Dedicated Storage Service

User Guide

Issue	01
Date	2018-04-30

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Permissions Management

1.1 Creating a User and Granting DSS Permissions

This chapter describes how to use **IAM** to implement fine-grained permissions control for your DSS resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing DSS resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust a Huawei Cloud account or cloud service to perform efficient O&M on your DSS resources.

If your Huawei Cloud account does not require individual IAM users, skip this section.

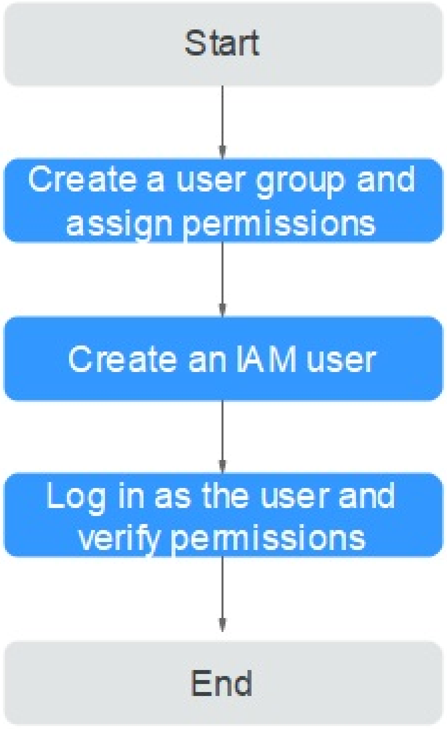
This section describes the procedure for granting permissions (see **Figure 1-1**).

Prerequisites

Learn about the permissions (see **DSS Permissions**) supported by DSS and choose policies or roles according to your requirements.

Process Flow

Figure 1-1 Process for granting DSS permissions


Start
Create a user group and assign permissions
Create an IAM user
Log in as the user and verify permissions
End

1. **Create a user group and assign permissions** to it.

   Create a user group on the IAM console, and attach the **DSS ReadOnlyAccess** policy to the group.

2. **Create a user** and add it to a user group.

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Log in** and verify permissions.

   Log in to the DSS console using the created user and verify that the user only has read permissions for DSS.

   - Choose **Service List** > **Dedicated Distributed Storage Service**. Then click **Apply for Storage Pool** on the DSS console. If a message appears indicating that you have insufficient permissions to perform the operation, the **DSS ReadOnlyAccess** policy has already taken effect.
   - Choose any other service in **Service List**. If a message appears indicating that you have insufficient permissions to access the service, the **DSS ReadOnlyAccess** policy has already taken effect.

1.2 DSS Custom Policies

Custom policies can be created to supplement the system-defined policies of DSS. For the actions supported for custom policies, see **Permissions Policies and Supported Actions**.

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 **Creating a Custom Policy**. The following section contains examples of common custom policies.

Example Custom Policies

- Example 1: Authorize a user to query storage pools.

```
{
        "Version": "1.1",
        "Statement": [
                {
                        "Action": [
                                "dss:pools:get",
                                "dss:pools:list",
                        ],
                        "Effect": "Allow"
                }
        ]
}
```

2 Storage Pool Management

2.1 Expanding a Storage Pool

Scenarios

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

NOTE

The additional capacity added is charged based on the remaining time of the storage pool. After capacity expansion, the storage pool expiration time remains unchanged.

Procedure

Step 1 Log in to the management console.

Step 2 Choose **Dedicated Distributed Storage Service** > **Storage Pools**. In the storage pool list, locate the storage pool you want to expand and click **Expand Capacity** in the **Operation** column.

Step 3 Set parameter **Add Capacity (TB)** as prompted. After the operation is complete, click **Submit**.

Step 4 On the displayed page, confirm the expansion details.

- If you do not need to modify the specifications, click **Submit** and perform the subsequent operations as prompted.
- If you need to modify the specifications, click **Previous** to modify parameters.

Step 5 After the order is paid, view the storage pool status.

- If the storage pool status is **Expanding**, your order is being processed and the storage resources are being expanded.
- When the storage pool status changes to **In-use**, the storage pool has been expanded and your order is completed.

----End

3 Disk Management

3.1 Detaching a Disk

3.1.1 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 detached system disk will not be automatically deleted, and it will still be billed. To avoid unintended charges, you can delete or unsubscribe from the disk if it is no longer needed.

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.

NOTE

- For an attached system disk, the disk function is displayed as **System disk**, and the disk status is displayed as **In-use** in the disk list. After a system disk is detached from the server, the disk function changes to **Bootable disk**, and the status changes to **Available**.
- 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.

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 **Yes** to detach the disk.

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

Step 8 (Optional) 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 device name selected.

- To re-attach and use it as a system disk, see **Attaching an Existing System Disk**.
- To re-attach and use it as a data disk, see **Attaching an Existing Non-Shared Disk**.

----End

3.1.2 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 an EVS disk from a running server. For details, see **Detaching an EVS Disk from a Running ECS**.
- 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.

Detaching a Non-shared Disk

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

- If you need to view the server information, perform the following procedure:
  a. In the disk list, click the name of the to-be-detached disk.
     The disk details page is displayed.
  b. Click the **Servers** tab to view the servers where the target disk has been attached.
  c. Click □ to select the server and click **Detach Disk**.
     The **Detach Disk** dialog box is displayed.
  d. Click **Yes** to detach the disk.
- If you do not need to view the server information, perform the following procedure:
  a. In the disk list, locate the row that contains the target disk and choose **More** > **Detach** in the **Operation** column.
     The **Detach Disk** dialog box is displayed.
  b. Click **Yes** to detach the disk.

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

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

----End

Detaching a Shared Disk

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

- If you need to view the server information, perform the following procedure:
  a. In the disk list, click the name of the to-be-detached disk.
     The disk details page is displayed.
  b. Click the **Servers** tab to view the servers where the target disk has been attached.
  c. Click □ to select the servers and click **Detach Disk**.
     Shared disks support batch detachment so that you can select multiple servers at a time.

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

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

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

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

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

  b. Click □ to select the servers.

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

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

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

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

----End

3.2 Attaching an Existing Disk

3.2.1 Attaching an Existing System Disk

Scenarios

This section describes how to attach an existing system disk.

System disks can only be attached offline, which means that the server must be in the **Stopped** state.

You can view the disk function in the disk list. A disk can be attached to a server and used as the system disk only when its function is **Bootable disk** and its status is **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.
- Non-shared, yearly/monthly disks purchased together with or later added to a yearly/monthly ECS cannot be attached to other ECSs.

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

A disk can be attached to a server and used as system disk only when its function is **Bootable disk** and its status is **Available**.

Step 4 Select the server and then select a device name from the drop-down list. Ensure that the disk and server are in the same AZ and that the server is in the **Stopped** state.

One device name can be attached with one disk only. For the mapping between device names displayed on the management console and those on the server, see **What Is the Mapping Between Device Names and Disks?** in the *Elastic Cloud Server User Guide*.

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

----End

3.2.2 Attaching an Existing Non-Shared Disk

Scenarios

This topic describes how to attach an existing non-shared disk to a server and use it as data disk. A non-shared disk can be attached to only one server.

You can view the disk information in the disk list. A disk can be attached to a server and used as a data disk when all of the following conditions are met:

- Disk Sharing: Disabled
- Function: Bootable disk or Data disk
- Status: 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 system disk or data disk depending on the device name selected.

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

A disk can be attached to a server and used as a system disk only when its function is **Bootable disk** and its status is **Available**.

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

One device name can be attached with one disk only. For the mapping between device names displayed on the management console and those on the server, see **What Is the Mapping Between Device Names and Disks?** in the *Elastic Cloud Server User Guide*.

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

----End

3.2.3 Attaching an Existing Shared Disk

Scenarios

This topic describes how to attach an existing shared disk to a server and use it as data disk.

You can view the disk information in the disk list. A disk can be attached to a server and used as a data disk when all of the following conditions are met:

- Disk Sharing: Enabled
- Function: Data disk
- Status: In-use or 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 system disk or data disk depending on the device name selected.

A shared disk can be attached to a maximum of 16 servers. If the target shared disk is in the **In-use** state, ensure that the maximum number of servers that the disk can be attached to has not been reached.

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

- Shared disks support batch attachment so that you can attach a shared disk to multiple servers at a time. The left area in the **Attach Disk** dialog box shows the server list. After you select the target servers, the selected servers will be displayed in the right area.
- A shared disk can be attached only when the disk status is **Available** or **In-use**.

Step 4 Select the server you want to attach the disk and choose a device name from the drop-down list. Ensure that the selected server and the target disk are in the same AZ.

One device name can be attached with one disk only. If a device name has been used, it will no longer be displayed in the drop-down list and cannot be selected.

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

NOTICE

If you simply attach a shared disk to multiple servers, files cannot be shared between the servers. To share files between servers, build a shared file system or deploy a cluster management system.

----End

3.3 Deleting a Disk

Scenarios

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

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

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

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

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

----End

3.4 Expanding the Capacity of a Disk

3.4.1 Introduction to Expansion Scenarios

What Is Capacity Expansion?

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

Both system disks and data disks can be expanded. A system disk can be expanded to up to 1 TB, and a data disk to 32 TB. Currently, disk capacities can be expanded only. Capacity reduction is not supported.

How to Expand the Disk Capacity?

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

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

After the disk capacity has been expanded, the additional disk space needs to be allocated to an existing partition or a new partition.

- To expand a disk attached to a Windows server, see **Extending Partitions and File Systems on Windows**.
- To expand a disk attached to a Linux server using fdisk, see **Extending Partitions and File Systems on Linux (fdisk)**.
- To expand a disk attached to a Linux server using parted, see **Extending Partitions and File Systems on Linux (parted)**.
- To expand a SCSI disk attached to a Linux server using fdisk, see **Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)**.

NOTICE

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

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

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

3.4.2 Expanding an In-use Disk

Scenarios

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

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

- During such an expansion, the server containing the to-be-expanded disk must be in the **Running** or **Stopped** state.
- Shared disks must be expanded when they are in the **Available** state. For details, see **Expanding an Available Disk**.
- Currently, only some server OSs support the expansion of In-use disks. Therefore, ensure that your server OS meets the requirements for expanding In-use disks before operation. **Table 3-1** lists the server OSs, including the OS images listed on the **Public Images** page of the IMS console and others, that support In-use disk expansion.

  If the server OS does not support capacity expansion of In-use disks, 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.

Table 3-1 Supported OSs

OS	Version
CentOS	7.4 64bit
	7.3 64bit
	7.2 64bit
	6.8 64bit
	6.7 64bit
	6.5 64bit
Debian	8.6.0 64bit
	8.5.0 64bit
Fedora	25 64bit
	24 64bit
SUSE	SUSE Linux Enterprise Server 12 SP2 64bit
	SUSE Linux Enterprise Server 12 SP1 64bit
	SUSE Linux Enterprise Server 11 SP4 64bit
	SUSE Linux Enterprise Server 12 64bit
OpenSUSE	42.2 64bit
	42.1 64bit
Oracle Linux Server release	7.3 64bit
	7.2 64bit

OS	Version
	6.8 64bit
	6.7 64bit
Ubuntu Server	16.04 64bit
	14.04 64bit
	14.04.4 64bit
Windows	Windows Server 2008 R2 Enterprise 64bit
	Windows Server 2012 R2 Standard 64bit
	Windows Server 2016 Standard 64bit
Red Hat Enterprise Linux	7.3 64bit
	6.8 64bit

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

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

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

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

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

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

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

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

NOTE

If the disk status is **Expanding**, you are not allowed to modify the specifications of the ECS where the disk is attached.

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

The operation method varies depending on the server OS.

- For Windows OSs, see **Extending Partitions and File Systems on Windows**.
- For Linux OSs,
  - If the fdisk partitioning tool will be used, see **Extending Partitions and File Systems on Linux (fdisk)**, **Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)**, or **Extending Partitions and File Systems of a System Disk on Linux (fdisk)**.
  - If the parted partitioning tool will be used, see **Extending Partitions and File Systems on Linux (parted)**.

----End

3.4.3 Expanding an Available Disk

Scenarios

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

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

Procedure

Step 1 Log in to the management console.

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

The disk list page is displayed.

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

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

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

The expansion page is displayed.

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

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

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

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

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

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

> NOTE
>
> If the disk status is **Expanding**, you are not allowed to modify the specifications of the ECS where the disk is attached.

Step 8 Attach the expanded disk to a server. For details, see the following topics:

- **Attaching an Existing System Disk**
- **Attaching an Existing Non-Shared Disk**
- **Attaching an Existing Shared Disk**

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

The operation method varies depending on the server OS.

- For Windows OSs, see **Extending Partitions and File Systems on Windows**.
- For Linux OSs,
  - If the fdisk partitioning tool will be used, see **Extending Partitions and File Systems on Linux (fdisk)**, **Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)**, or **Extending Partitions and File Systems of a System Disk on Linux (fdisk)**.
  - If the parted partitioning tool will be used, see **Extending Partitions and File Systems on Linux (parted)**.

----End

3.4.4 Extending Partitions and File Systems on Windows

Scenarios

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

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

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

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

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

NOTICE

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

Prerequisites

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

System Disk: Add Additional Space to the Original Volume

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

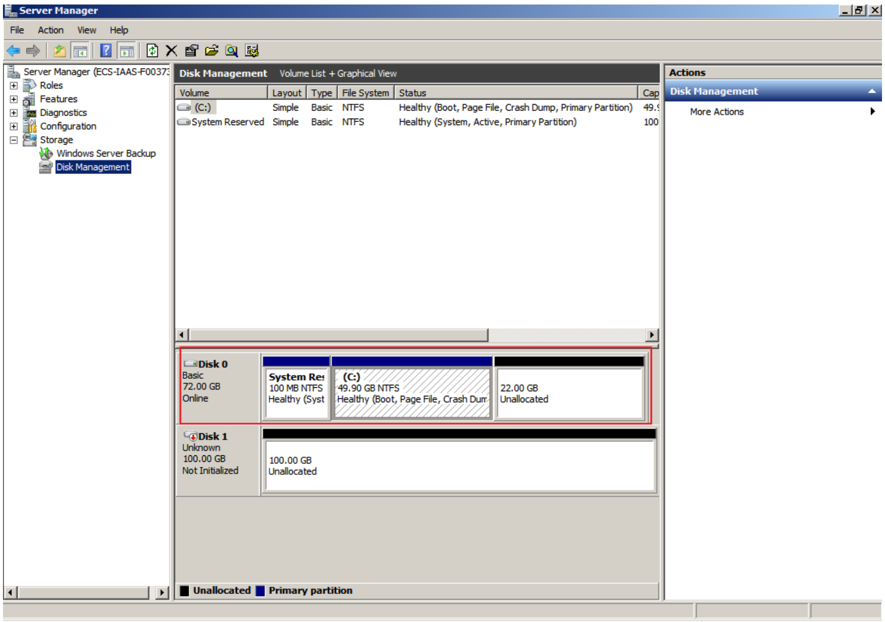
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

Figure 3-1 Disk Management (system disk)



Volume	Layout	Type	File System	Status	Cap
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	49.!
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100

Disk 0
Basic
72.00 GB
Online

System Res
100 MB NTFS
Healthy (Syst

(C:)
49.90 GB NTFS
Healthy (Boot, Page File, Crash Dum

22.00 GB
Unallocated

Disk 1
Unknown
100.00 GB
Not Initialized

100.00 GB
Unallocated

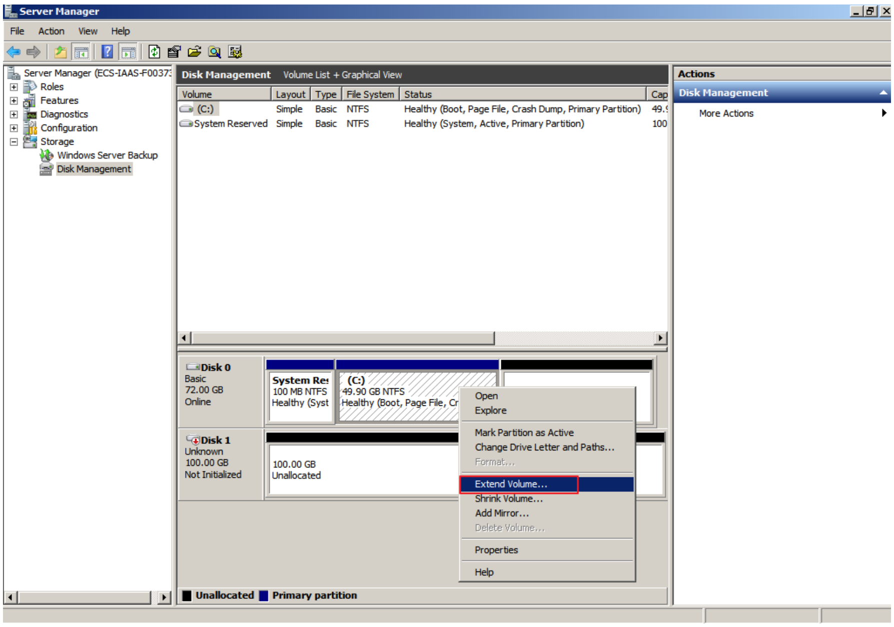
NOTE

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

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

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

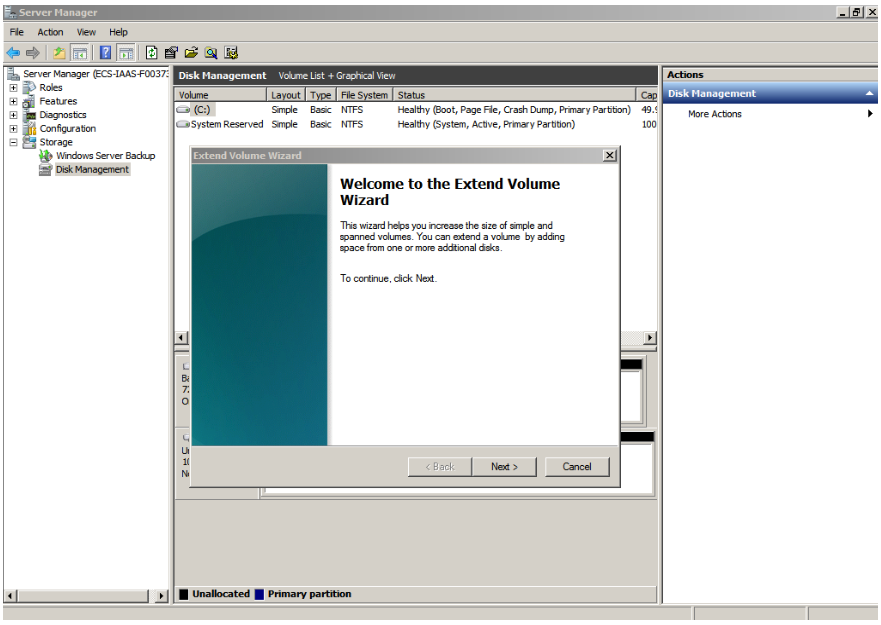
Figure 3-2 Choosing **Extend Volume**



Volume	Layout	Type	File System	Status	Cap
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	49.9
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100

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

Figure 3-3 Extend Volume Wizard



Volume	Layout	Type	File System	Status	Cap
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	49.9
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100

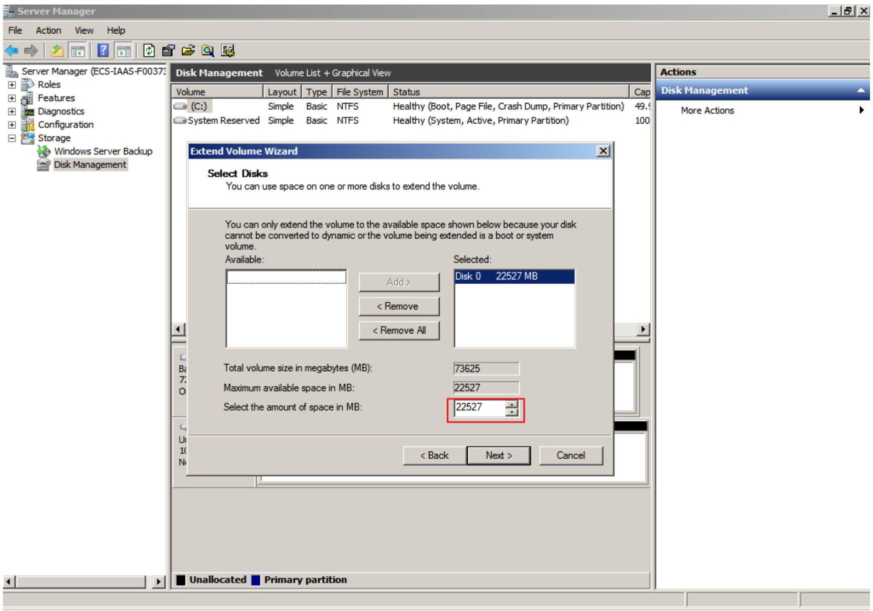
Welcome to the Extend Volume Wizard

This wizard helps you increase the size of simple and spanned volumes. You can extend a volume by adding space from one or more additional disks.

To continue, click Next.

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

Figure 3-4 Selecting space



Volume	Layout	Type	File System	Status	Cap
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	49.9
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100

Extend Volume Wizard

Select Disks

You can use space on one or more disks to extend the volume.

You can only extend the volume to the available space shown below because your disk cannot be converted to dynamic or the volume being extended is a boot or system volume.

Available:

Selected:

Disk 0 22527 MB

Total volume size in megabytes (MB): 73625

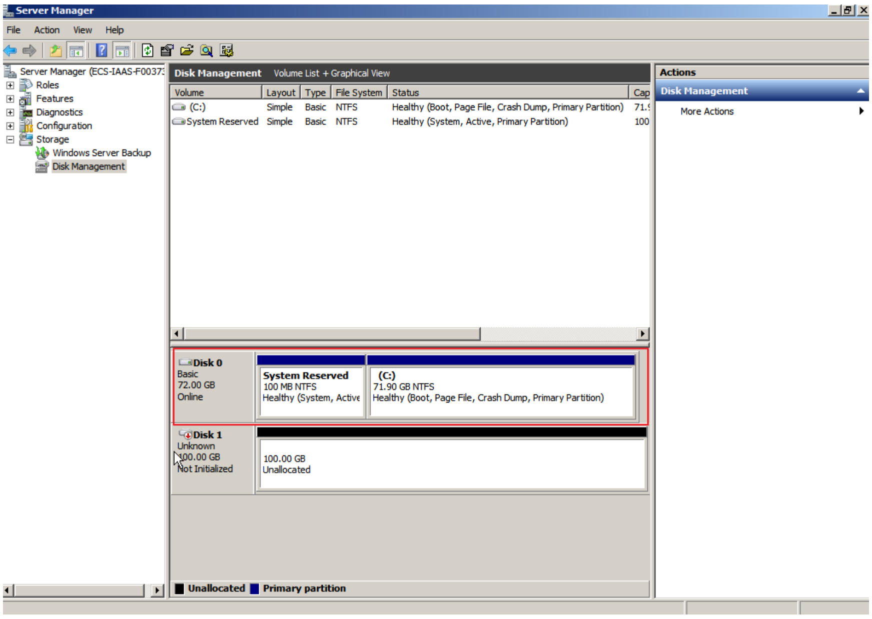
Maximum available space in MB: 22527

Select the amount of space in MB: 22527

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

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

Figure 3-5 Successful capacity expansion



Volume	Layout	Type	File System	Status	Cap
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump, Primary Partition)	71.9
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Partition)	100

Disk 0
Basic
72.00 GB
Online

System Reserved
100 MB NTFS
Healthy (System, Active

(C:)
71.90 GB NTFS
Healthy (Boot, Page File, Crash Dump, Primary Partition)

Disk 1
Unknown
100.00 GB
Not Initialized

100.00 GB
Unallocated

----End

Data Disk: Add Additional Space to a New Volume

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

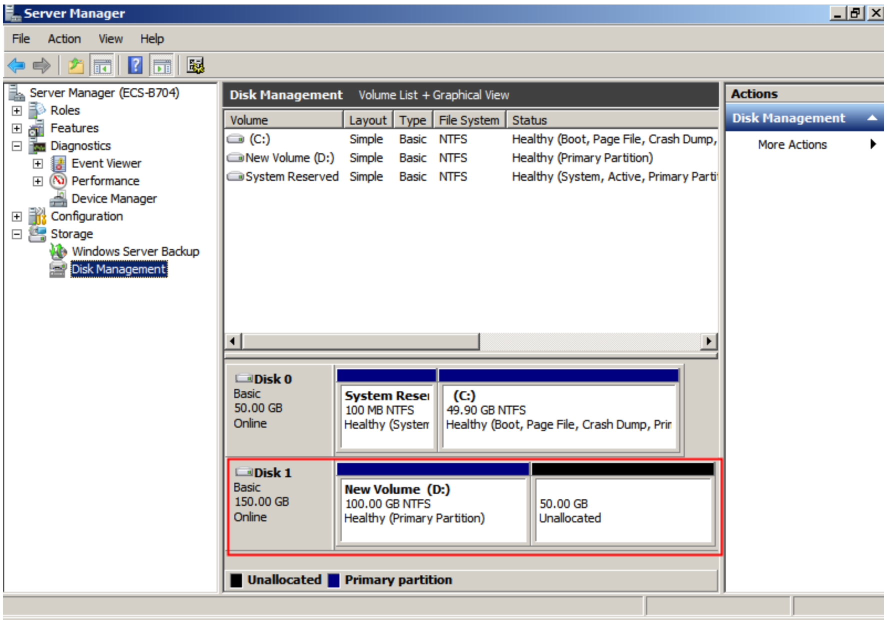
Step 1 On the desktop of the server, right-click **Computer** and choose **Manage** from the shortcut menu.

The **Server Manager** window is displayed.

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

The **Disk Management** window is displayed.

Figure 3-6 Disk Management (data disk)



Volume	Layout	Type	File System	Status
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump,
New Volume (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Parti

Disk 0
Basic
50.00 GB
Online

System Resei
100 MB NTFS
Healthy (Systen

(C:)
49.90 GB NTFS
Healthy (Boot, Page File, Crash Dump, Prir

Disk 1
Basic
150.00 GB
Online

New Volume (D:)
100.00 GB NTFS
Healthy (Primary Partition)

50.00 GB
Unallocated

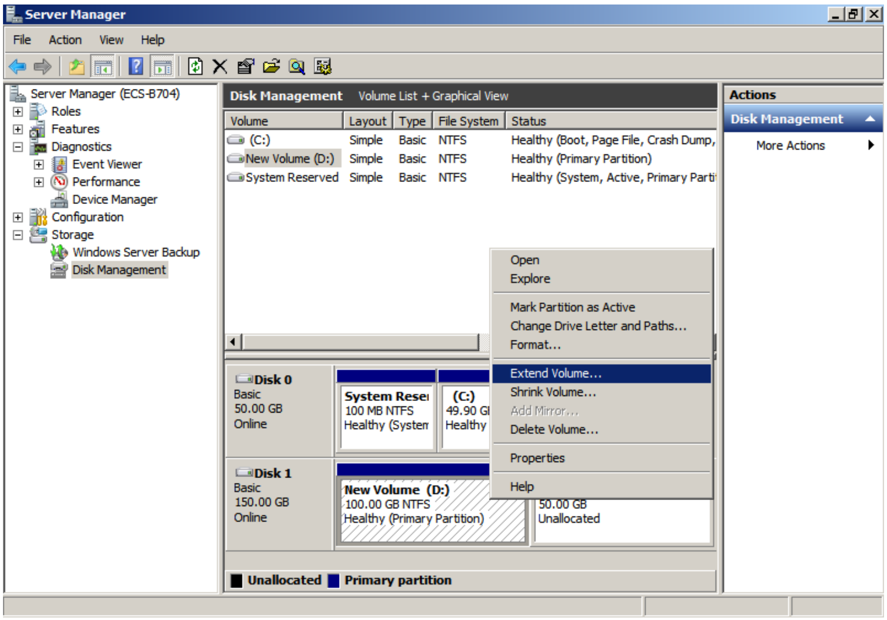
NOTE

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

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

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

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



Volume	Layout	Type	File System	Status
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump,
New Volume (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Parti

Disk 0
Basic
50.00 GB
Online

System Reser
100 MB NTFS
Healthy (Systen

(C:)
49.90 GI
Healthy

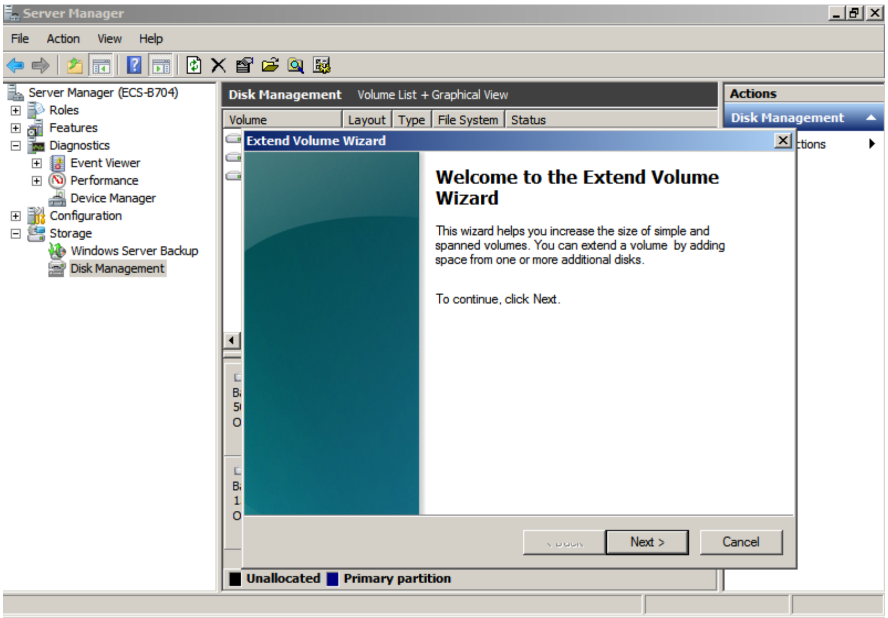
Disk 1
Basic
150.00 GB
Online

New Volume (D:)
100.00 GB NTFS
Healthy (Primary Partition)

50.00 GB
Unallocated

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

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



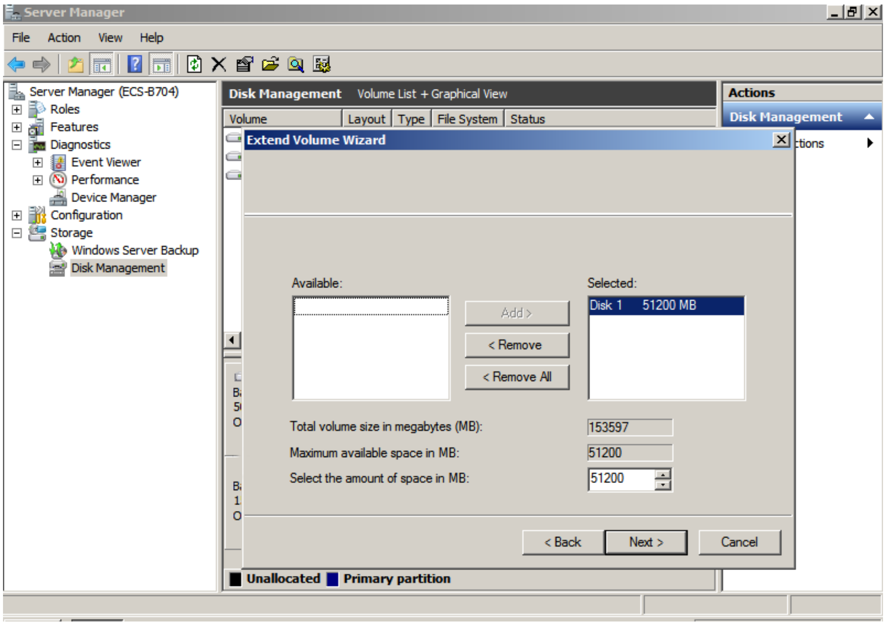
Welcome to the Extend Volume Wizard

This wizard helps you increase the size of simple and spanned volumes. You can extend a volume by adding space from one or more additional disks.

To continue, click Next.

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

Figure 3-9 Selecting space (Windows Server 2008)



Available:

Selected:
Disk 1 51200 MB

Total volume size in megabytes (MB): 153597

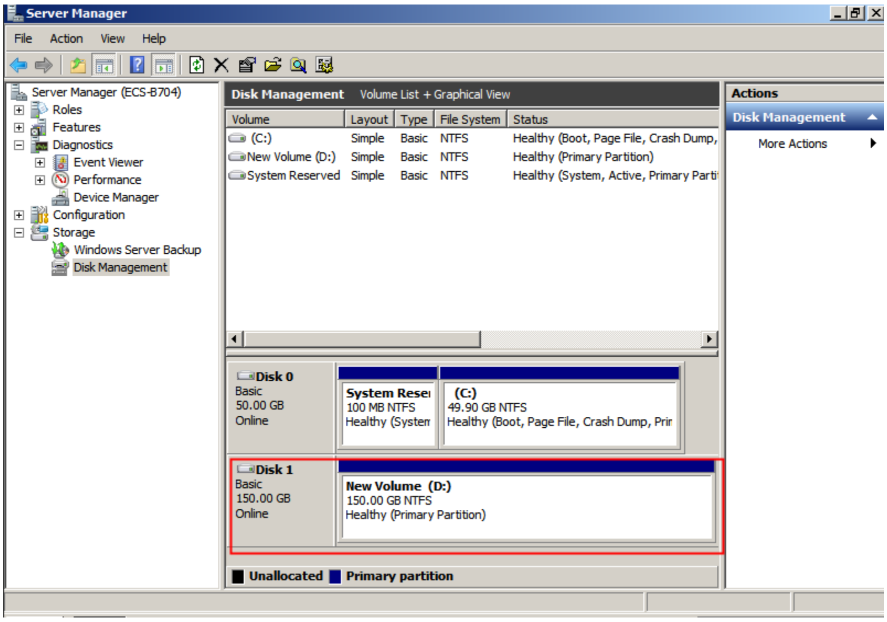
Maximum available space in MB: 51200

Select the amount of space in MB: 51200

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

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

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



Volume	Layout	Type	File System	Status
(C:)	Simple	Basic	NTFS	Healthy (Boot, Page File, Crash Dump,
New Volume (D:)	Simple	Basic	NTFS	Healthy (Primary Partition)
System Reserved	Simple	Basic	NTFS	Healthy (System, Active, Primary Parti

Disk 0
Basic
50.00 GB
Online

System Resei
100 MB NTFS
Healthy (Systen

(C:)
49.90 GB NTFS
Healthy (Boot, Page File, Crash Dump, Prir

Disk 1
Basic
150.00 GB
Online

New Volume (D:)
150.00 GB NTFS
Healthy (Primary Partition)

----End

3.4.5 Extending Partitions and File Systems on Linux (fdisk)

Scenarios

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

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

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

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

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

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

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

- Re-create an existing partition (services will be interrupted).

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

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

NOTICE

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

Prerequisites

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

Checking File Systems on the To-be-expanded Disk

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

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

mount *Disk partition Mount point*

Example command:

mount /dev/xvdb1 /mnt/sdc

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

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

df -TH

Information similar to the following is displayed:

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

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

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

ll *Mounting directory*

Example command:

ll /mnt/sdc

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

----End

Viewing the Partition Style

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

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

fdisk -l

Information similar to the following is displayed:

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

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

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

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


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

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


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

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

----End

Creating a New Partition

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

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

fdisk -l

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

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

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

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

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

Step 2 Run the following command to enter fdisk:

fdisk /dev/xvda

Information similar to the following is displayed:

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

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


Command (m for help):
```

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

Information similar to the following is displayed:

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

NOTE

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

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

To view the example operations in case that the maximum number of primary partitions is not reached for the system disk, see **Extending Partitions and File Systems of a System Disk on Linux (fdisk)**.

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

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

Information similar to the following is displayed:

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

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

The default end sector is used in this example.

Information similar to the following is displayed:

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

Command (m for help):
```

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

Information similar to the following is displayed:

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

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

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

Command (m for help):
```

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

Information similar to the following is displayed:

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

Calling ioctl() to re-read partition table.

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

The partition is created.

NOTE

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

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

partprobe

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

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

mkfs -t ext4 /dev/xvda6

NOTE

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

Information similar to the following is displayed:

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

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

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

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

mount /dev/xvda6 /opt

Information similar to the following is displayed:

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

NOTE

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

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

df -TH

Information similar to the following is displayed:

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

----End

Re-creating an Existing Partition

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

NOTICE

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

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

fdisk -l

Information similar to the following is displayed:

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

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

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

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

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

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

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

- If the additional space is not included, refresh the capacity according to **Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)**.
- If the additional space is included, take note of the start and end sectors of the **dev/xvdb1** partition and then go to **Step 2**. These values will be used in the following operations.

Step 2 Run the following command to unmount the partition:

umount /mnt/sdc

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

fdisk /dev/xvdb

Information similar to the following is displayed:

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

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

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

Command (m for help):
```

NOTE

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

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

Entering **n** creates a new partition.

Information similar to the following is displayed:

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

There are two types of disk partitions:

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

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

Information similar to the following is displayed:

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

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

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

Information similar to the following is displayed:

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

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

NOTE

Data will be lost if the following operations are performed:

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

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

Information similar to the following is displayed:

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

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

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

Information similar to the following is displayed:

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

The partition is successfully created.

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

Information similar to the following is displayed:

```
Command (m for help): p

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

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

Command (m for help):
```

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

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

Information similar to the following is displayed:

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

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

The partition is created.

NOTE

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

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

- For the **ext3** or **ext4** file system

  a. Run the following command to check the correctness of the file system on **/dev/xvdb1**:

     e2fsck -f /dev/xvdb1

     Information similar to the following is displayed:

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

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

     resize2fs /dev/xvdb1

     Information similar to the following is displayed:

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

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

     mount /dev/xvdb1 /mnt/sdc

- For the **xfs** file system

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

mount /dev/xvdb1 /mnt/sdc

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

sudo xfs_growfs /dev/xvdb1

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

df -TH

----End

Setting Automatic Mounting at System Start

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

NOTE

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

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

blkid *Disk partition*

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

blkid /dev/xvdb1

Information similar to the following is displayed:

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

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

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

vi /etc/fstab

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

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

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

The preceding content is used for reference only. Add the information that is used in the environment. The parameters are described as follows:

- The first column indicates the partition UUID obtained in **Step 1**.
- The second column indicates the directory on which the partition is mounted. You can query the mount point using the **df -TH** command.
- The third column indicates the file system format of the partition. You can query the file system format using the **df -TH** command.

- The fourth column indicates the partition mount option. Normally, this parameter is set to **defaults**.
- The fifth column indicates the Linux dump backup option.
  - **0**: not use Linux dump backup. Normally, dump backup is not used, and you can set this parameter to **0**.
  - **1**: use Linux dump backup.
- The sixth column indicates the fsck option, that is, whether to use fsck to check the attached disk during startup.
  - **0**: not use fsck.
  - If the mount point is the root partition (**/**), this parameter must be set to **1**.

    When this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** so that 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.

----End

3.4.6 Extending Partitions and File Systems on Linux (parted)

Scenarios

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

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

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

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

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

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

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.
- Re-create an existing partition (services will be interrupted).

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

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

NOTICE

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

Prerequisites

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

Viewing the Partition Style

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

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

lsblk

Information similar to the following is displayed:

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

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

parted *Disk name*

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

parted /dev/xvdb

Information similar to the following is displayed:

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

```
Partition Table: gpt
Disk Flags:

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

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

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

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

----End

Creating a New Partition

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

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

lsblk

Information similar to the following is displayed:

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

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

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

parted *System disk*

In this example, run the following command:

parted /dev/xvda

Information similar to the following is displayed:

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

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

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

Information similar to the following is displayed:

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

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

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

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

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```

Step 7 Set the file system type and size for the new partition.

Value **83886080** indicates the start sector of the **dev/xvda2** partition you created, and value **167772159** indicates the end sector. The two values are used for reference only. You can determine the number of partitions and partition sizes based on your service requirements.

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```

NOTE

The file system type may fail to set in this step. Therefore, reconfigure the file system type according to **Step 10** after the partition is created.

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

- Run the **fdisk -l** command and take note of the value.
- Enter **-1s**, and the value displayed is the desired value.

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

Information similar to the following is displayed:

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

Number  Start      End         Size       Type     File system  Flags
 1      2048s      83886079s   83884032s  primary  ext4
 2      83886080s  167772159s  83886080s  primary
```

The **dev/xvda2** partition is created.

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

Step 10 Run the following command to set the file system format for the new partition:

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

mkfs -t ext4 /dev/xvda2

NOTE

The procedure for setting the **xfs** file system is the same as that for the **ext3** or **ext4** file system. The command for creating the **xfs** file system is **mkfs -t xfs /dev/xvda2**.

Information similar to the following is displayed:

```
[[root@ecs-1120 linux]# mkfs -t ext4 /dev/xvda2
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
?32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
?4096000, 7962624

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

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

Step 11 Run the following command to mount the new partition on a space-demanding directory, for example **/opt**:

mount /dev/xvda6 /opt

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# mount /dev/xvda2 /opt
[root@ecs-1120 linux]#
```

NOTE

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

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

df -TH

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
```

```
/dev/xvda1     ext4       43G  8.3G   33G  21% /
devtmpfs       devtmpfs  885M     0  885M   0% /dev
tmpfs          tmpfs     894M     0  894M   0% /dev/shm
tmpfs          tmpfs     894M   18M  877M   2% /run
tmpfs          tmpfs     894M     0  894M   0% /sys/fs/cgroup
tmpfs          tmpfs     179M     0  179M   0% /run/user/2000
tmpfs          tmpfs     179M     0  179M   0% /run/user/0
tmpfs          tmpfs     179M     0  179M   0% /run/user/1001
/dev/xvda2     ext4       43G   51M   40G   1% /opt
```

----End

Re-creating an Existing Partition

The following example shows you how to make use of the additional space of a disk attached to a server by re-creating the **/dev/xvdc1** partition mounted on **/mnt/sdc**. Because the **/dev/xvdc** disk only has one partition, this partition is regarded as the partition at the disk end. During the partition re-creation, services will be interrupted.

NOTICE

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

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

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 sdc]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0   10G  0 part /mnt/sdc
```

In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the existing partition takes 10 GB. The **/dev/xvdc1** partition is at the end of the disk and has been mounted on **/mnt/sdc**.

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

- If the additional space is not included, refresh the capacity according to **Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)**.
- If the additional space is included, go to **Step 2**.

Step 2 Run the following command to unmount the partition:

umount /mnt/sdc

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

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# umount /mnt/sdc
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0   10G  0 part
```

Step 4 Run the following command to enter parted to allocate the additional space of the data disk to a partition:

parted *Data disk*

In this example, run the following command:

parted /dev/xvdc

Information similar to the following is displayed:

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

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

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

Information similar to the following is displayed:

```
(parted) mkpart
Partition type?  primary/extended? p
File system type?  [ext2]? ext4
Start? 83886080
End? 1677722159
```

Step 7 The **/dev/xvdc1** partition number is **1**. Therefore, enter **rm 1** and press **Enter** to delete the partition.

Step 8 Enter **p** and press **Enter** to check whether the **/dev/xvdc1** partition has been deleted.

Information similar to the following is displayed:

```
(parted) rm 1
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdc: 125829120s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End        Size        File system  Name  Flags
```

Step 9 Enter **mkpart opt** *2048s 125829119* and press **Enter** to re-create the partition.

2048 specifies the start sector recorded in **Step 6**, and *125829119* specifies the end sector, which must be greater than or equal to the end sector recorded in **Step 6**.

Information similar to the following is displayed:

```
(parted) mkpart opt 2048s 125829119s
Warning: You requested a partition from 2048s to 125829199s (sectors 2048..125829199).
```

```
The closest location we can manage is 2048s to 125829036s (sectors 2048..125829036).
Is this still acceptable to you?
Yes/No? Yes
```

Enter **Yes** as prompted to set the end sector.

If the following warning message is displayed, enter **Ignore** to ignore the performance warning. The warning message will not be displayed if the start sector with the optimal disk performance has been entered. In this example, **2048s** is one of such start sectors, so the system does not display the warning message.

```
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? Ignore
```

NOTE

Data will be lost if the following operations are performed:

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

Step 10 Enter **p** and press **Enter** to check whether the **/dev/xvdc1** partition has been re-created.

Information similar to the following is displayed:

```
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdb: 125829120s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start     End         Size        File system  Name  Flags
 1      2048s     125829086s  125827039s  ext4         opt
```

The **/dev/xvdc1** partition has been re-created.

Step 11 Enter **q** and press **Enter** to exit parted.

Step 12 Perform the following operations based on the file system of the disk:

- For the **ext3** or **ext4** file system

  a. Run the following command to check the correctness of the file system on **/dev/xvdc1**:

     e2fsck -f /dev/xvdc1

     Information similar to the following is displayed:

     ```
     [root@ecs-1120 linux]# e2fsck -f /dev/xvdb2
     e2fsck 1.42.9 (28-Dec-2013)
     Pass 1: Checking inodes, blocks, and sizes
     Pass 2: Checking directory structure
     Pass 3: Checking directory connectivity
     Pass 4: Checking reference counts
     Pass 5: Checking group summary information
     /dev/xvdc1: 11/655360 files (0.0% non-contiguous), 83137/2620928 blocks
     ```

  b. Run the following command to extend the size of the file system on **/dev/xvdc1**:

     resize2fs /dev/xvdc1

     Information similar to the following is displayed:

     ```
     [root@ecs-1120 linux]# resize2fs /dev/xvdc1
     resize2fs 1.42.9 (28-Dec-2013)
     Resizing the filesystem on /dev/xvdc1 to 15728379 (4k) blocks.
     The filesystem on /dev/xvdc1 is now 15728379 blocks long.
     ```

c. Run the following command to view the disk partition information after the partition re-creation:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0    60G  0 part
```

In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the additional 50 GB has been allocated to the **dev/xvdc1** partition.

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

mount /dev/xvdc1 /mnt/sdc

- For the **xfs** file system

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

mount /dev/xvdc1 /mnt/sdc

b. Run the following command to extend the size of the file system on **/dev/xvdc1**:

sudo xfs_growfs /dev/xvdc1

c. Run the following command to view the disk partition information after the partition re-creation:

lsblk

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0   80G  0 disk
├─xvda1 202:1    0   40G  0 part /
└─xvda2 202:2    0   40G  0 part /opt
xvdb    202:16   0  350G  0 disk
├─xvdb1 202:17   0  100G  0 part
└─xvdb2 202:18   0  200G  0 part
xvdc    202:32   0   60G  0 disk
└─xvdc1 202:33   0    60G  0 part
```

In the command output, the total capacity of the **/dev/xvdc** disk is 60 GB, in which the additional 50 GB has been allocated to the **dev/xvdc1** partition.

Step 13 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-1120 linux]# mount /dev/xvdc1 /mnt/sdc
[root@ecs-1120 linux]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/xvda1     ext4       43G  8.3G   33G  21% /
devtmpfs       devtmpfs  885M     0  885M   0% /dev
tmpfs          tmpfs     894M     0  894M   0% /dev/shm
tmpfs          tmpfs     894M   18M  877M   2% /run
tmpfs        tmpfs    894M    0  894M   0% /sys/fs/cgroup
tmpfs        tmpfs    179M    0  179M   0% /run/user/2000
tmpfs        tmpfs    179M    0  179M   0% /run/user/0
tmpfs        tmpfs    179M    0  179M   0% /run/user/1001
/dev/xvda2     ext4      43G   51M   40G   1% /opt
/dev/xvdc1     ext4      64G   55M   60G   1% /mnt/sdc
```

The **/dev/xvdc1** partition has been mounted on the **/mnt/sdc** directory.

----End

Setting Automatic Mounting at System Start

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

NOTE

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

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

blkid *Disk partition*

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

blkid /dev/xvdb1

Information similar to the following is displayed:

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

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

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

vi /etc/fstab

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

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

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

The preceding content is used for reference only. Add the information that is used in the environment. The parameters are described as follows:

- The first column indicates the partition UUID obtained in **Step 1**.
- The second column indicates the directory on which the partition is mounted. You can query the mount point using the **df -TH** command.
- The third column indicates the file system format of the partition. You can query the file system format using the **df -TH** command.
- The fourth column indicates the partition mount option. Normally, this parameter is set to **defaults**.
- The fifth column indicates the Linux dump backup option.

  - **0**: not use Linux dump backup. Normally, dump backup is not used, and you can set this parameter to **0**.
  - **1**: use Linux dump backup.
- The sixth column indicates the fsck option, that is, whether to use fsck to check the attached disk during startup.
  - **0**: not use fsck.
  - If the mount point is the root partition (**/**), this parameter must be set to **1**.

    When this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** so that 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.

----End

3.4.7 Extending Partitions and File Systems of a SCSI Data Disk on Linux (fdisk)

Scenarios

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

This topic uses SUSE Linux Enterprise Server 11 SP4 64bit to describe how to allocate the additional space of an attached SCSI data disk to a partition using fdisk.

Currently, a disk has been attached to a server, and the original disk capacity is 10 GB. In addition, a 10-GB space has been added to the disk on the management console, and the total disk capacity should be 20 GB. However, the additional space cannot be viewed on the server. The following operations guide you to view the additional space and extend the partition.

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

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

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

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

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.
- Re-create an existing partition (services will be interrupted).

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

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

NOTICE

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

Prerequisites

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

Re-creating an Existing Partition

The following procedure shows you how to make use of the additional space of a disk attached to a server by extending the **/dev/sda1** partition and mounting the partition on **/mnt/sdc**. During this process, services will be interrupted.

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

fdisk -l

Information similar to the following is displayed:

```
ecs-xen-02:/home/linux # fdisk -l

Disk /dev/xvda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00065c40

    Device Boot      Start         End      Blocks   Id  System
/dev/xvda1            2048    41945087    20971520   82  Linux swap / Solaris
/dev/xvda2   *    41945088    83892223    20973568   83  Linux
/dev/xvda3        83892224   209715199    62911488   83  Linux

Disk /dev/sda: 10.7 GB, 10737418240 bytes
64 heads, 32 sectors/track, 10240 cylinders, total 20971520 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2f1c057a
```

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1           2048    20971519    10484736   83  Linux
```

In the command output, 10 GB has been added to the **/dev/sda** data disk on the management console, and the total capacity should be 20 GB. However, the additional space is not included in the command output. In this case, run the following command to update the data disk capacity on the server:

Step 2 Run the following command to update the data disk capacity on the server:

echo 1 > /sys/class/scsi_device/*%d:%d:%d:%d*/device/rescan &

In the command, **%d:%d:%d:%d** indicates a folder in the **/sys/class/scsi_device/** directory and can be obtained using **ll /sys/class/scsi_device/**.

Information similar to the following is displayed: (**2:0:0:0** indicates the folder to be obtained.)

```
cs-xen-02:/sys/class/scsi_device # ll /sys/class/scsi_device/
total 0
lrwxrwxrwx 1 root root 0 Sep 26 11:37 2:0:0:0 -> ../../devices/xen/vscsi-2064/host2/target2:0:0/2:0:0:0/
scsi_device/2:0:0:0
```

Example command:

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

Step 3 After the disk capacity is updated, run the following command to view the disk partition information again:

fdisk -l

Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # fdisk -l

Disk /dev/xvda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders, total 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00065c40

   Device Boot      Start         End      Blocks   Id  System
/dev/xvda1            2048    41945087    20971520   82  Linux swap / Solaris
/dev/xvda2   *    41945088    83892223    20973568   83  Linux
/dev/xvda3        83892224   209715199    62911488   83  Linux

Disk /dev/sda: 21.5 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2f1c057a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1           2048    20971519    10484736   83  Linux
```

In the command output, the additional space has been added to the **/dev/sda** data disk, and the update succeeded. The **/dev/sda** disk has one partition only, **/dev/sda1**. Take note of start and end sectors of the **/dev/sda1** partition. These values will be used during the partition re-creation. The partition's start sector is **2048**, and its end sector is **20971519**.

Step 4 Run the following command to unmount the partition:

umount /mnt/sdc

Step 5 Run the following command to enter fdisk and enter **d** to delete the **/dev/sda1** partition:

fdisk /dev/sda1

Information similar to the following is displayed:

```
[ecs-xen-02:/sys/class/scsi_device # fdisk /dev/sda

Command (m for help): d
Selected partition 1

Command (m for help):
```

Step 6 Enter **n** and press **Enter** to create a new partition.

Entering **n** creates a new partition.

Information similar to the following is displayed:

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
```

There are two types of disk partitions:

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

Step 7 Ensure that the entered partition type is the same as the partition had before. In this example, a primary partition is used. Therefore, enter **p** and press **Enter** to create a primary partition.

Information similar to the following is displayed:

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

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

Step 8 Ensure that the entered partition number is the same as the partition had before. In this example, partition number **1** is used. Therefore, enter **1** and press **Enter**.

Information similar to the following is displayed:

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

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

NOTE

Data will be lost if the following operations are performed:

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

Step 9 Ensure that the entered start sector is the same as the partition had before. In this example, start sector **2048** is recorded in **Step 3**. Therefore, enter **2048** and press **Enter**.

Information similar to the following is displayed:

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

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

Step 10 Ensure that the entered end sector is larger than or equal to the end sector recorded in **Step 3**. In this example, the recorded end sector is **20971519**, and the default end sector is used. Therefore, enter **41943039** and press **Enter**.

Information similar to the following is displayed:

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

Command (m for help):
```

The partition is created.

Step 11 Enter **p** and press **Enter** to view details about the new partition.

Information similar to the following is displayed: (Details about the **/dev/sda1** partition are displayed.)

```
Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
64 heads, 32 sectors/track, 20480 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x2f1c057a

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1            2048    41943039    20970496   83  Linux
Command (m for help):
```

Step 12 Enter **w** and press **Enter** to write the changes to the partition table.

Information similar to the following is displayed: (The partition is successfully created.)

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

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

NOTE

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

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

- For the **ext3** or **ext4** file system

  a. Run the following command to check the correctness of the file system on **/dev/sda1**:

     e2fsck -f /dev/sda1

     Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # e2fsck -f /dev/sda1
e2fsck 1.41.9 (22-Aug-2009)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sda1: 11/655360 files (0.0% non-contiguous), 79663/2621184 blocks
```

  b. Run the following command to extend the size of the file system on **/dev/sda1**:

     resize2fs /dev/sda1

     Information similar to the following is displayed:

```
ecs-xen-02:/sys/class/scsi_device # resize2fs /dev/sda1
resize2fs 1.41.9 (22-Aug-2009)
Resizing the filesystem on /dev/sda1 to 5242624 (4k) blocks.
The filesystem on /dev/sda1 is now 5242624 blocks long.
```

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

mount /dev/sda1 /mnt/sdc

- For the **xfs** file system

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

mount /dev/sda1 /mnt/sdc

b. Run the following command to extend the size of the file system on **/dev/sda1**:

sudo xfs_growfs /dev/sda1

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

df -TH

----End

Setting Automatic Mounting at System Start

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

NOTE

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

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

blkid *Disk partition*

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

blkid /dev/xvdb1

Information similar to the following is displayed:

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

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

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

vi /etc/fstab

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

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

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

The preceding content is used for reference only. Add the information that is used in the environment. The parameters are described as follows:

- The first column indicates the partition UUID obtained in **Step 1**.
- The second column indicates the directory on which the partition is mounted. You can query the mount point using the **df -TH** command.
- The third column indicates the file system format of the partition. You can query the file system format using the **df -TH** command.
- The fourth column indicates the partition mount option. Normally, this parameter is set to **defaults**.
- The fifth column indicates the Linux dump backup option.
  - **0**: not use Linux dump backup. Normally, dump backup is not used, and you can set this parameter to **0**.
  - **1**: use Linux dump backup.
- The sixth column indicates the fsck option, that is, whether to use fsck to check the attached disk during startup.
  - **0**: not use fsck.
  - If the mount point is the root partition (**/**), this parameter must be set to **1**.

    When this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** so that 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.

----End

3.4.8 Extending Partitions and File Systems of a System Disk on Linux (fdisk)

Scenarios

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

This topic uses CentOS 7.4 64bit to describe how to allocate the additional system disk space to a partition using fdisk.

This document also describes how to create new partitions for system disks. See the following topics for more information:

- To create new partitions using fdisk in CentOS 7.0 64bit, see **Creating a New Partition**.
- To create new partitions using parted in CentOS 7.0 64bit, see **Creating a New Partition**.

Currently, a disk has been attached to a server, and the original disk capacity is 40 GB. In addition, a 40-GB space has been added to the disk on the management console, and the total disk capacity should be 80 GB. The following operations guide you to use the additional space and extend the partition.

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

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

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

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

  If the MBR partition style is used, ensure that the disk capacity does not exceed 2 TB and the number of partitions does not reach the upper limit after the expansion.
- Re-create an existing partition (services will be interrupted).

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

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

NOTICE

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

Prerequisites

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

Creating a New Partition

The following example shows you how to make use of the additional capacity of a system disk attached to a server by creating a new partition. The system disk already has the **/dev/vda1** partition, which is mounted on **/**. In the following operations, new partition **/dev/vda2** will be created and mounted on **/opt**. During this process, services are not interrupted.

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

fdisk -l

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# fdisk -l
```

```
Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f

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

In the command output, the capacity of the **dev/vda** system disk is 80 GB, in which the in-use **dev/vda1** partition takes 40 GB and the additional 40 GB has not been allocated.

Step 2 Run the following command to enter fdisk:

fdisk /dev/vda

Information similar to the following is displayed:

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

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


Command (m for help):
```

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

Information similar to the following is displayed:

```
Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
```

There are two types of disk partitions:

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

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

Information similar to the following is displayed:

```
Select (default p): p
Partition number (2-4, default 2):
```

Step 5 Enter the serial number of the primary partition and press **Enter**. Partition number **2** is used in this example. Therefore, enter **2** and press **Enter.**

Information similar to the following is displayed:

```
Partition number (2-4, default 2): 2
First sector (83886080-167772159, default 83886080):
```

Step 6 Enter the new partition's start sector, for example the default value, and press **Enter**.

The default start sector is used in this example.

Information similar to the following is displayed:

```
First sector (83886080-167772159, default 83886080):
Using default value 83886080
Last sector, +sectors or +size{K,M,G} (83886080-167772159,default 167772159):
```

Step 7 Enter the new partition's end sector and press **Enter**.

The default end sector is used in this example.

Information similar to the following is displayed:

```
Last sector, +sectors or +size{K,M,G} (83886080-167772159,
default 167772159):
Using default value 167772159
Partition 2 of type Linux and of size 40 GiB is set

Command (m for help):
```

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

Information similar to the following is displayed:

```
Command (m for help): p

Disk /dev/vda: 85.9 GB, 85899345920 bytes, 167772160 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x0008d18f

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1   *        2048    83886079    41942016   83  Linux
/dev/vda2        83886080   167772159    41943040   83  Linux
Command (m for help):
```

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

Information similar to the following is displayed:

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

Calling ioctl() to re-read partition table.

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

The partition is created.

NOTE

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

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

partprobe

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

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

mkfs -t ext4 /dev/vda2

NOTE

The procedure for setting the **xfs** file system is the same as that for the **ext3** or **ext4** file system. The command for creating the **xfs** file system is **mkfs -t xfs /dev/vda2**.

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# mkfs -t ext4 /dev/vda2
mke2fs 1.42.9 (28-Dec-2013)
```

```
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
2621440 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2157969408
320 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624

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

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

Step 12 Run the following command to mount the new partition on a space-demanding directory, for example **/opt**:

mount /dev/vda2 /opt

Information similar to the following is displayed:

```
[root@ecs-bab9 test]# mount /dev/vda2 /opt
[root@ecs-bab9 test]#
```

NOTE

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

Step 13 Run the following command to view the mount result:

df -TH

Information similar to the following is displayed:

```
[root@ecs-2220 ~]# df -TH
Filesystem     Type      Size  Used Avail Use% Mounted on
/dev/vda1      ext4       43G  2.0G   39G   5% /
devtmpfs       devtmpfs  509M     0  509M   0% /dev
tmpfs          tmpfs     520M     0  520M   0% /dev/shm
tmpfs          tmpfs     520M  7.2M  513M   2% /run
tmpfs          tmpfs     520M     0  520M   0% /sys/fs/cgroup
tmpfs          tmpfs     104M     0  104M   0% /run/user/0
/dev/vda2      ext4       43G   51M   40G   1% /opt
```

----End

Setting Automatic Mounting at System Start

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

NOTE

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

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

blkid *Disk partition*

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

blkid /dev/xvdb1

Information similar to the following is displayed:

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

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

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

vi /etc/fstab

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

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

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

The preceding content is used for reference only. Add the information that is used in the environment. The parameters are described as follows:

- The first column indicates the partition UUID obtained in **Step 1**.
- The second column indicates the directory on which the partition is mounted. You can query the mount point using the **df -TH** command.
- The third column indicates the file system format of the partition. You can query the file system format using the **df -TH** command.
- The fourth column indicates the partition mount option. Normally, this parameter is set to **defaults**.
- The fifth column indicates the Linux dump backup option.
  - **0**: not use Linux dump backup. Normally, dump backup is not used, and you can set this parameter to **0**.
  - **1**: use Linux dump backup.
- The sixth column indicates the fsck option, that is, whether to use fsck to check the attached disk during startup.
  - **0**: not use fsck.
  - If the mount point is the root partition (**/**), this parameter must be set to **1**.

    When this parameter is set to **1** for the root partition, this parameter for other partitions must start with **2** so that 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.

----End

3.5 Managing Encrypted Disks

Relationships Between Encrypted Disks and Backups

The encryption function can be used to encrypt system disks, data disks, and backups. The details are as follows:

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

For how to create encrypted disks, see **Create a Disk**.

Creating Encrypted Disks

Before you use the disk encryption function, KMS access rights need to be granted to EVS. If you have the Security Administrator permission, grant KMS access rights directly. If you do not have this permission, contact a user with the security administrator permissions to grant KMS access rights to EVS, then repeat the preceding operations.

For how to create encrypted disks, see **Create a Disk**.

Detaching Encrypted Disks

Before you detach a disk encrypted by a custom key, check whether the custom key is disabled or scheduled for deletion. If the custom key is unavailable, the disk can still be used, but normal read/write operations are not guaranteed permanently. If the disk is detached and then re-attached, re-attaching this disk will fail. In this case, do not detach the disk and restore the custom key status first.

The restoration method varies depending on the current CMK status. For details, see **Disk Encryption**.

If the custom key is available, the disk can be detached and re-attached, and data on the disk will not be lost.

For how to detach an encrypted disk, see **Detaching a Data Disk**.

3.6 Managing Shared 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 the shared disk to the ECSs in the same ECS group to improve service reliability.

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

NOTICE

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

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

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

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

Attaching Shared Disks

A common disk can only be attached to one server, whereas a shared disk can be attached to up to 16 servers.

Deleting Shared Disks

Because a shared disk can be attached to multiple servers, ensure that a shared disk is detached from all the servers before deletion.

For details about how to delete a shared disk, see **Deleting a Disk**.

Expanding Shared Disks

Shared disks must be expanded when they are in the **Available** state. For details, see **Expanding an Available Disk**.

3.7 Managing Backups

Scenarios

DSS disk backups are created using CBR.

This section describes how to configure a backup policy for a disk. With backup policies configured, data on DSS disks can be periodically backed up to improve data security.

> NOTE
>
> Backups can be created for disks only when the disks are in the **Available** or **In-use** state.

Purchasing 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**.

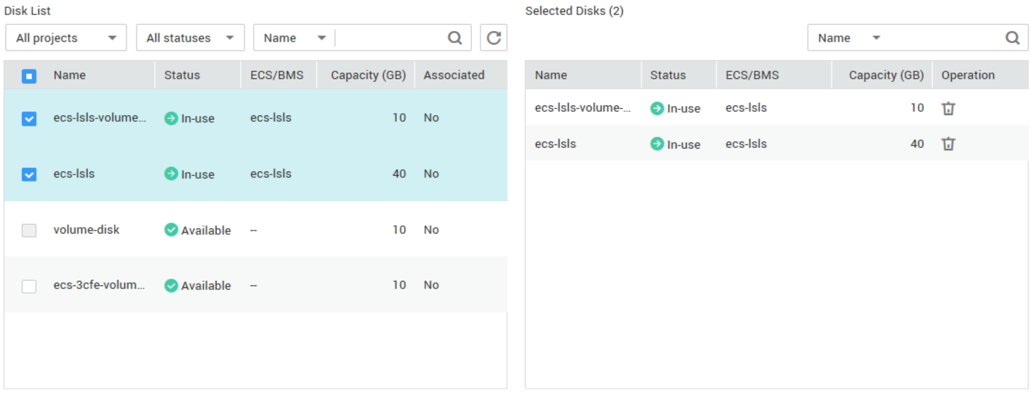
Step 2 In the upper right corner, click **Buy Disk Backup Vault**.

Step 3 Select a billing mode.

- Yearly/monthly is a prepaid billing mode. You are billed based on the subscription duration you specify. This mode provides lower prices and is ideal when the resource use duration is predictable.
- Pay-per-use is a postpaid billing mode. You are billed based on your resource usage. With this mode, you can increase or delete resources at any time. Fees are deducted from your account balance.

Step 4 (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.

Figure 3-11 Selecting disks



Disk List

■	Name	Status	ECS/BMS	Capacity (GB)	Associated
☑	ecs-lsls-volume...	In-use	ecs-lsls	10	No
☑	ecs-lsls	In-use	ecs-lsls	40	No
□	volume-disk	Available	--	10	No
□	ecs-3cfe-volum...	Available	--	10	No

Selected Disks (2)

Name	Status	ECS/BMS	Capacity (GB)	Operation
ecs-lsls-volume-...	In-use	ecs-lsls	10	🗑
ecs-lsls	In-use	ecs-lsls	40	🗑

NOTE

- Only **Available** and **In-use** disks can be selected.
- You can associate desired disks with the vault you created later if you skip this step.

Step 5 Specify a vault capacity ranging from the total sizes of disks to 10,485,760 GiB.

Step 6 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 7 If you have subscribed to Enterprise Project, add the vault to an existing enterprise project.

An enterprise project facilitates project-level management and grouping of cloud resources and users. The default project is **default**.

Step 8 (Optional) Add tags to the vault.

A tag consists of a key-value pair. Tags are used to identify, classify, and search for vaults. Vault tags are used to filter and manage vaults only. You can add up to 10 tags for a vault.

Table 3-2 describes the tag parameters.

Table 3-2 Tag parameters

Parameter	Description	Example Value
Key	A tag key of a vault must be unique. You can customize the key or select the key of an existing tag created in TMS. A tag key: ● Can contain 1 to 36 Unicode characters. ● Cannot be left blank, cannot start or end with spaces, or contain non-printable ASCII (0-31) characters or the following special characters: =*<>\,|/	Key_0001
Value	A tag value can be repetitive or left blank. A tag value: ● Can contain 0 to 43 Unicode characters. ● Can be an empty string, cannot start or end with spaces, or contain non-printable ASCII (0-31) characters or the following special characters: =*<>\,|/	Value_0001

Step 9 Specify a name for the vault.

The name can contain 1 to 64 characters including digits, letters, Chinese characters, underscores (_), and hyphens (-), for example, **vault-612c**.

NOTE

You can use the default name, which is in the format of **vault**_*xxxx*.

Step 10 Specify the subscription duration if you select **Yearly/Monthly** for **Billing Mode**. The validity period ranges from 1 month to 3 years.

Determine whether to enable auto renewal. If you select **Auto Renewal**:

- Your subscription will be renewed each month for monthly billing.
- Your subscription will be renewed each year for yearly billing.

Step 11 Click **Next**. Confirm the purchase details and continue.

Step 12 Complete the payment as prompted.

Step 13 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 **Vault Management**.

----End

A Change History

Released On	Description
2018-04-30	This issue is the first official release.