Dedicated Distributed Storage Service

User Guide (Paris Region)

Issue	01
Date	2018-07-30

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.

Contents

1 Introduction

1.1 What Is Dedicated Storage Service?

Dedicated Storage Service (DSS) provides you with dedicated storage pools which are physically isolated from other pools to ensure high security. With data redundancy and cache acceleration technologies, DSS delivers highly reliable, durable, low-latency, and stable storage resources. By flexibly interconnecting with various compute services, such as Dedicated Cloud (DeC), Elastic Cloud Server (ECS) and Bare Metal Server (BMS), DSS is suitable for different scenarios, including high performance computing (HPC), online analytical processing (OLAP), and mixed loads.

Function Characteristics

- Rich specifications
  - High I/O: Suitable for scenarios that require high performance, high read/write speed, and real-time data storage.
  - Ultra-high I/O: Excellent for read/write-intensive scenarios that require extremely high performance and read/write speed, and low latency.
- Elastic scalability
  - On-demand capacity improves resource utilization.
  - Linear performance increase can be achieved with capacity expansion.
- Security and reliability
  - Distributed storage with three data replicas ensures 99.9999999% durability.
  - System disks and data disks support data encryption with zero application awareness.
- Backup and restoration
  - Backups can be created for a DSS disk, and the backup data can be used to restore the disk data, maximizing data security and correctness and ensuring service security.

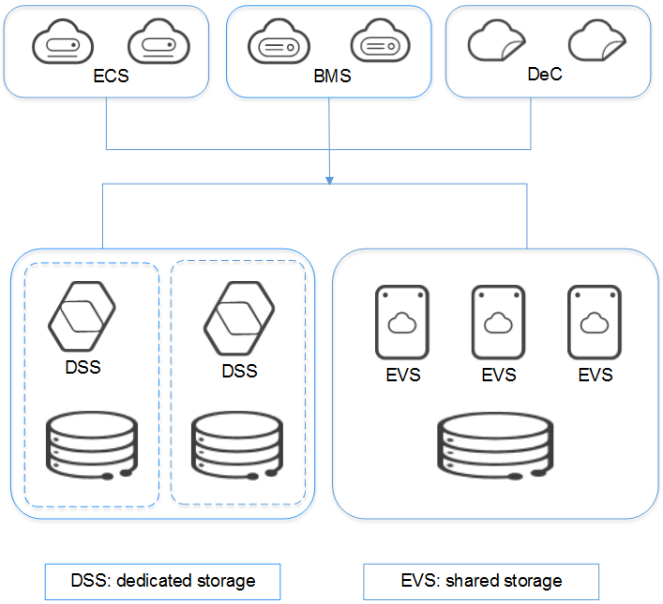
Differences Between DSS and EVS

Table 1-1 Differences between DSS and EVS

Service	Overall Introduction	Storage Category	Typical Application Scenarios	Performance
DSS	DSS provides exclusive physical storage resources for users. The storage pools are physically isolated, and data durability reaches 99.9999999%. Multiple types of compute services, including DeC, ECS and BMS, can be interconnected with DSS at the same time. DSS has abundant features to guarantee data security and reliability.	Dedicated storage pools, which means that storage pools are physically isolated and resources are exclusively used.	• Interconnection with compute services, such as ECS and BMS, in a dedicated cloud. • Interconnection with compute services, such as ECS and BMS, in a non-dedicated cloud. • Mixed load. DSS supports hybrid deployment of HPC, database, email, OA, and web applications. • High-performance computing • OLAP applications	• High I/O storage pool: The initial specification is 13.6 TB, which can be expanded to a maximum of 435.2 TB in 13.6 TB increments. The maximum IOPS is 1500 IOPS/TB. • Ultra-high I/O storage pool: The initial specification is 7.225 TB, which can be expanded to a maximum of 289 TB in 7.225 TB increments. The maximum IOPS is 8000 IOPS/TB.

Service	Overall Introduction	Storage Category	Typical Application Scenarios	Performance
EVS	Elastic Volume Service (EVS) provides scalable block storage that features high reliability, high performance, and rich specifications for servers.	Shared storage pools	• Enterprise office applications • Development and testing • Enterprise applications, including SAP, Microsoft Exchange, and Microsoft SharePoint • Distributed file systems • Various databases, including MongoDB, Oracle, SQL Server, MySQL, and PostgreSQL	EVS disks start at 10 GB and can be expanded as required in 1 GB increments to a maximum of 32 TB.

Figure 1-1 Differences between DSS and EVS



1.2 Region and AZ

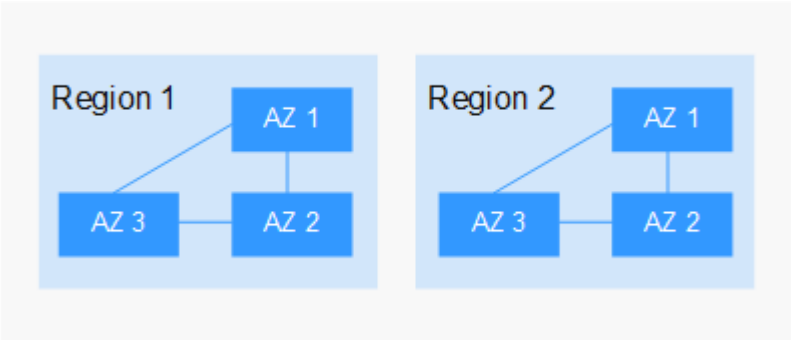
Concept

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

- A region is a physical data center, which is completely isolated to improve fault tolerance and stability. The region that is selected during resource creation cannot be changed after the resource is created.
- An AZ is a physical location where resources use independent power supplies and networks. A region contains one or more AZs that are physically isolated but interconnected through internal networks. Because AZs are isolated from each other, any fault that occurs in one AZ will not affect others.

Figure 1-2 shows the relationship between regions and AZs.

Figure 1-2 Regions and AZs



Selecting a Region

Select a region closest to your target users for lower network latency and quick access.

Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For lower network latency, deploy resources in the same AZ.

Regions and Endpoints

Before you use an API to call resources, specify its region and endpoint. For more details, see **Regions and Endpoints**.

1.3 Storage Pool Types

DSS provides two types of storage pools, which differ in I/O performance and price. You can select the storage pool type based on your service requirements.

The disk type must be consistent with the storage pool type you selected.

Application Scenarios

- High I/O storage pool supports only high I/O disks. It can deliver a maximum of 1500 IOPS per TB and a minimum of 6 ms read/write latency. This type of storage pools is designed for mainstream high-performance, high-reliability application scenarios, such as enterprise applications, large-scale development and testing, and web server logs.
- Ultra-high I/O storage pool supports only ultra-high I/O disks. It can deliver a maximum of 8000 IOPS per TB and a minimum of 1 ms read/write latency. This type of storage pools is perfect for read/write-intensive application scenarios. For example, the distributed file systems in the HPC scenarios or NoSQL and relational databases in I/O-intensive scenarios.

Performance

Key metrics of the storage pool performance include read/write I/O latency, IOPS, and throughput.

- IOPS: Number of read/write operations performed per second
- Throughput: Amount of data successfully transmitted per second, that is, the amount of data read from and written to the pool
- Read/write I/O latency: Minimum interval between two consecutive read/write operations

Table 1-2 Storage pool performance

Metric	High I/O	Ultra-high I/O
IOPS	1500 IOPS/TB	8000 IOPS/TB
Read/write I/O latency	6 ms to 10 ms	1 ms to 3 ms

1.4 Storage Pool Capacity Description

Table 1-3 Storage pool capacity description

Type	Description
Requested Capacity	Specifies the capacity of the storage pool that you apply for.

Type	Description
Raw Capacity	Specifies the raw capacity of the storage pool that you apply for. The requested capacity of a storage pool is no less than 85% of its raw capacity.
Total Available Capacity	Specifies the total available capacity of a storage pool.
Allocated Capacity	Specifies the storage pool capacity that has been allocated. Includes the capacity allocated to: • Volumes of VMs, bare metal servers, and containers • Advanced services such as SFS Turbo and RDS • Snapshots created during backup creation
Used Capacity	Specifies the storage pool physical capacity that has been used. Includes the capacity already used by: • Volumes of VMs, bare metal servers, and containers • Advanced services such as SFS Turbo and RDS • Snapshots created during backup creation

Table 1-4 Storage pool capacity calculation example

Parameter	Capacity
Requested Capacity	27.2 TB
Raw Capacity	32 T
Total Available Capacity	27.2*1024 G=27852 G
Allocated Capacity	7330 GB
Used Capacity	432 GB

1.5 What Are DSS Disks?

DSS disks are essentially dedicated EVS disks, which can be used as scalable block storage for servers. With high reliability, high performance, and rich specifications,

DSS disks can be used for distributed file systems, development and testing environments, data warehouse applications, and HPC scenarios to meet diverse service requirements. Servers that DSS support include Elastic Cloud Servers (ECSs) and Bare Metal Servers (BMSs).

DSS disks are sometimes just referred to as disks in this document.

1.6 DSS Three-Copies of Data Mechanism

What Is the Three-Copies of Data Mechanism?

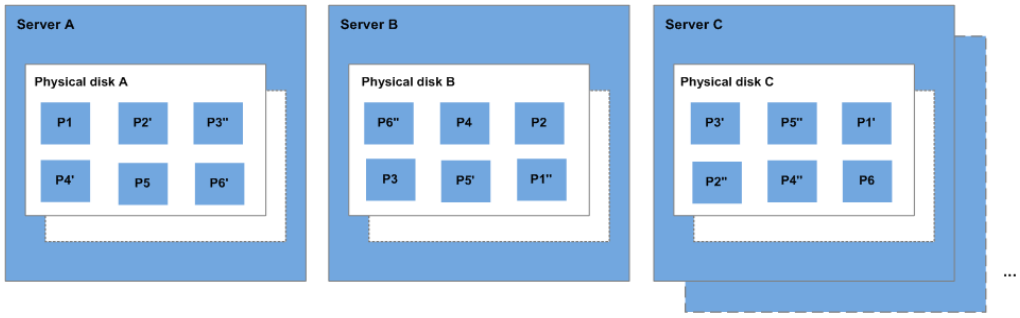
The backend storage system of DSS employs the three-copies of data mechanism to guarantee data reliability. In this mechanism, one piece of data is by default divided into multiple 1 MB data blocks. Each data block is saved in three copies, and these copies are stored on different nodes in the system according to the distributed algorithms.

The three copies of data mechanism has the following characteristics:

- The storage system saves the data copies on different disks of different servers, ensuring that services are not interrupted in case of a single physical device failure.
- The storage system guarantees tight consistency between the data copies.

For example, for data block P1 on physical disk A of server A, the storage system backs up its data to P1'' on physical disk B of server B and to P1' on physical disk C of server C. Data blocks P1, P1', and P1'' are the three copies of the same data block. If physical disk A where P1 resides is faulty, P1' and P1'' can continue providing storage services, ensuring service continuity.

Figure 1-3 Three-copies of data mechanism



How Does the Three-Copies of Data Mechanism Keep Data Consistency?

Data consistency includes the following two aspects: When an application writes a piece of data to the system, the three copies of the data in the storage system must be consistent. When any of the three copies is read by the application later, the data on this copy is consistent with the data previously written to it.

The three copies of data mechanism keeps data consistency in the following ways:

- Data is simultaneously written to the three copies of the data.

  When an application writes data, the storage system writes it to the three copies of the data simultaneously. In addition, the system returns the write

success response to the application only after the data has been written to all of its copies.

- Storage system automatically restores the damaged copy in case of a data read failure.

  When an application fails to read data, the system automatically identifies the failure cause. If the data cannot be read from a physical disk sector, the system reads the data from another copy of the data on another node and writes it back to the original disk sector. This mechanism ensures the number of data copies and data consistency among data copies.

How Does the Three-Copies of Data Mechanism Rapidly Rebuild Data?

Each physical disk in the storage system stores multiple data blocks, whose copies are scattered on the nodes in the system according to certain distribution rules. When a physical server or disk fault is detected, the storage system automatically rebuilds the data. Since the copies of data blocks are scattered on different nodes, the storage system will start the data rebuild on multiple nodes simultaneously during a data restore, with only a small amount of data on each node. In this way, the system eliminates the potential performance bottlenecks that may occur when a large amount of data needs to be rebuilt on a single node, and therefore minimizes the adverse impacts exerted on upper-layer applications.

Figure 1-4 shows the data rebuild process.

Figure 1-4 Data rebuild process

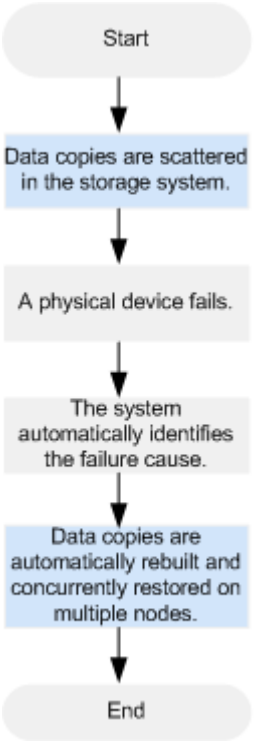
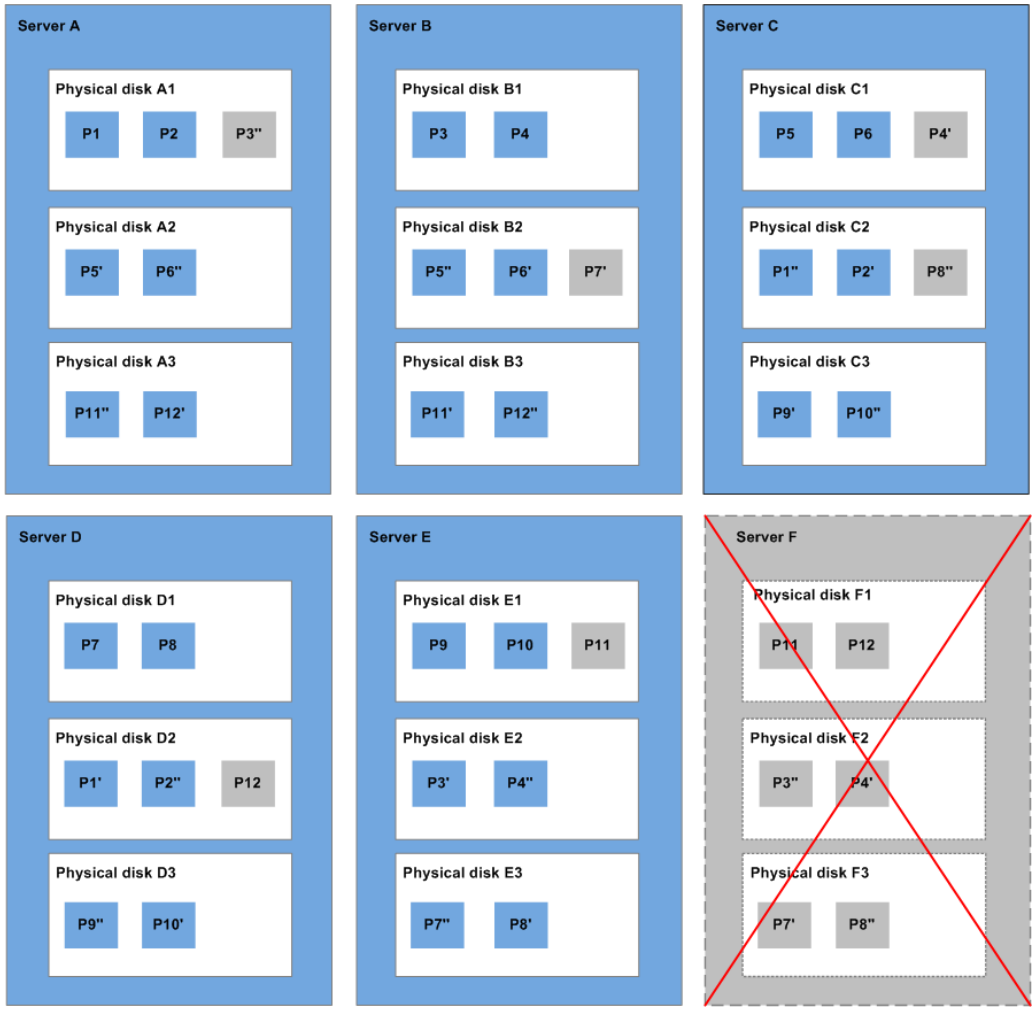


Figure 1-5 shows the data rebuild principle. For example, if physical disks on server F are faulty, the data blocks on these physical disks will be rebuilt on the physical disks of other servers.

Figure 1-5 Data rebuild principle



What Are the Differences Between Three-Copies of Data and Disk Backup

The three-copies of data mechanism improves the reliability of the data stored on DSS disks. It is used to tackle data loss or inconsistency caused by physical device faults.

Whereas, disk backup is used to prevent data loss or inconsistency caused by misoperation, viruses, or hacker attacks. Therefore, you are advised to create backups to back up the DSS disk data on a timely basis.

1.7 Device Types and Usage Instructions

What Device Types Are Available?

There are two EVS device types: Virtual Block Device (VBD) and Small Computer System Interface (SCSI).

- VBD is the default EVS disk device type. VBD EVS disks support only basic read/write SCSI commands.

- SCSI EVS disks support transparent SCSI command transmission and allow the server OS to directly access the underlying storage media. Besides basic read/write SCSI commands, SCSI disks support advanced SCSI commands.

Common Application Scenarios and Usage Instructions of SCSI EVS Disks

- SCSI EVS disks: BMSs support only SCSI EVS disks.
- Shared SCSI EVS disks: Shared SCSI EVS disks must be used together with a distributed file system or cluster software. Because most cluster applications, such as Windows MSCS, Veritas VCS, and Veritas CFS, require the usage of SCSI reservations, you are advised to use shared EVS disks with SCSI.

  SCSI reservations take effect only when shared SCSI EVS disks are attached to ECSs in the same ECS group. For more information about shared EVS disks, see **Shared Disks and Usage Instructions**.

Do I Need to Install a Driver for SCSI EVS Disks?

To use SCSI EVS disks, you need to install a driver for certain server OSs.

- BMS

  Both the Windows and Linux images for BMSs are pre-installed with the required SDI card driver. Therefore, no driver needs to be installed.

- KVM ECS

  You are advised to use SCSI EVS disks with KVM ECSs. Linux images and Windows images for KVM ECSs already have the required driver. Therefore, no driver needs to be installed for KVM ECSs.

  NOTE

  ECS virtualization types are categorized into KVM and Xen. For details, see **Product Introduction** > **ECS Types** in the *Elastic Cloud Server User Guide*.

- Xen ECS

  Due to driver limitations, you are advised not to use SCSI EVS disk with Xen ECSs.

  However, a few images support SCSI EVS disks on Xen ECSs. For the supported images, see **Table 1-5**.

  NOTE

  After confirming that the OS images of Xen ECSs support SCSI EVS disks, determine whether you need to install the driver:

  - Public Windows images are preinstalled with the Paravirtual SCSI (PVSCSI) driver. Therefore, no driver needs to be installed.
  - Private Windows images are not preinstalled with the PVSCSI driver. You need to download and install it explicitly.

    For details, see **(Optional) Optimizing Windows Private Images** in the *Image Management Service User Guide*.
  - Linux images are not preinstalled with the PVSCSI driver. You need to obtain the source code of the open-source Linux driver at **https://github.com/UVP-Tools/SAP-HANA-Tools**.

Table 1-5 OSs supporting SCSI EVS disks

Virtualization Type	OS	
Xen	Windows	See the Windows images listed on the **Public Images** page. Log in to the management console, choose **Image Management Service**, click the **Public Images** tab, and select **ECS image** and **Windows** from the drop-down lists, respectively.
	Linux	• SUSE Linux Enterprise Server 11 SP4 64bit (The kernel version is 3.0.101-68-default or 3.0.101-80-default.) • SUSE Linux Enterprise Server 12 64bit (The kernel version is 3.12.51-52.31-default.) • SUSE Linux Enterprise Server 12 SP1 64bit (The kernel version is 3.12.67-60.64.24-default.) • SUSE Linux Enterprise Server 12 SP2 64bit (The kernel version is 4.4.74-92.35.1-default.)

1.8 Shared Disks and Usage Instructions

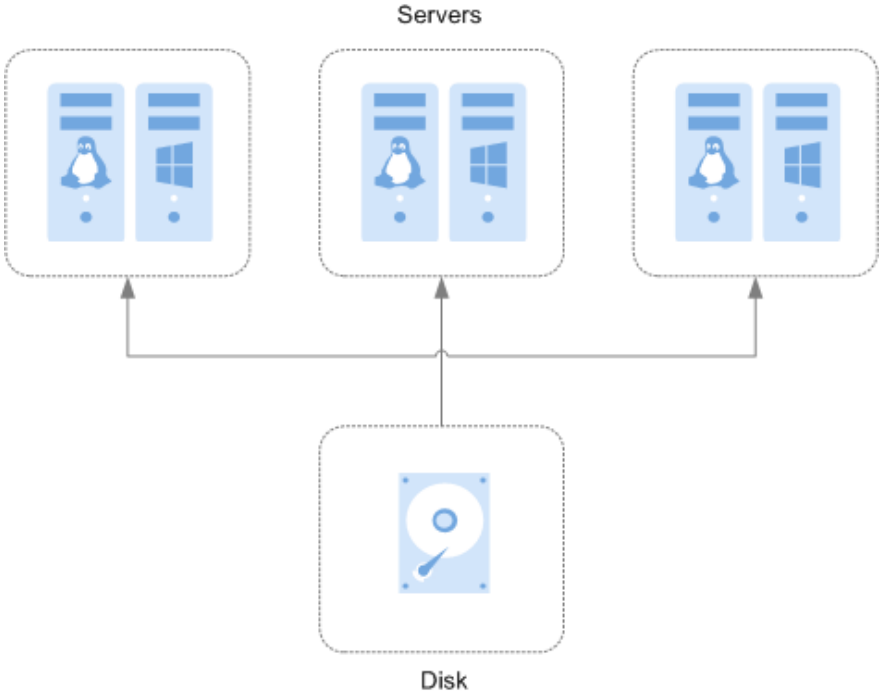
DSS disks can be classified into non-shared disks and shared disks based on whether a disk can be attached to multiple servers. A non-shared disk can only be attached to one server, whereas a shared disk can be attached to multiple servers.

What Are Shared Disks?

Shared disks are block storage devices that support concurrent read/write operations and can be attached to multiple servers. Shared disks feature multiple attachments, high concurrency, high performance, and high reliability. A shared disk can be attached to a maximum of 16 servers. **Figure 1-6** shows its application scenario.

Currently, shared disks can be used as data disks only and cannot be used as system disks.

Figure 1-6 Application scenario of shared disks



Application Scenarios and Precautions for Shared Disks

Shared disks are usually used for enterprise key applications that require cluster deployment and high availability (HA). These applications demand concurrent access to a disk from multiple servers. Before you attach a shared disk to multiple servers, the disk device type needs to be determined. The device type can be either VBD or SCSI.

Because most cluster applications, such as Windows MSCS, Veritas VCS, and Veritas CFS, require the usage of SCSI reservations, you are advised to use shared disks with SCSI. If a SCSI disk is attached to a Xen ECS for use, you must install the driver. For details, see **Device Types and Usage Instructions**.

You can create shared VBD disks or shared SCSI disks.

- Shared VBD disks: The device type of a newly created shared disk is VBD by default. Such disks can be used as virtual block storage devices, but do not support SCSI reservations. If SCSI reservations are required for your applications, create shared SCSI disks.
- Shared SCSI disks: These disks support SCSI reservations.

NOTICE

- To improve data security, you are advised to use SCSI reservations together with the anti-affinity policy of an ECS group. That said, ensure that the shared SCSI disk is only attached to ECSs in the same anti-affinity ECS group.
- If an ECS does not belong to any anti-affinity ECS group, you are advised not to attach shared SCSI disks to this ECS. Otherwise, SCSI reservations may not work properly, which may put your data at risk.

Concepts of the anti-affinity ECS group and SCSI reservations:

- The anti-affinity policy of an ECS group allows ECSs to be created on different physical servers to improve service reliability.

  For details about ECS groups, see **Managing ECS Groups** in the *Elastic Cloud Server User Guide*.
- The SCSI reservation mechanism uses a SCSI reservation command to perform SCSI reservation operations. If an ECS sends such a command to a disk, the disk is displayed as locked to other ECSs, preventing the data damage that may be caused by simultaneous read/write operations to the disk from multiple ECSs.
- ECS groups and SCSI reservations have the following relationship: A SCSI reservation on a single disk cannot differentiate multiple ECSs on the same physical host. For that reason, if multiple ECSs that use the same shared disk are running on the same physical host, SCSI reservations will not work properly. Therefore, you are advised to use SCSI reservations only on ECSs that are in the same ECS group, thus having a working anti-affinity policy.

Advantages of Shared Disks

- Multiple attachments: A shared disk can be attached to a maximum of 16 servers.
- High-performance: When multiple servers concurrently access a shared ultra-high I/O disk, random read/write IOPS can reach up to 160,000.
- High-reliability: Shared disks support both manual and automatic backup, delivering highly reliable data storage.
- Wide application scenarios: Shared disks can be used for Linux RHCS clusters where only VBD disks are needed. Whereas, they can also be used for Windows MSCS and Veritas VCS clusters that require SCSI reservations.

Specifications of Shared Disks

Key metrics of the disk performance include read/write I/O latency, IOPS, and throughput.

- IOPS: Number of read/write operations performed by a disk per second
- Throughput: Amount of data successfully transmitted by a disk per second, that is, the amount of data read from and written to a disk
- Read/write I/O latency: Minimum interval between two consecutive read/write operations of a disk

  Single-queue access latencies of different types of disks are as follows:
  - Common I/O: 10 ms to 15 ms
  - Ultra-high I/O: 1 ms to 3 ms

Table 1-6 Disk performance data

Metric	Common I/O	Ultra-high I/O
IOPS per GB/disk	1	50

Metric	Common I/O	Ultra-high I/O
Min. IOPS/disk	100	100
Max. IOPS/disk	1,000	20,000
IOPS Burst Limit/disk	1,000	10,000
Max. Throughput	40 MB/s	320 MB/s
Number of servers that can be attached to	A shared disk can be attached to a maximum of 16 servers.	

NOTE

To test the performance of a shared disk, the following requirements must be met:

- The shared disk must be attached to multiple ECSs or BMSs.
- If the shared disk is attached to multiple ECSs, these ECSs must belong to the same anti-affinity ECS group.

  If these ECSs fail to meet the anti-affinity requirement, the optimal performance of the shared disk cannot be achieved.

Data Sharing Principle and Common Usage Mistakes of Shared Disks

A shared disk is essentially the disk that can be attached to multiple servers for use, which is similar to a physical disk in that the disk can be attached to multiple physical servers, and each server can read data from and write data into any space on the disk. If the data read/write rules, such as the read/write sequence and meaning, between these servers are not defined, data read/write interference between servers or other unpredictable errors may occur.

Though shared disks are block storage devices that provide shared access for servers, shared disks do not have the cluster management capability. Therefore, you need to deploy a cluster system to manage shared disks. Common cluster management systems include Windows MSCS, Linux RHCS, Veritas VCS, and Veritas CFS.

If shared disks are not managed by a cluster system, the following issues may occur:

- Data inconsistency caused by read/write conflicts

  When a shared disk is attached to two servers (server A and server B), server A cannot recognize the disk spaces allocated to server B, vice versa. That said, a disk space allocated to server A may be already used by server B. In this case, repeated disk space allocation occurs, which leads to data errors.

  For example, a shared disk has been formatted into the ext3 file system and attached to server A and server B. Server A has written metadata into the file system in space R and space G. Then server B has written metadata into space E and space G. In this case, the data written into space G by server A will be replaced. When the metadata in space G is read, an error will occur.
- Data inconsistency caused by data caching

When a shared disk is attached to two servers (server A and server B), the application on server A has read the data in space R and space G, then cached the data. At that time, other processes and threads on server A would then read this data directly from the cache. At the same time, if the application on server B has modified the data in space R and space G, the application on server A cannot detect this data change and still reads this data from the cache. As a result, the user cannot view the modified data on server A.

For example, a shared disk has been formatted into the ext3 file system and attached to server A and server B. Both servers have cached the metadata in the file system. Then server A has created a new file (file F) on the shared disk, but server B cannot detect this modification and still reads data from its cached data. As a result, the user cannot view file F on server B.

Before you attach a shared disk to multiple servers, the disk device type needs to be determined. The device type can be either VBD or SCSI. Shared SCSI disks support SCSI reservations. Before using SCSI reservations, you need to install a driver in the server OS and ensure that the OS image is included in the compatibility list.

1.9 Disk Encryption

What Is Disk Encryption?

In case your services require encryption for the data stored on disks, EVS provides you with the encryption function. You can encrypt new disks. Keys used by encrypted disks are provided by the Key Management Service (KMS), which is secure and convenient. Therefore, you do not need to establish and maintain the key management infrastructure.

Keys Used for Disk Encryption

The keys provided by KMS include a Default Master Key and Customer Master Keys (CMKs).

- Default Master Key: A key that is automatically created by EVS through KMS and named **evs/default**.

  The Default Master Key cannot be disabled and does not support scheduled deletion.

- CMKs: Keys created by users. You may use existing CMKs or create new CMKs to encrypt disks. For details, see **Management** > **Creating a CMK** in the *Key Management Service User Guide*.

If disks are encrypted using CMKs and a CMK is then disabled or scheduled for deletion, the disks encrypted by this CMK can no longer be read from or written to and data on these disks may never be restored. See **Table 1-7** for more information.

Table 1-7 Impact of CMK unavailability

CMK Status	Impact	How to Restore
Disabled	• If an encrypted disk is attached to a server, the disk cannot be accessed or data on the disk cannot be restored after a period of time or even permanently. If this disk is detached later, it cannot be attached again. • If an encrypted disk is not attached to any server, it cannot be attached anymore.	Enable the CMK. For details, see **Managing CMKs** > **Enabling One or More CMKs** in the *Key Management Service User Guide*.
Scheduled deletion		Cancel the scheduled deletion for the CMK. For details, see **Managing CMKs** > **Canceling the Scheduled Deletion of One or More CMKs** in the *Key Management Service User Guide*.
Deleted		Data on the disks can never be restored.

Relationships Between Encrypted Disks and Backups

The encryption function can be used for system disks, data disks, and backups. The detailed descriptions are as follows:

- The system disk encryption depends on images. For details, see the *Image Management Service User Guide*.
- The encryption attribute of an existing disk cannot be changed. You can create new disks and determine whether to encrypt the disks or not.
- When a disk is created from a backup, the encryption attribute of the new disk will be consistent with that of the backup's source disk.

Before using the encryption function, you need to grant KMS access rights to EVS. If you have the right granting permission, grant KMS access rights directly. If you do not have the permission, contact a user with the security administrator rights to add the security administrator rights for you. Then, grant KMS access rights to EVS. For details, see **Who Can Use the Encryption Feature?**

For details about how to create an encrypted disk, see **Step 2: Create a Disk**.

Who Can Use the Disk Encryption Function?

- The security administrator (having the Security Administrator rights) can grant the KMS access rights to EVS for using the disk encryption function.
- When a common user who does not have the Security Administrator rights needs to use the disk encryption function, the condition varies depending on whether the user is the first one ever in the current region or project to use this feature.
  - If the common user is the first one ever in the current region or project to use the feature, the user must contact a user having the Security

Administrator rights to grant the KMS access rights to EVS. Then, the common user can use the disk encryption function.

- If the common user is not the first one ever in the current region or project to use the feature, the common user can use the disk encryption function directly.

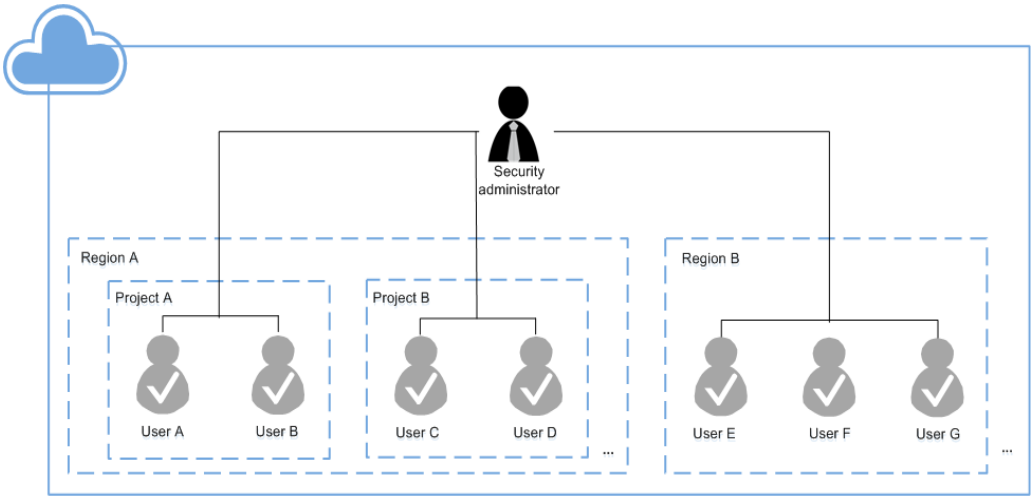
From the perspective of a tenant, as long as the KMS access rights have been granted to EVS in a region, all users in the same region can directly use the disk encryption feature.

If there are multiple projects in the current region, KMS access rights need to be granted to each project in this region.

Application Scenarios of Disk Encryption

Figure 1-7 shows the user relationships under regions and projects from the perspective of a tenant. The following example uses region B to describe the two application scenarios of the disk encryption function.

Figure 1-7 User relationships



- If the security administrator uses the encryption function for the first time ever, the operation process is as follows:

  a. Grant the KMS access rights to EVS.

  After KMS access rights have been granted, the system automatically creates a Default Master Key and names it **evs/default**. DMK can be used for disk encryption.

  NOTE

  Disk encryption relies on KMS. When the encryption function is used for the first time ever, KMS access rights need to be granted to EVS. After KMS access rights have been granted, all users in this region can use the encryption function, without requiring the KMS access rights to be granted again.

  b. Select a key.

  You can select one of the following keys:

    - DMK: **evs/default**
    - CMKs: Existing or newly created CMKs. For details, see **Creating a CMK** in the *Key Management Service User Guide*.

  After the security administrator has used the disk encryption function, all users in Region B can directly use the encryption function.

- If User E (common user) uses the encryption function for the first time ever, the operation process is as follows:

  a. When user E uses the encryption function, and the system prompts a message indicating that the KMS access rights have not been granted to EVS.

  b. Contact the security administrator to grant the KMS access rights to EVS.

  After the KMS access rights have been granted to EVS, User E as well as all users in Region B can directly use the disk encryption function and do not need to contact the security administrator to grant the KMS access rights to EVS again.

1.10 DSS and Other Services

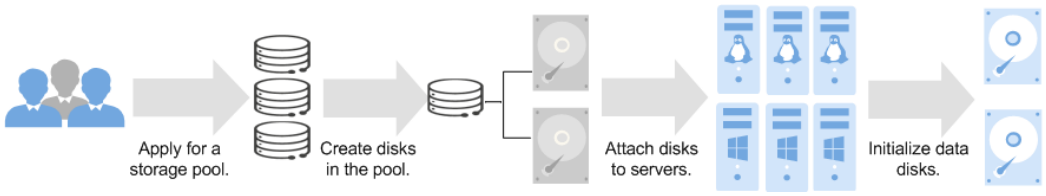
- Elastic Cloud Server (ECS): DSS disks can be attached to ECSs and used as scalable block storage devices.
- Bare Metal Server (BMS): SCSI DSS disks can be attached to BMSs and used as scalable block storage devices.
- CBR: The CBR service can be used to back up disk data to ensure the reliability and security of the server data.

2 Quick Start

2.1 Operation Procedure

Figure 2-1 shows the basic DSS operation procedure.

Figure 2-1 Basic operation procedure



1. Before using DSS, apply for a storage pool first. For more information, see **Step 1: Apply for a Storage Pool**. In a dedicated cloud, you can apply for multiple storage pools.
2. After the storage pool you requested is available, create disks in the storage pool to make use of the storage pool space. For more information, see **Step 2: Create a Disk**.
3. Attach the disks to servers. For more information, see the following topics:
   - **Attaching a Non-Shared Disk**
   - **Attaching a Shared Disk**
4. A disk cannot be used right away after being attached to a server. You must log in to the server and initialize the disk. For more information, see the following topics:
   - **Introduction to Data Disk Initialization Scenarios and Partition Styles**
   - Windows
     - **Initializing a Data Disk in Windows (Windows Server 2008)**
     - **Initializing a Data Disk in Windows (Windows Server 2016)**
     - **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**

- **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2012)**

– Linux

- **Initializing a Data Disk in Linux (fdisk)**
- **Initializing a Data Disk in Linux (parted)**
- **Initializing a Data Disk Greater Than 2 TB in Linux (parted)**

2.2 Step 1: Apply for a Storage Pool

Scenarios

DSS provides you with dedicated, physical storage resources, which can be flexibly interconnected with various compute services, such as ECS, BMS, and DeC, and are suitable to a wide-range of scenarios, including HPC, OLAP, and a mixed of loads. Before using DSS, you need to apply for a storage pool first.

This topic describes how to apply for a storage pool. In a dedicated cloud, you can apply for multiple storage pools.

Procedure

Step 1 Log in to the management console.

Step 2 Click in the upper left corner and select a region and project.

Step 3 Choose **Storage** > **Dedicated Storage Service** to switch to the **Dedicated Storage Service** page.

Step 4 Click **Apply for Storage Pool**.

Step 5 Apply for a storage pool by referring to the following information:

1. Customer Service

   To apply for a storage pool, click **here** to send an email or contact **customer service**.

2. Requirements Confirmation

   Fill out a storage pool **application form** and send it to your sales representative.

3. Resource Provisioning

   Customer service will process your application. After the storage pool is provisioned, we will inform you through your sales representative or email.

4. Resource Confirmation

   Log in to the DSS console and confirm that the storage pool you requested is available.

Table 2-1 Parameters required for enabling DSS

Parameter	Description
Domain Name	Specifies the domain name of the storage pool. For details, see **Obtaining the Domain Name**.
Storage Pool Name	Specifies the storage pool name, which can contain a maximum of 255 characters.
Region	Specifies the region where the storage pool belongs. For details, see **Obtaining the Region**.
AZ	Specifies the AZ where the storage pool belongs.
Type	Specifies the storage pool type. – High I/O – Ultra-high I/O
Capacity (TB)	Specifies the storage pool capacity.

----End

2.3 Step 2: Create a Disk

Scenarios

DSS storage capabilities are implemented through DSS disks. You need to create disks to make use of the storage resources you requested.

This topic describes how to create a disk in a DSS storage pool.

Precautions

When disks are created in a storage pool, the type of DSS disks will be the same as that of the storage pool.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Storage** > **Dedicated Storage Service** to switch to the **Dedicated Storage Service** page.

Step 3 Choose **Dedicated Storage Service** > **Disks**.

NOTE

If you have not applied for a storage pool, apply for a storage pool by referring to **Step 1: Apply for a Storage Pool**. If the storage pool has not been deployed, wait until the storage pool is deployed and then perform subsequent operations.

Step 4 On the displayed page, click **Create Disk**, or in the storage pool list, locate the storage pool in which the disk will be created and click **Create Disk** in the **Operation** column.

Step 5 Configure basic information about the disk by referring to **Table 2-2**.

Table 2-2 Parameter description

Name	Sub-Parameter	Description	Example Value
Region	-	Mandatory Specifies the region where the tenant belongs.	-
AZ	-	Mandatory Specifies the availability zone (AZ) where you want to create the disk. **NOTE** • Disks can only be attached to the servers in the same AZ. • The AZ of a disk cannot be changed after the disk has been created.	-
Disk Specifications	Disk Type	Mandatory Select a storage pool from the drop-down list. The disk type is the same as the storage pool type.	Ultra-high I/O
	Capacity (GB)	Mandatory	100GB
	Select Data Source • Create from backup	Optional • If you choose **Create from backup**, the backup data is used to create the disk. Click **Select Data Source** and choose **Create from backup**. On the displayed page, select the target backup and click **OK**. **NOTE** – One backup cannot be used for concurrent disk creation operations at the same time. For example, if you are creating disk A from a backup, this backup can be used to create another disk only after disk A has been created. – If a disk is created from a backup of a system disk, the new disk can be used as a data disk only.	• Create from backup: autobackup-001

Name	Sub-Parameter	Description	Example Value
Disk Name	-	Mandatory • If you create disks one by one, this parameter value is used as the actual disk name. The name can contain a maximum of 64 characters. • If you create disks in a batch, this parameter value is used as the prefix of disk names, and one disk name will be composed of this parameter value and a four-digit number. The name can contain a maximum of 59 characters.	For example, if you create two disks and set **volume** for **Disk Name**, the EVS disk names will be **volume-0001** and **volume-0002**.

Step 6 Click **Next**.

Step 7 In the disk list, view the disk status.

When the disk status changes to **Available**, the disk is successfully created.

----End

2.4 Step 3: Attach a Disk

2.4.1 Attaching a Non-Shared Disk

Scenarios

Independently created disks are data disks. In the disk list, the function of such disks is displayed as **Data disk**, and the status is displayed as **Available**. In this case, you need to attach the data disks to servers for use.

This topic describes how to attach a non-shared DSS disk to a server. A non-shared DSS disk can be attached to one server only.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Storage** > **Dedicated Storage Service**.

The **Dedicated Storage Service** details page is displayed.

Step 3 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 4 Locate the target disk in the list and click **Attach** in the **Operation** column.

The **Attach Disk** dialog box is displayed.

Step 5 Select the server and then select a device name from the drop-down list. Ensure that the disk and server are in the same AZ.

Return to the disk list page. The disk status is **Attaching**, indicating that the disk is being attached to the server. When the disk status changes to **In-use**, the disk is successfully attached.

Step 6 Initialize the disk.

After the disk has been attached to a server, the disk can be used only after you have initialized it. For details, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

----End

2.4.2 Attaching a Shared Disk

Scenarios

Independently created disks are data disks. In the disk list, the function of such disks is displayed as **Data disk**, and the status is displayed as **Available**. In this case, you need to attach the data disks to servers for use.

This topic describes how to attach a shared DSS disk to servers. A shared DSS disk can be attached to a maximum of 16 servers.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Storage** > **Dedicated Storage Service**.

The **Dedicated Storage Service** details page is displayed.

Step 3 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 4 Locate the target disk in the list and click **Attach** in the **Operation** column.

Shared disks support batch attachment so that you can attach a shared disk to multiple servers at a time. The left area in the **Attach Disk** dialog box shows the server list. After you select the target servers, the selected servers will be displayed in the right area.

Step 5 Select servers and select a device name from the drop-down list for each server you selected. Ensure that the disk and servers are in the same AZ.

Return to the disk list page. The disk status is **Attaching**, indicating that the disk is being attached to the servers. When the disk status changes to **In-use**, the disk is successfully attached.

NOTICE

If you simply attach a shared disk to multiple servers, files cannot be shared between the servers as shared DSS disks do not have the cluster capability. Therefore, build a shared file system or deploy a cluster management system if you need to share files between servers.

----End

2.5 Initialize a Data Disk

2.5.1 Introduction to Data Disk Initialization Scenarios and Partition Styles

Scenarios

After a disk is attached to a server, you need to log in to the server to initialize the disk, that is, format the disk. You must initialize a disk before accessing it.

- System disk

  A system disk does not require manual initialization because it is automatically created and initialized upon server creation. The default disk partition style is master boot record (MBR).

- Data disk
  - If a data disk is created along with a server, it will be automatically attached to the server.
  - If a data disk is created separately, you need to manually attach it to a server.

  In both cases, you must initialize the data disk before using it. Choose a proper disk partition style based on your service plan.

Constraints

A disk created from a data source does not need to be initialized. Such a disk contains the data of the data source in the beginning. Initializing the disk may clear the initial data on this disk.

Disk Partition Styles

Table 2-3 lists the common disk partition styles. In Linux, different disk partition styles require different partitioning tools.

Table 2-3 Disk partition styles

Disk Partition Style	Maximum Disk Capacity Supported	Maximum Number of Partitions Supported	Linux Partitioning Tool
Master Boot Record (MBR)	2 TB	• 4 primary partitions • 3 primary partitions and 1 extended partition With MBR, one may create several primary partitions and an extended partition. An extended partition must be divided into several logical partitions before use. For example, if 6 partitions need to be created, you can create the partitions in the following two ways: • 3 primary partitions and 1 extended partition, with the extended partition divided into 3 logical partitions • 1 primary partition and 1 extended partition, with the extended partition divided into 5 logical partitions	• fdisk • parted
GUID Partition Table (GPT)	18 EB 1 EB = 1048576 TB	Unlimited Disk partitions created using GPT are not categorized.	parted

NOTICE

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Partitioning Operation Guide

For a disk smaller than 2 TB, see the following sections:

- **Initializing a Data Disk in Windows (Windows Server 2008)**
- **Initializing a Data Disk in Windows (Windows Server 2016)**

- **Initializing a Data Disk in Linux (fdisk)**
- **Initializing a Data Disk in Linux (parted)**

For a disk larger than 2 TB, see the following sections:

- **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**
- **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2012)**
- **Initializing a Data Disk Greater Than 2 TB in Linux (parted)**

2.5.2 Initializing a Data Disk in Windows (Windows Server 2008)

Scenarios

This section uses Windows Server 2008 R2 Enterprise 64bit to describe how to initialize a data disk attached to a server running Windows.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. For details, see **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Procedure

Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

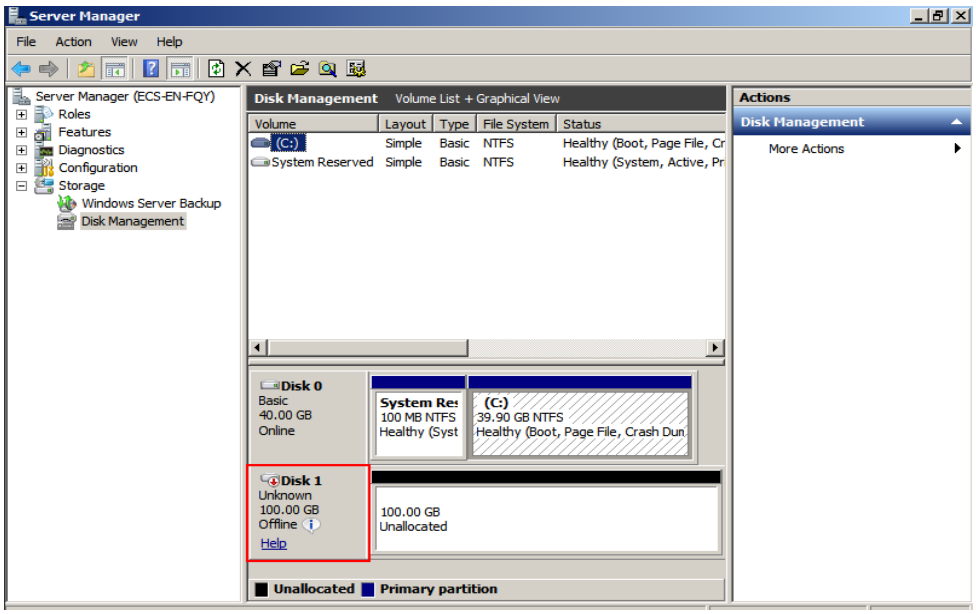
The **Server Manager** window is displayed.

Step 2 In the navigation tree, choose **Storage** > **Disk Management**.

The **Disk Management** window is displayed.

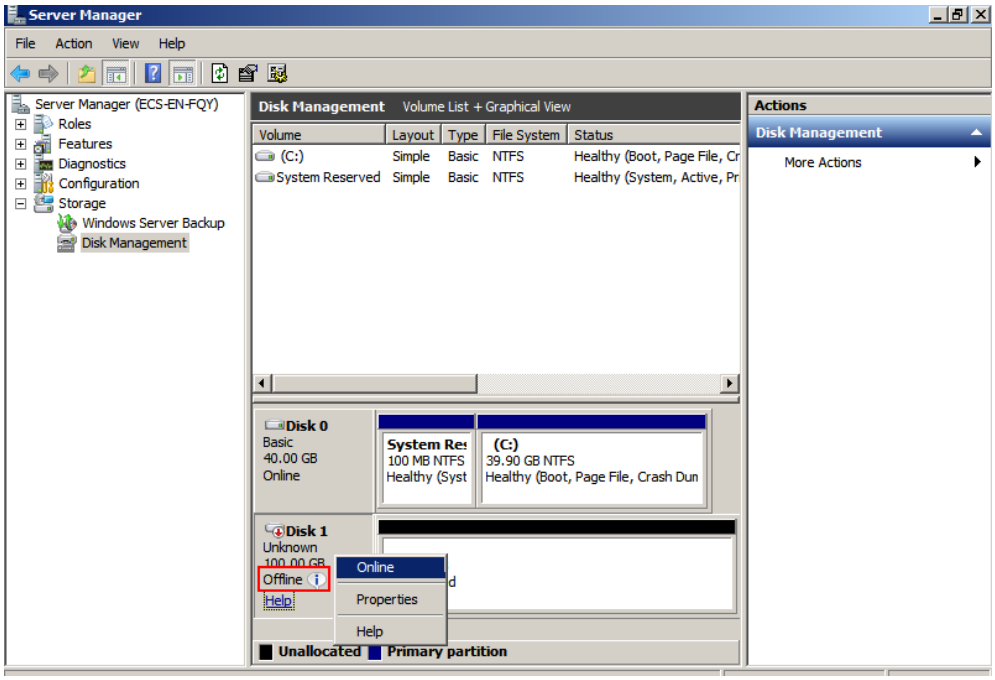
- If **Figure 2-2** is displayed, the new disk is offline. Go to **Step 3**.
- If **Figure 2-5** is displayed, the **Initialize Disk** window is prompted. Go to **Step 5**.

Figure 2-2 Disk Management



Step 3 Disks are displayed in the right pane. In the **Disk 1** area, right-click **Offline** and choose **Online** from the shortcut menu to online the disk.

Figure 2-3 Online the disk

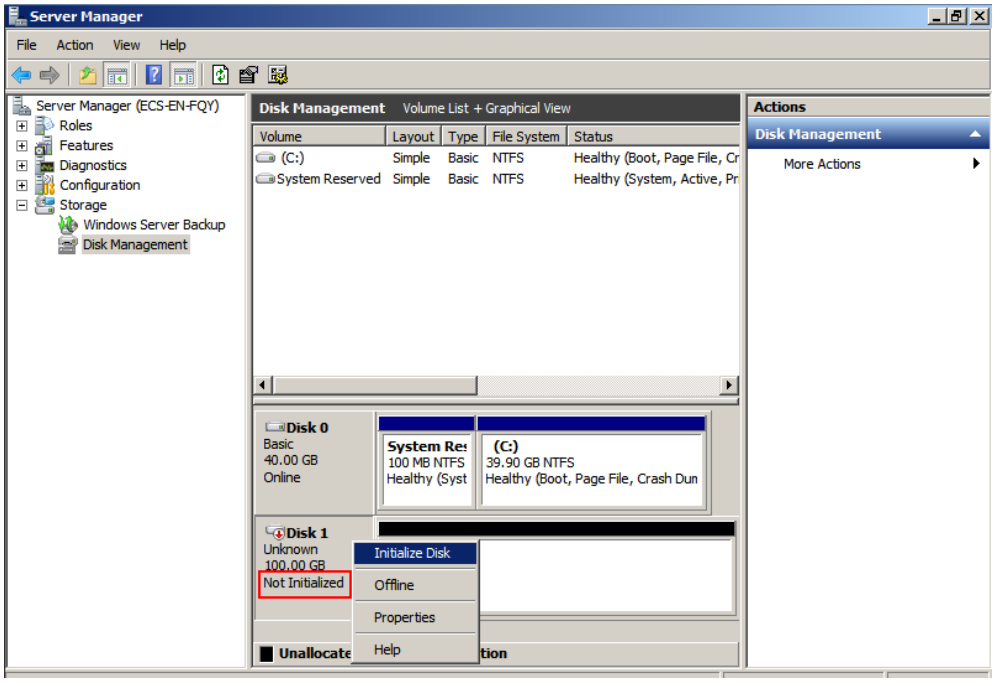


NOTE

If the disk is offline, you need to online the disk before initializing it.

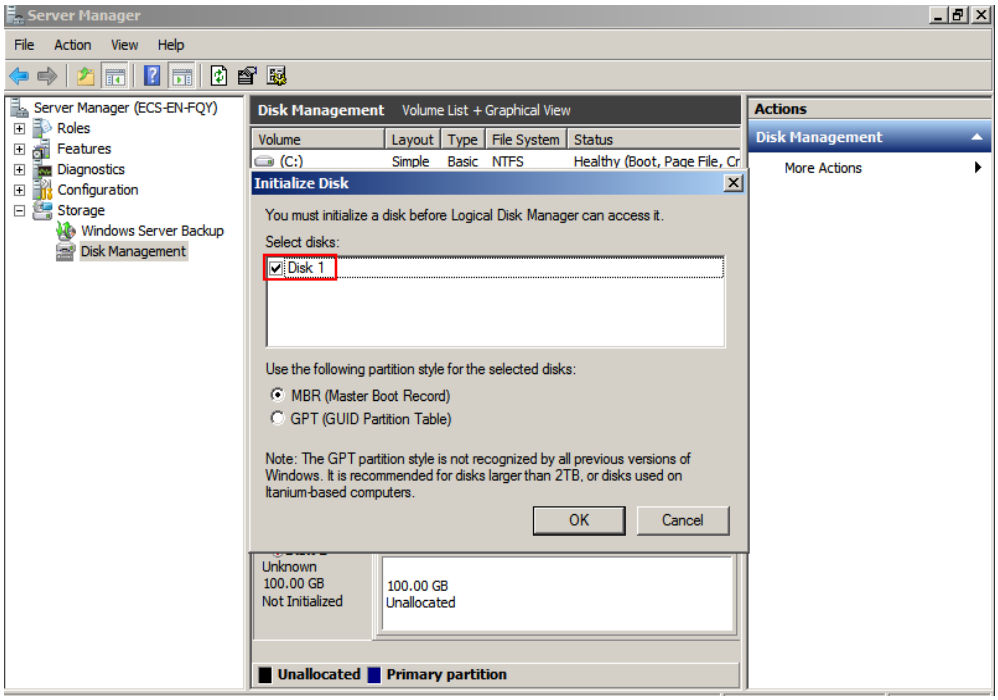
Step 4 After making the disk online, the disk status changes from **Offline** to **Not Initialized**. Right-click the disk status and choose **Initialize Disk** from the shortcut menu, as shown in **Figure 2-4**.

Figure 2-4 Initialize Disk



Step 5 In the **Initialize Disk** dialog box, select the target disk, click **MBR (Master Boot Record)** or **GPT (GUID Partition Table)**, and click **OK**, as shown in **Figure 2-5**.

Figure 2-5 Unallocated space



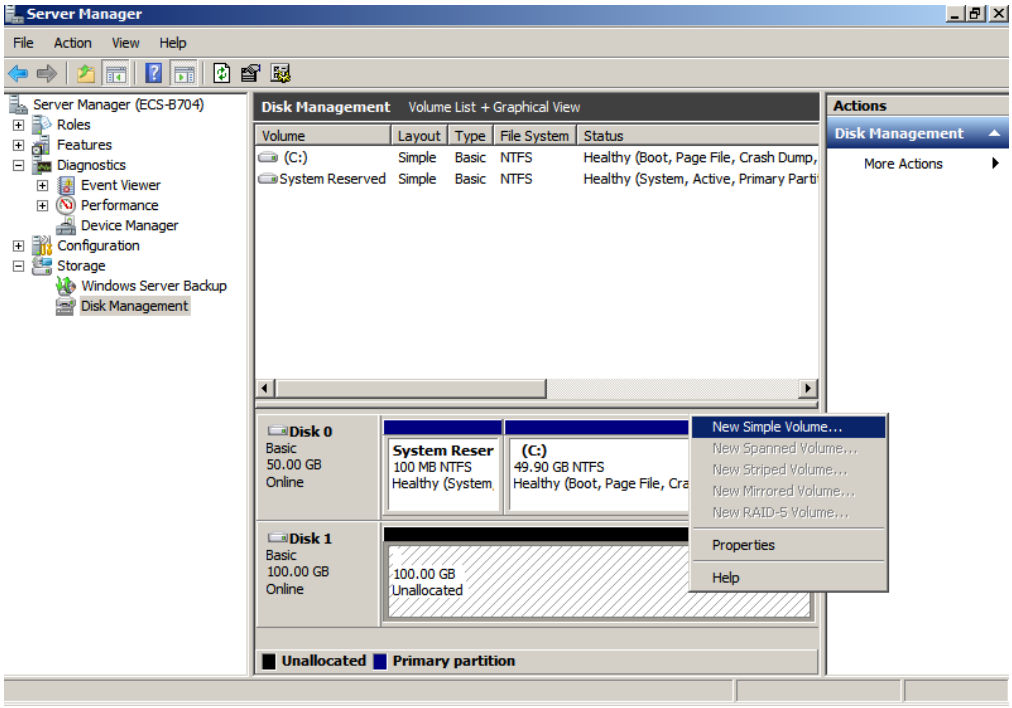
NOTICE

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

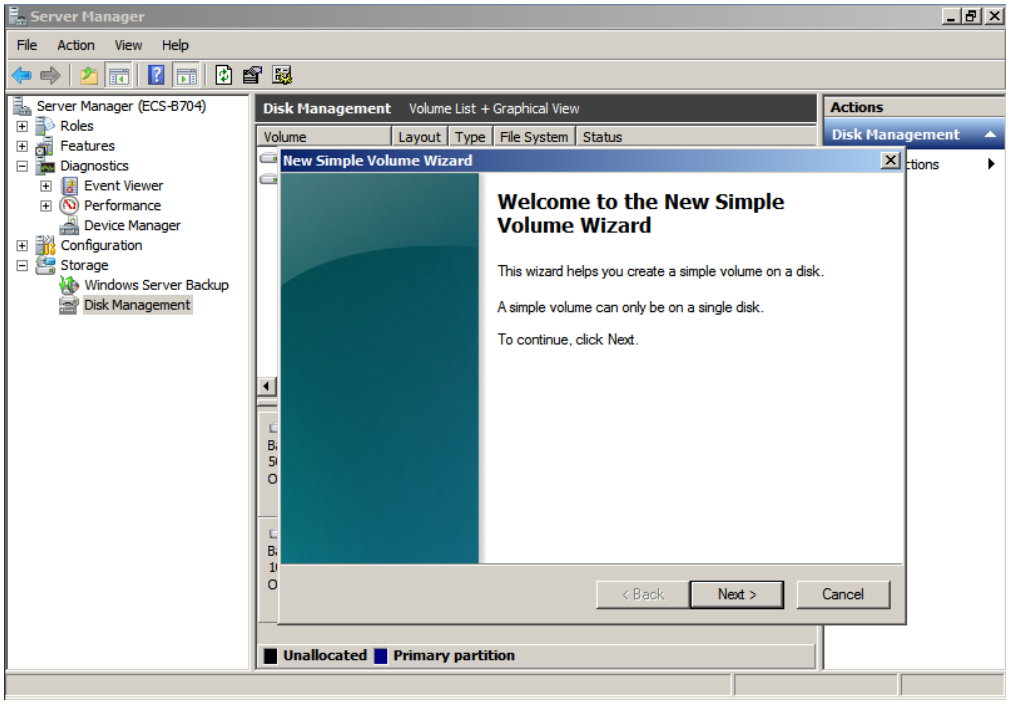
Step 6 Right-click at the unallocated space and choose **New Simple Volume** from the shortcut menu, as shown in **Figure 2-6**.

Figure 2-6 New Simple Volume



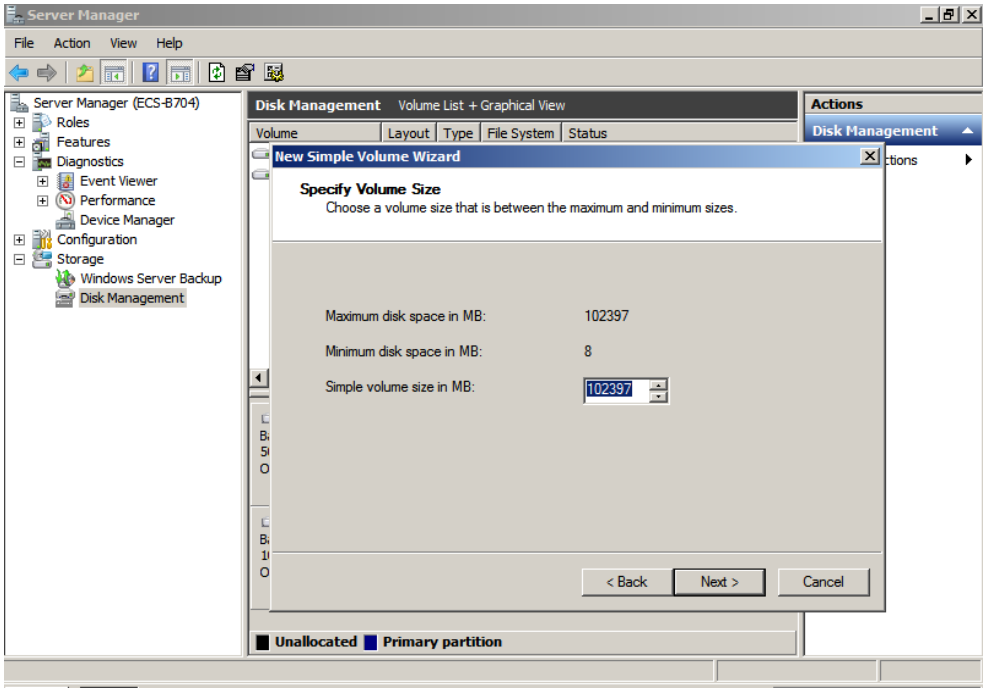
Step 7 On the displayed **New Simple Volume Wizard** window, click **Next**.

Figure 2-7 New Simple Volume Wizard



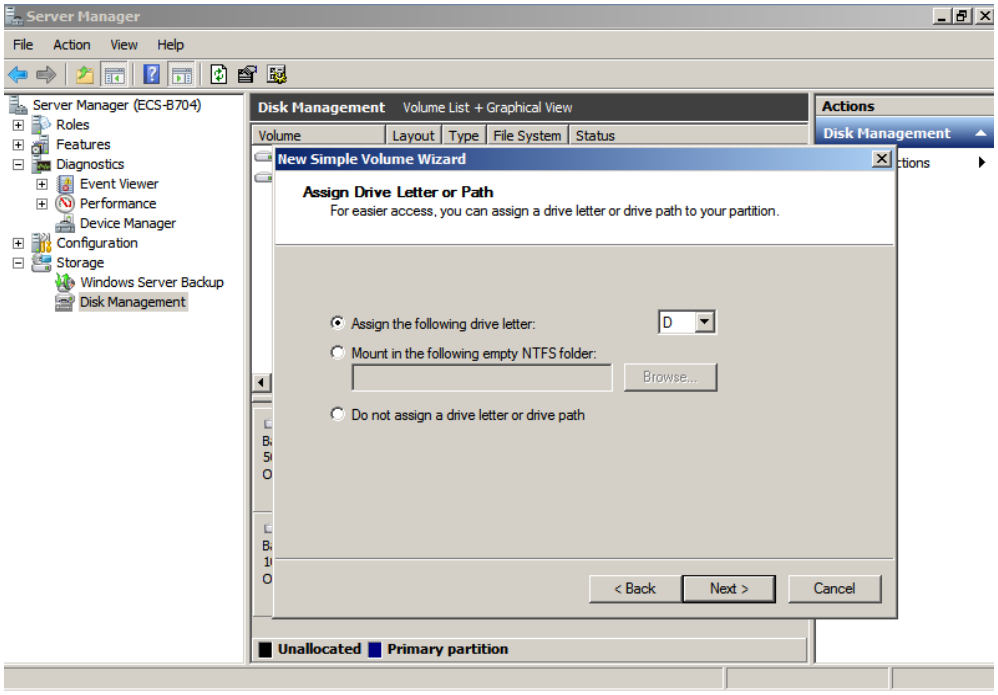
Step 8 Specify the volume size and click **Next**. The default value is the maximum size.

Figure 2-8 Specify Volume Size



Step 9 Assign the driver letter and click **Next**.

Figure 2-9 Assign Driver Letter or Path



Step 10 Select **Format this volume with the following settings**, set parameters based on the actual requirements, and select **Perform a quick format**. Then, click **Next**.

Figure 2-10 Format Partition

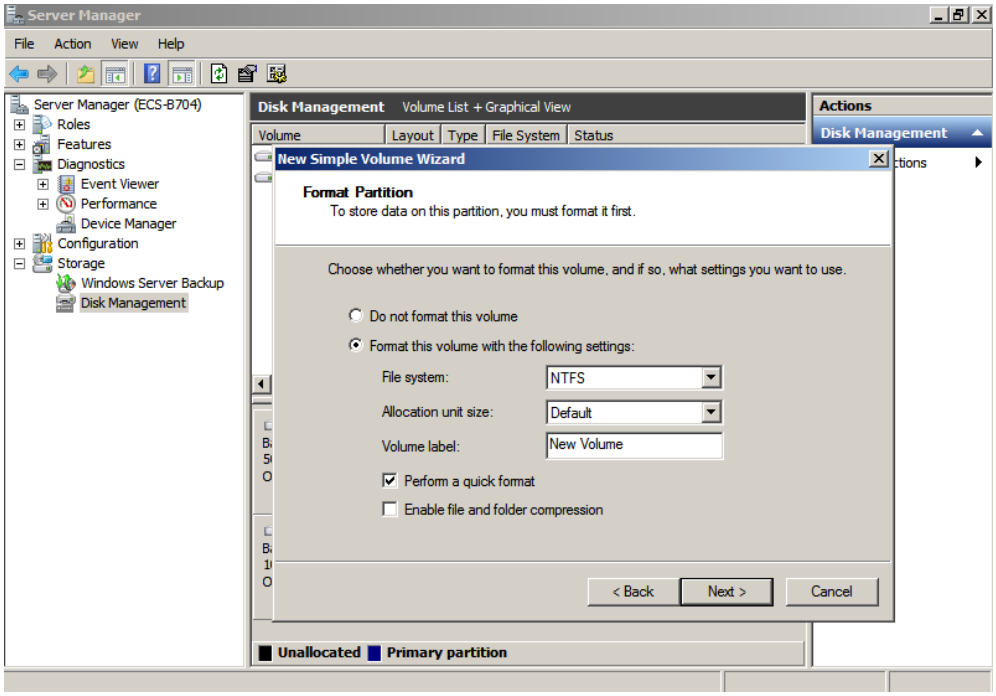
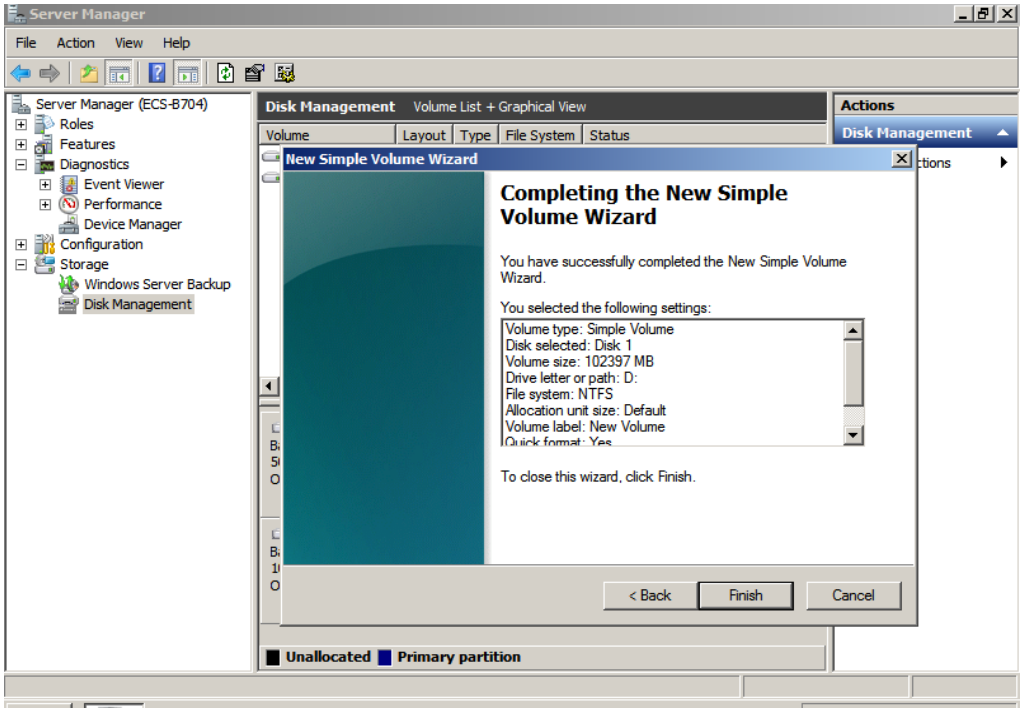


Figure 2-11 Completing the partition creation

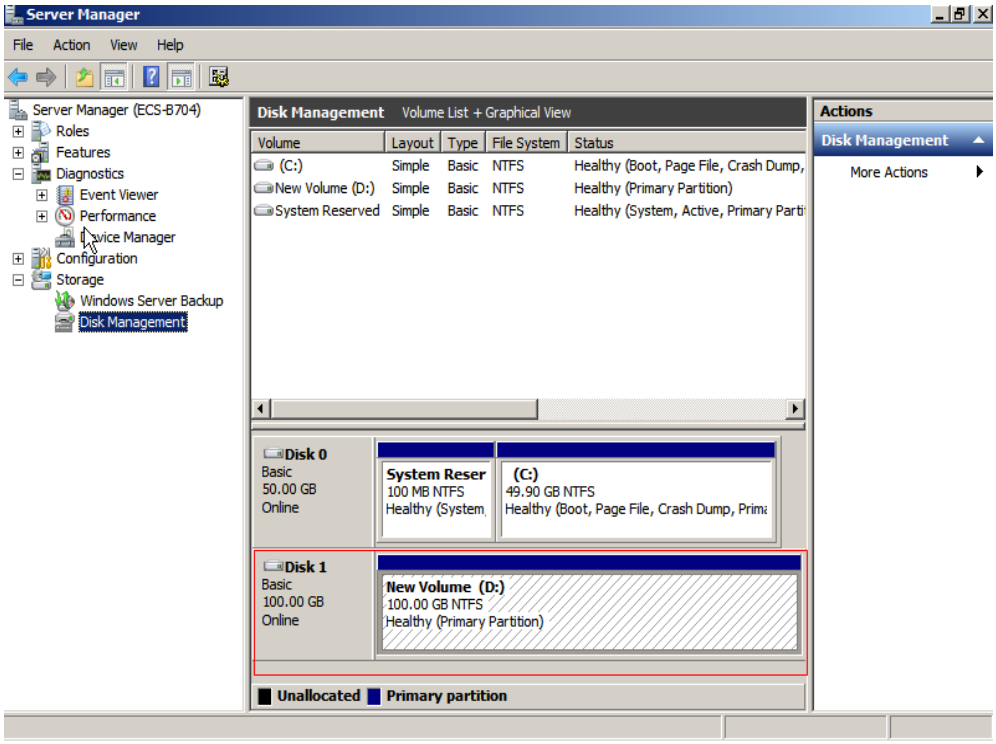


NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 11 Click **Finish**. Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully, as shown in **Figure 2-12**.

Figure 2-12 Disk initialization succeeded



----End

2.5.3 Initializing a Data Disk in Windows (Windows Server 2016)

Scenarios

This section uses Windows Server 2016 Standard 64bit to describe how to initialize a data disk attached to a server running Windows.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. For details, see **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Procedure

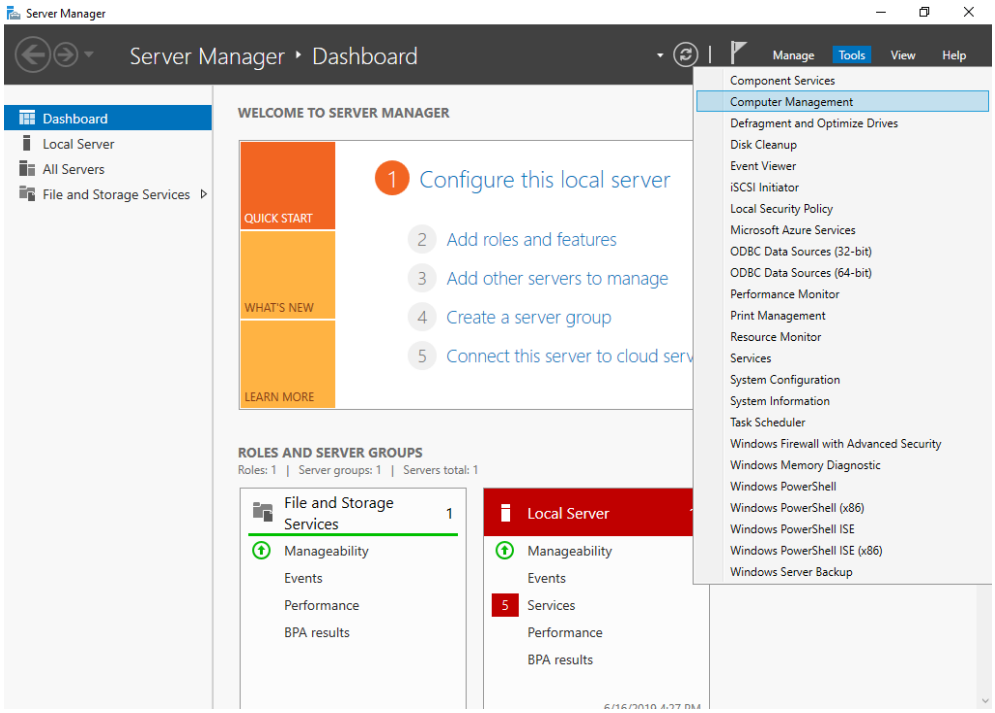
Step 1 On the desktop of the server, click the start icon in the lower left corner.

The **Windows Server** window is displayed.

Step 2 Click **Server Manager**.

The **Server Manager** window is displayed.

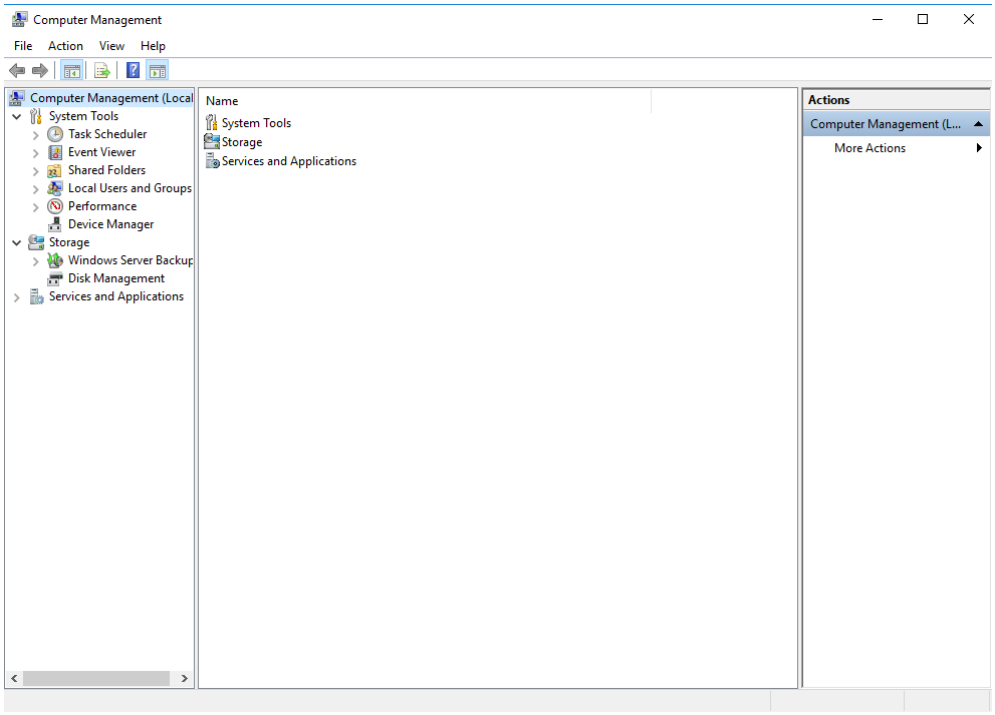
Figure 2-13 Server Manager



Step 3 In the upper right corner, choose **Tools** > **Computer Management**.

The **Computer Management** window is displayed.

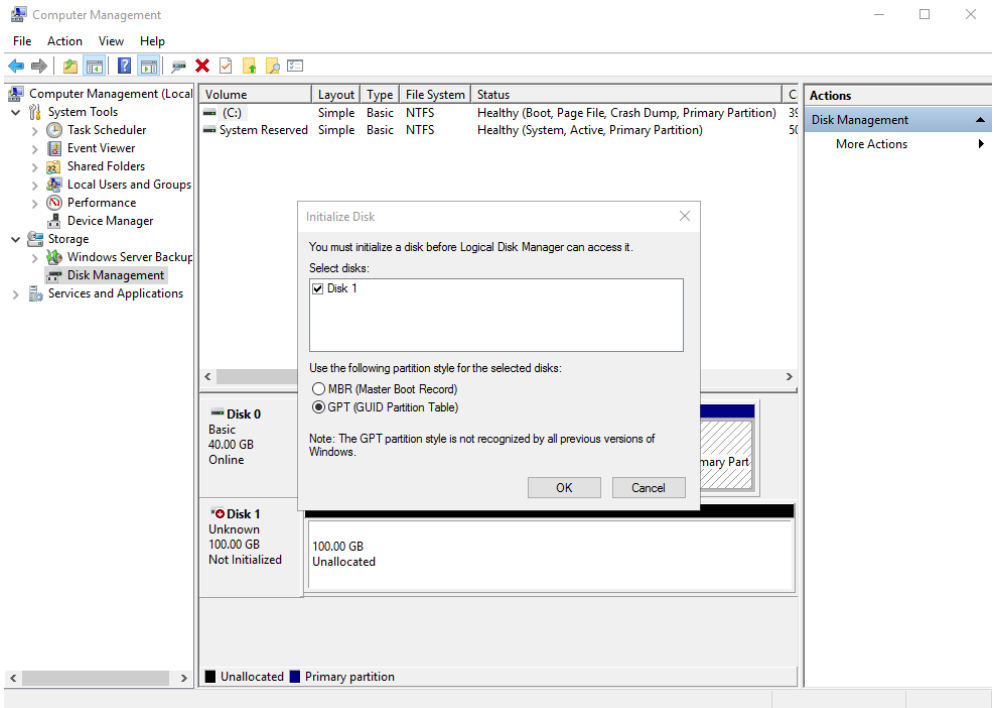
Figure 2-14 Computer Management



Step 4 Choose **Storage** > **Disk Management**.

Disks are displayed in the right pane. If there is a disk that is not initialized, the system will prompt you with the **Initialize Disk** dialog box.

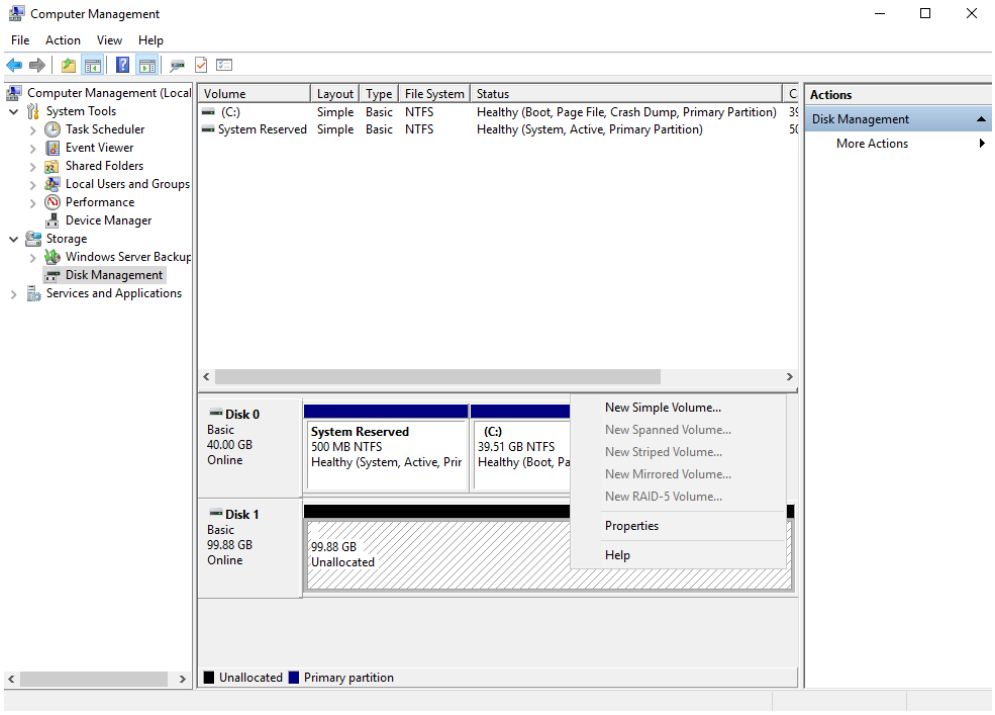
Figure 2-15 Disk list



Step 5 In the **Initialize Disk** dialog box, the to-be-initialized disk is selected. Select a disk partition style and click **OK**. In this example, **GPT (GUID Partition Table)** is selected.

The **Computer Management** window is displayed.

Figure 2-16 Computer Management (Windows Server 2016)



NOTICE

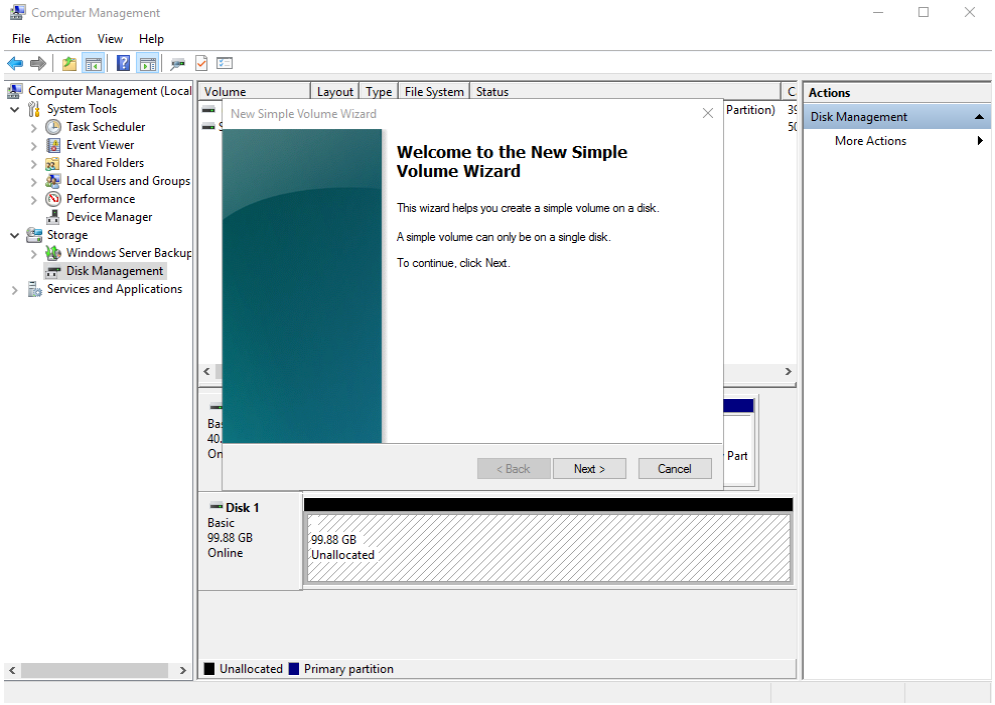
The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Step 6 Right-click at the unallocated disk space and choose **New Simple Volume** from the shortcut menu.

The **New Simple Volume Wizard** window is displayed.

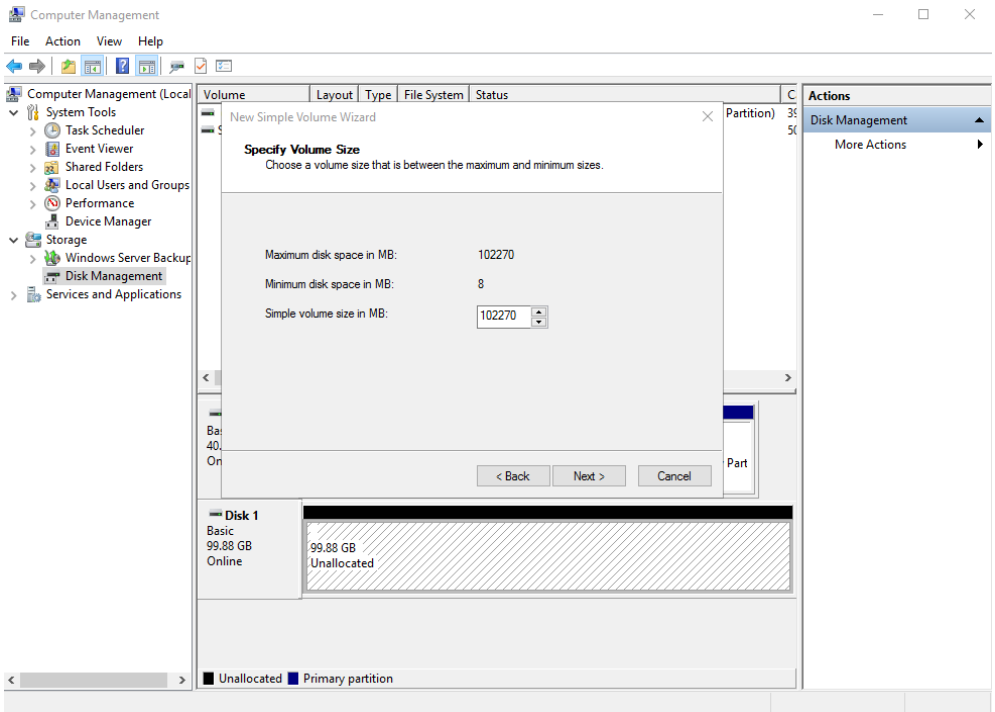
Figure 2-17 New Simple Volume Wizard (Windows Server 2016)



Step 7 Follow the prompts and click **Next**.

The **Specify Volume Size** page is displayed.

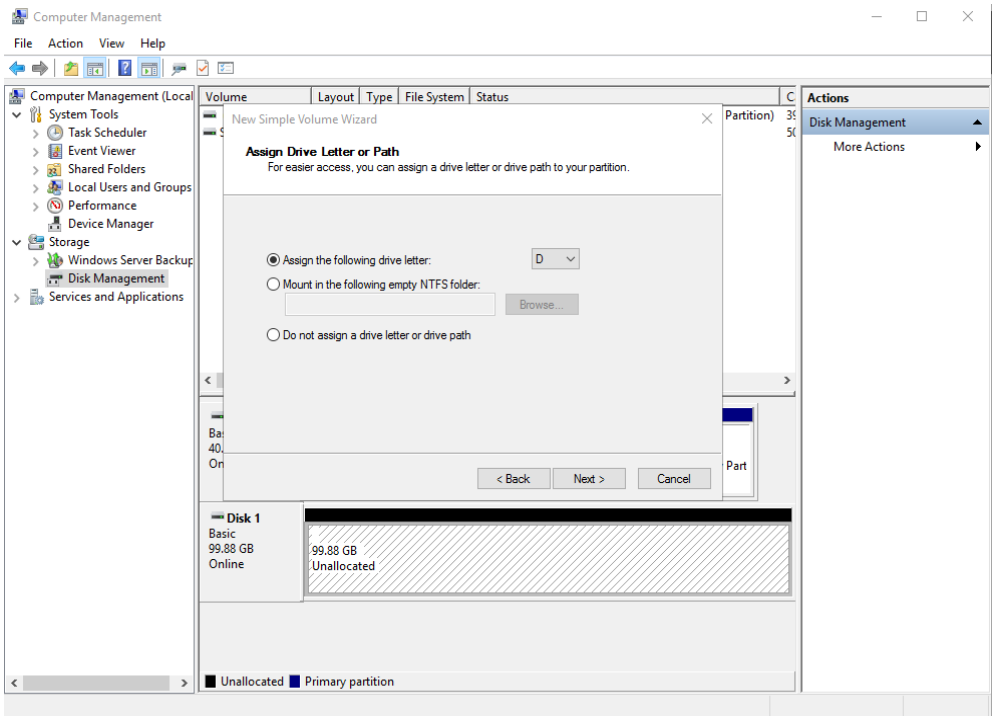
Figure 2-18 Specify Volume Size (Windows Server 2016)



Step 8 Specify the volume size and click **Next**. The system selects the maximum volume size by default. You can specify the volume size as required. In this example, the default setting is used.

The **Assign Drive Letter or Path** page is displayed.

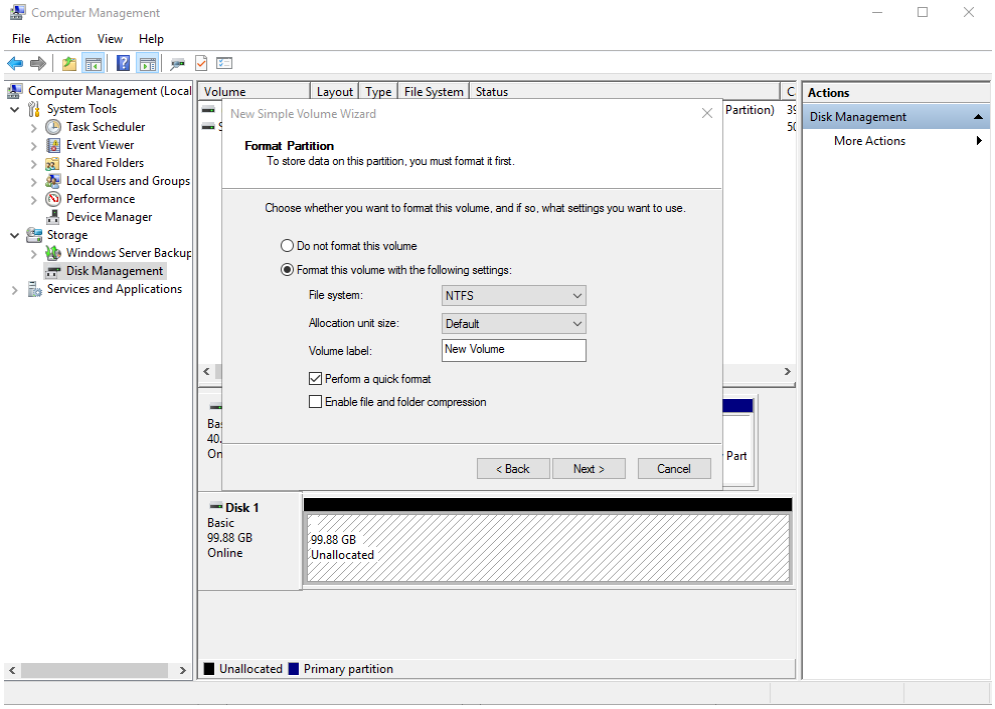
Figure 2-19 Assign Driver Letter or Path (Windows Server 2016)



Step 9 Assign a drive letter or path to your partition and click **Next**. The system assigns drive letter D by default. In this example, the default setting is used.

The **Format Partition** page is displayed.

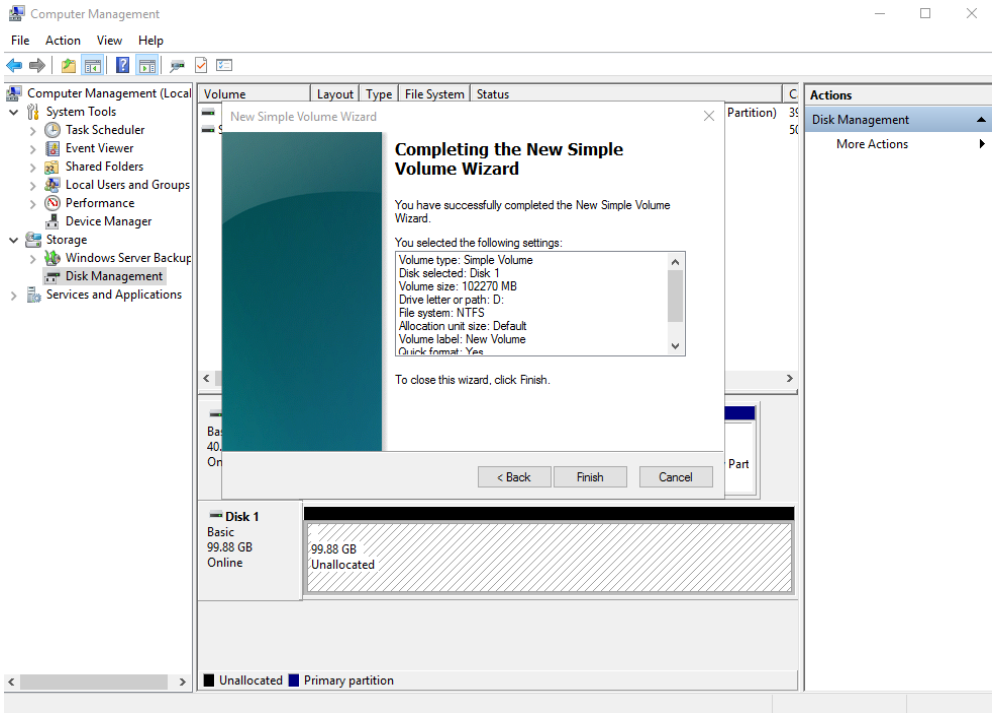
Figure 2-20 Format Partition (Windows Server 2016)



Step 10 Specify format settings and click **Next**. The system selects the NTFS file system by default. You can specify the file system type as required. In this example, the default setting is used.

The **Completing the New Simple Volume Wizard** page is displayed.

Figure 2-21 Completing the New Simple Volume Wizard (Windows Server 2016)



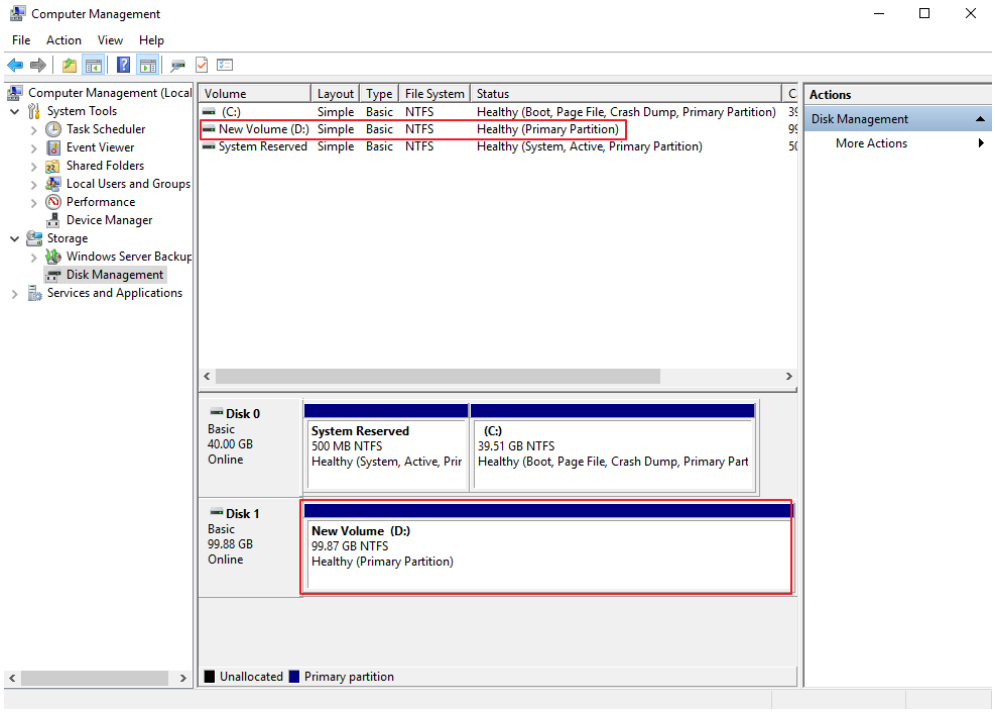
NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 11 Click **Finish**.

Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully, as shown in **Figure 2-22**.

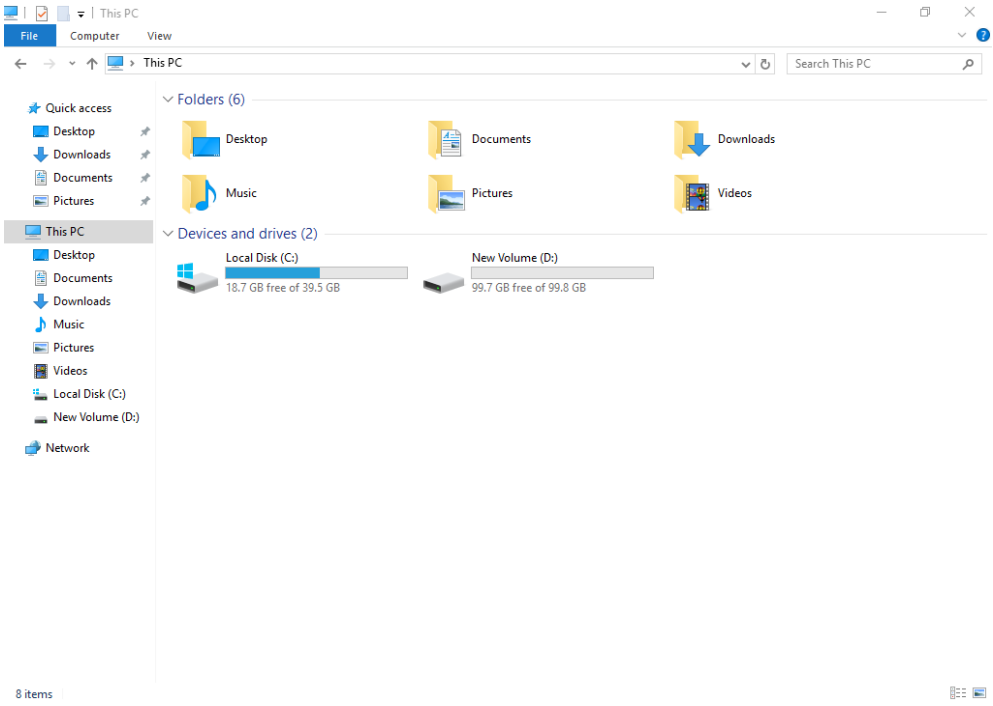
Figure 2-22 Disk initialization succeeded (Windows Server 2016)



Step 12 After the volume is created, click  on the task bar and check whether a new volume appears in **This PC**. In this example, New Volume (D:) is the new volume.

If New Volume (D:) appears, the disk is successfully initialized and no further action is required.

Figure 2-23 This PC (Windows Server 2016)



----End

2.5.4 Initializing a Data Disk in Linux (fdisk)

Scenarios

This section uses CentOS 7.4 64bit to describe how to initialize a data disk attached to a server running Linux and use fdisk to partition the data disk.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. In Linux, if you choose to use the GPT partition style, the fdisk partitioning tool cannot be used. Use the parted partitioning tool instead. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Creating and Mounting a Partition

The following example shows you how a new primary partition can be created on a new data disk that has been attached to a server. The primary partition will be

created using fdisk, and MBR is the default partition style. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured with automatic mounting at system start.

Step 1 Run the following command to query information about the new data disk:

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000bcb4e

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux

Disk /dev/vdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

In the command output, the server contains two disks. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk.

Step 2 Run the following command to enter fdisk to partition the new data disk:

fdisk *New data disk*

In this example, run the following command:

fdisk /dev/vdb

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# fdisk /dev/vdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x38717fc1.

Command (m for help):
```

Step 3 Enter **n** and press **Enter** to create a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
```

There are two types of disk partitions:

- Choosing **p** creates a primary partition.
- Choosing **e** creates an extended partition.

NOTE

If the MBR partition style is used, a maximum of 4 primary partitions, or 3 primary partitions and 1 extended partition can be created. The extended partition cannot be used directly and must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

Step 4 In this example, a primary partition is created. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
Select (default p): p
Partition number (1-4, default 1):
```

Partition number indicates the serial number of the primary partition. The value ranges from **1** to **4**.

Step 5 Enter the serial number of the primary partition and press **Enter**. Primary partition number **1** is used in this example. One usually starts with partition number **1** when partitioning an empty disk.

Information similar to the following is displayed:

```
Partition number (1-4, default 1): 1
First sector (2048-209715199, default 2048):
```

First sector indicates the start sector. The value ranges from **2048** to **209715199**, and the default value is **2048**.

Step 6 Select the default start sector **2048** and press **Enter**.

The system displays the start and end sectors of the partition's available space. You can customize the value within this range or use the default value. The start sector must be smaller than the partition's end sector.

Information similar to the following is displayed:

```
First sector (2048-209715199, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-209715199, default 209715199):
```

Last sector indicates the end sector. The value ranges from **2048** to **209715199**, and the default value is **209715199**.

Step 7 Select the default end sector **209715199** and press **Enter**.

The system displays the start and end sectors of the partition's available space. You can customize the value within this range or use the default value. The start sector must be smaller than the partition's end sector.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (2048-209715199, default 209715199):
Using default value 209715199
Partition 1 of type Linux and of size 100 GiB is set

Command (m for help):
```

A primary partition has been created for the new data disk.

Step 8 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed:

```
Command (m for help): p
```



```
Disk /dev/vdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x38717fc1

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048   209715199   104856576   83  Linux

Command (m for help):
```

Details about the **/dev/vdb1** partition are displayed.

Step 9 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

The partition is created.

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q**.

Step 10 Run the following command to synchronize the new partition table to the OS:

partprobe

Step 11 Run the following command to set the file system format for the new partition:

mkfs -t *File system format* **/dev/vdb1**

In this example, run the following command to set the **ext4** file system for the new partition:

mkfs -t ext4 /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26214144 blocks
1310707 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2174746624
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a period of time. Observe the system running status and do not exit.

NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 12 Run the following command to create a mount point:

mkdir *Mount point*

In this example, run the following command to create the **/mnt/sdc** mount point:

mkdir /mnt/sdc

Step 13 Run the following command to mount the new partition on the created mount point:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vdb1** on **/mnt/sdc**:

mount /dev/vdb1 /mnt/sdc

Step 14 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda1      ext4       43G  1.9G   39G   5% /
devtmpfs       devtmpfs  2.0G     0  2.0G   0% /dev
tmpfs          tmpfs     2.0G     0  2.0G   0% /dev/shm
tmpfs          tmpfs     2.0G  9.1M  2.0G   1% /run
tmpfs          tmpfs     2.0G     0  2.0G   0% /sys/fs/cgroup
tmpfs          tmpfs     398M     0  398M   0% /run/user/0
/dev/vdb1      ext4       106G   63M  101G   1% /mnt/sdc
```

New partition **/dev/vdb1** is mounted on **/mnt/sdc**.

NOTE

If the server is restarted, the mounting will become invalid. You can set automatic mounting for partitions at system start by modifying the **/etc/fstab** file. For details, see **Setting Automatic Mounting at System Start**.

----End

Setting Automatic Mounting at System Start

Modify the **fstab** file to set automatic disk mounting at server start.

The following procedure shows how to set automatic disk mounting at server start by using UUIDs to identify disks in the **fstab** file. You are advised not to use device names to identify disks in the file because a device name may change (for example, from /dev/vdb1 to /dev/vdb2) during the server stop or start, resulting in improper server running after restart.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

In this example, run the following command to query the UUID of the **/dev/vdb1** partition:

blkid /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

The UUID of the **/dev/vdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc              ext4    defaults        0 2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

Step 6 Perform the following operations to verify the automatic mounting function:

1. Run the following command to unmount the partition:

   umount *Disk partition*

   In this example, run the following command:

   umount /dev/vdb1

2. Run the following command to reload all the content in the **/etc/fstab** file:

   mount -a

3. Run the following command to query the file system mounting information:

   mount | grep *Mount point*

   In this example, run the following command:

   mount | grep /mnt/sdc

   If information similar to the following is displayed, the automatic mounting function takes effect:

   ```
   root@ecs-test-0001 ~]# mount | grep /mnt/sdc
   /dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
   ```

----End

2.5.5 Initializing a Data Disk in Linux (parted)

Scenarios

This section uses CentOS 7.4 64bit to describe how to initialize a data disk attached to a server running Linux and use parted to partition the data disk.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. In Linux, if you choose to use the GPT partition style, the fdisk partitioning tool cannot be used. Use the parted partitioning tool instead. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Creating and Mounting a Partition

The following example shows you how a new partition can be created on a new data disk that has been attached to a server. The partition will be created using parted, and GPT is used as the partition style. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured automatic mounting at system start.

Step 1 Run the following command to query information about the new data disk:

lsblk

Information similar to the following is displayed:

```
root@ecs-test-0001 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    253:0    0   40G  0 disk
└─vda1 253:1    0   40G  0 part /
vdb    253:16   0  100G  0 disk
```

In the command output, the server contains two disks. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk.

Step 2 Run the following command to enter parted to partition the new data disk:

parted *New data disk*

In this example, run the following command:

parted /dev/vdb

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# parted /dev/vdb
GNU Parted 3.1
```



```
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Step 3 Enter **p** and press **Enter** to view the current disk partition style.

Information similar to the following is displayed:

```
(parted) p
Error: /dev/vdb: unrecognised disk label
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted)
```

In the command output, the **Partition Table** value is **unknown**, indicating that no partition style is set for the new disk.

Step 4 Run the following command to set the disk partition style:

mklabel *Disk partition style*

In this example, run the following command to set the partition style to GPT: (Disk partition styles can be MBR or GPT.)

mklabel gpt

NOTICE

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Step 5 Enter **p** and press **Enter** to view the disk partition style.

Information similar to the following is displayed:

```
(parted) mklabel gpt
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 107GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

(parted)
```

In the command output, the **Partition Table** value is **gpt**, indicating that the disk partition style is GPT.

Step 6 Enter **unit s** and press **Enter** to set the measurement unit of the disk to sector.

Step 7 Run the following command and press **Enter**:

mkpart *Partition name Start sector End sector*

In this example, run the following command:

mkpart test 2048s 100%

In this example, one partition is created for the new data disk. Variable *2048s* indicates the disk start sector, and variable *100%* indicates the disk end sector. The two values are used for reference only. You can determine the number of partitions and the partition size based on your service requirements.

Information similar to the following is displayed:

```
(parted) mkpart opt 2048s 100%
(parted)
```

Step 8 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed:

```
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 209715200s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End         Size        File system  Name  Flags
 1      2048s  209713151s  209711104s               test

(parted)
```

Step 9 Enter **q** and press **Enter** to exit parted.

Information similar to the following is displayed:

```
(parted) q
Information: You may need to update /etc/fstab.
```

You can set automatic disk mounting by updating the **/etc/fstab** file. Before updating the file, set the file system format for the partition and mount the partition on the mount point.

Step 10 Run the following command to view the disk partition information:

lsblk

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda    253:0    0   40G  0 disk
└─vda1 253:1    0   40G  0 part /
vdb    253:16   0  100G  0 disk
└─vdb1 253:17   0  100G  0 part
```

In the command output, **/dev/vdb1** is the partition you created.

Step 11 Run the following command to set the file system format for the new partition:

mkfs -t *File system format* **/dev/vdb1**

In this example, run the following command to set the **ext4** file system for the new partition:

mkfs -t ext4 /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
```



```
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26213888 blocks
1310694 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2174746624
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a period of time. Observe the system running status and do not exit.

NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 12 Run the following command to create a mount point:

mkdir *Mount point*

In this example, run the following command to create the **/mnt/sdc** mount point:

mkdir /mnt/sdc

Step 13 Run the following command to mount the new partition on the created mount point:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vdb1** on **/mnt/sdc**:

mount /dev/vdb1 /mnt/sdc

Step 14 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda1      ext4       43G  1.9G   39G   5% /
devtmpfs       devtmpfs  2.0G     0  2.0G   0% /dev
tmpfs          tmpfs     2.0G     0  2.0G   0% /dev/shm
tmpfs          tmpfs     2.0G  9.0M  2.0G   1% /run
tmpfs          tmpfs     2.0G     0  2.0G   0% /sys/fs/cgroup
tmpfs          tmpfs     398M     0  398M   0% /run/user/0
/dev/vdb1      ext4      106G   63M  101G   1% /mnt/sdc
```

New partition **/dev/vdb1** is mounted on **/mnt/sdc**.

NOTE

If the server is restarted, the mounting will become invalid. You can set automatic mounting for partitions at system start by modifying the **/etc/fstab** file. For details, see **Setting Automatic Mounting at System Start**.

----End

Setting Automatic Mounting at System Start

Modify the **fstab** file to set automatic disk mounting at server start.

The following procedure shows how to set automatic disk mounting at server start by using UUIDs to identify disks in the **fstab** file. You are advised not to use device names to identify disks in the file because a device name may change (for example, from /dev/vdb1 to /dev/vdb2) during the server stop or start, resulting in improper server running after restart.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

In this example, run the following command to query the UUID of the **/dev/vdb1** partition:

blkid /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

The UUID of the **/dev/vdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc              ext4    defaults        0 2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

Step 6 Perform the following operations to verify the automatic mounting function:

1. Run the following command to unmount the partition:

   umount *Disk partition*

   In this example, run the following command:

   umount /dev/vdb1

2. Run the following command to reload all the content in the **/etc/fstab** file:

mount -a

3. Run the following command to query the file system mounting information:

   mount | grep *Mount point*

   In this example, run the following command:

   mount | grep /mnt/sdc

   If information similar to the following is displayed, the automatic mounting function takes effect:

```
root@ecs-test-0001 ~]# mount | grep /mnt/sdc
/dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
```

----End

2.5.6 Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)

Scenarios

This section uses Windows Server 2008 R2 Standard 64bit to describe how to initialize a data disk whose capacity is larger than 2 TB. In the following operations, the capacity of the example disk is 3 TB.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. For details, see **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Procedure

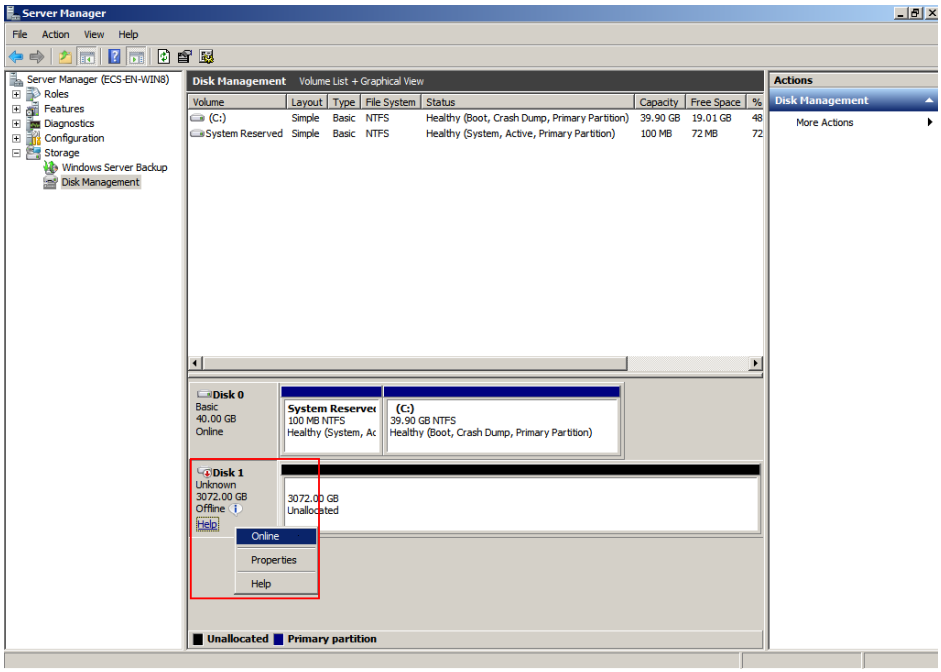
Step 1 On the desktop of the server, click **Start**.

The **Start** window is displayed.

Step 2 Right-click **Computer** and choose **Manage** from the short-cut menu.

The **Server Manager** window is displayed.

Figure 2-24 Server Manager (Windows Server 2008)

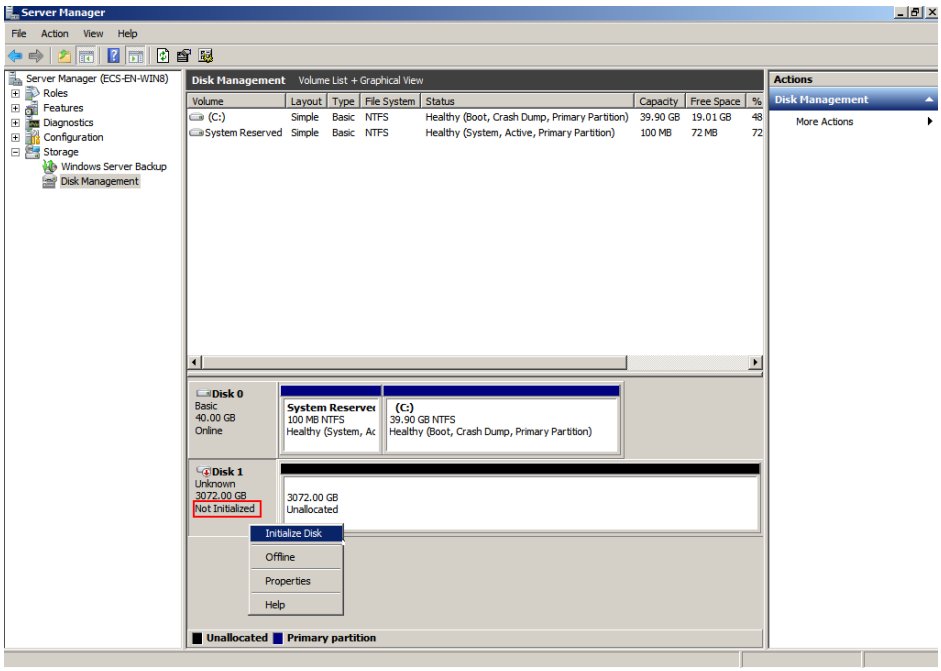


Step 3 Disks are listed in the right pane. If the new disk is in the offline state, bring it online before initialize it.

In the **Disk 1** area, right-click and choose **Online** from the shortcut menu.

When the Disk 1 status changes from **Offline** to **Not Initialized**, the disk has been brought online.

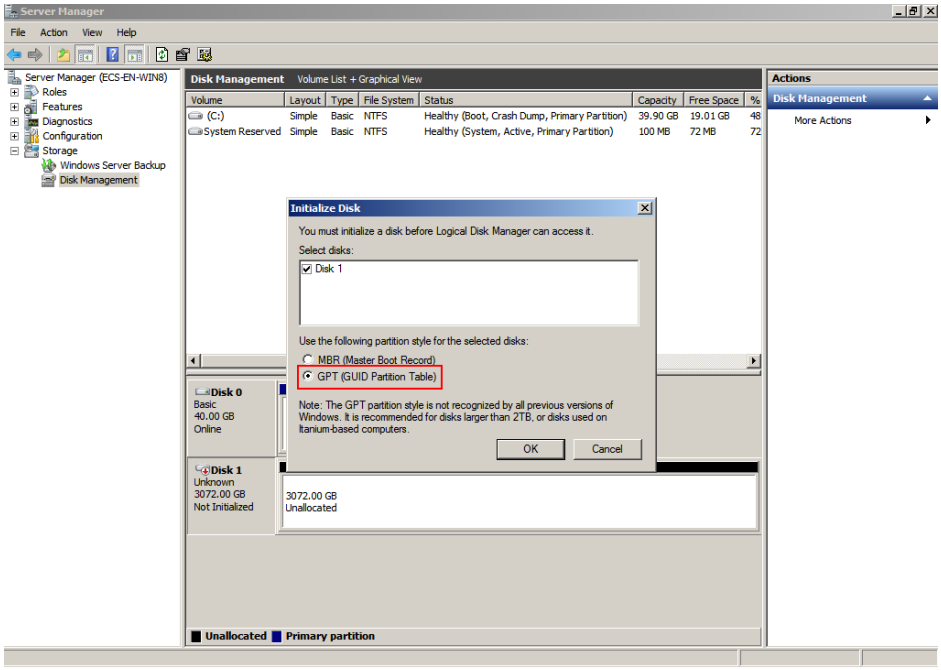
Figure 2-25 Bring online succeeded (Windows Server 2008)



Step 4 In the **Disk 1** area, right-click and choose **Initialize Disk** from the shortcut menu.

The **Initialize Disk** dialog box is displayed.

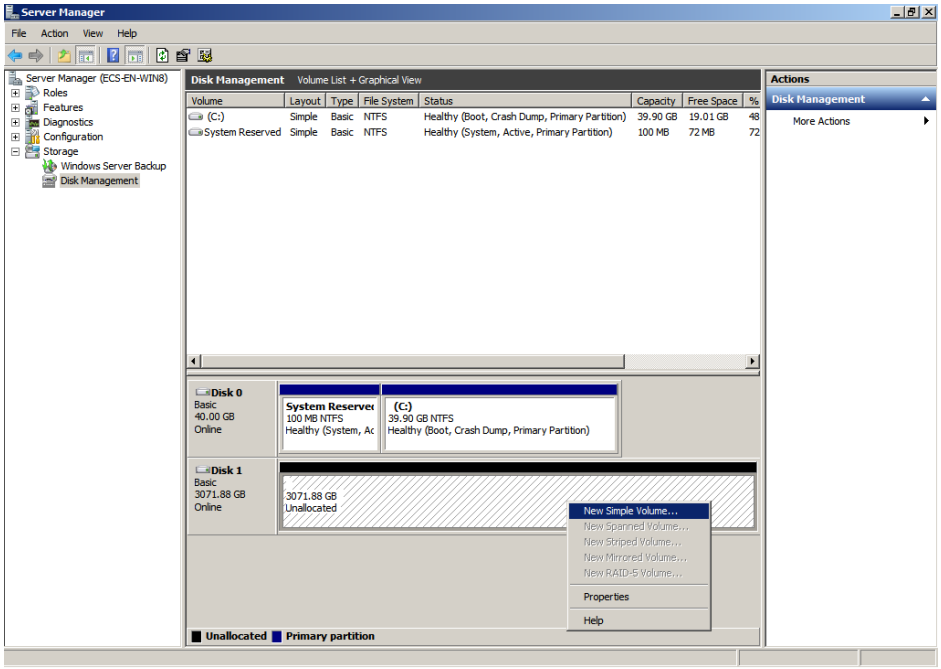
Figure 2-26 Initialize Disk (Windows Server 2008)



Step 5 In the **Initialize Disk** dialog box, the to-be-initialized disk is selected. In this example, the disk capacity is larger than 2 TB. Therefore, select **GPT (GUID Partition Table)** and click **OK**.

The **Server Manager** window is displayed.

Figure 2-27 Server Manager (Windows Server 2008)



NOTICE

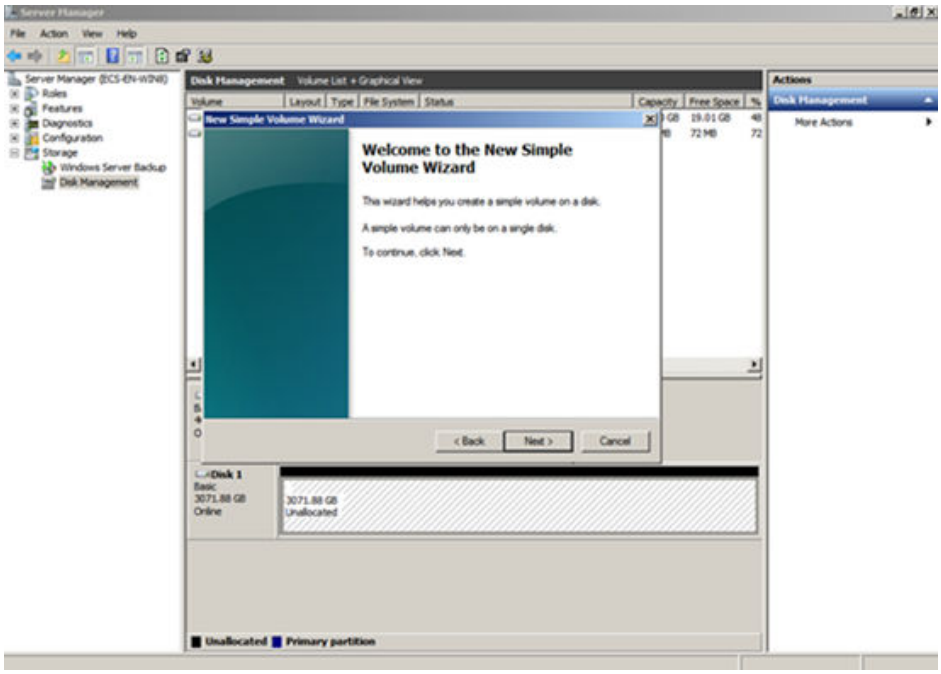
The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Step 6 Right-click at the unallocated disk space and choose **New Simple Volume** from the shortcut menu.

The **New Simple Volume Wizard** window is displayed.

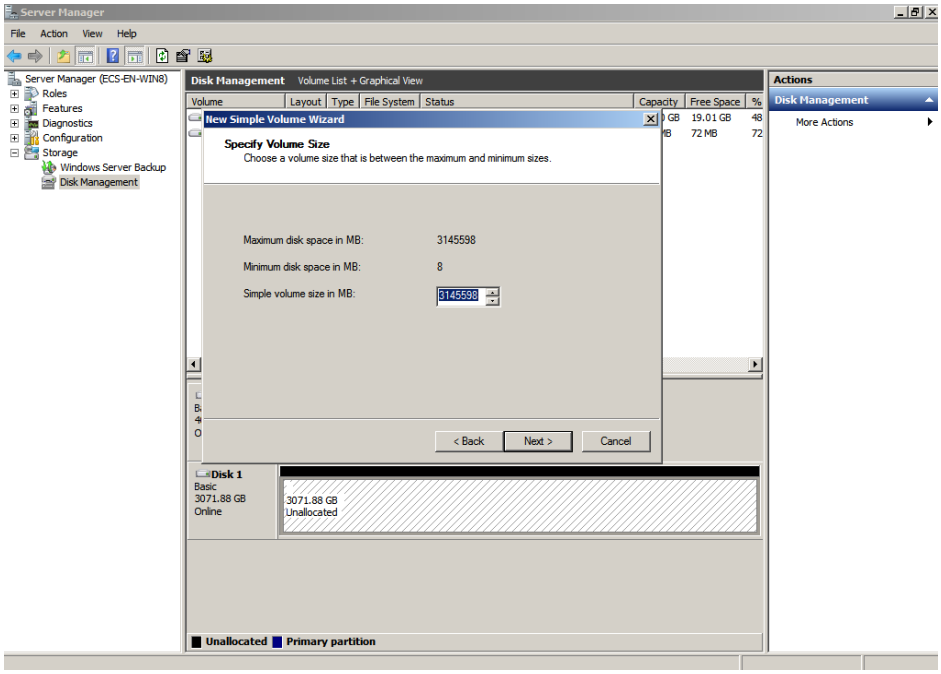
Figure 2-28 New Simple Volume Wizard (Windows Server 2008)



Step 7 Follow the prompts and click **Next**.

The **Specify Volume Size** page is displayed.

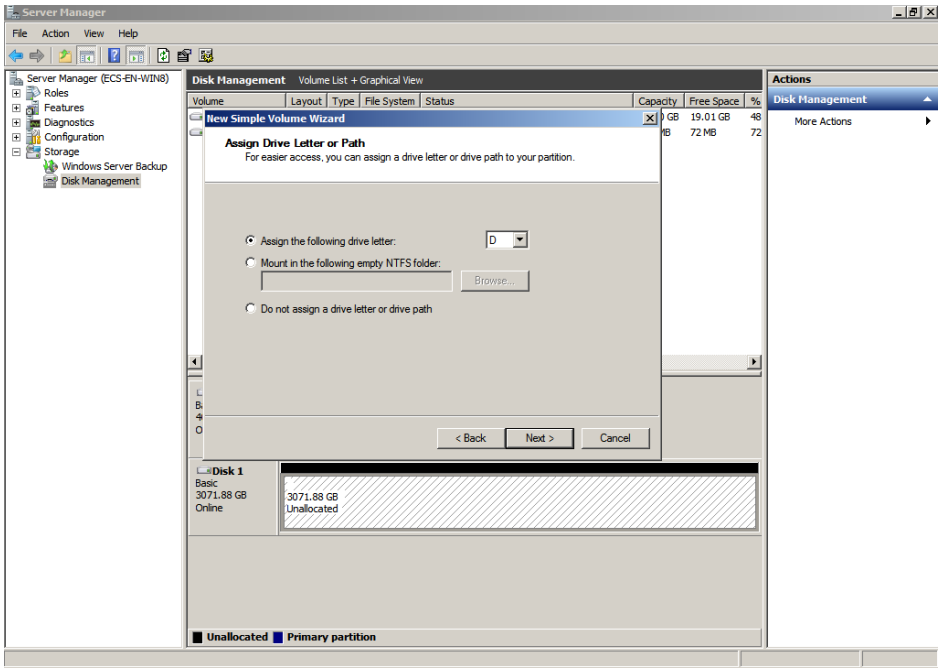
Figure 2-29 Specify Volume Size (Windows Server 2008)



Step 8 Specify the volume size and click **Next**. The system selects the maximum volume size by default. You can specify the volume size as required. In this example, the default setting is used.

The **Assign Drive Letter or Path** page is displayed.

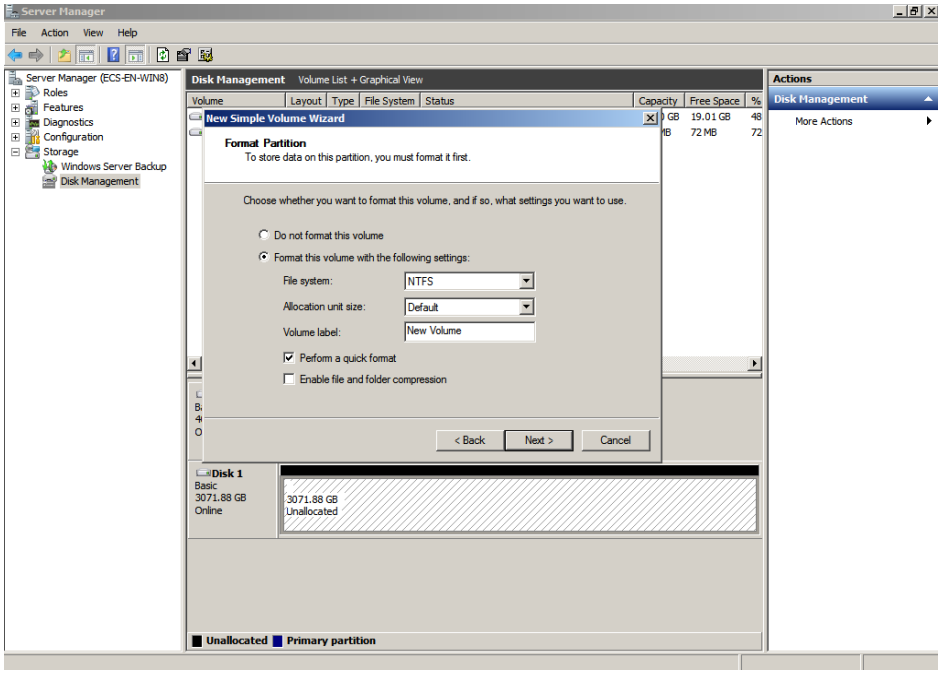
Figure 2-30 Assign Driver Letter or Path (Windows Server 2008)



Step 9 Assign a drive letter or path to your partition and click **Next**. The system assigns drive letter D by default. In this example, the default setting is used.

The **Format Partition** page is displayed.

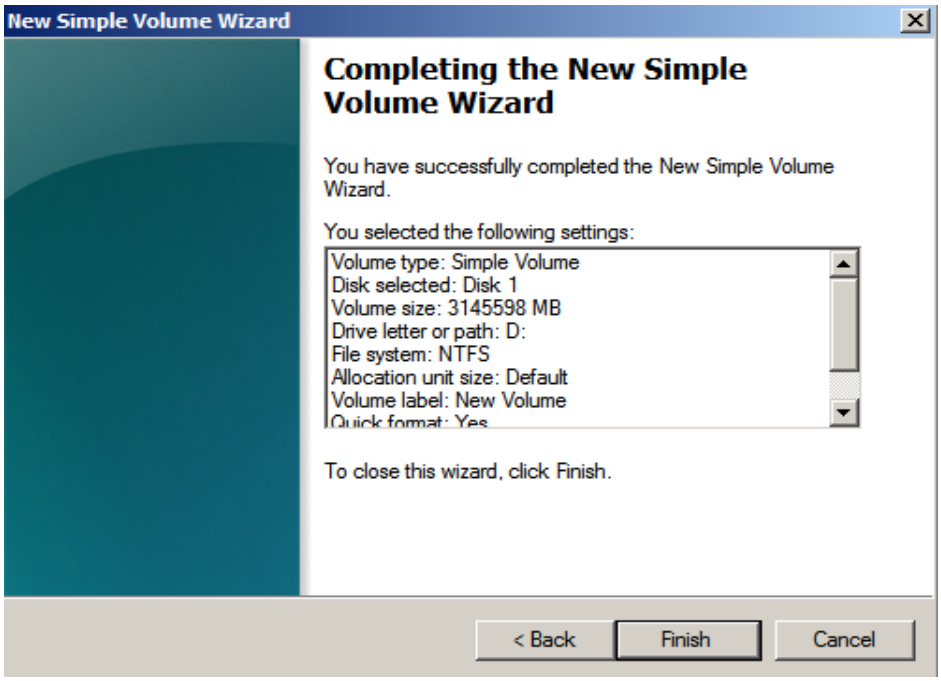
Figure 2-31 Format Partition (Windows Server 2008)



Step 10 Specify format settings and click **Next**. The system selects the NTFS file system by default. You can specify the file system type as required. In this example, the default setting is used.

The **Completing the New Simple Volume Wizard** page is displayed.

Figure 2-32 Completing the New Simple Volume Wizard



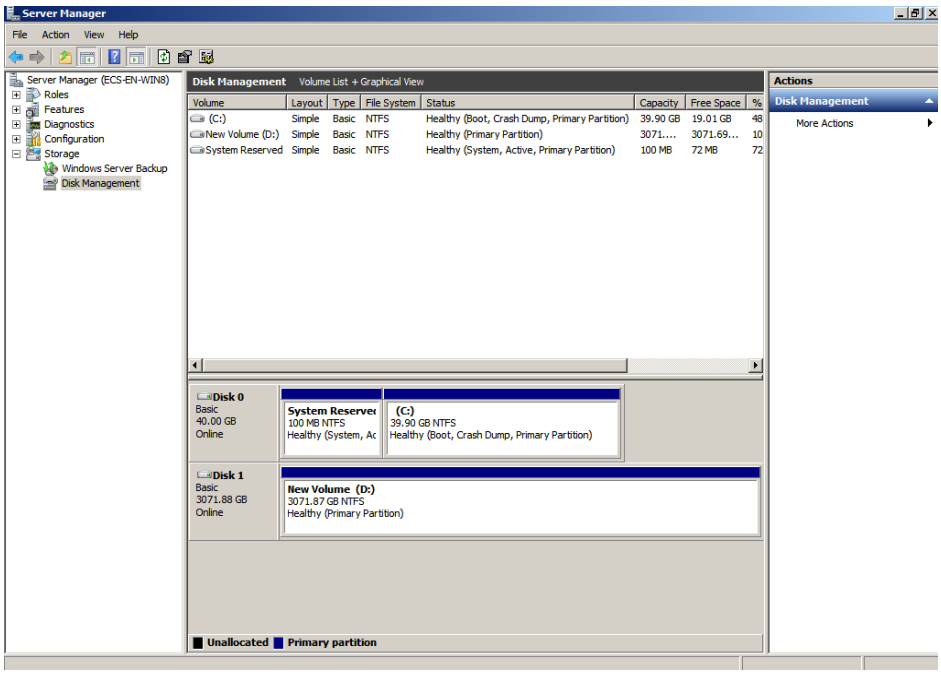
NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 11 Click **Finish**.

Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully, as shown in **Figure 2-33**.

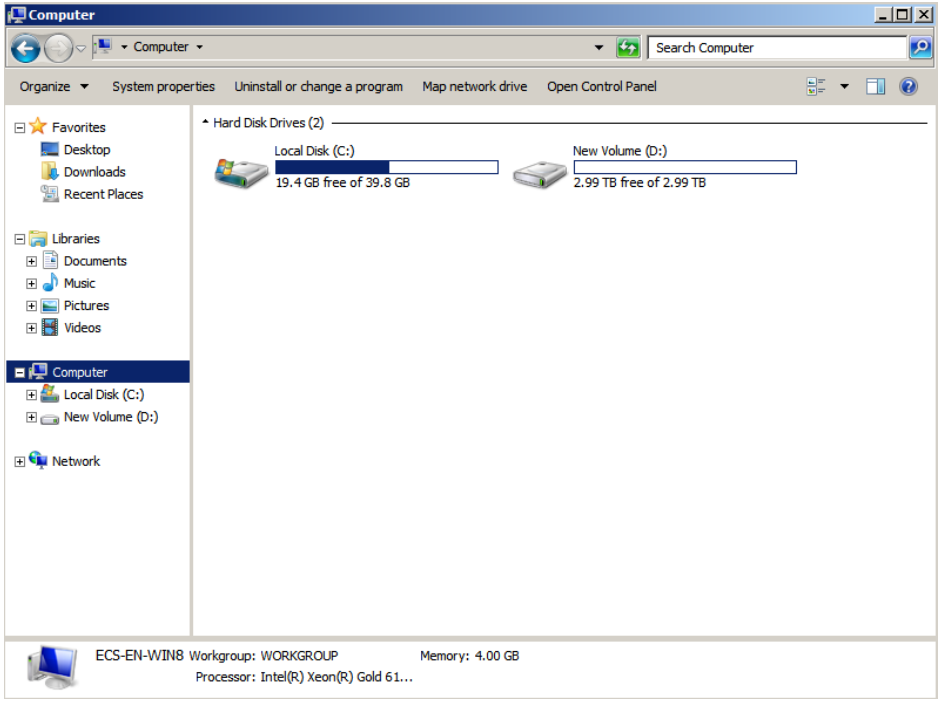
Figure 2-33 Disk initialization succeeded (Windows Server 2008)



Step 12 After the volume is created, click  and check whether a new volume appears in **Computer**. In this example, New Volume (D:) is the new volume.

If New Volume (D:) appears, the disk is successfully initialized and no further action is required.

Figure 2-34 Computer (Windows Server 2008)



----End

2.5.7 Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2012)

Scenarios

This section uses Windows Server 2012 R2 Standard 64bit to describe how to initialize a data disk whose capacity is larger than 2 TB. In the following operations, the capacity of the sample disk is 3 TB.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. For details, see **Initializing a Data Disk Greater Than 2 TB in Windows (Windows Server 2008)**. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

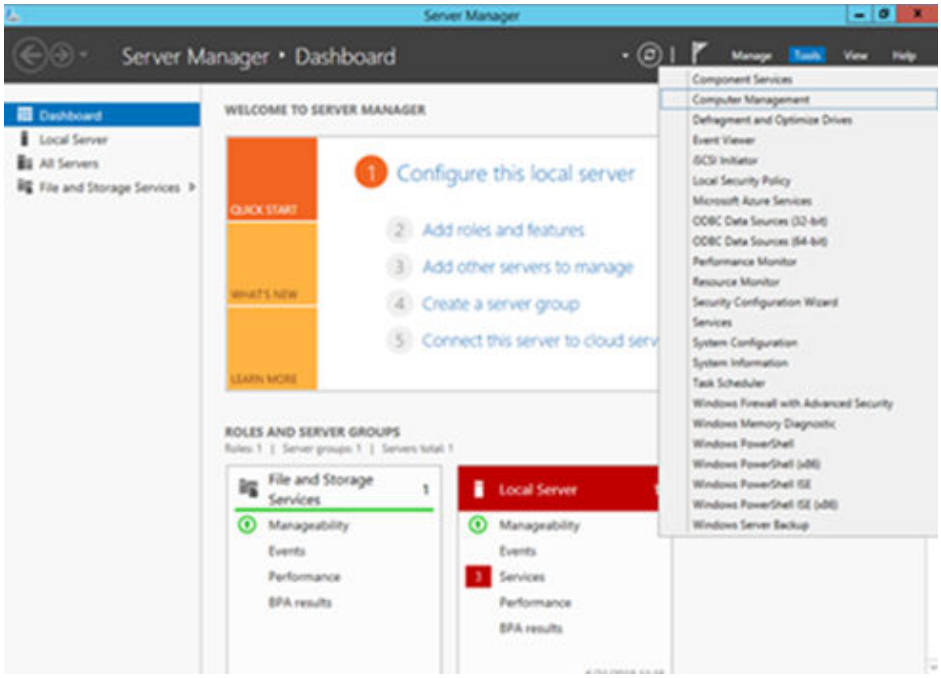
- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Procedure

Step 1 On the desktop of the server, click  in the lower area.

The **Server Manager** window is displayed.

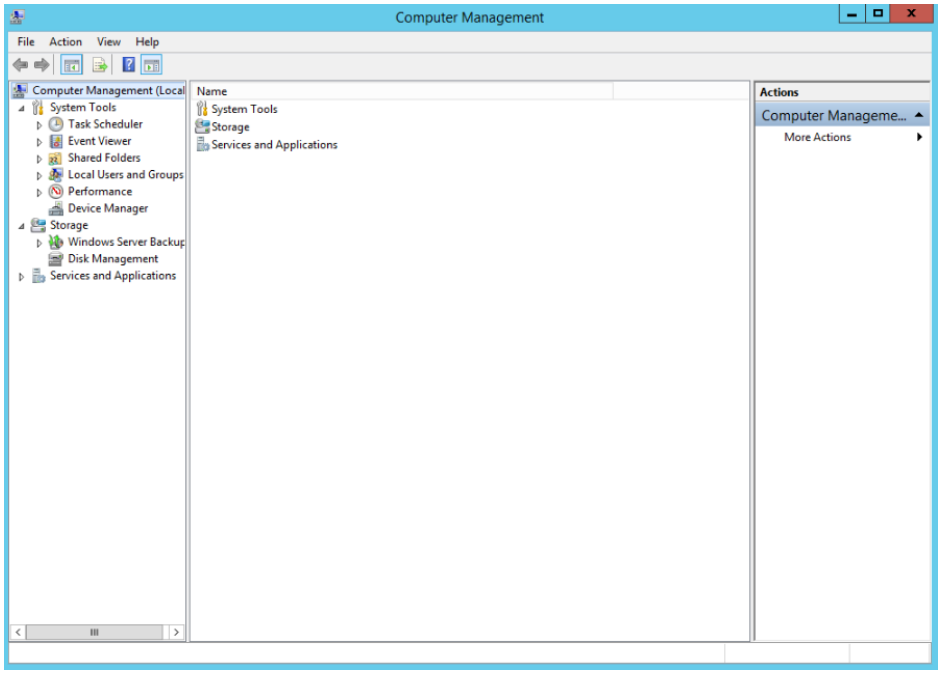
Figure 2-35 Server Manager (Windows Server 2012)



Step 2 In the upper right corner, choose **Tools** > **Computer Management**.

The **Computer Management** window is displayed.

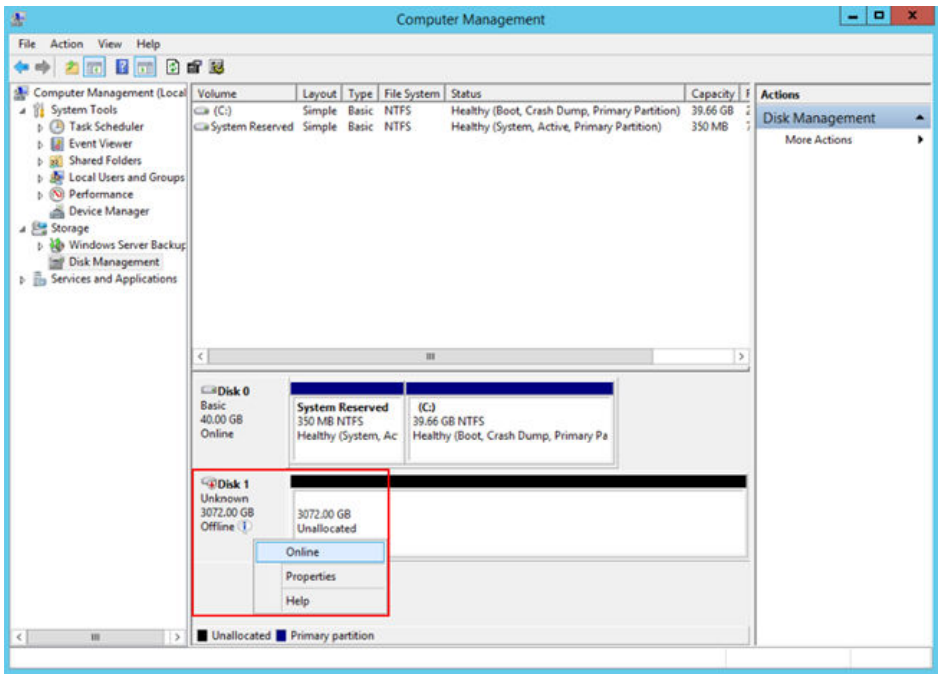
Figure 2-36 Computer Management window (Windows Server 2012)



Step 3 Choose **Storage** > **Disk Management**.

Disks are displayed in the right pane.

Figure 2-37 Disk Management list (Windows Server 2012)

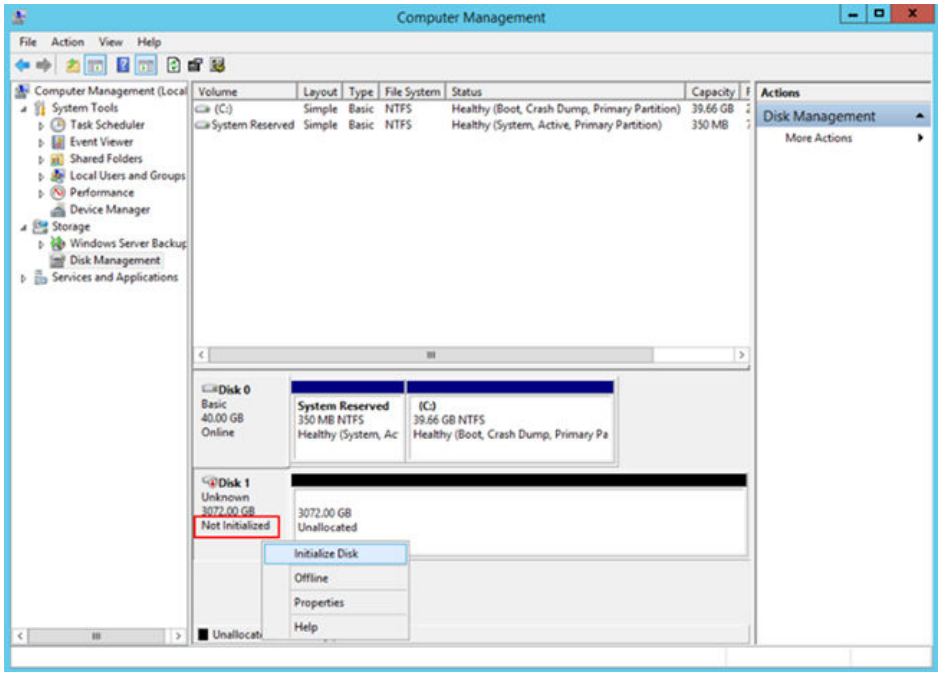


Step 4 (Optional) If the new disk is in the offline state, bring it online before initializing it.

In the **Disk 1** area, right-click and choose **Online** from the shortcut menu.

When the status of Disk 1 changes from **Offline** to **Not Initialized**, the disk has been brought online.

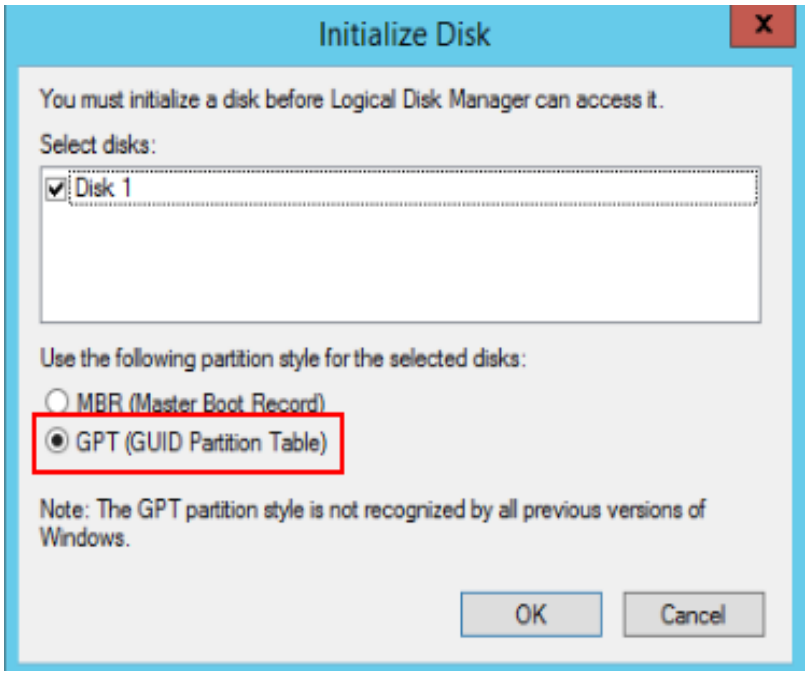
Figure 2-38 Bring online succeeded (Windows Server 2012)



Step 5 In the **Disk 1** area, right-click and choose **Initialize Disk** from the shortcut menu.

The **Initialize Disk** dialog box is displayed.

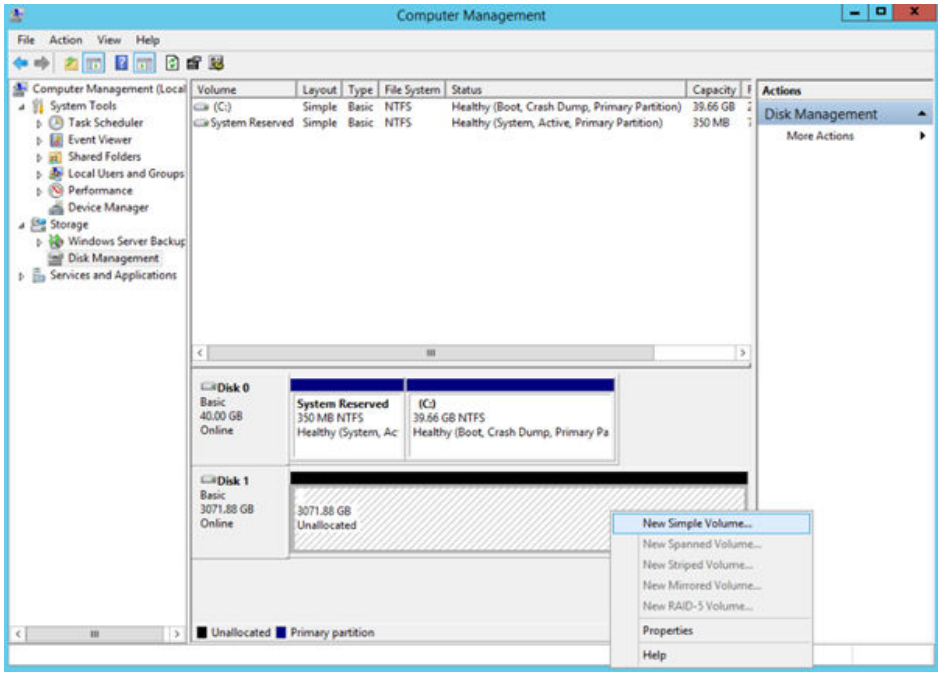
Figure 2-39 Initialize Disk (Windows Server 2012)



Step 6 In the **Initialize Disk** dialog box, the to-be-initialized disk is selected. In this example, the disk capacity is larger than 2 TB. Therefore, select **GPT (GUID Partition Table)** and click **OK**.

The **Computer Management** window is displayed.

Figure 2-40 Computer Management (Windows Server 2012)



NOTICE

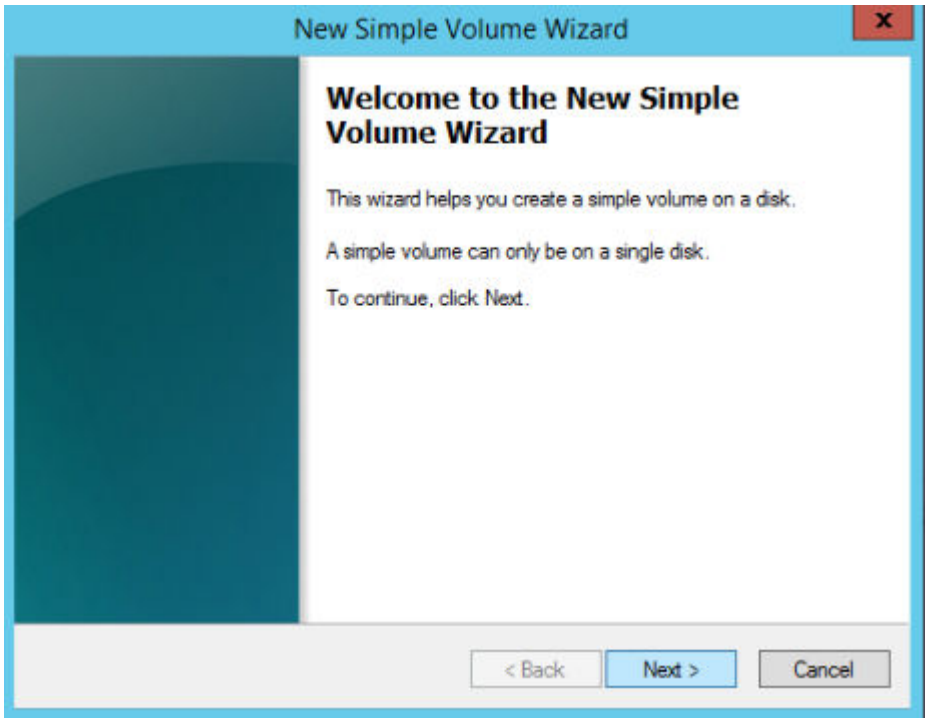
The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Step 7 Right-click at the unallocated disk space and choose **New Simple Volume** from the shortcut menu.

The **New Simple Volume Wizard** window is displayed.

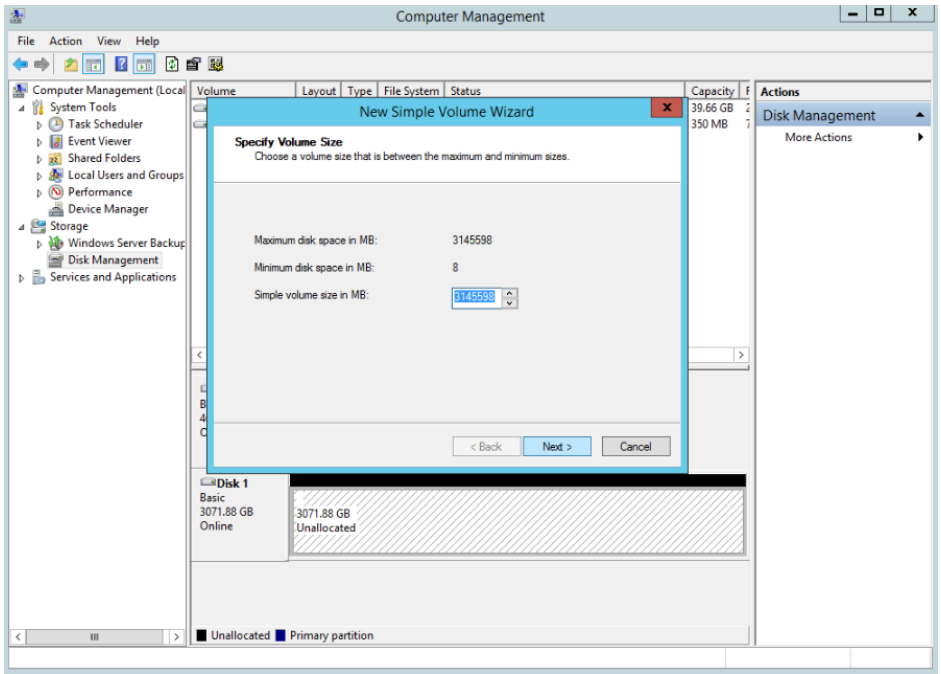
Figure 2-41 New Simple Volume Wizard (Windows Server 2012)



Step 8 Follow the prompts and click **Next**.

The **Specify Volume Size** page is displayed.

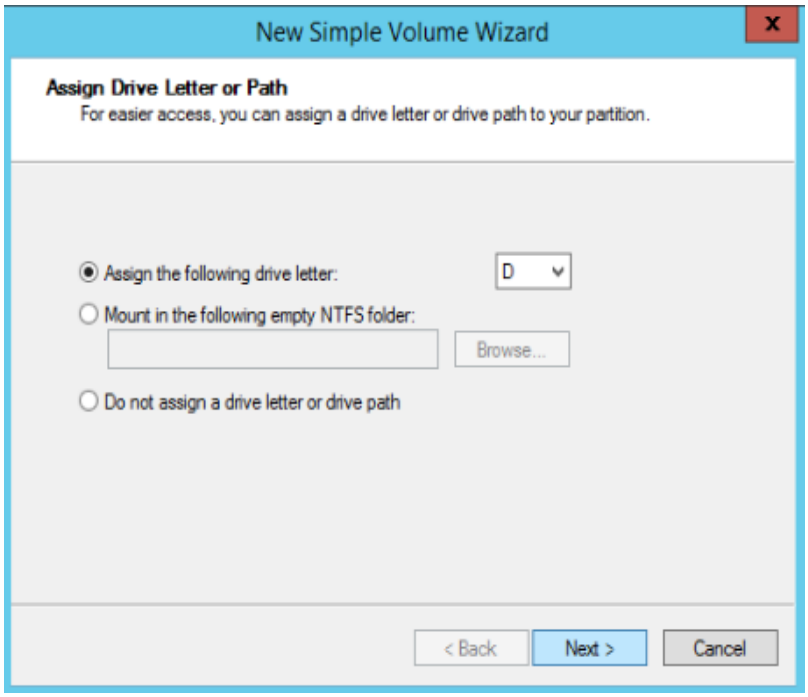
Figure 2-42 Specify Volume Size (Windows Server 2012)



Step 9 Specify the volume size and click **Next**. The system selects the maximum volume size by default. You can specify the volume size as required. In this example, the default setting is used.

The **Assign Drive Letter or Path** page is displayed.

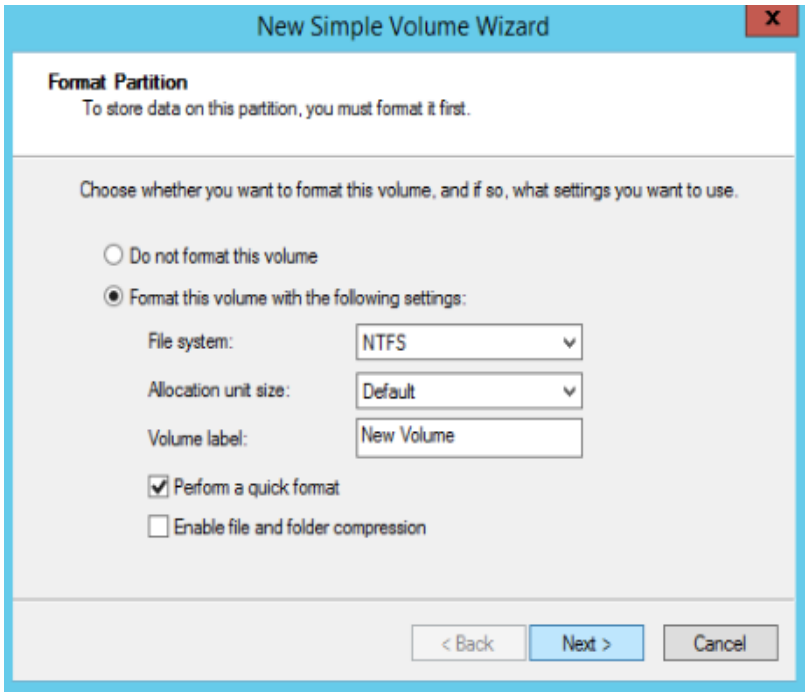
Figure 2-43 Assign Driver Letter or Path (Windows Server 2012)



Step 10 Assign a drive letter or path to your partition and click **Next**. The system assigns drive letter D by default. In this example, the default setting is used.

The **Format Partition** page is displayed.

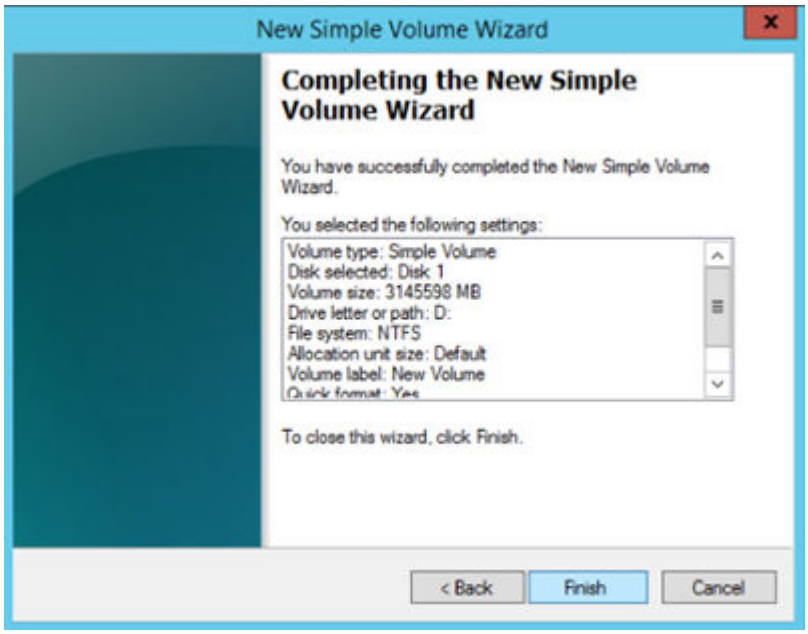
Figure 2-44 Format Partition (Windows Server 2012)



Step 11 Specify format settings and click **Next**. The system selects the NTFS file system by default. You can specify the file system type as required. In this example, the default setting is used.

The **Completing the New Simple Volume Wizard** page is displayed.

Figure 2-45 Completing the New Simple Volume Wizard (Windows Server 2012)



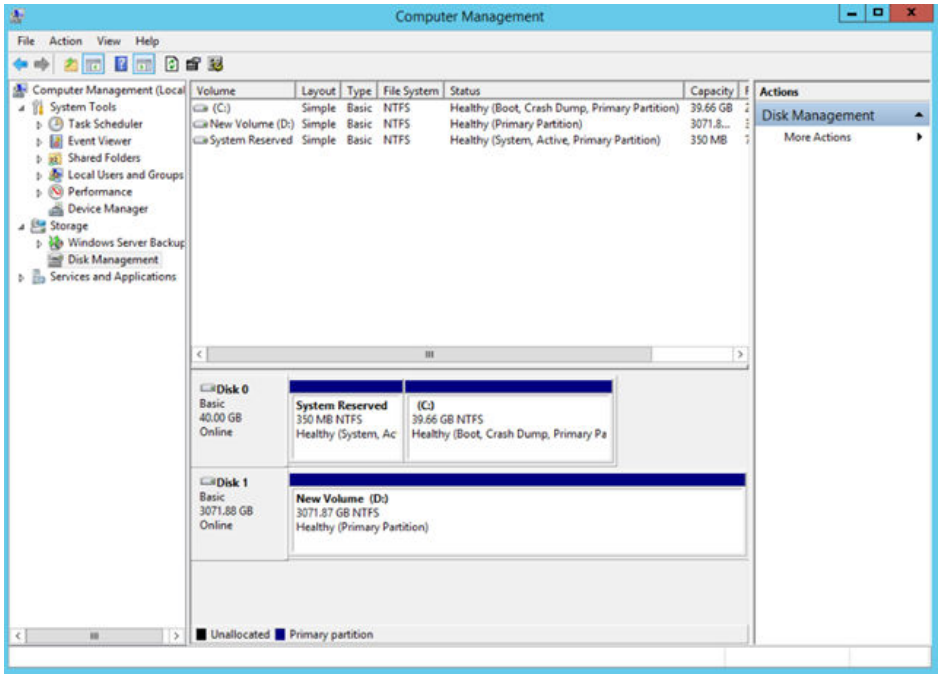
NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 12 Click **Finish**.

Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully, as shown in **Figure 2-46**.

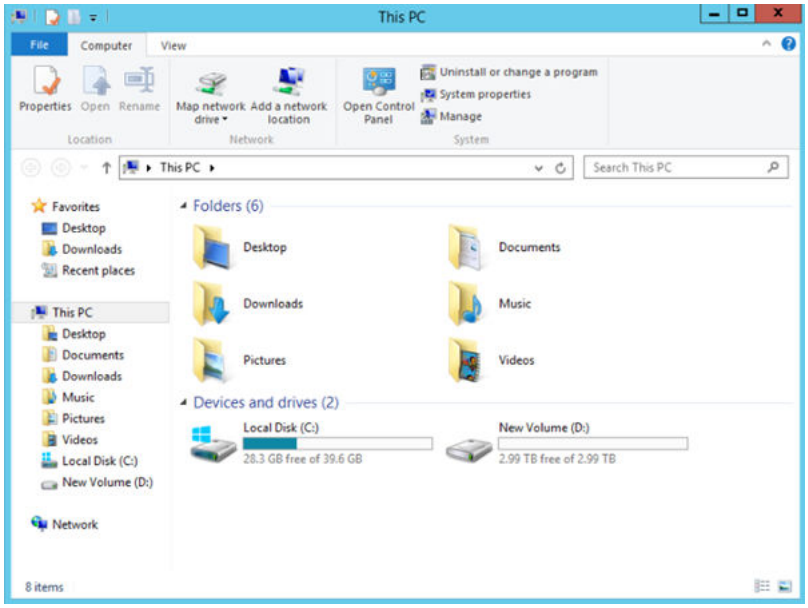
Figure 2-46 Disk initialization succeeded (Windows Server 2012)



Step 13 After the volume is created, click  and check whether a new volume appears in **This PC**. In this example, New Volume (D:) is the new volume.

If New Volume (D:) appears, the disk is successfully initialized and no further action is required.

Figure 2-47 This PC (Windows Server 2012)



----End

2.5.8 Initializing a Data Disk Greater Than 2 TB in Linux (parted)

Scenarios

This section uses CentOS 7.4 64bit to describe how to use parted to initialize a data disk whose capacity is larger than 2 TB. In the following operations, the capacity of the sample disk is 3 TB.

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TB. In Linux, if you choose to use the GPT partition style, the fdisk partitioning tool cannot be used. Use the parted partitioning tool instead. For details about disk partition styles, see **Introduction to Data Disk Initialization Scenarios and Partition Styles**.

The method for initializing a disk varies depending on the OS running on the server. This document is used for reference only. For the detailed operations and differences, see the product documents of the corresponding OS.

Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.

Creating and Mounting a Partition

The following example shows you how a new partition can be created on a new data disk that has been attached to a server. The partition will be created using parted, and GPT is used as the partition style. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured automatic mounting at system start.

Step 1 Run the following command to query information about the new data disk:

lsblk

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├─vda1 253:1    0   1G  0 part /boot
└─vda2 253:2    0  39G  0 part /
vdb    253:16   0   3T  0 disk
```

In the command output, the server contains two disks. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk.

Step 2 Run the following command to enter parted to partition the new data disk:

parted *New data disk*

In this example, run the following command:

parted /dev/vdb

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Step 3 Enter **p** and press **Enter** to view the current disk partition style.

Information similar to the following is displayed:

```
(parted) p
Error: /dev/vdb: unrecognised disk label
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 3299GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted)
```

In the command output, the **Partition Table** value is **unknown**, indicating that no partition style is set for the new disk.

Step 4 Run the following command to set the disk partition style:

mklabel *Disk partition style*

In this example, run the following command to set the disk partition style to GPT: (Disk partition styles can be MBR or GPT.)

mklabel gpt

NOTICE

The maximum disk capacity supported by MBR is 2 TB, and that supported by GPT is 18 EB. Because a data disk currently supports up to 32 TB, use the GPT partition style if your disk capacity is larger than 2 TB.

If you change the disk partition style after the disk has been used, the data on the disk will be cleared. Therefore, select a proper disk partition style when initializing the disk.

Step 5 Enter **p** and press **Enter** to view the disk partition style.

Information similar to the following is displayed:

```
(parted) mklabel gpt
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 3299GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End  Size  File system  Name  Flags

(parted)
```

Step 6 Enter **unit s** and press **Enter** to set the measurement unit of the disk to sector.

Step 7 Run the following command and press **Enter**:

mkpart *Partition name Start sector End sector*

In this example, run the following command:

mkpart opt 2048s 100%

In this example, one partition is created for the new data disk. Value **2048s** indicates the disk start sector, and **100%** indicates the disk end sector. The two values are used for reference only. You can determine the number of partitions and the partition size based on your service requirements.

Information similar to the following is displayed:

```
(parted) mkpart opt 2048s 100%
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
```

If the preceding warning message is displayed, enter **Ignore** to ignore the performance warning.

Step 8 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed:

```
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 6442450944s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End          Size         File system  Name  Flags
 1      2048s  6442448895s  6442446848s               opt
```

Details about the **dev/vdb1** partition are displayed.

Step 9 Enter **q** and press **Enter** to exit parted.

Step 10 Run the following command to view the disk partition information:

lsblk

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├─vda1 253:1    0   1G  0 part /boot
└─vda2 253:2    0  39G  0 part /
vdb    253:16   0   3T  0 disk
└─vdb1 253:17   0   3T  0 part
```

In the command output, **/dev/vdb1** is the partition you created.

Step 11 Run the following command to set the file system format for the new partition:

mkfs -t *File system format* **/dev/vdb1**

In this example, run the following command to set the **ext4** file system for the new partition:

mkfs -t ext4 /dev/vdb1

Information similar to the following is displayed:


```
[root@ecs-centos74 ~]# mkfs -t ext4 /dev/vdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
201326592 inodes, 805305856 blocks
40265292 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2952790016
24576 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000, 214990848, 512000000, 550731776, 644972544

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a period of time. Observe the system running status and do not exit.

NOTICE

The partition sizes supported by file systems vary. Therefore, you are advised to choose an appropriate file system based on your service requirements.

Step 12 Run the following command to create a mount point:

mkdir *Mount point*

In this example, run the following command to create the **/mnt/sdc** mount point:

mkdir /mnt/sdc

Step 13 Run the following command to mount the new partition on the created mount point:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vdb1** on **/mnt/sdc**:

mount /dev/vdb1 /mnt/sdc

Step 14 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda2      ext4       42G  1.5G   38G   4% /
devtmpfs       devtmpfs  2.0G     0  2.0G   0% /dev
tmpfs          tmpfs     2.0G     0  2.0G   0% /dev/shm
tmpfs          tmpfs     2.0G  8.9M  2.0G   1% /run
tmpfs          tmpfs     2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/vda1      ext4       1.1G  153M  801M  17% /boot
```



```
tmpfs          tmpfs     398M     0  398M   0% /run/user/0
/dev/vdb1      ext4      3.3T   93M  3.1T   1% /mnt/sdc
```

New partition **dev/vdb1** is mounted on **/mnt/sdc**.

----End

Setting Automatic Mounting at System Start

Modify the **fstab** file to set automatic disk mounting at server start.

The following procedure shows how to set automatic disk mounting at server start by using UUIDs to identify disks in the **fstab** file. You are advised not to use device names to identify disks in the file because a device name may change (for example, from /dev/vdb1 to /dev/vdb2) during the server stop or start, resulting in improper server running after restart.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

In this example, run the following command to query the UUID of the **/dev/vdb1** partition:

blkid /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# blkid /dev/vdb1
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
```

The UUID of the **/dev/vdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df /mnt/sdc              ext4    defaults        0 2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

Step 6 Perform the following operations to verify the automatic mounting function:

1. Run the following command to unmount the partition:
   umount *Disk partition*
   In this example, run the following command:
   umount /dev/vdb1
2. Run the following command to reload all the content in the **/etc/fstab** file:
   mount -a

3. Run the following command to query the file system mounting information:

   mount | grep *Mount point*

   In this example, run the following command:

   mount | grep /mnt/sdc

   If information similar to the following is displayed, the automatic mounting function takes effect:

   ```
   root@ecs-test-0001 ~]# mount | grep /mnt/sdc
   /dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
   ```

----End

2.6 Obtaining the Domain Name

Step 1 Log in to the management console.

Step 2 Click the username in the upper right corner. In the displayed area, select **My Credential**.

Step 3 In the **Account Information** area, obtain the domain name of the user.

----End

2.7 Obtaining the Region

Step 1 Log in to the management console.

Step 2 Check the information next to the  icon in the upper left corner on the management console. The information there indicates the region.

----End

3 Management

3.1 Storage Pool Management

3.1.1 Expanding a Storage Pool

Scenarios

When your storage pool space is insufficient, you can expand the storage pool capacity.

Applying for Expansion

Step 1 Contact customer service.

To apply for expanding a storage pool, click **here** to send an email or contact **customer service**.

Step 2 Confirm requirements.

Fill out a storage pool **capacity expansion application form** and send it to your sales representative.

Step 3 Expand the capacity.

Customer service will process your application. After the storage pool capacity is expanded, they will inform you through your sales representative or email.

Table 3-1 Parameters in the capacity expansion application form

Parameter	Description	Example Value
Domain Name	Specifies the domain name of the storage pool. For details, see **Obtaining the Domain Name**.	John Snow

Parameter	Description	Example Value
Storage Pool Name	Specifies the storage pool name.	DSS_UXN
ID	Specifies the storage pool ID.	da098bee-2dc2-4bfe-9d21-69bdc580f5ed
Region	Specifies the region where the storage pool belongs. For details, see **Obtaining the Region**.	-
AZ	Specifies the AZ where the storage pool belongs.	-
Capacity (TB)	Specifies the capacity to be expanded.	100 TB

----End

3.1.2 Deleting a Storage Pool

Prerequisites

- The storage pool status is **Available**.
- No disk in the storage pool has been attached.

Applying for Deletion

Step 1 Contact customer service.

To apply for the deletion of a storage pool, click **here** to send an email or contact customer service.

Step 2 Confirm requirements.

Step 3 Fill out a storage pool **deletion application form** and send it to your sales representative.

Step 4 Reclaim resource.

Customer service will delete your storage pool and reclaim resources in the pool. After the storage pool is deleted and resources are reclaimed, they will inform you through your sales representative or email.

Table 3-2 Parameters in the deletion application form

Parameter	Description	Example Value
Domain Name	Specifies the domain name of the storage pool. For details, see section **Obtaining the Domain Name**.	John Snow
Region	Specifies the region where the storage pool belongs. For details, see section **Obtaining the Region**.	-
AZ	Specifies the AZ where the storage pool belongs.	-
Name	Specifies the storage pool name.	dss_01
ID	Specifies the storage pool ID.	da098bee-2dc2-4bfe-9d21-69bdc580f5ed

----End

3.2 Disk Management

3.2.1 Detaching a Disk

3.2.1.1 Detaching a System Disk

Scenarios

A system disk can only be detached offline, that is, its server must be in the **Stopped** state before the system disk is detached. Therefore, you need to first stop the server and then detach the system disk.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Computing**, click **Elastic Cloud Server**.

The **Elastic Cloud Server** page is displayed.

Step 3 In the server list, locate the row that contains the server whose system disk is to be detached, click **More** in the **Operation** column, and choose **Stop**.

When the server status changes to **Stopped**, the server has been stopped.

Step 4 Click the name of this server.

The server details page is displayed.

Step 5 Click the **Disks** tab to view the system disk attached to the server.

Step 6 Locate the row that contains the system disk and click **Detach**.

The **Detach Disk** dialog box is displayed.

Step 7 Click **OK** to detach the disk.

After the operation succeeded, the detached system disk is no longer displayed under the **Disks** tab.

----End

3.2.1.2 Detaching a Data Disk

Scenarios

Data disks can be detached online or offline, which means that the server containing the to-be-detached data disk can either be in the **Running** or **Stopped** state.

- ECS

  Detach a disk online. For details, see **Storage** > **Detaching an EVS Disk from a Running ECS** in the *Elastic Cloud Server User Guide*.

- BMS

  Currently, SCSI disks can be attached to BMSs and used as data disks. You can detach a data disk either from a running or stopped BMS.

For a data disk attached to a server, the disk function is displayed as **Data disk**, and the disk status is displayed as **In-use** in the disk list. After the data disk has been detached from the server, the disk function remains unchanged, the disk status changes to **Available** for a non-shared data disk, and the disk status changes to **Available** for a shared data disk after it is detached from all its servers.

Detaching a Non-shared Disk

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 Determine whether to view the server information before detaching the disk.

- If you need to view the server information, perform the following procedure:
  a. In the disk list, click the name of the to-be-detached disk.

     The disk details page is displayed.

  b. Click the **Attachments** tab to view the servers where the target disk has been attached.

  c. Click to select the server and click **Detach Disk**.

The **Detach Disk** dialog box is displayed.

d. Click **OK** to detach the disk.

- If you do not need to view the server information, perform the following procedure:

a. In the disk list, locate the row that contains the target disk and click **Detach** in the **Operation** column.

The **Detach Disk** dialog box is displayed.

b. Click **OK** to detach the disk.

The disk list is displayed. The disk status is **Detaching**, indicating that the disk is being detached from the server.

When the status changes to **Available**, the disk is successfully detached.

----End

Detaching a Shared Disk

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 Determine whether to view the server information before detaching the disk.

- If you need to view the server information, perform the following procedure:

a. In the disk list, click the name of the to-be-detached disk.

The disk details page is displayed.

b. Click the **Attachments** tab to view the servers where the target disk has been attached.

c. Click to select the servers and click **Detach Disk**.

Shared disks support batch detachment so that you can select multiple servers at a time.

The **Detach Disk** dialog box is displayed.

d. Click **OK** to detach the disk.

- If you do not need to view the server information, perform the following procedure:

a. In the disk list, locate the row that contains the target disk and click **Detach** in the **Operation** column.

The **Detach Disk** dialog box is displayed.

b. Click to select the servers.

Shared disks support batch detachment so that you can select multiple servers at a time.

c. Click **OK** to detach the disk.

The disk list is displayed. The disk status is **Detaching**, indicating that the disk is being detached from the server.

If the shared EVS disk has been attached to multiple servers and needs to be detached from only some of its servers, the disk status will go back to **In-use** after the disk has been detached from the target servers. The disk status changes to **Available** only when it has been detached from all the servers.

----End

3.2.2 Deleting a Disk

Scenarios

If a DSS disk is no longer used, you can release the virtual resources by deleting the disk from the system.

- Before deleting a disk, ensure that the disk status is **Available**, **Error**, **Expansion failed**, **Restoration failed**, or **Rollback failed**.
- Before you delete a shared disk, ensure that the disk has been detached from all its servers.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 In the disk list, locate the row that contains the target disk and choose **More** > **Delete**.

Step 4 (Optional) If multiple disks are to be deleted, select in front of each disk and click **Delete** in the upper area of the list.

Step 5 In the displayed dialog box, confirm the information and click **OK**.

----End

3.2.3 Expanding the Capacity of a Disk

3.2.3.1 Introduction to Expansion Scenarios

What Is Capacity Expansion?

If the capacity of an existing disk is insufficient, you can expand the disk capacity.

Both system disks and data disks can be expanded. Currently, disk capacities can be expanded only. Capacity reduction is not supported.

How to Expand the Disk Capacity?

You can expand the disk capacities when the disks are in the **In-use** or **Available** state.

- Expanding an In-use disk means that the to-be-expanded disk has been attached to a server. Currently, only some server OSs support the expansion of In-use disks. Therefore, ensure that your server OS meets the requirements before you expand an In-use disk. For details, see **Expanding an In-use Disk**.
- Expanding an Available disk means that the to-be-expanded disk has not been attached to any server. For details, see **Expanding an Available Disk**.

NOTICE

When a disk has been put to use, you must check the disk partition style before expanding its capacity. The details are as follows:

- If the MBR partition style is used, the maximum disk capacity supported is 2 TB (2048 GB), and the disk space exceeding 2 TB cannot be allocated and used.
- If the GPT partition style is used, the maximum disk capacity supported is 18 EB (19327352832 GB). A data disk supports up to 32 TB (32768 GB) so that you can expand the capacity of a data disk to up to 32 TB when GPT is used.

If the in-use partition style is MBR and the disk capacity needs to be expanded to over 2 TB, change the partition style from MBR to GPT. Ensure that the disk data has been backed up before changing the partition style because services will be interrupted and data on the disk will be cleared during this change.

3.2.3.2 Expanding an In-use Disk

Scenarios

Currently, disk capacities can be expanded, but cannot be reduced.

Expanding an In-use disk means that the to-be-expanded disk has been attached to a server.

- During such an expansion, the server containing the to-be-expanded disk must be in the **Running** or **Stopped** state.
- Shared disks must be expanded when they are in the **Available** state. For details, see **Expanding an Available Disk**.
- Currently, only some server OSs support the expansion of In-use disks. Therefore, ensure that your server OS meets the requirements for expanding In-use disks before operation. **Table 3-3** lists the server OSs, including the OS images listed on the **Public Images** page of the IMS console and others, that support In-use disk expansion.

  If your server OS does not support capacity expansion of In-use disks, detach the disk and then expand the disk capacity. Otherwise, you may need to stop the server and then start it to make the expansion takes effect.

Table 3-3 Supported OSs

OS	Version
CentOS	7.4 64bit

OS	Version
	7.3 64bit
	7.2 64bit
	6.8 64bit
	6.7 64bit
	6.5 64bit
Debian	8.6.0 64bit
	8.5.0 64bit
Fedora	25 64bit
	24 64bit
SUSE	SUSE Linux Enterprise Server 12 SP2 64bit
	SUSE Linux Enterprise Server 12 SP1 64bit
	SUSE Linux Enterprise Server 11 SP4 64bit
	SUSE Linux Enterprise Server 12 64bit
OpenSUSE	42.2 64bit
	42.1 64bit
Oracle Linux Server release	7.3 64bit
	7.2 64bit
	6.8 64bit
	6.7 64bit
Ubuntu Server	16.04 64bit
	14.04 64bit
	14.04.4 64bit
Windows	Windows Server 2008 R2 Enterprise 64bit
	Windows Server 2012 R2 Standard 64bit
	Windows Server 2016 Standard 64bit
Redhat Linux Enterprise	7.3 64bit
	6.8 64bit

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 Determine whether to view the server information before expanding the disk.

- If you need to view the server information, perform the following procedure:
  a. In the disk list, click the name of the to-be-expanded disk.
     The disk details page is displayed.
  b. Click the **Attachments** tab to view the server where the target disk has been attached.
  c. Click **Expand Capacity** in the upper right corner of the page.
     The expansion page is displayed.
- If you do not need to view the server information, perform the following procedure:
  a. In the disk list, locate the row that contains the target disk and choose **More** > **Expand Capacity** in the **Operation** column.
     The expansion page is displayed.

Step 4 Set the **Add Capacity (GB)** parameter as prompted and click **Next**.

Step 5 On the **Details** page, check the disk information again.

- If you do not need to modify the specifications, click **Submit** to start the expansion.
- If you need to modify the specifications, click **Previous** to modify parameters.

After the specifications are submitted, the disk list page is displayed.

Step 6 In the disk list, view the capacity of the expanded disk.

When the disk status changes from **Expanding** to **In-use**, the additional capacity has been added to the disk and the expansion succeeded.

NOTE

If the expansion fails, customer service personnel will contact you and help you handle this error. Do not perform any operations on the disk before the customer service personnel contact you. If you require that the error be handled as soon as possible, contact our customer service personnel. The disk will no longer be charged once its status changes to **Expansion failed**.

Step 7 After the capacity expansion has succeeded, perform subsequent operations for the additional space.

The operation method varies depending on the server OS.

- For Windows OSs, see **Performing Post-Expansion Operations for a Windows Disk**.
- For Linux OSs,

- If the fdisk partitioning tool will be used, see **Performing Post-Expansion Operations for a Disk in Linux (fdisk)**, **Performing Post-Expansion Operations for a SCSI Data Disk in Linux (fdisk)**, or **Performing Post-Expansion Operations for a System Disk in Linux (fdisk)**.
- If the parted partitioning tool will be used, see **Performing Post-Expansion Operations for a Disk in Linux (parted)**.

----End

3.2.3.3 Expanding an Available Disk

Scenarios

Currently, disk capacities can be expanded, but cannot be reduced.

Expanding an **Available** disk means that the to-be-expanded disk has not been attached to a server.

Shared disks cannot be expanded when they are in the **In-use** state. Therefore, you need to detach the disk from all its servers and then perform the expansion. Once the disk status changes to **Available**, you can expand the disk according to the operations provided in this topic.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 (Optional) If the to-be-expanded disk has been attached to a server, detach it first. For details, see **Detaching a Disk**.

When the disk status changes to **Available**, the disk is successfully detached.

Step 4 In the disk list, locate the row that contains the target disk and choose **More** > **Expand Capacity** in the **Operation** column.

The expansion page is displayed.

Step 5 Set the **Add Capacity (GB)** parameter as prompted and click **Next**.

Step 6 On the **Details** page, check the disk information again.

- If you do not need to modify the specifications, click **Submit** to start the expansion.
- If you need to modify the specifications, click **Previous** to modify parameters.

After the specifications are submitted, the disk list page is displayed.

Step 7 In the disk list, view the capacity of the expanded disk.

When the disk status changes from **Expanding** to **Available**, the additional capacity has been added to the disk and the expansion succeeded.

NOTE

If the expansion fails, customer service personnel will contact you and help you handle this error. Do not perform any operations on the disk before the customer service personnel contact you. If you require that the error be handled as soon as possible, contact our customer service personnel. The disk will no longer be charged once its status changes to **Expansion failed**.

Step 8 Attach the expanded disk to a server. For details, see **Step 3: Attach a Disk**.

Step 9 After the capacity expansion has succeeded, perform subsequent operations for the additional space.

The operation method varies depending on the server OS.

- For Windows OSs, see **Performing Post-Expansion Operations for a Windows Disk**.
- For Linux OSs,
  - If the fdisk partitioning tool will be used, see **Performing Post-Expansion Operations for a Disk in Linux (fdisk)**, **Performing Post-Expansion Operations for a SCSI Data Disk in Linux (fdisk)**, or **Performing Post-Expansion Operations for a System Disk in Linux (fdisk)**.
  - If the parted partitioning tool will be used, see **Performing Post-Expansion Operations for a Disk in Linux (parted)**.

----End

3.2.3.4 Performing Post-Expansion Operations for a Windows Disk

Scenarios

After expanding the disk capacity on the management console, you need to log in to the Windows server to allocate the additional space to an existing volume or a new volume.

This topic uses Windows Server 2008 R2 Enterprise as the sample OS and provides the following capacity expansion methods:

- System disk: Add the additional space to volume (C:).
- Data disk: If volume (D:) already exists, add the additional space to volume (D:).

  If you want to create a new volume such as volume (E:), expanding the disk is not the right choice. You need to create a new disk, attach it to server, and initialize it for use.

The method for allocating the additional space varies depending on the server OS. This document is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

NOTICE

Performing the expansion operations with caution. Misoperation may lead to data loss or exceptions. Therefore, you are advised to use CBR to back up the disk data before expansion. For details, see **Managing a Backup**.

Prerequisites

- You have logged in to a server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.
- You have attached the disk to the server, and the additional space has not been allocated.

System Disk: Add Additional Space to Volume (C:)

In this example, the system disk has 50 GB originally, and 22 GB is added on the management console. The following procedure describes how to add this 22 GB to volume (C:) on the server. After the operation is complete, volume (C:) will have 72 GB of capacity and can be used as a system volume.

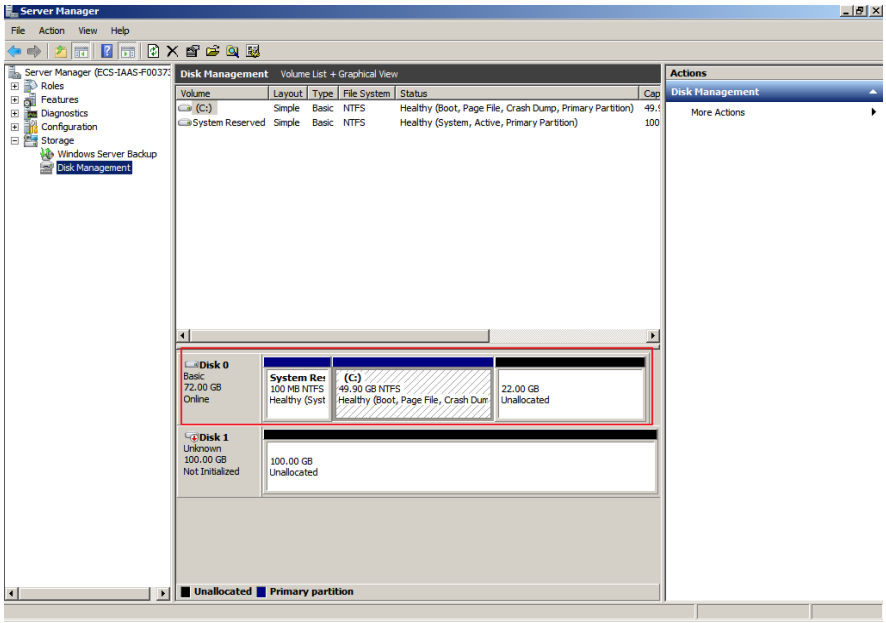
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

Step 2 In the navigation tree, choose **Storage** > **Disk Management**.

The **Disk Management** window is displayed.

Figure 3-1 Disk Management (system disk)



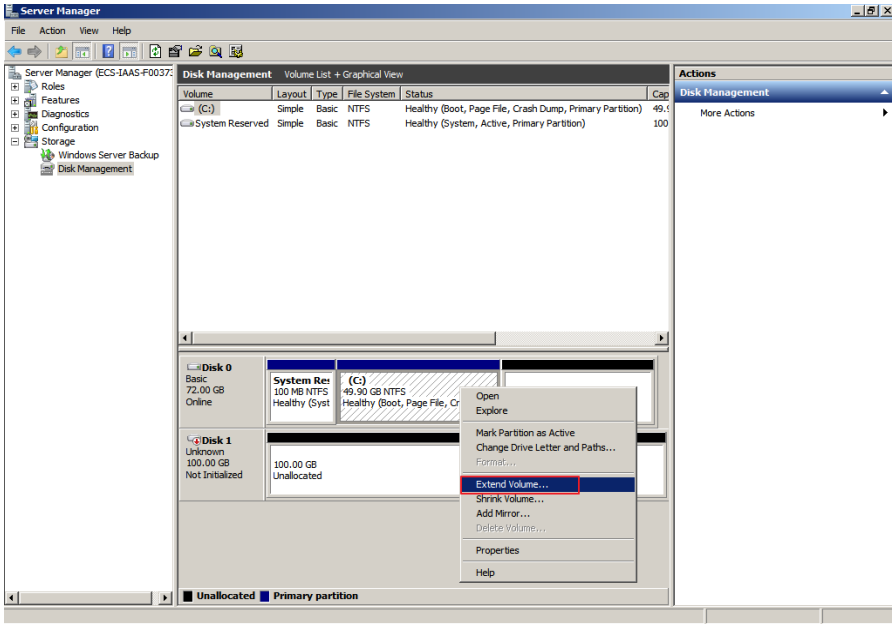
NOTE

If you cannot view the additional space, right-click **Disk Management** and choose **Refresh** from the shortcut menu.

Step 3 On the **Disk Management** page, select the disk and partition that needs to be extended. The current partition size and unallocated disk space are displayed.

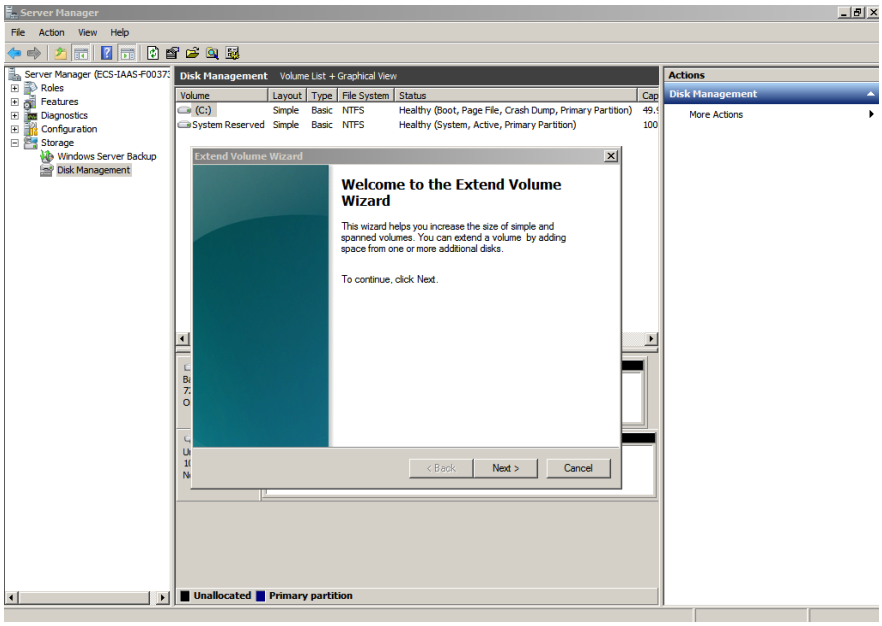
Step 4 Right-click the selected disk and choose **Extend Volume**.

Figure 3-2 Choosing **Extend Volume**



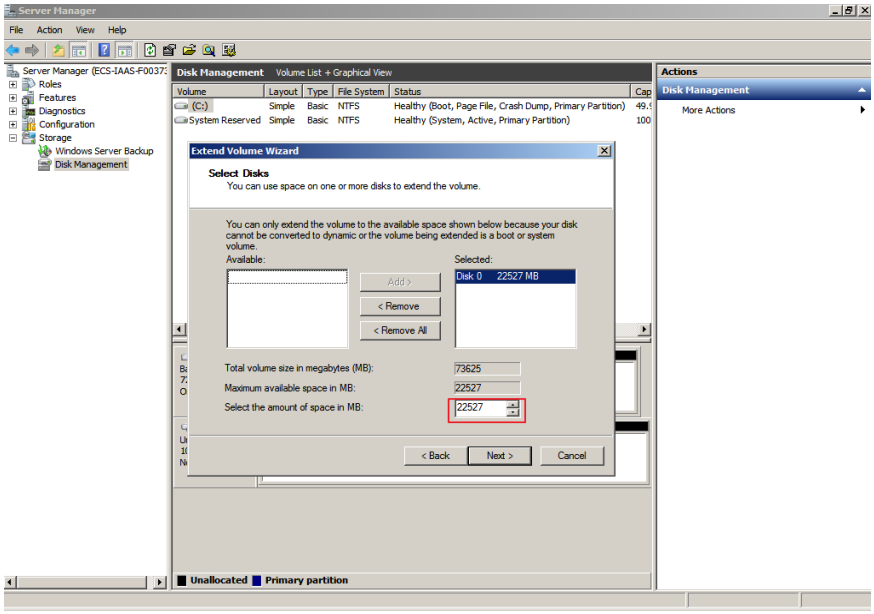
Step 5 On the displayed **Extend Volume Wizard** page, click **Next**.

Figure 3-3 Extend Volume Wizard



Step 6 In the text box to the right of **Select the amount of space in MB** shown in **Figure 3-4**, enter the amount of the additional capacity and click **Next**.

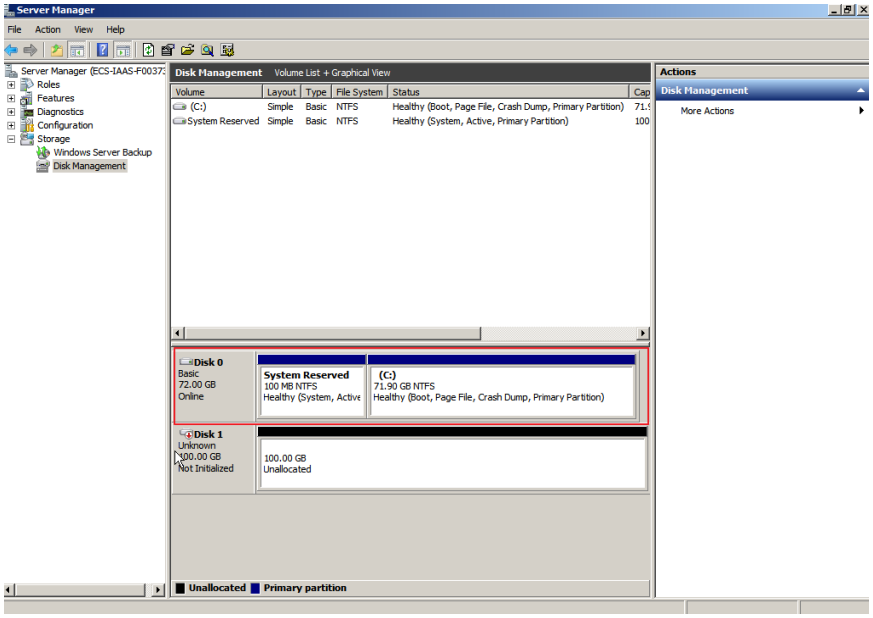
Figure 3-4 Selecting space



Step 7 Click **Finish** to complete the wizard.

After the expansion has succeeded, the disk capacity is greater than the original capacity.

Figure 3-5 Successful capacity expansion



----End

Data Disk: Add Additional Space to Volume (D:)

In this example, the data disk has 100 GB originally, and 50 GB is added on the management console. The following procedure describes how to add this 50 GB to volume (D:) on the server. After the operation is complete, volume (D:) has 150 GB of capacity and can be used as a data volume.

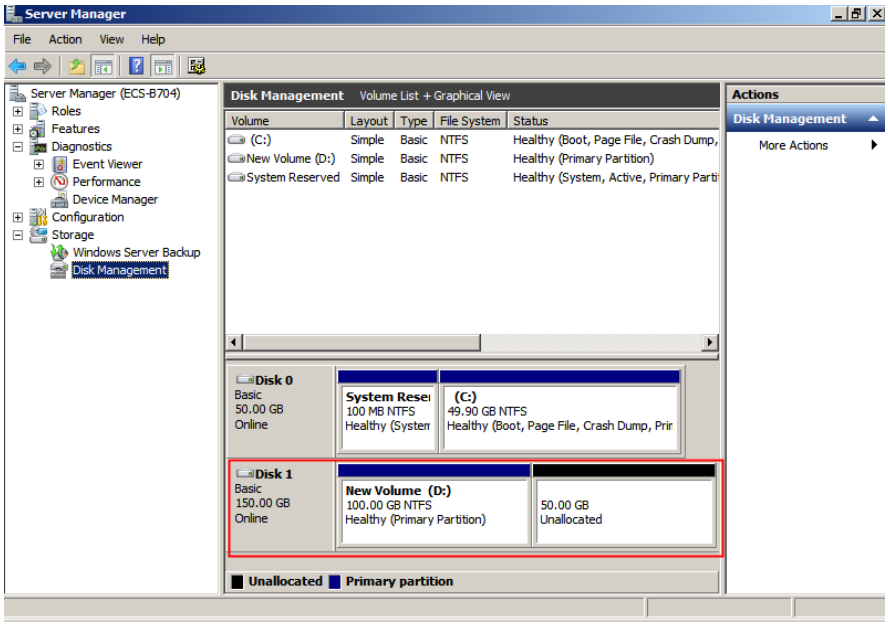
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

Step 2 In the navigation tree, choose **Storage** > **Disk Management**.

The **Disk Management** window is displayed.

Figure 3-6 Disk Management (data disk)



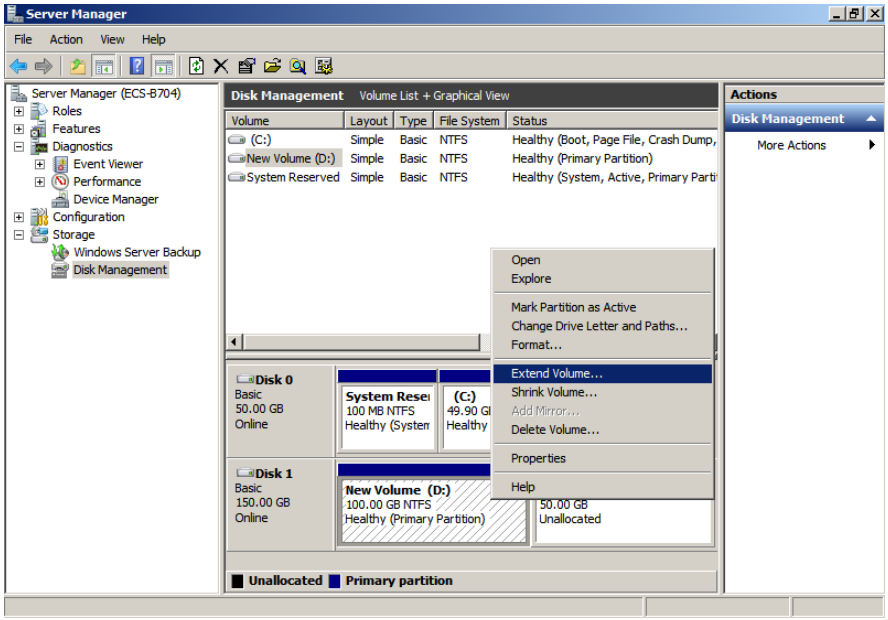
NOTE

If you cannot view the additional space, right-click **Disk Management** and choose **Refresh** from the shortcut menu.

Step 3 On the **Disk Management** page, select the disk and partition that needs to be extended. The current partition size and unallocated disk space are displayed.

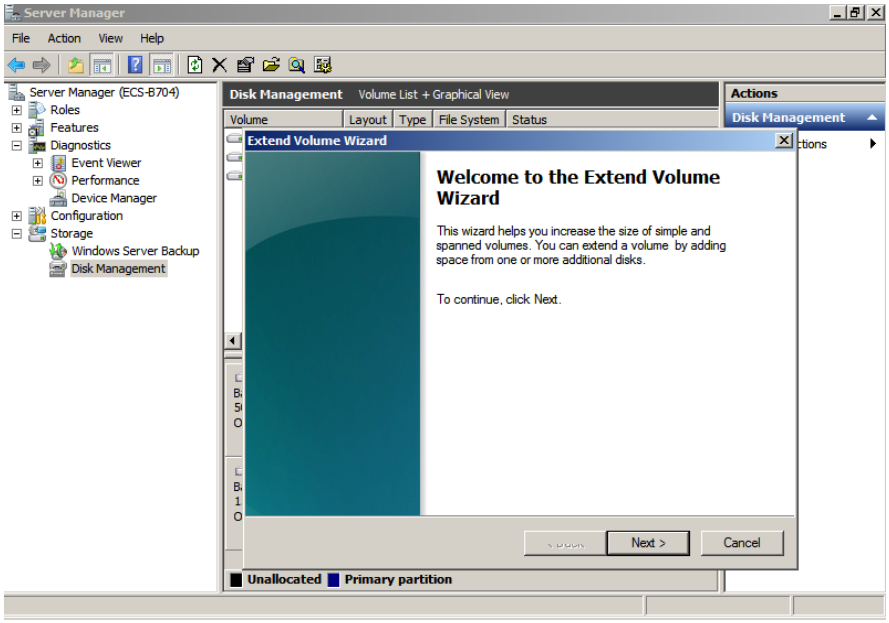
Step 4 Right-click the selected disk and choose **Extend Volume**.

Figure 3-7 Choosing Extend Volume (Windows Server 2008)



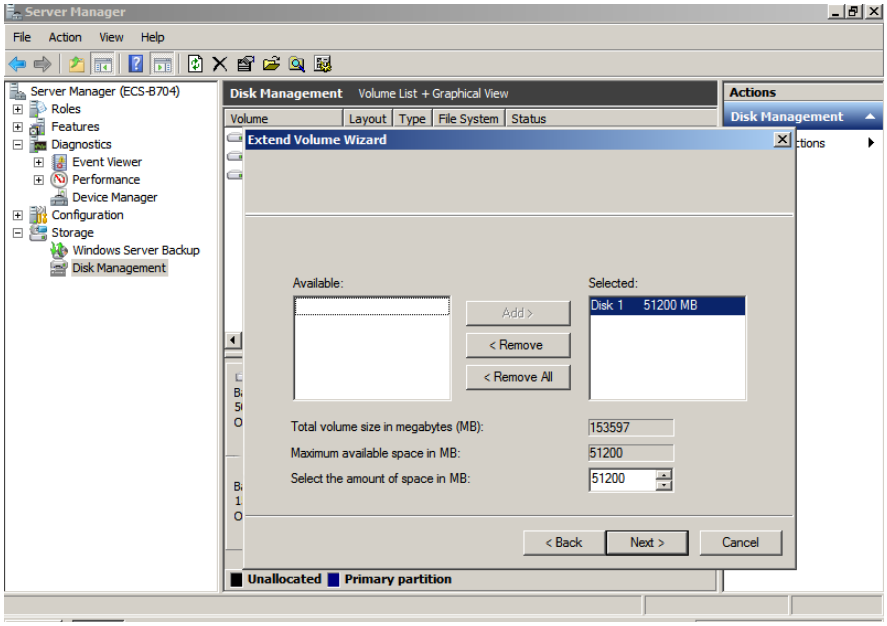
Step 5 On the displayed **Extend Volume Wizard** page, click **Next**.

Figure 3-8 Extend Volume Wizard (Windows Server 2008)



Step 6 In the text box to the right of **Select the amount of space in MB** shown in **Figure 3-9**, enter the amount of the additional capacity and click **Next**.

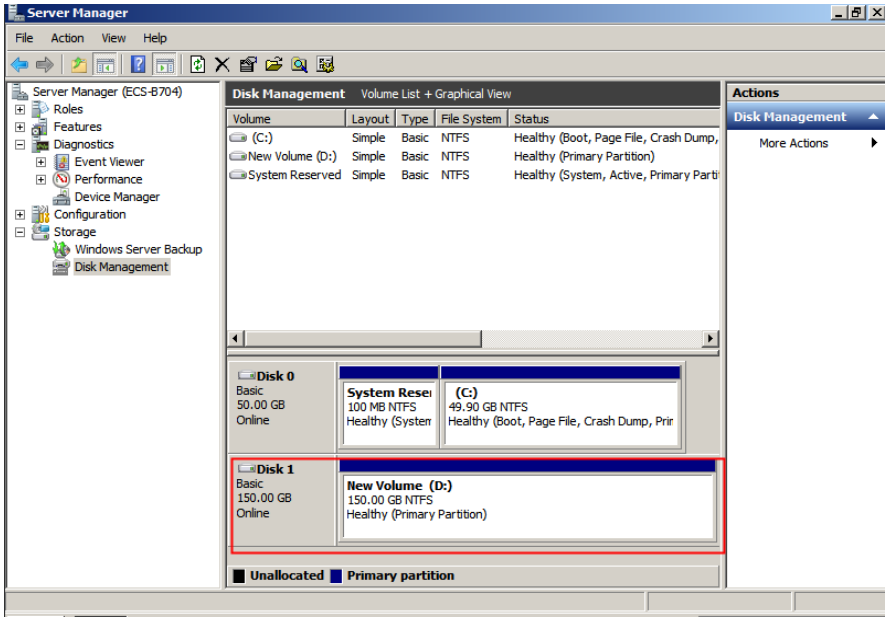
Figure 3-9 Selecting space (Windows Server 2008)



Step 7 Click **Finish** to complete the wizard.

After the expansion has succeeded, the disk capacity is greater than the original capacity.

Figure 3-10 Capacity expansion succeeded (Windows Server 2008)



----End

3.2.3.5 Performing Post-Expansion Operations for a Disk in Linux (fdisk)

Scenarios

In Linux, after the capacity expansion succeeded, the additional disk space needs to be allocated to an existing partition or a new partition.

This topic uses CentOS 7.0 64bit to describe how to allocate the additional disk space to a partition using fdisk.

The method for allocating the additional space varies depending on the server OS. This document is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Based on your service requirements and disk condition, you can choose either of the following ways to allocate the additional disk space:

- Create a new partition (services will not be interrupted).

  Creating a new partition after expansion does not require the original partitions to be unmounted. Therefore, the impacts on services are minor than expanding an existing partition. This method is recommended for system disks or disks carrying services that cannot be interrupted.

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.

- Expand an existing partition (services will be interrupted).

  If the MBR partition style is used and the number of partitions has reached the upper limit, you can only allocate the additional space to an existing partition. Expanding an existing partition does not delete its data, but requires the partition to unmount. Therefore, services will be interrupted.

  If the MBR partition style is used and the disk capacity after expansion will exceed 2 TB, the space exceeding 2 TB cannot be used. To make use of that space, change the disk partition style from MBR to GPT. Data on the disk will be cleared during such a change. Therefore, back up the disk data before changing the partition style.

NOTICE

Performing the expansion operations with caution. Misoperation may lead to data loss or exceptions. Therefore, you are advised to use CBR to back up the disk data before expansion. For details, see **Managing a Backup**.

Prerequisites

- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.
- You have attached the disk to the server, and the additional space has not been allocated.

Checking File Systems on the To-be-expanded Disk

Before expanding the disk capacity, check whether the file systems on the disk can be properly mounted.

Step 1 (Optional) If there is an unmounted partition, run the following command to mount the partition on the specified directory:

mount *Disk partition Mount point*

Example command:

mount /dev/xvdb1 /mnt/sdc

If the system returns a mount error, check whether the file system is correctly created. For example, a user may create the file system for the **/dev/xvdb** disk instead of the **/dev/xvdb1** partition. In this case, what mounted is actually the **/dev/xvdb** disk, but not the **/dev/xvdb1** partition.

Step 2 Run the following command to view the disk mounting information:

df -TH

Information similar to the following is displayed:

```
[root@ecs-b656 test]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda2     xfs        11G  7.4G  3.2G  71% /
devtmpfs       devtmpfs  4.1G     0  4.1G   0% /dev
tmpfs          tmpfs     4.1G   82k  4.1G   1% /dev/shm
tmpfs          tmpfs     4.1G  9.2M  4.1G   1% /run
tmpfs          tmpfs     4.1G     0  4.1G   0% /sys/fs/cgroup
/dev/xvda3     xfs       1.1G   39M  1.1G   4% /home
/dev/xvda1     xfs       1.1G  131M  915M  13% /boot
/dev/xvdb1     ext4       11G   38M  9.9G   1% /mnt/sdc
```

In the command output, the file system of the **/dev/xvdb1** partition is **ext4**, and the partition is mounted on **/mnt/sdc**.

Step 3 Run the following command to switch to the mounting directory to view the files on the disk:

ll *Mounting directory*

Example command:

ll /mnt/sdc

If the files on the disk can be viewed, the disk is normal and can be expanded.

----End

Viewing the Partition Style

Before allocating the additional space, query the current disk partition style. If MBR is used, you can use either the fdisk or parted partitioning tool. If GPT is used, you must use the parted partitioning tool.

Step 1 Run the following command to view the current disk partition style:

fdisk -l

Information similar to the following is displayed:


```
[root@ecs-1120 linux]# fdisk -l

Disk /dev/xvda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000c5712

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1            2048    83886079    41942016   83  Linux
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own
discretion.

Disk /dev/xvdb: 161.1 GB, 161061273600 bytes, 314572800 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt


#         Start          End     Size  Type            Name
 1          34    209715166    100G  Microsoft basic opt
 2   209715167    314572766     50G  Microsoft basic opt1
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own
discretion.

Disk /dev/xvdc: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt


#         Start          End     Size  Type            Name
 1          34     16777215      8G  Microsoft basic opt
 2    16777216     83884031     32G  Microsoft basic opt
```

In the command output, parameter **Disk label type** indicates the disk partition style. Value **dos** indicates the MBR partition style, and value **gpt** indicates the GPT partition style.

----End

Creating a New Partition

The following example shows you how to make use of the additional space of a system disk by creating a new partition and mount the partition on **/opt** without interrupting services.

Step 1 Run the following command to view the disk partition information:

fdisk -l

Information similar to the following is displayed: (**/dev/xvda** is the system disk.)

```
[root@ecs-bab9 test]# fdisk -l

Disk /dev/xvda: 64.4 GB, 64424509440 bytes, 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000cc4ad

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *        2048     2050047     1024000   83  Linux
```



```
/dev/xvda2       2050048   22530047   10240000  83  Linux
/dev/xvda3      22530048   24578047    1024000  83  Linux
/dev/xvda4      24578048   83886079   29654016   5  Extended
/dev/xvda5      24580096   26628095    1024000  82  Linux swap / Solaris
```

Step 2 Run the following command to enter fdisk:

fdisk /dev/vda

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# fdisk /dev/vda
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

Step 3 Enter **n** and press **Enter** to create a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
All primary partitions are in use
Adding logical partition 6
First sector (26630144-83886079, default 26630144):
```

NOTE

If the MBR partition style is used, a maximum of 4 primary partitions, or 3 primary partitions and 1 extended partition can be created. The extended partition cannot be used directly and must be divided into logical partitions before use.

In this example, the maximum number of primary partitions has been reached for the system disk, and the disk already has 5 partitions (3 primary partitions and 2 logical partitions). Therefore, the system automatically creates a sixth logical partition in the extended partition.

To view the example operations in case that the maximum number of primary partitions is not reached for the system disk, see **Performing Post-Expansion Operations for a System Disk in Linux (fdisk)**.

Step 4 Enter the new partition's start sector, for example the default value, and press **Enter**.

The start sector must be greater than end sector of the existing partition.

Information similar to the following is displayed:

```
First sector (26630144-83886079, default 26630144):
Using default value 26630144
Last sector, +sectors or +size{K,M,G} (26630144-83886079, default 83886079):
```

Step 5 Enter the new partition's end sector and press **Enter**.

The default end sector is used in this example.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (26630144-83886079, default 83886079):
Using default value 83886079
Partition 6 of type Linux and of size 27.3 GiB is set

Command (m for help):
```

Step 6 Enter **p** and press **Enter** to view the new partition.

Information similar to the following is displayed:

```
Disk /dev/xvda: 64.4 GB, 64424509440 bytes, 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
```



```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000cc4ad

Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *        2048     2050047     1024000   83  Linux
/dev/xvda2         2050048    22530047    10240000   83  Linux
/dev/xvda3        22530048    24578047     1024000   83  Linux
/dev/xvda4        24578048    83886079    29654016    5  Extended
/dev/xvda5        24580096    26628095     1024000   82  Linux swap / Solaris
/dev/xvda6        26630144    83886079    28627968   83  Linux

Command (m for help):
```

Step 7 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

The partition is created.

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q**.

Step 8 Run the following command to synchronize the new partition table to the OS:

partprobe

Step 9 Run the following command to set the file system format for the new partition:

(The ext4 file system is used in this example.)

mkfs -t ext4 /dev/xvda6

NOTE

The procedure for setting the **xfs** file system is the same as that for the **ext3** or **ext4** file system. The command for creating the **xfs** file system is **mkfs -t xfs /dev/xvda6**.

Information similar to the following is displayed:

```
[root@ecs-bab9 test]# mkfs -t ext4 /dev/xvda6
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1790544 inodes, 7156992 blocks
357849 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2155872256
219 block groups
32768 blocks per group, 32768 fragments per group
8176 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000
```



```
Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a while, and you need to observe the system running status. Once **done** is displayed in the command output, the formatting is complete.

Step 10 Run the following command to mount the new partition on a space-demanding directory, for example **/opt**:

mount /dev/xvda6 /opt

Information similar to the following is displayed:

```
[root@ecs-bab9 test]# mount /dev/xvda6 /opt
[root@ecs-bab9 test]#
```

NOTE

If the new partition is mounted on a directory that is not empty, the subdirectories and files in the directory will be hidden. Therefore, you are advised to mount the new partition on an empty directory or a new directory. If the new partition must be mounted on a directory that is not empty, move the subdirectories and files in this directory to another directory temporarily. After the partition is successfully mounted, move the subdirectories and files back.

Step 11 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-bab9 test]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda2     xfs        11G  7.4G  3.2G  71% /
devtmpfs       devtmpfs  4.1G     0  4.1G   0% /dev
tmpfs          tmpfs     4.1G   82k  4.1G   1% /dev/shm
tmpfs          tmpfs     4.1G  9.2M  4.1G   1% /run
tmpfs          tmpfs     4.1G     0  4.1G   0% /sys/fs/cgroup
/dev/xvda3     xfs       1.1G   39M  1.1G   4% /home
/dev/xvda1     xfs       1.1G  131M  915M  13% /boot
/dev/xvda6     ext4       29G   47M   28G   1% /opt
```

----End

Expanding an Existing Partition

The following example shows you how to make use of the additional space of a disk attached to a server by recreating the **/dev/xvdb1** partition and mounting the partition on **/mnt/sdc**. During this process, services will be interrupted.

NOTICE

During an expansion, the additional space is added to the end of the disk. When the disk has multiple partitions, only the partition at the end of this disk can be expanded.

Step 1 Run the following command to view the disk partition information:

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-b656 test]# fdisk -l

Disk /dev/xvda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000cc4ad

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1   *        2048     2050047     1024000   83  Linux
/dev/xvda2         2050048    22530047    10240000   83  Linux
/dev/xvda3        22530048    24578047     1024000   83  Linux
/dev/xvda4        24578048    83886079    29654016    5  Extended
/dev/xvda5        24580096    26628095     1024000   82  Linux swap / Solaris

Disk /dev/xvdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xb00005bd

Device Boot      Start         End      Blocks   Id  System
/dev/xvdb1          2048    20971519    10484736   83  Linux
```

In the command output, the total capacity of the **/dev/xvdb** disk is 21.5 GB, and the disk has only one partition, **dev/xvdb1**. The partition's start sector is **2048**, and its end sector is **20971519**.

View the **/dev/xvdb** capacity and check whether the additional space is included.

- If the additional space is not included, refresh the capacity according to **Performing Post-Expansion Operations for a SCSI Data Disk in Linux (fdisk)**.
- If the additional space is included, take note of the start and end sectors of the **dev/xvdb1** partition and then go to **Step 2**. These values will be used in the follow-up operations.

Step 2 Run the following command to unmount the partition:

umount /mnt/sdc

Step 3 Run the following command to enter fdisk and enter **d** to delete the **/dev/xvdb1** partition:

fdisk /dev/xvdb

Information similar to the following is displayed:

```
[root@ecs-b656 test]# fdisk /dev/xvdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): d
Selected partition 1
Partition 1 is deleted

Command (m for help):
```

NOTE

After deleting the partition, recreate the partition according to the following steps, and data on this disk will not be lost.

Step 4 Enter **n** and press **Enter** to create a new partition.

Entering **n** creates a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
Partition type:
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
```

There are two types of disk partitions:

- Choosing **p** creates a primary partition.
- Choosing **e** creates an extended partition.

Step 5 Ensure that the entered partition type is the same as the partition had before. In this example, a primary partition is used. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
Select (default p): p
Partition number (1-4, default 1):
```

In the command output, **Partition number** specifies the primary partition number.

Step 6 Ensure that the entered partition number is the same as the partition had before. In this example, partition number **1** is used. Therefore, enter **1** and press **Enter**.

Information similar to the following is displayed:

```
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048):
```

In the command output, **First sector** specifies the start sector.

NOTE

Data will be lost if the following operations are performed:

- Select a start sector other than the partition had before.
- Select an end sector smaller than the partition had before.

Step 7 Ensure that the entered start sector is the same as the partition had before. In this example, start sector **2048** is recorded in **Step 1**. Therefore, enter **2048** and press **Enter**.

Information similar to the following is displayed:

```
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
```

In the command output, **Last sector** specifies the end sector.

Step 8 Ensure that the entered end sector is greater than or equal to the end sector recorded in **Step 1**. In this example, the recorded end sector is **20971519**, and the default end sector is used. Therefore, enter **41943039** and press **Enter**.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
Using default value 41943039
Partition 1 of type Linux and of size 20 GiB is set
Command (m for help):
```

The partition is successfully created.

Step 9 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed:

```
Command (m for help): p

Disk /dev/xvdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0xb00005bd

   Device Boot      Start         End      Blocks   Id  System
/dev/xvdb1            2048    41943039    20970496   83  Linux

Command (m for help):
```

Details about the **/dev/xvdb1** partition are displayed.

Step 10 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

The partition is created.

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q**.

Step 11 Perform the following operations based on the file system of the disk:

- For the **ext3** or **ext4** file system
  a. Run the following command to check the correctness of the file system on **/dev/xvdb1**:

     e2fsck -f /dev/xvdb1

     Information similar to the following is displayed:

     ```
     [root@ecs-b656 test]# e2fsck -f /dev/xvdb1
     e2fsck 1.42.9 (28-Dec-2013)
     Pass 1: Checking inodes, blocks, and sizes
     Pass 2: Checking directory structure
     Pass 3: Checking directory connectivity
     Pass 4: Checking reference counts
     Pass 5: Checking group summary information
     /dev/xvdb1: 11/655360 files (0.0% non-contiguous), 83137/2621184 blocks
     ```

  b. Run the following command to extend the size of the file system on **/dev/xvdb1**:

     resize2fs /dev/xvdb1

     Information similar to the following is displayed:

     ```
     [root@ecs-b656 test]# resize2fs /dev/xvdb1
     resize2fs 1.42.9 (28-Dec-2013)
     Resizing the filesystem on /dev/xvdb1 to 5242624 (4k) blocks.
     The filesystem on /dev/xvdb1 is now 5242624 blocks long.
     ```

  c. Run the following command to mount the new partition on **/mnt/sdc**:

     mount /dev/xvdb1 /mnt/sdc

- For the **xfs** file system

a. Run the following command to mount the new partition on **/mnt/sdc**:

mount /dev/xvdb1 /mnt/sdc

b. Run the following command to extend the size of the file system on **/dev/xvdb1**:

sudo xfs_growfs /dev/xvdb1

Step 12 Run the following command to view the mount result:

df -TH

----End

Setting Automatic Mounting at System Start

To automatically mount partitions at system starts, do not specify partitions, for example, **/dev/xvdb1**, in **/etc/fstab** because the sequence of cloud devices, and therefore their names may change during the server stop or start. You are advised to use the UUID in **/etc/fstab** to set automatic mounting at system start.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

For example, run the following command to query the UUID of the **/dev/xvdb1** partition:

blkid /dev/xvdb1

Information similar to the following is displayed:

```
[root@ecs-b656 test]# blkid /dev/xvdb1
/dev/xvdb1: UUID="1851e23f-1c57-40ab-86bb-5fc5fc606ffa" TYPE="ext4"
```

The UUID of the **/dev/xvdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc     ext3 defaults    0   2
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc     ext4 defaults    0   2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

----End

3.2.3.6 Performing Post-Expansion Operations for a Disk in Linux (parted)

Scenarios

In Linux, after the capacity expansion succeeded, the additional disk space needs to be allocated to an existing partition or a new partition.

This topic uses EulerOS 2.0 64bit to describe how to allocate the additional disk space to a partition using parted.

The method for allocating the additional space varies depending on the server OS. This document is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Based on your service requirements and disk condition, you can choose either of the following ways to allocate the additional disk space:

- Create a new partition (services will not be interrupted).

  Creating a new partition after expansion does not require the original partitions to be unmounted. Therefore, the impacts on services are minor than expanding an existing partition. This method is recommended for system disks or disks carrying services that cannot be interrupted.

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.

- Expand an existing partition (services will be interrupted).

  If the MBR partition style is used and the number of partitions has reached the upper limit, you can only allocate the additional space to an existing partition. Expanding an existing partition does not delete its data, but requires the partition to unmount. Therefore, services will be interrupted.

  If the MBR partition style is used and the disk capacity after expansion will exceed 2 TB, the space exceeding 2 TB cannot be used. To make use of that space, change the disk partition style from MBR to GPT. Data on the disk will be cleared during such a change. Therefore, back up the disk data before changing the partition style.

NOTICE

Performing the expansion operations with caution. Misoperation may lead to data loss or exceptions. Therefore, you are advised to use CBR to back up the disk data before expansion. For details, see **Managing a Backup**.

Prerequisites

- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.
- You have attached the disk to the server, and the additional space has not been allocated.

Viewing the Partition Style

Before allocating the additional space, query the current disk partition style. If MBR is used, you can use either the fdisk or parted partitioning tool. If GPT is used, you must use the parted partitioning tool.

Step 1 Run the following command to view the disk information:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   40G  0 disk
└─xvda1 202:1    0   40G  0 part /
xvdb    202:16   0  150G  0 disk
├─xvdb1 202:17   0  100G  0 part /mnt/sdc
└─xvdb2 202:18   0   50G  0 part /mnt/opt
xvdc    202:32   0   40G  0 disk
├─xvdc1 202:33   0    8G  0 part
└─xvdc2 202:34   0   32G  0 part
```

Step 2 Run the following command and enter **p** to view the current disk partition style:

parted *Disk name*

For example, run the following command to view the partition style of the **/dev/xvdb** disk:

parted /dev/xvdb

Information similar to the following is displayed:

```
root@ecs-1120 linux]# parted /dev/xvdb
GNU Parted 3.1
Using /dev/xvdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 161GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      17.4kB  107GB   107GB   ext4         opt
 2      107GB   161GB   53.7GB  ext4         opt1
```

In the command output, parameter **Partition Table** indicates the disk partition style. Value **msdos** indicates the MBR partition style, and value **gpt** indicates the GPT partition style.

Step 3 Enter **q** and press **Enter** to exit parted.

Step 4 Check the disk partition style of other disks. For details, see **Step 2** to **Step 3**.

----End

Creating a New Partition

The following example shows you how to make use of the additional space of a system disk by creating a new partition and mount the partition on **/opt** without interrupting services.

Step 1 Run the following command to view the disk partition information:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
└─xvda1 202:1    0   40G  0 part /
xvdb    202:16   0  250G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0   50G  0 part
xvdc    202:32   0   40G  0 disk
├─xvdc1 202:33   0    8G  0 part
└─xvdc2 202:34   0   32G  0 part
```

In the command output, the capacity of the **dev/xvda** system disk is 80 GB, in which the in-use **dev/xvda1** partition takes 40 GB and the additional 40 GB has not been allocated.

Step 2 Run the following command to enter parted to allocate the additional space of the system disk to a partition:

parted *System disk*

In this example, run the following command:

parted /dev/xvda

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# parted /dev/xvda
GNU Parted 3.1
Using /dev/xvda
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

Step 3 Enter **unit s** and press **Enter** to set the measurement unit of the disk to sector.

Step 4 Enter **p** and press **Enter** to view the current disk partition style.

Information similar to the following is displayed:

```
(parted) unit s
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 167772160s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start  End        Size       Type     File system  Flags
 1      2048s  83886079s  83884032s  primary  ext4
```

Step 5 Enter **mkpart** and press **Enter** to create a new partition.

Step 6 In this example, a primary partition is created. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```


Step 7 Set the file system type and size for the new partition.

Value **83886080** indicates the start sector of the **dev/xvda2** partition you created, and value **167772159** indicates the end sector. The two values are used for reference only. You can determine the number of partitions and partition sizes based on your service requirements.

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```

NOTE

The file system type may fail to set in this step. Therefore, reconfigure the file system type according to **Step 10** after the partition is created.

The maximum sector can be obtained in either of the following ways:

- Run the **fdisk -l** command and take note of the value.
- Enter **-1s**, and the value displayed is the desired value.

Step 8 Enter **p** and press **Enter** to view the new partition.

Information similar to the following is displayed:

```
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 167772160s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start      End         Size       Type     File system  Flags
 1      2048s      83886079s   83884032s  primary  ext4
 2      83886080s  167772159s  83886080s  primary
```

The **dev/xvda2** partition is created.

Step 9 Enter **q** and press **Enter** to exit parted.

Step 10 Run the following command to set the file system format for the new partition:

(The ext4 file system is used in this example.)

mkfs -t ext4 /dev/xvda2

NOTE

The procedure for setting the **xfs** file system is the same as that for the **ext3** or **ext4** file system. The command for creating the **xfs** file system is **mkfs -t xfs /dev/xvda2**.

Information similar to the following is displayed:

```
[[root@ecs-1120 linux]# mkfs -t ext4 /dev/xvda2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
```



```
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
?32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
?4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a while, and you need to observe the system running status. Once **done** is displayed in the command output, the formatting is complete.

Step 11 Run the following command to mount the new partition on a space-demanding directory, for example **/opt**:

mount /dev/xvda6 /opt

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# mount /dev/xvda2 /opt
[root@ecs-1120 linux]#
```

NOTE

If the new partition is mounted on a directory that is not empty, the subdirectories and files in the directory will be hidden. Therefore, you are advised to mount the new partition on an empty directory or a new directory. If the new partition must be mounted on a directory that is not empty, move the subdirectories and files in this directory to another directory temporarily. After the partition is successfully mounted, move the subdirectories and files back.

Step 12 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda1     ext4       43G  8.3G   33G  21% /
devtmpfs       devtmpfs  885M     0  885M   0% /dev
tmpfs          tmpfs     894M     0  894M   0% /dev/shm
tmpfs          tmpfs     894M   18M  877M   2% /run
tmpfs          tmpfs     894M     0  894M   0% /sys/fs/cgroup
tmpfs          tmpfs     179M     0  179M   0% /run/user/2000
tmpfs          tmpfs     179M     0  179M   0% /run/user/0
tmpfs          tmpfs     179M     0  179M   0% /run/user/1001
/dev/xvda2     ext4       43G   51M   40G   1% /opt
```

----End

Expanding an Existing Partition

The following example shows you how to make use of the additional space of a disk attached to a server by recreating the **/dev/xvdc1** partition mounted on **/mnt/sdc**. Because the **/dev/xvdc** disk only has one partition, this partition is regarded as the partition at the disk end. During the partition recreation, services will be interrupted.

NOTICE

During an expansion, the additional space is added to the end of the disk. When the disk has multiple partitions, only the partition at the end of this disk can be expanded.

Step 1 Run the following command to view the disk partition information:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 sdc]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0    10G  0 part /mnt/sdc
```

In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the existing partition takes 10 GB. The **/dev/xvdc1** partition is at the end of the disk and has been mounted on **/mnt/sdc**.

View the **/dev/xvdc** capacity and check whether the additional space is included.

- If the additional space is not included, refresh the capacity according to **Performing Post-Expansion Operations for a SCSI Data Disk in Linux (fdisk)**.
- If the additional space is included, go to **Step 2**.

Step 2 Run the following command to unmount the partition:

umount /mnt/sdc

Step 3 Run the following command to view the unmount result:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# umount /mnt/sdc
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0    10G  0 part
```

Step 4 Run the following command to enter parted to allocate the additional space of the data disk to a partition:

parted *Data disk*

In this example, run the following command:

parted /dev/xvdc

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# parted /dev/xvdc
GNU Parted 3.1
Using /dev/xvdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

Step 5 Enter **unit s** and press **Enter** to set the measurement unit of the disk to sector.

Step 6 In this example, a primary partition is created. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```

Step 7 The **/dev/xvdc1** partition number is **1**. Therefore, enter **rm 1** and press **Enter** to delete the partition.

Step 8 Enter **p** and press **Enter** to check whether the **/dev/xvdc1** partition has been deleted.

Information similar to the following is displayed:

```
(parted) rm 1
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdc: 125829120s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End        Size       File system  Name  Flags
```

Step 9 Enter **mkpart opt** *2048s 125829119* and press **Enter** to recreate the partition.

2048 specifies the start sector recorded in **Step 6**, and *125829119* specifies the end sector, which must be greater than or equal to the end sector recorded in **Step 6**.

Information similar to the following is displayed:

```
(parted) mkpart opt 2048s 125829119s
Warning: You requested a partition from 2048s to 125829199s (sectors 2048..125829199).
The closest location we can manage is 2048s to 125829036s (sectors 2048..125829036).
Is this still acceptable to you?
Yes/No? Yes
```

Enter **Yes** as prompted to set the end sector.

If the following warning message is displayed, enter **Ignore** to ignore the performance warning. The warning message will not be displayed if the start sector with the best disk performance has been entered. In this example, **2048s** is one of such start sectors. Therefore, the system does not display the warning message.

```
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
```

NOTE

Data will be lost if the following operations are performed:

- Select a start sector other than the partition had before.
- Select an end sector smaller than the partition had before.

Step 10 Enter **p** and press **Enter** to check whether the **/dev/xvdc1** partition has been recreated.

Information similar to the following is displayed:

```
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 125829120s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start      End         Size        File system  Name  Flags
 1      2048s      125829086s  125827039s  ext4         opt
```

The **/dev/xvdc1** partition has been recreated.

Step 11 Enter **q** and press **Enter** to exit parted.

Step 12 Perform the following operations based on the file system of the disk:

- For the **ext3** or **ext4** file system

  a. Run the following command to check the correctness of the file system on **/dev/xvdc1**:

  e2fsck -f /dev/xvdc1

  Information similar to the following is displayed:

  ```
  [root@ecs-1120 linux]# e2fsck -f /dev/xvdb2
  e2fsck 1.42.9 (28-Dec-2013)
  Pass 1: Checking inodes, blocks, and sizes
  Pass 2: Checking directory structure
  Pass 3: Checking directory connectivity
  Pass 4: Checking reference counts
  Pass 5: Checking group summary information
  /dev/xvdc1: 11/655360 files (0.0% non-contiguous), 83137/2620928 blocks
  ```

  b. Run the following command to extend the size of the file system on **/dev/xvdc1**:

  resize2fs /dev/xvdc1

  Information similar to the following is displayed:

  ```
  [root@ecs-1120 linux]# resize2fs /dev/xvdc1
  resize2fs 1.42.9 (28-Dec-2013)
  Resizing the filesystem on /dev/xvdc1 to 15728379 (4k) blocks.
  The filesystem on /dev/xvdc1 is now 15728379 blocks long.
  ```

  c. Run the following command to view the disk partition information after the partition recreation:

  lsblk

  Information similar to the following is displayed:

  ```
  [root@ecs-1120 linux]# lsblk
  NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
  NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
  xvda    202:0    0   80G  0 disk
  ├─xvda1 202:1    0   40G  0 part /
  └─xvda2 202:2    0   40G  0 part /opt
  xvdb    202:16   0  350G  0 disk
  ├─xvdb1 202:17   0  100G  0 part
  └─xvdb2 202:18   0  200G  0 part
  xvdc    202:32   0   60G  0 disk
  └─xvdc1 202:33   0    60G  0 part
  ```

  In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the additional 50 GB has been allocated to the **dev/xvdc1** partition.

d. Run the following command to mount the new partition on **/mnt/sdc**:

mount /dev/xvdc1 /mnt/sdc

- For the **xfs** file system

a. Run the following command to mount the new partition on **/mnt/sdc**:

mount /dev/xvdc1 /mnt/sdc

b. Run the following command to extend the size of the file system on **/dev/xvdc1**:

sudo xfs_growfs /dev/xvdc1

c. Run the following command to view the disk partition information after the partition recreation:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0    60G  0 part
```

In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the additional 50 GB has been allocated to the **dev/xvdc1** partition.

Step 13 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# mount /dev/xvdc1 /mnt/sdc
[root@ecs-1120 linux]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda1     ext4       43G  8.3G   33G  21% /
devtmpfs       devtmpfs  885M     0  885M   0% /dev
tmpfs          tmpfs     894M     0  894M   0% /dev/shm
tmpfs          tmpfs     894M   18M  877M   2% /run
tmpfs          tmpfs     894M     0  894M   0% /sys/fs/cgroup
tmpfs          tmpfs     179M     0  179M   0% /run/user/2000
tmpfs          tmpfs     179M     0  179M   0% /run/user/0
tmpfs          tmpfs     179M     0  179M   0% /run/user/1001
/dev/xvda2     ext4       43G   51M   40G   1% /opt
/dev/xvdc1     ext4       64G   55M   60G   1% /mnt/sdc
```

The **/dev/xvdc1** partition has been mounted on the **/mnt/sdc** directory.

----End

Setting Automatic Mounting at System Start

To automatically mount partitions at system starts, do not specify partitions, for example, **/dev/xvdb1**, in **/etc/fstab** because the sequence of cloud devices, and therefore their names may change during the server stop or start. You are advised to use the UUID in **/etc/fstab** to set automatic mounting at system start.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

For example, run the following command to query the UUID of the **/dev/xvdb1** partition:

blkid /dev/xvdb1

Information similar to the following is displayed:

```
[root@ecs-b656 test]# blkid /dev/xvdb1
/dev/xvdb1: UUID="1851e23f-1c57-40ab-86bb-5fc5fc606ffa" TYPE="ext4"
```

The UUID of the **/dev/xvdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc     ext3 defaults    0   2
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc     ext4 defaults    0   2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

----End

3.2.3.7 Performing Post-Expansion Operations for a SCSI Data Disk in Linux (fdisk)

Scenarios

In Linux, after the capacity expansion succeeded, the additional disk space needs to be allocated to an existing partition or a new partition.

This topic uses SUSE Linux Enterprise Server 11 SP4 64bit to describe how to allocate the additional space of an attached SCSI data disk to a partition using fdisk.

Currently, a disk has been attached to a server, and the original disk capacity is 10 GB. In addition, a 10-GB space has been added to the disk on the management console, and the total disk capacity should be 20 GB. However, the additional space cannot be viewed on the server. The following operations guide you to view the additional space and extend the partition.

The method for allocating the additional space varies depending on the server OS. This document is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Based on your service requirements and disk condition, you can choose either of the following ways to allocate the additional disk space:

- Create a new partition (services will not be interrupted).

  Creating a new partition after expansion does not require the original partitions to be unmounted. Therefore, the impacts on services are minor than expanding an existing partition. This method is recommended for system disks or disks carrying services that cannot be interrupted.

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.

- Expand an existing partition (services will be interrupted).

  If the MBR partition style is used and the number of partitions has reached the upper limit, you can only allocate the additional space to an existing partition. Expanding an existing partition does not delete its data, but requires the partition to unmount. Therefore, services will be interrupted.

  If the MBR partition style is used and the disk capacity after expansion will exceed 2 TB, the space exceeding 2 TB cannot be used. To make use of that space, change the disk partition style from MBR to GPT. Data on the disk will be cleared during such a change. Therefore, back up the disk data before changing the partition style.

NOTICE

Performing the expansion operations with caution. Misoperation may lead to data loss or exceptions. Therefore, you are advised to use CBR to back up the disk data before expansion. For details, see **Managing a Backup**.

Prerequisites

- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.
- You have attached the disk to the server, and the additional space has not been allocated.

Expanding an Existing Partition

The following procedure shows you how to make use of the additional space of a disk attached to a server by extending the **/dev/sda1** partition and mounting the partition on **/mnt/sdc**. During this process, services will be interrupted.

Step 1 Run the following command to view the disk partition information:

fdisk -l

Information similar to the following is displayed:

```
ecs-xen-02:/home/linux # fdisk -l

Disk /dev/xvda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
```



```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00065c40

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1            2048    41945087    20971520   82  Linux swap / Solaris
/dev/xvda2   *    41945088    83892223    20973568   83  Linux
/dev/xvda3        83892224   209715199    62911488   83  Linux

Disk /dev/sda: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2f1c057a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1             2048    20971519    10484736   83  Linux
```

In the command output, 10 GB has been added to the **/dev/sda** data disk on the management console, and the total capacity should be 20 GB. However, the additional space is not included in the command output. In this case, run the following command to update the data disk capacity on the server:

Step 2 Run the following command to update the data disk capacity on the server:

echo 1 > /sys/class/scsi_device/*%d:%d:%d:%d*/device/rescan &

In the command, **%d:%d:%d:%d** indicates a folder in the **/sys/class/scsi_device/** directory and can be obtained using **ll /sys/class/scsi_device/**.

Information similar to the following is displayed: (**2:0:0:0** indicates the folder to be obtained.)

```
cs-xen-02:/sys/class/scsi_device # ll /sys/class/scsi_device/
total 0
lrwxrwxrwx 1 root root 0 Sep 26 11:37 2:0:0:0 -> ../../devices/xen/vscsi-2064/host2/target2:0:0/2:0:0:0/
scsi_device/2:0:0:0
```

Example command:

echo 1 > /sys/class/scsi_device/2:0:0:0/device/rescan &

Step 3 After the disk capacity is updated, run the following command to view the disk partition information again:

fdisk -l

Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # fdisk -l

Disk /dev/xvda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00065c40

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1            2048    41945087    20971520   82  Linux swap / Solaris
/dev/xvda2   *    41945088    83892223    20973568   83  Linux
/dev/xvda3        83892224   209715199    62911488   83  Linux

Disk /dev/sda: 21.5 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```



```
Disk identifier: 0x2f1c057a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            2048    20971519    10484736   83  Linux
```

In the command output, the additional space has been added to the **/dev/sda** data disk, and the update succeeded. The **/dev/sda** disk has one partition only, **/dev/sda1**. Take note of start and end sectors of the **/dev/sda1** partition. These values will be used during the partition recreation. The partition's start sector is **2048**, and its end sector is **20971519**.

Step 4 Run the following command to unmount the disk partition:

umount /mnt/sdc

Step 5 Run the following command to enter fdisk and enter **d** to delete the **/dev/sda1** partition:

fdisk /dev/sda1

Information similar to the following is displayed:

```
[ecs-xen-02:/sys/class/scsi_device # fdisk /dev/sda

Command (m for help): d
Selected partition 1

Command (m for help):
```

Step 6 Enter **n** and press **Enter** to create a new partition.

Entering **n** creates a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
```

There are two types of disk partitions:

- Choosing **p** creates a primary partition.
- Choosing **e** creates an extended partition.

Step 7 Ensure that the entered partition type is the same as the partition had before. In this example, a primary partition is used. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
p
Partition number (1-4, default 1):
```

In the command output, **Partition number** specifies the primary partition number.

Step 8 Ensure that the entered partition number is the same as the partition had before. In this example, partition number **1** is used. Therefore, enter **1** and press **Enter**.

Information similar to the following is displayed:

```
Partition number (1-4, default 1): 1
First sector (2048-41943039, default 2048):
```

In the command output, **First sector** specifies the start sector.

NOTE

Data will be lost if the following operations are performed:

- Select a start sector other than the partition had before.
- Select an end sector smaller than the partition had before.

Step 9 Ensure that the entered start sector is the same as the partition had before. In this example, start sector **2048** is recorded in **Step 3**. Therefore, enter **2048** and press **Enter**.

Information similar to the following is displayed:

```
First sector (2048-41943039, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
```

In the command output, **Last sector** specifies the end sector.

Step 10 Ensure that the entered end sector is larger than or equal to the end sector recorded in **Step 3**. In this example, the recorded end sector is **20971519**, and the default end sector is used. Therefore, enter **41943039** and press **Enter**.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039):
Using default value 41943039

Command (m for help):
```

The partition is created.

Step 11 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed: (Details about the **/dev/sda1** partition are displayed.)

```
CCommand (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2f1c057a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            2048    41943039    20970496   83  Linux
Command (m for help):
```

Step 12 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed: (The partition is successfully created.)

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q**.

Step 13 Perform the following operations based on the file system of the disk:

- For the **ext3** or **ext4** file system

a. Run the following command to check the correctness of the file system on **/dev/sda1**:

e2fsck -f /dev/sda1

Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # e2fsck -f /dev/sda1
e2fsck 1.41.9 (22-Aug-2009)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda1: 11/655360 files (0.0% non-contiguous), 79663/2621184 blocks
```

b. Run the following command to extend the size of the file system on **/dev/sda1**:

resize2fs /dev/sda1

Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # resize2fs /dev/sda1
resize2fs 1.41.9 (22-Aug-2009)
Resizing the filesystem on /dev/sda1 to 5242624 (4k) blocks.
The filesystem on /dev/sda1 is now 5242624 blocks long.
```

c. Run the following command to mount the new partition on **/mnt/sdc**:

mount /dev/sda1 /mnt/sdc

- For the **xfs** file system

a. Run the following command to mount the new partition on **/mnt/sdc**:

mount /dev/sda1 /mnt/sdc

b. Run the following command to extend the size of the file system on **/dev/sda1**:

sudo xfs_growfs /dev/sda1

Step 14 Run the following command to view the mount result:

df -TH

----End

Setting Automatic Mounting at System Start

To automatically mount partitions at system starts, do not specify partitions, for example, **/dev/xvdb1**, in **/etc/fstab** because the sequence of cloud devices, and therefore their names may change during the server stop or start. You are advised to use the UUID in **/etc/fstab** to set automatic mounting at system start.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

For example, run the following command to query the UUID of the **/dev/xvdb1** partition:

blkid /dev/xvdb1

Information similar to the following is displayed:

```
[root@ecs-b656 test]# blkid /dev/xvdb1
/dev/xvdb1: UUID="1851e23f-1c57-40ab-86bb-5fc5fc606ffa" TYPE="ext4"
```

The UUID of the **/dev/xvdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc    ext3 defaults    0   2
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc    ext4 defaults    0   2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

----End

3.2.3.8 Performing Post-Expansion Operations for a System Disk in Linux (fdisk)

Scenarios

In Linux, after the capacity expansion succeeded, the additional disk space needs to be allocated to an existing partition or a new partition.

This topic uses CentOS 7.4 64bit to describe how to allocate the additional system disk space to a partition using fdisk.

This document also describes how to create new partitions for system disks. See the following topics for more information:

- To create new partitions using fdisk in CentOS 7.0 64bit, see **Creating a New Partition**.
- To create new partitions using parted in CentOS 7.0 64bit, see **Creating a New Partition**.

Currently, a disk has been attached to a server, and the original disk capacity is 40 GB. In addition, a 40-GB space has been added to the disk on the management console, and the total disk capacity should be 80 GB. The following operations guide you to use the additional space and extend the partition.

The method for allocating the additional space varies depending on the server OS. This document is used for reference only. For the detailed operations and differences, see the corresponding OS documents.

Based on your service requirements and disk condition, you can choose either of the following ways to allocate the additional disk space:

- Create a new partition (services will not be interrupted).

  Creating a new partition after expansion does not require the original partitions to be unmounted. Therefore, the impacts on services are minor than expanding an existing partition. This method is recommended for system disks or disks carrying services that cannot be interrupted.

If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.

- Expand an existing partition (services will be interrupted).

  If the MBR partition style is used and the number of partitions has reached the upper limit, you can only allocate the additional space to an existing partition. Expanding an existing partition does not delete its data, but requires the partition to unmount. Therefore, services will be interrupted.

  If the MBR partition style is used and the disk capacity after expansion will exceed 2 TB, the space exceeding 2 TB cannot be used. To make use of that space, change the disk partition style from MBR to GPT. Data on the disk will be cleared during such a change. Therefore, back up the disk data before changing the partition style.

NOTICE

Performing the expansion operations with caution. Misoperation may lead to data loss or exceptions. Therefore, you are advised to use CBR to back up the disk data before expansion. For details, see **Managing a Backup**.

Prerequisites

- You have logged in to the server.
  - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.
  - For how to log in to a BMS, see the *Bare Metal Server User Guide*.
- You have attached the disk to the server, and the additional space has not been allocated.

Creating a New Partition

The following example shows you how to make use of the additional capacity of a system disk attached to a server by creating a new partition. The system disk already has the **/dev/vda1** partition, which is mounted on **/**. In the following operations, new partition **/dev/vda2** will be created and mounted on **/opt**. During this process, services are not interrupted.

Step 1 Run the following command to view the disk partition information:

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# fdisk -l

Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux
```


In the command output, the capacity of the **dev/vda** system disk is 80 GB, in which the in-use **dev/vda1** partition takes 40 GB and the additional 40 GB has not been allocated.

Step 2 Run the following command to enter fdisk:

fdisk /dev/vda

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# fdisk /dev/vda
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help):
```

Step 3 Enter **n** and press **Enter** to create a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
Partition type:
  p   primary (1 primary, 0 extended, 3 free)
  e   extended
```

There are two types of disk partitions:

- Choosing **p** creates a primary partition.
- Choosing **e** creates an extended partition.

Step 4 In this example, a primary partition is created. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

```
Select (default p): p
Partition number (2-4, default 2):
```

Step 5 Enter the serial number of the primary partition and press **Enter**. Partition number **2** is used in this example. Therefore, enter **2** and press **Enter.**

Information similar to the following is displayed:

```
Partition number (2-4, default 2): 2
First sector (83886080-167772159, default 83886080):
```

Step 6 Enter the new partition's start sector, for example the default value, and press **Enter**.

The default start sector is used in this example.

Information similar to the following is displayed:

```
First sector (83886080-167772159, default 83886080):
Using default value 83886080
Last sector, +sectors or +size{K,M,G} (83886080-167772159,default 167772159):
```

Step 7 Enter the new partition's end sector and press **Enter**.

The default end sector is used in this example.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (83886080-167772159,
default 167772159):
Using default value 167772159
Partition 2 of type Linux and of size 40 GiB is set

Command (m for help):
```


Step 8 Enter **p** and press **Enter** to view the new partition.

Information similar to the following is displayed:

```
Command (m for help): p

Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux
/dev/vda2        83886080   167772159    41943040   83  Linux
Command (m for help):
```

Step 9 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

The partition is created.

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q**.

Step 10 Run the following command to synchronize the new partition table to the OS:

partprobe

Step 11 Run the following command to set the file system format for the new partition:

(The ext4 file system is used in this example.)

mkfs -t ext4 /dev/vda2

NOTE

The procedure for setting the **xfs** file system is the same as that for the **ext3** or **ext4** file system. The command for creating the **xfs** file system is **mkfs -t xfs /dev/vda2**.

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# mkfs -t ext4 /dev/vda2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
```



```
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

The formatting takes a while, and you need to observe the system running status. Once **done** is displayed in the command output, the formatting is complete.

Step 12 Run the following command to mount the new partition on a space-demanding directory, for example **/opt**:

mount /dev/vda2 /opt

Information similar to the following is displayed:

```
[root@ecs-bab9 test]# mount /dev/vda2 /opt
[root@ecs-bab9 test]#
```

NOTE

If the new partition is mounted on a directory that is not empty, the subdirectories and files in the directory will be hidden. Therefore, you are advised to mount the new partition on an empty directory or a new directory. If the new partition must be mounted on a directory that is not empty, move the subdirectories and files in this directory to another directory temporarily. After the partition is successfully mounted, move the subdirectories and files back.

Step 13 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda1      ext4       43G  2.0G   39G   5% /
devtmpfs       devtmpfs  509M     0  509M   0% /dev
tmpfs          tmpfs     520M     0  520M   0% /dev/shm
tmpfs          tmpfs     520M  7.2M  513M   2% /run
tmpfs          tmpfs     520M     0  520M   0% /sys/fs/cgroup
tmpfs          tmpfs     104M     0  104M   0% /run/user/0
/dev/vda2      ext4       43G   51M   40G   1% /opt
```

----End

Setting Automatic Mounting at System Start

To automatically mount partitions at system starts, do not specify partitions, for example, **/dev/xvdb1**, in **/etc/fstab** because the sequence of cloud devices, and therefore their names may change during the server stop or start. You are advised to use the UUID in **/etc/fstab** to set automatic mounting at system start.

NOTE

UUID is the unique character string for disk partitions in a Linux system.

Step 1 Run the following command to query the partition UUID:

blkid *Disk partition*

For example, run the following command to query the UUID of the **/dev/xvdb1** partition:

blkid /dev/xvdb1

Information similar to the following is displayed:

```
[root@ecs-b656 test]# blkid /dev/xvdb1
/dev/xvdb1: UUID="1851e23f-1c57-40ab-86bb-5fc5fc606ffa" TYPE="ext4"
```

The UUID of the **/dev/xvdb1** partition is displayed.

Step 2 Run the following command to open the **fstab** file using the vi editor:

vi /etc/fstab

Step 3 Press **i** to enter the editing mode.

Step 4 Move the cursor to the end of the file and press **Enter**. Then, add the following information:

```
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc    ext3 defaults    0   2
UUID=1851e23f-1c57-40ab-86bb-5fc5fc606ffa /mnt/sdc    ext4 defaults    0   2
```

Step 5 Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

----End

3.2.4 Managing an Encrypted Disk

Relationships Between Encrypted Disks and Backups

The encryption function can be used for system disks, data disks, and backups. The detailed descriptions are as follows:

- System disk encryption depends on the image of the server OS. If the server is created from an encrypted image, the system disk will be an encrypted disk. For details, see **Encrypting an Image** in the *Image Management Service User Guide*.
- The encryption attribute of an existing disk cannot be changed. You can create new disks and determine whether to encrypt the disks or not.
- When a disk is created from a backup, the encryption attribute of the new disk will be consistent with that of the backup's source disk.
- When a backup is created for a disk, the encryption attribute of the backup is the same as that of the disk.

For details about how to create an encrypted disk, see **Step 2: Create a Disk**.

Creating an Encrypted Disk

Before you use the disk encryption function, KMS access rights need to be granted to EVS. If you have the Security Administrator permission, grant KMS access rights directly. If you do not have this permission, contact a user with the security administrator permission to grant KMS access rights to EVS, then repeat the preceding operations.

For details about how to create an encrypted disk, see **Step 2: Create a Disk**.

Detaching an Encrypted Disk

Before you detach a disk encrypted by a CMK, check whether the CMK is disabled or scheduled for deletion. If the CMK is unavailable, the disk can still be used, but normal read/write operations are not permanently guaranteed. If the disk is detached and then re-attached, re-attaching this disk will fail. In this case, do not detach the disk and restore the CMK status first.

The restoration method varies depending on the current CMK status. For details, see **Disk Encryption**.

If the CMK is available, the disk can be detached and re-attached, and data on the disk will not be lost.

For details about how to detach an encrypted disk, see **Detaching a Data Disk**.

3.2.5 Managing a Shared Disk

How to Use Shared VBD and SCSI Disks?

You can create shared VBD disks or shared SCSI disks. It is recommended that you attach the shared disk to the ECSs in the same ECS group to improve service reliability.

- Shared VBD disks: The device type of a newly created shared disk is VBD by default. Such disks can be used as virtual block storage devices, but do not support SCSI reservations. If SCSI reservations are required for your applications, create shared SCSI disks.
- Shared SCSI disks: These disks support SCSI reservations.

NOTICE

- To improve data security, you are advised to use SCSI reservations together with the anti-affinity policy of an ECS group. That said, ensure that shared SCSI disk is only attached to ECSs in the same anti-affinity ECS group.
- If an ECS does not belong to any anti-affinity ECS group, you are advised not to attach shared SCSI disks to this ECS. Otherwise, SCSI reservations may not work properly, which may put your data at risk.

Concepts of the anti-affinity ECS group and SCSI reservations:

- The anti-affinity policy of an ECS group allows ECSs to be created on different physical servers to improve service reliability.

  For details about ECS groups, see **Managing ECS Groups** in the *Elastic Cloud Server User Guide*.
- The SCSI reservation mechanism uses a SCSI reservation command to perform SCSI reservation operations. If an ECS sends such a command to a disk, the disk is displayed as locked to other ECSs, preventing the data damage that may be caused by simultaneous read/write operations to the disk from multiple ECSs.
- ECS groups and SCSI reservations have the following relationship: A SCSI reservation on a disk cannot differentiate multiple ECSs on the same physical host. For that reason, if multiple ECSs that use the same shared

disk are running on the same physical host, SCSI reservations will not work properly. Therefore, you are advised to use SCSI reservations only on ECSs that are in the same ECS group, thus having a working anti-affinity policy.

Attaching a Shared Disk

A common disk can only be attached to one server, whereas a shared disk can be attached to up to 16 servers.

For details about how to attach a shared disk, see **Attaching a Shared Disk**.

Deleting a Shared Disk

Because a shared disk can be attached to multiple servers, ensure that a shared disk is detached from all the servers before deletion.

For details about how to delete a shared disk, see **Deleting a Disk**.

Expanding a Shared Disk

Shared disks must be expanded when they are in the **Available** state. For details, see **Expanding an Available Disk**.

3.2.6 Managing a Backup

Scenarios

DSS disk backups are created using the CBR service.

This topic describes how to configure a backup policy for a disk. With backup policies configured, data on DSS disks can be periodically backed up to improve data security.

Configuring a Backup Policy

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Distributed Storage Service** > **Disks**.

The disk list page is displayed.

Step 3 In the disk list, locate the disk whose data you want to back up and choose **More** > **Configure Backup Policy** in the **Operation** column.

The **Configure Backup Policy** dialog box is displayed.

Step 4 In the backup policy list, locate the target backup policy to be associated and click **Associate**.

NOTE

If the disk has been associated with a backup policy, you need to disassociate the disk with its backup policy and then associate it with the new policy. For details, see **Data Backup Using a Backup Policy** in the *Volume Backup Service User Guide*.

Step 5 (Optional) To create a new backup policy, click **Edit Backup Policy** to switch to the **Volume Backup Service** page.

For details, see **Data Backup Using a Backup Policy** in the *Volume Backup Service User Guide*.

Step 6 In the displayed **Associate Backup Policy** dialog box, click **OK**.

After the association is complete, the system automatically backs up the data on the disk according to the backup policy.

Step 7 In the displayed **Associate Backup Policy** dialog box, click **OK**.

After the association is complete, the system automatically backs up the data on the disk according to the backup policy.

----End

4 FAQs

4.1 What Are the Risks of Not Expanding the Storage Pool Capacity?

In the process of using a DSS storage pool, when the ratio of **Used Capacity (GB)** to **Total Available Capacity (GB)** exceeds 75%, the system will remind you to expand the capacity. If the utilization stays high all the time, the DSS storage pool will be write protected, increasing the possibility of service interruption. It is recommended that you expand the capacity when the storage utilization reaches 75% to ensure that there is enough space available on the disk.

4.2 How Many Statuses Does a Storage Pool Have?

A storage pool has several statuses. **Table 4-1** lists the meaning of each status and the operations for each status.

Table 4-1 Storage pool status

Status	Description	Allowed Operation
Deploying	The storage pool is being deployed.	-
Available	The storage pool is successfully created, and you can create disks in the storage pool.	Creating disks
Expanding	The storage pool capacity is being expanded and cannot be used.	-

4.3 How Many Statuses Does a Disk Have?

A disk has several statuses. Table 4-2 lists the meaning of each status and the operations for each status.

Table 4-2 Disk status description

Status	Description	Allowed Operation
In-use	The disk is attached to a server and in use.	• Detaching • Creating backups **NOTE** If a shared disk is in the **In-use** state, the disk can be attached.
Available	The disk is successfully created and has not been attached to any server.	• Attaching • Expanding • Deleting
Creating	The disk is being created.	None
Attaching	The disk is being attached to a server.	None
Detaching	The disk is being detached from a server.	None
Deleting	The disk is being deleted.	None
Expanding	The capacity of the disk is being expanded.	None
Uploading	Data on the disk is being uploaded to an image. This status occurs when you create an image from a server.	None
Downloading	Data is being downloaded from an image to the disk. This status occurs when you create a server.	None
Error	An error occurs when you try to create a disk.	Deleting
Deletion failed	An error occurs when you try to delete a disk.	No operations can be performed.
Expansion failed	An error occurs when you try to expand the capacity of a disk.	Deleting

NOTE

If a DSS disk status is **Error**, **Deletion failed**, or **Expansion failed**, you can rectify the error by following the steps provided in **What Should I Do If an Error Occurs on My DSS Disk?**

4.4 Troubleshooting and Impacts on the DSS Usage

- Case one

  If server or disk failures cause a disk being removed from the storage pool, the total available capacity becomes smaller. After the fault is rectified, the total available capacity can be restored to the original value.

- Case two

  If a server or disk is faulty, and no disk is removed from the storage pool, the storage pool is degraded, which does not affect the use of the storage pool. After the fault is rectified, the storage pool becomes normal.

  NOTE

  Storage pool degradation refers to that the number of data copies for some data in a storage pool is reduced from three copies to two copies and cannot be automatically restored without manual troubleshooting.

- Case three

  If a server or disk is faulty, your services may be interrupted. For example, if the used capacity is 98% and a disk is removed due to server or disk failures, the total available capacity decreases. As a result, the proportion of the used capacity to the total available capacity reaches nearly 100%. The write protection is enabled for the storage pool, and your services are interrupted.

4.5 Can I Attach a Disk to Multiple Servers?

A non-shared disk can be attached to one server only.

A shared disk can be attached to a maximum of 16 servers.

NOTE

Shared disks are a type of DSS disks and can be attached to multiple servers.

4.6 Will Data in the DSS Disk Be Lost When the DSS Disk Is Detached?

Not necessarily.

To prevent data loss when you detach a DSS disk, perform the following operations:

1. Stop the ECS to which the to-be-detached disk has been attached.
2. After server has been stopped, detach the disk.

4.7 What Should I Do If an Error Occurs on My DSS Disk?

If an error occurs, the disk may show one of the states listed in **Table 4-3**. Take the following measures to handle the exceptions.

Table 4-3 Solutions for disk errors

Error Status	Handling Suggestion
Error	Delete the disk in the **Error** state and create another one.
Deletion failed	Contact customer service.
Expansion failed	Customer service personnel will contact you and help you handle this error. Do not perform any operations on the disk before they contact you. If you require that the error be handled as soon as possible, contact customer service.

4.8 What Are the Restrictions on Attaching a Disk to an ECS?

- The disk and the target ECS must be located in the same AZ.
- For a non-shared disk, the DSS disk must be in **Available** state.
  For a shared disk, the target ECS must be in the **In-use** or **Available** state.
- The target ECS must be in **Running** or **Stopped** state.
- A frozen disk cannot be attached to an ECS.
- A SCSI disk cannot be used as an ECS system disk.

4.9 What Are the Precautions for Detaching a Disk from an ECS?

Scenarios

A disk attached to an ECS can function as a system disk or data disk.

- Disks mounted on **/dev/sda** or **/dev/vda** function as system disks. Currently, system disks must be detached offline. In this case, you must stop the ECS before detaching a system disk.
- Disks mounted on other mount points functions as data disks. In addition to offline detachment, a data disk can also be detached online if the server OS supports online detachment. In this case, you do not need to stop the running ECS.

This FAQ describes how to detach a disk from a running ECS.

Constraints

- The disk to be detached must be mounted on a mount point other than **/dev/sda** or **/dev/vda**.

  Disks mounted on **/dev/sda** or **/dev/vda** are system disks and cannot be detached from running ECSs.
- Before detaching a disk from a running Windows ECS, make sure that VMTtools have been installed on the ECS and that the tools are running properly.
- Before detaching a disk from a running Windows ECS, ensure that no program is reading data from or writing data to the disk. Otherwise, data will be lost.
- SCSI disks cannot be detached from running Windows ECSs.
- Before detaching a disk from a running Linux ECS, you must log in to the ECS and run the **umount** command to cancel the association between the disk and the file system. In addition, ensure that no program is reading data from or writing data to the disk. Otherwise, the disk will fail to be detached.

OSs Supporting Disk Detachment from a Running ECS

OSs supporting disk detachment from a running ECS include two parts:

- For the first part, see "Formats and OSs Supported for External Image Files" in *Image Management Service User Guide*.
- For the second part, see **Table 4-4**.

Table 4-4 OSs supporting disk detachment from a running ECS

OS	Version
CentOS	7.3 64bit
	7.2 64bit
	6.8 64bit
	6.7 64bit
Debian	8.6.0 64bit
	8.5.0 64bit
Fedora	25 64bit
	24 64bit
SUSE	SUSE Linux Enterprise Server 12 SP2 64bit
	SUSE Linux Enterprise Server 12 SP1 64bit
	SUSE Linux Enterprise Server 11 SP4 64bit
	SUSE Linux Enterprise Server 12 64bit

OS	Version
OpenSUSE	42.2 64bit
	42.1 64bit
Oracle Linux Server release	7.3 64bit
	7.2 64bit
	6.8 64bit
	6.7 64bit
Ubuntu Server	16.04 64bit
	14.04 64bit
	14.04.4 64bit
Windows (SCSI disks cannot be detached from a running ECS.)	Windows Server 2008 R2 Enterprise 64bit
	Windows Server 2012 R2 Standard 64bit
	Windows Server 2016 R2 Standard 64bit
Redhat Linux Enterprise	7.3 64bit
	6.8 64bit

NOTE

Online detachment is not supported by the ECSs running OSs not listed in the preceding table. For such ECSs, stop the ECSs before detaching disks from them to prevent any possible problems from occurring.

4.10 Why My Disk Cannot Be Attached to a Server?

If your disk cannot be attached, verify that your disk meets the following conditions:

- The disk is in the **Available** state if it is a non-shared disk.
- The disk is in the **Available** or **In-use** state if it is a shared disk, and the maximum number of servers that the disk can be attached to is not reached.
- The disk is not added to any replication pair in SDRS. If it has been added to a replication pair, delete the replication pair and try again.
- The disk is not frozen.
- The disk is not locked by any service.

A Change History

Released On	Description
2019-10-10	This issue is the second official release, which incorporates the following change: Optimized the overall document structure and content.
2018-07-30	This issue is the first official release.

