

**OceanStor DJ
V1R2C10
Technical White Paper**

Issue 01
Date 2018-03-30

Copyright © Huawei Technologies Co., Ltd. 2018. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People's Republic of China

Website: <http://www.huawei.com>

Email: support@huawei.com

Contents

1 Overview of OceanStor DJ	5
1.1 Challenges to Traditional Data Centers	5
1.2 Benefits of OceanStor DJ	6
2 Application Scenarios	1
2.1 Setting up Independent Private Clouds.....	2
2.2 Interconnecting with Third-Party (VMware) Private Clouds.....	3
2.3 FusionCloud Private Cloud.....	3
3 Design Principles	5
4 Basic Functions and Principles	7
4.1 Storage Resource Virtualization.....	7
4.2 Automatic Service Deployment	8
4.3 Data Protection as a Service	9
4.4 Multi-Tenant Resource Management.....	10
4.5 Storage Service Monitoring	11
4.6 OceanStor DJ Extending Native OpenStack.....	12
4.6.1 OceanStor DJ Enhancing the Storage Service Capabilities	12
4.6.2 Comparison of Features Between Native OpenStack and OceanStor DJ	13
5 Networking and Environment Requirements	16
5.1 Deployment Technology	16
5.1.1 Microservice Architecture of Container-Based Deployment	16
5.1.2 Evolution to FusionCloud.....	17
5.2 Deployment-Related Concepts	18
5.3 Deployment Principles.....	18
5.4 Networking Overview	19
5.4.1 Networking of Single-Node Deployment	19
5.4.2 Networking of Three-Node Cluster Deployment.....	21
5.5 Typical Global Networking.....	23
6 Technical Features	24
6.1 Block Storage Service.....	24
6.1.1 Introduction.....	24
6.1.2 Technical Principles	25

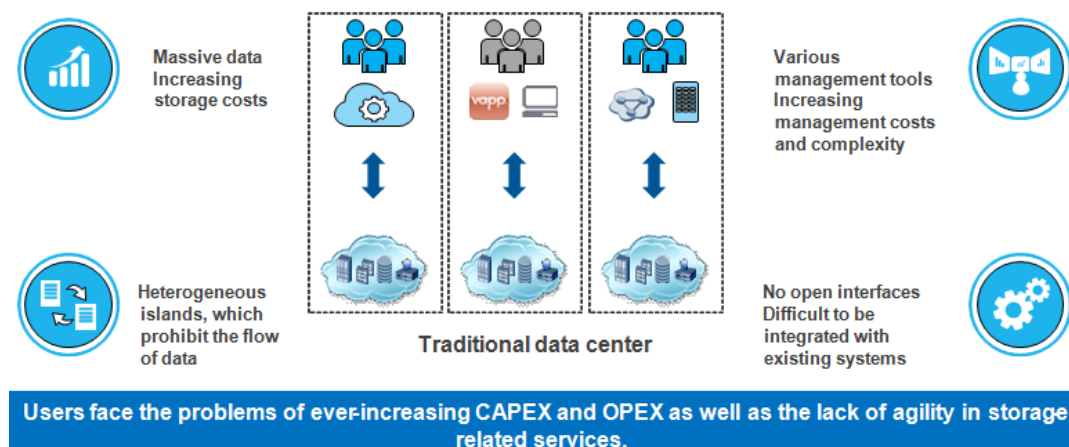
6.1.3 Service Process	29
6.2 File Share Service	30
6.2.1 Introduction.....	30
6.2.2 Technical Principles	31
6.2.3 Service Process	33
6.3 Object Storage Service.....	33
6.3.1 Introduction.....	33
6.3.2 Technical Principles	34
6.3.3 Service Process	35
6.4 Volume Backup Service	36
6.4.1 Introduction.....	36
6.4.2 Technical Principles	37
6.4.3 Service Process	37
6.5 Volume Replication Service.....	38
6.5.1 Introduction.....	38
6.5.2 Technical Principles	41
6.5.3 Service Process	43
6.6 Volume HyperMetro Service	44
6.6.1 Introduction.....	44
6.6.2 Technical Principles	45
6.6.3 Service Process	46
6.7 File Replication Service	47
6.7.1 Introduction.....	47
6.7.2 Technical Principles	48
6.7.3 Service Process	49
6.8 File HyperMetro Service	50
6.8.1 Introduction.....	50
6.8.2 Technical Principles	51
6.8.3 Service Process	51
6.9 Application Storage Service.....	52
6.9.1 Introduction.....	52
6.9.2 Technical Principles	53
6.9.3 Service Process	54
6.10 Multi-Region Management.....	54
6.10.1 Introduction.....	54
6.10.2 Technical Principles	56
6.10.3 Service Process	57
6.11 Host Management	57
6.11.1 Introduction.....	57
6.11.2 Technical Principles	58
6.11.3 Service Process	59
6.12 VMware vRO and vRA Integration	59

6.12.1 Introduction.....	59
6.12.2 Technical Principles	61
6.12.3 Service Process	61
6.13 VMware VASA Integration.....	62
6.13.1 Introduction.....	62
6.13.2 Technical Principles	63
6.13.3 Service Process	64
6.14 Connection to Odin.....	64
6.14.1 Introduction.....	64
6.14.2 Technical Principles	65
6.14.3 Service Process	66
6.15 Storage Service O&M and Monitoring	66
6.15.1 Introduction.....	66
6.15.2 Technical Principles	71
6.16 Architecture Reliability.....	71
6.17 Network Reliability	72
6.18 Data Reliability.....	72
7 Serviceability	74
7.1 Installation and Deployment	74
7.2 Alarm Management	74
7.3 Maintenance Modes.....	74
7.4 Health Check	75
7.5 Information Collection.....	75
7.6 Run Logs and Operation Logs	75
7.7 NTP Time Synchronization.....	75
8 Acronyms and Abbreviations.....	76

1 Overview of OceanStor DJ

With the development of the mobile Internet, Internet of Things, and Internet of everything, the amount of data being processed has been increasing exponentially. Users face many challenges (shown in Figure 1-1) when handling data storage, optimization, use, and management.

Figure 1-1 Challenges to traditional data centers



1.1 Challenges to Traditional Data Centers

- Low resource utilization and ever-increasing O&M cost
 - Heterogeneous storage devices are purchased in several batches during data center construction, resulting in diversified storage resources.
 - As the quantities, versions, and types of storage devices increase, storage resource management becomes increasingly complex.
 - Storage resources on different devices gradually form heterogeneous islands among which data cannot flow. Storage capacity fragments increase. Planning and allocating storage space becomes increasingly difficult.

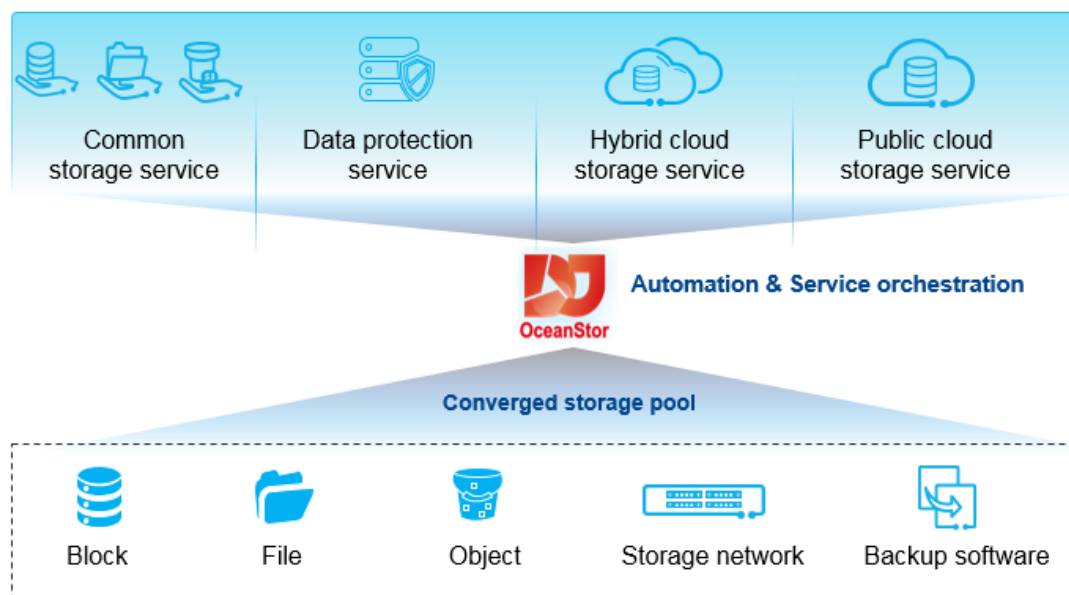
- Long time required for service rollout
 The rollout of medium- and large-sized data services involves service planning as well as purchase, configuration, and provisioning of storage resources. The rollout usually takes several months.
- Closed architecture of storage infrastructures, leading to the difficulty in integrating with other systems
 - Vendor lock-in, capacity expansion, reconstruction, and service provisioning across heterogeneous devices are difficult.
 - Data centers face problems related to resource consolidation and quick provisioning and cannot be transformed to cloud data centers.

Huawei OceanStor DJ software supports storage resource virtualization, automatic resource allocation, and storage resource as a service. Therefore, it can resolve the preceding problems faced by customers.

1.2 Benefits of OceanStor DJ

OceanStor DJ is a piece of service-driven storage control software. It masks underlying storage devices and describes services in the customers' language, enabling users who do not understand storage terminologies to easily apply for appropriate storage resources.

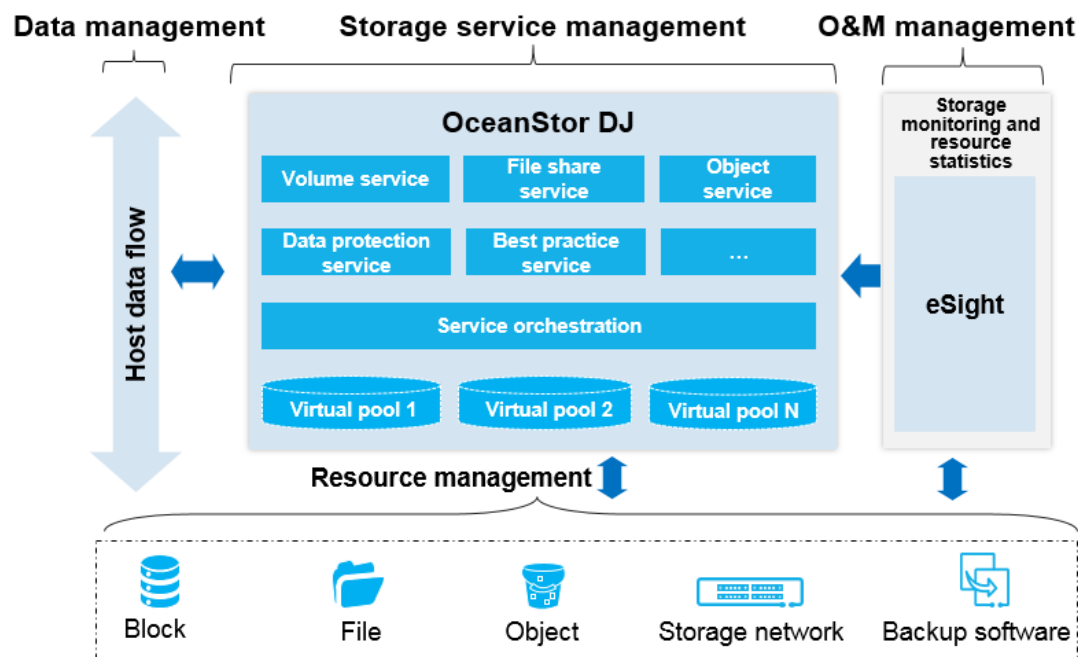
Figure 1-2 OceanStor DJ positioning



OceanStor DJ centrally manages storage resources, orchestrates service catalogs, and provides storage services and data application services on demand, which improves the operation efficiency of data centers. The industry has proposed the idea of control plane-based storage resource pool management in outband mode to manage various existing storage devices of enterprises, provide good expandability, and simplify the O&M of enterprises' IT departments. That is, management channels are used to achieve storage resource virtualization, automatic configuration, and data protection as a service to resolve customers' problems, without changing the existing data channels. With control plane-based storage resource pool

management, vendors separate storage policy management from hardware and focus on customer requirements for storage features. Customers can consolidate multiple types of storage resources on multiple storage devices based on their needs instead of adding storage functions on a storage array. OceanStor DJ can support and use various features of all storage hardware.

Figure 1-3 Service capabilities of OceanStor DJ



OceanStor DJ has the following features:

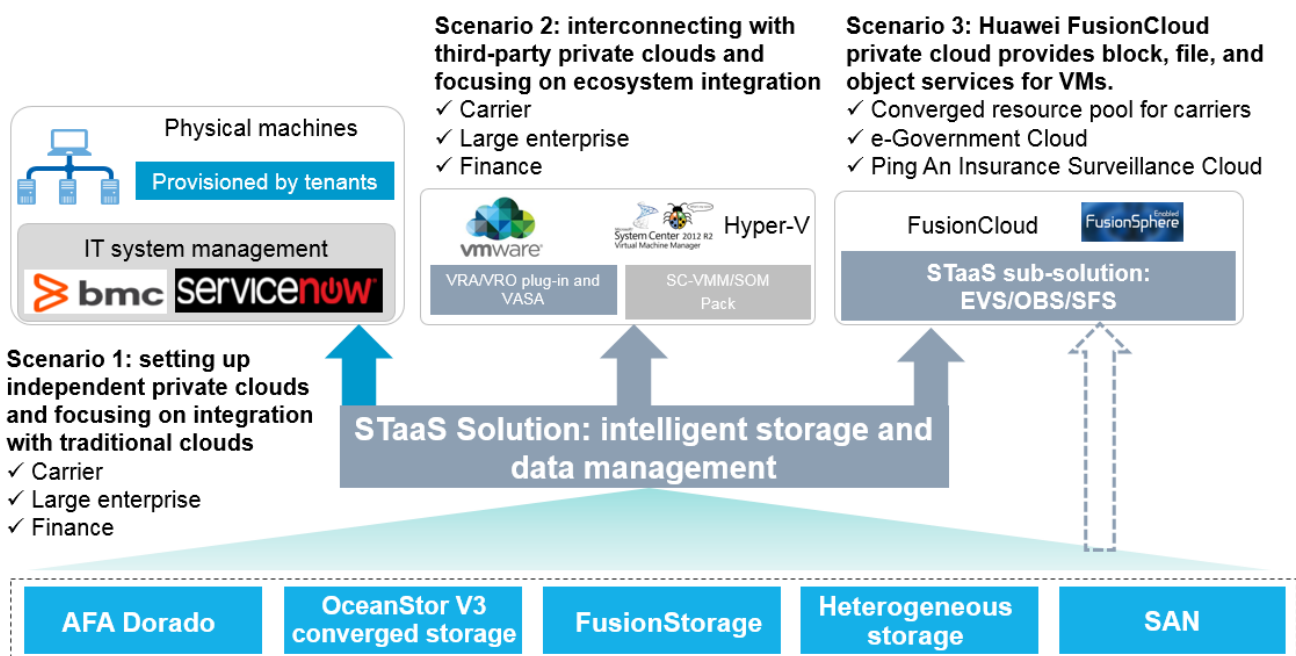
- Supports storage resource virtualization on the control plane and storage service level creation based on customer requirements.
- Allows administrators to configure management policies for storage and data services.
- Supports progressive construction of storage and data services.
- Supports global automatic management.
- Provides users with a self-service portal or various APIs.
- Supports the seamless expansion of the storage infrastructure architecture, improving reliability or performance (such as the QoS and storage service level configuration).
- Allows users to monitor and manage storage resources and costs in an open and transparent manner.

2 Application Scenarios

Software-defined storage (SDS) is among hotspots and development trends in the storage industry in recent years, and others include software-defined networking (SDN) and software-defined data center (SDDC). Similar to SDN, SDS allows flexible management in addition to precise system storage access. SDS is abstracted from hardware storage and can become a shared pool that is unrestricted by physical systems, thereby improving resource utilization. SDS can be deployed and provisioned based on software management, and simplified using policy-based automatic management.

OceanStor DJ is SDS at management and control planes, and is a part of Huawei storage as a service (STaaS) solution. OceanStor DJ applies to the following three scenarios.

Figure 2-1 OceanStor DJ ecosystem panorama

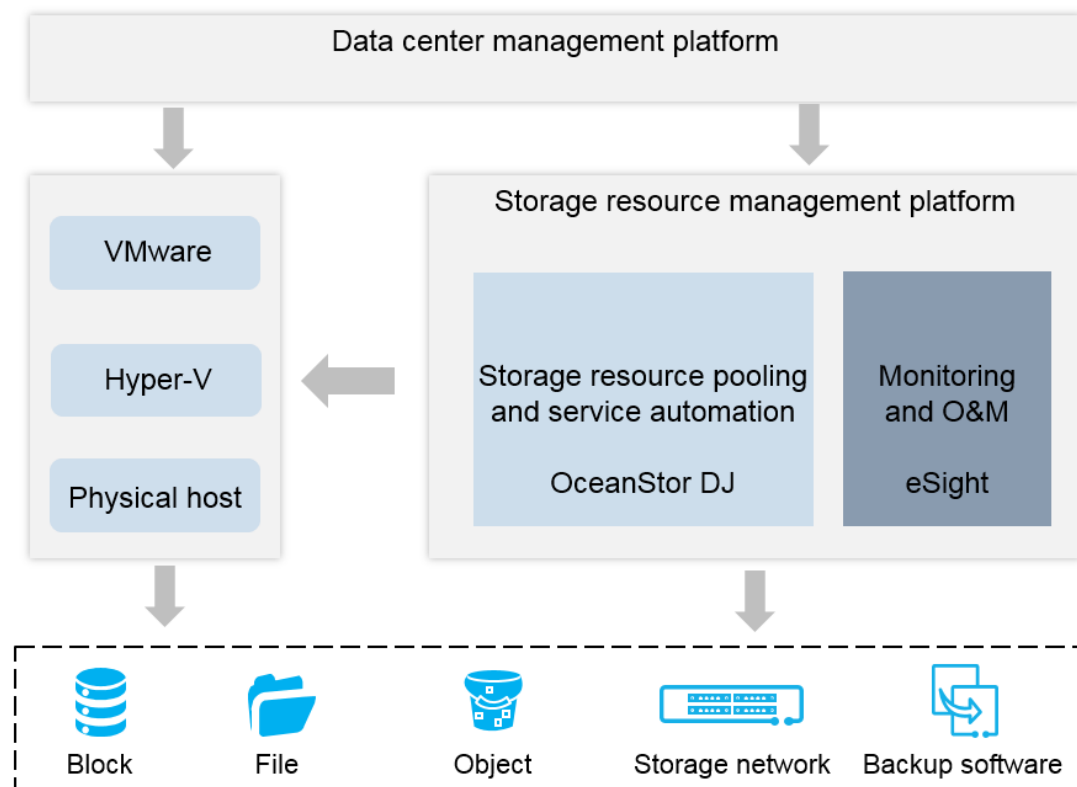


2.1 Setting up Independent Private Clouds

In this scenario, OceanStor DJ focuses on automatic management of traditional storage devices. OceanStor DJ provides traditional data centers with a unified interface for storage resource management and provides automatic resource scheduling capabilities as services. OceanStor DJ implements storage pooling for traditional data centers, improves the resource scheduling automation rate, enhances O&M efficiency, and reduces error probability. The main capabilities of OceanStor DJ are as follows:

- Provides storage pooling for multiple data centers and eliminates vendor lock-in so that heterogeneous storage devices from different vendors can be managed in a unified manner. Establishes unified virtual resource pools and corresponding service levels.
- Supports automatic storage for multiple data centers and minute-level service rollout.
- Provides REST APIs to open key capabilities and integrate with existing IT systems.
- Supports unified management of multiple data centers one-click service provisioning.

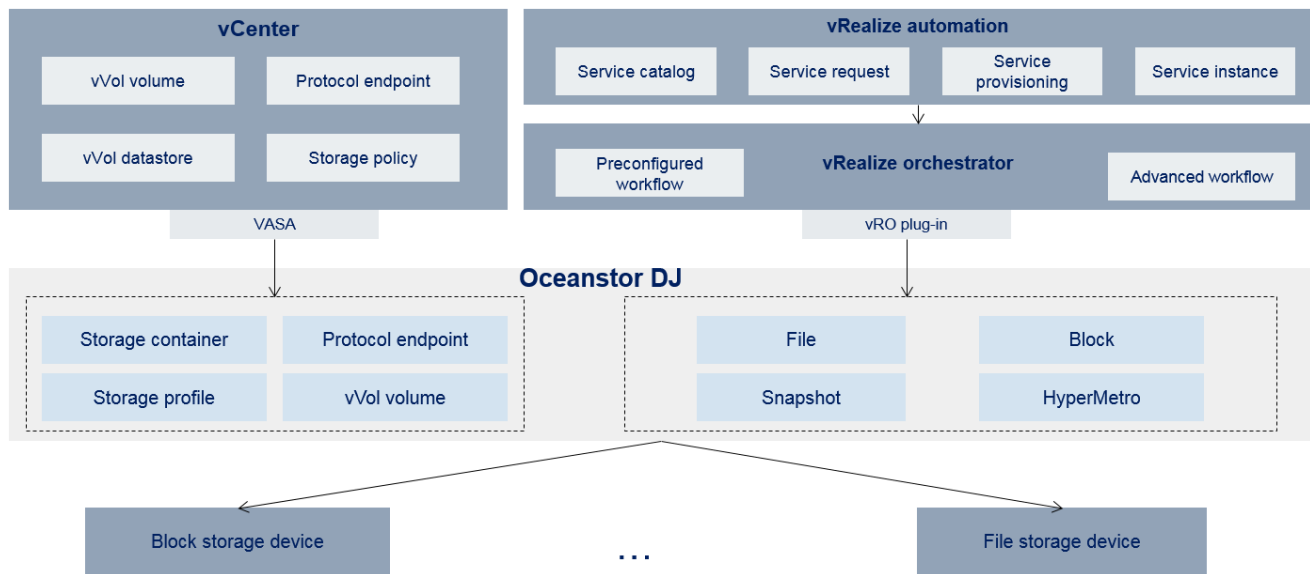
Figure 2-2 Setting up independent private clouds and focusing on unified and automatic storage management



2.2 Interconnecting with Third-Party (VMware) Private Clouds

When interconnecting with the VMware private cloud, OceanStor DJ focuses on the ecosystem integration with VMware. If a customer already uses VMware, OceanStor DJ can interwork with the existing private cloud.

Figure 2-3 Integration between OceanStor DJ and the VMware private cloud



VASA integration capabilities include:

- Supports VASA 2.0 and VVol 1.0, as well as ESXi 6.0 and 6.5.
- Supports the VVol and SPBM functions.
- Displays storage functions (QoS, Tiering, Thin, and Thick) associated with service levels (SLs) to vCenter.
- STaaS admin: manages VASA-related resources, including storage containers, storage configuration files, protocol endpoints, and VVols.

The capabilities of the vRO plug-in include:

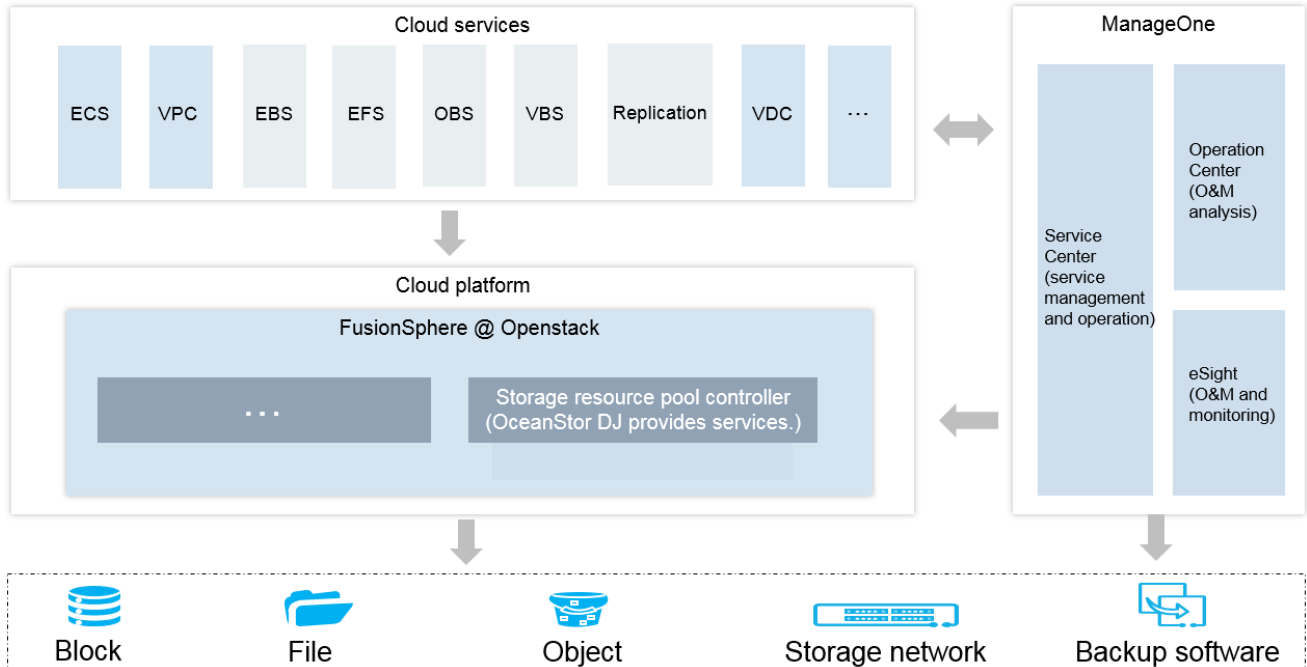
Provides over 30 storage services, such as file/block storage resource provisioning, HyperMetro, and replication.

2.3 FusionCloud Private Cloud

FusionCloud is a converged cloud computing solution developed by Huawei. It provides convergence of hardware resource pools from heterogeneous vendors, computing, storage, and network architecture convergence. In different enterprise IT scenarios, FusionCloud provides enterprises with an efficient, easy-to-build, and open cloud computing solution, and provides users with elastic and automatic infrastructure, on-demand services, and more agile IT services. By integrating with FusionCloud, OceanStor DJ provides Elastic Volume Service

(EVS), Scalable File Service (SFS), and Object Storage Service (OBS) that meet the requirements of FusionCloud VMs.

Figure 2-4 Resource pools and storage service components in the cloud solution



OceanStor DJ provides the following capabilities for the FusionCloud private cloud:

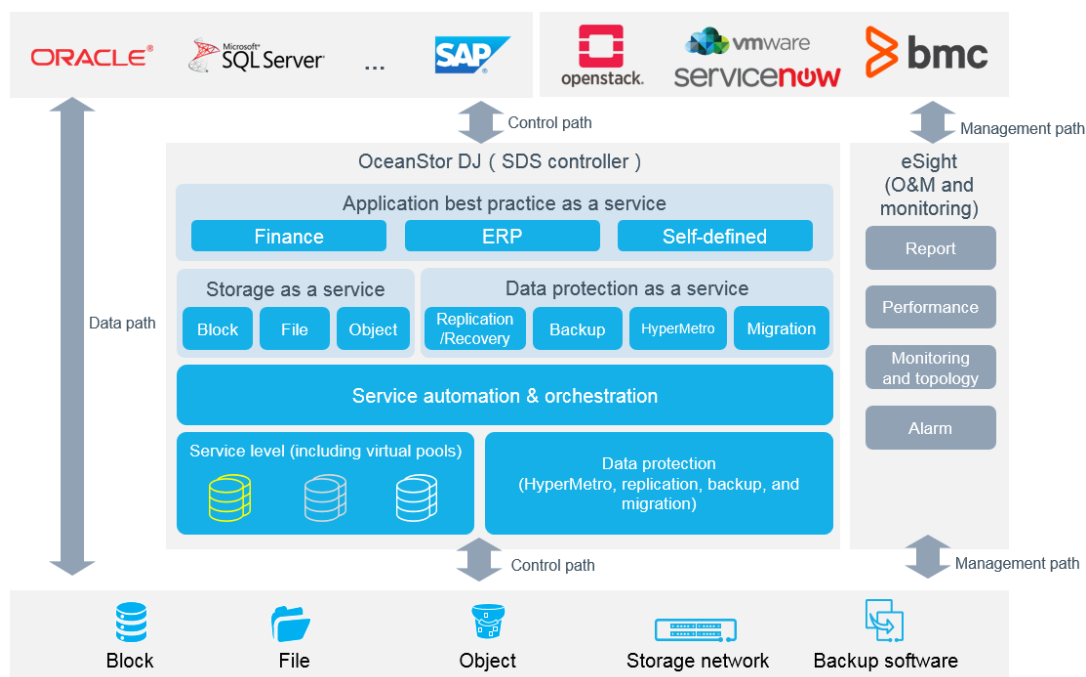
- Provides on-demand block, file, and object services.
- Provides O&M plane alarming, monitoring, and reporting.
- Meets customer data storage requirements for e-Government Cloud and policing cloud.

3 Design Principles

OceanStor DJ is service-driven storage control software that manages storage resources in a centralized manner and provides service-driven and automatic data services. Based on enhanced storage-related components of OpenStack, OceanStor DJ implements storage as a service and data protection as a service.

OceanStor DJ decouples applications from underlying storage, eliminating lock-in of traditional device and application vendors. In the cloud environment, OceanStor DJ provides storage, data protection, and other capabilities in XaaS mode, facilitating the transfer of the value chain to software and services.

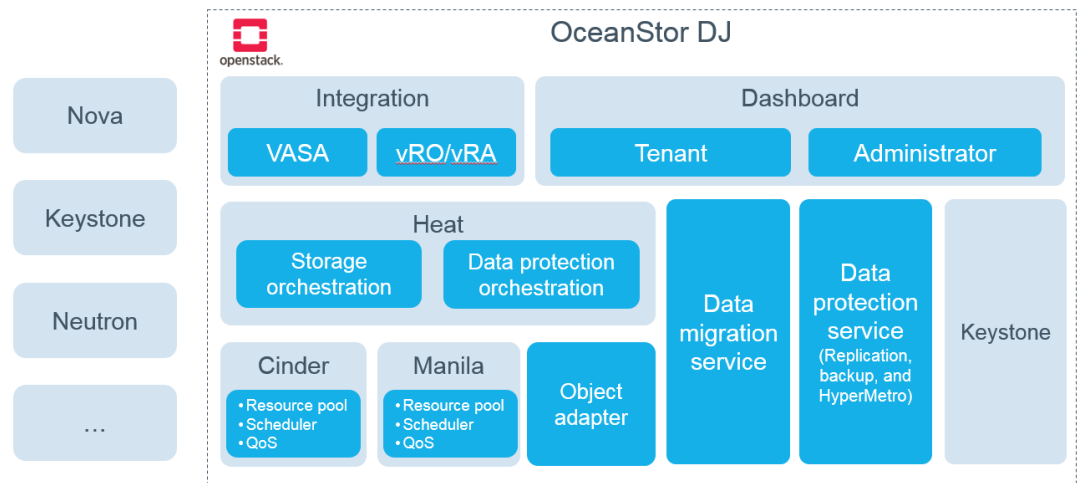
Figure 3-1 Functional architecture of OceanStor DJ



- Upper layer: storage application as a service
 - OceanStor DJ supports the allocation of storage resources, including file, block, and object storage resources, by providing storage functions as software and by decoupling access protocols.

- As an open platform, OceanStor DJ integrates Huawei and third-party software, including data protection software, and provides Data Protection As A Service (DPaaS) capabilities.
- OceanStor DJ is compatible with standard OpenStack APIs.
- Middle layer: automatic service deployment
 - Storage resources are provisioned on demand and automatically deployed through resource scheduling and task orchestration based on policies.
 - OceanStor DJ supports the self-service function, which improves resource allocation efficiency.
- Lower layer: storage resource virtualization
 - Enterprise-class storage resource virtualization based on the control plane helps to implement unified management and improve resource utilization.
 - OceanStor DJ supports standard hardware, which can be expanded elastically and quickly to carry new services and loads.

Figure 3-2 OceanStor DJ implementation architecture



Based on the open architecture of OpenStack, architecture compliance, management model, and API compatibility, OceanStor DJ enhances modules of OpenStack to meet the service objectives of storage resource scheduling and provision of storage resources as a service.

4 Basic Functions and Principles

OceanStor DJ provides the following basic functions:

- Storage resource virtualization
- Automatic service deployment
- Data protection as a service

In addition, OceanStor DJ enhances two functions: multi-tenant resource management and storage service monitoring.

This chapter describes these functions.

4.1 Storage Resource Virtualization

Storage resource virtualization of OceanStor DJ breaks the barriers among storage devices. It abstracts devices' storage resources that can provide the same capabilities to form a service level. Users do not need to know which storage arrays provide storage resources for their applications, but only need to pay attention to the service quality of service levels.

Storage resource virtualization is implemented by the OpenStack Cinder and Manila modules extended by OceanStor DJ.

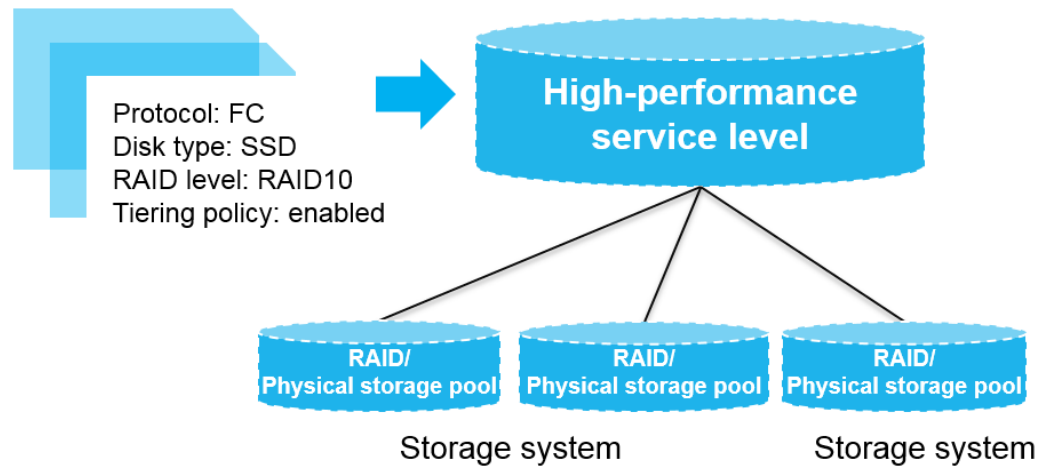
For example, a block storage service level is a logical group constructed based on VolumeType of OpenStack Cinder. The virtualization principles are as follows:

OceanStor DJ first obtains the basic information about devices, including the disk type, supported protocol type, thin/thick provisioning, and storage space size, from various vendors through the standard Cinder Driver interfaces provided by OpenStack. Then, OceanStor DJ groups storage pools that meet the conditions specified for a service level by an administrator and presents capabilities of the storage pools as the service level for follow-up resource provisioning.

For example, a user wants to create a service level with high performance. The user defines high performance as follows:

- The FC protocol is used.
- SSD disks are used.
- The RAID level is RAID10.
- A tiering policy is enabled.

Figure 4-1 Service level with high performance



The working principles are as follows:

1. OceanStor DJ discovers storage devices and obtains related information, including the storage pool list, supported protocol type, disk type, RAID level, and tiering status (enabled or disabled).
2. OceanStor DJ groups storage pools that meet the conditions specified for a service level by an administrator and presents capabilities of the storage pools as the service level for follow-up resource provisioning.

NOTE

The difference between storage service levels and storage pools is that a service level is a group of storage pools created based on certain conditions. Storage pools on devices from different vendors can be added into the same resource pool if they meet the same grouping conditions.

4.2 Automatic Service Deployment

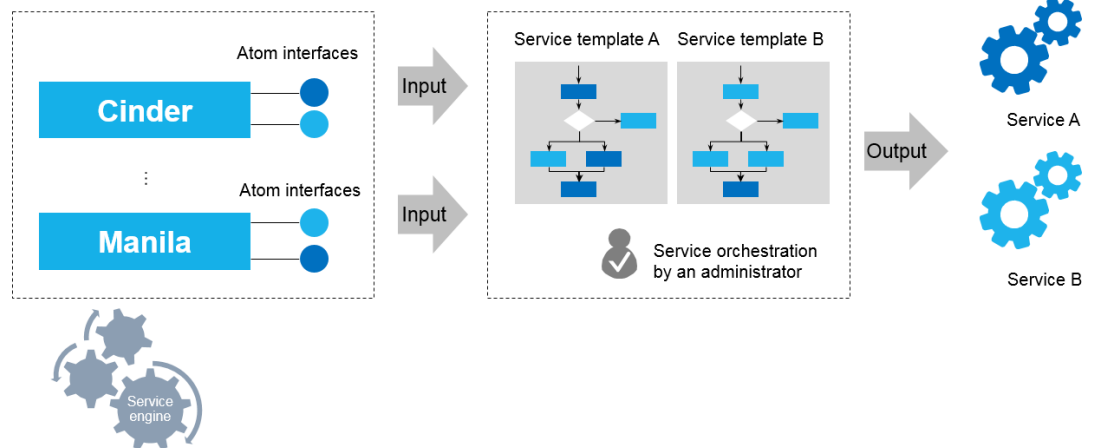
Automatic service deployment is implemented by OpenStack Driver and upper-layer service deployment and orchestration are implemented by OpenStack Heat extended by OceanStor DJ.

OpenStack Heat provides the basic framework of the IT resource orchestration engine. By extending the framework, vendors can orchestrate diversified IT resource services. OceanStor DJ implements storage service customization and orchestration using the extended OpenStack Heat module so it can provide various storage service capabilities. In addition, OceanStor DJ has a service catalog, which enables users to manage self-defined services or modify existing service catalogs to implement new services.

The purpose of automatic service deployment is to simplify the storage use by users. For example, if a user wants to apply for block storage resources, the user only needs to enter the desired storage capacity and volume count. Then, the service engine of OceanStor DJ invokes modules based on the preset services to complete related tasks.

Assume that a volume needs to be attached to a host. Cinder is invoked to complete volume assignment. The Fibre Channel switch module of Cinder is invoked to complete automatic zoning on the Fibre Channel switch.

Figure 4-2 Conceptual diagram of automatic service deployment



The service template (including service orchestration) combines OpenStack-related atom interfaces based on service requirements to provide services (catalog). Figure 4-2 illustrates the process.

The process is described as follows:

1. An administrator sets a service template package (including the service operation GUI and template file). The system automatically saves it.
2. When applying for storage resources, a user selects the desired service catalog, fills in volume information, and submits the template. The system then applies for storage resources.
3. The system selects storage devices that meet requirements in the resource pool to complete volume creation and attachment.

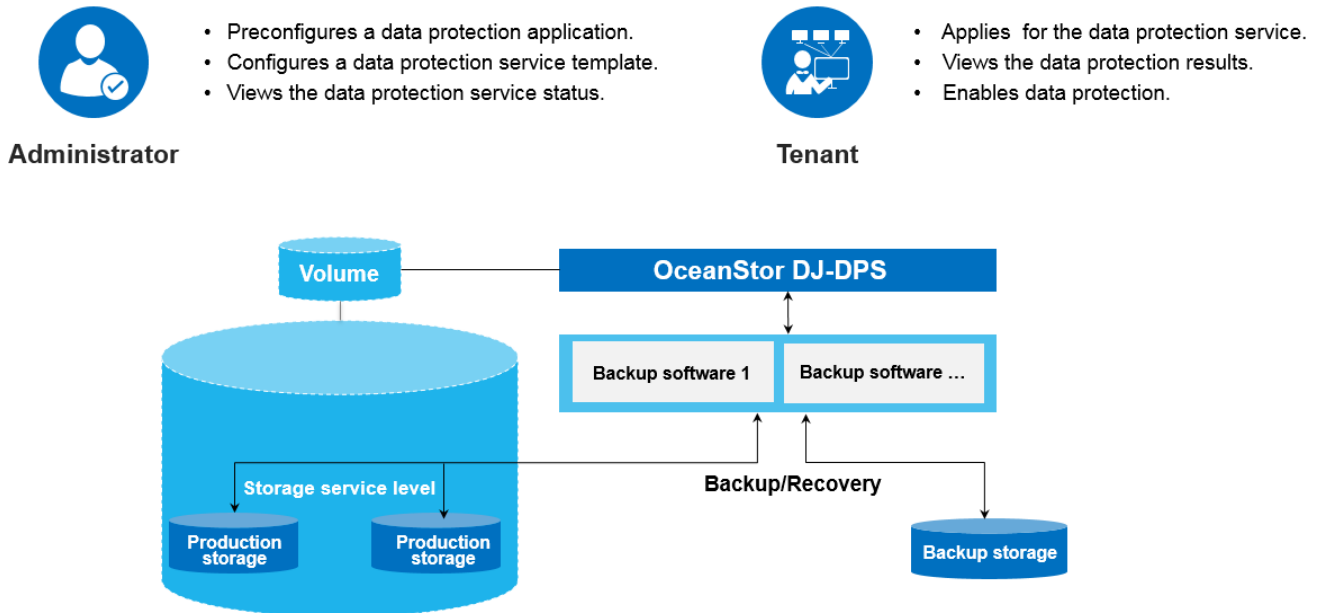
4.3 Data Protection as a Service

Data protection as a service is implemented by the Data Protection Service (DPS) component of OceanStor DJ. DPS provides the multi-tenant self-service volume backup, replication, and HyperMetro services. For volumes provisioned to the cloud platform, tenant users can use service catalogs to apply for the backup service and configure automatic protection policies. DPS automatically invokes the data protection software capability to back up data. Tenant users can view data copies and select data copies to restore data.

The advantages of data protection as a service are as follows:

- Easy use of data protection services by tenant users
- Template-based data protection service configuration by administrators
- Automatic execution of data protection services

Figure 4-3 Conceptual diagram of data protection as a service



4.4 Multi-Tenant Resource Management

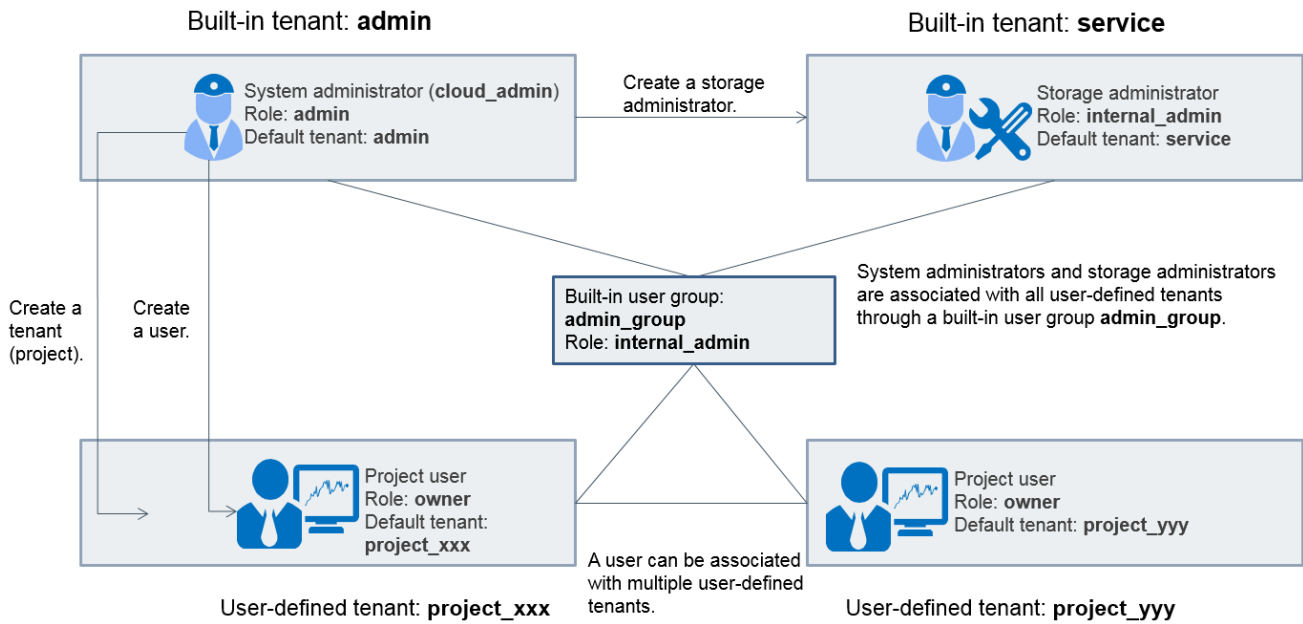
OceanStor DJ isolates storage resources of different tenants on the management plane. Tenant management involves project management, role management, and user management.

The system supports three roles: system administrator, storage administrator, and tenant user (same as project user).

- **System administrator:** role with permission for all system administration. The only built-in user of the system is **cloud_admin**.
- **Storage administrator:** role with permission for all system administration, except project, user, and role management. Storage administrators are responsible for the management of all storage resources, such as storage device management, resource pool planning, and storage service level definition.
- **Tenant user:** Tenant users use storage resources. They can apply for various storage resources through service catalogs.

Each user belongs to a default tenant. By default, system administrators and storage administrators belong to the built-in tenants **admin** and **service**, respectively. They are associated with all user-defined tenants through a built-in user group **admin_group**. After a user is associated with a tenant through a role, the user has the resource operation permissions of the tenant. The specific permissions are determined by the role.

Figure 4-4 Multi-tenant (project) model



Each tenant can apply for various storage resources, such as volumes, file systems, and buckets. A block storage service level or file storage service level can be visible to one or multiple tenants.

4.5 Storage Service Monitoring

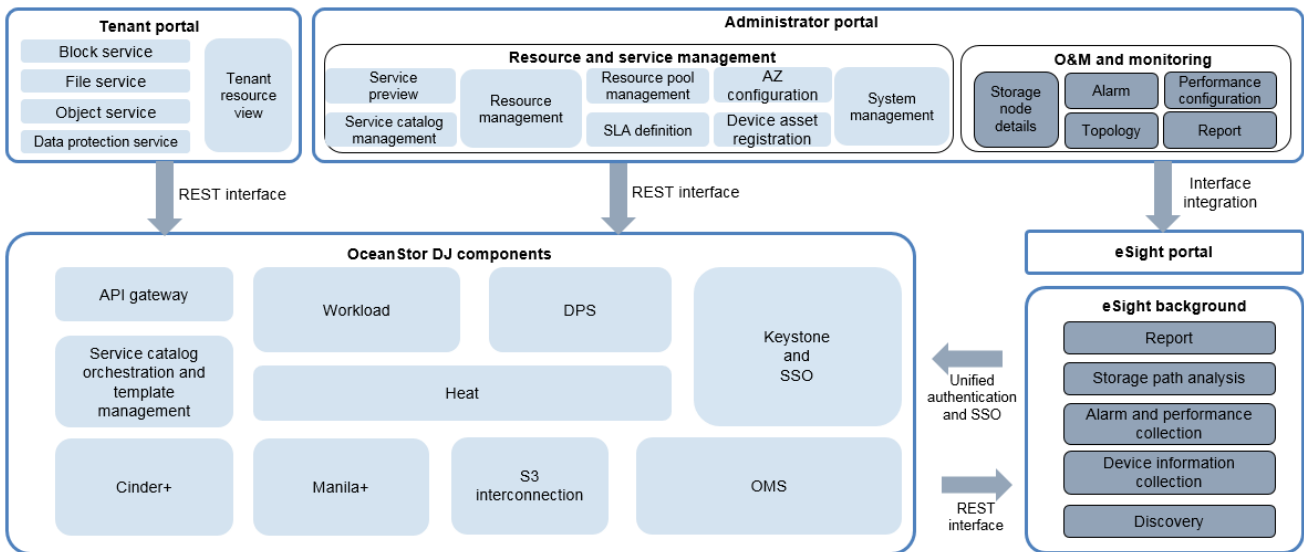
Users can use functions presented as services by OceanStor DJ to implement automatic modification of storage configurations. OceanStor DJ can interconnect with eSight, a piece of Huawei-developed monitoring software, to monitor storage devices, switches, and hosts managed by OceanStor DJ and output reports on monitoring data, so that customers' O&M requirements are met.

Adaption is made on both OceanStor DJ and eSight for their interconnection. The adaption conforms to the following rules:

- The OceanStor DJ GUI presents monitoring data and reports of OceanStor DJ-related storage objects, including service levels, storage arrays, and hosts.
- The eSight GUI presents monitoring data and reports of other storage objects.

In this manner, function de-coupling is ensured and customer requirements are met.

Figure 4-5 Architecture of OceanStor DJ and eSight



4.6 OceanStor DJ Extending Native OpenStack

4.6.1 OceanStor DJ Enhancing the Storage Service Capabilities

Table 4-1 Capabilities enhanced by OceanStor DJ

Enhanced Item	Native OpenStack	OceanStor DJ
Storage system capability collection	None	Storage Pool, Disk, RAID, Protocol, Thin, Tier, and Storage QoS are used for storage service level planning and the resource scheduler.
Storage back-end configuration	The administrator manually configures Cinder.conf/Manila.conf to start the back end.	Configures storage service levels on the administrator interface. Generates Cinder.conf/Manila.conf automatically to start the back end.
Storage resource pool planning	None	The administrator plans storage service levels in advance based on functions and capabilities to manage these storage pools. The planning is combined with the scheduling of resource provisioning.
Storage resource scheduler	Basic capacity match scheduler	Volume types and back-end storage pools are in one-to-many mapping relationship. Expansion schedulers: <ul style="list-style-type: none"> • Capacity match scheduler • Pool match scheduler

Enhanced Item	Native OpenStack	OceanStor DJ
		<ul style="list-style-type: none"> • Affinity scheduler • Capacity minimum fragments scheduler • Port balance scheduler
Automatic storage configuration	Performs volume provisioning using back-end scheduling.	<ul style="list-style-type: none"> • Performs volume provisioning using back-end scheduling. • Replication, HyperMetro, and backup configurations can be expanded. • Zoning of Fibre Channel switches is automated. • SmartQoS configuration of Huawei V3 series is automated.
Data protection	Volume backup from Cinders to Swifts	<ul style="list-style-type: none"> • Invokes the third-party backup software to back up volumes. • Uses the replication capability of V3 series to implement volume and file replication data protection. • Uses the HyperMetro capability of V3 series to implement volume and file HyperMetro data protection.
Storage service catalog	None	<ul style="list-style-type: none"> • Common volume services, Windows/Linux volume services, and VMware services • Volume services with replication capability and HyperMetro capability • CIFS and NFS share services

4.6.2 Comparison of Features Between Native OpenStack and OceanStor DJ

Table 4-2 Comparison of features between native OpenStack and OceanStor DJ

Category	Function/Feature	Native OpenStack	OceanStor DJ
Device addition and management	Fibre Channel switches (Cisco/Brocade)	Manual	Automatic
	FC SAN devices and other third-party storage arrays	Manual	Automatic
Network management	Fibre Channel switch zoning	Supported	Supported with enhancement (port

Category	Function/Feature	Native OpenStack	OceanStor DJ
			selection allowed)
Storage resource pooling	Device information discovery (such as VSAN, fabric, physical pool, and port information)	Not supported	Supported
	Physical array management (physical pool creation)	Not supported	Supported
	Storage service level management	Not supported	Supported
	QoS and volume type configuration	Supported	Supported
	Capacity scheduling, port balance, anti-fragmentation, and affinity scheduling and automated zoning algorithm	Not supported	Supported
Storage as a service	Automatic resource allocation for block, file, and object storage	Supported	Supported
	Storage resource allocation templates based on best practices in the industry to achieve optimal configuration by one click	Not supported	Supported
	Volume backup, replication, and HyperMetro as a service	Not supported	Supported
	DR and service level change	Not supported	Supported (in implementation)
	Snapshot management and rollback	Not supported	Supported
	Host and host group management	Not supported	Supported
O&M alarms	Management and backup of configuration information, and one-click importing of	Not supported	Supported

Category	Function/Feature	Native OpenStack	OceanStor DJ
	configuration information for restoration		
	Interworking with the cloud O&M platform, storage resource management, reports on resource information, and prediction report on resource usage	Not supported	Supported

5 Networking and Environment Requirements

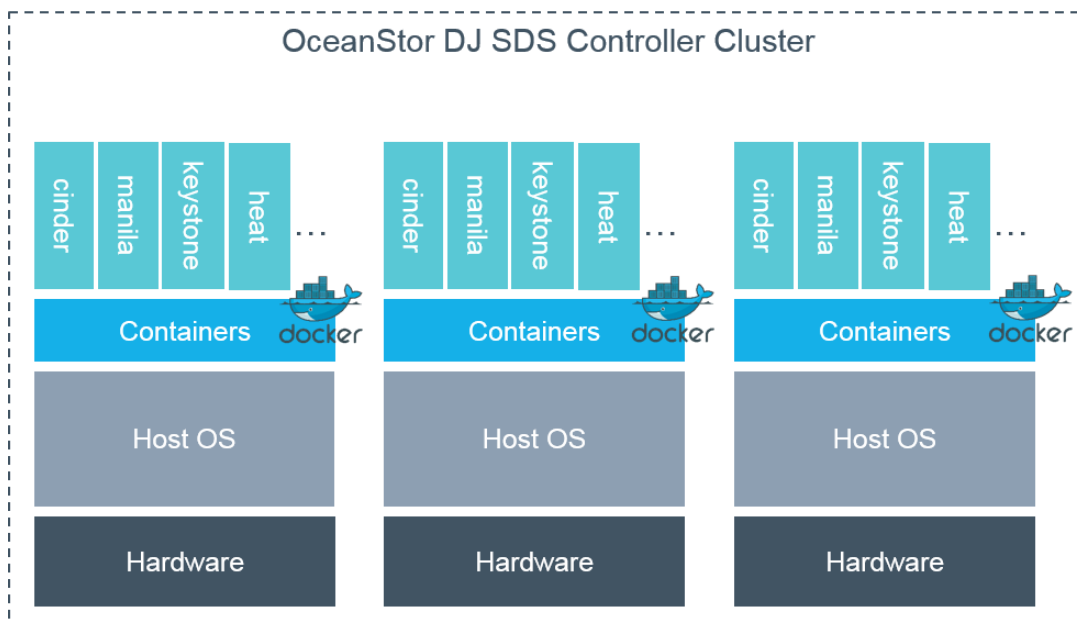
5.1 Deployment Technology

5.1.1 Microservice Architecture of Container-Based Deployment

Application deployment is a complex task. Apart from deployment, the following must be considered: dependency check, application expansion, and component update without affecting applications. The Docker container and the service-oriented design pattern can simplify and solve these issues. If an application consists of manageable components classified by function and these components can package all dependencies they need, it is easy to deploy the application in any architecture, expand the application, and update the components.

OceanStor DJ uses the latest microservice architecture of container-based deployment in the industry to deploy services and decouple components. Particularly, OceanStor DJ supports online upgrade without users knowing, greatly improving user experience and system reliability. OceanStor DJ supports container-based deployment of VMs and physical machines.

Figure 5-1 Docker deployment

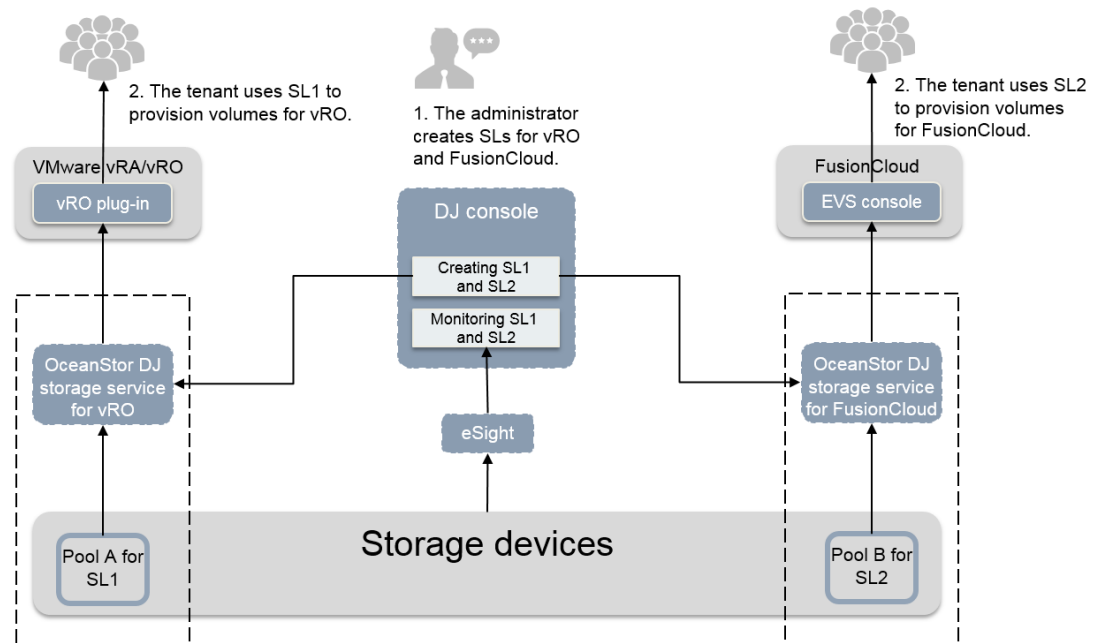


5.1.2 Evolution to FusionCloud

OceanStor DJ can be used to manage OceanStor DJ and FusionCloud resource pools at the same time so that storage administrators can manage two types of resource pools on only one GUI. OceanStor DJ supports the evolution of inventory storage management.

Original storage resources (for example, those provided for traditional IT systems) are provisioned using OceanStor DJ. However, FusionCloud does not manage traditional IT system resource pools. Storage administrators can manage existing storage resource pools and FusionCloud storage resource pools at the same time. On the OceanStor DJ GUI, administrators can create standard SLA storage resource pools for traditional IT systems and FusionCloud. Tenants still provision resources on OceanStor DJ and FusionCloud, respectively.

Figure 5-2 Interconnecting OceanStor DJ with the FusionCloud private cloud and unifying service levels



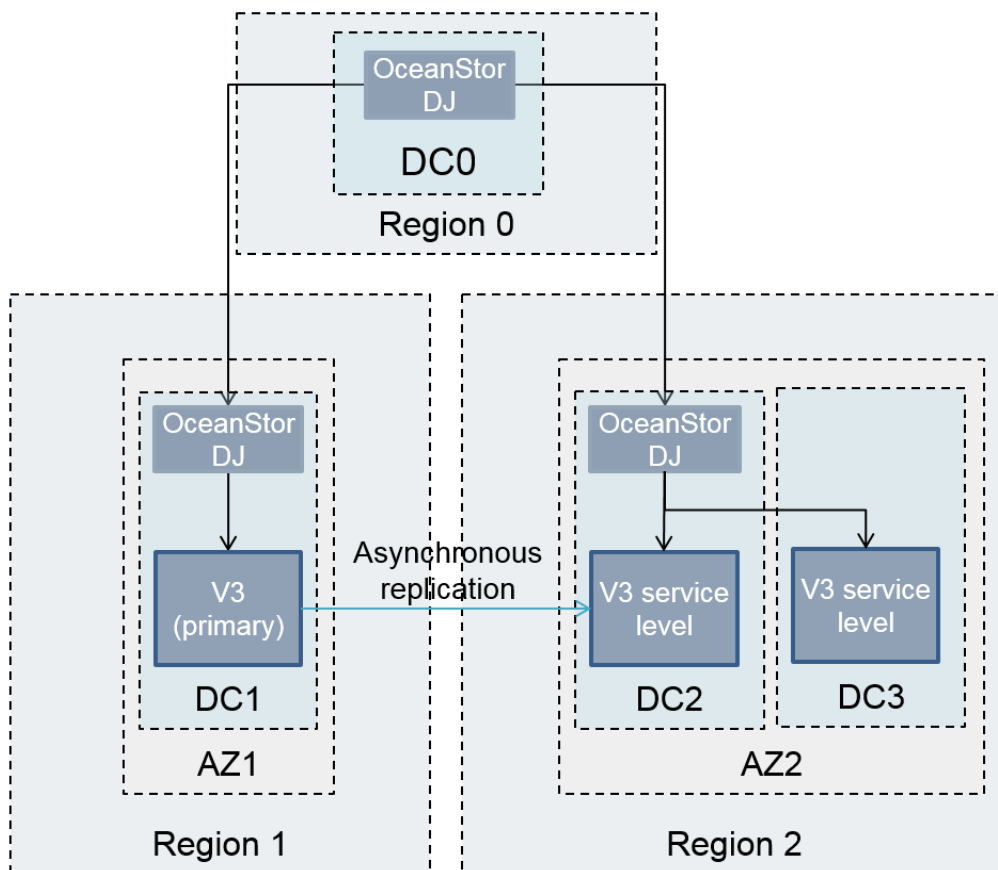
5.2 Deployment-Related Concepts

- **Region:** a physical zone, such as Central China and East China. A region is a geographical area with the access latency (for example, 100 ms) as the radius. A region may have one or multiple data centers (DCs). Different regions are usually in different geographical locations and have geographic disaster recovery functions. If the distance exceeds that determined by the latency, the service quality cannot be ensured. In this case, another region needs to be planned.
- **AZ:** availability zone. It is a concept from OpenStack. Different AZs have independent wind, fire, water, and electricity facilities, and are physically isolated. A Virtual Private Cloud (VPC) can be deployed across AZs. An original AZ contains only one OpenStack instance, while a current AZ contains multiple OpenStack instances.
- **DC:** data center. It is a concept from OpenStack. One AZ may carry multiple DCs.

5.3 Deployment Principles

One set of OceanStor DJ needs to be deployed for each region and can manage one or multiple data centers in this region. As shown in the Figure 5-3, OceanStor DJ manages DC1 in region 1, and DC2 and DC3 in region 2.

Figure 5-3 Multi-region deployment networking



5.4 Networking Overview

OceanStor DJ deployment scenarios include:

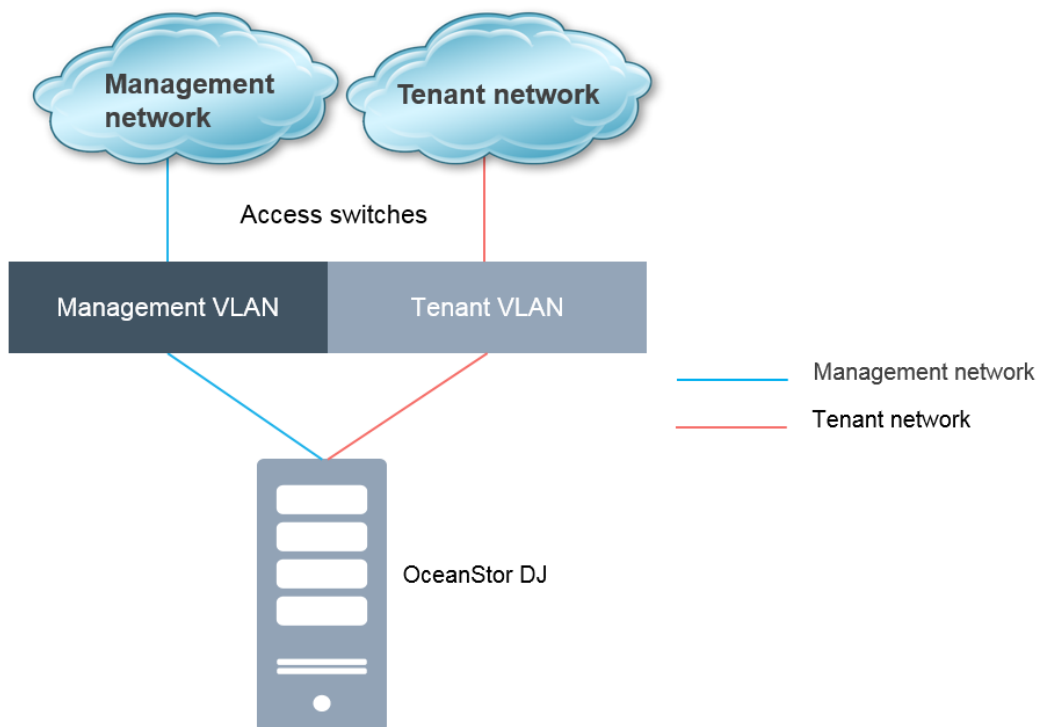
- Single-node deployment
- Three-node cluster deployment

Three-node cluster deployment is recommended because it provides higher reliability.

5.4.1 Networking of Single-Node Deployment

Figure 5-4 shows the networking of single-node deployment.

Figure 5-4 Single-node deployment



Network description:

- Two network ports corresponding to the IP address on each network plane of the OceanStor DJ node are bonded in active/standby mode. This can prevent single points of failure, improving reliability.
- Two VLANs are configured on the access switches. Network planes are isolated by the VLANs.

Table 5-1 describes the network planes and IP address planning in OceanStor DJ single-node deployment.

Table 5-1 Network planes and IP address planning in single-node deployment

Network Plane	IP Address Name	Quantity	Description
Management network	Management IP address	1	Used by the OceanStor DJ node to communicate with storage devices, hosts, Fibre Channel switches, management UI, and third-party APIs. It belongs to the same network segment as the management network of backend storage does. The administrator and tenant access ports share one IP address.

Network Plane	IP Address Name	Quantity	Description
Tenant network	Tenant IP address	1	Used to carry the tenant UI and service requests from tenants as well as implement the access of third-party APIs. It belongs to the same network segment as the customer's tenant service network does. The administrator and tenant access ports share one IP address.



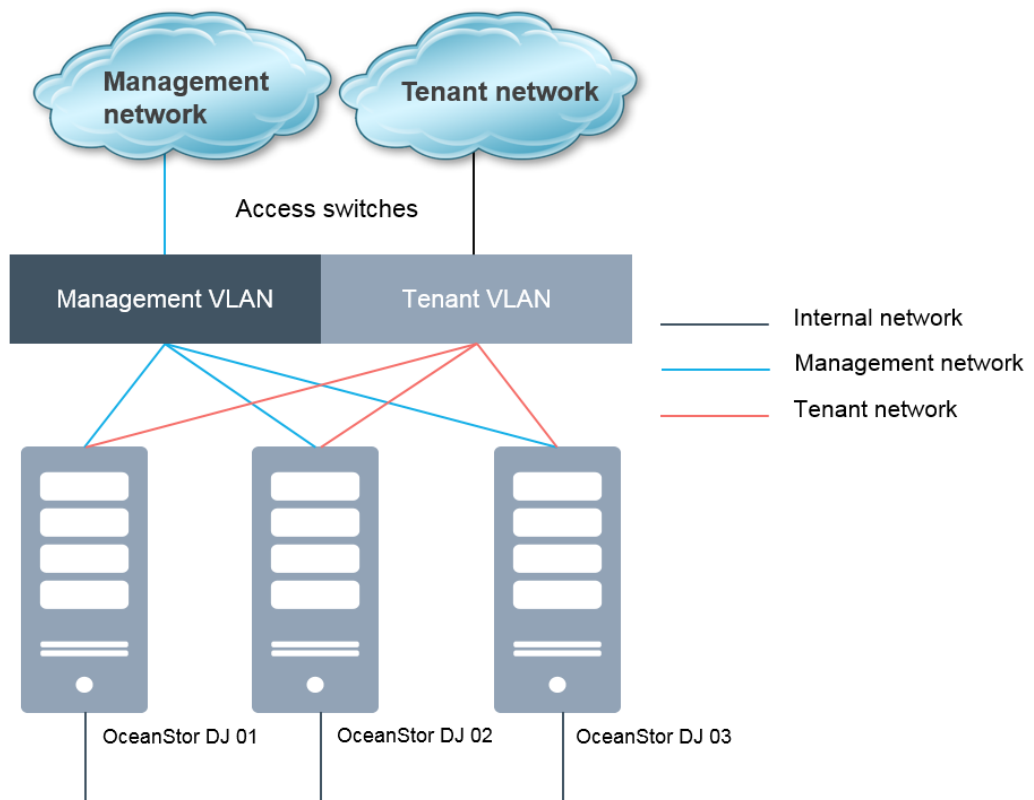
NOTE

The management network and tenant network can be deployed on the same network segment or different ones.

5.4.2 Networking of Three-Node Cluster Deployment

Figure 5-5 illustrates the typical networking of three-node cluster deployment.

Figure 5-5 Three-node cluster deployment



Network description:

- Two network ports corresponding to the IP address on each network plane of the OceanStor DJ nodes are bonded in active/standby mode.
- The customer provides two access switches, which are stacked.
- Three VLANs are configured on the access switches. Network planes are isolated by the VLANs.

Table 5-2 describes the network planes and IP address planning in OceanStor DJ three-node cluster deployment.

Table 5-2 Network planes and IP address planning in three-node deployment

Network Plane	IP Address Name	Quantity	Description
Management network	Management IP address	3	Used by the OceanStor DJ nodes to communicate with storage devices, hosts, Fibre Channel switches, management UI, and third-party APIs. It belongs to the same network segment as the management network of backend storage does.
	Floating IP address	1	
Tenant network	Tenant IP address	3	Used to carry the tenant UI and service requests from tenants as well as implement the access of third-party APIs. It belongs to the same network segment as the customer's tenant service network does.
	Floating IP address	1	
Internal network	Internal IP address	3	Used for high availability (HA) heartbeat communication and internal communication among nodes. It is planned based on the internal network's IP address segments provided by the customer.
	Floating IP address segment	≥ 5	

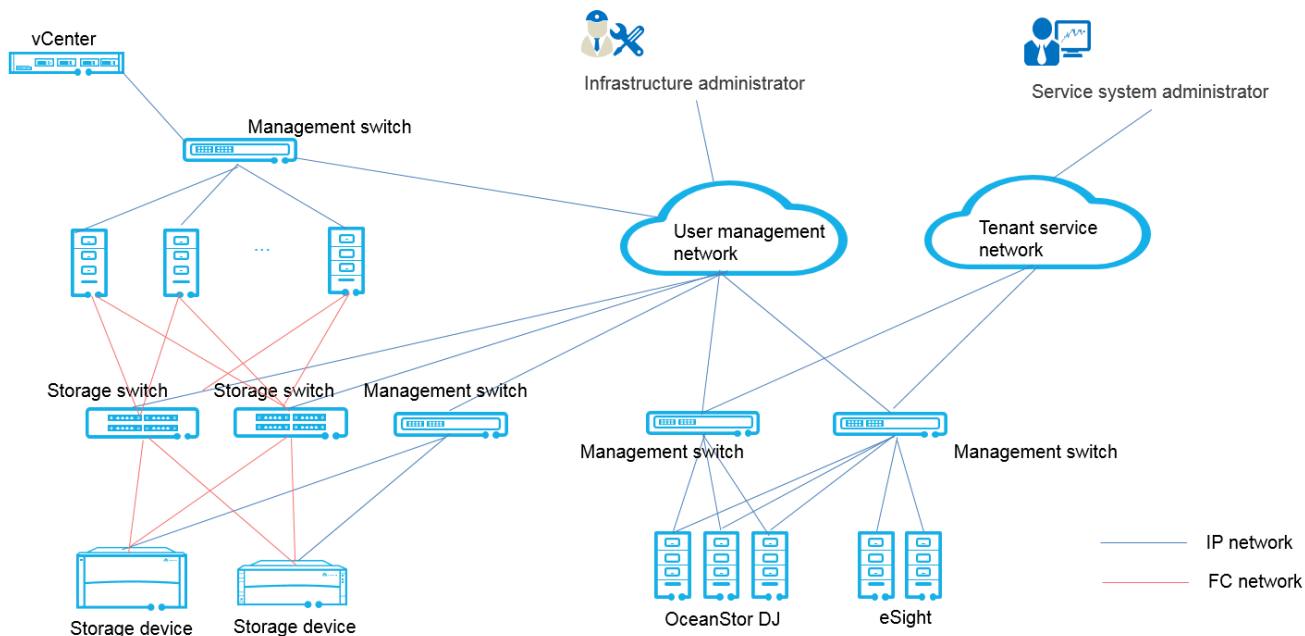


NOTE

The management network, tenant network, and internal network can be deployed on the same network segment or different ones.

5.5 Typical Global Networking

Figure 5-6 Typical global networking



OceanStor DJ supports switch cascading. That is, when multiple physical switches form a network, you can register the master switch to automatically create zones and broadcast zone information to the entire storage network.

6 Technical Features

6.1 Block Storage Service

6.1.1 Introduction

The block storage service provided by OceanStor DJ supports block resource pool management, block storage service level setting, volume allocation, and volume snapshot creation. The block storage service provides a simple service portal for users. It also provides RESTful APIs through which users can conveniently integrate the block storage service into their upper-layer O&M system. OceanStor DJ uses block storage service levels to define groups of volume capabilities or features, for example, resource pool to which a volume belongs, whether thin provisioning is enabled, whether storage tiering is enabled, migration policy used for tiering, QoS settings, and data protection capabilities. Users can also define other capability requirements.

The block storage service provides a scheduling layer. When users submit storage resource requirements through the GUI or APIs, the scheduling layer searches for the most suitable storage devices to provide storage space based on the requirements. The scheduling parameters (or capabilities) include the region, storage resource type, capacity, I/O, bandwidth, affinity, and data protection capabilities. The scheduling algorithm automatically matches the requirements with the capabilities of underlying storage devices to find the most suitable storage devices and the corresponding storage pools for resource allocation. Table 6-1 describes the scheduling algorithms supported by the block storage service.

Table 6-1 Block storage scheduling algorithms

Scheduling Algorithm	Description
Availability zone-based scheduling	Selects storage devices in a specified availability zone (AZ) and allocates volumes to them.
Affinity-based scheduling	Selects a storage device where a specified volume resides and allocates volumes to it.
Anti-affinity-based scheduling	Selects a storage device different from the one where a specified volume resides and allocates volumes to it.
QoS-based scheduling	Selects storage devices supporting QoS capabilities, allocates volumes to them, and configures a QoS policy for the volumes.

Scheduling Algorithm	Description
Huawei storage port scheduling	<p>When a zone is created on a switch, ports can be scheduled in either of the following modes for Huawei storage devices:</p> <ul style="list-style-type: none"> • Default mode: The available ports on the controller of each storage array are sorted by the number of hosts mapped to them and any two among the ports with the fewest hosts mapped are selected. One port is selected on a host, and a zone is created on a switch. Multipathing is supported. If there are two fabrics, zones are created based on the same rules. • Specified mode: When creating a service level, you can specify one or more array ports so that the volumes created using the service level can use the specified ports during mapping. <p>For non-Huawei storage devices, port selection algorithms for creating zones are implemented by vendors.</p>
Data protection capability-based scheduling	<p>Selects storage devices configured with the backup, replication, or HyperMetro capability and allocates volumes to them.</p> <p>Supports snapshot and snapshot rollback.</p>

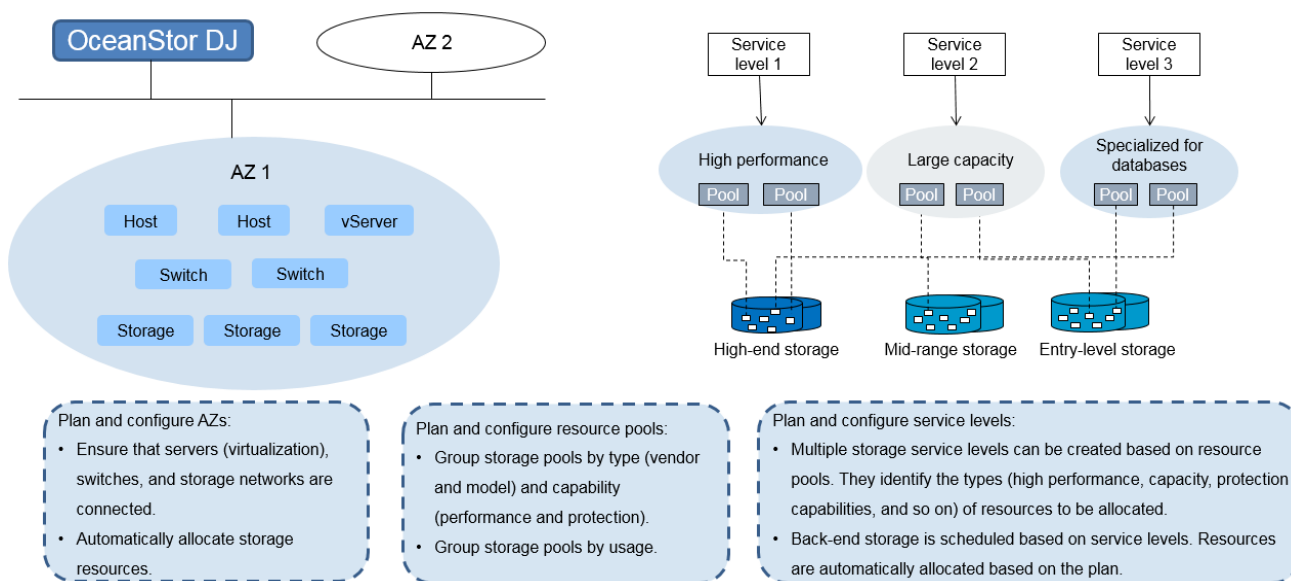
6.1.2 Technical Principles

- Service level-based service

In OceanStor DJ, each block storage service level can comprise multiple storage pools on storage devices from different vendors. Each block resource pool can be associated with multiple block storage service levels. Storage service levels determine the types of volumes that can be created. Storage administrators can create, delete, or modify block storage service levels.

When creating a volume, a tenant selects a block storage service level, which determines the capabilities or features of the volume. Based on the selected block storage service level, OceanStor DJ then selects a storage pool on a suitable storage device, creates a volume in the storage pool, and configures the volume to meet the block storage service level requirements.
- Block storage resource virtualization

Figure 6-1 Conceptual diagram of block storage resource virtualization



OceanStor DJ can virtualize block storage resources at the logical layer. Block storage resource virtualization is implemented by adding storage devices of different models from different vendors to OceanStor DJ and grouping storage pools on the storage devices based on certain rules (such as performance and protection capabilities) to form different storage service levels. When users apply for block storage services, OceanStor DJ allocates storage resources from the service levels. Users do not need to know which storage devices provide storage resources.

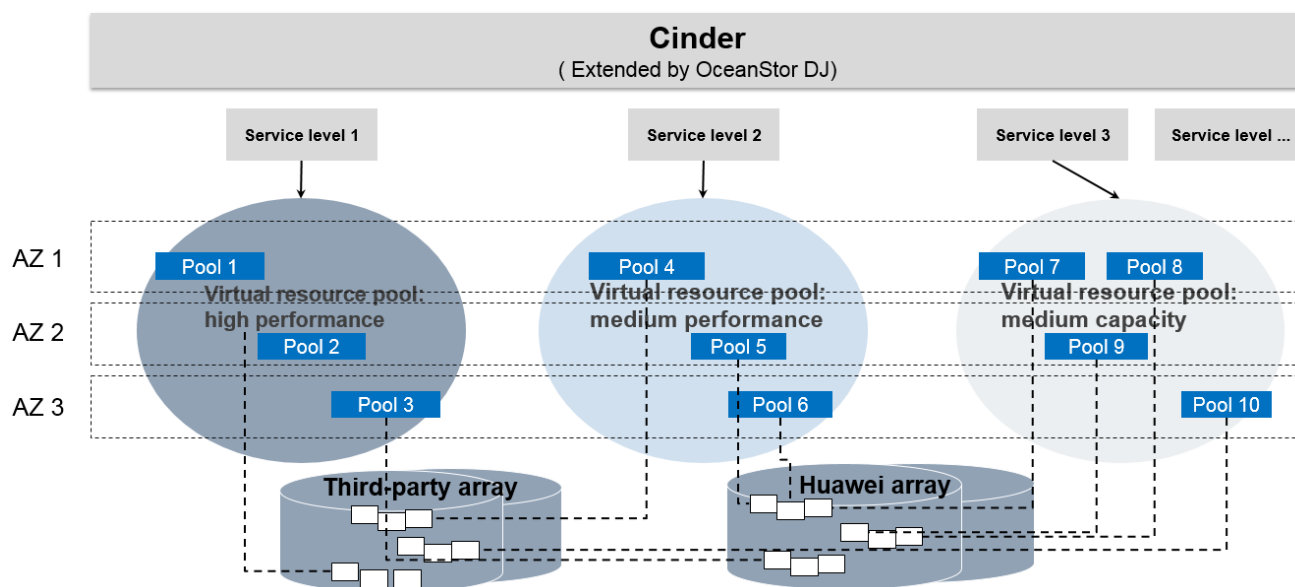
As shown in Figure 6-1, an AZ contains physical hosts, fabrics, storage devices, and other objects. A storage administrator can configure the AZ and define objects in it. When defining objects, ensure that all the storage devices and physical hosts within the AZ can communicate with one another through fabrics.

Therefore, an AZ defines a fully-connected domain. Any host can connect to any storage device in the domain. The issue that allocated storage space cannot be accessed due to a failed physical connection will never occur. In the actual configuration, storage administrators should plan the AZ scope based on the fabric scope due to physical connectivity requirements.

When a user applies for a volume creation service, OceanStor DJ schedules storage capacity within the specified AZ so that storage capacity is allocated only to physical hosts within the AZ.

- Elastic resource scheduling by IOPS, bandwidth, and latency

Figure 6-2 Conceptual diagram of resource scheduling

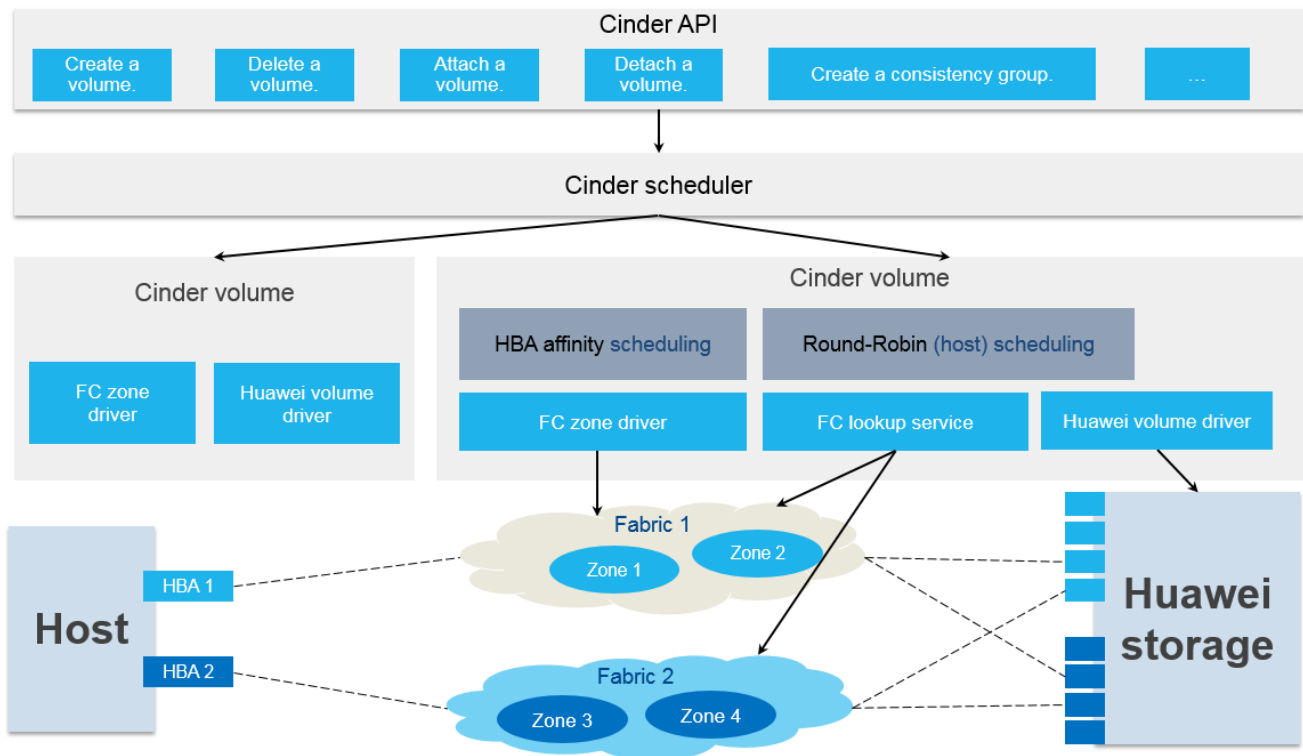


When applying for the volume creation service, a user can define the volume I/O processing capabilities (IOPS, bandwidth, and latency) in the QoS attributes. That is, the user can set the upper and lower limits of IOPS, latency, and bandwidth. These are QoS settings. QoS is configured when a block storage service level is defined. That is, when the volume is created, the selected block storage service level determines the requirements on I/O capabilities. Volumes created based on the same block storage service level have similar requirements on I/O capabilities.

After the requirements on I/O capabilities are set, OceanStor DJ will create volumes based on the capabilities. The I/O capabilities depend on QoS processing capabilities provided by storage devices. For example, the license supporting SmartQoS needs to be activated for Huawei OceanStor V3 storage devices. When a user applies for a volume with QoS capabilities, OceanStor DJ searches for a storage device that can ensure QoS, creates a volume on the storage device, and sets QoS parameters for the volume. When providing volume I/O access externally, the storage device controls I/O processing based on QoS requirements.

- High availability guarantee

Figure 6-3 Conceptual diagram of high availability guarantee



High availability must be ensured for volume access paths. Therefore, multiple access paths must be provided, and volumes must be allocated to storage devices with multipathing capabilities. OceanStor supports multipathing scheduling based on fabrics.

When multiple fabrics are configured in an AZ, OceanStor DJ automatically creates Fibre Channel zones on fabric managers based on ports of volume-residing storage devices and ports of hosts that will access the volumes. In this manner, access paths are established, and the whole process is called automatic zoning.

As shown in Figure 6-3, OceanStor DJ creates Fibre Channel zones on two fabrics based on multipathing requirements. Storage ports in the same zone are connected to different controllers, and host ports in the same zone are located on different host bus adapters (HBAs). This ensures the reliability of the full path from the host to the storage device. The zone creation rule is as follows: DJ + Host name + Array flag + Initiator. The created zones are small zones. That is, each zone contains only one array port and one host initiator.

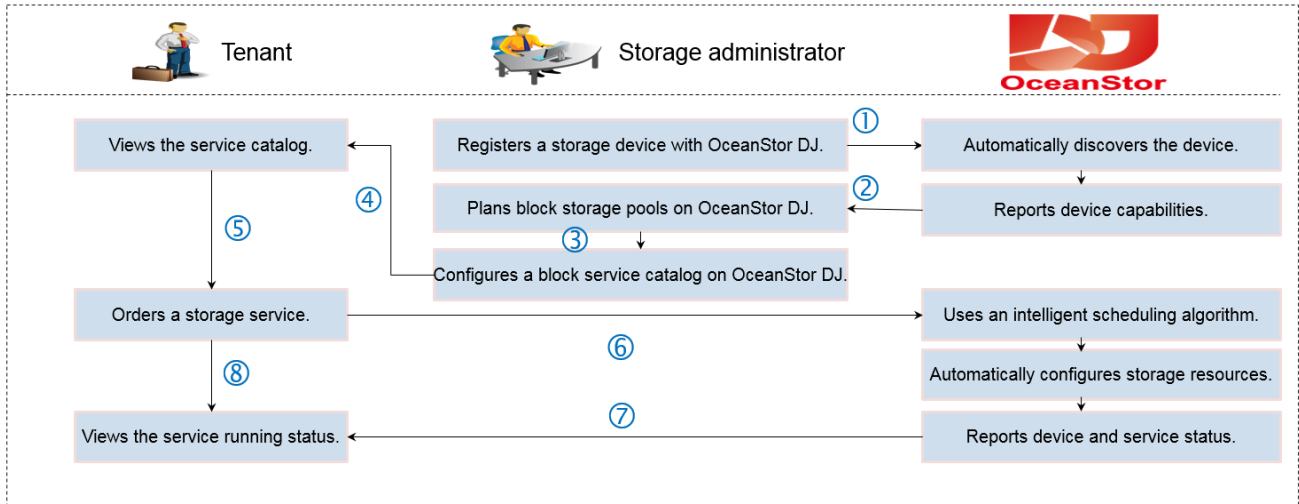
Multiple ports of a storage device are connected to fabrics. Therefore, the port used to create a zone is dynamically determined during automatic zoning. The processing capabilities (such as bandwidth) of each port are limited. To ensure balanced volume access, OceanStor DJ must consider port processing capabilities in port selection.

OceanStor DJ balances scheduling based on the number of hosts accessing storage ports. During volume allocation to hosts and automatic zoning, OceanStor DJ ensures that each Fibre Channel port of a storage device is accessed by the same number of host ports. In addition, the volume access requests from the same host pass through the same port on a storage device.

6.1.3 Service Process

Figure 6-4 shows the block storage service process.

Figure 6-4 Block storage service process



The block storage service process is described as follows:

1. A system administrator enters the address, access account, and password to register a block storage array.
2. After OceanStor DJ discovers the block storage array, it collects the status of the storage array.
3. The system administrator plans storage pools based on the type, usage, capacity, and other properties of the storage array, and configures the service catalog according to the design.
4. A tenant user views the service catalog in OceanStor DJ.
5. The tenant user selects and orders the desired storage service.
6. OceanStor DJ automatically configures storage resources based on the type of service selected by the tenant user.
7. OceanStor DJ provides the tenant user with the devices involved in the storage service and the service status.
8. The tenant user views the service running status.

When applying for the volume allocation service, a user does not need to specify a storage device or manually perform volume creation, mapping, zoning, attachment, or disk scan. Instead, the user only needs to specify the AZ, service level, and host or VM. Then, OceanStor DJ allocates storage resources, configures the Fibre Channel network, and attaches storage resources to the host at the background, ensuring that the volume is immediately available on the host.

6.2 File Share Service

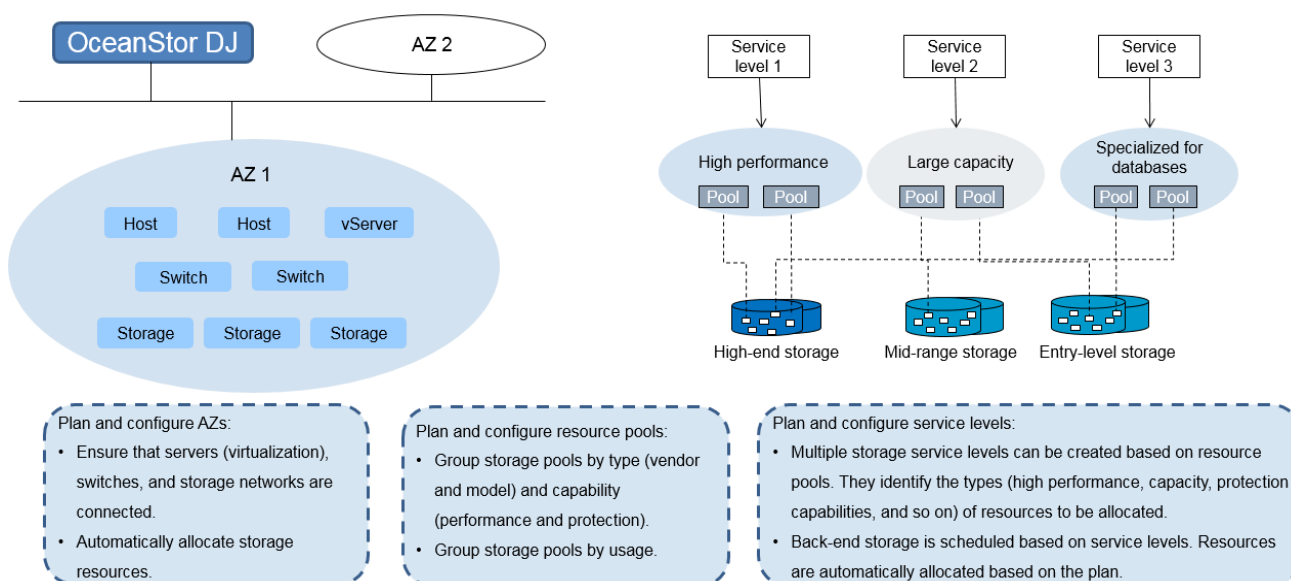
6.2.1 Introduction

OceanStor DJ provides users with a simple and easy-to-use file storage service portal, which supports the creation of file shares and file share snapshots. File storage services can be invoked through REST APIs at the background so that users can conveniently integrate the services into their upper-layer O&M system. Administrators can add storage devices, configure domains (LDAP, AD, and NIS domains) for share authentication, configure network share policies, and configure tenant quotas on configuration pages.

When applying for a file storage service, a tenant does not need to specify a storage device, but only needs to select a file storage service level created by an administrator. After receiving the service request, OceanStor DJ completes storage device selection, file share creation, network configuration, and access rule setting at the background. Then the tenant can use the file storage service immediately.

When a tenant user applies for a file service and the file service supports the quota feature, OceanStor DJ obtains the attributes for setting quotas for file shares so that the user can create subdirectories in file systems and set quotas for the subdirectories, enabling flexible management.

Figure 6-5 Conceptual diagram of file storage resource virtualization



6.2.2 Technical Principles

- Service level-based service

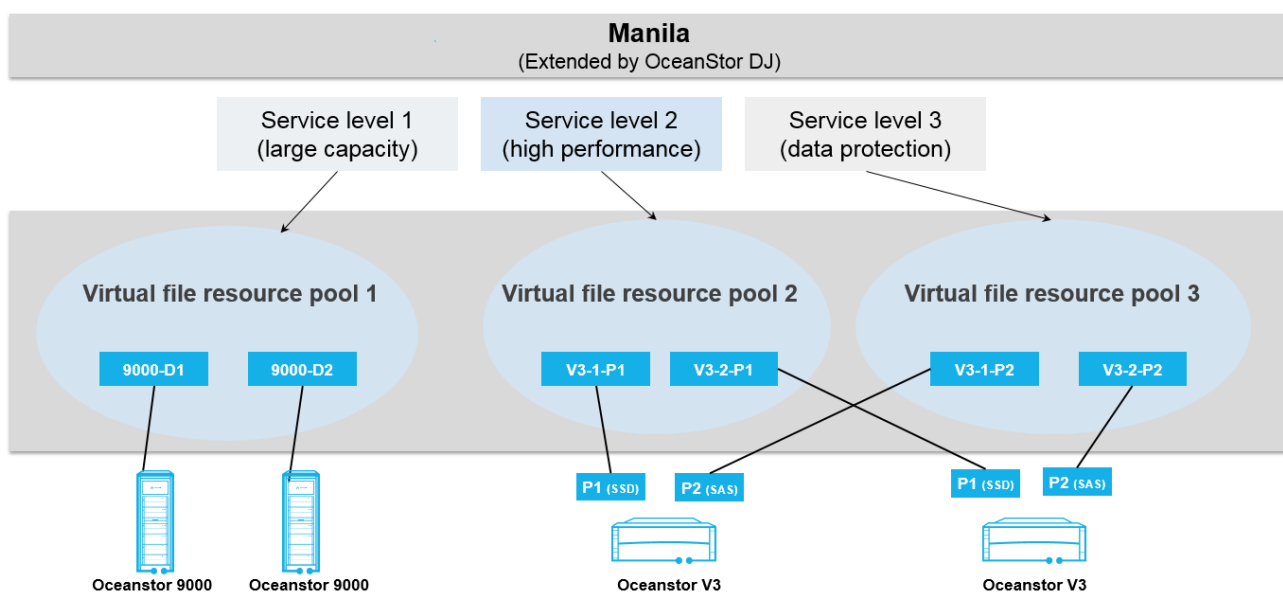
In OceanStor DJ, each file service level can comprise multiple storage pools on a storage array from a certain vendor. File storage service levels determine the characteristics of file shares that can be created. The file share characteristics include large capacity, high I/O performance, thin, deduplication, and so on. Storage administrators can create, delete, or modify file storage service levels.

When creating a file share, a tenant selects a file storage service level, which determines the capabilities or features of the file share. Based on the selected file storage service level, OceanStor DJ then selects a storage pool on a suitable storage device, creates a file share in the storage pool, and configures the file share to meet the service level requirements.

OceanStor DJ provides file share services that allow tenants to apply for Common Internet File System (CIFS) or Network File System (NFS) shares without having to know which storage devices provide file storage capacity. The system then automatically completes resource scheduling at the background. Tenants can define their own storage share networks, and obtain IP addresses from the defined storage share networks for the shares that they apply for. Tenants can configure access control by specifying users (for CIFS) or IP addresses (for NFS) that can access shares. Administrators can configure domains (including LDAP and AD domains) for share authentication.

- File resource virtualization

Figure 6-6 Conceptual diagram of file resource virtualization



Storage pools (P1 (SSD) and P2 (SAS) shown in Figure 6-6) for file services are accessed to OceanStor DJ through virtual file resource pools. A virtual file resource pool contains multiple physical storage pools. Each virtual file resource pool must correspond to the same type of storage devices. If there are multiple types of resource pools on a storage device, each resource pool can belong to the corresponding file resource pool. A file resource pool can span multiple storage devices. File storage service levels determine the characteristics of shares that can be created. Storage administrators can create, delete, or modify file storage service levels. A file storage service level defines a

group of file service capabilities or features, for example, resource pool to which the storage space used by the file storage service belongs and whether thin provisioning is enabled. Administrators can also define other metadata.

When creating a share, a tenant selects a file storage service level, which determines the capabilities or features of the share. Based on the selected file storage service level, OceanStor DJ then selects a storage pool on a suitable storage device, creates a file system in the storage pool, and creates a share in the file system to meet the service level requirements.

- Network management modes

In a cloud environment, cloud resources (such as computing, network, and storage resources) can be provided for multiple tenants. Each tenant can apply for cloud resources, and resources of different tenants are isolated.

OceanStor DJ provides file share services based on the extended OpenStack Manila. Tenants can apply for file shares and obtain IP addresses from their own networks for the file shares. Specifically, before using file share services, tenants usually apply for network resources (networks and subnets). They can create file share networks based on their own network resources. When applying for file shares, tenants apply for IP addresses in the file share networks and provide file shares through the IP addresses. OceanStor DJ supports the following network management modes for file share network generation:

- Neutron network (supported in the OpenStack environment): The cloud platform of tenants uses the Neutron service to manage network resources. When creating share servers for tenants, OceanStor DJ applies for IP addresses through the Neutron service. The IP addresses are in the private networks that the tenants apply for from the Neutron network.
- Nova network (supported in the OpenStack environment): The cloud platform of tenants uses the Nova service to manage network resources. When creating share servers for tenants, OceanStor DJ applies for IP addresses through the Nova service. The IP addresses are in the private networks that the tenants apply for from the Nova network.
- Offline network scheduling: Storage administrators specify an IP address pool for share services. When creating share servers for tenants, OceanStor DJ dynamically obtains IP addresses from the IP address pool. In this mode, the file service implementation does not require external network management, but only requires specific IP address segments reserved for file share services.
- Fixed IP addresses of storage devices: Storage administrators specify fixed IP addresses of storage devices for share service provisioning. Tenants share IP addresses and physical ports of storage devices.

- Security and access control

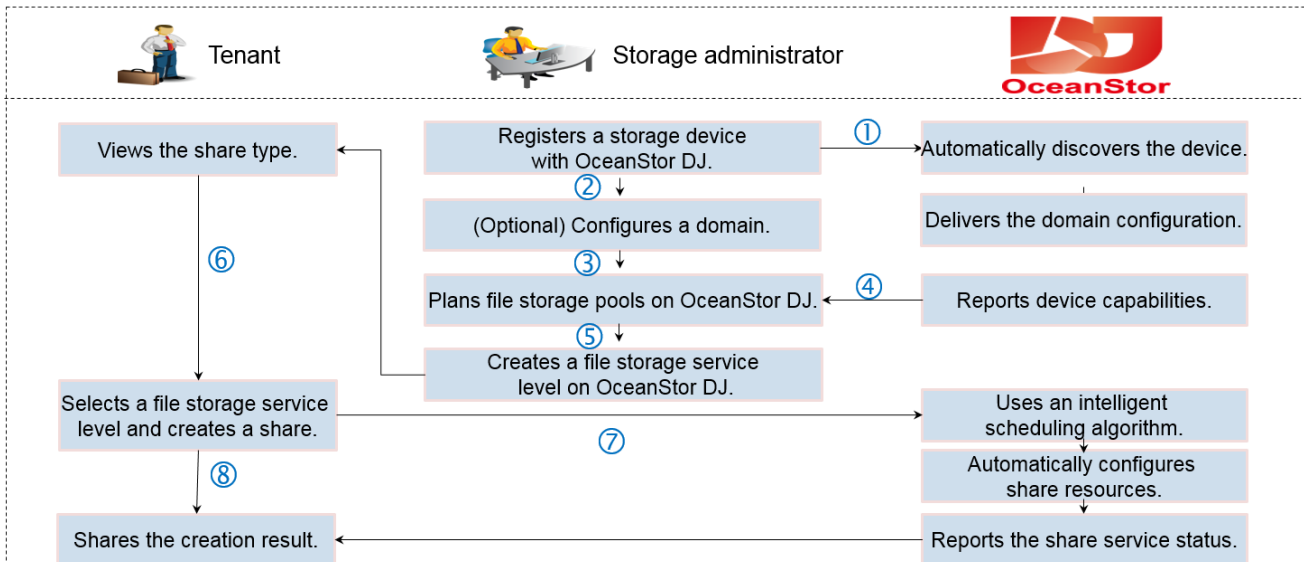
Tenants can configure external authentication services (such as the LDAP, NIS, and AD domains) for their shares. They can also configure external authentication services for each share network. After tenants configure external authentication services, OceanStor DJ automatically configures the services for storage devices. When tenants want to read/write files through file shares, they need to provide authentication accounts, which have been registered in the LDAP or AD domain. After receiving tenants' authentication information, storage devices verify the authentication information on the LDAP or AD domain server. With this feature, tenants of the cloud platform can use a unified authentication server to implement account authentication.

In addition, tenants can specify users (for CIFS shares) or IP addresses (for NFS shares) that can access each file share. Only the specified (domain) users or IP addresses can access the share.

6.2.3 Service Process

Figure 6-7 shows the file share service process.

Figure 6-7 File share service process



The file share service process is described as follows:

1. A system administrator enters the address, access account, and password to discover a file storage array. After OceanStor DJ discovers the file storage array, it collects the status of the storage array.
2. The system administrator sets the domain (NIS, LDAP, or AD) information for share authentication.
3. The system administrator plans storage pools based on the type, usage, capacity, and other properties of the storage array.
4. OceanStor DJ queries the capabilities of the storage array.
5. The system administrator creates file storage service levels.
6. A tenant user selects a file storage service level and creates a file share.
7. (Optional) A tenant can specify the file share name (for Huawei OceanStor V3 series and OceanStor 9000) when creating a file share.
8. OceanStor DJ filters storage arrays based on the file storage service level and completes file share creation.
9. OceanStor DJ returns the file share creation result to the tenant user.

6.3 Object Storage Service

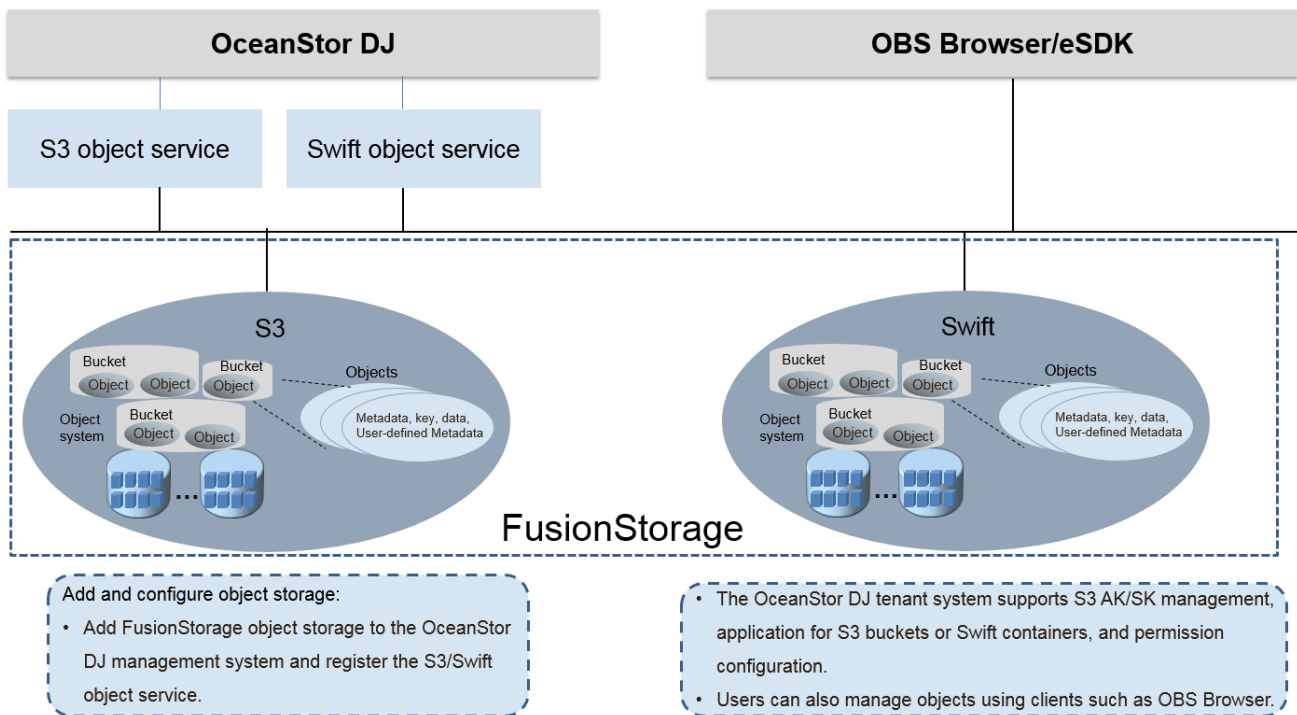
6.3.1 Introduction

FusionStorage is a core component in Huawei Server SAN solution. It can be deployed on multiple general-purpose x86 servers to consolidate the local disks on all the servers into

virtual storage resource pools to provide the block storage function. OceanStor DJ provides an easy-to-use object storage service portal. OceanStor DJ works with FusionStorage 6.0 to support S3 bucket and Swift container allocation services. FusionStorage 6.0 provides standard S3 or Swift APIs through which object storage services can be integrated into upper-layer application systems.

S3 or Swift is specified when FusionStorage 6.0 is deployed.

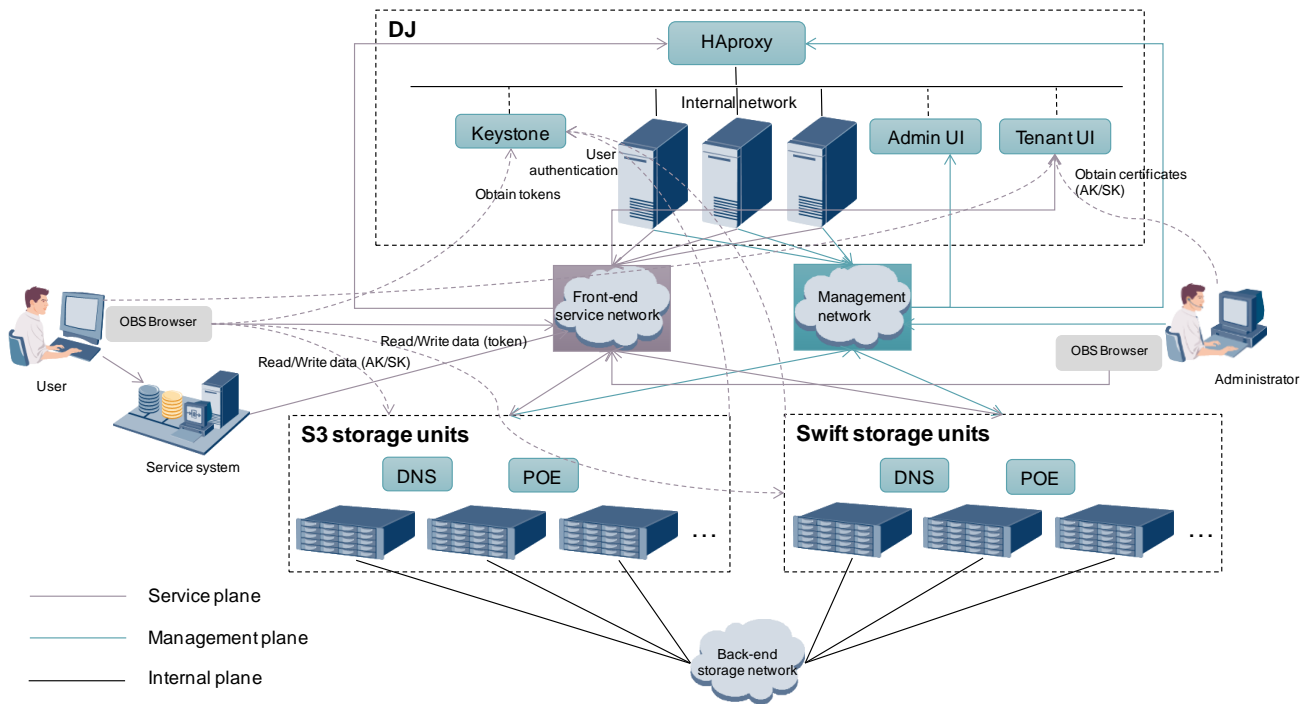
Figure 6-8 Conceptual diagram of the object storage service



6.3.2 Technical Principles

- OceanStor DJ uses OpenStack Keystone to implement standard authentication for object storage services.
- OceanStor DJ provides a self-service portal for tenants. This portal supports S3 AK/SK management, application for S3 buckets or Swift containers, and permission configuration.
- OceanStor DJ provides an independent management GUI, which supports unified tenant and user management as well as quota configuration for tenants.

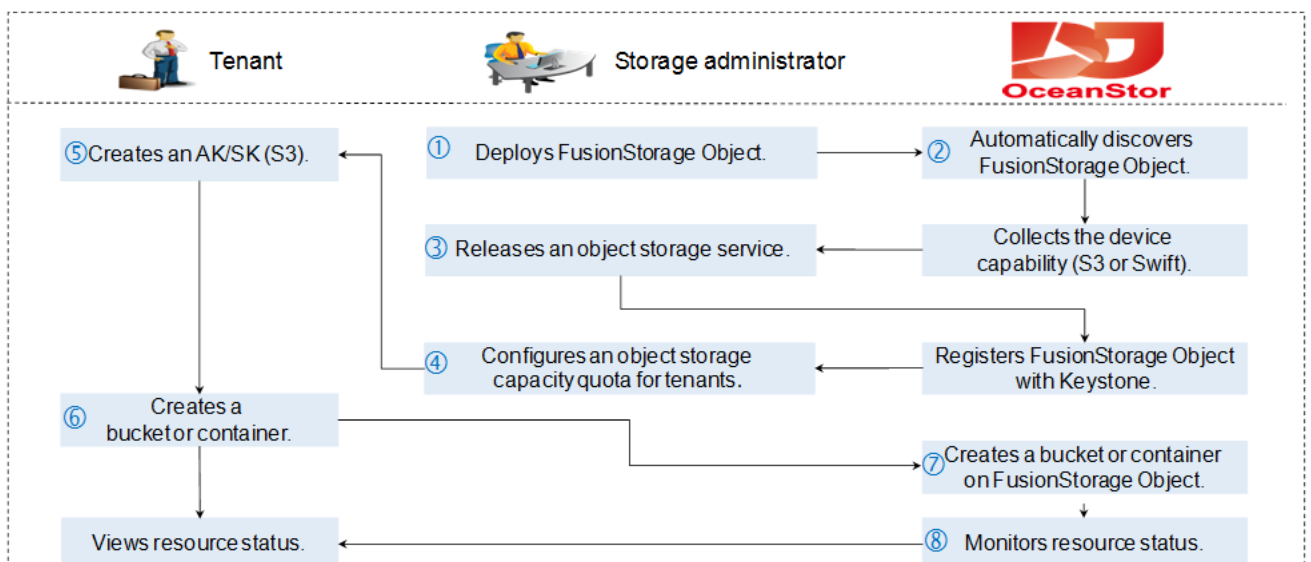
Figure 6-9 Networking for object storage services



6.3.3 Service Process

Figure 6-10 shows the object storage service process.

Figure 6-10 Object storage service process



The object storage service process is described as follows:

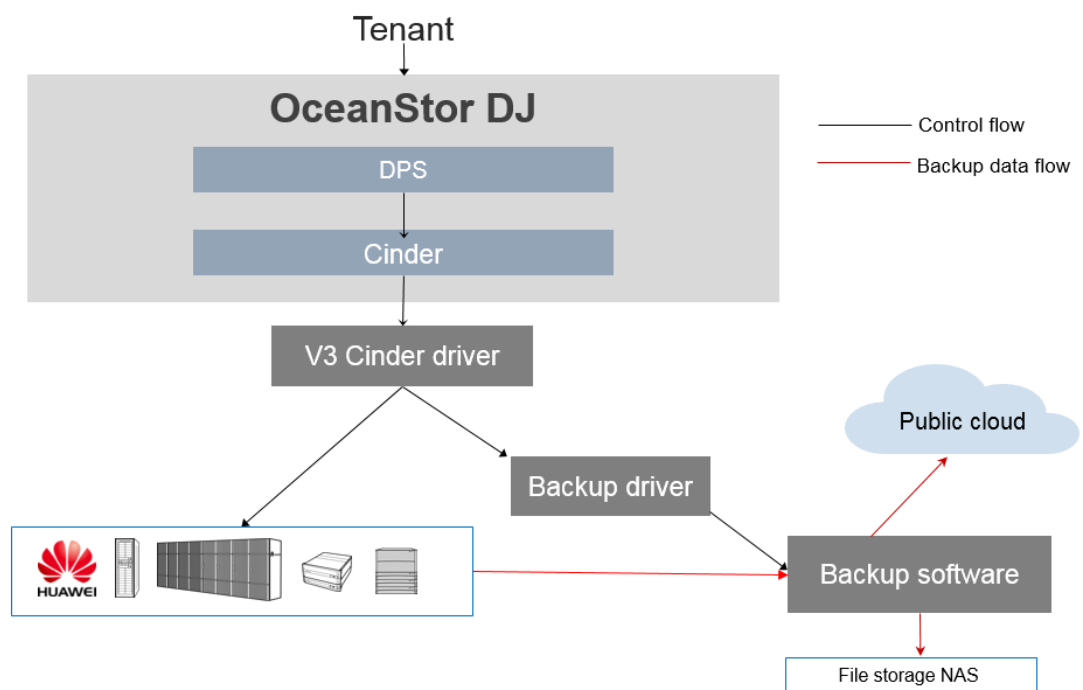
1. A storage administrator uses OceanStor Toolkit to deploy FusionStorage Object.
2. OceanStor DJ discovers FusionStorage Object and collects its properties and object service capability (S3 or Swift).
3. The storage administrator releases an object storage service on OceanStor DJ, registers FusionStorage Object with Keystone, and registers the object storage service with Keystone.
4. The storage administrator configures an object storage capacity quota for tenants on OceanStor DJ.
5. For an S3 service, a tenant user creates an AK/SK on the OceanStor DJ GUI.
6. The tenant user creates an S3 bucket or Swift container on the OceanStor DJ GUI.
7. OceanStor DJ schedules FusionStorage Object to create an S3 bucket or Swift container.
8. OceanStor DJ monitors storage resource status. After applying for resources, the tenant user can view the resource status and manage permissions on the bucket or container on the OceanStor DJ GUI.

6.4 Volume Backup Service

6.4.1 Introduction

The volume backup service abstracts volume backup capabilities based on block storage resource virtualization and creates a service catalog for self-service by tenants. With the volume backup service, tenants can back up important service data, ensuring data reliability. When data is lost, tenants can use backup data for restoration, minimizing the loss of key data. Storage data can be automatically backed up to the cloud using backup software. Major features include: consistency backup, backup plan execution, and consistency group copy addition and deletion.

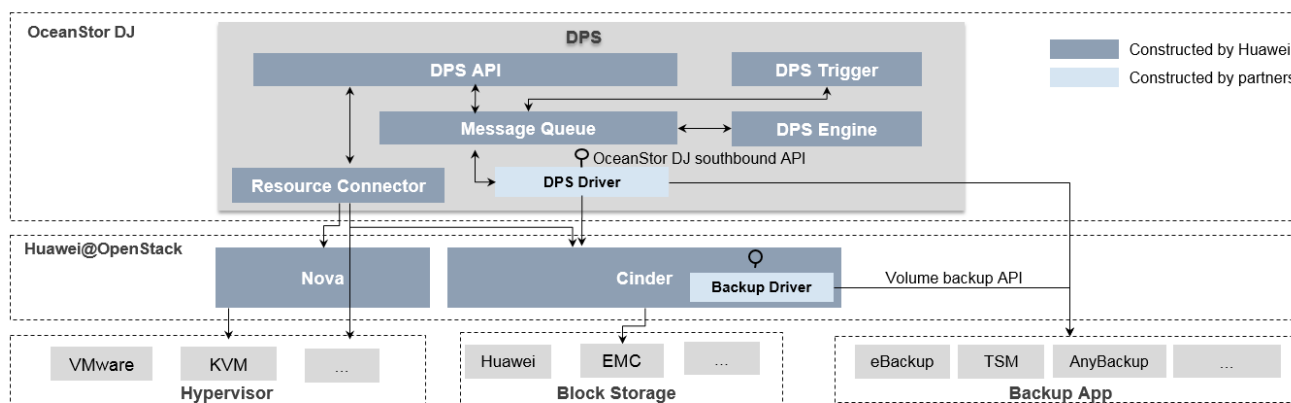
Figure 6-11 Conceptual diagram of the backup service



6.4.2 Technical Principles

The DPS component of OceanStor DJ provides a data protection service platform. It has a unique innovative scale-out scheduling engine and can support large-scale protection service scheduling. With the extended DPS Driver, DPS supports the access of various protection resources and software. Data protection as a service enables customers to use multiple pieces of backup software instead of only one, creating more business opportunities for partners.

Figure 6-12 DPS architecture



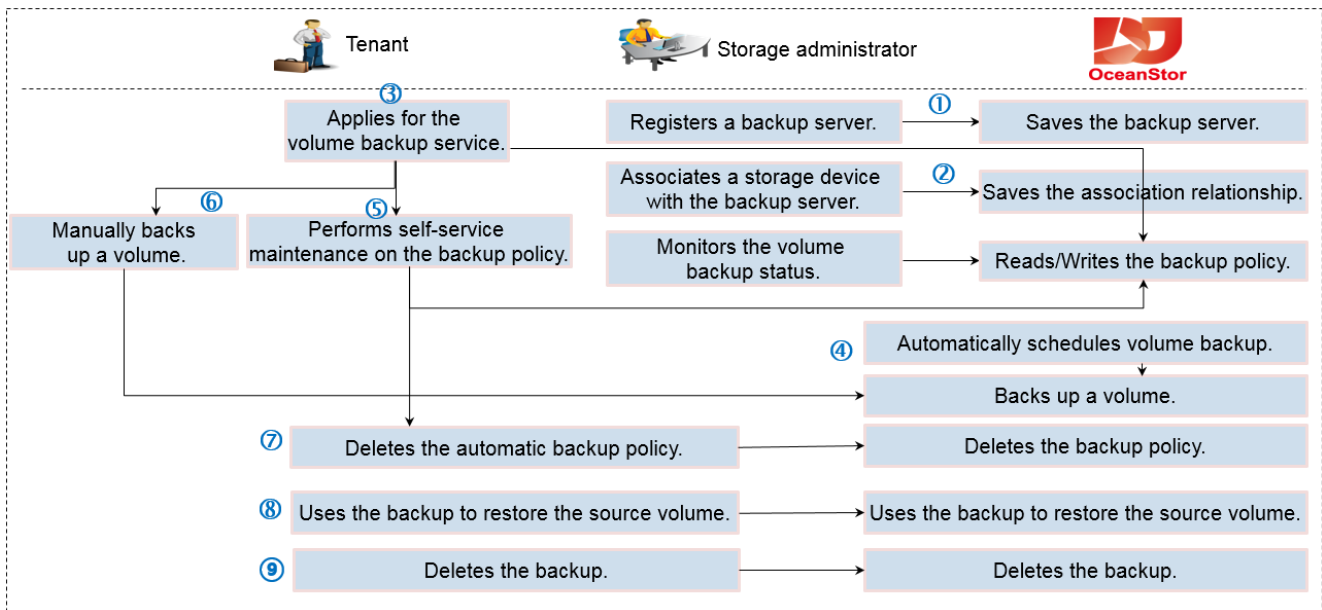
The components of DPS are described as follows:

- **DPS APIs:** REST APIs that provide protection services externally, such as volume backup and restoration.
- **Message Queue:** message bus, that is, message transmission channel.
- **DPS Engine:** selects proper data protection technologies based on data protection types and executes data protection based on policies.
- **Resource Connector:** obtains information about data protection resources such as VMs and volumes.
- **DPS Driver:** drives Cinder to implement volume backup or drives data protection software to execute data protection. DPS Driver is southbound protection driver interface defined in DPS.
- **Backup Driver:** southbound protection driver interface defined in Cinder.
- **DPS Trigger:** triggers scheduled tasks of DPS.

6.4.3 Service Process

Figure 6-13 shows the volume backup service process.

Figure 6-13 Volume backup service process



The volume backup service process is described as follows:

1. An administrator registers a backup server and saves the connection parameters into the database.
2. The administrator associates a storage device with the backup server.
3. A tenant user applies for the volume backup service.
4. The system automatically schedules volume backup.
5. The tenant user performs self-service maintenance on the backup policy.
6. The tenant user manually backs up a volume.
7. The tenant user deletes the automatic backup policy.
8. The tenant user uses the backup copy to restore the source volume or create a volume.
9. The tenant user deletes the backup copy.

6.5 Volume Replication Service

6.5.1 Introduction

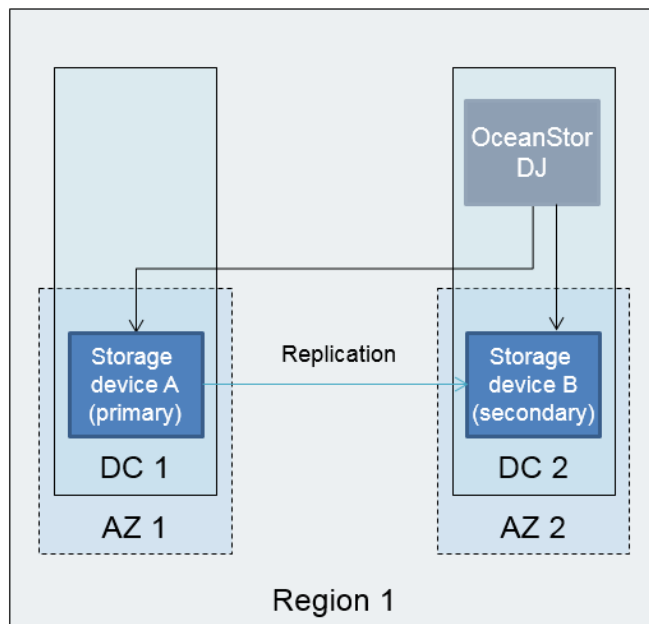
Data is very important for enterprises. To prevent production data loss caused by environmental disasters and to ensure service continuity, OceanStor DJ provides primary/secondary replication services for single volumes and consistency groups. OceanStor DJ supports the following replication modes:

- Synchronous replication
 The primary and secondary sites replicate data from each other in real time to maximize data consistency and minimize data loss in the event of a disaster.
- Asynchronous replication

The primary and secondary sites replicate data from each other periodically to minimize the service performance deterioration caused by the latency of remote data transfer.

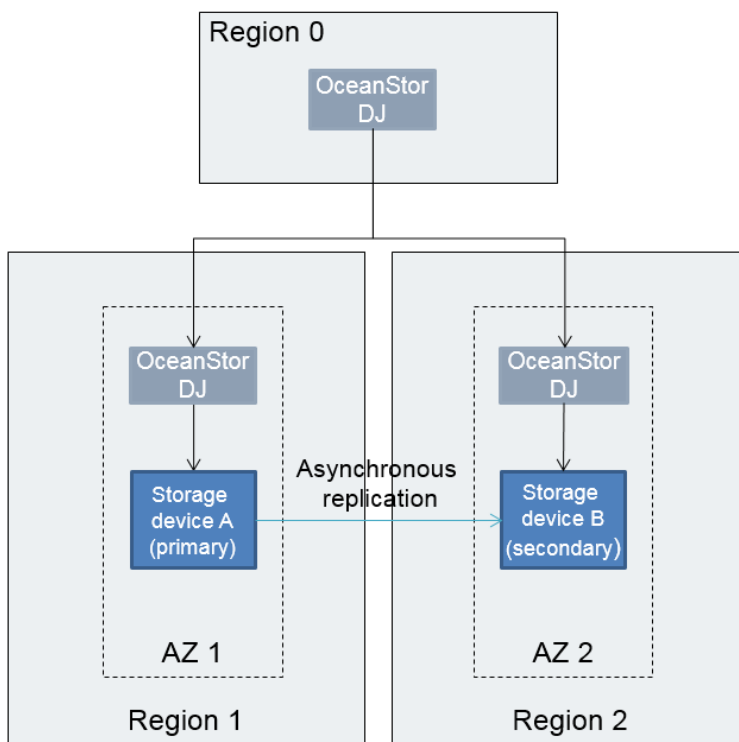
To provide volume replication services, at least two storage devices are required. They are deployed at the production end and disaster recovery end. Two application scenarios are supported: volume replication across multiple DCs and volume replication in a single DC, as shown in Figure 6-14. Asynchronous replication applies to volume replication across multiple DCs, and synchronous replication applies to volume replication in a single DC.

Figure 6-14 Networking for volume replication across multiple DCs (centralized management)



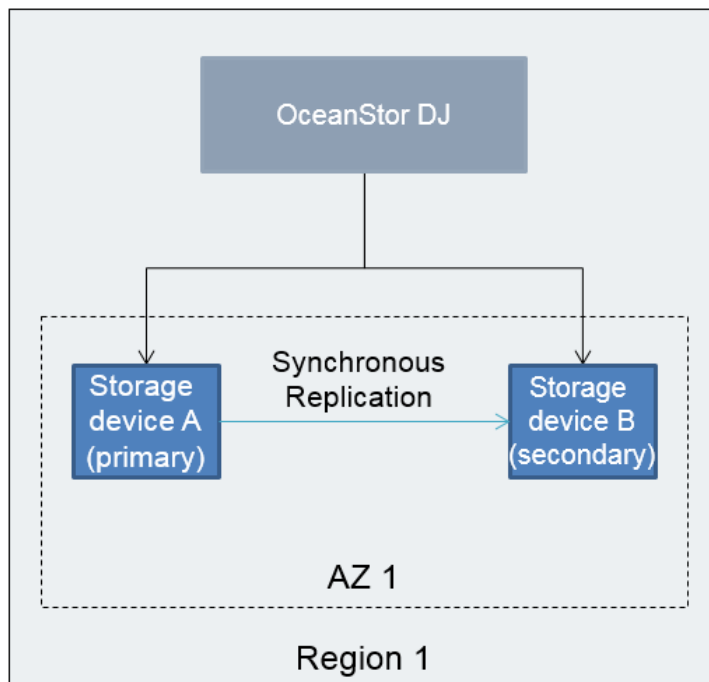
If storage devices at all DCs are connected to the same storage management network, you can deploy one set of OceanStor DJ in the disaster recovery DC to manage storage resources of both the production DC and disaster recovery DC and to execute volume replication operations.

Figure 6-15 Networking for volume replication across multiple DCs (distributed management)



If the storage resources of multiple DCs are beyond the management specifications of one set of OceanStor DJ or if the storage management networks of multiple DCs are isolated and storage devices of the DCs cannot be connected to the same storage management network, you can deploy one set of OceanStor DJ in each secondary DC at the production end and disaster recovery end and one set of OceanStor DJ in the primary DC. All volume replication operations can be performed on the OceanStor DJ GUI in the primary DC.

Figure 6-16 Networking for volume replication in a single DC



In the scenario of volume replication in a single DC, only one set of OceanStor DJ needs to be deployed to centrally manage production volumes and disaster recovery volumes and to execute volume replication operations. Storage devices at the production end and disaster recovery end can belong to the same AZ or different AZs.

6.5.2 Technical Principles

Volume replication services of OceanStor DJ use the same set of service components to support two deployment architectures: distributed management and centralized management. The set of service components includes Portal, Cinder, and DRExtend.

- Portal: provides the volume replication interface for tenant users and administrators.
- Cinder: There is one Cinder at each of the production end and disaster recovery end. Cinder creates production volumes or disaster recovery volumes on the local storage device.
- DRExtend: There is one DRExtend at each of the production end and disaster recovery end. DRExtend creates and manages replication relationships on the local storage device.

Figure 6-17 Volume replication service: distributed management by multiple OceanStor DJs

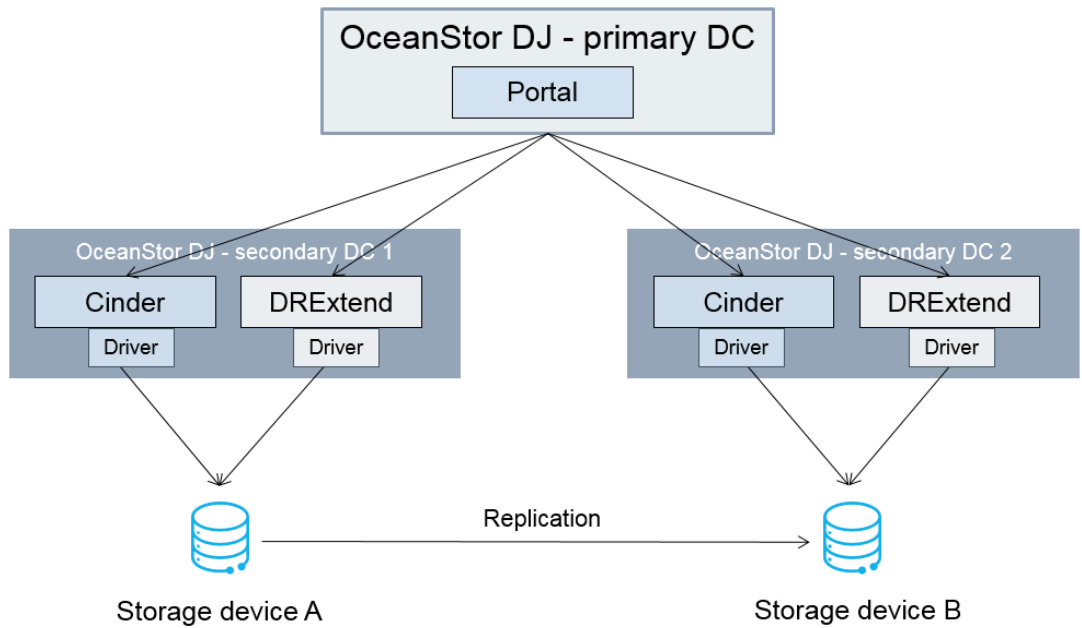
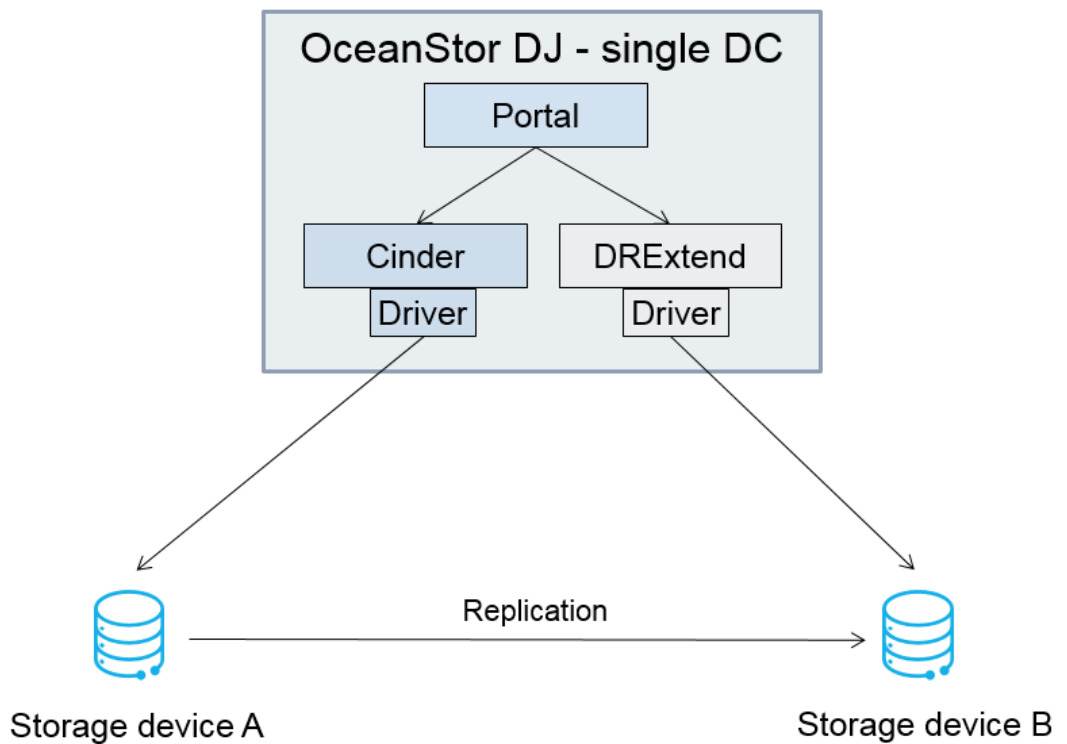
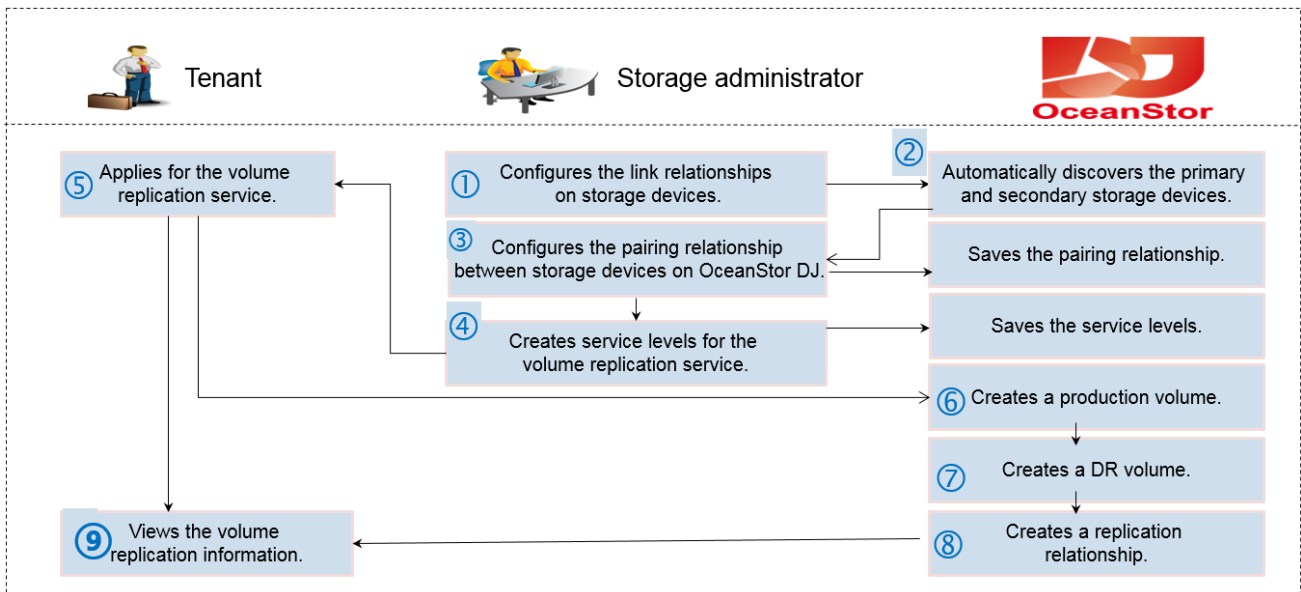


Figure 6-18 Volume replication service: centralized management by a single OceanStor DJ



6.5.3 Service Process

Figure 6-19 Volume replication service process



The volume replication service process is described as follows:

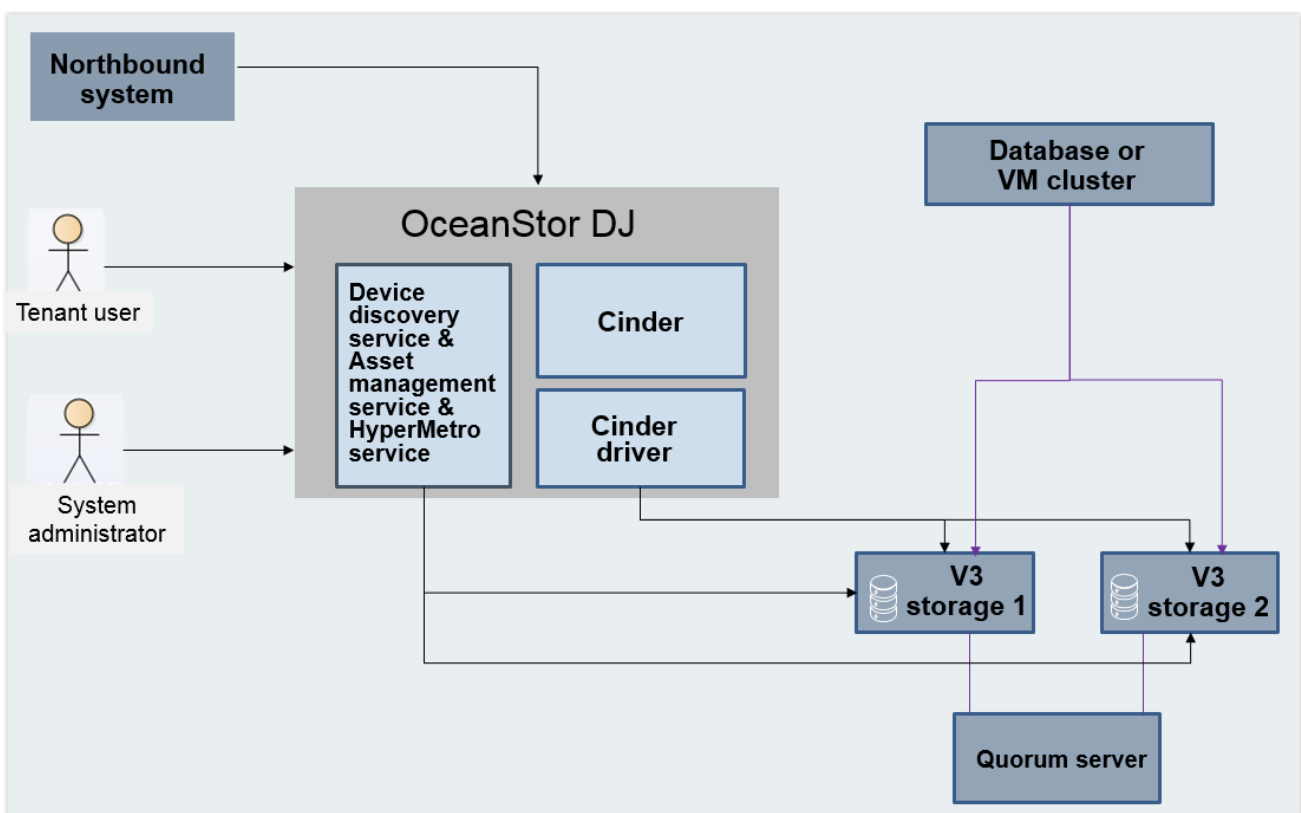
1. A storage administrator configures the link relationship between the storage devices at the production end and disaster recovery end.
2. OceanStor DJ automatically discovers the primary and secondary storage devices.
3. The storage administrator configures the pairing relationship between the storage devices at the production end and disaster recovery end on OceanStor DJ.
4. The storage administrator creates block storage service levels for the production end and disaster recovery end.
5. A tenant user selects the production AZ, block storage service level of a production volume, disaster recovery AZ, and block storage service level of a disaster recovery volume to apply for a volume replication service.
6. OceanStor DJ sends a command for creating a production volume to the storage device at the production end. The storage device then creates a production volume.
7. OceanStor DJ sends a command for creating a disaster recovery volume to the storage device at the disaster recovery end. The storage device then creates a disaster recovery volume.
8. OceanStor DJ sends a command for creating a replication relationship to the storage devices at the production end and disaster recovery end. The storage devices configure the replication relationship, which involves the replication mode and period.
9. The tenant user views the volume replication information. The tenant user can also perform data synchronization, fault-triggered switching, or primary/secondary switching on the GUI.

6.6 Volume HyperMetro Service

6.6.1 Introduction

OceanStor DJ provides the HyperMetro service for volumes. When provisioning volumes based on the service, OceanStor DJ selects the Cinder backend (back-end storage) with the HyperMetro capability, as well as creates a primary volume and a secondary volume, establishes a HyperMetro pair relationship, and completes consistency group configuration through the Cinder driver. If a user selects a service level with the HyperMetro capability enabled when applying for a volume service, OceanStor DJ will drive the HyperMetro capability of storage devices to automatically complete volume HyperMetro service configuration.

Figure 6-20 Conceptual diagram of the volume HyperMetro service



The system administrator completes basic configuration (including storage device pairing, storage pool management, HyperMetro-supporting resource pool management, and storage service level definition) for the volume HyperMetro service. The tenant user or northbound system can (through REST APIs) apply for the volume HyperMetro service. OceanStor DJ completes automatic configuration and resource provisioning for the volume HyperMetro service.

The device discovery service discovers storage systems.

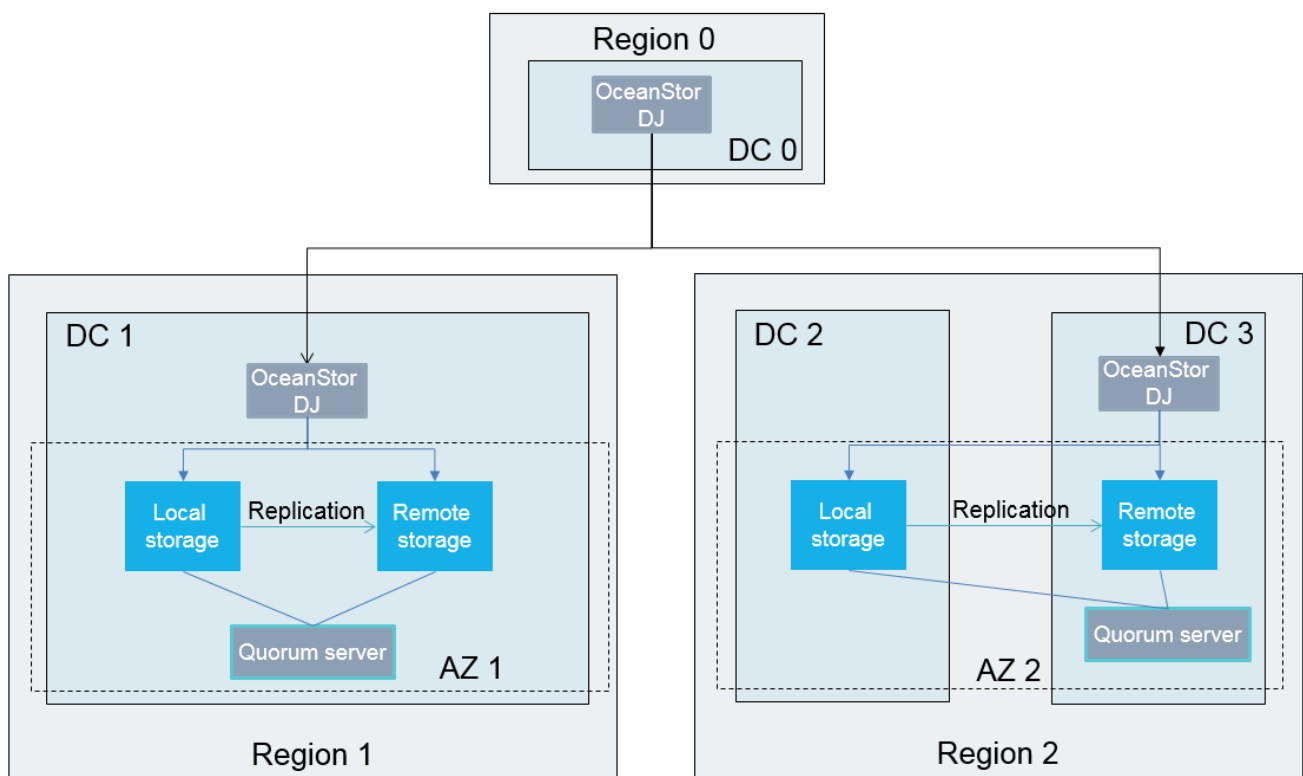
The asset management service manages storage system assets, including the information collection of physical storage pools, ports, and volumes.

The HyperMetro service configures the HyperMetro capability (including storage device pairing, link check, quorum server configuration, HyperMetro domain configuration, and storage pool pairing) and Cinder backend for OceanStor V3 storage devices.

6.6.2 Technical Principles

The HyperMetro service is based on the HyperMetro capability of storage devices in AZs. If a fault occurs on one storage device, host services are automatically switched to the other storage device, ensuring zero service interruption and zero data loss. In addition, the two storage devices are associated with a quorum server. If the link between the two storage devices fails, the quorum server determines which storage device will provide services for the host. Figure 6-21 illustrates the logical architecture.

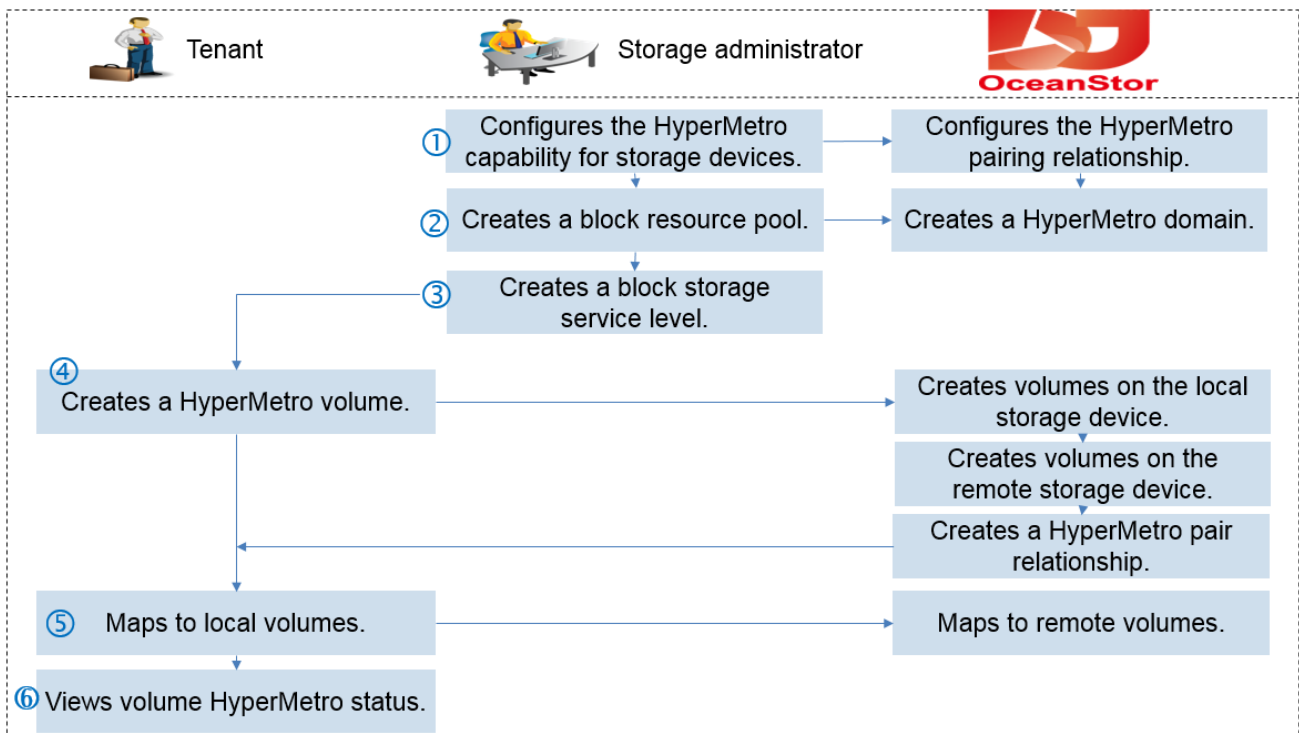
Figure 6-21 Logical architecture for the HyperMetro service



- In a single-DC scenario, as shown in region 1, one set of OceanStor DJ is deployed to manage resources in the DC. A HyperMetro pair is created between volumes on the local and remote storage devices in AZ 1 to implement real-time synchronization of volume data between the two storage devices and to provide read/write services for a host.
- In a multi-DC scenario, as shown in region 2, one set of OceanStor DJ is deployed in the DC where the remote storage device resides to manage resources in multiple DCs. A HyperMetro pair is created between volumes on the local and remote storage devices in DC 2 and DC 3 in AZ 2 to implement real-time synchronization of volume data between the two DCs and to provide read/write services for a host.
- In a multi-region scenario, as shown in Figure 6-21, one set of OceanStor DJ is deployed in each of region 0, region 1, and region 2 to manage resources in the regions. An administrator can also configure a primary region (region 0) and use OceanStor DJ in the primary region to manage resources in secondary regions.

6.6.3 Service Process

Figure 6-22 Volume HyperMetro service process



1. An administrator selects a remote storage device that will provide the HyperMetro capability for the local storage device (the local and remote storage devices must belong to the same AZ), and configures a HyperMetro relationship between the two storage devices. The administrator associates a quorum server with the storage devices and creates quorum server links, to create a HyperMetro domain. Then, the administrator configures a storage pool pair on the two storage devices.
2. The administrator creates a block resource pool that contains the local storage pool in the storage pool pair.
3. The administrator selects the created block resource pool to create a block storage service level with the HyperMetro capability.
4. A project user selects the preset **Create Volume with HyperMetro Capability** template, sets the basic volume information, and selects the block storage service level with the HyperMetro capability enabled to create HyperMetro volumes. OceanStor DJ automatically creates a local volume and a remote volume on the local and remote storage devices respectively and creates a HyperMetro pair between the two volumes.
5. The project user finds the created local volume and manually maps it to a host. If a HyperMetro volume is mapped to a host through FC, ensure that the network communication is normal between the remote device and the host. In addition, you must manually configure zones on the fabric where the remote device is located. Based on the mapping information of the local volume, OceanStor DJ automatically maps the remote volume to the same host. The project user logs in to the host and attaches the local and remote volumes to the host. When the host delivers a read/write request to a storage device, volume data is synchronized between the two storage devices in real time. Both

storage devices can provide read/write services for the host. If a fault occurs on a storage device, the host accesses data on the other storage device, ensuring service continuity.

6. The project user views volume HyperMetro information to learn the data status, link status, and HyperMetro pair status of the local and remote storage devices. In addition, the project user can manually perform or pause data synchronization.

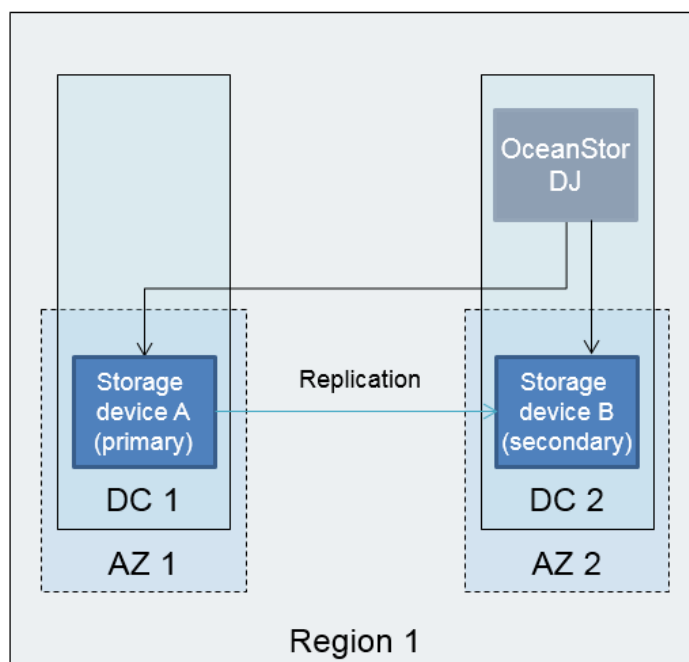
6.7 File Replication Service

6.7.1 Introduction

Data is very important for enterprises. To prevent production data loss caused by environmental disasters and to ensure service continuity, OceanStor DJ provides primary/secondary replication services for file shares. OceanStor DJ supports only asynchronous file replication. In asynchronous remote replication, data is periodically synchronized to minimize service performance deterioration caused by the latency of long-distance data transmission.

To provide file replication services, at least two storage devices are required. They are deployed at the production end and disaster recovery end. Two application scenarios are supported: file replication in a single region and file replication across regions, as shown in Figure 6-23.

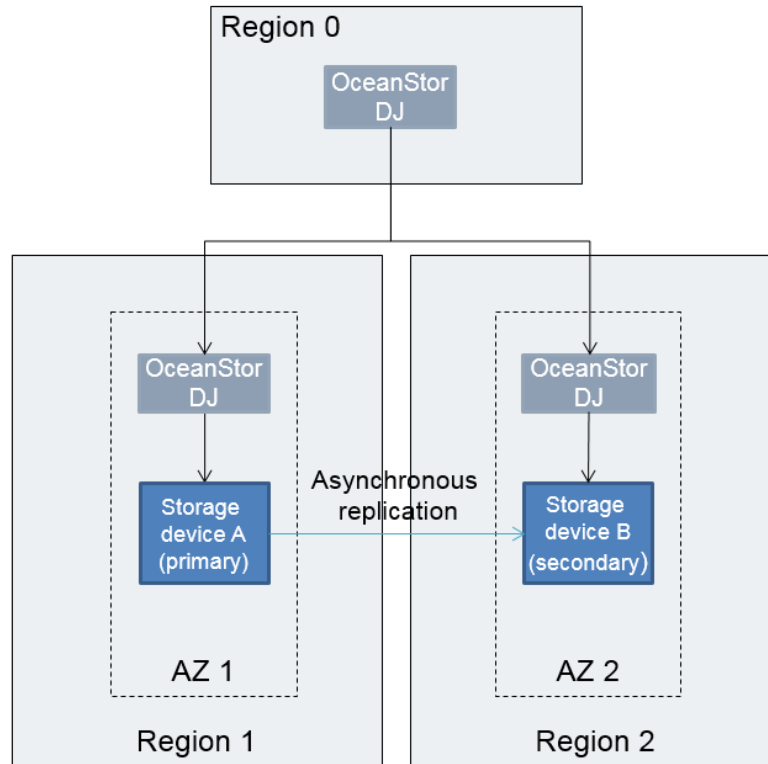
Figure 6-23 Networking for file replication in a single region



If storage devices at all DCs are connected to the same storage management network, you can deploy one set of OceanStor DJ in the disaster recovery DC or production DC to manage storage resources of both the production DC and disaster recovery DC and to perform file replication. In the scenario of file replication in a single DC, only one set of OceanStor DJ needs to be deployed to centrally manage production volumes and disaster recovery volumes

and to perform volume replication. Storage devices at the production end and disaster recovery end can belong to the same AZ or different AZs.

Figure 6-24 Networking for file replication across multiple regions



If the storage resources of multiple DCs are beyond the management specifications of one set of OceanStor DJ or if the storage management networks of multiple DCs are isolated and storage devices of the DCs cannot be connected to the same storage management network, you can deploy one set of OceanStor DJ in each secondary DC at the production end and disaster recovery end and one set of OceanStor DJ in the primary DC. All file replication operations can be performed on the OceanStor DJ GUI in the primary DC. File replication can be performed between secondary DCs or between a primary DC and a secondary DC.

6.7.2 Technical Principles

The file replication service and file HyperMetro service of OceanStor DJ use the same service components, including Portal, Manila, HyperFile, and Heat. OceanStor DJ supports two deployment modes: multiple DCs in multiple regions and a single DC in a single region. The components have the following functions in file replication:

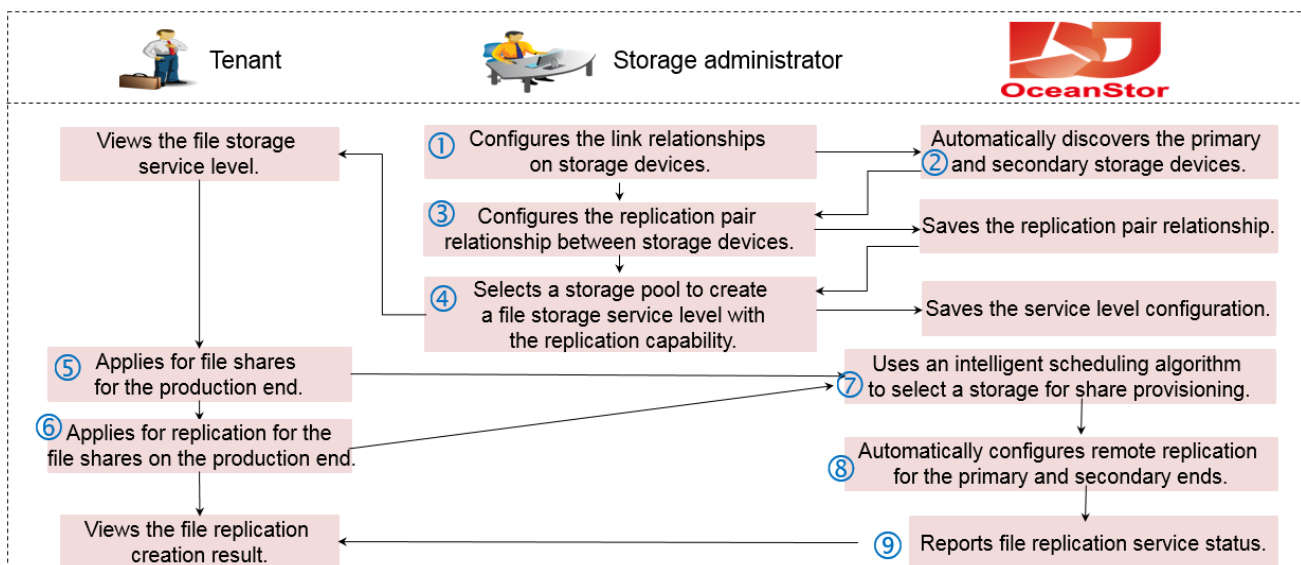
- Portal: provides the file replication interface for tenant users and administrators.
- Manila: provisions production file shares or disaster recovery file shares on storage devices.
- HyperFile: provides the capability of creating and managing replication relationships of local storage devices.
- Heat: implements the complete process of replication service provisioning by invoking APIs of different components. For example, when remote replication is created for existing file shares, Heat of the primary DC invokes Manila of the secondary DC to

provision file shares for the disaster recovery end, and then invokes HyperFile of the production end to configure the replication relationship of the file system.

The service invoking process is the same regardless of whether the multi-DC or single-DC deployment mode is used. File shares are first provisioned for the production end and then for the disaster recovery end, and a replication pair is created between file shares of both ends. In the multi-DC deployment mode, APIs of components are invoked across DCs; while in the single-DC deployment mode, APIs are invoked in the same DC. When file replication is set up between two secondary nodes, API invoking requests of tenants are transferred by HyperFile on the management node of the primary DC.

6.7.3 Service Process

Figure 6-25 File replication service process



1. A storage administrator configures the link relationship between the storage devices at the production end and disaster recovery end.
2. OceanStor DJ automatically discovers the primary and secondary storage devices.
3. The storage administrator configures the pairing relationship between the storage devices at the production end and disaster recovery end on OceanStor DJ.
4. The storage administrator creates file storage service levels for the production end and disaster recovery end.
5. A tenant user applies for file shares for the AZs and file service levels on the production end.
6. The tenant user applies for the file replication service for the regions, AZs, and file service levels on the disaster recovery end.
7. OceanStor DJ, based on the application from the tenant, sends a command for creating production file shares to the storage device on the production end, and sends a command for creating disaster recovery file shares to the storage device on the disaster recovery end.
8. OceanStor DJ sends a command for creating a replication relationship to the storage devices at the production end and disaster recovery end. The storage devices configure the replication relationship, which involves the replication mode and period.

9. The tenant user views the file replication information. The tenant user can also perform data synchronization, fault-triggered switching, or primary/secondary switching on the GUI.

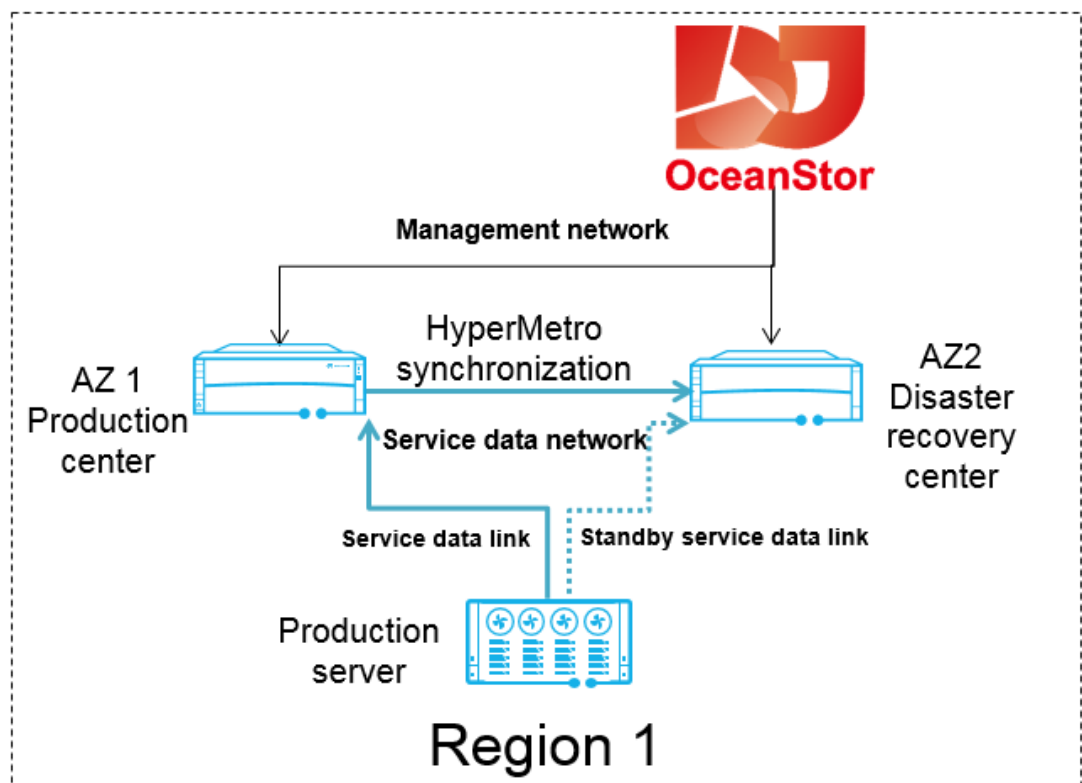
6.8 File HyperMetro Service

6.8.1 Introduction

Data is very important for enterprises. To prevent production data loss caused by environmental disasters and to ensure service continuity, OceanStor DJ provides HyperMetro services for file shares. Currently, the file HyperMetro service supports only OceanStor V3 storage devices.

To provide the file HyperMetro service, at least two OceanStor V3 storage devices need to be deployed on the primary and secondary ends of HyperMetro file shares, respectively. The production server is connected to the primary end of HyperMetro for reading and writing service data. The service data is replicated to the secondary end of HyperMetro. When services on the primary end fail or a network fault occurs, the secondary end automatically takes over the shared services and networks of the primary end. OceanStor DJ supports only file HyperMetro in a single DC, as shown in Figure 6-26.

Figure 6-26 Networking for file HyperMetro in a single DC



In the scenario of file HyperMetro in a single DC, only one set of OceanStor DJ needs to be deployed to centrally manage production volumes and disaster recovery volumes and to

perform file HyperMetro operations. Storage devices at the production end and disaster recovery end can belong to the same AZ or different AZs.

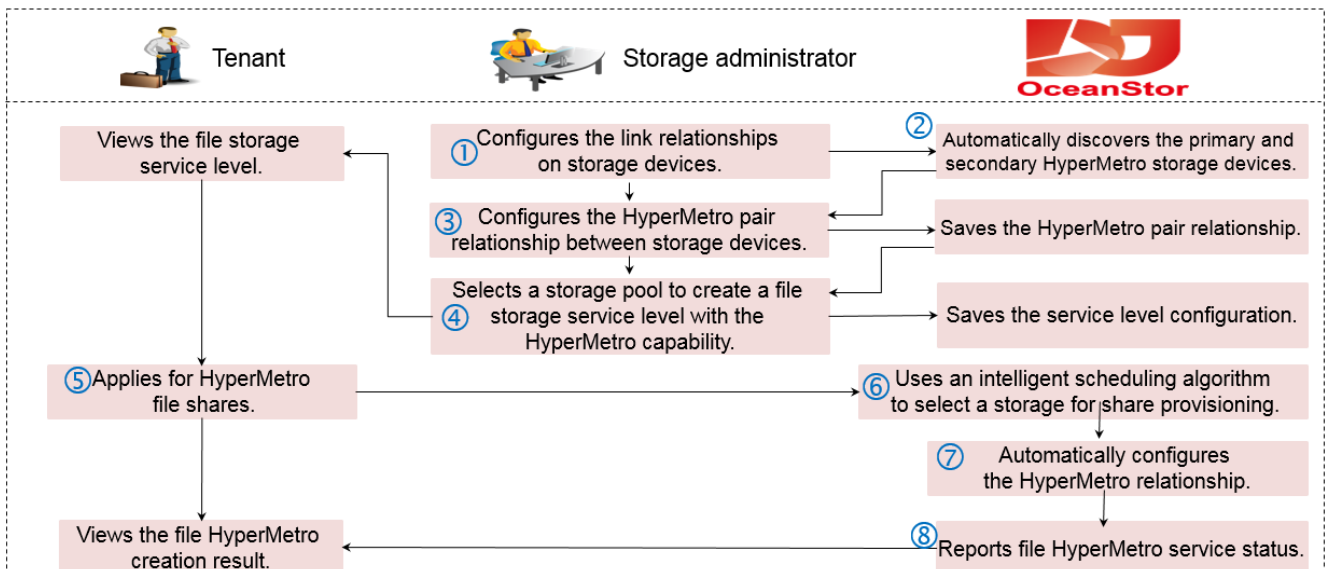
6.8.2 Technical Principles

The file replication service and file HyperMetro service of OceanStor DJ use the same service components, including Portal, Manila, HyperFile, and Heat. OceanStor DJ supports only file HyperMetro in a single region. The components have the following functions in file HyperMetro:

- Portal: provides the file HyperMetro interface for tenant users and administrators.
- Manila: provisions primary or secondary HyperMetro file shares on local storage devices.
- HyperFile: provides the capability of creating and managing HyperMetro relationships.
- Heat: implements the complete process of HyperMetro service provisioning by invoking APIs of different components. For example, when the file HyperMetro service is applied for by one click, Heat invokes Manila to provision file shares for the primary HyperMetro end and then also invokes Manila to provision file shares for the secondary HyperMetro end, and then invokes HyperFile to configure the HyperMetro relationship of the file system.

6.8.3 Service Process

Figure 6-27 File HyperMetro service process



1. A storage administrator configures the link relationship between the storage devices at the production end and disaster recovery end.
2. OceanStor DJ automatically scans for storage devices with HyperMetro links.
3. The storage administrator configures the HyperMetro pairing relationship for the storage devices with HyperMetro links.
4. The administrator selects a HyperMetro storage pool on OceanStor DJ to create file storage service levels with the HyperMetro capability enabled.

5. A tenant user applies for HyperMetro file shares for AZs and HyperMetro file service levels.
6. OceanStor DJ, based on the application from the tenant, schedules storage devices on the primary HyperMetro end to create files shares and then schedules storage devices on the secondary HyperMetro end to create files shares.
7. OceanStor DJ creates a HyperMetro pair between the primary and secondary shares and performs initial data synchronization.
8. The tenant user views the HyperMetro information of the file shares. The tenant user can configure the secondary HyperMetro end to writeable and readable on the GUI.

6.9 Application Storage Service

6.9.1 Introduction

Application storage services aim to provide cloud platform users with one-click services, including application, provisioning, and optimization of IT resources, for specific industry applications. IT resources include storage resources required for the operation of an industry application. The quantities, configurations, and proportions of the resources determine whether the operating environment of the application is efficient and cost-effective.

The IT resources required by an application should meet the performance and capacity requirements of the application operation and do not cause waste (that is, cost-effective). Application storage services provide each specific industry with a common platform and tools, which are used to generate storage resource application, provisioning, and optimization services. Storage resources generated through these services are optimal. To achieve the optimal IT resources, the following elements are introduced:

- Characteristic parameters of an application: determine the scale and configuration of IT resources required by the application. For example, the number of e-banking users determines the quantities of computing and storage resources that need to be allocated. There may be one or multiple characteristic parameters. When applying for IT resources for an application, an application administrator needs to provide characteristic parameters, which will be used by the system to compute required IT resources.
- Types of IT resources required by an application: include the deployment environment (physical machines or virtual machines), storage resource type (block storage or file storage), allocation relationships between storage resources and computing resources. Different applications require different IT resources.
- IT resource computation formulas: used to compute the capacities, quantities, and configurations of required IT resources based on an application's characteristic parameters.

The preceding three elements are called application workload data. IT resources can be provisioned so long as the three elements of an application are provided. How to determine the three elements to obtain the optimal IT resources relies on the experience of industry experts. Application storage services of OceanStor DJ provide the platform and tools required for resolving the problem.

OceanStor DJ defines a standard workload-based template format for applications. Any user (such as an industry expert) can create templates based on specific workload data.

NOTE

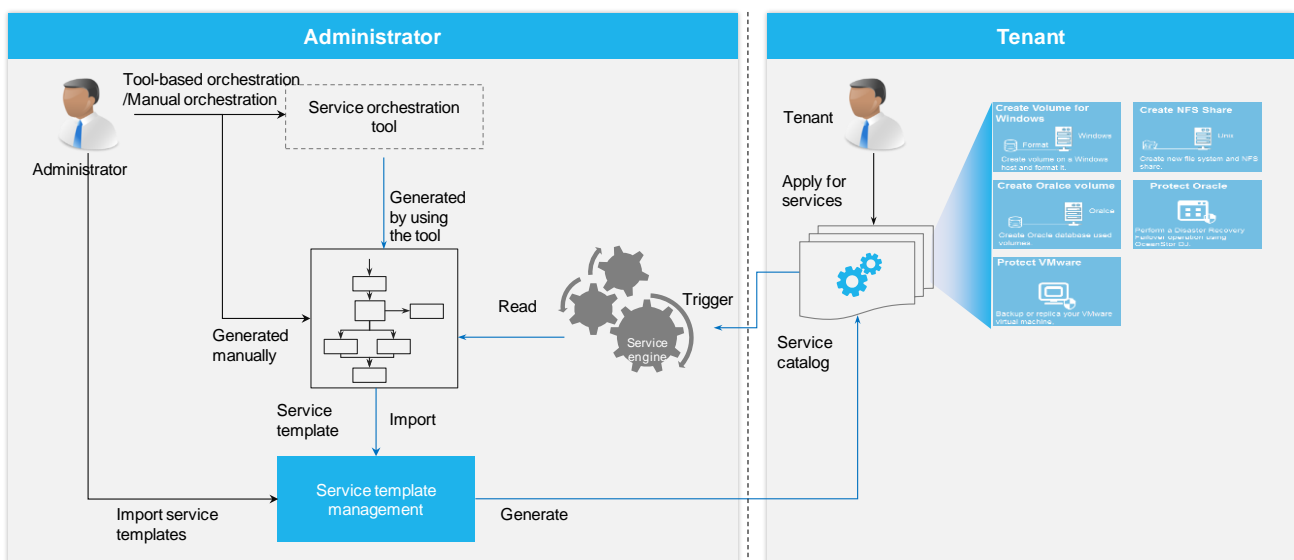
The templates can be generated by using the template development tool. Based on service requirements of customers, administrators need to enter related resource information and resource allocation calculating formulas in the tool to generate templates.

OceanStor DJ provides the workload-based template management function. A template can be imported into OceanStor DJ for reuse. OceanStor DJ is preconfigured with several typical industry workload-based templates, which can be used conveniently. Users can also create a workload-based template based on application characteristics of an industry and import the template into OceanStor DJ. Then, the template can be used by different departments in the industry in different time segments. OceanStor DJ automatically classifies and manages workload-based templates by industry.

OceanStor DJ provides the template parsing engine and IT resource provisioning engine. OceanStor DJ parses imported templates and dynamically generates application storage services. When applying for an application storage service, a user needs to fill in the application's characteristic parameters. The template parsing engine uses IT resource computation formulas to compute the required IT resources based on the characteristic parameters. The IT resources include servers (quantity and performance), network (VLANs and IP addresses), storage (volumes and file systems), and association between the resources. Then, the resource provisioning engine automatically creates and configures the resources. Finally, the IT resources are provisioned in one-click mode.

6.9.2 Technical Principles

Figure 6-28 Conceptual diagram of service management



OceanStor DJ provides operation interfaces for system administrators and tenants. System administrators are responsible for managing service catalogs. Templates can be written by system administrators, generated by using the template development tool, or written by experts and imported by system administrators. In addition, OceanStor DJ is preconfigured with some service templates based on industry solution best practices.

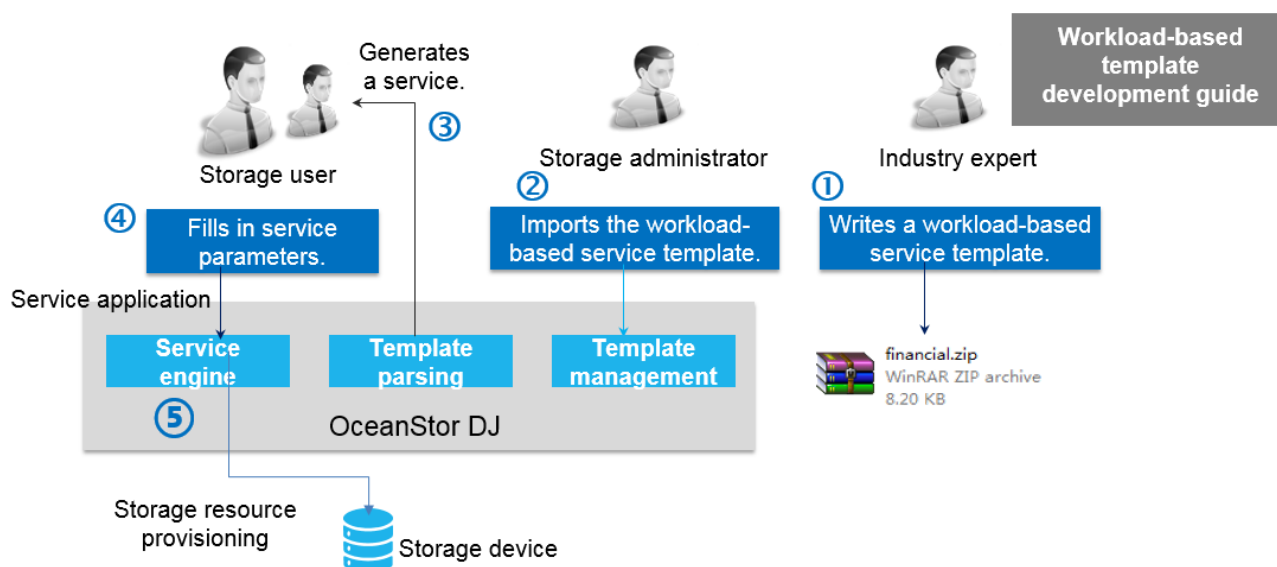
System administrators can import, modify, or delete industry templates.

After any template is imported into OceanStor, a service page is generated. Users can browse all the imported industry application service templates and apply for services. OceanStor DJ then automatically creates related IT resources.

Users can manage IT resources that they have obtained. OceanStor manages IT resources by group. When a user applies for a service, OceanStor DJ creates a group of IT resources. The user can view IT resources in the group or delete the whole IT resource group.

6.9.3 Service Process

Figure 6-29 Application storage service process



The service provisioning process is described as follows (an industry template used for provisioning volumes is used as an example):

1. A consulting team creates a workload-based storage service template. (The template can be written manually or generated by using the template development tool. The template development tool is provided for industry experts to directly generate templates).
2. A system administrator loads the template.
3. OceanStor DJ parses the template and generates the corresponding service catalog.
4. A service department user fills in service parameters (such as the number of e-banking users) based on the service catalog to apply for storage resources.
5. OceanStor DJ selects storage arrays that meet requirements in the resource pool and creates and attaches volumes.

6.10 Multi-Region Management

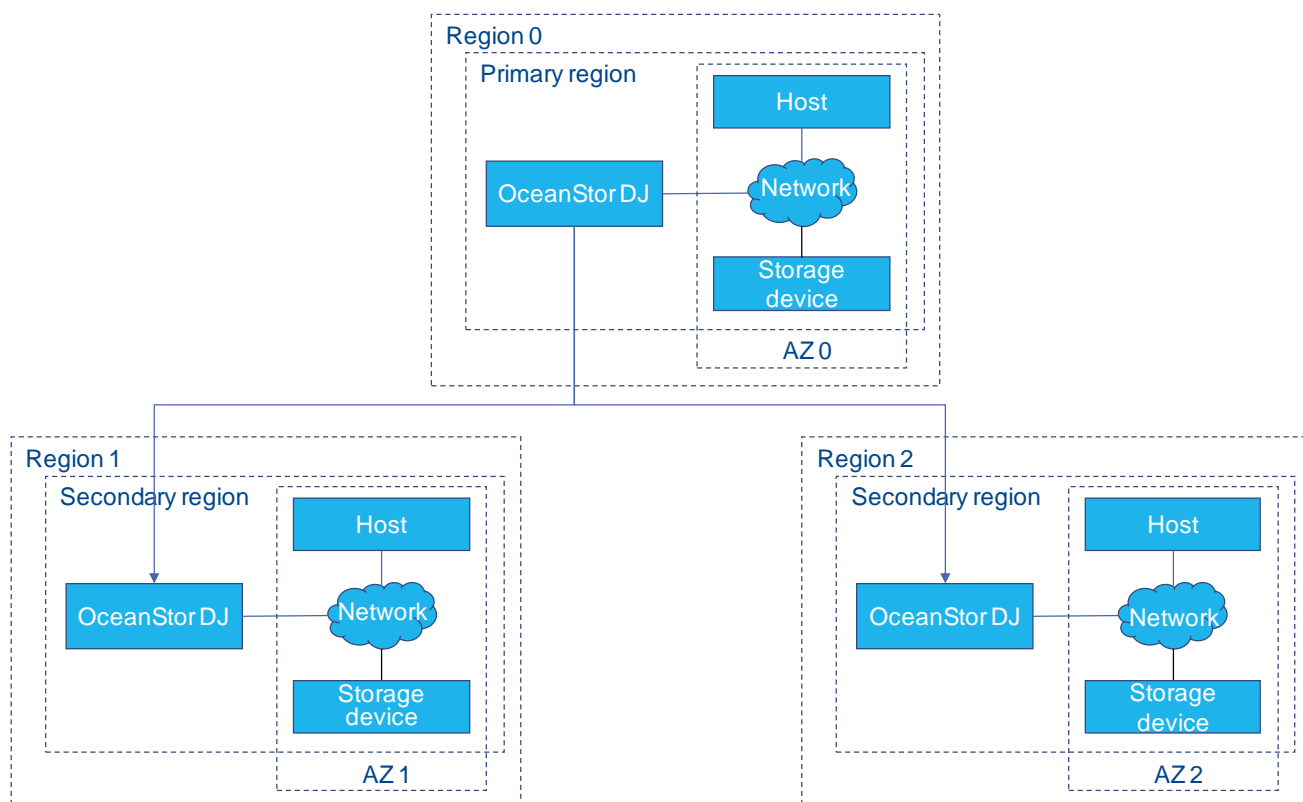
6.10.1 Introduction

The multi-region management feature enables administrators to manage multiple regions on the same GUI. If a customer wants to construct multiple regions and requires that one region should manage resources of all the other regions, an administrator needs to deploy an OceanStor DJ in the primary region and an OceanStor DJ in each secondary region. After being deployed, the OceanStor DJs in secondary regions register with the primary region.

After successful registration, the administrator can log in to the OceanStor DJ administrator GUI of the primary region to view and modify information about secondary regions. The administrator can also switch among resource views of different regions to manage resources of the regions in a unified manner. Tenants can also switch among resource views of different regions to apply for storage resources in the specified regions.

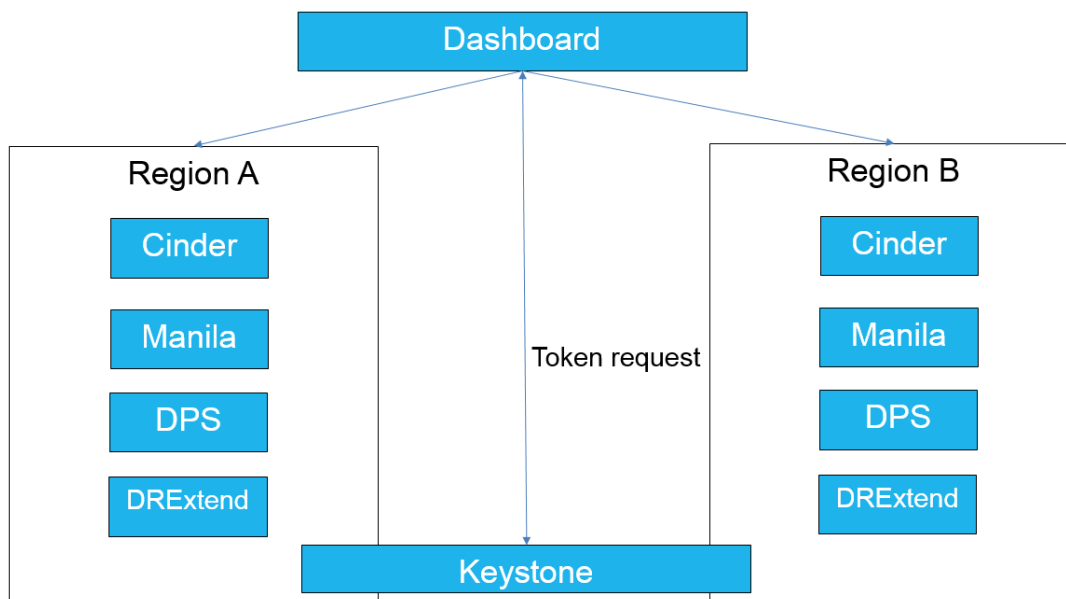
The multi-region management feature is used in cross-region resource management scenarios. When a large number of storage devices are required to support services and the resource management scope and permissions are limited (by specifications) in each region, the multi-region management feature needs to be used. This feature increases the resource management scale and implements resource isolation.

Figure 6-30 Networking for multi-region management



6.10.2 Technical Principles

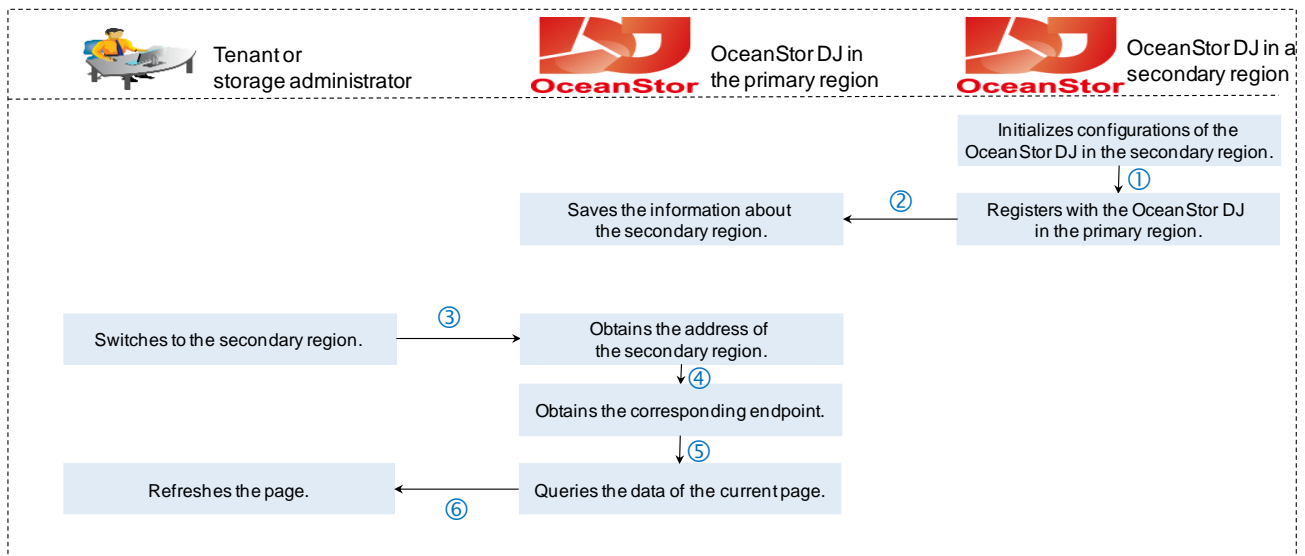
Figure 6-31 Principles of multi-region management



Multi-region management is inherited from the OpenStack multiple-region solution. In this solution, one set of OceanStor DJ is deployed in each region. When installing an OceanStor DJ, users need to specify whether the OceanStor DJ belongs to the primary region or a secondary region. After an OceanStor DJ is installed in a secondary region, components of the OceanStor DJ interconnect with Keystone of OceanStor DJ in the primary region. Storage devices in a secondary region can be managed by the primary region using OceanStor DJ in the secondary region so that unified authentication and centralized management can be implemented.

6.10.3 Service Process

Figure 6-32 Region switching process



The region switching process is described as follows:

1. When installing an OceanStor DJ in a secondary region, a storage administrator initializes configurations of the OceanStor DJ.
2. The OceanStor DJ in the secondary region registers with that in the primary region.
3. A tenant or storage administrator switches from the primary region to the secondary region through Dashboard.
4. The OceanStor DJ in the primary region obtains the address of the secondary region.
5. The OceanStor DJ in the primary region obtains the data of the current page on the OceanStor DJ in the secondary region.
6. After the switching, the tenant or storage administrator refreshes the page. The OceanStor DJ management information of the secondary region is displayed.

6.11 Host Management

6.11.1 Introduction

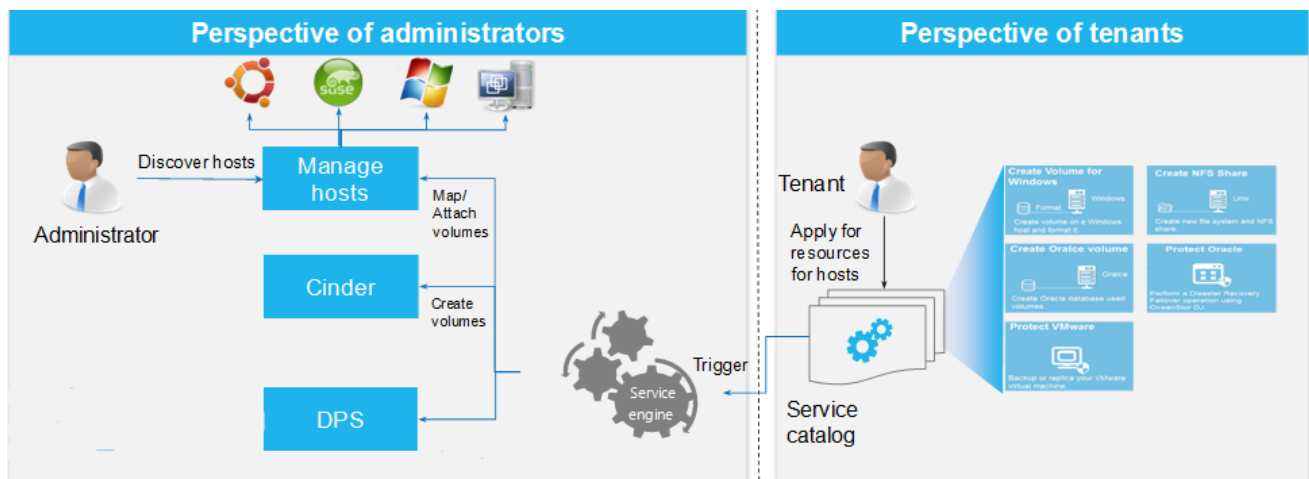
The host management function provides cloud platform users with end-to-end resource allocation and data protection services for hosts. Huawei wants to implement allocation, management, optimization, and protection of storage resources from the perspective of user services because users' applications are installed on hosts. Management of host groups is supported. With this function, you can allocate resources in the units of host groups.

The resource allocation and data protection for hosts involve the following scenarios:

- Basic host management (administrator): Administrators first obtain host information using a basic management protocol (such as SSH). Then, administrators can view or modify host information and status.
- Application for storage resources for hosts (tenant): After administrators obtain host information through OceanStor DJ, tenants can apply for storage resources for hosts. During the application process, tenants select the desired hosts, and then OceanStor DJ automatically executes resource allocation and data protection for the hosts.
- Support for host groups: OceanStor DJ provides the host cluster feature of OceanStor V3 as a service for managing hosts. After an administrator adds hosts to OceanStor DJ and allocates the hosts to a tenant, users in this tenant can use the hosts to create host groups and assign block resources to the host groups. In scenarios where the customer's existing network contains production data, you can use the host takeover function to manage production data after the data is migrated to Huawei storage.

6.11.2 Technical Principles

Figure 6-33 Host management principles



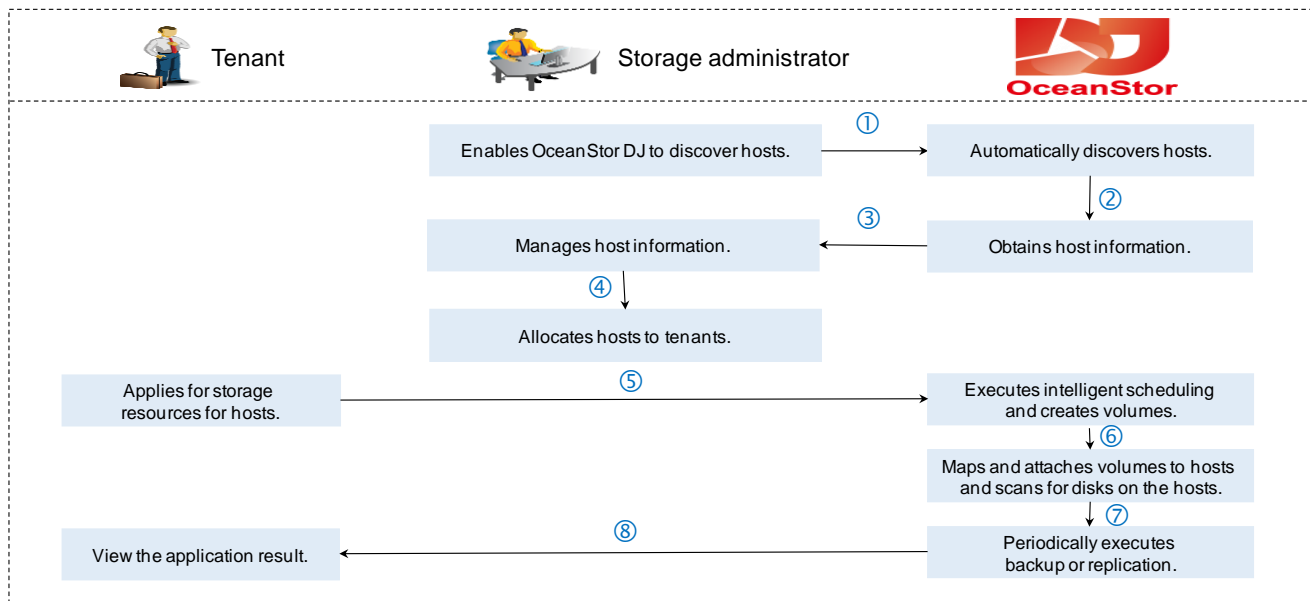
Administrators can manage host information, including discovering, modifying, and deleting host information. In addition, administrators can configure operating system types and other information for hosts.

Tenants apply for storage resources for hosts through service catalogs. When selecting service levels, tenants can determine whether to select data protection capabilities. After service applications are submitted successfully, OceanStor DJ automatically invokes Cinder and host management components to execute corresponding operations to meet tenants' requirements.

Administrators can add, delete, modify, and view host groups, add hosts to and remove hosts from host groups, as well as refresh initiator information when initiators of hosts or arrays are changed. After a host group has been created, tenant users can allocate volumes to hosts in the host group.

6.11.3 Service Process

Figure 6-34 Process of applying for resources for hosts



The process of applying for resources for hosts is described as follows:

1. A storage administrator enables OceanStor DJ to discover hosts by using a protocol such as SSH.
2. OceanStor DJ obtains the information about the discovered hosts.
3. The reported host information is saved into the database. The storage administrator can manage the host information on the OceanStor DJ GUI.
4. The storage administrator can allocate the host information to different tenants to determine the tenants' operation scopes.
5. A tenant applies for storage resources for hosts in the operation scope. After the tenant submits a service application, OceanStor DJ executes scheduling based on the selected service level, selects the appropriate storage device, and allocates storage resources.
6. OceanStor DJ maps and attaches volumes to the hosts and scans for disks on the hosts.
7. If the service level selected by the tenant is enabled with data protection capabilities, OceanStor DJ automatically executes backup or replication for the volumes on the hosts.
8. The tenant views the service application result on the resource page.

6.12 VMware vRO and vRA Integration

6.12.1 Introduction

VMware vRealize Orchestrator (vRO) is an automation and management engine provided by VMware. Huawei vRO Plugin, complying with VMware vRO API standards, is a workflow automation plug-in for managing Huawei storage devices. Packed with common workflows, vRO Plugin enables administrators to start workflows for automatic task processing. In addition, VMware vRO provides a friendly designer. This allows users to customize workflows and design complex automation tasks based on

pre-configured workflows provided by vRO Plugin. In this way, automatic configuration management of storage services is implemented. VMware vRealize Automation (vRA) is an IT automation tool based on vRO. It enables you to create and manage customized basic architecture, applications, and customized IT services (XaaS) to achieve IT automation.

The main application scenarios of VMware vRO are as follows:

- **Block storage service:** allocates and maps LUNs to ESXi hosts. In vCenter, the LUNs are formatted to VMFS datastores (big LUNs) and virtual machine disks (VMDKs) based on the datastore. Volumes with HyperMetro capabilities can be allocated.
- **File share service:** creates file shares and configures access permissions on ESXi hosts. In vCenter, the shares will be formatted to NFS datastores and VMDKs will be allocated to VMs based on the datastore.

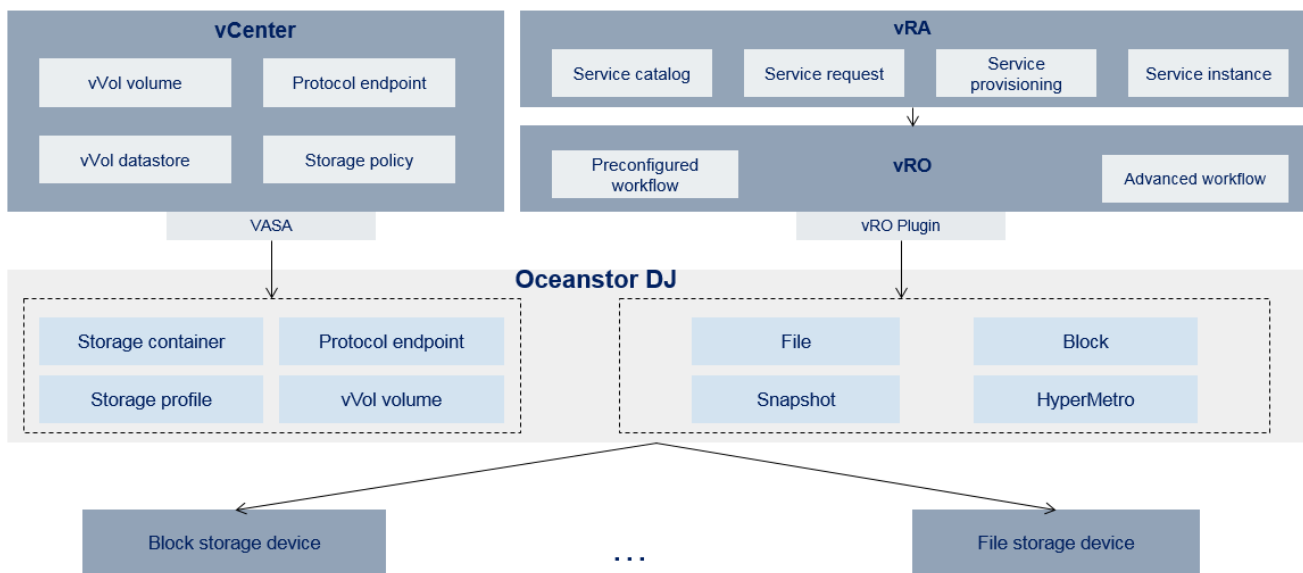
Table 6-2 List of vRO workflows

Service Catalog	File Service	Block Service
Advanced service	N/A	1) Create Datastore for Cluster
		2) Delete Datastore from Cluster
Basic service	1) Create file system	1) Create volume
	2) Delete file system	2) Delete volume
	3) Expand file system	3) Expand volume
	4) Query mount path by file system ID	4) Query WWN by volume ID
	5) Query file system ID by mount path	5) Query volume ID by WWN
	6) Add file system access rules	6) Volume mapping
	7) Delete file system access rules	7) Volume unmapping
		8) Create volume for host
		9) Attach Volume to HG
Snapshot service	1) Create file system snapshot	1) Create volume snapshot
	2) Delete file system snapshot	2) Create new volume from snapshot
		3) Delete volume snapshot
HyperMetro-based disaster recovery service	N/A	1) Create volume with AA DR capability
		2) Delete volume with AA DR capability
Replication-based disaster recovery	N/A	1) Create volume with replication capability

service	2) Delete volume with replication capability
---------	--

6.12.2 Technical Principles

Figure 6-35 vRO working principles



vRO Plugin adopts the Client/Server architecture. Its client is responsible for orchestration, management, and startup of workflows and its server is responsible for the running of workflows. vRO Plugin provides OceanStor DJ integration- and management-related workflows. The server provides and manages the storage services provided by OceanStor DJ through vRO Plugin.

OceanStor DJ pools storage resources and provides the resources as services. It orchestrates storage resources provided by Huawei storage and third-party storage, provides storage services in the form of storage service levels, such as block storage service levels and file storage service levels, and provides protection services, such as the snapshot feature and HyperMetro. vRO Plugin can invoke all services through RESTful APIs. If a feature is involved with multiple APIs, vRO Plugin will be also responsible for the feature's processing logic and encapsulation.

6.12.3 Service Process

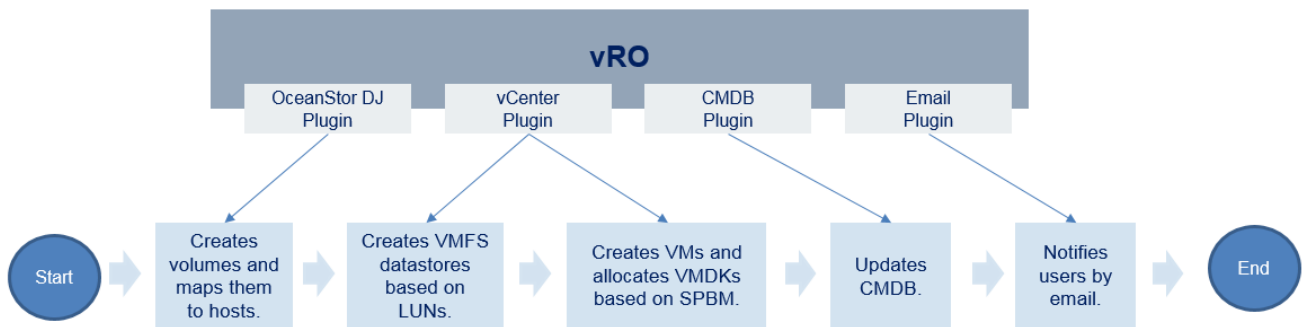
After vRO Plugin is installed on OceanStor DJ, you can use workflows in VMware. In normal cases, users install and deploy two components of VMware vRealize suite: vRO Plugin and vRealize Automation (vRA). With these two components installed, workflows can be used in three modes:

- vRO administrators run the OceanStor DJ-related pre-configured workflows on the vRO platform independently.
- On the vRO platform, vRO administrators perform composite orchestration on OceanStor DJ-related pre-configured workflows and other workflows.

- On the vRA platform, vRA administrators release storage workflows provided by OceanStor DJ as storage services, and vRA tenant users can apply for these released storage services on the tenant GUI.

Figure 6-36 illustrates the process how to combine the workflow for creating and mapping volumes for hosts on OceanStor DJ and a workflow provided by vCenter, CMDB, Email to an automatically executed high-level workflow through composite orchestration.

Figure 6-36 vRO service process



6.13 VMware VASA Integration

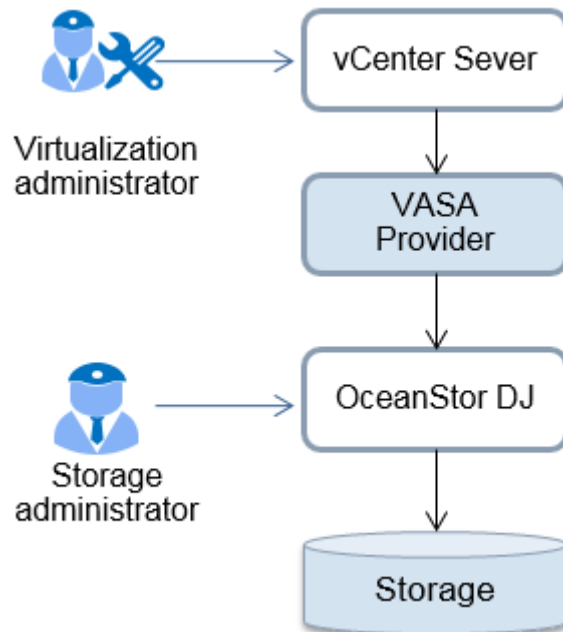
6.13.1 Introduction

VASA Provider 2.0 (VASA Provider for short) is developed by Huawei and meets the VASA standard. It enables vCenter Server to manage Huawei storage devices. VASA Provider supports the VMware virtual volume (VVOL). OceanStor DJ is service-driven storage control software that manages storage resources and provides service-driven and automated data services in the STaaS Solution. In the STaaS Solution, VASA provider and OceanStor DJ are integrated and work together to implement the VVOL function, simplifying the provisioning and management of VVOL resources in large-scale storage environment.

In the scenario where the STaaS Solution interconnects with VMware, OceanStor DJ integrated with VASA Provider interconnects with VMware vCenter to query and configure VASA-related resources. Functions provided include:

- Adding storage devices
- Configuring VASA service levels
- Querying Storage Container
- Querying Storage Profile
- Configuring and mapping PE LUN groups
- Managing VVols

Figure 6-37 Conceptual diagram of the interconnection between OceanStor DJ and VASA Provider



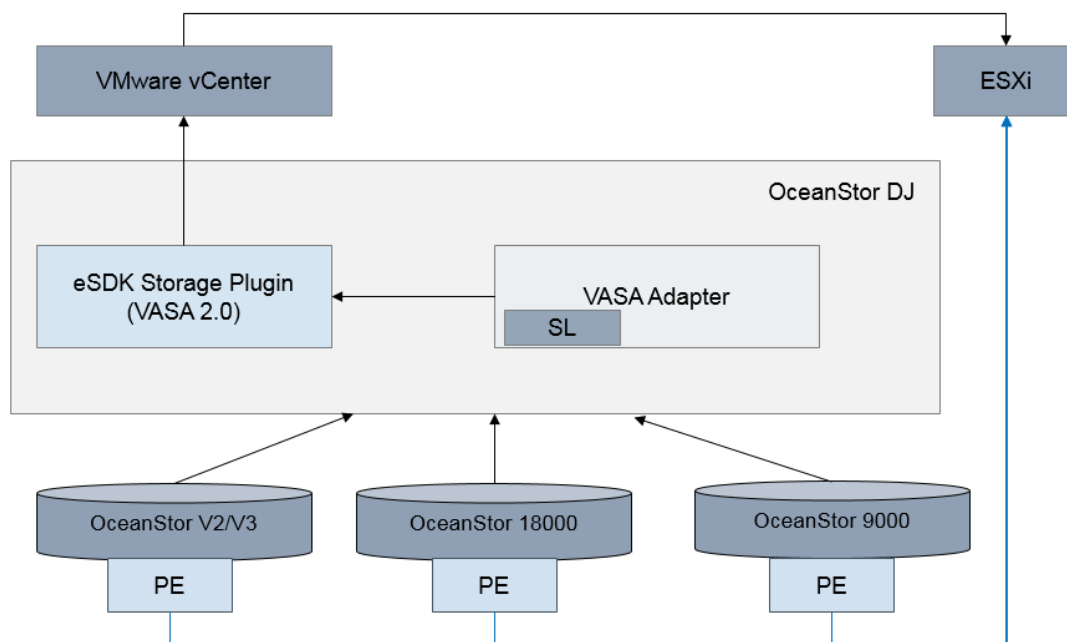
OceanStor DJ and vCenter have different functions in storage resource management. In the small-LUN scenario, OceanStor DJ initializes the configurations of PE and SC, while vCenter creates VVol datastores and allocates VVols based on SPBM. In the big-LUN scenario, OceanStor DJ creates volumes and maps them to hosts, or creates file systems and exports them to hosts, while vCenter creates VMFS/NFS datastores based on LUNs and allocates VMDKs based on SPBM.

6.13.2 Technical Principles

OceanStor DJ integrated with eSDK Storage Plugin (VASA 2.0, that is, VASA Provider 2.0 standard interface) can be used by VMware to manage Huawei storage devices with VVol enabled. In this way, OceanStor DJ has the following capabilities:

- Provides unified management of storage devices and vCenter clusters, and uses Huawei storage devices that support the VVol feature on demand.
- Provides unified block storage service levels to configure Storage Container and Storage Profile for VASA Provider.
- Creates and maps PE LUN groups, and configures VASA PE (Protocol Endpoint) to simplify multi-end operations.
- Creates VVols. Proper physical pools in Storage Container are selected based on Storage Profile.
- Creates VVol snapshots, and provides Clone and FastClone.
- Uses VASA Provider to save VVol metadata and Storage Profile.

Figure 6-38 VASA feature description



6.13.3 Service Process

The VMware VASA integration service process includes the following steps:

1. Install VASA Provider to interconnect it with OceanStor DJ HAProxy and database.
2. Enable VASA Provider on OceanStor DJ.
3. Add a storage device to OceanStor DJ.
4. Register the storage device with VASA Provider.
5. Create a VASA service level on OceanStor DJ (VASA Provider generates Storage Container and Storage Profile).
6. Interconnect OceanStor DJ with vCenter.
7. Generate a host group (corresponding to the vCenter cluster) on OceanStor DJ.
8. Create a PE LUN group.
9. Map the PE LUN group to the host group.
10. Register VASA Provider with vCenter.
11. Use VVols to create a VM based on VVol datastores on vCenter.

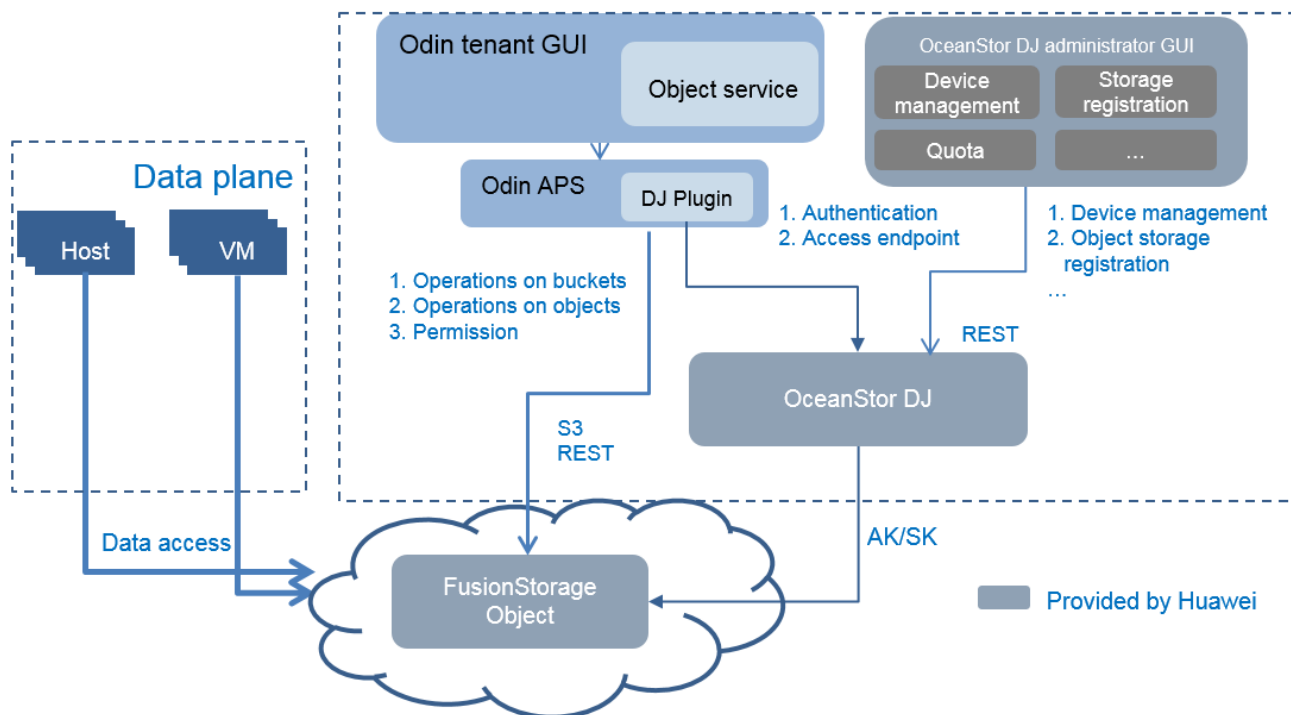
6.14 Connection to Odin

6.14.1 Introduction

Odin provides software for the entire cloud ecosystem, including web server management, server virtualization, configuration delivery, and charging automation. By integrating OceanStor DJ with Odin, customers can use STaaS to implement automatic storage

management on the Odin GUI. Operations can be performed on buckets, objects, tenants, and AK/SK.

Figure 6-39 Conceptual diagram of the interconnection between OceanStor DJ and Odin

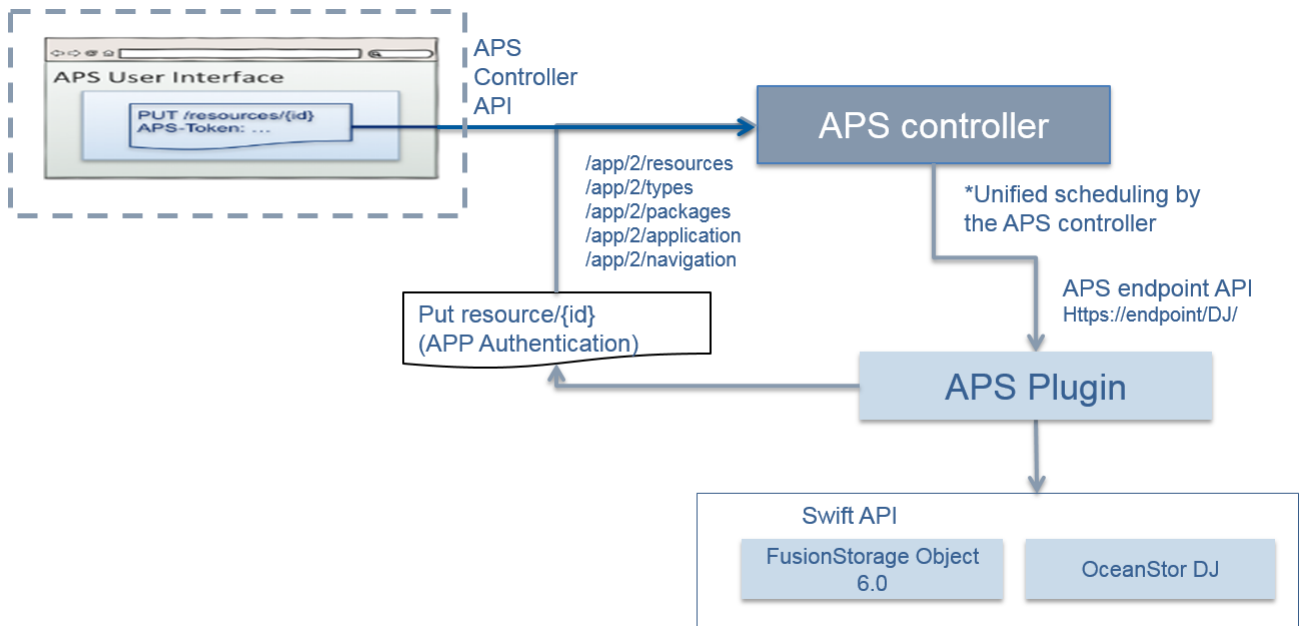


6.14.2 Technical Principles

Design of the interconnection between OceanStor DJ and Odin includes:

- **Front-end portal:** APS provides a general development package and an integrated tool package for the user portal.
- **Service logic processing:** Front-end portal requests received through the REST API adopts the universal interface model defined by APS (resource type, application program, navigation, and UI running).
- **Back-end logic:** APS Plugin for interconnecting with OceanStor DJ is developed to adapt to and forward upper-layer requests.

Figure 6-40 Implementation architecture of the interconnection between OceanStor DJ and Odin



For the current version of OceanStor DJ, only POC tests are preformed for the interconnection between OceanStor DJ and Odin.

6.14.3 Service Process

The typical interconnection process is as follows:

1. After discovering an object device, configure the device to be connected based on S3 protocol on OceanStor DJ.
2. Deploy StaaS Plugin. OceanStor DJ services can be viewed on Odin.
3. Users purchase services used for product deployment on Odin. Then Odin invokes the **cloud_admin** administrator to create accounts and create projects on OceanStor DJ. A default bucket and a default project administrator are created at the same time, and the project administrator has all operation permissions on the bucket.
4. Enterprise administrators can create buckets, create users, set ACLs, and manage objects.
5. After being granted with permission using ACLs, other users can use buckets to manage objects.

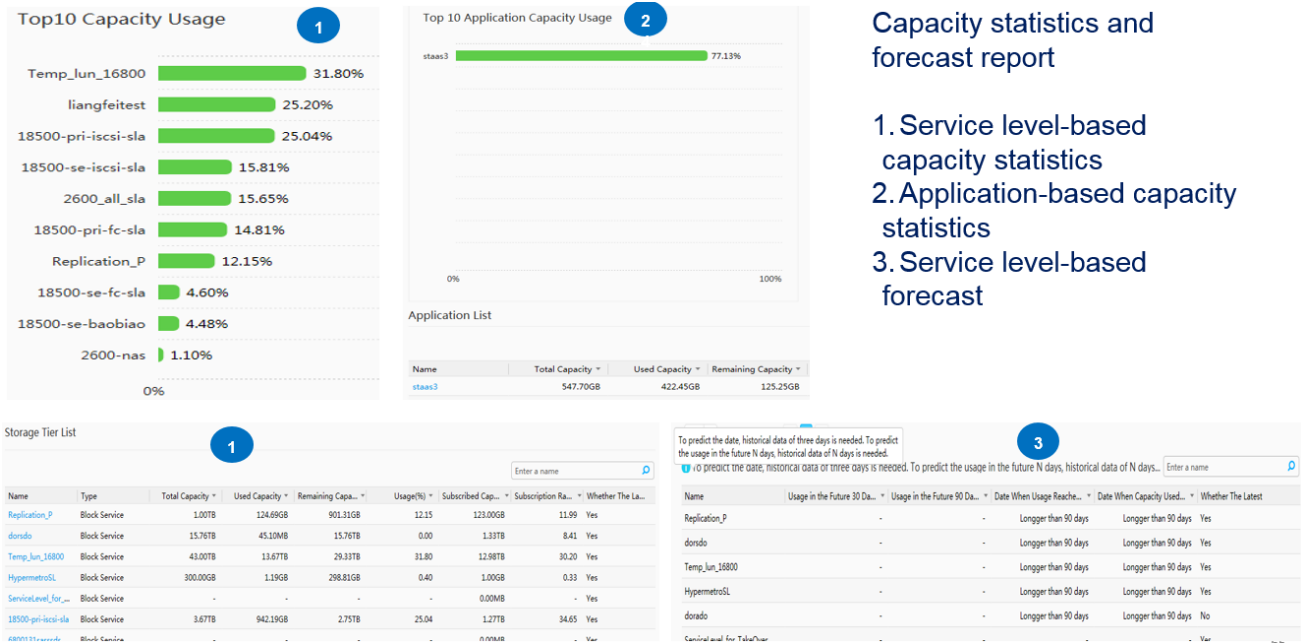
6.15 Storage Service O&M and Monitoring

6.15.1 Introduction

To meet the requirements on O&M and monitoring reports in STaaS solutions, OceanStor DJ interworks with eSight to provide a unified portal for users to use the service provisioning capabilities of OceanStor DJ and the monitoring and management functions of eSight. In this way, administrators can perform operations on a single GUI. The application scenarios of storage service monitoring and report generation can be classified into two categories: administrator scenarios and tenant scenarios.

- Administrator scenarios:
 - Interworking between OceanStor DJ and eSight. Capacity statistics based on different dimensions are supported.

Figure 6-41 Example capacity statistics and forecast report



Capacity statistics and forecast report

1. Service level-based capacity statistics
2. Application-based capacity statistics
3. Service level-based forecast

- Alarm viewing, notification, clearing, and export. Supported alarms are as follows:

Table 6-3 Supported alarms

Alarm Type	Alarm Source
Capacity threshold alarms of storage service levels	OceanStor DJ
Remote link fault alarms	Storage array
Quota threshold alarms of tenants	OceanStor DJ
Capacity threshold alarms of physical pools	Storage array
Capacity threshold alarms of volumes	eSight
Capacity threshold alarms of file shares	Storage array
OceanStor DJ alarms	OceanStor DJ
eSight alarms	eSight
Array alarms	Storage array

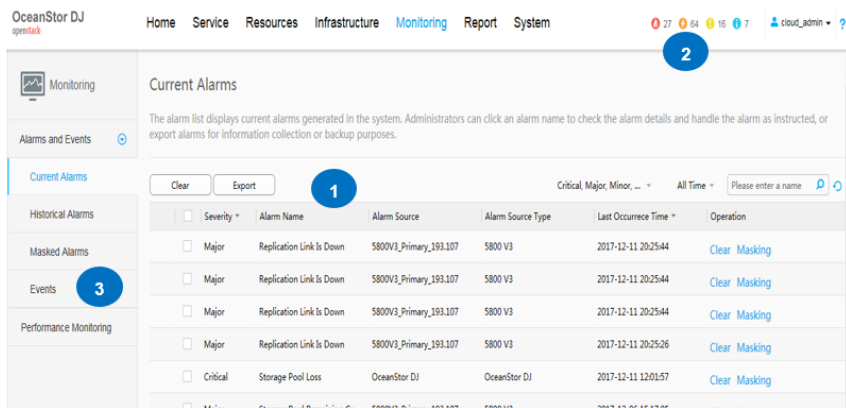
- Event viewing and export. Supported events are as follows:

Table 6-4 Supported events

Event Type	Event Source
Performance threshold events of volumes	OceanStor DJ
Performance threshold events of block storage service levels	OceanStor DJ

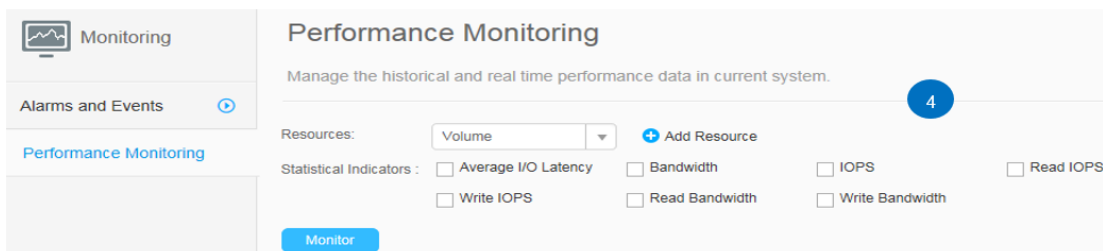
Figure 6-42 shows an example alarm report.

Figure 6-42 Example alarm report



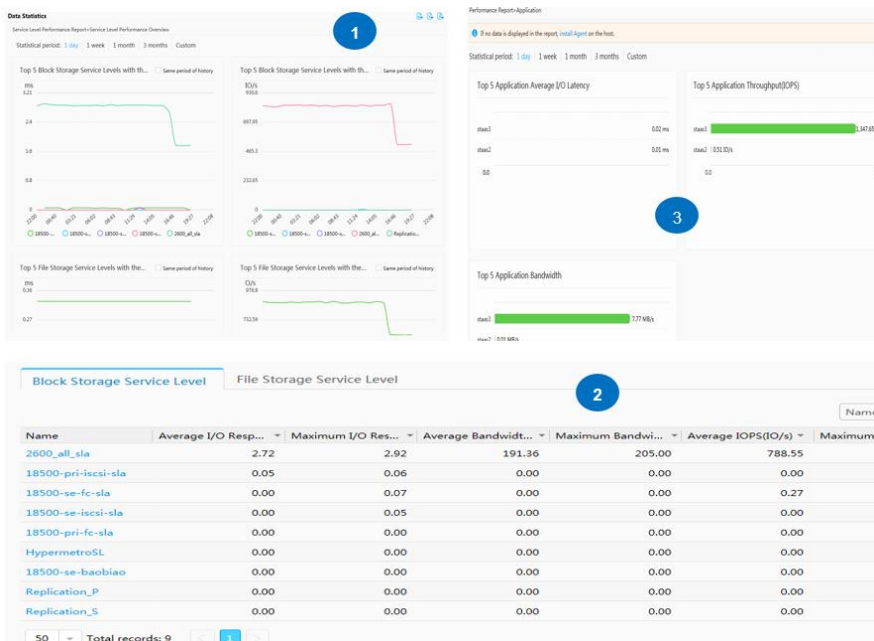
Alarm report

1. Supports alarms based on different dimensions.
2. Filters, queries, and clears alarms.
3. Views historical events.
4. Monitors performance.



- Performance monitoring: Administrators can view real-time and historical monitoring data of all volumes and file shares and export the data.

Figure 6-43 Example performance monitoring report



Performance monitoring report

1. Based on time windows
2. Based on service levels
3. Based on applications

- Report generation: Administrators can view the capacity of storage service level-centered resources and capacity usage of resources, including storage service levels, service systems, and storage arrays.
- Trend prediction: Administrators can view the prediction on capacity usage of all resources, including storage service levels, service systems, and storage arrays.
- Topology: Administrators can view topologies of service systems and host. These topologies are provided to present resource users with end-to-end mappings.
- Chargeback feature: Administrators can generate and export reports based on hosts, volumes, devices, or service levels to meet customers' chargeback requirements.

Figure 6-44 Example chargeback report

Name	Status	Mode	Start Time	End Time	Operation
2017-11-23 09:59:57_manual	Completed	Manual	2017-11-23 09:59:57 UTC+08...	2017-11-23 10:06:39 UTC+08...	Download
2017-11-20 21:13:43_manual	Completed	Manual	2017-11-20 21:13:43 UTC+08...	2017-11-20 21:22:15 UTC+08...	Download
2017-11-20 19:37:37_manual	Completed	Manual	2017-11-20 19:37:37 UTC+08...	2017-11-20 19:46:23 UTC+08...	Download
2017-11-20 19:26:55_manual	Completed	Manual	2017-11-20 19:26:55 UTC+08...	2017-11-20 19:36:18 UTC+08...	Download
2017-11-20 15:14:20_manual	Completed	Manual	2017-11-20 15:14:20 UTC+08...	2017-11-20 15:16:46 UTC+08...	Download

- Chargeback report
1. Can be automatically generated.
 2. Can be downloaded.
 3. Contains detailed resource information.
 4. Chargeback data can be used for charging.

Project	Customer	Hostname	WWN	OS	OS Version	Quantity (wn)	Hostgroup	Total (GB)	Used (GB)	Location
staas1		windows2012r2	21000024ff53 21000024ff53	Windows		2	HostGroup001	700	126.75	
staas1		windows2012r2	21000024ff53 21000024ff53	Windows		2	duankouteststaas	0	0	
staas1		windows2012r2	21000024ff53 21000024ff53	Windows		2	staas	0	0	
staas1		windows2012r2	21000024ff53 21000024ff53	Windows		2	zhanglichen	0	0	
staas2		redhat6.5	21000024ff05 21000024ff05	RedHat	6.5	2				
staas3		8.46.191.222	21000024ff7e 21000024ff7e	VMware ESXi	6.0.0	2				
staas3		Attach144	iqn.1996.21000024ff5b 21000024ff5b 21000024ff8f 21000024ff8f	SUSE	11	5	Attach144	19	0.115	
staas3		Attach144	iqn.1996.21000024ff5b 21000024ff5b 21000024ff8f 21000024ff8f	SUSE	11	5	HG Attach144	0	0	
staas3		Attach144	iqn.1996.21000024ff5b 21000024ff5b 21000024ff8f 21000024ff8f	SUSE	11	5	Attach144	1	1.188	
staas3		Attach144	iqn.1996.21000024ff5b 21000024ff5b 21000024ff8f 21000024ff8f	SUSE	11	5	FC	30	30.938	
staas3		Attach144	iqn.1996.21000024ff5b 21000024ff5b 21000024ff8f 21000024ff8f	SUSE	11	5	MR ATTACH	1	1.188	

Project	Hostgroup	Storage System	ServiceLevel_for_TakeOver_Volume(Total)(GB)	Service Level_f	18500-se-	18500-se-	dorsdo(Total)(GB)	dorsdo(Used)(GB)
staas3	Attach144	18500V3_Secondary	0	0	1	1.188	0	0
staas3	Attach144	18500V3_Primary	35	2.375	0	0	0	0
staas3	Attach144	Huawei.Storage_DrdoV3.6000.115	0	0	0	0	19	0.115
staas3	FC	18500V3_Secondary	30	30.938	0	0	0	0
staas4	HG_8.46.191.152	Huawei.Storage_DrdoV3.6000.115	0	0	0	0	0	0
staas4	HG_8.46.191.152	6800V3_Secondary_194.87	0	0	0	0	0	0
staas4	HG_8.46.191.152	18500V3_Primary	0	0	0	0	0	0

- Tenant scenarios
 - Alarming viewing and clearance. Supported alarms are as follows:

Table 6-5 Supported alarms

Alarm Type	Event Source
Capacity threshold alarms of volumes	eSight
Capacity threshold alarms of files	Storage array

- Events viewing. Supported events are as follows:

Table 6-6 Supported events

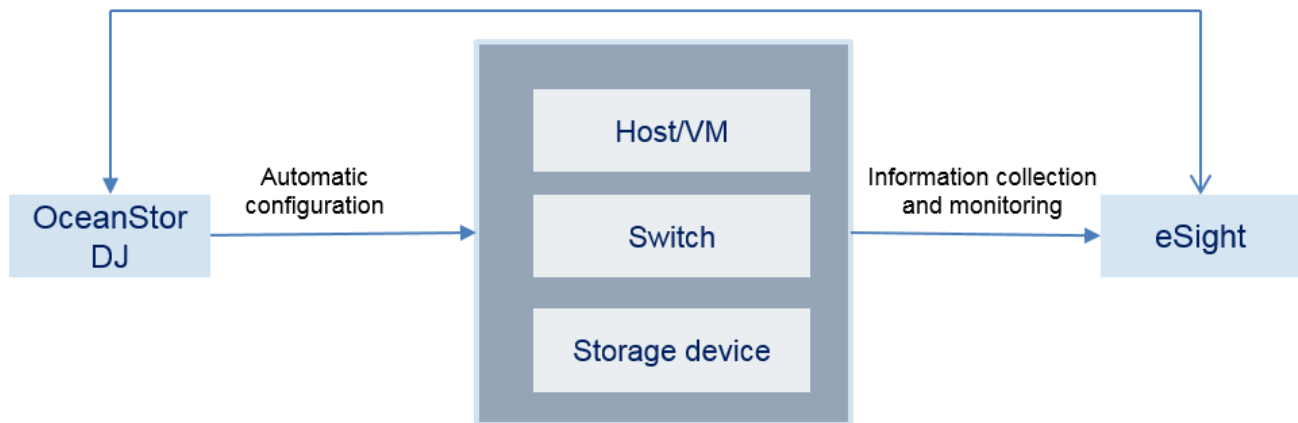
Event Type	Event Source
Performance threshold events of volumes	OceanStor DJ

- Performance monitoring: Tenant users can view real-time and historical monitoring data of all volumes and file shares.
- Trend prediction: Tenant users can view the prediction on capacity usage of all resources, including storage service levels, service systems, and arrays.

6.15.2 Technical Principles

Table 6-7 Interworking between OceanStor DJ and eSight

OceanStor DJ provides O&M and metering interfaces for eSight, which reports performance and capacity statistics information to OceanStor DJ.



OceanStor DJ collects metering data of volumes and files, data protection services on each component by using Celimeter, and provides O&M and metering interfaces for eSight. In this manner, eSight, according to the statistical data of capacity and performance, assesses the capabilities of the storage systems and storage pools and provides the assessment information to OceanStor DJ for scheduling and dynamic adjustment of QoS capabilities during service provisioning.

- Running status monitoring and fault alarm: By monitoring the running status of storage pools and their residing-storage devices, OceanStor DJ provides reports and alarm information to administrators, as well as storage path analysis and fault diagnosis.
- Capacity and performance statistics and analysis: Administrators can view statistics reports and trend analysis on capacity and performance of volumes, file systems, pools, and storage systems.
- Support for scheduling and dynamic adjustment of QoS capabilities: OceanStor DJ collects performance statistics on performance of storage devices and pools, assesses the QoS capabilities as the basis of QoS capability scheduling during service provisioning, and adjusts QoS policies for users in time based on the performance monitoring on allocated resources.

6.16 Architecture Reliability

OceanStor DJ can be deployed as a three-node cluster and employs a distributed architecture. The same service software is deployed on the three nodes. Each node can carry services and process service requests from users.

- Node-level reliability
If any node in a distributed OceanStor DJ cluster is faulty, its services are not affected, because they are distributed to other nodes for processing. When the faulty node is recovered, it can process service requests again.

When a large number of service requests are sent to the OceanStor DJ cluster, the cluster automatically distributes the requests among the nodes through HAProxy to achieve load balancing.

- **Server-level reliability**

All service processes of OceanStor DJ are distributed with multiple instances (active-active) on each node. A fault of a single process has no impact on services.

OceanStor DJ supports the software watchdog function. With this function, the system monitoring process on each node monitors the status of service processes on this node. If a service process is faulty, the system monitoring process automatically restarts it.

With the decentralized architecture, any node or service fault (man-made or mechanical fault) does not have any impact on system availability. With load balancing among the nodes, the impact of an access node or service fault on system availability can be reduced.

6.17 Network Reliability

For the networking diagram, see section 5.4.2 "Networking of Three-Node Cluster Deployment."

OceanStor DJ consists of the management plane, tenant plane, and internal plane.

Each plane uses dual-plane networking. Each node on each plane provides two ports to interconnect with two external switches. The failure of any port or switch on any plane does not cause the node or the whole system to become unavailable.

Nodes in the OceanStor DJ cluster communicate with one another through the internal plane. The internal plane is isolated from the management plane and tenant plane. Attacks on any plane do not affect other planes.



NOTE

Customers need to deploy dual-plane and isolation networks according to the OceanStor DJ networking guide.

6.18 Data Reliability

OceanStor DJ uses Gauss databases, which work in active/standby mode. Data is synchronized between the active and standby databases. Therefore, there are two copies of all data.

Data backup and restoration

OceanStor DJ supports manual and automatic data backup. By default, data is backed up locally. If a third-party FTP/NFS backup server is configured, data can also be backed up to the third-party server. The FTP server supports connectivity tests.

If the number of backup files does not exceed the maximum number allowed, new backup files do not replace old ones. When the number of backup files exceeds the maximum number, OceanStor DJ automatically deletes the earliest backup files.

After an important operation (such as an important data modification), an exception may occur or the operation may not achieve the expected results. In this case, users can perform a rollback during which backup data can be used for restoration.

In addition, OceanStor DJ can be reinstalled after a system breakdown. After the reinstallation, backup data can be used for restoration (in this scenario, the system data is restored to that at the backup time).

7 Serviceability

7.1 Installation and Deployment

The three-node OceanStor DJ cluster supports the physical machine and virtual machine environment as well as one-click installation.

Concurrent one-click installation is supported for the three nodes in the cluster. To install the three nodes, users only need to enter required data in the installation planning tool to generate an installation configuration file. During the installation, no manual intervention is required.

OceanStor DJ can be deployed using OceanStor Toolkit.

7.2 Alarm Management

OceanStor DJ supports alarm management. It automatically reports detected faults and rectifies them. If a fault cannot be automatically rectified, users can view uncleared alarm information and clear it by performing recommended actions. If an alarm has been handled, users can manually clear it.

OceanStor DJ supports northbound SNMP and can report alarms to a third-party alarm management platform. SNMP supports connectivity tests.

OceanStor DJ supports alarm exporting. Exported alarms can be easily viewed and analyzed for fault rectification.

OceanStor DJ can manage only alarms of its own but cannot manage alarms of storage devices. Alarms of storage devices are managed by eSight or a third-party alarm management platform.

7.3 Maintenance Modes

OceanStor DJ supports system maintenance on the GUI or CLI. On the CLI, users can configure system parameters and view the running status and details of all nodes and services in the system. When exceptions occur, users can restart services or perform switchovers (for active/standby services such as GaussDB) to quickly recover services.

7.4 Health Check

To ensure stable long-term operation of the OceanStor DJ system, maintenance engineers can use a health check tool (OceanStor Toolkit) to periodically check the system, find faults based on check reports, and rectify the faults by performing recommended actions.

7.5 Information Collection

OceanStor DJ allows users to use an information collection tool (OceanStor Toolkit) to export logs. By viewing and analyzing the exported logs, users can understand the system running status and rectify potential faults to ensure normal system operation.

7.6 Run Logs and Operation Logs

OceanStor DJ supports run logs and operation logs.

Run logs record the real-time running status of processes in the system. When the run log file reaches a specified size, it is compressed and backed up. The backup file is used to track historical process execution information.

Operation logs record user operations in the system. When operation logs accumulate to a specified count, they are exported and saved in a local file with one or more copies. The local file is used to track historical user operation information.

Run logs and operation logs can be backed up to a third-party FTP backup server.

7.7 NTP Time Synchronization

OceanStor DJ automatically synchronizes system time among the three nodes. It can also use an external NTP server as the clock source for time synchronization.

8 Acronyms and Abbreviations

Acronym or Abbreviation	Full Spelling
AZ	availability zone
DC2	Distributed Cloud Data Center
DPaaS	Data Protection as a Service
DPS	Data Protection Service
EVS	Elastic Volume Service
OBS	Object Storage Service
OC	Operation Center (DC2)
QoS	Quality of Service
SFS	Scalable File Service
SC	Service Center (DC2)
SL	service level
VBS	Volume Backup Service
XaaS	X as a service