



Elastic Volume Service

User Guide

Date **2024-04-15**

Contents

1 Overview..... 1

1.1 What Is EVS?..... 1

1.2 Disk Types and Performance..... 2

1.3 Device Types and Usage Instructions..... 5

1.4 Shared EVS Disks and Usage Instructions..... 6

1.5 EVS Backup..... 9

1.6 EVS Snapshot..... 11

1.7 Differences Between EVS Backups and EVS Snapshots..... 12

1.8 Permissions..... 14

1.9 EVS and Other Services..... 16

1.10 Basic Concepts..... 16

1.10.1 Region and AZ..... 16

2 Getting Started..... 18

2.1 Process Overview..... 18

2.2 Create an EVS Disk..... 20

2.3 Attach an EVS Disk..... 25

2.3.1 Attaching a Non-Shared Disk..... 25

2.3.2 Attaching a Shared Disk..... 27

2.4 Initialize an EVS Data Disk..... 29

2.4.1 Introduction to Data Disk Initialization Scenarios and Partition Styles..... 29

2.4.2 Initializing a Windows Data Disk (Windows Server 2008)..... 31

2.4.3 Initializing a Windows Data Disk (Windows Server 2019)..... 38

2.4.4 Initializing a Linux Data Disk (fdisk)..... 46

2.4.5 Initializing a Linux Data Disk (parted)..... 52

2.4.6 Initializing a Windows Data Disk Larger Than 2 TiB (Windows Server 2008)..... 58

2.4.7 Initializing a Windows Data Disk Larger Than 2 TiB (Windows Server 2012)..... 66

2.4.8 Initializing a Linux Data Disk Larger Than 2 TiB (parted)..... 74

3 Disk Capacity Expansion..... 81

3.1 Expansion Overview..... 81

3.2 Expanding Capacity for an In-use EVS Disk..... 83

3.3 Expanding Capacity for an Available EVS Disk..... 86

3.4 Extending Disk Partitions and File Systems (Windows Server 2008)..... 87

3.5 Extending Disk Partitions and File Systems (Linux).....	105
3.5.1 Partition and File System Extension Preparations (Linux).....	105
3.5.2 Extending Partitions and File Systems for System Disks (Linux).....	109
3.5.3 Extending Partitions and File Systems for Data Disks (Linux).....	119
3.5.4 Extending Partitions and File Systems for SCSI Disks (Linux).....	139
4 Detaching an EVS Disk.....	152
4.1 Detaching a System Disk.....	152
4.2 Detaching a Data Disk.....	153
5 Deleting EVS Disks.....	156
6 Permissions Management.....	157
6.1 Creating a User and Granting EVS Permissions.....	157
6.2 EVS Custom Policies.....	158
7 Managing Shared EVS Disks.....	160
8 Managing EVS Backups.....	162
9 Managing EVS Snapshots.....	164
9.1 Snapshot Overview.....	164
9.2 Creating a Snapshot.....	165
9.3 Deleting a Snapshot.....	167
9.4 Viewing Details of a Snapshot.....	168
9.5 Rolling Back Data from a Snapshot.....	168
9.6 Creating an EVS Disk from a Snapshot.....	169
10 Managing EVS Transfers.....	171
11 Viewing EVS Monitoring Data.....	174
12 Managing Quotas.....	176
12.1 Querying EVS Resource Quotas.....	176
12.2 Increasing EVS Resource Quotas.....	176
13 FAQ.....	178
13.1 General.....	178
13.1.1 How Do I Start Using a Newly Disk?.....	178
13.1.2 Can EVS Disks Be Used Directly for Storage?.....	178
13.1.3 Can EVS Disks Be Used Alone?.....	178
13.1.4 How Can I View My Disk Details?.....	178
13.1.5 Can I Change the AZ of My Disk?.....	179
13.1.6 What Are the Differences Between System Disks and Data Disks?.....	179
13.1.7 How Can I Download My EVS Disk Data to a Local PC?.....	179
13.1.8 How Can I Export the Original Data After I Changed My Server OS from Windows to CentOS?...	179
13.1.9 What Are the Differences Between MBR and GPT Partition Styles?.....	180
13.2 Attachment.....	181

13.2.1 Why Can't I View the Attached Data Disk on the Server?.....	181
13.2.2 Can I Attach a Disk to Multiple Servers?.....	184
13.2.3 Can I Attach a Disk to a Server in Another AZ?.....	184
13.2.4 How Can I Add a Data Disk to an Existing Server?.....	184
13.2.5 Can I Attach Different Types of Disks to the Same Server?.....	185
13.2.6 What Should I Do If a Linux EVS Disk Is Attached to a Windows Server?.....	185
13.2.7 How Do I Obtain My Disk Device Name in the ECS OS Using the Device Identifier Provided on the Console?.....	185
13.3 Capacity Expansion.....	189
13.3.1 Can I Reduce or Temporarily Expand the Disk Capacity?.....	189
13.3.2 What Are the Differences Between Expanding Capacity by Expanding an EVS Disk and Creating a New EVS Disk?.....	189
13.3.3 Will My Disk Data Be Lost After I Expand the Disk Capacity?.....	189
13.3.4 Do I Need to Restart the Server After Expanding the Disk Capacity?.....	189
13.3.5 Do I Need to Detach an EVS Disk Before Expanding Its Capacity?.....	190
13.3.6 What Should I Do If My Disk Capacity Exceeds 2 TiB After Expansion?.....	190
13.3.7 Why Did My Disk Capacity Remain Unchanged on the Server After Capacity Expansion?.....	191
13.3.8 How Do I Extend the File System of an Unpartitioned Data Disk in Linux?.....	191
13.3.9 How Do I View the Disk Partition Style in Linux?.....	193
13.4 Detachment.....	196
13.4.1 If I Detach a Disk, Will I Lose the Data on My Disk?.....	196
13.4.2 Why Can't I Detach My Disk?.....	196
13.5 Capacity.....	196
13.5.1 What Is the Maximum Capacity Supported for the System and Data Disks?.....	197
13.5.2 What Should I Do If My Disk Starts to Run Out of Space?.....	197
13.5.3 What Should I Do If I Use fdisk to Initialize a Disk Larger Than 2 TiB and Then the Space in Excess of 2 TiB Cannot Be Displayed?.....	197
13.5.4 How Can I View My Disk Usage?.....	197
13.5.5 Why the Space of My New Disk Is Full After I Uploaded Only 500 MB of Files to the Disk?.....	205
13.6 Snapshot.....	207
13.6.1 What Are the Typical Causes of a Snapshot Creation Failure?.....	207
13.6.2 Does EVS Support Automatic Snapshot Creation?.....	207
13.6.3 Can I Create Snapshots for Multiple Disks at a Time?.....	207
13.6.4 How Is a Snapshot Created for My Disk?.....	207
13.6.5 Why Can't I Roll Back My Disk Data from a Snapshot?.....	208
13.6.6 Can I Roll Back the Data from a Snapshot After Reinstalling the OS or Formatting the Disk?.....	208
13.6.7 How Is the Snapshot Size Calculated?.....	208
13.6.8 Do Snapshots Take Space on the Disk?.....	208
13.6.9 Can I Perform Multiple Rollback Operations for a Snapshot?.....	208
13.6.10 Can I Replicate Snapshots to Other Regions or Accounts?.....	208
13.6.11 Why Can't I Find My Snapshot?.....	209
13.6.12 Can I Use a Snapshot to Create a Disk and Roll Back the Snapshot at Almost the Same Time?.....	209
13.7 Sharing.....	209

13.7.1 Do I Have to Deploy a Cluster to Use Shared Disks?.....209

13.7.2 How Many Servers Can I Attach a Shared Disk to?..... 209

13.7.3 How Can I Attach a Shared Disk to Multiple Servers?..... 209

13.7.4 Can a Shared Disk Be Attached to Servers That Belong to Different Accounts?..... 209

13.7.5 Can I Attach a Shared Disk to Servers Running Different OSs?.....210

A Appendix..... 211

A.1 EVS Disk Status..... 211

A.2 EVS Snapshot Status..... 212

B Change History..... 214

1 Overview

1.1 What Is EVS?

Overview

Elastic Volume Service (EVS) offers scalable block storage for cloud servers. With high reliability, high performance, and a variety of specifications, EVS disks can be used for distributed file systems, development and test environments, data warehouses, and high-performance computing (HPC) applications. Cloud servers that EVS supports include Elastic Cloud Servers (ECSs) and BMS.

EVS disks are similar to hard disks in PCs. They must be attached to servers for use and cannot be used alone. You can initialize EVS disks, create file systems on them, and store data persistently on them.

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

EVS Advantages

EVS has the following advantages:

Table 1-1 EVS advantages

Advantage	Description	Related Knowledge
Various disk types	EVS provides a variety of disk types for you to choose from, and EVS disks can be used as data disks and system disks for servers. You can select an appropriate disk type that best suits your budget and service requirements.	Disk Types and Performance
Elastic scalability	The EVS disk capacity ranges from 10 GiB to 32 TiB. When it no longer meets your needs, you can expand the disk capacity up to 32 TiB in increments of 1 GiB, without interrupting your applications.	Expansion Overview

Advantage	Description	Related Knowledge
	Besides the disk capacity limit, the additional space you can add cannot exceed the remaining quota. You can increase the quota if the remaining quota is insufficient.	Managing Quotas
High security and reliability	Data redundancy functions, such as backups and snapshots, safeguard the disk data, preventing incorrect data caused by application exceptions or attacks.	EVS Backup
Real-time monitoring	On Cloud Eye, you can monitor the disk health and operating status at any time.	Viewing EVS Monitoring Data

1.2 Disk Types and Performance

EVS disks are classified based on the disk I/O performance. EVS disks differ in performance and price. Choose the disk type most appropriate for your applications.

EVS Performance

EVS performance metrics include:

- IOPS: Number of read/write operations performed by an EVS disk per second
- Throughput: Amount of data read from and written into an EVS disk per second
- Read/write I/O latency: Minimum interval between two consecutive read/write operations on an EVS disk

Single-queue access latencies of different types of EVS disks are as follows:

- Common I/O: 5 ms to 10 ms
- High I/O: 1 ms to 3 ms
- Ultra-high I/O: 1 ms

Table 1-2 EVS performance data

Parameter	Common I/O	High I/O	Ultra-high I/O
IOPS per GiB/EVS disk	2	8	50
Max. IOPS/EVS disk	2,200	5,000	50,000
Baseline IOPS/EVS disk	500	1,800	1,800

Parameter	Common I/O	High I/O	Ultra-high I/O
Disk IOPS	Min. (2,200, 500 + 2 x Capacity)	Min. (5,000, 1,800 + 8 x Capacity)	Min. (50,000, 1,800 + 50 x Capacity)
IOPS burst limit/EVS disk	2,200	5,000	16,000
Max. throughput	50 MiB/s	150 MiB/s	350 MiB/s
API name NOTE This API name indicates the value of the volume_type parameter in the EVS API. It does not represent the type of the underlying hardware device.	SATA	SAS	SSD
Typical application scenarios	Business applications that require large capacity, normal read/write speed, and few transactions, such as enterprise office applications and small-scale testing. When higher I/O performance is required, it is recommended that you select ultra-high I/O or high I/O EVS disks.	Mainstream applications requiring high performance and high reliability, such as large-scale development and test environments, web server logs, and enterprise applications. Typical enterprise applications include SAP applications, Microsoft Exchange, and Microsoft SharePoint.	Read/write-intensive applications that require ultra-high I/O and throughput, such as distributed file systems used in HPC scenarios or NoSQL and relational databases used in I/O-intensive scenarios. Typical databases include MongoDB, Oracle, SQL Server, MySQL, and PostgreSQL databases.

Calculating Disk IOPS Limit

To calculate the IOPS limit of a disk, obtain the smaller value of the following two values:

- Max. IOPS/disk
- Baseline IOPS/disk + IOPS per GiB x Disk capacity

The following example uses an ultra-high I/O EVS disk with a maximum IOPS of 33,000.

- If the disk capacity is 100 GiB, the disk IOPS limit is calculated as follows:
 $\text{Disk IOPS limit} = \text{Min. } (33,000, 1,500 + 50 \times 100)$
The disk IOPS limit is 6,500, the smaller value between 33,000 and 6,500.
- If the disk capacity is 1,000 GiB, the disk IOPS limit is calculated as follows:
 $\text{Disk IOPS limit} = \text{Min. } (33,000, 1,500 + 50 \times 1,000)$
The disk IOPS limit is 33,000, the smaller value between 33,000 and 51,500.

Disk Burst Capability and Principles

EVS disks have burst capability, which allows a small-capacity disk to surpass its maximum IOPS within a certain period of time.

Disks with burst capability are well-suited for speeding up server startup. In most cases, system disks have small capacities. For example, the IOPS of a 50-GiB ultra-high I/O disk without burst capability can reach only 4,000 ($1,500 + 50 \times 50$). If the disk has burst capability, its IOPS can burst up to 16,000.

The following example uses an ultra-high I/O EVS disk with the IOPS burst limit of 16,000.

- If the disk capacity is 100 GiB, the disk has a maximum IOPS of 6,500, but it can burst to 16,000 IOPS in a certain duration.
- If the disk capacity is 1,000 GiB, the disk has a maximum IOPS of 33,000. The disk maximum IOPS already exceeds its burst IOPS 10,000, and the disk does not use the burst capability.

The following describes the burst IOPS consumption and reservation.

A token bucket is used to handle burst I/O operations. The number of initial tokens in the bucket is calculated as follows:

$\text{Number of initial tokens} = \text{Burst duration} \times \text{IOPS burst limit}$

In the following example, a 100-GiB ultra-high I/O EVS disk is used, and the fixed burst duration is 1800s. Therefore, the number of initial tokens is 28,800,000 ($1,800 \times 16,000$).

- Token production rate: This rate equals the disk maximum IOPS, which is 6,500 tokens/s.
- Token consumption rate: This rate is calculated based on the I/O usage. Each I/O request consumes a token. The maximum consumption rate is 10,000 tokens/s, which is the larger value between the disk burst IOPS and maximum IOPS.

Consumption principles

When the token consumption rate is greater than the production rate, the number of tokens decreases accordingly, and eventually the disk IOPS will be consistent with the token production rate (the maximum IOPS). In this example, the disk can burst for approximately 3,032 seconds [$28,800,000 / (16,000 - 6,500)$].

Reservation principles

When the token consumption rate is smaller than the production rate, the number of tokens increases accordingly, enabling the disk to regain the burst capability. In

this example, if the disk is suspended for approximately 4,431 seconds (28,800,000/6,500), the token bucket will be filled up with tokens.

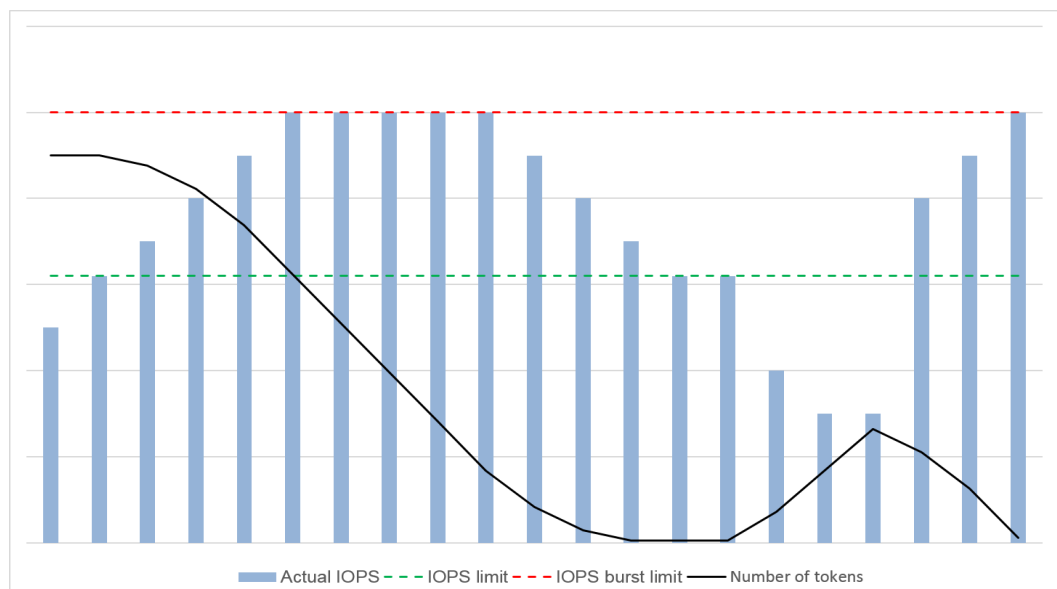
NOTE

As long as there are tokens in the token bucket, the disk has the burst capability.

Figure 1-1 shows the token consumption and reservation principles. The blue bars indicate the disk IOPS usage, the green dashed line represents the maximum IOPS, the red dashed line indicates the IOPS burst limit, and the black curve indicates the changes of the number of tokens.

- When the number of tokens is greater than zero, the disk IOPS can exceed 6,500 and has the capability to reach 16,000, the IOPS burst limit.
- When the number of tokens is zero, the disk does not have the burst capability, and the maximum IOPS is 6,500.
- When the disk IOPS is less than 6,500, the number of tokens starts to increase, and the disk can regain the burst capability.

Figure 1-1 Burst capability diagram



1.3 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 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.

Device type is configured during creation. It cannot be changed after the disk has been created.

Common Application Scenarios and Usage Instructions of 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 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 EVS Disks and Usage Instructions](#).

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

To use SCSI EVS disks, a cloud server must have a SCSI driver installed. If the SCSI driver is not pre-installed, you need to install it manually.

Check whether you need to manually install the driver based on the server type.

- 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

Currently, only KVM ECSs are supported.

- If neither of the server types are used, you need to manually install the driver on your server.

1.4 Shared EVS Disks and Usage Instructions

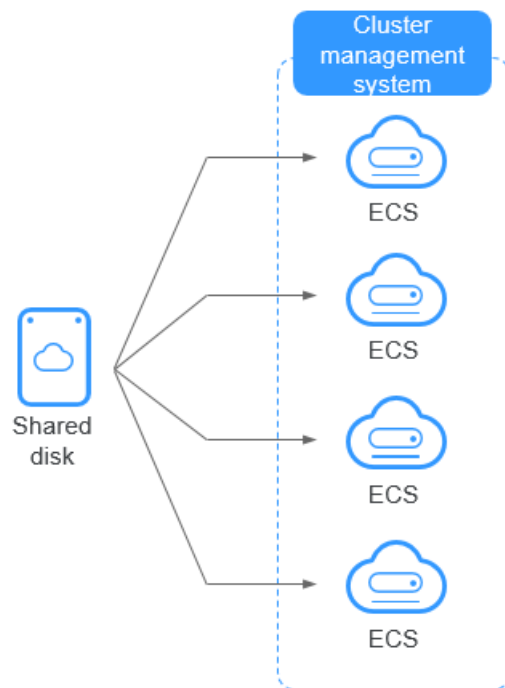
What Are Shared EVS Disks?

Shared EVS disks are block storage devices that support concurrent read/write operations and can be attached to multiple servers. Shared EVS disks feature multiple attachments, high-concurrency, high-performance, and high-reliability. They are usually used for enterprise business-critical applications that require cluster deployment and high availability (HA). Multiple servers can access the same shared EVS disk at the same time.

A shared EVS disk can be attached to a maximum of 16 servers. Servers that EVS supports include ECSs and BMSs. To share files, you need to deploy a shared file system or a cluster management system, such as Windows MSCS, Veritas VCS, or CFS.

NOTICE

You must set up a shared file system or cluster management system before using shared EVS disks. If you directly attach a disk to multiple servers, the sharing function will not work and data may be overwritten.

Figure 1-2 Application scenario of shared EVS disks

Usage Precautions

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

You can create shared VBD disks or shared SCSI disks. It is recommended that you attach a 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 EVS disks.
- Shared SCSI disks: Such 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 disks are 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 an EVS 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 EVS disk cannot differentiate multiple ECSs on the same physical host. For that reason, if multiple ECSs that use the same shared EVS disk are running on the same physical host, SCSI reservations will not work properly. So 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

- Multiple attachments: A shared EVS disk can be attached to a maximum of 16 servers.
- High-performance: The random read/write IOPS of a shared ultra-high I/O disk can reach up to 160,000.
- Wide range of use: Shared EVS disks can be used for Linux RHCS clusters where only VBD EVS disks are needed. They can also be used for Windows MSCS and Veritas VCS clusters that require SCSI reservations.

Specifications and Performance

Shared EVS disks have the same specifications and performance as non-shared EVS disks.

Data Sharing Principle and Common Usage Mistakes

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

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

- Data inconsistency caused by read/write conflicts
When a shared EVS 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 EVS 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 EVS 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 EVS 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 EVS disk to multiple servers, the disk device type needs to be determined. The device type can be either VBD or SCSI. Shared SCSI EVS 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.

NOTICE

If you simply attach a shared EVS disk to multiple servers, files cannot be shared between the servers as shared EVS 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.

1.5 EVS Backup

What Is EVS Backup?

Cloud Disk Backup provided by Cloud Backup and Recovery (CBR) allows you to create backups for your EVS disks while servers are running. If data loss or damage occurred due to virus invasions, accidental deletions, or software/hardware faults, you can use backups to restore data, guaranteeing your data integrity and security.

For more information, see the *Cloud Backup and Recovery User Guide*.

CBR Architecture

CBR involves backups, vaults, and policies.

Backup

A backup is a copy of a particular chunk of data and is usually stored elsewhere so that it may be used to restore the original data in the event of data loss. There are the following types of backups:

- Cloud disk backup: provides snapshot-based backups for EVS disks.
- Cloud server backup: uses the consistency snapshot technology to protect data for ECSs. Backups of non-database servers are non-database server backups, and those of database servers are application-consistent backups.

Vault

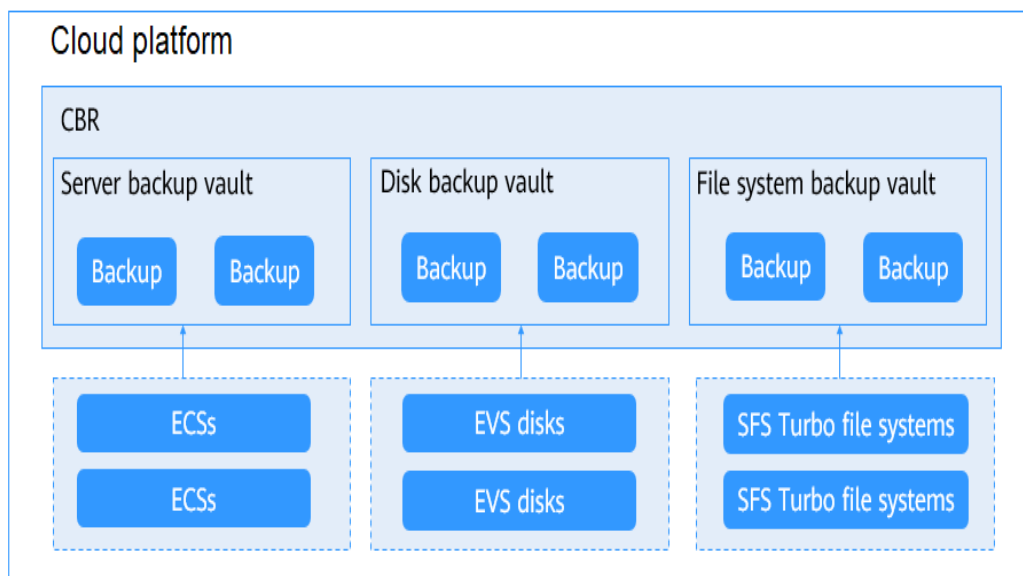
CBR stores backups in vaults. Before creating a backup, you need to create at least one vault and associate the resources you want to back up with the vaults. Then the resources can be backed up to the associated vaults.

Different types of resources must be backed up to different types of vaults. For example, cloud servers must be backed up to server backup vaults, not disk backup vaults or any other types of vaults.

Policy

- A backup policy defines when you want to take a backup and for how long you would retain each backup.

Figure 1-3 CBR architecture



Application Scenarios

Create and apply backup policies to schedule periodic backups for your EVS disks. You can use the backup data to create new EVS disks or restore to source disks.

EVS backup can help address your following needs:

- Create and apply backup policies to schedule periodic backups for your EVS disks. You can use the backup data to create new EVS disks or restore to source disks.
- Share backups with other users. You can use the backups shared by other users to create new EVS disks.

1.6 EVS Snapshot

What Is EVS Snapshot?

An EVS snapshot is a complete copy or image of the disk data at a specific point in time. Snapshots can be used as a disaster recovery (DR) approach, and you can use snapshots to fully restore data to the time when the snapshot was taken. You can create snapshots for disks on the console or via the API.

EVS snapshots are sometimes referred to as snapshots in this document.

You can create snapshots to rapidly save the disk data at specified time points. In addition, you can use snapshots to create new disks so that the created disks will contain the snapshot data in the beginning.

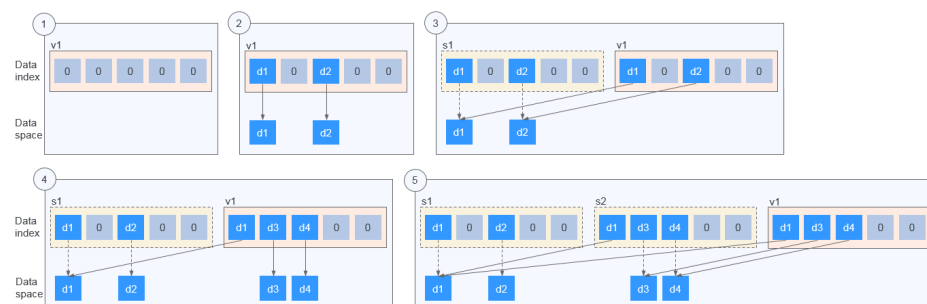
Snapshot Principles

Snapshots and backups are different in that a backup saves the data as another copy in the storage system other than on the disk, whereas a snapshot establishes a relationship between the snapshot and disk data.

The following example describes the snapshot principle by creating snapshots s1 and s2 for disk v1 at different time points:

1. Create disk v1, which contains no data.
2. Write data d1 and d2 to disk v1. Data d1 and d2 are written to new spaces.
3. Create snapshot s1 for disk v1 that is modified in 2. Data d1 and d2 are not saved as another copy elsewhere. Instead, the relationship between snapshot s1 and data d1 and d2 is established.
4. Write data d3 to disk v1 and change data d2 to d4. Data d3 and d4 are written to new spaces, and data d2 is not overwritten. The relationship between snapshot s1 and data d1 and d2 is still valid. Therefore, snapshot s1 can be used to restore data if needed.
5. Create snapshot s2 for disk v1 that is modified in 4. The relationship between s2 and data d1, d3, and d4 is established.

Figure 1-4 Snapshot principle



Application Scenarios

The snapshot function helps address your following needs:

- Routine data backup

You can create snapshots for disks on a timely basis and use snapshots to recover your data in case that data loss or data inconsistency occurred due to unintended operations, viruses, or attacks.

- Rapid data restoration

You can create a snapshot or multiple snapshots before an application software upgrade or a service data migration. If an exception occurs during the upgrade or migration, service data can be rapidly restored to the time point when the snapshot was created.

For example, a fault occurred on system disk A of server A, and therefore server A cannot be started. As system disk A is already faulty, data on system disk A cannot be restored by rolling back snapshots. But, you can create disk B using an existing snapshot of system disk A and attach disk B to a properly running server, for example server B. In this case, server B obtains the data of system disk A from disk B.

NOTE

When rolling back data from snapshots, data can only be rolled back to the original disk, and a rollback to a different disk is not possible.

- Multi-service quick deployment

You can use a snapshot to create multiple disks containing the same initial data, and these disks can be used as data resources for various services, for example data mining, report query, and development and testing. This method protects the initial data and creates disks rapidly, meeting diverse service requirements.

1.7 Differences Between EVS Backups and EVS Snapshots

Both EVS backups and EVS snapshots provide redundancies for improved disk data reliability. [Table 1-3](#) lists the differences between them.

Table 1-3 Differences between backups and snapshots

Metric	Storage Solution	Data Synchronizati on	DR Range	Service Recovery
Backup	Backups are stored in OBS, instead of disks. This ensures data restoration upon disk damage or corruption.	A backup is a copy of a disk taken at a given point of time and is stored in a different location. Automatic backup can be performed based on backup policies. Deleting a disk will not delete its backups.	A backup and its source disk reside in different AZs.	To restore data and recover services, you can restore the backups to their original disks or create new disks from the backups.
Snapshot	Snapshots are stored on the same disk as the original data. NOTE Creating a backup requires a certain amount of time because data needs to be transferred to OBS. Creating or rolling back a snapshot consumes less time than creating a backup.	A snapshot is the state of a disk at a specific point in time and is stored on the same disk. If the disk is deleted, all its snapshots will also be deleted. For example, if you reinstalled or changed the server OS, snapshots of the system disk were also automatically deleted. Snapshots of the data disks can be used as usual.	A snapshot and its source disk reside in the same AZ.	You can use a snapshot to roll back its original disk or create a disk from the snapshot.

1.8 Permissions

If you need to assign different permissions to employees in your enterprise to access your EVS resources, IAM is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you securely access your cloud resources.

With IAM, you can control access to specific cloud resources. For example, if you want some resource management personnel in your enterprise to view EVS resources but do not want them to delete EVS resources or perform any other high-risk operations, you can grant permission to view EVS resources but not permission to delete them.

If your account does not require IAM for permissions management, you can skip this section.

IAM is a free service. You only pay for the resources in your account. For more information about IAM, see section "Service Overview" in the *Identity and Access Management User Guide*.

EVS Permissions

New IAM users do not have any permissions assigned by default. You need to first add them to one or more groups and attach policies or roles to these groups. The users then inherit permissions from the groups and can perform specified operations on cloud services based on the permissions they have been assigned.

EVS is a project-level service deployed for specific regions. To assign EVS permissions to a user group, specify the scope as region-specific projects and select a project for the permissions to take effect. If **All projects** is selected, the permissions will take effect for the user group in all region-specific projects. When accessing EVS, users need to switch to a region where they have been authorized to use EVS.

You can grant users permissions by using roles and policies.

- **Roles:** A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other roles on which the permissions depend to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- **Policies:** A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization, meeting requirements for secure access control. For example, you can grant ECS users only the permissions for managing a certain type of ECSs. Most policies define permissions based on APIs. For the API actions supported by EVS, see section "Permissions Policies and Supported Actions" in the *Elastic Volume Service API Reference*.

Table 1-4 lists all the system-defined roles and policies supported by EVS.

Table 1-4 System-defined roles and policies supported by EVS

Role/Policy Name	Description	Type	Dependency
EVS FullAccess	Full permissions for EVS. Users granted these permissions can create, attach, detach, query, and delete EVS resources, and expand capacity of EVS disks.	System-defined policy	None
EVS ReadOnlyAccess	Read-only permissions for EVS. Users granted these permissions can view EVS resource data only.	System-defined policy	None
Server Administrator	Full permissions for EVS	System role	None

Table 1-5 lists the common operations supported by each system-defined policy of EVS. Select the policies as required.

Table 1-5 Common operations supported by each system-defined policy of EVS

Operation	EVS FullAccess	EVS ReadOnlyAccess
Creating disks	√	x
Viewing the disk list	√	√
Viewing disk details	√	√
Attaching disks	√	x
Detaching disks	√	x
Deleting disks	√	x
Expanding disk capacities	√	x
Creating snapshots	√	x
Deleting snapshots	√	x
Rolling back snapshot data	√	x
Changing disk names	√	x

1.9 EVS and Other Services

Table 1-6 EVS and other services

Interactive Function	Related Service	Reference
EVS disks can be attached to ECSs and used as scalable block storage devices.	ECS	• Attaching a Non-Shared Disk • Attaching a Shared Disk
After EVS is enabled, the performance metrics of monitored disks can be viewed through Cloud Eye without installing any additional plug-in. The monitored metrics include Disk Read Rate, Disk Write Rate, Disk Read Requests, and Disk Write Requests.	Cloud Eye	Viewing EVS Monitoring Data

1.10 Basic Concepts

1.10.1 Region and AZ

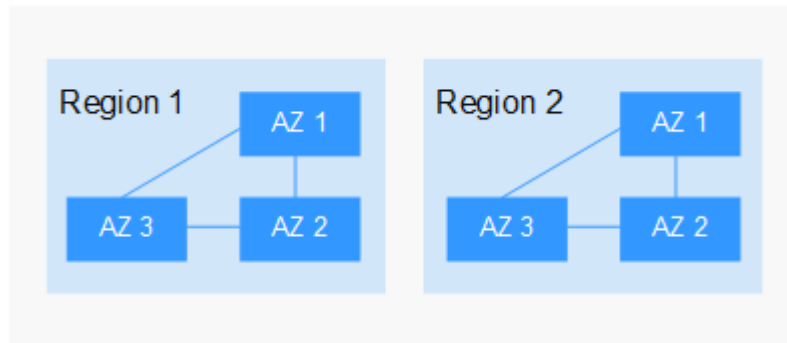
Concept

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 1-5](#) shows the relationship between regions and AZs.

Figure 1-5 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

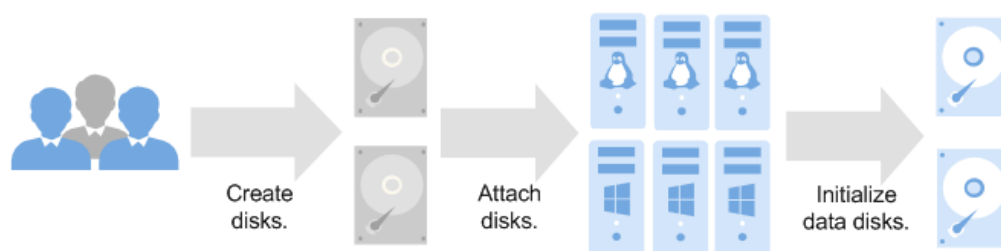
Before you use an API to call resources, specify its region and endpoint. For more details, see [Regions and Endpoints](#).

2 Getting Started

2.1 Process Overview

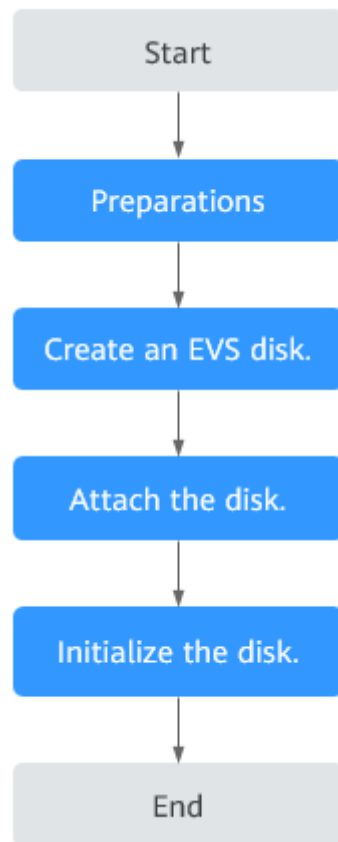
[Figure 2-1](#) shows the EVS process overview.

Figure 2-1 Process overview



[Figure 2-2](#) shows how to purchase a data disk separately.

Figure 2-2 Process overview



1. **Preparations:** Register an account on the console and obtain the permissions required to servers and disks.
2. **Create a disk.** Configure the disk parameters, including the disk type, size, name, and other information. For details, see [Create an EVS Disk](#).
3. **Attach the data disk.** Attach the separately created disk to a server. For details, see the following sections:
 - [Attaching a Non-Shared Disk](#)
 - [Attaching a Shared Disk](#)
4. **Initialize the data disk.** Log in to the server and initialize the data disk before using it. For details about how to initialize the disk, see the following sections:
 - [Introduction to Data Disk Initialization Scenarios and Partition Styles](#)
 - Windows
 - [Initializing a Windows Data Disk \(Windows Server 2008\)](#)
 - [Initializing a Windows Data Disk \(Windows Server 2019\)](#)
 - [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#)
 - [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2012\)](#)

- Linux
 - [Initializing a Linux Data Disk \(fdisk\)](#)
 - [Initializing a Linux Data Disk \(parted\)](#)
 - [Initializing a Linux Data Disk Larger Than 2 TiB \(parted\)](#)

2.2 Create an EVS Disk

Scenarios

EVS disks can be used as system disks or data disks for servers. You can create data disks on the EVS console, or create them together with system disks on the ECS console.

This section describes how to create data disks on the EVS console.

Constraints

Table 2-1 Constraints on creating disks

Created On	Description
The EVS console	• Disks created on the EVS console are data disks. You need to manually attach them to servers. • Disks can only be attached to servers in the same region and AZ. Once created, the region and AZ cannot be changed. • There are quantity and capacity quotas on EVS disks, so properly plan the number of disks and total disk capacity your workloads require. For details, see Managing Quotas.
The ECS console	• System disks can only be created together with servers and are automatically attached. • Data disks created together with servers or added after the server creation are automatically attached. • Disks will have the same billing mode as their server if the disks are created together with the server. • By default, disks created with ECSs are VBD disks, and those created with BMSs are SCSI disks.
-	Capacities of multiple disks cannot be combined, and the capacity of a single disk cannot be split.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Click **Create Disk**.

Step 4 Configure basic disk information according to [Table 2-2](#).

Table 2-2 Disk parameters

Parameter	Sub-Parameter	Description	Example Value
Region	-	Mandatory Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.	-
AZ	-	Mandatory 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 The available disk types are as follows: • Common I/O • High I/O • Ultra-high I/O NOTE When a disk is created from a snapshot, the disk type of the new disk will be consistent with that of the snapshot's source disk.	Ultra-high I/O
	Disk Size (GiB)	Mandatory The disk size. Only data disks can be created on the current page, and the disk size ranges from 10 GiB to 32,768 GiB.	100 GiB

Parameter	Sub-Parameter	Description	Example Value
Data Source (Optional)	Create from - Backup - Snapshot - Image	Optional - If you choose Create from Backup, the backup data is used to create the disk. Click Create from and choose 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. - If you choose Create from Snapshot, the snapshot data is used to create the disk. Click Create from and choose Snapshot. On the displayed page, select the target snapshot and click OK. NOTE - The device type of the new disk is the same as that of the snapshot's source disk. For details about the disk creation from snapshots, see Creating an EVS Disk from a Snapshot. - If you choose Create from Image, the image data is used to create the disk. Click Create from and choose Image. On the displayed page, select the target image and click OK. NOTE - The device type of the new disk is the same as that of the image's source disk.	Create from Backup: autobackup-001

Parameter	Sub-Parameter	Description	Example Value
Automatic Backup	-	CBR lets you back up EVS disks and ECSs and use the backups to restore data. After you configure automatic backup, the system will associate the EVS disk with the backup vault and apply the selected policy to the vault to periodically back up the disk. • Do not use: Skip this configuration if backup is not required. If you need backup protection after a disk has been created, log in to the CBR console, locate the desired vault, and associate the disk with the vault. • Use existing: 1. Vault: Select an existing vault from the drop-down list. 2. Backup Policy: Select a backup policy from the drop-down list, or go to the CBR console and configure a desired one. • Auto assign: 1. Enter a vault name, which can contain a maximum of 64 characters, including letters, digits, underscores (_), and hyphens (-), for example, vault-f61e. The default naming rule is vault_XXXX. 2. Enter the vault capacity, which is required for backing up the disk. The vault capacity cannot be less than the size of the disk to be backed up. The value ranges from the disk size to 10,485,760 in the unit of GiB. 3. Select a backup policy from the drop-down list, or go to the CBR console and configure a desired one.	-

Parameter	Sub-Parameter	Description	Example Value
More	Advanced Settings - Share - SCSI	Optional - Share If you select Share, a shared disk is created. A shared disk can be attached to up to 16 servers. If you do not select Share, a non-shared disk is created, and the disk can be attached to one a server only. If you select both SCSI and Share, a shared SCSI disk is created. NOTE The sharing attribute of a disk cannot be changed after the disk has been created. For details about shared EVS disks, see Managing Shared EVS Disks. - SCSI If you select SCSI, a SCSI disk is created. Such disks allow the server OS to directly access the underlying storage media and send SCSI commands to the disks. If you do not select SCSI, a VBD disk is created. That said, the disk device type is VBD, the default device type. NOTE The device type of a disk cannot be changed after the disk has been created. For details about the ECS types, OSs, and ECS software supported by SCSI EVS disks, see Device Types and Usage Instructions.	-
Disk Name	-	Mandatory - If you create disks individually, 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 .

Parameter	Sub-Parameter	Description	Example Value
Quantity	-	Optional The number of disks to be created. The default value is set to 1 , which means only one disk is created. Currently, you can create up to 100 disks at a time.	1

Step 5 Click **Next**.

Step 6 On the **Details** page, check the disk details.

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

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

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

----End

2.3 Attach an EVS Disk

2.3.1 Attaching a Non-Shared Disk

Scenarios

Separately created EVS 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.

A system disk must be created together with an ECS and is automatically attached. In the disk list, the function of such disks is displayed as **System disk**, and the status is displayed as **In-use**. After a system disk is detached from an ECS, the disk function changes to **Bootable disk**, and the status changes to **Available**.

NOTE

Bootable disks are the system disks detached from servers. A bootable disk can be re-attached to a server and be used as a system disk or data disk depending on the disk function selected.

This section describes how to attach a non-shared disk.

Prerequisites

- The non-shared disk status is **Available**.

Constraints

- Cloud servers created from ISO images are only used for OS installation. They have limited functions and cannot have EVS disks attached.
- A non-shared disk can be attached to one server only.
- The disk and the server must be in the same region and AZ.
- A shared disk can be attached only when the servers' statuses are **Running** or **Stopped**.
- A frozen disk cannot be attached.

Attaching the Disk on the EVS Console

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Locate the target disk in the list and click **Attach**.

Step 4 Select the server and then select the disk function from the drop-down list. Ensure that the disk and server are in the same AZ.

One device name can be used for one disk only. For how to obtain the disk name in the OS, see section "How Do I Obtain My Disk Name in the ECS OS Using the Device Identifier Provided on the Console?" in the *Elastic Cloud Server User Guide*.

Step 5 Click **OK** to return to the disk list page.

The status of the disk 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

Attaching the Disk on the ECS Console

1. Log in to the management console.
2. Under **Computing**, choose **Elastic Cloud Server**.
3. In the search box above the upper right corner of the ECS list, enter the ECS name, IP address, or ID for search.
4. Click the name of the target ECS.
The page providing details about the ECS is displayed.
5. Click the **Disks** tab. Then, click **Attach Disk**.
The **Attach Disk** dialog box is displayed.
6. Select the target disk and specify it as the system disk or a data disk.
 - For KVM ECSs, you can specify the disk as the system disk or a data disk but cannot specify a specific device name.

- For Xen ECSs, you can specify a specific device name, such as `/dev/vdb`.

 NOTE

- If no disks are available, click **Create Disk** in the lower part of the list.
 - For the restrictions on attaching disks, see FAQ "What Are the Requirements for Attaching an EVS Disk to an ECS?" in the *Elastic Cloud Server User Guide*.
7. Click **OK**.
- After the disk is attached, you can view the information about it on the **Disks** tab.

Follow-Up Operations

If you are attaching a new disk, you must then log in to the server and initialize the disk before it can be used. To learn how to initialize disks, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

2.3.2 Attaching a Shared Disk

Scenarios

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

Prerequisites

- The shared disk status is **In-use** or **Available**.

Constraints

NOTICE

If you simply attach a shared disk to multiple servers, files cannot be shared among them. Because there are no mutually agreed data read/write rules among servers, read and write operations from them may interfere with each other, or unpredictable errors may occur. To share files between servers, set up a shared file system or a clustered management system first.

- A shared disk can be attached to a maximum of 16 servers. These servers and the shared disk must be in the same AZ within a region.
- A shared, **In-use** disk can be attached to other servers only when the maximum number of servers that the disk can be attached to has not been reached.
- All the servers attached with a shared disk must run either Windows or Linux. For example, if you attach a shared disk to multiple Windows servers and then detach it from these servers, the shared disk cannot be attached to Linux servers later. This is because Windows and Linux support different file systems and cannot identify the original file system on the disk. Improper operations may damage the original file system.

- A shared disk can only be used as a data disk. It cannot be used as a system disk.
- Cloud servers created from ISO images are only used for OS installation. They have limited functions and cannot have EVS disks attached.
- A shared disk can be attached only when the servers' statuses are **Running** or **Stopped**.
- A frozen disk cannot be attached.

Attaching the Disk on the EVS Console

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Locate the target disk in the list and click **Attach**.

Shared disks support batch attachment, so you can attach a shared disk to multiple servers. 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 4 Select the target servers you want to attach the shared disk. Ensure that the disk and servers are in the same AZ. After you select the servers, **Data disk** is automatically entered as the disk function for each server.

Step 5 Click **OK** to return to the disk list page.

The status of the disk 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.

----End

Attaching the Disk on the ECS Console

1. Log in to the management console.
2. Under **Compute**, click **Elastic Cloud Server**.
3. In the search box above the upper right corner of the ECS list, enter the ECS name, IP address, or ID for search.
4. Click the name of the target ECS.
The page providing details about the ECS is displayed.
5. Click the **Disks** tab. Then, click **Attach Disk**.
The **Attach Disk** page is displayed.
6. Select the target disk and specify it as the system disk or a data disk.
 - For Xen ECSs, you can specify a specific device name, such as **/dev/sdb**.
 - For KVM ECSs, you can specify the disk as the system disk or a data disk but cannot specify a specific device name.

 NOTE

- If no disks are available, click **Create Disk** in the lower part of the list.
- For the restrictions on attaching disks, see FAQ "What Are the Requirements for Attaching an EVS Disk to an ECS?" in the *Elastic Cloud Server User Guide*.

7. Click **OK**.

After the disk is attached, you can view the information about it on the **Disks** tab.

Follow-Up Operations

If you are attaching a new disk, you must then log in to the server and initialize the disk before it can be used. To learn how to initialize disks, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

2.4 Initialize an EVS Data Disk

2.4.1 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 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 an appropriate partition style based on your service plan.

Prerequisites

- The disk has been attached to a server.
- Data may be lost after a disk is initialized, so you are advised to create [snapshots](#) or [backups](#) before initializing a disk.

Constraints

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

- Initializing a disk does not change the server's IP address or the disk ID.
- Initializing a disk does not delete the snapshots created for the disk, so you can still roll back snapshots to the original disk after the disk is initialized.

Disk Partition Styles

Table 2-3 lists the common disk partition styles. In Linux, different 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 TiB	• 4 primary partitions • 3 primary partitions and 1 extended partition With MBR, you can create several primary partitions and one extended partition. The extended partition must be divided into logical partitions before use. For example, if 6 partitions need to be created, you can create them 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 EiB 1 EiB = 1048576 TiB	Unlimited Disk partitions created using GPT are not categorized.	parted

NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

Partitioning Operation Guide

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

- [Initializing a Windows Data Disk \(Windows Server 2008\)](#)
- [Initializing a Windows Data Disk \(Windows Server 2019\)](#)
- [Initializing a Linux Data Disk \(fdisk\)](#)
- [Initializing a Linux Data Disk \(parted\)](#)

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

- [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#)
- [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2012\)](#)
- [Initializing a Linux Data Disk Larger Than 2 TiB \(parted\)](#)

2.4.2 Initializing a Windows Data Disk (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 TiB, and that supported by GPT is 18 EiB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TiB. For details, see [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#). To learn more about disk partition styles, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

Procedure

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

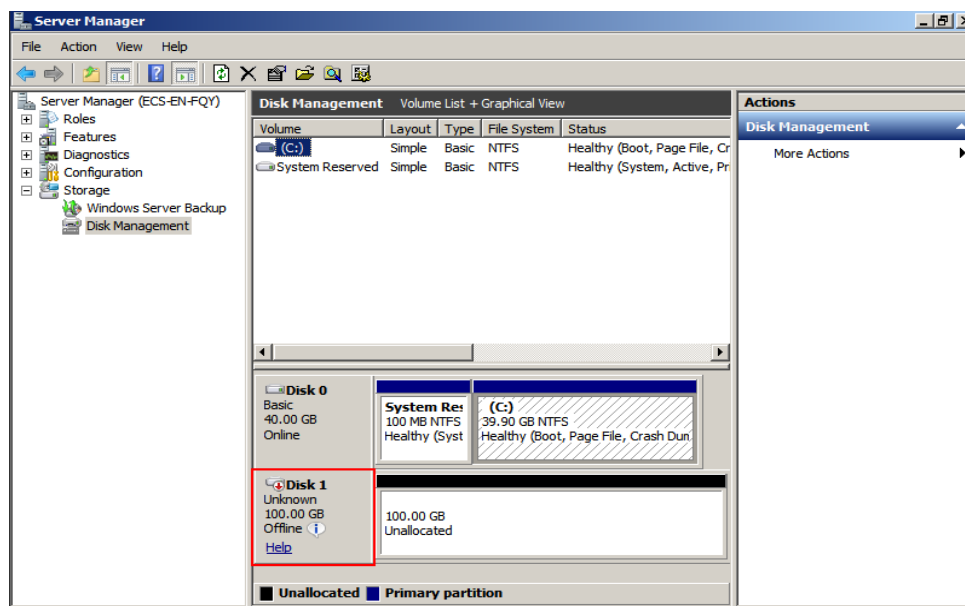
The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

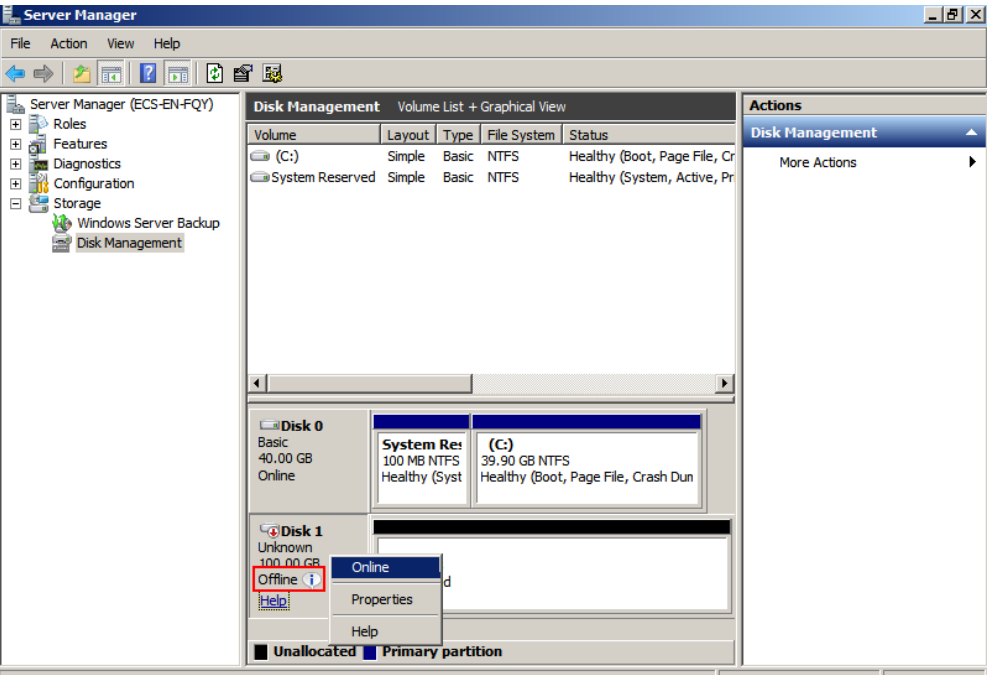
- If [Figure 2-3](#) is displayed, the new disk is offline. Go to [Step 3](#).
- If [Figure 2-6](#) is displayed, the **Initialize Disk** window is prompted. Go to [Step 5](#).

Figure 2-3 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 2-4 Online the disk

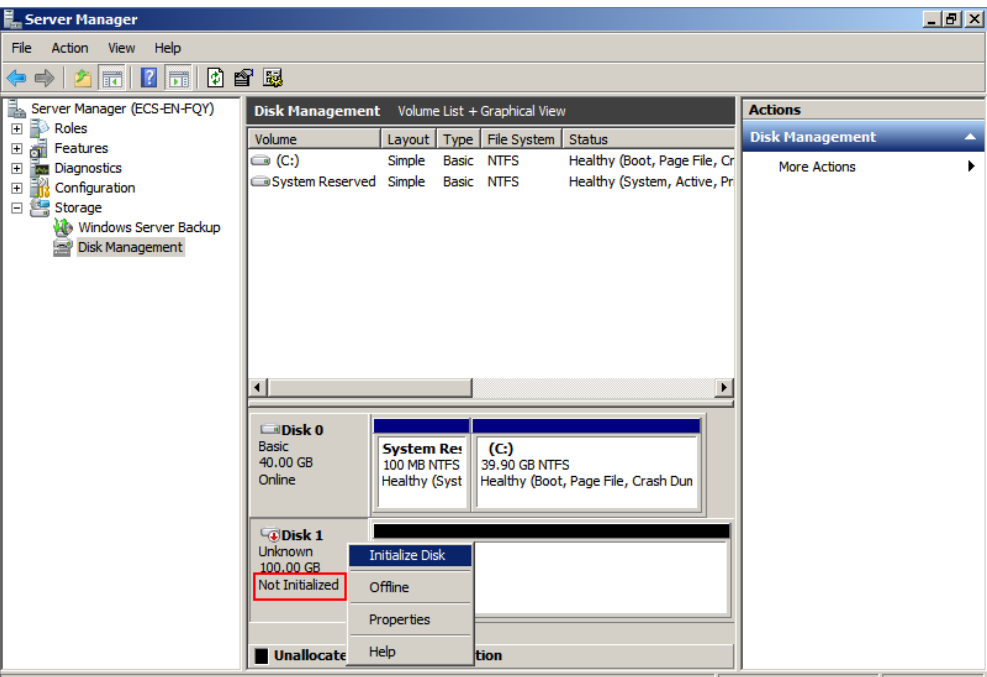


NOTE

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

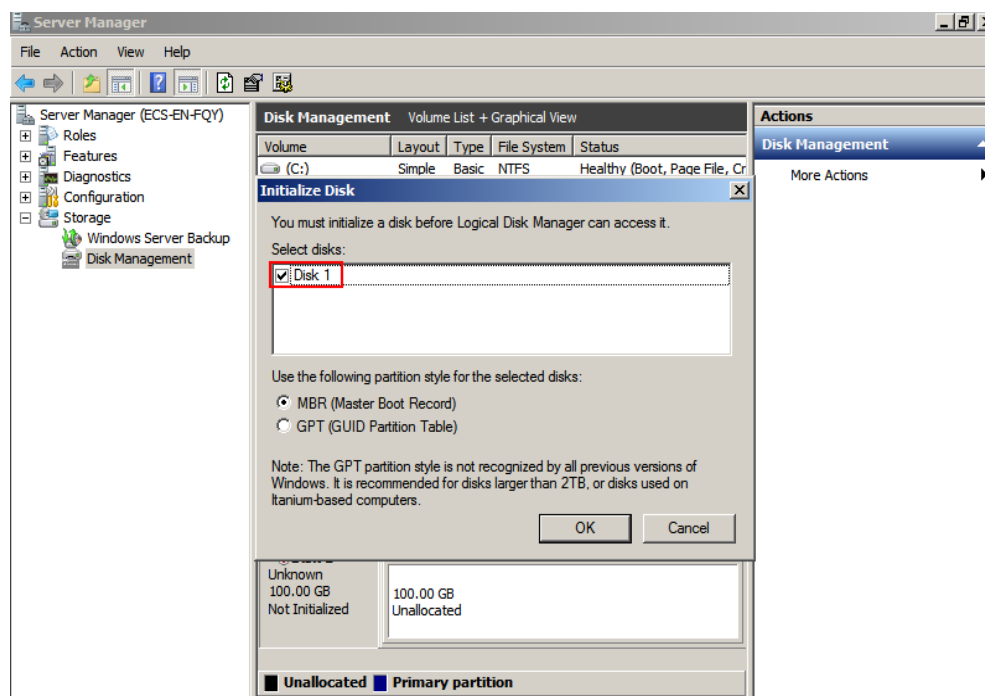
- Step 4** After making the disk online, the disk status changes from **Offline** to **Not Initialized**. Right-click the disk status and choose **Initialize Disk** from the shortcut menu.

Figure 2-5 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**.

Figure 2-6 Unallocated space



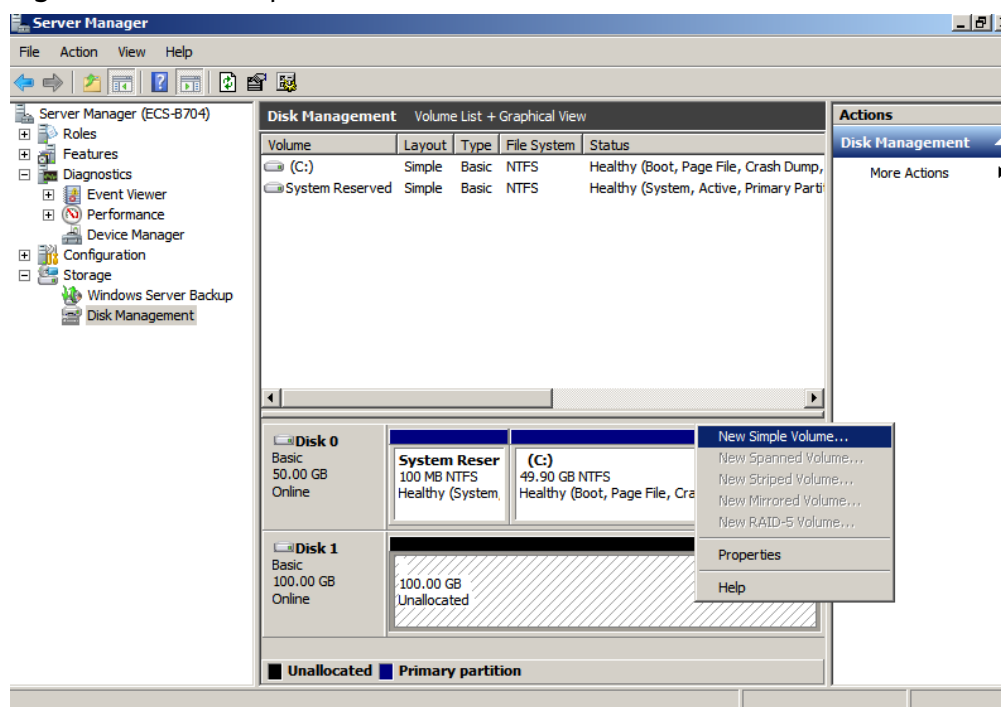
NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

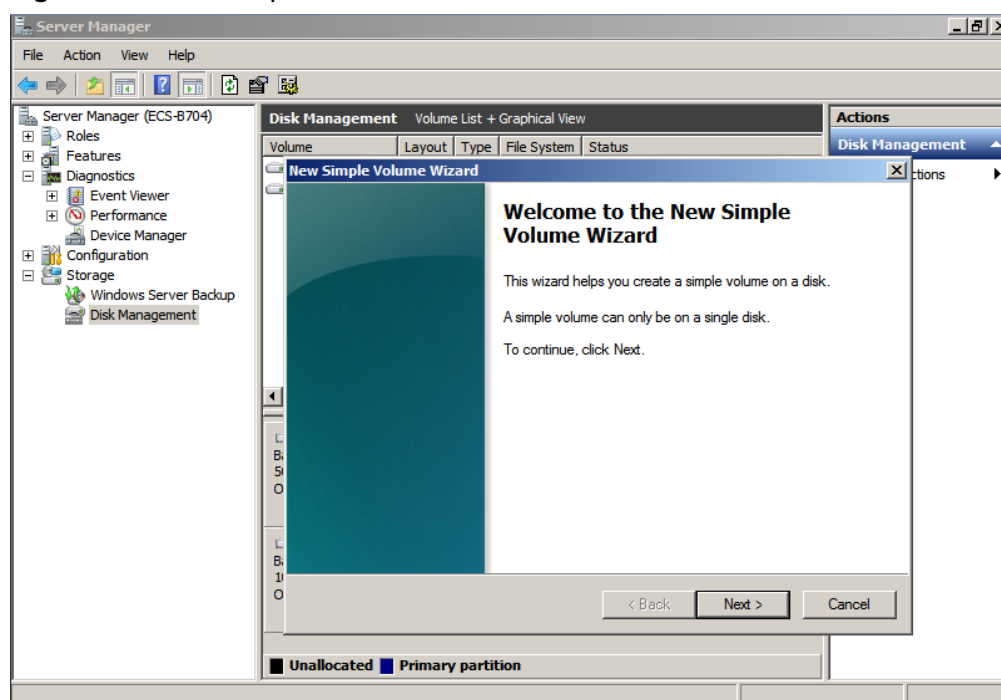
- Step 6** Right-click at the unallocated space and choose **New Simple Volume** from the shortcut menu.

Figure 2-7 New Simple Volume



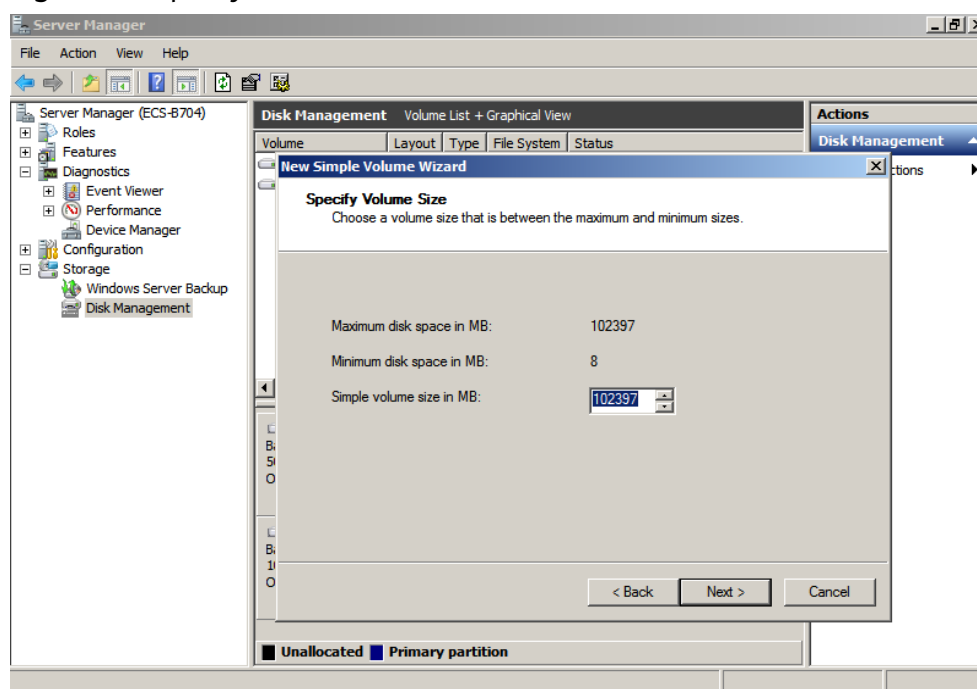
Step 7 On the displayed **New Simple Volume Wizard** window, click **Next**.

Figure 2-8 New Simple Volume Wizard



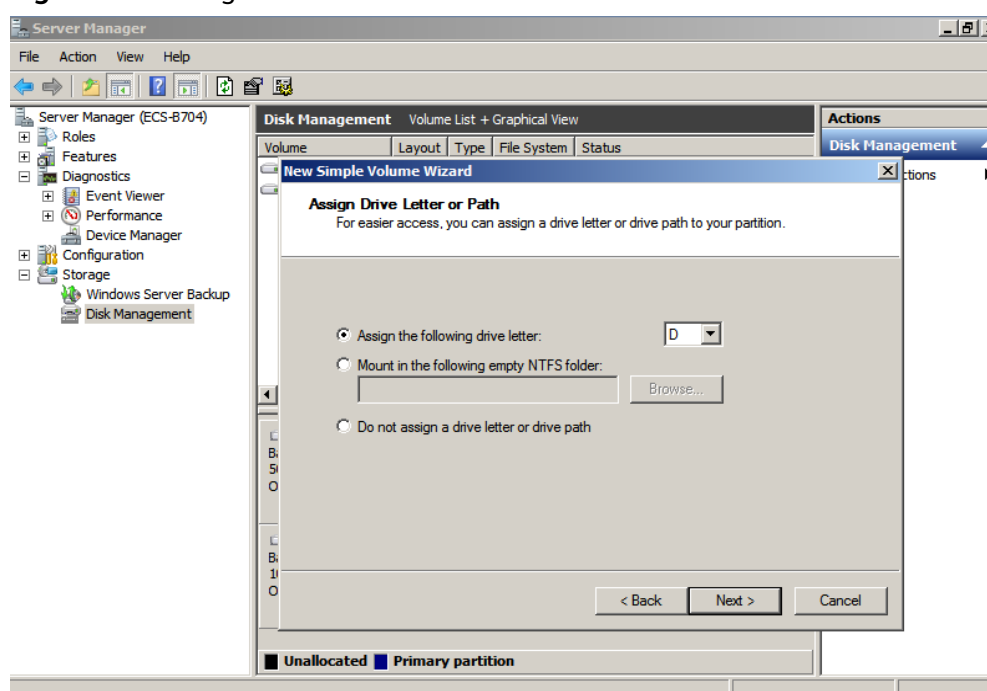
Step 8 Specify the volume size and click **Next**. The default value is the maximum size.

Figure 2-9 Specify Volume Size



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

Figure 2-10 Assign Drive Letter or Path



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

Figure 2-11 Format Partition

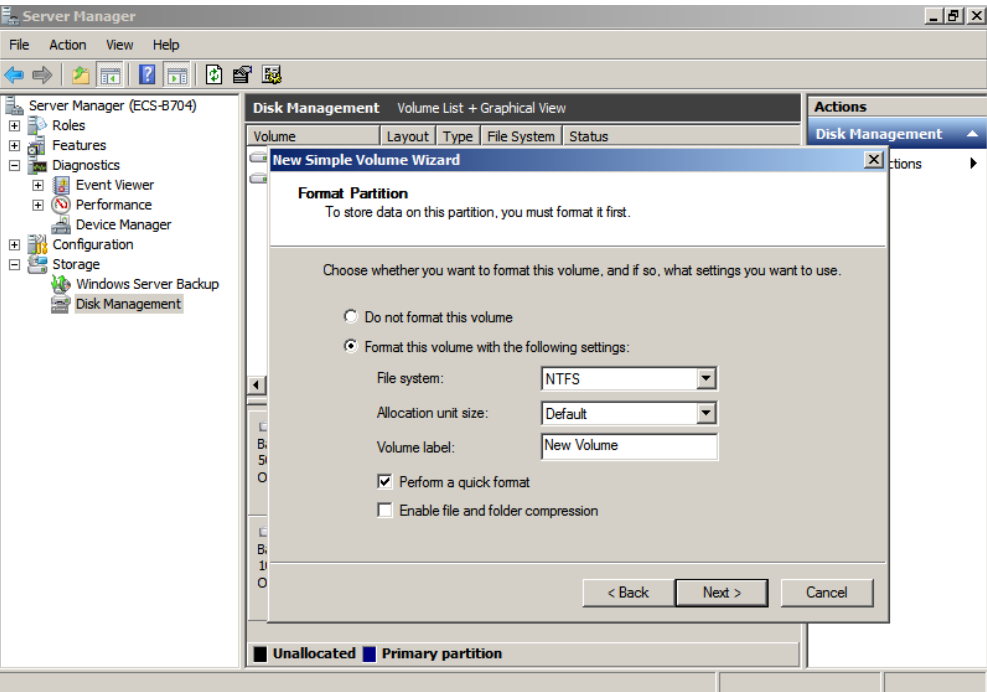
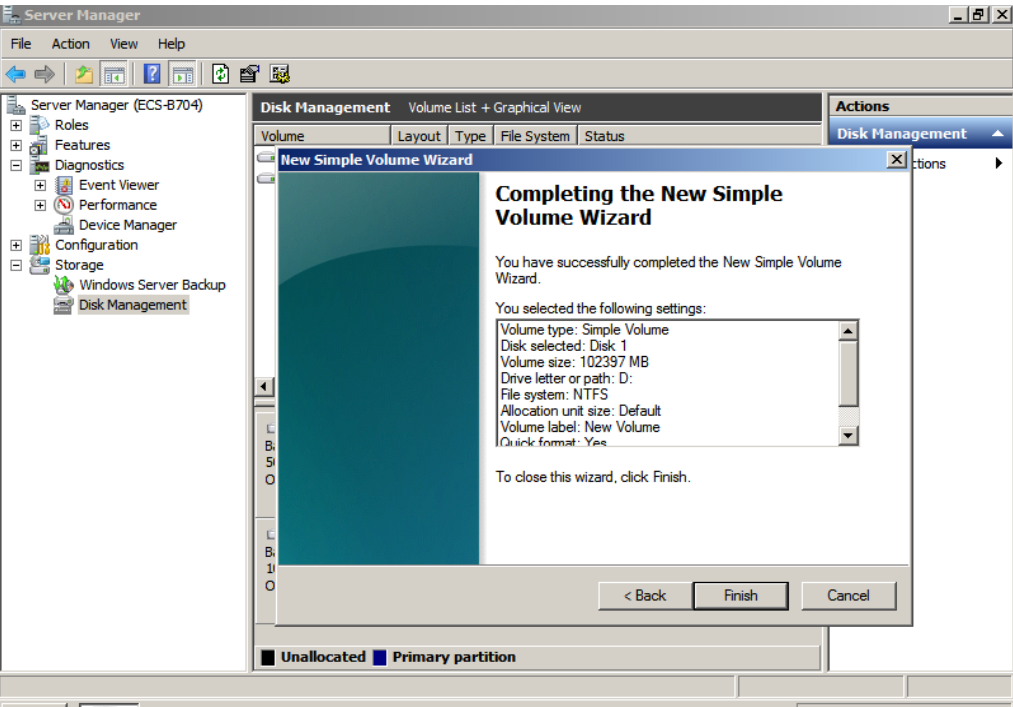


Figure 2-12 Completing the partition creation

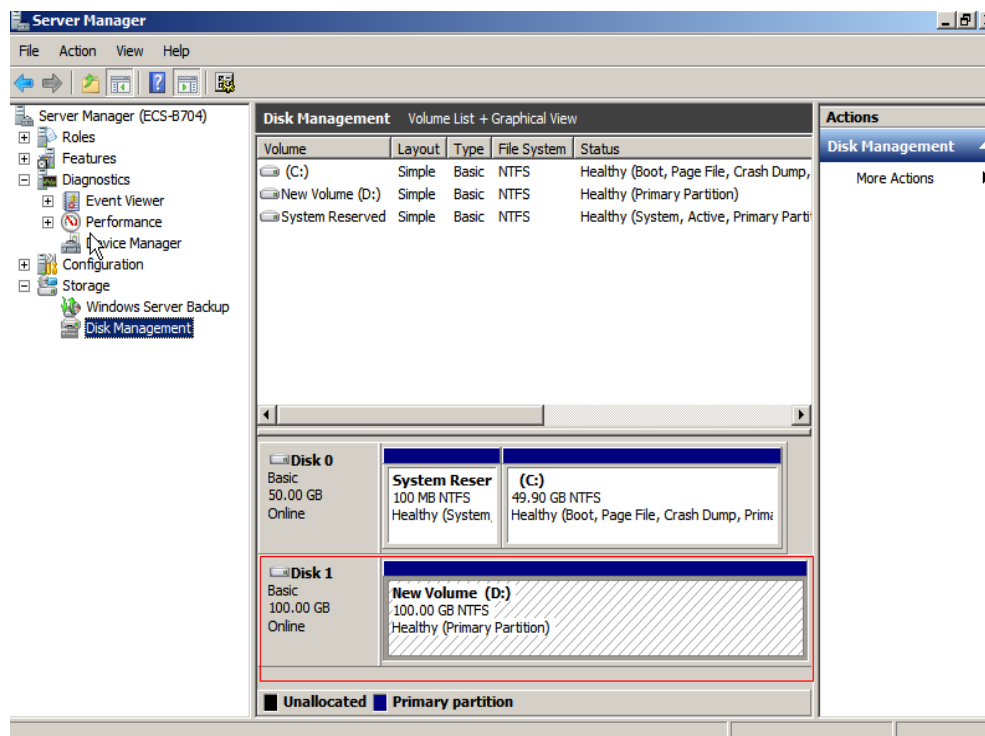


NOTICE

The partition sizes supported by file systems vary. Choose an appropriate file system format 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.

Figure 2-13 Disk initialization succeeded



----End

2.4.3 Initializing a Windows Data Disk (Windows Server 2019)

Scenarios

This section uses Windows Server 2019 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 TiB, and that supported by GPT is 18 EiB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TiB. For details, see [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#). To learn more about disk partition styles, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

Procedure

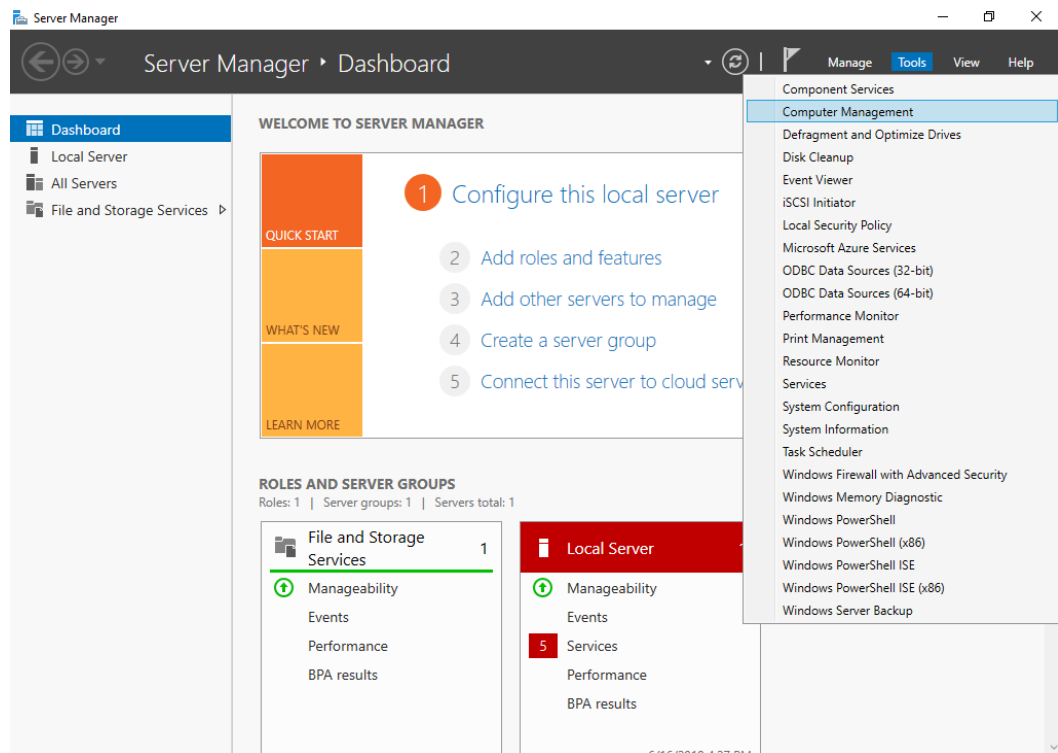
Step 1 On the desktop of the server, click the start icon in the lower left corner.

The **Windows Server** window is displayed.

Step 2 Click **Server Manager**.

The **Server Manager** window is displayed.

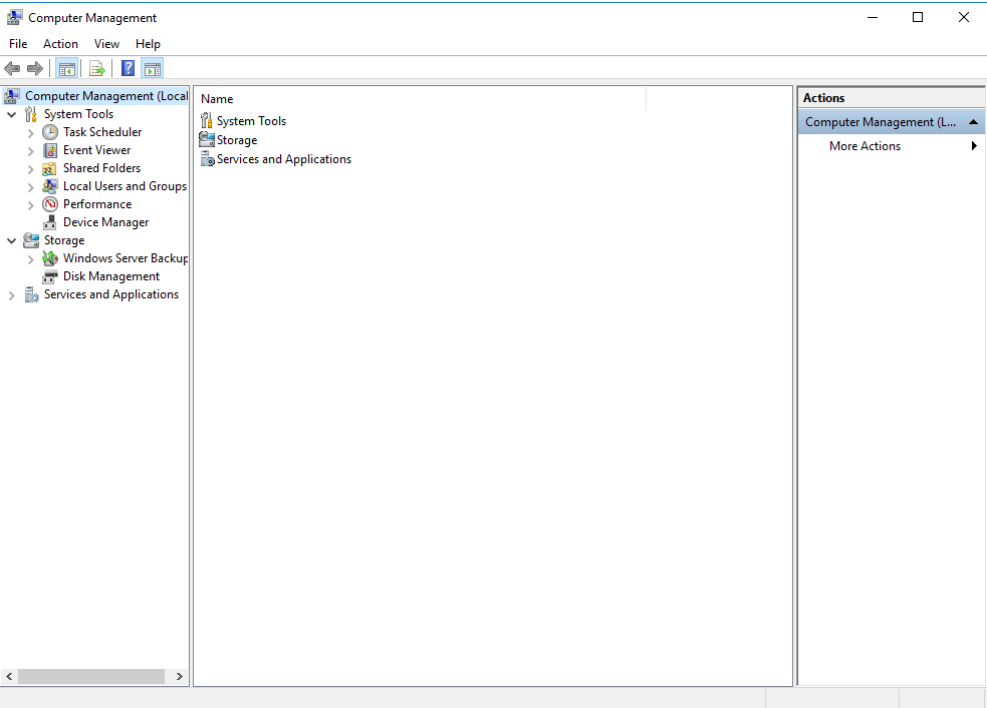
Figure 2-14 Server Manager



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

The **Computer Management** window is displayed.

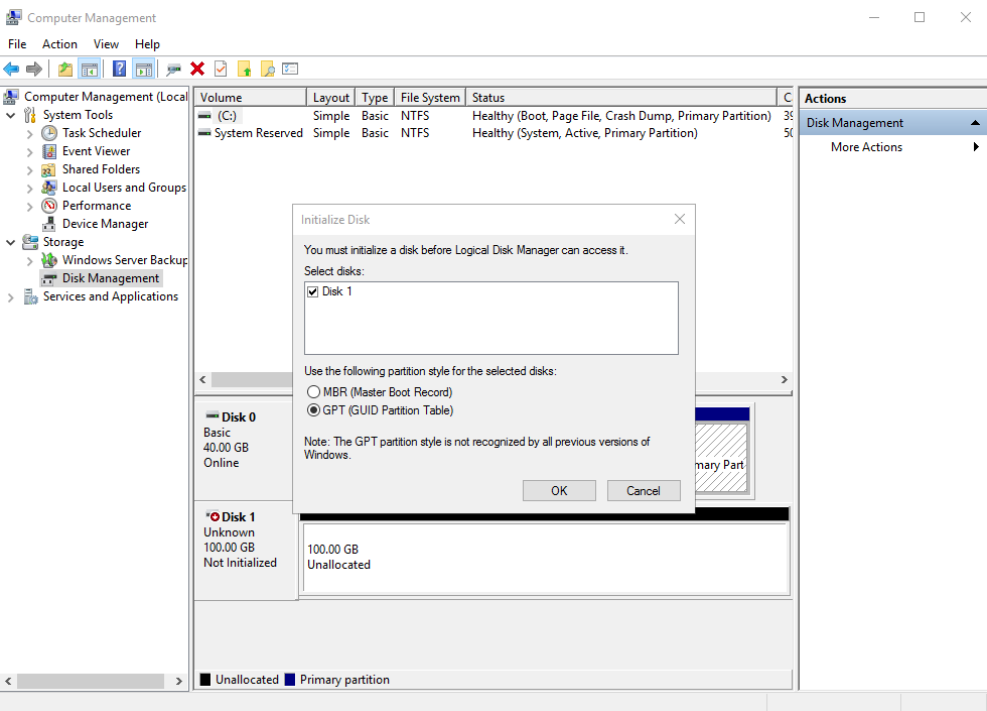
Figure 2-15 Computer Management



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.

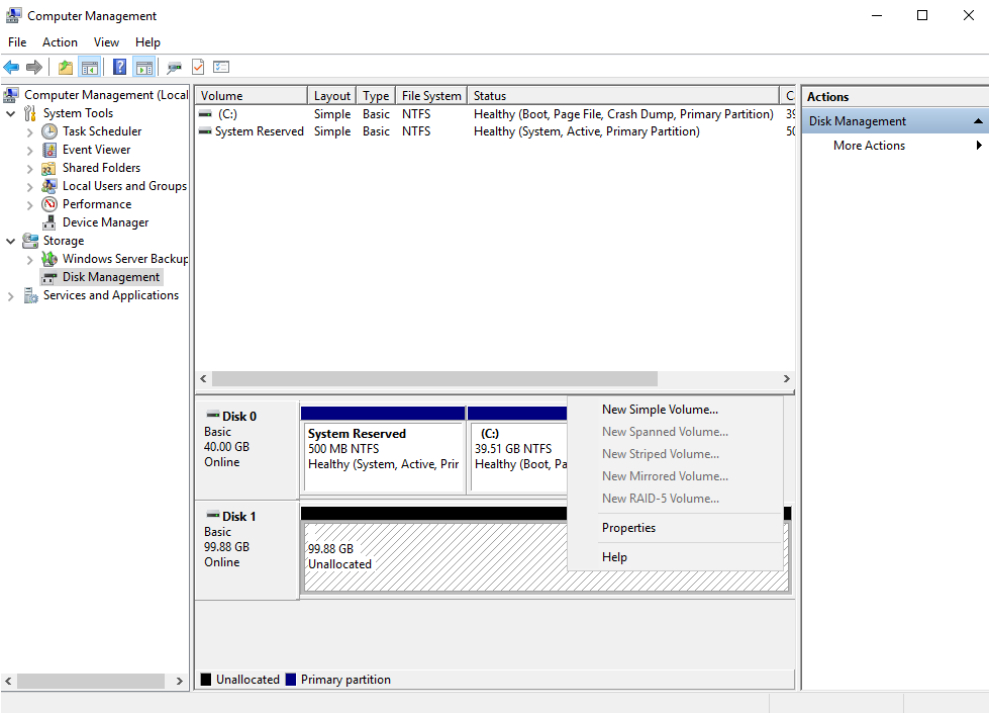
Figure 2-16 Disk list



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 2-17 Computer Management



NOTICE

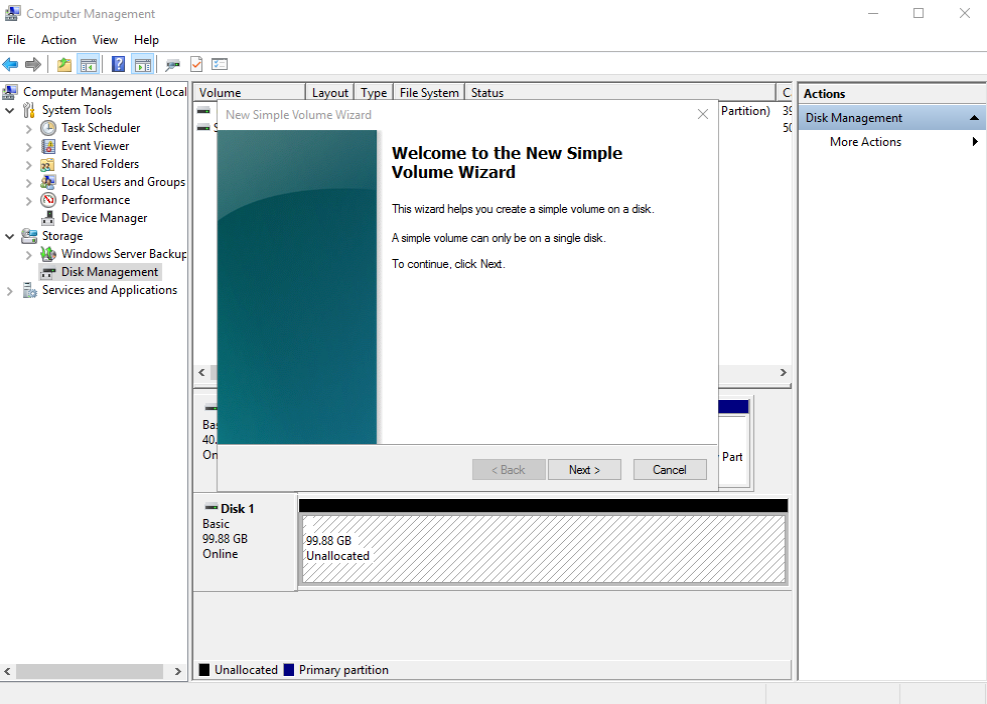
The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

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

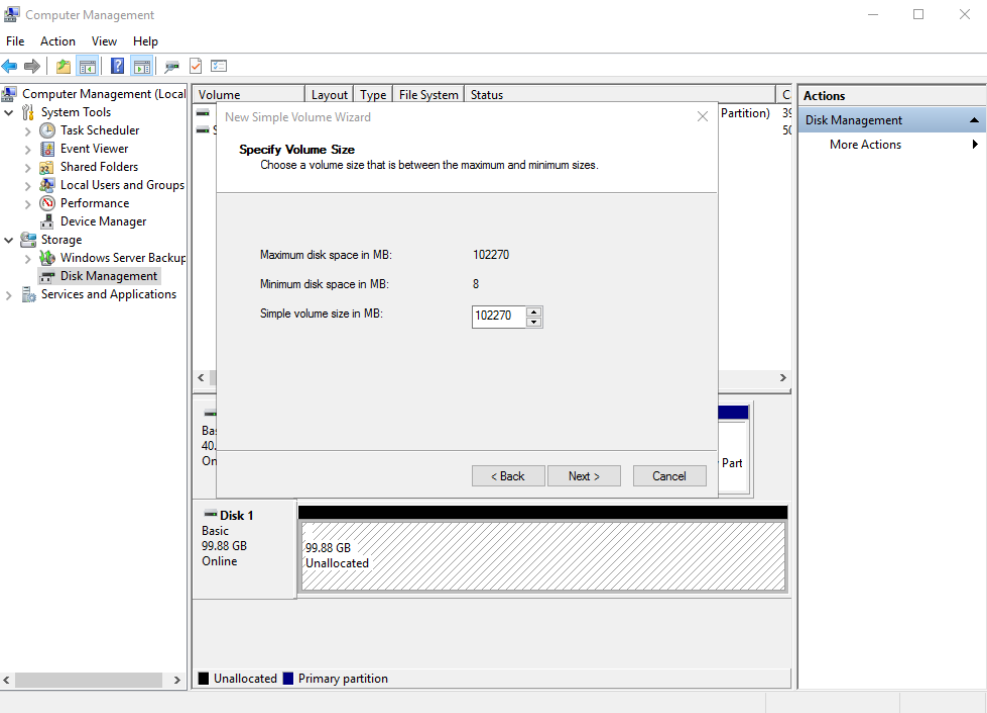
The **New Simple Volume Wizard** window is displayed.

Figure 2-18 New Simple Volume Wizard



Step 7 Follow the prompts and click **Next**.
The **Specify Volume Size** page is displayed.

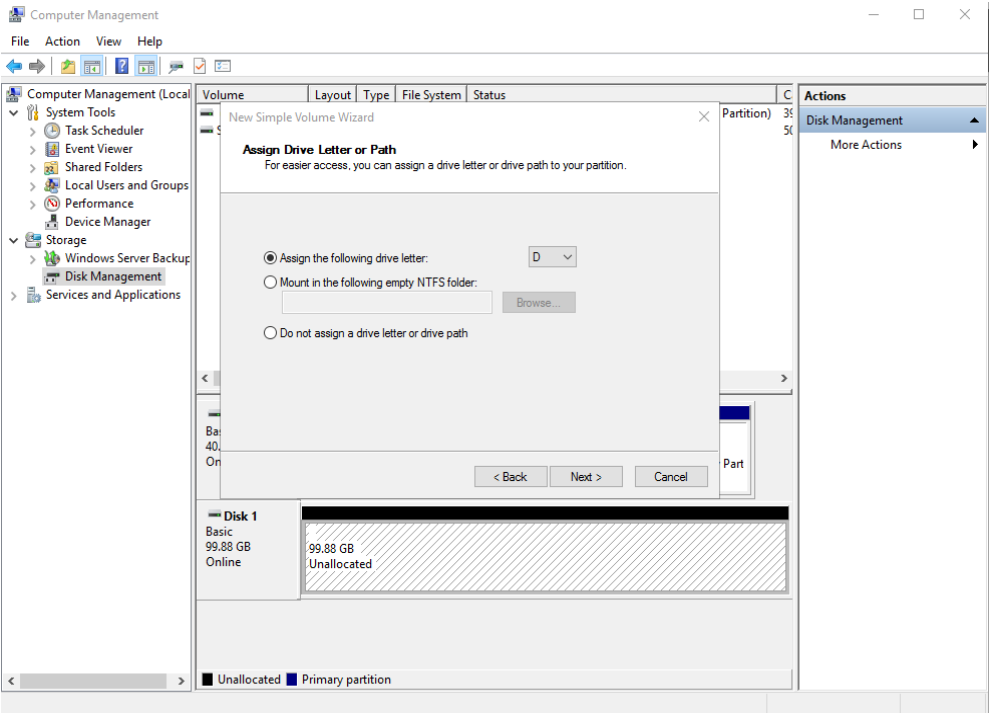
Figure 2-19 Specify Volume Size



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.

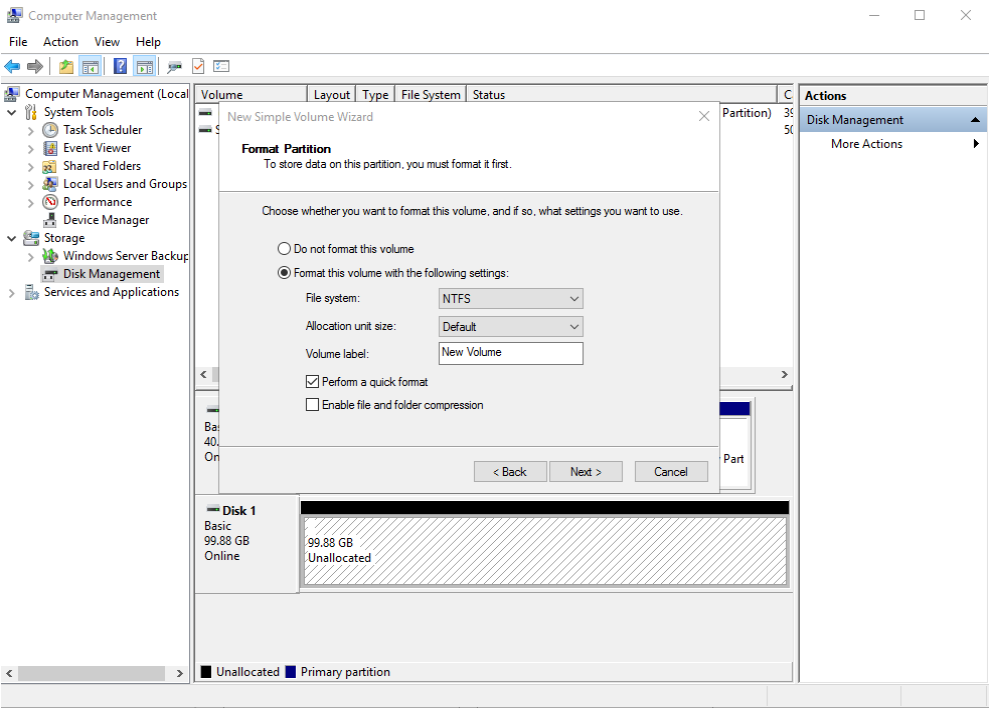
Figure 2-20 Assign Drive Letter or Path



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.

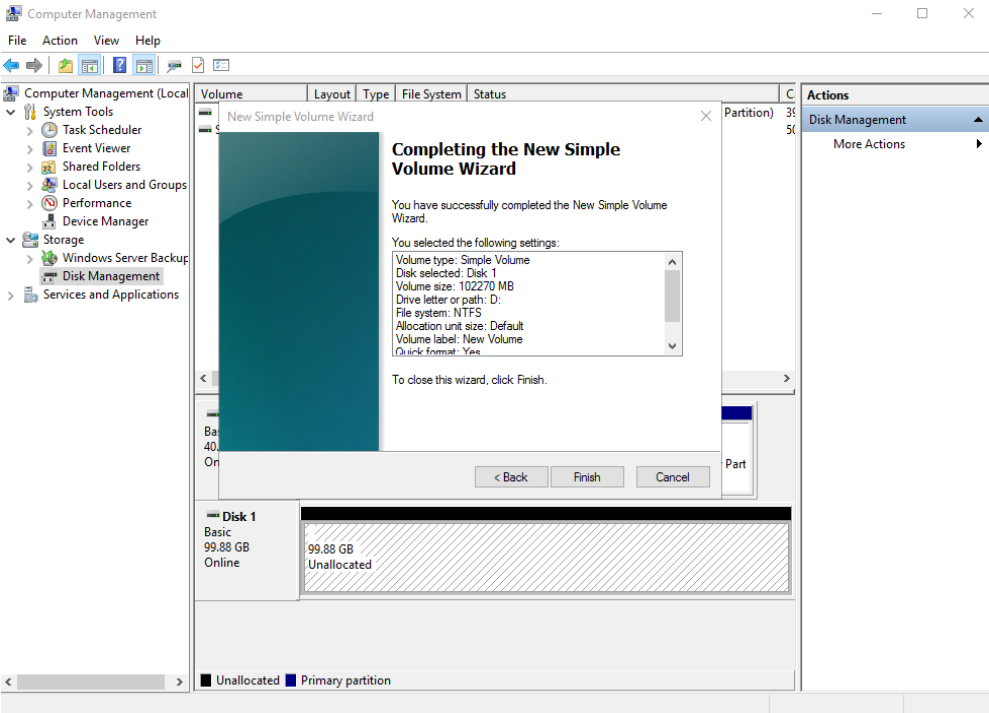
Figure 2-21 Format Partition



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 2-22 Completing the New Simple Volume Wizard



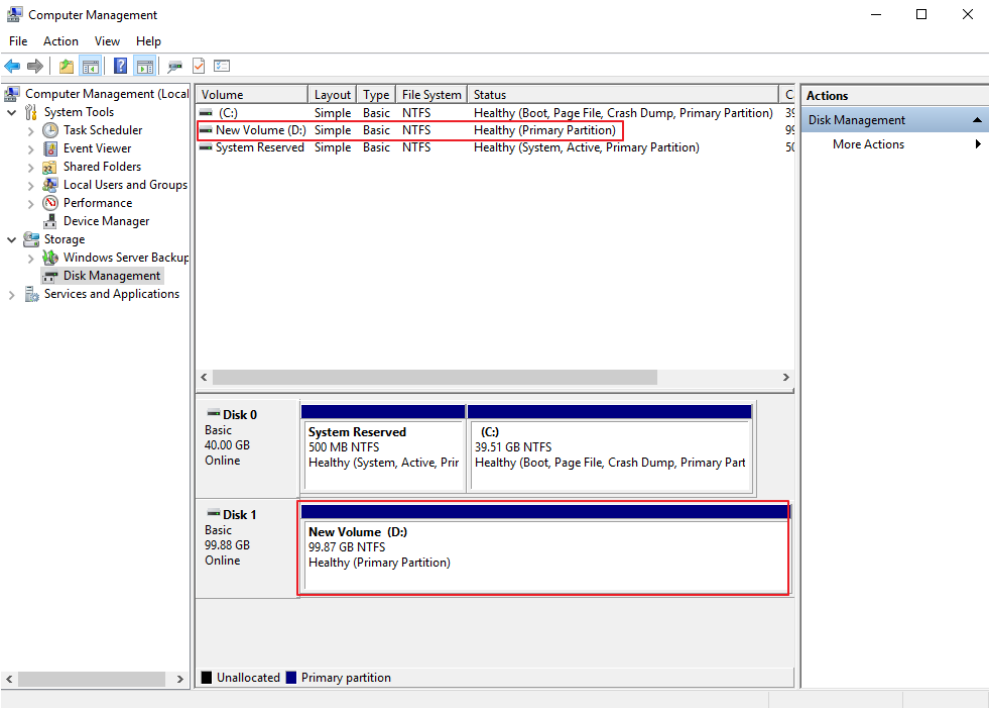
NOTICE

The partition sizes supported by file systems vary. Choose an appropriate file system format based on your service requirements.

Step 11 Click **Finish**.

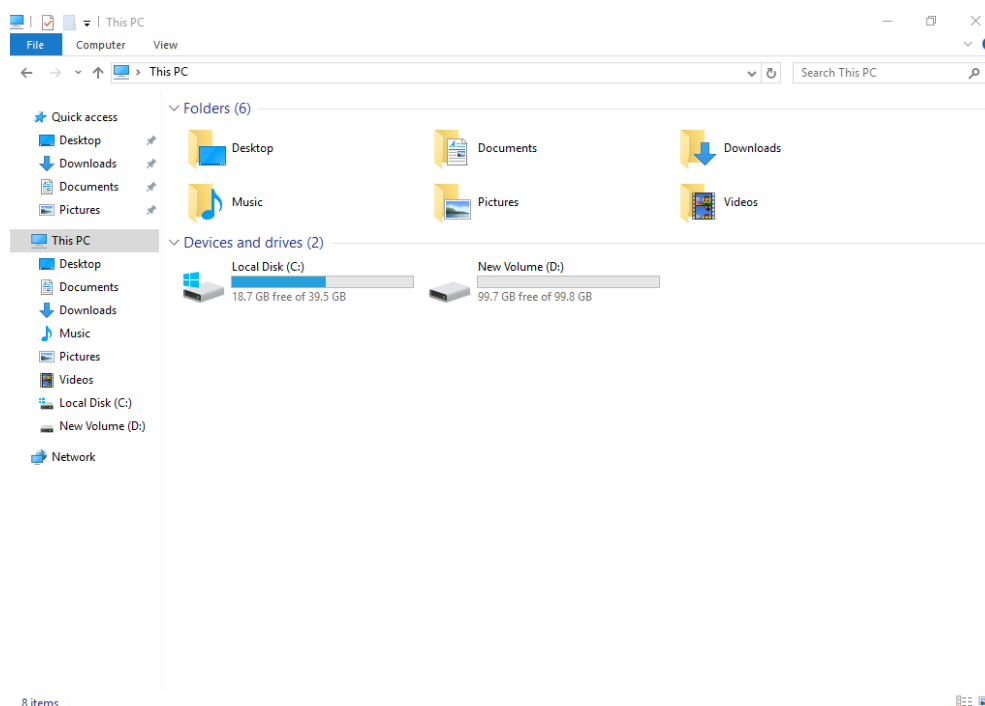
Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully.

Figure 2-23 Disk initialized



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 2-24 This PC

----End

2.4.4 Initializing a Linux Data Disk (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 partition size that MBR supports is 2 TiB and that GPT supports is 18 EiB. If the disk size you need to partition is greater than 2 TiB, partition the disk using GPT.

The fdisk partitioning tool is suitable only for MBR partitions, and the parted partitioning tool is suitable for both MBR and GPT partitions. For more information, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

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 will be used. Furthermore, the partition will be formatted using the file system, mounted on `/mnt/sdc`, and configured to mount automatically at startup.

Step 1 Query what block devices are available on the server.

fdisk -l

Information similar to the following is displayed:

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

```
Disk /dev/vda: 42.9 GiB, 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 GiB, 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, this server contains two disks. `/dev/vda` and `/dev/vdb`. `/dev/vda` is the system disk, and `/dev/vdb` is the new data disk.

Step 2 Launch `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 MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

Step 4 Enter **p** and press **Enter** to create a primary partition in this example.

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 print the partition details.

Information similar to the following is displayed:

Command (m for help): p

```
Disk /dev/vdb: 107.4 GiB, 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 Synchronize the new partition table to the OS.

partprobe

Step 11 Format the new partition with a desired file system format.

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

In this example, the **ext4** format is used 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. Choose an appropriate file system format based on your service requirements.

Step 12 Create a mount point.

mkdir *Mount point*

In this example, the **/mnt/sdc** mount point is created.

mkdir /mnt/sdc

 NOTE

The **/mnt** directory exists on all Linux systems. If the mount point cannot be created, it may be that the **/mnt** directory has been accidentally deleted. You can run **mkdir -p /mnt/sdc** to create the mount point.

Step 13 Mount the new partition on the created mount point.

mount *Disk partition Mount point*

In this example, the **/dev/vdb1** partition is mounted on **/mnt/sdc**.

mount /dev/vdb1 /mnt/sdc

Step 14 Check 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
```

You should now see that partition **/dev/vdb1** is mounted on **/mnt/sdc**.

 NOTE

After the server is restarted, the disk will not be automatically mounted. You can modify the **/etc/fstab** file to configure automount at startup. For details, see [Configuring Automatic Mounting at System Start](#).

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at startup. You can use **fstab** to configure your data disks to mount automatically. This operation will not affect the existing data.

The example here uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names to identify disks in the file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a server stop or start. This can even prevent the server from booting up.

NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi /etc/fstab

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID:** The UUID obtained in [Step 1](#).
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.

- **0**: not use fsck.
- If the mount point is the root partition (/), this parameter must be set to **1**.
If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

2.4.5 Initializing a Linux Data Disk (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 partition size that MBR supports is 2 TiB and that GPT supports is 18 EiB. If the disk size you need to partition is greater than 2 TiB, partition the disk using GPT.

The fdisk partitioning tool is suitable only for MBR partitions, and the parted partitioning tool is suitable for both MBR and GPT partitions. For more information, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

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 will be used. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured to mount automatically at startup.

Step 1 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, this server contains two disks. **/dev/vda** and **/dev/vdb**. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk.

Step 2 Launch 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: 107GiB
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 Set the disk partition style.

mklabel *Disk partition style*

This command lets you control whether to use MBR or GPT for your partition table. In this example, GPT is used.

mklabel gpt

NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

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: 107GiB
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 Create a new partition.

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, starting on **2048** and using **100%** of the rest of the disk. 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 print the partition details.

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 configure automatic mounting by updating the **/etc/fstab** file. Before doing so, format the partition with a desired file system and mount the partition on the mount point.

Step 10 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 Format the new partition with a desired file system format.

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

In this example, the **ext4** format is used 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. Choose an appropriate file system format based on your service requirements.

Step 12 Create a mount point.

mkdir *Mount point*

In this example, the **/mnt/sdc** mount point is created.

mkdir /mnt/sdc

 NOTE

The **/mnt** directory exists on all Linux systems. If the mount point cannot be created, it may be that the **/mnt** directory has been accidentally deleted. You can run **mkdir -p /mnt/sdc** to create the mount point.

Step 13 Mount the new partition on the created mount point.

mount *Disk partition Mount point*

In this example, the **/dev/vdb1** partition is mounted on **/mnt/sdc**.

mount /dev/vdb1 /mnt/sdc

Step 14 Check 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
```

You should now see that partition **/dev/vdb1** is mounted on **/mnt/sdc**.

 NOTE

After the server is restarted, the disk will not be automatically mounted. You can modify the **/etc/fstab** file to configure automount at startup. For details, see [Configuring Automatic Mounting at System Start](#).

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at startup. You can configure the **fstab** file of a that has data. This operation will not affect the existing data.

The following example uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names (like **/dev/vdb1**) to identify disks in the file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a stop or start. This can even prevent your from booting up.

NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi /etc/fstab

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID:** The UUID obtained in [Step 1](#).
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.

- **0**: not use fsck.
- If the mount point is the root partition (/), this parameter must be set to **1**.
If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

2.4.6 Initializing a Windows Data Disk Larger Than 2 TiB (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 TiB. In the following operations, the capacity of the example disk is 3 TiB.

The maximum disk capacity supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TiB. For details, see [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#). To learn more about disk partition styles, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

Procedure

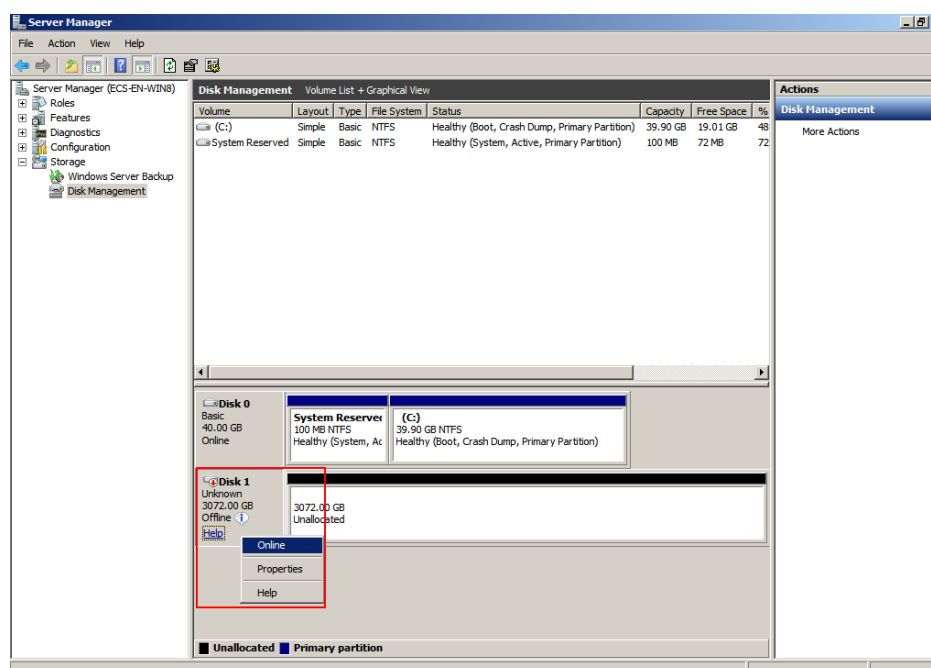
Step 1 On the desktop of the server, click **Start**.

The **Start** window is displayed.

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

The **Server Manager** window is displayed.

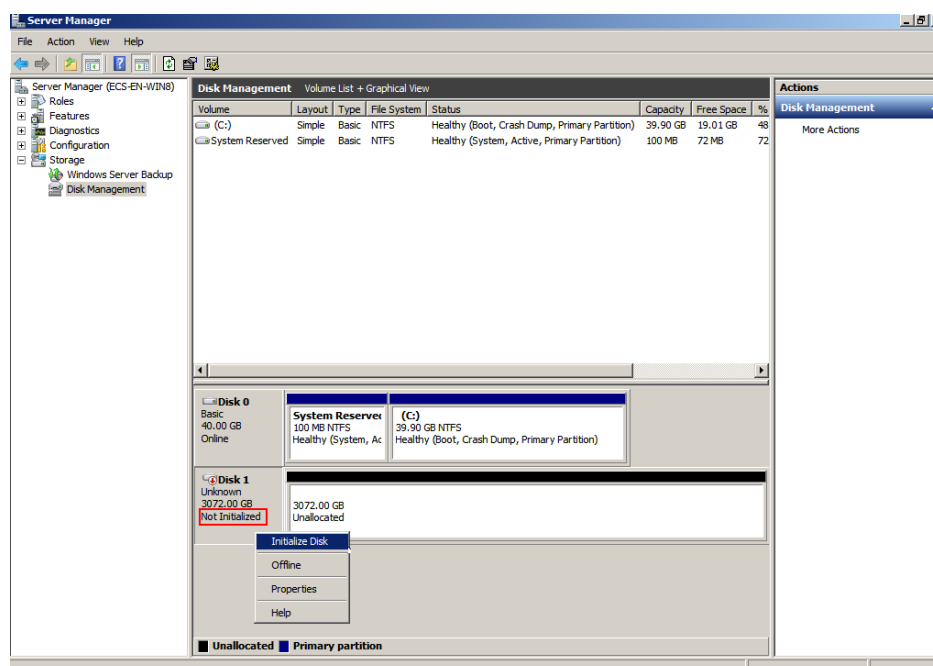
Figure 2-25 Server Manager (Windows Server 2008)



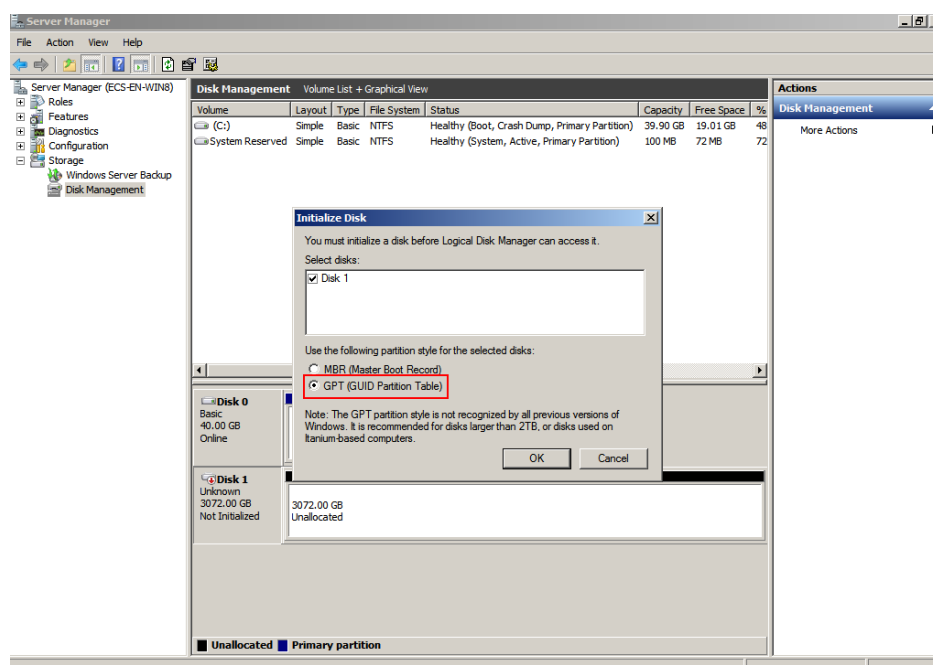
Step 3 Disks are listed in the right pane. If the new disk is offline, bring it online before initializing it.

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 2-26 Bring online succeeded (Windows Server 2008)

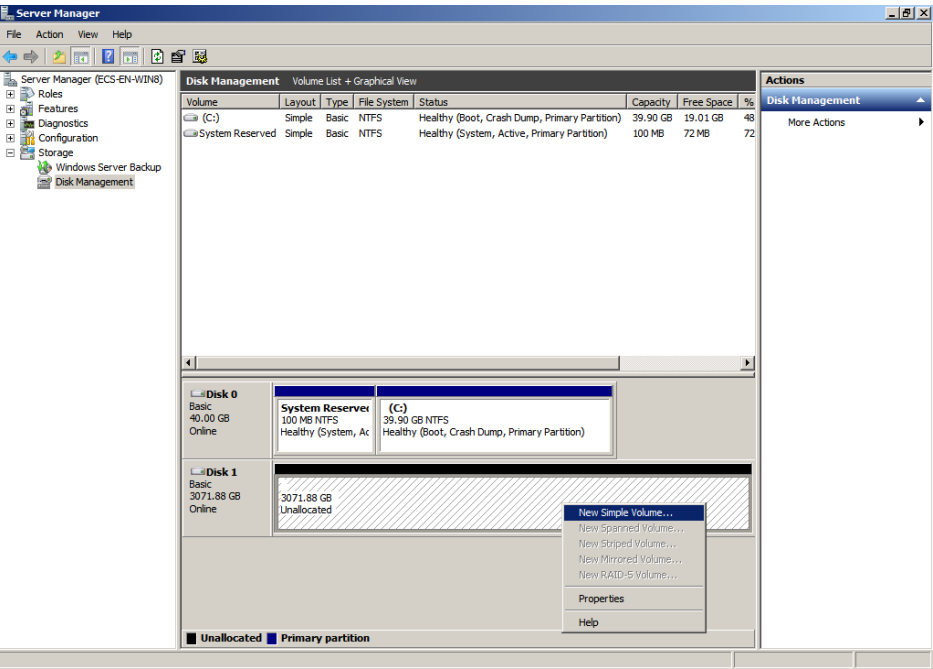
Step 4 In the **Disk 1** area, right-click and choose **Initialize Disk** from the shortcut menu.
The **Initialize Disk** dialog box is displayed.

Figure 2-27 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 TiB. Therefore, select **GPT (GUID Partition Table)** and click **OK**.

The **Server Manager** window is displayed.

Figure 2-28 Server Manager (Windows Server 2008)



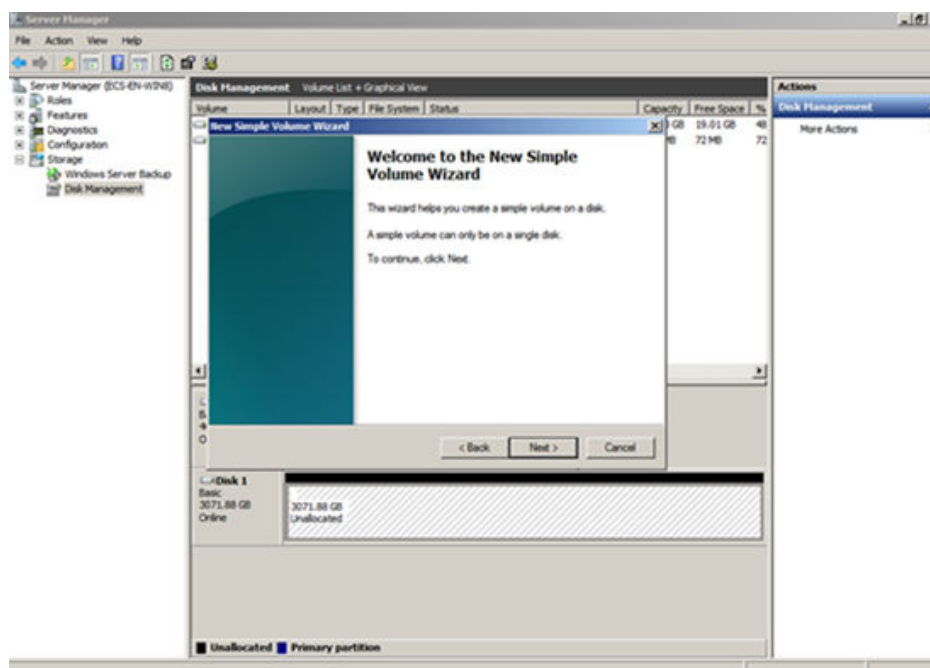
NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

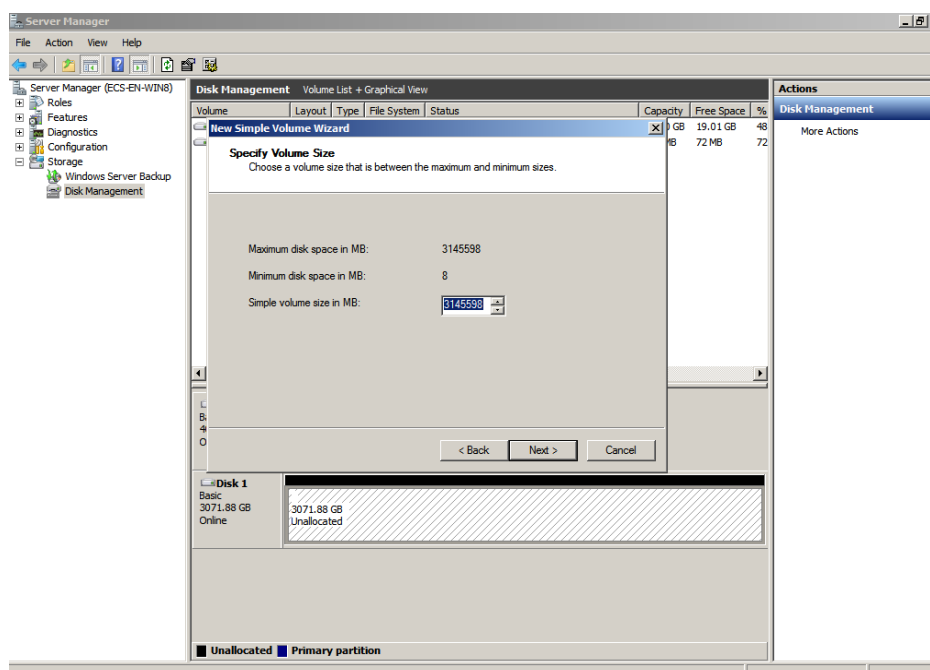
Step 6 Right-click at the unallocated disk space and choose **New Simple Volume** from the shortcut menu.

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

Figure 2-29 New Simple Volume Wizard (Windows Server 2008)

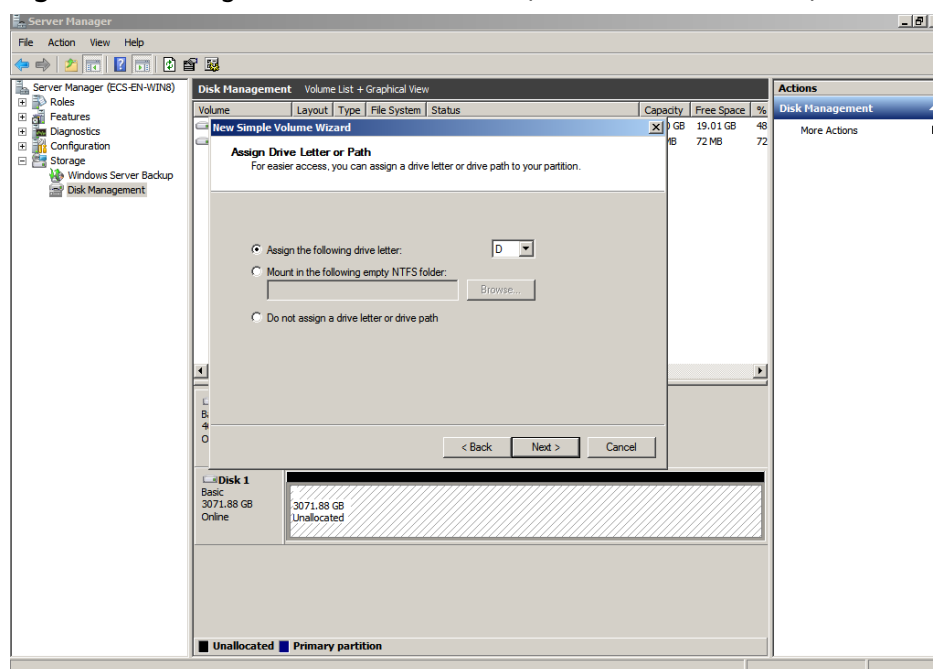
Step 7 Follow the prompts and click **Next**.

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

Figure 2-30 Specify Volume Size (Windows Server 2008)

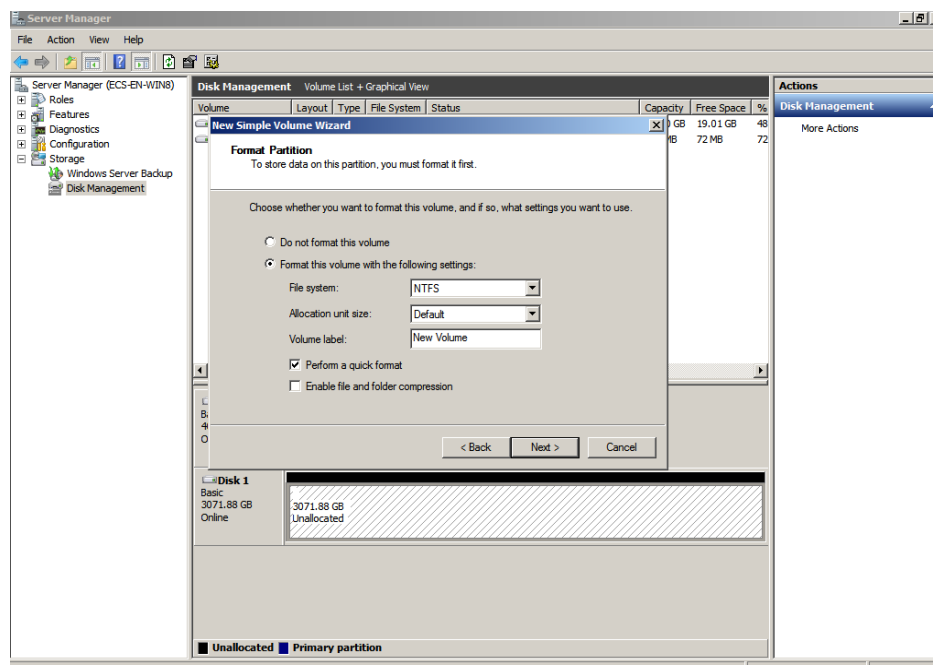
Step 8 Specify the volume size and click **Next**. The system selects the maximum volume size by default. You can specify the volume size as required. In this example, the default setting is used.

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

Figure 2-31 Assign Drive 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.

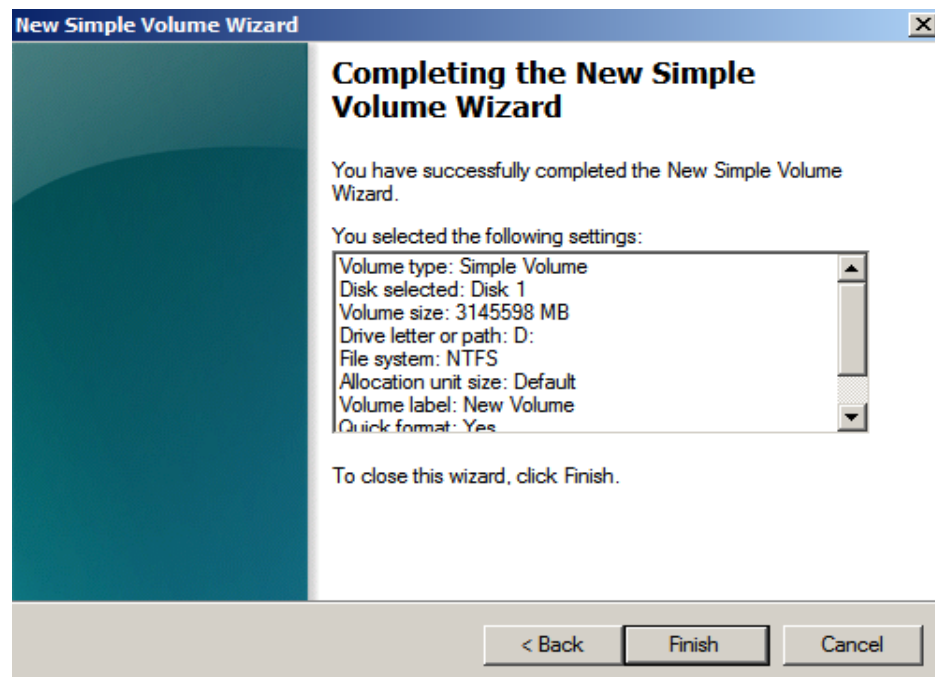
The **Format Partition** page is displayed.

Figure 2-32 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 2-33 Completing the New Simple Volume Wizard



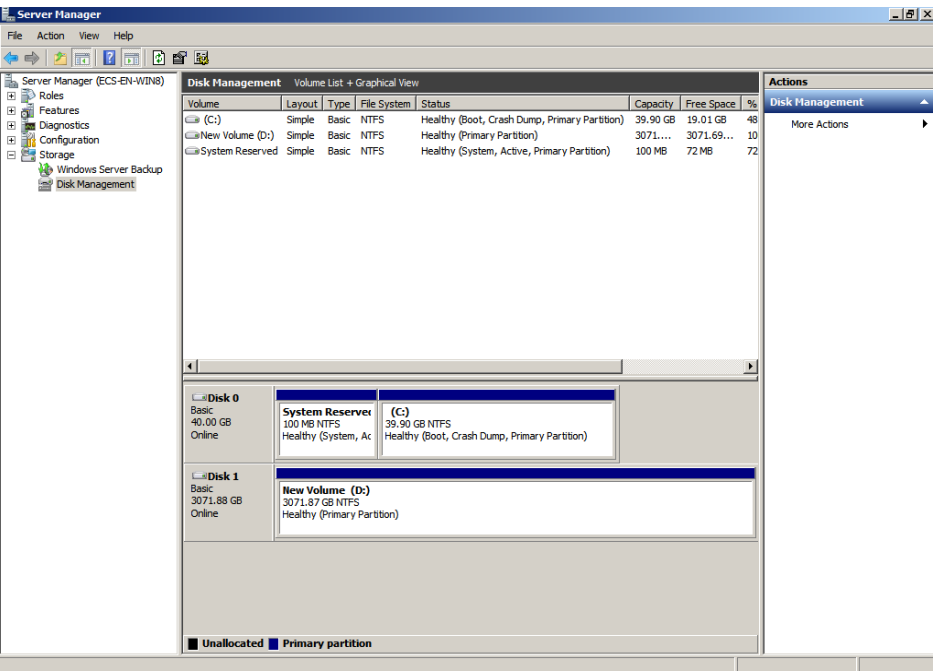
NOTICE

The partition sizes supported by file systems vary. Choose an appropriate file system format 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.

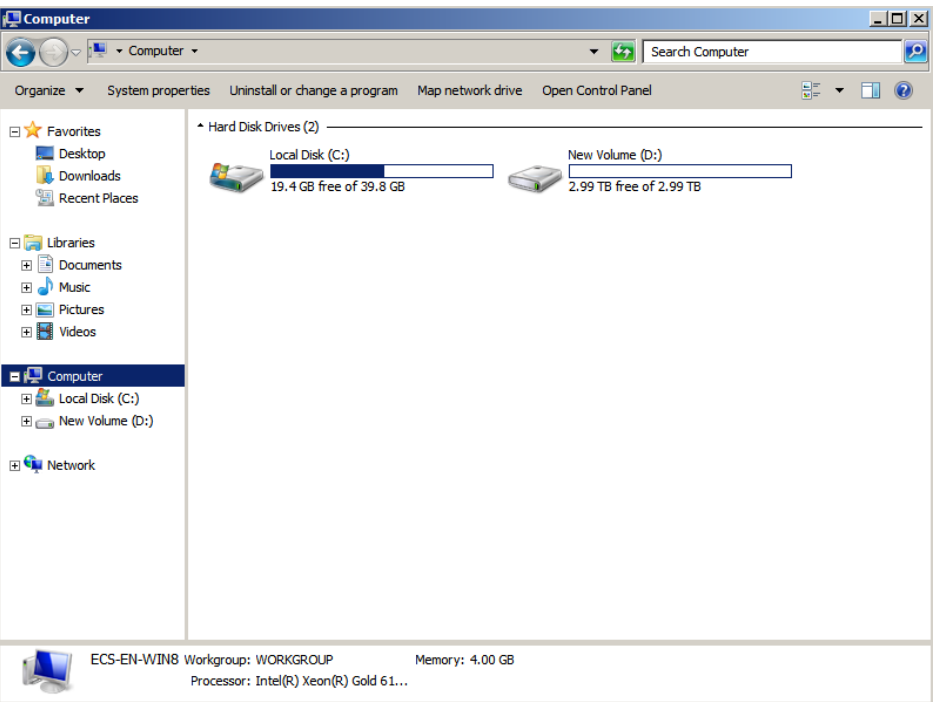
Figure 2-34 Disk initialization succeeded (Windows Server 2008)



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 2-35 Computer (Windows Server 2008)



----End

2.4.7 Initializing a Windows Data Disk Larger Than 2 TiB (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 TiB. In the following operations, the capacity of the example disk is 3 TiB.

The maximum disk capacity supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Therefore, use the GPT partition style if your disk capacity is larger than 2 TiB. For details, see [Initializing a Windows Data Disk Larger Than 2 TiB \(Windows Server 2008\)](#). To learn more about disk partition styles, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

Prerequisites

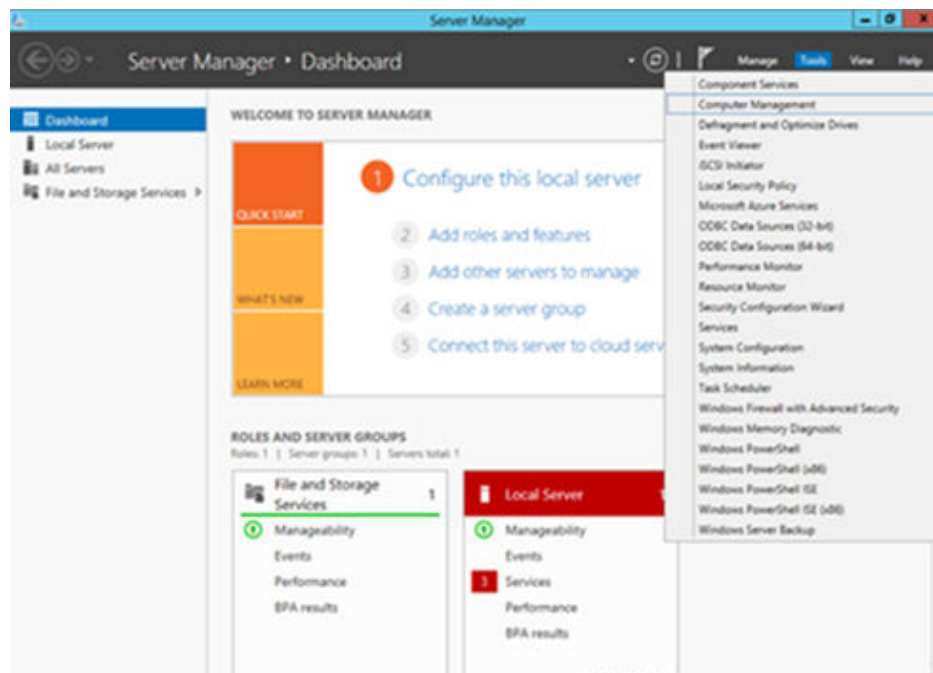
- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
 - For how to log in to an ECS, see the *Elastic Cloud Server User Guide*.

Procedure

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

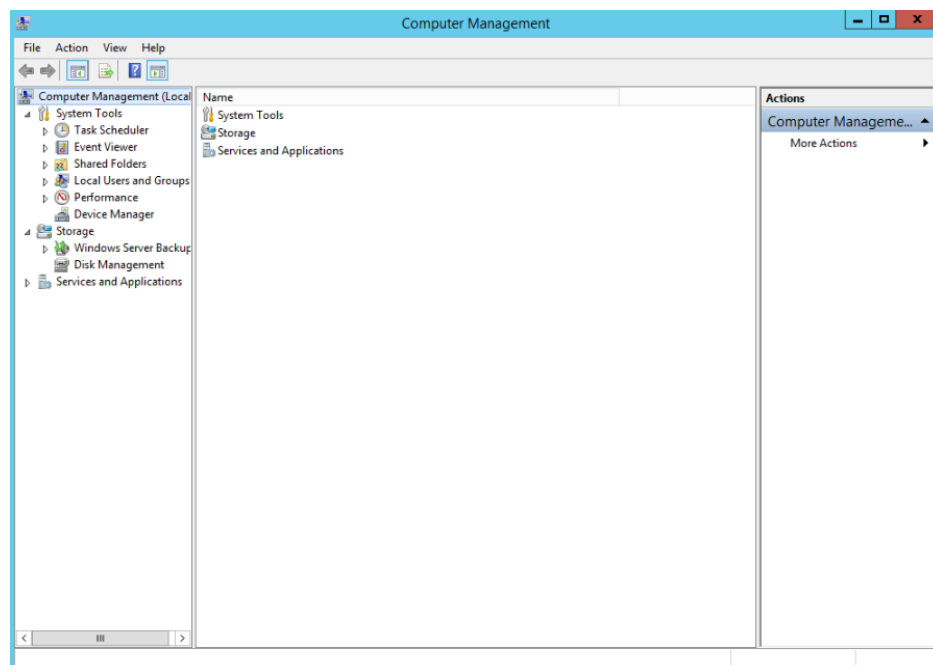
The **Server Manager** window is displayed.

Figure 2-36 Server Manager (Windows Server 2012)

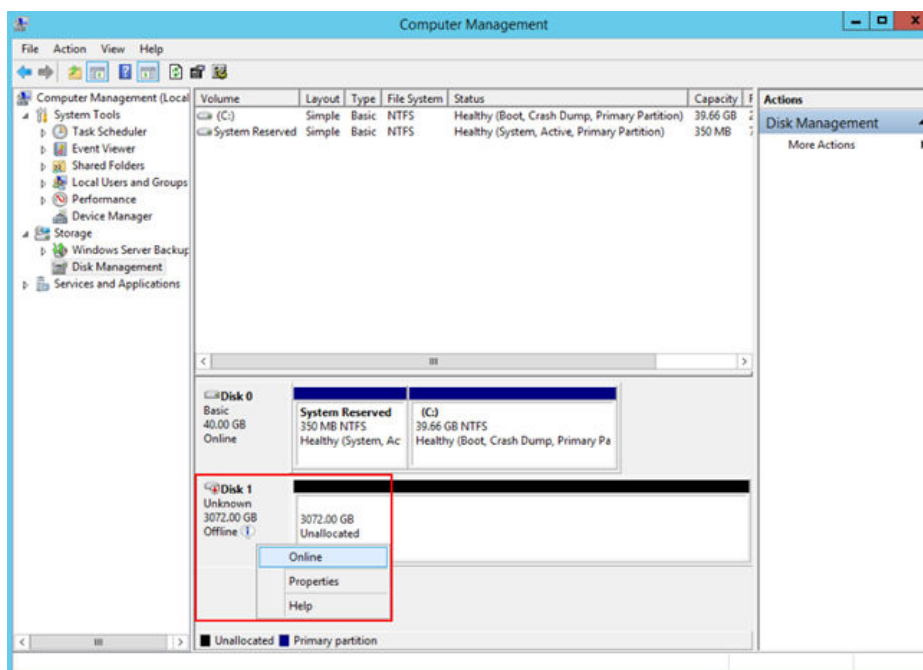


Step 2 In the upper right corner, choose **Tools > Computer Management**.
The **Computer Management** window is displayed.

Figure 2-37 Computer Management window (Windows Server 2012)



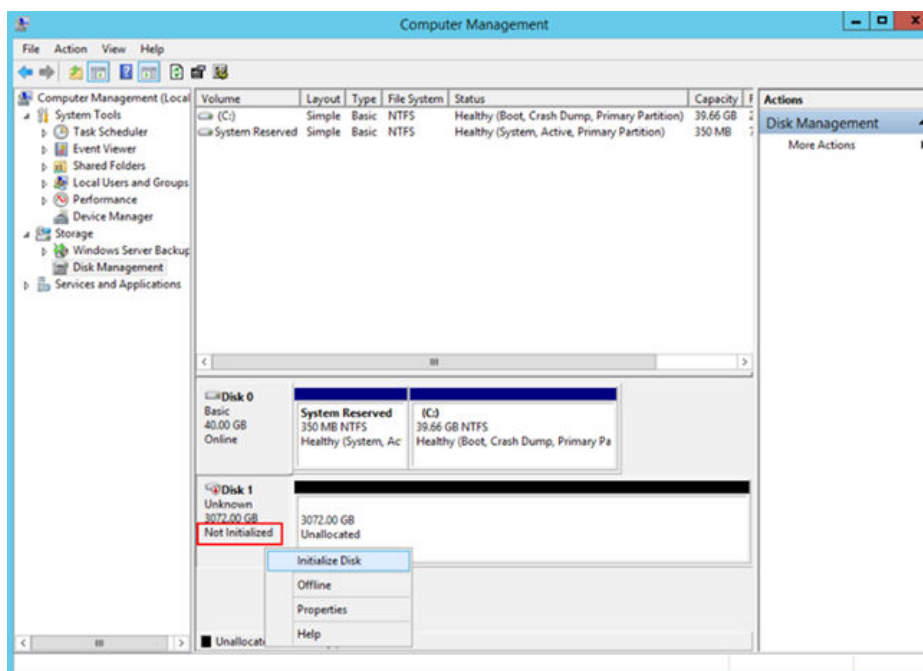
Step 3 Choose **Storage > Disk Management**.
Disks are displayed in the right pane.

Figure 2-38 Disk Management list (Windows Server 2012)

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

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

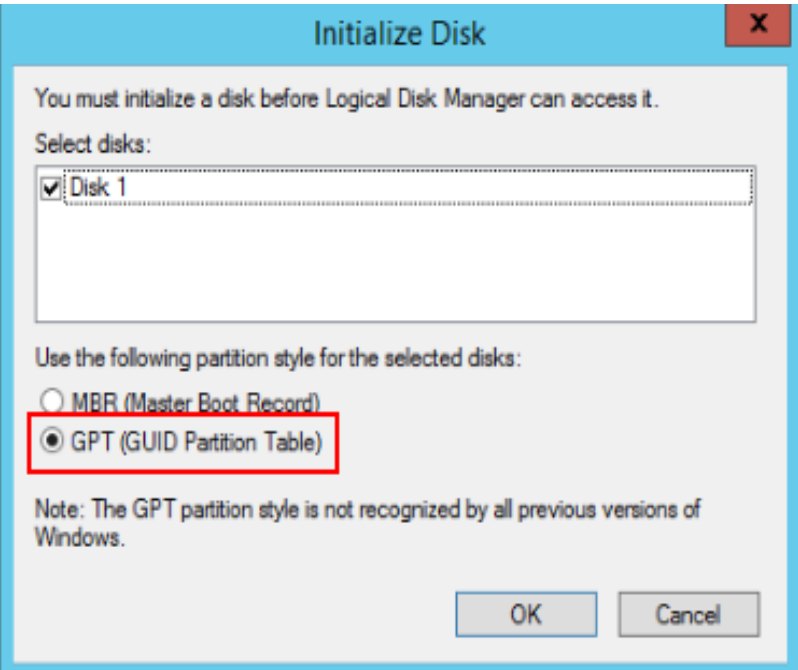
When the status of Disk 1 changes from **Offline** to **Not Initialized**, the disk has been brought online.

Figure 2-39 Bring online succeeded (Windows Server 2012)

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

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

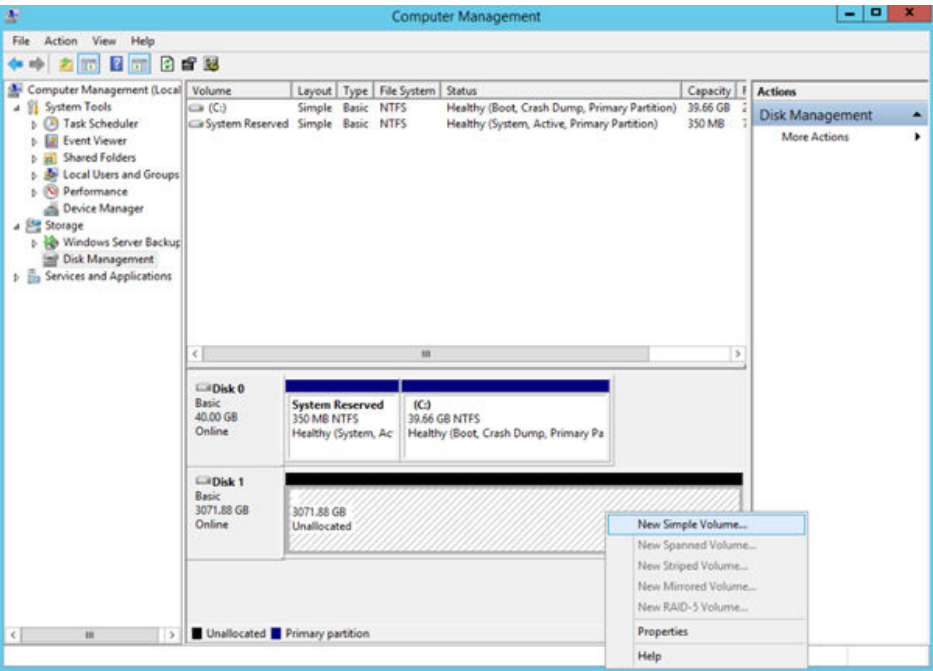
Figure 2-40 Initialize Disk (Windows Server 2012)



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 TiB. Therefore, select **GPT (GUID Partition Table)** and click **OK**.

The **Computer Management** window is displayed.

Figure 2-41 Computer Management (Windows Server 2012)



NOTICE

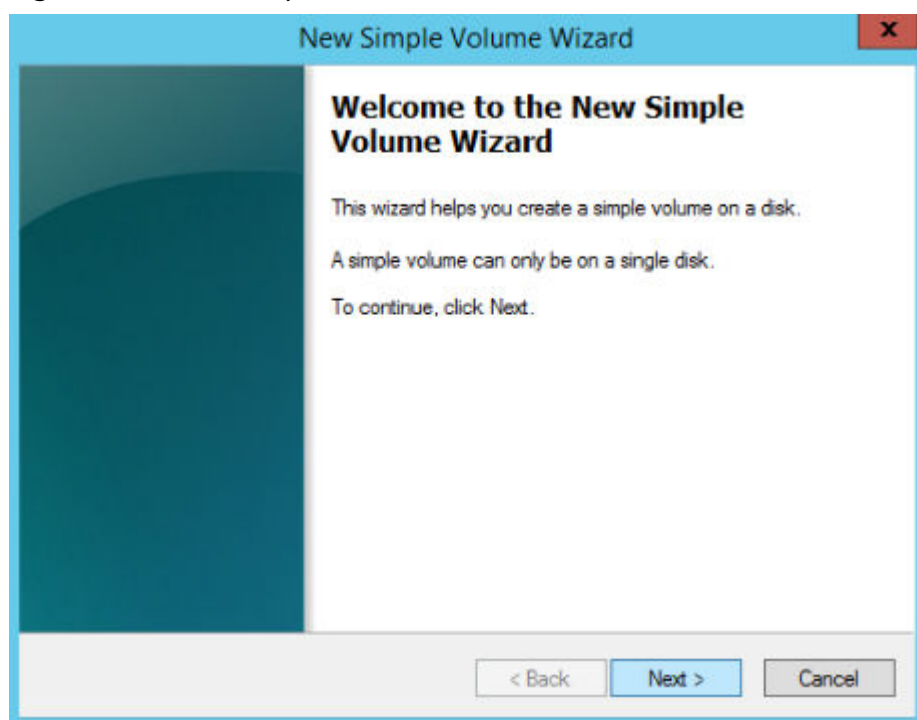
The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

- 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.

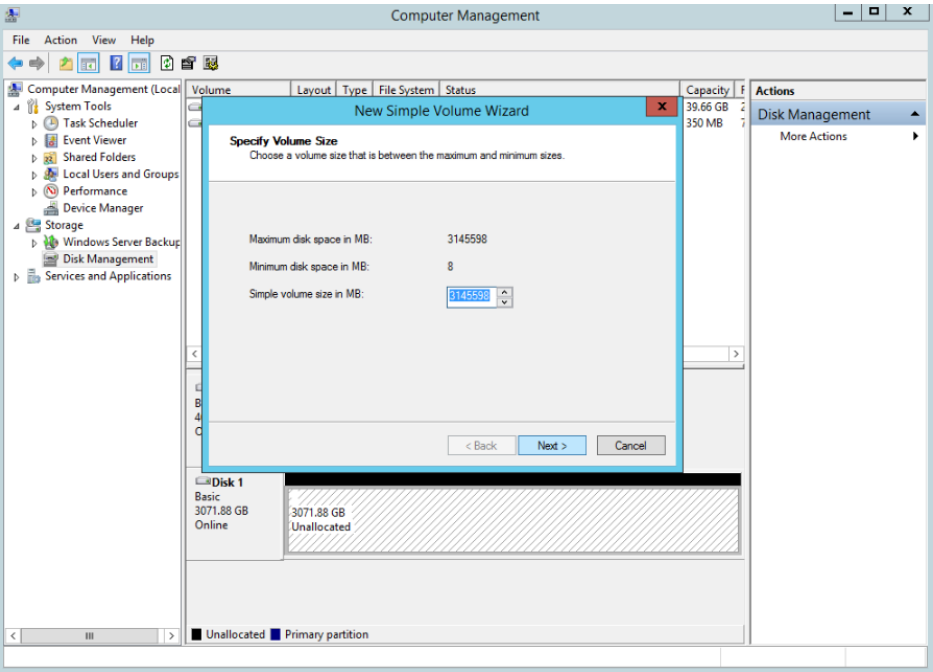
Figure 2-42 New Simple Volume Wizard (Windows Server 2012)



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

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

Figure 2-43 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.

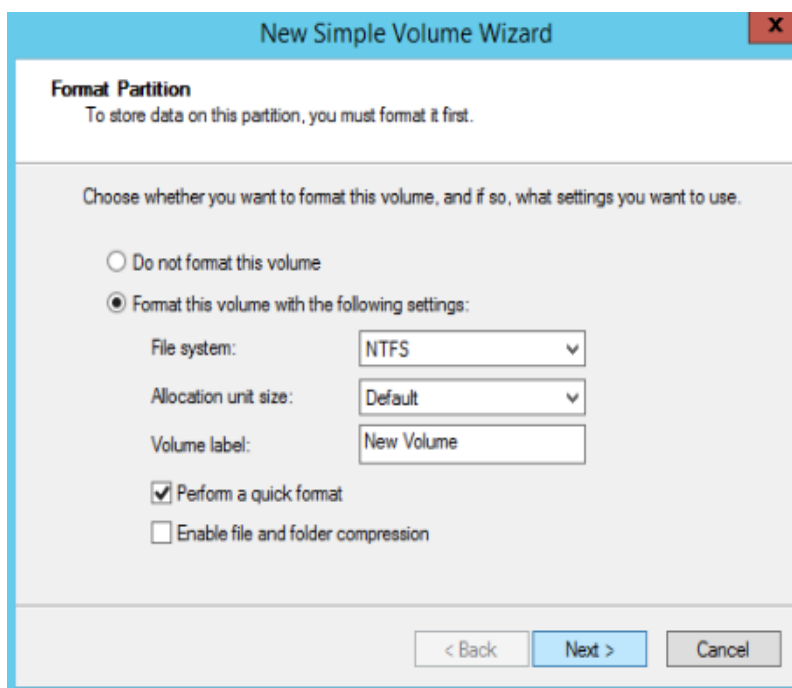
Figure 2-44 Assign Drive Letter or Path (Windows Server 2012)



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.

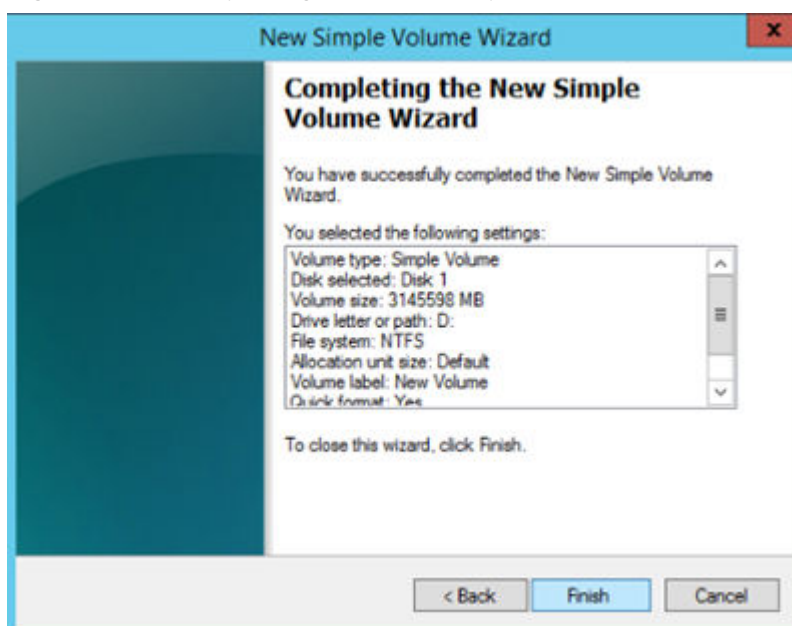
Figure 2-45 Format Partition (Windows Server 2012)



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 2-46 Completing the New Simple Volume Wizard (Windows Server 2012)



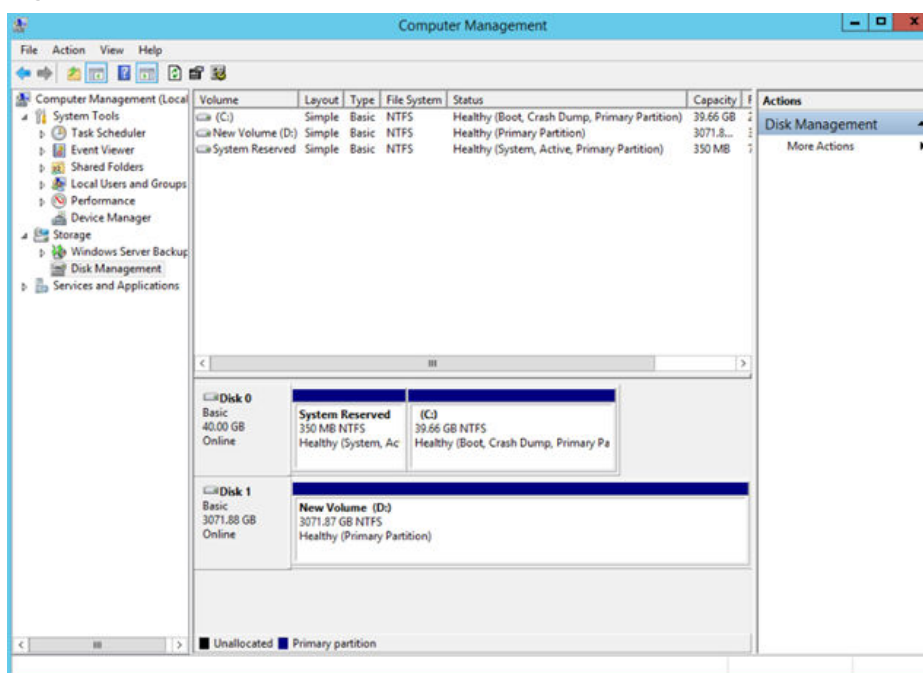
NOTICE

The partition sizes supported by file systems vary. Choose an appropriate file system format based on your service requirements.

Step 12 Click **Finish**.

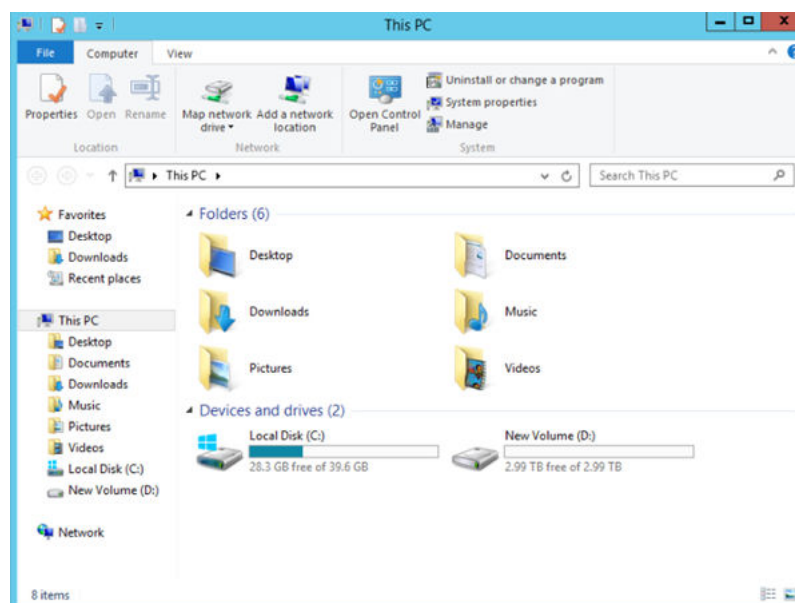
Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization has finished successfully.

Figure 2-47 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 2-48 This PC (Windows Server 2012)

----End

2.4.8 Initializing a Linux Data Disk Larger Than 2 TiB (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 TiB. In the following operations, the capacity of the example disk is 3 TiB.

The maximum partition size that MBR supports is 2 TiB and that GPT supports is 18 EiB. If the disk size you need to partition is greater than 2 TiB, partition the disk using GPT.

The fdisk partitioning tool is suitable only for MBR partitions, and the parted partitioning tool is suitable for both MBR and GPT partitions. For more information, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

The method for initializing a disk varies slightly 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.

NOTICE

When using a disk for the first time, if you have not initialized it, including creating partitions and file systems, the additional space added to this disk in an expansion later may not be normally used.

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*.

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 will be used. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured to mount automatically at startup.

Step 1 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, this server contains two disks. **/dev/vda** and **/dev/vdb**. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk.

Step 2 Launch 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: 3299GiB
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 Set the disk partition style.

mklabel *Disk partition style*

The disk partition style can be MBR or GPT. If the disk capacity is greater than 2 TiB, use GPT.

mklabel gpt

NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

If the partition style is changed after the disk has been used, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT after a disk has been used, it is recommended that you back up the disk data before the change.

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: 3299GiB
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 Create a new partition.

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, starting on **2048** and using **100%** of the rest of the disk. 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 print the partition details.

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 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 Format the new partition with a desired file system format.

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

In this example, the **ext4** format is used 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. Choose an appropriate file system format based on your service requirements.

Step 12 Create a mount point.

mkdir *Mount point*

In this example, the **/mnt/sdc** mount point is created.

mkdir /mnt/sdc

 NOTE

The **/mnt** directory exists on all Linux systems. If the mount point cannot be created, it may be that the **/mnt** directory has been accidentally deleted. You can run **mkdir -p /mnt/sdc** to create the mount point.

Step 13 Mount the new partition on the created mount point.

mount *Disk partition Mount point*

In this example, the **/dev/vdb1** partition is mounted on **/mnt/sdc**.

mount /dev/vdb1 /mnt/sdc

Step 14 Check 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
```

You should now see that partition **/dev/vdb1** is mounted on **/mnt/sdc**.

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at startup. You can configure the **fstab** file of a that has data. This operation will not affect the existing data.

The following example uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names (like **/dev/vdb1**) to identify disks in the file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a stop or start. This can even prevent your from booting up.

 NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi /etc/fstab

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID:** The UUID obtained in [Step 1](#).
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.
 - **0:** not use fsck.
 - If the mount point is the root partition (**/**), this parameter must be set to **1**.

If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

3 Disk Capacity Expansion

3.1 Expansion Overview

What Is Capacity Expansion?

If the capacity of an existing disk is insufficient, you can expand the disk capacity to increase the storage space.

Both system disks and data disks can be expanded. Disk capacity can only be expanded, not reduced.

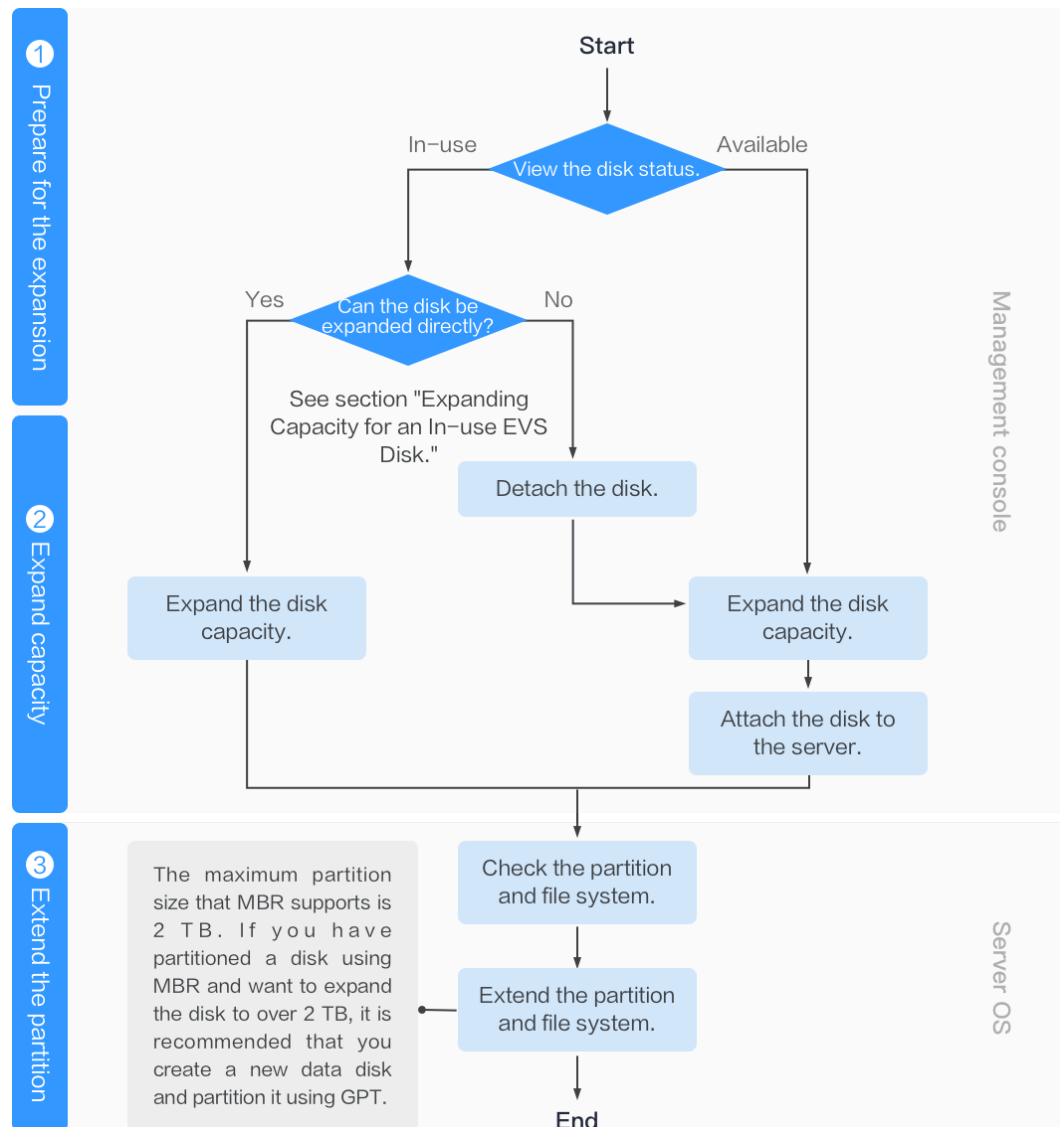
 NOTE

If you detach a system disk and then attach it to an ECS as a data disk, the maximum capacity of this disk is still 1 TiB.

How to Expand the Disk Capacity?

A capacity expansion operation includes the following steps:

1. [Expand the disk capacity on the management console.](#)
2. [Log in to the server and extend the disk partition and file system.](#)

Figure 3-1 Capacity expansion procedure

Expand the Disk Capacity on the Management Console

Choose a proper expansion method based on the disk status.

- For an In-use disk:
 - The disk has been attached to a server. Check whether the disk can be expanded in the In-use state by referring to [Constraints](#).
 - If yes, expand the disk capacity according to [Expanding Capacity for an In-use EVS Disk](#).
 - If no, detach the disk. Then, expand the disk capacity according to [Expanding Capacity for an Available EVS Disk](#).
- For an Available disk:
 - The disk has not been attached to any server and can be directly expanded by referring to [Expanding Capacity for an Available EVS Disk](#).
 - A shared disk can be expanded only when its status is **Available**.

Log In to the Server and Extend the Disk Partition and File System

After the disk has been expanded on the management console, only the disk storage capacity is enlarged, but its additional space cannot be used directly. You must log in to the server and extend the disk partition and file system. For details, see [Table 3-1](#).

Table 3-1 Extending the disk partition and file system

Capacity After Expansion	Extend Disk Partition and File System
Disk capacity $\leq$ 2 TiB	- Windows: Extending Disk Partitions and File Systems (Windows Server 2008) - Linux: Partition and File System Extension Preparations (Linux)
Disk capacity > 2 TiB	- GPT partition style: Extending Disk Partitions and File Systems (Windows Server 2008) or Partition and File System Extension Preparations (Linux) - MBR partition style: Not supported The maximum disk capacity that MBR supports is 2 TiB, and the disk space exceeding 2 TiB cannot be used. If your disk uses MBR and you need to expand the disk capacity to over 2 TiB, 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.

Related Operations

For more information, see [FAQs > Capacity Expansion](#).

3.2 Expanding Capacity for an In-use EVS Disk

Scenarios

This section describes how to expand the capacity of an In-use EVS disk on the management console. The In-use status indicates that the disk has been attached to a server. You do not need to detach the disk when expanding an In-use disk.

Constraints

- Disk capacity can be expanded, but cannot be reduced.
- When expanding an In-use disk, the server attached with this disk must be in the **Running** or **Stopped** state.
- A shared disk in the **In-use** state cannot be expanded. To expand such a disk, you must detach it from all its servers, wait until its status changes to **Available**, and then expand its capacity. For more information, see [Expanding Capacity for an Available EVS Disk](#).
- Only some server OSs support capacity expansion of In-use disks. For servers without such support, detach the disk and then expand its capacity. Otherwise, you may need to stop and then start the server after the expansion to make the additional space available.

Perform the following operations to check whether your server OS supports capacity expansion of In-use disks:

- a. Check your server image. Both public images and private images listed on the console support the capacity expansion of In-use disks.

How to view: Log in to the management console. In the navigation pane

on the left, click  and choose **Compute > Image Management Service**. On the **Public Images** tab, view the images of the **ECS image** type.

- b. If your server OS does not appear in the image list, check whether it is included in [Table 3-2](#).

If yes, you can expand the disk. Otherwise, you must detach the disk and then expand its capacity. For details, see [Expanding Capacity for an Available EVS Disk](#).

Table 3-2 OSs that support the capacity expansion of In-use disks

OS	Version
CentOS 8	8.0 64-bit or later
CentOS 7	7.2 64-bit or later
CentOS 6	6.5 64-bit or later
Debian	8.5.0 64-bit or later
Fedora	24 64-bit or later
SUSE 12	SUSE Linux Enterprise Server 12 64-bit or later
SUSE 11	SUSE Linux Enterprise Server 11 SP4 64-bit
OpenSUSE	42.1 64-bit or later
Oracle Linux Server release 7	7.2 64-bit or later
Oracle Linux Server release 6	6.7 64-bit or later

OS	Version
Ubuntu Server	14.04 64-bit or later
Red Hat Enterprise Linux 7	7.3 64-bit
Red Hat Enterprise Linux 6	6.8 64-bit
EulerOS	2.2 64-bit or later
Windows Server 2016	Windows Server 2016 R2 Enterprise 64-bit
Windows Server 2012	Windows Server 2012 R2 Standard 64-bit
Windows Server 2008	Windows Server 2008 R2 Enterprise 64-bit

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Choose a way to expand the disk by determining whether you want to check server information first.

- If yes, 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 **Servers** 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 no, perform the following procedure:
 - a. In the disk list, locate the row that contains the target disk and click **Expand Capacity** in the **Operation** column.
The expansion page is displayed.

Step 4 Set the **New Capacity** parameter and click **Next**.

Step 5 On the **Details** page, check the disk details.

- Click **Submit** to start the expansion.
- Click **Previous** to change the settings.

After the configuration is submitted, go back to the disk list page.

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

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

- Step 7** Log in to the server and extend the partition and file system after the disk has been expanded on the console, because previous steps only enlarge the disk space.

The operations vary depending on the server OS.

- In Windows, see [Extending Disk Partitions and File Systems \(Windows Server 2008\)](#).
- In Linux, see [Partition and File System Extension Preparations \(Linux\)](#).

----End

3.3 Expanding Capacity for an Available EVS Disk

Scenarios

This section describes how to expand the capacity of an Available EVS disk on the management console. The Available status indicates that the disk has not been attached to any server.

Constraints

- Disk capacity can be expanded, but cannot be reduced.
- A shared disk in the **In-use** state cannot be expanded. To expand such a disk, you must detach it from all its servers, wait until its status changes to **Available**, and then expand its capacity.

Procedure

- Step 1** Log in to the management console.

- Step 2** Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

- Step 3** In the disk list, locate the row that contains the target disk and click **Expand Capacity** in the **Operation** column.

The expansion page is displayed.

- Step 4** Set the **New Capacity** parameter and click **Next**.

- Step 5** On the **Details** page, check the disk details.

- Click **Submit** to start the expansion.
- Click **Previous** to change the settings.

After the configuration is submitted, go back to the disk list page.

- Step 6** In the disk list, view the capacity of the target disk.

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

- Step 7** Attach the disk to the server. For details, see [Attach an EVS Disk](#).

- Step 8** Log in to the server and extend the partition and file system after the disk has been expanded on the console, because previous steps only enlarge the disk space.

The operations vary depending on the server OS.

- In Windows, see [Extending Disk Partitions and File Systems \(Windows Server 2008\)](#).
- In Linux, see [Partition and File System Extension Preparations \(Linux\)](#).

----End

3.4 Extending Disk Partitions and File Systems (Windows Server 2008)

Scenarios

After a disk is expanded on the management console, the disk size is enlarged, but the additional space cannot be used directly.

In Windows, you must allocate the additional space to an existing partition or a new partition.

This section uses Windows Server 2008 R2 Enterprise 64bit as the sample OS to describe the expansion methods:

- For a system disk:
 - If volume (C:) already exists, you can add the additional space to volume (C:) and use it as a system volume. For details, see [System Disk: Add the Additional Space to the Original Volume](#).
 - If volume (C:) already exists, you can create a new volume such as volume (F:) with the additional space and use the new volume as a data volume. For details, see [System Disk: Create a New Volume with the Additional Space](#).
- For a data disk:
 - If volume (D:) already exists, you can add the additional space to volume (D:) and use it as a data volume. For details, see [Data Disk: Add the Additional Space to the Original Volume](#).
 - If volume (D:) already exists, you can create a new volume such as volume (E:) with the additional space and use the new volume as a data volume. For details, see [Data Disk: Create a New Volume with the Additional Space](#).

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

NOTICE

Performing the expansion operations with caution. Incorrect operations may lead to data loss or exceptions, so you are advised to back up the disk data using CBR before expansion. For details about using CBR, see [Managing EVS Backups](#).

Prerequisites

- You have expanded the disk capacity and attached the disk to a server on the management console. For details, see [Expanding Capacity for an In-use EVS Disk](#) or [Expanding Capacity for an Available EVS Disk](#).
- 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*.

System Disk: Add the Additional Space to the Original Volume

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

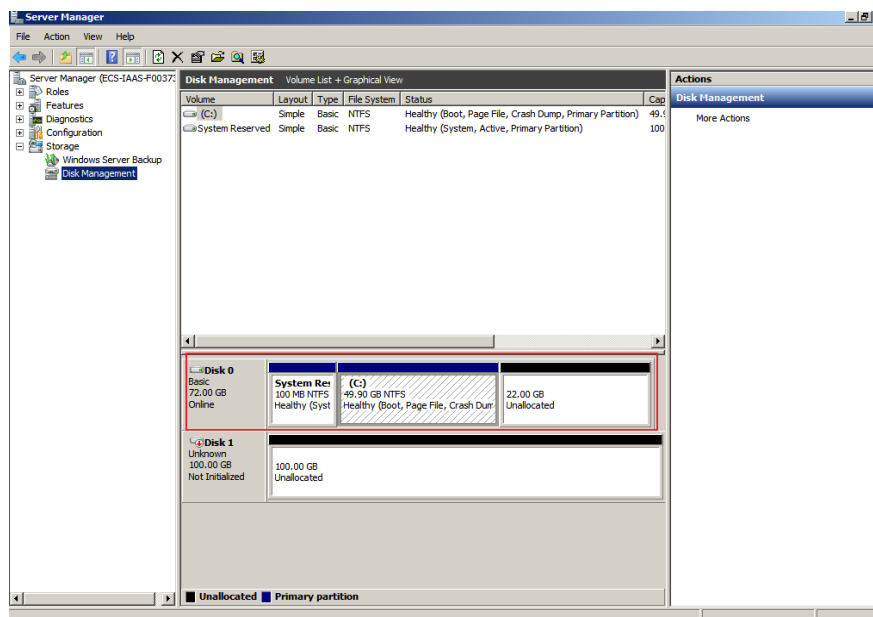
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

Figure 3-2 Disk Management (system disk)



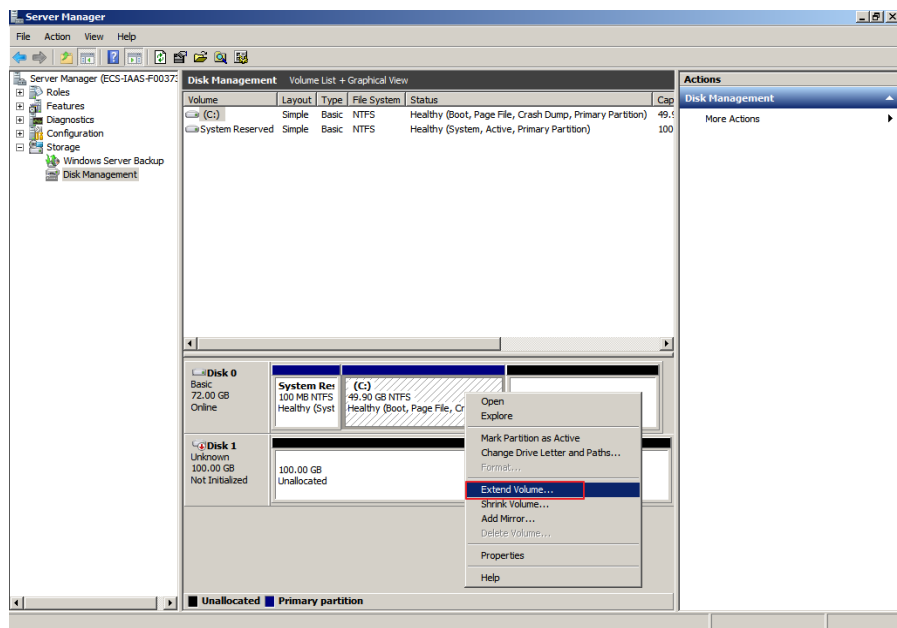
NOTE

If you cannot see 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 volume that you want to extend. The current volume size and unallocated space are displayed.

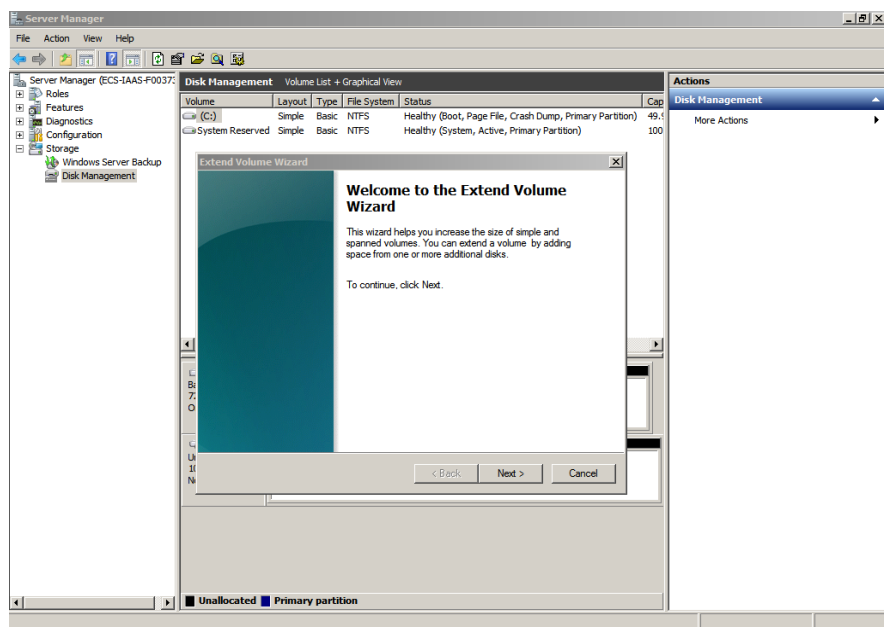
Step 4 Right-click the target volume and choose **Extend Volume**.

Figure 3-3 Choosing **Extend Volume**



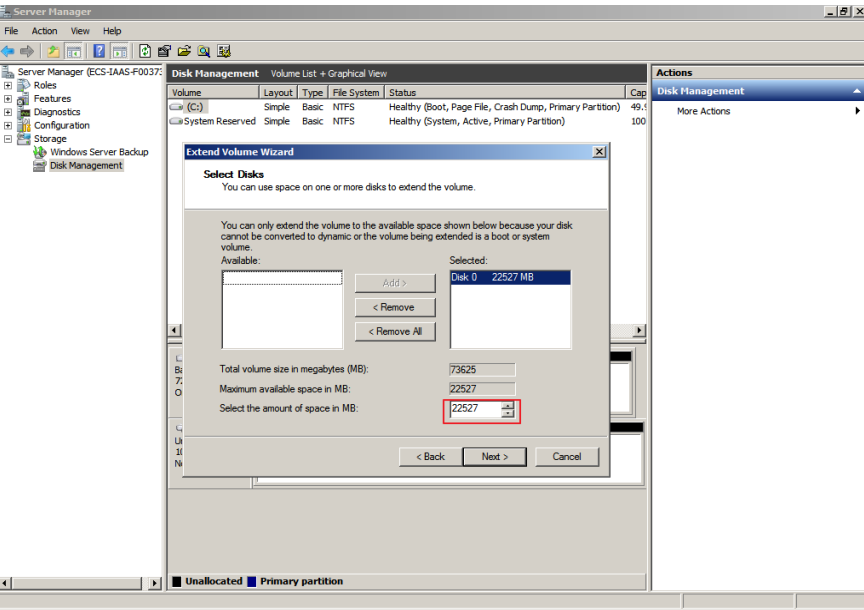
Step 5 On the displayed **Extend Volume Wizard** window, click **Next**.

Figure 3-4 Extend Volume Wizard



Step 6 In the text box to the right of **Select the amount of space in MB**, enter the amount of the additional space and click **Next**.

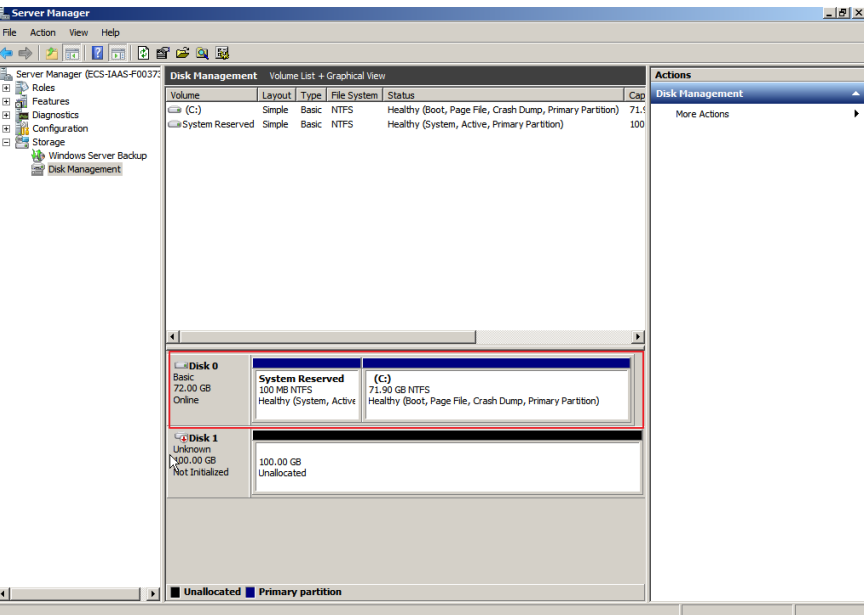
Figure 3-5 Selecting space



Step 7 Click **Finish**.

After the expansion succeeded, the partition size is larger than the original size.

Figure 3-6 Capacity expansion succeeded



----End

System Disk: Create a New Volume with the Additional Space

In this example, the system disk has 40 GiB originally, and 60 GiB is added on the management console. The following procedure describes how to use this 60 GiB to create a new volume, for example volume (F:), on the server. After the operation is complete, new volume (F:) has 60 GiB of capacity and can be used as a data volume.

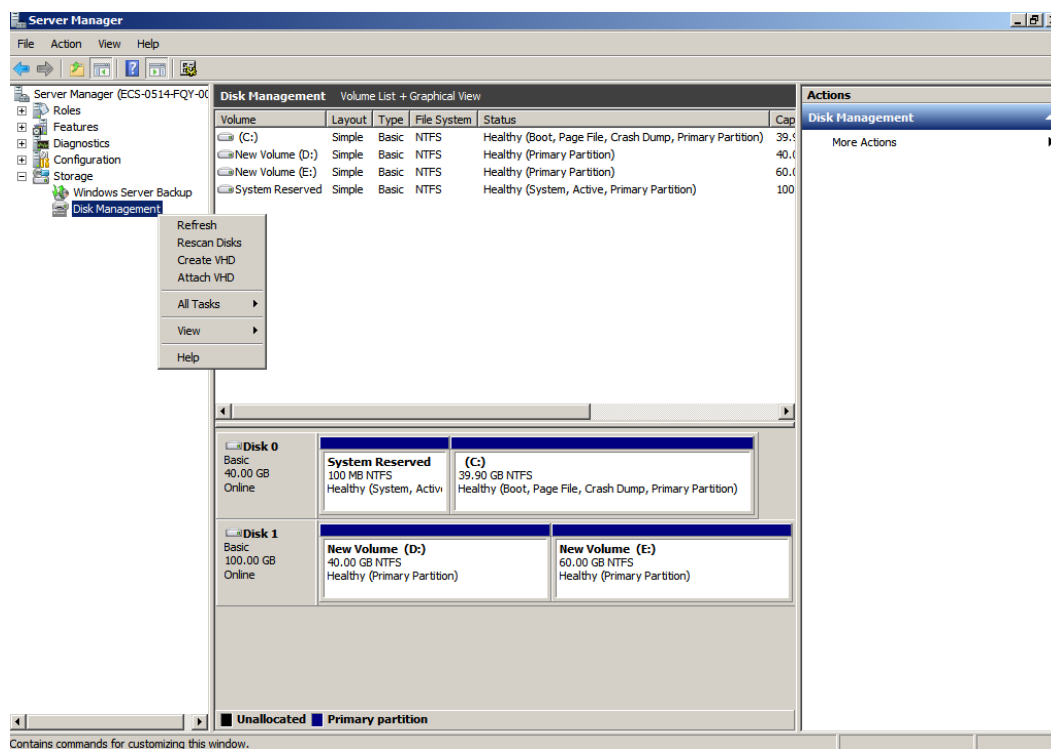
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

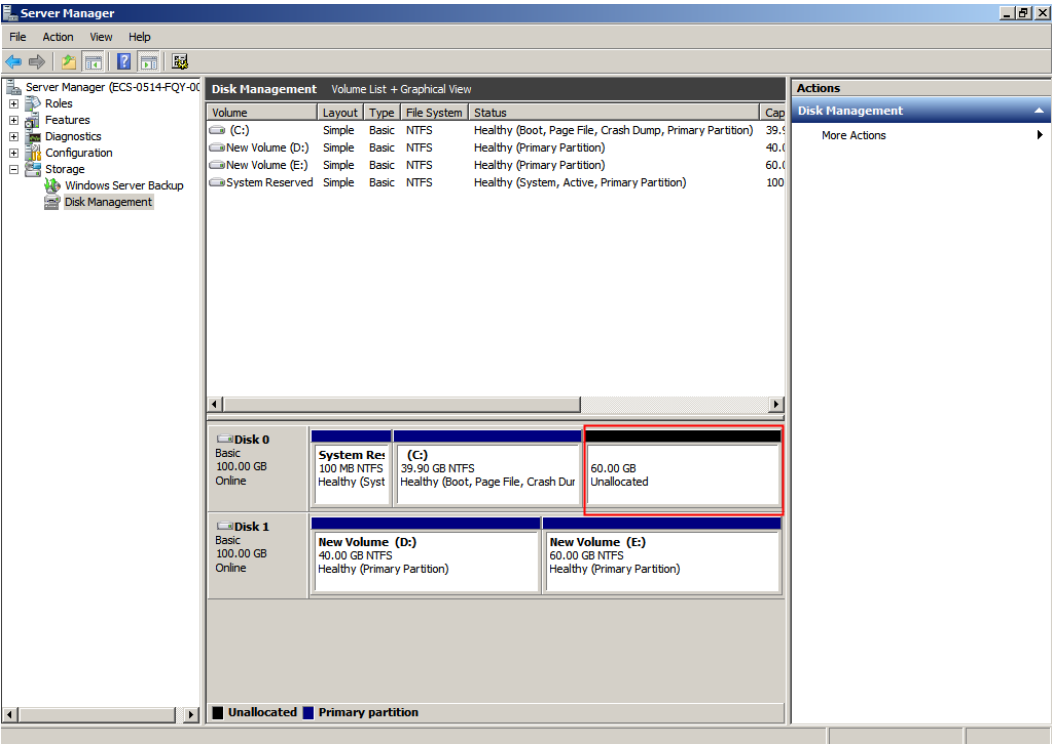
Figure 3-7 Refresh (system disk)



Step 3 If you cannot see the additional space, right-click **Disk Management** and choose **Refresh** from the shortcut menu.

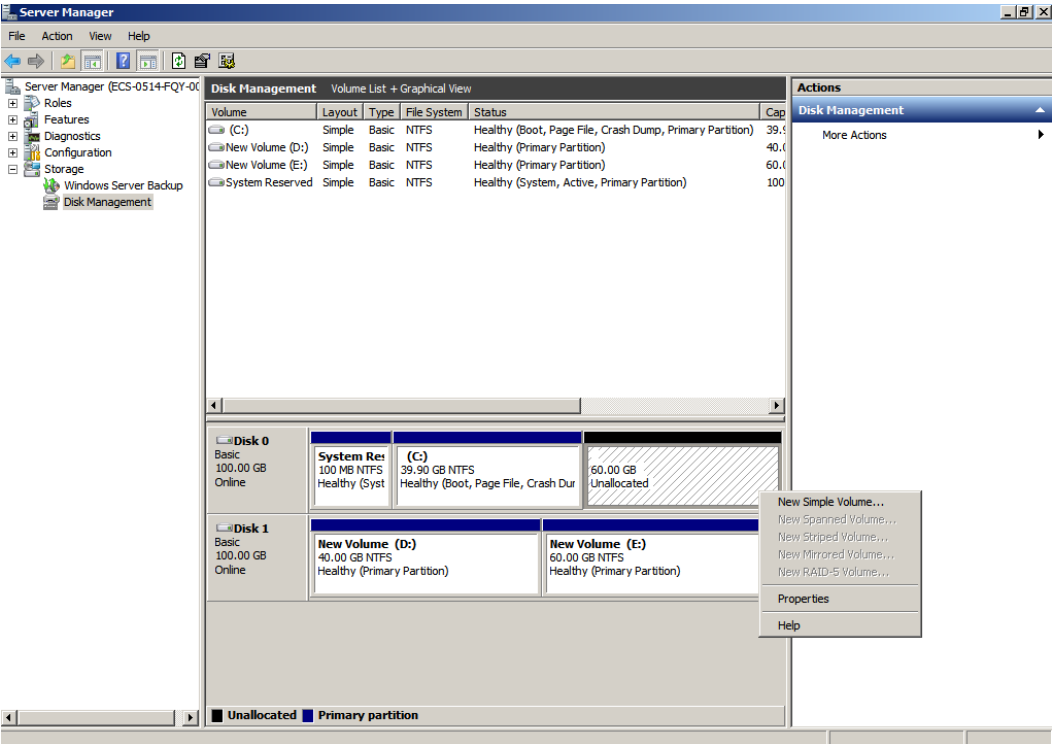
After the refresh, the additional space is displayed in the right area and is unallocated.

Figure 3-8 Unallocated disk space (system disk)



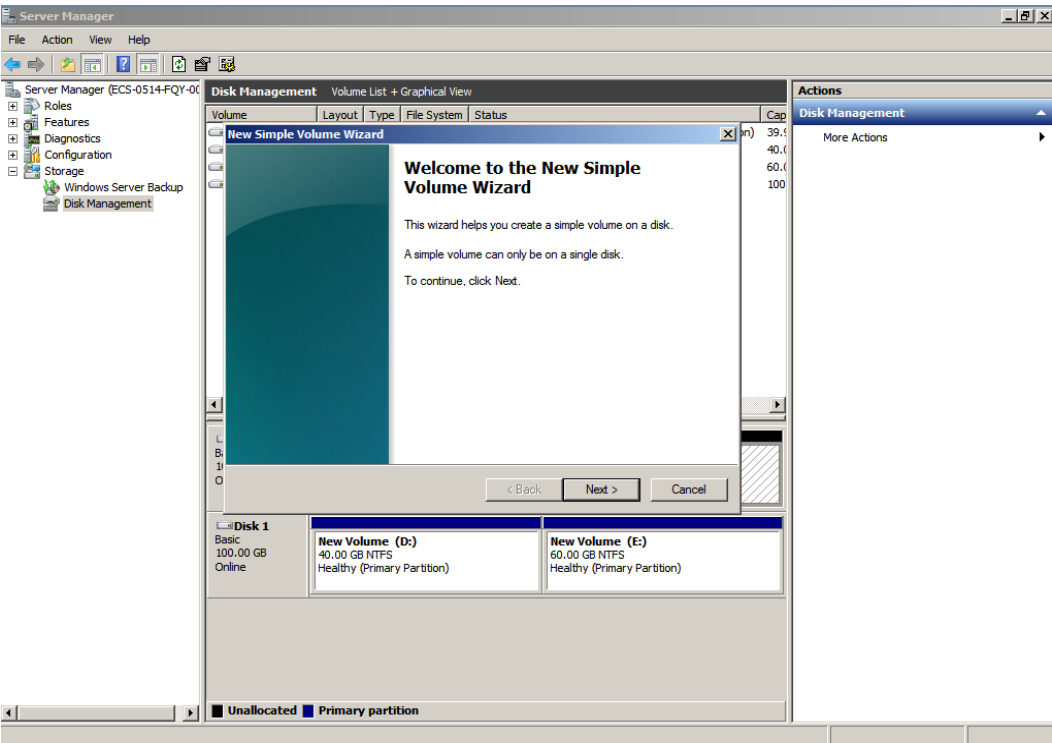
Step 4 In the **Unallocated** area of **Disk 0**, right-click the blank area and choose **New Simple Volume**.

Figure 3-9 New Simple Volume (system disk)



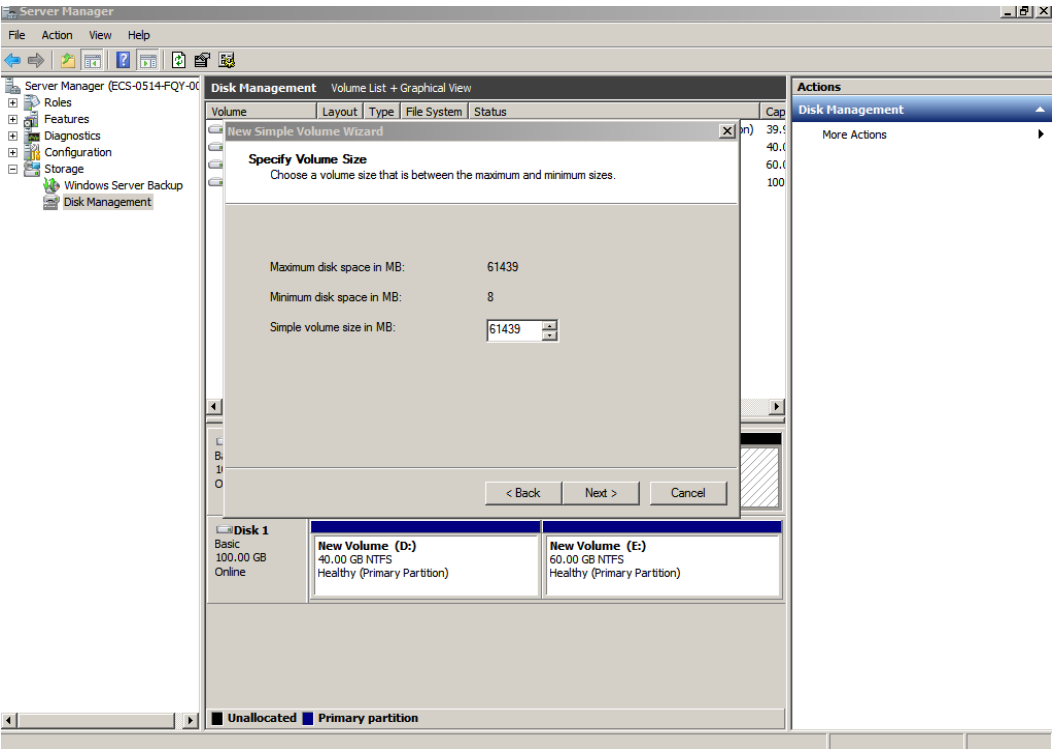
Step 5 On the displayed **New Simple Volume Wizard** window, click **Next**.

Figure 3-10 New Simple Volume Wizard (system disk)



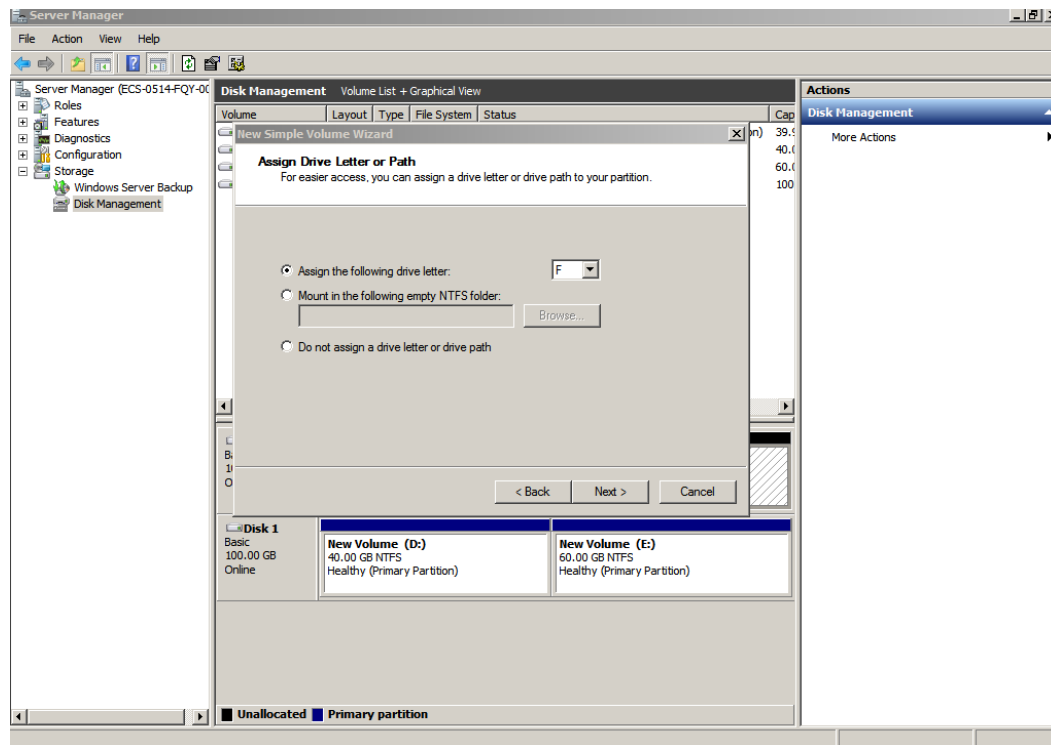
Step 6 On the displayed **Specify Volume Size** page, set **Simple volume size in MB** and click **Next**. In this example, the default size is used.

Figure 3-11 Specify Volume Size (system disk)



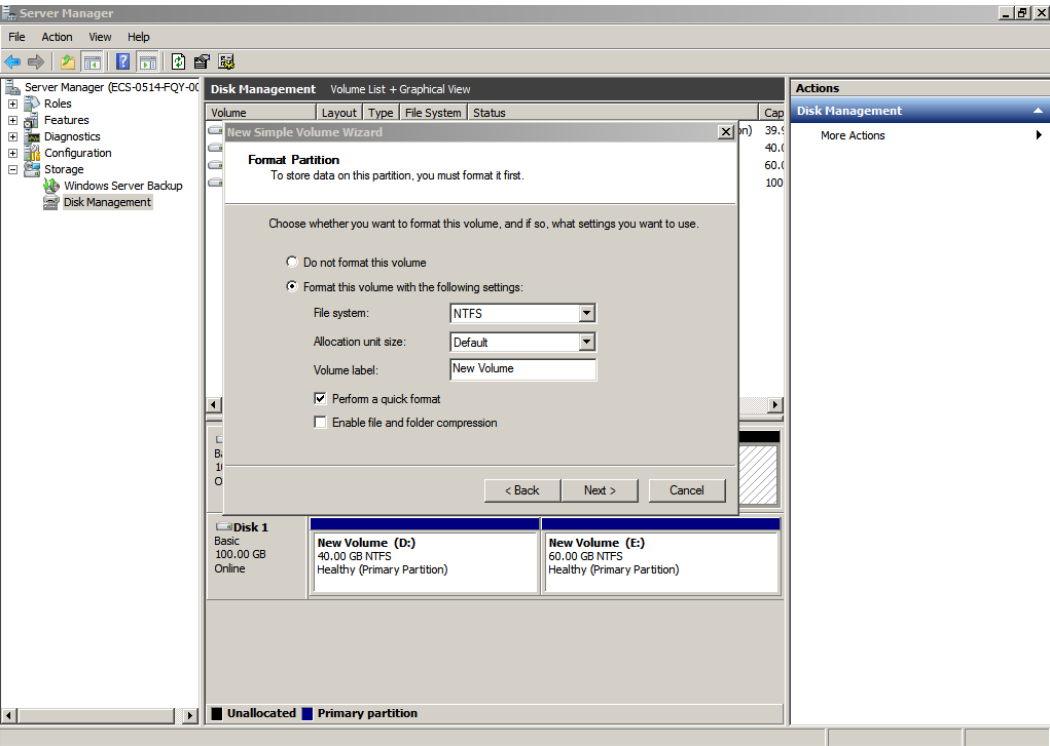
- Step 7** On the displayed **Assign Drive Letter and Path** page, click **Assign the following drive letter**, select a drive letter, and click **Next**. In this example, drive letter **F** is selected.

Figure 3-12 Assign Drive Letter or Path (system disk)



- Step 8** On the displayed **Format Partition** page, click **Format this volume with the following settings**, set parameters based on the requirements, and select **Perform a quick format**. Then, click **Next**.

Figure 3-13 Format Partition (system disk)



Step 9 Click **Finish**.

After the expansion succeeded, new volume (F:) is displayed.

Figure 3-14 Completing the New Simple Volume Wizard (new volume F:)

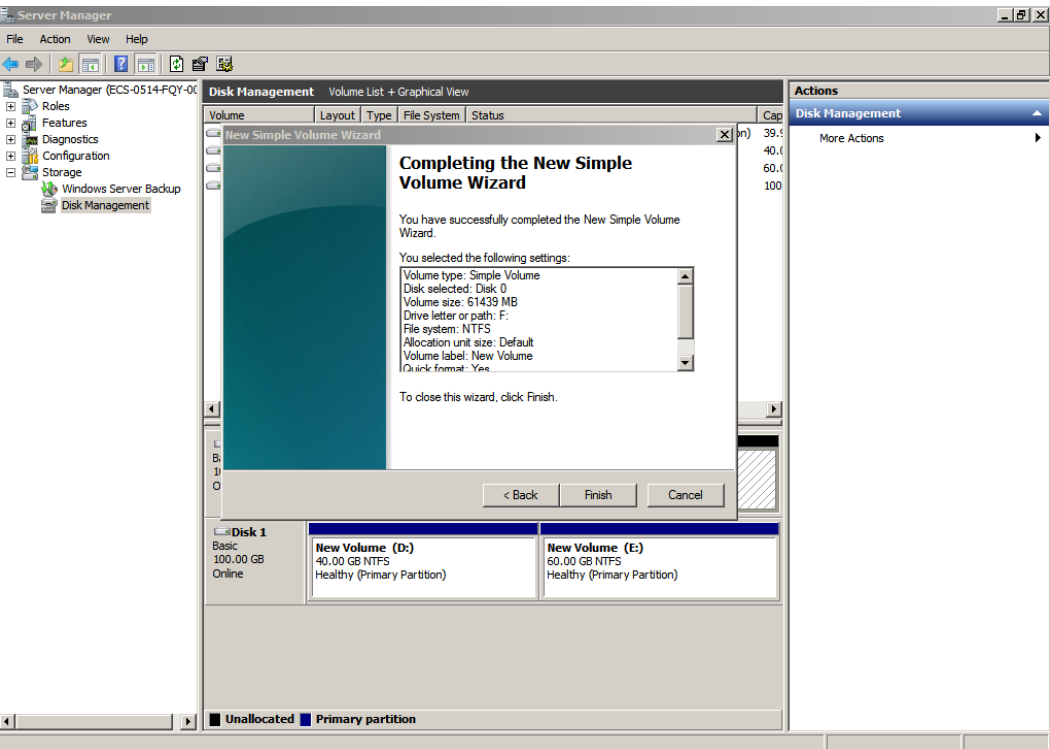
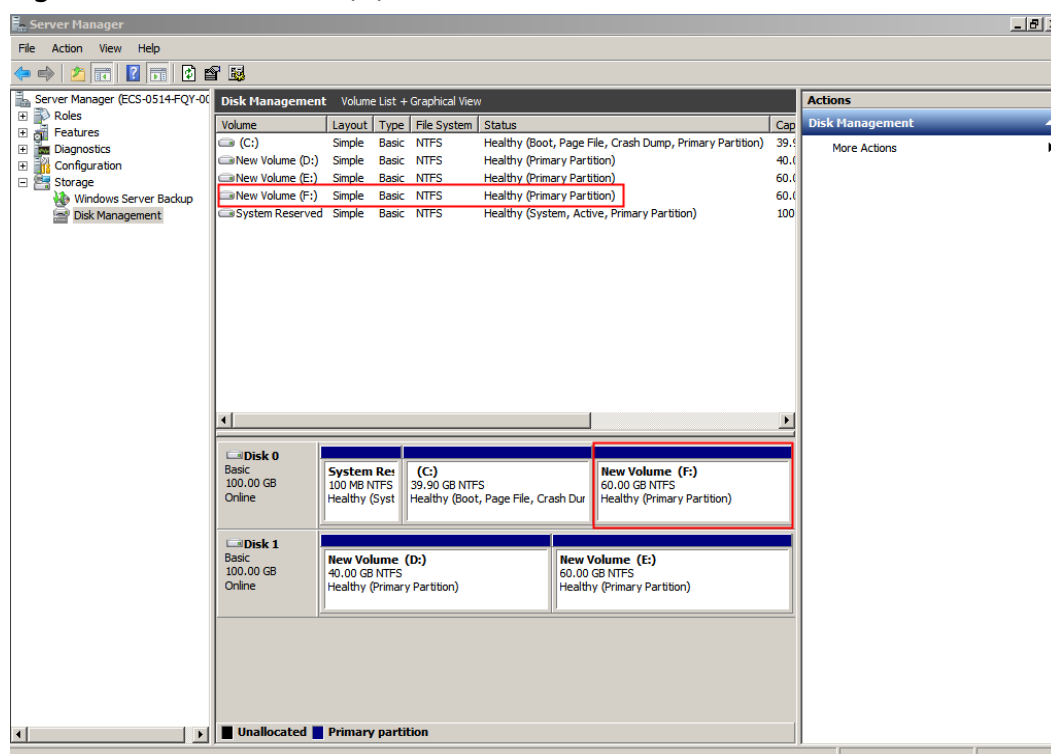


Figure 3-15 New Volume (F:)

-----End

Data Disk: Add the Additional Space to the Original Volume

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

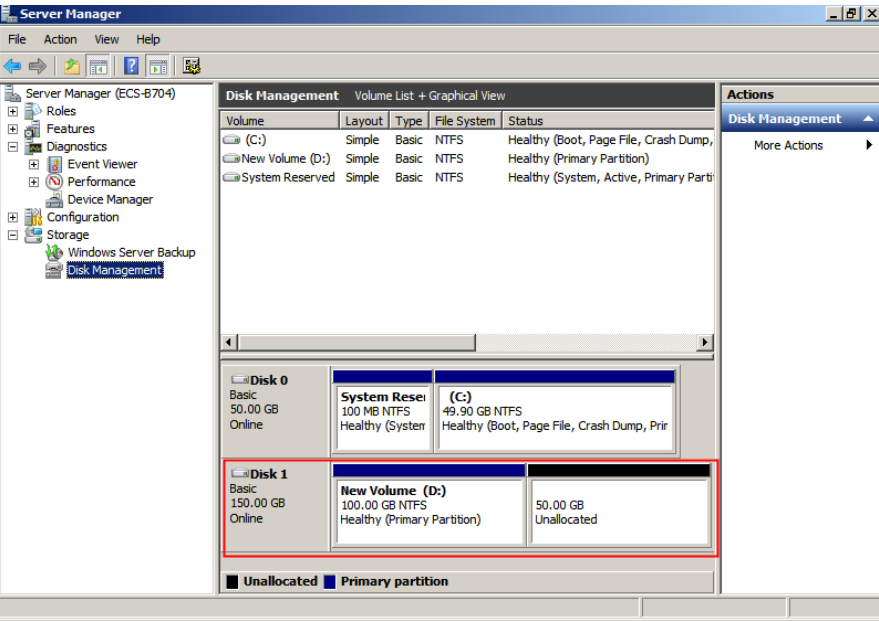
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

Figure 3-16 Disk Management (data disk)



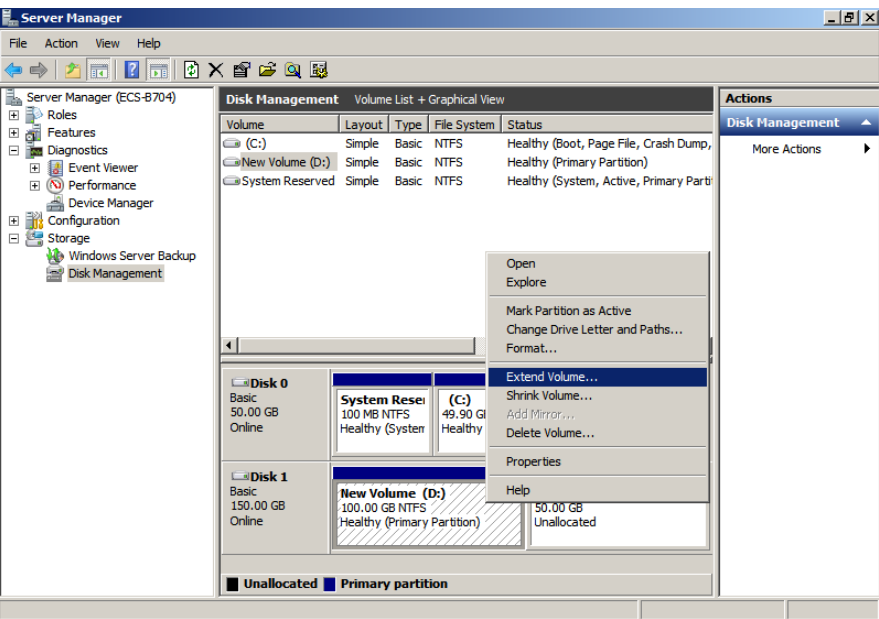
 NOTE

If you cannot see 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 volume that you want to extend. The current volume size and unallocated space are displayed.

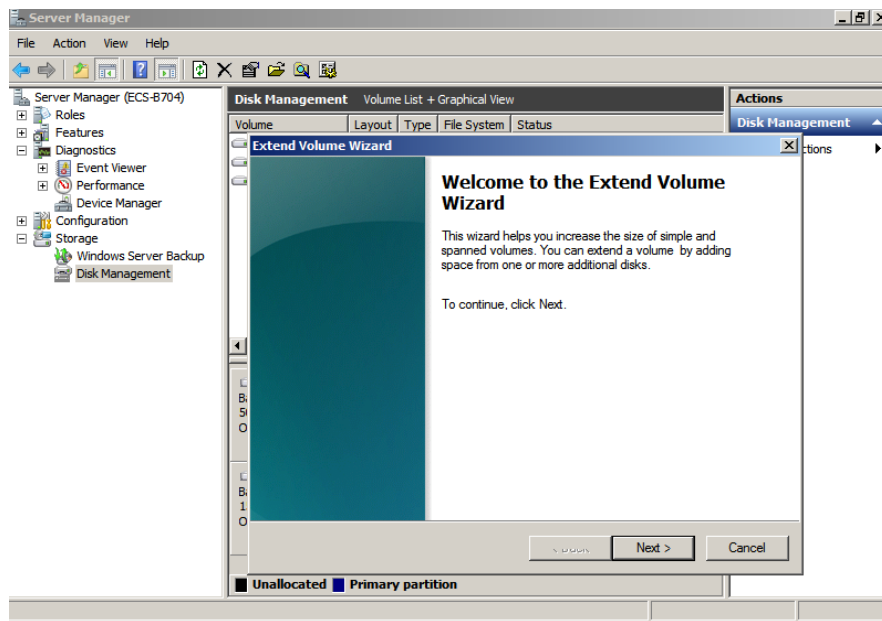
Step 4 Right-click the target volume and choose **Extend Volume**.

Figure 3-17 Choosing Extend Volume (Windows Server 2008)



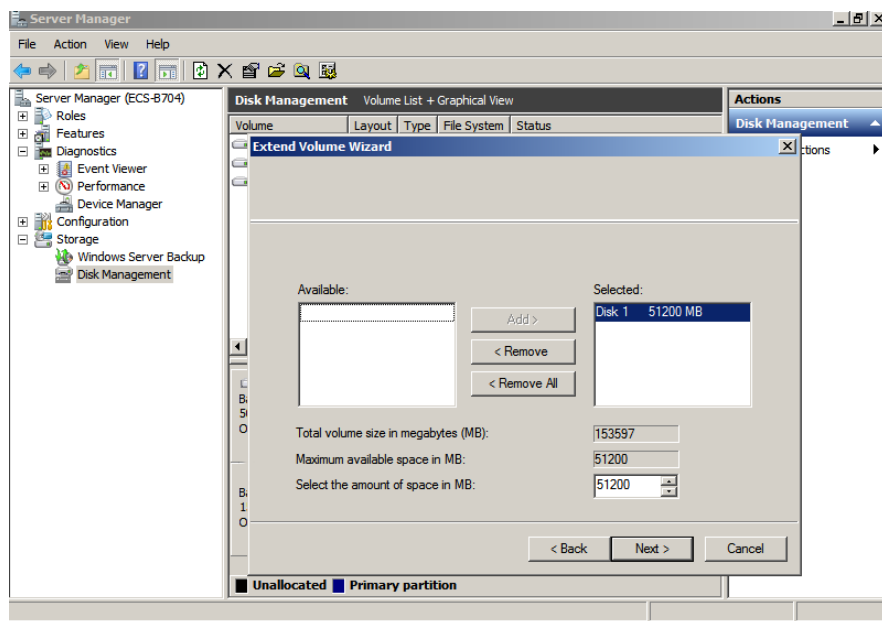
Step 5 On the displayed **Extend Volume Wizard** window, click **Next**.

Figure 3-18 Extend Volume Wizard (Windows Server 2008)



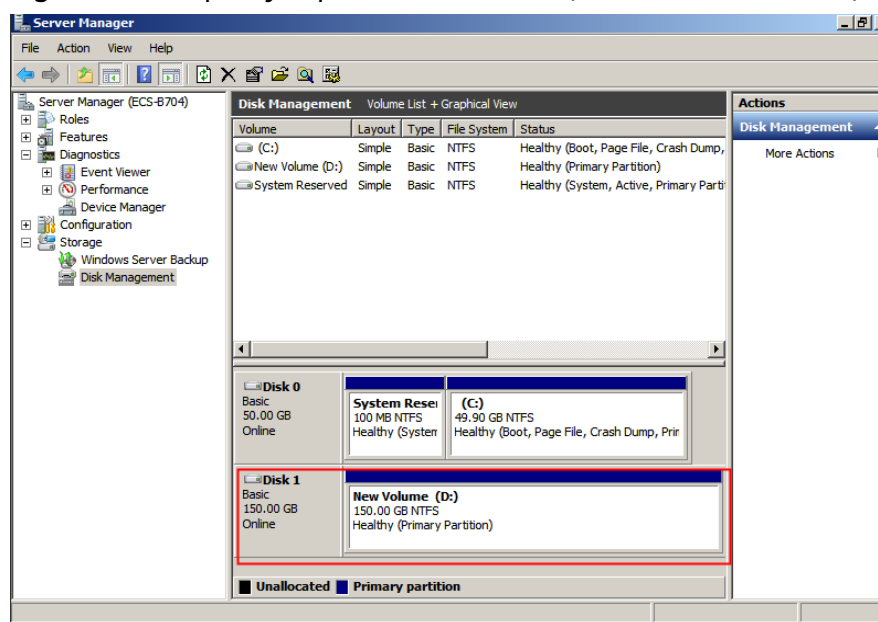
Step 6 In the text box to the right of **Select the amount of space in MB**, enter the amount of the additional space and click **Next**.

Figure 3-19 Selecting space (Windows Server 2008)



Step 7 Click **Finish**.

After the expansion succeeded, the partition size is larger than the original size.

Figure 3-20 Capacity expansion succeeded (Windows Server 2008)

----End

Data Disk: Create a New Volume with the Additional Space

In this example, the data disk has 40 GiB originally, and 60 GiB is added on the management console. The following procedure describes how to use this 60 GiB to create a new volume, for example volume (E:), on the server. After the operation is complete, new volume (E:) has 60 GiB of capacity and can be used as a data volume.

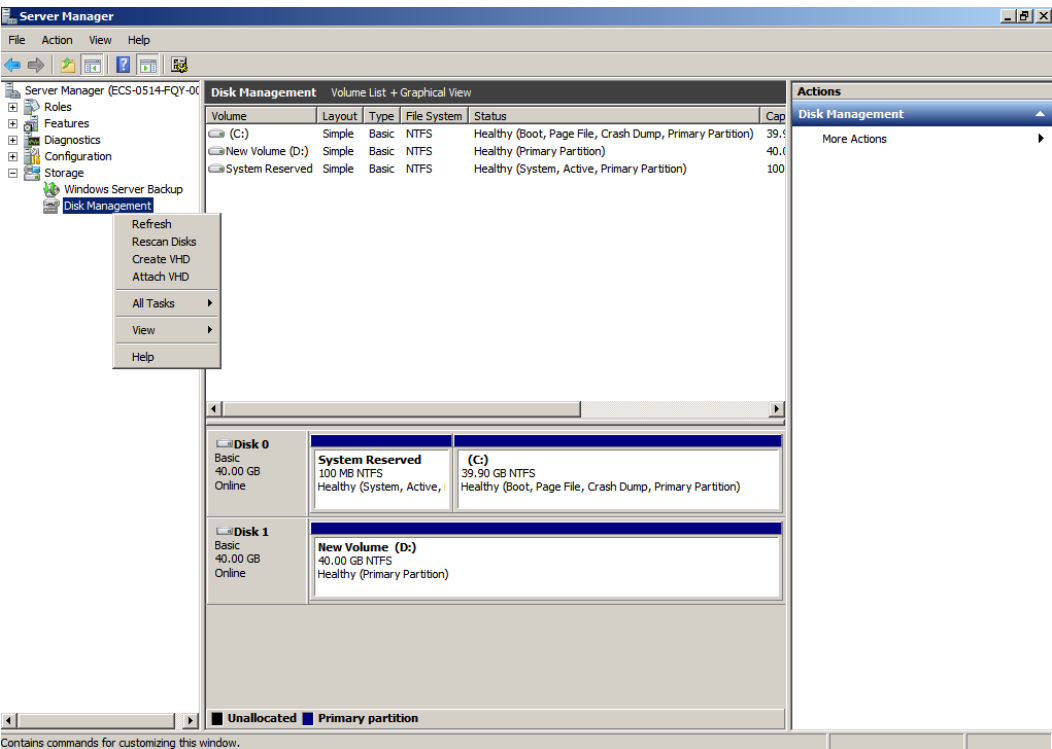
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

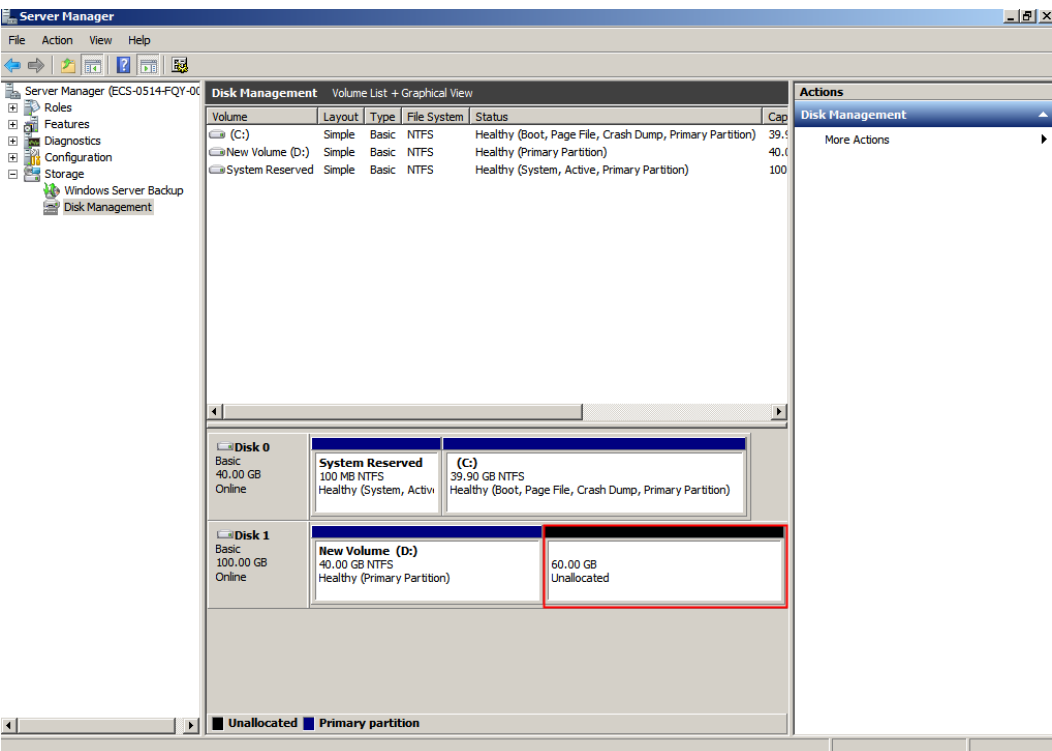
Figure 3-21 Refresh (data disk)



Step 3 If you cannot see the additional space, right-click **Disk Management** and choose **Refresh** from the shortcut menu.

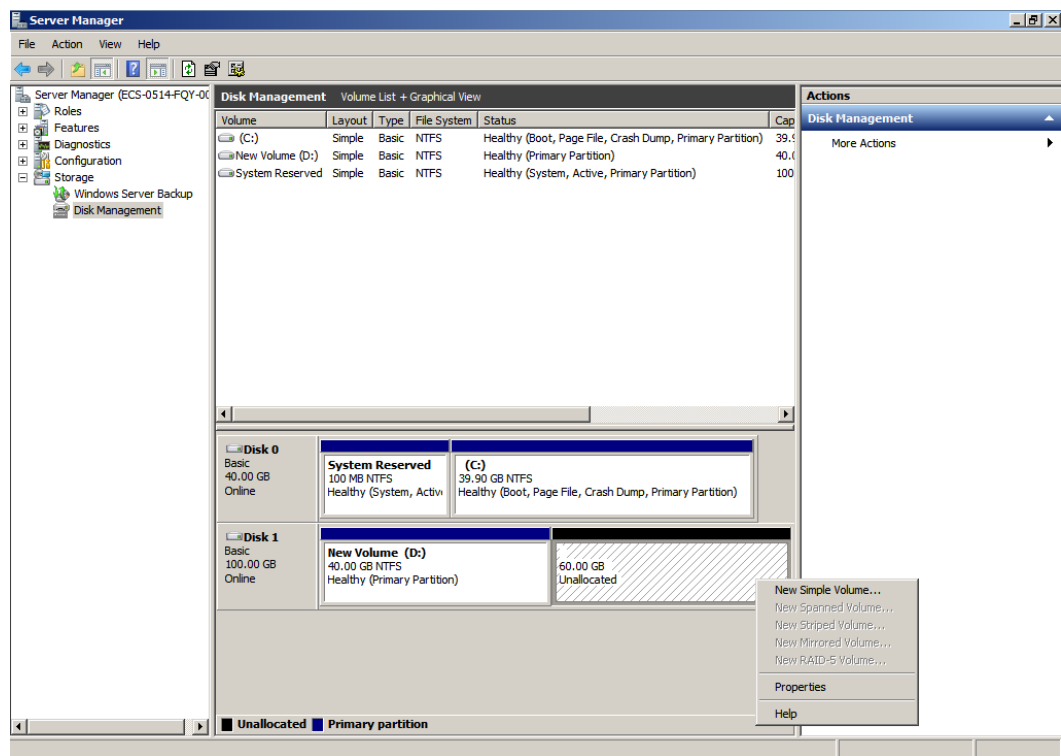
After the refresh, the additional space is displayed in the right area and is unallocated.

Figure 3-22 Unallocated disk space (data disk)



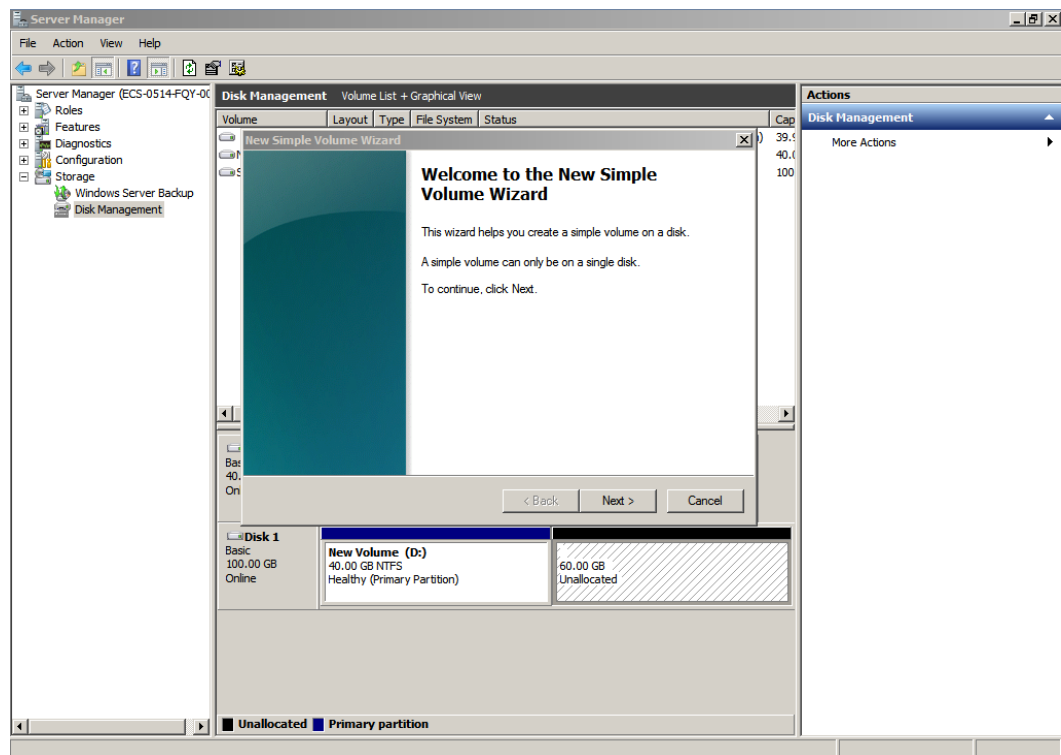
Step 4 In the **Unallocated** area of **Disk 1**, right-click the blank area and choose **New Simple Volume**.

Figure 3-23 New Simple Volume (data disk)



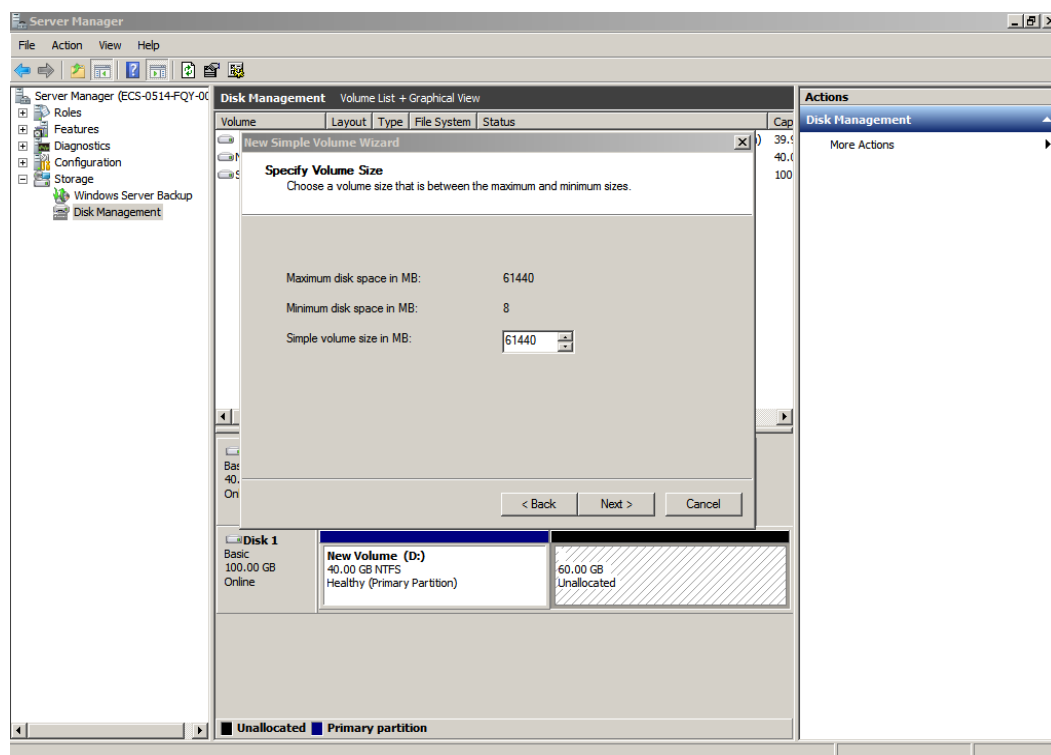
Step 5 On the displayed **New Simple Volume Wizard** window, click **Next**.

Figure 3-24 New Simple Volume Wizard (data disk)



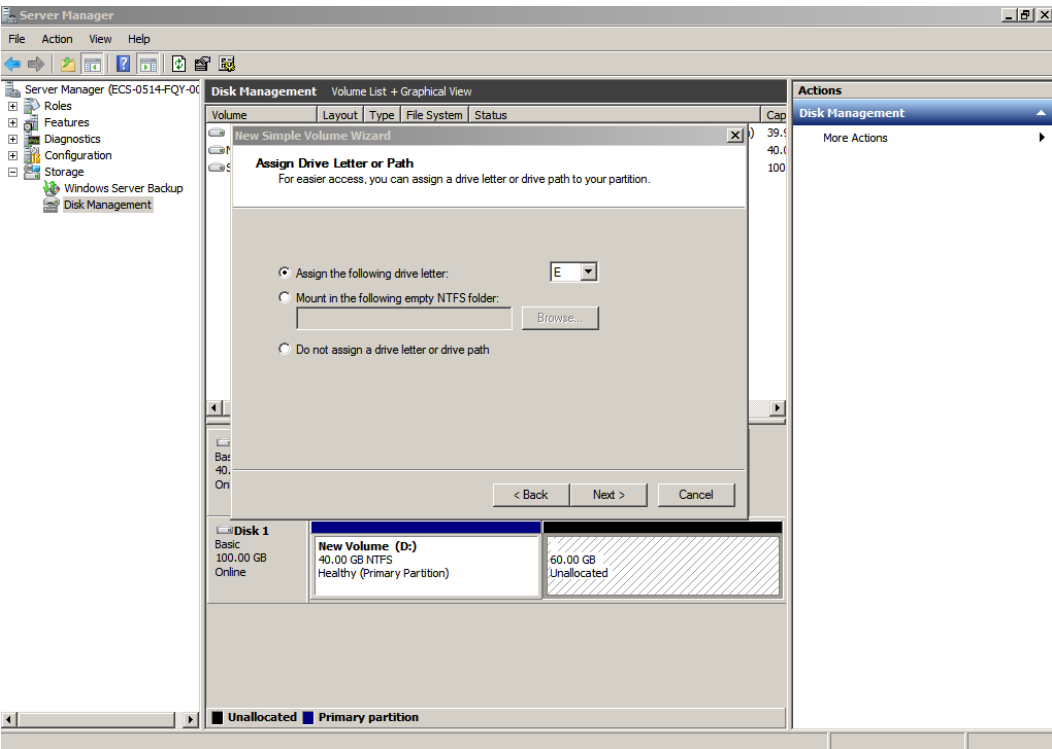
- Step 6** On the displayed **Specify Volume Size** page, set **Simple volume size in MB** and click **Next**. In this example, the default size is used.

Figure 3-25 Specify Volume Size (data disk)



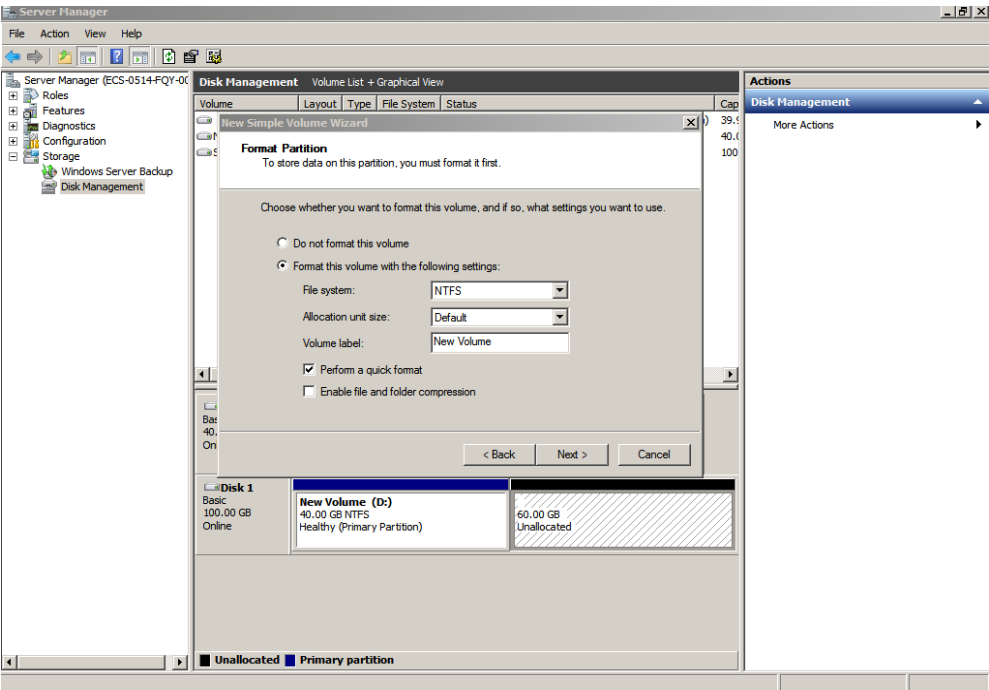
- Step 7** On the displayed **Assign Drive Letter and Path** page, click **Assign the following drive letter**, select a drive letter, and click **Next**. In this example, drive letter **E** is selected.

Figure 3-26 Assign Drive Letter or Path (data disk)



Step 8 On the displayed **Format Partition** page, click **Format this volume with the following settings**, set parameters based on the requirements, and select **Perform a quick format**. Then, click **Next**.

Figure 3-27 Format Partition (data disk)



Step 9 Click **Finish**.
After the expansion succeeded, new volume (E:) is displayed.

Figure 3-28 Completing the New Simple Volume Wizard (new volume E:)

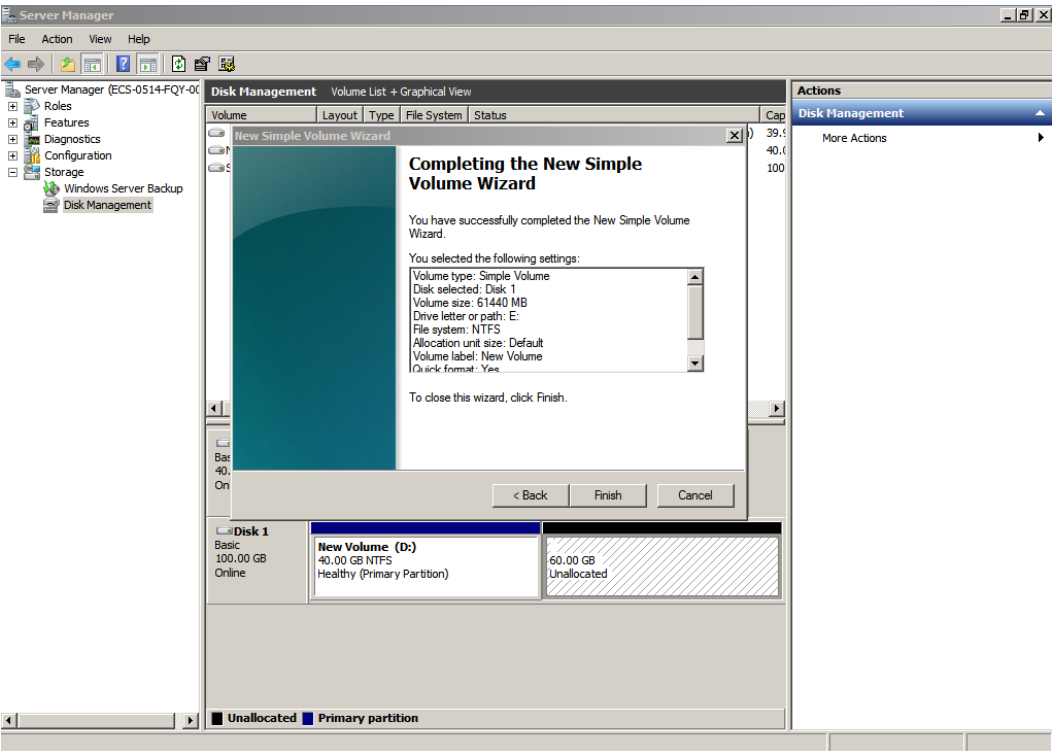
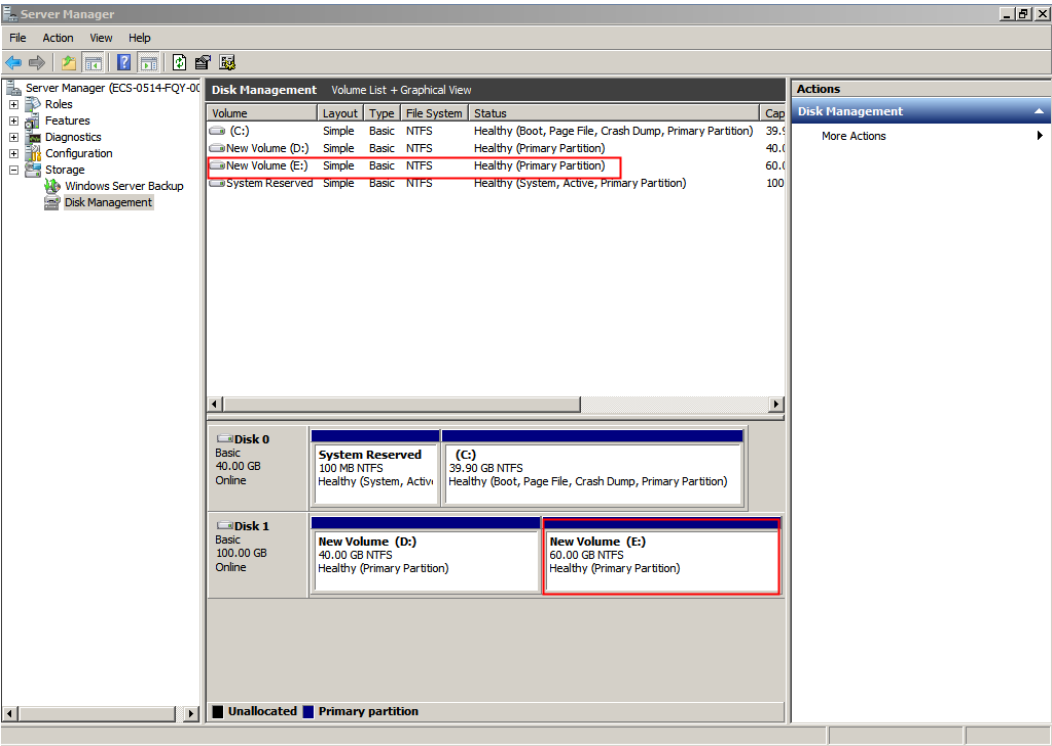


Figure 3-29 New Volume (E:)



----End

3.5 Extending Disk Partitions and File Systems (Linux)

3.5.1 Partition and File System Extension Preparations (Linux)

Before extending the disk partition and file system, you must check the disk partition style and file system format, and then select the appropriate operation accordingly.

1. To view the disk partition style, see the following methods:
 - [Method 1: Check Partition Style and File System Format Using fdisk](#)
 - [Method 2: Check Partition Style and File System Format Using parted](#)
2. To select the appropriate operation, see [Table 3-3](#).

Table 3-3 Disk partition and file system extension scenarios

Disk	Scenario	Method
System disk	Create a new MBR partition with the additional space.	Creating a New MBR Partition
	Allocate the additional space to an existing MBR partition.	• Extending an Existing MBR Partition (Kernel Version Later Than 3.6.0) • Extending an Existing MBR Partition (Kernel Version Earlier Than 3.6.0)
Data disk	Create a new MBR partition with the additional space.	Creating a New MBR Partition
	Allocate the additional space to an existing MBR partition.	Extending an Existing MBR Partition
	Create a new GPT partition with the additional space.	Creating a New GPT Partition
	Allocate the additional space to an existing GPT partition.	Extending an Existing GPT Partition
SCSI data disk	Create a new MBR partition with the additional space.	Creating a New MBR Partition
	Allocate the additional space to an existing MBR partition.	Extending an Existing MBR Partition

 NOTE

The maximum disk capacity that MBR supports is 2 TiB, and the disk space exceeding 2 TiB cannot be used.

If your disk uses MBR and you need to expand the disk capacity to over 2 TiB, 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.

Method 1: Check Partition Style and File System Format Using fdisk

Step 1 Run the following command to view all the disks attached to the server:

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 150G  0 disk
└─vdb1 253:17   0 100G  0 part /mnt/sdc
```

In this example, data disk **/dev/vdb** already has partition **/dev/vdb1** before capacity expansion, and the additional 50 GiB added has not been allocated yet. Therefore, **/dev/vdb** has 150 GiB, and **/dev/vdb1** has 100 GiB.

 NOTE

If you run **lsblk** and find out that disk **/dev/vdb** has no partitions, format the disk by referring to [How Do I Extend the File System of an Unpartitioned Data Disk in Linux?](#) and expand the capacity. Otherwise, the additional space cannot be used after expansion.

Step 2 Run the following command to view the current disk partition style:

fdisk -l

Information similar to the following is displayed:

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

Disk /dev/vda: 42.9 GiB, 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: 161.1 GiB, 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: dos
Disk identifier: 0x38717fc1

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

The value in the **System** column indicates the disk partition style. Value **Linux** indicates the MBR partition style. Value **GPT** indicates the GPT partition style.

- If the disk partitions displayed are inconsistent with those obtained in [Step 1](#), the possible reason may be that existing partitions uses GPT and there is unallocated disk space. In this case, you cannot query all the partitions using **fdisk -l**. Go to [Method 2: Check Partition Style and File System Format Using parted](#).
- If the disk partitions displayed are consistent with those obtained in [Step 1](#), continue with the following operations.

Step 3 Run the following command to view the partition's file system format:

blkid *Disk partition*

In this example, run the following command:

blkid /dev/vdb1

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

In the command output, the **TYPE** value is **ext4**, indicating that **/dev/vdb1**'s file system format is **ext4**.

Step 4 Run the following command to view the file system status:

ext*: **e2fsck -n** *Disk partition*

xfs: **xfs_repair -n** *Disk partition*

In this example, the ext4 file system is used. Therefore, run the following command:

e2fsck -n /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# e2fsck -n /dev/vdb1
e2fsck 1.42.9 (28-Dec-2013)
Warning! /dev/vdb1 is mounted.
Warning: skipping journal recovery because doing a read-only filesystem check.
/dev/vdb1: clean, 11/6553600 files, 459544/26214144 blocks
```

If the file system status is **clean**, the file system is normal. Otherwise, rectify the faulty and then perform the capacity expansion.

----End

Method 2: Check Partition Style and File System Format Using parted

Step 1 Run the following command to view all the disks attached to the server:

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 150G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
```

In this example, data disk **/dev/vdb** already has partition **/dev/vdb1** before capacity expansion, and the additional 50 GiB added has not been allocated yet. Therefore, **/dev/vdb** has 150 GiB, and **/dev/vdb1** has 100 GiB.

 NOTE

If you run **lsblk** and find out that disk **/dev/vdb** has no partitions, format the disk by referring to [How Do I Extend the File System of an Unpartitioned Data Disk in Linux?](#) and expand the capacity. Otherwise, the additional space cannot be used after expansion.

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

parted Disk

For example, run the following command to view **/dev/vdb**'s partition style:

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) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 161GiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
1       1049kB  107GiB  107GiB  ext4         test

(parted)
```

Partition Table indicates the disk partition style. **Partition Table: msdos** means MBR, **Partition Table: gpt** means GPT, and **Partition Table: loop** means that the whole disk is partitioned.

- If the following error information is displayed, enter **Fix**.

Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another operating system believes the disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?

The GPT partition table information is stored at the start of the disk. To reduce the risk of damage, a backup of the information is saved at the end of the disk. When you expand the disk capacity, the end of the disk changes accordingly. In this case, enter **Fix** to move the backup file of the information to new disk end.

- If the following warning information is displayed, enter **Fix**.

Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of the space (an extra 104857600 blocks) or continue with the current setting? Fix/Ignore? Fix

Enter **Fix** as prompted. The system automatically sets the GPT partition style for the additional space.

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

----End

3.5.2 Extending Partitions and File Systems for System Disks (Linux)

Scenarios

After a disk is expanded on the management console, the disk size is enlarged, but the additional space cannot be used directly.

In Linux, you must allocate the additional space to an existing partition or a new partition.

This section uses CentOS 7.4 64bit and CentOS 6.5 64bit as the sample OSs to describe how to extend the disk partition using `growpart` and `fdisk`. The method for allocating the additional space varies with the server OS. This section is used for reference only. For detailed operations and differences, see the corresponding OS documents.

For how to query the Linux kernel version, see [Querying the Linux Kernel Version](#).

- [Extending an Existing MBR Partition \(Kernel Version Later Than 3.6.0\)](#)
- [Extending an Existing MBR Partition \(Kernel Version Earlier Than 3.6.0\)](#)
- [Creating a New MBR Partition](#)

NOTICE

Performing the expansion operations with caution. Incorrect operations may lead to data loss or exceptions, so you are advised to back up the disk data using CBR before expansion. For details about using CBR, see [Managing EVS Backups](#).

Prerequisites

- You have expanded the disk capacity and attached the disk to a server on the management console. For details, see [Expanding Capacity for an In-use EVS Disk](#) or [Expanding Capacity for an Available EVS Disk](#).
- 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*.

Constraints

The additional space can only be added to the last partition of the disk.

Querying the Linux Kernel Version

Run the following command to query the Linux kernel version:

```
uname -a
```

Then, perform corresponding operations depending on whether the Linux kernel version is later than 3.6.0.

- For CentOS 7.4 64bit, information similar to the following is displayed:
Linux ecs-test-0001 3.10.0-957.5.1.el7.x86_64 #1 SMP Fri Feb 1 14:54:57 UTC 2019 x86_64 x86_64
The kernel version is 3.10.0, which is later than 3.6.0. For subsequent operations, see [Extending an Existing MBR Partition \(Kernel Version Later Than 3.6.0\)](#).
- For CentOS 6.5 64bit, information similar to the following is displayed:
Linux ecs-test-0002 2.6.32-754.10.1.el6.x86_64 #1 SMP Tue Jan 15 17:07:28 UTC 2019 x86_64
The kernel version is 2.6.32, which is earlier than 3.6.0. In this case, the disk partition and file system extension take effect only after a server reboot. For subsequent operations, see [Extending an Existing MBR Partition \(Kernel Version Earlier Than 3.6.0\)](#).

Extending an Existing MBR Partition (Kernel Version Later Than 3.6.0)

CentOS 7.4 64bit is used as the sample OS. Originally, system disk **/dev/vda** has 40 GiB and one partition (**/dev/vda1**), and then 60 GiB is added to the disk. The following procedure shows you how to allocate the additional 60 GiB to the existing MBR partition **/dev/vda1**.

Step 1 (Optional) Run the following command to install the growpart tool:

yum install cloud-utils-growpart

NOTE

You can run **growpart** to check whether the growpart tool has been installed. If the command output displays the tool usage instructions, the tool has been installed and you do not need to install it again.

Step 2 Run the following command to view the total capacity of the **/dev/vda** system disk:

fdisk -l

Information similar to the following is displayed:

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

Disk /dev/vda: 107.4 GiB, 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: 0x000bcb4e
```

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

Step 3 Run the following command to view the capacity of the **/dev/vda1** partition:

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   2.0G  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
```

Step 4 Run the following command to extend the partition using growpart:

growpart *System disk Partition number*

In this example, run the following command:

growpart /dev/vda 1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# growpart /dev/vda 1
CHANGED: partition=1 start=2048 old: size=83884032 end=83886080 new: size=209713119,end=209715167
```

Step 5 Run the following command to extend the file system of the partition:

resize2fs *Disk partition*

In this example, run the following command:

resize2fs /dev/vda1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# resize2fs /dev/vda1
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vda1 is mounted on /; on-line resizing required
old_desc_blocks = 5, new_desc_blocks = 13
The filesystem on /dev/vda1 is now 26214139 blocks long.
```

NOTE

If the error message "open: No such file or directory while opening /dev/vdb1" is returned, an incorrect partition is specified. Run **df -TH** to view the disk partitions.

Step 6 Run the following command to view the new capacity of the **/dev/vda1** partition:

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      106G  2.0G  99G   2% /
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
```

----End

Extending an Existing MBR Partition (Kernel Version Earlier Than 3.6.0)

NOTICE

If the OS kernel version is earlier than 3.6.0, the extension of an existing MBR disk partition and file system takes effect only after a server reboot, and services will be interrupted.

CentOS 6.5 64bit is used as the sample OS. Originally, system disk **/dev/vda** has 40 GiB and one partition (**/dev/vda1**), and then 60 GiB is added to the disk. The following procedure shows you how to allocate the additional 60 GiB to the existing MBR partition **/dev/vda1**.

Step 1 (Optional) Run the following command to install the growpart tool:

```
yum install cloud-utils-growpart
```

 NOTE

You can run **growpart** to check whether the growpart tool has been installed. If the command output displays the tool usage instructions, the tool has been installed and you do not need to install it again.

Step 2 Run the following command to install the dracut-modules-growroot tool:

```
yum install cloud-utils-growpart
```

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# yum install cloud-utils-growpart
Loaded plugins: fastestmirror, security
Setting up Install Process
Determining fastest mirrors
...
Package cloud-utils-growpart-0.27-10.el6.x86_64 already installed and latest version
Nothing to do
```

Step 3 Run the following command to regenerate the initramfs file:

```
dracut -f
```

Step 4 Run the following command to view the total capacity of the **/dev/vda** system disk:

```
fdisk -l
```

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# fdisk -l

Disk /dev/vda: 107.4 GiB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0004e0be

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1  *           1         5222     41942016   83  Linux
```

Step 5 Run the following command to view the capacity of the **/dev/vda1** partition:

```
df -TH
```

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# df -TH
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/vda1      ext4   43G   1.7G   39G   5% /
tmpfs          tmpfs  2.1G     0   2.1G   0% /dev/shm
```

Step 6 Run the following command to extend the partition using growpart:

```
growpart System disk Partition number
```

In this example, run the following command:

```
growpart /dev/vda 1
```

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# growpart /dev/vda 1
CHANGED: partition=1 start=2048 old: size=83884032 end=83886080 new: size=209710462,end=209712510
```

Step 7 Run the following command to restart the server:

reboot

After the server is restarted, reconnect to the server and perform the following steps.

Step 8 Run the following command to extend the file system of the partition:

resize2fs *Disk partition*

In this example, run the following command:

resize2fs /dev/vda1

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# resize2fs /dev/vda1
resize2fs 1.41.12 (17-May-2010)
The filesystem is already 26213807 blocks long. Nothing to do!
```

 NOTE

If the error message "open: No such file or directory while opening /dev/vdb1" is returned, an incorrect partition is specified. Run **df -TH** to view the disk partitions.

Step 9 Run the following command to view the new capacity of the **/dev/vda1** partition:

df -TH

Information similar to the following is displayed:

```
[root@ecs-test-0002 ~]# df -TH
Filesystem      Type  Size  Used Avail Use% Mounted on
/dev/vda1       ext4  106G  1.7G  99G   2% /
tmpfs           tmpfs  2.1G   0  2.1G   0% /dev/shm
```

----End

Creating a New MBR Partition

Originally, system disk **/dev/vda** has 40 GiB and one partition (**/dev/vda1**), and then 40 GiB is added to the disk. The following procedure shows you how to create a new MBR partition **/dev/vda2** with this 40 GiB.

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 GiB, 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 GiB, in which the in-use **dev/vda1** partition takes 40 GiB and the additional 40 GiB 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.

 NOTE

If MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition 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 (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 and press **Enter**. In this example, the default start sector is used.

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 (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**. In this example, the default end sector is used.

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} (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 GiB, 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:

mkfs -t *File system Disk partition*

- Sample command of the ext* file system:
(The ext4 file system is used in this example.)

mkfs -t ext4 /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
```

- Sample command of the xfs file system:

mkfs -t xfs /dev/vda2

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# mkfs -t xfs /dev/vda2
meta-data=/dev/vda2          isize=512    agcount=4, agsize=2621440 blks
        =                   sectsz=512   attr=2, projid32bit=1
        =                   crc=1        finobt=0, sparse=0
data      =                   bsize=4096   blocks=10485760, imaxpct=25
        =                   sunit=0      swidth=0 blks
naming    =version2          bsize=4096   ascii-ci=0 ftype=1
log       =internal log     bsize=4096   blocks=5120, version=2
        =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none             extsz=4096   blocks=0, rtextents=0
```

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 (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /opt

Step 13 Run the following command to mount the new partition:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vda2** on **/opt**:

mount /dev/vda2 /opt

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 14 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
```

NOTE

If the server is restarted, the mounting will become invalid. You can modify the `/etc/fstab` file to configure automount at startup. See the following part for details.

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at server startup. You can configure the **fstab** file of a server that has data. This operation will not affect the existing data.

The following example uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names (like `/dev/vdb1`) to identify disks in the file because device names are assigned dynamically and may change (for example, from `/dev/vdb1` to `/dev/vdb2`) after a server stop or start. This can even prevent your server from booting up.

NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the `/dev/vdb1` partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi `/etc/fstab`

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID**: The UUID obtained in [Step 1](#).
- **Mount point**: The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem**: The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option**: The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump**: The Linux dump backup option.
 - **0**: Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1**: Linux dump backup is used.
- **fsck**: The fsck option, which means whether to use fsck to check the disk during startup.
 - **0**: not use fsck.
 - If the mount point is the root partition (/), this parameter must be set to **1**.

If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

3.5.3 Extending Partitions and File Systems for Data Disks (Linux)

Scenarios

After a disk is expanded on the management console, the disk size is enlarged, but the additional space cannot be used directly.

In Linux, you must allocate the additional space to an existing partition or a new partition.

This section uses CentOS 7.4 64bit as the sample OS to describe how to extend an MBR or GPT partition. The method for allocating the additional space varies with the server OS. This section is used for reference only. For detailed operations and differences, see the corresponding OS documents.

- [Creating a New MBR Partition](#)
- [Extending an Existing MBR Partition](#)
- [Creating a New GPT Partition](#)
- [Extending an Existing GPT Partition](#)

NOTICE

Performing the expansion operations with caution. Incorrect operations may lead to data loss or exceptions, so you are advised to back up the disk data using CBR before expansion. For details about using CBR, see [Managing EVS Backups](#).

Prerequisites

- You have expanded the disk capacity and attached the disk to a server on the management console. For details, see [Expanding Capacity for an In-use EVS Disk](#) or [Expanding Capacity for an Available EVS Disk](#).
- 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*.

Constraints

- The additional space of a data disk cannot be added to the root partition. To extend the root partition, expand the system disk instead.
- The additional space can only be added to the last partition of the disk.

Creating a New MBR Partition

Originally, data disk `/dev/vdb` has 100 GiB and one partition (`/dev/vdb1`), and then 50 GiB is added to the disk. The following procedure shows you how to create a new MBR partition `/dev/vdb2` with this 50 GiB.

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

```
fdisk -l
```

Information similar to the following is displayed:

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

Disk /dev/vda: 42.9 GiB, 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: 161.1 GiB, 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: dos
Disk identifier: 0x38717fc1

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

Step 2 Run the following command to enter fdisk:

fdisk *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.


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
Select (default p):
```

There are two types of disk partitions:

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

NOTE

If MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition 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 (2-4, default 2):
```

Partition number indicates the serial number of the primary partition. Because partition number 1 has been used, the value ranges from 2 to 4.

- 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 (209715200-314572799, default 209715200):
```

First sector indicates the start sector. The value ranges from 209715200 to 314572799, and the default value is 209715200.

- Step 6** Enter the new partition's start sector and press **Enter**. In this example, the default start sector is used.

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 (209715200-314572799, default 209715200):
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-314572799, default 314572799):
```

Last sector indicates the end sector. The value ranges from 209715200 to 314572799, and the default value is 314572799.

- Step 7** Enter the new partition's end sector and press **Enter**. In this example, the default end sector is used.

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} (209715200-314572799, default 314572799):
Using default value 314572799
Partition 2 of type Linux and of size 50 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/vdb: 161.1 GiB, 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: dos
Disk identifier: 0x38717fc1

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1           2048     209715199     104856576    83  Linux
/dev/vdb2    209715200     314572799     52428800    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.
```

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 Disk partition*

- Sample command of the ext* file system:

mkfs -t ext4 /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb2
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
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 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

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

- Sample command of the xfs file system:

mkfs -t xfs /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t xfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
       =                   sectsz=512   attr=2,  projid32bit=1
       =                   crc=1        finobt=0, sparse=0
data      =                   bsize=4096   blocks=13107200, imaxpct=25
       =                   sunit=0       swidth=0 blks
naming    =version2          bsize=4096   ascii-ci=0 ftype=1
log       =internal log      bsize=4096   blocks=6400, version=2
       =                   sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
```

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 (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

Step 13 Run the following command to mount the new partition:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vdb2** on **/mnt/test**:

mount /dev/vdb2 /mnt/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 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
/dev/vdb2       ext4      53G   55M   50G    1% /mnt/test
```

 NOTE

If the server is restarted, the mounting will become invalid. You can modify the **/etc/fstab** file to configure automount at startup. For details, see [Configuring Automatic Mounting at System Start](#).

----End

Extending an Existing MBR Partition

NOTICE

If the additional space is allocated to an existing partition, data on the disk will not be cleared but you must use **umount** to unmount the existing partition. In this case, services will be affected.

Originally, data disk **/dev/vdb** has 150 GiB and two partitions (**/dev/vdb1** and **/dev/vdb2**), and then 80 GiB is added to the disk. The following procedure shows you how to add this 80 GiB to the existing MBR partition **/dev/vdb2**.

NOTICE

During an expansion, the additional space is added to the end of the disk. Therefore, if the disk has multiple partitions, the additional space can only be allocated to the partition at the disk end.

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

fdisk -l

Information similar to the following is displayed:

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

```
Disk /dev/vda: 42.9 GiB, 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: 247.0 GiB, 246960619520 bytes, 482344960 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
/dev/vdb2		209715200	314572799	52428800	83	Linux

In the command output, take note of the partition's start and end sectors. In this example, **/dev/vdb2**'s start sector is **209715200**, and its end sector is **314572799**.

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

- If the additional space is not included, refresh the capacity according to [Extending Partitions and File Systems for SCSI Disks \(Linux\)](#).
- If the additional space is included, take note of the start and end sectors of the target partition and then go to [Step 2](#). These values will be used in the subsequent operations.

Step 2 Run the following command to unmount the partition:

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb2

Step 3 Run the following command to enter fdisk:

fdisk *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.

Command (m for help):

Step 4 Run the following command to delete the partition to be extended:

1. Enter **d** and press **Enter** to delete the partition.

Information similar to the following is displayed:

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

```
Partition number (1,2, default 2):
```

2. Enter the partition number and press **Enter** to delete the partition. In this example, enter **2**.

Information similar to the following is displayed:

```
Partition number (1,2, default 2): 2
```

```
Partition 2 is deleted
```

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

NOTE

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

Step 5 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
```

```
Select (default p):
```

There are two types of disk partitions:

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

NOTE

If MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

Step 6 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 (2-4, default 2):
```

Partition number indicates the serial number of the primary partition.

Step 7 Ensure that entered partition number is the same as the partition had before. In this example, partition number **2** is used. Therefore, enter **2** and press **Enter**.

Information similar to the following is displayed:

```
Partition number (2-4, default 2): 2
First sector (209715200-482344959, default 209715200):
```

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 8 Ensure that the entered start sector is the same as the partition had before. In this example, start sector **209715200** is recorded in [Step 1](#). Therefore, enter **209715200** and press **Enter**.

Information similar to the following is displayed:

```
First sector (209715200-482344959, default 209715200):
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-482344959, default 482344959):
```

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

Step 9 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 **314572799**, and the default end sector is used. Therefore, enter **482344959** and press **Enter**.

Information similar to the following is displayed:

```
Using default value 209715200
Last sector, +sectors or +size{K,M,G} (209715200-482344959, default 482344959):
Using default value 482344959
Partition 2 of type Linux and of size 130 GiB is set
Command (m for help):
```

The partition is created.

Step 10 Enter **p** and press **Enter** to print the partition details.

Information similar to the following is displayed:

```
Command (m for help): p

Disk /dev/vdb: 247.0 GiB, 246960619520 bytes, 482344960 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
/dev/vdb2		209715200	482344959	136314880	83	Linux

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

Step 11 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.

 NOTE

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

Step 12 Run the following command to synchronize the new partition table to the OS:

partprobe

Step 13 Perform the following operations based on the file system of the disk:

- For the **ext*** file system
 - a. Run the following command to check the correctness of the file system on the partition:

e2fsck -f *Disk partition*

In this example, run the following command:

e2fsck -f /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# e2fsck -f /dev/vdb2
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/vdb2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. Run the following command to extend the file system of the partition:

resize2fs *Disk partition*

In this example, run the following command:

resize2fs /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# resize2fs /dev/vdb2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vdb2 to 34078720 (4k) blocks.
The filesystem on /dev/vdb2 is now 34078720 blocks long.
```

 NOTE

If the error message "open: No such file or directory while opening /dev/vdb1" is returned, an incorrect partition is specified. Run **df -TH** to view the disk partitions.

- c. (Optional) Run the following command to create a mount point:
Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

- d. Run the following command to mount the partition:

mount *Disk partition Mount point*

In this example, run the following command to mount partition **/dev/vdb2** on **/mnt/test**:

mount /dev/vdb2 /mnt/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.

- For the **xfs** file system

- (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

- Run the following command to mount the partition:

mount *Disk partition Mount point*

In this example, run the following command to mount partition **/dev/vdb2** on **/mnt/test**:

mount /dev/vdb2 /mnt/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.

- Run the following command to extend the file system of the partition:

sudo xfs_growfs *Disk partition*

In this example, run the following command:

sudo xfs_growfs /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# sudo xfs_growfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
=                               sectsz=512   attr=2,    projid32bit=1
=                               crc=1      finobt=0,   spinodes=0
data     =                       bsize=4096   blocks=13107200, imaxpct=25
=                               sunit=0      swidth=0 blks
naming   =version2             bsize=4096   ascii-ci=0 ftype=1
log      =internal             bsize=4096   blocks=6400, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                 extsz=4096   blocks=0, rtextents=0
data blocks changed from 13107200 to 34078720.
```

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
/dev/vdb2    ext4    138G  63M  131G  1% /mnt/test
```

----End

Creating a New GPT Partition

Originally, data disk **/dev/vdb** has 100 GiB and one partition (**/dev/vdb1**), and then 50 GiB is added to the disk. The following procedure shows you how to create a new GPT partition **/dev/vdb2** with this 50 GiB.

Step 1 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 150G  0 disk
└─vdb1 253:17  0 100G  0 part /mnt/sdc
```

Step 2 Run the following command to enter parted:

parted 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 **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 disk partition information.

Information similar to the following is displayed:

```
(parted) unit s
(parted) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 314572800s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start End      Size      File system Name  Flags
```

```
1    2048s 209713151s 209711104s ext4    test
(parted)
```

In the command output, take note of the partition's end sector. In this example, the end sector of the **/dev/vdb1** partition is **209713151s**.

- If the following error information is displayed, enter **Fix**.

Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another operating system believes the disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?

The GPT partition table information is stored at the start of the disk. To reduce the risk of damage, a backup of the information is saved at the end of the disk. When you expand the disk capacity, the end of the disk changes accordingly. In this case, enter **Fix** to move the backup file of the information to new disk end.

- If the following warning information is displayed, enter **Fix**.

Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of the space (an extra 104857600 blocks) or continue with the current setting?
Fix/Ignore? Fix

Enter **Fix** as prompted. The system automatically sets the GPT partition style for the additional space.

Step 5 Run the following command and press **Enter**:

mkpart *Partition name Start sector End sector*

In this example, run the following command:

mkpart data 209713152s 100%

In this example, the additional space is used to create a new partition. In [Step 4](#), the end sector of partition **dev/vdb1** is **209713151s**. Therefore, the start sector of the new partition **dev/vdb2** is set to **209713152s** and the end sector **100%**. This start and end sectors are for reference only. You can plan the number of partitions and partition size based on service requirements.

Information similar to the following is displayed:

```
(parted) mkpart data 209713152s 100%
(parted)
```

NOTE

The maximum sector can be obtained in either of the following ways:

- Query the disk's maximum end sector. For details, see [Step 2](#) to [Step 4](#).
- Enter **-1s** or **100%**, and the value displayed is the maximum end sector.

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

Information similar to the following is displayed:

```
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 314572800s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start      End          Size         File system  Name  Flags
 1      2048s     209713151s  209711104s  ext4         test
 2      209713152s 314570751s  104857600s                data
```

(parted)

Step 7 Run the following command to set the file system format for the new partition:

mkfs -t *File system Disk partition*

- Sample command of the ext* file system:

mkfs -t ext4 /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb2
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
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 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

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

- Sample command of the xfs file system:

mkfs -t xfs /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t xfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
       =                       sectsz=512   attr=2, projid32bit=1
       =                       crc=1        finobt=0, sparse=0
data      =                       bsize=4096   blocks=13107200, imaxpct=25
       =                       sunit=0      swidth=0 blks
naming   =version2           bsize=4096   ascii-ci=0 ftype=1
log      =internal log      bsize=4096   blocks=6400, version=2
       =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none              extsz=4096   blocks=0, rtextents=0
```

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 8 (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

Step 9 Run the following command to mount the new partition:

mount *Disk partition Mount point*

In this example, run the following command to mount the new partition **/dev/vdb2** on **/mnt/test**:

```
mount /dev/vdb2 /mnt/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 10 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
/dev/vdb2       ext4      53G   55M   50G    1% /mnt/test
```

NOTE

If the server is restarted, the mounting will become invalid. You can modify the **/etc/fstab** file to configure automount at startup. For details, see [Configuring Automatic Mounting at System Start](#).

----End

Extending an Existing GPT Partition

NOTICE

If the additional space is allocated to an existing partition, data on the disk will not be cleared but you must use **umount** to unmount the existing partition. In this case, services will be affected.

Originally, data disk **/dev/vdb** has 150 GiB and two partitions (**/dev/vdb1** and **/dev/vdb2**), and then 80 GiB is added to the disk. The following procedure shows you how to add this 80 GiB to the existing GPT partition **/dev/vdb2**.

During an expansion, the additional space is added to the end of the disk. Therefore, if the disk has multiple partitions, the additional space can only be allocated to the partition at the disk end.

Step 1 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 230G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
  └─vdb2 253:18 0 50G 0 part /mnt/test
```

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

- If the additional space is not included, refresh the capacity according to [Extending Partitions and File Systems for SCSI Disks \(Linux\)](#).
- If the additional space is included, go to [Step 2](#).

Step 2 Run the following command to unmount the partition:

```
umount Disk partition
```

In this example, run the following command:

```
umount /dev/vdb2
```

Step 3 Run the following command to view the unmount result:

```
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 230G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
  └─vdb2 253:18 0 50G 0 part
```

Step 4 Run the following command to enter parted:

```
parted 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 5 Enter **unit s** and press **Enter** to set the measurement unit of the disk to sector.

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

Information similar to the following is displayed:

```
(parted) unit s
(parted) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 167772160
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 482344960s
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	2048s	209713151s	209711104s	ext4	test	
2	209713152s	314570751s	104857600s	ext4	data	

```
(parted)
```

Take note of the start and end sectors of the **/dev/vdb2** partition. These values will be used during the partition recreation. In this example, the partition's start sector is **209713152s**, and its end sector is **314570751s**.

- If the following error information is displayed, enter **Fix**.

```
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that
another operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
```

The GPT partition table information is stored at the start of the disk. To reduce the risk of damage, a backup of the information is saved at the end of the disk. When you expand the disk capacity, the end of the disk changes accordingly. In this case, enter **Fix** to move the backup file of the information to new disk end.

- If the following warning information is displayed, enter **Fix**.

```
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all
of the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
```

Enter **Fix** as prompted. The system automatically sets the GPT partition style for the additional space.

Step 7 Enter **rm** and the partition number, and then press **Enter**. In this example, partition number **2** is used.

Information similar to the following is displayed:

```
(parted) rm
Partition number? 2
(parted)
```

Step 8 Run the following command to re-create the partition and press **Enter**:

mkpart *Partition name Start sector End sector*

In this example, run the following command:

mkpart data 209713152s 100%

- Ensure that the entered start sector is the same as the partition had before. In this example, start sector **209713152s** is recorded in [Step 6](#). Therefore, enter **209713152s**.
- Ensure that the entered end sector is greater than the partition had before. In this example, the end sector recorded in [Step 6](#) is **314570751s**, and all the additional space needs to be allocated to **dev/vdb2**. Therefore, enter **100%**.

Information similar to the following is displayed:

```
(parted) mkpart data 209713152s 100%
(parted)
```

 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 Enter **p** and press **Enter** to view the partition information.

Information similar to the following is displayed:

```
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 482344960s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start      End          Size         File system  Name  Flags
 1    2048s    209713151s  209711104s  ext4        test
 2    209713152s 482342911s  272629760s  ext4        data

(parted)
```

Step 10 Perform the following operations based on the file system of the disk:

- For the **ext*** file system
 - a. Run the following command to check the correctness of the file system on the partition:

e2fsck -f *Disk partition*

In this example, run the following command:

e2fsck -f /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# e2fsck -f /dev/vdb2
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/vdb2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. Run the following command to extend the file system of the partition:

resize2fs *Disk partition*

In this example, run the following command:

resize2fs /dev/vdb2

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# resize2fs /dev/vdb2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/vdb2 to 34078720 (4k) blocks.
The filesystem on /dev/vdb2 is now 34078720 blocks long.
```

 NOTE

If the error message "open: No such file or directory while opening /dev/vdb1" is returned, an incorrect partition is specified. Run **df -TH** to view the disk partitions.

- c. (Optional) Run the following command to create a mount point:
Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

```
mkdir /mnt/test
```

- d. Run the following command to mount the partition:

```
mount Disk partition Mount point
```

In this example, run the following command to mount partition **/dev/vdb2** on **/mnt/test**:

```
mount /dev/vdb2 /mnt/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.

- For the **xfs** file system

- a. (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

```
mkdir Mount point
```

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

```
mkdir /mnt/test
```

- b. Run the following command to mount the partition:

```
mount Disk partition Mount point
```

In this example, run the following command to mount partition **/dev/vdb2** on **/mnt/test**:

```
mount /dev/vdb2 /mnt/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.

- c. Run the following command to extend the file system of the partition:

```
sudo xfs_growfs Disk partition
```

In this example, run the following command:

```
sudo xfs_growfs /dev/vdb2
```

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# sudo xfs_growfs /dev/vdb2
meta-data=/dev/vdb2          isize=512    agcount=4, agsize=3276800 blks
        =                   sectsz=512   attr=2,    projid32bit=1
        =                   crc=1        finobt=0,   spinodes=0
data      =                   bsize=4096   blocks=13107200, imaxpct=25
        =                   sunit=0      swidth=0 blks
naming    =version2          bsize=4096   ascii-ci=0 ftype=1
```

```
log      =internal      bsize=4096   blocks=6400, version=2
          =              sectsz=512   sunit=0   blks, lazy-count=1
realtime =none          extsz=4096   blocks=0, rtextents=0
data blocks changed from 13107200 to 34078720.
```

Step 11 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
/dev/vdb2       ext4     138G   63M  131G   1% /mnt/test
```

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at server startup. You can configure the **fstab** file of a server that has data. This operation will not affect the existing data.

The following example uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names (like **/dev/vdb1**) to identify disks in the file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a server stop or start. This can even prevent your server from booting up.

NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi /etc/fstab

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID:** The UUID obtained in [Step 1](#).
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.
 - **0:** not use fsck.
 - If the mount point is the root partition (/), this parameter must be set to **1**.

If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

3.5.4 Extending Partitions and File Systems for SCSI Disks (Linux)

Scenarios

After a disk is expanded on the management console, the disk size is enlarged, but the additional space cannot be used directly.

In Linux, you must allocate the additional space to an existing partition or a new partition.

This section uses CentOS 7.4 64bit as the sample OS to describe how to extend an MBR partition of a SCSI data disk. The method for allocating the additional space varies with the server OS. This section is used for reference only. For detailed operations and differences, see the corresponding OS documents.

- [Creating a New MBR Partition](#)
- [Extending an Existing MBR Partition](#)

NOTICE

Performing the expansion operations with caution. Incorrect operations may lead to data loss or exceptions, so you are advised to back up the disk data using CBR before expansion. For details about using CBR, see [Managing EVS Backups](#).

Prerequisites

- You have expanded the disk capacity and attached the disk to a server on the management console. For details, see [Expanding Capacity for an In-use EVS Disk](#) or [Expanding Capacity for an Available EVS Disk](#).
- 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 a New MBR Partition

Originally, data disk **/dev/sda** has 50 GiB and one partition (**/dev/sda1**), and then 50 GiB is added to the disk. The following procedure shows you how to create a new MBR partition **/dev/sda2** with this 50 GiB.

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

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# fdisk -l
```

```
Disk /dev/vda: 42.9 GiB, 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/sda: 107.4 GiB, 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: 0x915ffe6a
```

```
Device Boot      Start         End      Blocks   Id  System
/dev/sda1        2048    104857599   52427776   83  Linux
```

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

- If the additional space is not included, refresh the capacity according to [Step 2](#).
- If the additional space is included, go to [Step 3](#).

Step 2 (Optional) Run the following command to update the capacity of the SCSI data disk:

1. Run the following command to update the 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
```

In this example, run the following command:

```
echo 1 > /sys/class/scsi_device/2:0:0:0/device/rescan &
```

2. After the disk capacity is updated, run the following command to view the disk partition information again:

```
fdisk -l
```

If the additional space is included, go to [Step 3](#).

Step 3 Run the following command to enter fdisk:

```
fdisk Disk
```

In this example, run the following command:

```
fdisk /dev/sda
```

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# fdisk /dev/sda
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 4 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
Select (default p):
```

There are two types of disk partitions:

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

 NOTE

If MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

Step 5 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):
```

Partition number indicates the serial number of the primary partition. Because partition number 1 has been used, the value ranges from **2** to **4**.

Step 6 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 (104857600-209715199, default 104857600):
```

First sector indicates the start sector. The value ranges from **104857600** to **209715199**, and the default value is **104857600**.

Step 7 Enter the new partition's start sector and press **Enter**. In this example, the default start sector is used.

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 (104857600-209715199, default 104857600):
Using default value 104857600
Last sector, +sectors or +size{K,M,G} (104857600-209715199, default 209715199):
```

Last sector indicates the end sector. The value ranges from **104857600** to **209715199**, and the default value is **209715199**.

Step 8 Enter the new partition's end sector and press **Enter**. In this example, the default end sector is used.

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} (104857600-209715199, default 209715199):
Using default value 209715199
```

```
Partition 2 of type Linux and of size 50 GiB is set
```

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

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

Information similar to the following is displayed:

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

```
Disk /dev/sda: 107.4 GiB, 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: 0x915ffe6a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1		2048	104857599	52427776	83	Linux
/dev/sda2		104857600	209715199	52428800	83	Linux

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

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.
```

```
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.
```

NOTE

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

Step 11 Run the following command to synchronize the new partition table to the OS:

partprobe

Step 12 Run the following command to set the file system format for the new partition:

mkfs -t *File system Disk partition*

- Sample command of the ext* file system:

mkfs -t ext4 /dev/sda2

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# mkfs -t ext4 /dev/sda2
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
3276800 inodes, 13107200 blocks
655360 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2162163712
400 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
```

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

- Sample command of the xfs file system:

mkfs -t xfs /dev/sda2

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# mkfs -t xfs /dev/sda2
meta-data=/dev/sda2          isize=512    agcount=4, agsize=3276800 blks
         =                  sectsz=512   attr=2, projid32bit=1
         =                  crc=1        finobt=0, sparse=0
data      =                  bsize=4096   blocks=13107200, imaxpct=25
         =                  sunit=0      swidth=0 blks
naming    =version2          bsize=4096   ascii-ci=0 ftype=1
log       =internal log      bsize=4096   blocks=6400, version=2
         =                  sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
```

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 13 (Optional) Run the following command to create a mount point:

Perform this step if you want to mount the partition on a new mount point.

mkdir Mount point

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

mkdir /mnt/test

Step 14 Run the following command to mount the new partition:

mount Disk partition Mount point

In this example, run the following command to mount the new partition **/dev/sda2** on **/mnt/test**:

mount /dev/sda2 /mnt/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 15 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# 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/sda1  ext4   53G  55M  50G  1% /mnt/sdc
/dev/sda2  ext4   53G  55M  50G  1% /mnt/test
```

NOTE

If the server is restarted, the mounting will become invalid. You can modify the `/etc/fstab` file to configure automount at startup. For details, see [Configuring Automatic Mounting at System Start](#).

----End

Extending an Existing MBR Partition

NOTICE

If the additional space is allocated to an existing partition, data on the disk will not be cleared but you must use **umount** to unmount the existing partition. In this case, services will be affected.

Originally, SCSI data disk `/dev/sda` has 100 GiB and two partitions (`/dev/sda1` and `/dev/sda2`), and then 50 GiB is added to the disk. The following procedure shows you how to add this 50 GiB to the existing MBR partition `/dev/sda2`.

During an expansion, the additional space is added to the end of the disk. Therefore, if the disk has multiple partitions, the additional space can only be allocated to the partition at the disk end.

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

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# fdisk -l

Disk /dev/vda: 42.9 GiB, 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/sda: 161.1 GiB, 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: dos
Disk identifier: 0x915ffe6a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1          2048    104857599     52427776   83  Linux
/dev/sda2    104857600    209715199     52428800   83  Linux
```

In the command output, take note of the partition's start and end sectors. In this example, `/dev/sda2`'s start sector is **104857600**, and its end sector is **209715199**.

View the `/dev/sda` capacity and check whether the additional space is included.

- If the additional space is not included, refresh the capacity according to [Step 2](#).

- If the additional space is included, take note of the start and end sectors of the target partition and then go to [Step 3](#). These values will be used in the subsequent operations.

Step 2 (Optional) Run the following command to update the capacity of the SCSI data disk:

1. Run the following command to update the 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
```

In this example, run the following command:

```
echo 1 > /sys/class/scsi_device/2:0:0:0/device/rescan &
```

2. After the disk capacity is updated, run the following command to view the disk partition information again:

```
fdisk -l
```

If the additional space is included, take note of the start and end sectors of the target partition and then go to [Step 3](#). These values will be used in the subsequent operations.

Step 3 Run the following command to unmount the partition:

```
umount Disk partition
```

In this example, run the following command:

```
umount /dev/sda2
```

Step 4 Run the following command to enter fdisk:

```
fdisk Disk
```

In this example, run the following command:

```
fdisk /dev/sda
```

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# fdisk /dev/sda
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 5 Run the following command to delete the partition to be extended:

1. Enter **d** and press **Enter** to delete the partition.

Information similar to the following is displayed:

```
Command (m for help): d
Partition number (1,2, default 2):
```

2. Enter the partition number and press **Enter** to delete the partition. In this example, enter **2**.

Information similar to the following is displayed:

```
Partition number (1,2, default 2): 2
Partition 2 is deleted
```

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

 NOTE

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

- Step 6** 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
```

```
Select (default p):
```

There are two types of disk partitions:

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

 NOTE

If MBR is used, a maximum of four primary partitions, or three primary partitions plus one extended partition can be created. The extended partition must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

- 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:

```
Select (default p): p
```

```
Partition number (2-4, default 2):
```

Partition number indicates the serial number of the primary partition.

- Step 8** Ensure that entered partition number is the same as the partition had before. In this example, partition number **2** is used. Therefore, enter **2** and press **Enter**.

Information similar to the following is displayed:

```
Partition number (2-4, default 2): 2
```

```
First sector (104857600-314572799, default 104857600):
```

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 **104857600** is recorded in [Step 1](#) or [Step 2](#). Therefore, enter **104857600** and press **Enter**.

Information similar to the following is displayed:

```
First sector (104857600-314572799, default 104857600):
Using default value 104857600
Last sector, +sectors or +size{K,M,G} (104857600-314572799, default 314572799):
```

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

- Step 10** Ensure that the entered end sector is greater than or equal to the end sector recorded in [Step 1](#) or [Step 2](#). In this example, the recorded end sector is **209715199**, and the default end sector is used. Therefore, enter **314572799** and press **Enter**.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (104857600-314572799, default 314572799):
Using default value 314572799
Partition 2 of type Linux and of size 100 GiB is set

Command (m for help):
```

The partition is created.

- Step 11** Enter **p** and press **Enter** to print the partition details.

Information similar to the following is displayed:

```
Command (m for help): p

Disk /dev/sda: 161.1 GiB, 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: dos
Disk identifier: 0x915ffe6a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1             2048     104857599     52427776    83  Linux
/dev/sda2    104857600     314572799     104857600    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.

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.
```

NOTE

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

- Step 13** Run the following command to synchronize the new partition table to the OS:
- partprobe**

- Step 14** Perform the following operations based on the file system of the disk:

- For the **ext*** file system

- a. Run the following command to check the correctness of the file system on the partition:

e2fsck -f *Disk partition*

In this example, run the following command:

e2fsck -f /dev/sda2

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# e2fsck -f /dev/sda2
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/sda2: 11/3276800 files (0.0% non-contiguous), 251790/13107200 blocks
```

- b. Run the following command to extend the file system of the partition:

resize2fs *Disk partition*

In this example, run the following command:

resize2fs /dev/sda2

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# resize2fs /dev/sda2
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/sda2 to 26214400 (4k) blocks.
The filesystem on /dev/sda2 is now 26214400 blocks long.
```

- c. (Optional) Run the following command to create a mount point:
Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

- d. Run the following command to mount the partition:

mount *Disk partition Mount point*

In this example, run the following command to mount partition **/dev/sda2** on **/mnt/test**:

mount /dev/sda2 /mnt/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.

- For the **xfs** file system

- a. (Optional) Run the following command to create a mount point:
Perform this step if you want to mount the partition on a new mount point.

mkdir *Mount point*

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

mkdir /mnt/test

- b. Run the following command to mount the partition:

mount *Disk partition Mount point*

In this example, run the following command to mount partition **/dev/sda2** on **/mnt/test**:

mount /dev/sda2 /mnt/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.

- c. Run the following command to extend the file system of the partition:

sudo xfs_growfs *Disk partition*

In this example, run the following command:

sudo xfs_growfs /dev/sda2

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# sudo xfs_growfs /dev/sda2
meta-data=/dev/sda2          isize=512    agcount=4, agsize=3276800 blks
=                               sectsz=512   attr=2,    projid32bit=1
=                               crc=1      finobt=0,   spinodes=0
data     =                       bsize=4096   blocks=13107200, imaxpct=25
=                               sunit=0      swidth=0 blks
naming   =version2               bsize=4096   ascii-ci=0  ftype=1
log      =internal              bsize=4096   blocks=6400, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0,   rtextents=0
data blocks changed from 13107200 to 26214400df .
```

Step 15 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-scsi ~]# 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/sda1 ext4 53G 55M 50G 1% /mnt/sdc
/dev/sda2 ext4 106G 63M 101G 1% /mnt/test
```

----End

Configuring Automatic Mounting at System Start

The **fstab** file controls what disks are automatically mounted at server startup. You can configure the **fstab** file of a server that has data. This operation will not affect the existing data.

The following example uses UUIDs to identify disks in the **fstab** file. You are advised not to use device names (like **/dev/vdb1**) to identify disks in the file

because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a server stop or start. This can even prevent your server from booting up.

 NOTE

UUIDs are the unique character strings for identifying partitions in Linux.

Step 1 Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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"
```

Carefully record the UUID, as you will need it for the following step.

Step 2 Open the **fstab** file using the vi editor.

vi /etc/fstab

Step 3 Press **i** to enter 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
```

The preceding information is used for reference only. The line starting with **UUID** is the information added. Edit this line from left to right to match the following format:

- **UUID:** The UUID obtained in [Step 1](#).
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.
 - **0:** not use fsck.
 - If the mount point is the root partition (**/**), this parameter must be set to **1**.

If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

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

The system saves the configurations and exits the vi editor.

Step 6 Verify that the disk is auto-mounted at startup.

1. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

2. Reload all the content in the **/etc/fstab** file.

mount -a

3. 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, automatic mounting has been configured:

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

----End

4 Detaching an EVS Disk

4.1 Detaching a System Disk

Scenarios

If the file system on your system disk is damaged and your server cannot be started, you can detach the system disk and attach it to another server as a data disk. After the file system is fixed, you can re-attach the disk to the original server as the system disk.

If you no longer need a system disk or want to replace it with a new one, you can detach the system disk.

A system disk can only be detached offline, which means that its server must be in the **Stopped** state before the system disk is detached. To detach a system disk from a running server, you must first stop the server and then detach the disk.

Constraints

- A system disk can only be detached offline, which means that its server must be in the **Stopped** state before you can detach it. To detach a system disk from a running server, you must first stop the server and then detach the disk.
- After a system disk is detached, some operations cannot be performed on the original server and the system disk. The restricted operations are as follows:
 - Server: starting the server, remote login, resetting the password, changing server billing mode, changing server specifications, changing the OS, reinstalling the OS, creating images, creating backups, adding disks, changing the security group, and changing the VPC

Procedure

Step 1 Log in to the management console.

Step 2 Under **Compute**, 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 had succeeded, the detached system disk is no longer displayed under the **Disks** tab.

----End

Related Operations

For more detachment FAQs, see [Detachment](#).

4.2 Detaching a Data Disk

Scenarios

If you want to use a data disk on another server in the same region and AZ, you can detach the data disk and then attach it to that server.

If a data disk is no longer required, you can detach it and then delete it.

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

- ECS

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

- BMS

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.

NOTE

For an attached data disk, 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.

Precautions

Data may be lost after you detach an encrypted disk. For more information, see [If I Detach a Disk, Will I Lose the Data on My Disk?](#).

Prerequisites

- Before detaching an EVS disk from a running Windows ECS, ensure that no programs are reading data from or writing data to the disk. Otherwise, data will be lost.
- Before detaching an EVS 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 programs are reading data from or writing data to the disk. Otherwise, detaching the disk will fail.

Detaching a Non-shared Disk

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Choose a way to detach the disk by determining whether you want to check server information first.

- If yes, 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 server 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 no, 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 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Choose a way to detach the disk by determining whether you want to check server information first.

- If yes, 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**.
Shared EVS 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 no, 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 server.
Shared EVS disks support batch detachment so that you can select multiple servers at a time.
 - c. Click **OK** to detach the disk.

The disk list page 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

5 Deleting EVS Disks

Scenarios

If an EVS disk is no longer used, you can release the virtual resources by deleting it.

- The disk status is **Available**, **Error**, **Expansion failed**, **Restoration failed**, or **Rollback failed**.
- The disk is not locked by any service.
- The shared disk has been detached from all its servers.

NOTICE

- When you delete a disk, all the disk data including the snapshots created for this disk will be deleted.
 - A deleted disk cannot be recovered.
-

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 In the disk list, locate the row that contains the target disk, click **More** in the **Operation** column, and choose **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

6 Permissions Management

6.1 Creating a User and Granting EVS Permissions

You can use IAM for fine-grained permissions control for your EVS resources. With IAM, you can:

- Create IAM users for workforce based on your enterprise's organizational structure. Each IAM user has their own identity credentials for accessing EVS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or cloud service to perform efficient O&M on your EVS resources.

If your account does not require individual IAM users, you may skip over this section.

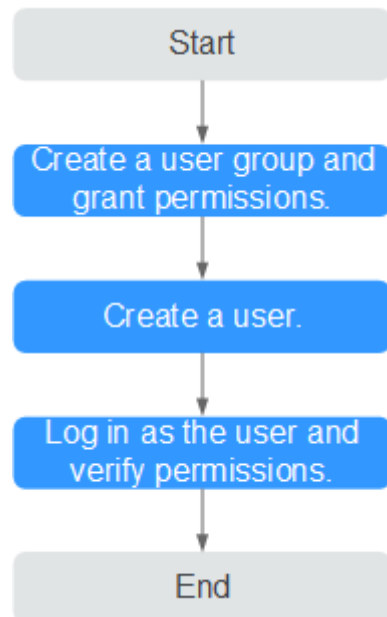
This section describes the procedure for granting permissions (see [Figure 6-1](#)).

Prerequisites

Before granting permissions to user groups, learn about system-defined permissions in section "Permissions Management" in the *Elastic Volume Service User Guide* for EVS.

Process Flow

Figure 6-1 Process for granting EVS permissions



1. On the IAM console, create a user group and grant it permissions (**EVS ReadOnlyAccess** as an example).
2. Create an IAM user and add it to the created user group.
Create a user on the IAM console and add the user to the group created in **1**.
3. Log in as the IAM user and verify permissions.
In the authorized region, perform the following operations:
 - Choose **Service List > Elastic Volume Service**. Then click **Create Disk** on the EVS console. If a message appears indicating that you have insufficient permissions to perform the operation, the **EVS ReadOnlyAccess** policy is in effect.
 - Choose another service from **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **EVS ReadOnlyAccess** policy is in effect.

6.2 EVS Custom Policies

You can create custom policies to supplement the system-defined policies of EVS. For the actions supported for custom policies, see section "Permissions Policies and Supported Actions" in the *Elastic Volume Service API Reference*.

You can create custom policies in either of the following ways:

- Visual editor: Select cloud services, actions, resources, and request conditions. This does not require knowledge of policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For operation details, see section "Creating a Custom Policy" in the *Identity and Access Management User Guide*. The following section contains examples of common EVS custom policies.

Example Custom Policies

- Example 1: Allowing users to create disks.

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Action": [
        "evs:volumes:list",
        "evs:volumes:get",
        "evs:quotas:get",
        "evs:volumeTags:list",
        "evs:types:get",
        "evs:volumes:create",
        "ecs:cloudServerFlavors:get",
        "ecs:cloudServers:list",
        "bss:balance:view",
        "bss:order:pay",
        "bss:order:update"
      ],
      "Effect": "Allow"
    }
  ]
}
```

- Example 2: Denying disk deletion

A policy with only "Deny" permissions must be used in conjunction with other policies to take effect. If the permissions assigned to a user contain both "Allow" and "Deny", the "Deny" permissions take precedence over the "Allow" permissions.

The following method can be used if you need to assign permissions of the **EVS FullAccess** policy to a user but you want to prevent the user from deleting EVS disks. Create a custom policy for denying disk deletion, and attach both policies to the group to which the user belongs. Then, the user can perform all operations on disks except deleting disks. The following is an example of a deny policy:

```
{
  "Version": "1.1",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "evs:volumes:delete"
      ]
    }
  ]
}
```

7 Managing Shared EVS Disks

How to Use Shared VBD and SCSI Disks

You can create shared VBD disks or shared SCSI disks. It is recommended that you attach a 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 EVS disks.
- Shared SCSI disks: Such 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 disks are 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 an EVS 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 EVS disk cannot differentiate multiple ECSs on the same physical host. For that reason, if multiple ECSs that use the same shared EVS disk are running on the same physical host, SCSI reservations

will not work properly. So you are advised to use SCSI reservations only on ECSs that are in the same ECS group, thus having a working anti-affinity policy.

Constraints on Shared Disks

- A shared disk can be attached to a maximum of 16 servers.
- The sharing attribute of a disk cannot be changed after the disk is created.
- Shared disks can only be used as data disks. The sharing function is not supported for system disks.
- A shared file system or cluster management system must be set up before you can properly use a shared disk. If you simply attach a shared disk to multiple servers, the sharing function will not work and data may be overwritten.
- When a shared disk is attached to multiple servers, the total performance of the disk on all servers cannot exceed the maximum allowed on a single disk.

Attaching a Shared EVS Disk

A non-shared EVS disk can only be attached to one server, whereas a shared EVS disk can be attached to up to 16 servers.

For details, see [Attaching a Shared Disk](#).

Deleting a Shared EVS Disk

Because a shared EVS disk can be attached to multiple servers, ensure that the shared EVS disk is detached from all the servers before deletion.

For details, see [Deleting EVS Disks](#).

Expanding a Shared EVS Disk

Shared EVS disks must be expanded when they are in the **Available** state. For details, see [Expanding Capacity for an In-use EVS Disk](#).

Related Operations

For more disk sharing FAQs, see [Sharing](#).

8 Managing EVS Backups

Scenarios

EVS disk backups are created using the CBR service. For details, see **Creating a Cloud Disk Backup** in the *Cloud Backup and Recovery User Guide*.

You can configure a backup policy for disks. With backup policies configured, data on EVS disks can be periodically backed up to improve data security.

Constraints

- Backups can be created only when the disks are in the **Available** or **In-use** state.

Creating a Disk Backup Vault and Applying a Backup Policy

Step 1 Log in to the CBR console.

1. Log in to the management console.
2. Choose **Storage > Cloud Backup and Recovery > Cloud Disk Backups**.

Step 2 In the upper right corner, click **Create Disk Backup Vault**.

Step 3 (Optional) In the disk list, select the disks you want to back up. After disks are selected, they are added to the list of selected disks.

NOTE

- Only **Available** and **In-use** disks can be selected.
- You can also associate disks with the vault you are creating later if you skip this step.

Step 4 Specify a vault capacity ranging from the total sizes of disks to 10,485,760 GiB.

Step 5 Configure auto backup.

- If you select **Configure**, you must then select an existing backup policy or create a new one. After the vault is created, the system applies this backup policy to the vault, and all disks associated with this vault will be automatically backed up based on this policy.
- If you select **Skip**, disks associated with this vault are not automatically backed up.

Step 6 Specify a name for the vault.

The name can contain 1 to 64 characters including digits, letters, underscores (_), and hyphens (-), for example, **vault-612c**.

 NOTE

You can use the default name, which is in the format of **vault_XXXX**.

Step 7 Click **Next**.

Step 8 Complete the creation as prompted.

Step 9 Go back to the disk backup page. The vault you created is displayed in the list.

You can associate disks to the new vault or create backups for the disks. For details, see section "Vault Management" in the *Cloud Backup and Recovery User Guide*.

-----End

9 Managing EVS Snapshots

9.1 Snapshot Overview

What Is EVS Snapshot?

An EVS snapshot is a complete copy or image of the disk data at a specific point in time. Snapshots can be used as a disaster recovery (DR) approach, and you can use snapshots to fully restore data to the time when the snapshot was taken. You can create snapshots for disks on the console or via the API.

EVS snapshots are sometimes referred to as snapshots in this document.

You can create snapshots to rapidly save the disk data at specified time points. In addition, you can use snapshots to create new disks so that the created disks will contain the snapshot data in the beginning.

Application Scenarios

The snapshot function helps address your following needs:

- **Routine data backup**
You can create snapshots for disks on a timely basis and use snapshots to recover your data in case that data loss or data inconsistency occurred due to unintended operations, viruses, or attacks.
- **Rapid data restoration**
You can create a snapshot or multiple snapshots before an application software upgrade or a service data migration. If an exception occurs during the upgrade or migration, service data can be rapidly restored to the time point when the snapshot was created.
For example, a fault occurred on system disk A of server A, and therefore server A cannot be started. As system disk A is already faulty, data on system disk A cannot be restored by rolling back snapshots. But, you can create disk B using an existing snapshot of system disk A and attach disk B to a properly running server, for example server B. In this case, server B obtains the data of system disk A from disk B.

 NOTE

When rolling back data from snapshots, data can only be rolled back to the original disk, and a rollback to a different disk is not possible.

- **Multi-service quick deployment**

You can use a snapshot to create multiple disks containing the same initial data, and these disks can be used as data resources for various services, for example data mining, report query, and development and testing. This method protects the initial data and creates disks rapidly, meeting diverse service requirements.

Operation Overview

You can create snapshots according to [Creating a Snapshot](#) to rapidly save the disk data at specified points in time.

If a data loss happened, you can roll back the disk data to the time when the snapshot was created based on [Rolling Back Data from a Snapshot](#). In addition, you can create a new disk from the snapshot so that the disk will contain the snapshot data in the beginning. For details, see [Creating an EVS Disk from a Snapshot](#).

When a snapshot is no longer needed, delete it according to [Deleting a Snapshot](#) to release the virtual resources.

9.2 Creating a Snapshot

Scenarios

You can create an EVS snapshot on the management console to save the EVS disk data at a specific time point.

 NOTE

Creating snapshots does not affect the performance of the disk.

Constraints

- A maximum of 7 snapshots can be created for one disk.
- Snapshots can be created for both system disks and data disks.
- Snapshots can be created only for available or in-use disks.

Creating a Snapshot on the Disks Page

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 In the disk list, locate the row that contains the target disk, click **Create Snapshot** in the **Operation** column.

Configure the basic settings for the snapshot according to [Table 9-1](#).

Table 9-1 Snapshot parameter

Parameter	Description	Example Value
Snapshot Name	Mandatory The name can contain a maximum of 64 characters.	snapshot-01

Step 4 Click **Create Now**.

Step 5 On the displayed **Details** page, view the details of the snapshot.

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

Step 6 Go back to the **Snapshots** page to view the snapshot creation information.

After the snapshot status changes to **Available**, the snapshot has been created.

-----End

Creating a Snapshot on the Snapshots Page

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 In the navigation pane on the left, choose **Elastic Volume Service > Snapshots**.

On the **Snapshots** page, click **Create Snapshot**.

Configure the basic settings for the snapshot according to [Table 9-2](#).

Table 9-2 Snapshot parameters

Parameter	Description	Example Value
Region	Mandatory After you select a region, disks in the selected region will be displayed for you to choose from.	-
Snapshot Name	Mandatory The name can contain a maximum of 64 characters.	snapshot-01
Select Disk	Mandatory Select a disk based on which the snapshot will be created.	volume-01

Step 4 Click **Create Now**.

Step 5 On the displayed **Details** page, view the details of the snapshot.

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

Step 6 Go back to the **Snapshots** page to view the snapshot creation information.

After the snapshot status changes to **Available**, the snapshot has been created.

----End

9.3 Deleting a Snapshot

Scenarios

If a snapshot is no longer needed, you can delete it to release the virtual resources.

Constraints

- The snapshot status must be **Available** or **Error**.
- If a disk is deleted, all the snapshots created for this disk will also be deleted.
- If a snapshot is deleted, disks rolled back and created from this snapshot are not affected.
- If you have reinstalled or changed the server OS, snapshots of the system disk are automatically deleted. Snapshots of the data disks can be used as usual.
- A snapshot whose name starts with **autobk_snapshot_vbs_**, **manualbk_snapshot_vbs_**, **autobk_snapshot_csbs_**, or **manualbk_snapshot_csbs_** is automatically generated during backup. You can only view details of such snapshots and cannot delete them.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

Step 3 In the navigation pane on the left, choose **Elastic Volume Service > Snapshots**.

The snapshot list page is displayed.

Step 4 In the snapshot list, locate the row that contains the target snapshot and click **Delete** in the **Operation** column.

Step 5 (Optional) If multiple snapshots are to be deleted, select ☐ in front of each snapshot and click **Delete** in the upper area of the list.

Step 6 In the displayed dialog box, confirm the information and click **Yes**.

If the snapshot disappears from the snapshot list, the snapshot is deleted successfully.

----End

9.4 Viewing Details of a Snapshot

Scenarios

This section describes how to view the details of a snapshot.

Procedure

- Step 1** Log in to the management console.
- Step 2** Under **Storage**, click **Elastic Volume Service**.
- Step 3** In the navigation pane on the left, choose **Elastic Volume Service > Snapshots**.
The snapshot list page is displayed.
- Step 4** Locate the row that contains the target snapshot and click  in front of the snapshot name.
On the details page, view the snapshot information, such as the snapshot size.
----End

9.5 Rolling Back Data from a Snapshot

Scenarios

If data on an EVS disk is incorrect or damaged, you can roll back data from a snapshot to the source disk.

Constraints

- A snapshot can be rolled back only to its source disk. Rollback to another disk is not possible.
- A snapshot can be rolled back only when the snapshot status is **Available** and its source disk status is **Available** (not attached to any server) or **Rollback failed**. If the source disk is attached, detach the disk first.
- A snapshot whose name starts with **autobk_snapshot_vbs_**, **manualbk_snapshot_vbs_**, **autobk_snapshot_csbs_**, or **manualbk_snapshot_csbs_** is automatically generated during backup. Such a snapshot can only be viewed. It cannot be used to roll back the disk data.
- If you first roll back the snapshot to the original disk, you cannot use the snapshot to create a new disk then.

Procedure

- Step 1** Log in to the management console.
- Step 2** Under **Storage**, click **Elastic Volume Service**.
- Step 3** In the navigation pane on the left, choose **Elastic Volume Service > Snapshots**.

The snapshot list page is displayed.

Step 4 In the snapshot list, locate the row that contains the target snapshot and click **Roll Back Disk** in the **Operation** column.

Step 5 In the displayed dialog box, click **Yes**.

The snapshot list is displayed. After the snapshot status changes from **Rolling back** to **Available**, the data rollback is successful.

----End

9.6 Creating an EVS Disk from a Snapshot

Scenarios

This section describes how to create an EVS disk on the **Snapshots** page. Besides, you can also create an EVS disk from a snapshot by specifying the **Create from snapshot** parameter on the disk creation page. For details, see [Create an EVS Disk](#).

Constraints

- The disk type, device type, and snapshot attributes of the new disk are the same as those of the snapshot's source disk.
- Batch disk creation is not possible, and the quantity parameter must be set to 1.
- The AZ of the disk is the same as that of the snapshot's source disk and cannot be changed.
- A snapshot whose name starts with **autobk_snapshot_vbs_**, **manualbk_snapshot_vbs_**, **autobk_snapshot_csbs_**, or **manualbk_snapshot_csbs_** is automatically generated during backup. Such a snapshot can only be viewed. It cannot be used to create new disks.
- If you first roll back the snapshot to the original disk, you cannot use the snapshot to create a new disk then.

Procedure

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

Step 3 In the navigation pane on the left, choose **Elastic Volume Service > Snapshots**.

The snapshot list page is displayed.

Step 4 In the snapshot list, locate the row that contains the target snapshot and click **Create Disk** in the **Operation** column.

Step 5 Configure the disk parameters. For details, see parameter descriptions and operations provided in [Create an EVS Disk](#).

 NOTE

A maximum of 128 disks can be created from a snapshot.

If you create a disk from a snapshot, the disk capacity must be greater than or equal to the snapshot size. In the condition that you do not specify the disk capacity, if the snapshot size is smaller than 10 GiB, the default capacity 10 GiB will be used as the disk capacity; if the snapshot size is greater than 10 GiB, the snapshot size will be used as the disk capacity.

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**

10 Managing EVS Transfers

Scenarios

EVS transfer allows you to transfer disks from one account to another. After a transfer succeeds, the ownership of the disk belongs to the target account only.

Users can use disk transfer via API only. For more information, see chapter "EVS Transfer" in the *Elastic Volume Service API Reference*.

Constraints

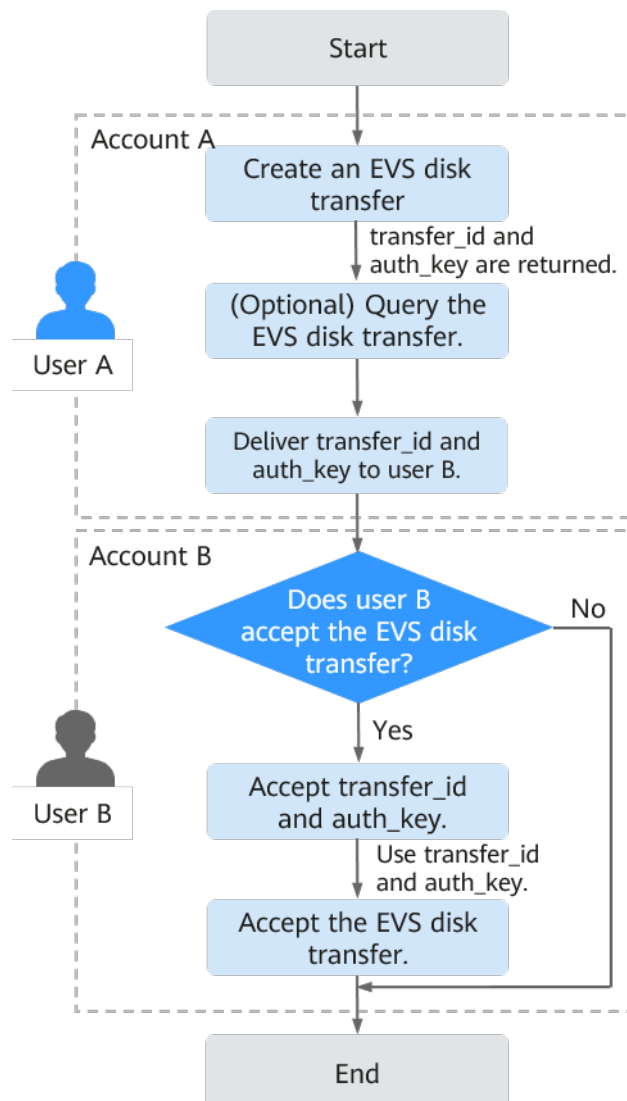
- EVS disks with backups and snapshots available cannot be transferred.
- EVS disks associated with backup policies cannot be transferred.
- EVS disks used as system disks cannot be transferred.

Procedure

The following example shows you how to transfer an EVS disk from account A to account B. User A belongs to account A, and user B belongs to account B. User A creates the transfer. User B accepts the transfer using the transfer ID (**transfer_id**) and authentication key (**auth_key**). After the transfer has been accepted, the transfer is complete. [Figure 10-1](#) shows the basic transfer process.

NOTE

- **transfer_id** specifies the disk transfer ID. Each EVS disk transfer has a transfer ID, and user B uses this ID to accept the disk transfer. The transfer ID expires after user B accepts the transfer.
- **auth_key** specifies the identity authentication key of the disk transfer. Each EVS disk transfer has an authentication key, and user B uses this key for authentication when accepting the disk transfer.

Figure 10-1 EVS disk transfer process

Step 1 User A creates the transfer. For details, see "Creating a Disk Transfer" in the *Elastic Volume Service API Reference*.

After the transfer is successfully created, **transfer_id** and **auth_key** are returned.

Step 2 (Optional) User A views the disk transfer. For details, see "Querying Details of a Disk Transfer" in the *Elastic Volume Service API Reference*. If multiple disk transfers have been created, user A can query all disk transfers. For details, see "Querying All Disk Transfers" or "Querying Details of All Disk Transfers" in the *Elastic Volume Service API Reference*.

Step 3 User A delivers the returned **transfer_id** and **auth_key** to user B.

Step 4 Check whether user B is going to accept the disk transfer.

- If yes, go to **Step 5**.
- If no, no further action is required.

User A can delete the unaccepted disk transfer. For details, see "Deleting a Disk Transfer" in the *Elastic Volume Service API Reference*.

Step 5 User B accepts **transfer_id** and **auth_key**.

Step 6 User B accepts the transfer through **transfer_id** and **auth_key**. For details, see "Accepting a Disk Transfer" in the *Elastic Volume Service API Reference*.

----End

11

Viewing EVS Monitoring Data

Description

This section describes monitored metrics reported by EVS to Cloud Eye as well as their namespaces and dimensions. You can use the console or APIs provided by Cloud Eye to query the metrics of the monitored objects and alarms generated for EVS.

Namespace

SYS.EVS

Metrics

Table 11-1 EVS metrics

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period
disk_device_read_bytes_rate	Disk Read Bandwidth	Number of bytes read from the monitored disk per second Unit: Bytes/s	≥ 0 bytes/s	EVS disk	5 minutes in average
disk_device_write_bytes_rate	Disk Write Bandwidth	Number of bytes written to the monitored disk per second Unit: Bytes/s	≥ 0 bytes/s	EVS disk	5 minutes in average
disk_device_read_requests_rate	Disk Read IOPS	Number of read requests sent to the monitored disk per second Unit: Requests/s	≥ 0 Requests/s	EVS disk	5 minutes in average

Metric ID	Metric Name	Description	Value Range	Monitored Object	Monitoring Period
disk_device_write_requests_rate	Disk Write IOPS	Number of write requests sent to the monitored disk per second Unit: Requests/s	≥ 0 Requests/s	EVS disk	5 minutes in average

Dimension

Key	Value
disk_name	Server ID-drive letter, for example, 6f3c6f91-4b24-4e1b-b7d1-a94ac1cb011d-vda (vda is the drive letter)

Viewing Monitoring Data

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 In the EVS disk list, click the name of the disk you want to view the monitoring data.

The disk details page is displayed.

Step 4 On the **Attachments** tab, locate the row that contains the server and click **View Monitoring Data** in the **Operation** column.

The monitoring graphs page is displayed.

Step 5 View the disk monitoring data by metric or monitored duration.

For more information about Cloud Eye, see the *Cloud Eye User Guide*.

----End

12 Managing Quotas

12.1 Querying EVS Resource Quotas

Scenarios

Quotas are enforced for service resources on the platform to prevent unforeseen spikes in resource usage. Quotas can limit the number or amount of resources available to users, such as the number of EVS disks, the capacity of EVS disks, and the number of EVS snapshots.

Users can perform the following operations to view the resource quota details.

Procedure

Step 1 Log in to the management console.

Step 2 In the upper right corner of the page, click  .
The **Service Quota** page is displayed.

Step 3 View the used and total quota of each type of resources on the displayed page.
If a quota cannot meet service requirements, apply for a higher quota.

----End

12.2 Increasing EVS Resource Quotas

Scenarios

If any resource quota no longer meets your service requirements, you can apply for a higher quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. If you need to adjust a quota, contact the operations administrator.

Before contacting the operations administrator, make sure that the following information has been obtained:

- Account name, which can be obtained by performing the following operations:
Log in to the management console using the cloud account, click the username in the upper right corner, select **My Credentials** from the drop-down list, and obtain the account name on the **My Credentials** page.
- Quota information, which includes service name, quota type, and required quota

13 FAQ

13.1 General

13.1.1 How Do I Start Using a Newly Disk?

A newly disk must be attached to a server and then initialized in the server OS before you can use it.

13.1.2 Can EVS Disks Be Used Directly for Storage?

No.

EVS disks must be attached to cloud servers before use. You cannot use EVS disks alone to store data.

13.1.3 Can EVS Disks Be Used Alone?

No.

EVS disks must be attached to servers before you can use them.

13.1.4 How Can I View My Disk Details?

To do so, perform the following operations:

Step 1 Log in to the management console.

Step 2 Under **Storage**, click **Elastic Volume Service**.

The disk list page is displayed.

Step 3 Locate the row that contains the target disk and view the disk specifications, attributes, and billing mode.

Step 4 (Optional) Click the disk name to view more information, such as the disk backup and snapshot information.

View more information on the **Summary** tab.

----End

13.1.5 Can I Change the AZ of My Disk?

No.

The AZ of a disk cannot be changed after you have created the disk. If you want to change the AZ, delete the disk and create a new one.

13.1.6 What Are the Differences Between System Disks and Data Disks?

- A system disk runs the server OS. It is like drive C in a PC.
When a server is created, a system disk is automatically created and attached. You cannot create a system disk separately. The maximum size of a system disk is 1,024 GiB.
- Data disks store the server data. They are like drive D, drive E, and drive F in a PC.
Data disks can be created during or after the server creation. If you create data disks during the server creation, the system will automatically attach the data disks to the server. If you create data disks after the server creation, you need to manually attach the data disks. The maximum size of a data disk is 32,768 GiB.

If one system disk already meets your business needs, you do not need to create data disks. As your business grows, you can create data disks when needed.

If the disk paths in your service systems cannot be changed or are difficult to change, you are advised to create data disks according to your system planning.

13.1.7 How Can I Download My EVS Disk Data to a Local PC?

EVS disk data cannot be directly saved to a local PC. It is recommended that you use a third-party tool, such as FTP, to download the data.

13.1.8 How Can I Export the Original Data After I Changed My Server OS from Windows to CentOS?

Solution:

1. Install the ntfsprogs software to enable Linux to access the NTFS file system.
yum install ntfsprogs
2. View the data disks previously attached to Windows.
parted -l
3. Mount the data disks.
mount -t ntfs-3g *Data disk path* *Mount point*

13.1.9 What Are the Differences Between MBR and GPT Partition Styles?

Table 13-1 lists the common disk partition styles. In Linux, different partition styles require different partitioning tools.

Table 13-1 Disk partition styles

Disk Partition Style	Maximum Disk Capacity Supported	Maximum Number of Partitions Supported	Linux Partitioning Tool
Master Boot Record (MBR)	2 TiB	4 primary partitions 3 primary partitions and 1 extended partition With MBR, you can create several primary partitions and one extended partition. The extended partition must be divided into logical partitions before use. For example, if 6 partitions need to be created, you can create them 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	You can use either of the following tools: - fdisk - parted
Guid Partition Table (GPT)	18 EiB 1 EiB = 1048576 TiB	Unlimited Disk partitions created using GPT are not categorized.	parted

NOTICE

The maximum disk size supported by MBR is 2 TiB, and that supported by GPT is 18 EiB. Because an EVS data disk currently supports up to 32 TiB, use GPT if your disk size is larger than 2 TiB.

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

13.2 Attachment

13.2.1 Why Can't I View the Attached Data Disk on the Server?

Troubleshooting

Table 13-2 Possible causes

OS	Possible Cause	Solution
Linux	- New data disks are not formatted and partitioned by default, and an unformatted disk will not be listed in the command output. You must manually initialize the disk. - If a data disk cannot be found after the server is restarted, automatic partition mounting at system start may not be configured.	Linux Data Disk
Windows	New data disks are not formatted and partitioned by default. Only formatted and partitioned drives show up in the resource manager. You must manually initialize the disk.	Windows Data Disk

Linux Data Disk

Symptom: A data disk has been attached to a Linux server on the management console, but the disk cannot be viewed on the server.

Run **df -TH** to view the disk information. CentOS 7.4 is used in this example. The normal command output is as follows:

```
[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
```

Unlike the normal command output, only system disk **/dev/vda1** is visible, but data disk **/dev/vdb1** is missing from the command output.

Cause Analysis:

- Cause 1:** New data disks are not formatted and partitioned by default, and an unformatted disk will not be listed in the command output. You must manually initialize the disk.

For details, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

- **Cause 2:** If a data disk cannot be found after the server is restarted, automatic partition mounting at system start may not be configured. Perform the following steps:

- a. Mount the data disk.

mount *Disk partition Mount point*

In this example, run the following command:

mount /dev/vdb1 /mnt/sdc

Perform the following steps to configure auto mount at system start:

- b. Query the partition UUID.

blkid *Disk partition*

In this example, the UUID of the **/dev/vdb1** partition is queried.

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.

- c. Open the **fstab** file using the vi editor.

vi /etc/fstab

Press **i** to enter editing mode.

- d. 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
```

In this example, the line starting with "UUID" is the information added. Edit this line to match the following format:

- **UUID:** The UUID obtained in **b**.
- **Mount point:** The directory on which the partition is mounted. You can query the mount point using **df -TH**.
- **Filesystem:** The file system format of the partition. You can query the file system format using **df -TH**.
- **Mount option:** The partition mount option. Usually, this parameter is set to **defaults**.
- **Dump:** The Linux dump backup option.
 - **0:** Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**.
 - **1:** Linux dump backup is used.
- **fsck:** The fsck option, which means whether to use fsck to check the disk during startup.
 - **0:** The fsck option is not used.
 - If the mount point is the root partition (**/**), this parameter must be set to **1**.

If this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** because the system checks the partitions in the ascending order of the values.

- e. Press **Esc**, enter **:wq**, and press **Enter**.

The system saves the configurations and exits the vi editor.

Verify that the disk is auto-mounted at startup.

- i. Unmount the partition.

umount *Disk partition*

In this example, run the following command:

umount /dev/vdb1

- ii. Reload all the content in the **/etc/fstab** file.

mount -a

- iii. Query the file system mounting information.

mount | grep Mount point

In this example, run the following command:

mount | grep /mnt/sdc

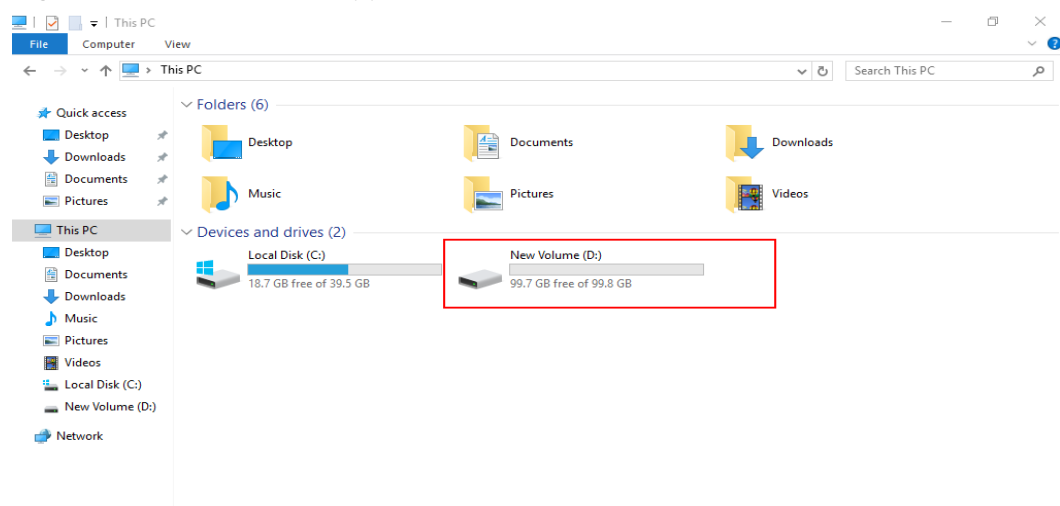
If information similar to the following is displayed, auto mount has taken effect:

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

Windows Data Disk

Symptom: A data disk has been attached to a Windows server on the management console, but the disk cannot be viewed on the server. For example, Volume (D:) was not shown in **This PC** of a Windows server running Windows Server 2012. Normally, Volume (D:) appears, as shown in [Figure 13-1](#).

Figure 13-1 Volume (D:) appears



Solution: New data disks are not formatted and partitioned by default. Only formatted and partitioned drives show up in **This PC**. You must manually initialize the disk before it can be viewed here.

For details, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

13.2.2 Can I Attach a Disk to Multiple Servers?

- A non-shared disk can only be attached to one server.
- A shared disk can be attached to up to 16 servers.

NOTE

- Shared disks are a type of EVS disks that can be attached to multiple servers.
- To use shared disks, you must set up a shared file system or cluster management system. If you directly attach a disk to multiple servers, the disk sharing attribute does not work and data may be overwritten.

13.2.3 Can I Attach a Disk to a Server in Another AZ?

No.

Disks and the servers you attach the disks to must be in the same AZ. The same is true for shared disks. Shared disks can only be attached to the servers in the same AZ.

13.2.4 How Can I Add a Data Disk to an Existing Server?

Data disks can be created during or after the server creation. If you create data disks during the server creation, the system will automatically attach the data disks to the server. If you create data disks after the server creation, you need to manually attach the data disks.

- On a Windows server:
 - If a data disk is created along with the server, you need to log in to the server and initialize the disk. The data disk will be visible after the initialization succeeds.
 - If no data disk is created along with the server, you need to create a data disk and attach it to the server. Then, you need to log in to the server and initialize the disk. The data disk will be visible after the initialization succeeds.
- On a Linux server:
 - If a data disk is created along with the server, you need to log in to the server and initialize the disk. The data disk will be visible after the initialization succeeds and the disk has been mounted via the **mount** command.
 - If no data disk is created along with the server, you need to create a data disk and attach it to the server. Then, you need to log in to the server and initialize the disk. The data disk will be visible after the initialization succeeds and the disk has been mounted via the **mount** command.

For details, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

13.2.5 Can I Attach Different Types of Disks to the Same Server?

Yes. Different types of EVS disks can be attached to the same server. You only need to make sure that these disks and the server are in the same AZ.

13.2.6 What Should I Do If a Linux EVS Disk Is Attached to a Windows Server?

You are not advised to attach a Linux EVS disk to a Windows server or attach a Windows EVS disk to a Linux server.

The disk information may fail to be displayed due to the inconsistent file systems. To solve this issue, you need to initialize and partition the disk again. Formatting the disk will destroy any data the disk has contained, so you should back up the data first.

13.2.7 How Do I Obtain My Disk Device Name in the ECS OS Using the Device Identifier Provided on the Console?

Scenarios

You find that the device name displayed in the ECS OS is different from that displayed on the management console and you cannot determine which disk name is correct. This section describes how to obtain the disk name used in an ECS OS according to the device identifier on the console.

Obtaining the Disk ID of an ECS on the Console

1. Log in to the management console.
2. Under **Computing**, choose **Elastic Cloud Server**.
3. Click the target ECS name in the ECS list.
The ECS details page is displayed.
4. Click the **Disks** tab and then click  to expand the disk information.
5. Check the device type and ID of the disk.

NOTE

If **Device Identifier** is not displayed on the page, stop the ECS and restart it.

– KVM ECS

- If **Device Type** is **VBD**, use a serial number or BDF to obtain the disk device name.

If you use a serial number (recommended) to obtain the disk name, see [Using a Serial Number to Obtain the Disk Name \(Windows\)](#) and [Using a Serial Number to Obtain a Disk Device Name \(Linux\)](#).

If you use a BDF to obtain the disk device name, see [Using a BDF to Obtain a Disk Device Name \(Linux\)](#). (BDF cannot be used to obtain the disk name of Windows ECSs.)

- If **Device Type** is **SCSI**, use a WWN to obtain the disk name. For details, see [Using a WWN to Obtain the Disk Name \(Windows\)](#) and [Using a WWN to Obtain a Disk Device Name \(Linux\)](#).

Using a Serial Number to Obtain the Disk Name (Windows)

If a serial number is displayed on the console, use either of the following methods to obtain the disk name.

cmd

1. Start **cmd** in a Windows OS as an administrator and run either of the following commands:

wmic diskdrive get serialnumber

wmic path win32_physicalmedia get SerialNumber

wmic path Win32_DiskDrive get SerialNumber

NOTE

A serial number is the first 20 digits of a disk UUID.

For example, if the serial number of a VBD disk on the console is 97c876c0-54b3-460a-b, run either of the following commands to obtain the serial number of the disk on the ECS OS:

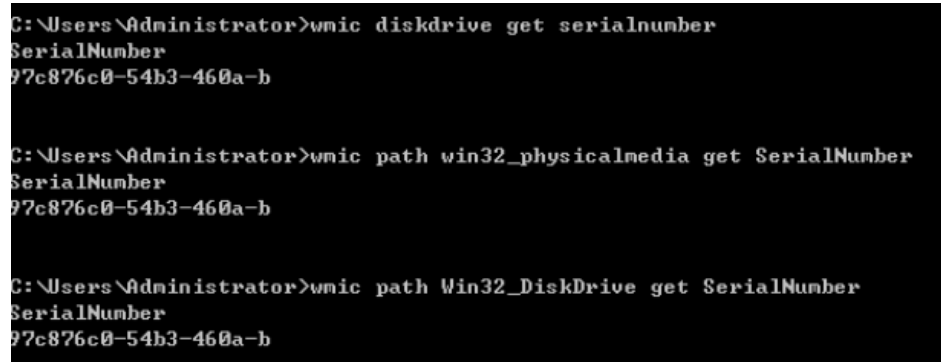
wmic diskdrive get serialnumber

wmic path win32_physicalmedia get SerialNumber

wmic path Win32_DiskDrive get SerialNumber

Information similar to the following is displayed:

Figure 13-2 Obtaining the disk serial number



```
C:\Users\Administrator>wmic diskdrive get serialnumber
SerialNumber
97c876c0-54b3-460a-b

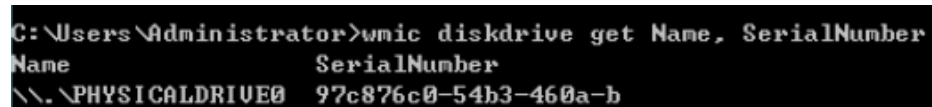
C:\Users\Administrator>wmic path win32_physicalmedia get SerialNumber
SerialNumber
97c876c0-54b3-460a-b

C:\Users\Administrator>wmic path Win32_DiskDrive get SerialNumber
SerialNumber
97c876c0-54b3-460a-b
```

2. Run the following command to check the disk corresponding to the serial number:

wmic diskdrive get Name, SerialNumber

Figure 13-3 Checking the disk corresponding to the serial number

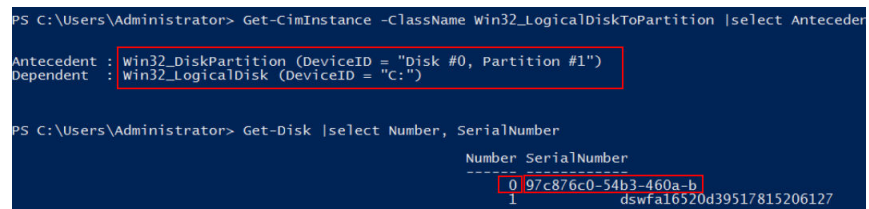


```
C:\Users\Administrator>wmic diskdrive get Name, SerialNumber
Name                      SerialNumber
\\.\PHYSICALDRIVE0       97c876c0-54b3-460a-b
```

PowerShell

1. Start PowerShell as an administrator in a Windows OS.
2. Run the following command to check the disk on which the logical disk is created:
 - Windows Server 2012 or later
 - i. Run the following command to check the disk on which the logical disk is created:
Get-CimInstance -ClassName Win32_LogicalDiskToPartition | select Antecedent, Dependent | fl
As shown in [Figure 13-4](#), the disk is **Disk 0**.
 - ii. Run the following command to view the mapping between the serial number and the disk:
Get-Disk |select Number, SerialNumber
As shown in [Figure 13-4](#), the disk is **Disk 0**.

Figure 13-4 Viewing the disk on which the logical disk is created



```
PS C:\Users\Administrator> Get-CimInstance -ClassName Win32_LogicalDiskToPartition |select Antecedent, Dependent | fl
Antecedent : Win32_DiskPartition (DeviceID = "Disk #0, Partition #1")
Dependent  : Win32_LogicalDisk (DeviceID = "C:")

PS C:\Users\Administrator> Get-Disk |select Number, SerialNumber
Number SerialNumber
-----
0 97c876c0-54b3-460a-b1d5wfa16520d39517815206127
```

- Versions earlier than Windows 2012
 - i. Run the following command to check the disk on which the logical disk is created:
Get-WmiObject -Class Win32_PhysicalMedia |select Tag, Serialnumber
 - ii. Run the following command to view the mapping between the serial number and the disk:
Get-WmiObject -Class Win32_LogicalDiskToPartition |select Antecedent, Dependent | fl

Using a Serial Number to Obtain a Disk Device Name (Linux)

If a serial number is displayed on the console, run either of the following commands to obtain the device name.

```
# udevadm info --query=all --name=/dev/xxx | grep ID_SERIAL
```

```
# ll /dev/disk/by-id/*
```

NOTE

A serial number is the first 20 digits of a disk UUID.

For example, if the serial number of the VBD disk is 62f0d06b-808d-480d-8, run either of the following commands:

```
# udevadm info --query=all --name=/dev/vdb | grep ID_SERIAL
```

ll /dev/disk/by-id/*

The following information is displayed:

```
[root@ecs-ab63 ~]# udevadm info --query=all --name=/dev/vdb | grep ID_SERIAL
E: ID_SERIAL=62f0d06b-808d-480d-8
[root@ecs-ab63 ~]# ll /dev/disk/by-id/*
lrwxrwxrwx 1 root root 9 Dec 30 15:56 /dev/disk/by-id/virtio-128d5bfd-f215-487f-9 -> ../vda
lrwxrwxrwx 1 root root 10 Dec 30 15:56 /dev/disk/by-id/virtio-128d5bfd-f215-487f-9-part1 -> ../vda1
lrwxrwxrwx 1 root root 9 Dec 30 15:56 /dev/disk/by-id/virtio-62f0d06b-808d-480d-8 -> ../vdb
```

/dev/vdb is the disk device name.

Using a BDF to Obtain a Disk Device Name (Linux)

1. Run the following command to use a BDF to obtain the device name:

```
ll /sys/bus/pci/devices/BDF disk ID/virtio*/block
```

For example, if the BDF disk ID of the VBD disk is 0000:02:02.0, run the following command to obtain the device name:

```
ll /sys/bus/pci/devices/0000:02:02.0/virtio*/block
```

The following information is displayed:

```
[root@ecs-ab63 ~]# ll /sys/bus/pci/devices/0000:02:02.0/virtio*/block
total 0
drwxr-xr-x 8 root root 0 Dec 30 15:56 vdb
```

/dev/vdb is the disk device name.

Using a WWN to Obtain the Disk Name (Windows)

1. Obtain the device identifier on the console by referring to [Obtaining the Disk ID of an ECS on the Console](#).
2. Manually convert the WWN.

For example, the obtained WWN (device identifier) is 68886030000**3252ffa**16520d39517815.

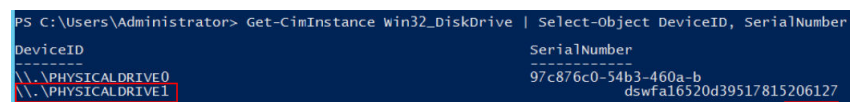
- a. Obtain the 21st to 17th digits that are counted backwards (**3252f**).
- b. Convert a hexadecimal (**3252f**) to a decimal (**206127**).

3. Start PowerShell as an administrator in a Windows OS.
4. Run the following command:

```
Get-CimInstance Win32_DiskDrive | Select-Object DeviceID, SerialNumber
```

5. In the command output, the disk whose serial number ends with **206127** is the disk corresponding to the WWN.

Figure 13-5 Disk with the serial number ending with **206127**



```
PS C:\Users\Administrator> Get-CimInstance Win32_DiskDrive | Select-Object DeviceID, SerialNumber
DeviceID                               SerialNumber
-----
\\.\PHYSICALDRIVE0                     97c876c0-54b3-460a-bdswfa16520d39517815206127
\\.\PHYSICALDRIVE1                     dswfa16520d39517815206127
```

Using a WWN to Obtain a Disk Device Name (Linux)

1. Log in to the ECS as user **root**.
2. Run the following command to view the disk device name:

```
ll /dev/disk/by-id |grep WWN|grep scsi-3
```

For example, if the WWN obtained on the console is 6888603000008b32fa16688d09368506, run the following command:

```
ll /dev/disk/by-id |grep 6888603000008b32fa16688d09368506|grep scsi-3
```

The following information is displayed:

```
[root@host-192-168-133-148 block]# ll /dev/disk/by-id/ |grep 6888603000008b32fa16688d09368506 |  
grep scsi-3  
lrwxrwxrwx 1 root root 9 May 21 20:22 scsi-36888603000008b32fa16688d09368506 -> ../../sda
```

13.3 Capacity Expansion

13.3.1 Can I Reduce or Temporarily Expand the Disk Capacity?

No. The disk capacity can only be expanded, and temporary capacity expansion is not supported.

13.3.2 What Are the Differences Between Expanding Capacity by Expanding an EVS Disk and Creating a New EVS Disk?

The differences are as follows:

- Expanding an EVS disk is when you expand the capacity of an existing EVS disk. Some systems let you expand the capacity of EVS disks in use. In this case, services are not interrupted.
- If you create a new EVS disk and attach it to a server that already has an existing EVS disk, the new EVS disk and the original EVS disk are attached to the same server but independent from each other.

13.3.3 Will My Disk Data Be Lost After I Expand the Disk Capacity?

Data will not be deleted during a system disk or data disk capacity expansion. However, incorrect operations during an expansion may result in data loss or exceptions. Exercise caution when performing capacity expansions. You are advised to back up the disk data before expanding capacity.

13.3.4 Do I Need to Restart the Server After Expanding the Disk Capacity?

An EVS disk can be expanded either in the Available or In-use state. Expanding the disk capacity on the management console enlarges the disk capacity, but you still need to log in to the server and extend the disk partitions and file systems to make that additional space usable. You may need to restart the server during the partition and file system extension. The details are as follows:

- After expanding an In-use disk on the management console, log in to the server and view the disk capacity.
 - If the additional space can be viewed, you can extend the partition and file system and a restart is not required.
 - If the additional space cannot be viewed, the server OS may not be included in the compatibility list. In this case, you should stop and then

start the server (do not restart the server). When the additional space can be viewed, extend the partition and file system.

- After expanding an Available disk on the management console, attach the disk to the server and extend the partition and file system on the server. In this case, a server restart is not required.

13.3.5 Do I Need to Detach an EVS Disk Before Expanding Its Capacity?

An expansion consists of two phases:

1. Expand the disk capacity on the management console.
 - A shared, in-use disk cannot be expanded. You must detach the shared disk from all its servers and then expand its capacity.
 - A non-shared, in-use disk can be expanded, and you can leave the disk attached during expansion as long as the following conditions are met:
 - The disk's server is in the **Running** or **Stopped** state.
 - The disk's server OS supports the expansion of In-use disks.
2. Log in to the server and create a new partition or allocate the additional space to one that is already there.
 - In Windows, there are no partition extensions that require the partition to be unmounted first.
 - In Linux:
 - When allocating the additional space to an existing partition, that is, extending an existing partition, you must use the **umount** command to unmount the partition first.
 - When allocating the additional space to a new partition, that is, creating a new partition, you do not need to unmount the existing partition.

13.3.6 What Should I Do If My Disk Capacity Exceeds 2 TiB After Expansion?

An EVS system disk can be as large as 1 TiB (1,024 GiB). You can expand the capacity of a system disk to up to 1 TiB.

An EVS data disk can be as large as 32 TiB (32,768 GiB).

- With MBR, any disk space in excess of 2 TiB cannot be allocated and used, because the maximum disk capacity supported by MBR is 2 TiB (2,048 GiB).
In this case, if you want to expand the disk capacity to over 2 TiB, 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 deleted during this change.
- With GPT, you can expand the capacity of a data disk to up to 32 TiB because the maximum disk capacity supported by GPT is 18 EiB (19,327,352,832 GiB).
If the in-use partition style is GPT, see the following methods:

- Windows:
Extending Disk Partitions and File Systems (Windows Server 2016)
- Linux:
Extending Partitions and File Systems for Data Disks (Linux)

13.3.7 Why Did My Disk Capacity Remain Unchanged on the Server After Capacity Expansion?

After expanding disk capacity on the management console, you must log in to the server and extend the disk partition and file system for the extra capacity to become available. Otherwise, you cannot view the additional space on the server.

To extend disk partitions and file systems, see the following sections:

- [Extending Disk Partitions and File Systems \(Windows Server 2008\)](#)
- [Partition and File System Extension Preparations \(Linux\)](#)

13.3.8 How Do I Extend the File System of an Unpartitioned Data Disk in Linux?

Scenarios

If no partition but only a file system is created on a data disk, extend the file system according to the following operations:

Run the **lsblk** command. Information similar to the following is displayed:

```
[root@ecs-test ~]# 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   60G  0 disk /mnt/sdc
```

In the command output, no partition but only a file system is created on data disk **/dev/vdb**.

In the following example, CentOS 7.4 64bit is used as the sample OS, data disk **/dev/vdb** has 10 GiB, no partition but only a file system is created on the disk, and additional 50 GiB has been added to this data disk on the management console. The following steps show how to extend this 50 GiB to the file system.

- [Extending the EXT* File System](#)
- [Extending the XFS File System](#)

The way you allocate additional space depends on the OS. This example is used for reference only. For the detailed operations and differences, see the corresponding OS documentations.

Extending the EXT* File System

Step 1 Run the following command to extend the file system:

```
resize2fs Disk name
```

In this example, run the following command:

resize2fs /dev/vdb

Information similar to the following is displayed:

```
[root@ecs-test ~]# resize2fs /dev/vdb
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/vdb is mounted on /root/test; on-line resizing required
old_desc_blocs = 2, old_desc_blocs = 8
[17744.521535] EXT4-fs (vdb): resizing filesystem from 26214400 to 15728640 blocks
[17744.904470] EXT4-fs (vdb): resized filesystem to 15728640
The filesystem on /dev/vdb is now 15728640 blocks long.
```

Step 2 Run the following command to view the result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-test ~]# 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/vdb        ext4      64G   55M   61G   1% /mnt/sdc
```

----End

Extending the XFS File System

Step 1 Run the following command to extend the file system:

xfs_growfs *Disk name*

In this example, run the following command:

xfs_growfs /dev/vdb

Information similar to the following is displayed:

```
[root@ecs-test ~]# xfs_growfs /dev/vdb
meta-data=/dev/vdb            isize=512    agcount=4, agsize=655360 blks
        =                       sectsz=512   attr=2,    projid32bit=1
        =                       crc=1        finobt=0,   spinodes=0
data      =                       bsize=4096   blocks=2621440, imaxpct=25
        =                       sunit=0      swidth=0 blks
naming    =version2           bsize=4096   ascii-ci=0 ftype=1
log        =internal          bsize=4096   blocks=2560, version=2
        =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none               extsz=4096   blocks=0, rtextents=0
data blocks changed from 2621440 to 15728640.
```

Step 2 Run the following command to view the result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-test ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda1       ext4      40G   2.3G   35G   7% /
devtmpfs        devtmpfs  1.9G   0   1.9G   0% /dev
tmpfs           tmpfs     1.9G   0   1.9G   0% /dev/shm
tmpfs           tmpfs     1.9G   8.6M   1.9G   1% /run
tmpfs           tmpfs     1.9G   0   1.9G   0% /sys/fs/cgroup
```

```
tmpfs      tmpfs    379M    0 379M    0% /run/user/0
/dev/vdb    xfs        60G    34M    60G    1% /mnt/sdc
```

----End

13.3.9 How Do I View the Disk Partition Style in Linux?

You can use either `fdisk` or `parted` to view the disk partition style.

- [Method 1: Check Partition Style and File System Format Using `fdisk`](#)
- [Method 2: Check Partition Style and File System Format Using `parted`](#)

Method 1: Check Partition Style and File System Format Using `fdisk`

Step 1 Run the following command to view all the disks attached to the server:

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 150G  0 disk
└─vdb1 253:17   0 100G  0 part /mnt/sdc
```

In this example, data disk **/dev/vdb** already has partition **/dev/vdb1** before capacity expansion, and the additional 50 GiB added has not been allocated yet. Therefore, **/dev/vdb** has 150 GiB, and **/dev/vdb1** has 100 GiB.

NOTE

If you run **lsblk** and find out that disk **/dev/vdb** has no partitions, format the disk by referring to [How Do I Extend the File System of an Unpartitioned Data Disk in Linux?](#) and expand the capacity. Otherwise, the additional space cannot be used after expansion.

Step 2 Run the following command to view the current disk partition style:

fdisk -l

Information similar to the following is displayed:

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

Disk /dev/vda: 42.9 GiB, 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: 161.1 GiB, 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: dos
Disk identifier: 0x38717fc1

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

The value in the **System** column indicates the disk partition style. Value **Linux** indicates the MBR partition style. Value **GPT** indicates the GPT partition style.

- If the disk partitions displayed are inconsistent with those obtained in [Step 1](#), the possible reason may be that existing partitions uses GPT and there is unallocated disk space. In this case, you cannot query all the partitions using **fdisk -l**. Go to [Method 2: Check Partition Style and File System Format Using parted](#).
- If the disk partitions displayed are consistent with those obtained in [Step 1](#), continue with the following operations.

Step 3 Run the following command to view the partition's file system format:

blkid *Disk partition*

In this example, run the following command:

blkid /dev/vdb1

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

In the command output, the **TYPE** value is **ext4**, indicating that **/dev/vdb1**'s file system format is **ext4**.

Step 4 Run the following command to view the file system status:

ext*: **e2fsck -n** *Disk partition*

xfs: **xfs_repair -n** *Disk partition*

In this example, the ext4 file system is used. Therefore, run the following command:

e2fsck -n /dev/vdb1

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# e2fsck -n /dev/vdb1
e2fsck 1.42.9 (28-Dec-2013)
Warning! /dev/vdb1 is mounted.
Warning: skipping journal recovery because doing a read-only filesystem check.
/dev/vdb1: clean, 11/6553600 files, 459544/26214144 blocks
```

If the file system status is **clean**, the file system is normal. Otherwise, rectify the faulty and then perform the capacity expansion.

----End

Method 2: Check Partition Style and File System Format Using parted

Step 1 Run the following command to view all the disks attached to the server:

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 150G 0 disk
└─vdb1 253:17 0 100G 0 part /mnt/sdc
```

In this example, data disk **/dev/vdb** already has partition **/dev/vdb1** before capacity expansion, and the additional 50 GiB added has not been allocated yet. Therefore, **/dev/vdb** has 150 GiB, and **/dev/vdb1** has 100 GiB.

NOTE

If you run **lsblk** and find out that disk **/dev/vdb** has no partitions, format the disk by referring to [How Do I Extend the File System of an Unpartitioned Data Disk in Linux?](#) and expand the capacity. Otherwise, the additional space cannot be used after expansion.

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

parted Disk

For example, run the following command to view **/dev/vdb**'s partition style:

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) p
Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another
operating system believes the
disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?
Fix/Ignore/Cancel? Fix
Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of
the space (an extra 104857600
blocks) or continue with the current setting?
Fix/Ignore? Fix
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 161GiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
1       1049kB  107GiB  107GiB  ext4         test

(parted)
```

Partition Table indicates the disk partition style. **Partition Table: msdos** means MBR, **Partition Table: gpt** means GPT, and **Partition Table: loop** means that the whole disk is partitioned.

- If the following error information is displayed, enter **Fix**.

Error: The backup GPT table is not at the end of the disk, as it should be. This might mean that another operating system believes the disk is smaller. Fix, by moving the backup to the end (and removing the old backup)?

The GPT partition table information is stored at the start of the disk. To reduce the risk of damage, a backup of the information is saved at the end of the disk. When you expand the disk capacity, the end of the disk changes accordingly. In this case, enter **Fix** to move the backup file of the information to new disk end.

- If the following warning information is displayed, enter **Fix**.

Warning: Not all of the space available to /dev/vdb appears to be used, you can fix the GPT to use all of the space (an extra 104857600 blocks) or continue with the current setting?
Fix/Ignore? Fix

Enter **Fix** as prompted. The system automatically sets the GPT partition style for the additional space.

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

----End

13.4 Detachment

13.4.1 If I Detach a Disk, Will I Lose the Data on My Disk?

Data on a disk will not be lost after the disk is detached, and the disk can be re-attached later if needed.

Disk Detachment Process

- For disks not supporting online detachment:
 - a. Stop the server where the disk was attached.
 - b. After the server has stopped, detach the disk.
- For disks supporting online detachment:

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

13.4.2 Why Can't I Detach My Disk?

EVS disks can be used as system disks or data disks, but the way you detach each one is different.

- System disks: A system disk can only be detached offline. You must first stop the server that uses this system disk and then detach the disk.

NOTE

In Linux, a system disk is typically mounted on **/dev/vda**. In Windows, a system disk is normally **Volume (C:)**.

- Data disks: A data disk can be detached regardless of whether it is offline or online.

NOTE

In Linux, a data disk is typically mounted on a mount point other than **/dev/vda**. In Windows, a data disk is normally a volume other than **Volume (C:)**.

- Offline detachment: The server must be in the **Stopped** state. If it is not, stop the server and then detach the data disk.
- Online detachment: Some OSs support online detachment. In this case, you do not need to stop the server before detaching the data disk. For more information, see **Storage > Detaching an EVS Disk from a Running ECS** in the *Elastic Cloud Server User Guide*.

13.5 Capacity

13.5.1 What Is the Maximum Capacity Supported for the System and Data Disks?

The maximum capacity supported for a system disk is 1024 GiB.

The maximum capacity supported for a data disk is 32768 GiB.

13.5.2 What Should I Do If My Disk Starts to Run Out of Space?

If your disk space starts to fill up, you can:

1. Create a new disk and attach it to the server. For details, see [Create an EVS Disk](#).
2. Expand the capacity of the existing disk. Both system disks and data disks can be expanded. An expansion operation includes two steps:
 - a. Expand the disk capacity on the console.
 - b. Log in to the server and extend the partition and file system.

For details, see [Disk Capacity Expansion](#)

Differences Between Expanding an EVS Disk and Creating an EVS Disk

The differences are as follows:

- Expanding an EVS disk is when you expand the capacity of an existing EVS disk. Some systems let you expand the capacity of EVS disks in use. In this case, services are not interrupted.
- If you create a new EVS disk and attach it to a server that already has an existing EVS disk, the new EVS disk and the original EVS disk are attached to the same server but independent from each other.

13.5.3 What Should I Do If I Use fdisk to Initialize a Disk Larger Than 2 TiB and Then the Space in Excess of 2 TiB Cannot Be Displayed?

If your disk capacity is greater than 2 TiB, do not use fdisk to partition the disk. Or any space in excess of 2 TiB will be unable to show up after the disk is partitioned.

In this case, use parted to repartition the disk and choose the GPT partition style because MBR does not support disks over 2 TiB.

For details, see [Introduction to Data Disk Initialization Scenarios and Partition Styles](#).

13.5.4 How Can I View My Disk Usage?

You can view your disk usages in either of the following ways:

- View disk usages manually.
The details depend on the OS. This FAQ uses Windows Server 2008, Windows Server 2016, and Linux as samples to describe how to view the disk usage.

- [Viewing Disk Usage in Linux](#)
- [Viewing Disk Usage in Windows Server 2008](#)
- [Viewing Disk Usage in Windows Server 2016](#)
- [Installing Agent to View Disk Usage](#)

Viewing Disk Usage in Linux

In this section, CentOS 7.4 64bit is used as an example. The details depend on if you need to view the available space or not.

- To query the total capacity only, run **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  40G  0 disk
├─vdb1 253:17  0  40G  0 part
```

In the command output, the server has two disks, **/dev/vda** and **/dev/vdb**. System disk **/dev/vda** has 40 GiB of capacity, as does data disk **/dev/vdb**.

- To query the total capacity and display the space available as well, run **df -TH**. Ensure that the disk has been attached and initialized before running this command.

Information similar to the following is displayed:

```
[root@ecs-0001 ~]# 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/vdb1       ext4       43G   51M   40G   1% /mnt/sdc
```

In the command output, the server has two partitions, **/dev/vda1** and **/dev/vdb1**. Partition **/dev/vda1** is used to deploy the OS, and its total capacity, used capacity, and available capacity are 43 GiB, 2 GiB, and 39 GiB, respectively. Partition **/dev/vdb1**'s total capacity, used capacity, and available capacity are 43 GiB, 51 MiB, and 40 GiB, respectively.

Viewing Disk Usage in Windows Server 2008

In this section, Windows Server 2008 R2 Enterprise 64bit is used as an example.

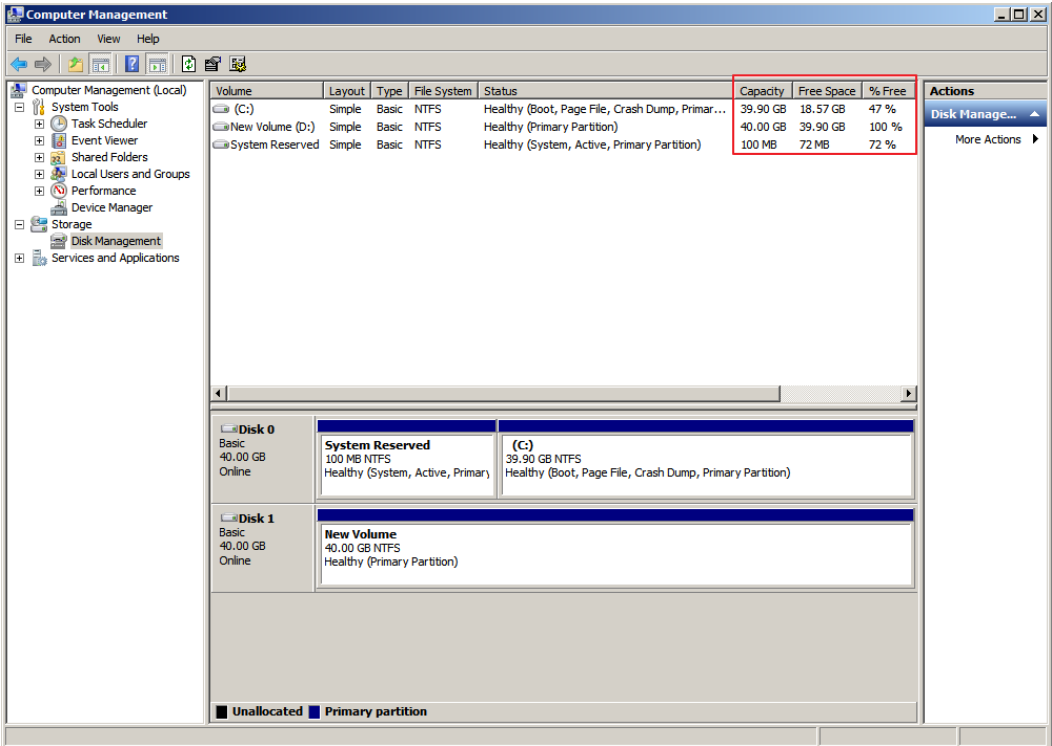
- Step 1** On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The sizes and available spaces of the volumes on the current disk are displayed in the middle pane.

Figure 13-6 Disk Management page



----End

Viewing Disk Usage in Windows Server 2016

In this section, Windows Server 2016 Standard 64bit is used as an example.

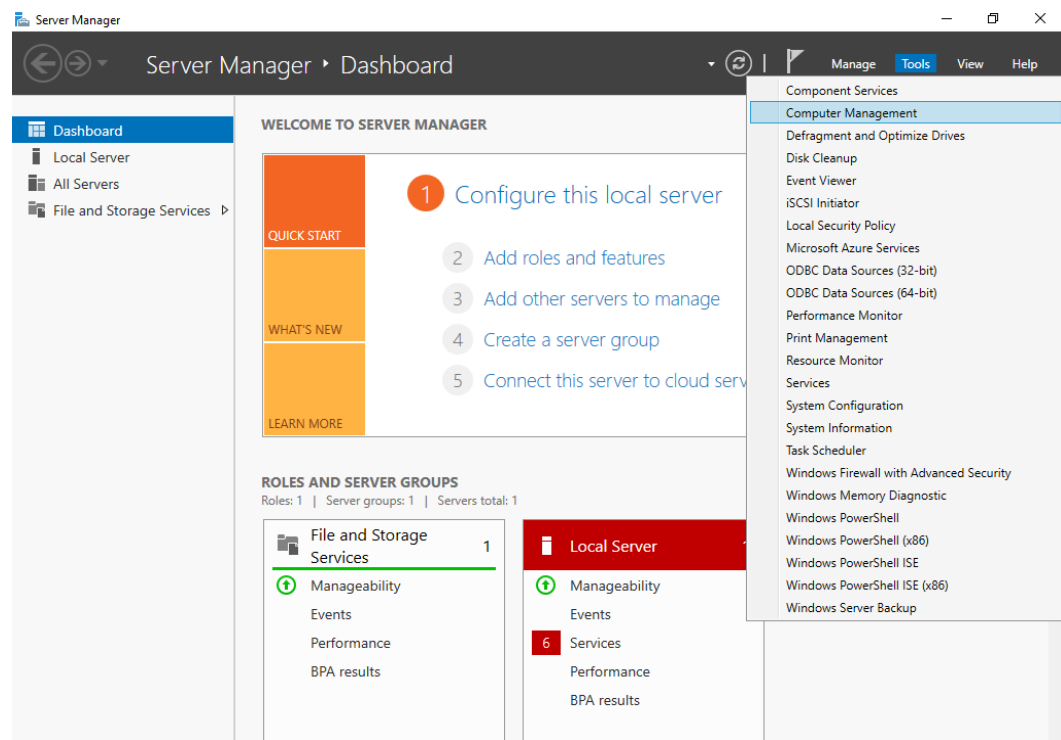
Step 1 On the desktop of the server, click the start icon in the lower left corner.

The **Windows Server** window is displayed.

Step 2 Click **Server Manager**.

The **Server Manager** window is displayed.

Figure 13-7 Server Manager page

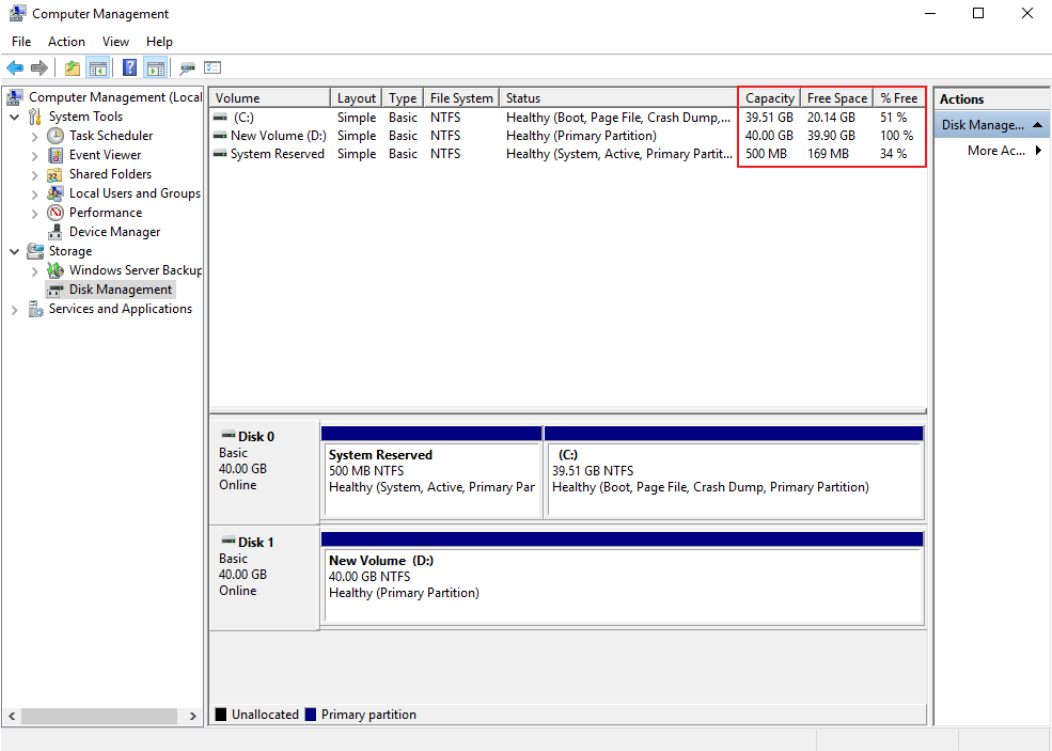


Step 3 In the upper right corner, choose **Tools** > **Computer Management**.

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

In the middle pane, you can view the sizes and available spaces of the volumes on the disk.

Figure 13-8 Disk list page



----End

Installing Agent to View Disk Usage

Some disk monitoring metrics require that the agent to be installed.

For details about how to install the Agent on an ECS, see section "Installing and Configuring the Agent on a Linux ECS or BMS" in the *Cloud Eye User Guide*.

Table 13-3 Disk metrics

Metric	Parameter	Description	Value Range	Monitored Object & Dimension	Monitoring Period (Raw Data)
disk_free	(Agent) Available Disk Space	Free space on the disks Unit: GB - Linux: Run the df -h command to check the value in the Avail column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). - Windows: Use the WMI interface to call GetDiskFreeSpaceExW API to obtain disk space data. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~).	≥ 0	ECS - Mount point	1 minute

Metric	Parameter	Description	Value Range	Monitored Object & Dimension	Monitoring Period (Raw Data)
disk_total	(Agent) Disk Storage Capacity	Total space on the disks, including used and free Unit: GB - Linux: Run the df -h command to check the value in the Size column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). - Windows: Use the WMI interface to call GetDiskFreeSpaceExW API to obtain disk space data. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~).	≥ 0	ECS - Mount point	1 minute

Metric	Parameter	Description	Value Range	Monitored Object & Dimension	Monitoring Period (Raw Data)
disk_used	(Agent) Used Disk Space	Used space on the disks Unit: GB - Linux: Run the df -h command to check the value in the Used column. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). - Windows: Use the WMI interface to call GetDiskFreeSpaceExW API to obtain disk space data. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~).	≥ 0	ECS - Mount point	1 minute

Metric	Parameter	Description	Value Range	Monitored Object & Dimension	Monitoring Period (Raw Data)
disk_usage_percent	(Agent) Disk Usage	Percentage of total disk space that is used, which is calculated as follows: Disk Usage = Used Disk Space/Disk Storage Capacity Unit: percent - Linux: It is calculated as follows: Used/Size. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~). - Windows: Use the WMI interface to call GetDiskFreeSpaceExW API to obtain disk space data. The path of the mount point prefix cannot exceed 64 characters. It must start with a letter, and contain only digits, letters, hyphens (-), dots (.), and swung dashes (~).	0-100	ECS - Mount point	1 minute

13.5.5 Why the Space of My New Disk Is Full After I Uploaded Only 500 MB of Files to the Disk?

Troubleshoot this issue by performing the following steps:

1. Check whether the disk partition usage is 100% or almost 100%.
`df -h`

Figure 13-9 Checking the partition usage

```
[root@ecs-a058 /]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        7.8G   0  7.8G   0% /dev
tmpfs           7.8G  20K  7.8G   1% /dev/shm
tmpfs           7.8G  8.7M  7.8G   1% /run
tmpfs           7.8G   0  7.8G   0% /sys/fs/cgroup
/dev/vda1       99G   99G   0 100% /
tmpfs           1.6G   0  1.6G   0% /run/user/0
```

In this example, the **/dev/vda1** partition usage is 100%.

2. Check the disk space usage.

```
df -i
```

Figure 13-10 Checking the disk space usage

```
[root@ecs-a058 /]# df -i
Filesystem      Inodes    IUsed   IFree IUse% Mounted on
devtmpfs        2030569    369 2030200     1% /dev
tmpfs           2033158     2 2033156     1% /dev/shm
tmpfs           2033158    507 2032651     1% /run
tmpfs           2033158     16 2033142     1% /sys/fs/cgroup
/dev/vda1       6553600 139249 6414351     3% /
tmpfs           2033158     1 2033157     1% /run/user/0
```

In this example, the disk space usage is low.

3. Check the deleted process files in the system.

```
ls -l | grep deleted
```

Figure 13-11 Checking the deleted process files in the system

```
[root@ecs-a058 /]# ls -l | grep delete
agetty    2687      root    txt      REG      253,1      49640    1319348 /usr/sbin/agetty:5ea3f3
mysql    8019      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8020      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8020      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8020      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8020      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8020      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8021      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8021      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8021      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8021      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8021      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8022      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8022      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8022      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8022      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8022      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8023      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8023      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8023      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8023      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8023      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8024      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8024      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8024      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8024      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8024      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8025      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8025      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8025      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8025      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8025      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
mysqld   8019 8026      mysql   5u       REG      253,1      0        528810   /tmp/ibTWxfsx (deleted)
mysqld   8019 8026      mysql   6u       REG      253,1      0        528811   /tmp/ib0KEqPb (deleted)
mysqld   8019 8026      mysql   7u       REG      253,1      0        528812   /tmp/ibX0VBcQ (deleted)
mysqld   8019 8026      mysql   8u       REG      253,1      0        528813   /tmp/ib73Mfc9 (deleted)
mysqld   8019 8026      mysql   14u      REG      253,1      0        528814   /tmp/ibQ0UESN (deleted)
```

Roughly calculate the total size of the deleted files based on the returned command output. If it is almost the same as the used space of the disk, the disk space may be used up by the deleted processes that have not been released.

4. Go to the location of a deleted file to check whether the file is still there.
`ll /tmp/`

Note that variable `/tmp/` in the command indicates the path of the deleted file.

5. If the file is not there, run the following command to terminate the process, or restart the server to release the used space.

```
kill -9 PID
```

Note that variable `PID` in the command indicates the process ID.

6. Check that the process is terminated.

```
lsof | grep deleted
```

7. Check that the disk partition usage is no longer 100%.

```
df -h
```

13.6 Snapshot

13.6.1 What Are the Typical Causes of a Snapshot Creation Failure?

A snapshot creation will fail if its source disk is in an intermediate state, such as **Attaching** and **Expanding**, or an abnormal state, such as **Error** and **Restoration failed**.

Ensure that a disk is in the **In-use** or **Available** state before creating a snapshot.

13.6.2 Does EVS Support Automatic Snapshot Creation?

No, but you can develop a program to periodically create snapshots through APIs. Alternatively, you may use the CBR service and enable auto backup to create backups periodically. For details, see section "Creating a Backup Policy" in the *Cloud Backup and Recovery User Guide*.

13.6.3 Can I Create Snapshots for Multiple Disks at a Time?

No.

Snapshots cannot be created for multiple disks at the same time. You must create snapshots for disks one by one.

13.6.4 How Is a Snapshot Created for My Disk?

There are two types of snapshots: manual snapshots you create on-demand and automatic snapshots created by the system.

- Manual snapshots: You may manually create snapshots to rapidly save the disk data at specific points of time. For details about how to create snapshots, see [Creating a Snapshot](#).

- Automatic snapshots: During the creation of a cloud server backup, or a disk backup with the CBR service, the system automatically creates a snapshot and saves the latest snapshot for each disk. If the disk already has a backup and a new backup is created, the system will automatically delete the old snapshot and save the latest one generated. This snapshot is free of charge. You can view the snapshot details only but cannot perform any operations on it.

13.6.5 Why Can't I Roll Back My Disk Data from a Snapshot?

Possible causes are as follows:

- Snapshot data can be rolled back only when the status of the snapshot's source disk is **Available** or **Rollback failed**. If the snapshot's source disk is **In-use**, detach the disk and then roll back the snapshot data. After the rollback succeeds, re-attach the disk.
- A snapshot whose name starts with **autobk_snapshot_vbs_**, **manualbk_snapshot_vbs_**, **autobk_snapshot_csbs_**, or **manualbk_snapshot_csbs_** is automatically generated during backup. Such a snapshot can only be viewed. It cannot be used to roll back the disk data.

13.6.6 Can I Roll Back the Data from a Snapshot After Reinstalling the OS or Formatting the Disk?

- If you have reinstalled or changed the server OS, snapshots of the system disk are automatically deleted. Snapshots of the data disks can be used as usual.
- If you have formatted a disk, snapshots can still be used to roll back the disk data.

13.6.7 How Is the Snapshot Size Calculated?

When you create a snapshot for a disk for the first time, the snapshot is a full snapshot, and its size is the same as the disk size. Subsequent snapshots created for the same disk are incremental snapshots, so they are only as large as the amount of data that has been added.

13.6.8 Do Snapshots Take Space on the Disk?

No.

Snapshots are stored on the physical disks that provide storage resources for EVS disks. Therefore, snapshots do not use the EVS disk space.

13.6.9 Can I Perform Multiple Rollback Operations for a Snapshot?

Yes. You can roll back a snapshot to the source disk multiple times as needed.

13.6.10 Can I Replicate Snapshots to Other Regions or Accounts?

No. Snapshots cannot be replicated to other regions or accounts.

13.6.11 Why Can't I Find My Snapshot?

Possible causes are:

- You have manually deleted the snapshot.
- You have deleted the disk based on which the snapshot was created. If a disk is deleted, all the snapshots created for this disk will be automatically deleted.
- You have reinstalled or changed the server OS. System disk snapshots will be automatically deleted after a server OS has been reinstalled or changed.

13.6.12 Can I Use a Snapshot to Create a Disk and Roll Back the Snapshot at Almost the Same Time?

There are two conditions:

- If you first use the snapshot to create a new disk, you can then roll back the snapshot to the original disk.
- If you first roll back the snapshot to the original disk, you cannot use the snapshot to create a new disk then.

13.7 Sharing

13.7.1 Do I Have to Deploy a Cluster to Use Shared Disks?

Yes.

If you simply attach a shared disk to multiple servers, files cannot be shared among them. Because there are no mutually agreed data read/write rules among servers, read and write operations from them may interfere with each other, or unpredictable errors may occur.

Shared EVS disks do not have cluster management capabilities. You need to build a clustered system for data sharing, such as Windows MSCS, Veritas VCS, and Veritas CFS clusters.

13.7.2 How Many Servers Can I Attach a Shared Disk to?

A shared disk can be attached to up to 16 servers.

13.7.3 How Can I Attach a Shared Disk to Multiple Servers?

A shared disk can be attached to multiple servers on the management console. You can choose to attach it to servers one by one or in a batch.

For details, see [Attaching a Shared Disk](#).

13.7.4 Can a Shared Disk Be Attached to Servers That Belong to Different Accounts?

No. A shared disk can only be attached to servers that belong to the same account and are in the same AZ.

13.7.5 Can I Attach a Shared Disk to Servers Running Different OSs?

- It is recommended that you do not simultaneously attach a shared disk to servers running Linux and Windows.
- If a shared disk is attached to servers running different versions of the same OS type, it can be used normally. For example, a shared disk attached to one server running CentOS 6 and another server running CentOS 7 can work fine.

A Appendix

A.1 EVS Disk Status

An EVS disk has several statuses. [Table A-1](#) lists EVS disk statuses, the meaning of each status, and the operations a disk in each status allows.

Table A-1 Disk status details

EVS Disk Status	Status Description	Allowed Operation
In-use	The EVS disk is attached to a server and is in use.	- Detaching - Expanding capacity NOTE A shared In-use EVS disk can be attached. A shared In-use EVS disk cannot be expanded.
Available	The EVS disk has not been attached to any server and can be attached.	- Attaching - Expanding capacity - Deleting
Creating	The EVS disk is being created.	None
Attaching	The EVS disk is being attached to a server.	None
Detaching	The EVS disk is being detached from a server.	None
Deleting	The EVS disk is being deleted.	None
Expanding	The capacity of the EVS disk is being expanded.	None

EVS Disk Status	Status Description	Allowed Operation
Uploading	Data on the EVS 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 EVS disk. This status occurs when you create a server.	None
Error	An error occurs when you try to create an EVS disk.	Deleting
Deletion failed	An error occurs when you try to delete the EVS disk.	None
Expansion failed	An error occurs when you try to expand the capacity of the EVS disk.	Deleting
Awaiting transfer	The EVS disk is awaiting for a transfer.	None

A.2 EVS Snapshot Status

An EVS snapshot has several statuses. [Table A-2](#) lists the EVS snapshot statuses, the meaning of each status, and the operations a snapshot in each status allows.

Table A-2 Snapshot status details

Snapshot Status	Description	Allowed Operation
Creating	The snapshot is being created.	No operations are allowed.
Available	The snapshot is successfully created.	• Creating EVS disks using snapshots • Deleting snapshots • Rolling back snapshots to source EVS disks
Deleting	The snapshot is being deleted.	No operations are allowed.
Error	An error occurs when you try to create a snapshot.	Deleting
Deletion failed	An error occurs when you try to delete a snapshot.	No operations are allowed.

Snapshot Status	Description	Allowed Operation
Rolling back	The snapshot is rolling back data. NOTE - When you roll back from a snapshot, you can only roll back data to the source EVS disk. Rollback to a specified disk is not supported. - A snapshot can only be used for rollback when its source disk is in the Available or Rollback failed state.	No operations are allowed.

B Change History

Released On	Description
2024-04-15	This issue is the first official release.