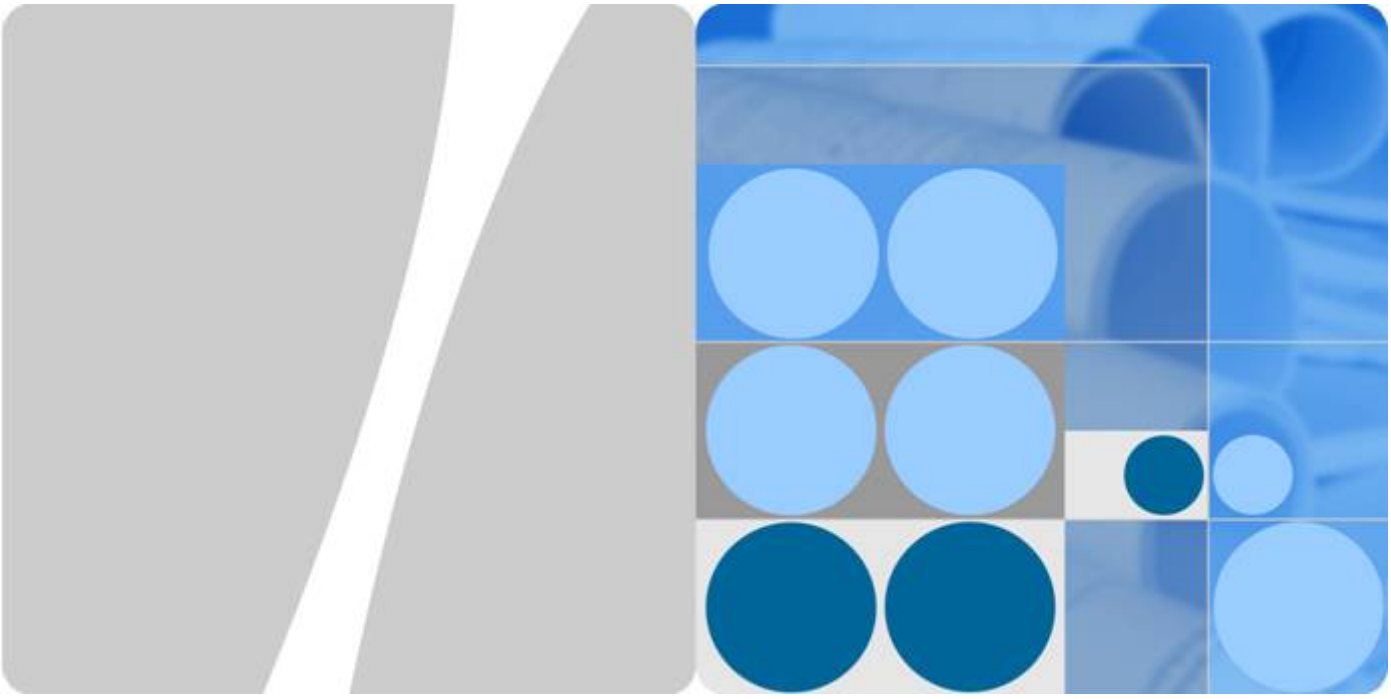


Image Management Service

User Guide

Date **2024-06-12**

Contents

1 Overview

1.1 What Is Image Management Service?

Overview

An image is a cloud server or disk template that contains an operating system (OS), service data, or necessary software.

Image Management Service (IMS) provides image lifecycle management. You can create ECSs from public, private, or shared images. You can also create a private image from a cloud server or an external image file to make it easier to migrate workloads to the cloud or on the cloud.

Image Types

IMS provides public, private, and shared images. Public images are provided by the cloud platform, private images are created by users, and shared images are private images that other users shared with you.

Image Type	Description
Public	A public image is a standard, widely used image. It contains an OS and preinstalled public applications and is available to all users. Public images are very stable and their OS and any included software have been officially authorized for use. If a public image does not contain the environments or software you need, you can use a public image to create an ECS and then deploy the required environments or software on it.

Image Type	Description
Private	A private image contains an OS or service data, preinstalled public applications, and a user's personal applications. Private images are only available to the users who created them. A private image can be a system disk image, data disk image, ISO image, or full-ECS image. ● A system disk image contains an OS and preinstalled software for various services. You can use a system disk image to create ECSs and migrate your services to the cloud. ● A data disk image contains only service data. You can use a data disk image to create EVS disks and use them to migrate your service data to the cloud. ● An ISO image is created from an external ISO image file. It is a special image that is not available on the ECS console. ● A full-ECS image contains an OS, preinstalled software, and service data. A full-ECS image is created using differential backups and the creation takes less time than creating a system or data disk image that has the same disk capacity.
Shared	A shared image is a private image another user has shared with you. For more information about shared images, see "Sharing Images" in *Image Management Service User Guide*.

IMS Functions

IMS provides:

- Public images that contain common OSs
- Creation of a private image from an ECS or external image file
- Public image management, such as searching for images by OS type, name, or ID, and viewing the image ID, system disk capacity, and image features such as user data injection and disk hot swap
- Private image management, such as modifying image attributes, sharing images, and replicating images
- Creation of ECSs using an image

Access Methods

The public cloud provides a web-based service management platform (a management console). You can access the IMS service through HTTPS APIs or from the management console.

- API

  If you need to integrate IMS into a third-party system for secondary development, use APIs to access the IMS service. For details, see *Image Management Service API Reference*.

- Management console

  If no integration with a third-party system is needed, use the management console. Log in to the management console and choose **Computing** > **Image Management Service** on the homepage.

1.2 Product Advantages

IMS provides convenient, secure, flexible, and efficient image management. Images allow you to deploy services faster, more easily and more securely.

Saving Time and Effort

- Deploying services on cloud servers is much faster and easier when you use images.
- A private image can be created from an ECS, a BMS, or an external image file. It can be a system disk, data disk, or full-ECS image that suites your different needs.
- Private images can be transferred between accounts, regions, or cloud platforms through image sharing, replication, and export.

Secure

- Public images use mainstream OSs such as Ubuntu and CentOS. These OSs have been thoroughly tested to provide secure and stable services.
- Multiple copies of image files are stored on Object Storage Service (OBS), which provides excellent data reliability and durability.
- Private images can be encrypted for data security by using envelope encryption provided by Key Management Service (KMS).

Flexible

- You can manage images through the management console or using APIs.
- You can use a public image to deploy a general-purpose environment, or use a private image to deploy a custom environment.
- You can use IMS to migrate servers to the cloud or on the cloud, and back up server running environments.

Unified

- IMS provides a self-service platform to simplify image management and maintenance.
- IMS allows you to batch deploy and upgrade application systems, improving O&M efficiency and ensuring consistency.
- Public images comply with industry standards. Preinstalled components only include clean installs, and only kernels from well-known third-party vendors are used to make it easier to transfer images from or to other cloud platforms.

Comparison Between Image-based Deployment and Manual Deployment

Table 1-1 Image-based deployment and manual deployment

Item	Image-based Deployment	Manual Deployment
Time required	2 to 5 minutes	1 to 2 days
Complexity	Quickly create ECSs by using public images or private images.	Select an appropriate OS, database, and various software packages based on your service requirements. Then, install and commission them.
Security	You only need to identify sources of shared images. Public and private images have been thoroughly tested to ensure security and stability.	The security depends on the skills of the R&D or O&M personnel.

1.3 Application Scenarios

- Migrating servers to the cloud or on the cloud

  You can import local images to the cloud platform and use the images to quickly create cloud servers for service migration to the cloud. A variety of image types can be imported, including VHD, VMDK, QCOW2, and RAW.

  You can also share or replicate images across regions to migrate ECSs between accounts and regions.
- Deploying a specific software environment

  Use shared images to quickly build custom software environments without having to manually configure environments or install any software. This is especially useful for Internet startups.
- Batch deploying software environments

  Prepare an ECS with an OS, the partition arrangement you prefer, and software installed to create a private image. You can use the image to create batch clones of your custom ECS.
- Backing up server environments

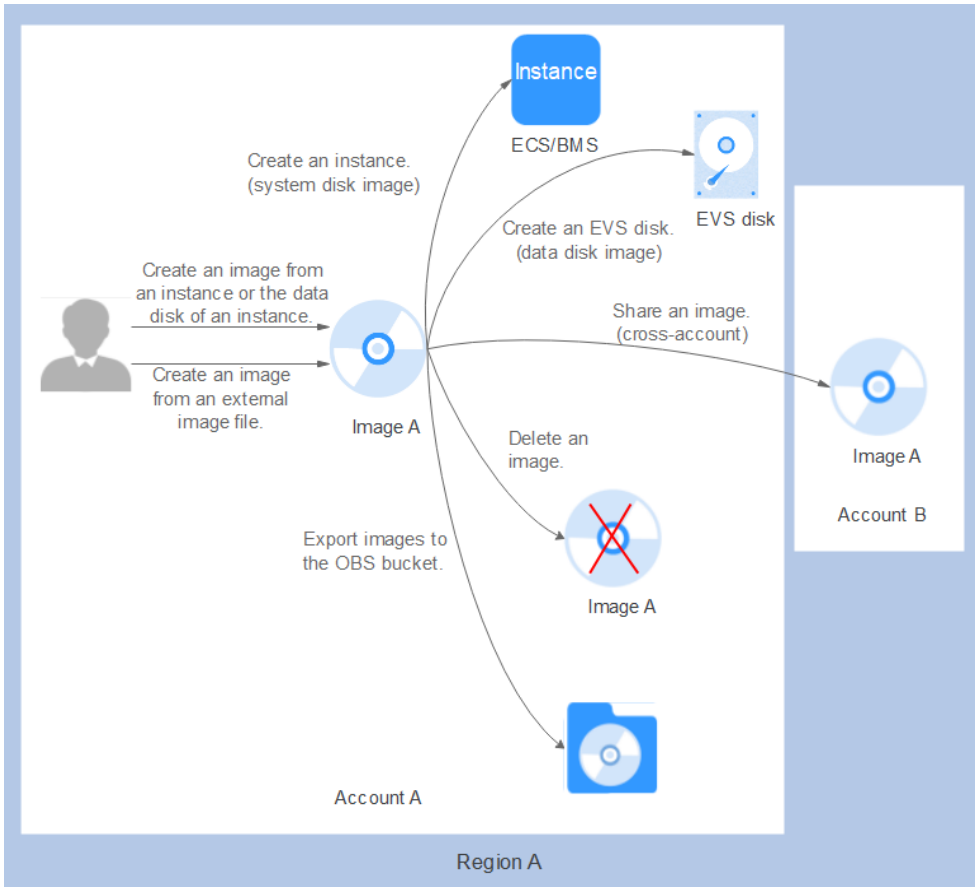
  Create an image from an ECS to back up the ECS. If the ECS breaks down due to software faults, you can use the image to restore the ECS.

1.4 Features

Private Image Lifecycle

After you create a private image, you can use it to create cloud servers or EVS disks. You can also share the image with other tenants or replicate it to other regions. **Figure 1-1** shows the lifecycle of a private image.

Figure 1-1 Private Image Lifecycle



Features

Table 1-2 Creating a private image

Feature	Description	Helpful Link
Creating a system disk image from an ECS or BMS	After creating a cloud server, you can set it up, installing whatever software or application environment you need, and then use the preconfigured server to create a system disk image. You can create new cloud servers with the custom configurations from the image, which frees you from a lot of repetitive work.	• **Creating a System Disk Image from a Windows ECS** • **Creating a System Disk Image from a Linux ECS** • **Creating a BMS System Disk Image**

Feature	Description	Helpful Link
Creating a system disk image from an external image file	You can import a system disk from your local PC or other cloud platforms, and use the imported image to create new cloud servers or reinstall or change the OSs of existing cloud servers.	• **Creating a Windows System Disk Image from an External Image File** • **Creating a Linux System Disk Image from an External Image File** • **Quickly Importing an Image File**
Creating a system disk image from an ISO file	In contrast with other image formats, an ISO file can be used only after it is decompressed using a tool, such as UltraISO or VirtualBox. For details about the image creation process, see the **Related Operations** column in the table.	• **Creating a Windows System Disk Image from an ISO File** • **Creating a Linux System Disk Image from an ISO File**
Creating a data disk image from an ECS	A data disk image only contains user data. You can create a data disk image from an ECS and then use the image to create new EVS disks. This is a convenient way to migrate data from an ECS to EVS disks.	**Creating a Data Disk Image from an ECS**
Creating a data disk image from an external image file	You can import the data disk image of a local server or a server on another cloud platform to and then the image can be used to create EVS disks.	**Creating a Data Disk Image from an External Image File**
Creating a full-ECS image from an ECS, a CSBS backup, or a CBR backup	You can use an ECS with data disks to create a full-ECS image, complete with an OS, various applications, and your service data. The full-ECS image then can be used to quickly provision identical ECSs for data migration. A full-ECS image can be created by using an ECS, a CSBS backup, or a CBR backup.	• **Creating a Full-ECS Image from an ECS** • **Creating a Full-ECS Image from a CBR Backup**
Creating an ECS from a private image	After a system disk image or full-ECS image is created, you can click **Apply for Server** in the row that contains the image to create an ECS.	**Creating an ECS from an Image**

Table 1-3 Managing private images

Feature	Description	Helpful Link
Modifying an image	To facilitate private image management, you can modify the following attributes of an image: name, description, minimum memory, maximum memory, and advanced functions such as NIC multi-queue and SR-IOV driver.	**Modifying an Image**
Sharing images	You can share an image with other accounts. These accounts can use your shared private image to quickly create ECSs or EVS disks.	• **Sharing Images** • **Image Sharing**
Exporting images	You can export private images to your OBS bucket and download them to your local PC for backup.	• **Exporting an Image** • **Image Exporting**
Encrypting images	You can create encrypted images to improve data security. The encryption mode is KMS envelope encryption. Encrypted images can be created from external image files or encrypted ECSs.	• **Encrypting Images**
Replicating images within a region	By replicating images within a region, you can convert encrypted and unencrypted images into each other or enable some advanced features, for example, quick instance provisioning.	**Replicating Images**
Tagging an image	You can tag your private images for easy management and search.	**Tagging an Image**
Exporting image list	You can export the public or private image list in a given region in CSV format, facilitating local maintenance and query.	**Exporting Image List**
Deleting images	You can delete images that will be no longer used. Deleting an image does not affect the ECSs created from that image.	**Deleting Images**

1.5 Constraints

This section describes the constraints on using IMS.

- **Creating a private image**

- **Importing a private image**
- **Sharing images**
- **Replicating an image**
- **Exporting an image**
- **Encrypting an image**
- **Deleting images**
- **Creating cloud servers from an image**
- **Tagging an image**

Table 1-4 Constraints on creating a private image

Item	Constraint
Maximum number of private images that can be created in a region	50 If you need more, submit a service ticket to increase your quota.
Maximum number of concurrent tasks for creating private images	40 **NOTE** Currently, only one image can be created in each task.
Creating a system disk image from an ECS	• The ECS must be in the **Stopped** or **Running** state.
Creating a data disk image from an ECS	• The ECS must be in the **Stopped** or **Running** state. • A data disk image can be used to create only one data disk at a time.
Disk capacity	• The system disk capacity of an ECS or a BMS used to create a system disk image must be no greater than 1 TB. If it is greater than 1 TB for an ECS, you can only use the ECS to create a full-ECS image. • The data disk capacity of an ECS used to create a data disk image must be no greater than 1 TB. If it is greater than 1 TB, you can only use the ECS to create a full-ECS image.
Creating a full-ECS image from an ECS or a CBR backup	• The ECS must be in the **Stopped** or **Running** state. • A CSBS or CBR backup can be used to create only one full-ECS image at a time. • Only full-ECS images created from CBR backups can be shared. Other full-ECS images cannot be shared. • A full-ECS image cannot be replicated within a region or be exported.

Table 1-5 Constraints on importing a private image

Item	Constraint
Importing a system disk image from an external image file	For details about constraints on external image files, see **Preparing an Image File** or **Preparing an Image File**.
Importing a system disk image from an ISO file	• Register the ISO file as an ISO image, use the ISO image to create a temporary ECS, install an OS and related drivers on the ECS, and use the ECS to create a system disk image. • The ISO image cannot be replicated, exported, or encrypted.
Importing a data disk image from an external image file	The data disk capacity can be 40–2048 GB, and it must also be at least as big as the data disk in the image file.
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	The image size cannot exceed 128 GB. If the image size is between 128 GB and 1 TB, convert the image file into the RAW or ZVHD2 format and import the image through fast import. • For details about how to convert the image file format, see **image format conversion**. • For details about fast import, see **Quickly Importing an Image File**.

Table 1-6 Constrains on sharing images

Item	Constraint
Maximum number of tenants an image can be shared with	System disk image or data disk image: 128 Full-ECS image: 10
Maximum number of shared images that a tenant can receive	No limit
Private image status	**Normal**
Image sharing	Encrypted images and full-ECS images created from a CSBS backup cannot be shared with others.

Item	Constraint
Region	• There are constraints on the region when cloud servers are created from a shared image. For example, a shared image can be used to create cloud servers only in the same region.

Table 1-7 Constraints on replicating an image

Item	Constraint
Maximum size of an image	128 GB
Maximum number of concurrent replication tasks per tenant	5
Private image status	**Normal**
Replicating images within a region	• Full-ECS images cannot be replicated within the same region. • Private images created using ISO files do not support in-region replication.

Table 1-8 Constraints on exporting an image

Item	Constraint
Maximum size of an exported image	1 TB Images larger than 128 GB only support fast export. For details about fast export, see **What Are the Differences Between Import/Export and Fast Import/Export?**.
Formats of exported image files	VMDK, VHD, QCOW2, ZVHD, and ZVHD2
Private image status	**Normal**

Item	Constraint
Exporting an image	• Encrypted images cannot be exported through fast export. • An image can only be exported to a Standard bucket that is in the same region as the image. • The following private images cannot be exported: – Full-ECS images – ISO images – Private images created from a Windows, SUSE, Red Hat, Ubuntu, or Oracle Linux public image • The image size must be less than 1 TB. Images larger than 128 GB support only fast export.

Table 1-9 Constraints on other image operations

Operation	Item	Constraint
Encrypting an image	Creating an encrypted image from an encrypted ECS or an external image file	• An encrypted image cannot be shared with other tenants. • The key used for encrypting an image cannot be changed.
Deleting images	Private image status	A published private image cannot be deleted.
Creating cloud servers from an image	Number of cloud servers that can be concurrently created using a system disk image	Recommended value: ≤ 100
Tagging an image	Maximum number of tags that can be added to a private image	10

Other Constraints

- If an ECS is frozen due to overdue payment, it cannot be used to create a private image. You must renew the ECS before using it to create a private image.
- A private image containing a 32-bit OS cannot be used to create an ECS with larger than 4 GB of memory because the total available address space for a 32-bit OS is 4 GB.

1.6 Supported OSs

1.6.1 OSs Supported by Different Types of ECSs

This section describes the OSs supported by different types of ECSs.

x86 ECSs

- **Table 1-10** lists the OSs supported by the following ECSs:
  General computing S6
  Memory-optimized M6
- **Table 1-11** lists the OSs supported by the following ECSs:
  General computing-plus C6

NOTE

It is recommended that you use the official OS release versions. Do not tailor or customize the release versions, or problems may occur.

OS vendors do not always update OS release versions regularly. Some versions are no longer maintained, and these deprecated versions no longer receive security patches. Ensure that you read the update notifications from OS vendors and update your OS so that it runs properly.

Table 1-10 Supported OS versions

OS	OS Version
Windows	Windows Server 2008 R2 Standard/Enterprise/Datacenter/Web Windows Server 2012 Standard/Datacenter Windows Server 2012 R2 Standard/Datacenter Windows Server 2016 Standard/Datacenter Windows Server 2019 Standard/Datacenter Windows Server Core Version 1709
CentOS	64-bit: CentOS 6.10, 6.9, 6.8, 6.7, 6.6, 6.5, 6.4, and 6.3 64-bit: CentOS 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, and 7.0
Ubuntu	64-bit: Ubuntu 22.04, 20.04, 18.04, 16.04, 14.04, and 12.04 Server
EulerOS	64-bit: EulerOS 2.5, 2.3, and 2.2
Red Hat	64-bit: Red Hat 6.10, 6.9, 6.8, 6.7, 6.6, 6.5, and 6.4 64-bit: Red Hat 7.6, 7.5, 7.4, 7.3, 7.2, 7.1, and 7.0 64-bit: Red Hat 8.0
SUSE Linux Enterprise	64-bit: SLES 11 SP4 and 11 SP3 64-bit: SLES 12 SP4, 12 SP3, 12 SP2, 12 SP1, and 12 64-bit: SLES 15

OS	OS Version
Debian	64-bit: Debian 8.0.0–8.10.0 64-bit: Debian 9.8.0, 9.7.0, 9.6.0, 9.5.0, 9.4.0, 9.3.0, and 9.0.0 64-bit: Debian 10.0.0 64-bit: Debian 11.1.0
openSUSE	64-bit: openSUSE 13.2 64-bit: openSUSE Leap 15.0 and 15.1 64-bit: openSUSE Leap 42.3, 42.2, and 42.1
Fedora	64-bit: Fedora 22–29
CoreOS	64-bit: CoreOS 2079.4.0
FreeBSD	64-bit: FreeBSD 11.0
openEuler	64-bit: openEuler 20.03

Table 1-11 Supported OS versions

OS	OS Version	Kernel Version
Windows	Windows Server 2008 R2 Enterprise/Datacenter/Web/ Standard Windows Server 2012 R2 Standard/Datacenter Windows Server 2016 Standard/Datacenter Windows Server 2019 Datacenter Windows Server Version 1709 Datacenter	10.0.14393 6.1.7600 6.0.6002 6.1.7600 6.3.9600
CentOS	64-bit: CentOS 6 CentOS 7	2.6.32-754.10.1.e16.x86_64 2.6.32-696.16.1.el6.x86_64 2.6.32-754.10.1.el6.x86_64 2.6.32-754.11.1.e16.x86_64 3.10.0-514.10.2.el7.x86_64 3.10.0-693.11.1.el7.x86_64 3.10.0-862.9.1.el7.x86_64 3.10.0-957.5.1.e17.x86_64 3.10.0-957.10.1.e17.x86_64

OS	OS Version	Kernel Version
Ubuntu	64-bit: Ubuntu 14.04 Server Ubuntu 16.04 Server Ubuntu 18.04 Server	4.15.0-52-56 4.4.0-151-178 4.4.0-104-generic 4.4.0-141-generic 4.4.0-142-generic 4.4.0-145-generic 4.15.0-34-generic 4.15.0-45-generic 4.15.0-47-generic
EulerOS	64-bit: EulerOS 2.2 EulerOS 2.3	3.10.0-327.62.59.83.h162.x86_64 3.10.0-514.44.5.10.h198.x86_64 3.10.0-327.59.59.46.h38.x86_64 3.10.0-327.62.59.83.h96.x86_64 3.10.0-327.62.59.83.h128.x86_64 3.10.0-514.44.5.10.h121.x86_64 3.10.0-514.44.5.10.h142.x86_64
Red Hat	64-bit: Red Hat 6 Red Hat 7	2.6.32-358.6.2.el6.x86_64 2.6.32-431.20.3.el6 2.6.32-504.12.2.el6 2.6.32-573.el6.x86_64 2.6.32-696.1.1.el6.x86_64 2.6.32-696.10.2.el6.x86_64 2.6.32-754.el6.x86_64 3.10.0-229.1.2.el7.x86_64 3.10.0-327.36.1.el7.x86_64 3.10.0-514.36.1.el7 3.10.0-514.6.1.el7.x86_64 3.10.0-693.11.6.el7.x86_64 3.10.0-862.3.2.el7.x86_64
SUSE Linux Enterprise	64-bit: SLES 11 SLES 12	3.0.101-108.18-default 3.12.74-60.64.40-default 4.4.103-92.53-default 4.4.120-92.70-default 4.4.121-92.92

OS	OS Version	Kernel Version
Debian	64-bit: Debian 8 Debian 9	4.9.168-1+deb9u3 3.2.0-4-686-pae 3.2.0-4-amd64 3.16.0-4-amd64 4.9.0-3-amd64 4.9.0-4-amd64 4.9.0-8-amd64 4.9.0-9-amd64 4.19.0-5-amd64
openSUSE	64-bit: openSUSE 15.0 openSUSE 15.1	4.4.103-18.41-default 3.0.101-108.18-default
Fedora	64-bit: Fedora 2x	5.1.11-200.fc29.x86_64 4.5.5-300.fc24.x86_64 4.20.8-200.fc29.x86_64 5.2.8-200.fc30.x86_64 4.8.6-300.fc25.x86_64
openEuler	64-bit: openEuler 20.03	4.19.90-2003.4.0.0036.oel.x86_64

1.6.2 External Image File Formats and Supported OSs

External File Formats

Image files in VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ISO, ZVHD2, or ZVHD format can be used to create private images. Select whichever format best meeting your requirements.

Supported OSs

When you upload an external image file to an OBS bucket on the management console, the OS contained in the image file will be checked. **Table 1-12** lists the OSs supported by external image files.

If the OS cannot be identified or is not supported:

- For Windows, **Other_Windows (64_bit)** or **Other_Windows (32_bit)** will be selected during image registration.
- For Linux, **Other_Linux (64_bit)** or **Other_Linux (32_bit)** will be selected during image registration.

NOTE

Uploading image files containing OSs not listed in **Table 1-12** and **Table 1-13** may fail. You are advised to contact the customer service before attempting to upload these image files.

Table 1-12 Supported OSs (x86)

OS	Version
Rocky Linux	Rocky Linux 8.5 64bit Rocky Linux 8.4 64bit Rocky Linux 8.3 64bit
AlmaLinux	AlmaLinux 8.4 64bit AlmaLinux 8.3 64bit
Windows	Windows 10 64bit Windows Server 2019 Standard 64bit Windows Server 2019 Datacenter 64bit Windows Server 2016 Standard 64bit Windows Server 2016 Datacenter 64bit Windows Server 2012 R2 Standard 64bit Windows Server 2012 R2 Essentials 64bit Windows Server 2012 R2 Datacenter 64bit Windows Server 2012 Datacenter 64bit Windows Server 2012 Standard 64bit Windows Server 2008 WEB R2 64bit Windows Server 2008 R2 Standard 64bit Windows Server 2008 R2 Enterprise 64bit Windows Server 2008 R2 Datacenter 64bit
SUSE	SUSE Linux Enterprise Server 15 SP3 64bit SUSE Linux Enterprise Server 15 SP2 64bit SUSE Linux Enterprise Server 15 SP1 64bit SUSE Linux Enterprise Server 15 64bit SUSE Linux Enterprise Server 12 SP5 64bit SUSE Linux Enterprise Server 12 SP4 64bit SUSE Linux Enterprise Server 12 SP3 64bit SUSE Linux Enterprise Server 12 SP2 64bit SUSE Linux Enterprise Server 12 SP1 64bit SUSE Linux Enterprise Server 11 SP4 64bit SUSE Linux Enterprise Server 11 SP3 64bit SUSE Linux Enterprise Server 11 SP3 32bit

OS	Version
Oracle Linux	Oracle Linux Server release 7.6 64bit Oracle Linux Server release 7.5 64bit Oracle Linux Server release 7.4 64bit Oracle Linux Server release 7.3 64bit Oracle Linux Server release 7.2 64bit Oracle Linux Server release 7.1 64bit Oracle Linux Server release 7.0 64bit Oracle Linux Server release 6.10 64bit Oracle Linux Server release 6.9 64bit Oracle Linux Server release 6.8 64bit Oracle Linux Server release 6.7 64bit Oracle Linux Server release 6.5 64bit
Red Hat	Red Hat Linux Enterprise 8.0 64bit Red Hat Linux Enterprise 7.9 64bit Red Hat Linux Enterprise 7.8 64bit Red Hat Linux Enterprise 7.6 64bit Red Hat Linux Enterprise 7.5 64bit Red Hat Linux Enterprise 7.4 64bit Red Hat Linux Enterprise 7.3 64bit Red Hat Linux Enterprise 7.2 64bit Red Hat Linux Enterprise 7.1 64bit Red Hat Linux Enterprise 7.0 64bit Red Hat Linux Enterprise 6.10 64bit Red Hat Linux Enterprise 6.9 64bit Red Hat Linux Enterprise 6.8 64bit Red Hat Linux Enterprise 6.7 64bit Red Hat Linux Enterprise 6.6 64bit Red Hat Linux Enterprise 6.6 32bit Red Hat Linux Enterprise 6.5 64bit Red Hat Linux Enterprise 6.4 64bit Red Hat Linux Enterprise 6.4 32bit

OS	Version
Ubuntu	Ubuntu 20.04 Server 64bit Ubuntu 19.04 Server 64bit Ubuntu 18.04.2 Server 64bit Ubuntu 18.04.1 Server 64bit Ubuntu 18.04 Server 64bit Ubuntu 16.04.6 Server 64bit Ubuntu 16.04.5 Server 64bit Ubuntu 16.04.4 Server 64bit Ubuntu 16.04.3 Server 64bit Ubuntu 16.04.2 Server 64bit Ubuntu 16.04 Server 64bit Ubuntu 14.04.5 Server 64bit Ubuntu 14.04.4 Server 64bit Ubuntu 14.04.4 Server 32bit Ubuntu 14.04.3 Server 64bit Ubuntu 14.04.3 Server 32bit Ubuntu 14.04.1 Server 64bit Ubuntu 14.04.1 Server 32bit Ubuntu 14.04 Server 64bit Ubuntu 14.04 Server 32bit
openSUSE	openSUSE 42.3 64bit openSUSE 42.2 64bit openSUSE 42.1 64bit openSUSE 15.3 64bit openSUSE 15.1 64bit openSUSE 15.0 64bit openSUSE 13.2 64bit openSUSE 11.3 64bit

OS	Version
CentOS	CentOS 8.3 64bit CentOS 8.2 64bit CentOS 8.1 64bit CentOS 8.0 64bit CentOS 8.0 64bit CentOS 7.9 64bit CentOS 7.8 64bit CentOS 7.7 64bit CentOS 7.6 64bit CentOS 7.5 64bit CentOS 7.4 64bit CentOS 7.3 64bit CentOS 7.2 64bit CentOS 7.1 64bit CentOS 7.0 64bit CentOS 7.0 32bit CentOS 6.10 64bit CentOS 6.10 32bit CentOS 6.9 64bit CentOS 6.8 64bit CentOS 6.7 64bit CentOS 6.7 32bit CentOS 6.6 64bit CentOS 6.6 32bit CentOS 6.5 64bit CentOS 6.5 32bit CentOS 6.4 64bit CentOS 6.4 32bit CentOS 6.3 64bit CentOS 6.3 32bit

OS	Version
Debian	Debian GNU/Linux 10.7.0 64bit Debian GNU/Linux 10.5.0 64bit Debian GNU/Linux 10.4.0 64bit Debian GNU/Linux 10.3.0 64bit Debian GNU/Linux 10.2.0 64bit Debian GNU/Linux 10.1.0 64bit Debian GNU/Linux 10.0.0 64bit Debian GNU/Linux 9.13.0 64bit Debian GNU/Linux 9.3.0 64bit Debian GNU/Linux 9.0.0 64bit Debian GNU/Linux 8.10.0 64bit Debian GNU/Linux 8.8.0 64bit Debian GNU/Linux 8.7.0 64bit Debian GNU/Linux 8.6.0 64bit Debian GNU/Linux 8.5.0 64bit Debian GNU/Linux 8.4.0 64bit Debian GNU/Linux 8.2.0 64bit Debian GNU/Linux 8.1.0 64bit
Fedora	Fedora 32 64bit Fedora 31 64bit Fedora 30 64bit Fedora 29 64bit Fedora 28 64bit Fedora 27 64bit Fedora 26 64bit Fedora 25 64bit Fedora 24 64bit Fedora 23 64bit Fedora 22 64bit
EulerOS	EulerOS 2.10 64bit EulerOS 2.9 64bit EulerOS 2.5 64bit EulerOS 2.3 64bit EulerOS 2.2 64bit EulerOS 2.1 64bit

OS	Version
CoreOS	CoreOS 1800.1.0 CoreOS 1745.2.0 CoreOS 1632.0.0 CoreOS 1520.8.0 CoreOS 1465.8.0 CoreOS 1298.5.0 CoreOS 1122.3.0 CoreOS 1122.2.0 CoreOS 1185.5.0 CoreOS 1068.10.0 CoreOS 1010.5.0 CoreOS 1298.6.0
openEuler	openEuler 20.03 64bit
NeoKylin	NeoKylin 7.6 64bit NeoKylin 7.4 64bit NeoKylin Server release 5.0 U2 64bit NeoKylin Linux Advanced Server release 7.0 U5 64bit

Table 1-13 Supported OSs (Arm)

OS	Version
AlmaLinux	AlmaLinux 8.4 64bit AlmaLinux 8.3 64bit
CentOS	CentOS 7.6 64bit CentOS 7.5 64bit CentOS 7.4 64bit
Debian	Debian GNU/Linux 10.2.0 64bit
EulerOS	EulerOS 2.10 64bit EulerOS 2.9 64bit EulerOS 2.8 64bit
Fedora	Fedora 29 64bit
Ubuntu	Ubuntu 20.04 Server 64bit Ubuntu 19.04 Server 64bit Ubuntu 18.04 Server 64bit
SUSE	SUSE Linux Enterprise Server 12 SP5 64bit

OS	Version
openEuler	openEuler 20.03 64bit
openSUSE	openSUSE 15.0 64bit
NeoKylin	NeoKylin V7 64bit NeoKylin 7.7 64bit
UnionTechOS	UOS 20 64bit
Kylin	Kylin V10 64bit Kylin Desktop V10 64bit
Kylinsec	KylinSec 3.3 64bit
NeoShine	iSoft 5.1 64bit

Related Operations

For how to upload an external image file, see **Uploading an External Image File** and **Uploading an External Image File**.

After an external image file is successfully uploaded, you can register this image file as a private image on the cloud platform. For details, see **Registering an External Image File as a Private Image** and **Registering an External Image File as a Private Image**.

1.6.3 OSs Supporting UEFI Boot Mode

The ECS boot mode can be BIOS or UEFI. For details about the differences between them, see **How Is BIOS Different from UEFI?**

Table 1-14 lists the OSs that support the UEFI boot mode.

Table 1-14 OSs supporting UEFI boot mode

OS	Version
Windows	Windows Server 2019 Datacenter 64bit
	Windows Server 2019 Standard 64bit
	Windows Server 2016 Standard 64bit
	Windows Server 2016 Datacenter 64bit
	Windows Server 2012 R2 Standard 64bit
	Windows Server 2012 R2 Datacenter 64bit
	Windows Server 2012 Essentials R2 64bit
	Windows Server 2012 Standard 64bit

OS	Version
	Windows Server 2012 Datacenter 64bit
	Windows 10 64bit
Ubuntu	Ubuntu 19.04 Server 64bit
	Ubuntu 18.04 Server 64bit
	Ubuntu 16.04 Server 64bit
	Ubuntu 14.04 Server 64bit
Red Hat	Red Hat Linux Enterprise 7.4 64bit
	Red Hat Linux Enterprise 7.3 64bit
	Red Hat Linux Enterprise 7.1 64bit
	Red Hat Linux Enterprise 7.0 64bit
	Red Hat Linux Enterprise 6.9 64bit
	Red Hat Linux Enterprise 6.6 32bit
	Red Hat Linux Enterprise 6.5 64bit
Oracle Linux	Oracle Linux Server release 7.4 64bit
	Oracle Linux Server release 6.9 64bit
openSUSE	openSUSE 42.1 64bit
SUSE	SUSE Linux Enterprise Server 12 SP5 64bit
	SUSE Linux Enterprise Server 12 SP1 64bit
	SUSE Linux Enterprise Server 11 SP3 64bit
Fedora	Fedora 29 64bit
	Fedora 24 64bit
Debian	Debian GNU/Linux 8.8.0 64bit
CentOS	CentOS 7.6 64bit
	CentOS 7.5 64bit
	CentOS 7.4 64bit
	CentOS 7.0 64bit
	CentOS 6.9 64bit
	CentOS 6.6 64bit
EulerOS	EulerOS 2.8 64bit
	EulerOS 2.5 64bit

OS	Version
	EulerOS 2.3 64bit
	EulerOS 2.2 64bit
openEuler	openEuler 20.03 64bit
NeoKylin	NeoKylin V7 64bit
UnionTechOS	UOS 20 64bit

1.7 Permissions

If you need to assign different permissions to personnel in your enterprise to access your images, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your resources.

With IAM, you can create IAM users and assign permissions to control their access to specific resources. For example, if you want some software developers in your enterprise to use images but do not want them to delete the images or perform any other high-risk operations, you can create IAM users and grant permission to use the images but not permission to delete them.

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

IAM is a free service. You pay only for the resources in your account. For more information about IAM, see **IAM Service Overview**.

IMS Permissions

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

IMS is a project-level service deployed for specific regions. When you set **Scope** to **Region-specific projects** and select the specified projects in the specified regions, the users only have permissions for images in the selected projects. If you set **Scope** to **All resources**, the users have permissions for images in all region-specific projects. When accessing IMS, the users need to switch to the authorized region.

You can grant permissions by using roles and policies.

- Roles: A coarse-grained authorization strategy provided by IAM to assign permissions based on users' job responsibilities. Only a limited number of service-level roles are available for authorization. Cloud services depend on each other. When you grant permissions using roles, you also need to attach

any existing role dependencies. Roles are not ideal for fine-grained authorization and least privilege access.

Table 1-15 System-defined IMS roles

Role	Description	Dependencies
IMS Administrator	Administrator permissions for IMS	This role depends on the **Tenant Administrator** role.
Server Administrator	Permissions for creating, deleting, querying, modifying, and uploading images	This role depends on the **IMS Administrator** role in the same project.

- Policies (recommended): A fine-grained authorization strategy that defines permissions required to perform operations on specific cloud resources under certain conditions. This type of authorization is more flexible and is ideal for least privilege access. For example, you can grant users only the permission to manage images of a certain type.

  A majority of fine-grained policies contain permissions for specific APIs, and permissions are defined using API actions. For the API actions supported by IMS, see **Permissions and Supported Actions**.

Table 1-16 System-defined policies for IMS

Policy	Description	Dependencies
IMS FullAccess	All permissions for IMS	None
IMS ReadOnlyAccess	Read-only permissions for IMS. Users with these permissions can only view IMS data.	None

Table 1-17 lists the common operations supported by system-defined permissions for IMS.

Table 1-17 Common operations supported by system-defined permissions

Operation	IMS FullAccess	IMS ReadOnlyAccess	IMS Administrator (Depending on Tenant Administrator)
Creating images	√	x	√
Deleting images	√	x	√
Querying images	√	√	√
Updating image information	√	x	√

Helpful Links

- What Is IAM?

1.8 Basic Concepts

1.8.1 Region and AZ

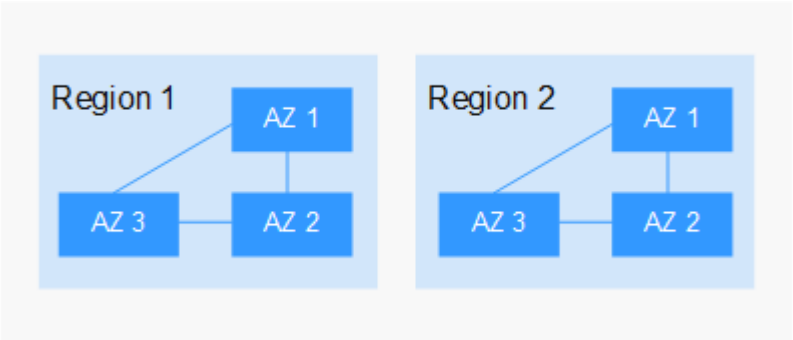
Concept

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

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

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

Figure 1-2 Regions and AZs



Selecting a Region

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

Selecting an AZ

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

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

Regions and Endpoints

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

1.8.2 Common Image Formats

IMS supports multiple image formats, but the system uses ZVHD or ZVHD2 by default.

Table 1-18 lists the common image formats.

Table 1-18 Common image formats

Image Format	Description	Remarks
ZVHD	This format uses the ZLIB compression algorithm and supports sequential read and write.	A universal format supported by IaaS OpenStack; a format supported for imported and exported images **NOTE** ZVHD image files do not support lazy loading. To import large ZVHD image files fast, convert them into ZVHD2 files first.
ZVHD2	This format uses the ZSTD algorithm and supports lazy loading.	A format for the lazy loading feature; a format supported for imported images
QCOW2	This is a disk image supported by the QEMU simulator. It is a file that indicates a block device disk of a fixed size. Compared with the RAW format, the QCOW2 format has the following features: • Supports a lower disk usage. • Supports Copy-On-Write (CoW). The image file only reflects disk changes. • Supports snapshots. • Supports zlib compression and encryption by following Advanced Encryption Standard (AES).	A format supported for imported and exported images

Image Format	Description	Remarks
VMDK	VMDK is a virtual disk format from VMware. A VMDK file represents a physical disk drive of the virtual machine file system (VMFS) on an ECS.	A format supported for imported and exported images
VHD	VHD is a virtual disk file format from Microsoft. A VHD file is a compressed file stored in the file system of the host machine. It mainly contains a file system required for starting ECSs.	A format supported for imported and exported images **NOTE** VHD image files do not support lazy loading. To import large VHD image files fast, convert them into ZVHD2 files first.
VHDX	VHDX is a new VHD format introduced into Hyper-V of Windows Server 2012 by Microsoft. Compared with the VHD format, VHDX has a larger storage capacity. It provides protection against data damage during power supply failures, and the disk structure alignment has been optimized to prevent performance degradation of new physical disks in a large sector.	A format supported for imported images
RAW	A RAW file can be directly read and written by ECSs. This format delivers higher I/O performance but does not support dynamic space expansion.	A format supported for imported images
QCOW	QCOW manages the space allocation of an image through the secondary index table. The secondary index uses the memory cache technology and needs the query operation, which results in performance loss. The performance of QCOW is inferior to that of QCOW2, and the read and write performance is inferior to that of RAW.	A format supported for imported images
VDI	VDI is the disk image file format used by the VirtualBOX virtualization software from Oracle. It supports snapshots.	A format supported for imported images

Image Format	Description	Remarks
QED	The QED format is an evolved version of the QCOW2 format. Its storage location query mode and data block size are the same as those of the QCOW2 format. However, QED implements Copy-On-Write (CoW) in a different way as it uses a dirty flag to replace the reference count table of QCOW2.	A format supported for imported images

1.9 Related Services

Figure 1-3 shows the relationships between IMS and other services.

Figure 1-3 IMS relationships with other services

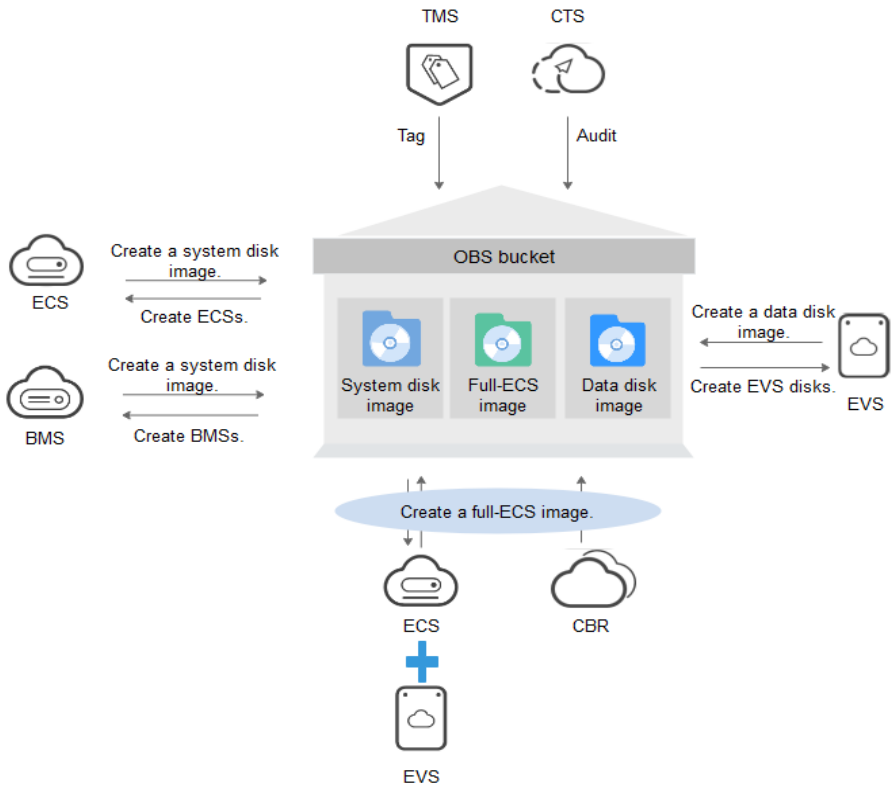


Table 1-19 Related services

Service	Relationship with IMS	Related Operation
Elastic Cloud Server (ECS)	You can use an image to create ECSs or use an ECS to create an image.	• **Creating an ECS from an Image** • **Creating a System Disk Image from a Windows ECS** • **Creating a System Disk Image from a Linux ECS**
Bare Metal Server (BMS)	You can use an image to create BMSs or use a BMS to create an image.	**Creating a BMS System Disk Image**
Object Storage Service (OBS)	Images are stored in OBS buckets. External image files to be uploaded to the system are stored in OBS buckets, and private images are exported to OBS buckets.	**Exporting an Image**
Data Encryption Workshop (DEW)	Images can be encrypted through envelope encryption of DEW to ensure data security. The keys used for encrypting images are stored in DEW.	**Encrypting Images**
Elastic Volume Service (EVS)	You can create a data disk image using a data disk of an ECS. The created data disk image can be used to create other EVS disks.	**Creating a Data Disk Image from an ECS**
Cloud Backup and Recovery (CBR)	You can use a CBR backup to create a full-ECS image.	**Creating a Full-ECS Image from a CBR Backup**
Tag Management Service (TMS)	You can add tags to images for convenient classification and search.	**Tagging an Image**

Service	Relationship with IMS	Related Operation
Cloud Trace Service (CTS)	CTS records IMS operations for query, auditing, or backtracking.	**Auditing Key Operations**

2 Creating a Private Image

2.1 Introduction

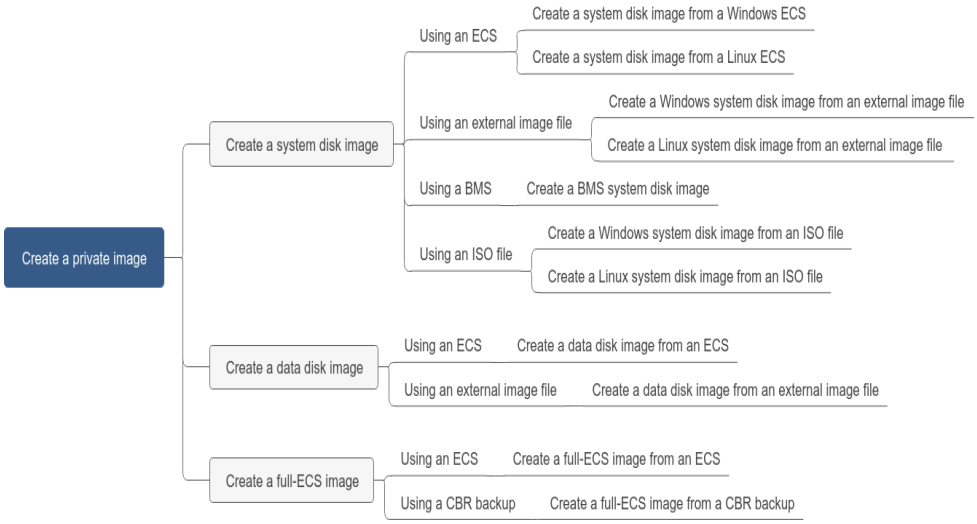
A private image is an image available only to the user who created it. It contains an OS, preinstalled public applications, and a user's personal applications. A private image can be a system disk image, data disk image, or full-ECS image. It can be created from a cloud server or an external image file.

Creating a private image does not affect the running of services on the cloud server or cause data loss.

This section describes how to create a private image using any of the following methods:

- **Creating a System Disk Image from a Windows ECS**
- **Creating a System Disk Image from a Linux ECS**
- **Creating a Windows System Disk Image from an External Image File**
- **Creating a Linux System Disk Image from an External Image File**
- **Creating a BMS System Disk Image**
- **Creating a Data Disk Image from an ECS**
- **Creating a Data Disk Image from an External Image File**
- **Creating a Full-ECS Image from an ECS**
- **Creating a Full-ECS Image from a CBR Backup**
- **Creating a Windows System Disk Image from an ISO File**
- **Creating a Linux System Disk Image from an ISO File**

Figure 2-1 Creating a private image



After a system disk image is created, you can use it to create an ECS or change the OS of an ECS.

2.2 Creating a System Disk Image from a Windows ECS

Scenarios

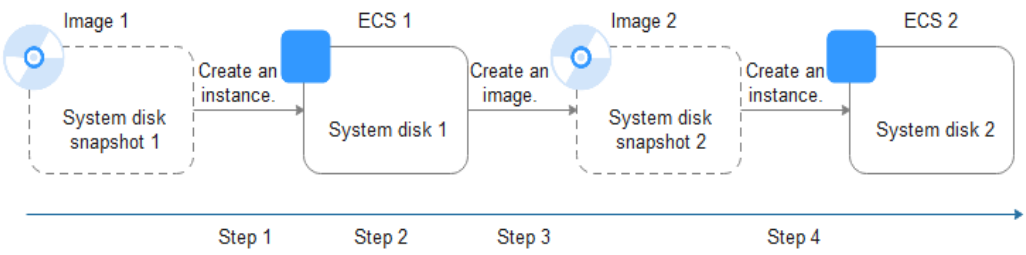
If you have created and configured a Windows ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software and environment preinstalled.

Creating a system disk image does not affect the running of services on the ECS or cause data loss.

Background

The following figure shows the process of creating a system disk image from an ECS.

Figure 2-2 Creating a system disk image and using it to create ECSs



- System disk images are often used for application scale-out. They can also be used for hybrid cloud deployment. You can create system disk images for resource synchronization on and off cloud. The procedure is as follows:
  a. Create a system disk image from an ECS.

     NOTE

     The ECS must be created from a private image. If it is created from a public image, the system disk image cannot be exported.

  b. Export the image to an OBS bucket. For details, see **Exporting an Image**.
  c. Download the image file from the OBS bucket.
- You can create an image from a running ECS.

  The image creation does not affect service running on the ECS.

  In this process, do not stop, start, or restart the ECS, or the image creation may fail.
- The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks.
- A system disk image will be created in the same region as the ECS that was used to create it.
- If an ECS has expired or been released, you can use the system disk image created from the ECS to restore it.

Prerequisites

Before creating a private image from an ECS:

- Delete any sensitive data the ECS may contain.
- Ensure that the ECS is in the **Running** or **Stopped** state.
- Check network configuration of the ECS and ensure that DHCP is configured for the NICs. Enable remote desktop connection if needed. For details, see **Setting the NIC to DHCP** and **Enabling Remote Desktop Connection**.
- Check whether Cloudbase-Init has been installed on the ECS. The user data injection function on the management console is only available for new ECSs that have this tool installed. You can use data injection, for example, to set the login password for a new ECS. For details, see **Installing and Configuring Cloudbase-Init**.
- Check and install VirtIO drivers to ensure that new ECSs created from the image support KVM virtualization and to improve network performance.

  For details, see steps **2** to **4** in **Optimization Process**.
- Run Sysprep to ensure that the SIDs of the new ECSs created from the image are unique within their domain. In a cluster deployment scenario, the SIDs must be unique. For details, see **Running Sysprep**.

NOTE

If an ECS is created from a public image, Cloudbase-Init has been installed by default. You can follow the guide in the prerequisites to verify the installation.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a system disk image.

1. Click **Create Image** in the upper right corner.
2. Set image parameters.

   Table 2-1 and Table 2-2 list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-1 Image type and source

Parameter	Description
Type	Select **Create Image**.
Image Type	Select **System disk image**.
Source	Select **ECS** and select an ECS with required configurations.

Table 2-2 Image information

Parameter	Description
Encryption	This parameter specifies whether the image will be encrypted. The value is provided by the system and cannot be changed. – Only an unencrypted private image can be created from an unencrypted ECS. – Only an encrypted private image can be created from an encrypted ECS.
Name	Set a name for the image.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or your account is an enterprise account. To enable this function, contact your customer manager. An enterprise project provides central management of cloud resources on a project.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.

Parameter	Description
Description	(Optional) Enter a description of the image.

3. Click **Apply Now**.
4. Confirm the settings and click **Submit Application**.

Step 3 Go back to the **Private Images** page and view the new system disk image.

The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks. When the image status changes to **Normal**, the image creation is complete.

NOTE

- Do not perform any operations on the selected ECS or its associated resources during image creation.
- An ECS created from an encrypted image is also encrypted. The key used for encrypting the ECS is the same as that used for encrypting the image.
- An image created from an encrypted ECS is also encrypted. The key used for encrypting the image is the same as that used for encrypting the ECS.

----End

Follow-up Procedure

After a system disk image is created, you can:

- Use the image to create new ECSs. For details, see **Creating an ECS from an Image**.
- Use the image to change the OSs of existing ECSs.

2.3 Creating a System Disk Image from a Linux ECS

Scenarios

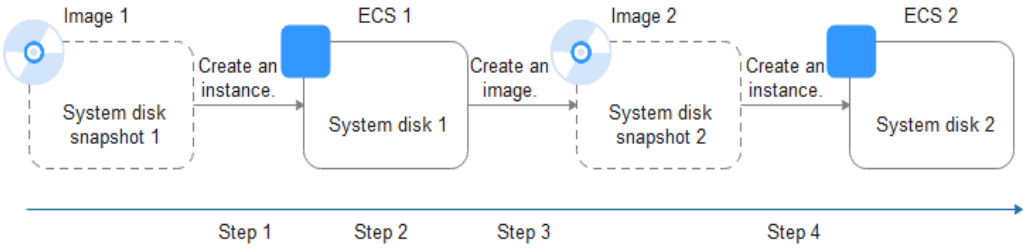
If you have created and configured a Linux ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software and environment preinstalled.

Creating a system disk image does not affect the running of services on the ECS or cause data loss.

Background

The following figure shows the process of creating a system disk image from an ECS.

Figure 2-3 Creating a system disk image and using it to create ECSs



- System disk images are often used for application scale-out. They can also be used for hybrid cloud deployment. You can create system disk images for resource synchronization on and off cloud. The procedure is as follows:
  a. Create a system disk image from an ECS.

     NOTE

     If the ECS is created from any of the following images, the system disk image cannot be exported:
     - ISO image
     - Private image created from a SUSE, Red Hat, Ubuntu, or Oracle Linux public image

  b. Export the image to an OBS bucket. For details, see **Exporting an Image**.
  c. Download the image file from the OBS bucket.
- You can create an image from a running ECS.

  The image creation does not affect service running on the ECS.

  In this process, do not stop, start, or restart the ECS, or the image creation may fail.
- The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks.
- A system disk image will be created in the same region as the ECS that was used to create it.
- If an ECS has expired or been released, you can use the system disk image created from the ECS to restore it.

Prerequisites

Before creating a private image from an ECS:

- Delete any sensitive data the ECS may contain.
- Ensure that the ECS is in the **Running** or **Stopped** state.
- Check network configuration of the ECS and ensure that DHCP is configured for the NICs. For details, see **Setting the NIC to DHCP**.
- Check whether Cloud-Init has been installed on the ECS. The user data injection function on the management console is only available for new ECSs that have this tool installed. You can use data injection, for example, to set the login password for a new ECS. For details, see **Installing Cloud-Init** and **Configuring Cloud-Init**.
- Delete any network rules to prevent NIC name drift on the ECSs created from the image. For details, see **Deleting Files from the Network Rule Directory**.

- To ensure that the ECSs created from the image support KVM virtualization, the Linux ECS used to create the image has to be modified. For instance, the disk IDs in the GRUB and fstab files need to be UUID and native KVM drivers need to be installed.

  For details, see **Optimization Process**.
- If multiple data disks are attached to an ECS used to create a private image, the ECSs created from the image may be unavailable. You need to detach all data disks from the ECS before using it to create an image. For details, see **Detaching Data Disks from an ECS**.
- If data disks have been attached to the ECS and automatic partition mounting has been configured in the fstab file for the ECS, delete these configurations from the file before using the ECS to create a system disk image.

NOTE

If an ECS is created from a public image, Cloud-Init has been installed by default. You can follow the guide to verify the installation.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a system disk image.

1. Click **Create Image** in the upper right corner.
2. Set image parameters.

   Table 2-3 and **Table 2-4** list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-3 Image type and source

Parameter	Description
Type	Select **Create Image**.
Image Type	Select **System disk image**.
Source	Select **ECS** and select an ECS with required configurations.

Table 2-4 Image information

Parameter	Description
Encryption	This parameter specifies whether the image will be encrypted. The value is provided by the system and cannot be changed. – Only an unencrypted private image can be created from an unencrypted ECS. – Only an encrypted private image can be created from an encrypted ECS.
Name	Set a name for the image.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or your account is an enterprise account. To enable this function, contact your customer manager. An enterprise project provides central management of cloud resources on a project.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

3. Click **Apply Now**.
4. Confirm the settings and click **Submit Application**.

Step 3 Go back to the **Private Images** page and view the new system disk image.

The time required for creating an image depends on the ECS system disk size, network quality, and the number of concurrent tasks. When the image status changes to **Normal**, the image creation is complete.

NOTE

- Do not perform any operations on the selected ECS or its associated resources during image creation.
- An ECS created from an encrypted image is also encrypted. The key used for encrypting the ECS is the same as that used for encrypting the image.
- An image created from an encrypted ECS is also encrypted. The key used for encrypting the image is the same as that used for encrypting the ECS.

----End

Follow-up Procedure

After a system disk image is created, you can:

- Use the image to create new ECSs. For details, see **Creating an ECS from an Image**.

- Use the image to change the OSs of existing ECSs.

2.4 Creating a Windows System Disk Image from an External Image File

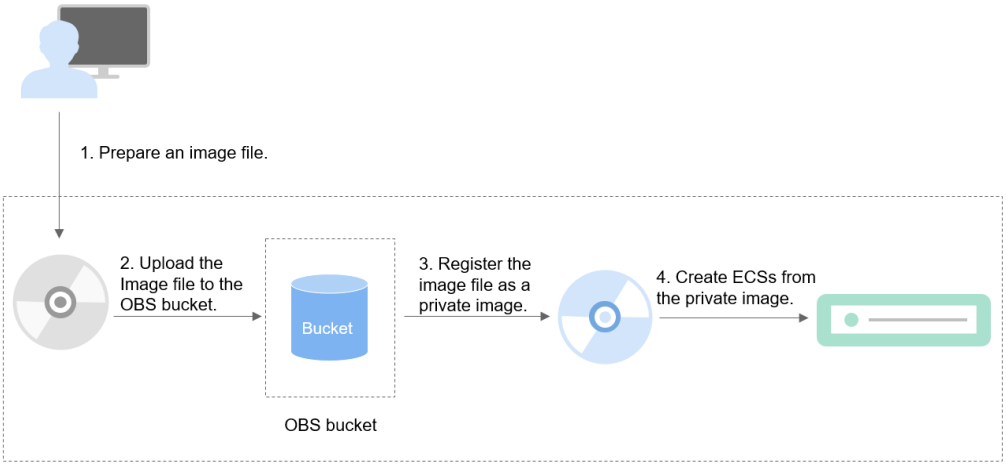
2.4.1 Overview

You can import a local image or a system disk image from another cloud platform to the current cloud. After an image is imported, you can use it to create ECSs or reinstall the OSs of existing ECSs.

Creation Process

Figure 2-4 shows the process of creating a private image.

Figure 2-4 Creating a Windows system disk image



As shown in the figure, the following steps are required to register an external image file as a private image:

1. Prepare an external image file that meets platform requirements. For details, see **Preparing an Image File**.
2. Upload the external image file to your OBS bucket. For details, see **Uploading an External Image File**.
3. On the management console, select the uploaded image file and register it as a private image. For details, see **Registering an External Image File as a Private Image**.
4. After the private image is registered, you can use it to create ECSs. For details, see **Creating a Windows ECS from an Image**.

2.4.2 Preparing an Image File

You need to prepare an image file that can be used to create a private image.

NOTE

Currently, a large image file (maximum: 1 TB) can be imported only in RAW or ZVHD2 format. In addition to the requirements described in **Table 2-6**, a bitmap file needs to be generated for each RAW image file. The bitmap file is uploaded together with the image file. For details, see **Quickly Importing an Image File**.

Initial Configuration for an Image File

The initial configuration must be completed on the source VM before an image file is exported from it. If you did not configure it, use the image file to create an ECS, configure the ECS, and use the ECS to create a private image. For details, see **What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?**

Table 2-5 Initial configuration for an image file

Configuration Item	How to Configure
Network	The NIC must be set to DHCP. Otherwise, the ECS startup or network capability will be abnormal. For details, see: **Setting the NIC to DHCP** The following operations are optional: • Enabling NIC multi-queue NIC multi-queue enables multiple vCPUs to process NIC interruptions, thereby improving network PPS and I/O performance. For details, see **How Do I Enable NIC Multi-Queue for an Image?** • Configuring dynamic assignment of IPv6 addresses IPv6 addresses are used to deal with IPv4 address exhaustion. If dynamic configuration is enabled in an image file, the ECSs created from this file will be automatically assigned an IPv6 address. These ECSs will support both IPv4 and IPv6 addresses. Configure dynamic assignment of IPv6 addresses. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Configuration Item	How to Configure
Tools	You are advised to install Cloudbase-Init. Cloudbase-Init is an open-source tool for cloud instance initialization. When creating ECSs from an image with Cloudbase-Init, you can use user data injection to inject customized initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloudbase-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs. For details, see **Installing and Configuring Cloudbase-Init**. If each of your ECSs requires a unique SID in a domain, run Sysprep after Cloudbase-Init is installed. For details, see **Running Sysprep**.
Drivers	An ECS can run properly only after KVM Guest OS drivers (VirtIO drivers) are installed on it. To ensure that ECSs support KVM and to improve network performance, VirtIO drivers must be installed for the image. **Installing VirtIO drivers**

Image File Properties

Table 2-6 Windows image file properties

Image File Property	Requirement
OS	• Windows Server 2008, Windows Server 2012, Windows Server 2016 • 32-bit or 64-bit • The OS cannot be bound to specific hardware. • The OS must support full virtualization. For details about the supported OS versions, see **External Image File Formats and Supported OSs**. These OSs support automatic configuration. For details, see **What Will the System Do to an Image File When I Use the File to Register a Private Image?** For other OSs, check and install Guest OS drivers. On the image registration page, select **Other Windows**. After the image is imported, whether the system is started depends on the driver integrity.
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD

Image File Property	Requirement
Image size	Maximum file size: 128 GB If the image size is between 128 GB and 1 TB, convert the image file into the RAW or ZVHD2 format and import the image using fast import. • For details about how to convert the image file format, see **image format conversion**. • For details about fast import, see **fast image file import**.

Other

- Currently, images with data disks cannot be created. The image file must contain only a system disk, and the system disk size must be [40 GB, 1024 GB].
- The initial password in the image file must contain uppercase letters, lowercase letters, digits, and special characters (!@$%^-_=+[{}]:,./?).
- The boot partition and system partition must be on the same disk.
- For an external image file, you need a tenant administrator account and password combination.
- Generally, the boot mode is BIOS in an image. Some OS images support the UEFI boot mode. For details, see **OSs Supporting UEFI Boot Mode**.
- The image file cannot be encrypted, or ECSs created from the registered image may not work properly.

2.4.3 Uploading an External Image File

You are advised to use OBS Browser+ to upload external image files to OBS buckets. For details, see *Object Storage Service User Guide*.

NOTE

- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to the OBS bucket.
- The storage class of the OBS bucket must be Standard.
- If you want to create a data disk image along with the system disk image, you also need to upload an image file containing data disks to the OBS bucket. You can create one system disk image and no more than three data disk images.

2.4.4 Registering an External Image File as a Private Image

Scenarios

Register an image file uploaded to the OBS bucket as a private image.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Register an external image file as a private image.

1. Click **Create Image** in the upper right corner.
2. Set image parameters.

   Table 2-7 and Table 2-8 list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-7 Image type and source

Parameter	Description
Type	Select **Import Image**.
Image Type	Select **System disk image**.
Source	Select the bucket storing the image file from the list and then select the image file.
Enable Fast Create	This parameter is available only when you select a ZVHD2 or RAW image file. This function enables fast image creation and supports import of large files (maximum: 1 TB) as long as the files to be uploaded are converted to ZVHD2 or RAW format and optimized. If you have a file that meets the requirements, select **Enable Fast Create** and select the confirmation information following **Image File Preparation**. **NOTE** To learn how to convert image file formats and generate bitmap files, see **Quickly Importing an Image File**.

Table 2-8 Image information

Parameter	Description
Enable automatic configuration	If you select this option, the system will automatically check and optimize the image file. For details, see **What Will the System Do to an Image File When I Use the File to Register a Private Image?**
Function	Specifies whether the image is used to create ECSs or BMSs. The value can be **ECS system disk image** or **BMS system disk image**. This section uses **ECS system disk image** as an example.

Parameter	Description
Boot Mode	This parameter is optional. The value can be **BIOS** or **UEFI**. For details about the differences between the two, see **How Is BIOS Different from UEFI?** For details about which OSs support UEFI boot, see **OSs Supporting UEFI Boot Mode**. The boot mode must be the same as that in the image file. You need to confirm which boot mode is used in the image file. After you select the correct boot mode, the boot mode will be configured for the image at the background. Select the right boot mode, or ECSs created using the image will not be able to boot up.
OS	To ensure that the image can be created and used properly, select an OS consistent with that in the image file. If you do not select an OS, the system attempts to automatically identify the OS in the image file. **NOTE** – If the system detects that the image file OS is different from the one you selected, the OS detected by the system will be used. – If the system cannot detect the OS in the image file, the OS you selected will be used. – If the OS you selected or identified by the system is incorrect, ECSs created from the image file may be affected.
System Disk (GB)	The system disk capacity (value range: 40 GB to 1024 GB). Ensure that this value is at least equal to the system disk capacity in the image file. **NOTE** If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk capacity based on **What Do I Do If the System Disk Capacity in a VHD Image File Exceeds the One I Have Specified on the Management Console When I Use This File to Register a Private Image?**
Data Disk (GB)	You can also add data disks to the image. You need to obtain an image file containing data disks in advance. This function is used to migrate VMs and data disks from other platforms to the current platform. To add data disks, click  , set the data disk capacity, and click **Select Image File**. In the displayed dialog box, select the target bucket and then the target image file containing the data disk. A maximum of three data disks can be added.
Name	Set a name for the image.

Parameter	Description
Encryption	(Optional) If you want to encrypt the image, select **KMS encryption** and select the key to be used from the key list. After you select **KMS encryption**, the system will create a default key **ims/default** for you. You can also select a key from the key list. For how to encrypt an image, see **Creating Encrypted Images**.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or your account is an enterprise account. An enterprise project provides central management of cloud resources on a project by project basis.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

3. Click **Apply Now**, confirm the configurations, and click **Submit Application**.

Step 3 Go back to the **Private Images** page. The image is successfully registered when its status becomes **Normal**.

If you add data disks during image creation, a system disk image and data disk images will be generated. The number of data disk images depends on the number of data disks you add (a maximum of 3).

NOTE

The time required for image registration is determined by the image file size. You may need to wait a long period of time for the image file to be successfully registered as a private image.

----End

2.4.5 Creating a Windows ECS from an Image

Scenarios

After registering an external image file as a private image on the cloud platform, you can use the image to create ECSs or change the OSs of existing ECSs.

This section describes how to create an ECS from an image.

Procedure

Create an ECS by referring to **Creating an ECS from an Image**.

Note the following when setting the parameters:

- **Region**: Select the region where the private image is located.
- **Specifications**: Select a flavor based on the OS type in the image and the OS versions described in **OSs Supported by Different Types of ECSs**.
- **Image**: Select **Private image** and then the created image from the drop-down list.
- (Optional) **Data Disk**: Add data disks. These data disks are created from a data disk image generated together with a system disk image. In this way, you can migrate the data of data disks together with system disk data from the VM on the original platform to the current cloud platform.

Follow-up Procedure

After a system disk image is created, you can use it to change the OS of an ECS.

2.5 Creating a Linux System Disk Image from an External Image File

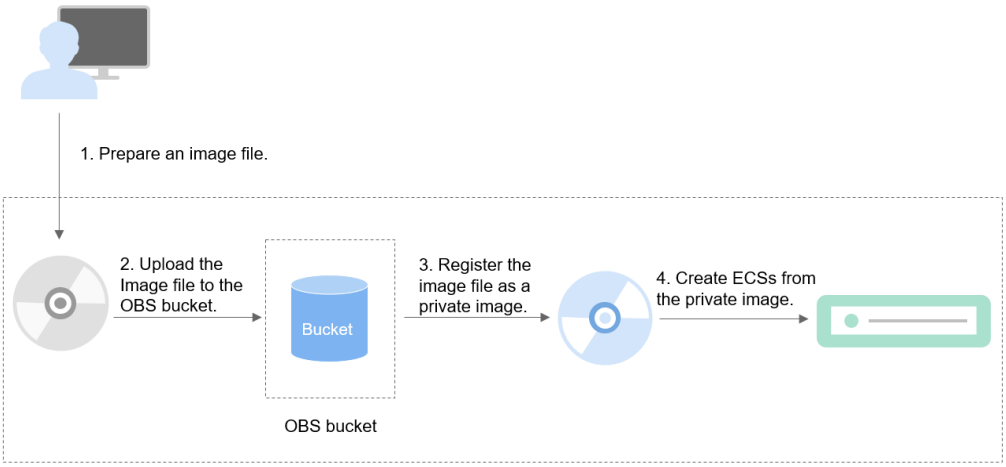
2.5.1 Overview

You can import a local image or a system disk image from another cloud platform to the current cloud. After an image is imported, you can use it to create ECSs or reinstall the OSs of existing ECSs.

Creation Process

Figure 2-5 shows the process of creating a private image.

Figure 2-5 Creating a Linux system disk image



The procedure is as follows:

1. Prepare an external image file that meets platform requirements. For details, see **Preparing an Image File**.
2. Upload the external image file to your OBS bucket. For details, see **Uploading an External Image File**.

3. On the management console, select the uploaded image file and register it as a private image. For details, see **Registering an External Image File as a Private Image**.
4. After the private image is registered, you can use it to create ECSs. For details, see **Creating a Linux ECS from an Image**.

2.5.2 Preparing an Image File

You need to prepare an image file that can be used to create a private image.

NOTE

Currently, a large image file (maximum: 1 TB) can be imported only in RAW or ZVHD2 format. In addition to the requirements described in **Table 2-10**, a bitmap file needs to be generated for each RAW image file. The bitmap file is uploaded together with the image file. For details, see **Quickly Importing an Image File**.

Initial Configuration for an Image File

The initial configuration must be completed on the source VM before an image file is exported from it. If you did not configure it, use the image file to create an ECS, configure the ECS, and use the ECS to create a private image. For details, see **What Do I Do If a Linux Image File Is Not Pre-Configured When I Use It to Register a Private Image?**

Table 2-9 Initial configuration for an image file

Configuration Item	How to Configure
Network	The NIC must be set to DHCP and files must be deleted from the network role directory. Otherwise, the ECS startup or network capability will be abnormal. For details, see: • **Deleting files from the network rule directory** • **Setting the NIC to DHCP** The following value-added operations are optional: • Enabling NIC multi-queue NIC multi-queue enables multiple vCPUs to process NIC interruptions, thereby improving network PPS and I/O performance. For details, see **How Do I Enable NIC Multi-Queue for an Image?** • Configuring dynamic assignment of IPv6 addresses IPv6 addresses are used to deal with IPv4 address exhaustion. If dynamic configuration is enabled in an image file, the ECSs created from this file will be automatically assigned an IPv6 address. These ECSs will support both IPv4 and IPv6 addresses. Configure dynamic assignment of IPv6 addresses. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Configuration Item	How to Configure
Tools	You are advised to install Cloud-Init. Cloud-Init is an open-source tool for cloud instance initialization. When creating ECSs from an image with Cloud-Init, you can use user data injection to inject customized initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloud-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs. For details, see **Installing Cloud-Init**.
Drivers	**Installing native KVM drivers**
File system	• **Changing the disk identifier in the GRUB configuration file to UUID** • **Changing the disk identifier in the fstab file to UUID**
Data disks	If multiple data disks are attached to the ECS used to create a private image, ECSs created from the image may be unavailable. Therefore, you need to detach all data disks from the ECS before using it to create a private image. For details, see **Detaching Data Disks from an ECS**.

Image File Properties

Table 2-10 Linux image file properties

Image File Property	Requirement
OS	• SUSE, Oracle Linux, Red Hat, Ubuntu, openSUSE, CentOS, Debian, Fedora, EulerOS, and NeoKylin • 32-bit or 64-bit • The OS cannot be bound to specific hardware. • The OS must support full virtualization. For details about the supported OS versions, see **External Image File Formats and Supported OSs**. These OSs support automatic configuration. For details, see **What Will the System Do to an Image File When I Use the File to Register a Private Image?** For other OSs, check and install VirtIO drivers (see **Installing Native KVM Drivers**). On the image registration page, select **Other Linux**. After the image is imported, whether the system is started depends on the driver integrity.

Image File Property	Requirement
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	Maximum file size: 128 GB If the image size is between 128 GB and 1 TB, convert the image file into the RAW or ZVHD2 format and import the image using fast import. • For details about how to convert the image file format, see **image format conversion**. • For details about fast import, see **fast image file import**.

Other

- Currently, images with data disks cannot be created. The image file must contain only a system disk, and the system disk size must be [40 GB, 1024 GB].
- The initial password in the image file must contain uppercase letters, lowercase letters, digits, and special characters (!@$%^-_=+[{}]:,./?).
- The boot partition and system partition must be on the same disk.
- Generally, the boot mode is BIOS in an image. Some OS images support the UEFI boot mode. For details, see "OSs Supporting UEFI Boot Mode" in *Image Service Management User Guide*.
- The image file cannot be encrypted, or ECSs created from the registered image may not work properly.
- The **/etc/fstab** file cannot contain automatic mounting information of non-system disks. Otherwise, the login to the created ECS may fail.
- If the external image file uses LVM as the system disk, ECSs created from the private image do not support file injection.
- If the VM where the external image file is located has been shut down, it must be a graceful shutdown. Otherwise, a blue screen may occur when the ECS created from the private image is started.

2.5.3 Uploading an External Image File

You are advised to use OBS Browser+ to upload external image files to OBS buckets. For details, see *Object Storage Service User Guide*.

NOTE

- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to the OBS bucket.
- The storage class of the OBS bucket must be Standard.
- If you want to create a data disk image along with the system disk image, you also need to upload an image file containing data disks to the OBS bucket. You can create one system disk image and no more than three data disk images.

2.5.4 Registering an External Image File as a Private Image

Scenarios

Register an image file uploaded to the OBS bucket as a private image.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
   The IMS console is displayed.

Step 2 Register an external image file as a private image.

1. Click **Create Image** in the upper right corner.
2. Set image parameters.
   Table 2-11 and Table 2-12 list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 2-11 Image type and source

Parameter	Description
Type	Select **Import Image**.
Image Type	Select **System disk image**.
Source	Select the bucket storing the image file from the list and then select the image file.
Enable Fast Create	This parameter is available only when you select a ZVHD2 or RAW image file. This function enables fast image creation and supports import of large files (maximum: 1 TB) as long as the files to be uploaded are converted to ZVHD2 or RAW format and optimized. If you have a file that meets the requirements, select **Enable Fast Create** and select the confirmation information following **Image File Preparation**. **NOTE** To learn how to convert image file formats and generate bitmap files, see **Quickly Importing an Image File**.

Table 2-12 Image information

Parameter	Description
Enable automatic configuration	If you select this option, the system will automatically check and optimize the image file. For details, see **What Will the System Do to an Image File When I Use the File to Register a Private Image?**
Function	Specifies whether the image is used to create ECSs or BMSs. The value can be **ECS system disk image** or **BMS system disk image**. This section uses **ECS system disk image** as an example.
Boot Mode	This parameter is optional. The value can be **BIOS** or **UEFI**. For details about the differences between the two, see **How Is BIOS Different from UEFI?** For details about which OSs support UEFI boot, see **OSs Supporting UEFI Boot Mode**. The boot mode must be the same as that in the image file. You need to confirm which boot mode is used in the image file. After you select the correct boot mode, the boot mode will be configured for the image at the background. Select the right boot mode, or ECSs created using the image will not be able to boot up.
OS	To ensure that the image can be created and used properly, select an OS consistent with that in the image file. If you do not select an OS, the system attempts to automatically identify the OS in the image file. **NOTE** – If the system detects that the image file OS is different from the one you selected, the OS detected by the system will be used. – If the system cannot detect the OS in the image file, the OS you selected will be used. – If the OS you selected or identified by the system is incorrect, ECSs created from the image file may be affected.
System Disk (GB)	The system disk capacity (value range: 40 GB to 1024 GB). Ensure that this value is at least equal to the system disk capacity in the image file. **NOTE** If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk capacity based on **What Do I Do If the System Disk Capacity in a VHD Image File Exceeds the One I Have Specified on the Management Console When I Use This File to Register a Private Image?**

Parameter	Description
Data Disk (GB)	You can also add data disks to the image. You need to obtain an image file containing data disks in advance. This function is used to migrate VMs and data disks from other platforms to the current platform. To add data disks, click  , set the data disk capacity, and click **Select Image File**. In the displayed dialog box, select the target bucket and then the target image file containing the data disk. A maximum of three data disks can be added.
Name	Set a name for the image.
Encryption	(Optional) If you want to encrypt the image, select **KMS encryption** and select the key to be used from the key list. After you select **KMS encryption**, the system will create a default key **ims/default** for you. You can also select a key from the key list. For how to encrypt an image, see **Creating Encrypted Images**.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or your account is an enterprise account. An enterprise project provides central management of cloud resources on a project by project basis.
Tag	(Optional) Set a tag key and a tag value for the image to make identification and management of your images easier.
Description	(Optional) Enter a description of the image.

3. Click **Apply Now**, confirm the configurations, and click **Submit Application**.

Step 3 Go back to the **Private Images** page. The image is successfully registered when its status becomes **Normal**.

If you add data disks during image creation, a system disk image and data disk images will be generated. The number of data disk images depends on the number of data disks you add (a maximum of 3).

NOTE

The time required for image registration is determined by the image file size. You may need to wait a long period of time for the image file to be successfully registered as a private image.

----End

2.5.5 Creating a Linux ECS from an Image

Scenarios

After registering an external image file as a private image on the cloud platform, you can use the image to create ECSs or change the OSs of existing ECSs.

This section describes how to create an ECS from an image.

Procedure

Create an ECS by referring to **Creating an ECS from an Image**.

Note the following when setting the parameters:

- **Region**: Select the region where the private image is located.
- **Specifications**: Select a flavor based on the OS type in the image and the OS versions described in **OSs Supported by Different Types of ECSs**.
- **Image**: Select **Private image** and then the created image from the drop-down list.
- (Optional) **Data Disk**: Add data disks. These data disks are created from a data disk image generated together with a system disk image. In this way, you can migrate the data of data disks together with system disk data from the VM on the original platform to the current cloud platform.

Follow-up Procedure

After a system disk image is created, you can use it to change the OS of an ECS.

2.6 Creating a BMS System Disk Image

For how to create a BMS private image, see *Bare Metal Server User Guide*.

2.7 Creating a Data Disk Image from an ECS

Scenarios

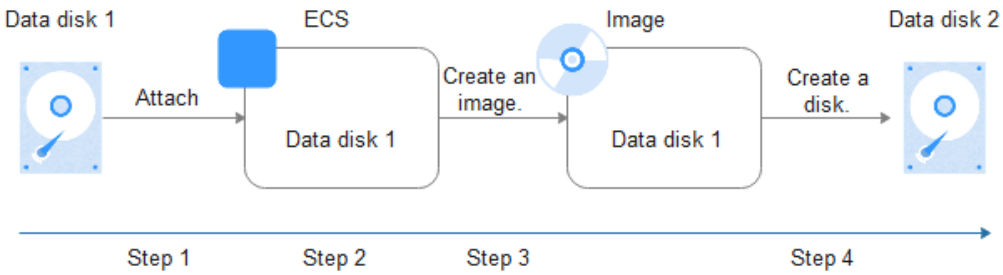
A data disk image contains only service data. You can create a data disk image from an ECS and then use the image to create new EVS disks. This is a convenient way to migrate data from an ECS to EVS disks.

For example, you can create a data disk image to clone the data of an ECS whose disk is about to expire.

Background

The following figure shows the process of creating a data disk image from an ECS.

Figure 2-6 Creating a data disk image and using it to create data disks



Prerequisites

- A data disk has been attached to the ECS, and the ECS is running or stopped. For details about how to attach a data disk, see *Elastic Cloud Server User Guide*.
- The data disk capacity of the ECS must be no greater than 1 TB.

  If the capacity is greater than 1 TB, you can only use the ECS to create a full-ECS image.

Procedure

Step 1 Access the IMS console.

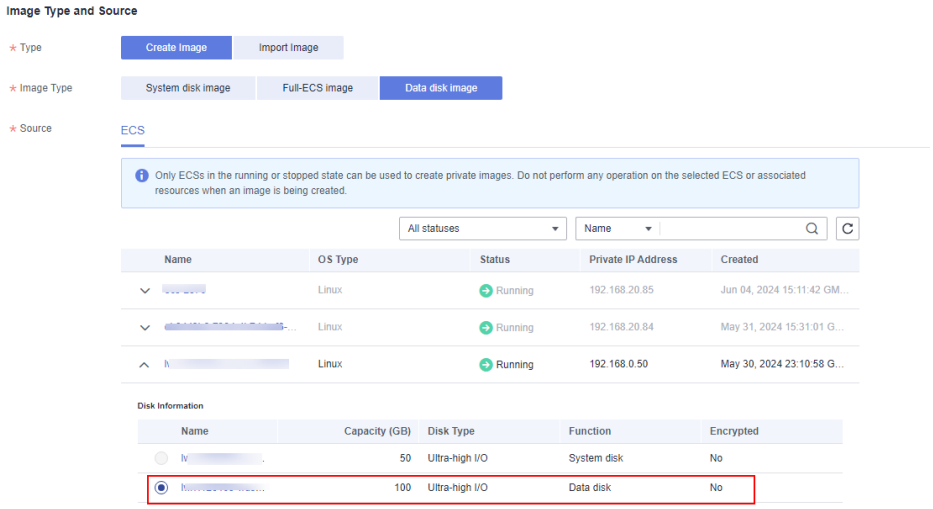
1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a data disk image.

1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Create Image** for **Type** and then select **Data disk image** for **Image Type**.
3. Select **ECS** for **Source** and then select a data disk of the ECS.

Figure 2-7 Creating a data disk image



4. In the **Image Information** area, set **Name**, **Tag**, and **Description**, and select an enterprise project.

   If the data disk is not encrypted, the private image created from it is also not encrypted. The encryption attribute cannot be changed during image creation. After the image is created, you can change its encryption attribute based on **Replicating Images Within a Region**.

5. Click **Apply Now**.
6. Confirm the settings and click **Submit Application**.

Step 3 Go back to the **Private Images** page and view the new data disk image.

----End

Follow-up Procedure

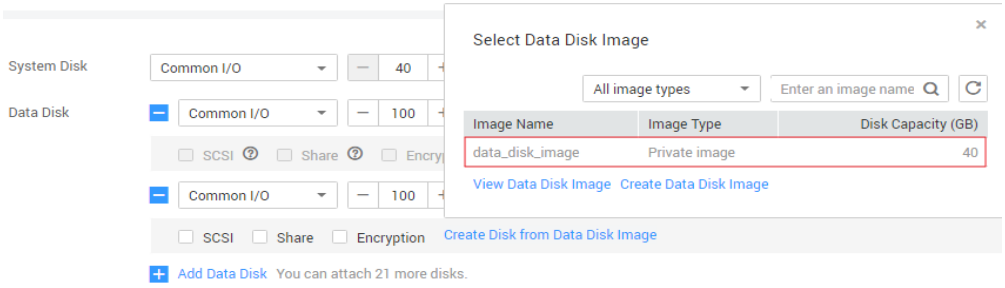
If you want to use the created data disk image to create an EVS disk and attach it to an ECS, you can perform either of the following operations:

- Locate the row that contains the created data disk image and click **Create Data Disk** to create one or multiple data disks. Then attach the data disks to an ECS.
- On the page for creating ECSs, click **Create Disk from Data Disk Image** and select the data disk image.

  NOTE

  In this way, a data disk image can be used to create a data disk for an ECS only once. For example, a data disk created from data disk image **data_disk_image** has been added to the ECS. No any other data disk created from this image can be added to the ECS.

Figure 2-8 Adding data disks



2.8 Creating a Data Disk Image from an External Image File

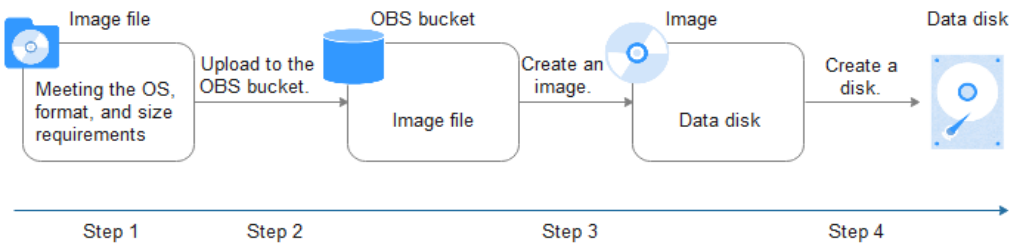
Scenarios

A data disk image contains only service data. You can create a data disk image using a local image file or an external image file (image file on another cloud platform). Then, you can use the data disk image to create EVS disks and migrate your service data to the cloud.

Background

The following figure shows the process of creating a data disk image from an external image file.

Figure 2-9 Creating a data disk image from an external image file



1. Prepare an external image file. The file must be in VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format. If you want to use an image file in other formats, convert the file into any of the listed formats before importing it to the cloud platform.
2. When uploading the external image file, you must select an OBS bucket with standard storage. For details, see **Uploading an External Image File**.
3. Create a data disk image. For details, see **Procedure**.
4. Use the data disk image to create data disks. For details, see **Follow-up Procedure**.

Procedure

Step 1 Access the IMS console.

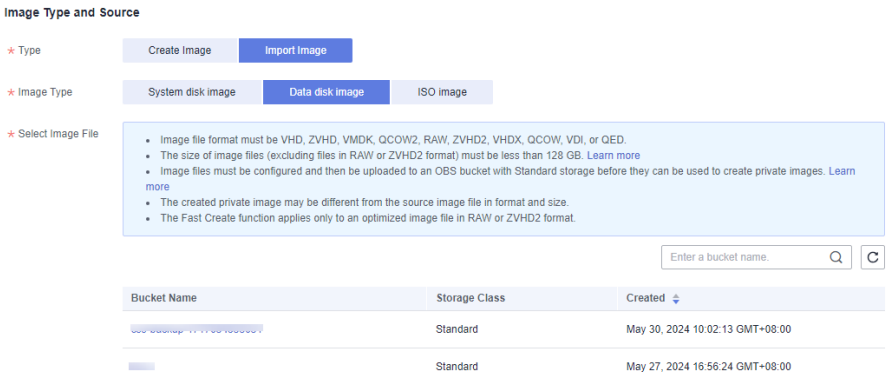
1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a data disk image.

1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Import Image** for **Type** and then select **Data disk image** for **Image Type**.
3. Select the bucket storing the image file from the list and then select the image file.

Figure 2-10 Creating a data disk image from an external image file



4. To register the image file using the Fast Create function, select **Enable Fast Create**.

   NOTE

   - Currently, this function supports only image files in ZVHD2 or RAW format.
   - For how to convert image file formats and generate bitmap files, see **Quickly Importing an Image File**.

   After you select **Enable Fast Create**, select the confirmation information following **Image File Preparation** if you have prepared the required files.

5. In the **Image Information** area, set the following parameters.
   - **OS Type**: The value is **Windows** or **Linux**.
   - **Data Disk**: The value ranges from 40 GB to 2048 GB and must be no less than the data disk capacity in the image file.
   - **Name**: Enter a name for the image.
   - (Optional) **Encryption**: If you want to encrypt the image, select **KMS encryption** and then select the key to be used from the key list.
   - **Enterprise Project**: Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or your account is an enterprise account. To enable this function, contact your customer manager. An enterprise project provides central management of cloud resources on a project.
   - (Optional) **Tag**: Set a tag key and a tag value for the image to easily identify and manage it.
   - (Optional) **Description**: Enter description of the image.
6. Click **Apply Now**.
7. Confirm the settings and click **Submit Application**.

Step 3 Go back to the **Private Images** page and view the new data disk image.

When the image status changes to **Normal**, the image creation is complete.

----End

Follow-up Procedure

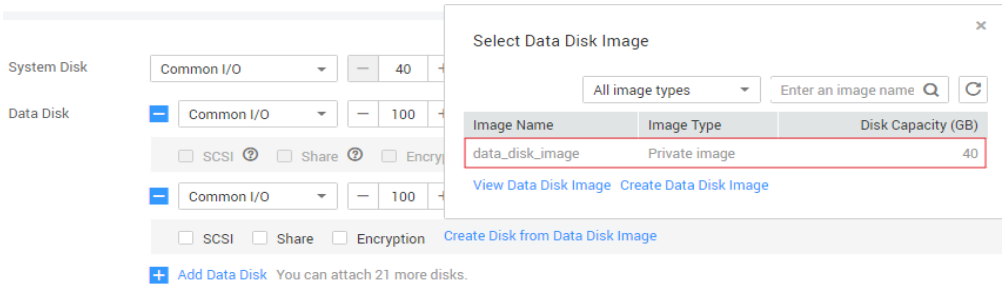
If you want to use the created data disk image to create an EVS disk and attach it to an ECS, you can perform either of the following operations:

- Locate the row that contains the created data disk image and click **Create Data Disk** to create one or multiple data disks. Then attach the data disks to an ECS.
- On the page for creating ECSs, click **Create Disk from Data Disk Image** and select the data disk image.

  NOTE

  In this way, a data disk image can be used to create a data disk for an ECS only once. For example, a data disk created from data disk image **data_disk_image** has been added to the ECS. No any other data disk created from this image can be added to the ECS.

Figure 2-11 Adding data disks



2.9 Creating a Full-ECS Image from an ECS

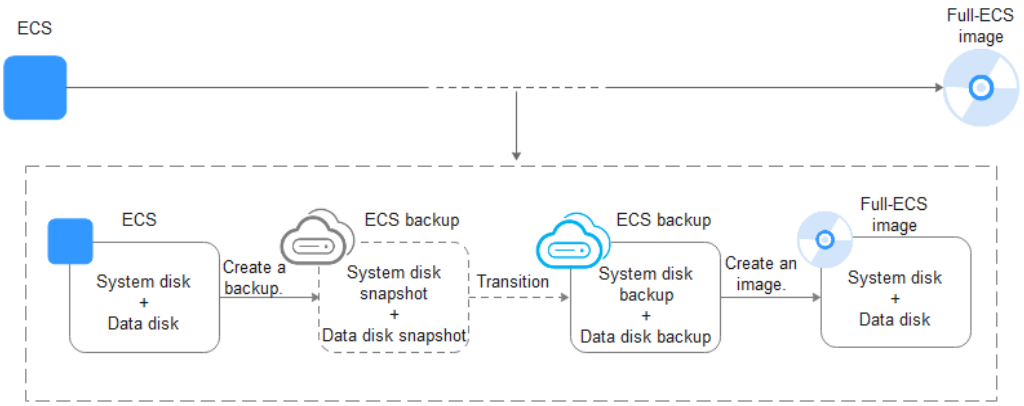
Scenarios

You can create an image of an entire ECS, including not just the OS, but also the software and all the service data. You can then use this image to migrate data by quickly provisioning exact clones of the original ECS.

Background

The following figure shows the process of creating an image from an entire ECS, with both the system and data disks included.

Figure 2-12 Creating a full-ECS image from an ECS



- The time required for creating a full-ECS image depends on the disk size, network quality, and the number of concurrent tasks.
- The ECS used to create a full-ECS image must be in **Running** or **Stopped** state. To create a full-ECS image containing a database, use a stopped ECS.
- If an ECS is in **Stopped** state, do not start it when you are using it to create a full-ECS image.
- When a full-ECS image is being created from an ECS, do not perform any operations on the ECS, or the image creation may fail.
- In **Figure 2-12**, if there are snapshots of the system disk and data disks but the ECS backup creation is not complete, the full-ECS image you create will only be available in the AZ where the source ECS is and can only be used to provision ECSs in this AZ. You cannot provision ECSs in other AZs in the region until the original ECS is fully backed up and the full-ECS image is in the **Normal** state.
- If you use a full-ECS image to change an ECS OS, only the system disk data can be written into the ECS. Therefore, if you want to restore or migrate the data disk data of an ECS by using a full-ECS image, you can only use the image to create a new ECS rather than use it to change the ECS OS.

Constraints

- When creating a full-ECS image from an ECS, ensure that the ECS has been properly configured, or the image creation may fail.
- A Windows ECS used to create a full-ECS image cannot have a spanned volume, or data may be lost when ECSs are created from that image.
- A Linux ECS used to create a full-ECS image cannot have a disk group or logical disk that contains multiple physical disks, or data may be lost when ECSs are created from that image.
- A full-ECS image cannot be replicated within a region or be exported.
- When creating a full-ECS image from a Windows ECS, you need to change the SAN policy of the ECS to OnlineAll. Otherwise, EVS disks attached to the ECSs created from the image may be offline.

  Windows has three types of SAN policies: **OnlineAll**, **OfflineShared**, and **OfflineInternal**.

  Table 2-13 SAN policies in Windows

Type	Description
OnlineAll	All newly detected disks are automatically brought online.
OfflineShared	All disks on sharable buses, such as iSCSI and FC, are left offline by default, while disks on non-sharable buses are kept online.
OfflineInternal	All newly detected disks are left offline.

  a. Execute **cmd.exe** and run the following command to query the current SAN policy of the ECS:

diskpart

b. Run the following command to view the SAN policy of the ECS:

san

- If the SAN policy is **OnlineAll**, run the **exit** command to exit DiskPart.
- If the SAN policy is not **OnlineAll**, go to **c**.

c. Run the following command to change the SAN policy of the ECS to **OnlineAll**:

san policy=onlineall

Procedure

Step 1 Access the IMS console.

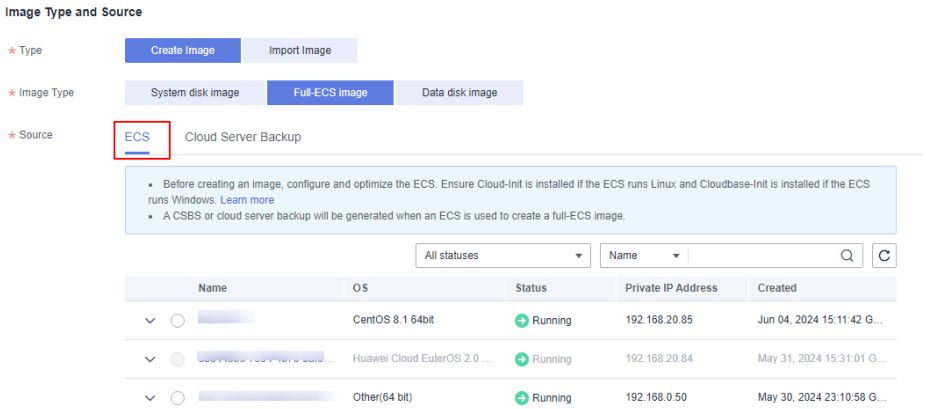
1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a full-ECS image.

1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Create Image** for **Type** and then select **Full-ECS image** for **Image Type**.
3. Select **ECS** for **Source** and then select an ECS from the list.

   Figure 2-13 Creating a full-ECS image using an ECS



4. Specify **Server Backup Vault** to store backups.

   The created full-ECS image and backup are stored in the server backup vault.

   If no server backup vault is available, click **Create Server Backup Vault** to create one. Ensure that you select **Backup** for **Protection Type**. For more information about CBR backups and vaults, see *Cloud Backup and Recovery User Guide*.
5. In the **Image Information** area, configure basic image details, such as the image name and description.
6. Click **Apply Now**.

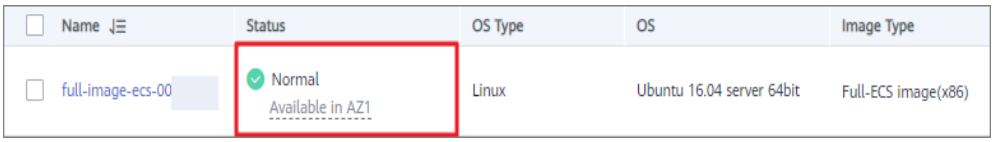
7. Confirm the settings and click **Submit Application**.

Step 3 Go back to the **Private Images** page and view the new full-ECS image.

- When the image status changes to **Normal**, the image creation is complete.
- If **Available in AZ***X* is displayed under **Normal** in the **Status** column for a full-ECS image, the backup for this ECS has not been created and only a disk snapshot is created. (**AZ***X* indicates the AZ where the source ECS of the image resides.)

  In this case, the full-ECS image can be used to provision ECSs only in the specified AZ. If you want to use this image to provision ECSs in other AZs of the region, you need to wait until **Available in AZ***X* disappears from under **Normal**, which indicates that the ECS backup has been successfully created. This process takes about 10 minutes, depending on the data volume of the source ECS.

  Figure 2-14 Full-ECS image status

----End

Follow-up Procedure

- If you want to use the full-ECS image to create ECSs, click **Apply for ECS** in the **Operation** column. On the displayed page, create ECSs by following the instructions in *Elastic Cloud Server User Guide*.

  NOTE

  When you use a full-ECS image to create an ECS:

  - The system and data disk information defaulted by the image will be automatically displayed.
  - If the full-ECS image contains multiple data disks, it takes some time to load and display the disk information.

- If you use a full-ECS image to change an ECS OS, only the system disk data can be written into the ECS. Therefore, if you want to restore or migrate the data disk data of an ECS by using a full-ECS image, you can only use the image to create a new ECS rather than use it to change the ECS OS.
- If you want to share the full-ECS image with other tenants, you can use either of the following methods:
  - If the ECS the full-ECS image was created from has a CSBS backup, you must first migrate the backup to CBR before you share the image because CSBS is being deprecated.
  - If the ECS has no such a backup, you can share the full-ECS image directly.

2.10 Creating a Full-ECS Image from a CBR Backup

Scenarios

You can use a Cloud Backup and Recovery (CBR) backup to create a full-ECS image, which can be used to create ECSs.

Background

- The CBR service provides backup services for EVS disks, BMSs, and ECSs, and supports restoring data of servers and disks using backups. If you have created a backup for an ECS using CBR, you can use the backup to create a full-ECS image.
- If you use a full-ECS image to change an ECS OS, only the system disk data can be written into the ECS. Therefore, if you want to restore or migrate the data disk data of an ECS by using a full-ECS image, you can only use the image to create a new ECS rather than use it to change the ECS OS.

Constraints

- When creating a full-ECS image from a CBR backup, ensure that the source ECS of the CBR backup has been properly configured, or the image creation may fail.
- A CBR backup can be used to create only one full-ECS image.
- If an ECS is in **Stopped** state, do not start it when you are using it to create a full-ECS image.
- A full-ECS image created from a CBR backup can be shared with other tenants. However, if it is a shared CBR backup, the full-ECS image created from it cannot be shared.
- A full-ECS image cannot be replicated within a region or be exported.

Procedure

Step 1 Access the IMS console.

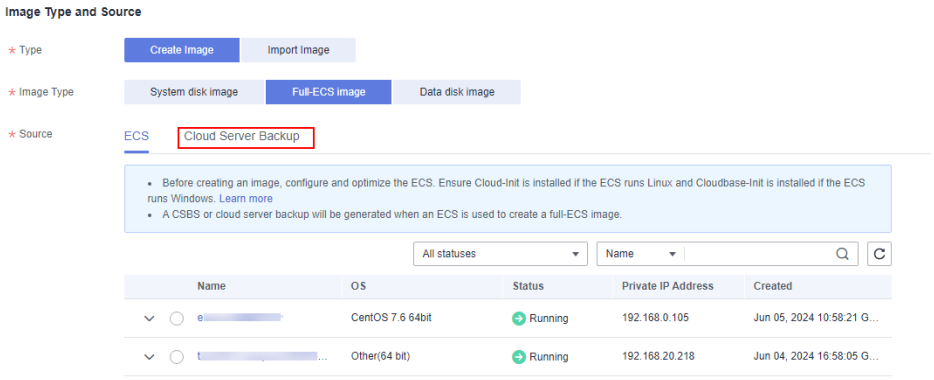
1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Create a full-ECS image.

1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Create Image** for **Type** and then select **Full-ECS image** for **Image Type**.
3. Select **Cloud Server Backup** for **Source** and then select an ECS from the list.

Figure 2-15 Creating a full-ECS image using a CBR backup



4. In the **Image Information** area, configure basic image details, such as the image name and description.
5. Click **Apply Now**.
6. Confirm the settings and click **Submit Application**.

Step 3 Switch back to the **Image Management Service** page to monitor the image status.

When the image status changes to **Normal**, the image creation is complete.

----End

Follow-up Procedure

After the full-ECS image creation is complete, you can perform the following operations:

- If you want to use the image to create ECSs, click **Apply for ECS** in the **Operation** column. On the displayed page, select **Private image** and then select the created full-ECS image. For details, see *Elastic Cloud Server User Guide*.

  NOTE

  When you use a full-ECS image to create an ECS:
  - The system and data disk information defaulted by the image will be automatically displayed.
  - If the full-ECS image contains multiple data disks, it takes some time to load and display the disk information.
- If you want to share the image with other tenants, click **More** in the **Operation** column and select **Share** from the drop-down list. In the displayed dialog box, enter the account names of the image recipients. For details, see **Sharing Specified Images**.
- If you use a full-ECS image to change an ECS OS, only the system disk data can be written into the ECS. Therefore, if you want to restore or migrate the data disk data of an ECS by using a full-ECS image, you can only use the image to create a new ECS rather than use it to change the ECS OS.

2.11 Creating a Windows System Disk Image from an ISO File

2.11.1 Overview

An ISO file is a disk image of an optical disc. A large number of data files can be compressed into a single ISO file. Likewise, to access the files stored in an ISO, the ISO file needs to be decompressed. For example, you can use a virtual CD-ROM to open an ISO file, or burn the ISO file to a CD or DVD and then use the CD-ROM to read the image.

This section describes how to create a Windows system disk image from an ISO file.

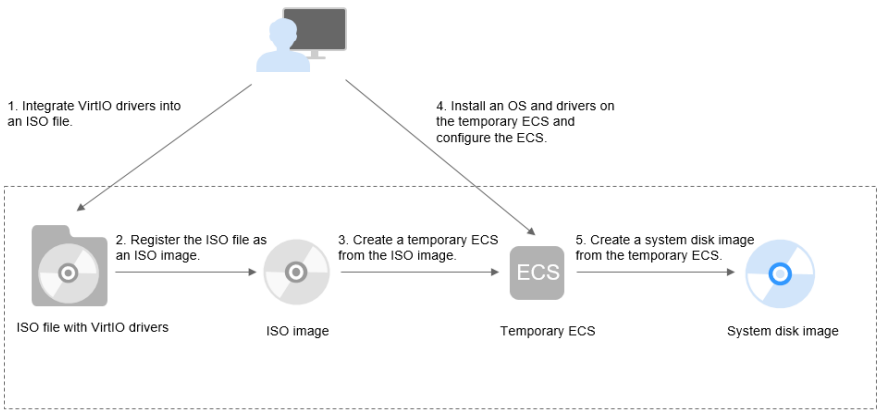
NOTE

This section is applicable only to the management console. If you are an API user, see "Creating an Image from an ISO File" in *Image Management Service User Guide*.

Creation Process

Figure 2-16 shows the process of creating a Windows system disk image from an ISO file.

Figure 2-16 Creating a Windows system disk image



The procedure is as follows:

1. Integrate VirtIO drivers into the ISO file.

   Windows uses Integrated Drive Electronics (IDE) disks and VirtIO NICs. Before registering an image on the cloud platform, integrate VirtIO drivers into the Windows ISO file. For details, see **Integrating VirtIO Drivers into an ISO File**.
2. Register the ISO file as an ISO image.

On the management console, register the ISO file with VirtIO drivers as an image. The image is an ISO image and cannot be used to provision ECSs. For details, see **Registering an ISO File as an ISO Image**.

3. Create a temporary ECS from the ISO image.

   Use the registered ISO image to create a temporary ECS. The ECS has no OS or driver installed. For details, see **Creating a Windows ECS from an ISO Image**.

4. Install an OS and necessary drivers for the temporary ECS and configure related settings.

   You need to install an OS and VirtIO drivers, and configure NICs. For details, see **Installing a Windows OS and VirtIO Drivers** and **Step 1** in **Configuring the ECS and Creating a Windows System Disk Image**.

5. Create a system disk image from the temporary ECS.

   On the management console, create a system disk image from the temporary ECS on which the installation and configuration have been completed. After the image is created, delete the temporary ECS to prevent it from occupying compute resources. For details, see **Creating a System Disk Image from a Windows ECS**.

Constraints

- An ISO image created from an ISO file is used only for creating a temporary ECS. It will not be available on the ECS console. You cannot use it to create ECSs or change ECS OSs. You need to install an OS on the temporary ECS and use that ECS to create a system disk image which can be used to create ECSs or change ECS OSs.
- A temporary ECS has limited functionality. For example, you cannot attach disks to it. You are not advised to use it as a normal ECS.

2.11.2 Integrating VirtIO Drivers into an ISO File

Scenarios

Windows uses IDE disks and VirtIO NICs. Before registering an image on the cloud platform, integrate VirtIO drivers into the Windows ISO file. Typically, an ISO file contains all the files that would be included on an optical disc. Some software can be installed only from a CD-ROM drive. So, a virtual CD-ROM drive is required.

This section uses AnyBurn and UltraISO as examples to describe how to integrate VirtIO drivers into an ISO file.

NOTE

- AnyBurn is lightweight CD/DVD/Blu-ray burning software with a free version.
- UltraISO is an ISO CD/DVD image file handling tool. A free trial version is limited to ISO files of 300 MB or less. You are advised to buy a standard version.

Prerequisites

You have obtained an ISO file.

NOTE

The ISO file name can contain only letters, digits, hyphens (-), and underscores (_). If the name does not meet the requirements, change it.

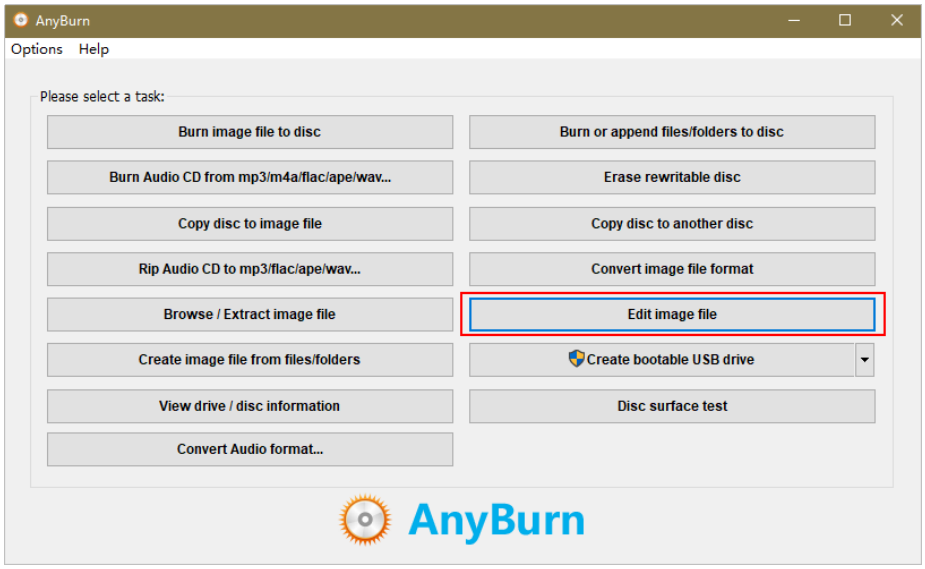
AnyBurn

1. Download **AnyBurn** and install it on your local PC.
2. Download VirtIO drivers.

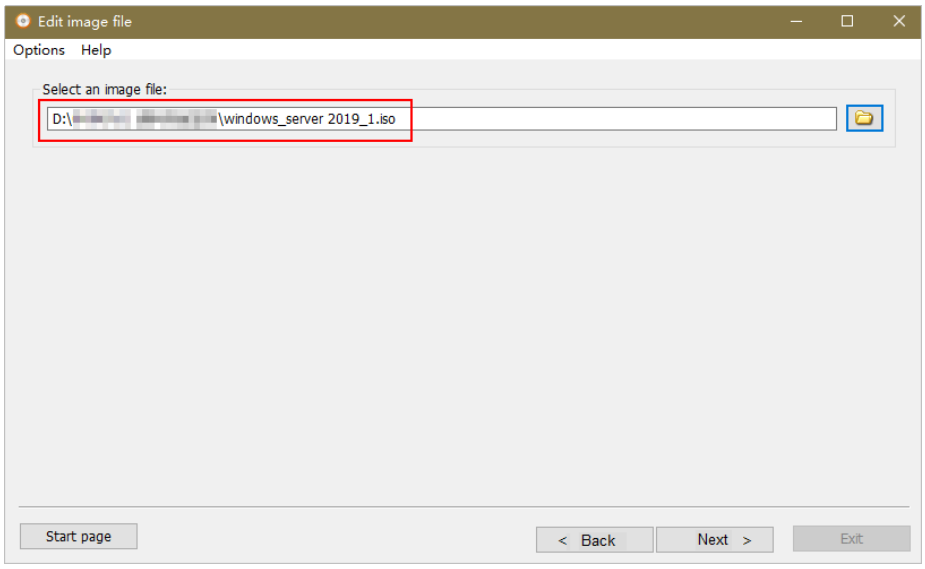
   https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso

   Other versions:

   https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/
3. Use AnyBurn to open the ISO file.
   a. Open AnyBurn and select **Edit Image File**.



   b. Select the ISO file and click **Next**.

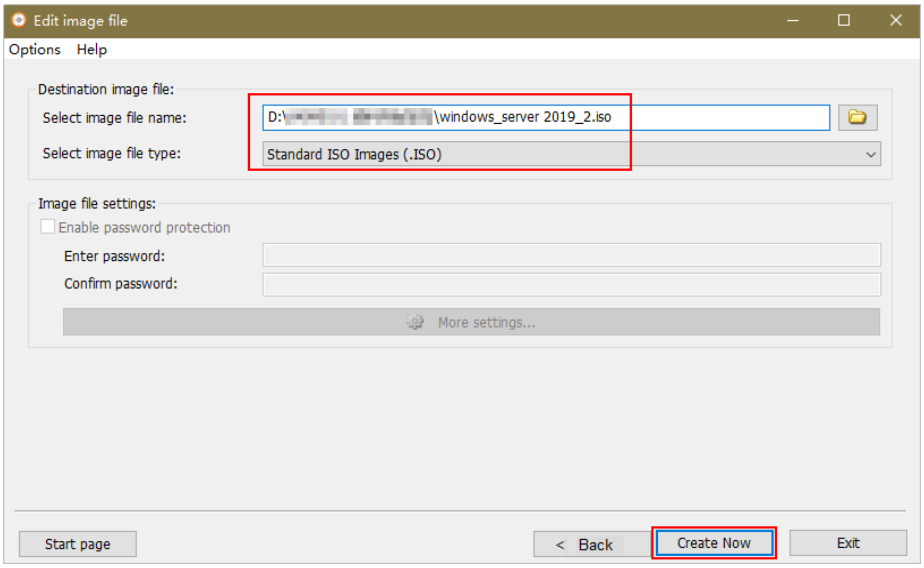


4. Edit the ISO file to integrate VirtIO drivers into it.

   a. Click **Add** and select all the files in **virtio-win.iso** that was downloaded in **2** to add them to the parent node of the ISO file.

   b. Select a path to save the new ISO file and specify a name for the new file. Select **ISO** as the file type. Click **Create Now**.

      After the new ISO file is generated, view VirtIO drivers in it.



UltraISO

1. Download UltraISO and install it on your local PC.

   Download address: **https://www.ultraiso.com/**

2. Download VirtIO drivers.

   https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso

   Other versions:

   https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/

3. Use UltraISO to open the ISO file.

   CAUTION

   Do not extract the ISO file or open it with any tool other than UltraISO, or the boot data will be lost.

4. Drag and drop the downloaded VirtIO driver files to the parent node of the ISO file.

5. Use UltraISO to export the ISO file with VirtIO drivers to an .iso file on your local PC.

2.11.3 Registering an ISO File as an ISO Image

Scenarios

Register an external ISO file on the cloud platform as a private image (ISO image). Before registering an image, upload the ISO file exported in **Integrating VirtIO Drivers into an ISO File** to the OBS bucket.

The ISO image cannot be replicated, exported, or encrypted.

Prerequisites

- The file to be registered must be in ISO format.
- The ISO image file has been uploaded to the OBS bucket. For details, see **Uploading an External Image File**.

Procedure

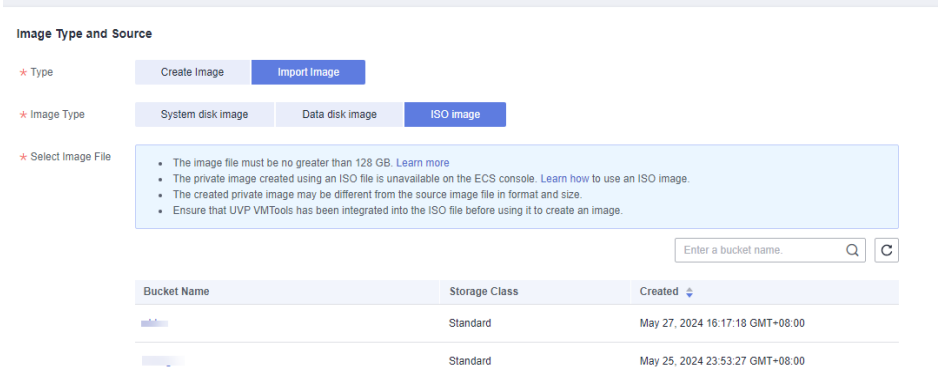
Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
   The IMS console is displayed.

Step 2 Register an ISO file as an ISO image.

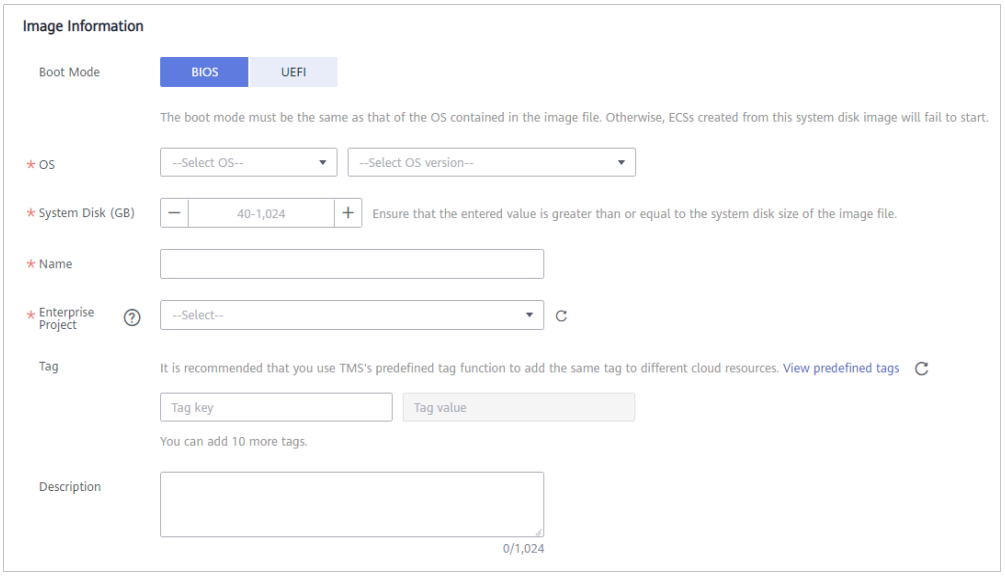
1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Import Image** for **Type** and then select **ISO image** for **Image Type**.
3. In the image file list, select the bucket and then the image file.

   Figure 2-17 Creating a private image from an ISO file



4. In the **Image Information** area, set the following parameters.

Figure 2-18 Configuring image information



- **Boot Mode**: Select **BIOS** or **UEFI**. Ensure that the selected boot mode is the same as that in the image file, or the ECSs created from this image will not be able to boot up.
- **OS**: Select the OS specified in the ISO file. To ensure that the image can be created and used properly, select an OS consistent with that in the image file.
- **System Disk**: Set the system disk capacity (value range: 40 GB to 1024 GB), which must be no less than the capacity of the system disk in the image file.
- **Name**: Enter a name for the image to be created.
- **Enterprise Project**: Select the enterprise project to which your images belong.
- **Tag**: (Optional) Add a tag to the image to be created.
- **Description**: (Optional) Enter image description as needed.

5. Click **Apply Now**.
6. Confirm the settings and click **Submit Application**.

Step 3 Switch back to the **Image Management Service** page to check the image status.

When the image status changes to **Normal**, the image is registered successfully.

----End

2.11.4 Creating a Windows ECS from an ISO Image

Scenarios

This section describes how to create an ECS from a registered ISO image.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Use an ISO image to create a Windows ECS.

1. Click the **Private Images** tab. Locate the row that contains the ISO image and click **Create ECS** in the **Operation** column.
2. Configure the ECS as prompted and click **OK**.

----End

Follow-up Procedure

After the ECS is created, you can log in remotely to continue with OS and drivers installation.

2.11.5 Installing a Windows OS and VirtIO Drivers

Scenarios

This section uses Windows Server 2019 64-bit as an example to describe how to install Windows on an ECS.

The installation procedure varies depending on the image file you use. Perform operations as prompted.

NOTE

Set the time zone, KMS address, patch server, input method, and language based on service requirements.

Prerequisites

You have remotely logged in to the ECS and entered the installation page.

Procedure

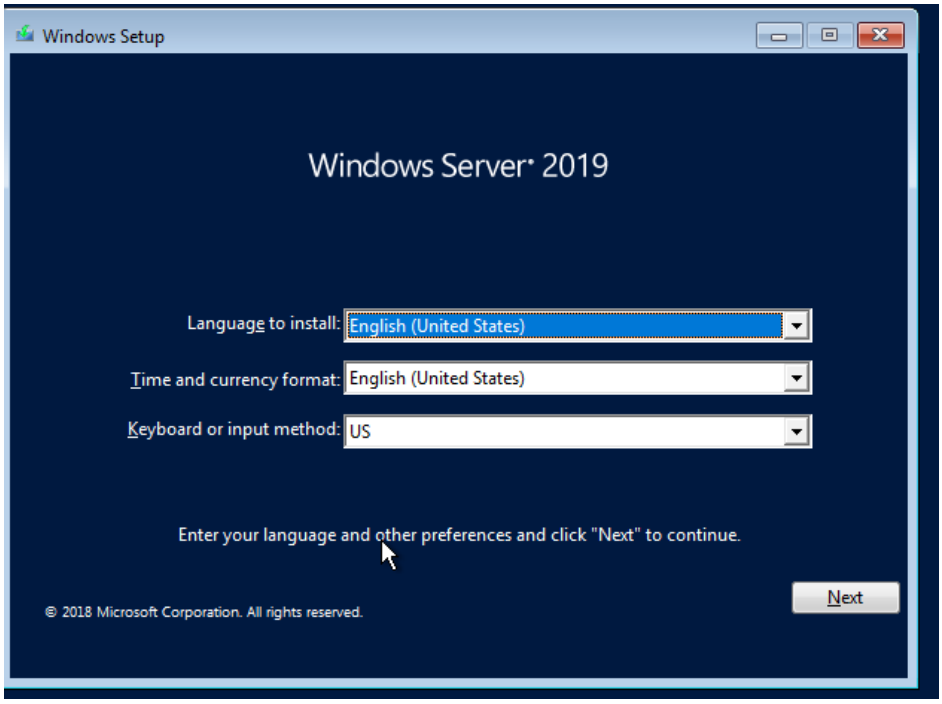
CAUTION

Do not stop or restart the ECS during the OS installation. Otherwise, the OS installation will fail.

Step 1 Install the Windows OS.

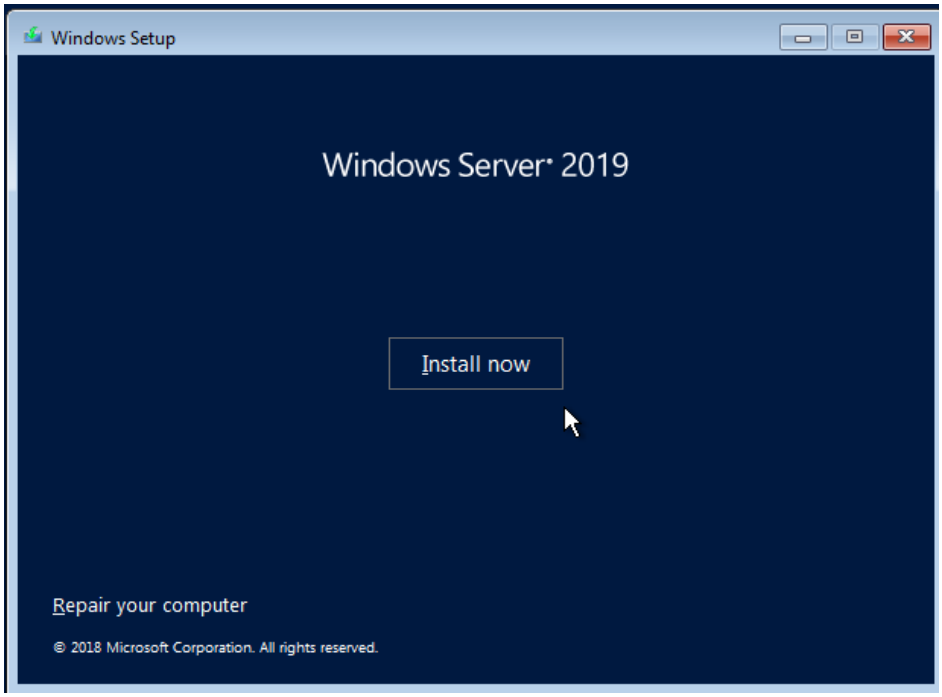
1. Configure Windows setup.

Figure 2-19 Windows setup



2. Click **Next**.
   The installation confirmation window is displayed.

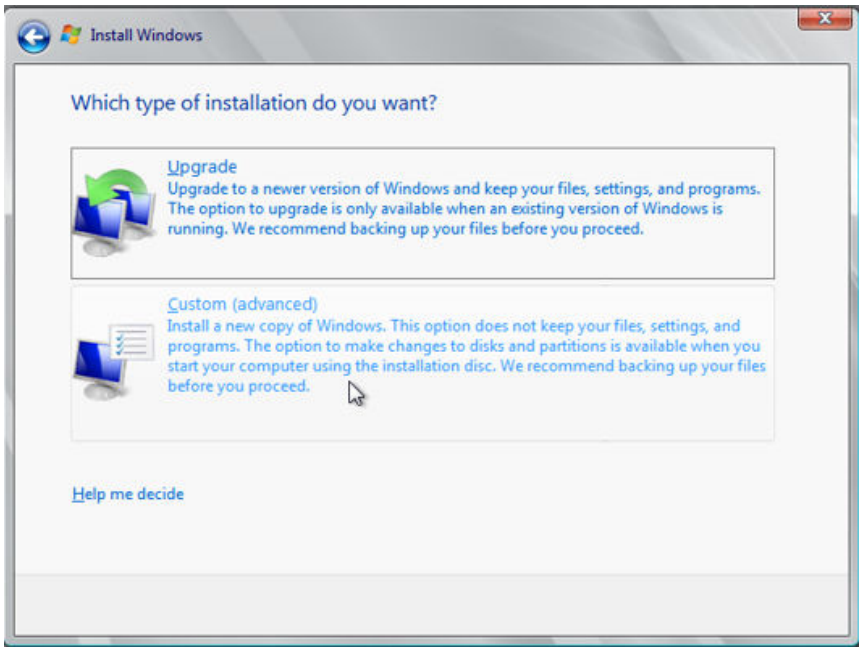
Figure 2-20 Installation confirmation window



3. Click **Install now**.
   The **Select the operating system you want to install** window is displayed.
4. Select the version of the OS to be installed and click **Next**.
   The **Please read the license terms** window is displayed.

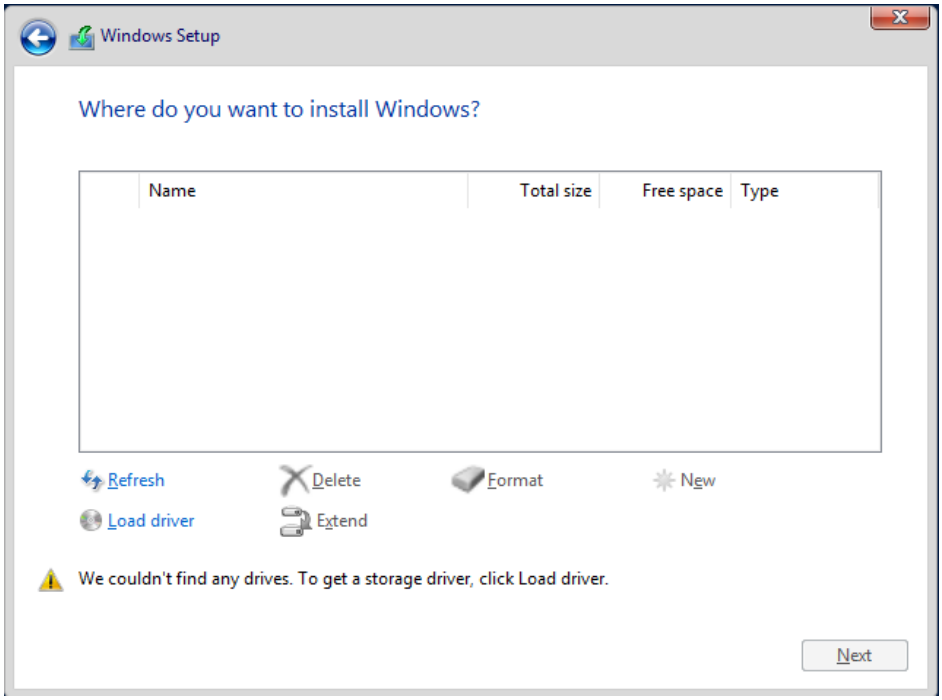
5. Select **I accept the license terms**, and click **Next**.
   The **Which type of installation do you want?** window is displayed.

   Figure 2-21 Installation type



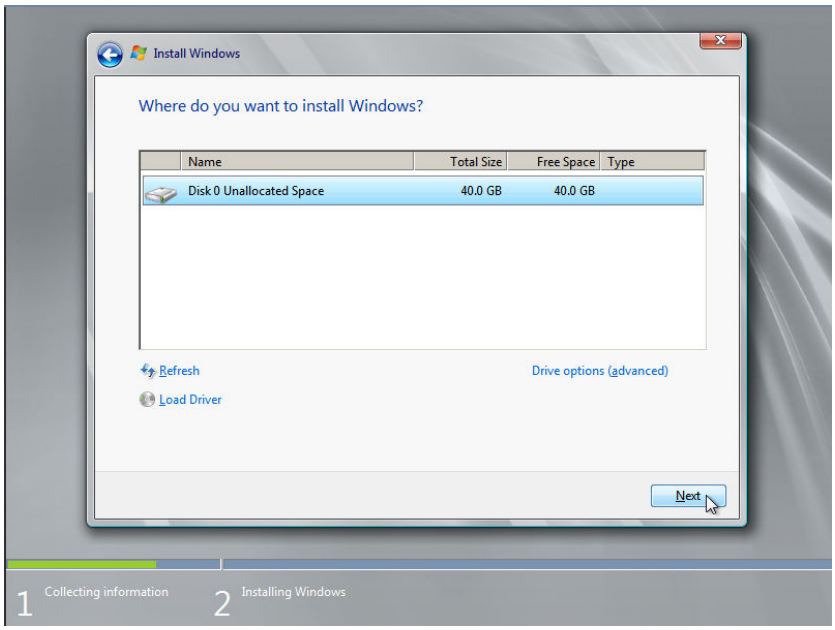
6. Select **Custom (advanced)**.
   The **Where do you want to install Windows?** window is displayed.
   - If the system displays a message indicating that no driver is found, go to **Step 1.7**.

   Figure 2-22 Installation path



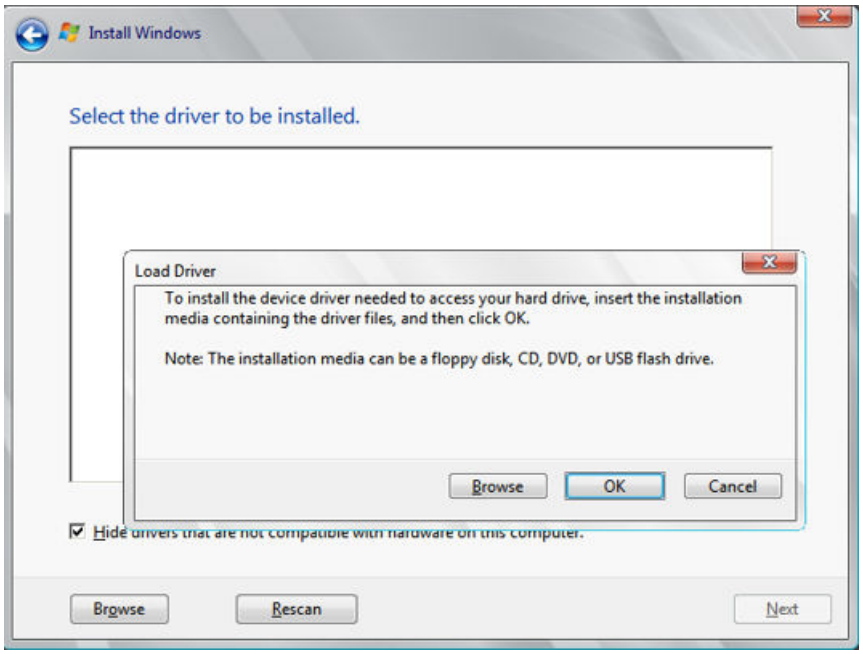
- If a disk is displayed, go to **Step 1.10**.

Figure 2-23 Installation path



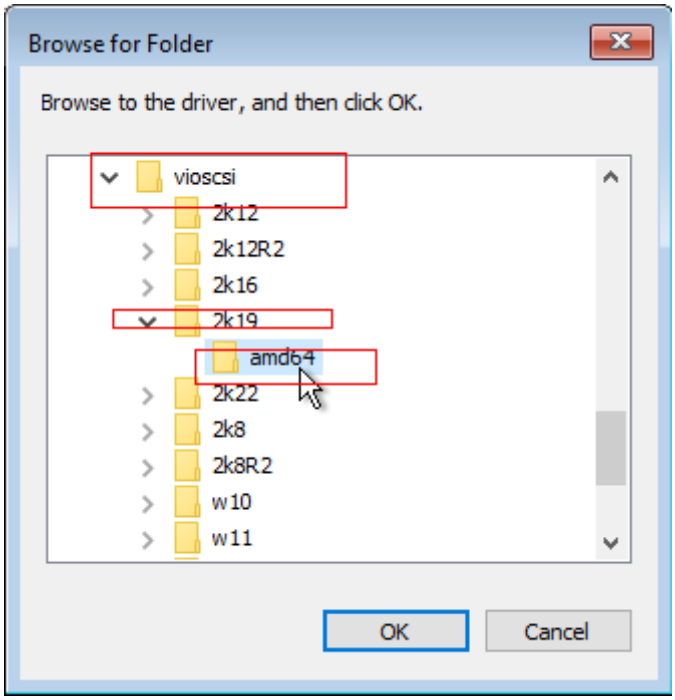
7. Click **Load Driver** and then **Browse**.

Figure 2-24 Loading drivers



8. Choose **vioscsi** > **2k19** > **amd64** and click **OK**.

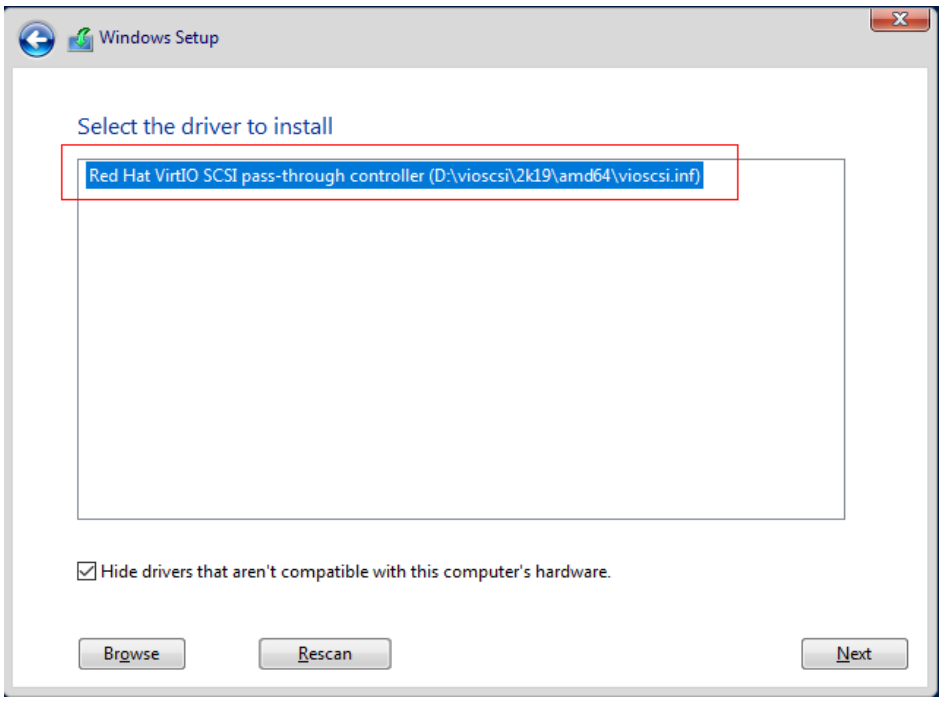
Figure 2-25 Browsing for a folder



9. Select the driver matching the OS and click **Next**.

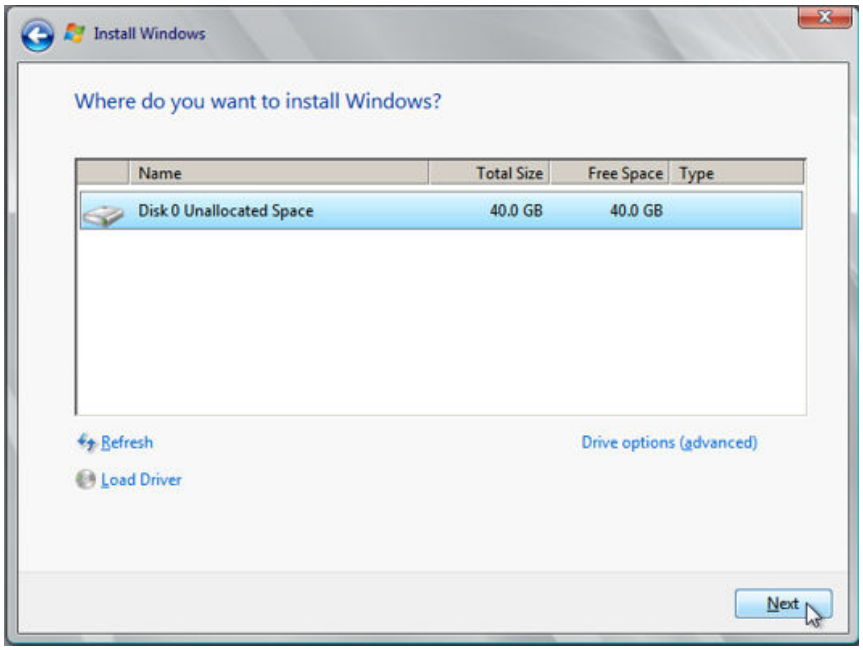
   The system may provide multiple drivers. Select **vioscsi.inf** shown in the following figure.

Figure 2-26 Selecting the driver to install



10. Select the disk and click **Next**.

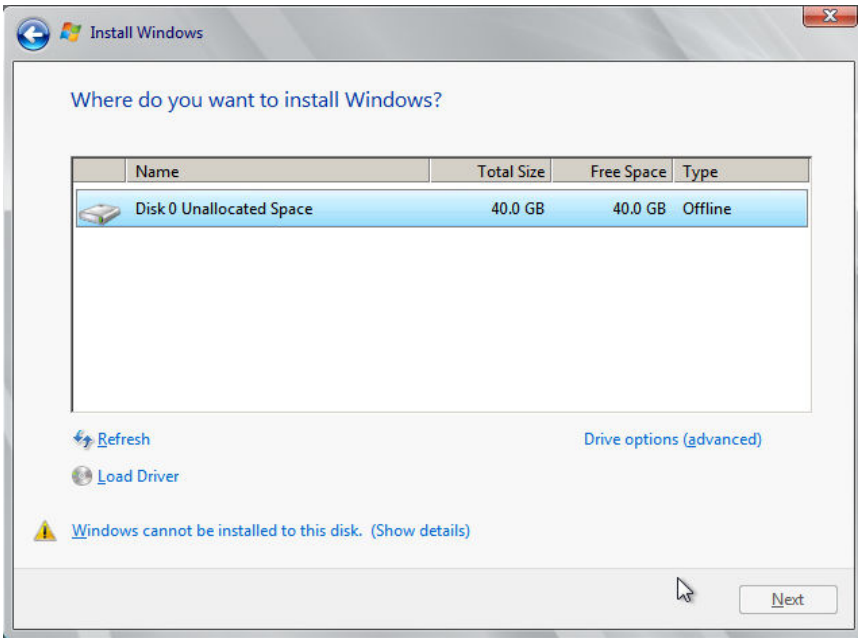
Figure 2-27 Installation path



NOTE

If the disk type is **Offline**, you can stop and then start the ECS, and restart the OS installation process.

Figure 2-28 Offline disk



11. The **Installing Windows** window is displayed, and the OS installation starts.

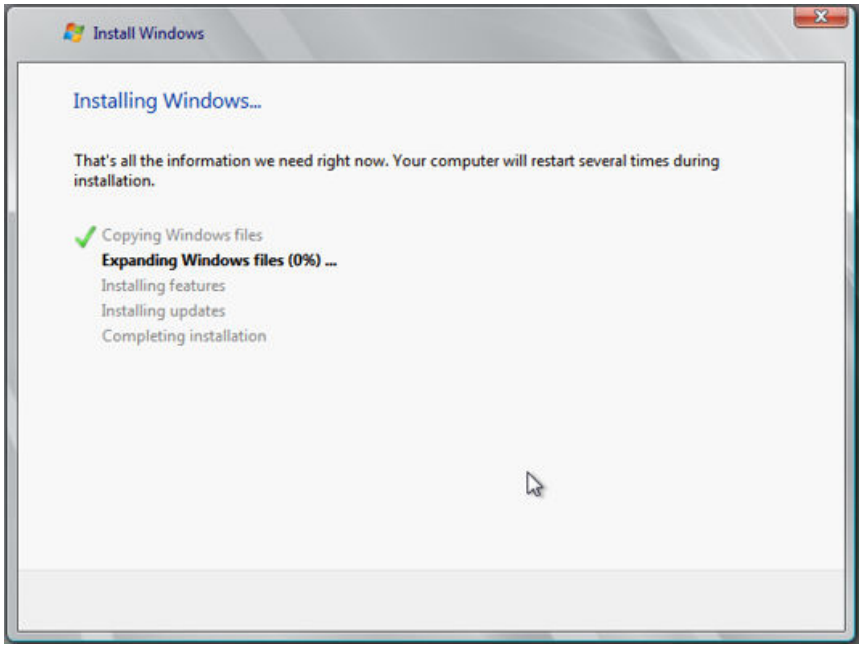
    The installation takes about 50 minutes. The ECS restarts during the installation. After the ECS successfully restarts, log in to it again and configure the OS as prompted.

NOTE

You are required to set a password for the OS user.

Supported special characters include !@$%^-_=+[{}]:,./?

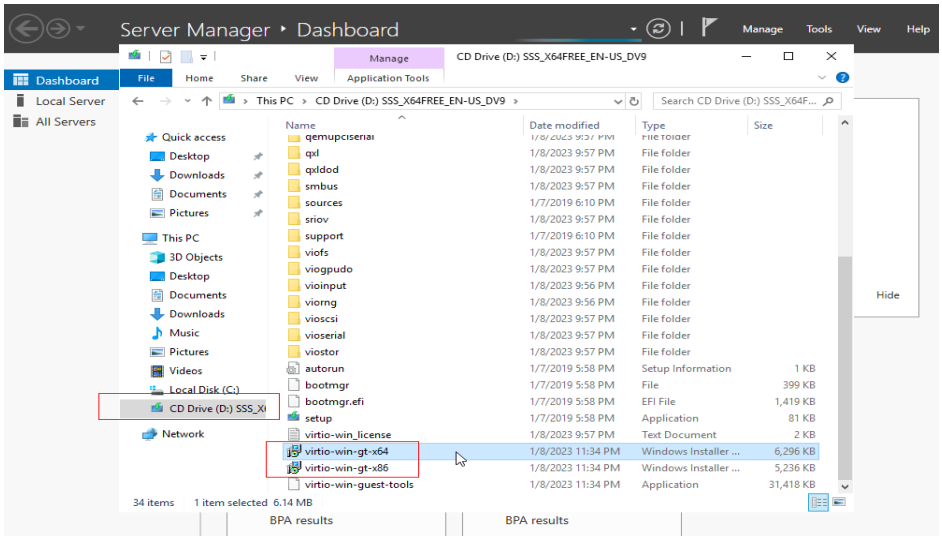
Figure 2-29 Installation progress



Step 2 Install related drivers.

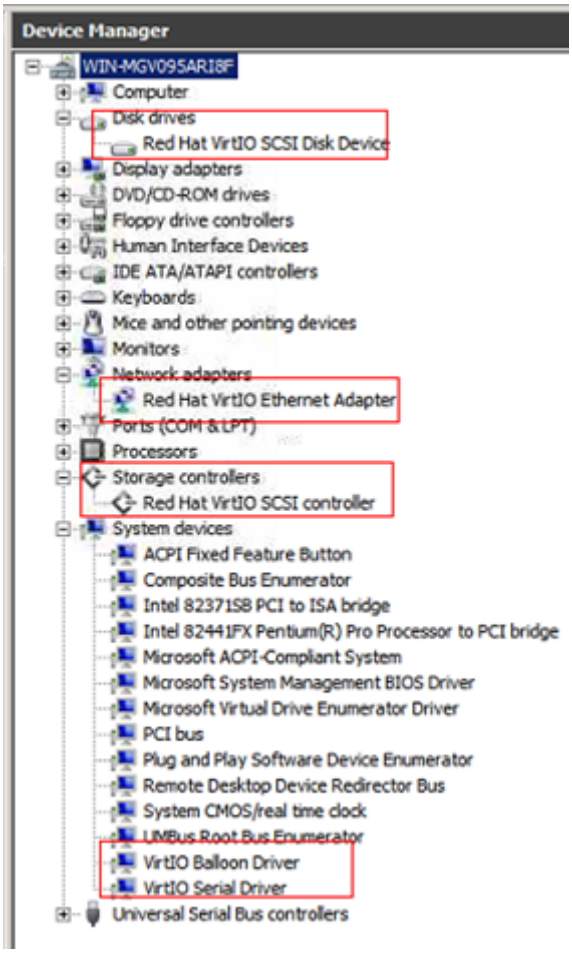
1. Open **Computer** and double-click the CD drive.

Figure 2-30 Starting the CD drive



2. Double-click **virtio-win-gt-x64** or **virtio-win-gt-x86**. Install drivers as prompted.
3. After the installation is complete, start **Device Manager** and check that all the drivers shown in the red box are successfully installed.

Figure 2-31 Device Manager



----End

2.11.6 Configuring the ECS and Creating a Windows System Disk Image

Scenarios

After installing an OS for the temporary ECS, configure the ECS and install Guest OS drivers provided by the cloud platform so that ECSs that will be created with this temporary ECS as a source can work properly.

This section describes how to configure a Windows ECS, install the Guest OS drivers, and create a Windows system disk image.

Procedure

Step 1 Configure the ECS.

1. Check whether NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in **Setting the NIC to DHCP**.
2. Enable remote desktop connection for the ECS as needed. For details about how to enable this function, see **Enabling Remote Desktop Connection**.

3. Install and configure Cloudbase-Init. User data injection on the management console is available for the new ECSs created from the image only after this tool is installed. For example, you can use data injection to set the login password for a new ECS. For details, see **Installing and Configuring Cloudbase-Init**.
4. (Optional) Configure value-added functions.
   - Enable NIC multi-queue. For details, see **How Do I Enable NIC Multi-Queue for an Image?**
   - Configure dynamic assignment of IPv6 addresses. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Step 2 Stop the ECS to make the configurations take effect.

Step 3 Use the ECS to create a Windows system disk image.

For details, see **Creating a System Disk Image from a Windows ECS**.

----End

Follow-up Procedure

After the system disk image is created, delete the temporary ECS in a timely manner to prevent it from occupying compute resources.

2.12 Creating a Linux System Disk Image from an ISO File

2.12.1 Overview

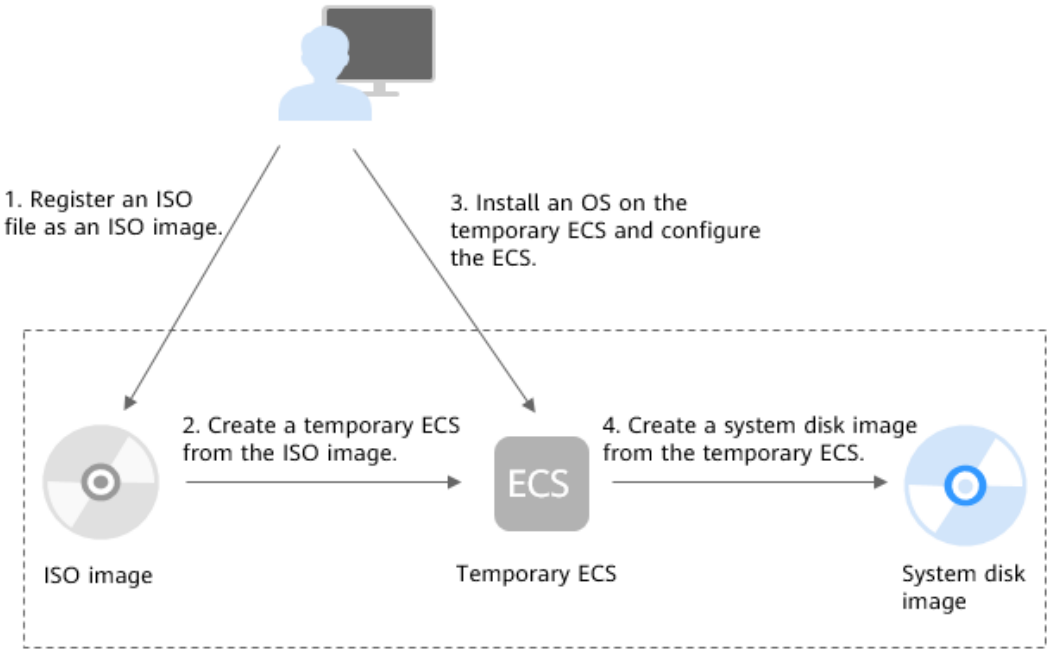
An ISO file is a disk image of an optical disc. A large number of data files can be compressed into a single ISO file. Likewise, to access the files stored in an ISO, the ISO file needs to be decompressed. For example, you can use a virtual CD-ROM to open an ISO file, or burn the ISO file to a CD or DVD and then use the CD-ROM to read the image.

This section describes how to create a Linux system disk image using an ISO file.

Creation Process

Figure 2-32 shows the process of creating a Linux system disk image from an ISO file.

Figure 2-32 Creating a Linux system disk image



The procedure is as follows:

1. Register an ISO file as an ISO image.

   On the management console, register the prepared ISO file as an image. The image is an ISO image and cannot be used to provision ECSs. For details, see **Registering an ISO File as an ISO Image**.

2. Create a temporary ECS from the ISO image.

   Use the registered ISO image to create a temporary ECS. The ECS has no OS or driver installed. For details, see **Creating a Linux ECS from an ISO Image**.

3. Install an OS and necessary drivers for the temporary ECS and configure related settings.

   The operations include installing an OS, installing native Xen and KVM drivers, configuring NIC attributes, and deleting files from the network rule directory. For details, see **Installing a Linux OS** and **Step 1** in **Configuring the ECS and Creating a Linux System Disk Image**.

4. Create a system disk image from the temporary ECS.

   On the management console, create a system disk image from the temporary ECS on which the installation and configuration have been completed. After the image is created, delete the temporary ECS to prevent it from occupying compute resources. For details, see **Creating a System Disk Image from a Linux ECS**.

Constraints

- An ISO image created from an ISO file is used only for creating a temporary ECS. It will not be available on the ECS console. You cannot use it to create ECSs or change ECS OSs. You need to install an OS on the temporary ECS and use that ECS to create a system disk image which can be used to create ECSs or change ECS OSs.

- A temporary ECS has limited functionality. For example, you cannot attach disks to it. You are not advised to use it as a normal ECS.

2.12.2 Registering an ISO File as an ISO Image

Scenarios

Register an external ISO file on the cloud platform as a private image (ISO image). Before registering an image, upload the ISO file to the OBS bucket.

The ISO image cannot be replicated, exported, or encrypted.

Prerequisites

- The file to be registered must be in ISO format.
- The ISO image file has been uploaded to the OBS bucket. For details, see **Uploading an External Image File**.

  NOTE

  The ISO image file name can contain only letters, digits, hyphens (-), and underscores (_). If the image file name does not meet the requirements, change the name before uploading the image file to the OBS bucket.

Procedure

Step 1 Access the IMS console.

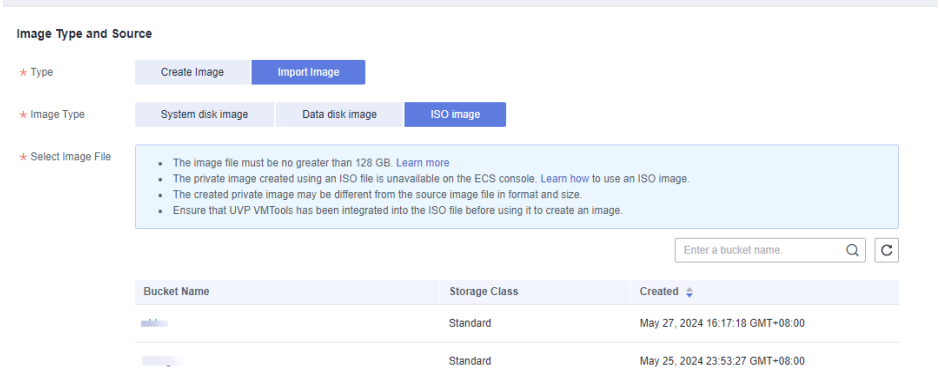
1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Register an ISO file as an ISO image.

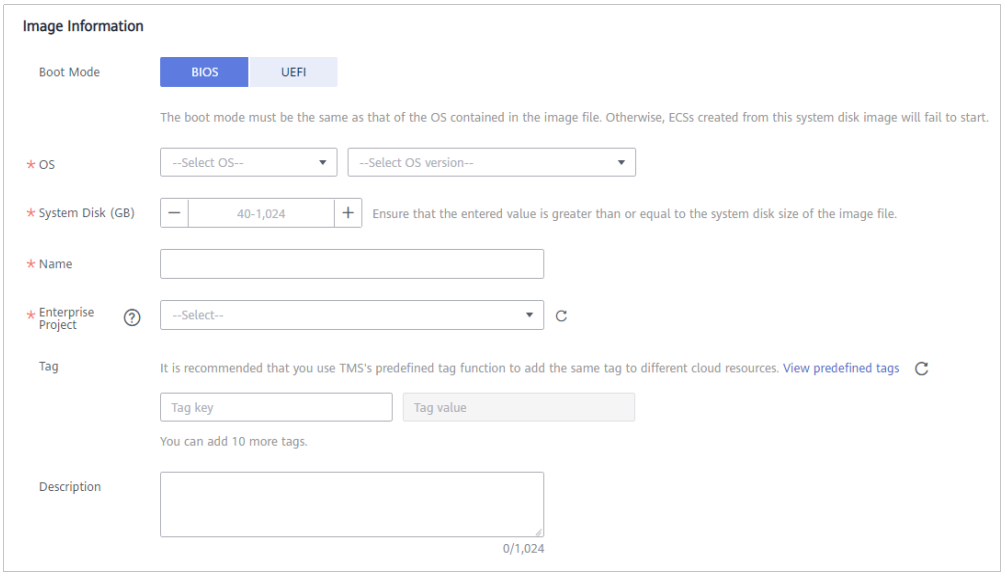
1. Click **Create Image** in the upper right corner.
2. In the **Image Type and Source** area, select **Import Image** for **Type** and then select **ISO image** for **Image Type**.
3. In the image file list, select the bucket and then the image file.

   Figure 2-33 Creating a private image from an ISO file



4. In the **Image Information** area, set the following parameters.

Figure 2-34 Configuring image information



- **Boot Mode**: Select **BIOS** or **UEFI**. Ensure that the selected boot mode is the same as that in the image file, or the ECSs created from this image will not be able to boot up.
- **OS**: Select the OS specified in the ISO file. To ensure that the image can be created and used properly, select an OS consistent with that in the image file.
- **System Disk**: Set the system disk capacity (value range: 40 GB to 1024 GB), which must be no less than the capacity of the system disk in the image file.
- **Name**: Enter a name for the image to be created.
- **Enterprise Project**: Select the enterprise project to which your images belong.
- **Tag**: (Optional) Add a tag to the image to be created.
- **Description**: (Optional) Enter image description as needed.

5. Click **Apply Now**.
6. Confirm the settings and click **Submit Application**.

Step 3 Switch back to the **Image Management Service** page to check the image status.

When the image status changes to **Normal**, the image is registered successfully.

----End

2.12.3 Creating a Linux ECS from an ISO Image

Scenarios

This section describes how to create an ECS from a registered ISO image.

Procedure

Step 1 Access the IMS console.

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.

   The IMS console is displayed.

Step 2 Use an ISO image to create a Linux ECS.

1. Click the **Private Images** tab. Locate the row that contains the ISO image and click **Create ECS** in the **Operation** column.
2. Configure the ECS as prompted and click **OK**.

----End

Follow-up Procedure

After the ECS is created, you can log in remotely to continue with OS and drivers installation.

2.12.4 Installing a Linux OS

Scenarios

This section uses CentOS 7 64-bit as an example to describe how to install Linux on an ECS.

The installation procedure varies depending on the image file you use. Perform operations as prompted.

NOTE

Set the time zone, repo source update address, input method, language, and other items based on service requirements.

Prerequisites

You have remotely logged in to the ECS and entered the installation page.

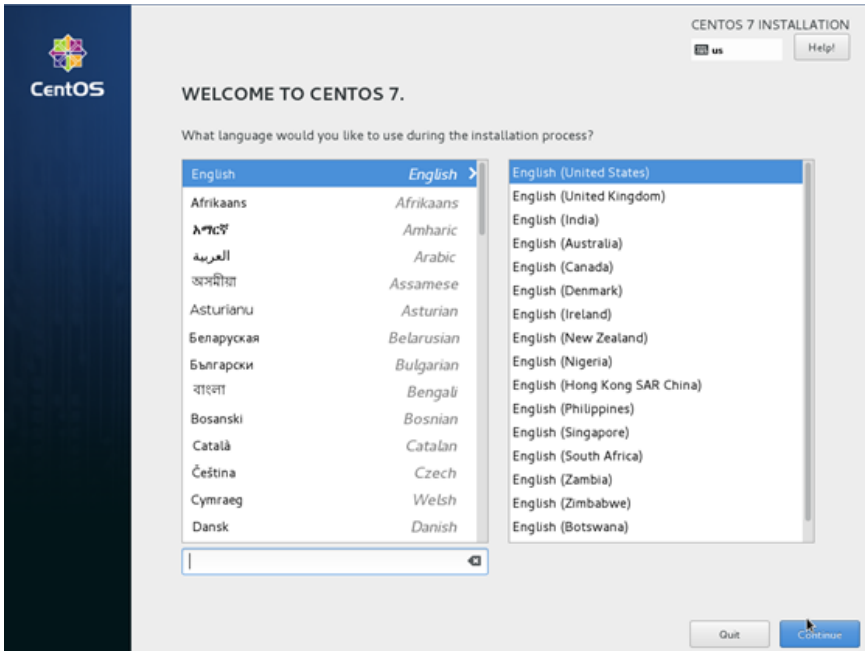
Procedure

CAUTION

Do not stop or restart the ECS during the OS installation. Otherwise, the OS installation will fail.

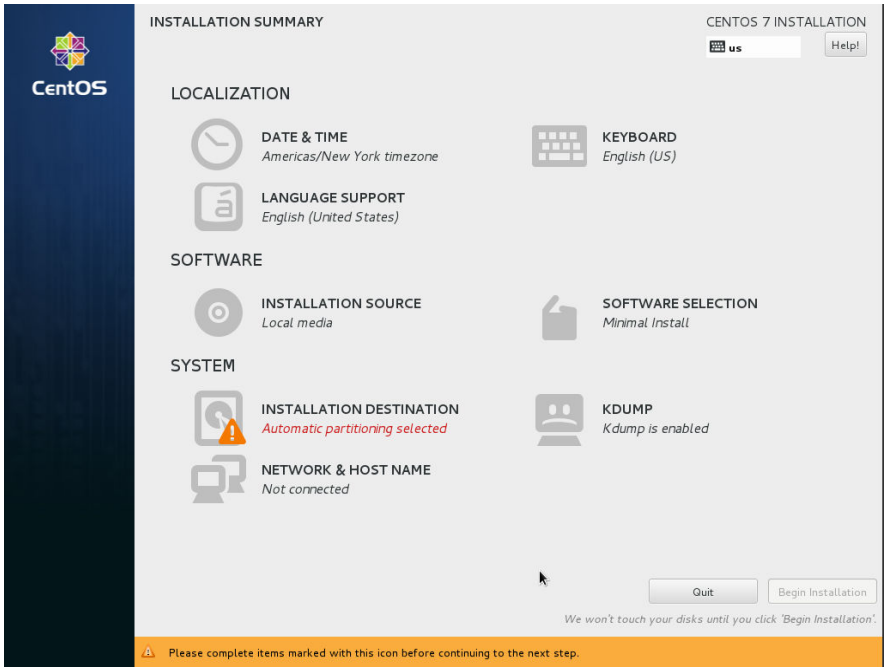
1. On the installation page, select the language and click **Continue**.

Figure 2-35 Installation page



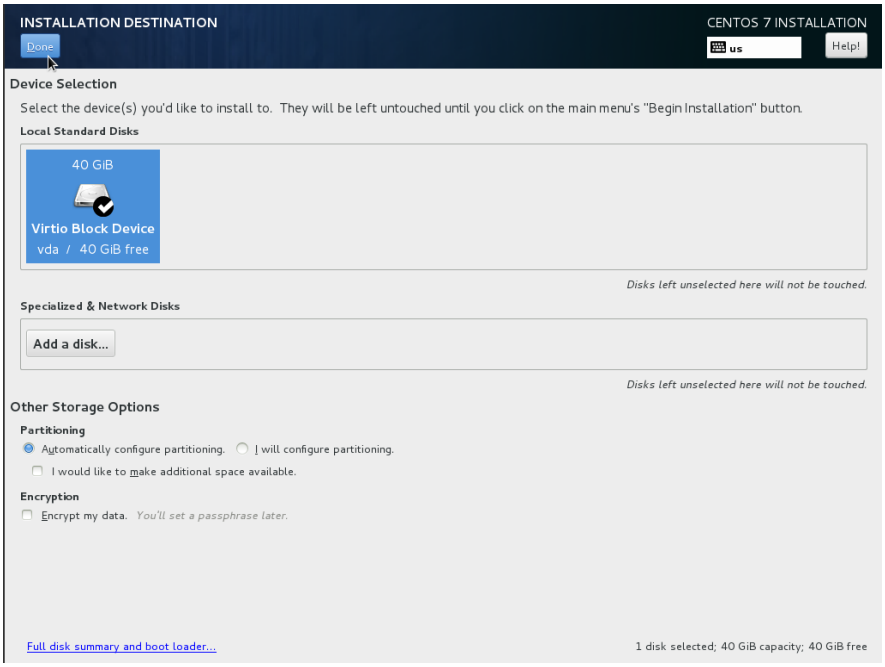
2. On the **INSTALLATION SUMMARY** page, choose **SYSTEM** > **INSTALLATION DESTINATION**.

Figure 2-36 INSTALLATION SUMMARY page



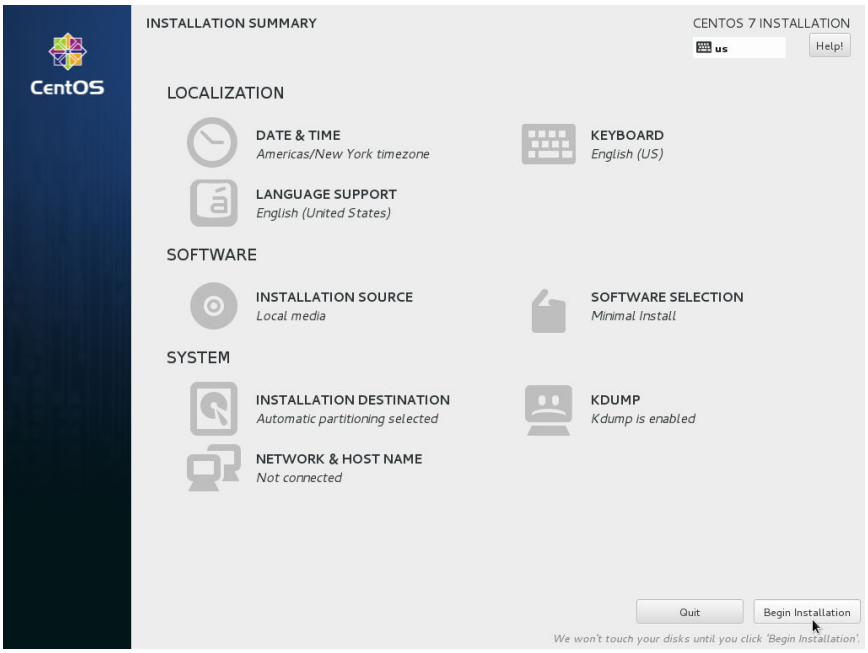
3. Select the target disk and click **Done**.

Figure 2-37 Installation location



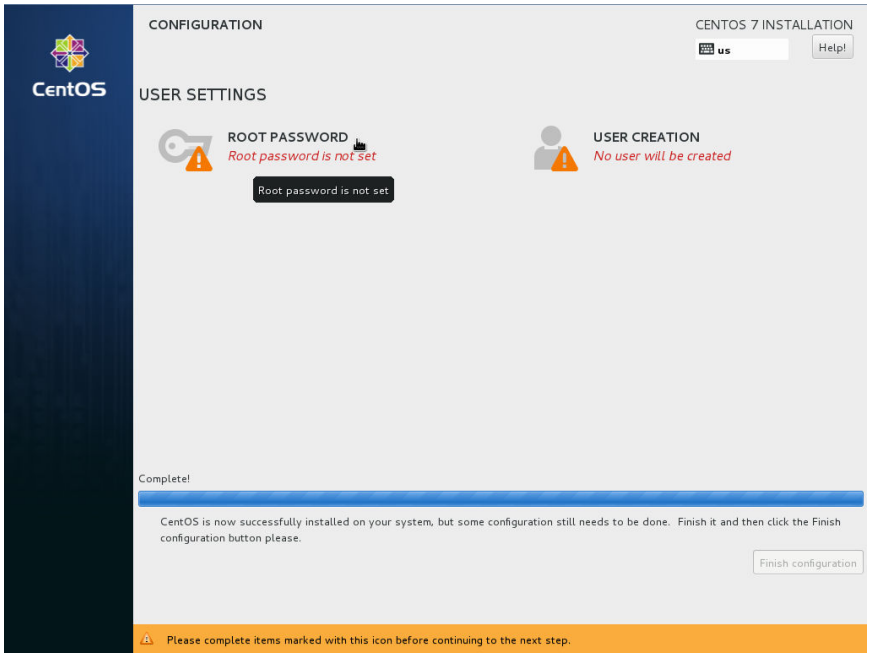
4. Click **Begin Installation**.

Figure 2-38 Starting installation



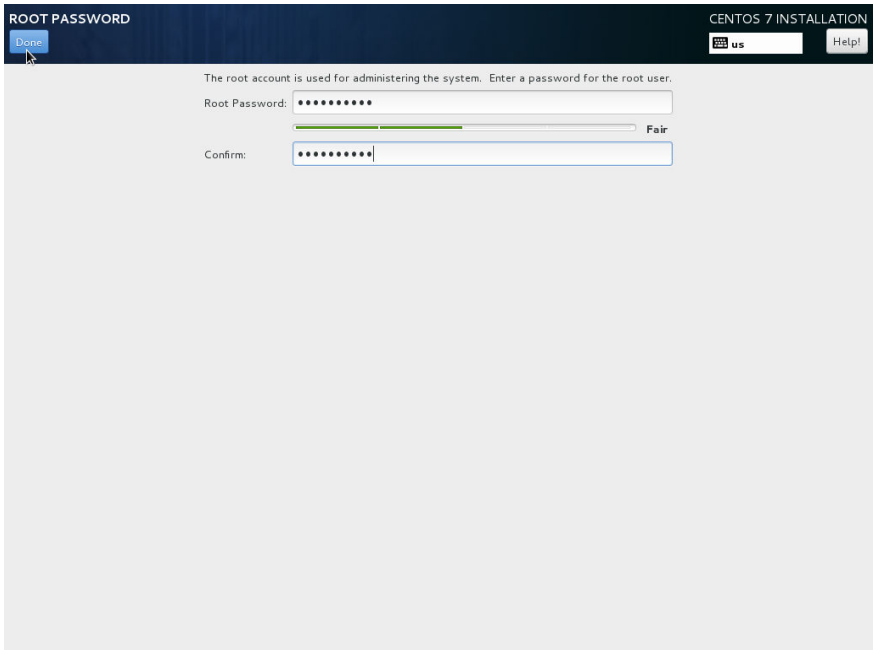
5. Wait for the automatic OS installation to complete. When the progress reaches 100%, CentOS is installed successfully.

Figure 2-39 Successful installation



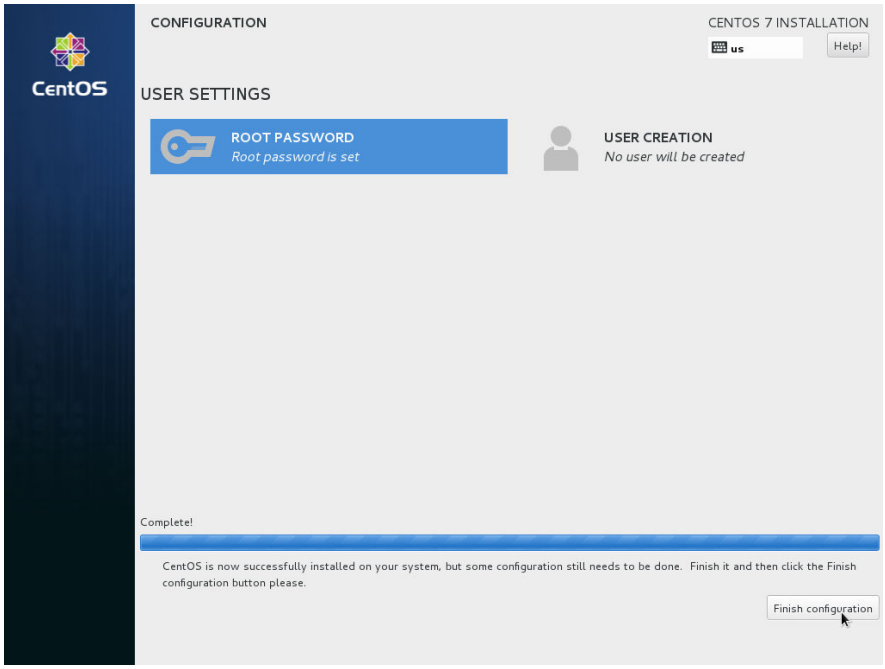
6. In the **USER SETTINGS** area, click **ROOT PASSWORD**.
   The **ROOT PASSWORD** page is displayed.
7. Set a password for user **root** as prompted and click **Done**.

Figure 2-40 Setting a password for user root



8. Click **Finish configuration**.

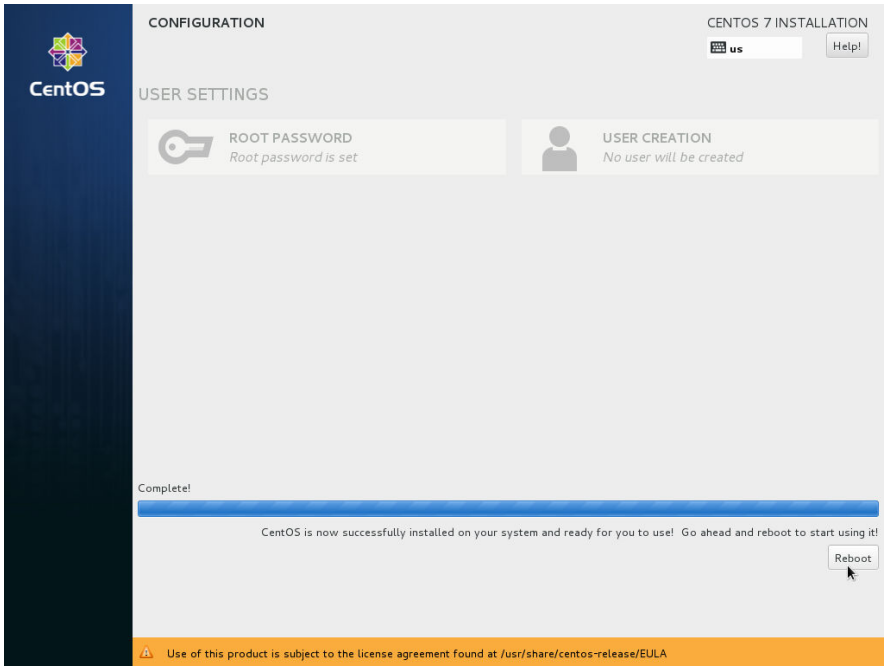
Figure 2-41 Completing configuration



9. Click **Reboot**.

   If you are prompted to install the OS again after the ECS is restarted, exit the VNC login page and restart the ECS on the console.

Figure 2-42 Restarting the ECS



2.12.5 Configuring the ECS and Creating a Linux System Disk Image

Scenarios

After installing an OS for the temporary ECS, configure the ECS and install KVM drivers to ensure that ECSs created from this temporary ECS can work properly.

This section describes how to configure a Linux ECS, install drivers, and create a Linux system disk image.

Procedure

Step 1 Configure the ECS.

1. Configure the network.
   - Run the **ifconfig** command to check whether the private IP address of the ECS is the same as that displayed on the console. If they are inconsistent, delete files from the network rule directory as instructed in **Deleting Files from the Network Rule Directory**.
   - Check whether NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in **Setting the NIC to DHCP**.
   - Run the **service sshd status** command to check whether SSH is enabled. If it is disabled, run the **service sshd start** command to enable it. Ensure that your ECS firewall, for example, Linux iptables, allows access to SSH.
2. Install drivers.

   To ensure that the network performance and basic functions of the ECSs created from the private image are normal, install KVM drivers on the ECS used to create the image.

   NOTE

   Disable your antivirus and intrusion detection software. You can enable them after installation of KVM drivers.

   - Install native KVM drivers. For details, see **Installing Native KVM Drivers**.
   - After the drivers are installed, you need to clear log files and historical records. For details, see **Clearing System Logs**.
3. Configure a file system.
   - Change the disk identifier in the GRUB configuration file to UUID. For details, see **Changing the Disk Identifier in the GRUB Configuration File to UUID**.
   - Change the disk identifier in the fstab file to UUID. For details, see **Changing the Disk Identifier in the fstab File to UUID**.
   - Clear the automatic attachment information of non-system disks in the **/etc/fstab** file to prevent impacts on subsequent data disk attachment. For details, see **Detaching Data Disks from an ECS**.
4. (Optional) Configure value-added functions.

- Install and configure Cloud-Init. For details, see **Installing Cloud-Init** and **Configuring Cloud-Init**.
- Enable NIC multi-queue. For details, see **How Do I Enable NIC Multi-Queue for an Image?**
- Configure dynamic assignment of IPv6 addresses. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Step 2 Create a Linux system disk image.

For details, see **Creating a System Disk Image from a Linux ECS**.

----End

Follow-up Procedure

After the system disk image is created, delete the temporary ECS in a timely manner to prevent it from occupying compute resources.

2.13 Importing an Image

You need to prepare an image file that meets the platform requirements.

Constraints

- For details about the restrictions on Windows image files, see **Preparing an Image file (Windows)**.
- For details about the restrictions on Linux image files, see **Preparing an Image file (Linux)**.

NOTE

- You are advised to complete network, tool, and driver configurations on the source VM and then export the image file. You can also complete the configurations on the created ECSs. For details, see **What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?** and **What Do I Do If a Linux Image File Is Not Pre-Configured When I Use It to Register a Private Image?**
- Currently, a large image file (maximum: 1 TB) can be imported only in RAW or ZVHD2 format. In addition to meeting the requirements for common image files, a bitmap file needs to be generated for each RAW image file. The bitmap file will be uploaded together with the image file. For details, see **Quickly Importing an Image File**.

Import

IMS provides multiple methods for importing images. You can select a method based on the image file type, format, or size.

Table 2-14 Importing an image

Format	File Size	Reference
VMDK, VHD, QCOW2, VHDX, QED, VDI, QCOW, or ZVHD	Not larger than 128 GB	• **Creating a Windows System Disk Image from an External Image File** • **Creating a Linux System Disk Image from an External Image File** • **Creating a Data Disk Image from an External Image File**
RAW or ZVHD2	No larger than 1 TB	• **Creating a Data Disk Image from an External Image File**
ISO	Not larger than 128 GB	• **Creating a Windows System Disk Image from an ISO File** • **Creating a Linux System Disk Image from an ISO File**

2.14 Quickly Importing an Image File

2.14.1 Overview

If an image file is larger than 128 GB, you can import it using fast import.

Constraints

- The image file must be in RAW or ZVHD2 format.
- The image file size cannot exceed 1 TB.

Methods

You can import an image file in any of the following methods depending on the file format:

- ZVHD2
  a. Optimize the image file.
  b. Upload the image file to an OBS bucket.
  c. Register the image file on the cloud platform.
- RAW
  a. Optimize the image file.
  b. Generate a bitmap file for the image file.
  c. Upload the image file and bitmap file to an OBS bucket.
  d. Register the image file on the cloud platform.
- Others

- If the file format is converted to ZVHD2:
  i. Optimize the image file.
  ii. Convert the image file format to ZVHD2.
  iii. Upload the image file to an OBS bucket.
  iv. Register the image file on the cloud platform.
- If the file format is converted to RAW:
  i. Optimize the image file.
  ii. Convert the image file format to RAW and generate a bitmap file for the image file.
  iii. Upload the image file and bitmap file to an OBS bucket.
  iv. Register the image file on the cloud platform.

NOTE

- The import of large files depends on lazy loading which defers loading of file data until it is needed. This reduces the initial loading time. However, RAW files do not support this feature. When you upload a RAW file, you need to upload its bitmap together.
- For details about how to optimize an image file, see **Optimization Process** (Windows) or **Optimization Process** (Linux) depending on the OS type specified in the image file.

Import Process

The following describes how to import an external image file. Assume that you need to convert the file format to ZVHD2 or RAW.

You can use **qemu-img-hw** or the open-source tool **qemu-img** to convert the image format. **qemu-img-hw** can only be used in Linux.

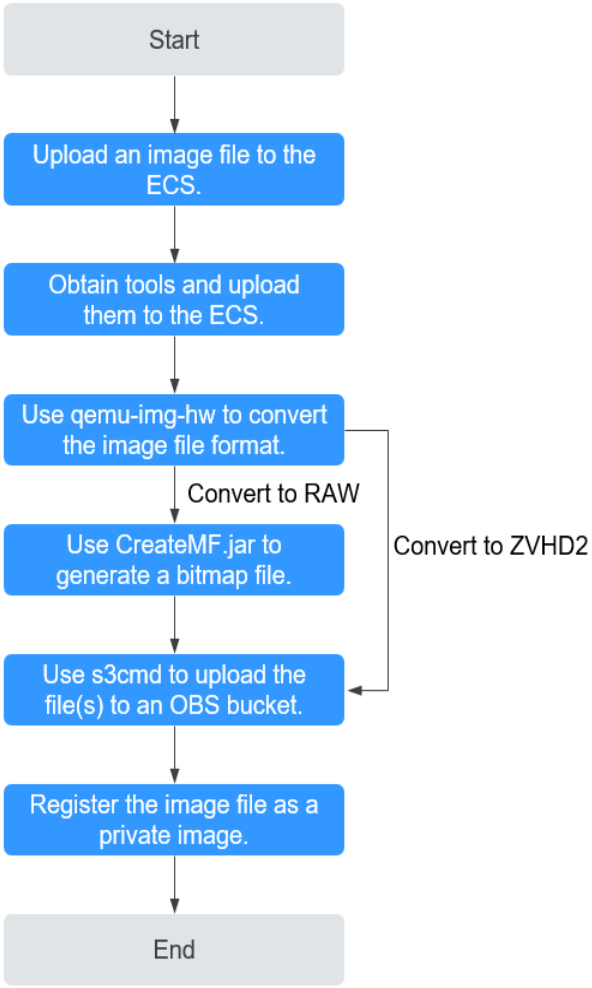
NOTE

The tool package contains **qemu-img-hw** (for converting image formats) and **CreateMF.jar** (for generating bitmap files).

- Linux

  You are advised to use an EulerOS ECS to convert the file format.

Figure 2-43 Import process



For details, see **Quickly Importing an Image File (Linux)**.

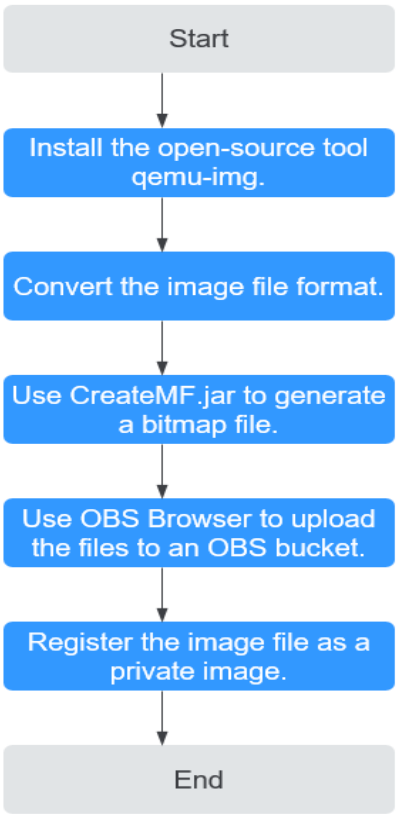
- Windows

  You are advised to use a local PC running Windows to convert the file format.

  NOTE

  qemu-img cannot convert image files to the ZVHD2 format. You need to convert an image file to the RAW format and then use **CreateMF.jar** to generate a bitmap file.

Figure 2-44 Import process (Windows)



For details, see **Quickly Importing an Image File (Windows)**.

2.14.2 Quickly Importing an Image File (Linux)

Scenarios

This section describes how to convert the format of an image file on a Linux server and then quickly import it to the cloud platform. You are advised to use an EulerOS ECS for converting image file formats and generating bitmap files.

In Linux, you are advised to use **qemu-img-hw** to convert image formats.

Prerequisites

- The image file has been optimized. For details, see **Optimization Process** (Windows) or **Optimization Process** (Linux). Ensure that the image file meets the requirements in **Table 2-6** (Windows) or **Table 2-10** (Linux).

  NOTE

  Select the reference content based on the OS type in the image file.

- You have created an ECS running EulerOS on the management console and bound an EIP to the ECS.
- An OBS bucket has been created on the management console.

Procedure

Step 1 Upload an image file.

- If the image file is uploaded from a Linux PC, run the **scp** command.

  For example, to upload **image01.qcow2** to the **/usr/** directory of the ECS, run the following command:

  scp /var/image01.qcow2 root@*xxx.xxx.xx.xxx*:/usr/

  xxx.xxx.xx.xxx indicates the EIP bound to the ECS.
- If the image file is uploaded from a Windows PC, use a file transfer tool, such as WinSCP, to upload the image file.

Step 2 Obtain the image conversion tool (**qemu-img-hw.zip**) and bitmap file generation tool (**createMF.zip**), upload them to the ECS, and decompress the packages.

Table 2-15 Tool packages

Tool Package	How to Obtain
qemu-img-hw.zip	**https://ecs-instance-driver.obs.ru-moscow-1.hc.sbercloud.ru/qemu-img-hw.zip**
createMF.zip	**https://ecs-instance-driver.obs.ru-moscow-1.hc.sbercloud.ru/createMF.zip**

Step 3 Use **qemu-img-hw** to convert the image format.

1. Go to the directory where **qemu-img-hw** is stored, for example, **/usr/qemu-img-hw**.

   cd /usr/qemu-img-hw
2. Run the following command to make **qemu-img-hw** executable:

   chmod +x qemu-img-hw
3. Execute **qemu-img-hw** to convert the image file format to ZVHD2 (recommended) or RAW.

   Command format:

   ./qemu-img-hw convert -p -O *Target_image_format Source_image_file Target_image_file*

   For example, run the following command to convert an **image01.qcow2** file to an **image01.zvhd2** file:

   ./qemu-img-hw convert -p -O zvhd2 image01.qcow2 image01.zvhd2

   – If the image file is converted to the ZVHD2 format, go to **Step 5**.

   – If the image file is converted to the RAW format, go to **Step 4**.

Step 4 Use **CreateMF.jar** to generate a bitmap file.

1. Ensure that JDK has been installed on the ECS.

   Run the following commands to check whether JDK is installed:

   source /etc/profile

   java -version

   If a Java version is displayed, JDK has been installed.

2. Run the following command to enter the directory where **CreateMF.jar** is stored:

   cd /usr/createMF

3. Run the following command to generate a bitmap file:

   java -jar CreateMF.jar */Original RAW file path/Generated .mf file path*

   Example:

   java -jar CreateMF.jar image01.raw image01.mf

> ⚠ CAUTION
>
> – The generated .mf bitmap file must have the same name as the RAW image file. For example, if the image file name is **image01.raw**, the generated bitmap name is **image01.mf**.

Step 5 Use **s3cmd** to upload the file(s) to an OBS bucket.

1. Install **s3cmd** on the ECS.

   If **s3cmd** has been installed, skip this step.

   a. Run the following command to install setuptools:

      yum install python-setuptools

   b. Run the following command to install wget:

      yum install wget

   c. Run the following commands to obtain the **s3cmd** software package:

      wget https://github.com/s3tools/s3cmd/archive/master.zip

      mv master.zip s3cmd-master.zip

   d. Run the following commands to install **s3cmd**:

      unzip s3cmd-master.zip

      cd s3cmd-master

      python setup.py install

2. Configure **s3cmd**.

   Run the following command to configure **s3cmd**:

   s3cmd --configure
   Access Key: *Enter an AK.*
   Secret Key: *Enter an SK.*
   Default Region: *Enter the region where the bucket is located.*
   S3 Endpoint: *Refer to the OBS endpoint.*
   DNS-style bucket+hostname:port template for accessing a bucket: *Enter a server address with a bucket name, for example, mybucket.obs.myclouds.com.*
   Encryption password: *Press Enter.*
   Path to GPG program: *Press Enter.*
   Use HTTPS protocol: *Specifies whether to use HTTPS. The value can be Yes or No.*
   HTTP Proxy server name: *Specifies the proxy address used to connect the cloud from an external network. (If you do not need it, press Enter.)*
   HTTP Proxy server port: *Specifies the proxy port used to connect to the cloud from an external network (If you do not need it, press Enter.)*
   Test access with supplied credentials? y
   (If "Success. Your access key and secret key worked fine :-)" is displayed, the connection is successful.)
   Save settings? y *(Specifies whether to save the configurations. If you enter y, the configuration will be saved.)*

NOTE

The configurations will be stored in **/root/.s3cfg**. If you want to modify these configurations, run the **s3cmd --configure** command to configure the parameters or run the **vi .s3cfg** command to edit the **.s3cfg** file.

3. Run the following command to upload the ZVHD2 image file (or the RAW image file and its bitmap file) to an OBS bucket.

   s3cmd put *image01.zvhd2* **s3://***mybucket***/**

CAUTION

The .mf bitmap file must be in the same OBS bucket as the RAW image file.

Step 6 Register a private image.

You can register a private image using the converted ZVHD2 or RAW file on the console or using an API.

Method 1: Register a private image on the console.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. In the upper right corner, click **Create Image**.
3. In the **Image Type and Source** area, select **Import Image** for **Type** and then select **System disk image** or **Data disk image** for **Image Type**.
4. Select the bucket storing the ZVHD2 or RAW image file and then select the image file. If the image file is in the RAW format, you also need to select its bitmap file.
5. Select **Enable Fast Create**, and select the sentence following **Image File Preparation**.
6. Set parameters as prompted.

   For details about the parameters, see **Registering an External Image File as a Private Image**.

CAUTION

- The OS must be the same as that in the image file.
- The system disk capacity must be greater than that specified in the image file.

  Run the following command to check the system disk capacity in the image file:

  qemu-img-hw info *test.zvhd2*

Method 2: Register a private image using an API.

You can use the POST /v2/cloudimages/quickimport/action API to quickly import an image file.

For details about how to call this API, see "Importing an Image File Quickly" in *Image Management Service API Reference*.

----End

Appendix 1: Common qemu-img-hw Commands

- Converting image file formats: **qemu-img-hw convert -p -O** *Target_image_format Source_image__file Target_image_file*

  The parameters are described as follows:

  -p: indicates the conversion progress.

  The part following **-O** (which must be in upper case) consists of the target image format, source image file, and target image file.

  For example, run the following command to convert a QCOW2 image file to a ZVHD2 file:

  qemu-img-hw convert -p -O zvhd2 test.qcow2 test.zvhd2
- Querying image file information: **qemu-img-hw info** *Source image file*

  An example command is **qemu-img-hw info test.zvhd2**.
- Viewing help information: **qemu-img-hw -help**

Appendix 2: Common Errors During qemu-img-hw Running

- Symptom:

  The following information is displayed when you run the **qemu-img-hw** command:

  ```
  ./qemu-img-hw: /lib64/libc.so.6: version `GLIBC_2.14' not found (required by ./qemu-img-hw)
  ```

  Solution:

  Run the **strings /lib64/libc.so.6 | grep glibc** command to check the glibc version. If the version is too early, install the latest version. Run the following commands in sequence:

  wget http://ftp.gnu.org/gnu/glibc/glibc-2.15.tar.gz

  wget http://ftp.gnu.org/gnu/glibc/glibc-ports-2.15.tar.gz

  tar -xvf glibc-2.15.tar.gz

  tar -xvf glibc-ports-2.15.tar.gz

  mv glibc-ports-2.15 glibc-2.15/ports

  mkdir glibc-build-2.15

  cd glibc-build-2.15

  ../glibc-2.15/configure --prefix=/usr --disable-profile --enable-add-ons --with-headers=/usr/include --with-binutils=/usr/bin

  NOTE

  If **configure: error: no acceptable C compiler found in $PATH** is displayed, run the **yum -y install gcc** command.

  make

  make install

- Symptom:

  The following information is displayed when you run the **qemu-img-hw** command:

  ```
  ./qemu-img-hw: error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory
  ```

  Solution: Run the **yum install libaio** command first.

2.14.3 Quickly Importing an Image File (Windows)

Scenarios

This section describes how to convert the format of an image file on a Windows server and then quickly import it to the cloud platform. You are advised to use a local Windows PC for converting image formats and generating bitmap files.

In Windows, use the open-source tool **qemu-img** to convert image formats. **qemu-img** supports conversion between image files of the VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED formats. Convert an image to the RAW format and then use the **CreateMF.jar** tool to generate a bitmap file.

Prerequisites

- The image file has been optimized. For details, see **Optimization Process** (Windows) or **Optimization Process** (Linux). Ensure that the image file meets the requirements in **Table 2-6** (Windows) or **Table 2-10** (Linux).

  NOTE

  Select the reference content based on the OS type in the image file.

- An OBS bucket has been created on the management console, and OBS Browser+ has been ready.

Procedure

Step 1 Install the open-source image conversion tool **qemu-img**.

Step 2 Run the **cmd** command to go to the **qemu-img** installation directory and run the **qemu-img** command to convert the image file to the RAW format.

For example, run the following command to convert an **image.qcow2** file to an **image.raw** file:

qemu-img convert -p -O raw image.qcow2 image.raw

Step 3 Use **CreateMF.jar** to generate a bitmap file.

1. Obtain the **CreateMF.jar** package and decompress it.

   Table 2-16 CreateMF.jar package

Tool Package	How to Obtain
createMF.zip	**https://ecs-instance-driver.obs.ru-moscow-1.hc.sbercloud.ru/createMF.zip**

2. Ensure that JDK has been installed in the current environment.

   You can verify the installation by running **cmd.exe** and then **java -version**. If Java version information is displayed, JDK has been installed.

3. Go to the directory where **CreateMF.jar** is stored.

   For example, if you have downloaded **CreateMF.jar** to **D:/test**, run the following commands to access the directory:

   D:

   cd test

4. Run the following command to generate a bitmap file for the RAW image file:

   java -jar CreateMF.jar D:/image01.raw D:/image01.mf

> ⚠ CAUTION
>
> – The generated .mf bitmap file must have the same name as the RAW image file. For example, if the image file name is **image01.raw**, the generated bitmap name is **image01.mf**.

Step 4 Use OBS Browser+ to upload the converted image file and its bitmap file to an OBS bucket.

You must upload the RAW image file and its bitmap file to the same OBS bucket.

Step 5 Register a private image.

You can register a private image using the converted ZVHD2 or RAW file on the console or using an API.

Method 1: Register a private image on the console.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. In the upper right corner, click **Create Image**.
3. In the **Image Type and Source** area, select **Import Image** for **Type** and then select **System disk image** or **Data disk image** for **Image Type**.
4. Select the bucket storing the ZVHD2 or RAW image file and then select the image file. If the image file is in the RAW format, you also need to select its bitmap file.
5. Select **Enable Fast Create**, and select the sentence following **Image File Preparation**.
6. Set parameters as prompted.

   For details about the parameters, see **Registering an External Image File as a Private Image**.

CAUTION

- The OS must be the same as that in the image file.
- The system disk capacity must be greater than that specified in the image file.

  Run the following command to check the system disk capacity in the image file:

  qemu-img-hw info *test.zvhd2*

Method 2: Register a private image using an API.

You can use the POST /v2/cloudimages/quickimport/action API to quickly import an image file.

For details about how to call this API, see "Importing an Image File Quickly" in *Image Management Service API Reference*.

----End

3 Managing Private Images

3.1 Modifying an Image

Scenarios

You can modify the following attributes of a private image:

- Name
- Description
- Minimum Memory
- Maximum Memory
- NIC Multi-Queue

  NIC multi-queue enables multiple CPUs to process NIC interruptions for load balancing. For details, see **How Do I Enable NIC Multi-Queue for an Image?**
- Boot Mode

Constraints

- You can only modify a private image in the **Normal** state.

Procedure

Use any of the following methods to modify an image:

Method 1:

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab to display the image list.
3. Locate the row that contains the image and click **Modify** in the **Operation** column.

4. In the **Modify Image** dialog box, modify the image.

Figure 3-1 Modifying an image

Method 2:

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab to display the image list.
3. On the image list, click the name of the target image.
4. On the image details page, click **Modify** in the upper right corner. In the **Modify Image** dialog box, modify image attributes.

Method 3:

The system allows you to quickly change the name of a private image.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.

2. Click the **Private Images** tab.
3. In the private image list, locate the target image and move the cursor to the **Name** column.
4. Click  to change the image name.
5. Click **OK**.

3.2 Exporting Image List

Scenarios

You can export the public or private image list in the current region as a CSV file to your local PC.

- For public images, the file describes the image name, image status, OS, image type, image creation time, system disk, and minimum memory.
- For private images, the file describes the image name, image ID, image status, OS, image type, image creation time, disk capacities, shared disks, image size, minimum memory, and encryption.

Exporting Private Image Information

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab and click  .
   The system will automatically export the private image list in the current region under your account to a local directory.

   NOTE

   The file name is in the format of **private-images-***Region ID-Export time*.

Exporting Public Image Information

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Public Images** tab and click  .
   The system will automatically export all public images in the current region to a local directory.

   NOTE

   The file name is in the format of **public-images-***Region ID-Export time*.

3.3 Checking the Disk Capacity of an Image

Scenarios

You can check the disk capacity of a private image.

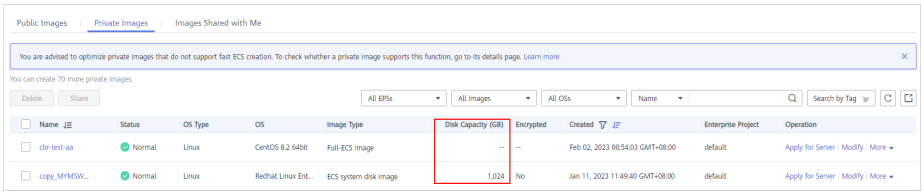
- To check the disk capacity of a system disk image, data disk image, or ISO image, see **Check the Disk Capacity of a System Disk Image, Data Disk Image, or ISO Image**.
- To check the disk capacity of a full-ECS image, see **Check the Disk Capacity of a Full-ECS Image**.

Check the Disk Capacity of a System Disk Image, Data Disk Image, or ISO Image

Check the disk capacity in the **Disk Capacity** column of the private image list.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab to display the image list.
3. Check the value in the **Disk Capacity** column. The unit is **GB**.

Figure 3-2 Checking the disk capacity of a system disk image, data disk image, or ISO image

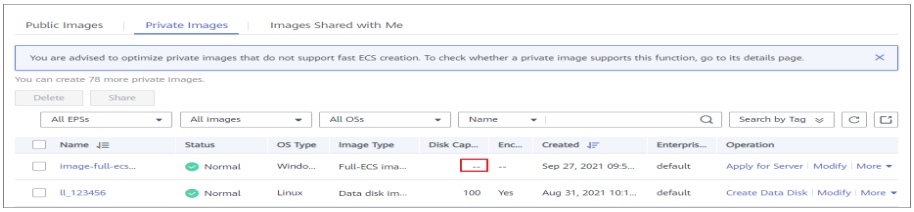


Check the Disk Capacity of a Full-ECS Image

The disk capacity of a full-ECS image is the sum of the system disk capacity and data disk capacity in the backup from which the full-ECS image is created.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab to display the image list.
   The value in the **Disk Capacity** column is **--**.

Figure 3-3 Checking the disk capacity of a full-ECS image



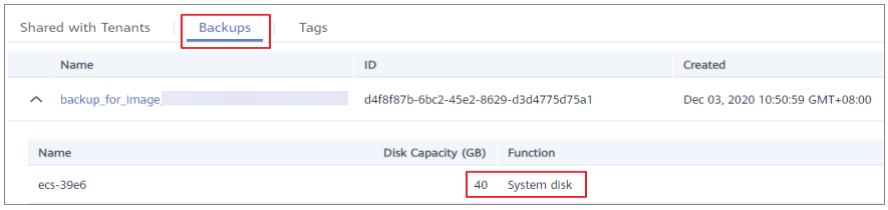
3. Click the full-ECS image name.
4. Click the **Backups** tab and view the capacities of the system disk and data disks in the backup.

   Disk capacity of a full-ECS image = Capacity of the system disk in the backup + Capacity of data disks in the backup

   For example:

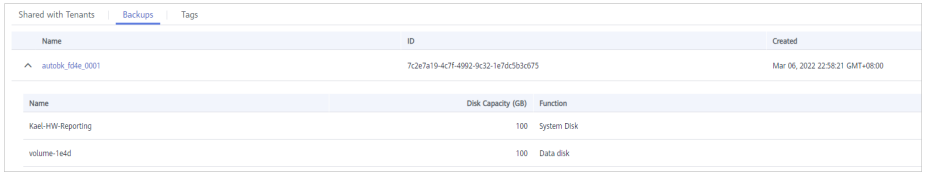
   - If the system disk capacity is 40 GB and no data disk is attached, the capacity of the full-ECS image disk is 40 GB.

     Figure 3-4 Checking backup details

   - If the system disk capacity is 40 GB and data disk capacity is 40 GB, the full-ECS image disk capacity is 80 GB.

     Figure 3-5 Checking backup details

3.4 Creating an ECS from an Image

Scenarios

You can use a public, private, or shared image to create an ECS.

- If you use a public image, the created ECS contains an OS and preinstalled public applications. You need to install applications as needed.
- If you use a private or shared image, the created ECS contains an OS, preinstalled public applications, and a user's personal applications.

Procedure

1. Access the IMS console.

a. Log in to the management console.

b. Under **Computing**, click **Image Management Service**.

The IMS console is displayed.

2. Click the **Public Images**, **Private Images**, or **Images Shared with Me** tab to display the image list.
3. Locate the row that contains your desired image and click **Apply for ECS** in the **Operation** column.
4. For details about how to create an ECS, see *Elastic Cloud Server User Guide*.

When you use a system disk image to create an ECS, you can set the ECS specifications and system disk type without considering those in the image, but the system disk capacity can only be larger than that in the image.

When you use a full-ECS image to create an ECS, the system and data disk information defaulted by the image will be automatically displayed. You can increase the capacity of a system disk or data disks, but cannot decrease it.

NOTE

If a full-ECS image contains multiple data disks, it takes some time to load and display the disk information.

3.5 Deleting Images

Scenarios

You can delete private images that will no longer be used.

- Deleted private images cannot be retrieved. Perform this operation only when absolutely necessary.
- After a private image is deleted, it cannot be used to create ECSs or EVS disks.
- After a private image is deleted, ECSs created from the image can still be used and are still billed. However, the OS cannot be reinstalled for the ECSs and ECSs with the same configuration cannot be created.
- Deleting the source image of a replicated image has no effect on the replicated image. Similarly, deleting a replicated image has no effect on its source.

Procedure

1. Access the IMS console.

a. Log in to the management console.

b. Under **Computing**, click **Image Management Service**.

The IMS console is displayed.

2. Click the **Private Images** tab to display the image list.
3. Locate the row that contains the image, choose **More** > **Delete** in the **Operation** column.

NOTE

To delete multiple images:

1. Select the images you want to delete in the image list.
2. Click **Delete** above the image list.

4. (Optional) Select **Delete cloud server backups of the full-ECS images**.

   This parameter is available only when you have selected full-ECS images from the image list.

   If you select this option, the system will delete CBR backups of the full-ECS images.

   NOTE

   If CBR backups failed to be deleted, the cause may be that these backups are being created and cannot be deleted. In this case, manually delete them as prompted.

5. Click **Yes**.

3.6 Sharing Images

3.6.1 Overview

You can share your private images with other tenants. The tenants who accept the shared images can use the images to create ECSs of the same specifications.

CAUTION

The cloud platform is not responsible for the integrity or security of shared images. When you use a shared image, ensure that the image is from a trusted sharer.

Constraints

- You can share images only within the region where they reside. To share an image across regions, you need to replicate the image to the target region first.
- A system disk image or data disk image can be shared with up to 128 tenants, and a full-ECS image can be shared with up to 10 tenants.
- Encrypted images cannot be shared.
- Only full-ECS images created from CBR backups can be shared. Other full-ECS images cannot be shared.

Procedure

If you want to share a private image with another tenant, the procedure is as follows:

1. You obtain the account name of the tenant.

   If the tenant is a multi-project user, you also need to obtain the project name from the tenant.

2. You share an image with the tenant.
3. The tenant accepts the shared image.

   After accepting the image, the tenant can use it to create ECSs.

Related FAQs

If you have any questions, see **Image Sharing FAQs**.

3.6.2 Obtaining the Account Name and Project Name

Scenarios

Before a tenant shares an image with you, you need to provide your account name. If you are a multi-project user, you also need to provide your project name. This section describes how to obtain your account name and project name.

Procedure

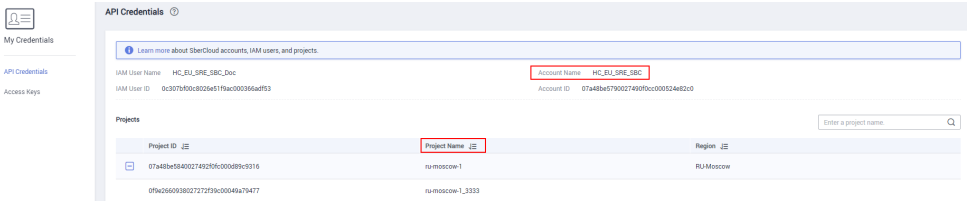
1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. Click the username in the upper right corner and select **My Credentials** from the drop-down list.

   On the **My Credentials** page, view the account name and project name (value in the **Project Name** column) in the project list.

   Images can be shared only within the region where they reside. So, obtain the project name in the same region.

   Figure 3-6 Viewing the account name and project name



3.6.3 Sharing Specified Images

Scenarios

After obtaining the account name from a tenant (if the tenant is a multi-project user, you also need to obtain the project name), you can share specified private images with the tenant. You can share a single image or multiple images as needed.

Prerequisites

- You have obtained the account name of the target tenant. (If the tenant is a multi-project user, you also need to obtain the project name.)

- Before sharing an image, ensure that any sensitive data has been deleted from the image.

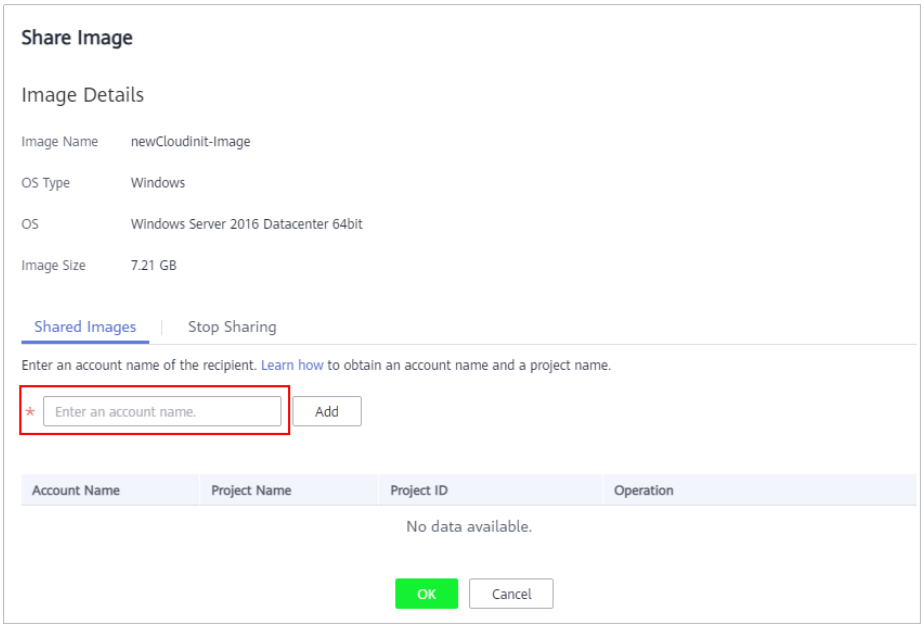
Procedure

- Share multiple images.
  a. Log in to the management console.
  b. Under **Computing**, click **Image Management Service**.
  c. Click the **Private Images** tab.
  d. Select the private images to share and click **Share** above the image list.
  e. In the **Share Image** dialog box, enter the account name of the target tenant and click **Add**. If the tenant is a multi-project user, you also need to select the project name.

     To add multiple target tenants, enter their account names (and project names) and then click **Add**.
  f. Click **OK**.
- Share a single image.
  a. Log in to the management console.
  b. Under **Computing**, click **Image Management Service**.
  c. Click the **Private Images** tab.
  d. Locate the row that contains the private image you are to share, click **More** in the **Operation** column, and select **Share** from the drop-down list.
  e. In the **Share Image** dialog box, enter the account name of the target tenant and click **Add**. If the tenant is a multi-project user, you also need to select the project name.

     To add multiple target tenants, enter their account names (and project names) and then click **Add**.

Figure 3-7 Sharing an image



f. Click **OK**.

Related Operations

After you share images with a tenant, the tenant can accept the shared images on the **Images Shared with Me** page on the IMS console. For detailed operations, see **Accepting or Rejecting Shared Images**.

3.6.4 Accepting or Rejecting Shared Images

Scenarios

After another tenant shares images with you, you will receive a message. You can choose to accept or reject all or some of the shared images.

NOTE

- If you are not in the same region as the tenant sharing the images with you, you will not receive the message.

Prerequisites

- Another tenant has shared images with you.
- If the shared image is a full-ECS image, you need to create a server backup vault to store the full-ECS image and the backups of the full-ECS image before accepting the shared image. When creating a server backup vault, set **Protection Type** to **Backup**.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Images Shared with Me** tab.
   A message is displayed above the image list asking you whether to accept the shared images.
   - To accept all the shared images, click **Accept All** in the upper right corner.
   - To accept some images, select the images and click **Accept**.
   - To reject some images, select the images and click **Reject**.

   NOTE

   If no message is displayed, check whether you have selected a correct region.
3. (Optional) In the **Accept Full-ECS Image** dialog box, select a server backup vault with the **Backup** protection type and click **OK**.
   This dialog box is displayed when the shared image is a full-ECS image.
   When accepting a full-ECS image, you must specify a vault for storing the CBR backups associated with the full-ECS image. The vault capacity must be no less than the total capacities of the system disk and data disk backups.

NOTE

For more information about server backup vaults, see *Cloud Backup and Recovery User Guide*.

Results

- **Pending**: If you do not immediately accept or reject a shared image, the image is in the **Pending** state.

  A pending shared image is not displayed in the shared image list.
- **Accepted**: After an image is accepted, it is displayed in the shared image list. You can use the image to create ECSs.
- **Rejected**: After an image is rejected, it is not displayed in the shared image list. You can click **Rejected Images** to view the images you have rejected and you can still choose to accept them.

Follow-up Procedure

After accepting a system disk image shared by another tenant, you can:

- Use the image to create one or more ECSs (select **Shared Image** during ECS creation). For details, see "Purchasing an ECS" in *Elastic Cloud Server User Guide*.
- Use the image to change the OS of existing ECSs. For details, see "Changing the OS" in *Elastic Cloud Server User Guide*.

After accepting a data disk image shared by another tenant, you can use the image to create EVS disks (locate the row that contains the image and click **Create Data Disk** in the **Operation** column).

3.6.5 Rejecting Accepted Images

Scenarios

You can reject accepted images if you no longer need them.

After an image is rejected, it will not be displayed on the **Images Shared with Me** page.

Prerequisites

You have accepted images shared by other users.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. Click the **Images Shared with Me** tab.
3. Determine the next step based on how many images you are to reject.

- To reject multiple images: select the images to be rejected and click **Reject** above the image list. In the displayed dialog box, click **Yes**.
- To reject a specific image: locate the image to be rejected and choose **More** > **Reject** in the **Operation** column. In the displayed dialog box, click **Yes**.

3.6.6 Accepting Rejected Images

Scenarios

If you want to use the shared images you have rejected, you can accept them from the list of rejected images.

Prerequisites

- You have rejected the images shared by others.
- The image owners have not stopped sharing the images.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Images Shared with Me** tab.
3. Click **Rejected Images**. All the rejected images are displayed.
4. Select the images you want to accept and click **Accept**.
5. Check the accepted images in the shared image list.

3.6.7 Stopping Sharing Images

Scenarios

You can stop sharing images. After you stop sharing an image:

- The image will be invisible to the recipient on the management console and no data will be returned when the recipient query the image through an API.
- The recipient cannot use the image to create an ECS or EVS disk, or change the OS of an ECS.
- The recipient cannot reinstall the OS of the ECSs created from the shared image or create instances identical with these ECSs.

Prerequisites

You have shared private images with others.

Procedure

1. Access the IMS console.

a. Log in to the management console.
b. Under **Computing**, click **Image Management Service**.
   The IMS console is displayed.
2. Click the **Private Images** tab.
3. Locate the row that contains the private image that you no longer want to share, and choose **More** > **Share** in the **Operation** column.
4. In the **Share Image** dialog box, click the **Stop Sharing** tab.
5. Select the account name for which you want to stop image sharing and click **OK**.

3.6.8 Adding Tenants Who Can Use Shared Images

Scenarios

In addition to the tenants you have shared images with, you can add more tenants who can use the shared images.

Prerequisites

- You have shared private images.
- You have obtained the account name of the tenant to be added. If the tenant is a multi-project user, you also need to obtain the project name.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab.
3. Click the image name to view image details.
4. Click **Add Tenant**.
5. In the **Add Tenant** dialog box, enter the account name (and select the project name if the tenant to be added is a multi-project user). Then, click **Add**.
   If you want to add multiple tenants, enter their account names (and select the project names if the tenants to be added are multi-project users). Then, click **Add**.

3.6.9 Deleting Image Recipients Who Can Use Shared Images

Scenarios

This section describes how to delete image recipients who can use shared images.

Prerequisites

- You have shared private images.

- You have obtained account names of the image recipients.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab.
3. Click the image name to view image details.
4. View the tenants who can use shared image.
5. Delete one or all of the recipients:
   – To delete a single image recipient, locate the target recipient and click **Delete**.
   – To delete all image recipients, click **Delete All** above the image recipient list.
6. Click **Yes**.

3.6.10 Replicating a Shared Image

Scenarios

Replicate a private image that was shared with you. The image is displayed in the private image list. You can export, share, and replicate this image, or use it to create ECSs.

Constraints

- Currently, only system and data disk images can be replicated. Full-ECS images are not supported.
- Currently, images can only be replicated within a region.
- An image to be replicated cannot be larger than 128 GB.
- An image cannot be replicated to generate an encrypted image.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. On the displayed IMS console, click the **Images Shared with Me** tab.
   Shared images that are accepted are displayed.
3. Locate a shared image, click **More** in the **Operation** column, and select **Replicate** from the drop-down list.
4. In the displayed **Replicate Image** dialog box, enter the name and description of the image you want to obtain.

5. Click **OK**.

   You can click the **Private Images** tab and view the creation progress of the image in the private image list. When the image status changes to **Normal**, the image creation is complete.

3.7 Exporting an Image

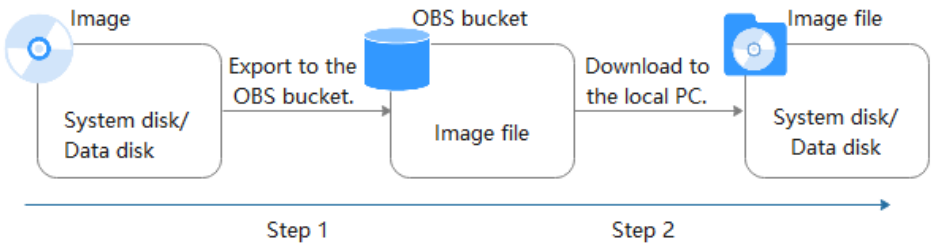
Scenarios

You can export a private image to a standard OBS bucket and then download it to your local PC.

Background

- You can reproduce cloud servers and their running environments in on-promises clusters or private clouds by exporting their images from the cloud platform. The following figure shows the process of exporting an image.

  Figure 3-8 Exporting an image



- The time required for exporting an image depends on the image size and the number of concurrent export tasks.
- You can export images in ZVHD2, QCOW2, VMDK, VHD, or ZVHD format. The default format of a private image is ZVHD2. Images exported in different formats may vary in size.
- If an image is larger than 128 GB, you can select **Enable** for **Fast Export** when exporting the image to an OBS bucket. The image will be exported as a ZVHD2 file. You can convert the image format after it is exported.

  NOTE

  Fast Export is unavailable for encrypted images. To export an encrypted image, decrypt it first.

Constraints

- An image can only be exported to a Standard bucket that is in the same region as the image.
- The following private images cannot be exported:
  - Full-ECS images
  - ISO images
  - Private images created from a Windows, SUSE, Red Hat, Ubuntu, or Oracle Linux public image

- The image size must be less than 1 TB. Images larger than 128 GB support only fast export.

Prerequisites

- You have Administrator permissions for OBS.
- An OBS bucket is available in the region where the private image is located.

  If no OBS bucket is available, create one by referring to *Object Storage Service User Guide*. Select **Standard** for **Storage Class**.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. Locate the row that contains the image to be exported, click **More** in the **Operation** column and select **Export**.
3. In the displayed **Export Image** dialog box, set the following parameters:
   - **Fast Export**: To export an image larger than 128 GB, you must enable fast export, and you cannot specify the format of the exported image (which can only be ZVHD2). After exporting the image, you can use **qemu-img-hw** to convert it to your desired format. For details, see **Step 3**.

     NOTE

     For details about differences between export and fast export, see **What Are the Differences Between Import/Export and Fast Import/Export?**
   - **Format**: Select one from **qcow2**, **vmdk**, **vhd**, and **zvhd** as you need.
   - **Name**: Enter a name that is easy to identify.
   - **Storage Path**: Click  to expand the bucket list and select an OBS bucket for storing the exported image.

     NOTE

     An image can only be exported to a Standard bucket that is in the same region as the image. So, only such buckets are available in the list.
4. Click **OK**.

   You can view the image export progress above the private image list.

Follow-up Procedure

After the image is exported successfully, you can download it from the OBS bucket through the management console or OBS Browser+.

3.8 Optimizing a Windows Private Image

3.8.1 Optimization Process

An ECS can run properly only after KVM Guest OS drivers (VirtIO drivers) are installed on it. To ensure that ECSs support KVM and to improve network performance, VirtIO drivers must be installed for the image.

1. Create an ECS from the Windows private image to be optimized and log in to the ECS.
2. Install VirtIO drivers which are needed to create KVM ECSs.

   For details, see **Installing VirtIO Drivers**.
3. On the ECS, choose **Control Panel** > **Power Options**. Click **Choose when to turn off the display**, select **Never** for **Turn off the display**, and save the changes.
4. Clear system logs and then stop the ECS.

   For details, see **Clearing System Logs**.
5. Create a Windows private image from the ECS.

3.8.2 Obtaining Required Software Packages

VirtIO Drivers

Download a VirtIO driver package from:

https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/

You can select a version as needed.

3.8.3 Installing VirtIO Drivers

Scenarios

This section only applies to KVM ECSs, which will replace Xen ECSs gradually. Before using an ECS or external image file to create a private image, ensure that VirtIO drivers have been installed in the OS so that ECSs created from this image can support KVM virtualization and the network performance can be improved.

CAUTION

If you do not install VirtIO drivers, ECS NICs cannot be detected. As a result, the ECSs cannot communicate with other resources.

If an ECS is created from a public image, VirtIO drivers have been installed by default.

Prerequisites

An EIP has been bound to the ECS. (This ECS is used to optimize a private image.)

Installing VirtIO Drivers

The following uses **virtio-win-gt-x64.msi** in **version virtio-win-0.1.189-1** as an example to describe how to install VirtIO drivers.

1. Log in to the Windows ECS using VNC.

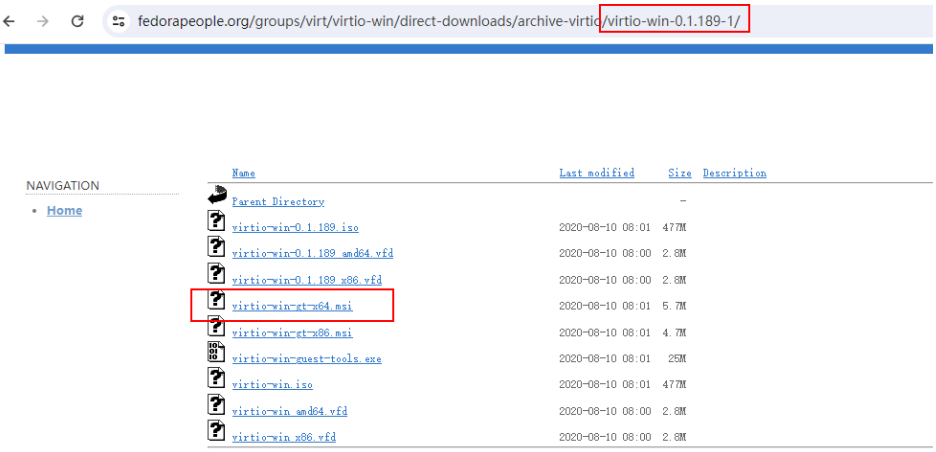
   For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

   NOTE

   You must log in to the ECS using VNC. Remote desktop connection is not allowed because the NIC driver needs to be updated during the installation but the NIC is in use for the remote desktop connection. As a result, the installation will fail.

2. Download a VirtIO driver package (**virtio-win-gt-x64.msi** as an example) of the required version by referring to **Obtaining Required Software Packages**.

   Figure 3-9 Downloading a driver package

3. After the download is complete, right-click **virtio-win-gt-x64.msi** and choose **Run as administrator** from the shortcut menu.

Figure 3-10 Starting the installation

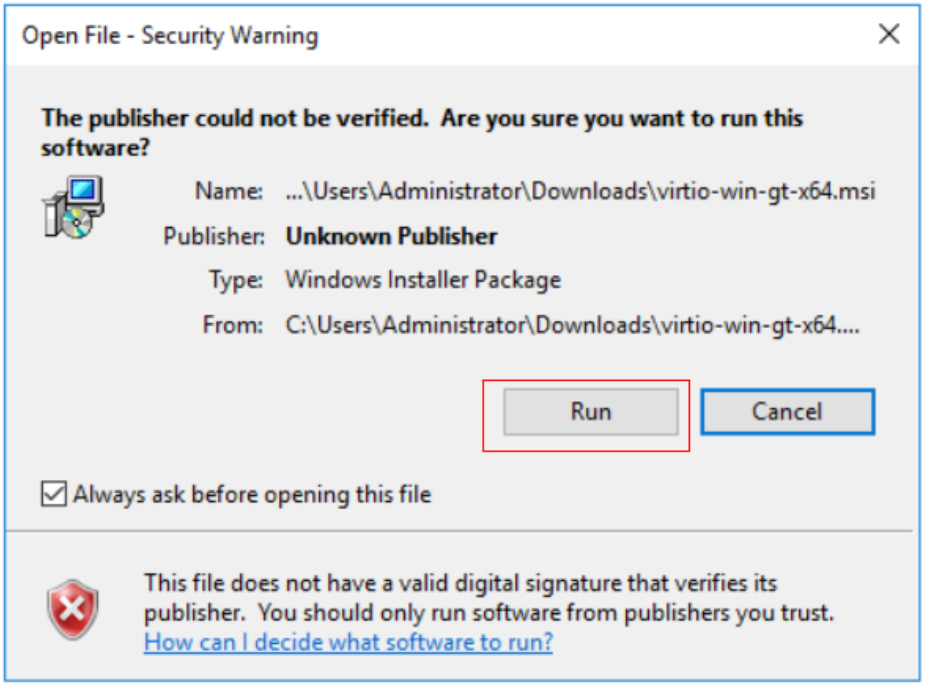


Figure 3-11 Installation wizard

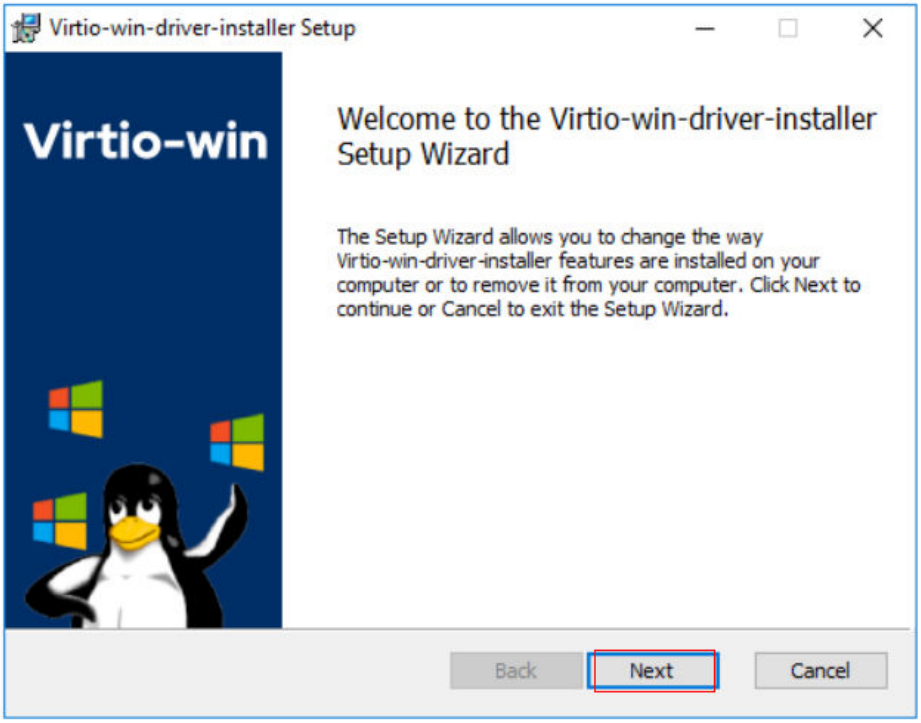
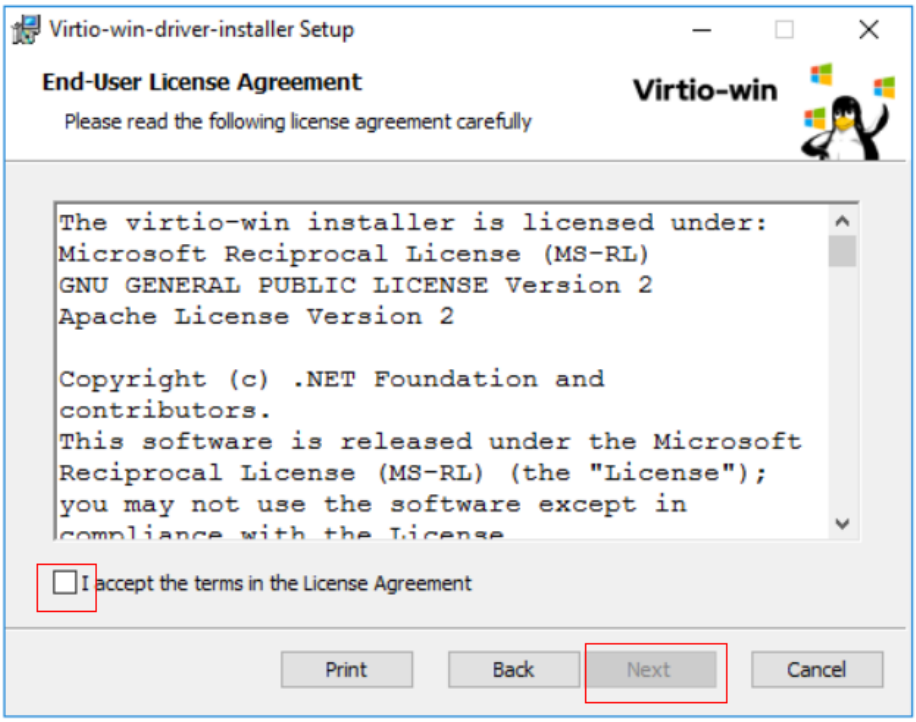


Figure 3-12 Accepting the agreement



Select the VirtIO drivers to be installed. In this example, select all VirtIO drivers.

Figure 3-13 Selecting VirtIO drivers to install

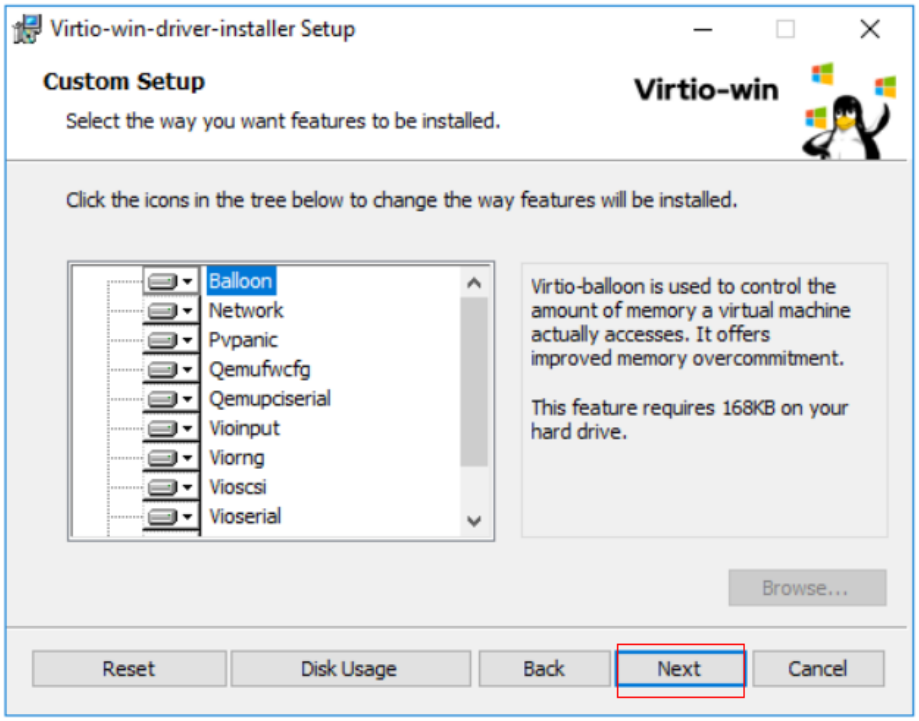
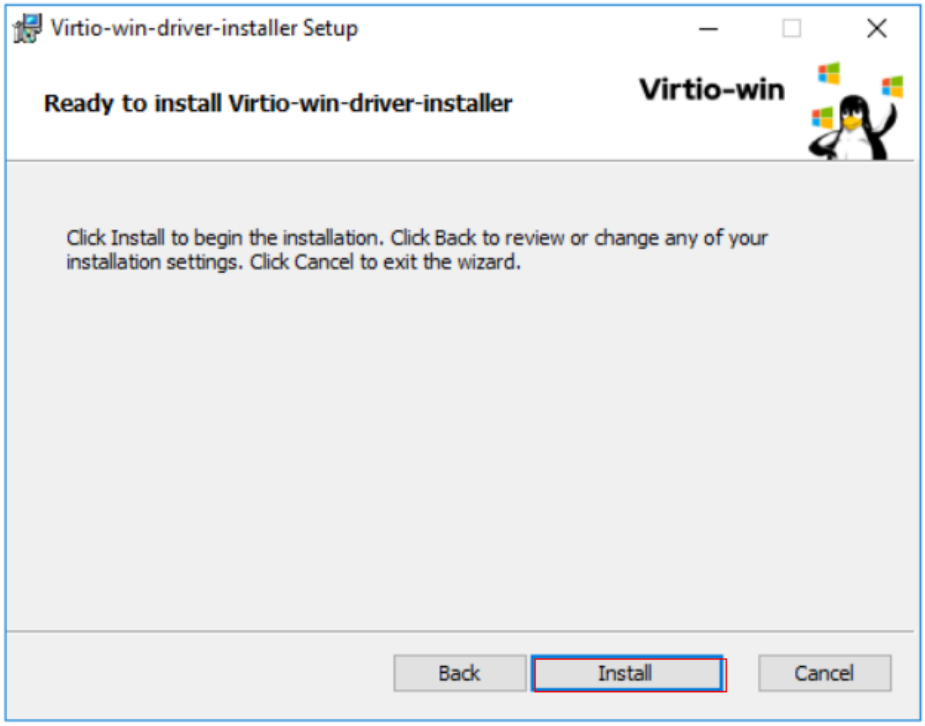
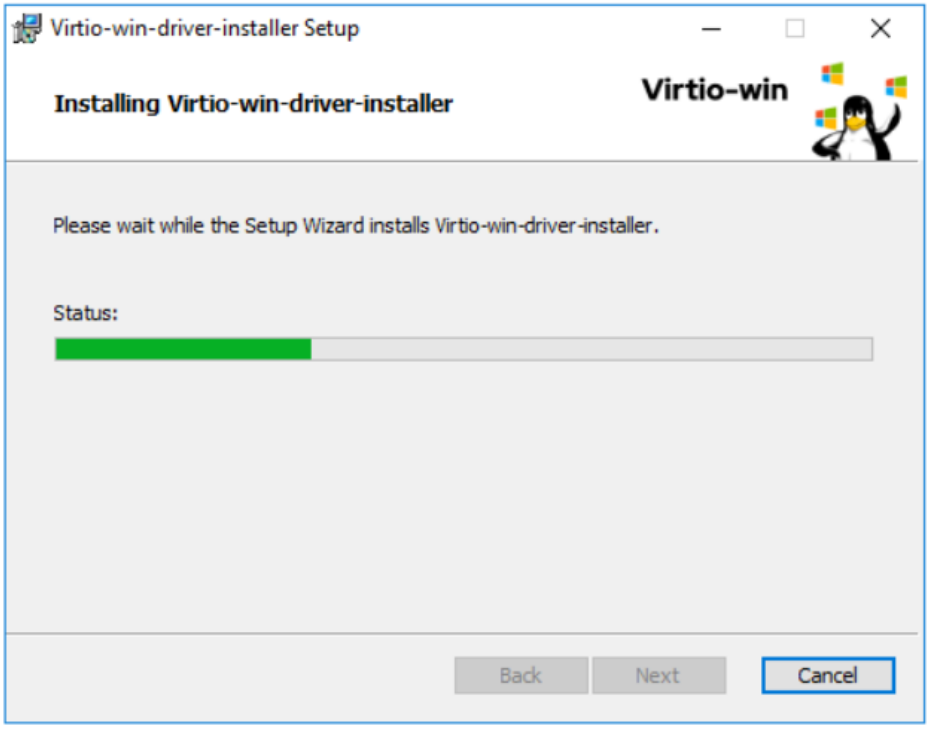


Figure 3-14 Proceeding with the installation.



4. Wait until the installation is complete.

Figure 3-15 Installation in process



5. Restart the ECS after the installation is complete.

Figure 3-16 Installation completed

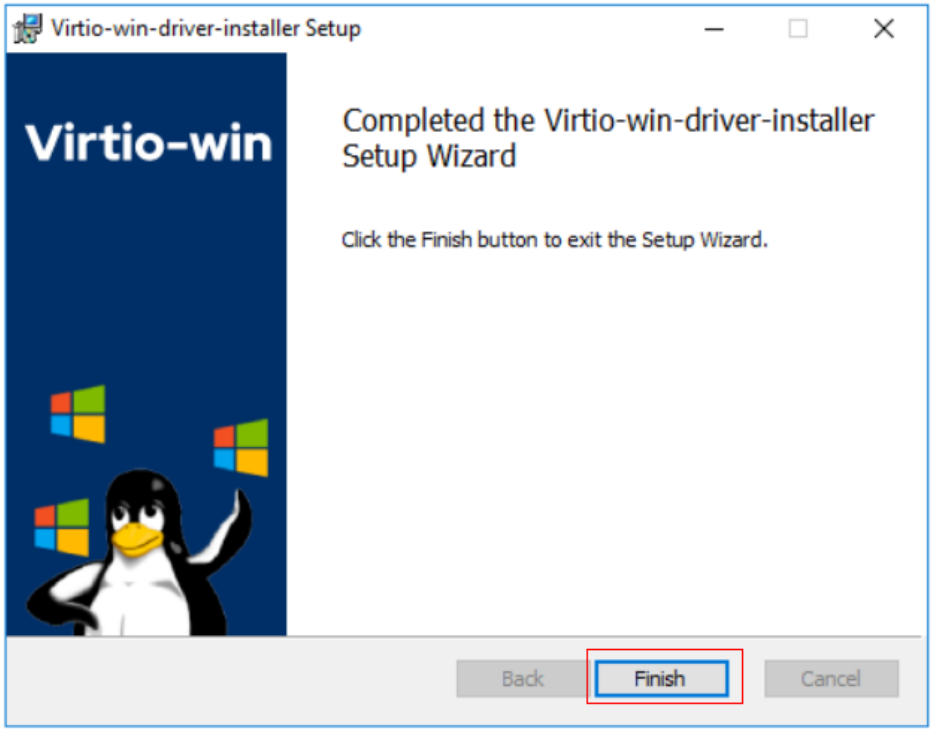
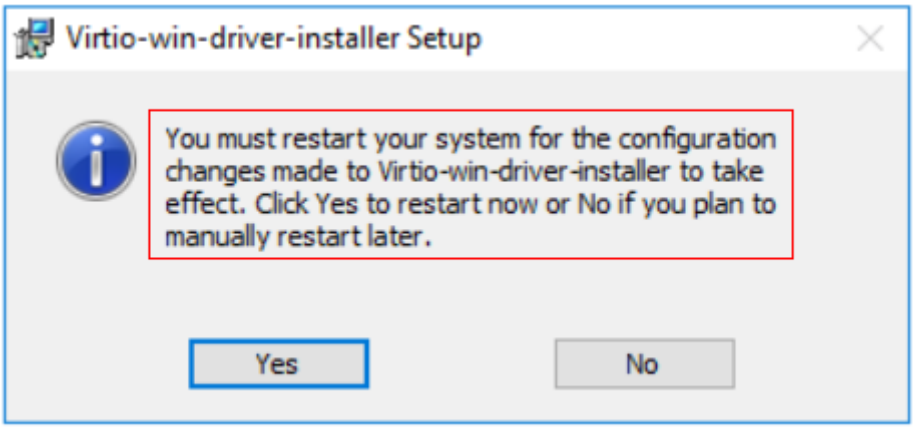


Figure 3-17 Restart prompt



6. After the restart, perform the operations in **Verifying the Installation** to verify that the VirtIO drivers have been successfully installed.

Verifying the Installation

Perform the following steps to verify the installation of the VirtIO drivers:

1. Open **Device Manager** and search for VirtIO drivers.
2. Check whether the VirtIO driver version and date displayed in **Device Manager** are the same as those of the VirtIO drivers you downloaded. If they are the same, the VirtIO drivers have been installed successfully.

Figure 3-18 Version and date of downloaded drivers

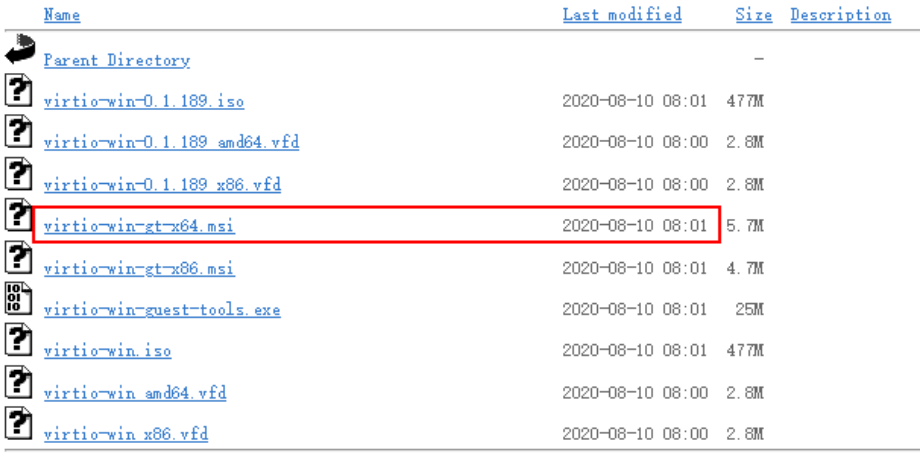
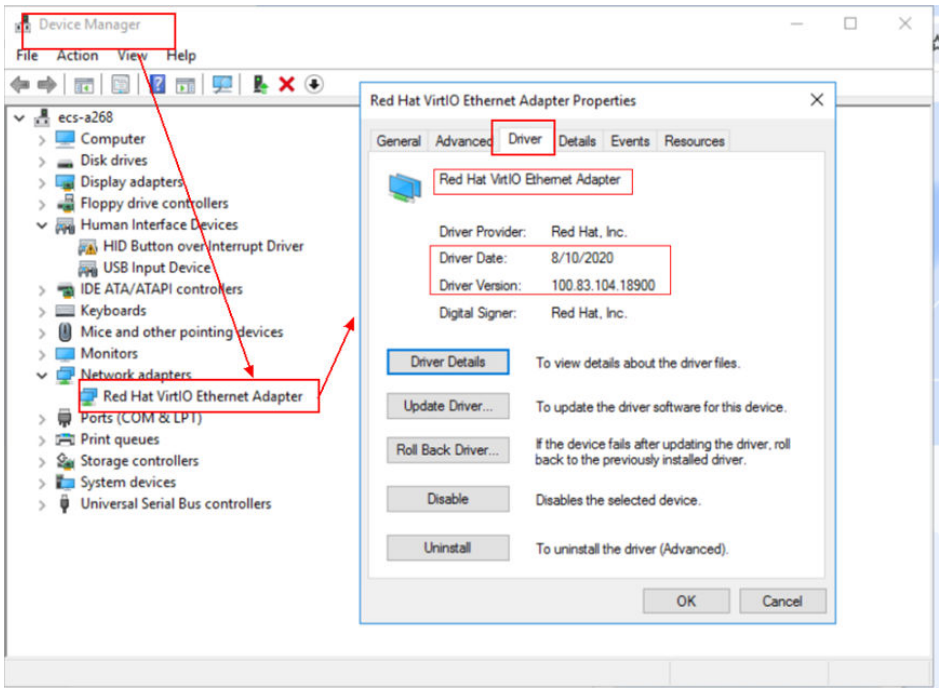


Figure 3-19 Version and date of drivers in Device Manager



3.8.4 Clearing System Logs

After installing PV and VirtIO drivers, perform the following operations to clear system logs:

1. For Windows Server 2008 and Windows Server 2012, right-click **Computer** and select **Manage**.
2. In the displayed dialog box, choose **System Tools** > **Event Viewer** > **Windows Logs** and delete logs of five items.
3. Stop the ECS.

3.9 Optimizing a Linux Private Image

3.9.1 Optimization Process

A Linux ECS can run properly only when native KVM (VirtIO) drivers have been installed on it and the disk ID in its GRUB configuration file and fstab file has been changed to UUID.

Preparations

1. Use the Linux image to be optimized to create an ECS, and start and log in to the ECS.
2. Check whether the private image needs to be optimized.

   For details, see **Checking Whether a Private Image Needs to be Optimized**.

Process

1. Change the disk ID in the GRUB configuration file to UUID.

   For details, see **Changing the Disk Identifier in the GRUB Configuration File to UUID**.
2. Change the disk ID in the fstab file to UUID.

   For details, see **Changing the Disk Identifier in the fstab File to UUID**.
3. Install native KVM drivers.

   For details, see **Installing Native KVM Drivers**.
4. Delete log files and historical records, and stop the ECS.

   For details, see **Clearing System Logs**.
5. Create a Linux private image from the ECS.

3.9.2 Checking Whether a Private Image Needs to be Optimized

- If the virtualization type is KVM and VirtIO drivers are not installed, optimization is required.
- If the virtualization type is KVM and VirtIO drivers are installed, optimization is not required.

Procedure

1. Check whether VirtIO drivers have been installed.
   - CentOS/EulerOS

     For initramfs, run the following command:

     lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

     For initrd, run the following command:

     lsinitrd /boot/initrd-`uname -r` | grep virtio

- Ubuntu/Debian

  lsinitramfs /boot/initrd.img-`uname -r` |grep virtio

- SUSE/openSUSE

  - SUSE 12 SP1/openSUSE 13 or earlier:

    lsinitrd /boot/initrd-`uname -r` | grep virtio

  - SUSE 12 SP1 or later than SUSE 12 SP1/openSUSE 13:

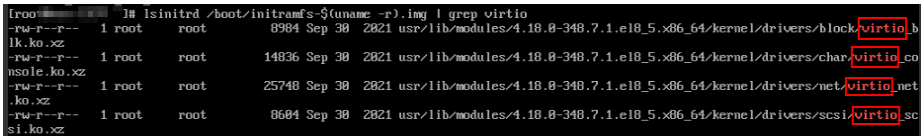
    For initramfs, run the following command:

    lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

    For initrd, run the following command:

    lsinitrd /boot/initrd-`uname -r` | grep virtio

If **virtio** is displayed, VirtIO drivers have been installed. For more information, see **Creating a Linux System Disk Image from an External Image File**.



Otherwise, VirtIO drivers have not been installed. Optimize the private image as instructed in **Process**.

3.9.3 Changing the Disk Identifier in the GRUB Configuration File to UUID

Scenarios

When optimizing a Linux private image, you need to change the disk identifier to UUID in the GRUB configuration file of the ECS.

Modify the **menu.lst** or **grub.cfg** configuration file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, **/boot/grub/grub.conf**, or **/boot/efi/EFI/euleros/grub.cfg**), and configure the boot partition using the UUID.

NOTE

The root partition identified in the configuration file varies depending on the OS. It may be **root=/dev/xvda** or **root=/dev/disk**.

Procedure

- Ubuntu 14.04: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub/grub.cfg** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required. The procedure is as follows:

  a. Log in to the ECS as user **root**.

  b. Run the following command to query all types of mounted file systems and device UUIDs:

     blkid

     The following information is displayed:

```
/dev/xvda1: UUID="ec51d860-34bf-4374-ad46-a0c3e337fd34" TYPE="ext3"
/dev/xvda5: UUID="7a44a9ce-9281-4740-b95f-c8de33ae5c11" TYPE="swap"
```

c. Run the following command to query the **grub.cfg** file:

cat /boot/grub/grub.cfg

The following information is displayed:

```
.....menuentry 'Ubuntu Linux, with Linux 3.13.0-24-generic' --class ubuntu --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.13.0-24-generic-advanced-
ec51d860-34bf-4374-ad46-a0c3e337fd34' {
recordfail
load_video
gfxmode $linux_gfx_mode
insmod gzio
insmod part_msdos
insmod ext2
if [ x$feature_platform_search_hint = xy ]; then
search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
else
search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
fi
echo 'Loading Linux 3.13.0-24-generic ...'
linux /boot/vmlinuz-3.13.0-24-generic root=/dev/xvda1 ro
echo 'Loading initial ramdisk ...'
initrd /boot/initrd.img-3.13.0-24-generic
}
```

d. Check whether the root partition in the **/boot/grub/grub.cfg** configuration file contains **root=/dev/xvda1** or **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34**.

- If **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34** is contained, the root partition is in the UUID format and requires no change.
- If **root=/dev/xvda1** is contained, the root partition is in the device name format. Go to **5**.

e. Identify the UUID of the root partition device based on **root=/dev/xvda1** (device name of the root partition) and the partition information obtained by running the **blkid** command.

f. Run the following command to open the **grub.cfg** file:

vi /boot/grub/grub.cfg

g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda1** to **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34**.

h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

i. Run the following command to verify the change:

cat /boot/grub/grub.cfg

The change is successful if information similar to the following is displayed:

```
.....menuentry 'Ubuntu Linux, with Linux 3.13.0-24-generic' --class ubuntu --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.13.0-24-generic-advanced-
ec51d860-34bf-4374-ad46-a0c3e337fd34' {
recordfail
load_video
gfxmode $linux_gfx_mode
insmod gzio
insmod part_msdos
```

```
insmod ext2
if [ x$feature_platform_search_hint = xy ]; then
search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
else
search --no-floppy --fs-uuid --set=root ec51d860-34bf-4374-ad46-a0c3e337fd34
fi
echo 'Loading Linux 3.13.0-24-generic ...'
linux /boot/vmlinuz-3.13.0-24-generic root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34 ro
echo 'Loading initial ramdisk ...'
initrd /boot/initrd.img-3.13.0-24-generic
}
```

- CentOS 6.5: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub/grub.conf** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required. The procedure is as follows:

  a. Log in to the ECS as user **root**.

  b. Run the following command to query all types of mounted file systems and device UUIDs:

     blkid

     The following information is displayed:

     ```
     /dev/xvda1: UUID="749d6c0c-990a-4661-bed1-46769388365a" TYPE="swap"
     /dev/xvda2: UUID="f382872b-eda6-43df-9516-5a687fecdce6" TYPE="ext4"
     ```

  c. Run the following command to query the **grub.conf** file:

     cat /boot/grub/grub.conf

     The following information is displayed:

     ```
     default=0
     timeout=5
     splashimage=(hd0,1)/boot/grub/splash.xpm.gz
     hiddenmenu
     title CentOS (2.6.32-573.8.1.el6.x86_64)
     root (hd0,1)
     kernel /boot/vmlinuz-2.6.32-573.8.1.el6.x86_64 ro root=/dev/xvda2 rd_NO_LUKS rd_NO_LVM
     LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16
     crashkernel=autoKEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
     initrd /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img
     ```

  d. Check whether the root partition in the **/boot/grub/grub.conf** configuration file contains **root=/dev/xvda2** or **root=UUID=f382872b-eda6-43df-9516-5a687fecdce6**.

     - If **root=UUID=f382872b-eda6-43df-9516-5a687fecdce6** is contained, the root partition is in the UUID format and requires no change.
     - If **root=/dev/xvda2** is contained, the root partition is in the device name format. Go to **5**.

  e. Identify the UUID of the root partition device based on **root=/dev/xvda2** (device name of the root partition) and the partition information obtained by running the **blkid** command.

  f. Run the following command to open the **grub.conf** file:

     vi /boot/grub/grub.conf

  g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda2** to **root=UUID=f382872b-eda6-43df-9516-5a687fecdce6**.

h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

i. Run the following command to verify the change:

cat /boot/grub/grub.conf

The change is successful if information similar to the following is displayed:

```
default=0
timeout=5
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-573.8.1.el6.x86_64)
root (hd0,1)
kernel /boot/vmlinuz-2.6.32-573.8.1.el6.x86_64 ro root=UUID=f382872b-eda6-43df-9516-5a687fecdce6 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=autoKEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img
```

- CentOS 7.0: Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub2/grub.cfg** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required.

  a. Log in to the ECS as user **root**.

  b. Run the following command to query all types of mounted file systems and device UUIDs:

  blkid

  The following information is displayed:

  ```
  /dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"
  /dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
  ```

  c. Run the following command to query the **grub.cfg** file:

  cat /boot/grub2/grub.cfg

  The following information is displayed:

  ```
  ......
  menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core)' --class fedora --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-advanced-4eb40294-4c6f-4384-bbb6-b8795bbb1130' {
  load_video
  set gfxpayload=keep
  insmod gzio
  insmod part_msdos
  insmod xfs
  set root='hd0,msdos2'
  if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint='hd0,msdos2'4eb40294-4c6f-4384-bbb6-b8795bbb1130
  else
  search --no-floppy --fs-uuid --set=root 4eb40294-4c6f-4384-bbb6-b8795bbb1130
  fi
  linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=/dev/xvda2 ro crashkernel=auto rhgb quiet LANG=en_US.UTF-8
  initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img
  }
  ```

  d. Check whether the root partition in the **/boot/grub2/grub.cfg** configuration file contains **root=/dev/xvda2** or **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130**.

- If **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130** is contained, the root partition is in the UUID format and requires no change.
- If **root=/dev/xvda2** is contained, the root partition is in the device name format. Go to **5**.

e. Identify the UUID of the root partition device based on **root=/dev/xvda2** (device name of the root partition) and the partition information obtained by running the **blkid** command.

f. Run the following command to open the **grub.cfg** file:

vi /boot/grub2/grub.cfg

g. Press **i** to enter editing mode and change the root partition to the UUID format, for example, from **root=/dev/xvda2** to **root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130**.

h. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

i. Run the following command to verify the change:

cat /boot/grub2/grub.cfg

The change is successful if information similar to the following is displayed:

```
.....
menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core)' --class fedora --class gnu-linux --
class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-
advanced-4eb40294-4c6f-4384-bbb6-b8795bbb1130' {
load_video
set gfxpayload=keep
insmod gzio
insmod part_msdos
insmod xfs
set root='hd0,msdos2'
if [ x$feature_platform_search_hint = xy ]; then
search --no-floppy --fs-uuid --set=root --hint='hd0,msdos2'4eb40294-4c6f-4384-bbb6-
b8795bbb1130
else
search --no-floppy --fs-uuid --set=root 4eb40294-4c6f-4384-bbb6-b8795bbb1130
fi
linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 ro crashkernel=auto rhgb quiet LANG=en_US.UTF-8
initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img
}
```

3.9.4 Changing the Disk Identifier in the fstab File to UUID

Scenarios

When optimizing a Linux private image, you need to change the disk identifier to UUID in the fstab configuration file of the ECS.

Procedure

- Take CentOS 7.0 as an example. Run **blkid** to obtain the UUIDs of all partitions. Modify the **/etc/fstab** file and use the partition UUIDs to configure automatic partition mounting.

1. Log in to the ECS as user **root**.

2. Run the following command to query all types of mounted file systems and device UUIDs:

   blkid

   The following information is displayed:

   ```
   /dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"
   /dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
   ```

3. Run the following command to query the **fstab** file:

   cat /etc/fstab

   The following information is displayed:

   ```
   [root@CTU1000028010 ~]# cat /etc/fstab
   /dev/xvda2  /      xfs     defaults    0 0
   /dev/xvda1  swap    swap    defaults    0 0
   ```

4. Check whether the disk identifier in the **fstab** file is the device name.
   - If the disk is represented by UUID, no further operation is required.
   - If the disk is represented by the device name, go to 5.

5. Run the following command to open the **fstab** file:

   vi /etc/fstab

6. Press **i** to enter editing mode and change the disk identifier in the **fstab** file to UUID.

- Take CentOS 7.1 as an example. Run **blkid** to obtain the UUIDs of all partitions. Modify the **/etc/fstab** file and use the partition UUIDs to configure automatic partition mounting.

1. Log in to the ECS as user **root**.
2. Run the following command to query all types of mounted file systems and device UUIDs:

   blkid

   ```
   /dev/xvda2: UUID="4eb40294-4c6f-4384-bbb6-b8795bbb1130" TYPE="xfs"
   /dev/xvda1: UUID="2de37c6b-2648-43b4-a4f5-40162154e135" TYPE="swap"
   ```

   Before the change:

   ```
   [root@CTU1000028010 ~]# cat /etc/fstab
   /dev/xvda2  /      xfs     defaults    0 0
   /dev/xvda1  swap    swap    defaults    0 0
   ```

   After the change:

   ```
   [root@CTU1000028010 ~]# cat /etc/fstab
   UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130  /      xfs     defaults    0 0
   UUID=2de37c6b-2648-43b4-a4f5-40162154e135  swap    swap    defaults    0 0
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.
4. Run the following command to verify the change:

   cat /etc/fstab

   The change is successful if information similar to the following is displayed:

   ```
   [root@CTU1000028010 ~]# cat /etc/fstab
   UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130  /      xfs     defaults    0 0
   UUID=2de37c6b-2648-43b4-a4f5-40162154e135  swap    swap    defaults    0 0
   ```

3.9.5 Installing Native KVM Drivers

Scenarios

When optimizing a Linux private image with Xen virtualization, you need to install native Xen and KVM drivers on the source ECS of the image.

This section describes how to install native KVM drivers.

CAUTION

If an ECS has no KVM drivers installed, the NICs of the ECS may not be detected and the ECS will be unable to communicate with other resources.

Prerequisites

- The virtualization type of the ECS is Xen.
- The kernel version must be later than 2.6.24.
- Disable your antivirus and intrusion detection software. You can enable them after the driver installation is complete.

Procedure

Modify the configuration file depending on the OS.

- CentOS, EulerOS

  Take CentOS 7.0 as an example. Modify the **/etc/dracut.conf** file. Add the VirtIO drivers to **add_drivers**. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

  For details, see **CentOS and EulerOS**.

- Ubuntu and Debian

  Modify the **/etc/initramfs-tools/modules** file. Add the VirtIO drivers. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/initramfs-tools/modules** file. Run the **update-initramfs -u** command to regenerate initrd.

  For details, see **Ubuntu and Debian**.

- SUSE and openSUSE
  - If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the **/etc/sysconfig/kernel** file and add Xen PV and VirtIO drivers to **INITRD_MODULES=""**. Xen PV drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **mkinitrd** command to regenerate initrd.
  - If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file and add Xen PV and VirtIO drivers to **add_drivers**. Xen PV drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **dracut -f** command to regenerate initrd.

- If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file and add Xen PV and VirtIO drivers to **add_drivers**. Xen PV drivers include xen-blkfront and xen-netfront. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

For details, see **SUSE and openSUSE**.

NOTE

For SUSE, run the following command to check whether xen-kmp (driver package for Xen PV) is installed:

rpm -qa |grep xen-kmp

If information similar to the following is displayed, xen-kmp is installed in the OS:

```
xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
```

If xen-kmp is not installed, obtain it from the ISO file and install it.

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected.

CentOS and EulerOS

1. Run the following command to open the **/etc/dracut.conf** file:

   vi /etc/dracut.conf

2. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add_drivers** (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
   # additional kernel modules to the default
   add_drivers+="xen-blkfront xen-netfront virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
   ......
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

4. Run the following command to regenerate initrd:

   dracut -f /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img

   If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

5. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitrd /boot/initramfs-`uname -r`.img | grep xen

   lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

   If the virtual file system is initrd, run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep xen

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   Assume that the virtual file system is initramfs. The following command output will be displayed:

   ```
   [root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep xen
   -rwxr--r--   1 root     root        54888 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
   ```

```
block/xen-blkfront.ko
-rwxr--r--  1 root  root  45664 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/
drivers/net/xen-netfront.ko

[root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
-rwxr--r--  1 root  root  23448 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
block/virtio_blk.ko
-rwxr--r--  1 root  root  50704 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/
drivers/net/virtio_net.ko
-rwxr--r--  1 root  root  28424 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
scsi/virtio_scsi.ko
drwxr-xr-x  2 root  root      0 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio
-rwxr--r--  1 root  root  14544 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio.ko
-rwxr--r--  1 root  root  21040 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_pci.ko
-rwxr--r--  1 root  root  18016 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
virtio/virtio_ring.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected. The drivers cannot be found by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

Ubuntu and Debian

1. Run the following command to open the **modules** file:

   vi /etc/initramfs-tools/modules

2. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to the **/etc/initramfs-tools/modules** file (the format varies depending on the OS).

```
[root@CTU10000xxxxx ~]#vi /etc/initramfs-tools/modules
......
# Examples:
#
# raid1
# sd_mOd
xen-blkfront
xen-netfront
virtio_blk
virtio_scsi
virtio_net
virtio_pci
virtio_ring
virtio
```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/initramfs-tools/modules** file.
4. Run the following command to regenerate initrd:

   update-initramfs -u

5. Run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitramfs /boot/initrd.img-`uname -r` |grep xen

   lsinitramfs /boot/initrd.img-`uname -r` |grep virtio

```
[root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep xen
lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen
lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen/netxen_nic.ko
lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback
```

```
lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback/xen-netback.ko
lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback
lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback/xen-blkback.ko

[root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
lib/modules/3.5.0-23-generic/kernel/drivers/scsi/virtio_scsi.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected. The drivers cannot be found by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

```
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
CONFIG_VIRTIO_BLK=y
CONFIG_VIRTIO_NET=y
CONFIG_VIRTIO=y
CONFIG_VIRTIO_RING=y
CONFIG_VIRTIO_PCI=y
CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y
CONFIG_XEN_BLKDEV_FRONTEND=y
CONFIG_XEN_NETDEV_FRONTEND=y
```

SUSE and openSUSE

If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the **/etc/sysconfig/kernel** file to add drivers. For details, see **scenario 1**.

If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file to add drivers. For details, see **scenario 2**.

If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file to add drivers. For details, see **scenario 3**.

- If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, perform the following steps:

  NOTE

  For SUSE, run the following command to check whether xen-kmp (driver package for Xen PV) is installed in the OS:

  rpm -qa |grep xen-kmp

  If information similar to the following is displayed, xen-kmp is installed:

  ```
  xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
  ```

  If xen-kmp is not installed, obtain it from the installation ISO and install it first.

  a. Run the following command to open the **/etc/sysconfig/kernel** file:

     vi /etc/sysconfig/kernel

  b. Add Xen PV and VirtIO drivers after **INITRD_MODULES=** (the format varies depending on the OS).

     ```
     SIA10000xxxxx:~ # vi /etc/sysconfig/kernel
     # (like drivers for scsi-controllers, for lvm or reiserfs)
     #
     INITRD_MODULES="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
     ```

  c. Run the **mkinitrd** command to regenerate initrd:

NOTE

If the virtual file system is not the default initramfs or initrd, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **menu.lst** or **grub.cfg** file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, or **/boot/grub2/grub.cfg**).

The following is an example initrd file of SUSE 11 SP4:

```
default 0
timeout 10
gfxmenu (hd0,0)/boot/message
title sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent console=ttyS0,115200n8 console=tty0
net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
initrd /boot/initrd.vmx
title Failsafe_sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent ide=nodma apm=off noresume edd=off
powersaved=off nohz=off highres=off processsor.max+cstate=1 nomodeset x11failsafe
console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1
showopts
initrd /boot/initrd.vmx
```

/boot/initrd.vmx in the **initrd** line is the initrd file actually used. Run the **dracut -f /boot/initrd.vmx** command. If the initrd file does not contain the **/boot** directory, such as **/initramfs-***xxx*, run the **dracut -f /boot/initramfs-***xxx* command.

d. Run the following commands to check whether Xen PVOPS and KVM VirtIO have been installed:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
xen-netfront.ko

SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

e. Restart the ECS.

f. Modify the **/boot/grub/menu.lst** file. Add **xen_platform_pci.dev_unplug=all** and modify the root configuration.

Before the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314
initrd /boot/initrd-3.0.76-0.11-default
```

After the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314 xen_platform_pci.dev_unplug=all
initrd /boot/initrd-3.0.76-0.11-default
```

NOTE

- Ensure that the root partition is in the UUID format.
- **xen_platform_pci.dev_unplug=all** is added to shield QEMU devices.
- For SUSE 11 SP1 64bit to SUSE 11 SP4 64bit, add **xen_platform_pci.dev_unplug=all** to the **menu.lst** file. For SUSE 12 or later, QEMU device shield is enabled by default, and you do not need to configure it.

g. Run the following commands to check whether Xen drivers exist in initrd:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/xen-netfront.ko

SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/virtio_net.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

- If the OS version is SUSE 12 SP1, perform the following steps:

a. Run the following command to open the **/etc/dracut.conf** file:

vi /etc/dracut.conf

b. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add-drivers** (the format varies depending on the OS).

```
[root@CTU10000xxxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
```

c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

d. Run the following command to regenerate initrd:

dracut -f /boot/initramfs-*File name*

If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

lsinitrd /boot/initramfs-`uname -r`.img | grep xen

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is initrd, run the following commands to check whether native Xen and KVM drivers have been installed:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

- If the OS version is later than SUSE 12 SP1 or openSUSE 13, perform the following steps:

  Take SUSE Linux Enterprise Server 12 SP2 (x86_64) as an example.

  a. Run the following command to open the **/etc/dracut.conf** file:

  vi /etc/dracut.conf

  b. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add_drivers** (the format varies depending on the OS).

  ```
  [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
  # additional kernel modules to the default
  add_drivers+="ata_piix ata_generic xen-blkfront xen-netfront virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
  ```

  c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

  d. Run the following command to regenerate initrd:

  dracut -f /boot/initramfs-*File name*

  If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

  e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

  lsinitrd /boot/initramfs-`uname -r`.img | grep xen

  lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

  If the virtual file system is initrd, run the following commands to check whether the native Xen and KVM drivers have been installed:

  lsinitrd /boot/initrd-`uname -r` | grep xen

  lsinitrd /boot/initrd-`uname -r` | grep virtio

  Assume that the virtual file system is initrd. The following command output will be displayed:

  ```
  sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
  -rw-r--r-- 1 root root 69575 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/xen-
  ```

```
blkfront.ko
-rw-r--r-- 1 root root 53415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/xen-netfront.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-hcall
-rwxr-xr-x 1 root root 8320 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-hcall/xen-hcall.ko

sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rw-r--r-- 1 root root 29335 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/virtio_blk.ko
-rw-r--r-- 1 root root 57007 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/virtio_net.ko
-rw-r--r-- 1 root root 32415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/scsi/virtio_scsi.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/kernel/drivers/virtio
-rw-r--r-- 1 root root 19623 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/virtio.ko
-rw-r--r-- 1 root root 38943 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/virtio_pci.ko
-rw-r--r-- 1 root root 24431 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/virtio_ring.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

3.9.6 Installing Native KVM Drivers

Scenarios

When optimizing a Linux private image, you need to install native KVM drivers on the ECS. If the drivers have been installed, skip this section.

CAUTION

If you do not install KVM drivers, NICs of the ECS may not be detected and the ECS cannot communicate with other resources.

Prerequisites

- The ECS needs to be optimized. For details, see **Checking Whether a Private Image Needs to be Optimized**.
- The ECS kernel must be later than 2.6.24.
- Disable your antivirus and intrusion detection software. You can enable the software after KVM drivers are installed.

Procedure

Modify the configuration file based on the OS version.

Table 3-1 Modifying configuration files for different OSs

OS	Configuration	Reference
CentOS/EulerOS	Take CentOS 7.0 as an example. 1. In the **/etc/dracut.conf** file, add VirtIO drivers to **add_drivers**, including virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. 2. Save and exit the **/etc/dracut.conf** file and run the **dracut -f** command to generate **initrd** again.	**CentOS and EulerOS**
Ubuntu/Debian	1. In the **/etc/initramfs-tools/modules** file, add VirtIO drivers, including virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. 2. Save and exit the **/etc/initramfs-tools/modules** file and run the **update-initramfs -u** command to generate **initrd** again.	**Ubuntu and Debian**
SUSE and openSUSE	If the OS version is earlier than SUSE 12 SP1 or openSUSE 13: 1. In the **/etc/sysconfig/kernel** file, add VirtIO drivers to **INITRD_MODULES=""**. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. 2. Run the **mkinitrd** command to generate **initrd** again.	**SUSE and openSUSE (Earlier than SUSE 12 SP1 or openSUSE 13)**
	If the OS version is SUSE 12 SP1: 1. In the **/etc/dracut.conf** file, add VirtIO drivers to **add_drivers**. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. 2. Run the **dracut -f** command to generate **initrd** again.	**SUSE and openSUSE (SUSE 12 SP1)**

OS	Configuration	Reference
	If the OS version is later than SUSE 12 SP1 or openSUSE 13: 1. In the **/etc/dracut.conf** file, add VirtIO drivers to **add_drivers**, including virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. 2. Save and exit the **/etc/dracut.conf** file and run the **dracut -f** command to generate **initrd** again.	**SUSE and openSUSE (Later than SUSE 12 SP1 or openSUSE 13)**

CentOS and EulerOS

1. Run the following command to open the **/etc/dracut.conf** file:

   vi /etc/dracut.conf

2. Press **i** to enter the editing mode and add VirtIO drivers to **add_drivers** (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
   # additional kernel modules to the default
   add_drivers+="virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
   ....
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

4. Run the following command to regenerate initrd:

   dracut -f */boot/initramfs-2.6.32-573.8.1.el6.x86_64.img*

   If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

5. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

   If the virtual file system is initrd, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   Assume that the virtual file system is initramfs. The following command output will be displayed:

   ```
   [root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
   -rwxr--r--   1 root     root        23448 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/block/virtio_blk.ko
   -rwxr--r--   1 root     root        50704 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/net/virtio_net.ko
   -rwxr--r--   1 root     root        28424 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/scsi/virtio_scsi.ko
   drwxr-xr-x   2 root     root            0 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio
   ```

```
-rwxr--r--  1 root  root  14544 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio.ko
-rwxr--r--  1 root  root  21040 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rwxr--r--  1 root  root  18016 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio_ring.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

Ubuntu and Debian

1. Run the following command to open the **modules** file:

   vi /etc/initramfs-tools/modules

2. Press **i** to enter the editing mode and add VirtIO drivers to the **/etc/initramfs-tools/modules** file (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]#vi /etc/initramfs-tools/modules
   ...
   # Examples:
   #
   # raid1
   # sd_mOd
   virtio_blk
   virtio_scsi
   virtio_net
   virtio_pci
   virtio_ring
   virtio
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/initramfs-tools/modules** file.

4. Run the following command to regenerate initrd:

   update-initramfs -u

5. Run the following command to check whether native KVM drivers have been installed:

   lsinitramfs /boot/initrd.img-`uname -r` |grep virtio

   ```
   [root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
   lib/modules/3.5.0-23-generic/kernel/drivers/scsi/virtio_scsi.ko
   ```

   NOTE

   If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

   ```
   [root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
   CONFIG_VIRTIO_BLK=y
   CONFIG_VIRTIO_NET=y
   CONFIG_VIRTIO=y
   CONFIG_VIRTIO_RING=y
   CONFIG_VIRTIO_PCI=y
   CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
   ```

SUSE and openSUSE (Earlier than SUSE 12 SP1 or openSUSE 13)

Modify the **/etc/sysconfig/kernel** file.

1. Run the following command to modify the **/etc/sysconfig/kernel** file:

   vi /etc/sysconfig/kernel

2. Add VirtIO drivers to **INITRD_MODULES=""** (the format of drivers depends on the OS).

   ```
   SIA10000xxxxx:~ # vi /etc/sysconfig/kernel
   # (like drivers for scsi-controllers, for lvm or reiserfs)
   #
   INITRD_MODULES="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
   ```

3. Run the **mkinitrd** command to generate **initrd** again.

   NOTE

   If the virtual file system is not the default initramfs or initrd, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **menu.lst** or **grub.cfg** file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, or **/boot/grub2/grub.cfg**).

   The following is an example initrd file of SUSE 11 SP4:

   ```
   default 0
   timeout 10
   gfxmenu (hd0,0)/boot/message
   title sles11sp4_001_[_VMX_]
   root (hd0,0)
   kernel /boot/linux.vmx vga=0x314 splash=silent console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
   initrd /boot/initrd.vmx
   title Failsafe_sles11sp4_001_[_VMX_]
   root (hd0,0)
   kernel /boot/linux.vmx vga=0x314 splash=silent ide=nodma apm=off noresume edd=off powersaved=off nohz=off highres=off processsor.max+cstate=1 nomodeset x11failsafe console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
   initrd /boot/initrd.vmx
   ```

   /boot/initrd.vmx in the **initrd** line is the **initrd** file actually used. Run the **dracut -f /boot/initrd.vmx** command. If the **initrd** file does not contain the **/boot** directory, such as **/initramfs-***xxx*, run the **dracut -f /boot/initramfs-***xxx* command.

4. Run the following command to check whether KVM VirtIO drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   ```
   SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
   -rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/virtio_scsi.ko
   -rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/virtio_blk.ko
   drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
   -rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_ring.ko
   -rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_pci.ko
   -rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio.ko
   -rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/virtio_net.ko
   ```

5. Restart the ECS.

6. Run the following command to check whether KVM drivers exist in initrd:

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   ```
   SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
   -rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/virtio_scsi.ko
   ```

```
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

SUSE and openSUSE (SUSE 12 SP1)

Modify the **/etc/dracut.conf** file.

1. Run the following command to open the **/etc/dracut.conf** file:

   vi /etc/dracut.conf

2. Press **i** to enter the editing mode and add VirtIO drivers to **add-drivers** (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
   # additional kernel modules to the default
   add_drivers+="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

4. Run the following command to regenerate initrd:

   dracut -f /boot/initramfs-*File name*

   If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

5. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

   If the virtual file system is initrd, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep virtio

SUSE and openSUSE (Later than SUSE 12 SP1 or openSUSE 13)

Modify the **/etc/dracut.conf** file.

Take SUSE Linux Enterprise Server 12 SP2 (x86_64) as an example.

1. Run the following command to open the **/etc/dracut.conf** file:

   vi /etc/dracut.conf

2. Press **i** to enter the editing mode and add VirtIO drivers to **add_drivers** (the format varies depending on the OS).

```
[root@CTU10000xxxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.
4. Run the following command to regenerate initrd:

   dracut -f /boot/initramfs-*File name*

   If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.
5. If the virtual file system is initramfs, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

   If the virtual file system is initrd, run the following command to check whether native KVM drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   Assume that the virtual file system is initrd. The following command output will be displayed:

```
sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rw-r--r-- 1 root root 29335 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/
virtio_blk.ko
-rw-r--r-- 1 root root 57007 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/
virtio_net.ko
-rw-r--r-- 1 root root 32415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/scsi/
virtio_scsi.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/kernel/drivers/virtio
-rw-r--r-- 1 root root 19623 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/virtio.ko
-rw-r--r-- 1 root root 38943 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_pci.ko
-rw-r--r-- 1 root root 24431 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_ring.ko
```

   NOTE

   If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following command to check whether the drivers are built-in ones in the kernel:

   cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

3.9.7 Clearing System Logs

Delete log files and historical records, and stop the ECS.

1. Run the following commands to delete redundant key files:

   echo > /$path/$to/$root/.ssh/authorized_keys

   An example command is **echo > /root/.ssh/authorized_keys**.

   echo > /$path/$to/$none-root/.ssh/authorized_keys

   An example command is **echo > /home/linux/.ssh/authorized_keys**.

2. Run the following command to clear log files in the **/var/log** directory:

   rm -rf /var/log/*

   NOTE

   Before deleting log files, back up log directories and log files required by application startup. For example, if the default Nginx log directory **/var/log/nginx** is deleted, Nginx may fail to be started.

3. Run the following commands to delete historical records:

   echo > /root/.bash_history

   history -c

3.10 Encrypting Images

3.10.1 Overview

IMS allows you to create encrypted images to ensure data security.

NOTE

To use the image encryption function, you must apply for KMS Administrator permissions.

Constraints

- DEW must be enabled.
- Encrypted images cannot be shared with others.
- The system disk of an ECS created from an encrypted image is also encrypted, and its key is the same as the image key.
- If an ECS has an encrypted system disk, private images created from the ECS are also encrypted.
- The key used for encrypting an image cannot be changed.
- If the key used for encrypting an image is disabled or deleted, the image is unavailable.

3.10.2 Creating Encrypted Images

You can create an encrypted image using an external image file or an encrypted ECS.

- Create an encrypted image using an external image file.

  When you register the external image file as a private image, select **KMS encryption** and select a key. For details, see **Creating a Windows System Disk Image from an External Image File** and **Creating a Linux System Disk Image from an External Image File**.

- Create an encrypted image using an encrypted ECS.

  When you use an ECS to create a private image, if the system disk of the ECS is encrypted, the private image created using the ECS is also encrypted. The key used for encrypting the image must be the same as that used for encrypting the system disk. For details, see **Creating a System Disk Image from a Windows ECS** and **Creating a System Disk Image from a Linux ECS**.

3.11 Replicating Images

Scenarios

You can convert encrypted and unencrypted images into each other or enable some advanced features (such as fast ECS creation from an image) using the in-region image replication function. You may need to replicate an image to:

- Replicate an encrypted image to an unencrypted one.

  Encrypted images cannot be shared. If you want to share an encrypted image, you can replicate it to an unencrypted one.
- Replicate an encrypted image to an encrypted one.

  Keys for encrypting the images cannot be changed. If you want to change the key of an encrypted image, you can replicate this image to a new one and encrypt the new image using an encryption key.
- Replicate an unencrypted image to an encrypted one.

  If you want to store an unencrypted image in an encrypted way, you can replicate this image as a new one and encrypt the new image using a key.
- Optimize a system disk image so that it can be used to quickly create ECSs.

  Fast Create greatly reduces the time required for creating ECSs from a system disk image. Currently, this feature is supported by all newly created system disk images by default. Existing system disk images may not support this function. You can optimize the images using the in-region image replication function. For example, if image A does not support fast ECS creation, you can replicate it to generate image copy_A that supports fast ECS creation.

Constraints

- Full-ECS images cannot be replicated within the same region.
- Private images created using ISO files do not support in-region replication.

Prerequisites

The images to be replicated are in the **Normal** state.

Procedure

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. Locate the row that contains the image to be replicated, click **More** in the **Operation** column, and select **Replicate**.
3. In the displayed **Replicate Image** dialog box, set the following parameters:
   - **Name**: Enter a name that is easy to identify.
   - **Enterprise Project**: Select an enterprise project from the drop-down list. This parameter is available only if you have enabled enterprise projects or

your account is an enterprise account. To enable this function, contact your customer manager.

- **Description**: This parameter is optional. Enter description of the replication.
- **Encryption**: If you want to encrypt the image or change a key, select **KMS encryption** and select the key you want to use from the drop-down list.

4. Click **OK**.

   On the **Private Images** page, view the replication progress. If the status of the new image becomes **Normal**, the image replication is successful.

3.12 Tagging an Image

Scenarios

You can use tags to classify images. You can add, modify, or delete image tags, or search for required images by tag in the image list.

NOTE

- When adding predefined tags to an image or searching for an image using predefined tags, you must have permission to access the Tag Management Service (TMS).

Constraints

An image can have a maximum of 10 tags.

Add, Delete, and Modify Image Tags

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.

      The IMS console is displayed.
2. Click the **Private Images** tab and click the image name to display the image details.
   - To modify an image tag, go to **3**.
   - To delete an image tag, go to **4**.
   - To add an image tag, go to **5**.
3. Click the **Tags** tab, locate the target tag, and click **Edit** in the **Operation** column. In the displayed dialog box, modify the tag.
4. Click the **Tags** tab, locate the target tag, and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.
5. Click the **Tags** tab and then **Add Tag**. In the displayed dialog box, add a tag.

Search for Private Images by Tag

1. Access the IMS console.

a. Log in to the management console.
b. Under **Computing**, click **Image Management Service**.
   The IMS console is displayed.
2. Click the **Private Images** tab and then **Search by Tag**.
3. Enter the tag key and value.
   Neither the tag key nor tag value can be empty. When the tag key and tag value are matched, the system automatically shows your desired private images.
4. Click  to add a tag.
   You can add multiple tags to search for private images. The system will display private images that match all tags.
5. Click **Search**.
   The system searches for private images based on tag keys or tag values.

3.13 Auditing Key Operations

3.13.1 IMS Operations Recorded by CTS

Scenarios

Cloud Trace Service (CTS) is a log audit service provided by the public cloud and intended for cloud security. It allows you to collect, store, and query cloud resource operation records and use these records for security analysis, compliance auditing, resource tracking, and fault locating.

You can use CTS to record IMS operations for later querying, auditing, and backtracking.

Prerequisites

You need to enable CTS before using it. If it is not enabled, IMS operations cannot be recorded. After being enabled, CTS automatically creates a tracker to record all your operations. The tracker stores only the operations of the last seven days. To store the operations for a longer time, store trace files in OBS buckets.

IMS Operations Recorded by CTS

Table 3-2 IMS operations that can be recorded by CTS

Operation	Resource Type	Trace Name
Creating an Image	ims	createImage
Modifying an image	ims	updateImage
Deleting images in a batch	ims	deleteImage

Operation	Resource Type	Trace Name
Replicating an image	ims	copyImage
Exporting an image	ims	exportImage
Adding a tenant that can use a shared image	ims	addMember
Modifying tenants that can use a shared image	ims	updateMember
Deleting tenants from the group where the members can use a shared image	ims	deleteMemeber

Table 3-3 Relationship between IMS operations and native OpenStack APIs

Operation	Trace Name	Service Type	Resource Type	OpenStack Component
Creating an Image	createImage	IMS	image	glance
Modifying/ Uploading an image	updateImage	IMS	image	glance
Deleting an image	deleteImage	IMS	image	glance
Tagging an image	addTag	IMS	image	glance
Deleting an image tag	deleteTag	IMS	image	glance
Adding a tenant that can use a shared image	addMember	IMS	image	glance
Modifying information about a tenant that can use a shared image	updateMember	IMS	image	glance

Operation	Trace Name	Service Type	Resource Type	OpenStack Component
Deleting a tenant from the group where the members can use a shared image	deleteMember	IMS	image	glance

3.13.2 Viewing Traces

Scenarios

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, the system starts recording operations on data in OBS buckets. CTS stores operation records generated in the last seven days.

This section describes how to query and export operation records of the last seven days on the CTS console.

- **Viewing Real-Time Traces in the Trace List of the New Edition**
- **Viewing Real-Time Traces in the Trace List of the Old Edition**

Viewing Real-Time Traces in the Trace List of the New Edition

1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. On the **Trace List** page, use advanced search to query traces. You can combine one or more filters.
   - **Trace Name**: Enter a trace name.
   - **Trace ID**: Enter a trace ID.
   - **Resource Name**: Enter a resource name. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.
   - **Resource ID**: Enter a resource ID. Leave this field empty if the resource has no resource ID or if resource creation failed.
   - **Trace Source**: Select a cloud service name from the drop-down list.
   - **Resource Type**: Select a resource type from the drop-down list.
   - **Operator**: Select one or more operators from the drop-down list.
   - **Trace Status**: Select **normal**, **warning**, or **incident**.

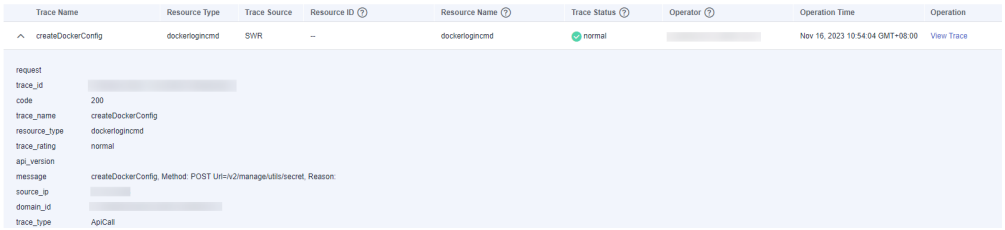
    - **normal**: The operation succeeded.
    - **warning**: The operation failed.
    - **incident**: The operation caused a fault that is more serious than the operation failure, for example, causing other faults.
  - Time range: Select **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
5. On the **Trace List** page, you can also export and refresh the trace list, and customize the list display settings.
  - Enter any keyword in the search box and press Enter to filter desired traces.
  - Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5000 records.
  - Click  to view the latest information about traces.
  - Click  to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled (), excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.
6. For details about key fields in the trace structure, see section "Trace References" > "Trace Structure" and section "Trace References" > "Example Traces".
7. (Optional) On the **Trace List** page of the new edition, click **Go to Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

Viewing Real-Time Traces in the Trace List of the Old Edition

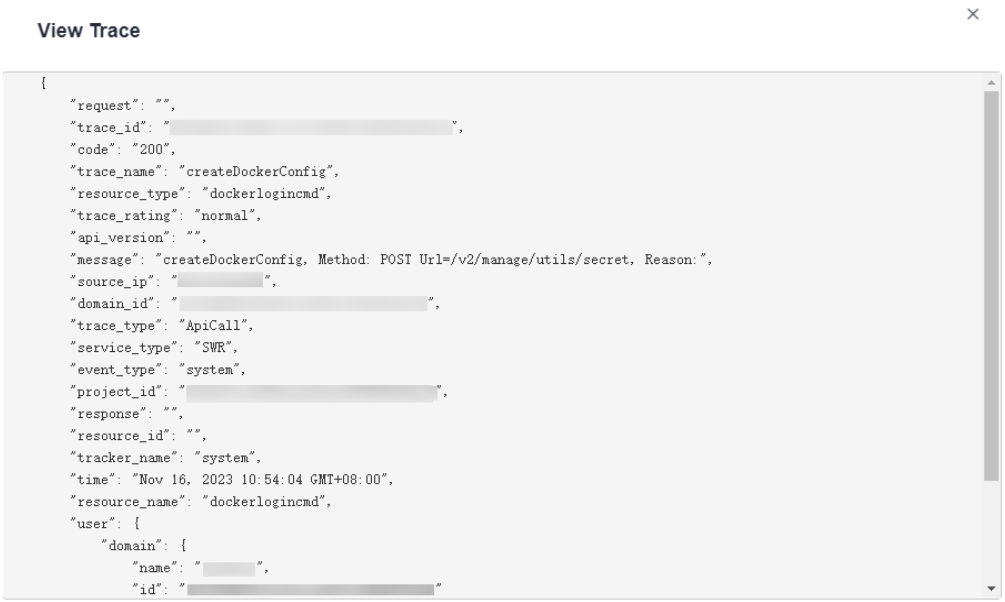
1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. Each time you log in to the CTS console, the new edition is displayed by default. Click **Go to Old Edition** in the upper right corner to switch to the trace list of the old edition.
5. Set filters to search for your desired traces. The following filters are available:
  - **Trace Type**, **Trace Source**, **Resource Type**, and **Search By**: Select a filter from the drop-down list.
    - If you select **Resource ID** for **Search By**, specify a resource ID.
    - If you select **Trace name** for **Search By**, specify a trace name.
    - If you select **Resource name** for **Search By**, specify a resource name.
  - **Operator**: Select a user.
  - **Trace Status**: Select **All trace statuses**, **Normal**, **Warning**, or **Incident**.
  - Time range: You can query traces generated during any time range in the last seven days.

- Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.

6. Click **Query**.
7. On the **Trace List** page, you can also export and refresh the trace list.
   - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
   - Click to view the latest information about traces.
8. Click on the left of a trace to expand its details.



9. Click **View Trace** in the **Operation** column. The trace details are displayed.



10. For details about key fields in the trace structure, see section "Trace References" > "Trace Structure" and section "Trace References" > "Example Traces" in the *CTS User Guide*.
11. (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

3.14 Converting the Image Format

Scenarios

You can import an image file in VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format to the cloud platform. Image files in other formats need to be converted before being imported. The open-source tool **qemu-img** is provided for you to convert image file formats.

Description

This section describes how to convert an image format on a local Windows or Linux PC.

Tool and Costs

Table 3-4 Tool and costs

Tool	Description	Costs
qemu-img	**qemu-img** is an open-source tool for converting image formats. You can obtain it from: **https://qemu.weilnetz.de/w64/**	Free

Constraints

- **qemu-img** supports the mutual conversion of image formats VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED.
- ZVHD and ZVHD2 are self-developed image file formats and cannot be identified by **qemu-img**.
- When you run a command to convert the format of VHD image files, use VPC to replace VHD. Otherwise, qemu-img cannot identify the image format.

  For example, to convert a CentOS 6.9 image file from VHD to QCOW2, run the following command:

  qemu-img convert -p -f vpc -O qcow2 centos6.9.vhd centos6.9.qcow2

Windows

1. Install qemu-img.
   a. Download the qemu-img installation package from **https://qemu.weilnetz.de/w64/**.
   b. Double-click the setup file to install qemu-img in **D:\Program Files\qemu** (an example installation path).
2. Configure environment variables.
   a. Choose **Start** > **Computer** and right-click **Properties**.
   b. Click **Advanced system settings**.
   c. In the **System Properties** dialog box, click **Advanced** > **Environment Variables**.
   d. In the **Environment Variables** dialog box, search for **Path** in the **System Variable** area and click **Edit**. Add **D:\Program Files\qemu** to **Variable Value**. Use semicolons (;) to separate variable values.

      NOTE

      If **Path** does not exist, add it and set its value to **D:\Program Files\qemu**.

e. Click **OK**.

3. Verify the installation.

   Choose **Start** > **Run**, enter **cmd**, and press **Enter**. In the **cmd** window, enter **qemu-img --help**. If the qemu-img version information is contained in the command output, the installation is successful.

4. Convert the image format.

   a. In the **cmd** window, run the following commands to switch to **D:\Program Files\qemu**:

      d:

      cd D:\Program Files\qemu

   b. Run the following command to convert the image file format from VMDK to QCOW2:

      qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2

      The parameters are described as follows:

      - **-p** indicates the image conversion progress.
      - **-f** indicates the source image format.
      - The part following **-O** (which must be in upper case) consists of the required format, source image file, and target image file.

      After the conversion is complete, the target image file is displayed in the directory where the source image file is located.

      The following information is displayed:

      ```
      # qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2
        (100.00/100%)
      ```

   c. Run the following command to query details about the converted image file in QCOW2 format:

      qemu-img info centos6.9.qcow2

      The following information is displayed:

      ```
      # qemu-img info centos6.9.qcow2
      image: centos6.9.qcow2
      file format: qcow2
      virtual size: 1.0G (1073741824 bytes)
      disk size: 200K
      cluster_size: 65536
      Format specific information:
          compat: 1.1
          lazy refcounts: false
      ```

Linux

1. Install qemu-img.

   – For Ubuntu or Debian, run the following command:

     apt install qemu-img

   – For CentOS, Red Hat, or Oracle, run the following command:

     yum install qemu-img

   – For SUSE or openSUSE, run the following command:

zypper install qemu-img

2. Run the following command to check whether the installation is successful:

   qemu-img -v

   If the version information and help manual of the qemu-img tool are contained in the command output, the installation is successful. If CentOS 7 is used, the command output is as follows:

```
[root@CentOS7 ~]# qemu-img -v
qemu-img version 1.5.3, Copyright (c) 2004-2008 Fabrice Bellard
usage: qemu-img command [command options]
QEMU disk image utility

Command syntax:
  check [-q] [-f fmt] [--output=ofmt] [-r [leaks | all]] [-T src_cache] filename
  create [-q] [-f fmt] [-o options] filename [size]
  commit [-q] [-f fmt] [-t cache] filename
  compare [-f fmt] [-F fmt] [-T src_cach]
```

3. Convert the image format. For example, perform the following steps to convert a VMDK image file running CentOS 7 to a QCOW2 image file:

   a. Run the following command to convert the image file format to QCOW2:

      qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2

      The parameters are described as follows:

      - **-p**: indicates the conversion progress.
      - **-f** indicates the source image format.
      - The part following **-O** (which must be in upper case) is the converted image format + source image file name + target image file name.

      After the conversion is complete, the target image file is displayed in the directory where the source image file is located.

      The following information is displayed:

```
[root@CentOS7 home]# qemu-img convert -p -f vmdk -O qcow2 centos6.9.vmdk centos6.9.qcow2
    (100.00/100%)
```

   b. Run the following command to query details about the converted image file in QCOW2 format:

      qemu-img info centos6.9.qcow2

      The following information is displayed:

```
[root@CentOS7 home]# qemu-img info centos6.9.qcow2
image: centos6.9.qcow2
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 200K
cluster_size: 65536
Format specific information:
    compat: 1.1
    lazy refcounts: false
```

Examples

- Scenario

  A pre-allocated image depends on two files: *xxxx*.**vmdk** (configuration file) and *xxxx*-**flat.vmdk** (data file) and cannot be directly imported to the cloud

platform. When you export a pre-allocated image file in VMDK monolithic Flat format from the VMware platform, you must convert its format to common VMDK or QCOW2 before it can be imported to the cloud platform.

The following uses the image files **centos6.9-64bit-flat.vmdk** and **centos6.9-64bit.vmdk** as an example to describe how to use qemu-img to convert image formats.

- Procedure

1. Run the following commands to query the image file details:

   ls -lh centos6.9-64bit*

   qemu-img info centos6.9-64bit.vmdk

   qemu-img info centos6.9-64bit-flat.vmdk

   The following information is displayed:

```
[root@CentOS7 tmp]# ls -lh centos6.9-64bit*
-rw-r--r--. 1 root root 10G Jun 13 05:30 centos6.9-64bit-flat.vmdk
-rw-r--r--. 1 root root 327 Jun 13 05:30 centos6.9-64bit.vmdk
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit.vmdk
image: centos6.9-64bit.vmdk
file format: vmdk
virtual size: 10G (10737418240 bytes)
disk size: 4.0K
Format specific information:
    cid: 3302005459
    parent cid: 4294967295
    create type: monolithicFlat
    extents:
        [0]:
            virtual size: 10737418240
            filename: centos6.9-64bit-flat.vmdk
            format: FLAT
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit-flat.vmdk
image: centos6.9-64bit-flat.vmdk
file format: raw
virtual size: 10G (10737418240 bytes)
disk size: 0
```

   NOTE

   The command output shows that the format of **centos6.9-64bit.vmdk** is VMDK and that of **centos6.9-64bit-flat.vmdk** is RAW. You can convert the format of only **centos6.9-64bit.vmdk**. For details about how to convert it, see **3**.

2. Run the following command to query the configuration of the pre-allocated image file:

   cat centos6.9-64bit.vmdk

   The following information is displayed:

```
[root@CentOS7 tmp]# cat centos6.9-64bit.vmdk
# Disk DescriptorFile
version=1
CID=c4d09ad3
parentCID=ffffffff
createType="monolithicFlat"

# Extent description
RW 20971520 FLAT "centos6.9-64bit-flat.vmdk" 0

# The Disk Data Base
#DDB

ddb.virtualHWVersion = "4"
ddb.geometry.cylinders = "20805"
```

```
ddb.geometry.heads = "16"
ddb.geometry.sectors = "63"
ddb.adapterType = "ide"
```

3. Place **centos6.9-64bit-flat.vmdk** and **centos6.9-64bit.vmdk** in the same directory. Run the following command to convert the format of **centos6.9-64bit.vmdk** to QCOW2 using qemu-img:

```
[root@CentOS7 tmp]# qemu-img convert -p -f vmdk -O qcow2 centos6.9-64bit.vmdk centos6.9-64bit.qcow2
    (100.00/100%)
```

4. Run the following command to query details about the converted image file in QCOW2 format:

   qemu-img info centos6.9-64bit.qcow2

   The following information is displayed:

```
[root@CentOS7 tmp]# qemu-img info centos6.9-64bit.qcow2
image: centos6.9-64bit.qcow2
file format: qcow2
virtual size: 10G (10737418240 bytes)
disk size: 200K
cluster_size: 65536
Format specific information:
    compat: 1.1
    lazy refcounts: false
```

4 Windows Operations

4.1 Setting the NIC to DHCP

Scenarios

If a private image is created from an ECS or external image file and the VM where the ECS or external image file is located is configured with a static IP address, you need to change the NIC attribute to DHCP so that the new ECSs created from the private image can dynamically obtain an IP address.

This section uses Windows Server 2008 R2 as an example to describe how to configure DHCP. For details about how to configure DHCP on ECSs running other OSs, see the relevant OS documentation.

NOTE

When registering an external image file as a private image, configure DHCP on the VM where the external image file is located. You are advised to configure DHCP on the VM and then export the image file.

Prerequisites

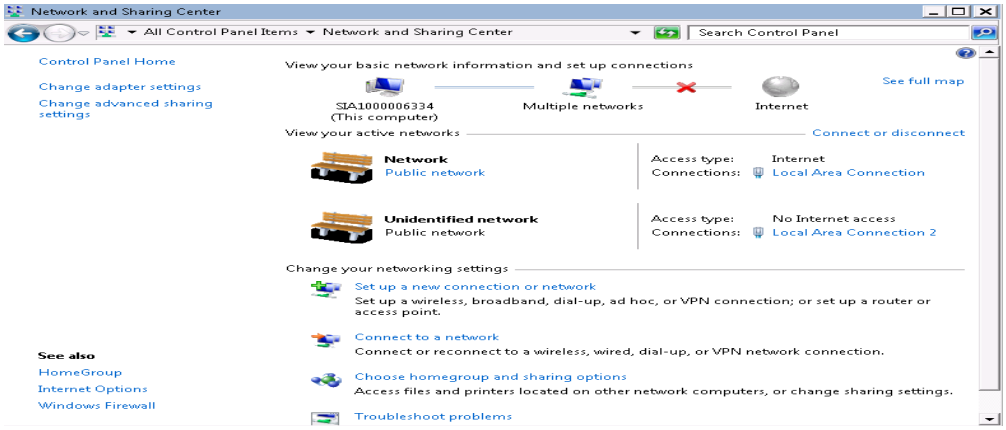
You have logged in to the ECS used to create a Windows private image.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Procedure

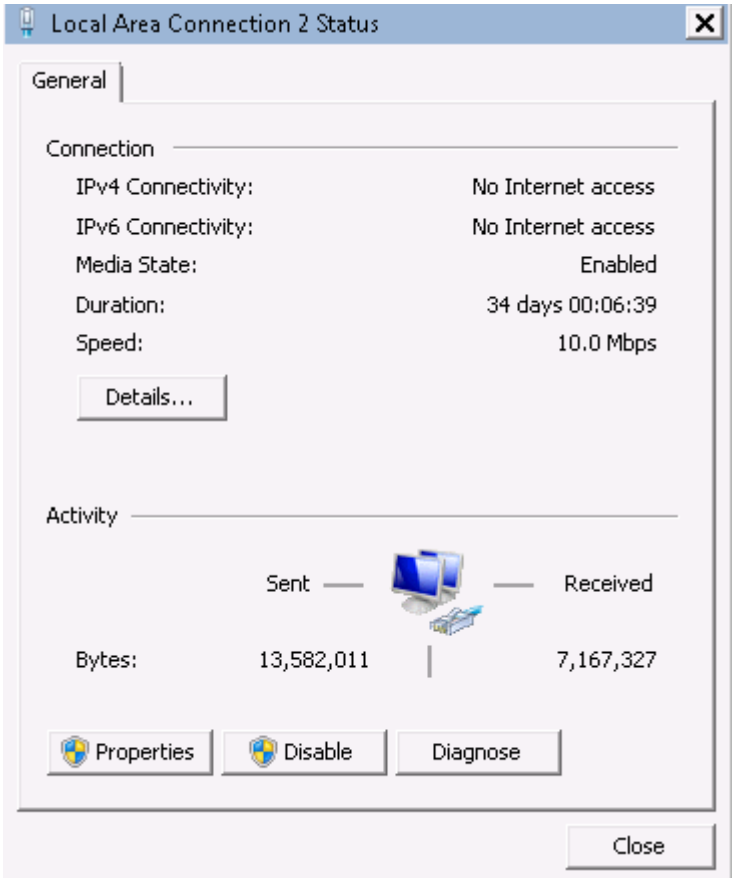
1. On the ECS, choose **Start** > **Control Panel**.
2. Click **Network and Internet Connections**.
3. Click **Network and Sharing Center**.

Figure 4-1 Network and Sharing Center



4. Select the connection configured with the static IP address. For example, click **Local Area Connection 2**.

Figure 4-2 Local Area Connection 2 Status

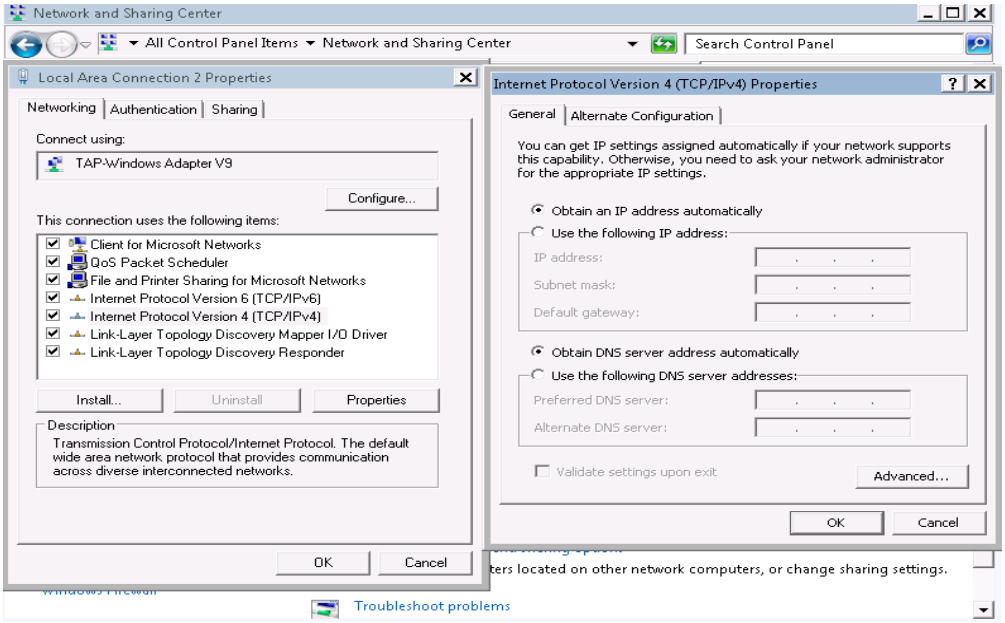


5. Click **Properties** and select the configured Internet protocol version.
6. On the **General** tab, select **Obtain an IP address automatically** and **Obtain DNS server address automatically** and click **OK**. Figure 4-3 shows the dialog box for configuring the IP address obtaining mode.

NOTE

You are advised to record the original network information so that you can restore the network if necessary.

Figure 4-3 Configuring the IP address obtaining mode



4.2 Enabling Remote Desktop Connection

Scenarios

If you want to remotely access an ECS, enable remote desktop connection for the source ECS when creating a private image. This function must be enabled for GPU-accelerated ECSs.

NOTE

When registering an external image file as a private image, enable remote desktop connection on the VM where the external image file is located. You are advised to enable this function on the VM and then export the image file.

Prerequisites

You have logged in to the ECS used to create a Windows private image.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Procedure

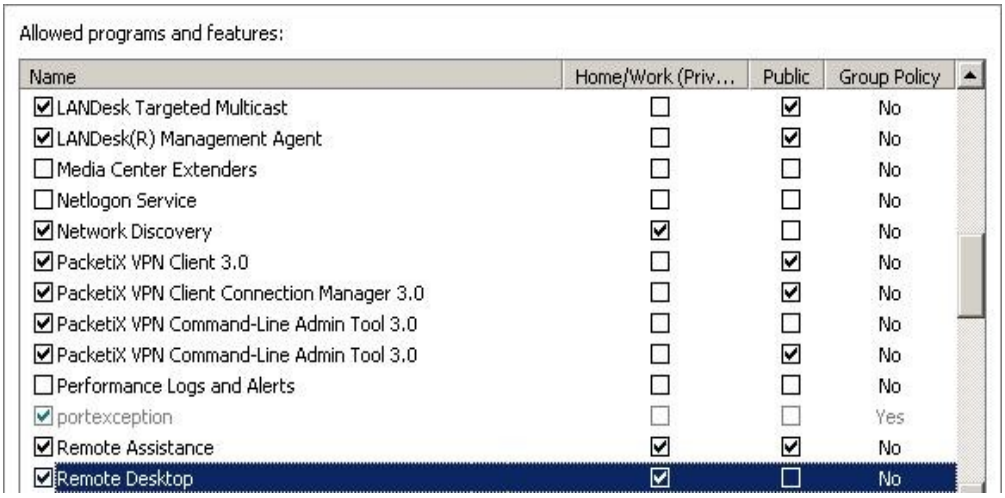
1. Before enabling this function, you are advised to set the resolution of the ECS to 1920×1080.

   On the ECS, choose **Start** > **Control Panel**. Under **Appearance and Personalization**, click **Adjust screen resolution**. Then select a proper value from the **Resolution** drop-down list box.

2. Choose **Start**, right-click **Computer**, and choose **Properties** from the shortcut menu.

3. Click **Remote settings**.
4. In the **Remote** tab, select **Allow connections from computers running any version of Remote Desktop (less secure)**.
5. Click **OK**.
6. Choose **Start** > **Control Panel** and navigate to **Windows Firewall**.
7. Choose **Allow a program or feature through Windows Firewall** in the left pane.
8. Select programs and features that are allowed by the Windows firewall for **Remote Desktop** based on your network requirements and click **OK** in the lower part.

Figure 4-4 Configuring remote desktop



4.3 Installing and Configuring Cloudbase-Init

Scenarios

To ensure that you can use the user data injection function to inject initial custom information into ECSs created from a private image (such as setting the ECS login password), install Cloudbase-Init on the ECS used to create the image.

- If Cloudbase-Init is not installed, you cannot configure an ECS. As a result, you can only use the password in the image file to log in to the ECS.
- By default, ECSs created from a public image have Cloudbase-Init installed. You do not need to install or configure Cloudbase-Init on such ECSs.
- For ECSs created from external image files, install and configure Cloudbase-Init by performing the operations in this section.

NOTE

Cloudbase-Init is open-source software. If the installed version has security vulnerabilities, you are advised to upgrade it to the latest version.

Prerequisites

- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Install Cloudbase-Init

1. On the Windows **Start** menu, choose **Control Panel** > **Programs** > **Programs and Features** and check whether Cloudbase-Init 1.1.2 is installed.
   - If Cloudbase-Init 1.1.2 is installed, skip the subsequent steps and go to **Configure Cloudbase-Init**.
   - If Cloudbase-Init is installed but the version is not 1.1.2, uninstall Cloudbase-Init and go to the next step.
   - If Cloudbase-Init is not installed, go to the next step.
2. Check whether the version of the OS is Windows desktop.
   - If yes, go to **3**.
   - If the OS is Windows Server, go to **4**.
3. Enable the administrator account (Windows 7 is used as an example).
   a. Click **Start** and choose **Control Panel** > **System and Security** > **Administrative Tools**.
   b. Double-click **Computer Management**.
   c. Choose **System Tools** > **Local Users and Groups** > **Users**.
   d. Right-click **Administrator** and select **Properties**.
   e. Deselect **Account is disabled**.
4. Download the Cloudbase-Init installation package.

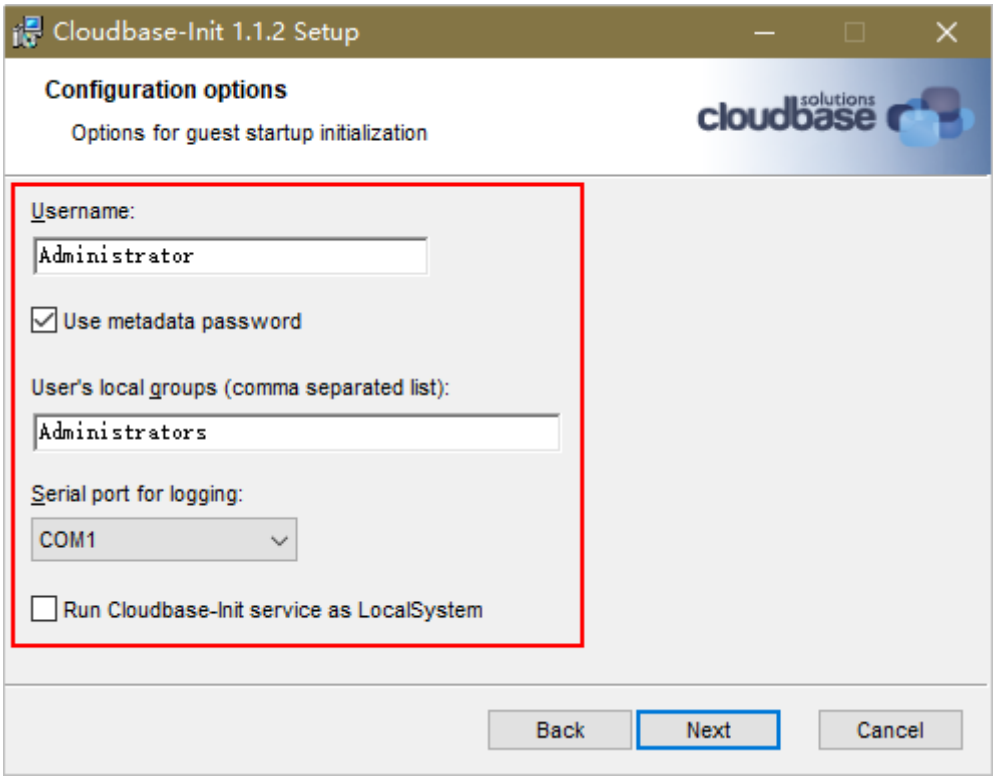
   Download the Cloudbase-Init installation package of the appropriate version based on the OS architecture from the Cloudbase-Init official website (**http://www.cloudbase.it/cloud-init-for-windows-instances/**).

   To obtain the stable version, visit the following paths:
   - 64-bit: **https://www.cloudbase.it/downloads/CloudbaseInitSetup_Stable_x64.msi**
   - 32-bit: **https://www.cloudbase.it/downloads/CloudbaseInitSetup_Stable_x86.msi**
5. Double-click the Cloudbase-Init installation package.
6. Click **Next**.
7. Select **I accept the terms in the License Agreement** and click **Next**.
8. Retain the default path and click **Next**.
9. In the **Configuration options** window, enter **Administrator** for **Username**, select **COM1** for **Serial port for logging**, and ensure that **Run Cloudbase-Init service as LocalSystem** is not selected.

   NOTE

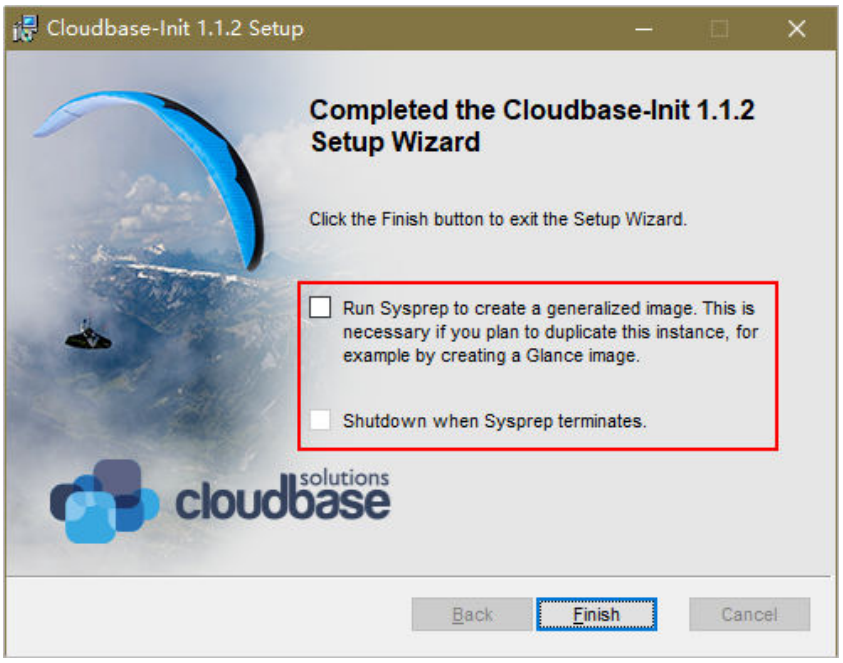
   The version number shown in the figure is for reference only.

Figure 4-5 Configuring parameters



10. Click **Next**.
11. Click **Install**.
12. In the **Files in Use** dialog box, select **Close the application and attempt to restart them** and click **OK**.
13. Check whether the version of the OS is Windows desktop.
    - If yes, go to **15**.
    - If no, go to **14**.
14. In the **Completed the Cloudbase-Init Setup Wizard** window, ensure that neither option is selected.

Figure 4-6 Completing the Cloudbase-Init installation



NOTE

The version number shown in the figure is for reference only.

15. Click **Finish**.

Configure Cloudbase-Init

1. Edit the configuration file **C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init.conf** in the Cloudbase-Init installation path.
   a. Add **netbios_host_name_compatibility=false** to the last line of the file so that the hostname supports a maximum of 63 characters.

      NOTE

      NetBIOS contains no more than 15 characters due to Windows system restrictions.

   b. Add **metadata_services=cloudbaseinit.metadata.services.httpservice.HttpService** to enable the agent to access the IaaS OpenStack data source.
   c. Add **plugins** to configure the plugins that will be loaded. Separate different plugins with commas (,). The information in bold is the keyword of each plugin.

      - The following plugins are loaded by default. You can keep all or some of them as needed.

        plugins=cloudbaseinit.plugins.common.localscripts.**LocalScriptsPlugin**,cloudbaseinit.plugins.common.mtu.**MTUPlugin**,cloudbaseinit.plugins.windows.createuser.**CreateUserPlugin**,cloudbaseinit.plugins.common.setuserpassword.**SetUserPasswordPlugin**,cloudbaseinit.plugins.common.sshpublickeys.**SetUserSSHPublicKeysPlugin**,cloudbaseinit.plugins.common.sethostname.**SetHostNamePlugin**,cloudbaseinit.plugins.windows.extendvolumes.**ExtendVolumesPlugin**,cloudbaseinit.plugins.common.userdata.**UserDataPlugin**,cloudbaseinit.plugins.windows.licensing.**WindowsLicensingPlugin**

        Plugin functions:

- **LocalScriptsPlugin** configures scripts.
- **MTUPlugin** configures MTU network interfaces.
- **CreateUserPlugin** creates a user.
- **SetUserPasswordPlugin** configures a password.
- **SetUserSSHPublicKeysPlugin** configures a key.
- **SetHostNamePlugin** configures a hostname.
- **ExtendVolumesPlugin** expands disk space.
- **UserDataPlugin** injects user data.
- **WindowsLicensingPlugin** activates Windows instances.

NOTE

If you may change the hostname of ECSs after they are created from this image and services on the ECSs are sensitive to hostname changes, you are not advised to configure the **SetHostNamePlugin** here.

- Optional plugins:
  plugins=cloudbaseinit.plugins.windows.winrmlistener.**ConfigWinRMListenerPlugin**,cloudbaseinit.plugins.windows.winrmcertificateauth.**ConfigWinRMCertificateAuthPlugin**

  Plugin functions:
  - **ConfigWinRMListenerPlugin** configures listening to remote logins.
  - **ConfigWinRMCertificateAuthPlugin** configures remote logins without password authentication.

  CAUTION

  The WinRM plug-ins use weak cryptographic algorithm, which may cause security risks. So, you are advised not to load the plug-ins.

d. (Optional) Add the following configuration items to configure the number of retry times and interval for obtaining metadata:

```
retry_count=40
retry_count_interval=5
```

e. (Optional) Add the following configuration item to prevent metadata network disconnections caused by the default route added by Windows:

```
[openstack]
add_metadata_private_ip_route=False
```

f. (Optional) If the Cloudbase-Init version is 0.9.12 or later, you can customize the length of the password.

Change the value of **user_password_length** to customize the password length.

g. (Optional) Add the following configuration item to ensure that time synchronization from BIOS persists through system restarts:

real_time_clock_utc=true

NOTE

The registry entry **RealTimeIsUniversal=1** allows the system to synchronize time from BIOS. If **real_time_clock_utc=true** is not configured, Cloudbase-Init will revert **RealTimeIsUniversal** back to **0**. As a result, the system cannot synchronize time from BIOS after a restart.

2. Release the current DHCP address so that the created ECSs can obtain correct addresses.

   In the Windows command line, run the following command to release the current DHCP address:

   ipconfig /release

   NOTE

   This operation will interrupt network connection and adversely affect ECS use. The network will automatically recover after the ECSs are started again.

3. When creating an image using a Windows ECS, you need to change the SAN policy of the ECS to **OnlineAll**. Otherwise, EVS disks attached to the ECSs created from the image may be offline.

   Windows has three types of SAN policies: **OnlineAll**, **OfflineShared**, and **OfflineInternal**.

   Table 4-1 SAN policies

Type	Description
OnlineAll	All newly detected disks are automatically brought online.
OfflineShared	All disks on sharable buses, such as iSCSI and FC, are left offline by default, while disks on non-sharable buses are kept online.
OfflineInternal	All newly detected disks are left offline.

   a. Execute **cmd.exe** and run the following command to query the current SAN policy of the ECS using DiskPart:

      diskpart

   b. Run the following command to view the SAN policy of the ECS:

      san

      - If the SAN policy is **OnlineAll**, run the **exit** command to exit DiskPart.
      - If the SAN policy is not **OnlineAll**, go to **3.c**.

   c. Run the following command to change the SAN policy of the ECS to **OnlineAll**:

      san policy=onlineall

4.4 Running Sysprep

Scenarios

Running Sysprep ensures that an ECS has a unique SID after it is added to a domain.

After installing Cloudbase-Init on an ECS, you need to decide whether the ECS needs to be added to a domain or whether it must have a unique SID. If yes, run Sysprep as instructed in this section.

Prerequisites

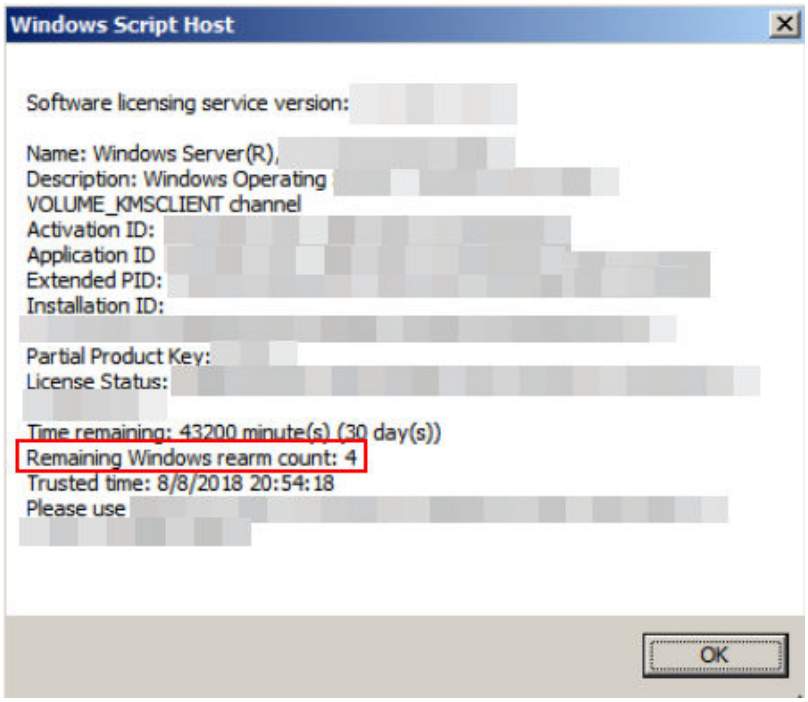
- Run Sysprep as the administrator.
- For a newly activated Windows ECS, you can run Sysprep only once at a time.
- If an ECS is created from an image file, only Sysprep provided by the image file can be used. In addition, Sysprep must always reside in the **%WINDIR%\system32\sysprep** directory.
- Windows must be in the activated state, and the remaining Windows rearm count must be greater than or equal to 1. Otherwise, the Sysprep encapsulation cannot be executed.

  Run the following command in the Windows command line and check how many times you can run Sysprep in the displayed **Windows Script Host** dialog box:

  slmgr.vbs /dlv

  If the value of **Remaining Windows rearm count** is **0**, you cannot run Sysprep.

  Figure 4-7 Windows Script Host



Procedure

1. Enter the Cloudbase-Init installation directory.

   C:\Program Files\Cloudbase Solutions is used as an example of the Cloudbase-Init installation directory. Switch to the root directory of drive C and run the following command to enter the installation directory:

   cd C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf

2. Run the following command to encapsulate Windows:

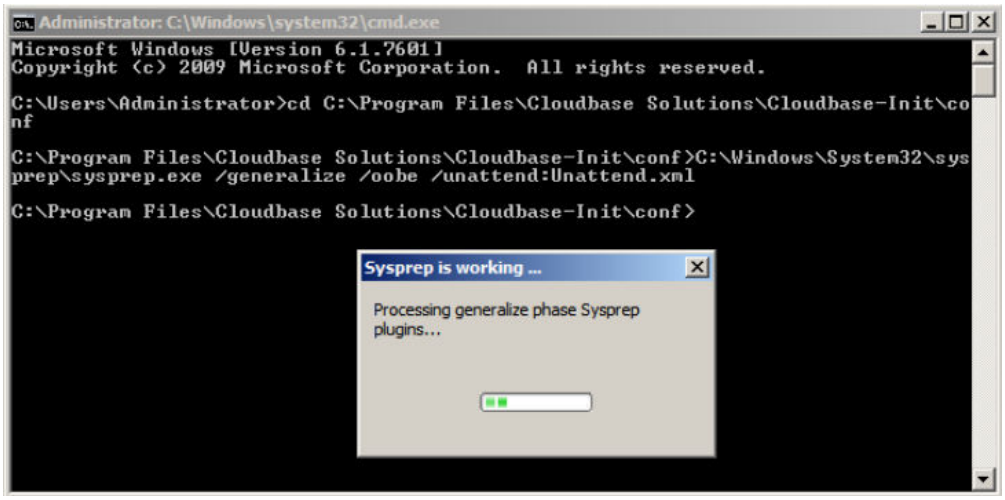
   C:\Windows\System32\sysprep\sysprep.exe /generalize /oobe /unattend:Unattend.xml

---

CAUTION

- Ensure that **/unattend:Unattend.xml** is contained in the preceding command. Otherwise, the username, password, and other important configuration information of the ECS will be reset, and you must configure the OS manually when you use ECSs created from the Windows private image.
- After this command is executed, the ECS will be automatically stopped. After the ECS is stopped, use the ECS to create an image. ECSs created using the image have unique SIDs. If you restart a Windows ECS on which Sysprep has been executed, Sysprep takes effect only for the current ECS. Before creating an image using the ECS, you must run Sysprep again.
- For Windows Server 2012 and Windows Server 2012 R2, the administrator password of the ECS will be deleted after Sysprep is executed on the ECS. You need to log in to the ECS and reset the administrator password. In this case, the administrator password set on the management console will be invalid. Keep the password you set secure.
- If a domain account is required for logins, run Sysprep on the ECS before using it to create a private image. For details about the impact of Sysprep operations, see **Why Is Sysprep Required for Creating a Private Image from a Windows ECS?**
- The Cloudbase-Init account of a Windows ECS is an internal account of the Cloudbase-Init agent. This account is used for obtaining metadata and completing relevant configuration when the Windows ECS starts. If you modify or delete this account, or uninstall the Cloudbase-Init agent, you will be unable to inject initial custom information into an ECS created from a Windows private image. Therefore, you are not advised to modify or delete the Cloudbase-Init account.

Figure 4-8 Running Sysprep



Follow-up Procedure

1. Create a private image from the ECS on which Sysprep is executed. For details, see **Creating a System Disk Image from a Windows ECS**.
2. You can use the image to create ECSs. Each ECS has a unique SID.

   Run the following command to query the ECS SID:

   whoami /user

Figure 4-9 ECS SID before Sysprep is executed

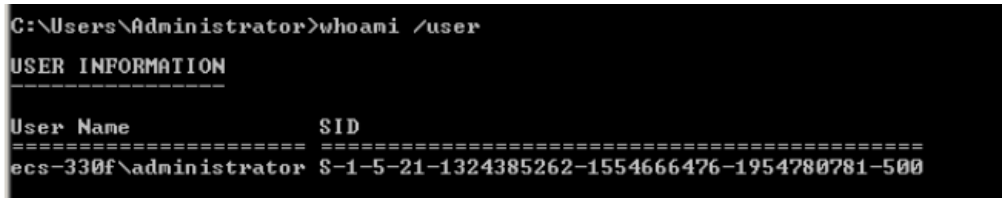
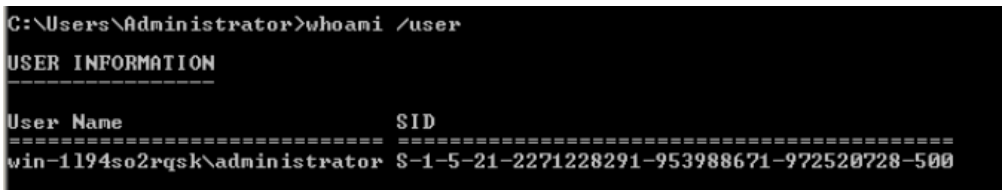


Figure 4-10 ECS SID after Sysprep is executed



5 Linux Operations

5.1 Setting the NIC to DHCP

Scenarios

If a private image is created from an ECS or external image file and the VM where the ECS or external image file is located is configured with a static IP address, you need to change the NIC attribute to DHCP so that the new ECSs created from the private image can dynamically obtain an IP address.

The configuration method varies depending on OSs.

NOTE

When registering an external image file as a private image, configure DHCP on the VM where the external image file is located. You are advised to configure DHCP on the VM and then export the image file.

Prerequisites

You have logged in to the ECS used to create a Windows private image.

For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

Ubuntu 18 or Later

1. Run **vi /etc/netplan/01-netcfg.yaml** on the ECS to open the **/etc/netplan/01-netcfg.yaml** file, and check whether the value of **dhcp4** is **true**.
   - If **dhcp4** is set to **true**, enter **:q** to exit the editor. No further action will be required.

     ```
     network:
       version:2
       renderer:NetworkManager
       ethernets:
           eth0:
             dhcp4: true
     ```

   - If **dhcp4** is set to **no** and a static IP address is configured, go to the next step.

```
network:
   version:2
   renderer:NetworkManager
   ethernets:
      eth0:
         dhcp4: no
         addresses: [192.168.1.109/24]
         gateway4: 192.168.1.1
         nameservers:
           addresses: [8.8.8.8,114.114.114.114]
```

2. Press **i** to enter the editing mode.

   Delete the static IP address settings and set **dhcp4** to **true**. You can also use a number sign (#) to comment out the static IP address settings.

```
network:
   version:2
   renderer:NetworkManager
   ethernets:
      eth0:
        dhcp4: true   # Set dhcp4 to true.
        #dhcp4: no    # Delete or comment out the static IP address settings.
        #addresses: [192.168.1.109]
        #gateway4: 192.168.1.1
        #nameservers:
        # addresses: [8.8.8.8,114.114.114.114]
```

3. If your ECS has more than one NIC, configure DHCP for all of them.

```
network:
   version:2
   renderer:NetworkManager
   ethernets:
       eth0:
          dhcp4: true
       eth1:
          dhcp4: true
       eth2:
          dhcp4: true
       eth3:
          dhcp4: true
```

4. Press **Esc**, enter **:wq**, and press **Enter** to save the settings and exit the vi editor.
5. Run the **netplan apply** command to make the settings take effect.

Ubuntu 16.04

1. Run the following command on the ECS to open the **/etc/network/interfaces** file:

   vi /etc/network/interfaces

   - If DHCP has been configured for all NICs, enter **:q** to exit the vi editor.

```
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet dhcp
```

   - If static IP addresses are set on the NICs, go to 2.

```
auto lo
iface lo inet loopback
auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.1.109
```

```
netmask 255.255.255.0
gateway 192.168.1.1
```

2. Press **i** to enter the editing mode.
3. Delete the static IP address settings and configure DHCP for the NICs.

   You can also use a number sign (#) to comment out the static IP address settings.

   ```
   auto lo
   iface lo inet loopback
   auto eth0
   iface eth0 inet dhcp
   ```

   If the ECS has multiple NICs, you must configure DHCP for all the NICs.

   ```
   auto lo
   iface lo inet loopback
   auto eth0
   iface eth0 inet dhcp
   auto eth1
   iface eth1 inet dhcp
   ```

4. Press **Esc**, enter **:wq**, and press **Enter**.

   The system saves the settings and exits the vi editor.

Related Operations

Configure DHCP to enable the ECS to obtain IP addresses continuously.

- For CentOS and EulerOS, use the vi editor to add **PERSISTENT_DHCLIENT="y"** to configuration file **/etc/sysconfig/network-scripts/ifcfg-ethX**.
- For SUSE Linux Enterprise, use the vi editor to set **DHCLIENT_USE_LAST_LEASE** to **no** in the configuration file **/etc/sysconfig/network/dhcp**.
- For Ubuntu 12.04 or later, upgrade dhclient to ISC dhclient 4.2.4 so that the NIC can consistently obtain IP addresses from the DHCP server. To perform the upgrade, you need to install isc-dhcp-server first.

5.2 Deleting Files from the Network Rule Directory

Scenarios

To prevent NIC name drift when you use a private image to create ECSs, you need to delete files from the network rule directory of the VM where the ECS or image file is located during the private image creation.

NOTE

When registering an external image file as a private image, delete files from the network rule directory on the VM where the external image file is located. You are advised to delete the files on the VM and then export the image file.

Prerequisites

An OS and VirtIO drivers have been installed on the ECS.

Procedure

1. Run the following command to query files in the network rule directory:

   ls -l /etc/udev/rules.d

2. Run the following commands to delete the files whose names contain **persistent** and **net** from the network rule directory:

   Example:

   rm /etc/udev/rules.d/*30*-**net_persistent**-*names*.**rules**

   rm /etc/udev/rules.d/*70*-**persistent-net.rules**

   The italic content in the commands varies depending on your environment.

   NOTE

   For CentOS 6 images, to prevent NIC name drift, you need to create an empty rules configuration file.

   Example:

   touch /etc/udev/rules.d/*75*-**persistent-net-generator.rules** //Replace *75* with the actual value in the environment.

3. Delete network rules.
   - If the OS uses the initrd system image, perform the following operations:
     i. Run the following command to check whether the initrd image file whose name starts with **initrd** and ends with **default** contains the **persistent** and **net** network device rule files (replace the italic content in the following command with the actual OS version):

     lsinitrd /boot/initrd-*2.6.32.12-0.7***-default |grep persistent|grep net**

     - If no, no further action is required.
     - If yes, go to **3.ii**.

     ii. Run the following command to back up the initrd image files (replace the italic part in the following command with the actual OS version):

     cp /boot/initrd-*2.6.32.12-0.7***-default /boot/initrd-***2.6.32.12-0.7***-default_bak**

     iii. Run the following command to generate the initrd file again:

     mkinitrd

   - If the OS uses the initramfs system image (such as Ubuntu), perform the following operations:
     i. Run the following command to check whether the initramfs image file whose name starts with **initrd** and ends with **generic** contains persistent and net rule files.

     lsinitramfs /boot/initrd.img-3.19.0-25-generic|grep persistent|grep net

     - If no, no further action is required.
     - If yes, go to **3.ii**.

     ii. Run the following command to back up the initrd image files:

     cp /boot/initrd.img-3.19.0-25-generic /boot/initrd.img-3.19.0-25-generic_bak

iii. Run the following command to generate the initramfs image files again:

update-initramfs -u

5.3 Installing Cloud-Init

Scenarios

To ensure that you can use the user data injection function to inject initial custom information into ECSs created from a private image (such as setting the ECS login password), install Cloud-Init on the ECS used to create the image.

- You need to download Cloud-Init from its official website. Therefore, you must bind an EIP to the ECS.
- If Cloud-Init is not installed, you cannot configure an ECS. As a result, you can only use the password in the image file to log in to the created ECSs.
- By default, ECSs created from a public image have Cloud-Init installed. You do not need to install or configure Cloud-Init on such ECSs.
- For ECSs created using an external image file, install and configure Cloud-Init by performing the operations in this section. For how to configure Cloud-Init, see **Configuring Cloud-Init**.

NOTE

Cloud-Init is open-source software. If the installed version has security vulnerabilities, you are advised to upgrade it to the latest version.

Prerequisites

- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Procedure

1. Check whether Cloud-Init has been installed.

   For details, see **Check Whether Cloud-Init Has Been Installed**.

2. Install Cloud-Init.

   You can install Cloud-Init using either of the following methods: **(Recommended) Install Cloud-Init Using the Official Installation Package** and **Install Cloud-Init Using the Official Source Code Package and pip**.

Check Whether Cloud-Init Has Been Installed

Perform the operations provided here to check whether Cloud-Init has been installed. The methods of checking whether Cloud-Init is installed vary depending on the OSs.

- If you are in a Python 3 environment, run the following command to check whether Cloud-Init is installed (Ubuntu22.0.4 is used as an example):

  which cloud-init

  - If information similar to the following is displayed, Cloud-Init has been installed:
    ```
    /usr/bin/cloud-init
    ```
  - If information similar to the following is displayed, Cloud-Init is not installed:
    ```
    /usr/bin/which: no cloud-init in (/usr/local/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin)
    ```

- If you are in a Python 2 environment, run the following command to check whether Cloud-Init is installed (CentOS 6 is used as an example):

  which cloud-init

  - If information similar to the following is displayed, Cloud-Init has been installed:
    ```
    cloud-init-0.7.5-10.el6.centos.2.x86_64
    ```
  - If no information is returned, Cloud-Init is not installed.

    NOTE

    To confirm Cloud-Init is really not installed, you are advised to run **rpm -qa |grep cloud-init** to check again. If either of **which cloud-init** and **rpm -qa |grep cloud-init** shows that Cloud-Init has been installed, Cloud-Init is installed.

If Cloud-Init has been installed, perform the following operations:

- Check whether to use the SSH certificate in the ECS OS. If the certificate is no longer used, delete it.
  - If the certificate is stored in a directory of user **root**, for example, */$path/$to/$root***/.ssh/authorized_keys**, run the following commands:

    cd /root/.ssh

    rm authorized_keys
  - If the certificate is not stored in a directory of user **root**, for example, */$path/$to/$none-root***/.ssh/authorized_keys**, run the following commands:

    cd /home/centos/.ssh

    rm authorized_keys
- Run the following command to delete the cache generated by Cloud-Init and ensure that the ECS created from the private image can be logged in by using the certificate:

  sudo rm -rf /var/lib/cloud/*

NOTE

Do not restart the ECS after performing the configuration. Otherwise, you need to configure it again.

(Recommended) Install Cloud-Init Using the Official Installation Package

The method of installing Cloud-Init on an ECS varies depending on the OS. Perform the installation operations as user **root**.

The following describes how to install Cloud-Init on an ECS running SUSE Linux, CentOS, Fedora, Debian, and Ubuntu. For other OS types, install the required type of Cloud-Init. For example, you need to install coreos-cloudinit on ECSs running CoreOS.

- SUSE Linux

  Paths for obtaining the Cloud-Init installation package for SUSE Linux

  https://ftp5.gwdg.de/pub/opensuse/repositories/Cloud:/Tools/

  http://download.opensuse.org/repositories/Cloud:/Tools/

  NOTE

  Select the required repo installation package in the provided paths.

  Take SUSE Enterprise Linux Server 12 as an example. Perform the following steps to install Cloud-Init:

  a. Log in to the ECS used to create a Linux private image.

  b. Run the following command to install the network installation source for SUSE Enterprise Linux Server 12:

     zypper ar https://ftp5.gwdg.de/pub/opensuse/repositories/Cloud:/Tools/SLE_12_SP3/Cloud:Tools.repo

  c. Run the following command to update the network installation source:

     zypper refresh

  d. Run the following command to install Cloud-Init:

     zypper install cloud-init

  e. Run the following commands to enable Cloud-Init to automatically start upon system boot:

     - SUSE 11

       chkconfig cloud-init-local on; chkconfig cloud-init on; chkconfig cloud-config on; chkconfig cloud-final on

       service cloud-init-local status; service cloud-init status; service cloud-config status; service cloud-final status

     - SUSE 12 and openSUSE 12/13/42

       systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

       systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  CAUTION

  For SUSE and openSUSE, perform the following steps to disable dynamic change of the ECS name:

  1. Run the following command to open the **dhcp** file using the vi editor:

     vi etc/sysconfig/network/dhcp

  2. Change the value of **DHCLIENT_SET_HOSTNAME** in the **dhcp** file to **no**.

- **CentOS**

  Table 5-1 lists the Cloud-Init installation paths for CentOS. Select the required installation package from the following addresses.

Table 5-1 Cloud-Init installation package addresses

OS Type	Version	How to Obtain
CentOS	6 32-bit	https://archives.fedoraproject.org/pub/archive/epel/6/i386/
	6 64-bit	https://archives.fedoraproject.org/pub/archive/epel/6/x86_64/
	7 64-bit	https://archives.fedoraproject.org/pub/epel/7/x86_64/Packages/e/

a. Run the following commands to install Cloud-Init:

yum install *Cloud-Init installation package address*/**epel-release-***x-y*.**noarch.rpm**

yum install cloud-init

NOTE

Cloud-Init installation package address indicates the address of the Cloud-Init epel-release installation package, and *x-y* indicates the version of the Cloud-Init epel-release required by the current OS. Replace them with the actual values according to **Table 5-1**.

- Take CentOS 6 64-bit as an example. If the version is 6.8, the command is as follows:

  yum install https://archives.fedoraproject.org/pub/archive/epel/6/x86_64/epel-release-6-8.noarch.rpm

- Take CentOS 7 64-bit as an example. If the version is 7.14, the command is as follows:

  yum install https://archives.fedoraproject.org/pub/epel/7/x86_64/Packages/e/epel-release-7-14.noarch.rpm

b. Run the following commands to enable Cloud-Init to automatically start upon system boot:

systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

- Fedora

  Before installing Cloud-Init, ensure that the network installation source address has been configured for the OS by checking whether the **/etc/yum.repo.d/fedora.repo** file contains the installation source address of the software package. If the file does not contain the address, configure the address by following the instructions on the Fedora official website.

  a. Run the following command to install Cloud-Init:

  yum install cloud-init

  b. Run the following commands to enable Cloud-Init to automatically start upon system boot:

  systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

- Debian and Ubuntu

  Before installing Cloud-Init, ensure that the network installation source address has been configured for the OS by checking whether the **/etc/apt/sources.list** file contains the installation source address of the software package. If the file does not contain the address, configure the address by following the instructions on the Debian or Ubuntu official website.

  a. Run the following commands to install Cloud-Init:

  apt-get update

  apt-get install cloud-init

  b. Run the following commands to enable Cloud-Init to automatically start upon system boot:

  systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

Install Cloud-Init Using the Official Source Code Package and pip

The following operations use Cloud-Init 0.7.9 as an example to describe how to install Cloud-Init.

1. Download the **cloud-init-0.7.9.tar.gz** source code package (version 0.7.9 is recommended) and upload it to the **/home/** directory of the ECS.

   Download **cloud-init-0.7.9.tar.gz** from the following path:

   https://launchpad.net/cloud-init/trunk/0.7.9/+download/cloud-init-0.7.9.tar.gz

2. Create a **pip.conf** file in the **~/.pip/** directory and edit the following content:

   NOTE

   If the **~/.pip/** directory does not exist, run the **mkdir ~/.pip** command to create it.

   ```
   [global]
   index-url  = https://<$mirror>/simple/
   trusted-host = <$mirror>
   ```

   NOTE

   Replace *<$mirror>* with a public network PyPI source.

   Public network PyPI source: **https://pypi.python.org/**

3. Run the following command to install the downloaded Cloud-Init source code package (select **--upgrade** as needed during installation):

   pip install [--upgrade] /home/cloud-init-0.7.9.tar.gz

   NOTE

   For details about how to install a Cloud-Init source code package, see **Cloud-Init Documentation**

4. Run the **cloud-init -v** command. Cloud-Init is installed successfully if the following information is displayed:

```
cloud-init 0.7.9
```

5. Enable Cloud-Init to automatically start upon system boot.
   - If the OS uses SysVinit to manage automatic start of services, run the following commands:

     chkconfig --add cloud-init-local; chkconfig --add cloud-init; chkconfig --add cloud-config; chkconfig --add cloud-final

     chkconfig cloud-init-local on; chkconfig cloud-init on; chkconfig cloud-config on; chkconfig cloud-final on

     service cloud-init-local status; service cloud-init status; service cloud-config status; service cloud-final status
   - If the OS uses Systemd to manage automatic start of services, run the following commands:

     systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

     systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

CAUTION

If you install Cloud-Init using the official source code package and pip, pay attention to the following:

1. Add user **syslog** to the **adm** group during the installation. If user **syslog** exists, add it to the **adm** group. For some OSs (such as CentOS and SUSE), user **syslog** may not exist. Run the following commands to create user **syslog** and add it to the **adm** group:

   useradd syslog

   groupadd adm

   usermod -g adm syslog
2. Change the value of **distro** in **system_info** in the **/etc/cloud/cloud.cfg** file based on the OS release version, such as **distro: ubuntu**, **distro: sles**, **distro: debian**, and **distro: fedora**.

5.4 Configuring Cloud-Init

Scenarios

You need to configure Cloud-Init after it is installed.

Prerequisites

- Cloud-Init has been installed.
- An EIP has been bound to the ECS.
- You have logged in to the ECS.
- The IP address obtaining mode of the ECS is DHCP.

Procedure

The following operations are required:

1. Configure Cloud-Init.
   For details, see **Configure Cloud-Init**.
2. Check whether Cloud-Init is successfully configured.
   For details, see **Check the Cloud-Init Configuration**.

Configure Cloud-Init

1. Configure the user permissions for logging in to the ECS. If you use a common account (not user **root**) to log in to the ECS, disable the SSH permissions of user **root** and remote login using a password to improve the ECS security.
   - You can remotely log in to the ECS using SSH and a key pair injected into your account. (It is recommended that you select the key pair login mode when creating an ECS.)
   - You can also use a random password to log in to the ECS through noVNC.

   Run the following command to open the **sshd_config** file using the vi editor:

   vi /etc/ssh/sshd_config
2. Change the value of **PasswordAuthentication** in the **sshd_config** file to **no**.

   NOTE

   For SUSE and openSUSE, change the values of the following parameters in the **sshd_config** file to **no**:
   - PasswordAuthentication
   - ChallengeResponseAuthentication
3. Run the following command to open the **cloud.cfg** file using the vi editor:

   vi /etc/cloud/cloud.cfg
4. (Optional) In **/etc/cloud/cloud.cfg**, set **apply_network_config** to **false**.

   This step is only for Cloud-Init 18.3 or later.

   Figure 5-1 Example configuration

   ```
   #      max_wait: 10 # (defaults to 120 seconds)
   +datasource_list: [ OpenStack ]
   +datasource:
   +  OpenStack:
   +    metadata_urls: ['http://                    ']
   +    max_wait: 120
   +    timeout: 5
   +    apply_network_config: false
   ```
5. Disable the SSH permissions of user **root** in **/etc/cloud/cloud.cfg**, add a common user (which is used for logging in to the ECS using VNC), and configure a password for the added user and assign sudo permissions to it.

NOTE

For Ubuntu and Debian, set the value of **manage_etc_hosts** in the **/etc/cloud/cloud.cfg** file to **localhost**. Otherwise, switching to user **root** may time out. For details about how to change the hostname of a cloud server, configure the SSH key, use Cloud-Init to add a user, and configure swap partitions, see **Cloud-Init Configuration FAQ**.

Take Ubuntu as an example.

- Run the following command to create script **/etc/cloud/set_linux_random_password.sh**, which is executable and can be used to generate random passwords:

  cat /etc/cloud/set_linux_random_password.sh

  The file content is as follows:

```
#!/bin/bash

password=$(cat /dev/urandom | tr -dc 'A-Za-z0-9!@#$%&+=' | head -c 9)

echo "linux:$password" | chpasswd
sed -i -e '/^Login/d' /etc/issue
sed -i -e '/^Initial/d' /etc/issue
sed -i -c -e '/^$/d' /etc/issue
echo -e "\nInitial login with linux:$password\n" >> /etc/issue
```

  NOTE

  You can run the **chmod +x /etc/cloud/set_linux_random_password.sh** command to add execute permissions of **set_linux_random_password.sh**.

- After you log in to the ECS, run the following commands to add a user-friendly prompt "Please change password for user linux after first login."

  echo -e '\e[1;31m##################################\e[0m' > /etc/motd

  echo -e '\e[1;31m# Important !!! #\e[0m' >> /etc/motd

  echo -e '\e[1;31m# Please change password for user linux after first login. #\e[0m' >> /etc/motd

  echo -e '\e[1;31m##################################\e[0m' >> /etc/motd

  echo -e '' >> /etc/motd

6. Add a common login user, set its password, assign sudo permissions to it, and use bootcmd to create a script used for generating a random password for each created ECS.

CAUTION

Ensure that the configuration file format (such as alignment and spaces) is consistent with the provided example.

```
system_info:
   # This will affect which distro class gets used
   distro: rhel
   # Default user name + that default users groups (if added/used)
   default_user:
     name: linux  #Username for login
     lock_passwd: False  #Login using a password is enabled. Note that some OSs use value 0 to enable the password login.
```

```
    gecos: Cloud User
    groups: users  #Optional. Add users to other groups that have been configured in /etc/group.
    passwd: $6$I63DBVKK
$Zh4lchiJR7NuZvtJHsYBQJIg5RoQCRLS1X2Hsgj2s5JwXI7KUO1we8WYcwbzeaS2VNpRmNo28vmxx
CyU6LwoD0
    sudo: ["ALL=(ALL) NOPASSWD:ALL"]  # Assign the root rights to the user.
    shell: /bin/bash  #Execute shell in bash mode.
  # Other config here will be given to the distro class and/or path classes
  paths:
    cloud_dir: /var/lib/cloud/
    templates_dir: /etc/cloud/templates/
  ssh_svcname: sshd

bootcmd:
- [cloud-init-per, instance, password, bash,
/etc/cloud/set_linux_random_password.sh]
```

NOTE

The value of **passwd** is encrypted using SHA512 (which is used as an example). For more details, see **https://cloudinit.readthedocs.io/en/latest/topics/examples.html**.

For details about how to encrypt a password and generate ciphertext, see the following (encrypting password **cloud.1234** is used as an example):

```
[root@** ~]# python -c "import crypt, getpass, pwd; print crypt.mksalt()"
$6$I63DBVKK
[root@** ~]# python -c "import crypt, getpass, pwd; print crypt.crypt('cloud.1234', '\$6\
$I63DBVKK')"
$6$I63DBVKK
$Zh4lchiJR7NuZvtJHsYBQJIg5RoQCRLS1X2Hsgj2s5JwXI7KUO1we8WYcwbzeaS2VNpRmNo28vmxx
CyU6LwoD0
```

7. Enable the agent to access the IaaS OpenStack data source.

   Add the following information to the last line of **/etc/cloud/cloud.cfg**:

```
datasource_list: [ OpenStack ]
datasource:
 OpenStack:
  metadata_urls: ['http://169.254.169.254']
  max_wait: 120
  timeout: 5
```

NOTE

- You can decide whether to set **max_wait** and **timeout**. The values of **max_wait** and **timeout** in the preceding example are only for reference.
- If the OS version is earlier than Debian 8 or CentOS 5, you cannot enable the agent to access the IaaS OpenStack data source.
- The default zeroconf route must be disabled for CentOS and EulerOS ECSs for accurate access to the IaaS OpenStack data source.

  echo "NOZEROCONF=yes" >> /etc/sysconfig/network

8. Prevent Cloud-Init from taking over the network in **/etc/cloud/cloud.cfg**.

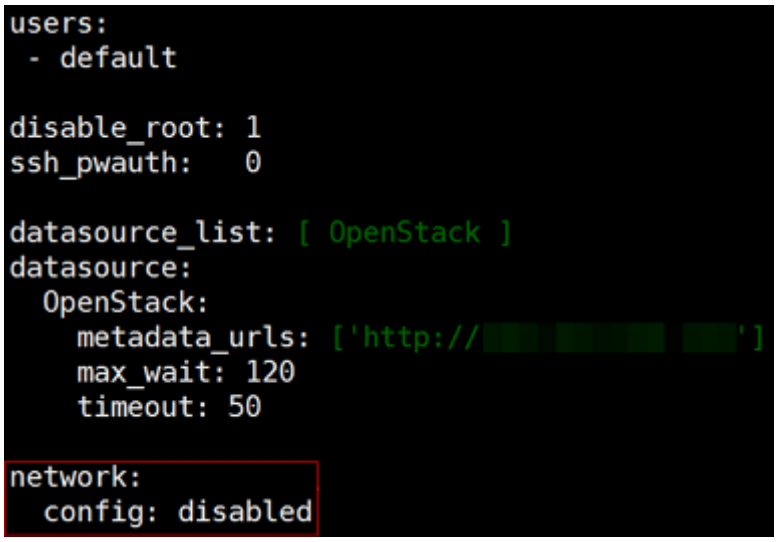
   If the Cloud-Init version is 0.7.9 or later, add the following content to **/etc/cloud/cloud.cfg**:

```
network:
 config: disabled
```

NOTE

The added content must be in the YAML format.

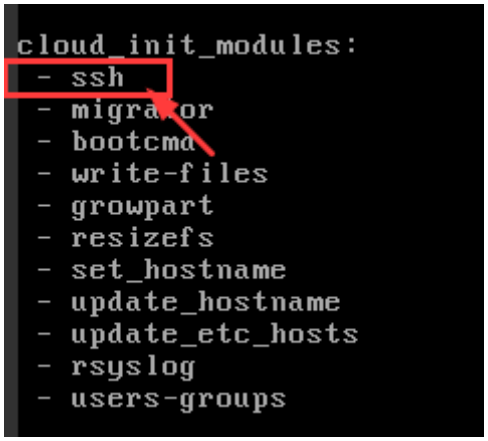
Figure 5-2 Preventing Cloud-Init from taking over the network



9. Modify **cloud_init_modules** in the **cloud.cfg** configuration file.

   Move **ssh** from the bottom to the top to speed up the SSH login.

   Figure 5-3 Speeding up the SSH login to the ECS



10. Modify the configuration so that the hostname of the ECS created from the image does not contain the **.novalocal** suffix and can contain a dot (.).

    a. Run the following command to modify the **__init__.py** file:

       vi /usr/lib/python2.7/site-packages/cloudinit/sources/__init__.py

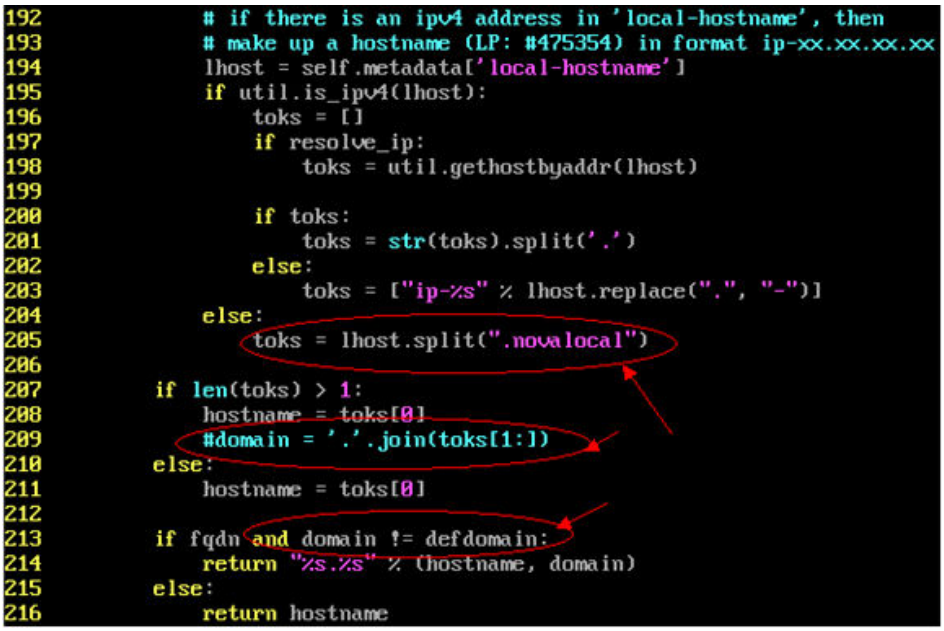
       Press **i** to enter editing mode. Modify the file content as follows based on the keyword **toks**:

```
if toks:
    toks = str(toks).split('.')
else:
    #toks = ["ip-%s" % lhost.replace(".", "-")]  # Comment out this line.
    toks = lhost.split(".novalocal")  # Add this line.

if len(toks) > 1:
    hostname = toks[0]
    #domain = '.'.join(toks[1:])   # Comment out this line.
else:
    hostname = toks[0]

if fqdn and domain != defdomain:
```

```
    #return hostname    # Comment out this line.
    return "%s.%s" % (hostname, domain) # Add this line.
else:
    return hostname
```

After the modification is complete, press **Esc** to exit the editing mode and enter **:wq!** to save the settings and exit.

Figure 5-4 Modifying the **__init__.py** file



b. Run the following command to switch to the **cloudinit/sources** folder:

   cd /usr/lib/python2.7/site-packages/cloudinit/sources/

c. Run the following commands to delete the **__init__.pyc** file and the optimized **__init__.pyo** file:

   rm -rf __init__.pyc

   rm -rf __init__.pyo

d. Run the following commands to clear the logs:

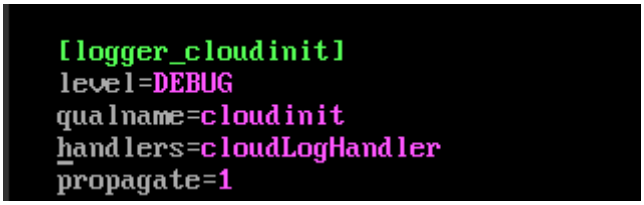
   rm -rf /var/lib/cloud/*

   rm -rf /var/log/cloud-init*

11. Run the following command to edit the **/etc/cloud/cloud.cfg.d/05_logging.cfg** file to use cloudLogHandler to process logs:

    vim /etc/cloud/cloud.cfg.d/05_logging.cfg

Figure 5-5 Setting the parameter value to **cloudLogHandler**



Check the Cloud-Init Configuration

Run the following command to check whether Cloud-Init has been properly configured:

cloud-init init --local

If Cloud-Init has been properly installed, the version information is displayed and no error occurs. For example, messages indicating lack of files will not be displayed.

NOTE

(Optional) Run the following command to set the password validity period to the maximum:

chage -M 99999 $user_name

user_name is a system user, such as user **root**.

You are advised to set the password validity period to **99999**.

5.5 Detaching Data Disks from an ECS

Scenarios

If multiple data disks are attached to the ECS used to create a private image, ECSs created from the image may be unavailable. Therefore, you need to detach all data disks from the ECS before using it to create a private image.

This section describes how to detach all data disks from an ECS.

Prerequisites

You have logged in to the ECS used to create a Linux private image.

Procedure

1. Check whether the ECS has data disks.

   Run the following command to check the number of disks attached to the ECS:

   fdisk -l

   - If the number is greater than 1, the ECS has data disks. Go to **2**.
   - If the number is equal to 1, no data disk is attached to the ECS. Go to **3**.

2. Run the following command to check the data disks attached to the ECS:

   mount

   - If the command output does not contain any EVS disk information, no EVS data disks need to be detached.

     ```
     /dev/vda1 on / type ext4 (rw,relatime,data=ordered)
     ```

   - If information similar to the following is displayed, go to **3**:

     ```
     /dev/vda1 on / type ext4 (rw,relatime,data=ordered)
     /dev/vdb1 on /mnt/test type ext4 (rw,relatime,data=ordered)
     ```

3. Delete the configuration information in the **fstab** file.

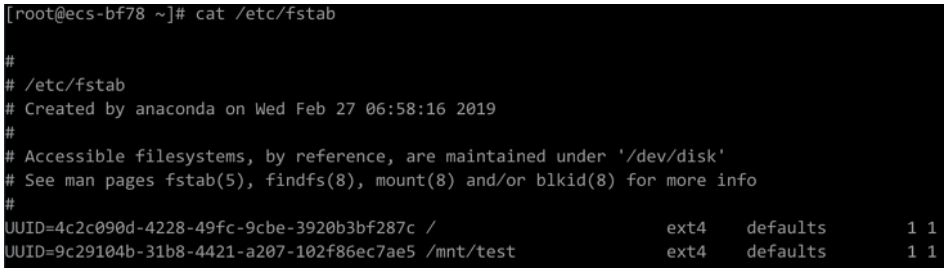
a. Run the following command to edit the **fstab** file:

vi /etc/fstab

b. Delete the disk configuration from the **fstab** file.

The **/etc/fstab** file contains information about the file systems and storage devices automatically attached to the ECS when the ECS starts. The configuration about data disks automatically attached to the ECS needs to be deleted, for example, the last line shown in the following figure.

Figure 5-6 EVS disk configuration in the **fstab** file



4. Run the following command to detach data disks from the ECS:

Run the following command to detach the disks:

umount */dev/vdb1*

5. Run the following command to check the data disks attached to the ECS:

mount

If the command output contains no information about the data disks, they have been detached from the ECS.

6 Permissions Management

6.1 Creating a User and Granting Permissions

Scenarios

This section describes how to use **Identity and Access Management** (IAM) to implement fine-grained permissions control over your images. With IAM, you can:

- Create IAM users for employees based on the organizational structure of your enterprise. Each IAM user has their own identity credentials for accessing images.
- Grant only the permissions required for users to perform a specific task.
- Entrust an account or cloud service to perform professional and efficient O&M on your images.

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

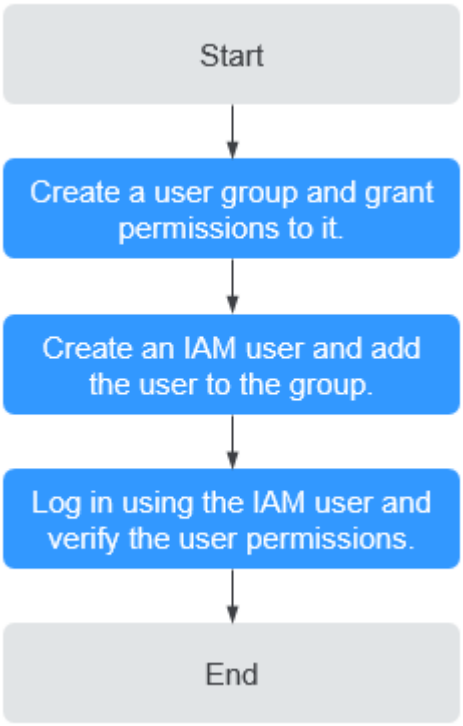
This section uses the **IMS ReadOnlyAccess** permission as an example to describe how to grant permissions to a user. **Figure 6-1** shows the process.

Prerequisites

Learn about the permissions (see **IMS Permissions**) supported by IMS. For the system permissions of other services, see **Permissions**.

Process Flow

Figure 6-1 Process for granting IMS permissions



1. **Create a user group and grant permissions to it**.
2. **Create an IAM user and add it to the user group**.
3. **Log in** and verify permissions.

   Log in to the management console using the IAM user, switch to a region where the permissions take effect, and verify the permissions (assume that the user has only the **IMS ReadOnlyAccess** permission).

   - In the **Service List**, choose **Image Management Service**. On the IMS console, perform operations except querying images, such as creating, modifying, and deleting an image.

     For example, click **Create Private Image** in the upper right corner. If you are prompted insufficient permissions, the **IMS ReadOnlyAccess** permission has taken effect.
   - Choose any other service in the **Service List**, such as **Virtual Private Cloud**. If a message appears indicating insufficient permissions to access the service, the **IMS ReadOnlyAccess** permission has taken effect.

6.2 Creating a Custom Policy

Scenarios

Custom policies can be created as a supplement to the system permissions of IMS. For the actions supported by custom policies, see **Permission Policies and Supported Actions**.

You can create custom policies in either of the following two ways:

- Visual editor: Select cloud services, actions, resources, and request conditions without the need to know policy syntax.
- JSON: Edit JSON policies from scratch or based on an existing policy.

For details, see **Creating a Custom Policy**. This section provides examples of common IMS custom policies.

Example Policies

- Example 1: Allowing users to create images

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "ims:serverImages:create"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "KMS:*:*"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "ecs:cloudServers:get",
            "ecs:servers:get",
            "ecs:serverVolumes:use",
            "ecs:cloudServers:list",
            "ecs:serverVolumeAttachments:list",
            "ecs:servers:list"

         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "bms:servers:list",
            "bms:servers:get",
            "bms:serverFlavors:get"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "evs:volumes:*"
         ]
      }
   ]
```

```
}
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "OBS:*:*"
         ]
      }
   ]
}
```

NOTE

The action required for creating an image is **ims:serverImages:create**. Others are dependent actions for creating an image.

- Example 2: Denying image deletion

  A deny policy must be used in conjunction with other policies to take effect. If the policies assigned to a user contain both Allow and Deny actions, the Deny actions take precedence over the Allow actions.

  The following method can be used if you need to assign the **IMS FullAccess** policy to a user but also forbid the user from deleting images. Create a custom policy for denying image deletion, and assign both the policies to the group the user belongs to. Then, the user can perform all operations on IMS except deleting images. The following is an example deny policy:

```
{
   "Version": "1.1",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": [
            "ims:images:delete"
         ]
      }
   ]
}
```

7 FAQs

7.1 Image Consulting

7.1.1 Basic Concepts

Images are classified as public, private, and shared.

Image Type	Description
Public	A public image is a standard, widely used image. It contains an OS and preinstalled public applications and is available to all users. Public images are very stable and their OS and any included software have been officially authorized for use. If a public image does not contain the environments or software you need, you can use a public image to create an ECS and then deploy the required environments or software on it.
Private	A private image contains an OS or service data, preinstalled public applications, and a user's personal applications. Private images are only available to the users who created them. A private image can be a system disk image, data disk image, ISO image, or full-ECS image. • A system disk image contains an OS and preinstalled software for various services. You can use a system disk image to create ECSs and migrate your services to the cloud. • A data disk image contains only service data. You can use a data disk image to create EVS disks and use them to migrate your service data to the cloud. • An ISO image is created from an external ISO image file. It is a special image that is not available on the ECS console. • A full-ECS image contains an OS, preinstalled software, and service data. A full-ECS image is created using differential backups and the creation takes less time than creating a system or data disk image that has the same disk capacity.

Image Type	Description
Shared	A shared image is a private image another user has shared with you. For more information about shared images, see "Sharing Images" in *Image Management Service User Guide*.

You can modify an image, share images, export images, encrypt images, replicate images within a region, export an image list, and delete images.

Table 7-1 Managing private images

Feature	Description	Helpful Link
Modifying an image	To facilitate private image management, you can modify the following attributes of an image: name, description, minimum memory, maximum memory, and advanced functions such as NIC multi-queue and SR-IOV driver.	**Modifying an Image**
Sharing images	You can share an image with other accounts. These accounts can use your shared private image to quickly create ECSs or EVS disks.	• **Sharing Images** • **Image Sharing**
Exporting images	You can export private images to your OBS bucket and download them to your local PC for backup.	• **Exporting an Image** • **Image Exporting**
Encrypting images	You can create encrypted images to improve data security. The encryption mode is KMS envelope encryption. Encrypted images can be created from external image files or encrypted ECSs.	• **Encrypting Images**
Replicating images within a region	By replicating images within a region, you can convert encrypted and unencrypted images into each other or enable some advanced features, for example, quick instance provisioning.	**Replicating Images**
Tagging an image	You can tag your private images for easy management and search.	**Tagging an Image**

Feature	Description	Helpful Link
Exporting image list	You can export the public or private image list in a given region in CSV format, facilitating local maintenance and query.	**Exporting Image List**
Deleting images	You can delete images that will be no longer used. Deleting an image does not affect the ECSs created from that image.	**Deleting Images**

7.1.2 How Do I Select an Image?

When creating an ECS, you can select an image based on the following factors:

- **Image Type**
- **OS**

Image Type

Images are classified into public images, private images, and shared images. A private image can be a system disk image, data disk image, or full-ECS image. For details, see **What Is Image Management Service?**

OS

When selecting an OS, consider the following factors:

- Architecture types

System Architecture	Applicable Memory	Constraints
32-bit	Smaller than 4 GB	• If the instance memory is greater than 4 GB, a 32-bit OS cannot be used. • A 32-bit OS only allows addressing within a 4 GB memory range. An OS with more than 4 GB memory cannot be accessed.
64-bit	4 GB or larger	If your application requires more than 4 GB of memory or the memory may need to be expanded to more than 4 GB, use a 64-bit OS.

- OS types

OS Type	Applicable Scenario	Constraints
Windows	• Programs developed for Windows (for example, .NET). • Databases such as SQL Server. (You need to install the database.)	The system disk must be at least 40 GB, and there must be at least 1 GB of memory.
Linux	• High-performance server applications (for example, Web) and working with common programming languages such as PHP and Python. • Databases such as MySQL. (You need to install the database.)	The system disk must be at least 40 GB, and there must be at least 512 MB of memory.

7.1.3 What Do I Do If I Cannot Find a Desired Image?

You can view OS types and versions on the **Public Images** page on the management console. If you cannot find a desired image, you have the following options:

- Download an image file from the official OS website and then use the file to create a private image. For details, see **Creating a Windows System Disk Image from an External Image File** or **Creating a Linux System Disk Image from an External Image File**. The external image file can be in the VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, or ZVHD format.
- If you already have an ISO file and the OS is supported by the cloud platform, you can create a private image as follows:
  a. Create a private image on the management console. For details, see **Creating a Windows System Disk Image from an ISO File** or **Creating a Linux System Disk Image from an ISO File**.
- If the image belongs to another tenant, ask the tenant to share it with you. For details about image sharing, see **Sharing Specified Images**.

7.1.4 How Do I Increase the Image Quota?

What Is Quota?

Quotas can limit the number or amount of resources available to users, such as the maximum number of ECSs or EVS disks that can be created.

If the existing resource quota cannot meet your service requirements, you can apply for a higher quota.

How Do I View My Quotas?

1. Log in to the management console.
2. Click  in the upper left corner and select the desired region and project.
3. In the upper right corner of the page, click .
   The **Service Quota** page is displayed.
4. View the used and total quota of each type of resources on the displayed page.
   If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment. If you need to adjust a quota, call the hotline or send an email to the customer service mailbox. Customer service personnel will timely process your request for quota adjustment and inform you of the real-time progress by making a call or sending an email.

Before dialing the hotline number or sending an email, make sure that the following information has been obtained:

- Account name, project name, and project ID, which can be obtained by performing the following operations:
  Log in to the management console using the cloud account, click the username in the upper right corner, select **My Credentials** from the drop-down list, and obtain the account name, project name, and project ID on the **My Credentials** page.
- Quota information, which includes:
  - Service name
  - Quota type
  - Required quota

Learn how to obtain the service hotline and email address.

7.1.5 What Are the Differences Between Images and Backups?

CBR and Image Management Service (IMS) have some complementary functions and can be used together in certain scenarios. Like CBR, IMS can also be used to back up ECSs.

Differences Between Backups and Images

Table 7-2 lists the differences between them.

Table 7-2 Differences between backups and images

Item	CBR	IMS
Concept	A backup contains the status, configuration, and data of a cloud server or disk stored at a specific time point for recovery in case of a fault. It is used to ensure data security and improve availability.	An image provides all information required for starting a cloud server. It is used to create a cloud server and deploy software environments in batches. A system disk image contains an OS and pre-installed application software for running services. A data disk image contains service data. A full-ECS image contains data of the system disk and data disks.
Usage method	• Data storage location: Unlike server or disk data, backups are stored in OBS. Deleting a disk will not clear its backups. • Operation object: A server or disk can be backed up at a given point in time. CBR supports automatic backup and automatic deletion by configuring backup policies. • Usage: Backups can be used to restore data to the original server or disk, or to create a new disk or full-ECS image. • Support exporting to a local PC: No	• Data storage location: Unlike server or disk data, backups are stored in OBS. If a server or disk that is created using an image is deleted, the image will not be cleared. • Operation object: The system disk and data disks of a server can be used to create private images. You can also create private images using external image files. • Usage: System disk images or full-ECS images can be used to create new servers, and data disk images can be used to create new disks for service migration. • Support exporting to a local PC: Yes However, full-ECS images cannot be exported to a local PC.
Application scenarios	• Data backup and restoration • Rapid service deployment and migration	• Server migration to the cloud or between clouds • Deploying a specific software environment • Deploying software environments in batches • Backing up server operating environments

Item	CBR	IMS
Advantages	Supports automatic backup. Data on a server or disk at a certain time point can be retained periodically or quantitatively. You can back up on-premises VMware VMs, synchronize the backups to the cloud, and then use the backups to restore data to new ECSs.	Supports system disk backup. You can import the data disk image of a local server or a server provided by another cloud platform to IMS and then use the image to create an EVS disk.

NOTE

Although backups and images are stored in OBS, you cannot view backup and image data in OBS, because they do not occupy your resources.

Relationship Between Backups and Images

1. You can use an ECS backup to create a full-ECS image.
2. Before creating a full-ECS image for an ECS, you need to back up the target ECS.
3. A backup is compressed when it is used to create an image, so the size of the generated image may be smaller than the backup size.

7.1.6 Can I Tailor an Image?

When you import an external image file, you are advised to import the image that contains the official OS release version. Do not tailor or highly customize the release version. Otherwise, problems may occur.

OS vendors do not always update OS release versions regularly. Some versions are no longer maintained, and these deprecated versions no longer receive security patches. Ensure that you read the update notifications from OS vendors and update your OS so that it runs properly.

7.1.7 How Can I Back Up the Current Status of an ECS for Restoration in the Case of a System Fault?

You can back up the ECS in any of the following ways:

- (Recommended) Use CBR to create a scheduled backup task for the ECS. If the ECS fails, select a backup corresponding to the time you want the ECS to restore to, create a full-ECS image from the backup, and use the image to apply for a new ECS or to reinstall the OS.
- Create a system disk image from the ECS. If the ECS fails, use the system disk image to apply for a new ECS or to reinstall the OS.
- Create a snapshot for the system disk of the ECS. If the ECS fails, you can roll it back from the snapshot.

7.1.8 How Can I Apply a Private Image to an Existing ECS?

- You can change the OS of the ECS later. When you change the OS, select the created private image. For details about how to change the OS, see "Changing the OS" in *Elastic Cloud Server User Guide*.

7.1.9 Can I Import Data from a Data Disk Image to a Data Disk?

No.

A data disk image can only be used to apply for a new disk and its data cannot be imported to a disk. To import the data, perform the following operations:

1. Use the data disk image to create a temporary disk.
2. Attach the temporary disk to the ECS where the target disk is located.
3. Copy data from the temporary disk to the target disk. Then, delete the temporary disk.

7.1.10 Can I Use Private Images of Other Tenants?

Yes.

Other tenants can share a private image with you. You can use it after accepting it. For details about image sharing, see **Sharing Specified Images**.

7.2 Image Creation

7.2.1 Image Creation FAQs

How Can I Use an ECS to Quickly Provision Identical ECSs?

If you have an ECS with applications deployed, you can use the ECS to create a private image and then use the image to create identical ECSs. In this way, you do not need to deploy applications repeatedly.

- **Creating a System Disk Image from a Windows ECS**
- **Creating a System Disk Image from a Linux ECS**
- **Creating ECSs from an Image**

How Many Private Images Can I Create Under an Account?

Currently, you can create a maximum of 100 private images under an account in a region.

Do I Have to Stop the ECS Before Using It to Create a Private Image?

No. You can create an image from a running ECS. However, if data is written to the ECS during image creation, that new data will not be included in the created image.

Where Can I View the Image Creation Progress? How Long Does It Take to Create an Image?

Log in to the management console. Choose **Computing** > **Image Management Service** and click the **Private Images** tab. Monitor the image creation progress in the **Status** column.

The image creation involves the installation of KVM drivers, OS kernel loading, and GRUB boot configuration, which may take a long time. In addition, the network speed, image file type, and disk size have an impact on how long image creation takes.

Can I Select a Private Image Created Under a Subaccount When Creating an ECS Under the Main Account?

Yes.

Private images created under a subaccount are visible to the main account and all the other subaccounts (if any) under the main account.

- If the private image is a system disk image or full-ECS image, you can select **Private Image** for **Image** when creating an ECS. Then, select this image from the drop-down list.
- If the private image is a data disk image, select **Create from image** for **Select Data Source** when creating an EVS disk. Then, select this image in the displayed dialog box.

In addition, private images created under the main account are visible to all of its subaccounts.

7.2.2 Full-ECS Image FAQs

What Is a Full-ECS Image?

A full-ECS image contains the OS, applications, and service data of an ECS. Generally, a full-ECS image is used to migrate all data of an ECS. For example:

- Sharing an ECS with other tenants
- Migrating data from an old ECS to a new one

Why Do I Have to Select a Vault When Creating a Full-ECS Image? Do I Need to Pay for the Vault?

When creating a full-ECS image from a CBR backup, you must select a vault. The vault is where your images and backups are stored. You need to pay for the vault.

Therefore, no matter which backup type you select, you need to pay for the vault. Selecting a vault does not mean that you need to pay extra fees.

Where Can I View the Data Disk Information of a Successfully Created Full-ECS Image?

After a full-ECS image is created, only the system disk information (**Disk Capacity**) is displayed in the image list and image details. You can view the data disk information on the CBR console.

The following describes how to view the data disk details in CBR:

1. In the private image list, click the full-ECS image name.

   Image details are displayed.
2. Locate **Source** and click the backup ID following it.

   The CBR details page is displayed.
3. Click the **Disk Backup** tab. Details about the system disk and data disks are displayed.

What Are the Restrictions on Using a Full-ECS Image?

- A full-ECS image cannot be exported. You are advised to create images for the system disk and data disks separately and then export the images.
- Only the full-ECS image created from a CBR backup is shareable with other tenants.
- A full-ECS image cannot be replicated within the same region.

7.2.3 Is There Any Difference Between the Image Created from a CSBS/CBR Backup and That Created from an ECS?

No.

You can create a full-ECS image from an ECS or a CBR backup.

When you create a full-ECS image from an ECS, the system first creates a backup for the ECS and then uses the backup to create an image. Therefore, the image is essentially created from an ECS backup no matter you use an ECS or a CSBS/CBR backup.

7.2.4 Why Can't I Find an ISO Image When I Want to Use It to Create an ECS or Change the OS of an ECS?

- An ISO image created from an ISO file is used only for creating a temporary ECS. It will not be available on the ECS console. You cannot use it to create ECSs or change ECS OSs. You need to install an OS on the temporary ECS and use that ECS to create a system disk image which can be used to create ECSs or change ECS OSs.
- You are not advised to use a temporary ECS as a normal ECS because it has limited functionality. For example, disks cannot be attached to it.

For details about how to create a private image using an ISO file, see:

- **Creating a Windows System Disk Image from an ISO File**
- **Creating a Linux System Disk Image from an ISO File**

7.2.5 How Do I Create a Full-ECS Image Using an ECS That Has a Spanned Volume?

An ECS used to create a Windows full-ECS image cannot have a spanned volume. If you attempt to create an image from an ECS with a spanned volume, when the image is used to create new ECSs, data may be lost.

If an ECS has a spanned volume, back up data in the spanned volume and then delete this volume from the ECS. Use the ECS to create a full-ECS image. Use the full-ECS image to create an ECS. Then, use the backup to create a spanned volume for the new ECS if necessary.

NOTE

If a Linux ECS has a volume group or a logical volume consisting of multiple physical volumes, to ensure you do not lose any data, back up data in the volume group or logical volume and delete the volume group or logical volume before using this ECS to create a full-ECS image.

7.2.6 Why Is Sysprep Required for Creating a Private Image from a Windows ECS?

Why Is Sysprep Required?

For a user that needs to be added to a domain and uses the domain account to log in to Windows, Sysprep is required before a private image is created. Otherwise, the image will contain information about the original ECS, especially the SID. ECSs with the same SID cannot be added to a domain. If Windows does not require any user or ECS to be added to a domain, you do not need to run Sysprep.

CAUTION

- Before running Sysprep, ensure that Windows is activated.
- For details about Sysprep, visit **https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-vista/cc721940(v=ws.10)?redirectedfrom=MSDN**.

Restrictions on Running Sysprep

Sysprep can only be used for configuring a new Windows installation. You can run Sysprep multiple times to install and configure Windows. However, you can reset and activate a Windows OS only three times, and you are not allowed to use Sysprep to re-configure an existing Windows OS.

NOTE

In the Windows command line, enter the following command to check how many times you can run Sysprep in the displayed **Windows Script Host** dialog box:

slmgr /dlv

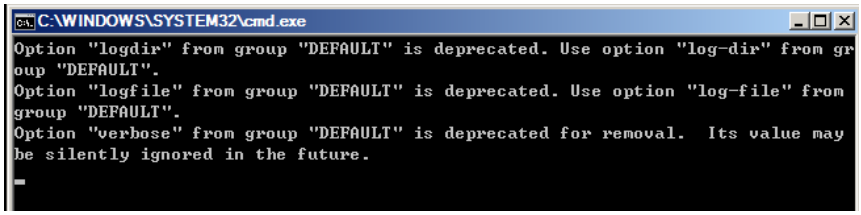
If the value of **Remaining Windows rearm count** is **0**, you cannot run Sysprep.

7.2.7 What Do I Do If an ECS Created from a Windows Image Failed to Start After Running Sysprep?

Symptom

1. After Sysprep is executed, the following message is displayed when you start the ECS.

   Figure 7-1 Message displayed

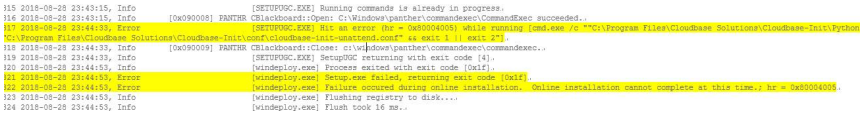
   Then, the following information is displayed in the dialog box:

   Windows could not parse or process the unattend answer file for pass [specialize]. A component or setting specified in the answer file does not exist. The error was detected while processing settings for component [Microsoft-Windows-Shell-Setup].

2. Click **OK**. The following information is displayed in the dialog box:
   The computer accidentally restarts or encounters an error. Windows installation cannot continue. Click OK to restart the computer and restart the installation.

3. Open **setupact.log** in **C:\Windows\Panther**. The log contains the following information.

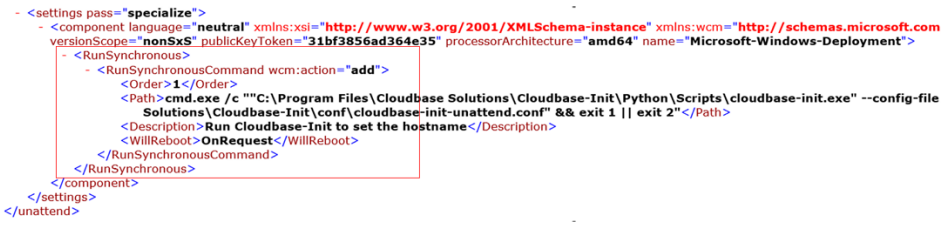
   Figure 7-2 Viewing ECS logs

Solution

1. Create an ECS from a public image. (You are advised to use a public image to create another ECS because Sysprep can be executed only for certain times.)
2. Create an **Unattend.xml** file or modify the **Unattend.xml** file provided by the system.
   - If you create an **Unattend.xml** file, ensure that the created file is used when you run Sysprep. For details about the file, visit:
     - https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/update-windows-settings-and-scripts-create-your-own-answer-file-sxs
     - https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/sysprep--system-preparation--overview
   - If you modify the **Unattend.xml** file (in the **C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf** directory), delete the **RunSynchronous** part from the file.

Figure 7-3 Deleting the RunSynchronous part



3. Run Sysprep. For details, see **Running Sysprep**.

   NOTICE

   If you use the **Unattend.xml** file created by yourself, check the **Unattend.xml** path when running Sysprep to ensure that the newly created **Unattend.xml** file is used.

4. Create an image from the ECS where Sysprep has been executed.

7.2.8 What Do I Do If I Cannot Create an Image in ZVHD2 Format Using an API?

Symptom

When you create a ZVHD2 image using an API, the image is created in the ZVHD format.

Solution

Check whether your token contains the **op_gated_lld** role (**op_gated_lld** is the OBT tag, which can be viewed in the body of the response message of the API used to obtain a user token). The ZVHD2 image has the lazy loading feature. If the current environment does not support this feature or this feature is in the OBT phase, the ZVHD2 image will fail to be created.

Contact the administrator to ensure that the current environment supports lazy loading, obtain a new token, and use the new token to create an image.

7.3 Image Sharing

7.3.1 Image Sharing FAQs

How Many Tenants Can I Share an Image with?

A system disk image or data disk image can be shared with up to 128 tenants, but a full-ECS image can only be shared with up to 10 tenants.

How Many Images Can Be Shared with Me?

There is no limit.

Do Shared Images Affect My Private Image Quota?

No.

I Shared an Image to an Account But the Account Did Not Accept or Reject the Image. Will My Image Sharing Quota Be Consumed?

No.

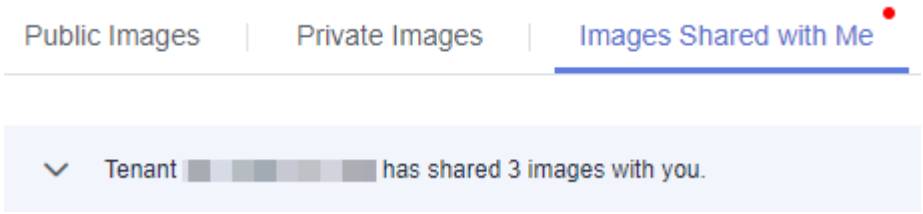
Where Can I View the Images Shared with Me?

Switch to the region where the shared image is located, choose **Service List** > **Computing** > **Image Management Service** > **Images Shared with Me**.

If you are a multi-project user, make clear which of your projects will receive the shared image. Switch to the region where the project resides and select the project. Then, choose **Service List** > **Computing** > **Image Management Service** > **Images Shared with Me**.

If the image is not accepted, a red dot is displayed on the **Images Shared with Me** tab page (as shown in **Figure 7-4**) and a message is displayed, asking you whether to accept the shared image. After the image is accepted, it is displayed in the list on the **Images Shared with Me** tab page.

Figure 7-4 Images Shared with Me



If I Want to Share a System Disk Image with Another Account, Should the Account Purchase an ECS in Advance?

No. The account can use the shared image to apply for ECSs.

Is There Any Restriction on the Region When I Create ECSs Using a Shared Image?

Yes. You can only create ECSs in the same region as the shared image.

Can I Share Images Shared with Me with Other Tenants?

You cannot directly share such images with other tenants. If you do need to do so, you can replicate a shared image to a private image and then share the private image.

Can I Use an Image I Have Shared with Others to Create an ECS?

Yes. After sharing an image with other tenants, you can still use the image to create an ECS and use the created ECS to create a private image.

What Are the Risks of Creating ECSs Using a Shared Image?

The image owner can view, stop sharing, or delete the image at any time. After the shared image is deleted, you will be unable to use the shared image to reinstall the OSs of the ECSs created from the shared image or create ECSs with the same configurations.

The cloud platform does not ensure the integrity or security of images shared by other accounts. For security reasons, you are advised to choose only images shared by trusted accounts.

What Are the Risks of Sharing Images with Other Tenants?

Data, files, and software may be disclosed. Before sharing an image, you must take care to delete any sensitive data or important files from the image. The image recipient can use the shared image to create ECSs and use the created ECSs to create private images. If the created private images are shared with other tenants, any data leakage that occurs can be quite widespread.

Can I Specify a Region or an AZ for Sharing an Image?

No. When sharing an image, you can only specify a project name. You cannot specify a region or an AZ. An image can only be shared within a given region, but once shared, it can be used in any AZ in that region.

Can I Restore My Data Disks from a Data Disk Image Shared by Another Account?

No. You can only use the shared image to apply for a new data disk and cannot use it to restore your existing data disks. However, you can use the new data disk for restoration by referring to **Can I Import Data from a Data Disk Image to a Data Disk?**

What Can I Do If I Want to Use a Rejected Image?

If you have rejected an image shared by another tenant, but now want to use it, two methods are available:

- Method 1

  Ask the image owner to add you to the tenants the image is shared with. For details, see **Adding Tenants Who Can Use Shared Images**.
- Method 2

  Accept the rejected image again. For details, see **Accepting Rejected Images**.

7.3.2 What Are the Differences Between Sharing Images and Replicating Images?

- Sharing images:

  You can only share images within a region with other users. To share an image across regions, replicate the image to the target region and then share it. For details, see **Overview**.

- Replicating images:
  - In-region: You can convert encrypted and unencrypted images into each other or enable some advanced features (such as fast ECS creation from an image) using the in-region image replication function.

The following table describes the details.

Scenario	Operation	Description	Helpful Links
Sharing	Share	The image is shared with another user in the same region. The target user can use the image (with the same ID as the source image) but the image owner is still the user who shared it.	For details, see **Sharing Specified Images**.
In-region replication under the same account	Replicate	This is used for conversion between encrypted images and unencrypted images or for enabling advanced features (such as fast ECS creation from an image).	For details, see **Replicating Images**.

7.3.3 What Do I Do If I Cannot Share My Images?

- Some images cannot be shared. Therefore, the **Share** option is not provided for them in the **Operation** column. The following images cannot be shared:
  - Encrypted images

7.4 OS

7.4.1 How Do I Select an OS?

- Linux

  Used for development platforms or services that run Linux. CentOS and Ubuntu are provided. CentOS is recommended.

  The system disk must be no less than 40 GB, and the memory must be no less than 512 MB.
- OS selection for servers that require memory greater than 4 GB

  Because 32-bit OSs allow addressing only within a 4 GB memory range, if the required memory capacity is 4 GB or larger, select a 64-bit OS.

7.4.2 How Is BIOS Different from UEFI?

Table 7-3 Differences between the UEFI and BIOS boot modes

Boot Mode	Description	Highlight
BIOS	Basic Input Output System (BIOS) stores important basic input/output programs of ECSs, system settings, self-test programs upon system startup, and automatic startup programs.	Provides basic settings and control for ECSs.
UEFI	Unified Extensible Firmware Interface (UEFI) is a specification that defines a software interface between an OS and platform firmware. UEFI can be used to automatically load an OS from a pre-boot operating environment.	Boots up or recovers from sleep state faster.

7.4.3 How Do I Delete Redundant Network Connections from a Windows ECS?

Method 1

1. Press **Win+R**. In the displayed dialog box, enter **regedit** and press **Enter** to open the registry editor.

   NOTE

   Modifying a registry may cause a system startup failure. So, back up the registry before modifying it.

2. Open the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT \CurrentVersion\NetworkList\Profiles

   Click each item under **Profiles** and query the **Data** column of **ProfileName** in the right pane.

3. Double-click **ProfileName** and set **Value Data** to the name of a new network.
4. Restart the ECS for the change to take effect.

Method 2

1. Press **Win+R**. In the displayed dialog box, enter **regedit** and press **Enter** to open the registry editor.

NOTE

Modifying a registry may cause a system startup failure. So, back up the registry before modifying it.

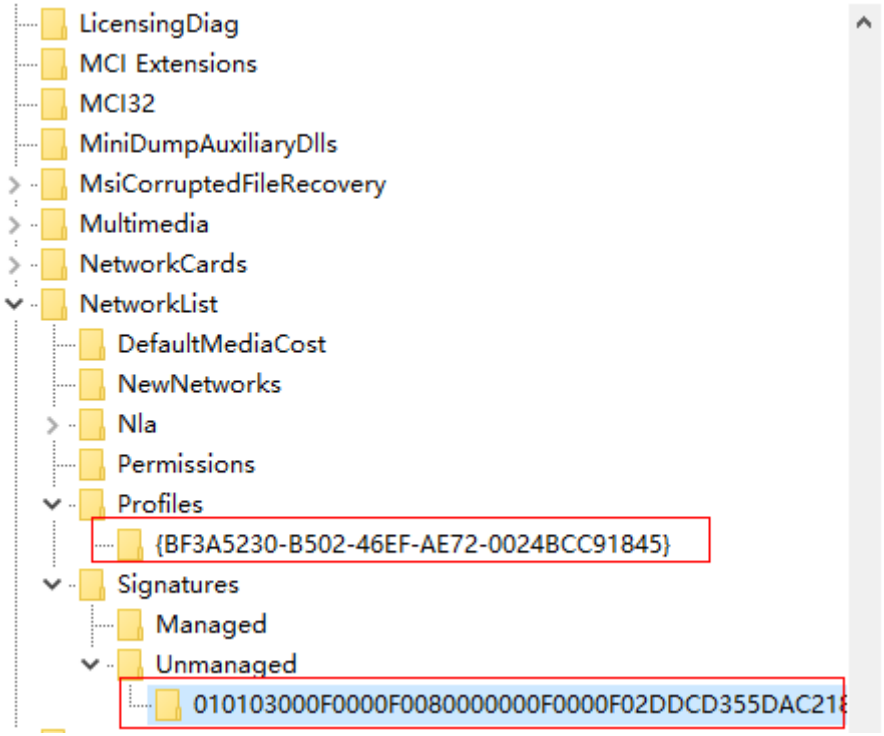
2. Open the following registry keys:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Signatures\Unmanaged

3. Delete the directories shown in the following figure.

Figure 7-5 Registry directory



7.4.4 What Do I Do If an ECS Starts Slowly?

Symptom

If an ECS starts slowly, you can change the default timeout duration to speed up the startup.

Solution

1. Log in to the ECS.
2. Run the following command to switch to user **root**:

   sudo su

3. Run the following command to query the version of the GRUB file:

   rpm -qa | grep grub

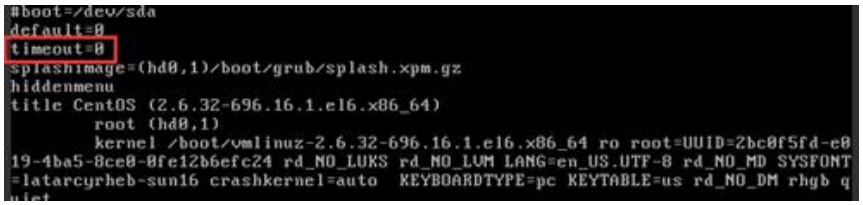
Figure 7-6 Querying the GRUB file version

4. Set **timeout** in the GRUB file to **0**.
   - If the GRUB file version is earlier than 2:

     Open **/boot/grub/grub.cfg** or **/boot/grub/menu.lst** and set **timeout** to **0**.
   - If the GRUB file version is 2:

     Open **/boot/grub2/grub.cfg** and set the value of **timeout** to **0**.

     Figure 7-7 Modifying the timeout duration



7.4.5 Why Can't I Find My Private Image When I Want to Use It to Create an ECS or Change the OS of an ECS?

When you create an ECS or change the OS of an existing ECS, some of your private images are not shown. One possible cause is that the x86 and Arm architectures are incompatible with each other, or that there is an incompatibility issue between UEFI and BIOS boot modes.

- If a private image is created from a x86 ECS, this image will be invisible to you when you create an Arm (Kunpeng) ECS or change the OS of an Arm (Kunpeng) ECS, and vice versa.
- If you use an external image file to create a private image and select the x86 architecture, this image will be invisible to you when you create an Arm (Kunpeng) ECS or change the OS of an Arm (Kunpeng) ECS, and vice versa.
- If a private image is created from an ECS in BIOS boot mode, this image will be invisible to you when you create an ECS in UEFI boot mode or change the OS of an ECS in UEFI boot mode, and vice versa.
- If you use an external image file to create a private image and select the BIOS boot mode, this image will be invisible to you when you create an ECS in UEFI boot mode or change the OS of an ECS in UEFI boot mode, and vice versa.

7.5 Image Importing

7.5.1 Can I Use Images in Formats Other Than the Specified Ones?

No. Currently, only the VMDK, VHD, RAW, QCOW2, VHDX, QED, VDI, QCOW, ZVHD2, ISO, and ZVHD formats are supported.

Images of the -flat.vmdk format and image file packages containing snapshot volumes or delta volumes are not supported. You can use **qemu-img** to convert

an image to one of the supported formats before uploading it to the cloud platform.

NOTE

For how to install and use **qemu-img** in Windows, visit:
https://cloudbase.it/qemu-img-windows/

7.5.2 What Are the Impacts If I Do Not Pre-configure an ECS Used to Create a Private Image?

Before using an ECS or external image file to create a private image, you need to pre-configure the ECS or the source VM of the image file. If you do not perform the pre-configuration, there will be the following impacts:

1. If you do not delete residual rule files from the **udev** directory, new ECSs will retain the configurations of the source ECS or image file. If you do not set the IP address assignment mode to DHCP, NICs of new ECSs will not start from eth0. You need to remotely log in to the new ECSs to perform configurations.
2. For Linux, the following issues may occur during the ECS creation:
   - Custom passwords cannot be injected.
   - Certificates cannot be injected.
   - Other custom configurations cannot be applied on new ECSs.
3. If you do not delete information about automatic disk attachment detection from the **fstab** file, new ECSs may fail to start.

7.5.3 How Do I Import an OVF or OVA File to the Cloud Platform?

Scenarios

Open Virtualization Appliance (OVA) is a single file (with the .ova extension) that archives all the files making up an Open Virtualization Format (OVF). OVF is a folder that contains the files required for defining and deploying VMs. An OVF folder always includes .ovf, .mf, and .vmdk files.

- An .ovf file is an XML descriptor that defines metadata of a VM, such as the name and hardware requirements, and contains reference information about other files in the OVF folder.
- An .mf file contains the SHA hash codes of all the files in the folder and is used to prevent the image file from being tampered with.
- A .vmdk file is a virtual disk file that is used to create a disk image. An OVF folder may contain multiple .vmdk files.

This section describes how to import OVF and OVA files to the cloud platform.

Procedure

Manually extract VMDK files from an OVF or OVA template and upload them to an OBS bucket. Then, you can select one from the bucket when you use an external file to create a system or data disk image.

NOTE

The following assumes that the OVF or OVA template contains only one VMDK file. If there are multiple VMDK files (for example, there are three VMDK files, one used as a system disk image file and the others as data disk image files), upload them to an OBS bucket and register them as a system disk image and data disk images, respectively.

- The source VM runs the Windows OS.
  - If you choose to export an OVF template named **MyVm** and save it to the **OvfLib** folder in drive C, the following files will be generated in the folder (the VMDK file can be uploaded to the cloud platform):

```
C
  OvfLib
    MyVm
      MyVm.ovf
    MyVm.mf
    MyVm-disk1.vmdk
```

  - If you choose to export an OVA template and name it **MyVm**, the **C:\MyVm.ova** file will be generated. The VMDK file extracted from **MyVm.ova** can be uploaded to the cloud platform.

    NOTE

    You can import an image file in the VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format to create a private image.

  For details, see **Creating a Windows System Disk Image from an External Image File** or **Creating a Data Disk Image from an External Image File**.

- The source VM runs the Linux OS.
  - If you choose to export an OVF template, upload the VMDK file generated in the folder to the cloud platform.
  - If you choose to export an OVA template and name it **MyVm**, perform the following operations:

    i. Run the following command to view the OVA file:

    file MyVm.ova

    The command output is as follows:

```
MyVm.ova: POSIX tar archive (GNU)
```

    MyVm.ova contains the following two files:

```
$tar tf MyVm.ova
MyVm.ovf
MyVm.vmdk
```

    ii. Run the following command to decompress **MyVm.ova**:

    tar xvf MyVm.ova

    The extracted folder contains the following files:

```
MyVm.ovf
MyVm.vmdk
```

    The image file in the VMDK format can be uploaded to the cloud platform.

    NOTE

    You can import an image file in the VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, QED, ZVHD, or ZVHD2 format to create a private image.

  For details, see **Creating a Linux System Disk Image from an External Image File** or **Creating a Data Disk Image from an External Image File**.

7.5.4 What Do I Do If I Configured an Incorrect OS or System Disk Capacity During Private Image Registration Using an Image File?

If you selected an incorrect OS, ECSs may fail to be created from the private image. If the configured system disk capacity is less than the one in the image file, image registration will fail.

In such cases, delete the incorrect image and create a new one using correct parameter settings.

7.5.5 What Do I Do If the System Disk Capacity in a VHD Image File Exceeds the One I Have Specified on the Management Console When I Use This File to Register a Private Image?

The possible causes may be:

1. You have specified a small value.

   Check the system disk capacity in the VHD image file. Specify a value no less than this value when you use the VHD image file to register an image.

2. After being converted using **qemu-img** or a similar tool, the VHD's virtual disk size becomes smaller than the actual system disk size. For details, see **https://bugs.launchpad.net/qemu/+bug/1490611**.

   Run the following command to check the VHD image file information:

   ```
   [xxxx@xxxxx test]$ qemu-img info 2g.vhd
   image: 2g.vhd
   file format: vpc
   virtual size: 2.0G (2147991552 bytes)
   disk size: 8.0K
   cluster_size: 2097152
   ```

   The virtual size is converted from the actual size (unit: byte) to an integer in GB. After the conversion, the output virtual size **2 GB** is smaller than the input actual size **2.0004 GB** (**2147991552 bytes**). You need to specify an integer larger than the actual size 2.0004 GB on the management console.

7.6 Image Exporting

7.6.1 Can I Download My Private Images to a Local PC?

Yes. You can download private images in VMDK, VHD, QCOW2, or ZVHD format as instructed in **Exporting an Image**.

7.6.2 Can I Use the System Disk Image of an ECS on a BMS After I Export It from the Cloud Platform?

No. The system disk image of an ECS is a VM file that contains a system running environment and does not have an installation boot program. Therefore, it cannot be used on a BMS.

7.6.3 Why Is the Image Size in an OBS Bucket Different from That Displayed in IMS?

Symptom

After a private image is exported to an OBS bucket, the image size in the bucket is different from that displayed in IMS. For example, the size of a private image is 1.04 GB on the IMS console. After it is exported to an OBS bucket, the size is displayed as 2.91 GB.

Cause Analysis

The size of an image in an OBS bucket varies depending on the file's storage format in the bucket.

7.6.4 Can I Download a Public Image to My Local PC?

Currently, you cannot directly download a public image. You can use the public image to create an ECS, use the ECS to create a private image, export the private image to your OBS bucket, and download the private image to your local PC.

Helpful links:

- **Creating a System Disk Image from a Windows ECS** or **Creating a System Disk Image from a Linux ECS**
- **Exporting an Image**

NOTE

- Windows, SUSE, Red Hat, Ubuntu, and Oracle Linux public images and the private images created from these public images cannot be exported.
- However, if a Windows, SUSE, Red Hat, Ubuntu, or Oracle Linux private image is created from an external image file, this private image can be exported.

7.6.5 What Are the Differences Between Import/Export and Fast Import/Export?

Item	Description	Helpful Link
Import	Import an external image file to the management console for creating a private image. External image files in the following formats can be imported: VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD. Maximum file size: 128 GB During the import, operations such as driver injection will be performed in the background. Therefore, the import takes a longer time than fast import.	• **Creating a Windows System Disk Image from an External Image File** • **Creating a Linux System Disk Image from an External Image File** • **Creating a Data Disk Image from an External Image File**

Item	Description	Helpful Link
Fast import	When importing an external image file in the RAW or ZVHD2 format to the management console, you can select **Enable Fast Create**. The system does not perform any operations such as driver injection. Verify that: • The image file converted to the RAW format has been optimized as required and a bitmap file has been generated for it. • The image file converted to the ZVHD2 format has been optimized as required. Maximum file size: 1 TB	**Quickly Importing an Image File**
Export	You can export private images to OBS buckets and download them to your local PC for further use on other cloud platforms. Maximum file size: 128 GB (If an image file is larger than 128 GB, use fast export to export it.) You can specify the format of the exported image file. Currently, only QCOW2, VMDK, VHD, and ZVHD are supported.	**Exporting an Image**
Fast export	On the **Export Image** page, select **Enable** following **Fast Export**. You cannot specify the format of the exported image file. After the export is complete, you can use a tool to convert the exported image to your desired format. The file size is not limited. Encrypted images do not support fast export.	**Exporting an Image**

7.6.6 What Do I Do If the Export Option Is Unavailable for My Image?

Some images cannot be exported. Therefore, the **Export** option is not provided for them in the **Operation** column. The following images cannot be exported:

- Public images
- Full-ECS images

- ISO images
- Private images created from a Windows or SUSE public image

7.7 Image Optimization

7.7.1 Must I Install Guest OS Drivers on an ECS?

Installing Guest OS drivers on an ECS improves your experience in using the ECS. In addition, it also ensures high reliability and stability of ECSs.

- Windows ECSs: Install VirtIO drivers on ECSs.
- Linux ECSs: Install VirtIO drivers and add them to initrd.

7.7.2 Why Do I Need to Install and Update VirtIO Drivers for Windows?

Why Do I Need to Install VirtIO Drivers?

VirtIO drivers are paravirtualized drivers that provide high-performance disks and NICs for ECSs.

- A standard Windows OS does not have VirtIO drivers.
- Public images have VirtIO drivers by default.
- You need to install VirtIO drivers for private images. For details, see **Installing VirtIO Drivers**.

Why Do I Need to Update VirtIO Drivers?

This ensures that known issues identified in the community or R&D tests can be avoided on the latest drivers.

When Do I Need to Update VirtIO Drivers?

After a major error is fixed, you are advised to update VirtIO drivers immediately. (This has not happened by now.)

After other issues are fixed, decide whether to update VirtIO drivers based on your needs.

What Do I Need to Do?

- Upgrade VirtIO drivers in Windows private images or running Windows ECSs.
- If you have any technical issue or question, contact the customer service.

7.7.3 What Will the System Do to an Image File When I Use the File to Register a Private Image?

You are advised to enable automatic configuration when registering a private image using an image file. Then, the system will perform the following operations:

Linux

- Check whether any PV drivers exist. If yes, the system deletes them.
- Modify the **grub** and **syslinux** configuration files to add the OS kernel boot parameters and change the disk partition name (**UUID=***UUID of the disk partition*).
- Change the names of the disk partitions in the **/etc/fstab** file (**UUID=***UUID of the disk partition*).
- Check whether the initrd file has IDE driver. If no, the system will load the IDE driver.
- Modify the X Window configuration file **/etc/X11/xorg.conf** to prevent display failures.
- Delete services of VMware tools.
- Record the latest automatic modification made to the image into **/var/log/rainbow_modification_record.log**.
- Copy built-in VirtIO drivers to **initrd** or **initramfs**. For details, see **External Image File Formats and Supported OSs**.

NOTE

For the following image files, the system does not copy this driver after **Enable automatic configuration** is selected:

- Image files whose **/usr** directory is an independent partition
- Fedora 29 64bit, Fedora 30 64bit, and CentOS 8.0 64bit image files that use the XFS file system
- SUSE 12 SP4 64bit image files that use the ext4 file system

Windows

- Restore the IDE driver so that the OS can use this driver for its initial start.
- Delete the registry keys of the mouse and keyboard and generate the registry keys again to ensure that the mouse and keyboard are available on the new cloud platform.
- Inject VirtIO drivers offline so that the OS can start properly.
- Restore DHCP. The OS will dynamically obtain information such as the IP address based on the DHCP protocol.

7.7.4 How Do I Configure an ECS or Image File Before I Use It to Create an Image?

ECS or Image File Configurations

Table 7-4 ECS configurations

OS	Configuration	Reference
Windows	• Set the NIC to DHCP. • Enable remote desktop connection. • (Optional) Install Cloudbase-Init. • Install Guest OS drivers (VirtIO drivers). • Run Sysprep.	**Creating a System Disk Image from a Windows ECS**
Linux	• Set the NIC to DHCP. • (Optional) Install Cloud-Init. • Delete files from the network rule directory. • Change the disk identifier in the GRUB configuration file to UUID. • Change the disk identifier in the fstab file to UUID. • Install native KVM drivers. • Detach data disks from the ECS.	**Creating a System Disk Image from a Linux ECS**

Table 7-5 Image file configurations

OS	Configuration	Reference
Windows	• Set the NIC to DHCP. • Enable remote desktop connection. • Install Guest OS drivers (VirtIO drivers). • (Optional) Install Cloudbase-Init. • (Optional) Enable NIC multi-queue. • (Optional) Configure an IPv6 address.	**Preparing an Image File**

OS	Configuration	Reference
Linux	• Delete files from the network rule directory. • Set the NIC to DHCP. • Install native KVM drivers. • Change the disk identifier in the GRUB configuration file to UUID. • Change the disk identifier in the fstab file to UUID. • Delete the automatic attachment information of non-system disks from the **/etc/fstab** file. • (Optional) Install Cloud-Init. • (Optional) Enable NIC multi-queue. • (Optional) Configure an IPv6 address.	**Preparing an Image File**

NOTE

- When registering an external image file as a private image, you are advised to perform the preceding operations on the VM where the external image file is located.
- When registering a Windows external image file as a private image, if the Guest OS drivers are installed, the cloud platform will check the image file after you select **Enable automatic configuration**. If the GuestOS drivers are not installed, the cloud platform will try to install them.

7.7.5 What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?

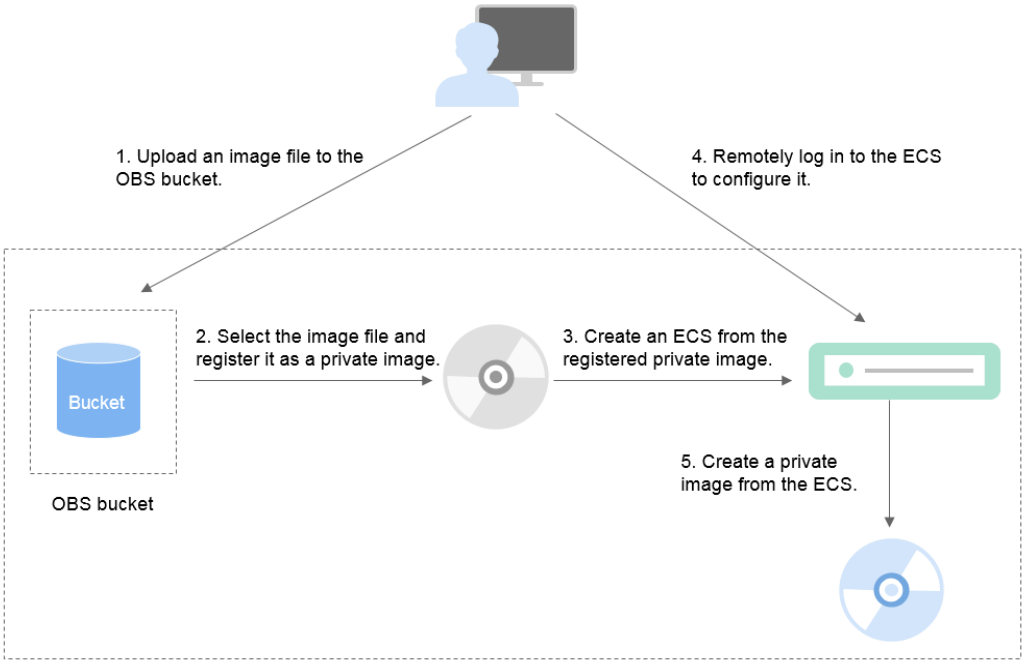
If an image file is not configured as instructed in **Table 2-5** before it is exported from the original platform, you can use it to create an ECS, configure the ECS, and use the ECS to create a private image. **Figure 7-8** shows the process.

CAUTION

An ECS can run properly only after KVM Guest OS drivers (VirtIO drivers) are installed on it. Without these drivers, the performance of this ECS will be affected and some functions will be unavailable. Ensure that the driver installation has been completed for the image file before it is exported from the original platform. Otherwise, the ECSs created from the image will fail to start.

Install VirtIO drivers. For details, see **Installing VirtIO Drivers**.

Figure 7-8 Image creation process



Step 1: Upload the Image File

Upload the external image file to an OBS bucket. For details, see **Uploading an External Image File**.

Step 2 Register the Image File as a Private Image

On the management console, select the uploaded image file and register it as a private image. For details, see **Registering an External Image File as a Private Image**.

Step 3: Create an ECS

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab.
3. Locate the row that contains the private image and click **Apply for ECS** in the **Operation** column.
4. Set parameters as promoted to create an ECS. Pay attention to the following:
   - Bind an EIP to the ECS so that you can upload installation packages to the ECS or download installation packages from the ECS.
   - You must add inbound rules for security groups of the ECS to ensure that the ECS can be accessed.
   - If the image file has Cloudbase-Init installed, set a password and log in to the ECS using the password as prompted. If Cloudbase-Init is not

installed, use the password or certificate contained in the image file to log in the ECS.

For details, see *Elastic Cloud Server User Guide*.

5. Perform the following steps to check whether the private image has been pre-configured:

   a. Check whether the ECS can be successfully started. If the start succeeds, Guest OS drivers have been installed for the image file on the original platform or the drivers have been automatically installed for the private image on the cloud platform. If the start failed, install Guest OS drivers for the image file on the original platform and start from **Step 1: Upload the Image File** again.

   b. Check whether you can log in to the ECS using your configured password or key. If you can, Cloudbase-Init has been installed. If you cannot, use the password or key contained in the image file to log in to the ECS and install Cloudbase-Init as instructed in **Installing and Configuring Cloudbase-Init**.

   c. Check whether NICs are set to DHCP by referring to **2** in **Step 4: Configure the ECS**.

   d. Use MSTSC to log in to the ECS. If the login is successful, remote desktop connection is enabled on the ECS. If the login fails, enable remote desktop connection by referring to **3** in **Step 4: Configure the ECS**.

   If the ECS meets the preceding requirements, the private image has been pre-configured. Skip **Step 4: Configure the ECS** and **Step 5: Create a Private Image from the ECS**.

Step 4: Configure the ECS

Remotely log in to the ECS created in **Step 3: Create an ECS** to configure it.

1. Log in to the ECS.
2. Check whether NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in **Setting the NIC to DHCP**.
3. Enable remote desktop connection for the ECS as needed. For details about how to enable this function, see **Enabling Remote Desktop Connection**.
4. (Optional) Configure value-added functions.
   - Install and configure Cloudbase-Init. For details, see **Installing and Configuring Cloudbase-Init**.
   - Enable NIC multi-queue. For details, see **How Do I Enable NIC Multi-Queue for an Image?**
   - Configure an IPv6 address. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Step 5: Create a Private Image from the ECS

For details, see **Creating a System Disk Image from a Windows ECS**.

(Optional) Clear the Environment

After the image registration is complete, delete the image file as well as the intermediate private image and ECS to prevent them from occupying storage and compute resources.

- Delete the image registered in **Step 2 Register the Image File as a Private Image**.
- Delete the ECS created in **Step 3: Create an ECS**.
- Delete the image file from the OBS bucket.

7.7.6 What Do I Do If a Linux Image File Is Not Pre-Configured When I Use It to Register a Private Image?

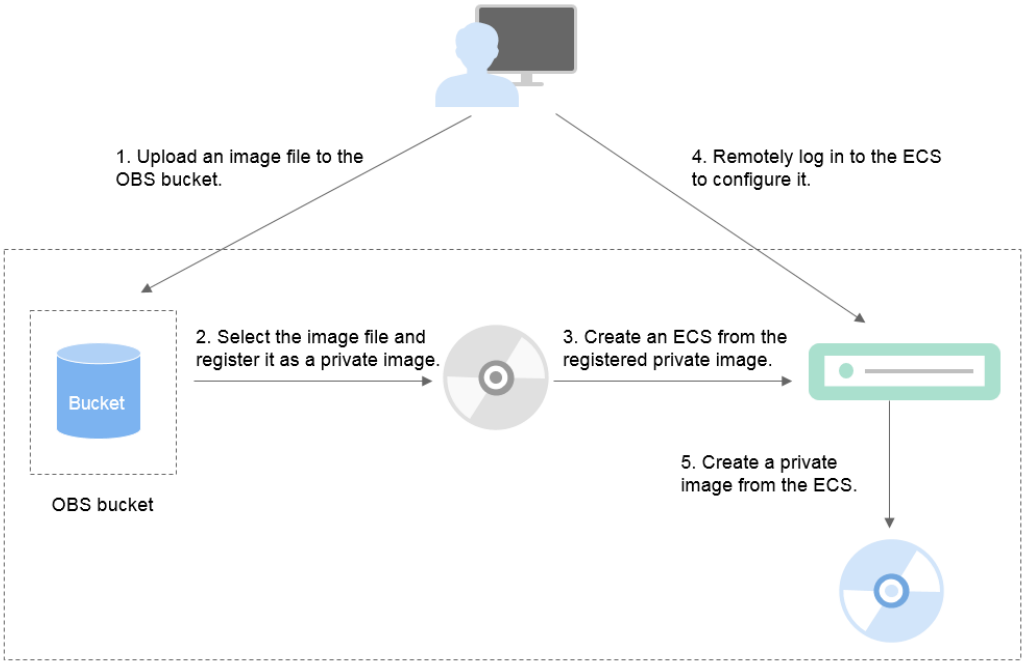
If an image file is not configured as instructed in **Table 2-9** before it is exported from the original platform, you can use it to create an ECS, configure the ECS, and use the ECS to create a private image. **Figure 7-9** shows the process.

CAUTION

An ECS can run properly only after KVM drivers are installed on it. If no such drivers are installed, the performance of the ECS will be affected and some functions will be unavailable. Ensure that KVM drivers have been installed for the image file before it is exported from the original platform. Otherwise, the ECSs created from the image will fail to start.

For details, see **Installing Native KVM Drivers**.

Figure 7-9 Image creation process



Step 1: Upload the Image File

Upload the external image file to an OBS bucket. For details, see **Uploading an External Image File**.

Step 2 Register the Image File as a Private Image

On the management console, select the uploaded image file and register it as a private image. For details, see **Registering an External Image File as a Private Image**.

Step 3: Create an ECS

Create an ECS from the private image.

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab.
3. Locate the row that contains the private image and click **Apply for ECS** in the **Operation** column.
4. Set parameters as promoted to create an ECS. Pay attention to the following:
   - You must add inbound rules for security groups of the ECS to ensure that the ECS can be accessed.
   - If Cloud-Init has been installed in the image file, set a login password as prompted. If Cloud-Init is not installed, use the password or certificate contained in the image file to log in.

   For details, see *Elastic Cloud Server User Guide*.
5. Perform the following steps to check whether the private image has been pre-configured:
   a. Check whether the ECS can be successfully started. If the start succeeds, KVM drivers have been installed for the external image file on the original platform or the drivers have been automatically installed for the private image on the cloud platform. If the start failed, install KVM drivers for the image file and start from **Step 1: Upload the Image File** again.
   b. Check whether you can log in to the ECS using your configured password or key. If you can, Cloud-Init has been installed. If you cannot, use the password or key contained in the image file to log in to the ECS and install Cloud-Init as instructed in **Installing Cloud-Init**.
   c. Check the network configuration by referring to **Step 4: Configure the ECS**.

   If the ECS meets the preceding requirements, the private image has been pre-configured. Skip **Step 4: Configure the ECS** and **Step 5: Create a Private Image from the ECS**.

Step 4: Configure the ECS

Remotely log in to the ECS created in **Step 3: Create an ECS** to configure it.

1. Log in to the ECS.
2. Configure the network.
   - Run the **ifconfig** command to check whether the private IP address of the ECS is the same as that displayed on the console. If they are inconsistent, delete files from the network rule directory as instructed in **Deleting Files from the Network Rule Directory**.
   - Check whether NICs are set to DHCP. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in **Setting the NIC to DHCP**.
   - Run the **service sshd status** command to check whether SSH is enabled. If it is disabled, run the **service sshd start** command to enable it. Ensure that your firewall (for example, Linux iptables) allows SSH access.
3. Configure a file system.
   - Change the disk identifier in the GRUB configuration file to UUID. For details, see **Changing the Disk Identifier in the GRUB Configuration File to UUID**.
   - Change the disk identifier in the fstab file to UUID. For details, see **Changing the Disk Identifier in the fstab File to UUID**.
   - Clear the automatic attachment information of non-system disks in the **/etc/fstab** file to prevent impacts on subsequent data disk attachment. For details, see **Detaching Data Disks from an ECS**.
4. (Optional) Configure value-added functions.
   - Install and configure Cloud-Init. For details, see **Installing Cloud-Init** and **Configuring Cloud-Init**.
   - Enable NIC multi-queue. For details, see **How Do I Enable NIC Multi-Queue for an Image?**
   - Configure an IPv6 address. For details, see **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**

Step 5: Create a Private Image from the ECS

Create a private image from the ECS. For details, see **Creating a System Disk Image from a Linux ECS**.

(Optional) Clear the Environment

After the image registration is complete, delete the image file as well as the intermediate private image and ECS to prevent them from occupying storage and compute resources.

- Delete the image registered in **Step 2 Register the Image File as a Private Image**.
- Delete the ECS created in **Step 3: Create an ECS**.
- Delete the image file from the OBS bucket.

7.7.7 How Do I Enable NIC Multi-Queue for an Image?

Scenarios

With the increase of network I/O bandwidth, a single vCPU cannot meet the requirement of processing NIC interruptions. NIC multi-queue allows multiple vCPUs to process NIC interruptions, thereby improving network PPS and I/O performance.

ECSs Supporting NIC Multi-Queue

NIC multi-queue can be enabled on an ECS only when the ECS specifications, virtualization type, and image meet the requirements described in this section.

- For details about the ECS flavors that support NIC multi-queue, see section "Instances" in *Elastic Cloud Server User Guide*.

  NOTE

  If the number of NIC queues is greater than 1, NIC multi-queue is supported.
- Only KVM ECSs support NIC multi-queue.
- The Linux public images listed in **Table 7-7** support NIC multi-queue.

  NOTE

  - Windows OSs have not commercially supported NIC multi-queue. If you enable NIC multi-queue for a Windows image, starting an ECS created using such an image may be slow.
  - You are advised to upgrade the kernel version of Linux ECSs to 2.6.35 or later. Otherwise, NIC multi-queue is not supported.

    Run the **uname -r** command to check the kernel version. If the version is earlier than 2.6.35, contact technical support to upgrade it.

Table 7-6 Windows ECSs that support NIC multi-queue

OS	Image	Supported By
Windows	Windows Server 2008 WEB R2 64bit	Private images
	Windows Server 2008 Enterprise SP2 64bit	Private images
	Windows Server 2008 R2 Standard/ Datacenter/Enterprise 64bit	Private images
	Windows Server 2008 R2 Enterprise 64bit_WithGPUdriver	Private images
	Windows Server 2012 R2 Standard 64bit_WithGPUdriver	Private images
	Windows Server 2012 R2 Standard/ Datacenter 64bit	Private images

Table 7-7 Linux ECSs that support NIC multi-queue

OS	Image	Supported By	NIC Multi-Queue Enabled by Default
Linux	Ubuntu 14.04/16.04 Server 64bit	Public images	Yes
	openSUSE 42.2 64bit	Public images	Yes
	SUSE Enterprise 12 SP1/SP2 64bit	Public images	Yes
	CentOS 6.8/6.9/7.0/7.1/7.2/7.3/7.4/7.5/7.6 64bit	Public images	Yes
	Debian 8.0.0/8.8.0/8.9.0/9.0.0 64bit	Public images	Yes
	Fedora 24/25 64bit	Public images	Yes
	EulerOS 2.2 64bit	Public images	Yes

Operation Instructions

Assume that an ECS has the required specifications and virtualization type.

- If the ECS was created using a public image listed in **ECSs Supporting NIC Multi-Queue**, NIC multi-queue has been enabled on the ECS by default. Therefore, you do not need to manually enable NIC multi-queue for it.
- If the ECS was created using an external image file with an OS listed in **ECSs Supporting NIC Multi-Queue**, perform the following operations to enable NIC multi-queue:
  a. **Register the External Image File as a Private Image**.
  b. **Set NIC Multi-Queue for the Image**.
  c. **Create an ECS from the Private Image**.
  d. **Enable NIC Multi-Queue**.

Register the External Image File as a Private Image

Register the external image file as a private image. For details, see **Registering an External Image File as a Private Image**.

Set NIC Multi-Queue for the Image

Windows OSs have not commercially supported NIC multi-queue. If you enable NIC multi-queue for a Windows image, starting an ECS created using such an image may be slow.

Use either of the following methods to set NIC multi-queue.

Method 1:

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. On the displayed **Private Images** page, locate the row that contains the target image and click **Modify** in the **Operation** column.
3. Set NIC multi-queue for the image.

Method 2:

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. On the displayed **Private Images** page, click the name of the target image.
3. In the upper right corner of the displayed image details page, click **Modify**. In the displayed **Modify Image** dialog box, set NIC multi-queue for the image.

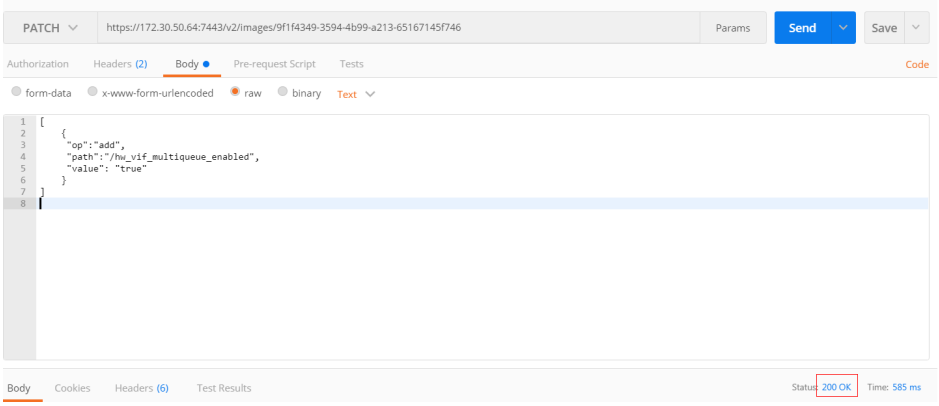
Method 3: Add **hw_vif_multiqueue_enabled** to the image using an API.

1. Obtain a token. For details, see **Calling APIs** > **Authentication** in *Image Management Service API Reference*.
2. Call an API to update image information. For details, see "Updating Image Information (Native OpenStack API)" in *Image Management Service API Reference*.
3. Add **X-Auth-Token** to the request header.
   The value of **X-Auth-Token** is the token obtained in step **1**.
4. Add **Content-Type** to the request header.
   The value of **Content-Type** is **application/openstack-images-v2.1-json-patch**.
   The request URI is in the following format:
   PATCH /v2/images/{*image_id*}
   The request body is as follows:

```
[
      {
       "op":"add",
       "path":"/hw_vif_multiqueue_enabled",
       "value": "true"
      }
]
```

Figure 7-10 shows an example request body for setting NIC multi-queue.

Figure 7-10 Example request body



Create an ECS from the Private Image

Use the registered private image to create an ECS. For details, see the *Elastic Cloud Server User Guide*. Note the following when setting the parameters:

- **Region**: Select the region where the private image is located.
- **Image**: Select **Private image** and then the desired image from the drop-down list.

Enable NIC Multi-Queue

KVM ECSs running Windows use private images to support NIC multi-queue.

For Linux ECSs, which run CentOS 7.4 as an example, perform the following operations to enable NIC multi-queue:

Step 1 Enable NIC multi-queue.

1. Log in to the ECS.
2. Run the following command to obtain the number of queues supported by the NIC and the number of queues with NIC multi-queue enabled:

   ethtool -l *NIC*
3. Run the following command to configure the number of queues used by the NIC:

   ethtool -L *NIC* **combined** *Number of queues*

Example:

```
[root@localhost ~]# ethtool -l eth0  #View the number of queues used by NIC eth0.
Channel parameters for eth0:
Pre-set maximums:
RX:             0
TX:             0
Other:          0
Combined: 4  #Indicates that a maximum of four queues can be enabled for the NIC.
Current hardware settings:
RX:             0
TX:             0
Other:          0
Combined: 1 #Indicates that one queue has been enabled.

[root@localhost ~]# ethtool -L eth0 combined 4 #Enable four queues on NIC eth0.
```

Step 2 (Optional) Enable irqbalance so that the system automatically allocates NIC interruptions to multiple vCPUs.

1. Run the following command to enable irqbalance:

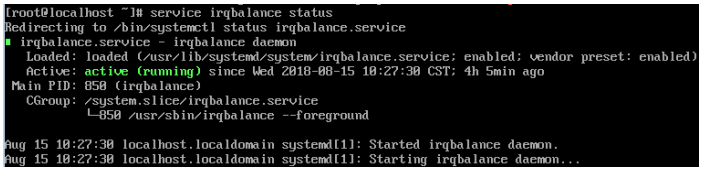
   service irqbalance start

2. Run the following command to view the irqbalance status:

   service irqbalance status

   If the **Active** value in the command output contains **active (running)**, irqbalance has been enabled.

   Figure 7-11 Enabled irqbalance



Step 3 (Optional) Enable interrupt binding.

Enabling irqbalance allows the system to automatically allocate NIC interruptions, improving network performance. If the improved network performance fails to meet your expectations, manually configure interrupt affinity on the target ECS.

The detailed operations are as follows:

Run the following script so that each ECS vCPU responds the interrupt requests initialized by one queue. That is, one queue corresponds to one interrupt, and one interrupt binds to one vCPU.

```
#!/bin/bash
service irqbalance stop

eth_dirs=$(ls -d /sys/class/net/eth*)
if [ $? -ne 0 ];then
    echo "Failed to find eth*  , sleep 30" >> $ecs_network_log
    sleep 30
    eth_dirs=$(ls -d /sys/class/net/eth*)
fi

for eth in $eth_dirs
do
    cur_eth=$(basename $eth)
    cpu_count=`cat /proc/cpuinfo| grep "processor"| wc -l`
    virtio_name=$(ls -l /sys/class/net/"$cur_eth"/device/driver/ | grep pci |awk {'print $9'})

    affinity_cpu=0
    virtio_input="$virtio_name""-input"
    irqs_in=$(grep "$virtio_input" /proc/interrupts | awk -F ":" '{print $1}')
    for irq in ${irqs_in[*]}
    do
        echo $((affinity_cpu%cpu_count)) > /proc/irq/"$irq"/smp_affinity_list
        affinity_cpu=$[affinity_cpu+2]
    done

    affinity_cpu=1
    virtio_output="$virtio_name""-output"
    irqs_out=$(grep "$virtio_output" /proc/interrupts | awk -F ":" '{print $1}')
    for irq in ${irqs_out[*]}
    do
        echo $((affinity_cpu%cpu_count)) > /proc/irq/"$irq"/smp_affinity_list
        affinity_cpu=$[affinity_cpu+2]
```

```
    done
done
```

Step 4 (Optional) Enable XPS and RPS.

XPS allows the system with NIC multi-queue enabled to select a queue by vCPU when sending a data packet.

```
#!/bin/bash
# enable XPS feature
cpu_count=$(grep -c processor /proc/cpuinfo)
dec2hex(){
  echo $(printf "%x" $1)
}
eth_dirs=$(ls -d /sys/class/net/eth*)
if [ $? -ne 0 ];then
    echo "Failed to find eth* , sleep 30" >> $ecs_network_log
    sleep 30
    eth_dirs=$(ls -d /sys/class/net/eth*)
fi
for eth in $eth_dirs
do
    cpu_id=1
    cur_eth=$(basename $eth)
    cur_q_num=$(ethtool -l $cur_eth | grep -iA5 current | grep -i combined | awk {'print $2'})
    for((i=0;i<cur_q_num;i++))
    do
        if [ $i -eq $ cpu_count ];then
            cpu_id=1
        fi
        xps_file="/sys/class/net/${cur_eth}/queues/tx-$i/xps_cpus"
        rps_file="/sys/class/net/${cur_eth}/queues/rx-$i/rps_cpus"
        cpuset=$(dec2hex "$cpu_id")
        echo $cpuset > $xps_file
        echo $cpuset > $rps_file
        let cpu_id=cpu_id*2
    done
done
```

----End

7.7.8 How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?

Scenarios

IPv6 addresses are used to deal with IPv4 address exhaustion. If an ECS uses an IPv4 address, the ECS can run in dual-stack mode after IPv6 is enabled for it. Then, the ECS will have two IP addresses to access the intranet and Internet: an IPv4 address and an IPv6 address.

In some cases, an ECS cannot dynamically acquire an IPv6 address even if it meets all the requirements in **Constraints**. You need to configure the ECS to dynamically acquire IPv6 addresses. For public images:

- By default, dynamic IPv6 address assignment is enabled for Windows public images. You do not need to configure it. The operations in **Windows Server 2012** and **Windows Server 2008** are for your reference only.
- Before enabling dynamic IPv6 address assignment for a Linux public image, check whether IPv6 has been enabled and then whether dynamic IPv6 address assignment has been enabled. Currently, IPv6 is enabled for all Linux public images.

Constraints

- Ensure that IPv6 has been enabled on the subnet where the ECS works.

  If IPv6 is not enabled on the subnet, enable it by referring to **Enabling IPv6 for an ECS**. IPv6 cannot be disabled once it is enabled.
- Ensure that **Self-assigned IPv6 address** is selected during ECS creation.
- After the ECS is started, its hot-swappable NICs cannot automatically acquire IPv6 addresses.
- Only ECSs can work in dual-stack mode and BMSs cannot.
- Only one IPv6 address can be bound to a NIC.

Procedure

- Windows: Windows Server 2012/2008 is used as an example to describe how to enable dynamic assignment of IPv6 addresses in Windows.
- Linux: Dynamic assignment of IPv6 addresses can be enabled automatically (recommended) or manually.

  If a private image created from a CentOS 6.x or Debian ECS with automatic IPv6 address assignment enabled is used to create an ECS in an environment that does not support IPv6, the ECS may start slow because of IPv6 address assignment timeout. You can set the timeout duration for assigning IPv6 addresses by referring to **Setting the Timeout Duration for IPv6 Address Assignment**.

Table 7-8 Enabling dynamic assignment of IPv6 addresses for different OSs

OS	Automatically/ Manually Enabling	Reference
Windows Server 2012	Automatically	**Windows Server 2012**
Windows Server 2008	Automatically	**Windows Server 2008**
Linux	Automatically (recommended)	**Linux (Automatically Enabling Dynamic Assignment of IPv6 Addresses)**
Linux	Manually	**Linux (Manually Enabling Dynamic Assignment of IPv6 Addresses)**

Enabling IPv6 for an ECS

NOTE

After IPv6 is enabled on the subnet where the ECS works, an IPv6 CIDR block is automatically assigned to the subnet. IPv6 cannot be disabled once it is enabled.

1. Log in to the management console.

2. Under **Computing**, click **Elastic Cloud Server**.
3. Click the target ECS to go to the detail page.
4. In the **ECS Information** area, click the VPC name.
5. Click the number in the **Subnets** column.

   The **Subnets** page is displayed.
6. In the subnet list, locate the target subnet and click its name.

   The subnet details page is displayed.
7. In the **Subnet Information** area, click **Enable** for **IPv6 CIDR Block**.
8. Click **Yes**.

Windows Server 2012

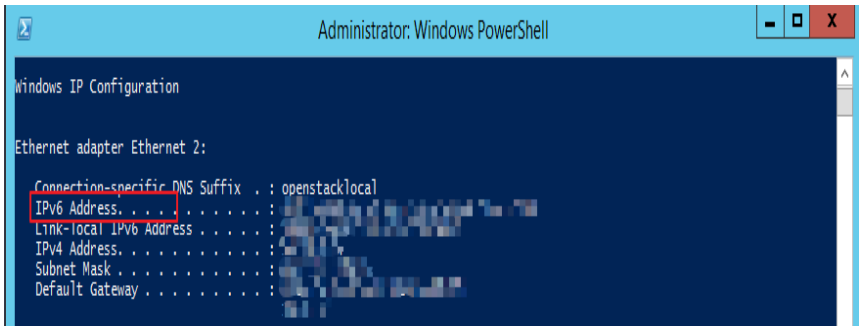
Step 1 Check whether IPv6 is enabled for the ECS.

Run the following command in the CMD window to check it:

ipconfig

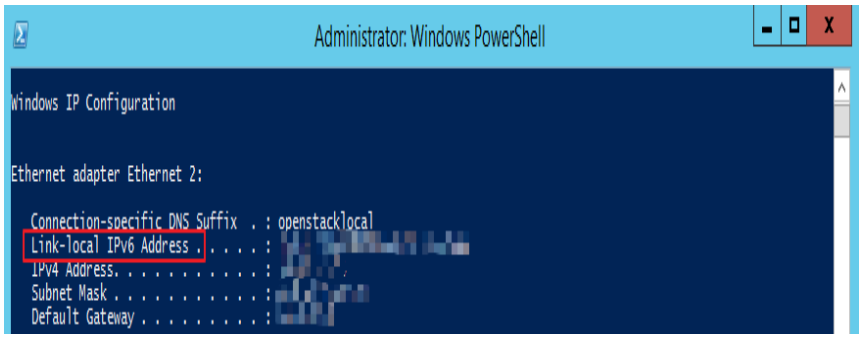
- If an IPv6 address and a link-local IPv6 address are displayed, IPv6 is enabled and dynamic IPv6 assignment is also enabled.

  Figure 7-12 Querying the IPv6 address

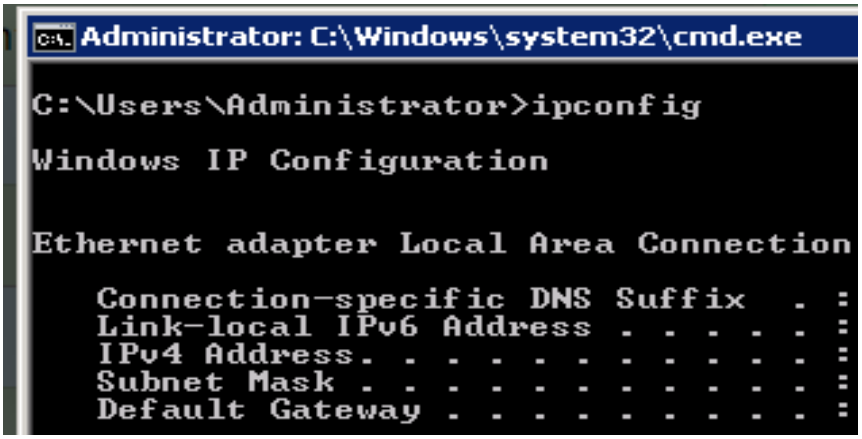
- If only a link-local IPv6 address is displayed, IPv6 is enabled but dynamic IPv6 assignment is not enabled. Go to **Step 2**.

  Figure 7-13 Link-local IPv6 address

- If neither an IPv6 address nor link-local IPv6 address is displayed, IPv6 is disabled. Go to **Step 3**.

Figure 7-14 IPv6 disabled



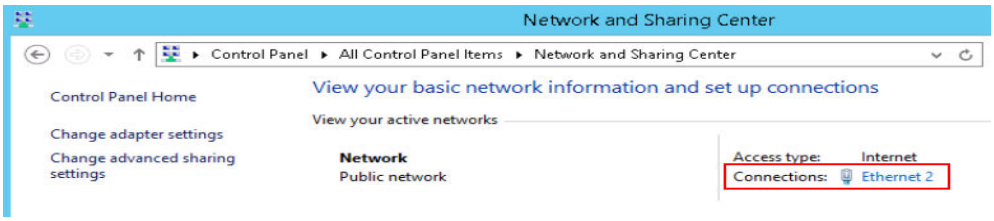
NOTE

By default, dynamic IPv6 address assignment is enabled for Windows public images, as shown in **Figure 7-12**. No additional configuration is required.

Step 2 Enable dynamic IPv6 address assignment.

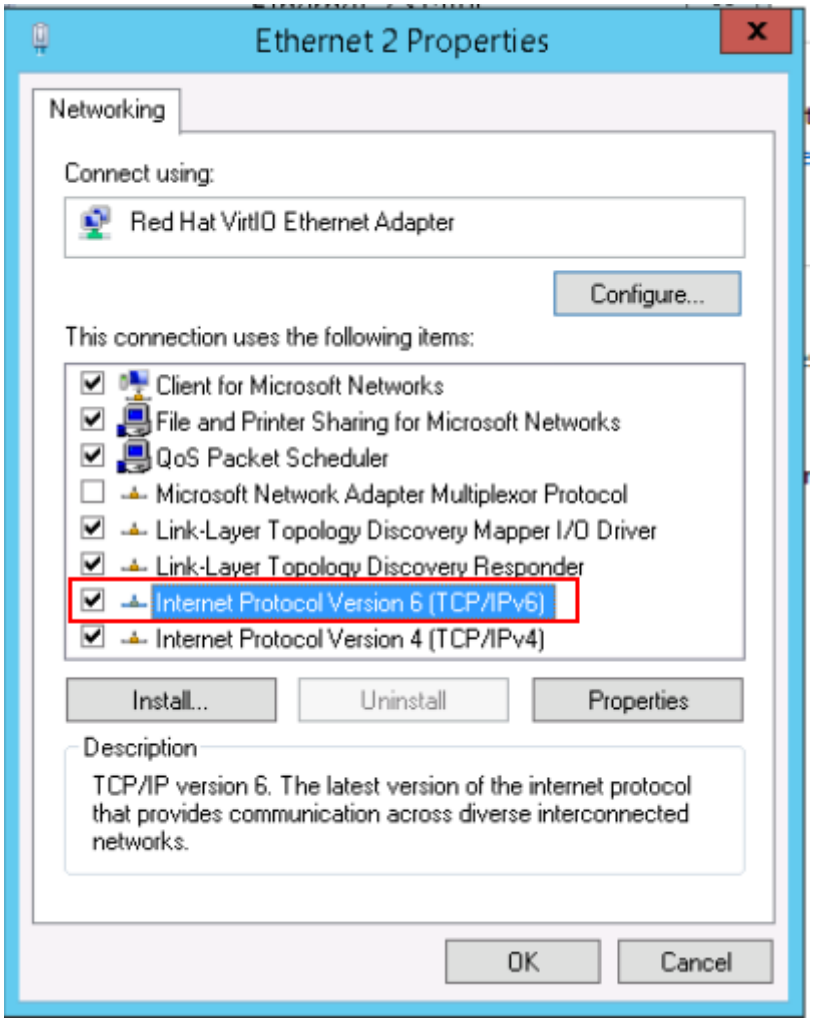
1. Choose **Start** > **Control Panel**.
2. Click **Network and Sharing Center**.
3. Click the Ethernet connection.

Figure 7-15 Ethernet connection



4. In the **Ethernet Status** dialog box, click **Properties** in the lower left corner.
5. Select **Internet Protocol Version 6 (TCP/IPv6)** and click **OK**.

Figure 7-16 Configuring dynamic IPv6 address assignment

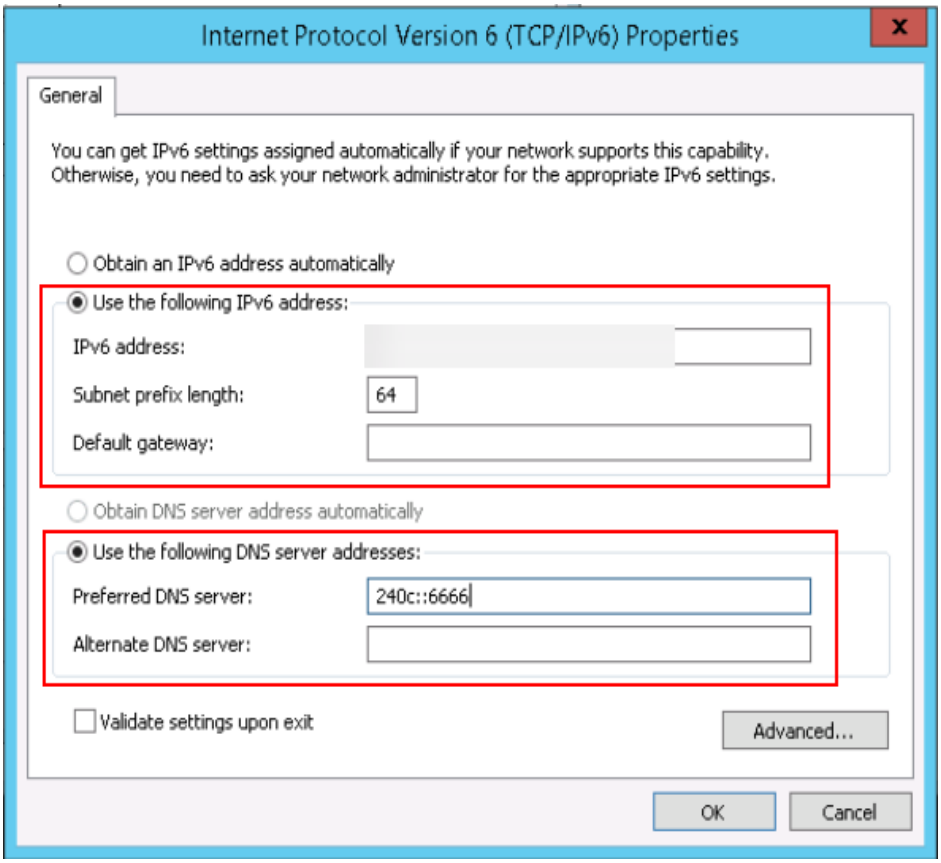


6. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

Step 3 Enable and configure IPv6.

1. In the **Internet Protocol Version 6 (TCP/IPv6) Properties** dialog box, configure an IPv6 address and a DNS server address.
   - **IPv6 address**: IPv6 address allocated during ECS creation. Obtain the value from the ECS list on the console.
   - **Subnet prefix length**: **64**
   - **Preferred DNS server**: **240c::6666** (recommended)

Figure 7-17 Configuring an IPv6 address and a DNS server address



2. (Optional) Run the following command depending on your ECS OS.

   For Windows Server 2012, run the following command in PowerShell or CMD:

   Set-NetIPv6Protocol -RandomizeIdentifiers disabled

3. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

----End

Windows Server 2008

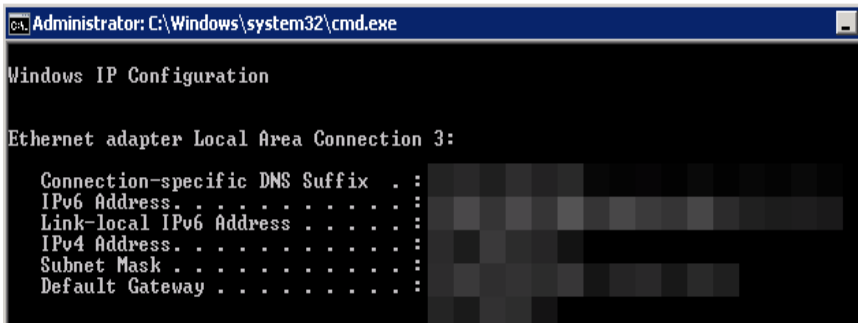
Step 1 Check whether IPv6 is enabled for the ECS.

Run the following command in the CMD window to check it:

ipconfig

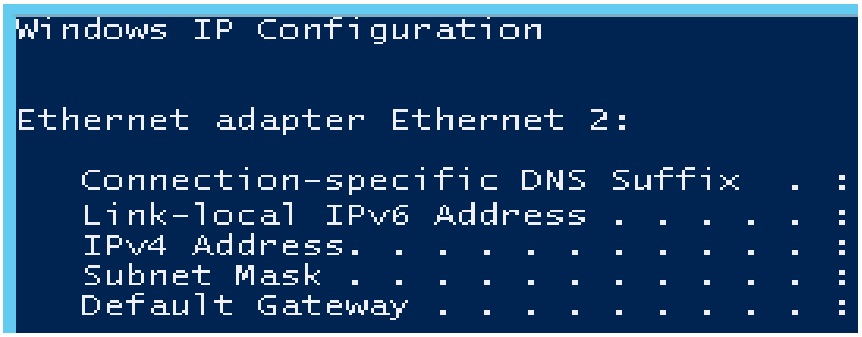
- If an IPv6 address and a link-local IPv6 address are displayed, IPv6 is enabled and dynamic IPv6 assignment is also enabled.

Figure 7-18 Querying the IPv6 address



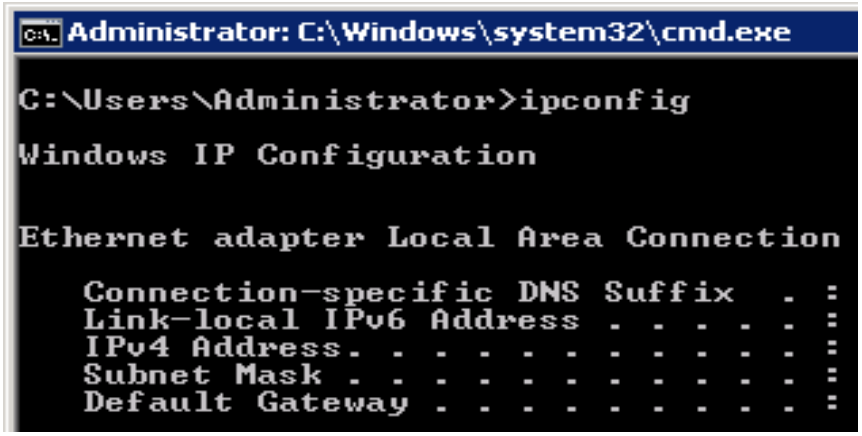
- If only a link-local IPv6 address is displayed, IPv6 is enabled but dynamic IPv6 assignment is not enabled. Go to **Step 2**.

Figure 7-19 Link-local IPv6 address



- If neither an IPv6 address nor link-local IPv6 address is displayed, IPv6 is disabled. Go to **Step 3**.

Figure 7-20 IPv6 disabled



NOTE

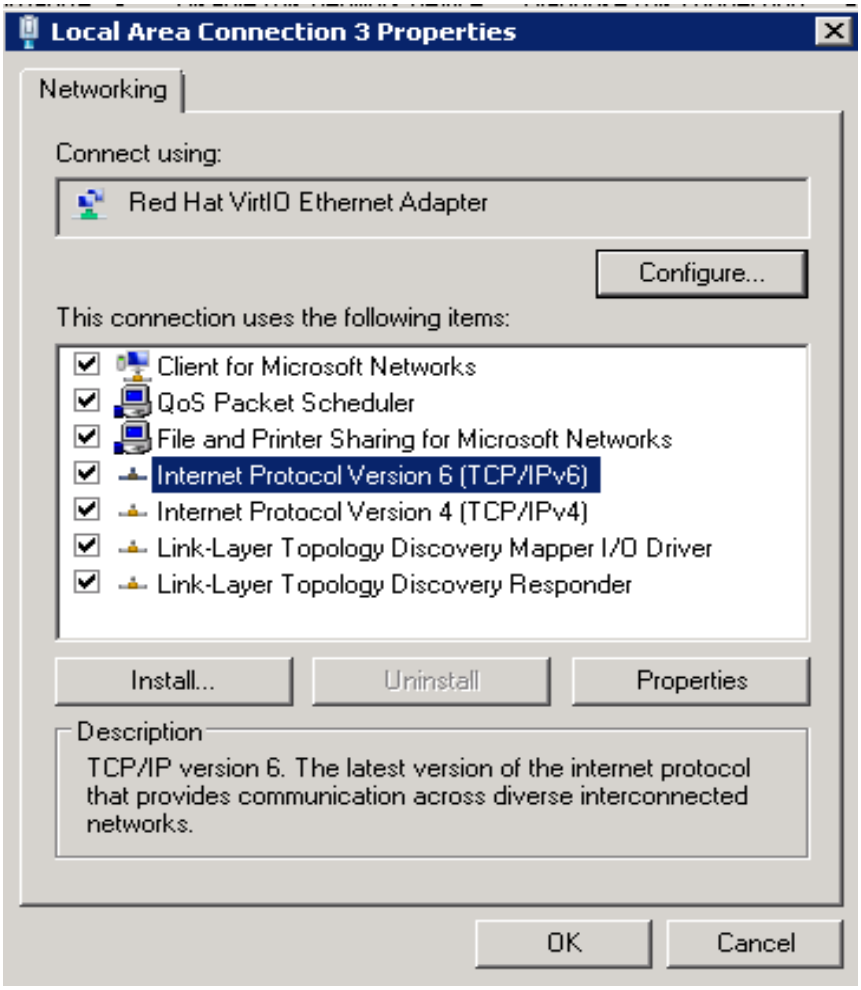
By default, dynamic IPv6 address assignment is enabled for Windows public images, as shown in **Figure 7-18**. No additional configuration is required.

Step 2 Enable dynamic IPv6 address assignment.

1. Choose **Start** > **Control Panel**.

2. Click **Network and Sharing Center**.
3. Click **Change adapter settings**.
4. Right-click the local network connection and choose **Properties**.
5. Select **Internet Protocol Version 6 (TCP/IPv6)** and click **OK**.

Figure 7-21 Configuring dynamic IPv6 address assignment

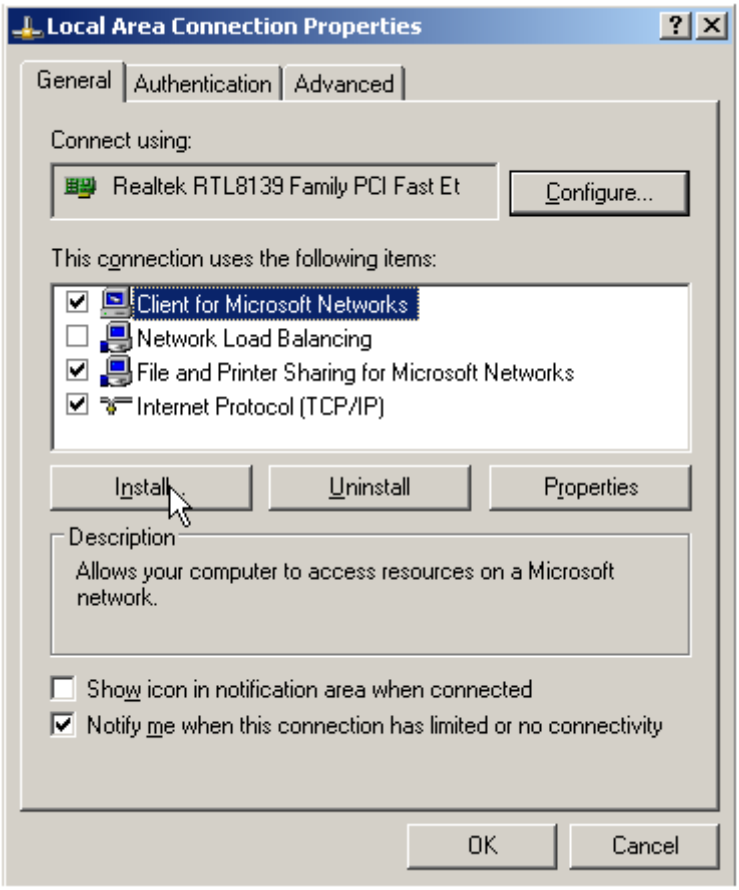


6. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

Step 3 Enable and configure IPv6.

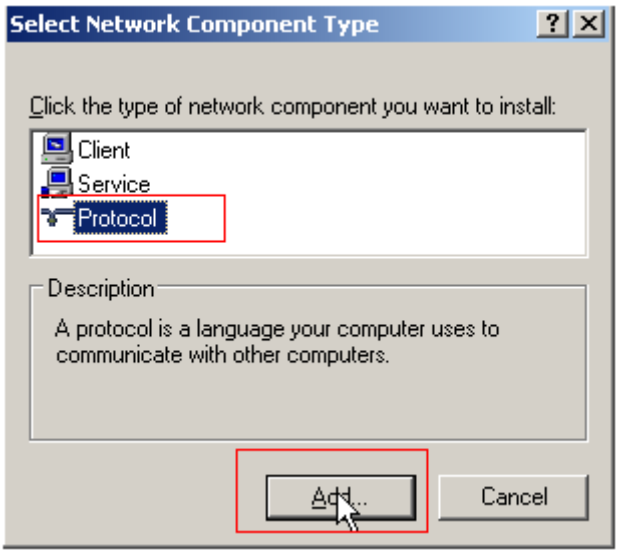
1. Choose **Start** > **Control Panel** > **Network Connection** > **Local Connection**.
2. Select **Properties**, select the following options, and click **Install**.

Figure 7-22 Enabling and configuring IPv6



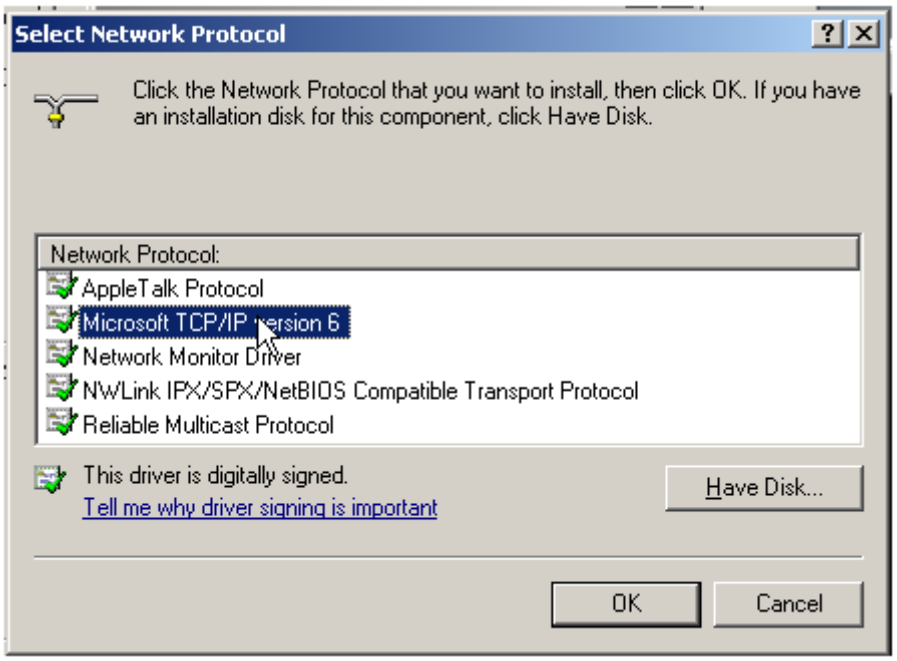
3. Select **Protocol** and click **Add**.

Figure 7-23 Adding the protocol



4. Select **Microsoft TCP/IP Version 6** and click **OK**.

Figure 7-24 Network protocols



5. (Optional) Run the following commands depending on your ECS OS.

   For Windows Server 2008, run the following command in PowerShell or CMD:

   netsh interface ipv6 set global randomizeidentifiers=disable

   Disable the local connection and then enable it again.

   To disable the local connection, choose **Start** > **Control Panel** > **Network and Internet** > **Network and Sharing Center** > **Change Adapter Options**. Right-click the local connection and choose **Disable** from the shortcut menu.

   To enable the local connection, choose **Start** > **Control Panel** > **Network and Internet** > **Network and Sharing Center** > **Change Adapter Options**. Right-click the local connection and choose **Enable** from the shortcut menu.

6. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

----End

Linux (Automatically Enabling Dynamic Assignment of IPv6 Addresses)

The **ipv6-setup-***xxx* tool can be used to enable Linux OSs to automatically acquire IPv6 addresses. *xxx* indicates a tool, which can be rhel or debian.

You can also enable dynamic IPv6 address assignment by following the instructions in **Linux (Manually Enabling Dynamic Assignment of IPv6 Addresses)**.

CAUTION

- When you run **ipv6-setup-***xxx*, the network service will be automatically restarted. As a result, the network is temporarily disconnected.
- If a private image created from a CentOS 6.x or Debian ECS with automatic IPv6 address assignment enabled is used to create an ECS in an environment that does not support IPv6, the ECS may start slow because of IPv6 address assignment timeout. Set the timeout duration for assigning IPv6 addresses to 30s by referring to **Setting the Timeout Duration for IPv6 Address Assignment** and try to create a new private image again.

Step 1 Run the following command to check whether IPv6 is enabled for the ECS:

ip addr

- If only an IPv4 address is displayed, IPv6 is disabled. Enable it by referring to **Setting the Timeout Duration for IPv6 Address Assignment**.

  Figure 7-25 IPv6 disabled

  ```
  eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
   link/ether fa:16:3e:          brd ff:ff:ff:ff:ff:ff
   inet                  brd                  scope global noprefixroute dynamic eth0
      valid_lft 1193sec preferred_lft 1193sec
  ```

- If a link-local address (starting with fe80) is displayed, IPv6 is enabled but dynamic assignment of IPv6 addresses is not enabled.

  Figure 7-26 IPv6 enabled

  ```
  eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether fa:16:3e:          brd ff:ff:ff:ff:ff:ff
   inet                  brd                  scope global noprefixroute dynamic eth0
      valid_lft 76391sec preferred_lft 76391sec
   inet6 fe80::f816:                /64 scope link
      valid_lft forever preferred_lft forever
  ```

- If the following address is displayed, IPv6 is enabled and an IPv6 address has been assigned:

  Figure 7-27 IPv6 enabled and an IPv6 address assigned

  ```
  eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
   link/ether fa:16:3e:75:af:4c brd ff:ff:ff:ff:ff:ff
   inet                  brd                  scope global noprefixroute dynamic eth0
      valid_lft 86395sec preferred_lft 86395sec
   inet6 2407:c080:802:                                  /128 scope global dynamic
      valid_lft 7496sec preferred_lft 7196sec
   inet6 fe80::f816:3eff:          /64 scope link noprefixroute
      valid_lft forever preferred_lft forever
  ```

NOTE

IPv6 is enabled for Linux public images by default, as shown in **Figure 7-26**.

Step 2 Enable IPv6 for the ECS.

1. Run the following command to check whether IPv6 is enabled for the kernel:

   sysctl -a | grep ipv6

   - If a command output is displayed, IPv6 is enabled.
   - If no information is displayed, IPv6 is disabled. Go to **Step 2.2** to load the IPv6 module.

2. Run the following command to load the IPv6 module:

   modprobe ipv6

3. Add the following content to the **/etc/sysctl.conf** file:

   net.ipv6.conf.all.disable_ipv6=0

4. Save the configuration and exit. Then, run the following command to load the configuration:

   sysctl -p

Step 3 Enable dynamic IPv6 address assignment for the ECS.

1. Download **ipv6-setup-rhel** or **ipv6-setup-debian** with a required version and upload it to the target ECS.

   ipv6-setup-*xxx* modifies the configuration file of a NIC to enable dynamic IPv6 address assignment or adds such a configuration file for a NIC, and then restarts the NIC or network service.

   Contact the administrator to obtain the download paths of **ipv6-setup-rhel** and **ipv6-setup-debian**.

2. Run the following command to make **ipv6-setup-***xxx* executable:

   chmod +x ipv6-setup-*xxx*

3. Run the following command to enable dynamic IPv6 address assignment for a NIC:

   ./ipv6-setup-*xxx* **--dev** [*dev*]

   Example:

   ./ipv6-setup-*xxx* **--dev eth0**

   NOTE

   - To enable dynamic IPv6 address assignment for all NICs, run the **./ipv6-setup-***xxx* command.
   - To learn how to use **ipv6-setup-***xxx*, run the **./ipv6-setup-***xxx* **--help** command.

----End

Linux (Manually Enabling Dynamic Assignment of IPv6 Addresses)

CAUTION

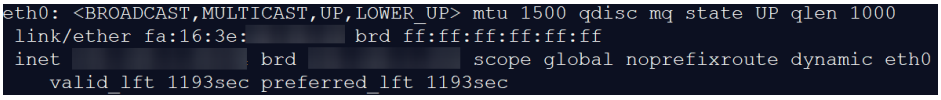
If a private image created from a CentOS 6.x or Debian ECS with automatic IPv6 address assignment enabled is used to create an ECS in an environment that does not support IPv6, the ECS may start slow because of IPv6 address assignment timeout. Set the timeout duration for assigning IPv6 addresses to 30s by referring to **Setting the Timeout Duration for IPv6 Address Assignment** and try to create a new private image again.

Step 1 Run the following command to check whether IPv6 is enabled for the ECS:

ip addr

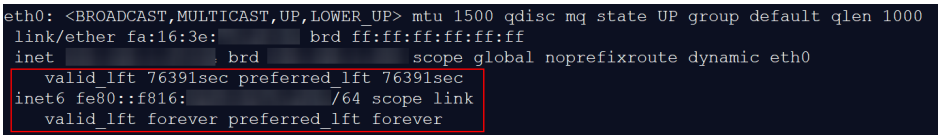
- If only an IPv4 address is displayed, IPv6 is disabled. Enable it by referring to **Step 2**.

Figure 7-28 IPv6 disabled



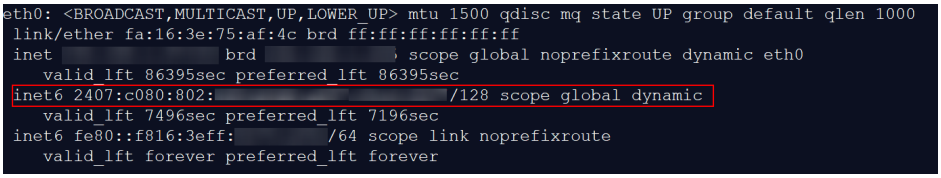
- If a link-local address (starting with fe80) is displayed, IPv6 is enabled but dynamic assignment of IPv6 addresses is not enabled.

Figure 7-29 IPv6 enabled



- If the following address is displayed, IPv6 is enabled and an IPv6 address has been assigned:

Figure 7-30 IPv6 enabled and an IPv6 address assigned



NOTE

IPv6 is enabled for Linux public images by default, as shown in **Figure 7-29**.

Step 2 Enable IPv6 for the ECS.

1. Run the following command to check whether IPv6 is enabled for the kernel:

   sysctl -a | grep ipv6

   - If a command output is displayed, IPv6 is enabled.
   - If no information is displayed, IPv6 is disabled. Go to **Step 2.2** to load the IPv6 module.

2. Run the following command to load the IPv6 module:

   modprobe ipv6

3. Add the following content to the **/etc/sysctl.conf** file:

   net.ipv6.conf.all.disable_ipv6=0

4. Save the configuration and exit. Then, run the following command to load the configuration:

   sysctl -p

Step 3 Enable dynamic IPv6 address assignment for the ECS.

- Ubuntu 18.04/20.04

  a. Run the following command to access **/etc/netplan/**:

  cd /etc/netplan

  b. Run the following command to list the configuration file:

  ls

Figure 7-31 Configuration file name

c. Run the following command to edit the configuration file:

vi 01-network-manager-all.yaml

d. Append the following content to the configuration file (pay attention to the yaml syntax and text indentation):

```
ethernets:
 eth0:
  dhcp6: true
```

Figure 7-32 Edited configuration file

Save the changes and exit.

e. Run the following command to make the changes take effect:

sudo netplan apply

- Ubuntu 22.04

a. Run the following command to access **/etc/netplan/**:

cd /etc/netplan

b. Run the following command to list the configuration file:

ls

Figure 7-33 Configuration file name

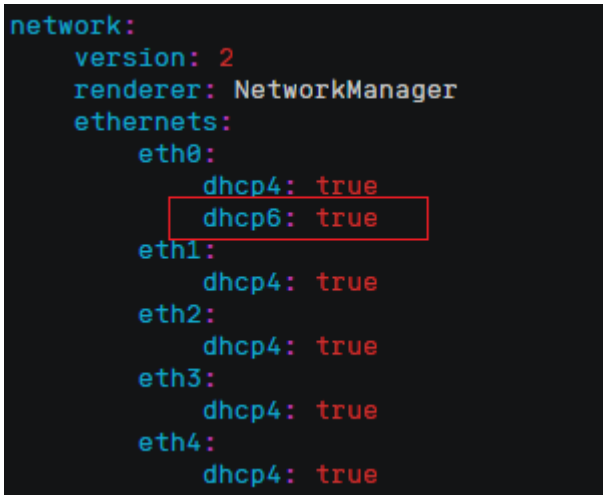
c. Run the following command to edit the configuration file:

vi 01-netcfg.yaml

d. Append the following content to the configuration file **01-netcfg.yaml** (pay attention to the yaml syntax and text indentation):

```
ethernets:
 eth0:
  dhcp6: true
```

Figure 7-34 Edited configuration file



Save the changes and exit.

e. Run the following command to make the changes take effect:

sudo netplan apply

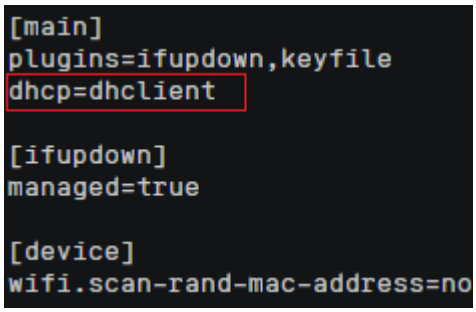
f. Run the following command to edit **/etc/NetworkManager/NetworkManager.conf**:

vi /etc/NetworkManager/NetworkManager.conf

g. Append the following content to the configuration file **NetworkManager.conf** (pay attention to the file format and indentation):

```
[main]
plugins=ifupdown,keyfile
dhcp=dhclient

[ifupdown]
managed=true

[device]
wifi.scan-rand-mac-address=no
```

Figure 7-35 Modification result



h. Run the following command for the configuration to take effect:

systemctl restart NetworkManager

- Debian

a. Add the following content to the **/etc/network/interfaces** file:

```
auto lo
iface lo inet loopback
```

```
auto eth0
iface eth0 inet dhcp
iface eth0 inet6 dhcp
   pre-up sleep 3
```

b. Add configurations for each NIC to the **/etc/network/interfaces** file. The following uses eth1 as an example:

```
auto eth1
iface eth1 inet dhcp
iface eth1 inet6 dhcp
   pre-up sleep 3
```

c. Run the following command to restart the network service:

service networking restart

NOTE

If no IPv6 address is assigned after the NICs are brought down and up, you can run this command to restart the network.

d. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

- CentOS, EulerOS, or Fedora

a. Open the configuration file **/etc/sysconfig/network-scripts/ifcfg-eth0** of the primary NIC.

Add the following configuration items to the file:

```
IPV6INIT=yes
DHCPV6C=yes
```

b. Edit the **/etc/sysconfig/network** file to add or modify the following line:

```
NETWORKING_IPV6=yes
```

c. For an ECS running CentOS 6, you need to edit the configuration files of its extension NICs. For example, if the extension NIC is eth1, you need to edit **/etc/sysconfig/network-scripts/ifcfg-eth1**.

Add the following configuration items to the file:

```
IPV6INIT=yes
DHCPV6C=yes
```

In CentOS 6.3, dhcpv6-client requests are filtered by **ip6tables** by default. So, you also need to add a rule allowing the dhcpv6-client request to the **ip6tables** file.

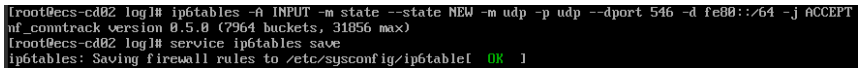
i. Run the following command to add the rule to **ip6tables**:

ip6tables -A INPUT -m state --state NEW -m udp -p udp --dport 546 -d fe80::/64 -j ACCEPT

ii. Run the following command to save the rule in **ip6tables**:

service ip6tables save

Figure 7-36 Example command

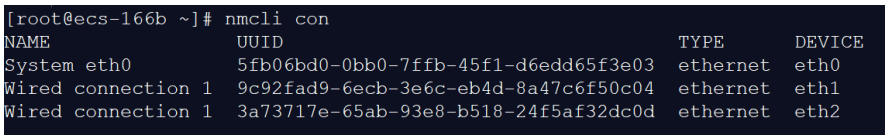


d. (Optional) For CentOS 7/CentOS 8, change the IPv6 link-local address mode of extension NICs to EUI64.

i. Run the following command to query the NIC information:

nmcli con

Figure 7-37 Querying NIC information



ii. Run the following command to change the IPv6 link-local address mode of eth1 to EUI64:

nmcli con modify "*Wired connection 1***" ipv6.addr-gen-mode eui64**

NOTE

The NIC information varies depending on the CentOS series. In the command, *Wired connection 1* needs to be replaced with the value in the **NAME** column of the queried NIC information.

iii. Run the following commands to bring eth1 down and up:

ifdown eth1

ifup eth1

e. Restart the network service.

i. For CentOS 6, run the following command to restart the network service:

service network restart

ii. For CentOS 7/EulerOS/Fedora, run the following command to restart the network service:

systemctl restart NetworkManager

f. Perform **Step 1** to check whether dynamic IPv6 address assignment is enabled.

- SUSE, openSUSE, or CoreOS

  SUSE 11 SP4 does not support dynamic IPv6 address assignment.

  No additional configuration is required for SUSE 12 SP1 or SUSE 12 SP2.

  No additional configuration is required for openSUSE 13.2 or openSUSE 42.2.

  No additional configuration is required for CoreOS 10.10.5.

----End

Setting the Timeout Duration for IPv6 Address Assignment

After automatic IPv6 address assignment is configured on an ECS running CentOS 6.x or Debian, the ECS will be created as a private image. When this image is used to create an ECS in an environment that IPv6 is unavailable, the ECS may start slow because acquiring an IPv6 address times out. Before creating the private image, you can set the timeout duration for acquiring IPv6 addresses to 30s as follows:

- CentOS 6.*x*:

  a. Run the following command to edit the **dhclient.conf** file:

  vi /etc/dhcp/dhclient.conf

  b. Press **i** to enter editing mode and add the timeout attribute to the file.

  ```
  timeout 30;
  ```

c. Enter **:wq** to save the settings and exit.

- Debian 7.5:
  a. Run the following command to edit the **networking** file:
  vi /etc/init.d/networking
  b. Press **i** to enter editing mode and add the timeout attribute.

Figure 7-38 Modification 1

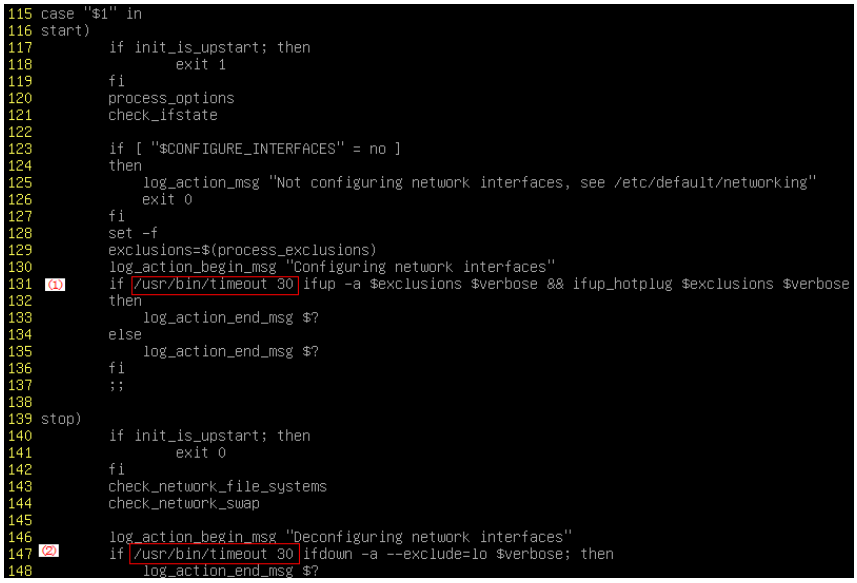
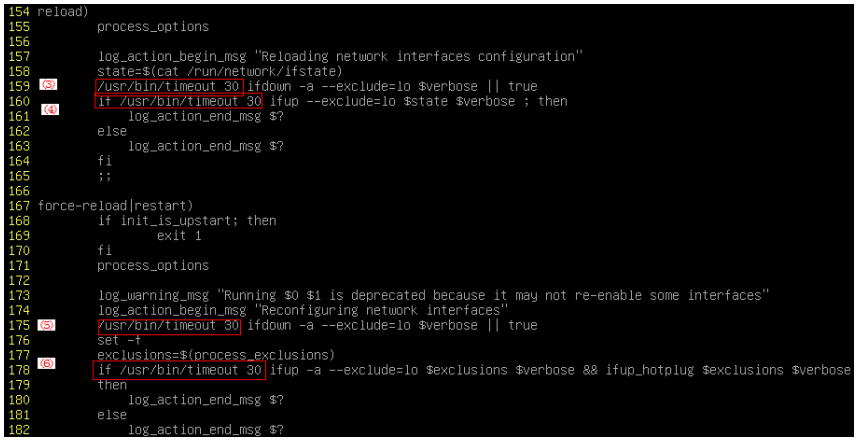


Figure 7-39 Modification 2



- Debian 8.2.0/8.8.0
  a. Run the following command to edit the **network-pre.conf** file:
  vi /lib/systemd/system/networking.service.d/network-pre.conf
  b. Press *i* to enter editing mode and add the timeout attribute to the file.

```
[Service]
TimeoutStartSec=30
```

- Debian 9.0
  a. Run the following command to edit the **networking.service** file:
  vi /etc/system/system/network-online.target.wants/networking.service

b. Press **i** to enter editing mode and change **TimeoutStartSec=5min** to **TimeoutStartSec=30**.

7.7.9 How Do I Make a System Disk Image Support Fast ECS Creation?

Scenarios

Fast Create greatly reduces the time required for creating ECSs from a system disk image. Currently, this feature is supported by all newly created system disk images by default. Some existing system disk images may not support this feature, you can make them support it through image replication.

For example, if image A does not support fast ECS creation, you can replicate it to generate image copy_A that supports fast ECS creation.

Constraints

Full-ECS images and ISO images cannot be configured using this method.

Check Whether an Image Supports Fast ECS Creation

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab to display the image list.
3. Click the name of the target image.
4. On the displayed image details page, check the value of **Fast ECS Creation**.

Configure an Image to Make It Support Fast ECS Creation

1. Locate the target system disk image, click **More** in the **Operation** column, and select **Replicate** from the drop-down list.
   The **Replicate Image** dialog box is displayed.
2. Set parameters based on **Replicating Images**.
3. After the image is successfully replicated, the generated image can be used to quickly create ECSs.

7.7.10 Whey Do I Fail to Install Guest OS Drivers on a Windows ECS?

Possible causes:

- Your image file was exported from a VMware VM, and VMware Tools was not uninstalled or not completely uninstalled.
- You have downloaded Guest OS drivers of an incorrect version for your Windows ECS.

- The disk space available for installing Guest OS drivers is insufficient. Ensure that the disk where Guest OS drivers are installed has at least 300 MB space available.

7.8 Image Replication

When Do I Need to Replicate an Image?

- In-region replication

  This is used for conversion between encrypted and unencrypted images or for enabling advanced features (such as fast ECS creation) for images. For details, see **Replicating Images**.

How Long Does It Take to Replicate an Image?

The time required for replicating an image depends on the network transmission speed and the number of tasks in the queue.

What Is the Charge for Image Replication?

- In-region replication

  The replicas of system disk and data disk images are stored in OBS buckets for free.

  NOTE

  Full-ECS images cannot be replicated within the same region.

7.9 Image Deletion

Will a Private Image Be Automatically Deleted If I Delete or Unsubscribe from the ECS Used to Create the Image?

No. Private images created using ECSs are stored in OBS buckets. Deleting or unsubscribing from the ECS used to create a private image does not affect the image.

Can I Delete an Image I Shared with Others?

- If an image is shared with a project, you can delete the image directly. The image recipient does not need to perform any operation. After you delete the image, the image recipient cannot use it any longer. Inform the image recipients to back up their data before you delete the image.

How Do I Delete a Shared Image? Does the Deletion Affect an ECS or EVS Disk Created from It?

Reject this image on the **Images Shared with Me** tab page. This does not affect an ECS or EVS disk created from it.

7.10 Image Encryption

How Can I Change an Unencrypted Image to an Encrypted One?

If you want to store an unencrypted image in an encrypted way, you can select an encryption key when you replicate the image. Then, the system will generate an encrypted replica of the unencrypted image.

Constraints

- Encrypted images cannot be shared with other tenants.
- The key used for encrypting an image cannot be changed.

7.11 Accounts and Permissions

7.11.1 What Do I Do If Private Images Cannot Be Found on the Enterprise Project Management Service Page After EPS Is Enabled?

Scenarios

If you cannot find the private images on the **Enterprise Project Management Service** page, add the private images to their associated enterprise project.

Procedure

1. Log in to the management console.
2. Under **Computing**, click **Image Management Service**.
3. Click the **Private Images** tab.
4. Locate the row that contains the image, click **More** in the **Operation** column, and select **Allocate to Enterprise Project**.
5. In the displayed dialog box, select the target enterprise project.

7.11.2 What Do I Do If I Cannot Create an Image from a CSBS Backup or BMS Using a Subaccount with the Allow_all Permission After EPS Is Enabled?

When an enterprise project subaccount is used to create an image, the system displays a message indicating that CSBS or BMS is not supported by EPS. This is because CSBS and BMS are not interconnected with EPS regionally or globally. The global resource viewing permission must be granted to the subaccount in IAM. For example, you can view resources of other cloud services if you have the Tenant Guest permission.

7.12 Cloud-Init

7.12.1 Cloud-Init Installation FAQ

You are advised to install Cloud-Init on the ECS that will be used to create a private image so that new ECSs created from the private image support custom configurations (for example, changing the ECS login password).

For details about how to install Cloud-Init, see **Installing Cloud-Init**.

For details about how to configure Cloud-Init, see **Configuring Cloud-Init**.

The following describes common problems you may encounter when installing Cloud-Init and their solutions.

Ubuntu 16.04/CentOS 7: Failed to Set Cloud-Init Automatic Start

- Symptom:

  After Cloud-Init is installed, run the following command to set Cloud-Init automatic start:

  systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  Information similar to the following is displayed:

  Figure 7-40 Failed to enable Cloud-Init to start automatically

  ```
  root@ecs-wjq-ubuntu14:~# systemctl enable cloud-init-local.service cloud-init.se
  rvice cloud-config.service cloud-final.service
  Failed to execute operation: Unit file is masked
  root@ecs-wjq-ubuntu14:~#
  ```

- Solution:

  a. Run the following command to roll back the configuration:

  systemctl unmask cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  b. Run the following command to set automatic start again:

  systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  c. Run the following command to check the Cloud-Init status:

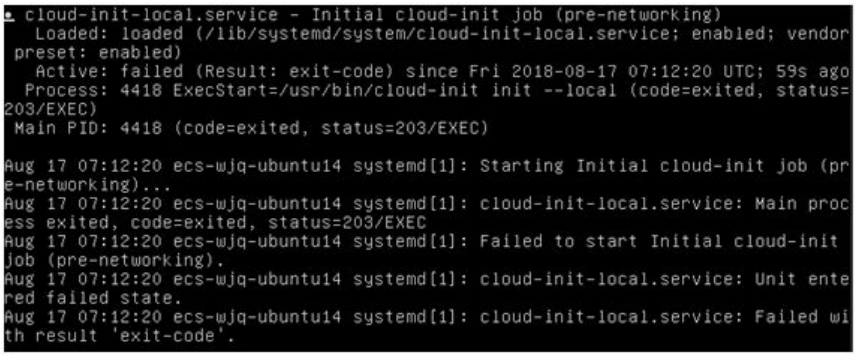
  systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

  As shown in the following figures, **failed** is displayed and all services are in the **inactive** state.

  Figure 7-41 Checking Cloud-Init status

  ```
  root@ecs-wjq-ubuntu14:~# systemctl status cloud-init-local.service
  ● cloud-init-local.service - Initial cloud-init job (pre-networking)
     Loaded: loaded (/lib/systemd/system/cloud-init-local.service; enabled; vendor
     Active: failed (Result: exit-code) since Fri 2018-08-17 07:12:20 UTC; 1min 25
    Process: 4418 ExecStart=/usr/bin/cloud-init init --local (code=exited, status=
   Main PID: 4418 (code=exited, status=203/EXEC)

  Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Starting Initial cloud-init job (pr
  Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Main proc
  Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Failed to start Initial cloud-init
  Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Unit ente
  Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Failed wi
  lines 1-11/11 (END)
  ```

Figure 7-42 Checking Cloud-Init status



This is because the address that the system uses to access Cloud-Init is redirected to **/usr/bin/**, but the actual installation path is **/usr/local/bin**.

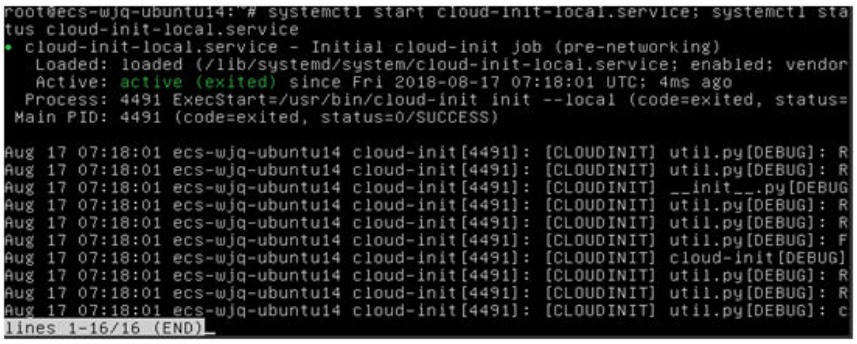
d. Run the following command to copy Cloud-Init to the **usr/bin** directory:

cp /usr/local/cloud-init /usr/bin/

e. Run the following command to restart Cloud-Init:

systemctl restart cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

Figure 7-43 Restarting Cloud-Init



f. Run the following command to check the Cloud-Init status:

systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

Ubuntu 14.04: chkconfig and systemctl Not Installed

- Symptom:

  chkconfig is not installed.

- Solution:

  Run the following commands to install chkconfig:

  apt-get update

  apt-get install sysv-rc-conf

  cp /usr/sbin/sysv-rc-conf /usr/sbin/chkconfig

  Run the following command to query the Cloud-Init version:

  cloud-init -v

  Information similar to the following is displayed:

```
-bash:/usr/bin/cloud-init:not found this command
```

Solution: Run the following command to copy Cloud-Init to the **usr/bin** directory:

cp /usr/local/bin/cloud-init /usr/bin/

Debian 9.5: Failed to Query the Cloud-Init Version and Set Automatic Start

1. Run the following command to query the Cloud-Init version:

   cloud-init -v

   Information similar to the following is displayed:

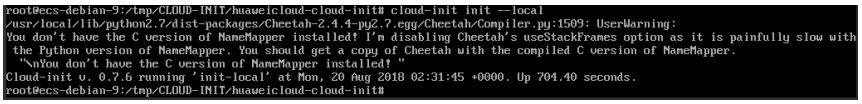
   ```
   -bash:/usr/bin/cloud-init:not found this command
   ```

   Solution: Run the **cp /usr/local/bin/cloud-init /usr/bin/** command to copy Cloud-Init to the **usr/bin** directory.

2. Run the **cloud-init init --local** command.

   Information similar to the following is displayed:

   Figure 7-44 Information returned when Cloud-Init automatic start successfully set

   Cause analysis: The compilation fails because GCC is not installed.

   Solution:

   Run the following command to install GCC. Then, install Cloud-Init again.

   yum -y install gcc

3. After Cloud-Init is installed, run the following command to set Cloud-Init automatic start:

   systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

   Information similar to the following is displayed.

   Figure 7-45 Prompt indicating the failure to set Cloud-Init automatic start

   Solution:

   a. Run the following command to roll back the configuration:

      systemctl unmask cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

   b. Run the following command to set automatic start again:

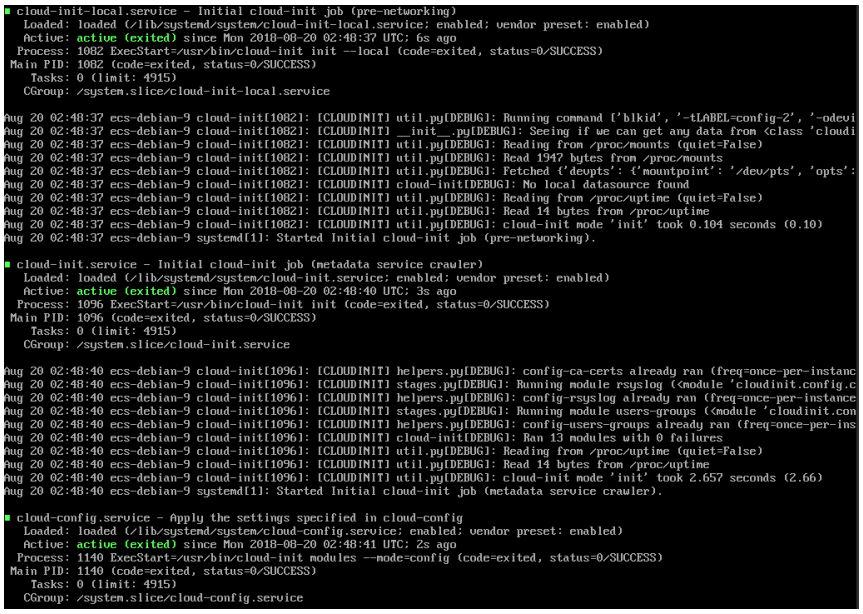
      systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

   c. Run the following command to restart Cloud-Init:

      systemctl restart cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

Run the **systemctl status** command to check the Cloud-Init status. Information similar to the following is displayed:

Figure 7-46 Verifying the service status



CentOS 7/Fedora 28: Required C Compiler Not Installed

- Symptom

  After Cloud-Init is successfully installed, run the following command:

  cloud-init init --local

  The following information is displayed:

  ```
  /usr/lib/python2.5/site-packages/Cheetah/Compiler.py:1532: UserWarning:
  You don't have the C version of NameMapper installed! I'm disabling Cheetah's useStackFrames
  option as it is painfully slow with the Python version of NameMapper. You should get a copy of
  Cheetah with the compiled C version of NameMapper.
    "\nYou don't have the C version of NameMapper installed!
  ```

- Cause analysis

  This alarm is generated because C version of NameMapper needs to be compiled when Cloud-Init is installed. However, GCC is not installed in the system, and the compilation cannot be performed. As a result, NameMapper is missing.

- Solution

  Run the following command to install GCC:

  yum -y install gcc

  Reinstall Cloud-Init.

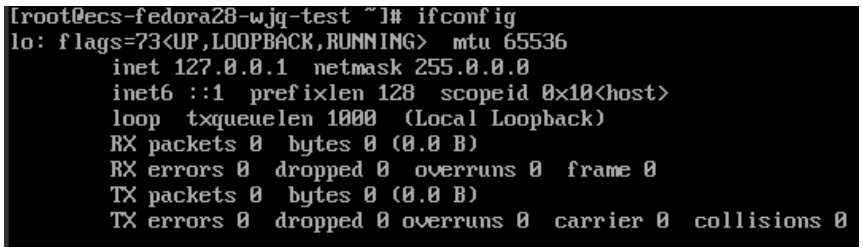
CentOS 7/Fedora: Failed to Use the New Password to Log In to an ECS Created from an Image

- Symptom

  After Cloud-Init is successfully installed on an ECS, an image is created from the ECS. You cannot use a new password to log in to the ECSs created from

this image. When you log in to the ECSs using the old password, you find that NICs of these ECSs are not started.

Figure 7-47 NIC not started



- Solution:

  Log in to the ECS used to create the image, open the DHCP configuration file **/etc/sysconfig/network-scripts/ifcfg-eth***X*, and comment out **HWADDR**.

7.12.2 What Can I Do with a Cloud-Init ECS?

Introduction to Cloud-Init

Cloud-Init is an open-source tool for cloud instance initialization. When creating ECSs from an image with Cloud-Init, you can use user data injection to inject customized initialization details (for example, an ECS login password) to the ECSs. You can also configure and manage a running ECS by querying and using metadata. If Cloud-Init is not installed, you cannot apply custom configurations to the ECSs. You will have to use the original password in the image file to log in to the ECSs.

Installation Methods

You are advised to install Cloud-Init or Cloudbase-Init on the ECS to be used to create a private image so that new ECSs created from the private image support custom configurations.

- For Windows OSs, download and install Cloudbase-Init.

  For how to install Cloudbase-Init, see **Installing and Configuring Cloudbase-Init**.

- For Linux OSs, download and install Cloud-Init.

  For how to install Cloud-Init, see **Installing Cloud-Init**.

  For how to configure Cloud-Init, see **Configuring Cloud-Init**.

7.12.3 What Do I Do If Injecting the Key or Password Using Cloud-Init Failed After NetworkManager Is Installed?

Symptom

A major cause is that the version of Cloud-Init is incompatible with that of NetworkManager. In Debian 9.0 and later versions, NetworkManager is incompatible with Cloud-Init 0.7.9.

Solution

Uninstall the current Cloud-Init and install Cloud-Init 0.7.6 or an earlier version.

For details about how to install Cloud-Init, see **Installing Cloud-Init**.

7.12.4 How Do I Install growpart for SUSE 11 SP4?

Scenarios

growpart for SUSE and openSUSE is an independent toolkit that does not start with **cloud-***. Perform operations in this section to install growpart.

Procedure

1. Run the following commands to check whether Cloud-Init and growpart have been installed:

   rpm -qa | grep cloud-init

   The command output is as follows:

   ```
   cloud-init-0.7.8-39.2
   ```

   rpm -qa | grep growpart

   The command output is as follows:

   ```
   growpart-0.29-8.1
   ```

2. Run the following command to uninstall Cloud-Init and growpart:

   zypper remove cloud-init growpart

3. Run the following commands to delete residual files:

   rm -fr /etc/cloud/*

   rm -fr /var/lib/cloud/*

4. Run the following command to install growpart:

   zypper install http://download.opensuse.org/repositories/home:/garloff:/OTC:/cloudinit/SLE_11_SP4/noarch/growpart-0.27-1.1.noarch.rpm

5. Run the following command to install python-oauth:

   zypper install http://download.opensuse.org/repositories/home:/garloff:/OTC:/cloudinit/SLE_11_SP4/x86_64/python-oauth-1.0.1-35.1.x86_64.rpm

6. Run the following command to install Cloud-Init:

   zypper install http://download.opensuse.org/repositories/home:/garloff:/OTC:/cloudinit/SLE_11_SP4/x86_64/cloud-init-0.7.6-27.23.1.x86_64.rpm

7. Run the following commands to check whether growpart, python-oauth, and Cloud-Init have been installed successfully:

   rpm -qa | grep growpart

   The command output is as follows:

   ```
   growpart-0.27-1.1
   ```

   rpm -qa | grep python-oauth

   The command output is as follows:

   ```
   python-oauthlib-0.6.0-1.5
   python-oauth-1.0.1-35.1
   ```

   rpm -qa | grep cloud-init

The command output is as follows:

```
cloud-init-0.7.6-27.19.1
```

8. Run the following command to check the configuration:

   chkconfig cloud-init-local on;chkconfig cloud-init on;chkconfig cloud-config on;chkconfig cloud-final on

7.12.5 Cloud-Init Configuration FAQ

How Do I Change the Hostname of a Cloud Server?

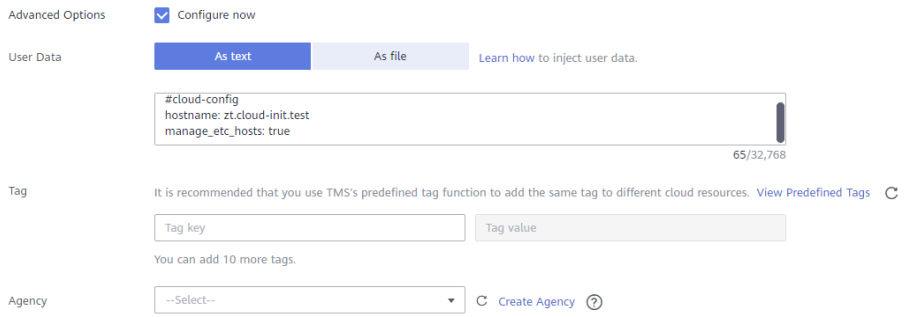
The static hostname of a Linux cloud server is user defined and injected using Cloud-Init during the cloud server creation. You can run a **hostname** command to change the hostname. Assume that the new hostname is **zt.cloud-init.test**.

1. When creating a cloud server, in the **Advanced Options** area, click **Configure now** and paste the following content to the **User Data** text box:

   #cloud-config

   hostname: *zt.cloud-init.test*

   manage_etc_hosts: true



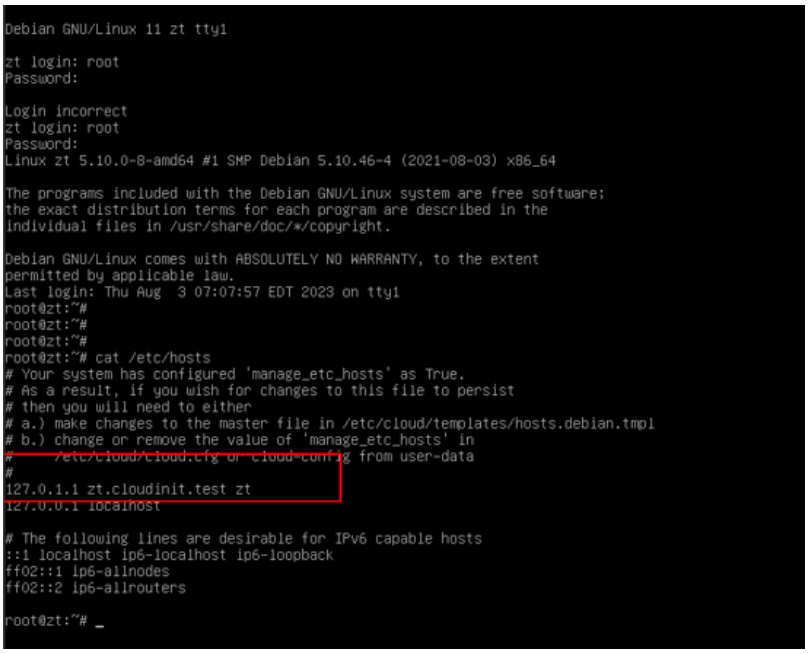
NOTE

Cloud-Init can read only the data starting with **#cloud-config**.

2. Log in to the cloud server and run the following command to check whether the hostname has been changed:

   cat /etc/hosts

   If the new hostname (**zt.cloud-init.test**) is displayed in the command output, the hostname has been changed.



How Do I Configure an SSH Key for a Cloud Server and Use Cloud-Init to Add a User?

If you want to use an SSH key pair instead of a password for more secure remote login, you can use a Cloud-Init script to add the key pair to the **~/.ssh/authorized_keys** file. The procedure is as follows:

1. Remotely log in to the cloud server in SSH mode.

   vim /etc/ssh/sshd_config

   PasswordAuthentication: yes

2. Generate a key pair.

   ssh-keygen -t rsa -f ~/.ssh/id_rsa -P

   cat ~/.ssh/id_rsa.pub

   Example:

   ```
   #cloud-config
   users:
     - default
     - name: myadminuser
       groups: sudo
       shell: /bin/bash
       sudo: ['ALL=(ALL) NOPASSWD:ALL']
       ssh-authorized-keys:
           - ssh-rsa Public key (for example, ~/.ssh/id_rsa.pub)
   ```

   NOTE

   - If the **#cloud-config** script contains the **- default** parameter, the new user will be added to the administrator list.
   - If the **#cloud-config** script does not contain the **- default** parameter, the new user will overwrite the original administrator.

3. Log in to the cloud server in SSH mode.

   ssh <*user*>**@**<*publicIpAddress*>

4. Open the **/etc/group** file and check whether the user is added to the cloud server and the specified user group:

   cat /etc/group

5. If the following information is displayed, the user in the **cloud_init_add_user.txt** file has been added to the cloud server and the specified user group:

   root:x:0:

   <snip />

   sudo:x:27:myadminuser

How Do I Create a Swap Partition?

Cloud-Init can be used to create swap partitions in Linux distributions. Conventionally, swap partitions are configured by the Linux agent (WALA) for different distributions. The following describes how to use Cloud-Init to create a swap partition.

1. Use Cloud-Init to create two partitions on a temporary disk. The first partition occupies 66% of the disk space. By default, it is an ext4 partition and is mounted to **/mnt**. The second partition is a swap partition, which occupies the remaining disk space and takes effect when the cloud server is started.

```
#cloud-config
disk_setup: # Disk settings
  ephemeral0:
    table_type: mbr (mbr or gpt)
    layout: [66, [33, 82]]  # 82 indicates the disk type of a swap partition.
    overwrite: True
fs_setup:    # Configure a file system on the partition.
  - device: ephemeral0.1
    filesystem: ext4
  - device: ephemeral0.2
    filesystem: swap
mounts:   # Use the first partition as ext4 and the second as swap.
  - ["ephemeral0.1", "/mnt"]
  - ["ephemeral0.2", "none", "swap", "sw", "0", "0"]
```

2. Check whether the swap partition is successfully created.

   swapon -s

   If the following information is displayed, the swap partition is successfully created:

```
Filename                Type        Size    Used    Priority
/dev/sdb2  partition   2494440 0   -1
```

7.13 ECS Creation

7.13.1 Can I Change the Image of a Purchased ECS?

Yes.

If you have selected an incorrect image or your service requirements have changed, you can change the image of your ECS.

You can change the image type (public, private, and shared images) and OS. For details, see "Changing the OS" in *Elastic Cloud Server User Guide*.

7.13.2 Can I Use a Private Image to Create ECSs with Different Hardware Specifications from the ECS Used to Create the Private Image?

Yes. You can specify the CPU, memory, bandwidth, data disks of the new ECSs if necessary. You can also specify their system disk capacity. The value must be smaller than 1024 GB but no less than the system disk capacity in the image.

7.13.3 Can I Specify the System Disk Capacity When I Create an ECS Using an Image?

Yes. However, the value must be smaller than 1024 GB but no less than the system disk capacity in the image.

7.13.4 What Do I Do If No Partition Is Found During the Startup of an ECS Created from an Imported Private Image?

Symptom

This may be caused by a disk partition ID change after the cross-platform image import. As a result, no partition can be found based on the original disk partition ID in the image. In this case, you need to change the disk partition in the image (**UUID=***UUID of the disk partition*).

Solution

The following uses openSUSE 13.2 as an example to describe how to change the partition name.

1. Run the following command to query the disk partition ID:

   ls -l /dev/disk/by-id/

   The example command output is as follows.

   ```
   total 0
   lrwxrwxrwx 1 root root 10 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001 -> ../../xvda
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part1 -> ../../xvda1
   lrwxrwxrwx 1 root root 12 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part10 -> ../../xvda10
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part2 -> ../../xvda2
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part5 -> ../../xvda5
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part6 -> ../../xvda6
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part7 -> ../../xvda7
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part8 -> ../../xvda8
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ata-QEMU_HARDDISK_QM00001-part9 -> ../../xvda9
   lrwxrwxrwx 1 root root 10 Jul 22 01:35 ata-QEMU_HARDDISK_QM00005 -> ../../xvde
   lrwxrwxrwx 1 root root 10 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001 -> ../../xvda
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part1 -> ../../xvda1
   lrwxrwxrwx 1 root root 12 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part10 -> ../../xvda10
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part2 -> ../../xvda2
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part5 -> ../../xvda5
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part6 -> ../../xvda6
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part7 -> ../../xvda7
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part8 -> ../../xvda8
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00001-part9 -> ../../xvda9
   lrwxrwxrwx 1 root root 10 Jul 22 01:35 scsi-SATA_QEMU_HARDDISK_QM00005 -> ../../xvde
   ```

   ata-QEMU_HARDDISK_xxx and **scsi-SATA_QEMU_HARDDISK_xxx** indicate that the disk of the ECS is simulated using Quick EMUlator (QEMU). The

content on the left of **->** is the disk partition ID, and that on the right of **->** is the partition name.

2. Run the following command to query the disk partition UUID:

   ls -l /dev/disk/by-uuid/

   The example command output is as follows.

   ```
   total 0
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 45ecd7a0-29da-4402-a017-4564a62308b8 -> ../../xvda5
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 55386c6a-9e32-41d4-af7a-e79596221f51 -> ../../xvda9
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 55f36660-9bac-478c-a701-7ecc5347f789 -> ../../xvda8
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 780f36bc-0ada-4c98-9a8d-44570d65333d -> ../../xvda1
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 b3b7c47f-6a91-45ef-80d6-275b1cc16e19 -> ../../xvda6
   lrwxrwxrwx 1 root root 11 Jul 22 01:35 ea63b55d-3b6e-4dcd-8986-956b72bac3e9 -> ../../xvda7
   lrwxrwxrwx 1 root root 12 Jul 22 01:35 eb3cc645-925e-4bc5-bedf-c2a6f3b65809 -> ../../xvda10
   ```

   The content on the left of **->** is the disk partition UUID, and that on the right of **->** is the partition name. Obtain the relationship between the disk partition name, partition ID, and partition UUID.

3. Run the following command to check the partition names in the **/etc/fstab** file:

   vi /etc/fstab

   The example command output is as follows.

   ```
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part5 / ext3 defaults,errors=panic 1 1
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part1 /boot ext3 defaults,errors=panic 1 2
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part6 /home ext3 nosuid,errors=panic 1 2
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part10 /opt ext3 defaults,errors=panic 1 2
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part7 /tmp ext3 nodev,nosuid,errors=panic 1 2
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part9 /usr ext3 defaults,errors=panic 1 2
   /dev/disk/by-id/scsi-SATA_QEMU_HARDDISK_QM00001-part8 /var ext3 nodev,nosuid,errors=panic 1 2
   sysfs /sys sysfs noauto 0 0
   proc /proc proc defaults 0 0
   usbfs /proc/bus/usb usbfs noauto 0 0
   devpts /dev/pts devpts mode=0620,gid=5 0 0
   /dev/cdrom /media/ udf,iso9660 noexec,noauto,nouser,nodev,nosuid 1 2
   tmpfs /dev/shm tmpfs noexec,nodev,nosuid 0 0
   ```

   The values in the first column are the disk partition IDs.

4. Press **i** to enter editing mode. Change the disk partition ID in the row that contains **/dev/disk/xxx** in the **/etc/fstab** file in step 3 to **UUID=***UUID of the disk partition* based on the query results in step 1 and step 2.

   The modified content is as follows.

   ```
   UUID=45ecd7a0-29da-4402-a017-4564a62308b8 / ext3 defaults,errors=panic 1 1
   UUID=780f36bc-0ada-4c98-9a8d-44570d65333d /boot ext3 defaults,errors=panic 1 2
   UUID=b3b7c47f-6a91-45ef-80d6-275b1cc16e19 /home ext3 nosuid,errors=panic 1 2
   UUID=eb3cc645-925e-4bc5-bedf-c2a6f3b65809 /opt ext3 defaults,errors=panic 1 2
   UUID=ea63b55d-3b6e-4dcd-8986-956b72bac3e9 /tmp ext3 nodev,nosuid,errors=panic 1 2
   UUID=55386c6a-9e32-41d4-af7a-e79596221f51 /usr ext3 defaults,errors=panic 1 2
   UUID=55f36660-9bac-478c-a701-7ecc5347f789 /var ext3 nodev,nosuid,errors=panic 1 2
   sysfs /sys sysfs noauto 0 0
   proc /proc proc defaults 0 0
   usbfs /proc/bus/usb usbfs noauto 0 0
   devpts /dev/pts devpts mode=0620,gid=5 0 0
   /dev/cdrom /media/ udf,iso9660 noexec,noauto,nouser,nodev,nosuid 1 2
   tmpfs /dev/shm tmpfs noexec,nodev,nosuid 0 0
   ```

   NOTE

   Ensure that the UUIDs are correct. Otherwise, the ECS cannot start properly.

5. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

6. Check the partition names in the system boot configuration file.

   The system boot configuration files vary depending on the OS. Confirm the boot configuration file of the current OS.

   - Grand Unified Boot Loader (GRUB) configuration file
     - /boot/grub/grub.conf
     - /boot/grub/menu.lst
     - /boot/grub/grub.cfg
     - /boot/grub2/grub.cfg
   - Syslinux configuration file
     - /extlinux.conf
     - /boot/syslinux/extlinux.conf
     - /boot/extlinux/extlinux.conf
     - /boot/syslinux/syslinux.cfg
     - /syslinux/syslinux.cfg
     - /syslinux.cfg

   The boot file in this example is **/boot/grub/menu.lst**. Run the following command to check it:

   vi /boot/grub/menu.lst

   ```
   default 0
   timeout 3
   title xxx Server OS - xxxxxx
   kernel /boot/vmlinuz-3.0.101-0.47.52-default root=/dev/disk/by-id/scsi-
   SATA_QEMU_HARDDISK_QM00001-part5 resume= memmap=0x2000000$0x3E000000
   nmi_watchdog=2 crashkernel=512M-:256M console=ttyS0,115200 console=tty0 xen_emul_unplug=all
   initrd /boot/initrd-3.0.101-0.47.52-default
   ```

7. Press **i** to enter editing mode and change the partition names in the system boot configuration file.

   Change the disk partition name in the **/boot/grub/menu.lst** file in **6** to **UUID=***UUID of the disk partition* based on the query results in **1** and **2**.

   ```
   default 0
   timeout 3
   title xxx Server OS - xxxxxx
   kernel /boot/vmlinuz-3.0.101-0.47.52-default root=UUID=45ecd7a0-29da-4402-a017-4564a62308b8
   resume= memmap=0x2000000$0x3E000000 nmi_watchdog=2 crashkernel=512M-:256M
   console=ttyS0,115200 console=tty0 xen_emul_unplug=all
   initrd /boot/initrd-3.0.101-0.47.52-default
   ```

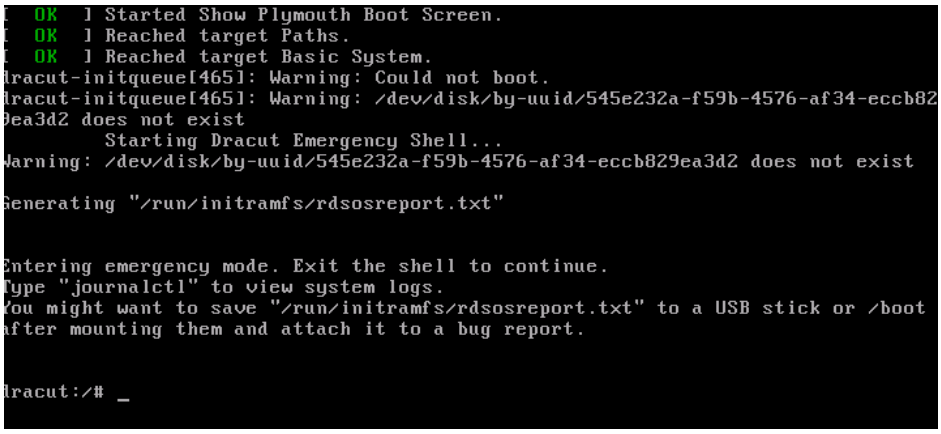
8. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

7.13.5 What Do I Do If the Disks of an ECS Created from a CentOS Image Cannot Be Found?

Symptom

Generally, this is because the xen-blkfront.ko module is not loaded during the startup. You need to modify OS kernel startup parameters. **Figure 7-48** shows the startup screen after the login to the ECS.

Figure 7-48 Startup screen



Solution

Perform the following operations to modify OS kernel boot parameters:

NOTE

These operations must be performed after the OS starts. You are advised to modify kernel boot parameters in the ECS used for creating the image.

1. Run the following command to log in to the OS:

   lsinitrd /boot/initramfs- `*uname* **-r** `**.img |grep -i xen**

   - If the command output contains **xen-blkfront.ko**, contact the customer service.
   - If no command output is displayed, go to 2.

2. Back up the GRUB configuration file.
   - If the ECS runs CentOS 6, run the following command:

     cp /boot/grub/grub.conf /boot/grub/grub.conf.bak
   - If the ECS runs CentOS 7, run the following command:

     cp /boot/grub2/grub.cfg /boot/grub2/grub.cfg.bak

3. Use the **vi** editor to open the GRUB configuration file. Run the following command (using CentOS 7 as an example):

   vi /boot/grub2/grub.cfg

4. Add **xen_emul_unplug=all** to the default boot kernel.

NOTE

Search for the line that contains **root=UUID=** and add **xen_emul_unplug=all** to the end of the line.

```
menuentry 'CentOS Linux (3.10.0-229.el7.x86_64) 7 (Core) with debugging' --class centos --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-229.el7.x86_64-advanced-bf3cc825-7638-48d8-8222-cd2f412dd0de' {
        load_video
        set gfxpayload=keep
        insmod gzio
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1'  bf3cc825-7638-48d8-8222-cd2f412dd0de
        else
          search --no-floppy --fs-uuid --set=root bf3cc825-7638-48d8-8222-cd2f412dd0de
        fi
        linux16 /boot/vmlinuz-3.10.0-229.el7.x86_64 root=UUID=bf3cc825-7638-48d8-8222-cd2f412dd0de xen_emul_unplug=all ro crashkernel=auto rhgb quiet  systemd.log_level=debug systemd.log_target=kmsg
        initrd16 /boot/initramfs-3.10.0-229.el7.x86_64.img
}
```

5. Press **Esc**, enter **:wq**, and press **Enter** to exit the vi editor.
6. Create an image using the ECS, upload and register the image on the cloud platform.

7.13.6 What Do I Do If an ECS Created from a Windows Image Failed to Start When I Have Enabled Automatic Configuration During Image Registration?

Symptom

This issue is probably caused by the failure of offline VirtIO driver injection.

Solution

When you inject VirtIO drivers for a Windows ECS offline, there are some restrictions:

- If the boot mode in the image file is UEFI, the VirtIO drivers cannot be injected offline.
- It is recommended that you disable Group Policy Object (GPO). Some policies may cause the failure of VirtIO driver injection offline.
- It is recommended that you stop antivirus software. Otherwise, the VirtIO drivers may fail to be injected offline.

To update VirtIO drivers, see **Optimizing a Windows Private Image**.

7.13.7 What Do I Do If an Exception Occurs When I Start an ECS Created from an Image Using the UEFI Boot Mode?

Symptom

An ECS created from a private image using the UEFI boot mode cannot start.

Possible Causes

The image OS uses the UEFI boot mode, but the uefi attribute is not added to the image.

Solution

1. Delete the ECS that failed to start.
2. Call the API to update the image attributes and change the value of **hw_firmware_type** to **uefi**.

   API URI: PATCH /v2/cloudimages/{*image_id*}

   For details about how to call the API, see "Updating Image Information" in *Image Management Service API Reference*.
3. Use the updated image to create an ECS.

7.14 Driver Installation

7.14.1 Must I Install Guest OS Drivers on an ECS?

Installing Guest OS drivers on an ECS improves your experience in using the ECS. In addition, it also ensures high reliability and stability of ECSs.

- Windows ECSs: Install VirtIO drivers on ECSs.
- Linux ECSs: Install VirtIO drivers and add them to initrd.

7.14.2 Why Do I Need to Install and Update VirtIO Drivers for Windows?

Why Do I Need to Install VirtIO Drivers?

VirtIO drivers are paravirtualized drivers that provide high-performance disks and NICs for ECSs.

- A standard Windows OS does not have VirtIO drivers.
- Public images have VirtIO drivers by default.
- You need to install VirtIO drivers for private images. For details, see **Installing VirtIO Drivers**.

Why Do I Need to Update VirtIO Drivers?

This ensures that known issues identified in the community or R&D tests can be avoided on the latest drivers.

When Do I Need to Update VirtIO Drivers?

After a major error is fixed, you are advised to update VirtIO drivers immediately. (This has not happened by now.)

After other issues are fixed, decide whether to update VirtIO drivers based on your needs.

What Do I Need to Do?

- Upgrade VirtIO drivers in Windows private images or running Windows ECSs.
- If you have any technical issue or question, contact the customer service.

7.14.3 Whey Do I Fail to Install Guest OS Drivers on a Windows ECS?

Possible causes:

- Your image file was exported from a VMware VM, and VMware Tools was not uninstalled or not completely uninstalled.
- You have downloaded Guest OS drivers of an incorrect version for your Windows ECS.
- The disk space available for installing Guest OS drivers is insufficient. Ensure that the disk where Guest OS drivers are installed has at least 300 MB space available.

7.14.4 How Do I Install VirtIO Drivers in Windows?

The installation only applies to KVM ECSs. Before using an ECS or external image file to create a private image, ensure that VirtIO drivers have been installed in the OS so that ECSs created from this image can support KVM virtualization and the network performance can be improved.

For details, see **Installing VirtIO Drivers**.

7.14.5 How Do I Install Native KVM Drivers in Linux?

When optimizing a Linux private image, you need to install native KVM drivers on the ECS from which the image will be created. If you ECS already has native KVM drivers installed, you do not need to install the drivers again.

For details, see **Installing Native KVM Drivers**.

7.14.6 How Do I Install Native KVM Drivers?

Scenarios

When optimizing a Linux private image with Xen virtualization, you need to install native Xen and KVM drivers on the source ECS of the image.

This section describes how to install native KVM drivers.

CAUTION

If an ECS has no KVM drivers installed, the NICs of the ECS may not be detected and the ECS will be unable to communicate with other resources.

Prerequisites

- The virtualization type of the ECS is Xen.
- The kernel version must be later than 2.6.24.
- Disable your antivirus and intrusion detection software. You can enable them after the driver installation is complete.

Procedure

Modify the configuration file depending on the OS.

- CentOS, EulerOS

  Take CentOS 7.0 as an example. Modify the **/etc/dracut.conf** file. Add the VirtIO drivers to **add_drivers**. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

  For details, see **CentOS and EulerOS**.

- Ubuntu and Debian

  Modify the **/etc/initramfs-tools/modules** file. Add the VirtIO drivers. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/initramfs-tools/modules** file. Run the **update-initramfs -u** command to regenerate initrd.

  For details, see **Ubuntu and Debian**.

- SUSE and openSUSE
  - If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the **/etc/sysconfig/kernel** file and add Xen PV and VirtIO drivers to **INITRD_MODULES=""**. Xen PV drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **mkinitrd** command to regenerate initrd.
  - If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file and add Xen PV and VirtIO drivers to **add_drivers**. Xen PV drivers include xen_vnif, xen_vbd, and xen_platform_pci. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Run the **dracut -f** command to regenerate initrd.
  - If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file and add Xen PV and VirtIO drivers to **add_drivers**. Xen PV drivers include xen-blkfront and xen-netfront. VirtIO drivers include virtio_blk, virtio_scsi, virtio_net, virtio_pci, virtio_ring, and virtio. Separate driver names with spaces. Save and exit the **/etc/dracut.conf** file. Run the **dracut -f** command to regenerate initrd.

  For details, see **SUSE and openSUSE**.

NOTE

For SUSE, run the following command to check whether xen-kmp (driver package for Xen PV) is installed:

rpm -qa |grep xen-kmp

If information similar to the following is displayed, xen-kmp is installed in the OS:

```
xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
```

If xen-kmp is not installed, obtain it from the ISO file and install it.

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected.

CentOS and EulerOS

1. Run the following command to open the **/etc/dracut.conf** file:

   vi /etc/dracut.conf

2. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add_drivers** (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
   # additional kernel modules to the default
   add_drivers+="xen-blkfront xen-netfront virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
   ......
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

4. Run the following command to regenerate initrd:

   dracut -f /boot/initramfs-2.6.32-573.8.1.el6.x86_64.img

   If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

5. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitrd /boot/initramfs-`uname -r`.img | grep xen

   lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

   If the virtual file system is initrd, run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitrd /boot/initrd-`uname -r` | grep xen

   lsinitrd /boot/initrd-`uname -r` | grep virtio

   Assume that the virtual file system is initramfs. The following command output will be displayed:

   ```
   [root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep xen
   -rwxr--r--   1 root     root        54888 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/block/xen-blkfront.ko
   -rwxr--r--   1 root     root        45664 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/net/xen-netfront.ko

   [root@CTU10000xxxxx home]# lsinitrd /boot/initramfs-`uname -r`.img | grep virtio
   -rwxr--r--   1 root     root        23448 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/block/virtio_blk.ko
   -rwxr--r--   1 root     root        50704 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/net/virtio_net.ko
   -rwxr--r--   1 root     root        28424 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/
   ```

```
scsi/virtio_scsi.ko
drwxr-xr-x  2 root  root     0 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio
-rwxr--r--  1 root  root  14544 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio.ko
-rwxr--r--  1 root  root  21040 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rwxr--r--  1 root  root  18016 Jul 16 17:53 lib/modules/2.6.32-573.8.1.el6.x86_64/kernel/drivers/virtio/virtio_ring.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected. The drivers cannot be found by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

Ubuntu and Debian

1. Run the following command to open the **modules** file:

   vi /etc/initramfs-tools/modules

2. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to the **/etc/initramfs-tools/modules** file (the format varies depending on the OS).

   ```
   [root@CTU10000xxxxx ~]#vi /etc/initramfs-tools/modules
   ......
   # Examples:
   #
   # raid1
   # sd_mOd
   xen-blkfront
   xen-netfront
   virtio_blk
   virtio_scsi
   virtio_net
   virtio_pci
   virtio_ring
   virtio
   ```

3. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/initramfs-tools/modules** file.

4. Run the following command to regenerate initrd:

   update-initramfs -u

5. Run the following commands to check whether native Xen and KVM drivers have been installed:

   lsinitramfs /boot/initrd.img-`uname -r` |grep xen

   lsinitramfs /boot/initrd.img-`uname -r` |grep virtio

   ```
   [root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep xen
   lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen
   lib/modules/3.5.0-23-generic/kernel/drivers/net/ethernet/qlogic/netxen/netxen_nic.ko
   lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback
   lib/modules/3.5.0-23-generic/kernel/drivers/net/xen-netback/xen-netback.ko
   lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback
   lib/modules/3.5.0-23-generic/kernel/drivers/block/xen-blkback/xen-blkback.ko

   [root@ CTU10000xxxxx home]# lsinitramfs /boot/initrd.img-`uname -r` |grep virtio
   lib/modules/3.5.0-23-generic/kernel/drivers/scsi/virtio_scsi.ko
   ```

NOTE

If you add built-in drivers to the initrd or initramfs file by mistake, the ECS will not be affected. The drivers cannot be found by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

```
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y
CONFIG_VIRTIO_BLK=y
CONFIG_VIRTIO_NET=y
CONFIG_VIRTIO=y
CONFIG_VIRTIO_RING=y
CONFIG_VIRTIO_PCI=y
CONFIG_VIRTIO_MMIO_CMDLINE_DEVICES=y
[root@ CTU10000xxxxx home]# cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y
CONFIG_XEN_BLKDEV_FRONTEND=y
CONFIG_XEN_NETDEV_FRONTEND=y
```

SUSE and openSUSE

If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, modify the **/etc/sysconfig/kernel** file to add drivers. For details, see **scenario 1**.

If the OS version is SUSE 12 SP1, modify the **/etc/dracut.conf** file to add drivers. For details, see **scenario 2**.

If the OS version is later than SUSE 12 SP1 or openSUSE 13, modify the **/etc/dracut.conf** file to add drivers. For details, see **scenario 3**.

- If the OS version is earlier than SUSE 12 SP1 or openSUSE 13, perform the following steps:

  NOTE

  For SUSE, run the following command to check whether xen-kmp (driver package for Xen PV) is installed in the OS:

  rpm -qa |grep xen-kmp

  If information similar to the following is displayed, xen-kmp is installed:

  ```
  xen-kmp-default-4.2.2_04_3.0.76_0.11-0.7.5
  ```

  If xen-kmp is not installed, obtain it from the installation ISO and install it first.

  a. Run the following command to open the **/etc/sysconfig/kernel** file:

     vi /etc/sysconfig/kernel

  b. Add Xen PV and VirtIO drivers after **INITRD_MODULES=** (the format varies depending on the OS).

     ```
     SIA10000xxxxx:~ # vi /etc/sysconfig/kernel
     # (like drivers for scsi-controllers, for lvm or reiserfs)
     #
     INITRD_MODULES="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
     ```

  c. Run the **mkinitrd** command to regenerate initrd:

     NOTE

     If the virtual file system is not the default initramfs or initrd, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **menu.lst** or **grub.cfg** file (**/boot/grub/menu.lst**, **/boot/grub/grub.cfg**, or **/boot/grub2/grub.cfg**).

     The following is an example initrd file of SUSE 11 SP4:

     ```
     default 0
     timeout 10
     ```

```
gfxmenu (hd0,0)/boot/message
title sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent console=ttyS0,115200n8 console=tty0
net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1 showopts
initrd /boot/initrd.vmx
title Failsafe_sles11sp4_001_[_VMX_]
root (hd0,0)
kernel /boot/linux.vmx vga=0x314 splash=silent ide=nodma apm=off noresume edd=off
powersaved=off nohz=off highres=off processsor.max+cstate=1 nomodeset x11failsafe
console=ttyS0,115200n8 console=tty0 net.ifnames=0 NON_PERSISTENT_DEVICE_NAMES=1
showopts
initrd /boot/initrd.vmx
```

/boot/initrd.vmx in the **initrd** line is the initrd file actually used. Run the **dracut -f /boot/initrd.vmx** command. If the initrd file does not contain the **/boot** directory, such as **/initramfs-***xxx*, run the **dracut -f /boot/initramfs-***xxx* command.

d. Run the following commands to check whether Xen PVOPS and KVM VirtIO have been installed:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
xen-netfront.ko

SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/
virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/
virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/
virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/
virtio_net.ko
```

e. Restart the ECS.

f. Modify the **/boot/grub/menu.lst** file. Add **xen_platform_pci.dev_unplug=all** and modify the root configuration.

Before the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314
initrd /boot/initrd-3.0.76-0.11-default
```

After the modification:

```
###Don't change this comment -YaST2 identifier: Original name: linux###
title SUSE Linux Enterprise Server 11SP4 - 3.0.76-0.11 (default)
root (hd0,0)
kernel /boot/vmlinuz-3.0.76-0.11-default root=UUID=4eb40294-4c6f-4384-bbb6-
b8795bbb1130 splash=silentcrashkernel=256M-:128M showopts vga=0x314
xen_platform_pci.dev_unplug=all
initrd /boot/initrd-3.0.76-0.11-default
```

NOTE

- Ensure that the root partition is in the UUID format.
- **xen_platform_pci.dev_unplug=all** is added to shield QEMU devices.
- For SUSE 11 SP1 64bit to SUSE 11 SP4 64bit, add **xen_platform_pci.dev_unplug=all** to the **menu.lst** file. For SUSE 12 or later, QEMU device shield is enabled by default, and you do not need to configure it.

g. Run the following commands to check whether Xen drivers exist in initrd:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

```
SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
-rwxr--r-- 1 root root 42400 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/xen-blkfront.ko
-rwxr--r-- 1 root root 44200 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/xen-netfront.ko

SIA10000xxxxx:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rwxr--r-- 1 root root 19248 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/scsi/virtio_scsi.ko
-rwxr--r-- 1 root root 23856 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/block/virtio_blk.ko
drwxr-xr-x 2 root root 0 Jul 12 14:53 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio
-rwxr--r-- 1 root root 15848 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_ring.ko
-rwxr--r-- 1 root root 20008 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio_pci.ko
-rwxr--r-- 1 root root 12272 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/virtio/virtio.ko
-rwxr--r-- 1 root root 38208 Jun 22 2012 lib/modules/2.6.32-279.el6.x86_64/kernel/drivers/net/virtio_net.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

- If the OS version is SUSE 12 SP1, perform the following steps:

a. Run the following command to open the **/etc/dracut.conf** file:

vi /etc/dracut.conf

b. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add-drivers** (the format varies depending on the OS).

```
[root@CTU10000xxxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="ata_piix ata_generic xen_vnif xen_vbd xen_platform_pci virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
```

c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

d. Run the following command to regenerate initrd:

dracut -f /boot/initramfs-*File name*

If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which

can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

lsinitrd /boot/initramfs-`uname -r`.img | grep xen

lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

If the virtual file system is initrd, run the following commands to check whether native Xen and KVM drivers have been installed:

lsinitrd /boot/initrd-`uname -r` | grep xen

lsinitrd /boot/initrd-`uname -r` | grep virtio

- If the OS version is later than SUSE 12 SP1 or openSUSE 13, perform the following steps:

  Take SUSE Linux Enterprise Server 12 SP2 (x86_64) as an example.

  a. Run the following command to open the **/etc/dracut.conf** file:

  vi /etc/dracut.conf

  b. Press **i** to enter editing mode and add Xen PV and VirtIO drivers to **add_drivers** (the format varies depending on the OS).

  ```
  [root@CTU10000xxxxx ~]# vi /etc/dracut.conf
  # additional kernel modules to the default
  add_drivers+="ata_piix ata_generic xen-blkfront xen-netfront virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
  ```

  c. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the change and exits the **/etc/dracut.conf** file.

  d. Run the following command to regenerate initrd:

  dracut -f /boot/initramfs-*File name*

  If the virtual file system is not the default initramfs, run the **dracut -f** *Name of the initramfs or initrd file actually used* command. The actual initramfs or initrd file name can be obtained from the **grub.cfg** file, which can be **/boot/grub/grub.cfg**, **/boot/grub2/grub.cfg**, or **/boot/grub/grub.conf** depending on the OS.

  e. If the virtual file system is initramfs, run the following commands to check whether native Xen and KVM drivers have been installed:

  lsinitrd /boot/initramfs-`uname -r`.img | grep xen

  lsinitrd /boot/initramfs-`uname -r`.img | grep virtio

  If the virtual file system is initrd, run the following commands to check whether the native Xen and KVM drivers have been installed:

  lsinitrd /boot/initrd-`uname -r` | grep xen

  lsinitrd /boot/initrd-`uname -r` | grep virtio

  Assume that the virtual file system is initrd. The following command output will be displayed:

  ```
  sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep xen
  -rw-r--r-- 1 root root 69575 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/xen-blkfront.ko
  -rw-r--r-- 1 root root 53415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/xen-netfront.ko
  drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-hcall
  -rwxr-xr-x 1 root root 8320 Sep 28 10:21 lib/modules/4.4.21-69-default/updates/pvdriver/xen-hcall/xen-hcall.ko
  ```

```
sluo-ecs-30dc:~ # lsinitrd /boot/initrd-`uname -r` | grep virtio
-rw-r--r-- 1 root root 29335 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/block/
virtio_blk.ko
-rw-r--r-- 1 root root 57007 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/net/
virtio_net.ko
-rw-r--r-- 1 root root 32415 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/scsi/
virtio_scsi.ko
drwxr-xr-x 2 root root 0 Sep 28 10:21 lib/modules/4.4.21-69-default/kernel/drivers/virtio
-rw-r--r-- 1 root root 19623 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio.ko
-rw-r--r-- 1 root root 38943 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_pci.ko
-rw-r--r-- 1 root root 24431 Oct 26 2016 lib/modules/4.4.21-69-default/kernel/drivers/virtio/
virtio_ring.ko
```

NOTE

If you add built-in drivers to the initrd or initramfs file, the ECS will not be affected. This makes it easy to modify the drivers. However, you cannot check the drivers by running the **lsinitrd** command. You can run the following commands to check whether built-in drivers are in the kernel:

cat /boot/config-`uname -r` | grep CONFIG_VIRTIO | grep y

cat /boot/config-`uname -r` | grep CONFIG_XEN | grep y

7.15 Image Tags

7.15.1 How Many Tags Can I Add to an Image?

An image can have a maximum of 10 tags.

7.15.2 How Do I Add, Delete, and Modify Image Tags?

NOTE

- When adding predefined tags to an image or searching for an image using predefined tags, you must have permission to access the Tag Management Service (TMS).

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab and click the image name to display the image details.
   - To modify an image tag, go to **3**.
   - To delete an image tag, go to **4**.
   - To add an image tag, go to **5**.
3. Click the **Tags** tab, locate the target tag, and click **Edit** in the **Operation** column. In the displayed dialog box, modify the tag.
4. Click the **Tags** tab, locate the target tag, and click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes**.
5. Click the **Tags** tab and then **Add Tag**. In the displayed dialog box, add a tag.

7.15.3 How Do I Search for Private Images by Tag?

NOTE

- When adding predefined tags to an image or searching for an image using predefined tags, you must have permission to access the Tag Management Service (TMS).

Search for Private Images by Tag

1. Access the IMS console.
   a. Log in to the management console.
   b. Under **Computing**, click **Image Management Service**.
      The IMS console is displayed.
2. Click the **Private Images** tab and then **Search by Tag**.
3. Enter the tag key and value.
   Neither the tag key nor tag value can be empty. When the tag key and tag value are matched, the system automatically shows your desired private images.
4. Click  to add a tag.
   You can add multiple tags to search for shared images. The system will display private images that match all tags.
5. Click **Search**.
   The system searches for private images based on tag keys or tag values.

A Change History

Released On	Description
2024-06-12	This issue is the twelfth official release. Added the following content: **Installing VirtIO Drivers** Deleted the "Installing UVP VMTools" section.
2023-09-25	This issue is the eleventh official release. Modified the following content: • Added Ubuntu 18 and later in **Setting the NIC to DHCP**. • Added the **plugins** configuration item in **Installing and Configuring Cloudbase-Init**. • Supplemented information in **Configuring Cloud-Init**. Added the following content: • **Cloud-Init Configuration FAQ**

Released On	Description
2023-05-10	This issue is the tenth official release. Added the following content: • **Product Advantages** • **Application Scenarios** • **Features** • **Constraints** • **Permissions** • **Creating a BMS System Disk Image** • **Tagging an Image** • **Permissions Management** • **Creating a User and Granting Permissions** • **Creating a Custom Policy** • **Basic Concepts** • **What Do I Do If I Cannot Find a Desired Image?** • **What Are the Differences Between Images and Backups?** • **Can I Tailor an Image?** • **How Can I Back Up the Current Status of an ECS for Restoration in the Case of a System Fault?** • **How Can I Apply a Private Image to an Existing ECS?** • **Can I Import Data from a Data Disk Image to a Data Disk?** • **Full-ECS Image FAQs** • **Is There Any Difference Between the Image Created from a CSBS/CBR Backup and That Created from an ECS?** • **What Do I Do If I Cannot Create an Image in ZVHD2 Format Using an API?** • **What Are the Differences Between Sharing Images and Replicating Images?** • **How Do I Select an OS?** • **Why Can't I Find My Private Image When I Want to Use It to Create an ECS or Change the OS of an ECS?** • **Image Replication** • **Image Deletion** • **What Do I Do If Private Images Cannot Be Found on the Enterprise Project Management Service Page After EPS Is Enabled?** • **What Do I Do If I Cannot Create an Image from a CSBS Backup or BMS Using a Subaccount with the Allow_all Permission After EPS Is Enabled?** • **Cloud-Init Installation FAQ** • **Can I Change the Image of a Purchased ECS?**

Released On	Description
	• **Driver Installation** • **Must I Install Guest OS Drivers on an ECS?** • **Why Do I Need to Install and Update VirtIO Drivers for Windows?** • **Whey Do I Fail to Install Guest OS Drivers on a Windows ECS?** • **How Do I Install VirtIO Drivers in Windows?** • **How Do I Install Native KVM Drivers in Linux?** • **Image Tags** • **How Many Tags Can I Add to an Image?** • **How Do I Add, Delete, and Modify Image Tags?** • **How Do I Search for Private Images by Tag?** Modified the following content: • Added the Tag Management Service (TMS) and updated the figure showing relationships between IMS and other services in **Related Services**. • Updated the figure in **Introduction**. • Added the **Enterprise Project** parameter in **Creating a System Disk Image from a Windows ECS** and **Creating a System Disk Image from a Linux ECS**. • Added restrictions on VMDK image files in **Preparing an Image File**. • Added configuration items **Function**, **Enterprise Project**, and **Tag** in **Registering an External Image File as a Private Image** and **Registering an External Image File as a Private Image**. • Added a screenshot of the console operations in **Creating a Data Disk Image from an External Image File**. • Added the description of sharing full-ECS image in **Creating a Full-ECS Image from an ECS**. • Added a screenshot of the console operations in **Quickly Importing an Image File (Linux)** and **Quickly Importing an Image File (Windows)**. • Added the **Boot Mode** attribute and updated the console operation screenshot accordingly in **Modifying an Image**. • Added new scenarios in **Deleting Images**. • Clarified that the cloud platform does not ensure the integrity or security of shared images and advised users to use images from a trusted sharer in **Overview**. • Added follow-up operations in **Accepting or Rejecting Shared Images**. • Clarified that users need to be billed for storing exported images in **Exporting an Image**.

Released On	Description
	• Added the parameter **Enterprise Project** in **Replicating Images**. • Added new questions and answers in **Image Creation FAQs**. • Added new questions and answers in **Image Sharing FAQs**. • Updated the console operation screenshot in **How Do I Make a System Disk Image Support Fast ECS Creation?**. Deleted the following content: Deleted "How Do I Configure a Linux Private Image to Make It Automatically Expand Its Root Partition?" from **Cloud-Init FAQ**.
2022-06-10	This issue is the ninth official release. Deleted the "Installing PV Drivers" section. Deleted information about PV drivers from the following sections: • **Preparing an Image File** • **Configuring the ECS and Creating a Windows System Disk Image** • **Optimization Process** • **What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?**
2022-02-15	This issue is the eighth official release. Added **How Do I Configure an ECS to Dynamically Acquire IPv6 Addresses?**
2021-08-30	This issue is the seventh official release. Modified the following content: • Added the method of downloading the quick import tool in **Quickly Importing an Image File**.
2021-07-31	This issue is the sixth official release. Added the following content: • **Permissions** • **Permissions Management** • **What Do I Do If I Cannot Share My Images?** Modified the following content: • Added the description of a full-ECS image's status in **Creating a Full-ECS Image from an ECS**.

Released On	Description
2021-06-30	This issue is the fifth official release. Added the following content: • **Why Can't I Find an ISO Image When I Want to Use It to Create an ECS or Change the OS of an ECS?** • **Can I Download My Private Images to a Local PC?** • **Can I Use the System Disk Image of an ECS on a BMS After I Export It from the Cloud Platform?** • **Why Is the Image Size in an OBS Bucket Different from That Displayed in IMS?** • **Can I Download a Public Image to My Local PC?** • **What Are the Differences Between Import/Export and Fast Import/Export?** • **What Do I Do If the Export Option Is Unavailable for My Image?**
2021-04-15	This issue is the fourth official release. Added the following content: • **Encrypting Images** Modified the following content: • Modified "Prerequisites" in **Creating a Data Disk Image from an ECS**. • Added the startup file **/boot/efi/EFI/euleros/grub.cfg** of EulerOS 2.9 in **Changing the Disk Identifier in the GRUB Configuration File to UUID**. • Added the configuration of Cloud-Init 18.3 and later versions in **Configuring Cloud-Init**.
2020-12-20	This issue is the third official release. Added the following content: **Checking the Disk Capacity of an Image**
2020-11-12	This issue is the second official release. Added **How Do I Import an OVF or OVA File to the Cloud Platform?**
2020-02-26	This issue is the first official release.