation for A				
		Prepare and take a snapshot (PrepareAndFinalize)	**Prepare**	**Take a snapshot (Finalize)**	**Restore**	**Delete**
○: Succeeds △: Succeeds but nothing is performed (The job ends normally without performing processing.) ×: Does not succeed -: The operation is not available						

■ When sharing the copy-source VOL among multiple snapshots

You can take a snapshot of volume B only if the volume is operable.

snapshot Type	snapshot Status of A	Snapshot operation for B				
		Prepare and take a snapshot (PrepareAndFinalize)	**Prepare**	**Take a snapshot (Finalize)**	**Restore**	**Delete**
Snapshot	Empty	○	○	○	○	○
	Preparing	○	○	○	○	○
	Prepared	○	○	○	○	○
	Normal	○	○	○	○	○
	Restoring	×	○	×	×	○
	Error	×	×	○	○	○
	Deleting	×	×	○	○	○
○: Succeeds △: Succeeds but nothing is performed (The job ends normally without performing processing.) ×: Does not succeed						

■ QoS settings of snapshots

QoS settings can be applied to a P-VOL, S-VOL, and P/S-VOL.

For details about the QoS function, see *About the Quality of Service (QoS) function*.

Chapter 6: Backing up and restoring volumes by using snapshots

> 📄 **Note:**
>
> Snapshots do not inherit QoS settings of the copy-source volumes.

Preparing to take a snapshot

Run either of the following commands with the copy-source volume specified to create a volume (S-VOL) to take a snapshot of.

- Preparation for taking a snapshot cannot be performed if there is a snapshot whose status is "Deleting" or "Error" on the copy-source volume.

- If the copy-source volume is a snapshot, the system can be prepared for taking a snapshot only when the copy-source snapshot is in Normal state.

> 📄 **Note:**
>
> Preparing to take a snapshot and taking a snapshot can be performed in one step. Batch processing simplifies the operation, but if the snapshot is to be taken after I/O is stopped, the I/O stop time will be longer because I/O must also be stopped for the time it takes to prepare to take the snapshot.
>
> Therefore, this manual separately describes the operation for preparing to take a snapshot and the operation for taking a snapshot.
>
> To sequentially perform the operation for preparing to take a snapshot and the operation for taking a snapshot, specify PrepareAndFinalize for the operationType parameter (in the case of CLI, --operation_type) when you perform step 3.
>
> Note that the operation for preparing to take a snapshot and the operation for taking a snapshot are sequentially performed if the operationType parameter (in the case of CLI, --operation_type) is omitted.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the volume for which you want to create a snapshot.

   If you want to specify a volume by its name in the CLI, verify the name of the intended volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain a list of S-VOL information by specifying the ID of the copy-source volume, and then confirm that there are no volumes for which snapshotStatus is "Deleting" or "Error".

   If you use the CLI, you can specify the name of the volume instead of its ID.

   REST API: GET /v1/objects/volumes/<*id*>/snapshot-volumes

   CLI: snapshot_volume_list

   If there are any volumes with snapshotStatus of "Deleting", wait for the respective processes to complete. If there are any volumes with "Error" snapshotStatus, delete the associated S-VOLs.

3. Prepare to take a snapshot.

   Run either of the following commands with the ID of the copy-source volume specified and "Prepare" specified for the operationType parameter.

   If you use the CLI, you can specify the name of the volume instead of its ID. The parameter is --operation_type and the specified value is "Prepare".

   REST API: POST /v1/objects/volumes/actions/create-snapshot/invoke

   CLI: volume_create_snapshot

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain a list of S-VOL information to verify that an S-VOL whose snapshotStatus is "Prepared" has been created.

   Run one of the following commands with the ID of the copy-source volume specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/<*id*>/snapshot-volumes

   CLI: snapshot_volume_list

   After performing step 4, it might take approximately one minute for the snapshot status to change. When you want to verify that the snapshot status has changed, wait for approximately one minute before verification.

Taking a snapshot (CLI or REST API)

Run either of the following commands with the copy-destination volume created in the preparatory step specified.

- If a snapshot in the Restoring state exists under the copy-source volume, a snapshot cannot be taken.

- You can take a snapshot only when the copy-destination snapshot is in Prepared state.

- If the copy-source volume is a snapshot, you can take a snapshot only when the copy-source snapshot is in Normal state.

> **Note:**
>
> - After a snapshot is taken, a write request for the P-VOL is processed with the CoW (Copy-on-Write) method. In the CoW method, the data existing before a write is performed is copied to the storage pool, and then the write is performed.
>
> - Immediately after creating a snapshot volume, whenever the data in the P-VOL is updated, the difference data is copied. If many write I/O operations are performed from the host continuously, a large amount of difference data will be copied. This will increase the load on the storage node, causing degradation in IOPS performance of write and read I/O operations. If IOPS performance is degraded, reduce the number of write I/O operations from the host.

> **Caution:**
>
> To keep data consistent, stop the host I/O until a snapshot has been taken. If the volume is for the OS system disk, take a snapshot after shutting down the OS. For a data disk volume, forcibly write file cache to disk and stop writing data to the drive (for example, with the sync command and fsfreeze command) or unmount the disk, and then take a snapshot. After a snapshot has been obtained, resume the host I/O operations. The steps for stopping the host I/O operations depend on the OS. Stop the host I/O operations according to the steps for the OS you are using.

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the volume for which you want to create a snapshot.

   If you want to specify a volume by its name in the CLI, verify the name of the intended volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain a list of S-VOL information by specifying the ID of the copy-source volume, and verify that there is no volume with snapshotStatus of "Restoring". Also, verify the ID of the copy-destination volume and make sure that snapshotStatus is "Prepared".

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/<*id*>/snapshot-volumes

   CLI: snapshot_volume_list

   After verification, go to the next step.

3. Take a snapshot.

   Run either of the following commands with the ID of the copy-source volume specified and "Finalize" specified for the operationType parameter.

   If you want to use the CLI, you can specify the name of the volume instead of its ID. The parameter is operation_type and the specified value is "Finalize".

   REST API: POST /v1/objects/volumes/actions/create-snapshot/invoke

   CLI: volume_create_snapshot

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/<*jobId*>

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain information about replication-destination volumes to verify that the snapshotStatus has changed to "Normal".

   REST API: GET /v1/objects/volumes/<*id*>

   CLI: volume_show

   After performing step 4, it might take approximately one minute for the snapshot status to change. When you want to verify that the snapshot status has changed, wait for approximately one minute before verification.

Deleting snapshots (CLI or REST API)

When snapshot volumes are no longer necessary, you can delete them as follows.

When you perform editing, note the following points:

- If the path (volume path) between the snapshot volume and the compute node is set, the snapshot volume cannot be deleted. Disconnect the snapshot volume from the compute node before deleting it.

- You can delete a snapshot only when the snapshot is in Prepared, Normal, or Error state.

- A snapshot that is the copy source of another existing snapshot cannot be deleted.

▪ To delete the S-VOL, run the snapshot deletion command.

▪ To delete the P-VOL, run the snapshot deletion command to delete all the snapshot volumes (S-VOL) that were created from the P-VOL to be deleted. Then, run the volume deletion command to delete the P-VOL (which is now a normal volume due to S-VOL deletion).

▪ To delete the P/S-VOL, run the snapshot deletion command to delete all the snapshot volumes (S-VOL) that were created from the P/S-VOL to be deleted. Then, run the volume deletion command to delete the P/S-VOL (which is now a S-VOL due to S-VOL deletion).

Before you begin

Required role: VpsStorage

Procedure

1. Verify the ID of the volume (P-VOL or S-VOL).

   If you use the CLI to specify P-VOL and S-VOL by name, check the names of P-VOL and S-VOL.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain information of the snapshot volume to be deleted, and verify that snapshotStatus is "Prepared", "Normal" or "Error".

   Run either of the following commands with the volume ID specified.

   If you want to use the CLI, you can specify the name of the volume instead of its ID.

   REST API: GET /v1/objects/volumes/*<id>*

   CLI: volume_show

   After verification, go to the next step.

3. Delete the snapshot volume.

   Run either of the following commands with the S-VOL ID of the volume to be deleted specified.

   If you use the CLI, you can specify the name of the S-VOL instead of its ID.

   REST API: POST /v1/objects/volumes/actions/delete-snapshot/invoke

   CLI: volume_delete_snapshot

   Verify the job ID which is displayed after the command is run.

4. Verify the state of the job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   If the job state is "Succeeded", the job is completed.

5. Obtain a list of information about S-VOLs and verify that S-VOLs are deleted.

Chapter 6: Backing up and restoring volumes by using snapshots

Run either of the following commands with the P-VOL volume ID specified.

If you use the CLI, you can specify a name instead of the ID of the volume.

REST API: GET /v1/objects/volumes/<*id*>/snapshot-volumes

CLI: snapshot_volume_list

> 📄 **Note:**
>
> If the snapshotStatus of an S-VOL becomes "Error" and the snapshot deletion command is unsuccessful even if the intended S-VOL is specified, you can delete all S-VOLs created from the P-VOL simultaneously by running the snapshot deletion command with the volume ID of the P-VOL and true specified for masterVolumeId and snapshotTree (for the REST API) and for --master_volume_id and --snapshot_tree (for the CLI), respectively. In performing this operation, if a command that performs an operation on a snapshot volume existing in the tree on the P-VOL is already being run by a prior job, the job might become unsuccessful. The job that runs the snapshot deletion command might also result in failure. If the job results in failure, rerun the command.
>
> For detailed operations, see *POST/v1/objects/volumes/actions/delete-snapshot/invoke* in the *Hitachi Virtual Storage Software Block REST API Reference* and *volume_delete_snapshot* in the *Hitachi Virtual Storage Software Block CLI Reference*.

Restoring P-VOLs from a snapshot (CLI or REST API)

The operation of restoring the copy-source volume data from the copy-destination volume data is called snapshot restore.

When restore is started and snapshotStatus becomes Restoring, the data in the restore-destination volume will virtually reflect the data in the restore-source volume, even if the restore copy is not yet complete, and the restored data can be accessed immediately.

When restore starts, access to the restore-source volume is suppressed. When the restore is complete, the restore-source volume will transition to the Prepared state. The volume that has transitioned to the Prepared state can take the next snapshot without the need to perform preparation for taking a snapshot.

▪ You can restore a volume only when snapshotStatus of the copy-destination (restore-source) volume is "Normal".

▪ You cannot restore a volume if a snapshot has been taken by using the copy-destination (restore-source) volume as the copy-source volume in a cascade configuration.
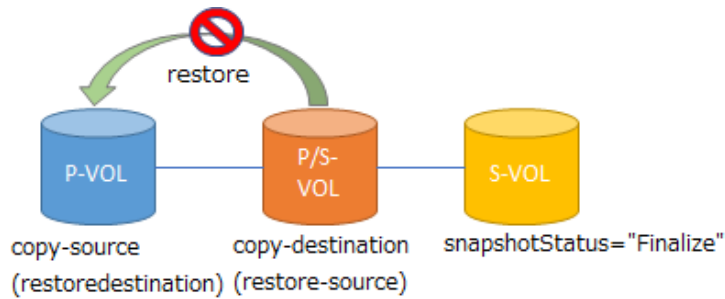
▪ If the copy-source (restore-destination) volume is a snapshot in a cascade configuration, a volume can be restored only when snapshotStatus of the copy-source (restore-destination) volume is "Normal".
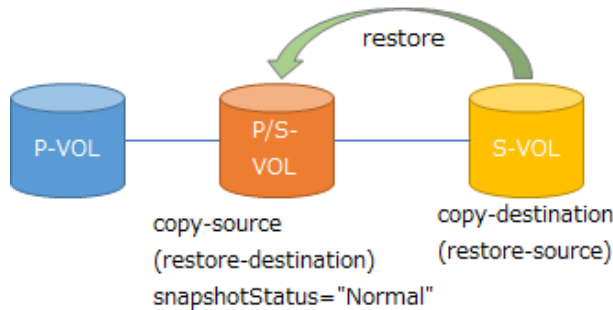
▪ When the copy-source volume is shared, if there is already a snapshot whose snapshotStatus is "Restoring" under the copy-source (restore-destination) volume, a new restore operation cannot be started.
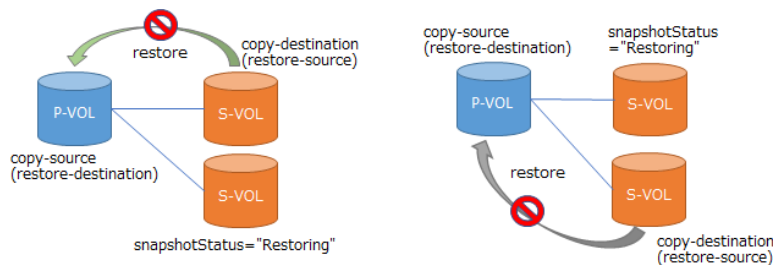
📄 **Note:**

- When a restore operation is performed, the data in the restore-source volume is written to the restore-destination volume. Because the data in the restore-destination volume before the data in the restore-source volume is written is copied to the storage pool here, the storage pool usage will increase. If data in snapshot volumes is lost during restore due to the storage pool being full, data in the restore-destination volume will also become unavailable.

- When restoring a volume, do so when IOPS to the restore-destination volume is low.

  If the data in the restore-destination volume is updated by a write I/O, differential data copy will run. Furthermore, copying data from the restore-source volume to the restore-destination volume by restore will also run. These copy operations increase the load in the storage node and deteriorate not only the IOPS performance of write I/O but also the IOPS performance of read I/O.

- If more failures than the redundant configuration allows occur during restoration, restoration processing might be unsuccessful. However, processing automatically restarts after the node is restored. In such cases, obtain the volume information, and then verify the progress of the restoration from snapshotStatus. If snapshotStatus is "Restoring", restoration is in progress. Wait until the processing finishes. If snapshotStatus is "Prepared", restoration has completed.

⚠️ **Caution:**

- If the storage cluster becomes abnormal during restoration due to failures exceeding redundancy (for example, a network switch failure, power failure, or an unintentional operation to shut down the storage node VMs* on multiple storage nodes), data in snapshot volumes will be lost and the data on the restoration-destination volume will become unavailable. Before performing restoration, back up data on the restoration-destination volume to another normal volume via hosts or other means, and record information of the restoration-destination volume by performing the operations described in *Obtaining information about individual volumes*.

  * (Bare metal) Storage nodes

- To keep the data consistent, stop the host I/O operations until a snapshot has been restored. If the volume is for the OS system disk, restore a snapshot from the restore-source volume (S-VOL) after shutting down the OS. For a data disk volume, restore a snapshot after unmounting the disk. In this case, the fsfreeze command cannot be used. After a snapshot restoration is completed, mount the file system, and then resume the host I/O operations. The steps for stopping the host I/O operations depend on the OS. Perform the process according to the steps for the OS you are using.

Before you begin

Required role: VpsStorage

Procedure

1. Obtain a list of S-VOL information by specifying the restore-destination volume, and verify that there is no volume with snapshotStatus of "Restoring".

    REST API: GET /v1/objects/volumes/<*id*>/snapshot-volumes

    CLI: snapshot_volume_list

    After verification, go to the next step.

2. Restore the intended P-VOL with data in snapshot volumes.

    Run either of the following commands with the restore-source volume (S-VOL) ID specified.

    If you use the CLI, you can specify a name instead of the ID of the volume.

    REST API: POST /v1/objects/volumes/actions/restore-snapshot/invoke

    CLI: volume_restore_snapshot

    Verify the job ID which is displayed after the command is run.

    You can check the progress of the restoration by obtaining the volume information, snapshotProgressRate. For how to obtain the volume information, see *Obtaining information about individual volumes* in this manual. For the volume ID, specify the ID of the restore-source volume.

3. Verify the state of the job.

    Run either of the following commands with the job ID specified.

    REST API: GET /v1/objects/jobs/<*jobId*>

    CLI: job_show

    If the job state is "Succeeded", the job is completed.

4. Obtain information about restore-source volumes (S-VOLs) to verify that the snapshotStatus has changed to "Prepared".

    REST API: GET /v1/objects/volumes/<*id*>

    CLI: volume_show

    After performing step 3, it might take approximately one minute for the snapshotStatus to change. When you want to verify that the snapshotStatus has changed, wait for approximately one minute before verification.

Obtaining P-VOL information for the volume (CLI or REST API)

The following information can be obtained.

- masterVolumeId: Volume ID (uuid) of the P-VOL

- vpsId: ID of the VPS to which the P-VOL belongs

- vpsName: Name of the VPS to which the P-VOL belongs

- qosParam: QoS-related parameter

  - upperLimitForIops: Upper limit for volume performance (IOPS)

  - upperLimitForTransferRate: Upper limit for volume performance (MiB/s)

  - upperAlertAllowableTime: Alert threshold value for the upper limit of volume performance (second)

  - upperAlertTime: The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC)

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the volume ID of the S-VOL that is created from the intended P-VOL.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain P-VOL information.

   Run either of the following commands with the S-VOL volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/*<id>*/master-volume

   CLI: master_volume_show

Obtaining a list of S-VOL information for the volume (CLI or REST API)

The following information can be obtained.

- snapshotVolumeId: Volume ID (uuid) of the S-VOL

- snapshotVolumeName: Name of the S-VOL

- snapshotVolumeNickname: Nickname of the S-VOL

- statusSummary: Status summary of the S-VOL

- status: Volume status

- snapshotStatus: Snapshot status

  It might take about one minute for snapshotStatus to transit. To confirm the transition of snapshotStatus, wait about one minute and then check again.

- snapshotProgressRate: Progress rate of preparation for taking snapshots of, deleting, and restoring the S-VOL

- snapshotTimestamp: Time when recording of difference data started for the P-VOL at the time of snapshot creation

- snapshotType: Snapshot type

- snapshotConcordanceRate: Match rate between the volume and one newer generation of S-VOL or copy-source volume (P-VOL)

- isWrittenInSvol: Whether the volume has been written from the compute node (host)

- vpsId: ID of the VPS to which the S-VOL belongs

- vpsName: Name of the VPS to which the S-VOL belongs

- qosParam: QoS-related parameter

  - upperLimitForIops: Upper limit for volume performance (IOPS)

  - upperLimitForTransferRate: Upper limit for volume performance (MiB/s)

  - upperAlertAllowableTime: Alert threshold value for the upper limit of volume performance (second)

  - upperAlertTime: The last time the upper limit of volume performance was continuously exceeded and the conditions for the alert threshold of the performance upper limit were met (UTC)

📄 **Note:**

The matching rate of the S-VOL is handled as follows. Whether the generation is older or newer can be verified from "snapshotTimestamp".

- "snapshotConcordanceRate" for the newest generation (S-VOL1) is the rate of matching with the P-VOL.

- "snapshotConcordanceRate" for the S-VOL2 is the rate of matching with the S-VOL3, which is one generation newer than the S-VOL2.

- "snapshotConcordanceRate" for the S-VOL3 is the rate of matching with the S-VOL1, which is one generation newer than the S-VOL3.

When data is written to the P-VOL, the matching rate of the S-VOL1 (newest generation) seemingly decreases because the data written to the P-VOL is assumed to be differential data. If data is written to an S-VOL, the matching rate of only the S-VOL decreases.
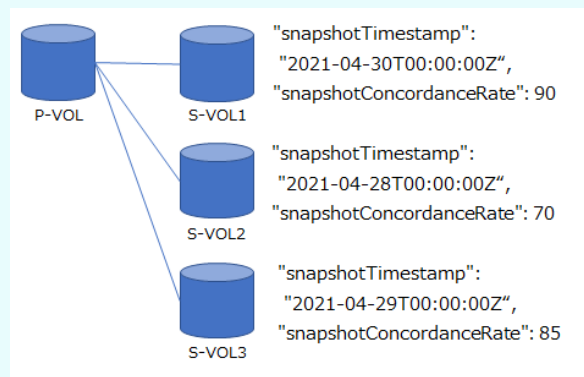
P-VOL S-VOL1

"snapshotTimestamp":
"2021-04-30T00:00:00Z",
"snapshotConcordanceRate": 90

S-VOL2

"snapshotTimestamp":
"2021-04-28T00:00:00Z",
"snapshotConcordanceRate": 70

S-VOL3

"snapshotTimestamp":
"2021-04-29T00:00:00Z",
"snapshotConcordanceRate": 85

Before you begin

Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Verify the volume ID of the P-VOL that is the source of the intended S-VOL.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain a list of S-VOL information.

   Run either of the following commands with the P-VOL volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/volumes/*<id>*/snapshot-volumes

   CLI: snapshot_volume_list

> 📄 **Note:**
>
> If the storage cluster stops while snapshot volumes are being deleted or restored, the delete or restore job disappears due to interruption or failure. As a result, the storage cluster stops. When you restart the storage cluster, the delete or restore operation will be automatically resumed.
>
> To verify the status of deleting or restoring snapshot volumes, obtain the volume information and verify snapshotStatus. When the delete operation ends, snapshotStatus changes to "Empty". When the restore operation ends, it changes to "Prepared".
>
> Note that the S-VOLs remain even if the snapshot volume is deleted and snapshotStatus is changed to "Empty" by the resumed delete operation (Automatic deletion does not take place if the storage cluster stops.). To delete the S-VOLs, run the snapshot deletion command again.

Chapter 7: Managing users

Overview of user management

To operate and set up the Virtual Storage Software block storage cluster, you must register with Virtual Storage Software block as a user.

For example, in REST APIs, you must specify your user ID and password ("*<user-ID>*:*<password>*") in the Authorization header for the request header encoded in Base64.

In the case of CLI, specify the user Id in the --user option and enter your password interactively.

Creating users and user groups

A user's operation privilege is determined by the roles set for the user group to which the user belongs. For example, only a user who belongs to a user group having the VpsSecurity role can create users. Ask a system administrator to create initial user groups and users that have the VpsSecurity role.

> ⚠ **Caution:**
>
> Be careful not to lose the passwords of valid users having the VpsSecurity role. If the passwords of all valid users having the VpsSecurity role are lost, ask a system administrator to change the passwords.

The only operation a user who is created can perform initially is to change the password. After changing the password, the user can perform any operations allowed for the given role. Ask a system administrator about password policies.

A user can be registered for more than one user group.

You can create new user groups.

Roles and available operations

The following table lists the roles and available operations. Create users according to the system operation guidelines.

Role	Available operations
VpsSecurity	Managing users, obtaining session information

Role	Available operations
VpsStorage	Deleting compute node information, registering or deleting compute node initiator information, registering or deleting compute node paths, allocating volumes to compute node paths or releasing connections between volumes and compute node paths, obtaining compute port information
	Creating, deleting, expanding, or editing settings of volumes
	Obtaining, deleting, or restoring snapshots
	Obtaining information about volume capacity or volume performance
	Obtaining VPS usage status
VpsMonitor	Obtaining compute port information
	Obtaining information about volume capacity or volume performance
	Obtaining VPS usage status

No role-based execution restriction is applied to the following operations:

- Verifying, creating, and deleting your own session

- Obtaining a message to be displayed in the GUI login window and in

- Obtaining versions of APIs

- Obtaining information about individual jobs

- Obtaining information about storage cluster master (primary)

- Obtaining information about control ports and internode ports

- Network settings for the storage cluster

- Storage cluster time settings

- Obtaining your own user information

- Changing your own password

- A user who has the VpsSecurity, VpsStorage, or VpsMonitor role can perform the following operations:

  - Obtaining volume information

  - Obtaining S-VOL and P-VOL information

  - Obtaining compute node information

  - Obtaining compute node initiator information

  - Obtaining compute node path information

  - Obtaining volumes and compute node connection information

  - Obtaining compute port information

  - Obtaining storage node network settings

Basic authentication, session authentication, and ticket authentication

To perform a storage cluster operation through a REST API, for example, send an authentication request to Virtual Storage Software block with credentials specified in the Authorization header for the request header.

Virtual Storage Software block supports three authentication methods: basic authentication, session authentication, and ticket authentication.

In basic authentication, a user ID and a password are used as credentials. In basic authentication, authentication is performed for each request.

In session authentication, a token is used as credentials, and authentication can be omitted for a period of time. Therefore, session authentication is useful in application-based automatic operations. A token is obtained by running a REST API or CLI for generating a session. For how to generate a token, see *Generating a session*.

User authentication settings and system requirements

The settings that are applied to user authentication are called user authentication settings. User authentication settings contain password complexity, password expiration time, lockout, and session parameters. System administrators set those values and VPS administrators can obtain them. See *Editing user authentication settings*.

Using an external authentication server

When linkage with an external authentication server is configured by the system administrator, authentication can be performed by using the user information registered in the external authentication server. Only an OpenLDAP or Active Directory (AD) external authentication server can be linked.

Obtaining detailed information about users (CLI or REST API)

The following information about the registered users can be obtained.

- userId: User ID

- userObjectId: User object ID

- passwordExpirationTime: Expiration time of the password

- isEnabled: Whether the user is valid

- userGroups: List of IDs of user groups that the user belongs to (user group IDs and the object ID of each user group)

- isBuiltIn: Whether the user is a built-in user

- authentication: Authentication type

- roleNames: Role of the user group

- isEnabledConsoleLogin:

  (Virtual machine) null

  (Bare metal) Whether the console interface can be used

- vpsId: ID of the VPS to which the user belongs

- privileges: List of the VPS information that the user can access

Before you begin

- Role required to obtain detailed information about a user: VpsSecurity

  Role-based execution is not subject to restriction for obtaining the user information about yourself.

Procedure

1. Verify the user ID.

   REST API: GET /v1/objects/users

   CLI: user_list

2. Obtain detailed information about users.

   Run either of the following commands with the user ID specified.

   REST API: GET /v1/objects/users/*<userId>*

   CLI: user_show

Changing your own password (CLI or REST API)

Note that only a user for which authentication is set to local can change their password.

If you change your password, your session is deleted.

Role-based execution is not subject to restriction.

Procedure

1. Change your own password.

   Run either of the following commands with the user ID, current password, and new password specified.

   REST API: PATCH /v1/objects/users/*<userId>*/password

   CLI: user_password_set

   After running the command, you receive a response indicating user information.

Chapter 8: Managing sessions

Overview of session management

A session is connection information between a user and a storage system.

A session can be generated by executing a REST API or CLI for generating a session. You can obtain a token by generating a session.

A generated session is deleted when:

- The session (token) expired.

- The session has timed out without being used.

- The cluster master node (primary) has failed over.

- The session is deleted, the user is edited or deleted, the password is changed, the user is deleted from a user group, or the user group to which the user belongs is edited.

Token expiration time (default: 24 hours) applies to the entire storage system. For details about how to change the token expiration time, see *Editing user authentication settings*.

You can specify the session timeout time (default: 30 minutes) not only for each session but for the entire storage system. To specify the session timeout time for each session, specify the time when sessions are created. If you omit this specification, the session timeout time will be applied to the entire storage system. For details about how to change the session timeout time of the entire storage system, see *Editing user authentication settings*.

> ⚠️ **Caution:**
>
> If the token obtained by generating a session has leaked out, the Virtual Storage Software block storage system might be operated by an unintended third party. Properly manage the obtained token so that it won't be leaked out.

Generating a session (CLI or REST API)

A token for session authentication is generated and the following information about the session is displayed. Session authentication is possible with the generated token.

Role-based execution is not subject to restriction.

- token: Token

- sessionId: Session ID (uuid)

- userId: User ID

- userObjectId: User object ID

- expirationTime: Session expiration time

- createdTime: Date and time when the session is generated

- lastAccessTime: Date and time when the session was used last

- roleNames: List of roles assigned to the user who retains this session

- vpsId: ID of the VPS to which the user who generated the session belongs

- privileges: List of the VPS information that the user who generated the session can access

Procedure

1. Generate a session.

   You can specify the time until a session times out.

   REST API: POST /v1/objects/sessions

   CLI: session_create

   After running the command, you receive a response indicating session information.

Chapter 9: Editing user authentication settings

Obtaining user authentication settings (CLI or REST API)

Role-based execution is not subject to restriction.

- passwordComplexitySetting: Specifies password complexity.

- passwordAgeSetting: Specifies the password expiration time.

- lockoutSetting: Specifies the lockout setting.

- sessionSetting: Specifies the session.

Procedure

1. Obtain the user authentication settings as follows.

   You can perform this for the cluster master node (primary) only.

   REST API: GET /v1/objects/user-auth-setting

   CLI: user_auth_setting_show

Chapter 10: Obtaining system performance and capacity information

Performance and capacity information of the storage system

You can obtain performance and capacity information about each resource that Virtual Storage Software block is collecting.

The performance information the user can obtain is divided into two types: low-temporal-resolution (hereinafter, low-resolution) information that is collected at 1-minute intervals and high-temporal-resolution (hereinafter, high-resolution) information that is collected at 5-second intervals. Low-resolution information refers to the information obtained from archival records. High-resolution information refers to the latest information. (Some information is only available as low-resolution information.)

Low-resolution information is retained for up to two days. Low-resolution information that exceeds the maximum retention days is deleted in the oldest order.

When a problem occurs in the storage system, obtain and utilize high-resolution information about the specific faulty resource for troubleshooting.

Regularly obtain and utilize low-resolution information to verify whether the system is running normally and any abnormality has occurred.

Obtaining a list of capacity information about volumes (CLI or REST API)

The following information can be obtained.

- id: Volume IDs (uuid)

- capacityUsage: Consumed amount

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Obtain a list of capacity information about volumes.

   REST API: GET /v1/objects/performances/volume-capacities

   CLI: volume_capacity_performance_list

Obtaining capacity information about individual volumes (CLI or REST API)

The following capacity information is obtained in low-resolution for the volume with the specified ID.

- id: Volume IDs (uuid)

- capacityUsage: Consumed amount

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Verify the ID of the volume for which you want to obtain capacity information.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain capacity information.

   Run either of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API: GET /v1/objects/performances/volume-capacities/*<id>*

   CLI: volume_capacity_performance_show

Obtaining a list of performance information about volumes (CLI or REST API)

The following information can be obtained.

- id: Volume IDs (uuid)

- readIOPS: Number of read operations per second

- writeIOPS: Number of write operations per second

- readTransferRate: Read transfer amount per second

Chapter 10: Obtaining system performance and capacity information

- writeTransferRate: Write transfer amount per second

- readResponseTime: Average time required to respond to the read command of the volume

- writeResponseTime: Average time required to respond to the write command of the volume

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Verify the IDs of the volumes for which you want to obtain performance information.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain a list of performance information.

   To obtain high-resolution performance information, run the command with the volume IDs specified in the query parameter. You can specify a maximum of 32 volume IDs.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API (low-resolution): GET /v1/objects/performances/volumes

   REST API (high-resolution): GET /v1/objects/detail-performances/volumes

   CLI (low-resolution): volume_performance_list

   CLI (high-resolution): volume_detail_performance_list

Obtaining performance information about individual volumes (CLI or REST API)

The following information can be obtained for the volume with the ID specified.

- id: Volume ID (uuid)

- readIOPS: Number of read operations per second

- writeIOPS: Number of write operations per second

- readTransferRate: Read transfer amount per second

- writeTransferRate: Write transfer amount per second

- readResponseTime: Average time required to respond to the read command of the volume

- writeResponseTime: Average time required to respond to the write command of the volume

Before you begin

Required role: VpsStorage or VpsMonitor

Procedure

1. Verify the ID of the volume for which you want to obtain performance information.

   If you use the CLI to specify a volume by name, check the name of the volume.

   REST API: GET /v1/objects/volumes

   CLI: volume_list

2. Obtain performance information.

   Run one of the following commands with the volume ID specified.

   If you use the CLI, you can specify a name instead of the ID of the volume.

   REST API (low-resolution): GET /v1/objects/performances/volumes/*<id>*

   REST API (high-resolution): GET /v1/objects/detail-performances/volumes/*<id>*

   CLI (low-resolution): volume_performance_show

   CLI (high-resolution): volume_detail_performance_show

Chapter 11: Obtaining VPS information

Obtaining a list of VPS information

The following information can be obtained for a VPS to which to belong.

- id: ID of a virtual private storage (VPS)

- name: Name of a virtual private storage (VPS)

- upperLimitForNumberOfUserGroups: Upper limit for the number of user groups for a virtual private storage (VPS)

- numberOfUserGroupsCreated: Number of created user groups for a virtual private storage (VPS)

- upperLimitForNumberOfUsers: Upper limit for the number of users for a virtual private storage (VPS)

- numberOfUsersCreated: Number of created users for a virtual private storage (VPS)

- upperLimitForNumberOfSessions: Upper limit for the number of sessions for a virtual private storage (VPS)

- numberOfSessionsCreated: Number of created sessions for a virtual private storage (VPS)

- upperLimitForNumberOfServers: Upper limit for the number of compute nodes for a virtual private storage (VPS)

- numberOfServersCreated: Number of created compute nodes for a virtual private storage (VPS)

- upperLimitForNumberOfHbas: Upper limit for the number of initiators for a virtual private storage (VPS)

- numberOfHbasCreated: Number of created initiators for a virtual private storage (VPS)

- upperLimitForNumberOfVolumeServerConnections: Upper limit for the amount of information about connections between volumes and compute nodes for a virtual private storage (VPS)

- numberOfVolumeServerConnectionsCreated: Amount of information about connections between created volumes and compute nodes for a virtual private storage (VPS)

- volumeSettings: Information about volumes of a virtual private storage (VPS)

  - poolId: ID of a storage pool to be used

  - upperLimitForNumberOfVolumes: Upper limit for the number of volumes for a virtual private storage (VPS)

  - numberOfVolumesCreated: Number of created volumes for a virtual private storage (VPS)

  - upperLimitForCapacityOfVolumes: Upper limit for the total capacity of volumes for a virtual private storage (VPS)

  - capacityOfVolumesCreated: Total capacity of created volumes for a virtual private storage (VPS)

  - upperLimitForCapacityOfSingleVolume: Upper limit for the capacity of a single volume for a virtual private storage (VPS)

  - qosParam

    - upperLimitForIops: Upper limit for volume performance [IOPS] for a virtual private storage (VPS)

    - upperLimitForThroughput: Upper limit for volume performance [MB/s] for a virtual private storage (VPS)

    - upperLimitForAlertThreshold: Alert threshold (second) for the upper limit of volume performance for a virtual private storage (VPS)

Before you begin

- Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Obtain a list of VPS information.

   REST API: GET /v1/objects/virtual-private-storages

   CLI: vps_list

Obtaining information about individual VPSs

The following information can be obtained for a VPS to which to belong.

- id: ID of a virtual private storage (VPS)

- name: Name of a virtual private storage (VPS)

- upperLimitForNumberOfUserGroups: Upper limit for the number of user groups for a virtual private storage (VPS)

- numberOfUserGroupsCreated: Number of created user groups for a virtual private storage (VPS)

- upperLimitForNumberOfUsers: Upper limit for the number of users for a virtual private storage (VPS)

- numberOfUsersCreated: Number of created users for a virtual private storage (VPS)

- upperLimitForNumberOfSessions: Upper limit for the number of sessions for a virtual private storage (VPS)

- numberOfSessionsCreated: Number of created sessions for a virtual private storage (VPS)

- upperLimitForNumberOfServers: Upper limit for the number of compute nodes for a virtual private storage (VPS)

- numberOfServersCreated: Number of created compute nodes for a virtual private storage (VPS)

- upperLimitForNumberOfHbas: Upper limit for the number of initiators for a virtual private storage (VPS)

- numberOfHbasCreated: Number of created initiators for a virtual private storage (VPS)

- upperLimitForNumberOfVolumeServerConnections: Upper limit for the amount of information about connections between volumes and compute nodes for a virtual private storage (VPS)

- numberOfVolumeServerConnectionsCreated: Amount of information about connections between created volumes and compute nodes for a virtual private storage (VPS)

- volumeSettings: Information about volumes of a virtual private storage (VPS)

  - poolId: ID of a storage pool to be used

  - upperLimitForNumberOfVolumes: Upper limit for the number of volumes for a virtual private storage (VPS)

  - numberOfVolumesCreated: Number of created volumes for a virtual private storage (VPS)

  - upperLimitForCapacityOfVolumes: Upper limit for the total capacity of volumes for a virtual private storage (VPS)

  - capacityOfVolumesCreated: Total capacity of created volumes for a virtual private storage (VPS)

  - upperLimitForCapacityOfSingleVolume: Upper limit for the capacity of a single volume for a virtual private storage (VPS)

  - qosParam

    - upperLimitForIops: Upper limit for volume performance [IOPS] for a virtual private storage (VPS)

    - upperLimitForThroughput: Upper limit for volume performance [MB/s] for a virtual private storage (VPS)

    - upperLimitForAlertThreshold: Alert threshold (second) for the upper limit of volume performance for a virtual private storage (VPS)

Before you begin

- Required role: VpsSecurity, VpsStorage, or VpsMonitor

Procedure

1. Obtain a list of VPS information.

   REST API: GET /v1/objects/virtual-private-storages

   CLI: vps_list

2. Obtain VPS information.

   Run either of the following commands with the VPS ID specified.

   REST API: GET /v1/objects/virtual-private-storages/*<id>*

   CLI: vps_show

Chapter 12: Obtaining common information

Obtaining versions and names of APIs (CLI or REST API)

The following information can be obtained.

Role-based execution is not subject to restriction.

- apiVersion: Versions of APIs

- productName: Names of APIs

Procedure

1. Obtain information about APIs such as version information.

   REST API: GET /configuration/version

   CLI: version_show

Obtaining a list of job information (CLI or REST API)

The following information about the job you issued can be obtained.

Role-based execution is not subject to restriction.

- jobId: IDs (uuid) of jobs

- self: URLs for accessing job information

- userId: IDs of the users who issued APIs that triggered creation of jobs

- status: Progress of jobs

- state: Status of jobs

- createdTime: Time when jobs were created

- updatedTime: Time when job status was updated

- completedTime: Time when jobs were completed

- request: Request information

- affectedResources: URLs for accessing the resources which are the operation targets of jobs

- error: Job error information

Procedure

1. Obtain a list of jobs.

   REST API: GET /v1/objects/jobs

   CLI: job_list

   > 📄 **Note:**
   >
   > - If you can't find job information, filter by the query parameter when you obtain a list of job information, or specify the job ID to get job information individually. For information about the query parameter, see *Hitachi Virtual Storage Software Block REST API Reference*.
   >
   > - Up to 100,000 jobs can be retained in a storage system. Job information that exceeds the maximum number of retentions is deleted in the oldest order. However, because it also includes jobs that run automatically inside the storage system, the total number of job information that can be obtained is less than this number.

Obtaining information about individual jobs (CLI or REST API)

The following information about the job you issued can be obtained.

Role-based execution is not subject to restriction.

- jobId: ID (uuid) of the intended job

- self: URL for accessing the information about the intended job

- userId: ID of the user who issued an API that triggered creation of the intended job

- status: Progress of the intended job

- state: Status of the intended job

- createdTime: Time when the intended job was created

- updatedTime: Time when the job status was updated

- completedTime: Time when the job was completed

- request: Request information

- affectedResources: URL for accessing the resource which is the operation target of the intended job

- error: Job error information

Procedure

1. Obtain the ID of the intended job.

   REST API: GET /v1/objects/jobs

   CLI: job_list

2. Obtain information about the intended job.

   Run either of the following commands with the job ID specified.

   REST API: GET /v1/objects/jobs/*<jobId>*

   CLI: job_show

   > 📄 **Note:**
   >
   > Up to 100,000 jobs can be retained in a storage system. Job information that exceeds the maximum number of retentions is deleted in the oldest order. However, because it also includes jobs that run automatically inside the storage system, the total number of job information that can be obtained is less than this number.

Obtaining messages to be displayed at login and during CLI Basic authentication (CLI or REST API)

You can obtain messages to be displayed in the warning banner during CLI Basic authentication.

Role-based execution restriction is not subject to restriction.

Procedure

1. Obtain messages.

   REST API: GET /configuration/login-message

   CLI: login_message_show

   If nothing is displayed, no message is set.

   > 📄 **Note:**
   >
   > If the CLI execution does not require authentication or it performs session authentication, nothing is displayed even if a message is set.

Appendix A: ALUA configuration guidelines

Required preparations

VMware

Common settings

In the case of VMware ESXi 6.X (including 5.X), apply the following updates and KBs.

- VMware ESXi5.5 Update3 and KB2146717

- VMware ESXi6.0 Update2 and KB2145663

Note that you do not need to change the setting value for SCSI Timeout because the SCSI Timeout value is 120 seconds by default.

Alternate path software setting

No additional settings are required.

Inter-guest cluster setting

For a device in which the inter-guest cluster (MSCS, etc.) uses reserved volumes (SCSI-2, SCSI-3), change "perennially-reserved" to TRUE from each VMware ESXi (see each KB). Make this setting irrespective of the storage configuration.

- ESXi/ESX hosts with visibility to RDM LUNs being used by MSCS nodes with RDMs may take a long time to start or during LUN rescan (1016106)

    https://kb.vmware.com/s/article/1016106

Windows

Common settings

Change the setting value of SCSI Timeout by referring to *Installing a compute node* in this manual.

Alternate path software setting (in the case of MPIO)

You must change the registry settings. Set the following registry values (see the following URL):

http://technet.microsoft.com/en-us/library/ee619749(v=ws.10).aspx

- Set PathVerifyEnabled to 1.

- Set UseCustomPathRecoveryInterval to 1.

- Set PathRecoveryInterval (default: 40) to a value smaller than PDORemovePeriod (default: 20).

Edit registry values carefully (see the following URL):

http://support.microsoft.com/kb/136393

Cluster settings

No additional settings are required.

Linux

Change the setting value of SCSI Timeout and multipath settings by referring to *Installing a compute node* in this manual.

ALUA setting procedure

VMware (in the case of NMP)

This procedure allows the compute node to recognize whether the ALUA is enabled (specified on the storage side).

No procedure is available for allowing the compute node to recognize an ALUA priority change set on the storage side. ALUA priority changes are recognized automatically.

Procedure

1. Verify that a dedicated SATP has not been created yet.

   If the following command outputs nothing, you can judge that an SATP has not been created. If an SATP has already been created, you do not need to perform the following steps.

   ```
   # esxcli storage nmp satp rule list | grep Hi-SDS
   ```

2. Create a dedicated SATP.

   ```
   # esxcli storage nmp satp rule add -V HITACHI -M "^Hi-SDS*" -s VMW_SATP_ALUA -c
   tpgs_on --psp="VMW_PSP_RR"
   ```

3. Reboot the compute node to allow the setting to be recognized.

Windows (in the case of MPIO)

No procedure is available for allowing the compute node to recognize whether ALUA is enabled (specified on the storage side).

To allow the compute node to recognize an ALUA priority change set on the storage side, scan hardware changes from the device manager. Run the following command for verification.

```
# mpclaim -s  -d XXX
```

Priority paths and non-priority paths are indicated as follows:

Priority path: Active/Optimized

Non-priority path: Active/Unoptimized

Linux system (in the case of Device Mapper)

The following procedure allows the compute node to recognize whether the ALUA is enabled (specified on the storage side).

Procedure

1. Change the settings for the following items in /etc/multipath.conf:

Item	Setting value
vendor	"HITACHI"
product	"Hi-SDS*"
path_grouping_policy	group_by_prio
prio	alua
detect_path_checker/ detect_checker [1]	no
path_checker [2]	readsector0
dev_loss_tmo	Refer to your distribution's documentation and set the maximum value.
1. The parameter name differs depending on your distribution. Refer to your distribution's documentation and set the appropriate parameter name.	
2. If a failure occurs in a storage node of Virtual Storage Software block, when a value other than readsector0 is set, paths might be wrongly blocked. For this reason, make sure that you set readsector0.	

2. Reload the alternate path daemon.

   # service multipathd reload

Appendix A: ALUA configuration guidelines

Example

No procedure is available for letting the compute node recognize an ALUA priority change set on the storage side. ALUA priority changes are recognized automatically.

You can verify the ALUA priority set on the storage by running the following command:

multipath -ll

Priority paths and non-priority paths are indicated as follows:

Priority path: prio=50 status=active

Non-priority path: prio=10 status=enable

Virtual Storage Software block does not support a mixed configuration of ALUA devices and non-ALUA devices.

Appendix B: Processing time for snapshot operations

Estimated processing time for snapshot operations

The processing time (in seconds) described here is estimated on the assumption that one snapshot operation is performed while no I/O is issued and no other operation is being performed. Estimate the processing time based on the table below.

- The processing time for restoration and deletion varies depending on totalCapacity and snapshotConcordanceRate of the volume.

- The processing time for PrepareAndFinalize and Prepare includes the time required for S-VOL creation.

- The processing time for restoration might be about twice as long as the time described in the table below depending on the target generation of restoration and the amount of differential data between the copy-source volume and the copy-destination volume of restoration. To restore data from a generation other than the latest one or from a generation of S-VOL to which data is written, estimate the processing time by doubling it.

- If the load of the copy processing of restoration has a high impact on I/O, the copy processing might be stopped by the storage controller. If the copy processing does not progress even if the storage node is not overloaded, collect the dump log file for the storage node and contact customer support.

totalCapacity [GiB]	snapshotConcordanceRate [%]	Prepare and take a snapshot (PrepareAndFinalize) [second]	Prepare(Prepare) [second]	Take a snapshot (Finalize) [second]	Restore[second]	Delete[second]
51200	100	36	38	0	9727	1350
16384	100	14	13	0	3723	573
4096	100	8	8	0	961	162
1024	100	7	6	0	241	54

totalCapacity [GiB]	snapshot ConcordanceRate [%]	Prepare and take a snapshot (PrepareAndFinalize) [second]	Prepare(Prepare) [second]	Take a snapshot (Finalize) [second]	Restore[second]	Delete[second]
256	100	6	6	0	61	29
128	100	8	6	0	60	22
128	75	-	-	-	2583	44
128	50	-	-	-	5045	74
128	0	-	-	-	9729	101
-: Not available						

You can delete all snapshot volumes (S-VOL or P/S-VOL) created from the P-VOL by running the snapshot deletion command with the volume ID of the P-VOL and true specified for masterVolumeId and snapshotTree (for the REST API) and for --master_volume_id and --snapshot_tree (for the CLI).

The deletion time can be calculated as follows:

Number of snapshot volumes[1] x deletion time per volume + snapshot configuration deletion time + 600 seconds[2]

1. Number of all the S-VOLs or P/S-VOLs that were created from the specified P-VOL.

2. Waiting time for copy stop processing required only when conflict with snapshot copy processing within the system occurs.

The deletion time per volume and snapshot configuration deletion time vary depending on the size of the P-VOL as follows:

Size of the P-VOL [GiB]	Deletion time per volume [second]	Snapshot configuration deletion time [second]
128	10	4
256	10	4
1024	10	4
4096	11	10
8192	11	15
16384	15	32

Appendix B: Processing time for snapshot operations

Size of the P-VOL [GiB]	Deletion time per volume [second]	Snapshot configuration deletion time [second]
32768	19	64
51200	24	92
65536	29	107
98304	41	161

Glossary

Auto recovery

See *Storage node auto-recovery* in the Glossary.

base license

A license that provides basic functionality.

blocked, blocking, blockage

A state for a storage or resources that comprise a storage where I/O operations cannot be performed.

BMC network

Network that connects the storage node BMC and the controller node. This network is used to operate the BMC from the controller node.

BMC port

The port that is on a storage node and is used for connection to the BMC network.

capacity balancing

Function of moving volumes automatically from high capacity usage storage controllers to low capacity usage storage controllers when capacity usage is not balanced among storage controllers.

cluster master node (primary)

A storage node within the storage cluster that has the role of managing the entire storage cluster.

cluster master node (secondary)

A storage node in the storage cluster that is responsible for managing the entire storage cluster in the event of failure of the cluster master node (primary).

cluster worker node

A storage node in the storage cluster that does not have the role of managing the entire storage cluster.

compute network

A network between a compute node and a storage node. Used for input / output of user data.

compute node

A node that the application of the user operates and instructs input / output of user data to the storage node. A host connected to the compute port.

compute port

(Virtual machine) The virtual port that is on a storage node and connects to the compute network.

(Bare metal) The port that is on a storage node and connects to the compute network.

configuration backup file

Backup file of storage cluster configuration information.

Configuration file

(Virtual machine) Generic term for VSS block configuration file and VM configuration file.

(Bare metal) A synonym for the VSS block configuration file.

Console interface

The interface of a storage node console (such as a virtual console via BMC).

control network

(Virtual machine) The network between the controller node and the storage node or maintenance node. It is used for Virtual Storage Software block management operation and communication with external service such as SNMP and NTP.

(Bare metal) The network between the controller node and the storage node. It is used for Virtual Storage Software block management operation and communication with external service such as SNMP and NTP.

control port

(Virtual machine) The virtual port that is on a storage node and connects to the control network.

(Bare metal) The port that is on a storage node and connects to the control network.

controller node

A management node used to instruct Virtual Storage Software block's management function (volume creation, etc.).

data migration

A functionality to migrate data from an external storage system into Virtual Storage Software block in volume units.

disk controller

Hardware required to use a drive.

drive

A physical device that stores user data and the OS. Common name for SSDs and HDDs.

drive data relocation

Function of balancing data capacity among storage nodes (to optimize capacity efficiency of each storage node) when capacity becomes unbalanced among storage nodes due to storage node addition or removal.

event log

A file that records the operation of the system. In Virtual Storage Software block, it refers to the log for the purpose of fault notification.

Failover

Switching the cluster master (secondary) to the cluster master (primary) in the event of failure of the cluster master (primary).

fault domain

A group of storage nodes sharing power system and network switch. A configuration for making it possible to continue the operation of storage even if the storage nodes in a group collectively become abnormal.

Glossary

initiator

An endpoint on the compute node side when accessing a volume from a compute node.

internode network

Network between storage nodes. Used for communication of user data and management information between storage nodes.

internode port

(Virtual machine) The virtual port that is on a storage node and connects to the internode network.

(Bare metal) The port that is on a storage node and connects to the internode network.

license key

Key to activate the corresponding license in Virtual Storage Software block.

maintenance blockage

See *Storage node maintenance blocking* in the Glossary.

maintenance node

VM that is configured inside some of the storage nodes, and which is used to configure and manage Virtual Storage Software block.

maintenance recovery

See *Storage node maintenance recovery* in the Glossary.

multi-tenancy function

Function to allow resources of a storage in a large storage system to be distributed to and shared by multiple tenants (companies and divisions). A storage distributed to each tenant is called VPS (Virtual Private Storage).

normal volume

Volume that is neither P-VOL, S-VOL, nor P/S-VOL.

other volume capacity

Total capacity of snapshot volumes (S-VOLs and P/S-VOLs).

OVA

An acronym for the Open Virtualization Appliance/Application. The following files are bundled into one tar ball.

1. OVF file contains the virtual machine attributes etc.

2. Disk image or ISO image created by certain Hypervisor software.

3. Manifest file contains hash value for each file (mf option).

4. Certification file for digital signage for Manifest files (cert option).

OVF

Acronym for Open Virtualization Format. OVF is a standard format designed to allow different virtualization software to exchange virtual machine image files with each other.

P-VOL

Volume of the copy source.

Glossary

P/S-VOL

Volume having both the P-VOL and S-VOL attributes in a snapshot tree in cascade configuration.

physical node

In an environment where storage is used, a physical server that belongs to that environment.

program product license

A license provided on a per-function basis.

protection domain

Setting for limiting the range of failure if an error occurs in a storage node or the network between storage nodes.

provisioned volume capacity

Total capacity of normal volumes and snapshot volumes (P-VOLs).

rebuild

Function of automatically restoring redundancy of data whose redundancy was reduced due to a drive failure or storage node failure.

Rebuild capacity

Capacity in a storage pool secured for Data rebuild at the time of drive failure.

Representative storage node

A storage node that is used to configure a storage cluster in the setup procedure for the bare metal model. This node is different from a cluster master node (primary).

S-VOL

The copy destination volume.

scale out

A method of increasing the number of CPUs, memory capacity, and the number of drives by adding storage nodes to improve system performance and capacity.

scope

The range of resources that users can operate. A scope is set for a user group. A scope for a user is determined according to the user group to which the user belongs.

snapshot volume

Volume that is either a P-VOL, S-VOL, or P/S-VOL.

spare node

Standby storage node used for the spare node function.

spare node function

Function to allow restoration of redundancy by performing spare node switchover. Spare node switchover from a faulty storage node to a storage node that is registered as a standby storage node in the storage cluster is performed when the faulty storage node cannot be restored by the auto-recovery function.

storage cluster

A virtual storage system built from multiple storage nodes.

storage controller

Part of Virtual Storage Software block processes that manage storage node capacities and volumes.

storage controller relocation

Function of optimizing the number of the storage controllers of each storage node when the number of the storage controllers becomes unbalanced among storage nodes due to storage node addition or removal.

storage node

Physical server to which the CPU, memory, and drives that comprise Virtual Storage Software block are assigned. Alternatively, this term refers to a process group of Virtual Storage Software block software running on storage nodes.

storage node addition

A process of adding a storage node to a storage cluster.

Storage node auto-recovery

Function to execute self-diagnosis and self-recovery by a storage node to recover the storage node from server failures due to software factors (firmware, driver, and so on) or due to temporary network problems between storage nodes.

Storage node maintenance blocking

Process of separating a storage node from a storage cluster temporarily and placing the storage node in a status that allows for part replacement or other maintenance.

Storage node maintenance recovery

Process of returning a storage node to the available status again after it was blocked by manual operation or due to a failure.

storage node removal

A process of removing a storage node from a storage cluster.

storage node replacement

A functionality or process that manually recovers a blocked storage node.

Replace the following to recover the blocked storage node.

(Virtual machine) Storage node VM

(Bare metal) Physical node

storage pool

Logical user data storage area that combines multiple drives.

storage software

The Virtual Storage Software block software that realizes a storage cluster.

system administrator

Administrator who manages the entire system.

Glossary

target

> An endpoint on the storage cluster side when accessing a volume from a compute node.

temporary volume capacity

> Total capacity of volumes created temporarily by Data migration and Capacity balance.

thin provisioning

> Method of creating a virtual storage in which the minimum required capacity is initially secured, and then expanded as required.

virtual machine (VM)

> Virtual machine.

virtual private storage

> Virtual storage logically divided from a storage cluster in a multi-tenancy configuration.

volume

> A logical device that mounts on a compute node to read or write user data.

volume migration

> Moving volumes (existing on a storage node to be removed) to another storage node.

volume path

> Connection information between a compute node and a volume. One of the setting information necessary for using a volume from a compute node.

VPS

> Acronym for Virtual Private Storage. See *virtual private storage* in the Glossary.

VPS administrator

> Administrator who manages a virtual private storage (VPS) in a multi-tenancy configuration.

Hitachi Vantara