

**Elastic Cloud Server**

# Getting Started

**Issue**            01  
**Date**             2022-09-15



**Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.**

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

## **Trademarks and Permissions**



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.

# 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

---

<b>1 Overview.....</b>	<b>1</b>
<b>2 Purchasing an ECS.....</b>	<b>2</b>
2.1 Overview.....	2
2.2 Step 1: Configure Basic Settings.....	2
2.3 Step 2: Configure Network.....	5
2.4 Step 3: Configure Advanced Settings.....	7
2.5 Step 4: Confirm.....	10
<b>3 Logging In to an ECS.....</b>	<b>12</b>
<b>4 Initializing EVS Data Disks.....</b>	<b>14</b>
4.1 Scenarios and Disk Partitions.....	14
4.2 Initializing Linux Data Disks Using Script.....	15
4.3 Initializing a Linux Data Disk (fdisk).....	19
4.4 Initializing a Linux Data Disk (parted).....	25
4.5 Initializing a Linux Data Disk Larger Than 2 TiB (parted).....	31
<b>5 Change History.....</b>	<b>37</b>

# 1 Overview

---

This section describes the methods of purchasing ECSs, and how to register with Huawei Cloud and complete real-name authentication before purchasing ECSs.

# 2 Purchasing an ECS

---

## 2.1 Overview

### Scenarios

ECSs are more cost-effective than physical servers. Within minutes, you can obtain ECS resources from the cloud service platform. ECS resources are flexible and on-demand. This section describes how to purchase an ECS on the management console.

**Creation process:**

- [Step 1: Configure Basic Settings](#)
- [Step 2: Configure Network](#)
- [Step 3: Configure Advanced Settings](#)
- [Step 4: Confirm](#)

## 2.2 Step 1: Configure Basic Settings

### Basic Settings

1. [Log in to the ECS console and buy an ECS.](#)
2. Select a region.  
ECSs in different regions cannot communicate with each other over an intranet. For low network latency and quick resource access, select the nearest region.
3. Select a billing mode, **Yearly/Monthly**, **Pay-per-use**, or **Spot pricing**.
  - In **Yearly/Monthly** billing mode, you can select a required duration. Then, the system deducts the fees incurred at one time based on the service price.

 NOTE

Yearly/Monthly ECSs cannot be deleted. If such an ECS is no longer required, go to the **Elastic Cloud Server** page, click **More** in the **Operation** column of this ECS, and select **Unsubscribe** to unsubscribe it.

- In **Pay-per-use** billing mode, after configuring basic settings, you do not need to select a required duration. Then, the system deducts the fees incurred based on the service duration.
- In **Spot pricing** billing mode, your purchased ECS is billed based on the service duration at a lower price than that of a pay-per-use ECS with the same specifications. However, a spot ECS may be reclaimed at any time based on the market price or changes in supply and demand. For more information about spot ECSs, see [Spot Pricing ECSs](#).

## 4. Select an AZ.

An AZ is a physical location that uses independent power supply and networks. AZs in the same region can communicate with each other over an intranet.

- To enhance application availability, create ECSs in different AZs.
- To shorten network latency, create ECSs in the same AZ.

 NOTE

Random AZ allocation is available on the console when you create an ECS. The system will use a hash algorithm to select an AZ as the default AZ based on your universally unique identifier (UUID).

The available ECS types and flavors vary depending on AZs. To view all supported ECS types and flavors on the cloud service platform, set **AZ** to **Random**. Then, the system automatically allocates an AZ according to your selected ECS flavor.

For example, S3 ECSs are available only in AZ 1; S2 ECSs are available in AZ 2 and AZ 3 and have been sold out in AZ 1. If you set **AZ** to **Random**, you can view both S3 and S2 ECSs. If you create an S3 ECS, the system automatically allocates it to AZ 1. If you create an S2 ECS, the system randomly allocates it to AZ 2 or AZ 3.

5. Set **Spot Type**.

This parameter is optional and is displayed only when **Billing Mode** is set to **Spot pricing**. You can select **Spot** or **Spot block**. For details, see [Spot Pricing ECSs](#).

6. Set **Specifications**.

The cloud platform provides various ECS types for different application scenarios. You can choose from existing ECS types and flavors in the list. Alternatively, you can enter a flavor or specify vCPUs and memory size to search for the flavor suited to your needs.

 NOTE

- Before selecting an ECS type, learn about the introduction and notes on each type of ECSs. For details, see [ECS Types](#).
- When purchasing an ECS, sold-out vCPU and memory resources cannot be selected. You can select **Hide sold-out specifications** to hide specifications that have been sold out.

## 7. Select an image.

- Public image

A public image is a standard, widely used image. It contains an OS and preinstalled public applications and is available to all users. You can

configure the runtime environment or software in the public image as needed.

– Private image

A private image is an image available only to the user who created it. It contains an OS, preinstalled public applications, and the user's private applications. Using a customized private image, you can create ECSs tailored to your needs in batches.

For instructions about how to create a private image, see [Creating a Private Image](#).

You can also select an encrypted image. For more information about encrypted images, see [Encrypting Images](#).

 NOTE

- If you use a full-ECS image to create an ECS, the EVS disks associated with the full-ECS image do not support the function of creating disks using a data disk image.
- If a full-ECS image is in **Normal** state and the system displays message "Available in AZx", the full-ECS image can be used to create ECSs in this AZ only, and the encryption attributes of the system and data disks of the created ECSs are the same as those of the system and data disks specified in the full-ECS image. Additionally, the SCSI, sharing attribute, and data encryption settings of the system and data disks cannot be modified during ECS creation.
- If a full-ECS image is in **Normal** state but the system does not display message "Available in AZx", the full-ECS image can be used to create ECSs in the entire region, and the encryption attributes of the system and data disks of the created ECSs are the same as those of the system and data disks specified in the full-ECS image. Additionally, the SCSI, sharing attribute, and data encryption settings of data disks can be modified during ECS creation.
- An ISO image created from an ISO file cannot be used to create ECSs. You need to install an OS and drivers on the ECS and use the ECS to create a system disk image first.
- ECSs created from ISO images cannot be directly used because some functions like disk attachment are restricted.
- To ensure that NIC multi-queue is enabled on an ECS created using a private image, configure NIC multi-queue when creating such a private image. NIC multi-queue routes NIC interrupt requests among multiple vCPUs for higher network packets per second (PPS) and bandwidth.

For details, see [How Do I Enable NIC Multi-Queue for an Image?](#)

– Shared image

A shared image is a private image shared by another user.

8. Set **System Disk** and **Data Disk** if required.

– System disk


For details about the disk types supported by ECS, see [Disk Types and Performance](#).

- If the image based on which an ECS is created is not encrypted, the system disk of the ECS is not encrypted. If the image based on which an ECS is created is encrypted, the system disk of the ECS is automatically encrypted. For details, see [\(Optional\) Encryption-related parameters](#).



- Data disk

You can create multiple data disks for an ECS and enable required functions for each data disk. When creating an ECS, you can add up to 23 data disks with customized sizes to it. After the ECS is created, you can add up to 23 VBD disks or 59 SCSI disks to it.

Click **Show**  and set the following functions if required:

    - **SCSI**: indicates that the device type of the data disk is SCSI if you select this option. For more information about SCSI disks and the ECSs that can be attached with SCSI disks, see [Device Types and Usage Instructions](#).
    - **Share**: indicates that the EVS disk is sharable if you select this option. Such an EVS disk can be attached to multiple ECSs.
    - **Encryption**: indicates that the data disk is encrypted if you select this option. For details, see [\(Optional\) Encryption-related parameters](#).
  - (Optional) Encryption-related parameters

To enable encryption, click **Create Xrole** to assign KMS access permissions to EVS. If you have rights granting permission, assign the KMS access permissions to EVS. If you do not have the permission, contact the user having the security administrator rights to assign the KMS access permissions.

    - **Encryption**: indicates that the EVS disk has been encrypted.
    - **Create Xrole**: assigns KMS access permissions to EVS to obtain KMS keys. After the permissions are assigned, follow-up operations do not require assigning permissions again.
    - **Xrole Name: EVSAccessKMS**: specifies that permissions have been assigned to EVS to obtain KMS keys for encrypting or decrypting EVS disks.
    - **KMS Key Name**: specifies the name of the key used by the encrypted EVS disk. You can select an existing key, or click **Create KMS Key** and create a new one on the KMS console. The default value is **evs/default**.
    - **KMS Key ID**: specifies the ID of the key used by the encrypted data disk.
9. Click **Next: Configure Network**.

## 2.3 Step 2: Configure Network

### Network Settings

1. Set **Network** by selecting an available VPC and subnet from the drop-down list and specifying a private IP address assignment mode.

VPC provides a dedicated network for your ECS. A VPC can contain subnets for further isolation. You can configure security groups per subnet to control access to cloud resources.

You can select an existing VPC or create a new one.

For more information about VPC, see [Virtual Private Cloud](#).

 **NOTE**

- Ensure that DHCP is enabled in the VPC to which the ECS belongs.
  - When you use VPC for the first time, the system automatically creates a VPC for you, including the security group and NIC.
2. (Optional) Add an extension NIC. You can add multiple extension NICs to an ECS and specify IP addresses for them (including primary NICs).

 **NOTE**

If you specify an IP address for a NIC when creating multiple ECSs in a batch:

- This IP address serves as the start IP address.
  - Ensure that the IP addresses required by the NICs are within the subnet, consecutive, and available.
  - The subnet with the specified IP address cannot overlap with other subnets.
3. Set **Security Group** by selecting an available security group from the drop-down list or creating a new one.

A security group controls ECS access within or between security groups by defining access rules. This enhances ECS security.

When creating an ECS, you can select multiple (recommended not more than five) security groups. In such a case, the access rules of all the selected security groups apply on the ECS.

 **NOTE**

Before initializing an ECS, ensure that the security group rules for the outbound direction meet the following requirements:

- **Protocol:** TCP
- **Port Range:** 80
- **Remote End:** 169.254.0.0/16

If you use the default security group rules for the outbound direction, the preceding requirements are met, and the ECS can be initialized. The default security group rules for the outbound direction are as follows:

- **Protocol:** ANY
- **Port Range:** ANY
- **Remote End:** 0.0.0.0/16

4. Set EIP.

An EIP is a static public IP address bound to an ECS in a VPC. Using the EIP, the ECS provides services externally.

The following options are provided:

- Auto assign  
The system automatically assigns an EIP for the ECS. The EIP provides a dedicated bandwidth that is configurable.
- Use existing  
An existing EIP is assigned for the ECS. When using an existing EIP, you are not allowed to create ECSs in a batch.
- Not required

Without an EIP, the ECS cannot access the Internet and is used in the private network or cluster only.

 **NOTE**

For a yearly/monthly ECS, **Auto assign** is unavailable to **EIP**. If an EIP is required, bind an existing EIP to the ECS. Alternatively, purchase an EIP that is billed in pay-per-use payment and then bind the EIP to the ECS.

5. Set **EIP Type**.

- Dynamic BGP provides automatic failover and load balancing capabilities and makes better routing decisions based on optimal paths when a network connection fails.
- Static BGP offers routing control and protects against route flapping, but cannot choose an optimal path in real time when a network connection fails.

6. Set **Billed By**.

This parameter is mandatory when **EIP** is set to **Auto assign**. If you select **Bandwidth** or **Traffic**, the system will allocate a dedicated bandwidth for you, and the bandwidth is dedicated for one EIP.

- **Bandwidth**: The billing will be based on the duration for which the bandwidth is used.
- **Traffic**: The billing will be based on the total traffic irrespective of the duration for which the bandwidth is used.
- **Shared bandwidth**: The bandwidth can be used by multiple EIPs.

 **NOTE**

- A bandwidth can be shared among a limited number of EIPs. If the number of EIPs cannot meet service requirements, switch to a higher shared bandwidth or apply for expanding the EIP quota of the existing bandwidth.
- EIPs that are billed yearly/monthly do not support shared bandwidths.
- When a shared bandwidth that is billed yearly/monthly expires, the system automatically deletes the bandwidth configuration and creates a dedicated bandwidth billed by traffic for the EIPs sharing the deleted bandwidth configuration.

7. Set **Bandwidth Size**.

Select the bandwidth based on service requirements. The unit is Mbit/s.

8. Click **Next: Configure Advanced Settings**.

## 2.4 Step 3: Configure Advanced Settings

### Advanced Settings

1. Set **ECS Name**.

The name can be customized but can contain only letters, digits, underscores (\_), hyphens (-), and periods (.).

If you want to purchase multiple ECSs at a time, the system automatically sequences these ECSs.

If multiple ECSs are created at the same time, the system automatically adds a hyphen followed by a four-digit incremental number to the end of each ECS

name. For example, if you enter **ecs**, the ECSs will be named **ecs-0001**, **ecs-0002**, .... If you create multiple ECSs again, the values in the new ECS names increase from the existing maximum value. For example, the existing ECS with the maximum number in name is **ecs-0010**. If you enter **ecs**, the names of the new ECSs will be **ecs-0011**, **ecs-0012**, .... When the value reaches **9999**, it will start from **0001**.

**Allow duplicate name:** allows ECS names to be duplicate. If you select **Allow duplicate name** and create multiple ECSs in a batch, the created ECSs will have the same name.

The **ECS Name** set in this step will be the initial host name in the ECS OS.

 **NOTE**

Consecutive periods (.) or hyphens (-) will be replaced with the first character to prevent unknown issues.

2. Set **Login Mode**.

Key pair authentication is more secure than password authentication. If you select **Password**, ensure that the password meets complexity requirements listed in [Table 2-1](#) to prevent malicious attacks.

- Key pair

A key pair is used for ECS login authentication. You can select an existing key pair, or click **Create Key Pair** and create a desired one.

 **NOTE**

If you use an existing key pair, make sure that you have saved the key file locally. Otherwise, logging in to the ECS will fail.

- Password

A username and its initial password are used for ECS login authentication. The initial password of user **root** is used for authenticating Linux ECSs. The passwords must meet the requirements described in [Table 2-1](#).

**Table 2-1** Password complexity requirements

Parameter	Requirement	Example Value
Password	<ul style="list-style-type: none"> <li>• Consists of 8 to 26 characters.</li> <li>• Contains at least three of the following character types:                             <ul style="list-style-type: none"> <li>- Uppercase letters</li> <li>- Lowercase letters</li> <li>- Digits</li> <li>- Special characters for Linux: !@%-_+=[:./^,}{?}</li> </ul> </li> <li>• Cannot contain the username or the username spelled backwards.</li> <li>• Cannot start with a slash (/) for Windows ECSs.</li> </ul>	YNbUwp! dUc9MClnv <b>NOTE</b> The example password is generated randomly. Do not use it.

 NOTE

The system does not periodically change the ECS password. It is recommended that you change your password regularly for security.

## 3. Configure Cloud Eye.

If you select certain public images, it is a good practice to enable detailed monitoring to collect OS metrics. These metrics help you understand the resource utilization and locate a fault.

After you enable detailed monitoring, an agent will be automatically installed on the ECS to provide 1-minute fine-grained monitoring of ECS metrics, such as CPU, memory, network, disk, and process.

For details about the monitoring metrics after the agent is installed, see [OS Monitoring Metrics Supported by ECSs with the Agent Installed](#).

4. Set **ECS Group (Optional)**.

An ECS group applies the anti-affinity policy to the ECSs in it so that the ECSs are automatically allocated to different hosts. This configuration is optional. For instructions about how to create an ECS group, see [Managing ECS Groups](#).

 NOTE

An existing ECS attached with a local disk cannot be added to an ECS group. To use ECS group functions, select an ECS group when creating an ECS.

## 5. Configure automatic recovery.

If a physical server accommodating ECSs breaks down, the ECSs with automatic recovery enabled will automatically be migrated to a functional server to minimize the impact on your services. During this process, the ECSs will restart. If automatic recovery is not enabled, the affected users must wait for the system administrator to recover ECSs once the physical server becomes faulty.

 NOTE

An ECS with any of the following resources cannot be automatically recovered:

- Local disk
- Passthrough FPGA
- Passthrough InfiniBand NIC

6. To use functions listed in **Advanced Options**, select **Configure now**. Otherwise, do not select it.

## – User Data

You can specify the user data. The user data will be automatically passed to the ECS when the ECS starts for the first time. This configuration is optional.

- **As text:** allows you to enter the user data in the text box.
- **As file:** enables the text to automatically inject a script file or other files into a specified directory on an ECS when you create the ECS.

For example, if you activate user **root** permission by passing a script file to an ECS, you can log in to the ECS as user **root**.

For detailed operations, see [Passing User Data to ECSs](#).

- Tag

This configuration is optional. You can tag an ECS to facilitate identification and management. You can add up to 10 tags to an ECS.

 **NOTE**

Tags added during ECS creation will also be added to the created EIP and EVS disks (including the system disk and data disks) of the ECS. If the ECS uses an existing EIP, the tags will not be added to the EIP.

After creating the ECS, you can view the tags on the pages providing details about the ECS, EIP, and EVS disks.

For details, see [Tag Management](#).

- Agency

This configuration is optional. When your ECS resources need to be shared with other accounts, or your ECS is delegated to professional personnel or team for management, the tenant administrator creates an agency in IAM and grants the ECS management permissions to the personnel or team. The delegated account can log in to the cloud system and switch to your account to manage resources. You do not need to share security credentials (such as passwords) with other accounts, ensuring the security of your account.

If you have created an agency in IAM, you can select the agency from the drop-down list and obtain specified operation permissions. For more information about agencies, see [Account Delegation](#).

7. Click **Next: Confirm**.

## 2.5 Step 4: Confirm

### Confirming the Order

1. On the **Confirm** page, view details about the ECS configuration.

2. Set **Enterprise Project**.

This function is provided for enterprise users.

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

Select an enterprise project from the drop-down list. For more details, see *Enterprise Management User Guide*.

3. Select the required duration and quantity for the ECS.

- **Required Duration**: sets the service duration if you select the **Yearly/Monthly** billing mode. The service duration ranges from one month to three years.

You can select **Auto-renew** to automatically renew yearly/monthly resources when they expire.

- **Monthly**: Your subscription will be automatically renewed each month.
- **Yearly**: Your subscription will be automatically renewed each year.

For details about auto-renewal, see [Auto-Renewal Rules](#).

- **Quantity:** sets the number of ECSs to be created.  
The remaining number of ECSs you are allowed to create is displayed. To ensure effective resource usage, an upper limit is set on the ECSs to be created. If the number of ECSs you need exceeds the maximum number of ECSs you are allowed to create, you can click **Increase Quota** to apply for a higher quota. You can purchase the ECSs you need after the request is approved.
  - **ECS Price:** shows the ECS configuration fee.  
For details, click **Pricing details**.
4. Confirm the configuration and click **Submit**.  
After the ECS is created, it will be started by default.

### Follow-up Procedure

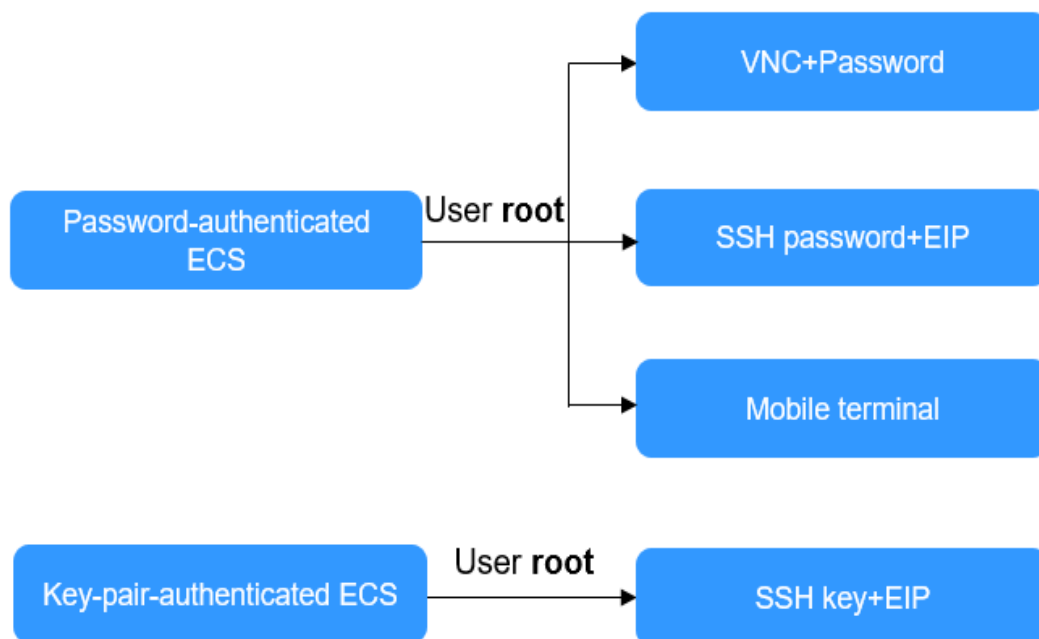
- After an ECS with automatic recovery enabled is created, you can check whether this function has been enabled by viewing **Failures**. For details, see [Viewing Failures](#).

# 3 Logging In to an ECS

## Logging In to a Linux ECS

The method of logging in to an ECS varies depending on the login authentication configured during ECS creation.

**Figure 3-1** Linux ECS login modes



- To log in to a password-authenticated ECS for the first time, use one of the following methods:
  - Using VNC on the management console with login username **root**  
For details, see [Login Using VNC](#).
  - Using an SSH password  
The login username is **root**, and the ECS must have an EIP bound.  
For details, see [Login Using an SSH Password](#).
  - From a mobile terminal



The login username is **root**, and the ECS must have an EIP bound.

For details, see [Logging In to a Linux ECS from a Mobile Terminal](#).

- To log in to a key-pair-authenticated ECS for the first time, use a tool, such as PuTTY or XShell, and the desired SSH key as user **root**. Ensure that the ECS has an EIP bound.

For instructions about how to log in to a Linux ECS using an SSH key, see [Login Using an SSH Key](#).

 **NOTE**

If you want to log in to an ECS using VNC provided on the management console, log in to the ECS using an SSH key, configure the login password, and use the password for login.

## Follow-up Procedure

- If you have added a data disk during ECS creation, you must initialize the data disk after logging in to the ECS.  
For details, see [Scenarios and Disk Partitions](#).
- Certain ECSs require the installation of a driver after you log in to them. For details about available ECS types as well as their functions and usage, see "Notes" in .

# 4 Initializing EVS Data Disks

---

## 4.1 Scenarios and Disk Partitions

If you have added a data disk during ECS creation, you must initialize the data disk after logging in to the ECS.

### Scenarios

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

- System disk  
A system disk does not require manual initialization because it is automatically created and initialized upon server creation. The default partition style is master boot record (MBR).
- Data disk
  - If a data disk is created along with a server, it will be automatically attached to the server.
  - If a data disk is created separately, you need to manually attach it to a server.

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

### Partitioning Operation Guide

[Table 4-1](#) lists the common disk partition styles. In Linux, different disk partition styles require different partitioning tools.

**Table 4-1** Disk partition styles

Disk Partition Style	Maximum Disk Capacity Supported	Maximum Number of Partitions Supported	Linux Partitioning Tool
Master Boot Record (MBR)	2 TiB	<ul style="list-style-type: none"><li>• 4 primary partitions</li><li>• 3 primary partitions and 1 extended partition</li></ul> <p>With MBR, you can create several primary partitions and one extended partition. The extended partition must be divided into logical partitions before use. For example, if 6 partitions need to be created, you can create them in the following two ways:</p> <ul style="list-style-type: none"><li>• 3 primary partitions and 1 extended partition, with the extended partition divided into 3 logical partitions</li><li>• 1 primary partition and 1 extended partition, with the extended partition divided into 5 logical partitions</li></ul>	<ul style="list-style-type: none"><li>• fdisk</li><li>• parted</li></ul>
GUID Partition Table (GPT)	18 EiB 1 EiB = 1048576 TiB	Unlimited Disk partitions created using GPT are not categorized.	parted

## 4.2 Initializing Linux Data Disks Using Script

### Scenarios

This section uses CentOS 7.4 64bit to describe how to initialize data disks attached to a server running Linux using the initialization script.

The formatting procedure varies depending on the server OS. This section is for reference only.

---

**NOTICE**

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

---

## Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see [Logging In to an ECS](#).
  - For how to log in to a BMS, see [Logging In to the BMS](#).

## Constraints

- This script uses the EXT4 file system by default. It cannot be used for the disks using LVM or VG.
- This script can only be used to format CentOS.

## Initializing New Disks

1. Log in to the server and run the following command to obtain the automatic disk initialization script:

```
wget https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/datadisk/LinuxVMDataDiskAutoInitialize.sh
```

### NOTE

If an abnormal output is returned, check whether the server has an EIP bound. When you are in a region other than **CN North-Beijing1**, the script can be obtained only after an EIP is bound to the server.

2. Run the following command to change the permission of the script:

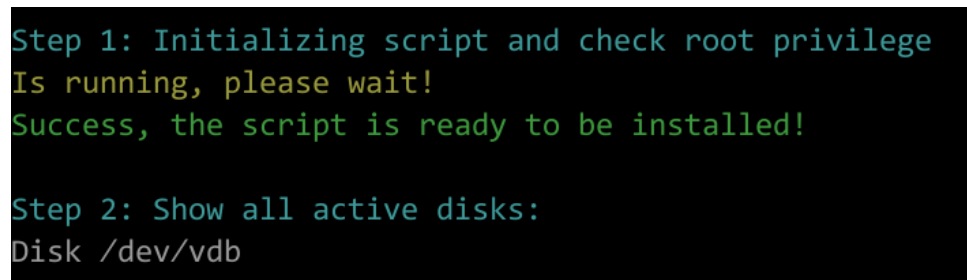
```
chmod +x LinuxVMDataDiskAutoInitialize.sh
```

3. Execute the following script to automatically detect the data disks to be initialized:

```
./LinuxVMDataDiskAutoInitialize.sh
```

The script automatically detects and displays the drive letters (for example, **/dev/vdb**) except the system disk on the server. Then, you need to enter the drive letter to be initialized, for example, **/dev/vdb**.

**Figure 4-1** Automatic detecting disks



```
Step 1: Initializing script and check root privilege
Is running, please wait!
Success, the script is ready to be installed!

Step 2: Show all active disks:
Disk /dev/vdb
```

4. Enter the drive letter and press **Enter**. The script automatically creates the partition for the disk and formats it.

**Figure 4-2** Entering the drive letter

```
Step 3: Please choose the disk(e.g.: /dev/vdb and q to quit):/dev/vdb
Step 4: The disk is partitioning and formatting
Is running, please wait!
Success, the disk has been partitioned and formatted!
```

5. Enter the directory on which you want to mount the partition, for example, **/data-test**.

**Figure 4-3** Entering the mounting directory

```
Step 5: Make a directory and mount it
Please enter a location to mount (e.g.: /mnt/data):/data-test
Success, the mount is completed!
```

The initialization is complete until the script has mounted the partition and configured automatic mounting upon server start.

**Figure 4-4** Disk initialization completed

```
Step 6: Write configuration to /etc/fstab and mount device
Success, the /etc/fstab is Write!

Step 7: Show information about the file system on which each FILE resides
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0   4.0M   0% /dev
tmpfs           16G    0   16G    0% /dev/shm
tmpfs           16G   8.6M   16G    1% /run
tmpfs           4.0M   0   4.0M   0% /sys/fs/cgroup
/dev/vda1       40G   2.1G   36G    6% /
tmpfs           16G   32K   16G    1% /tmp
/dev/sdb1       1.7T   28K   1.6T    1% /data-test

Step 8: Show the write configuration to /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Jul  7 04:01:51 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=257f1147-e890-4926-a897-4e9347cf1b97 / ext4 defaults 1 1
UUID=a0eeaf8f-bacd-4999-aaba-b6a5286be02c /data-test ext4 defaults 0 0
```

## Initializing Existing Disks

1. Log in to the server and run the following command to obtain the automatic disk initialization script:

```
wget https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/datadisk/LinuxVMDataDiskAutoInitialize.sh
```

2. Run the following command to change the permission of the script:

```
chmod -x LinuxVMDataDiskAutoInitialize.sh
```

3. Execute the following script to automatically detect the data disks to be initialized:

```
sh LinuxVMDataDiskAutoInitialize.sh
```

- The script automatically detects that the disk partition has been mounted and displays the mounting directory. Determine whether to unmount the partition. Enter **y** to unmount the partition, or enter **n** to exit the script. In this example, enter **y** to continue the initialization.

**Figure 4-5** Unmounting the disk partition

```
Your disk has been mounted:
/dev/vdb1 on /data-test type ext4 (rw,relatime,data=ordered)

Force uninstalling? [y/n]:y
Is running, please wait!
Success, the disk is unloaded!
```

- After the disk partition is unmounted, determine whether to format the disk. Enter **y** to format the disk, or enter **n** to exit the script.

**Figure 4-6** Formatting the disk

```
Ready to begin to format the disk? [y/n]:y
Is running, please wait!
Success, the disk has been formatted!

Step 4: The disk is partitioning and formatting
Is running, please wait!
Success, the disk has been partitioned and formatted!
```

- After the disk is formatted, the system automatically creates a partition and completes the formatting. Enter the directory on which you want to mount the partition, for example, **/data-test**.

**Figure 4-7** Entering the mounting directory

```
Step 5: Make a directory and mount it
Please enter a location to mount (e.g.: /mnt/data):/data-test
Success, the mount is completed!
```

The initialization is complete until the script has mounted the partition and configured automatic mounting upon server start.

**Figure 4-8** Disk formatting and partition mounting completed

```
Step 6: Write configuration to /etc/fstab and mount device
Success, the /etc/fstab is Write!

Step 7: Show information about the file system on which each FILE resides
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0    4.0M   0% /dev
tmpfs           16G    0    16G    0% /dev/shm
tmpfs           16G   8.6M   16G    1% /run
tmpfs           4.0M   0    4.0M   0% /sys/fs/cgroup
/dev/vda1       40G   2.1G   36G    6% /
tmpfs           16G   32K   16G    1% /tmp
/dev/sdb1       1.7T   28K   1.6T    1% /data-test

Step 8: Show the write configuration to /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Jul  7 04:01:51 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=257f1147-e890-4926-a897-4e9347cf1b97 / ext4 defaults 1 1
UUID=a0eeaf8f-bacd-4999-aaba-b6a5286be02c /data-test ext4 defaults 0 0
```

## 4.3 Initializing a Linux Data Disk (fdisk)

### Scenarios

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

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

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

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

---

#### NOTICE

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

---

### Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see [Logging In to an ECS](#).
  - For how to log in to a BMS, see [Logging In to the BMS](#).

### Creating and Mounting a Partition

The following example shows you how a new primary partition can be created on a new data disk that has been attached to a server. The primary partition will be created using fdisk, and MBR will be used. Furthermore, the partition will be formatted using the ext4 file system, mounted on **/mnt/sdc**, and configured to mount automatically at startup.

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

#### **fdisk -l**

Information similar to the following is displayed:

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

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

```
Disk identifier: 0x000bcb4e
  Device Boot   Start      End      Blocks  Id System
/dev/vda1 *    2048    83886079  41942016  83 Linux

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

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

**Step 2** Launch **fdisk** to partition the new data disk.

**fdisk** *New data disk*

In this example, run the following command:

**fdisk /dev/vdb**

Information similar to the following is displayed:

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

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

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

Command (m for help):
```

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

Information similar to the following is displayed:

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

There are two types of disk partitions:

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

 **NOTE**

If MBR is used, a maximum of 4 primary partitions, or 3 primary partitions plus 1 extended partition can be created. The extended partition must be divided into logical partitions before use.

Disk partitions created using GPT are not categorized.

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

Information similar to the following is displayed:

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

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

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



Information similar to the following is displayed:

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

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

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

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

Information similar to the following is displayed:

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

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

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

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

Information similar to the following is displayed:

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

Command (m for help):
```

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

**Step 8** Enter **p** and press **Enter** to print the partition details.

Information similar to the following is displayed:

```
Command (m for help): p

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

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

Command (m for help):
```

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

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

Information similar to the following is displayed:

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

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

The partition is created.

 NOTE

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

**Step 10** Synchronize the new partition table to the OS.

**partprobe**

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

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

In this example, the **ext4** format is used for the new partition.

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

Information similar to the following is displayed:

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

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

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

---

**NOTICE**

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

---

**Step 12** Create a mount point.

**mkdir *Mount point***

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

**mkdir /mnt/sdc**

 NOTE

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

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

**mount** *Disk partition Mount point*

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

**mount /dev/vdb1 /mnt/sdc**

**Step 14** Check the mount result.

**df -TH**

Information similar to the following is displayed:

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

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

#### NOTE

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

----End

## Configuring Automatic Mounting at System Start

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

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

#### NOTE

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

**Step 1** Query the partition UUID.

**blkid** *Disk partition*

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

**blkid /dev/vdb1**

Information similar to the following is displayed:

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

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

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

```
vi /etc/fstab
```

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

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

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

In this example, the line starting with "UUID" is the information added. Edit this line to match the following format:

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

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

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

The system saves the configurations and exits the vi editor.

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

1. Unmount the partition.

```
umount Disk partition
```

In this example, run the following command:

```
umount /dev/vdb1
```

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

```
mount -a
```

3. Query the file system mounting information.

```
mount | grep Mount point
```

In this example, run the following command:

```
mount | grep /mnt/sdc
```

If information similar to the following is displayed, automatic mounting has been configured:

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

----End

## 4.4 Initializing a Linux Data Disk (parted)

### Scenarios

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

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

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

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

---

#### NOTICE

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

---

### Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see [Logging In to an ECS](#).
  - For how to log in to a BMS, see [Logging In to the BMS](#).

### Creating and Mounting a Partition

The following example shows you how a new partition can be created on a new data disk that has been attached to a server. The partition will be created using parted, and GPT will be used. Furthermore, the partition will be formatted using the ext4 file system, mounted on `/mnt/sdc`, and configured to mount automatically at startup.

**Step 1** Query information about the new data disk.

#### lsblk

Information similar to the following is displayed:

```
root@ecs-test-0001 ~]# lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
vda 253:0 0 40G 0 disk
```

```
└─vda1 253:1 0 40G 0 part /  
vdb 253:16 0 100G 0 disk
```

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

**Step 2** Launch parted to partition the new data disk.

**parted** *New data disk*

In this example, run the following command:

**parted /dev/vdb**

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# parted /dev/vdb  
GNU Parted 3.1  
Using /dev/vdb  
Welcome to GNU Parted! Type 'help' to view a list of commands.  
(parted)
```

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

Information similar to the following is displayed:

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

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

**Step 4** Set the disk partition style.

**mklabel** *Disk partition style*

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

**mklabel gpt**

---

#### NOTICE

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

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

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

Information similar to the following is displayed:

```
(parted) mklabel gpt  
(parted) p  
Model: Virtio Block Device (virtblk)
```

```
Disk /dev/vdb: 107GiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start End Size File system Name Flags
(parted)
```

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

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

**Step 7** Create a new partition.

```
mkpart Partition name Start sector End sector
```

In this example, run the following command:

```
mkpart test 2048s 100%
```

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

Information similar to the following is displayed:

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

**Step 8** Enter **p** and press **Enter** to print the partition details.

Information similar to the following is displayed:

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

Number Start End Size File system Name Flags
1 2048s 209713151s 209711104s test
(parted)
```

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

Information similar to the following is displayed:

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

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

**Step 10** View the disk partition information.

```
lsblk
```

Information similar to the following is displayed:

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

```
vdb 253:16 0 100G 0 disk
└─vdb1 253:17 0 100G 0 part
```

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

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

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

In this example, the **ext4** format is used for the new partition.

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

Information similar to the following is displayed:

```
[root@ecs-test-0001 ~]# mkfs -t ext4 /dev/vdb1
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26213888 blocks
1310694 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2174746624
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

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

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

---

#### NOTICE

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

---

**Step 12** Create a mount point.

**mkdir *Mount point***

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

**mkdir /mnt/sdc**

#### NOTE

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

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

**mount *Disk partition Mount point***



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

```
mount /dev/vdb1 /mnt/sdc
```

**Step 14** Check the mount result.

```
df -TH
```

Information similar to the following is displayed:

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

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

#### NOTE

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

----End

## Configuring Automatic Mounting at System Start

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

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

#### NOTE

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

**Step 1** Query the partition UUID.

```
blkid Disk partition
```

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

```
blkid /dev/vdb1
```

Information similar to the following is displayed:

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

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

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

```
vi /etc/fstab
```

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

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

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

In this example, the line starting with "UUID" is the information added. Edit this line to match the following format:

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

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

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

The system saves the configurations and exits the vi editor.

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

1. Unmount the partition.

```
umount Disk partition
```

In this example, run the following command:

```
umount /dev/vdb1
```

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

```
mount -a
```

3. Query the file system mounting information.

```
mount | grep Mount point
```

In this example, run the following command:

```
mount | grep /mnt/sdc
```

If information similar to the following is displayed, automatic mounting has been configured:

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

----End

## 4.5 Initializing a Linux Data Disk Larger Than 2 TiB (parted)

### Scenarios

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

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

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

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

---

#### NOTICE

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

---

### Prerequisites

- A data disk has been attached to a server and has not been initialized.
- You have logged in to the server.
  - For how to log in to an ECS, see [Logging In to an ECS](#).
  - For how to log in to a BMS, see [Logging In to the BMS](#).

### Creating and Mounting a Partition

The following example shows you how a new partition can be created on a new data disk that has been attached to a server. The partition will be created using parted, and GPT will be used. Furthermore, the partition will be formatted using the ext4 file system, mounted on `/mnt/sdc`, and configured to mount automatically at startup.

**Step 1** Query information about the new data disk.

#### lsblk

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
vda  253:0  0 40G  0 disk
```

```
└─vda1 253:1 0 1G 0 part /boot
└─vda2 253:2 0 39G 0 part /
vdb 253:16 0 3T 0 disk
```

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

**Step 2** Launch `parted` to partition the new data disk.

**parted** *New data disk*

In this example, run the following command:

**parted /dev/vdb**

Information similar to the following is displayed:

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

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

Information similar to the following is displayed:

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

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

**Step 4** Set the disk partition style.

**mklabel** *Disk partition style*

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

**mklabel gpt**

---

#### NOTICE

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

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

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

Information similar to the following is displayed:

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

Number Start End Size File system Name Flags
(parted)
```

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

**Step 7** Create a new partition.

**mkpart** *Partition name Start sector End sector*

In this example, run the following command:

**mkpart opt 2048s 100%**

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

Information similar to the following is displayed:

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

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

**Step 8** Enter **p** and press **Enter** to print the partition details.

Information similar to the following is displayed:

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

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

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

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

**Step 10** View the disk partition information.

**lsblk**

Information similar to the following is displayed:

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

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

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

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

In this example, the **ext4** format is used for the new partition.

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

Information similar to the following is displayed:

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

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

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

---

#### NOTICE

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

---

**Step 12** Create a mount point.

**mkdir** *Mount point*

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

**mkdir /mnt/sdc**

#### NOTE

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

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

**mount** *Disk partition* *Mount point*

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

```
mount /dev/vdb1 /mnt/sdc
```

**Step 14** Check the mount result.

```
df -TH
```

Information similar to the following is displayed:

```
[root@ecs-centos74 ~]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
/dev/vda2       ext4      42G   1.5G   38G   4% /
devtmpfs        devtmpfs  2.0G   0     2.0G   0% /dev
tmpfs           tmpfs     2.0G   0     2.0G   0% /dev/shm
tmpfs           tmpfs     2.0G   8.9M   2.0G   1% /run
tmpfs           tmpfs     2.0G   0     2.0G   0% /sys/fs/cgroup
/dev/vda1       ext4      1.1G  153M   801M  17% /boot
tmpfs           tmpfs     398M   0     398M   0% /run/user/0
/dev/vdb1       ext4      3.3T   93M   3.1T   1% /mnt/sdc
```

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

----End

## Configuring Automatic Mounting at System Start

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

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

### NOTE

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

**Step 1** Query the partition UUID.

```
blkid Disk partition
```

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

```
blkid /dev/vdb1
```

Information similar to the following is displayed:

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

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

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

```
vi /etc/fstab
```

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

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

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

In this example, the line starting with "UUID" is the information added. Edit this line to match the following format:

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

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

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

The system saves the configurations and exits the vi editor.

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

1. Unmount the partition.

**umount** *Disk partition*

In this example, run the following command:

```
umount /dev/vdb1
```

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

```
mount -a
```

3. Query the file system mounting information.

```
mount | grep Mount point
```

In this example, run the following command:

```
mount | grep /mnt/sdc
```

If information similar to the following is displayed, automatic mounting has been configured:

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

----End



# 5 Change History

---

Released On	Description
2022-09-15	This issue is the first official release.