Dedicated Storage Service

User Guide

Issue 01

Date 2026-09-20

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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

Notice

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

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

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process.* For details about this process, visit the following web page:
https://www.huawei.com/en/psirt/vul-response-process
For vulnerability information, enterprise customers can visit the following web page:
https://securitybulletin.huawei.com/enterprise/en/security-advisory

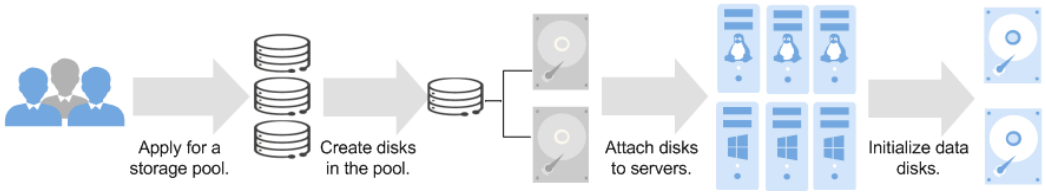
Contents

1 Process....1
2 Step 1: Apply for a Storage Pool....2
3 Step 2: Create a Disk....5
4 Step 3: Attach a Disk....11
4.1 Attaching a Non-Shared Disk....11
4.2 Attaching a Shared Disk....12
5 Initializing EVS Data Disks....14
5.1 Initialization Overview....14
5.2 Initializing a Linux Data Disk (Less Than or Equal to 2 TiB)....17
5.3 Initializing a Linux Data Disk (Greater Than 2 TiB)....23
5.4 Initializing a Windows Data Disk....29
6 Change History....37

1 Process

Figure 1-1 shows the basic process of using DSS.

Figure 1-1 Process of using DSS



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

2 Step 1: Apply for a Storage Pool

Scenarios

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

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

NOTE

DSS storage pools are constructed and deployed on demand. Contact your customer manager for detailed construction time.

Procedure

Step 1 Log in to the console.

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

Step 3 Choose **Storage** > **Dedicated Distributed Storage Service** to access the **Dedicated Distributed Storage Service** console.

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

Configure the basic information about the storage pool by referring to **Table 2-1**.

Figure 2-1 Apply for Storage Pool

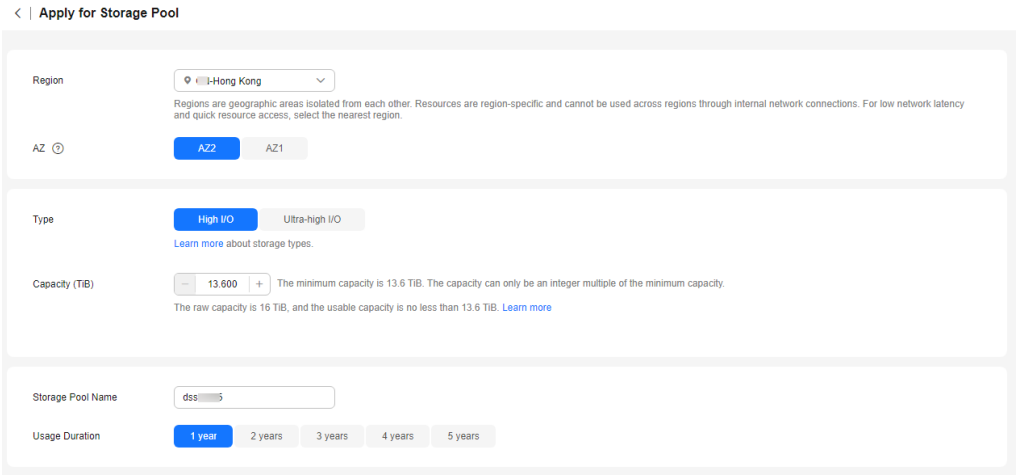


Table 2-1 Parameter description

Parameter	Description
Region	The region where the storage pool belongs.
AZ	The AZ where the storage pool belongs.
Type	• High I/O • Ultra-high I/O
Capacity (TB)	The storage pool capacity you requested. **Storage pool capacity description**
Storage Pool Name	The storage pool name, which can contain a maximum of 64 characters. The name can contain only letters, digits, underscores (_), and hyphens (-).
Usage Duration	You can choose 1 year to 5 years for the purchased duration.

After the configuration is complete, the details about the storage pool specifications are displayed.

Step 5 Click **Apply Now**, and the page for you to confirm the specifications is displayed.

You can confirm the details about the storage pool specifications.

- If you do not need to modify the specifications, click **Submit Application** and go to **Step 6**.
- If you need to modify the specifications, click **Previous** to modify parameters.

Step 6 Check the application result.

- On the displayed page, you can click **View Review Status** to view the order details.

- Or, click **Back to Storage Pool List** to return to the storage pool list.

Step 7 After the order is approved and paid, view the storage pool status.

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

Step 8 You can perform the following operations for your storage pool in the **Operation** column:

- Expand the storage pool capacity. For details, see **Expanding a Storage Pool**.
- Create disks. For details, see **Creating a Disk**.

----End

3 Step 2: Create a Disk

Scenarios

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

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

Notes

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

Procedure

Step 1 Log in to the console.

Step 2 Choose **Storage** > DSS to go to the **Dedicated Distributed Storage Service** console.

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

The disk list page is displayed.

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

Step 5 Configure the parameters based on **Table 3-1**.

Table 3-1 Parameter description

Name	Sub-Parameter	Description	Example Value
Region	-	Mandatory The region where the tenant belongs.	-

Name	Sub-Parameter	Description	Example Value
AZ	-	Mandatory The availability zone (AZ) where you want to create the disk. **NOTE** • Disks can only be attached to the servers in the same AZ. • The AZ of a disk cannot be changed after the disk has been created.	-
Disk Specifications	Disk Type	Mandatory Select a storage pool from the drop-down list. The disk type is the same as the storage pool type.	Ultra-high I/O
	Capacity (GB)	Mandatory The capacity of the disk. The capacity range is from 10 GB to the available capacity displayed on the page.	100 GB
	Select Data Source • Create from backup • Create from image	Optional • If you choose **Create from backup**, the backup data is used to create the disk. Click **Select Data Source** and choose **Create from backup**. On the displayed page, select the target backup and click **OK**. **NOTE** – A single backup cannot be used for concurrent disk creation operations. For example, if you are creating disk A from a backup, this backup can be used to create another disk only after disk A has been created. – If a disk is created from a backup of a system disk, the new disk can be used as a data disk only. • If you choose **Create from image**, the image data is used to create the disk. Click **Select Data Source** and choose **Create from image**. On the displayed page, select the target image and click **OK**. **NOTE** – The device type of the new disk is the same as that of the image's source disk. – The encryption attribute of the new disk is the same as that of the image's source disk.	• Create from backup: autobackup-001

Name	Sub-Parameter	Description	Example Value
Automatic Backup	-	CBR lets you back up disks and ECSs and use the backups to restore data. After you set automatic backup, the system will associate the disk with the backup vault and apply the selected policy to the vault to periodically back up the disk. ● Do not use: Skip this configuration if CBR is not required. If you still need to set backup protection after a disk is successfully purchased, log in to the CBR console, locate the target vault, and bind the disk to the vault. ● Use existing 1. Select an existing cloud backup vault from the drop-down list. 2. Select a backup policy from the drop-down list, or log in to the CBR console and configure a desired one. ● Buy new: 1. Set the name of the cloud backup vault, which is a character string consisting of 1 to 64 characters, including letters, digits, underscores (_), and hyphens (-), for example, **vault-f61e**. The default naming rule is **vault**_*xxxx*. 2. Enter the vault capacity, which is required for backing up the disk. The vault capacity cannot be less than the size of the disk to be backed up. The value ranges from the disk size to 10,485,760 in the unit of GB. 3. Select a backup policy from the drop-down list, or log in to the CBR console and configure a desired one.	-

Name	Sub-Parameter	Description	Example Value
More	Advanced settings • Share • SCSI • Encryption	Optional • Share: If you select **Share**, a shared disk is created. A shared disk can be attached to up to 16 servers. If you do not select **Share**, a non-shared disk is created, and the disk can be attached to one server only. When you select both **SCSI** and **Share**, a shared SCSI disk is created. **NOTE** The sharing attribute of a disk cannot be changed after the disk has been created. • SCSI: If you select **SCSI**, a SCSI disk is created. Such disks allow the server OS to directly access the underlying storage media and send SCSI commands to the disks. If you do not select **SCSI**, a VBD disk is created. That said, the disk device type is VBD, the default device type. **NOTE** The device type of a disk cannot be changed after the disk has been created. • Encryption: The encryption function on this console is used for data disk encryption only. System disk encryption relies on images. For details, see the *Image Management Service User Guide*. To use the disk encryption function, select **Encryption**. The displayed dialog box contains the following parameters: – Create Agency An agency is a trust relationship between two tenants or services. A tenant can create an agency to grant resource access rights to another tenant or service. If the KMS access rights are not granted to EVS, the **Create Agency** dialog box will be displayed. Otherwise, it will not be displayed. Click **Yes** to grant the KMS access rights to EVS. After the rights have	-

Name	Sub-Parameter	Description	Example Value
		been granted, EVS can obtain KMS keys to encrypt or decrypt disks. After the KMS access rights have been granted, follow-up operations do not require the rights to be granted again. – KMS Key Name **KMS Key Name** is displayed only after the KMS access rights have been granted. **KMS Key Name** is the identifier of the key, and you can use **KMS Key Name** to specify the KMS key that is to be used for encryption. You can select one of the following keys: Default Key: After the KMS access rights have been granted to EVS, the system automatically creates a Default Key **evs/default**. Custom keys: Existing or newly created keys. For details, see **Key Management Service** > **Creating a CMK** in the *Data Encryption Workshop User Guide*. **NOTE** – Before you use the disk encryption function, KMS access rights need to be granted to EVS. If you have the right to grant permissions, grant KMS access rights to EVS directly. If you do not have this permission, contact a user with the security administrator permissions to grant the KMS access rights to EVS, then repeat the preceding operations. – The encryption attribute of a disk cannot be changed after the disk is created.	

Name	Sub-Parameter	Description	Example Value
Disk Name	-	Mandatory ● If you create disks one by one, this parameter value is used as the actual disk name. The name can contain a maximum of 64 characters. ● If you create disks in a batch, this parameter value is used as the prefix of disk names, and one disk name will be composed of this parameter value and a four-digit number. The name can contain a maximum of 59 characters.	For example, if you create two disks and set **volume** for **Disk Name**, the disk names will be **volume-0001** and **volume-0002**.
Quantity	-	Optional The number of disks to be created. The default value is set to **1**, which means only one disk will be created. Currently, you can create up to 100 disks at a time. **NOTE** ● If the disk is created from a backup, batch creation is not possible, and this parameter must be set to **1**. ● If the disk is created from a snapshot, batch creation is not possible, and this parameter must be set to **1**.	1

Step 6 Click **Next**.

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

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

----End

4 Step 3: Attach a Disk

4.1 Attaching a Non-Shared Disk

Scenarios

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

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

NOTE

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

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

Procedure

Step 1 Log in to the console.

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

The **Dedicated Distributed Storage Service** console is displayed.

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

The disk list page is displayed.

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

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

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

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

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

Step 6 Initialize the disk.

After the disk has been attached, the disk can only be used after you have initialized it on the server. For details, see **Initializing EVS Data Disks**.

----End

4.2 Attaching a Shared Disk

Scenarios

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

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

Procedure

Step 1 Log in to the console.

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

The **Dedicated Distributed Storage Service** console is displayed.

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

The disk list page is displayed.

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

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

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

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 the servers. When the disk status changes to **In-use**, the disk is successfully attached.

NOTICE

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

----End

5 Initializing EVS Data Disks

5.1 Initialization Overview

Scenarios

- **System disk**

  When a server is created, a system disk is automatically initialized with Master Boot Record (MBR).

- **Newly created empty data disk**

  After you attach a newly created empty data disk to a server, you must initialize the disk including creating partitions, creating file systems, and mounting the partitions before you can use the disk.

  – If a data disk is created together with a server, the system automatically attaches it to the server. You only need to initialize it to make it available for use.

  – If a data disk is created explicitly, you need to first attach it to a server and then initialize it.

  For detailed operation instructions, see **Table 5-1**.

- **Existing data disk**

  An existing data disk is a disk created from a snapshot, a backup, or an image, or a disk detached from another server.

  – You can choose not to re-partition the disk, but use the existing partitions.

    ▪ In Linux, **create new mount points and mount the partitions**, and **configure auto mount at system start**.

    ▪ In Windows, no further action is required. You can simply use the existing partitions.

  – You can also re-initialize the data disk.

    Re-partitioning a disk will erase all the existing data on the disk, so you are advised to use snapshots to back up the disk data first.

- In Linux, unmount the partitions, delete them (by running **fdisk** *<disk-name>*, entering **d** and the partition number, and entering **w**), and then re-initialize the disk.
- In Windows, delete the partitions (using the volume deletion tool) and then re-initialize the disk.

For detailed initialization operations, see **Table 5-1**.

Impact on the System

- An initialization operation includes partitioning, which deletes all the data on the disk.
- If you change the partition style of a disk, data on the disk will be erased. Select an appropriate partition style when initializing disks.
- Initializing a disk does not delete the snapshots created for the disk, so you can still use snapshots to roll back data to the source disk after the disk is initialized.

Operation Instructions

Major initialization steps include **creating partitions and file systems, mounting partitions, and configuring auto mount at system startup**. In Linux, you can choose different partition styles based on your disk capacity. For details, see **Table 5-2**.

The maximum disk size that MBR supports is 2 TiB, and that GPT supports is 18 EiB. If your disk is greater than 2 TiB or you plan to expand it to over 2 TiB later, use GPT when initializing disks.

Table 5-1 Disk initialization instructions

Disk Capacity	Partition Style	OS	Reference
Capacity ≤ 2 TiB	Guid Partition Table (GPT) or MBR	Linux	**Initializing a Linux Data Disk (Less Than or Equal to 2 TiB)**
		Windows	**Initializing a Windows Data Disk**
Capacity > 2 TiB	GPT	Linux	**Initializing a Linux Data Disk (Greater Than 2 TiB)**
		Windows	**Initializing a Windows Data Disk**

Partition Styles

Common disk partition styles include MBR and GPT. In Linux, you can choose different partition styles based on your disk capacity, as described in **Partition Styles**.

Table 5-2 Partition styles

Partition Style	Max. Disk Size Supported	Max. Number of Partitions Supported	Linux Partitioning Tool	Reference
MBR	2 TiB	MBR partitions include primary partitions and extended partitions. A maximum of four primary partitions are supported. If you need more partitions, create one extended partition. The following is an example: • Four primary partitions • Three primary partitions and one extended partition The extended partition must be divided into logical partitions before use. For example, to create six partitions, you can create them in the following two ways: • Three primary partitions and one extended partition, with the extended partition divided into three logical partitions • One primary partition and one extended partition, with the extended partition divided into five logical partitions For how to create logical partitions, see **Creating a Logical Volume Using LVM**.	• fdisk • parted	**Initializing a Linux Data Disk (Less Than or Equal to 2 TiB)** **Initializing a Windows Data Disk**
GPT	18 EiB 1 EiB = 1048576 TiB	Not limited GPT partitions are not categorized.	parted	**Initializing a Linux Data Disk (Greater Than 2 TiB)** **Initializing a Windows Data Disk**

Common Disk Device Names in Linux

The following lists some common disk device names in Linux:

- System disk: /dev/vda, /dev/sda, and /dev/xvda
- Data disk: /dev/vd[b-z], /dev/sd[b-z], and /dev/xvd[b-z]
  Examples: /dev/vdb, /dev/vdc, /dev/sdb, /dev/sdc, /dev/xvdb, and /dev/xvdc
- Partition: /dev/vd[a-z][digit], /dev/sd[a-z][digit], /dev/xvd[a-z][digit]
  Examples: /dev/vda1, /dev/vda2, /dev/vdb1, /dev/vdb2

5.2 Initializing a Linux Data Disk (Less Than or Equal to 2 TiB)

Scenarios

After a newly purchased data disk is attached to a server, you must log in to the server and initialize the disk before you can use the disk. This section describes how to initialize a Linux data disk. The operations may vary depending on the server OS.

The maximum disk size that MBR supports is 2 TiB, and that GPT supports is 18 EiB. If your disk capacity is less than 2 TiB, you can use either the MBR or GPT partition style. However, if your disk is greater than 2 TiB or you plan to expand it to over 2 TiB later, use **GPT** when initializing disks.

Table 5-3 Initialization instructions

Operation	OS Requirements	Partition Style	Common File Systems	Example Configuration
Initializing a Data Disk Using fdisk	None	MBR	ext* (such as ext2, ext3, and ext4), xfs, and btrfs	• OS: CentOS 7.4 64-bit • Partitioning tool: fdisk • Device name: /dev/vdb • File system format: ext4 • Mount points: /mnt/sdc and /mnt/sdd • Partition 1: /dev/vdb1 – Size: 40 GiB – Partition style: MBR • Partition 2: /dev/vdb2 – Size: 60 GiB – Partition style: MBR
Initializing a Data Disk Using parted	None	• GPT • MBR		See **Initializing a Data Disk Using parted**.

Prerequisites

The disk has been attached to a server. For how to attach disks, see **Step 3: Attach a Disk**.

Constraints

- A disk created from a data source does not need to be initialized. Such a disk contains the source data in the beginning. Initializing the disk may clear the initial data on it. If you need to re-initialize the disk, you are advised to back up the disk data first. For how to back up data using CBR, see "Managing Backups." For how to back up data using snapshots, see "Creating a Snapshot."
- Initializing a disk does not delete the snapshots created for the disk, so you can still use snapshots to roll back data to the source disk after the disk is initialized.

Initializing a Data Disk Using fdisk

The following example shows you how to use fdisk to create two primary partitions (/dev/vdb1: 40 GiB; /dev/vdb2: 60 GiB) on the /dev/vdb data disk and set the partition style to MBR.

Step 1 Log in to the server as user **root**.

For how to log in to an ECS, see **How Do I Log In to My ECS?**

For how to log in to a BMS, see **Linux BMS Login Methods**.

Step 2 Create partitions. In this example, create two primary partitions, **/dev/vdb1** and **/dev/vdb2** for data disk **/dev/vdb**.

1. Check that the capacity of the **/dev/vdb** data disk is 100 GiB.

   lsblk

   ```
   [root@ecs-centos76 ~]# lsblk
   NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
   vda    253:0    0   40G  0 disk
   └─vda1 253:1    0   40G  0 part /
   vdb    253:16   0  100G 0 disk
   ```

   The command output shows that there are two disks. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk, whose 100 GiB is not partitioned.

2. Create the first primary partition **/dev/vdb1** for **/dev/vdb**.

   fdisk */dev/vdb*

   n

   p

   1

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

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

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

   Command (m for help): n
   Partition type:
   ```



```
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
Select (default p): p
Partition number (1-4, default 1): 1
```

- Entering **p** for **Partition type** creates a primary partition, and entering **e** creates an extended partition. The default value is **p**.
- **Partition number** indicates the partition serial number. Enter a value ranging from **1** to **4**.

3. Set **First sector** to **2048** and **Last sector** to **83886079** for partition **/dev/vdb1** (40 GiB). For how to calculate the values, see **Table 5-4**.

```
First sector (2048-209715199, default 2048): 2048
Last sector, +sectors or +size{K,M,G} (2048-209715199, default 209715199):83886079
Partition 1 of type Linux and of size 40 GB is set
```

Table 5-4 First and last sectors in this example are calculated as follows

Sector	/dev/vdb1 (40 GiB)	/dev/vdb2 (60 GiB)	Formula for Calculating the Value of sectors
First sector	2048 (The first sector of the **/dev/vdb** data disk is used.)	**Last sector of /dev/vdb1 + 1** = 83886079 + 1 = 83886080	**Value of sectors** = Capacity × 1073741824/512
Last sector	**Value of sectors – 1** = (40 × 1073741824/512) – 1 = 83886079	**First sector + Value of sectors – 1** = 83886080 + (60 × 1073741824/512) – 1 = 209715199	

4. Create the second primary partition **/dev/vdb2** for **/dev/vdb**.

n

p

2

```
Command (m for help): n
Partition type:
  p   primary (0 primary, 0 extended, 4 free)
  e   extended
Select (default p): p
Partition number (1-4, default 2): 2
```

5. Set the **First sector** to **83886080** and **Last sector** to **209715199** for partition **/dev/vdb2**.

```
First sector (83886080-209715199, default 83886080): 83886080
Last sector, +sectors or +size{K,M,G} (83886080-209715199, default 209715199):209715199
Partition 2 of type Linux and of size 60 GB is set
```

6. Check the sizes and partition styles of the new partitions.

p

```
Command (m for help): p

Disk /dev/vdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
```



```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x994727e5

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048    83886079    41942016   83  Linux
/dev/vdb2        83886080   209715199    62914560   83  Linux

Command (m for help):
```

Disk label type: dos indicates the MBR partition style.

NOTE

In case that you want to discard the changes made before, you can exit fdisk by entering **q** and press **Enter**. Then, re-create the partitions by referring to step 1 and step 2.

7. Write the changes to the partition table.

   w

   ```
   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

   If error message **-bash: partprobe: command not found** is returned, the system cannot identify the command. In this case, run **yum install -y parted** to install the command. Then, run the command again.

8. Synchronize the new partition table to the OS.

   partprobe

Step 3 Create file systems.

1. Create ext4 file systems for partitions **/dev/vdb1** (40 GiB) and **/dev/vdb2** (60 GiB).

   mkfs -t ext4 *<u>/dev/vdb1</u>*

   mkfs -t ext4 *<u>/dev/vdb2</u>*

   NOTE

   - **mkfs -t** *<file-system-format> <disk-partition-name>*: To create an xfs file system, the command is **mkfs -t xfs** *<disk-partition-name>*. To create a btrfs file system, the command is **mkfs -t btrfs** *<disk-partition-name>*.
   - It takes some time to create file systems. Do not exit before the system returns the following information:

   ```
   [root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb1
   mke2fs 1.42.9 (28-Dec-2013)
   Filesystem label=
   OS type: Linux
   Block size=4096 (log=2)
   Fragment size=4096 (log=2)
   Stride=0 blocks, Stripe width=0 blocks
   2621440 inodes, 10485504 blocks
   524275 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
```

2. Check whether the file system format is ext4.

   blkid *<u>/dev/vdb</u>*

```
[root@ecs-test-0001 ~]# blkid /dev/vdb
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
/dev/vdb2: UUID="0d6769k2-1745-9dsf-453d-hgd0b34267dj" TYPE="ext4"
```

Step 4 Create directories (mount points) and mount the new partitions on the created mount points.

1. Mount **/dev/vdb1** on **/mnt/sdc**.

   mkdir -p *<u>/mnt/sdc</u>*

   mount *<u>/dev/vdb1</u> <u>/mnt/sdc</u>*

2. Mount **/dev/vdb2** on **/mnt/sdd**.

   mkdir -p *<u>/mnt/sdd</u>*

   mount *<u>/dev/vdb2</u> <u>/mnt/sdd</u>*

3. View the mount results.

   lsblk

```
[root@ecs-test-0001 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├vda1 253:1    0  40G  0 part /
vdb    253:16   0 100G  0 disk
├vdb1 253:17   0  40G  0 part /mnt/sdc
├vdb2 253:18   0  60G  0 part /mnt/sdd
```

You should now see that partitions **/dev/vdb1** and **/dev/vdb2** are mounted on **/mnt/sdc** and **/mnt/sdd**.

Step 5 (Optional) Use the partition UUIDs to configure auto mount at startup.

UUIDs are the unique character strings for identifying partitions in Linux. Mounts become invalid after a system reboot. You can configure auto mount at startup by adding information of the new partitions into the **/etc/fstab** file.

NOTE

- You are advised not to use device names to identify disks in the **/etc/fstab** file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a server stops or starts. This can even prevent your server from booting up.
- This operation will not affect the existing data on the ECS.

1. Query the partition UUIDs.

   blkid *<u>/dev/vdb1</u>*

   blkid *<u>/dev/vdb2</u>*

```
[root@ecs-test-0001 ~]# blkid /dev/vdb
/dev/vdb1: UUID="0b3040e2-1367-4abb-841d-ddb0b92693df" TYPE="ext4"
/dev/vdb2: UUID="0d6769k2-1745-9dsf-453d-hgd0b34267dj" TYPE="ext4"
```


Take note of the partition UUIDs, which will be used in the next step. In this example, the partition UUIDs are **0b3040e2-1367-4abb-841d-ddb0b92693df** and **0d6769k2-1745-9dsf-453d-hgd0b34267dj**.

2. Configure auto mount at startup.

   vi /etc/fstab

   Press **i** to enter the editing mode, move the cursor to the end of the file, press **Enter**, and add the following content:

   ```
   UUID=0b3040e2-1367-4abb-841d-ddb0b92693df   /mnt/sdc    ext4   defaults    0   2
   UUID=0d6769k2-1745-9dsf-453d-hgd0b34267dj   /mnt/sdd    ext4   defaults    0   2
   ```

   Press **Esc**, enter **:wq**, and press **Enter** to save the settings and exit the vi editor.

   Table 5-5 Content description

Example Value	Description
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df	The UUID of the partition.
/mnt/sdc	The mount point of the partition.
ext4	The file system format of the partition.
defaults	The partition mount option. Normally, this parameter is set to **defaults**.
0	The Linux dump backup option. – **0**: Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**. – **1**: Linux dump backup is used.
2	The fsck option, which means whether to use fsck to check the disk during startup. – **2**: The check starts from the partitions whose mount points are non-root directories. **/** is the root directory. – **1**: The check starts from the partitions whose mount points are root directories. – **0**: The fsck option is not used.

Step 6 (Optional) Verify that auto mount takes effect.

You can restart the server to check whether auto mount takes effect. Alternatively, you can perform the following steps to simulate auto mount.

1. To verify auto mount, unmount the partitions first.

   umount ***/dev/vdb1***

   umount ***/dev/vdb2***

2. Reload all the content in the **/etc/fstab** file. **/etc/fstab** is a static file system table that contains the list of file systems that need to be automatically mounted during system startup.

mount -a

3. Query the file system mount information.

mount | grep *[/mnt/sdc](#)*

mount | grep *[/mnt/sdd](#)*

If information similar to the following is displayed, auto mount has taken effect:

```
root@ecs-test-0001 ~]# mount | grep /mnt/sdc
/dev/vdb1 on /mnt/sdc type ext4 (rw,relatime,data=ordered)
root@ecs-test-0001 ~]# mount | grep /mnt/sdd
/dev/vdb2 on /mnt/sdd type ext4 (rw,relatime,data=ordered)
```

----End

Initializing a Data Disk Using parted

You can use parted to create either MBR or GPT partitions. The only difference is that the command used to set the partition style is different. All other operations are the same. For the initialization instructions with parted, see **Initializing a Data Disk Using parted**.

Commands for setting partition styles:

MBR:

```
mklabel msdos
unit s
p
```

GPT:

```
mklabel gpt
unit s
p
```

5.3 Initializing a Linux Data Disk (Greater Than 2 TiB)

Scenarios

After a newly purchased data disk is attached to a server, you must log in to the server and initialize the disk before you can use the disk.

When your disk is greater than 2 TiB, you can only use the parted tool and create GPT partitions. The initialization operations may vary depending on the server OS. Perform initialization operations based on your server OS. To learn about the differences between MBR and GPT, see **Partition Styles**.

Operation	Partition Style	OS Requirements	Common File Systems	Partitioning Tool	Example Configuration
Initializing a Data Disk Using parted	GPT	None	ext* (such as ext2, ext3, and ext4), xfs, and btrfs	parted	• OS: CentOS 7.4 64-bit • Device name: /dev/vdb • File system format: ext4 • Mount point: /mnt/sdc • Partition name: /dev/vdb1 • Partition style: GPT • Size: 3 TiB

Prerequisites

The disk has been attached to a server. For how to attach disks, see **Step 3: Attach a Disk**.

Constraints

- A disk created from a data source does not need to be initialized. Such a disk contains the source data in the beginning. Initializing the disk may clear the initial data on it. If you need to re-initialize the disk, you are advised to back up the disk data first. For how to back up data using CBR, see "Managing Backups." For how to back up data using snapshots, see "Creating a Snapshot."
- Initializing a disk does not delete the snapshots created for the disk, so you can still use snapshots to roll back data to the source disk after the disk is initialized.

Initializing a Data Disk Using parted

Major initialization steps include creating partitions and file systems, mounting partitions, and configuring auto mount at system startup.

The following example shows you how to use parted to create a GPT partition on the /dev/vdb data disk.

Step 1 Log in to the server as user **root**.

For how to log in to an ECS, see **How Do I Log In to My ECS?**

For how to log in to a BMS, see **Linux BMS Login Methods**.

Step 2 Create partitions. In this example, create the **/dev/vdb1** partition on data disk **/dev/vdb**.

1. Check that the capacity of the **/dev/vdb** data disk is 3 TiB.

lsblk

```
[root@ecs-centos76 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├vda1 253:1    0  40G  0 part /
vdb    253:16   0   3T  0 disk
```

The command output shows that there are two disks. **/dev/vda** is the system disk, and **/dev/vdb** is the new data disk, whose 3 TiB is not partitioned.

2. Create a partition. Replace **/dev/vdb** in the command with your desired disk name.

   parted ***/dev/vdb***

   p

```
[root@ecs-centos74 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Error: /dev/vdb: unrecognised disk label
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 3299GB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
(parted)
```

   Partition Table: unknown means that no partition style is set for the new disk.

   NOTE

   If error message **-bash: parted: command not found** is returned, the system cannot identify the command. In this case, run **yum install -y parted** to install the command. Then, run the command again.

3. Set the partition style to GPT.

   mklabel *****gpt*

   unit s

   p

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

Number  Start  End  Size  File system  Name  Flags

(parted)
```


NOTE

- **mklabel** *<disk-partition-style>*: If your disk capacity is less than or equal to 2 TiB and you want to use parted to create an MBR partition, run the **mklabel msdos** command.
- **unit s**: This command sets the measurement unit of the disk to sector.
- If you change the partition style of a disk, data on the disk will be erased. Select an appropriate partition style when initializing disks.
- **The partition style (MBR or GPT) set here will apply to all subsequent partitions created on this disk. When you create partitions on this disk later, you do not need to perform this step again.**

4. Set the partition name to **/dev/vdb1** and allocate all the storage space to this partition.

   mkpart ***/dev/vdb1 2048s 100%***

   p

   NOTE

   - **mkpart** *<partition-name> <first-sector-value> <last-sector-value>*: In the example command, **2048s** is the start sector value, and **100%** indicates to allocate 100% of the disk space to the **/dev/vdb1** partition.
   - If you want to allocate the data disk capacity to two or more partitions, calculate the first and last sectors of the partitions based on the method provided in **Step 2**.

```
(parted) mkpart /dev/vdb1 2048s 100%
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 6442450944s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start       End          Size         File system  Name         Flags
 1      2048s       6442448895s  6442446848s               /dev/vdb1
```

5. Enter **q** and press **Enter**. Then run **lsblk** to view the new partition **/dev/vdb1**.

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

Step 3 Create file systems.

1. In this example, create an ext4 file system on the **/dev/vdb1** partition.

   mkfs -t ext4 ***/dev/vdb1***

   NOTE

   - **mkfs -t** *<file-system-format> <disk-partition-name>*: To create an xfs file system, run the **mkfs -t xfs** *<disk-partition-name>* command. To create a btrfs file system, run the **mkfs -t btrfs** *<disk-partition-name>* command.
   - It takes some time to create file systems. Do not exit until the system returns the following information:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
```



```
201326592 inodes, 805305856 blocks
40265292 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2952790016
24576 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872, 71663616, 78675968,
        102400000, 214990848, 512000000, 550731776, 644972544

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
[root@ecs-test-0001 ~]#
```

2. After the file system is created, run the following commands to check the file system format:

   parted *<data disk>*

   p

```
[root@ecs-test-0001 ~]# parted /dev/vdb
GNU Parted 3.1
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 3299GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name        Flags
 1      1049kB  3299GB  3299GB  ext4         /dev/vdb1

(parted) q
[root@ecs-test-0001 ~]#
```

   If **ext4** is displayed under **File system**, the creation is successful.

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

Step 4 Create a directory (mount point) and mount the new partition on the created mount point.

mkdir -p *<mount point>*

mount *<data disk>1 <mount point>*

lsblk

```
[root@ecs-test-0001 ~]# lsblk
NAME   MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda    253:0    0  40G  0 disk
├─vda1 253:1    0  40G  0 part /
vdb    253:16   0   3T 0 disk
├─vdb1 253:17   0   3T 0 part /mnt/sdc
```

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

Step 5 (Optional) Use the partition UUID to configure auto mount at startup.

UUIDs are the unique character strings for identifying partitions in Linux. Mounts become invalid after a system reboot. You can configure auto mount at startup by adding information of the new partitions into the **/etc/fstab** file.

NOTE

- You are advised not to use device names to identify disks in the **/etc/fstab** file because device names are assigned dynamically and may change (for example, from **/dev/vdb1** to **/dev/vdb2**) after a server stop or start. This can even prevent your server from booting up.
- This operation will not affect the existing data on the ECS.

1. Query the partition UUID.

   blkid ***/dev/vdb1***

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

   Take note of the partition UUID, which will be used in the next step. In this example, the UUID of the **/dev/vdb1** partition is **0b3040e2-1367-4abb-841d-ddb0b92693df**.

2. Configure auto mount at startup.

   vi /etc/fstab

   Press **i** to enter the editing mode, move the cursor to the end of the file, press **Enter**, and add the following content:

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

   Press **Esc**, enter **:wq**, and press **Enter** to save the settings and exit the vi editor.

   Table 5-6 Content description

Example Value	Description
UUID=0b3040e2-1367-4abb-841d-ddb0b92693df	The UUID of the partition.
/mnt/sdc	The mount point of the partition.
ext4	The file system format of the partition.
defaults	The partition mount option. Normally, this parameter is set to **defaults**.
0	– The Linux dump backup option. ▪ **0**: Linux dump backup is not used. Usually, dump backup is not used, and you can set this parameter to **0**. ▪ **1**: Linux dump backup is used.

Example Value	Description
2	– The fsck option, which means whether to use fsck to check the disk during startup. ▪ **2**: The check starts from the partitions whose mount points are non-root directories. **/** is the root directory. ▪ **1**: The check starts from the partitions whose mount points are root directories. ▪ **0**: The fsck option is not used.

Step 6 (Optional) Verify that auto mount takes effect.

You can restart the server to check whether auto mount takes effect. Alternatively, you can perform the following steps to simulate auto mount.

1. To verify auto mount, unmount the partition first.

umount ***/dev/vdb1***

2. Reload all the content in the **/etc/fstab** file. **/etc/fstab** is a static file system table that contains the list of file systems that need to be automatically mounted during system startup.

mount -a

3. Query the file system mount information.

mount | grep ***/mnt/sdc***

If information similar to the following is displayed, auto mount has taken effect:

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

----End

5.4 Initializing a Windows Data Disk

Scenarios

This section uses the example configurations below to describe how to use Disk Management Tool or a script to initialize a Windows data disk. The initialization operations may vary depending on the server OS. Perform initialization operations based on your server OS.

Initialization Method	Partition Style	Example Configuration
Initializing a Data Disk Manually	• GPT • MBR	• Version: Windows Server 2019 Standard (64-bit) • Disk name: Disk 1 • Size: 100 GiB • After the initialization: – Partition name: New volume (D:) – Partition style: GPT – File system format: NTFS

Prerequisites

The disk has been attached to a server. For how to attach disks, see **Step 3: Attach a Disk**.

Constraints

- A disk created from a data source does not need to be initialized. Such a disk contains the source data in the beginning. Initializing the disk may clear the initial data on it. If you need to re-initialize the disk, you are advised to back up the disk data first. For how to back up data using CBR, see "Managing Backups." For how to back up data using snapshots, see "Creating a Snapshot."
- Initializing a disk does not delete the snapshots created for the disk, so you can still use the snapshots to roll back data to the source disk after the disk is initialized.

Initializing a Data Disk

The following example shows you how to create a 100 GiB GPT partition with an NTFS file system on a server running Windows Server 2019.

Step 1 Log in to the server.

For how to log in to an ECS, see **How Do I Log In to My ECS?**

For how to log in to a BMS, see **Windows BMS Login Methods**.

Step 2 On the server desktop, right-click  and choose **Disk Management** from the shortcut menu.

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

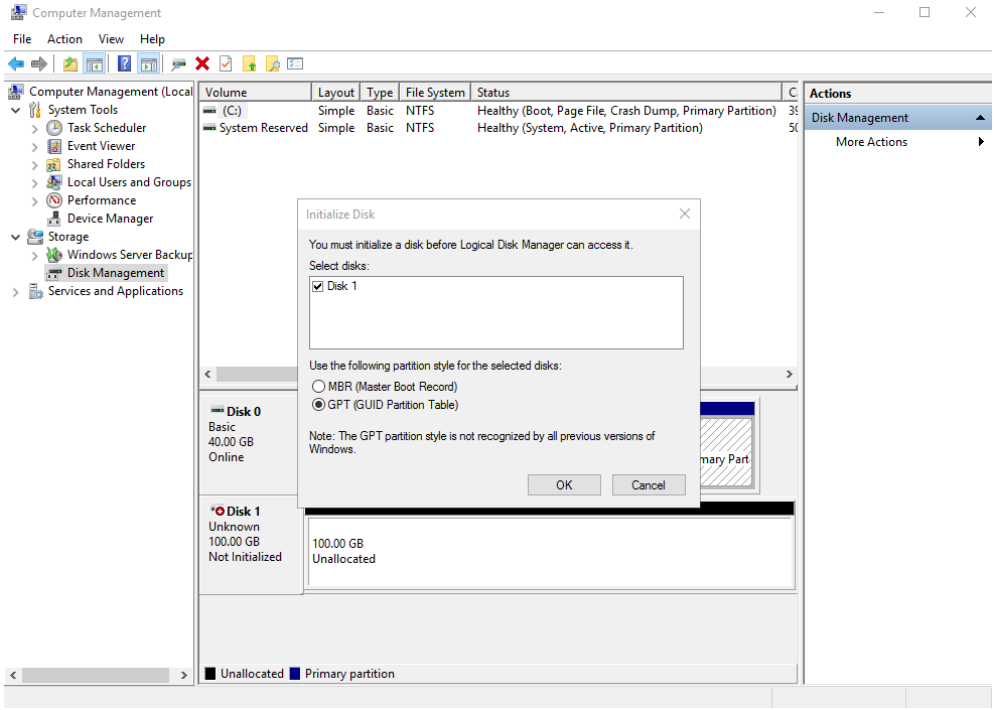
In the **Initialize Disk** dialog box, the to-be-initialized disk is selected. Select a partition style and click **OK**. In this example, **GPT (GUID Partition Table)** is selected.

NOTICE

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

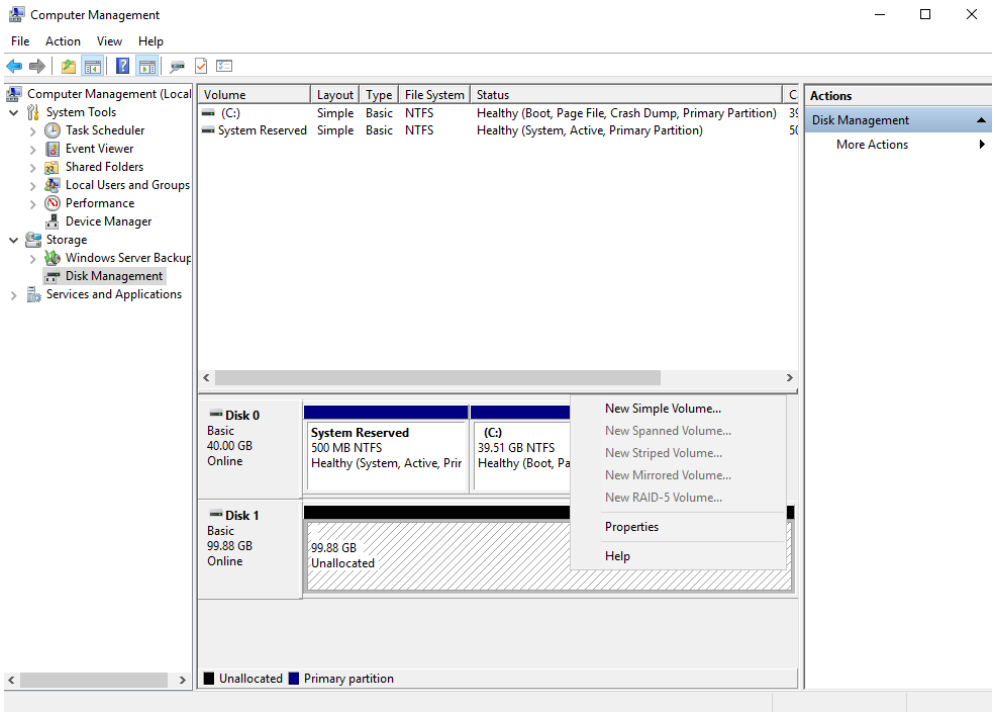
If the partition style of an in-use disk is changed, all data on the disk will be lost, so take care to select an appropriate partition style when initializing the disk. If you must change the partition style to GPT, it is recommended that you back up the disk data before the change.

Figure 5-1 Disk list



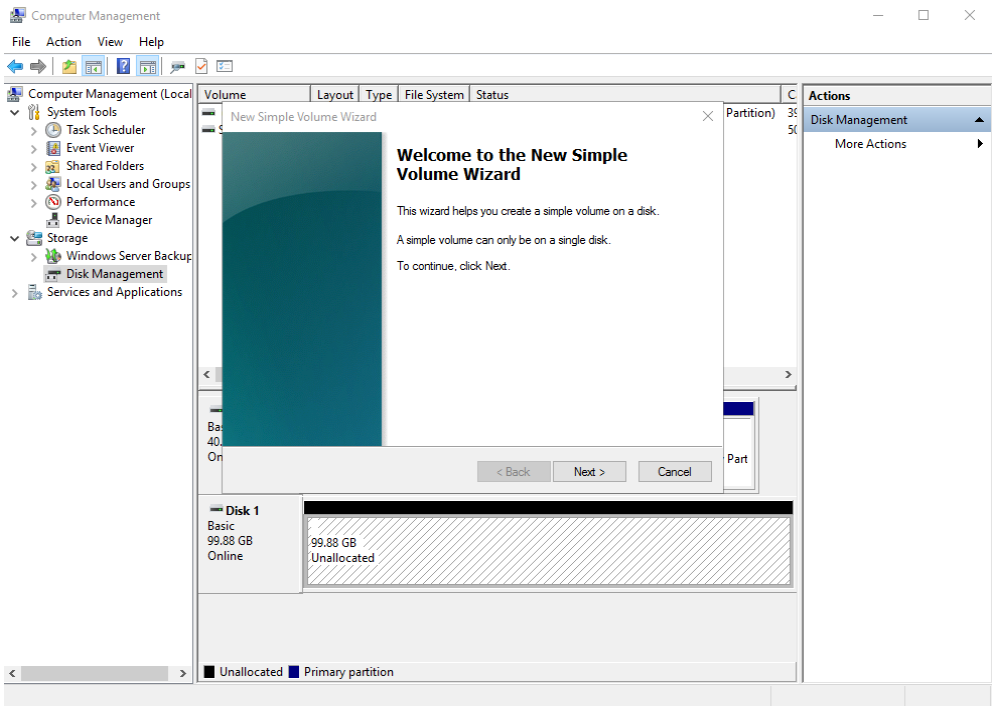
Step 3 In the **Unallocated** area of **Disk 1**, right-click the blank area and choose **New Simple Volume**.

Figure 5-2 Computer Management



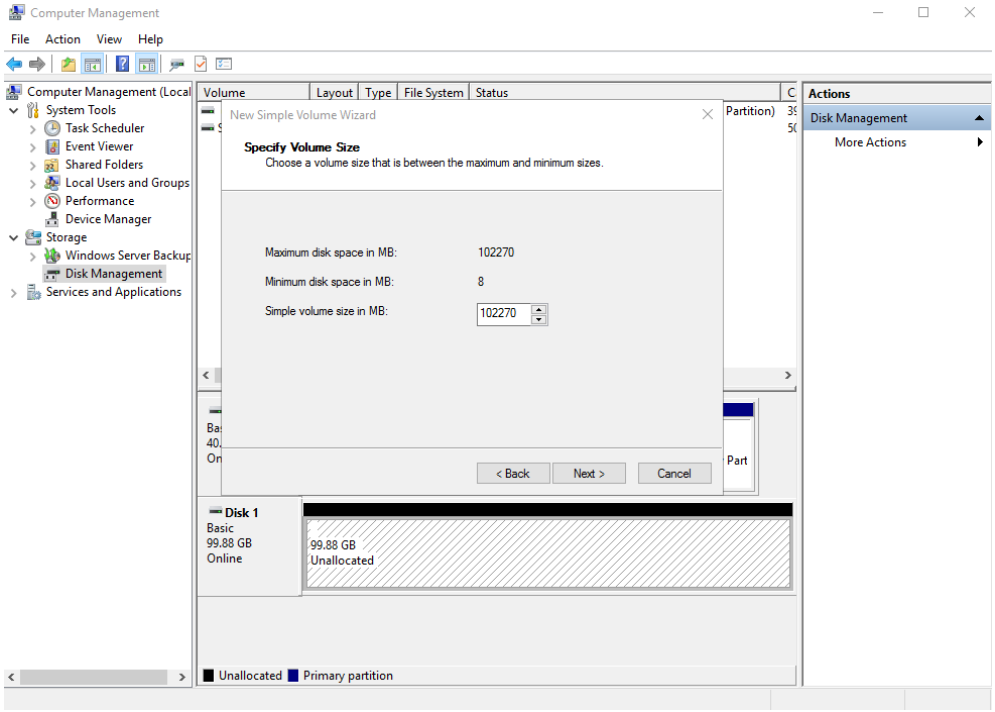
The **New Simple Volume Wizard** window is displayed.

Figure 5-3 New Simple Volume Wizard



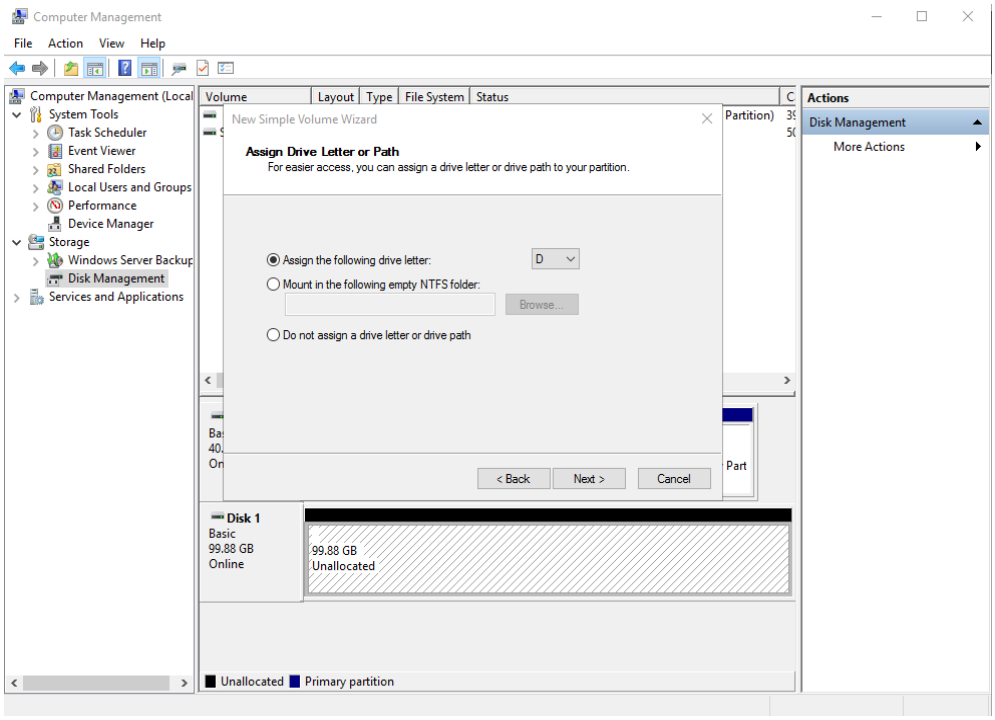
Step 4 Click **Next** to go to the **Specify Volume Size** page.

Figure 5-4 Specify Volume Size



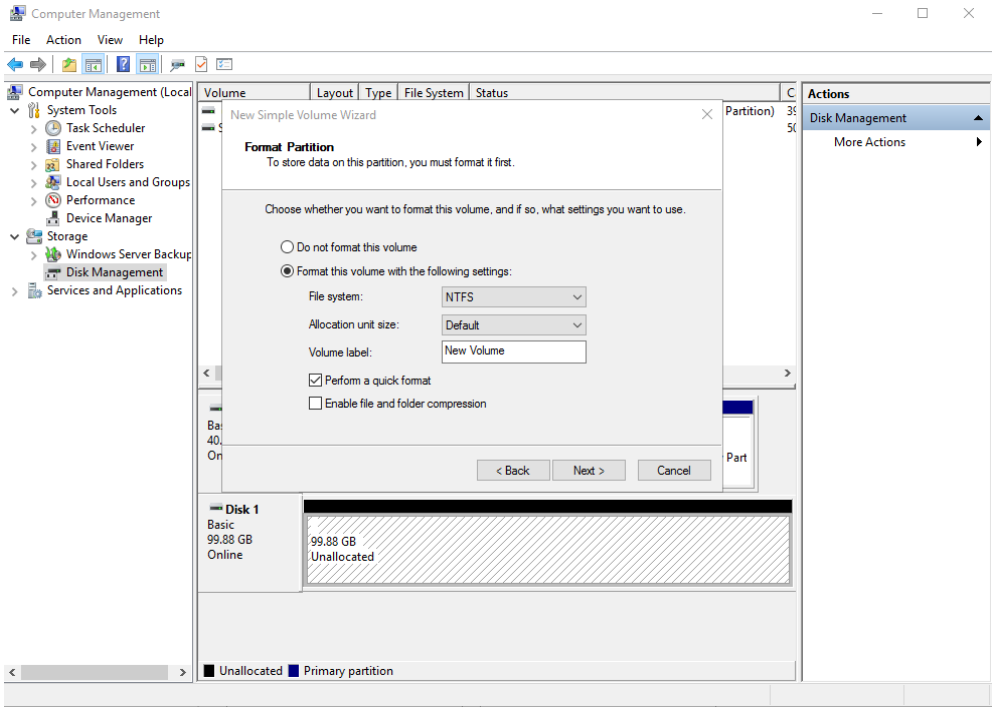
Step 5 Specify the volume size and click **Next**. The system selects the maximum volume size by default. You can specify the volume size as required. In this example, the default setting is used.

Figure 5-5 Assign Drive Letter or Path



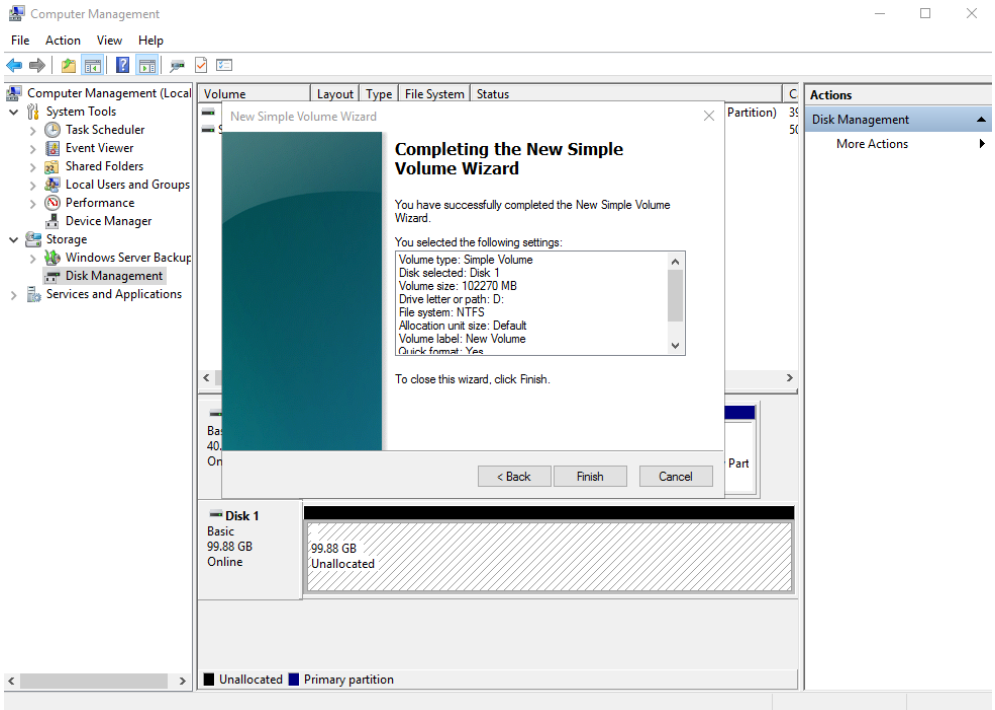
Step 6 Assign a drive letter or path to your partition and click **Next**. The system assigns drive letter D by default. In this example, the default setting is used.

Figure 5-6 Format Partition



Step 7 Specify format settings and click **Next**. The system selects the NTFS file system by default. You can specify a file system format as required. In this example, the default setting is used.

Figure 5-7 Completing the New Simple Volume Wizard



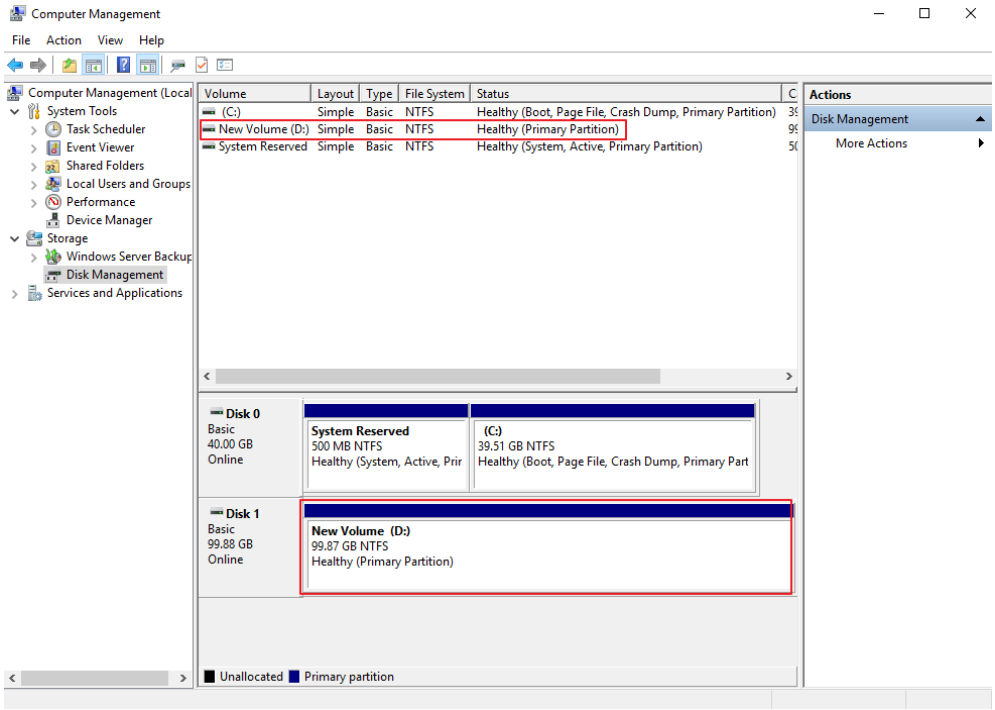
NOTICE

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

Step 8 Click **Finish**.

Wait for the initialization to complete. When the volume status changes to **Healthy**, the initialization succeeds.

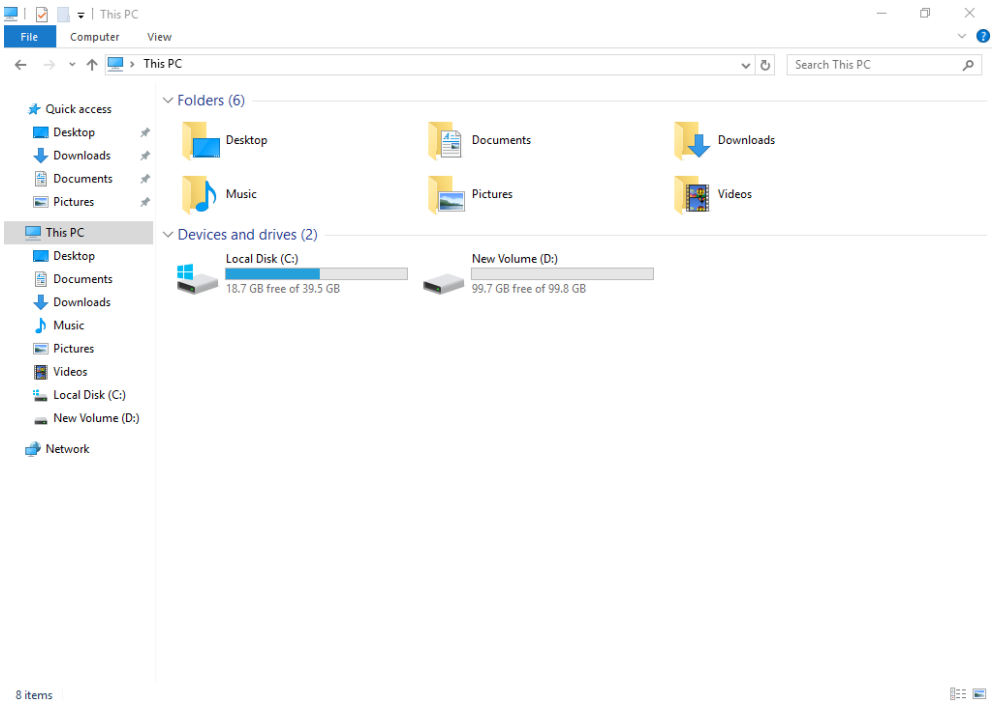
Figure 5-8 Disk initialized



Step 9 After the volume is created, click  on the task bar and check whether a new volume appears in the File Explorer. In this example, New Volume (D:) is the new volume.

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

Figure 5-9 File Explorer



----End

6 Change History

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

