FUJITSU

White Paper

Best Practices for OpenStack and FUJITSU Storage ETERNUS AF series and ETERNUS DX series

In cloud system configurations, it is important to select the best storage pool configuration that can utilize the advantages of both the infrastructure software and the storage systems.
The combination of OpenStack, which is infrastructure software, and the FUJITSU Storage ETERNUS AF series or ETERNUS DX series can provide the best storage pool configuration for cloud systems.

Table of Contents

Preface

OpenStack has been developed as infrastructure software that provides IaaS.
Since OpenStack consists of many components, configurations are flexible. At the same time, configurations should be selected after careful consideration, particularly since the component (Cinder) that manages the block storage is a relatively new function, and therefore the storage system selection requires thorough review.

OpenStack is licensed under the Apache Software License 2.0, which has several distributions. Although OpenStack can be operated on various platforms, Linux is the platform that is the most widely used.
And Red Hat Enterprise Linux is the most used Linux distribution. OpenStack distributions that are provided by Red Hat, Inc. are Red Hat Enterprise Linux OpenStack Platform (non-free) and RDO (free).

This document provides configurations for storage systems when used in systems with Red Hat Enterprise Linux 7, which is the leading and most popular distribution for OpenStack, and with RDO.
The procedures described from "3. Best Practice for RDO and the ETERNUS AF/DX" through "5. Best Practice Configuration" are current as of October 2015.

The information stated in this document is current as of November 2019.

■ Target Readers
   This document targets readers who have knowledge of OpenStack and the ETERNUS AF series or the ETERNUS DX series, and are looking to select a storage system for cloud infrastructure using OpenStack.

■ Naming Conventions
   The following abbreviation is used in this document.

   - FUJITSU Storage ETERNUS AF S3/S2 series and ETERNUS DX S5/S4 series ....ETERNUS AF/DX

1. OpenStack Overview

OpenStack consists of many components that interoperate with each other through the API.
The relationship of the components is described below.

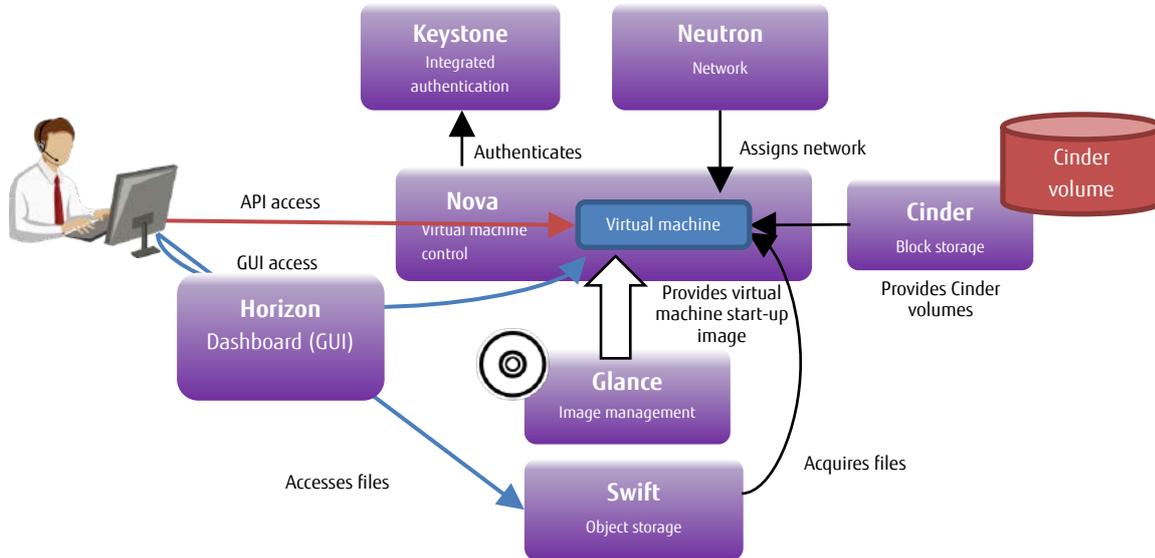Figure: Relationship of components and functional overview

Table: Functional overview of components

Component name	Functional overview
Nova	Function that is the basis of OpenStack which performs integrated management for component interoperability and creates, starts, and stops virtual machines
Horizon	Function that provides the OpenStack Web GUI (or dashboard)
KeyStone	Function that manages access authentication to the dashboards and APIs
Glance	Function that manages images (or templates for virtual machines)
Swift	Function that defines static files as objects, and distributes and saves them to an object storage
Cinder	Function that creates persistent block storage and manages connections to the virtual machines
Neutron	Function that creates and manages virtual network connection devices (such as routers and L2/L3 switches)

Nova, which is the core component, creates virtual machines in conjunction with other components. To start the virtual machines, Nova specifies a start-up image from Glance when creating the virtual machines.
After the virtual machines are created, Neutron creates a virtual network so that users can use the virtual machines.
Cinder is the function that makes persistent block storage available for the virtual machines. Swift is the function that makes an object storage available.
Cinder and Swift both have their own advantages and users need to select them depending on their purposes.

OpenStack operations are performed through the OpenStack's GUI-based dashboard (Horizon) or OpenStack's command APIs. The OpenStack interface is authenticated with KeyStone.

2. Use of Cinder and Swift

For OpenStack, block storage and object storage can be used as a storage system.
This chapter describes Cinder and Swift that use block storage and object storage, respectively, in OpenStack.

2.1 Block Storage and Object Storage

Disks that are allocated when a virtual machine starts are areas that are deleted when the virtual machine shuts down. Therefore, a persistent device must be prepared as an area to save data. For OpenStack, there are two types of persistent devices: block storage and object storage.
In systems using OpenStack, persistent devices must be provisioned using Cinder and Swift for data retention.

2.1.1 Cinder and Block Storage

Cinder manages the creation and backup of persistent block storage and provides them to the virtual machines.
Actual Cinder volumes are created in an external storage system. To use third-party storage systems, drivers that are provided by the storage vendors are required.
Although the typical purpose of creating Cinder volumes is to store data, Cinder volumes where instances can be started can also be created. In this case, specify the OS start-up image of Glance as a volume source to create Cinder volumes where instances can be started.
To use regular Cinder volumes, connect them to virtual machines that are already started. (The role of Cinder is to create and connect Cinder volumes.) After the Cinder volumes are connected to the virtual machines, and to make them available to users as areas for saving data, the Cinder volumes must be formatted and mounted the same way as general volumes using the OS commands. Cinder volumes that are in use can be disconnected and reconnected to other virtual machines with the data retained.

When compared with Swift, Cinder has advantages when small updates occur frequently because the access is based on blocks and is suited for DB servers in three-tier structure systems and for servers where logs in the system are integrally managed.

2.1.2 Swift and Object Storage

Swift provides a distributed object storage that is accessible regardless of whether the virtual machines are started or not. Except for authentication with KeyStone, environments for Swift can be configured independently from other components. Static files that are to be stored such as movies, images, and log files are recommended, and the files are integrally managed as objects. These objects are accessed, referred to, updated, and deleted through HTTP. (These objects can be accessed from OpenStack Horizon.) Virtual machine images that are managed by Glance, and backups of Cinder volumes can be considered as objects, and Swift can be selected as the destination to store the objects.
The objects are allocated to the Meta data to associate the actual data content with OpenStack and stored in a container (which is equivalent to a folder in Window) that is defined in Swift.
The advantages of Swift are described below:

  - Because identical data is automatically distributed to multiple nodes for data security, service does not stop if an error occurs at a single point. (This is file duplication, not RAID.)
  - Adding servers and storage systems to nodes can increase performance and capacity. Inexpensive devices such as HDDs for PCs can be added.

In general, the features of Swift are low cost and high reliability.
On the other hand, there are disadvantages when object storage is compared with block storage, such as high latency in a system that requires accessing and locking small amounts of data such as DB access since the access is based on objects (such as an image file and log file). In addition, the Swift API may not be supported for general business applications and legacy applications. When OpenStack is used with Swift-API-incompatible applications, the interface must be changed.

2.1.3 Usage Overview of Cinder and Swift

Cinder and Swift are used differently based on the user requirements.

Table: Functional comparison between Cinder and Swift

	Cinder (block storage)	Swift (object storage)
Access method	Access to a block device (such as /dev/vdc) through the virtual machines	HTTP REST API
Capacity	Specified by users when Cinder is created	Determined by the total capacity of available physical disks
Access unit	Per block	Per object
Effective usage	Cases when access and lock occur in a small unit such as log files and DB servers	Cases when there is a large amount of files such as images, sounds, and old log files that are rarely written to but frequently read

An example differentiating the use of Cinder and Swift is provided below.

Figure: Differentiating the use of Cinder and Swift
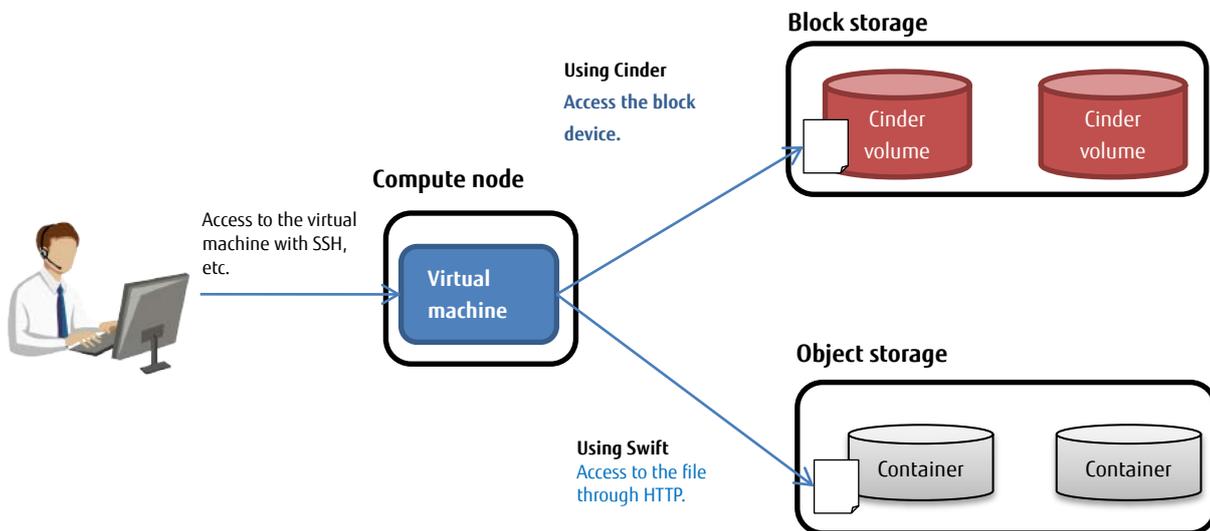
2.2 ETERNUS OpenStack VolumeDriver

ETERNUS OpenStack VolumeDriver is a program that supports the interoperability of the ETERNUS AF/DX and OpenStack.
The ETERNUS AF/DX can be used as block storage with the ETERNUS OpenStack VolumeDriver.
In addition, the ETERNUS AF/DX can be operated from the OpenStack standard interface (or dashboard) and user operations can be centralized.

2.2.1 Overview of ETERNUS OpenStack VolumeDriver

ETERNUS OpenStack VolumeDriver uses ETERNUS AF/DX SMI-S (which is a storage management interface) to manage the ETERNUS AF/DX.

■ Features
- This product can be downloaded free of charge from the Fujitsu website:
  https://www.fujitsu.com/global/support/products/computing/storage/download/
- Fibre Channel (FC) and iSCSI are supported as connection protocols.
- The supported storage system models are as follows:
  ETERNUS AF series
  ETERNUS DX S3 series[1]
    *1: The ETERNUS DX8100 S4 is excluded.

2.2.2 Features of ETERNUS OpenStack VolumeDriver

The following table shows the relationship between the ETERNUS OpenStack VolumeDriver functions and the ETERNUS AF/DX functions.

Table: Relationship between the ETERNUS OpenStack VolumeDriver functions and the ETERNUS AF/DX functions

ETERNUS OpenStack VolumeDriver functions	ETERNUS AF/DX functions
Create Volume	Creating volumes
Delete Volume	Deleting volumes
Attach Volume	Setting access paths to volumes - Adding FC hosts/iSCSI hosts - Creating/modifying LUN groups - Setting host affinities
Detach Volume	Deleting access paths to volumes - Modifying/deleting LUN groups - Deleting host affinities
Create Snapshot	Creating snapshots - Creating Snap Data Volumes (SDVs) - Starting SnapOPC
Delete Snapshot	Deleting snapshots - Stopping SnapOPC - Deleting Snap Data Volumes
Create Volume from Snapshot	Creating volumes from snapshots - Creating a new volume - Copying from a snapshot volume to a volume (Starting OPC)
Get Volume Stats	Obtaining free space information in the storage pool (or a RAID group/Thin Provisioning Pool)
Clone Volume	Cloning volumes - Starting OPC from an existing volume to the newly created volume

Copy Image to Volume	Creating a volume and copying the image managed in Glance to the volume
	- ETERNUS AF/DX functions
	  Creating volumes and setting/deleting access paths (Using Create Volume, Attach Volume, and Detach Volume)
	- Cinder operation
	  Downloading an image from Glance and writing it in the volume

Copy Volume to Image	Uploading volume data as an image to Glance
	- ETERNUS AF/DX functions
	  Setting/deleting access paths (Using Attach Volume and Detach Volume)
	- Cinder operation
	  Reading data from a volume and uploading the image to Glance

Extend Volume	Modifying the volume size (executable only for Thin Provisioning Volumes [TPVs])

Migrate Volume	Moving volumes to a different storage pool
	- ETERNUS AF/DX functions
	  Creating migration destination volumes/setting access paths to migration destination volumes,
	  Deleting migration source volumes/deleting access paths to migration source volumes
	  (Using Create Volume, Delete Volume, Attach Volume, and Detach Volume)
	- Cinder operation
	  Copying data in the migration source volume to the migration destination volume

Retype Volume	Changing the volume type - When the "--migration-policy on-demand" option is specified, Migrate Volume is used - No action is performed on the ETERNUS AF/DX side for cases other than above
Manage Volume	Registering information for ETERNUS AF/DX volumes in the OpenStack controller - Searching volumes
Unmanage Volume	Deleting only volume information that is registered in the OpenStack controller without deleting volumes from the ETERNUS AF/DX - No action is performed on the ETERNUS AF/DX side

* To execute the functions that use OPC/SnapOPC, a separate Advanced Copy Feature license is required.
  To use the snapshot function, a Snap Data Pool Volume (SDPV) must be created and a Snap Data Pool (SDP) must be enabled.
  A RAID group is also required because SDVs are created in RAID groups.
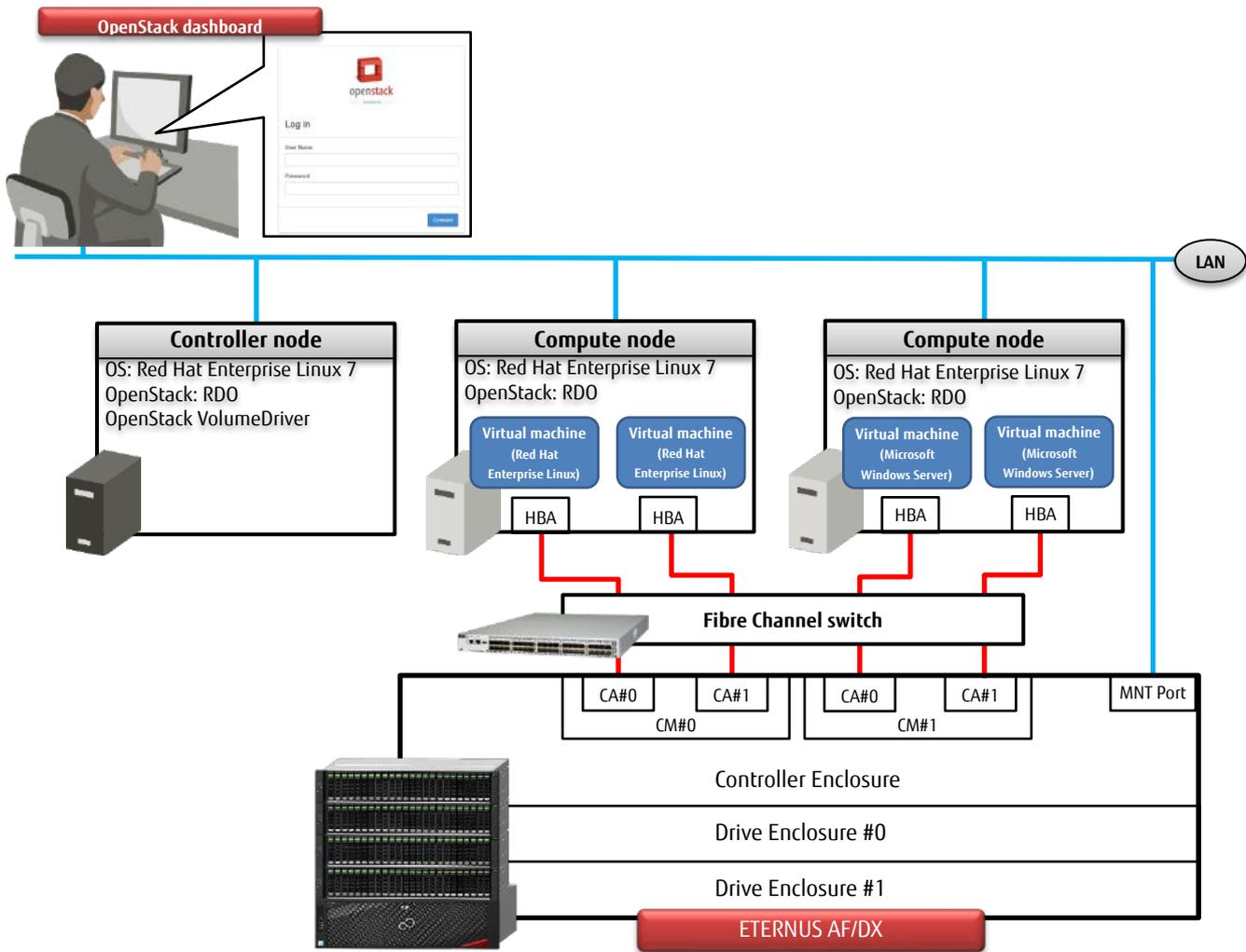
3. Best Practice for RDO and the ETERNUS AF/DX

3.1 Best Practice Configuration

This section describes the best practice configuration for OpenStack and the ETERNUS AF/DX using ETERNUS OpenStack VolumeDriver, allowing the virtual machines and the Cinder volumes to be operated integrally from the OpenStack dashboard.

This document shows the best practice in an environment with Linux (Red Hat Enterprise Linux 7), RDO (Otaca) for OpenStack, and ETERNUS OpenStack VolumeDriver 1.4.
In this configuration, the controller node and the compute nodes are configured separately.

Figure: Configuration overview of the best practice for OpenStack and the ETERNUS AF/DX

3.2 Best Practice for Storage Pools and Drives

The best practice for OpenStack and the ETERNUS AF/DX is a configuration where three types of drives are separated into three storage pools. The best practice configuration is described below.

Table: Best practice configuration

	Storage pool type	Volume type in the ETERNUS AF/DX	Drive type	Usage
Storage pool 1	Thin Provisioning Pool (TPP)	Thin Provisioning Volumes (TPVs)	SSDs *2	Areas where high-speed access is possible
Storage pool 2	Thin Provisioning Pool (TPP)	Thin Provisioning Volumes (TPVs)	Online SAS disks *3	Areas where enormous capacities can be used
Storage pool 3	RAID group	Snapshot related volumes (SDVs, SDPVs)	Online SAS disks *3	Areas for snapshots

*2: SSDs cannot be installed in the ETERNUS DX60 S5/S4. Storage pools 1 and 2 are the same area.

*3: Only SSDs can be installed in the ETERNUS AF S3/S2 series. Storage pools 1 and 2 are the same area and storage pool 3 is also configured with SSDs.

When a controller enclosure (CE) and two drive enclosures (DE) are connected, install drives and configure storage pools as shown in the figure below. (*4)

Figure: Storage pool configuration when two drive enclosures are connected
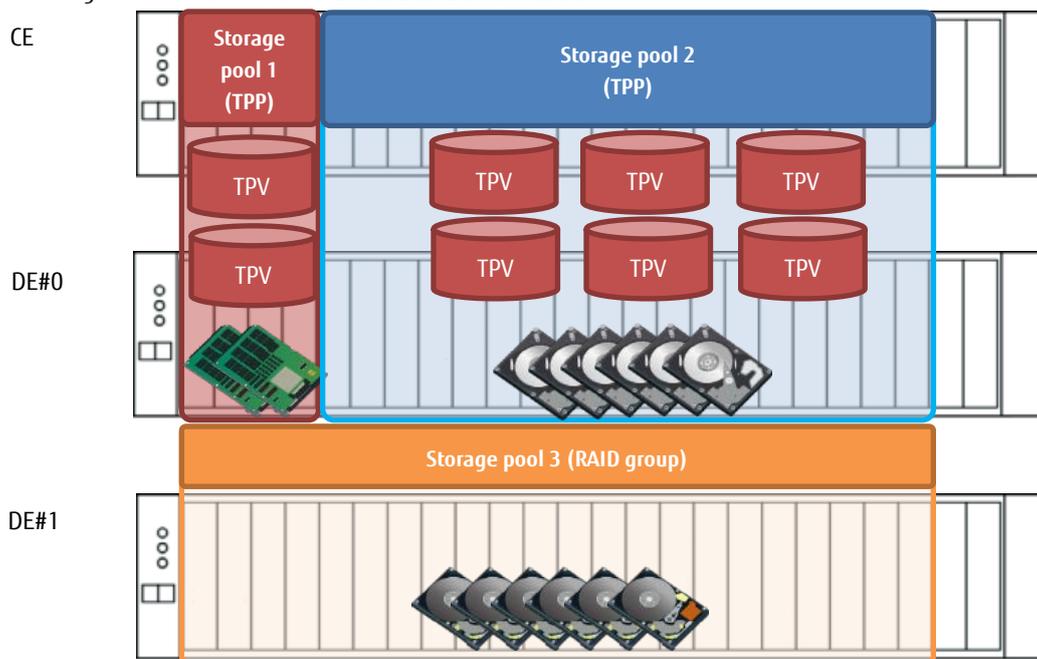
*4: For the ETERNUS DX60 S5/S4 (3.5" type) and the ETERNUS AF250 S2, the above configuration is not available because only one drive enclosure (DE) maximum can be connected.
For the ETERNUS AF150 S3, the above configuration is not available because drive enclosures (DE) cannot be connected to it.

For the ETERNUS DX60 S5/S4 (2.5" type), drive enclosures cannot be connected. Install drives (SAS disks) in the slots of the controller enclosure (CE) and configure two storage pools.

Figure: Example of a storage pool configuration for the ETERNUS DX60 S5/S4
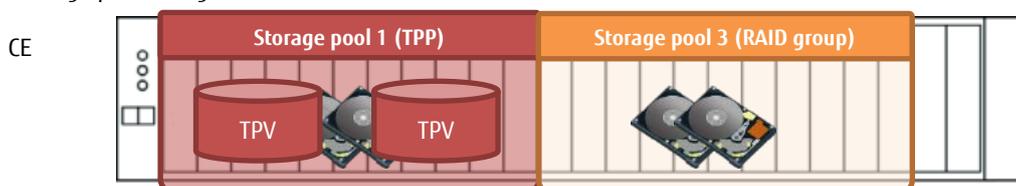
3.3 Best Practice for OpenStack Dashboard Utilization

The best practice for OpenStack and the ETERNUS AF/DX provides superior readiness, flexibility, and load reduction, enabling integrated operations through use of the OpenStack dashboard.

3.3.1 Readiness

Because operations (1) through (3) shown in the figure below can be performed from the OpenStack dashboard, they can be processed quickly.

Figure: OpenStack dashboard operations
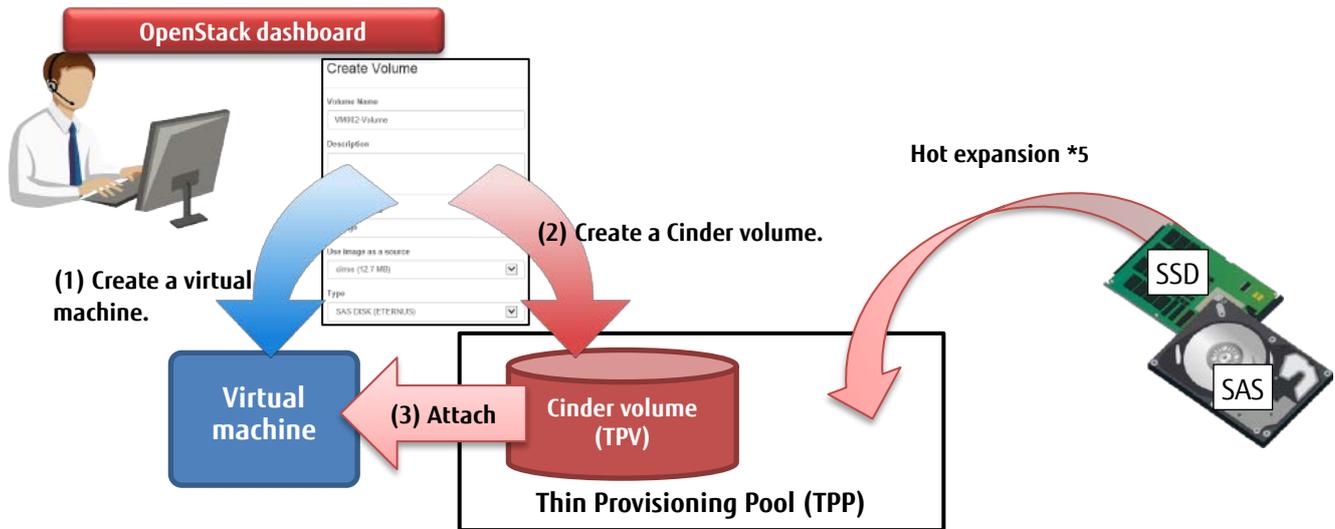
*5: Maintenance work of the ETERNUS AF/DX (such as changing and adding drives) must be performed using ETERNUS Web GUI.

3.3.2 Flexibility

Flexible operation is possible in the following cases:
    a. Virtual machines that require high-speed access responses
      Volumes that can quickly respond can be created just by selecting storage pool 1 that is configured with SSDs.
    b. Virtual machines that require large work capacity
      Volumes that have large capacities can be created just by selecting storage pool 2 that is configured with SAS disks.

Figure: Creating volumes according to the requirements for virtual machines
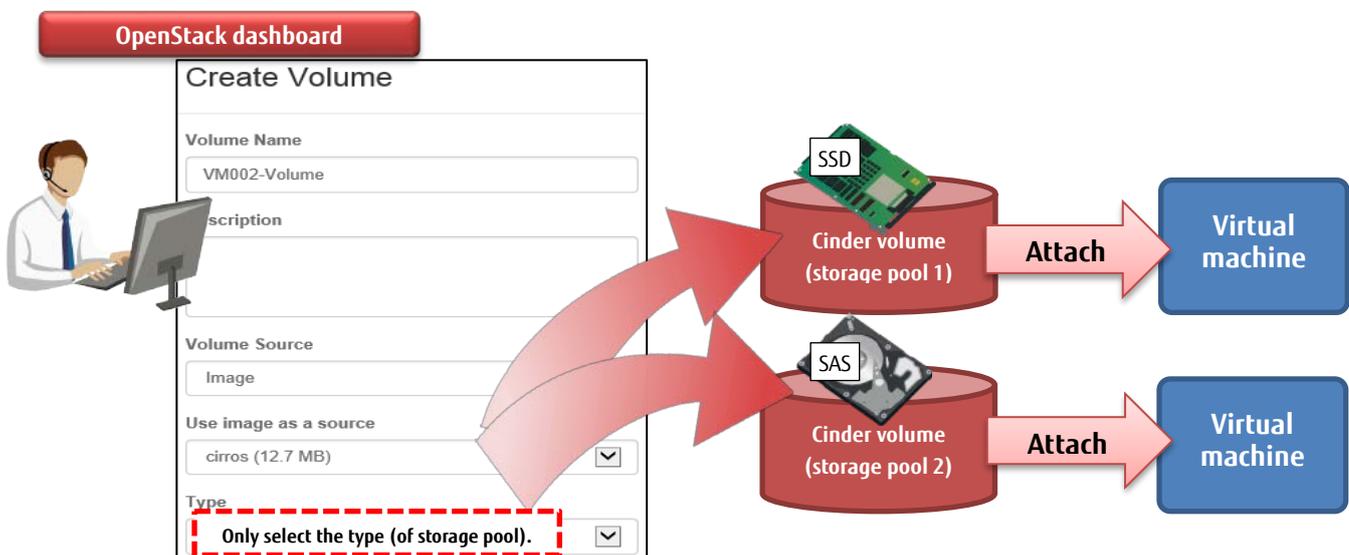
3.3.3 Reducing Loads

In the following cases, the load of the users and servers can be reduced.
- The user's load can be reduced because starting and stopping the virtual OS, and creating backups/snapshots can be integrally managed from the OpenStack dashboard.
- The server's load can be reduced because backups/snapshots are created using the Advanced Copy functions (*6) of the ETERNUS AF/DX.
    *6 To create clones of Cinder volumes, use the OPC function.
       To create snapshots of Cinder volumes, use the SnapOPC function.

Figure: Creating clones and snapshots from the OpenStack dashboard
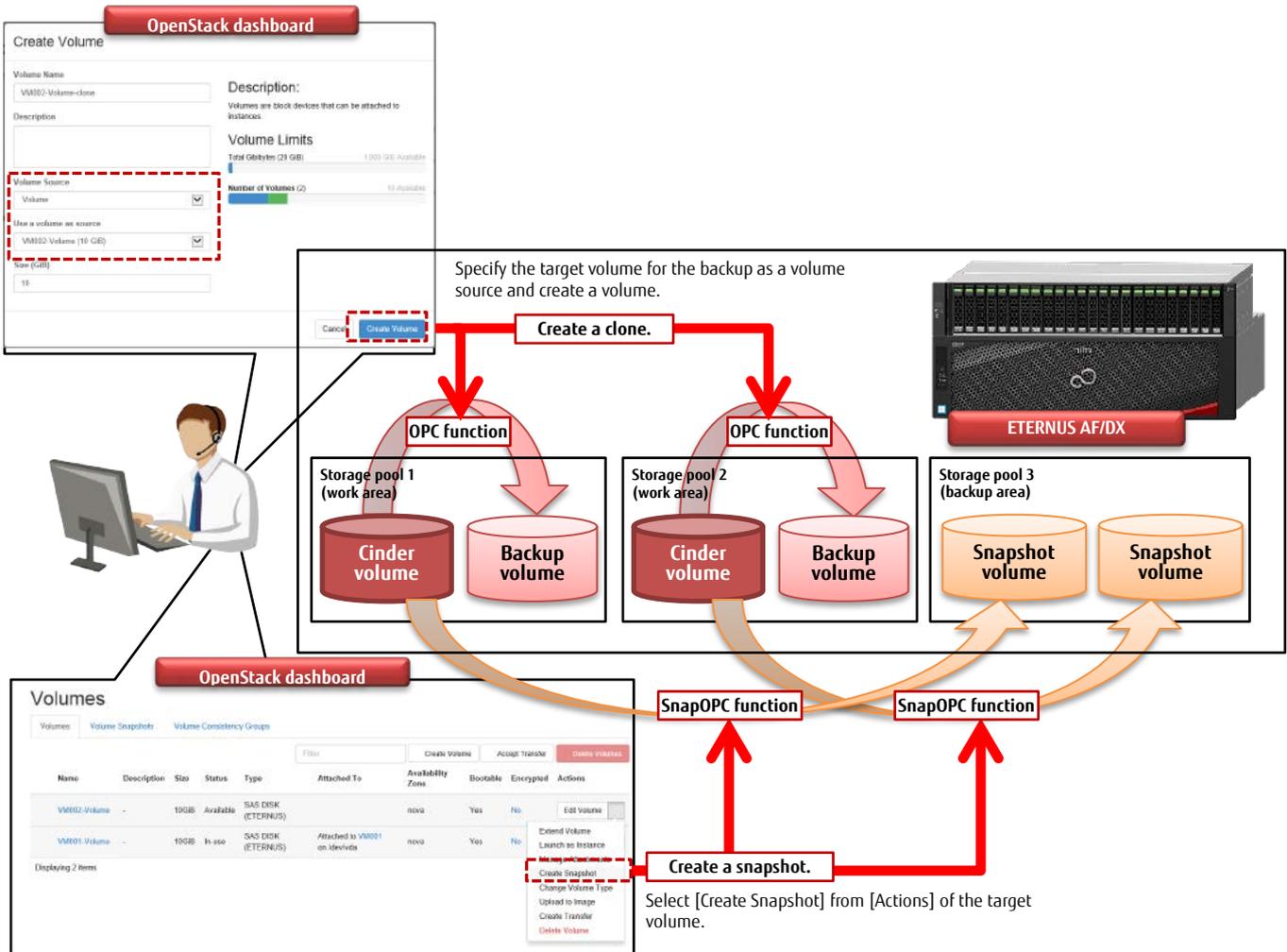
4. Best Practices for Configuring and Making Backups

This chapter provides both the best practices for configuration of virtual machines before operations have started and when backups are taken after operations have started.

4.1 Creating Virtual Machines According to Use

With the best practice configuration for OpenStack and the ETERNUS AF/DX, volumes can be created quickly from the OpenStack dashboard, according to the purpose of the virtual machines.

Both database servers and application servers require high performance. For these servers, select storage pool 1 which is configured with SSDs that are expensive but have a very high response performance.
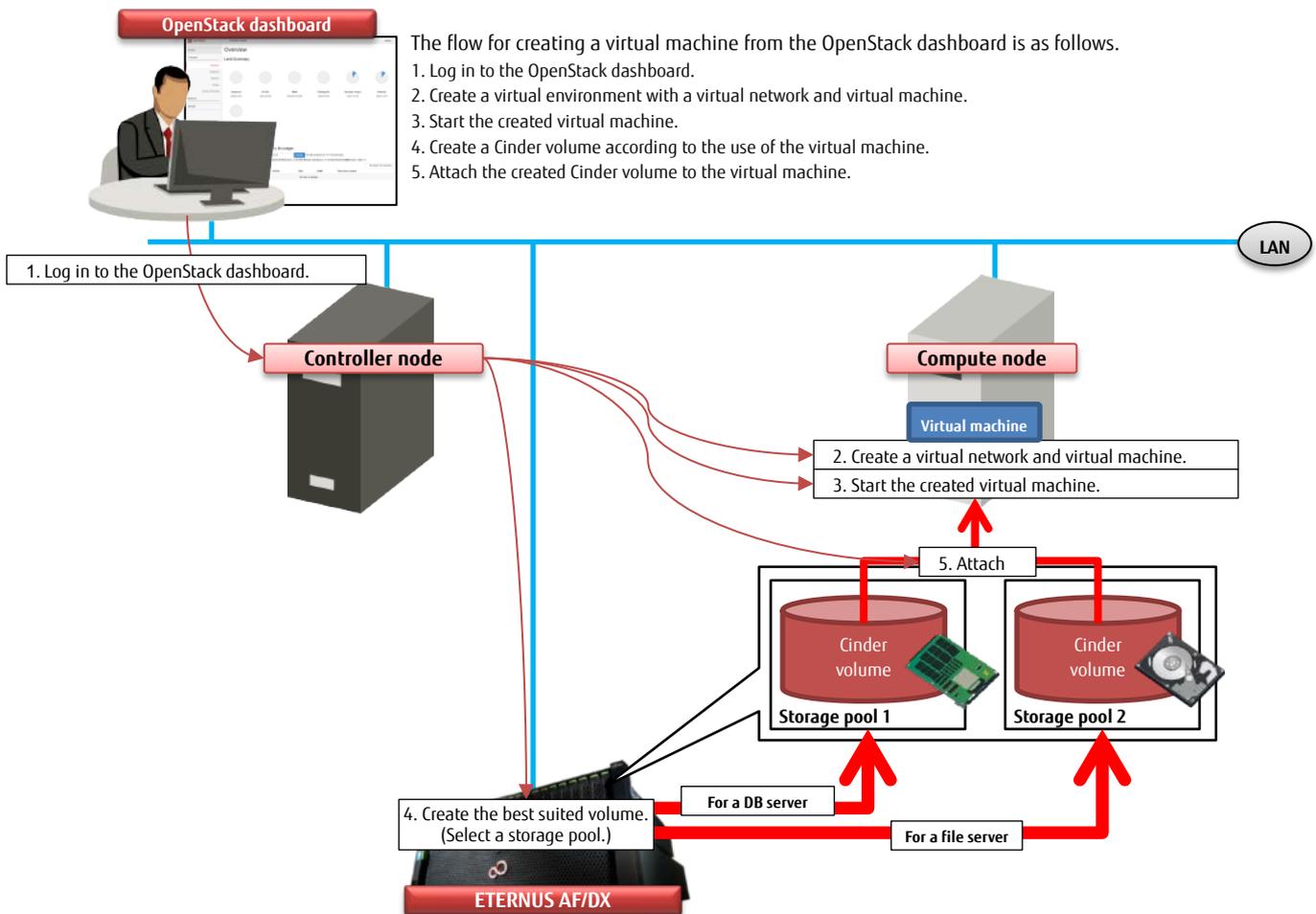File servers require a large amount of capacity for operation, but they do not require high performance because the file servers are accessed through the network.
For these servers, select storage pool 2 which is configured with Online SAS disks that are low cost and have large capacities.

The advantages of the best practice configuration are as follows:
- From creating virtual machines to creating Cinder volumes, operations can be managed by a single user from the OpenStack dashboard.
- Virtual machines can be created any time without the need for adjustments or requests to the engineer in charge of the storage system.
- Highly-reliable volumes, which are characteristic of the storage system, can be used.
- A suitable volume can be created just by selecting a storage pool.

Figure: Flow for creating virtual machines

The flow for creating a virtual machine from the OpenStack dashboard is as follows.
1. Log in to the OpenStack dashboard.
2. Create a virtual environment with a virtual network and virtual machine.
3. Start the created virtual machine.
4. Create a Cinder volume according to the use of the virtual machine.
5. Attach the created Cinder volume to the virtual machine.

1. Log in to the OpenStack dashboard.

Controller node

Compute node

Virtual machine

2. Create a virtual network and virtual machine.

3. Start the created virtual machine.

5. Attach

Cinder volume

Cinder volume

Storage pool 1

Storage pool 2

4. Create the best suited volume. (Select a storage pool.)

For a DB server

For a file server

ETERNUS AF/DX

LAN

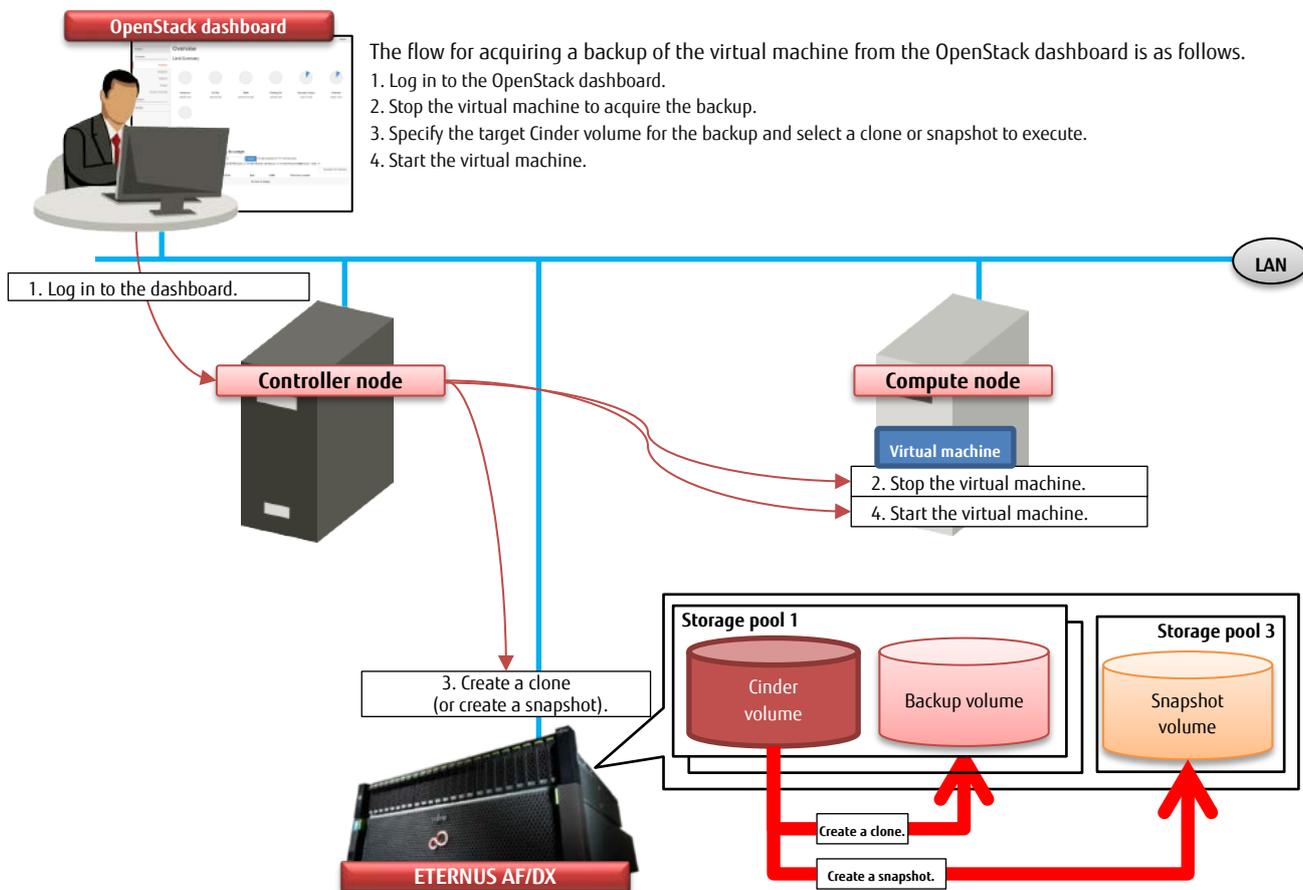## 4.2 Backups of Virtual Machines

For backups of virtual machines, full backups (clone) and generation management (snapshot) must be used separately according to the requirement.
In addition, to secure a backup point and to reduce load on the server, virtual machines must be stopped and adjustments must be made by the person in charge of the server.

The advantages of the best practice for OpenStack and the ETERNUS AF/DX with integrated operations are as follows.
- Adjustments by the person in charge of the server are not required because starting and stopping the virtual machines, and making backups, can be integrally performed from the OpenStack dashboard.
- Operations for acquiring clones and snapshots can be managed from the OpenStack dashboard.
- Load on the server does not occur because the Advanced Copy functions of the ETERNUS AF/DX are used.

Figure: Flow for virtual machine backups

The flow for acquiring a backup of the virtual machine from the OpenStack dashboard is as follows.
1. Log in to the OpenStack dashboard.
2. Stop the virtual machine to acquire the backup.
3. Specify the target Cinder volume for the backup and select a clone or snapshot to execute.
4. Start the virtual machine.

OpenStack dashboard

1. Log in to the dashboard.

LAN

Controller node

Compute node

Virtual machine

2. Stop the virtual machine.
4. Start the virtual machine.

3. Create a clone
(or create a snapshot).

Storage pool 1

Cinder volume

Backup volume

Storage pool 3

Snapshot volume

ETERNUS AF/DX

Create a clone.

Create a snapshot.

5. Best Practice Configuration

The best practice configuration for OpenStack and the ETERNUS AF/DX requires taking some specific items into consideration.
This chapter describes how to configure OpenStack and the ETERNUS AF/DX.

In this document, the best practice is configured with the following settings.

	Storage pool name in the ETERNUS AF/DX	Volume_backend_name in OpenStack	Section name	Volume type
Storage pool 1	openstack-ssd	ssd	FJFC_SSD	SSD
Storage pool 2	openstack-sas	sas	FJFC_SAS	SAS
Storage pool 3	openstack-snap	–	–	–

5.1 Configuring the ETERNUS AF/DX
Follow the instructions in the user's guide of the ETERNUS OpenStack VolumeDriver to configure the ETERNUS AF/DX.

> NOTE for the configuration
>   - At this point, create storage pools using ETERNUS Web GUI or CLI.

5.2 Installing the ETERNUS OpenStack VolumeDriver
Follow the instructions in the user's guide of the ETERNUS OpenStack VolumeDriver to install the ETERNUS OpenStack VolumeDriver.

> NOTES for the configuration
>   - The ETERNUS OpenStack VolumeDriver can be installed using either the interactive mode or the non-interactive mode.
>   - Regardless of whether the ETERNUS OpenStack VolumeDriver is installed using the interactive mode or the non-interactive mode, the Cinder configuration file, "/etc/cinder/cinder.conf", is updated according to the specified contents.
>   - The contents of cinder.conf can be modified after the OpenStack VolumeDriver is installed.

5.3 Configuration Files for the ETERNUS OpenStack VolumeDriver

This section describes the configuration files for the ETERNUS OpenStack VolumeDriver.
(For cinder.conf, only the descriptions that are related to the ETERNUS OpenStack VolumeDriver are shown.)

> NOTES for the configuration
> - When using two storage pools, enter two separate sections in cinder.conf.
> - Enter an ETERNUS AF/DX storage pool that corresponds with the volume type and the area for snapshots in the driver configuration file.

Example: "/etc/cinder/cinder.conf" of the best practice configuration (extract)

```
# A list of backend names to use. These backend names should be backed by a
# unique [CONFIG] group with its options (list value)
enabled_backends=FJFC_SSD,FJFC_SAS                                          (1)


[FJFC_SSD]                                                                  (2)
volume_driver=cinder.volume.drivers.fujitsu.eternus_dx_fc.FJDXFCDriver(3)
cinder_eternus_config_file=/etc/cinder/eternus_ssd.xml                      (4)
volume_backend_name=ssd                                                     (5)


[FJFC_SAS]                                                                  (2)
volume_driver=cinder.volume.drivers.fujitsu.eternus_dx_fc.FJDXFCDriver(3)
cinder_eternus_config_file=/etc/cinder/eternus_sas.xml                      (4)
volume_backend_name=sas                                                     (5)
```

Example of a driver configuration file: /etc/cinder/eternus_ssd.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<FUJITSU>
<EternusIP>xxx.xxx.xxx.xxx</EternusIP>                                      (6)
<EternusPort>5988</EternusPort>                                             (7)
<EternusUser>UserName</EternusUser>                                         (8)
<EternusPassword>Password</EternusPassword>                                 (9)
<EternusPool>openstack-ssd</EternusPool>                                    (10)
<EternusSnapPool>openstack-snap</EternusSnapPool>                           (11)
</FUJITSU>
```

(1)   Specify the names of the backend sections that are to be defined from line (2) onwards.

Enter parameters in lines (2) to (5), (10), and (11) for each storage pool that is to be used.

  (2)   Enter the name of the backend section that is to be defined.
  (3)   Enter this part according to the connection protocol to use. In the example, the connection protocol is FC.
          When the connection protocol is iSCSI, the description is "cinder.volume.drivers.fujitsu.eternus_dx_iscsi.FJDXISCSIDriver".
  (4)   Specify the driver configuration file that was specified with the installer.
          Specify a different file name for each backend in order to enter a storage pool name in the driver configuration file.
  (5)   Specify the "volume_backend_name" that is specified for the additional spec of the volume type.
  (10) Specify either the Thin Provisioning Pool name or the RAID group name that was created in the ETERNUS AF/DX.
  (11) Specify the RAID group name in which snapshots (SDV) are to be created.

(6)   Specify the IP address of the ETERNUS AF/DX.
(7)   Specify the connection port to the SMI-S of the ETERNUS AF/DX.
(8)   Specify the username that is used to connect to the SMI-S of the ETERNUS AF/DX.
(9)   Specify the password that is used to connect to the SMI-S of the ETERNUS AF/DX.

5.4 Associating the Volume Type with "volume_backend_name"
To create a Cinder volume using the OpenStack dashboard or CLI, specify the volume type.
Creating a volume type that corresponds with "volume_backend_name" allows for the creation of a Cinder volume in the storage pool.

> NOTES for the configuration
> - By specifying a volume type name that is easy to identify the use of the storage pool that was created in the ETERNUS AF/DX, volumes can be easily managed on OpenStack.
> - A volume type can be set using either the OpenStack dashboard or CLI.

5.4.1 Creating Volume Types
Create a volume type using the OpenStack dashboard.
Creation method: Select the admin project.
In the dashboard menu, select [Admin] → [System] → [Volumes] → the [Volume Types] tab.
Enter a name in [Name] (for example, SSD) and create the volume type.

Figure: Creating the "SSD" volume type

5.4.2 Adding Specs to the Volume Type

Add "volume_backend_name" in [Volume Type Extra Specs].

Addition method: Select [View Extra Specs] from [Actions] of the created volume type, and then click the [Create] button.

Figure: Volume Type Extra Specs
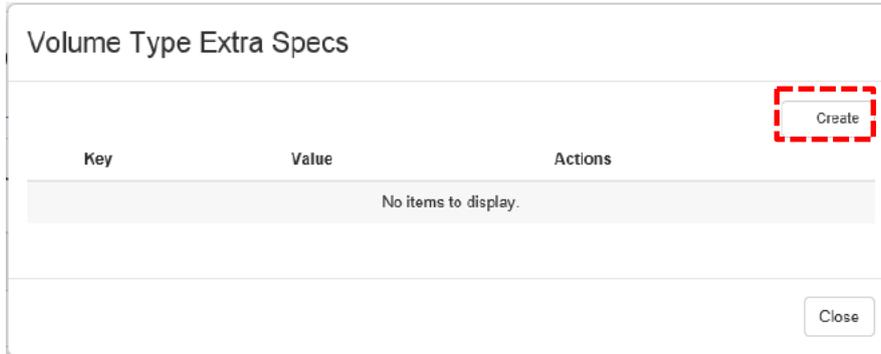
Create an additional spec in [Volume Type Extra Specs].

Table: Parameters for [Key] and [Value]

Item	Parameter to enter	Description
Key	volume_backend_name	Additional spec key for defining a volume type
Value	ssd	Parameter for volume_backend_name that is written in cinder.conf

Figure: Create Volume Type Extra Spec
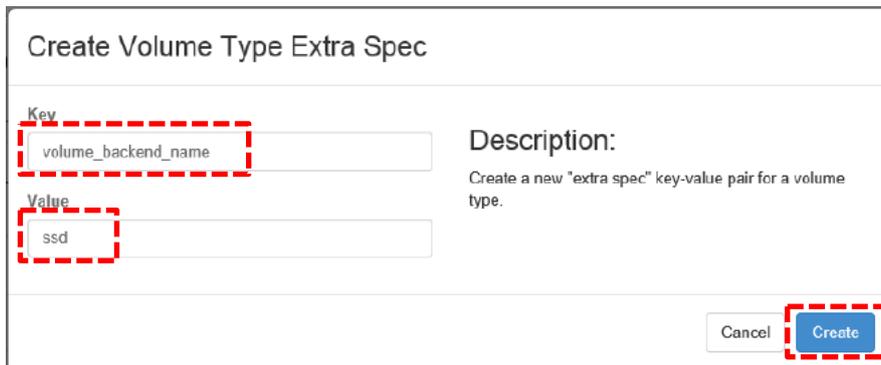
Create a volume type for "sas" as well.

Figure: After the volume type is created
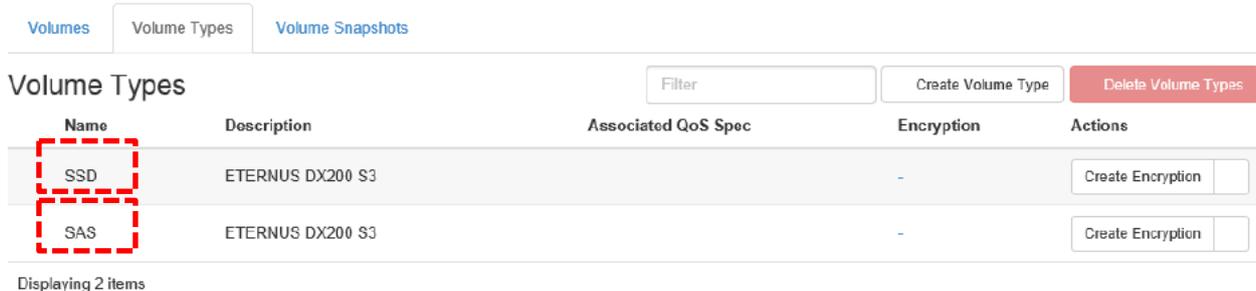
5.5 Selecting the Volume Type When Creating a Cinder Volume

Create a Cinder-managed volume in a storage pool of the ETERNUS AF/DX just by selecting the volume type when creating the volume from the OpenStack dashboard.

Figure: Screen for selecting the volume type

6. Related Solution

Storage Foundation Software "FUJITSU Storage ETERNUS SF" reduces the cost of configuration, operation, power consumption for the ETERNUS AF/DX storage system.
It ensures business continuity with high speed backup, disaster recovery and other data protection features.

"FUJITSU Storage ETERNUS SF" eases the integration of ETERNUS AF/DX in the OpenStack environment with the following features:
- Configuration, performance management and fault monitoring for the entire storage system (HBA, switches, storage devices)
- Volume remote copy feature

Product reference
    ETERNUS SF - Flexible Data Management
        https://www.fujitsu.com/global/products/computing/storage/software/eternus-sf/

7. Summary

Although the best practice for OpenStack and the ETERNUS AF/DX is a simple configuration where only three storage pools are created, it can be used in various situations. Because users can seamlessly operate virtual machines and Cinder volumes from the OpenStack dashboard, it only takes a short time to configure a virtual environment, and the load on the server and users is also reduced.

OpenStack and the ETERNUS AF/DX – a winning combination!