

Image Management Service

User Guide (Paris Regions)

Issue 10
Date 2024-12-18



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

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

Trademarks and Permissions



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

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

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

Contents

1 Overview.....	1
1.1 What Is Image Management Service?.....	1
1.2 Product Advantages.....	4
1.3 Application Scenarios.....	5
1.4 Features.....	5
1.5 Constraints.....	8
1.6 Supported OSs.....	12
1.6.1 External Image File Formats and Supported OSs.....	12
1.6.2 OSs Supporting UEFI Boot Mode.....	17
1.7 Permissions.....	19
1.8 Basic Concepts.....	21
1.8.1 Region and AZ.....	21
1.8.2 Common Image Formats.....	22
1.9 Related Services.....	24
2 Using IAM to Grant Access to IMS.....	27
2.1 Creating a User and Granting Permissions.....	27
2.2 Creating a Custom Policy.....	29
3 Creating a Private Image.....	31
3.1 Introduction.....	31
3.2 Creating a System Disk Image from a Windows ECS.....	32
3.3 Creating a System Disk Image from a Linux ECS.....	35
3.4 Creating a Windows System Disk Image from an External Image File.....	39
3.4.1 Overview.....	39
3.4.2 Preparing an Image File.....	39
3.4.3 Uploading an External Image File.....	42
3.4.4 Registering an External Image File as a Private Image.....	42
3.4.5 Creating a Windows ECS from an Image.....	45
3.5 Creating a Linux System Disk Image from an External Image File.....	45
3.5.1 Overview.....	45
3.5.2 Preparing an Image File.....	46
3.5.3 Uploading an External Image File.....	49
3.5.4 Registering an External Image File as a Private Image.....	49

3.5.5 Creating a Linux ECS from an Image.....	52
3.6 Creating a Data Disk Image from an External Image File.....	52
3.7 Creating a Full-ECS Image from an ECS.....	55
3.8 Creating a Full-ECS Image from a CSBS Backup.....	58
3.9 Creating a Full-ECS Image from a CBR Backup.....	59
3.10 Creating a Windows System Disk Image from an ISO File.....	61
3.10.1 Overview.....	61
3.10.2 Integrating VirtIO Drivers into an ISO File.....	63
3.10.3 Registering an ISO File as an ISO Image.....	65
3.10.4 Creating a Windows ECS from an ISO Image.....	67
3.10.5 Installing a Windows OS and VirtIO Drivers.....	68
3.10.6 Configuring the ECS and Creating a Windows System Disk Image.....	77
3.11 Creating a Linux System Disk Image from an ISO File.....	78
3.11.1 Overview.....	78
3.11.2 Registering an ISO File as an ISO Image.....	80
3.11.3 Creating a Linux ECS from an ISO Image.....	81
3.11.4 Installing a Linux OS.....	82
3.11.5 Configuring the ECS and Creating a Linux System Disk Image.....	87
3.12 Importing an Image.....	88
3.13 Fast Import of an Image File.....	89
3.13.1 Overview.....	89
3.13.2 Fast Import in Linux.....	92
3.13.3 Fast Import in Windows.....	97
4 Managing Private Images.....	100
4.1 Creating an ECS from an Image.....	100
4.2 Modifying an Image.....	101
4.3 Exporting an Image.....	102
4.4 Exporting Image List.....	104
4.5 Checking the Disk Capacity of an Image.....	105
4.6 Deleting Images.....	106
4.7 Sharing Images.....	107
4.7.1 Overview.....	107
4.7.2 Obtaining the Project ID.....	108
4.7.3 Sharing Specified Images.....	108
4.7.4 Accepting or Rejecting Shared Images.....	110
4.7.5 Rejecting Accepted Images.....	112
4.7.6 Accepting Rejected Images.....	112
4.7.7 Stopping Sharing Images.....	113
4.7.8 Adding Tenants Who Can Use Shared Images.....	113
4.7.9 Deleting Image Recipients Who Can Use Shared Images.....	114
4.8 Replicating Images Within a Region.....	114
4.9 Optimizing a Windows Private Image.....	116

4.9.1 Optimization Process.....	116
4.9.2 Viewing the Virtualization Type of a Windows ECS.....	116
4.9.3 Obtaining Required Software Packages.....	117
4.9.4 Installing PV Drivers.....	118
4.9.5 Installing VirtIO Drivers.....	119
4.9.6 Clearing System Logs.....	125
4.10 Optimizing a Linux Private Image.....	126
4.10.1 Optimization Process.....	126
4.10.2 Checking Whether a Private Image Needs to be Optimized.....	126
4.10.3 Uninstalling PV Drivers from a Linux ECS.....	128
4.10.4 Changing Disk Identifiers in the GRUB File to UUID.....	129
4.10.5 Changing Disk Identifiers in the fstab File to UUID.....	133
4.10.6 Installing Native Xen and KVM Drivers.....	134
4.10.7 Installing Native KVM Drivers.....	142
4.10.8 Clearing System Logs.....	148
4.11 Encrypting Images.....	149
4.11.1 Overview.....	149
4.11.2 Creating Encrypted Images.....	149
4.12 Converting the Image Format.....	150
4.12.1 Converting the Image Format Using qemu-img.....	150
4.12.2 Converting the Image Format Using qemu-img-hw.....	154
5 Windows Operations.....	158
5.1 Configuring DHCP.....	158
5.2 Enabling Remote Desktop Connection.....	160
5.3 Installing and Configuring Cloudbase-Init.....	161
5.4 Running Sysprep.....	167
5.5 Installing Special Windows Drivers.....	169
6 Linux Operations.....	171
6.1 Configuring DHCP.....	171
6.2 Deleting Files from the Network Rule Directory.....	173
6.3 Installing Cloud-Init.....	175
6.4 Configuring Cloud-Init.....	180
6.5 Detaching Data Disks from an ECS.....	186
7 Managing Tags.....	188
8 Managing Quotas.....	190
9 Auditing Key Operations.....	191
9.1 IMS Operations Audited by CTS.....	191
9.2 Viewing Traces.....	193
10 FAQs.....	195
10.1 Image Consulting.....	195

10.1.1 Basic Concepts.....	195
10.1.2 How Do I Select an Image?.....	197
10.1.3 What Do I Do If I Cannot Find a Desired Image?.....	198
10.1.4 What Are the Differences Between Images and Backups?.....	198
10.1.5 Can I Tailor an Image?.....	200
10.1.6 How Can I Back Up the Current Status of an ECS for Restoration in the Case of a System Fault?.....	200
10.1.7 How Can I Apply a Private Image to an Existing ECS?.....	201
10.1.8 Can I Import Data from a Data Disk Image to a Data Disk?.....	201
10.1.9 Can I Use Private Images of Other Accounts?.....	201
10.2 End-of-Support for OSs.....	201
10.2.1 What Do I Do If CentOS Linux Is No Longer Maintained?.....	201
10.3 Image Creation.....	203
10.3.1 General Creation FAQ.....	203
10.3.2 Full-ECS Image FAQ.....	204
10.3.3 How Can I Use a Backup to Create an EVS Disk or ECS?.....	205
10.3.4 Is There Any Difference Between the Image Created from a CSBS/CBR Backup and That Created from an ECS?.....	205
10.3.5 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?.....	205
10.3.6 How Do I Create a Full-ECS Image Using an ECS That Has a Spanned Volume?.....	205
10.3.7 Why Is Sysprep Required for Creating a Private Image from a Windows ECS?.....	206
10.3.8 How Do I Handle the Startup Failure of a Windows ECS Created from a Windows Image Generalized by Sysprep?.....	206
10.3.9 What Do I Do If I Cannot Create an Image in ZVHD2 Format Using an API?.....	208
10.4 Image Sharing.....	208
10.4.1 General Sharing FAQ.....	208
10.4.2 What Are the Differences Between Sharing Images and Replicating Images?.....	210
10.4.3 Why Can't I Share My Images?.....	211
10.5 OS.....	211
10.5.1 How Do I Select an OS?.....	211
10.5.2 How Is BIOS Different from UEFI?.....	211
10.5.3 How Do I Delete Redundant Network Connections from a Windows ECS?.....	212
10.5.4 What Do I Do If an ECS Starts Slowly?.....	213
10.5.5 What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?.....	214
10.5.6 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?.....	215
10.6 Image Import.....	215
10.6.1 Can I Use Images in Formats not Described in This Document?.....	215
10.6.2 What Are the Impacts If I Do Not Pre-configure an ECS Used to Create a Private Image?.....	215
10.6.3 What Do I Do If I Chose the Wrong OS or System Disk Capacity When Registering a Private Image?.....	216
10.6.4 Why Did My VHD Upload Fail? Why Does the System Say the System Disk in the VHD Image File Is Larger Than What I Specified on the Management Console?.....	216

10.7 Image Export.....	216
10.7.1 Can I Download My Private Images to a Local PC?.....	216
10.7.2 Can I Use the System Disk Image of an ECS on a BMS After I Export It from the Cloud Platform?	217
10.7.3 Why Is the Image Size in an OBS Bucket Different from That Displayed in IMS?.....	217
10.7.4 Can I Download a Public Image to My PC?.....	217
10.7.5 What Are the Differences Between Import/Export and Fast Import/Export?.....	218
10.7.6 Why the Export Option Is Unavailable for My Image?.....	219
10.8 Image Optimization.....	219
10.8.1 Must I Install Guest OS Drivers on an ECS?.....	219
10.8.2 Why Do I Need to Install and Update VirtIO Drivers for Windows?.....	219
10.8.3 What Will the System Do to an Image File When I Use the File to Register a Private Image?.....	220
10.8.4 How Do I Configure an ECS, a BMS, or an Image File Before I Use It to Create an Image?.....	221
10.8.5 What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?.....	223
10.8.6 What Do I Do If a Linux Image File Is Not Pre-Configured When I Use It to Register a Private Image?.....	226
10.8.7 How Do I Enable NIC Multi-Queue for an Image?.....	229
10.8.8 How Do I Make a System Disk Image Support Fast ECS Creation?.....	234
10.8.9 Why Did I Fail to Install Guest OS Drivers on a Windows ECS?.....	235
10.8.10 How Do I Install Native Xen and KVM Drivers?.....	235
10.9 Image Replication.....	242
10.10 Image Deletion.....	243
10.11 Image Encryption.....	243
10.12 Accounts and Permissions.....	243
10.12.1 What Do I Do If I Enabled EPS But Now I Cannot Find Private Images in My Enterprise Project?	243
10.12.2 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?.....	244
10.13 Cloud-Init.....	244
10.13.1 Cloud-Init Installation FAQ.....	244
10.13.2 What Can I Do with a Cloud-Init ECS?.....	248
10.13.3 What Do I Do If Installed NetworkManager and Now I Can't Inject the Key or Password Using Cloud-Init?.....	249
10.13.4 How Do I Install growpart for SUSE 11 SP4?.....	249
10.14 ECS Creation.....	250
10.14.1 Can I Change the Image of a Purchased ECS?.....	250
10.14.2 Can I Change the Specifications Defined by a Private Image When I Use the Image to Create an ECS?.....	251
10.14.3 Can I Specify the System Disk Capacity When I Create an ECS Using an Image?.....	251
10.14.4 What Do I Do If a Partition Is Not Found During the Startup of an ECS Created from an Imported Private Image?.....	251
10.14.5 What Do I Do If the Disks of a CentOS ECS Created from an Image Cannot Be Found?.....	254
10.14.6 What Do I Do If I Enabled Automatic Configuration During Image Registration for an ECS Created from a Windows Image and Now It Won't Start?.....	255

10.14.7 What Do I Do If an Exception Occurs When I Start an ECS Created from an Image Using UEFI Boot?.....	255
10.15 Driver Installation.....	256
10.15.1 Must I Install Guest OS Drivers on an ECS?.....	256
10.15.2 Why Do I Need to Install and Update VirtIO Drivers for Windows?.....	256
10.15.3 Why Did I Fail to Install Guest OS Drivers on a Windows ECS?.....	257
10.15.4 How Do I Install VirtIO Drivers in Windows?.....	257
10.15.5 How Do I Install Native KVM Drivers in Linux?.....	257
10.15.6 How Do I Install Native Xen and KVM Drivers?.....	257
10.16 Image Tags.....	265
10.16.1 How Many Tags Can I Add to an Image?.....	265
10.16.2 How Do I Add, Delete, and Modify Image Tags?.....	265
10.16.3 How Do I Search for Private Images by Tag?.....	266

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) allows you to manage the entire lifecycle of your images. You can create ECSs or BMSs 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.

Figure 1-1 Image types

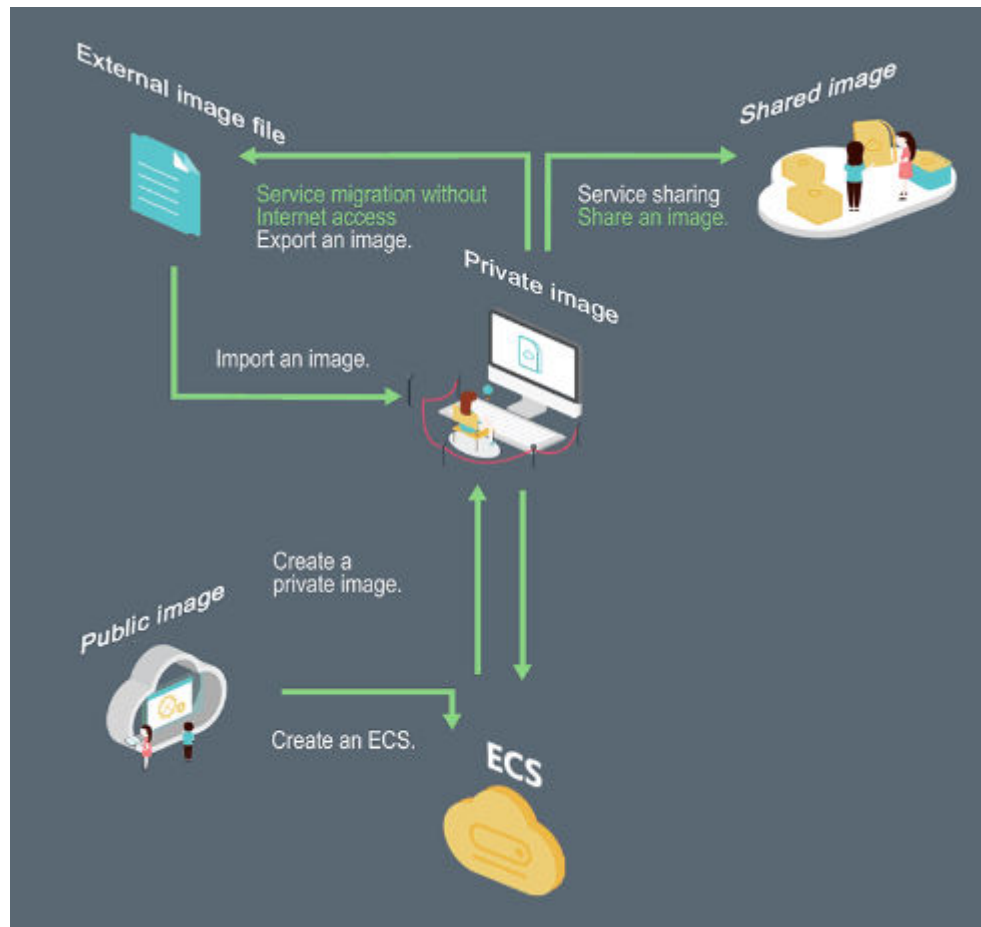


Image Type	Description
Public	A public image is a standard, widely used image. It contains an OS and preinstalled public applications and is available to all users. Public images are very stable and their OS and any included software have been officially authorized for use. If a public image does not 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	<p>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.</p> <p>A private image can be a system disk image, data disk image, ISO image, or full-ECS image.</p> <ul style="list-style-type: none">• 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	<p>A shared image is a private image another user has shared with you.</p> <p>For more information, see "Sharing Images" in <i>Image Management Service User Guide</i>.</p>

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

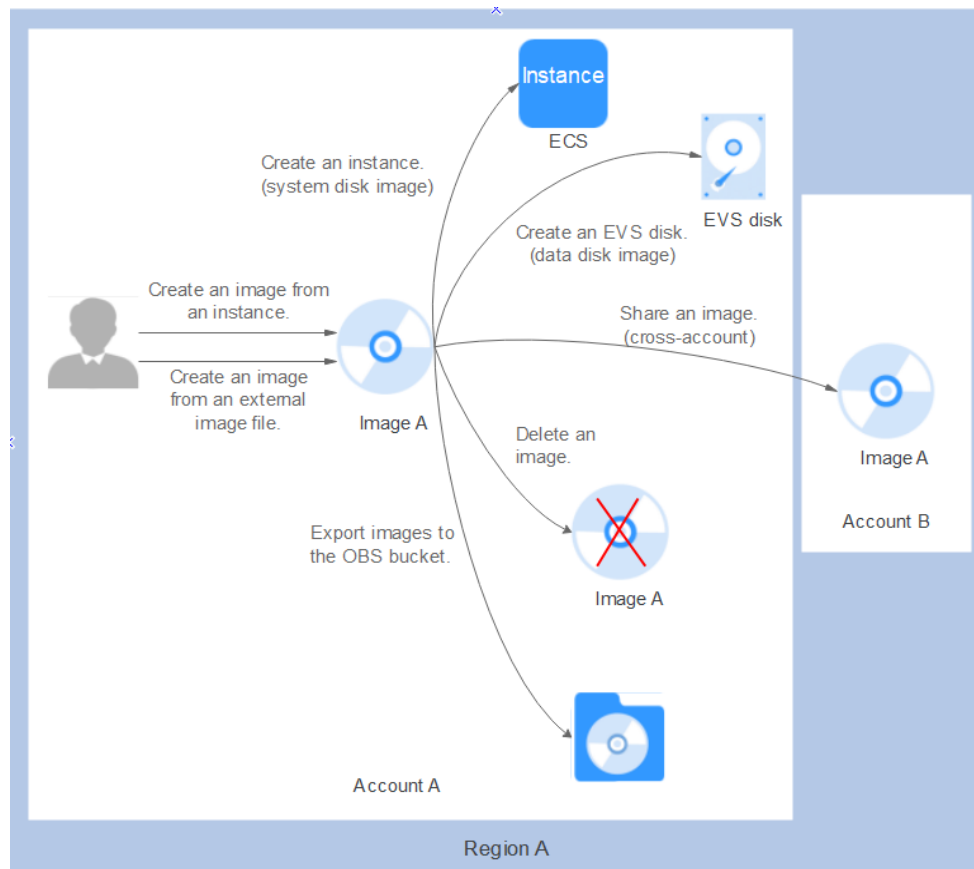
- Migrating servers to the cloud or on the cloud
You can import local images in various formats such as VHD, VMDK, QCOW2, and RAW to the cloud platform and use the images to quickly create cloud servers for service migration to the cloud.
- Deploying a specific software environment
Use shared images 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 it.

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. [Figure 1-2](#) shows the lifecycle of a private image.

Figure 1-2 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.	<ul style="list-style-type: none"> • Creating a System Disk Image from a Windows ECS • Creating a System Disk Image from a Linux ECS

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.	<ul style="list-style-type: none"> • Creating a Windows System Disk Image from an External Image File • Creating a Linux System Disk Image from an External Image File • Fast Import of an Image File
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 from an ECS, a CSBS backup, or a CBR backup.	<ul style="list-style-type: none"> • Creating a Full-ECS Image from an ECS • Creating a Full-ECS Image from a CSBS Backup • 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	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.	<ul style="list-style-type: none"> • Sharing Images • Image Sharing

Feature	Description	Helpful Link
Exporting images	You can export private images to your OBS bucket and download them to your local PC for backup.	<ul style="list-style-type: none"> • Exporting an Image • Image Export
Encrypting images	You can create encrypted images to improve data security. KMS envelope encryption is used. Encrypted images can be created from external image files or encrypted ECSs.	<ul style="list-style-type: none"> • Encrypting Images
Replicating images	By replicating images, you can convert encrypted and unencrypted images into each other or enable some advanced features, for example, fast instance provisioning.	Replicating Images Within a Region
Tagging an image	You can tag your private images for easy management and search.	Managing Tags
Exporting image list	You can export the public or private image list in a given region as a CSV file for 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	<ul style="list-style-type: none"> The ECS must be in the Stopped or Running state.
Creating a full-ECS image from an ECS or a CSBS or CBR backup	<ul style="list-style-type: none"> 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. A full-ECS image cannot be exported, replicated, or shared.

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	<ul style="list-style-type: none"> 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

Item	Constraint
Image size	<p>The image size cannot exceed 128 GB.</p> <p>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.</p> <ul style="list-style-type: none">• For details about how to convert the image file format, see Converting the Image Format Using qemu-img-hw.• For details about fast import, see Fast Import of 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	<ul style="list-style-type: none">• Encrypted images cannot be shared.
Region	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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.
Formats of exported image files	VMDK, VHD, QCOW2, ZVHD, and ZVHD2
Private image status	Normal
Exporting an image	<ul style="list-style-type: none">• 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:<ul style="list-style-type: none">– 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	<ul style="list-style-type: none">• An encrypted image cannot be shared with others.• 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 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 1-10](#) lists the OSs supported by external image files. (If you use a public image to create an ECS and then use the ECS to create a private image, the private image also supports the OSs listed in [Table 1-10](#).)

- For details about OSs supported by BIOS image files, see [Table 1-10](#). If the OS is not included in the table, either **Other Windows(64bit)** or **Other Windows(32bit)** will be used for Windows, and either **Other Linux(64bit)** or **Other Linux(32bit)** for Linux during image registration.
- For more information about the OSs supported by UEFI image files, see [OSs Supporting UEFI Boot Mode](#).

NOTE

- Uploading image files containing OSs not listed in [Table 1-10](#) may fail. You are advised to contact the customer service before uploading these image files.
- When uploading a CoreOS image file, set the OS type to CoreOS. Otherwise, the OS type will be set to **Other (64bit)**. In addition, ensure that coreos-cloudinit has been installed and configured for CoreOS. Automatic system upgrades must be disabled. Otherwise, they may make ECSs created from this image unavailable.

Table 1-10 Supported OSs

OS	Version
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 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 12 64bit SUSE Linux Enterprise Server 11 SP4 64bit SUSE Linux Enterprise Server 11 SP3 64bit SUSE Linux Enterprise Server 11 SP3 32bit SUSE Linux Enterprise Server 11 SP1 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 7.6 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 15.1 64bit openSUSE 15.0 64bit openSUSE 13.2 64bit openSUSE 11.3 64bit

OS	Version
CentOS	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
Debian	Debian GNU/Linux 10.0.0 64bit Debian GNU/Linux 9.3.0 64bit Debian GNU/Linux 9.0.0 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

OS	Version
Fedora	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 2.9 64bit EulerOS 2.5 64bit EulerOS 2.3 64bit EulerOS 2.2 64bit EulerOS 2.1 64bit
CoreOS	CoreOS 1068.10.0 CoreOS 1010.5.0 CoreOS 1298.6.0
openEuler	openEuler 20.03 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](#).

1.6.2 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 1-11](#) lists the OSs that support the UEFI boot mode.

Table 1-11 OSs supporting UEFI boot mode

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

OS	Version
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 2.7 64bit EulerOS 2.5 64bit EulerOS 2.3 64bit EulerOS 2.2 64bit
openEuler	openEuler 20.03 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 1-12 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" in *Elastic Cloud Server API Reference*.

Table 1-13 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 1-14 lists the common operations supported by system-defined permissions for IMS.

Table 1-14 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

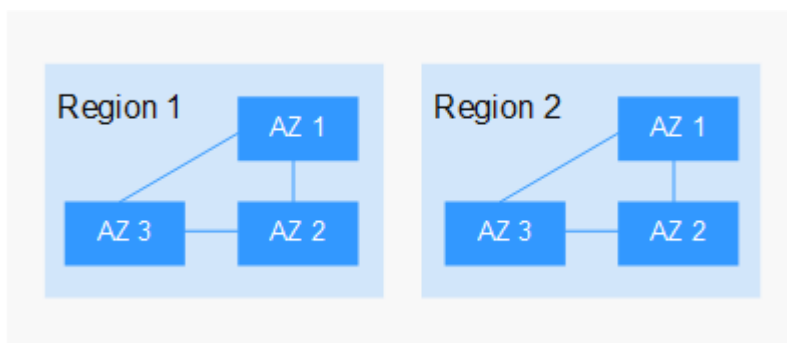
Concept

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 1-3 shows the relationship between regions and AZs.

Figure 1-3 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 1-15](#) lists the common image formats.

Table 1-15 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

Image Format	Description	Remarks
QCOW2	<p>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:</p> <ul style="list-style-type: none"> • 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
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.	<p>A format supported for imported and exported images</p> <p>NOTE VHD image files do not support lazy loading. To import large VHD image files fast, convert them into ZVHD2 files first.</p>
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

Image Format	Description	Remarks
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
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 1-4](#) shows the relationships between IMS and other services.

Figure 1-4 Related Services

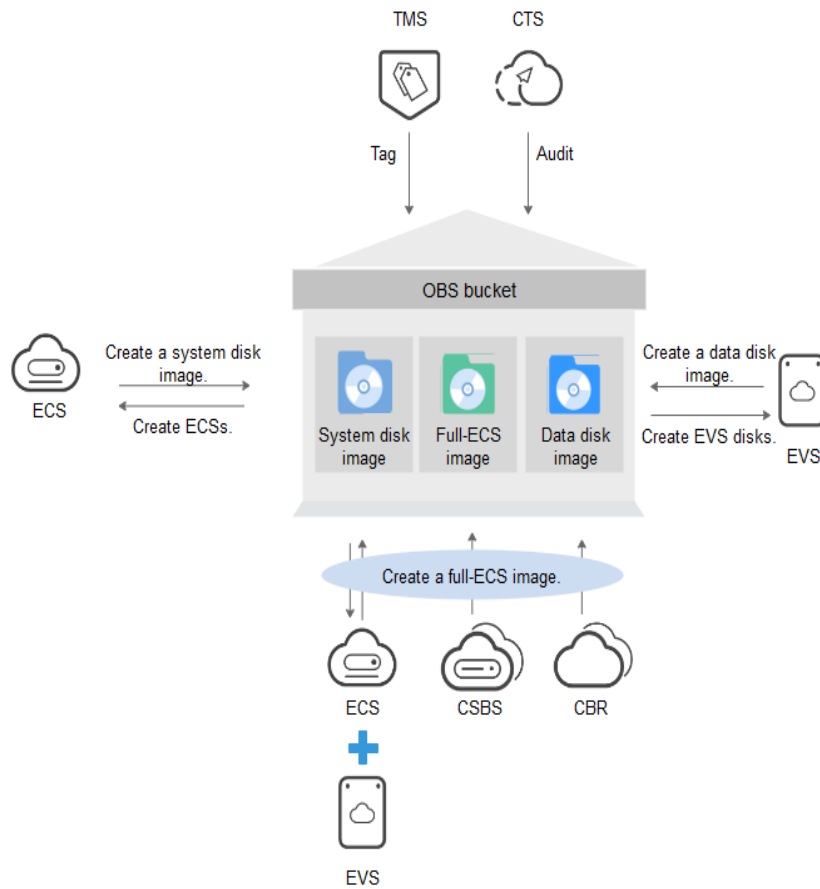


Table 1-16 Related services

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.	<ul style="list-style-type: none"> • Creating an ECS from an Image • Creating a System Disk Image from a Windows ECS • Creating a System Disk Image from a Linux ECS
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

Service	Relationship with IMS	Related Operation
Key Management Service (KMS)	KMS provides the keys used for encrypting images.	Encrypting Images
Cloud Server Backup Service (CSBS)	You can use a CSBS backup to create a full-ECS image.	Creating a Full-ECS Image from a CSBS Backup
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.	Managing Tags
Cloud Trace Service (CTS)	CTS records IMS operations for query, auditing, or backtracking.	Auditing Key Operations

2 Using IAM to Grant Access to IMS

2.1 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.

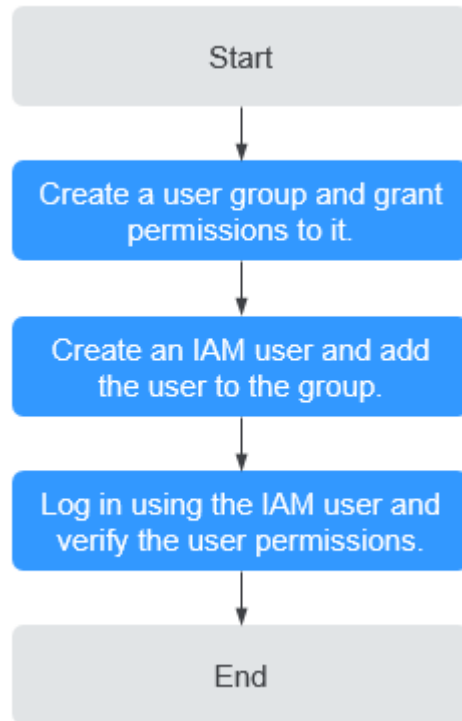
This section uses the **IMS ReadOnlyAccess** permission as an example to describe how to grant permissions to a user. [Figure 2-1](#) shows the process.

Prerequisites

Learn about the permissions (see [IMS Permissions](#)) supported by IMS. For the system permissions of other services, see [System Permissions](#).

Process Flow

Figure 2-1 Process for granting IMS permissions



1. **Create a user group and grant permissions to it.**
Create a user group on the IAM console, and grant the read-only permission to the group by assigning the **IMS ReadOnlyAccess** permission.
2. **Create an IAM user and add it to the user group.**
Create a user on the IAM console and add the user to the group created in 1.
3. Log in and verify permissions.
Log in to the 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.

2.2 Creating a Custom Policy

Scenarios

Custom policies can be created as a supplement to the system permissions of IMS. For the actions supported for custom policies, see "Permissions Policies and Supported Actions" in *Image Management Service API Reference*.

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" in *Identity and Access Management User Guide*. 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"
      ]
    }
  ]
}
```

3 Creating a Private Image

3.1 Introduction

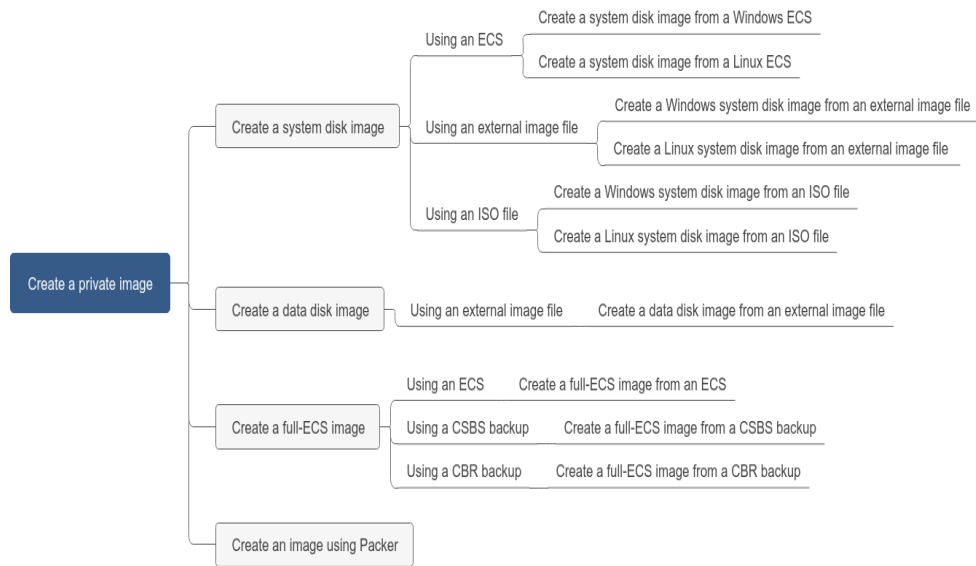
A private image is an image available only to the user who created it. It contains an OS, preinstalled public applications, and a user's personal applications. A private image can be a system disk image, data disk image, or full-ECS image. It can be created from a cloud server or an external image file.

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 Full-ECS Image from an ECS](#)
- [Creating a Full-ECS Image from a CSBS Backup](#)
- [Creating a Windows System Disk Image from an ISO File](#)
- [Creating a Linux System Disk Image from an ISO File](#)

Figure 3-1 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.

3.2 Creating a System Disk Image from a Windows ECS

Scenarios

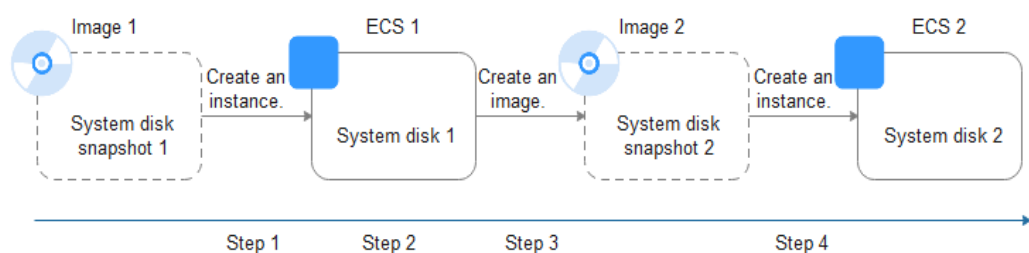
If you have created and configured a Windows ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software 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 3-2 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 [Configuring DHCP](#) and [Enabling Remote Desktop Connection](#).
- Install special drivers. The normal running and advanced functions of some ECSs depend on certain drivers. For example, GPU-accelerated ECSs depend on the Tesla and GRID/vGPU drivers. For details, see [Installing Special Windows Drivers](#).
- 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 PV and VirtIO drivers to ensure that new ECSs created from the image support both KVM and Xen virtualization and to improve network performance.

For details, see steps [2](#) to [5](#) 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 3-1 and **Table 3-2** list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 3-1 Image type and source

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

Table 3-2 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. <ul style="list-style-type: none">- 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.

Parameter	Description
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 **Create Now**.
4. Confirm the settings and click **Submit**.

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.

3.3 Creating a System Disk Image from a Linux ECS

Scenarios

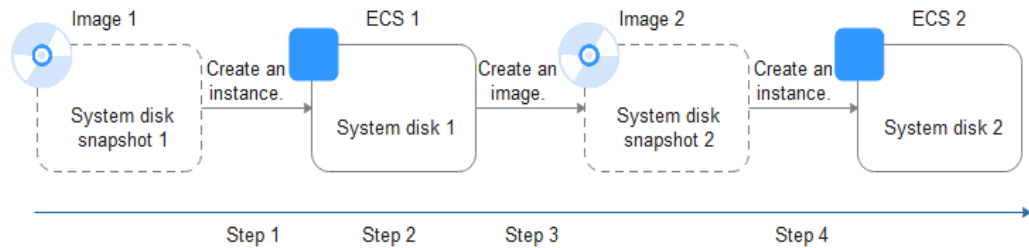
If you have created and configured a Linux ECS based on your service requirements (for example, by installing software and setting up an application environment), you can create a system disk image based on this configured ECS. Then, all new ECSs created from this image will have the same software and 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 3-3 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 [Configuring 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 both Xen and KVM virtualization, the Linux ECS used to create the image has to be modified. For instance, disk identifiers in the GRUB and fstab files need to be UUID and native Xen and KVM drivers need to be installed.
For details, see steps 2 to 6 in [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 3-3](#) and [Table 3-4](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 3-3 Image type and source

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

Table 3-4 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. <ul style="list-style-type: none">- 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 **Create Now**.
4. Confirm the settings and click **Submit**.

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.

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

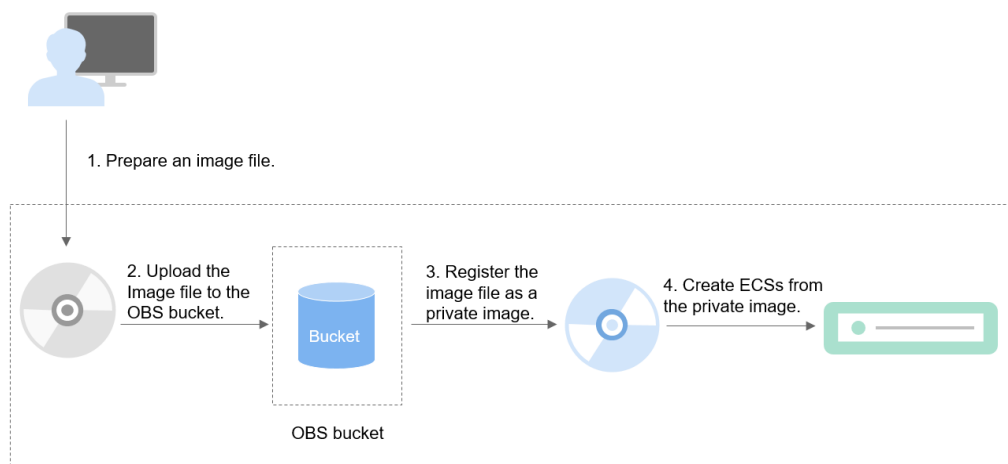
3.4.1 Overview

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 3-4 shows the process of creating a private image.

Figure 3-4 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](#).

3.4.2 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 3-6](#), a bitmap file needs to be generated alongside each RAW image file. The bitmap file is uploaded together with the image file. For details, see [Fast Import of 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 3-5 Initial configuration for an image file

Configuration Item	How to Configure
Network	<p>DHCP must be configured. Otherwise, the ECS startup or network capability will be abnormal. For details, see: Configuring DHCP</p> <p>The following operations are optional:</p> <ul style="list-style-type: none">Enabling NIC multi-queue NIC multi-queue enables multiple vCPUs to process NIC interrupts, thereby improving network PPS and I/O performance. For details, see How Do I Enable NIC Multi-Queue for an Image?
Tools	<p>You are advised to install Cloudbase-Init.</p> <p>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.</p> <p>For details, see Installing and Configuring Cloudbase-Init.</p> <p>If each of your ECSs requires a unique SID in a domain, run Sysprep after Cloudbase-Init is installed. For details, see Running Sysprep.</p>

Configuration Item	How to Configure
Drivers	<p>An ECS can run properly only after Xen Guest OS drivers (PV drivers) and KVM Guest OS drivers (VirtIO drivers) are installed on it. To ensure that ECSs support both Xen and KVM and to improve network performance, PV and VirtIO drivers must be installed for the image.</p> <ul style="list-style-type: none"> • Installing PV drivers • Installing VirtIO drivers <p>For GPU-accelerated ECSs, you need to install special drivers when creating a private image. For details, see Installing Special Windows Drivers.</p>

Image File Properties

Table 3-6 Windows image file properties

Image File Property	Requirement
OS	<ul style="list-style-type: none"> • 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. <p>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.</p>
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	<p>If an image is no larger than 128 GB, import it directly.</p> <p>If an image is between 128 GB and 1 TB, convert the image file into RAW or ZVHD2 and import it using fast import.</p> <ul style="list-style-type: none"> • For details about how to convert the image file format, see Converting the Image Format Using qemu-img-hw. • For details about fast import, see Fast Import of an Image File.

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.

3.4.3 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

- The bucket file and the image to be registered must belong to the same region.
- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to an OBS bucket.
- The storage class of the OBS bucket must be Standard.
- Do not upload image files using the OBS server-side encryption function.
- 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.

3.4.4 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 3-7](#) and [Table 3-8](#) list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.

Table 3-7 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select Image File for Source . Select the bucket storing the image file from the list and then select the image file.
Enable Fast Create	<p>This parameter is available only when you select a ZVHD2 or RAW image file.</p> <p>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.</p> <p>NOTE To learn how to convert image file formats and generate bitmap files, see Fast Import of an Image File.</p>

Table 3-8 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.
Boot Mode	<p>This parameter is optional. The value can be BIOS or UEFI. For details about the differences between them, see How Is BIOS Different from UEFI?</p> <p>For details about which OSs support UEFI boot, see OSs Supporting UEFI Boot Mode.</p> <p>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 right 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.</p>

Parameter	Description
OS	<p>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.</p> <p>NOTE</p> <ul style="list-style-type: none">- If the system identifies that the OS in the image file is different from the one you select here, the identified OS prevails.- If the system fails to identify an OS, the OS you select will be used.- If the OS you selected or identified by the system is inconsistent with the actual one, ECSs created from the image file may not work properly.
System Disk (GB)	<p>The system disk capacity (value range: 40 GB to 1024 GB). Ensure that this value is not less than the system disk capacity in the image file.</p> <p>NOTE</p> <p>If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk capacity based on Why Did My VHD Upload Fail? Why Does the System Say the System Disk in the VHD Image File Is Larger Than What I Specified on the Management Console?</p>
Data Disk (GB)	<p>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.</p> <p>To add data disks, click , configure 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.</p> <p>A maximum of three data disks can be added.</p>
Name	Set a name for the image.
Encryption	<p>(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.</p> <p>For how to encrypt an image, see Creating Encrypted Images.</p>
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 **Create Now**, confirm the configurations, and click **Submit**.

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

3.4.5 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.
- **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.

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

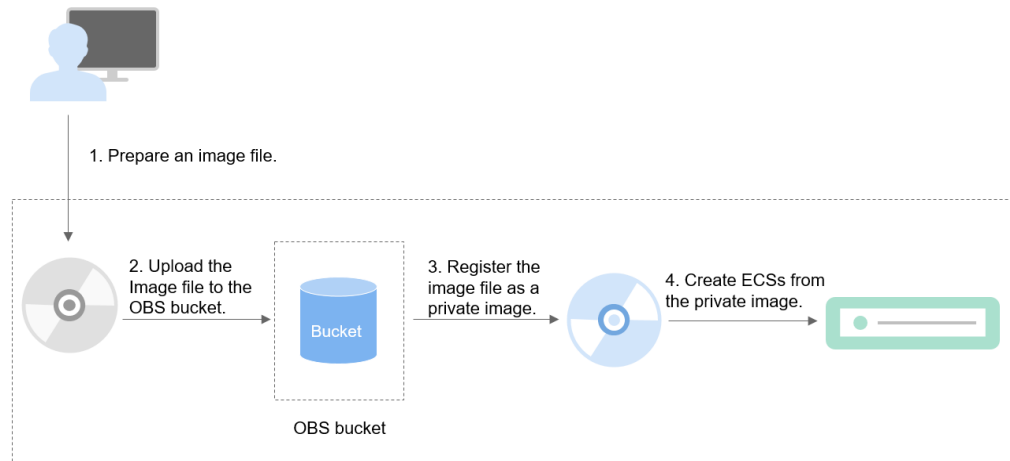
3.5.1 Overview

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 3-5 shows the process of creating a private image.

Figure 3-5 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](#).

3.5.2 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 3-10](#), a bitmap file needs to be generated alongside each RAW image file. The bitmap file is uploaded together with the image file. For details, see [Fast Import of 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 3-9 Initial configuration for an image file

Configuration Item	How to Configure
Network	<p>DHCP must be configured. Otherwise, the ECS startup or network capability will be abnormal. For details, see:</p> <ul style="list-style-type: none">• Deleting files from the network rule directory• Configuring DHCP <p>The following value-added operations are optional:</p> <ul style="list-style-type: none">• Enabling NIC multi-queue NIC multi-queue enables multiple vCPUs to process NIC interrupts, thereby improving network PPS and I/O performance. For details, see How Do I Enable NIC Multi-Queue for an Image?
Tools	<p>You are advised to install Cloud-Init.</p> <p>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 customize 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 these custom configurations to your ECSs, and you will have to use the original password in the image file to log in to the ECSs.</p> <p>For details, see Installing Cloud-Init.</p>
Drivers	Installing native KVM drivers
File system	<ul style="list-style-type: none">• Changing disk identifiers in the GRUB file to UUID• Changing disk identifiers in the fstab file to UUID
Data disks	<p>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.</p> <p>For details, see Detaching Data Disks from an ECS.</p>

Image File Properties

Table 3-10 Linux image file properties

Image File Property	Requirement
OS	<ul style="list-style-type: none">• 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. <p>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.</p>
Image format	VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD
Image size	Maximum file size: 128 GB

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.

3.5.3 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

- The bucket file and the image to be registered must belong to the same region.
- Only unencrypted external image files or those encrypted using SSE-KMS can be uploaded to an OBS bucket.
- The storage class of the OBS bucket must be Standard.
- Do not upload image files using the OBS server-side encryption function.
- 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.

3.5.4 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 3-11 and **Table 3-12** list the parameters in the **Image Type and Source** and **Image Information** areas, respectively.


Table 3-11 Image type and source

Parameter	Description
Type	Select System disk image .
Source	Select Image File for Source . Select the bucket storing the image file from the list and then select the image file.

Parameter	Description
Enable Fast Create	<p>This parameter is available only when you select a ZVHD2 or RAW image file.</p> <p>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.</p> <p>NOTE To learn how to convert image file formats and generate bitmap files, see Fast Import of an Image File.</p>

Table 3-12 Image information

Parameter	Description
Enable automatic configuration	<p>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?</p>
Function	<p>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.</p>
Boot Mode	<p>This parameter is optional. The value can be BIOS or UEFI. For details about the differences between them, see How Is BIOS Different from UEFI?</p> <p>For details about which OSs support UEFI boot, see OSs Supporting UEFI Boot Mode.</p> <p>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 right 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.</p>

Parameter	Description
OS	<p>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.</p> <p>NOTE</p> <ul style="list-style-type: none"> - If the system identifies that the OS in the image file is different from the one you select here, the identified OS prevails. - If the system fails to identify an OS, the OS you select will be used. - If the OS you selected or identified by the system is inconsistent with the actual one, ECSs created from the image file may not work properly.
System Disk (GB)	<p>The system disk capacity (value range: 40 GB to 1024 GB). Ensure that this value is not less than the system disk capacity in the image file.</p> <p>NOTE</p> <p>If the uploaded VHD image is generated using qemu-img or similar tools, check the system disk capacity based on Why Did My VHD Upload Fail? Why Does the System Say the System Disk in the VHD Image File Is Larger Than What I Specified on the Management Console?</p>
Data Disk (GB)	<p>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.</p> <p>To add data disks, click , configure 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.</p> <p>A maximum of three data disks can be added.</p>
Name	Set a name for the image.
Encryption	<p>(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.</p> <p>For how to encrypt an image, see Creating Encrypted Images.</p>
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 **Create Now**, confirm the configurations, and click **Submit**.

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

3.5.5 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.
- **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.

3.6 Creating a Data Disk Image from an External Image File

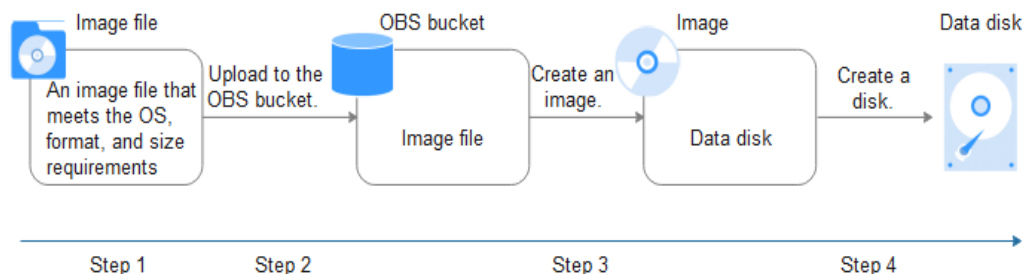
Scenarios

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 3-6 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 **Data disk image** for **Type**.
3. Select **Image File** for **Source**. Select the bucket storing the image file from the list and then select the image file.
4. To register the image file using Fast Create, select **Enable Fast Create**.

NOTE

- Currently, fast import is only available for ZVHD2 and RAW image files.
- For how to convert image file formats and generate bitmap files, see [Fast Import of 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 **Create Now**.
 7. Confirm the settings and click **Submit**.

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

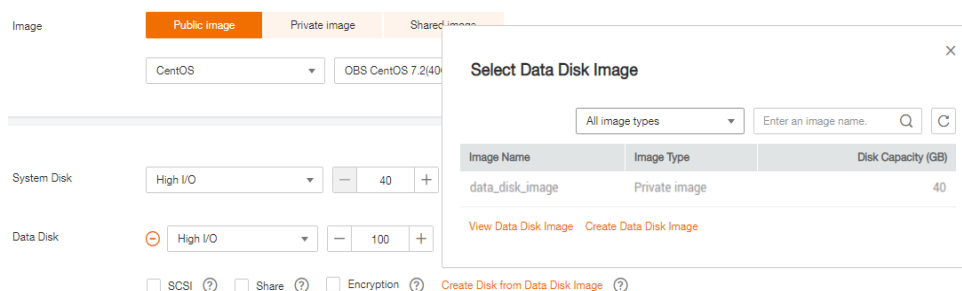
If you want to use the created data disk image to create an EVS disk and attach it to an ECS, you can perform either of the following operations:

- 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 3-7 Adding data disks



3.7 Creating a Full-ECS Image from an ECS

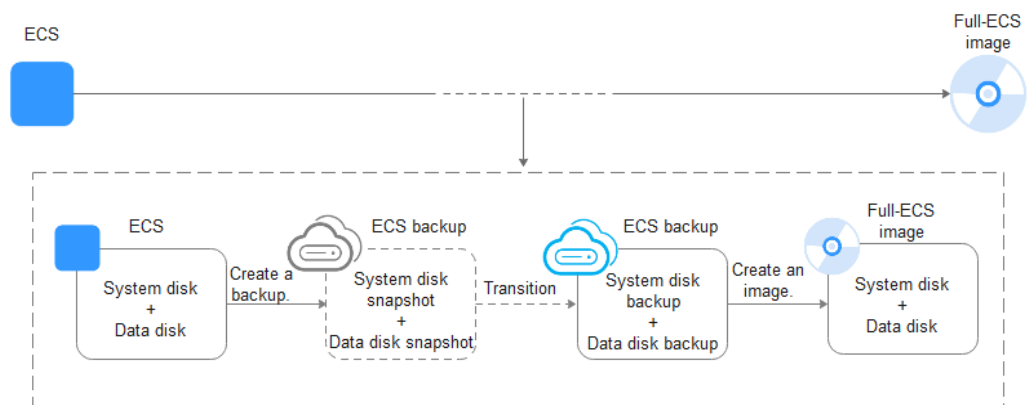
Scenarios

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 3-8 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 3-8**, 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.
- An ECS used to create a full-ECS image cannot contain a Dedicated Storage Service disk.
- A full-ECS image cannot be exported, replicated, or shared.
- 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 3-13 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.

- Execute **cmd.exe** and run the following command to query the current SAN policy of the ECS:
diskpart
- 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**.
- Run the following command to change the SAN policy of the ECS to **OnlineAll**:
san policy=onlineall

Procedure

Step 1 Access the IMS console.

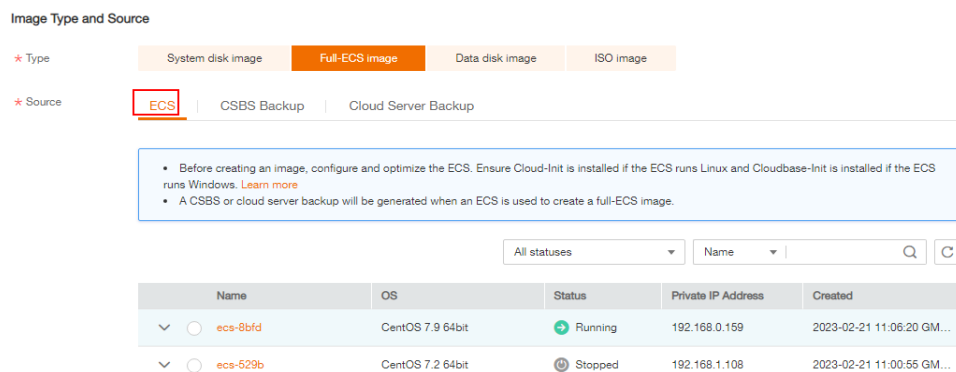
1. Log in to the management console.

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 **Full-ECS image** for **Type**.
3. Select **ECS** for **Source** and then select an ECS from the list.

Figure 3-9 Creating a full-ECS image using an ECS



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

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 AZX** 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. (**AZX** 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 AZX** 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 3-10 Full-ECS image status

Name	Status	OS Type	OS	Image Type
full-image-ecs-00	Normal Available in AZ1	Linux	Ubuntu 16.04 server 64bit	Full-ECS image(x86)

----End

Follow-up Procedure

- If you want to use the full-ECS image to create ECSs, click **Apply for Server** 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.

3.8 Creating a Full-ECS Image from a CSBS Backup

Scenarios

Create a full-ECS image from a CSBS backup. This image can then be used to create ECSs.

Constraints

- When creating a full-ECS image from a CSBS backup, ensure that the source ECS of the CSBS backup has been properly configured, or the image creation may fail.
- If an ECS is in **Stopped** state, do not start it when you are using it to create a full-ECS image.
- A CSBS backup used to create a full-ECS image cannot have shared disks.
- Only an available CSBS backup can be used to create a full-ECS image. A CSBS backup can be used to create only one full-ECS image.
- A full-ECS image cannot be exported, replicated, or shared.

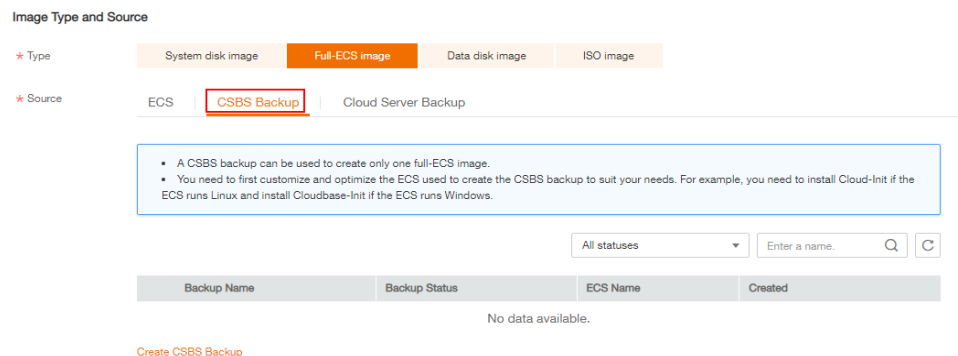
Procedure

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 **Full-ECS image** for **Type**.
3. Select **CSBS Backup** for **Source** and then select a backup from the list.

Figure 3-11 Creating a full-ECS image using a CSBS backup

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

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

- If you want to use the full-ECS image to create ECSs, click **Apply for Server** 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 a full-ECS image with other tenants, you must migrate the associated backup to CBR first because only full-ECS images created from CBR backups can be shared.

3.9 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

- CBR provides the backup service for EVS disks, BMSs, and ECSs. 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 cannot be exported, replicated, or shared.

Procedure

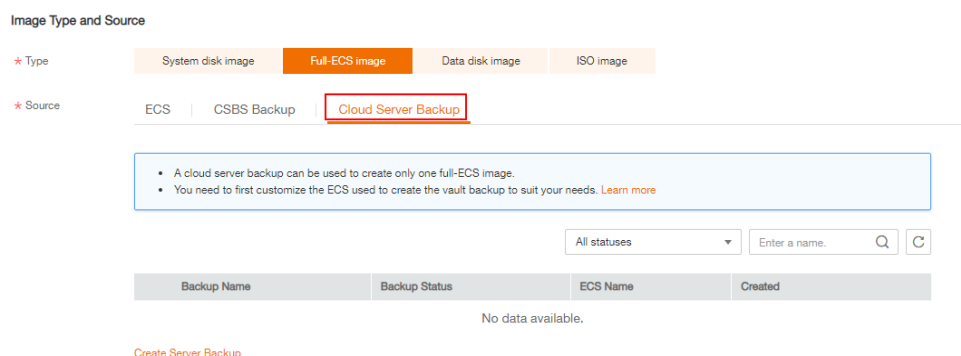
Step 1 Access the IMS console.

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 **Full-ECS image** for **Type**.
3. Select **Cloud Server Backup** for **Source** and then select an ECS from the list.

Figure 3-12 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 **Create Now**.
6. Confirm the settings and click **Submit**.

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 Server** in the **Operation** column. On the displayed page, select **Private image** and then select the 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.
- 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.

3.10 Creating a Windows System Disk Image from an ISO File

3.10.1 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.

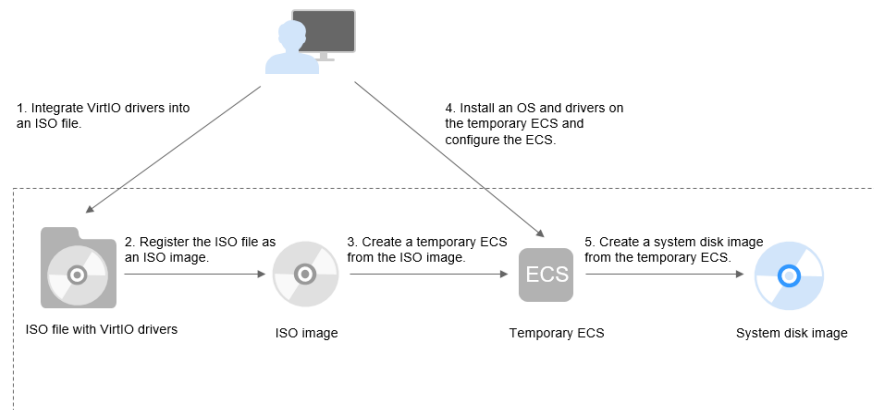
NOTE

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 3-13 shows the process of creating a Windows system disk image from an ISO file.

Figure 3-13 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, PV drivers, 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.

3.10.2 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.
- VirtIO is a standard interface for VMs to access host devices. It is used to improve the I/O performance between VMs and hosts. For details about VirtIO, see [VirtIO](#). For details about open source code of virtio-win/kvm-guest-drivers-windows, see <https://github.com/virtio-win/kvm-guest-drivers-windows>.

Prerequisites

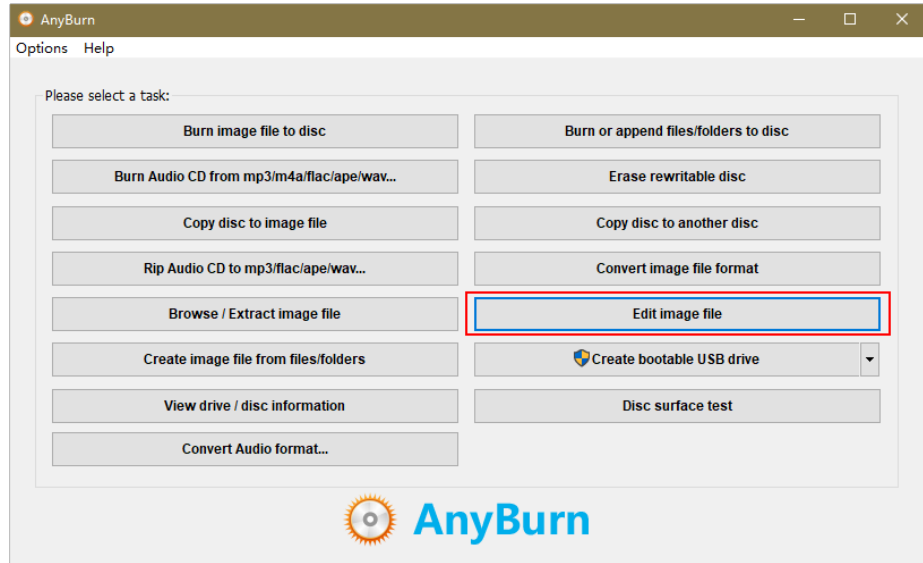
You have obtained an ISO file.

NOTE

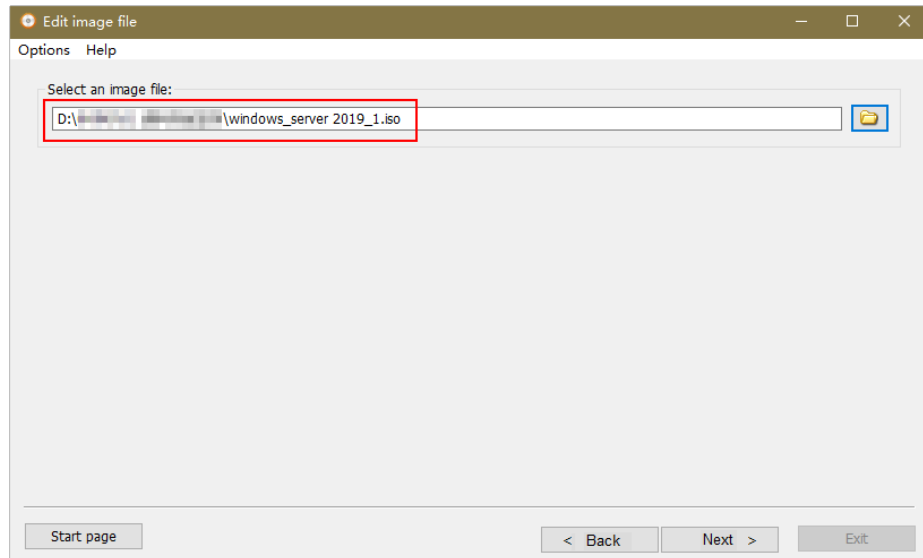
The ISO file name can contain only letters, digits, hyphens (-), and underscores (_).

AnyBurn

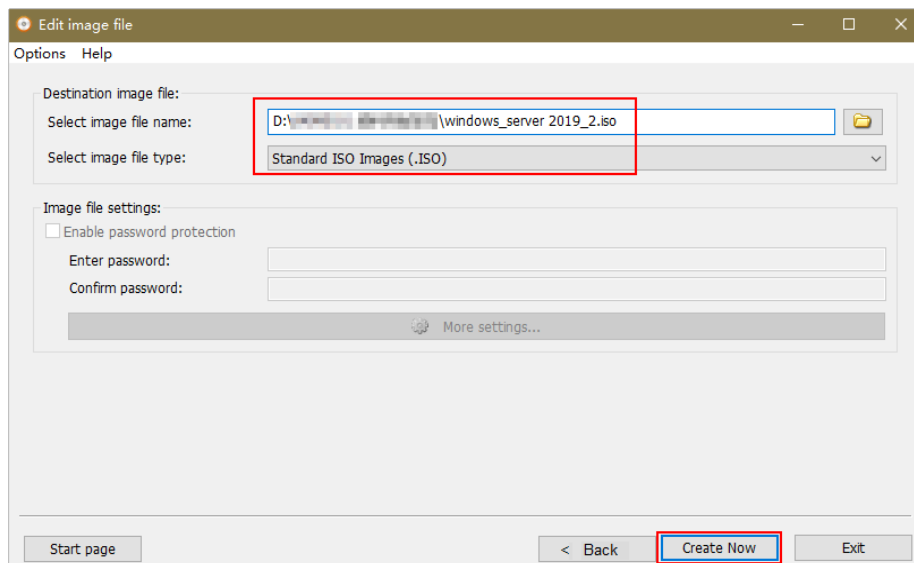
1. Download [AnyBurn](#) and install it on your local PC.
2. Download VirtIO drivers.
<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/stable-virtio/virtio-win.iso>
Other versions:
<https://fedorapeople.org/groups/virt/virtio-win/direct-downloads/archive-virtio/>
3. Use AnyBurn to open 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. Decompress the **virtio-win.iso** file downloaded in [2](#).
 - b. Click **Add**. Select all the decompressed files to add them to the parent node of the ISO file, and click **Next**.
 - c. 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.

3.10.3 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, encrypted, or exported.

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 **ISO image** for **Type**.
3. In the image file list, select the bucket and then the image file.

Figure 3-14 Creating a private image from an ISO file

Image Type and Source

* Type System disk image Full-ECS image Data disk image ISO image

* Source Image File

- The image file used to create a private image must be uploaded to an OBS bucket. Supported file types include ISO. A new image will be created with a format and size different from your original image file. [Learn more](#)

Bucket Name	Created
yrdyyr	2023-03-05 21:36:43 GMT+08:00
2023-2	2023-01-06 22:24:17 GMT+08:00

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

Figure 3-15 Configuring image information

Image Information

Boot Mode BIOS UEFI

The boot mode must be the same as that of the OS contained in the image file. Otherwise, ECSs created from this system disk image will fail to start. [View differences between BIOS and UEFI](#)

* OS --Select OS-- --Select OS version--

* System Disk (GB) -- 1-1,024 + Ensure that the entered value is greater than or equal to the system disk size of the image file.

* Name

Tag It is recommended that you use TMS's predefined tag function to add the same tag to different cloud resources. [View predefined tags](#)

You can add 10 more tags.

Description

0/1,024

- **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.
 - **Tag:** (Optional) Add a tag to the image to be created.
 - **Description:** (Optional) Enter image description as needed.
5. Click **Create Now**.
 6. Confirm the settings and click **Submit**.

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

3.10.4 Creating a Windows ECS from an ISO Image

Scenarios

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

Constraints

Dedicated Cloud (DeC) users cannot create ECSs from ISO images.

If the DeC service is enabled for a user in a specified region, the user will be a DeC user.

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.

NOTE

If you are a **DeC** user, the **Create ECS** button in the **Operation** column will be unavailable for you because a DeC user cannot use an ISO image to create an ECS.

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.

3.10.5 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

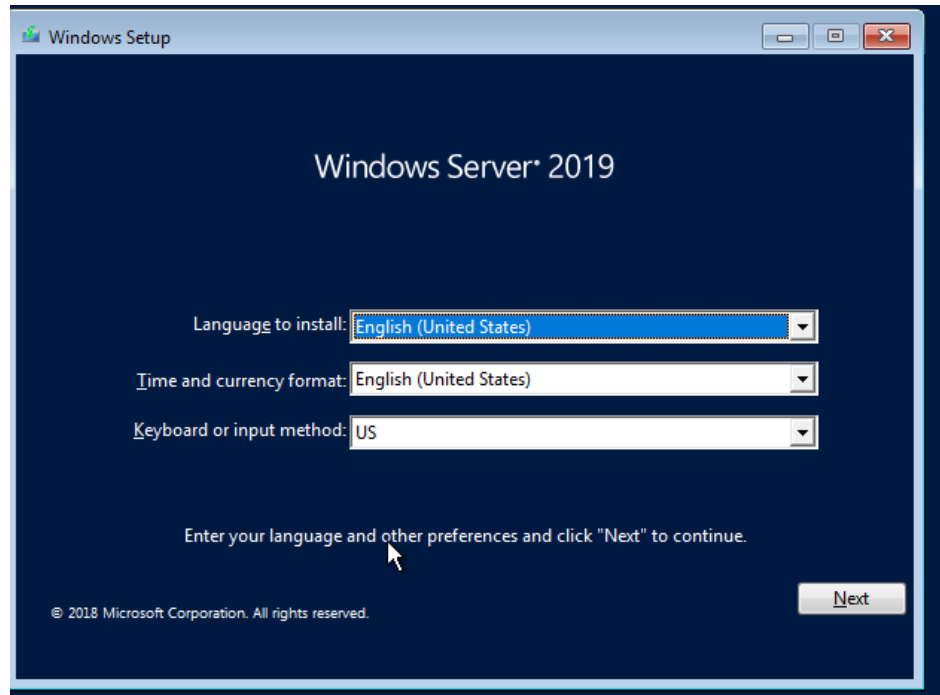
CAUTION

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

Step 1 Install the Windows OS.

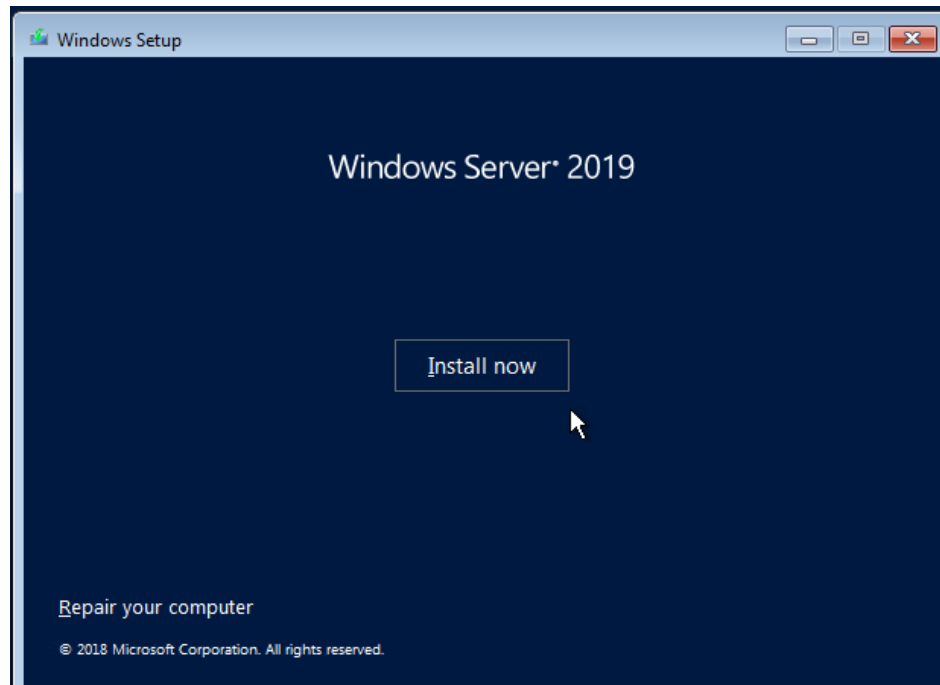
1. Configure Windows setup.

Figure 3-16 Windows setup



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

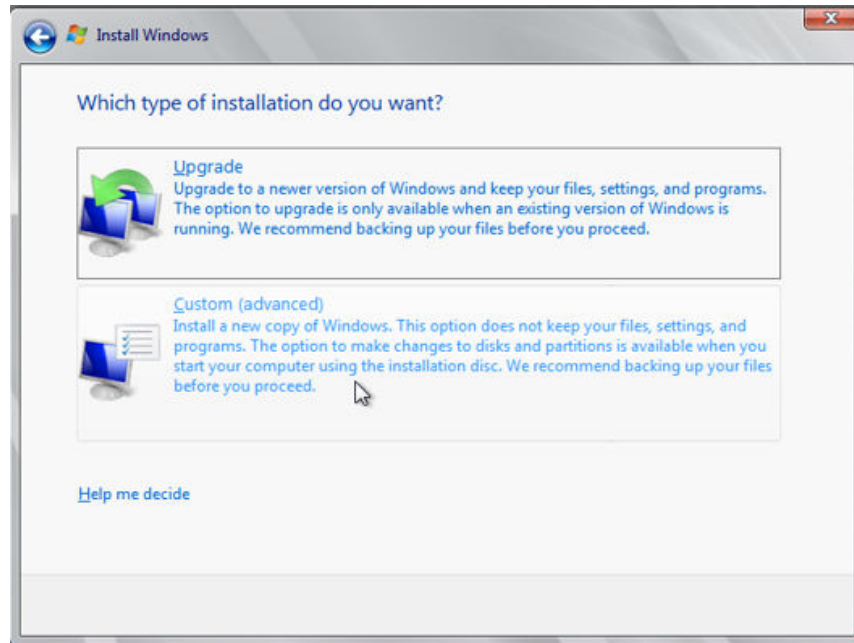
Figure 3-17 Installation confirmation



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

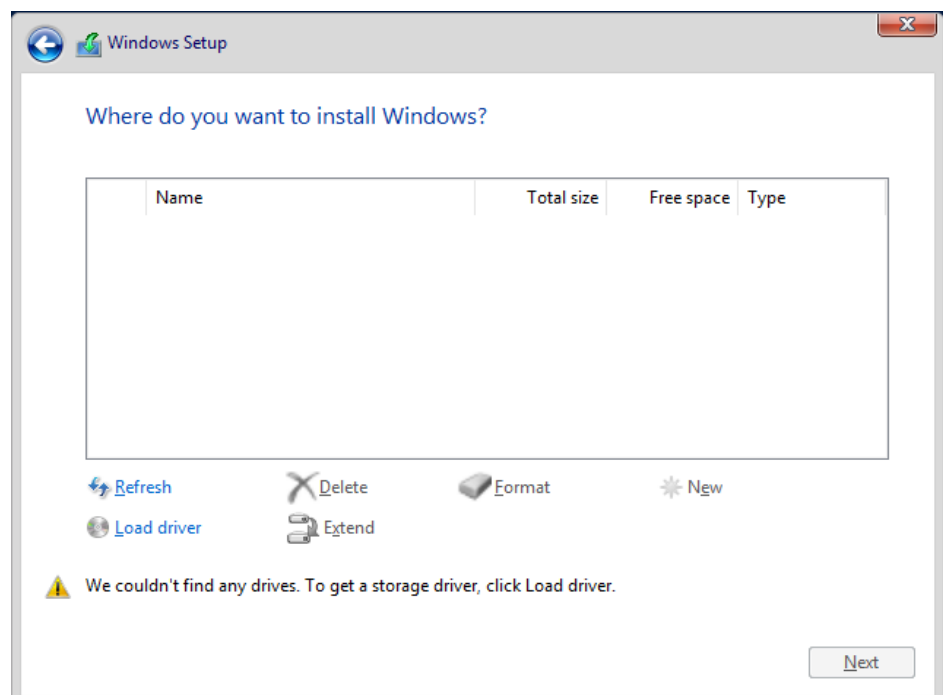
5. Select **I accept the license terms**, and click **Next**.
The **Which type of installation do you want?** dialog box is displayed.

Figure 3-18 Installation type



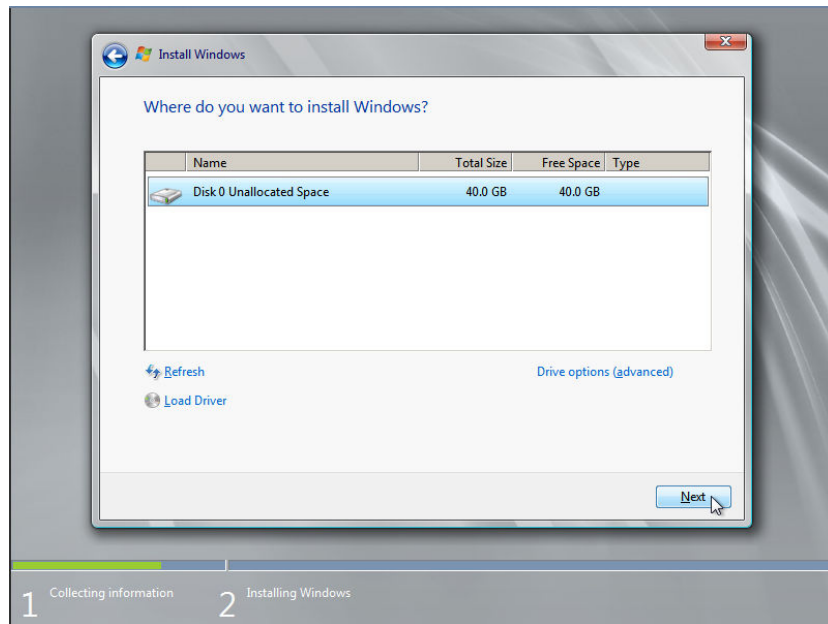
6. Select **Custom (advanced)**.
The **Where do you want to install Windows?** dialog box is displayed.
 - If the system displays a message indicating that no driver is found, go to **Step 1.7**.

Figure 3-19 Installation path



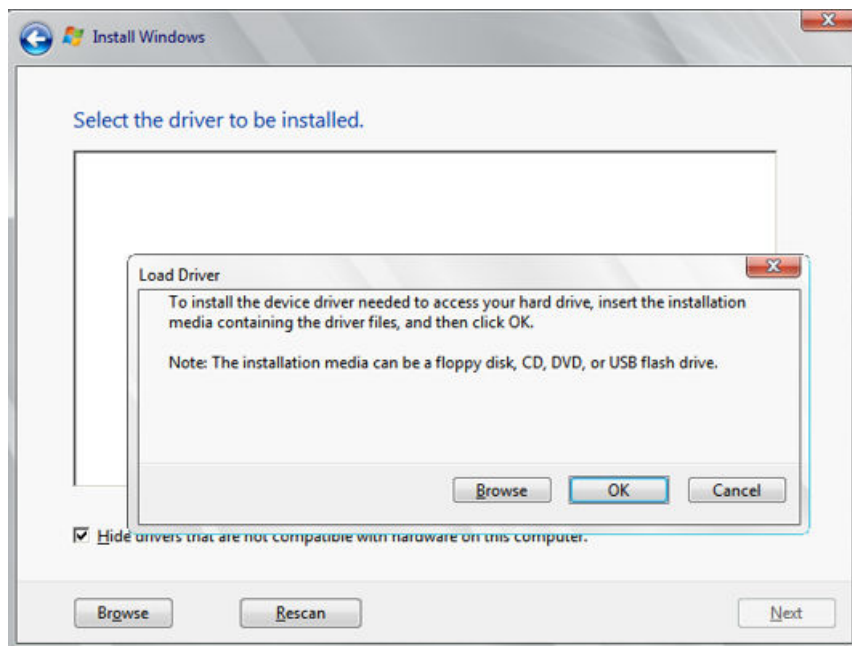
- If a disk is displayed, go to **Step 1.9**.

Figure 3-20 Installation path



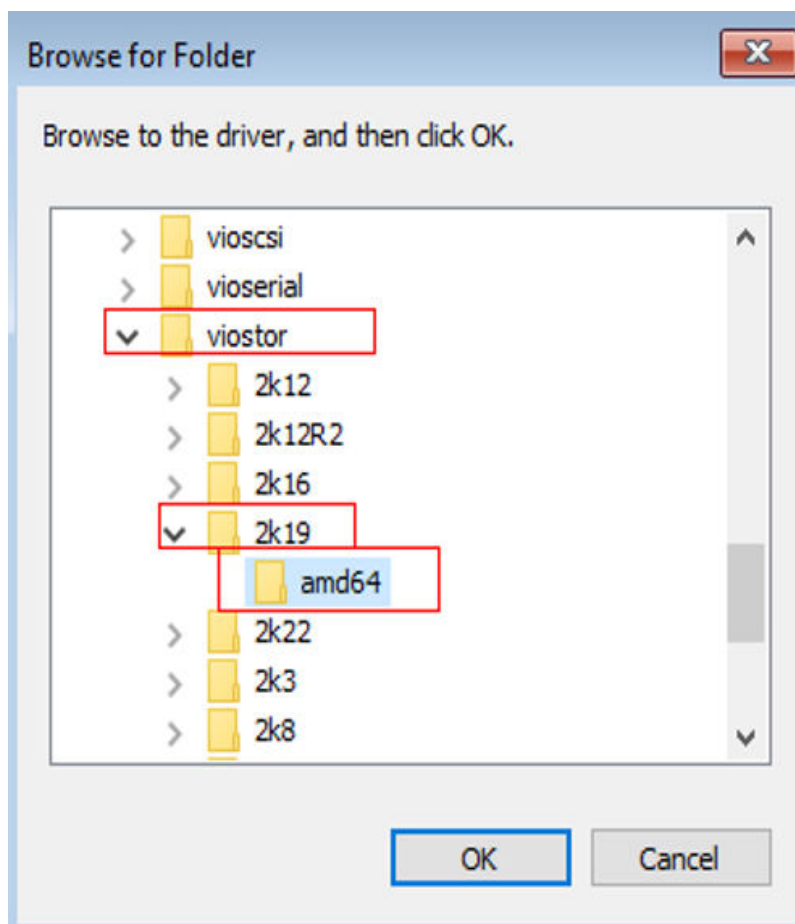
7. Click **Load driver** and then **Browse**.

Figure 3-21 Loading drivers



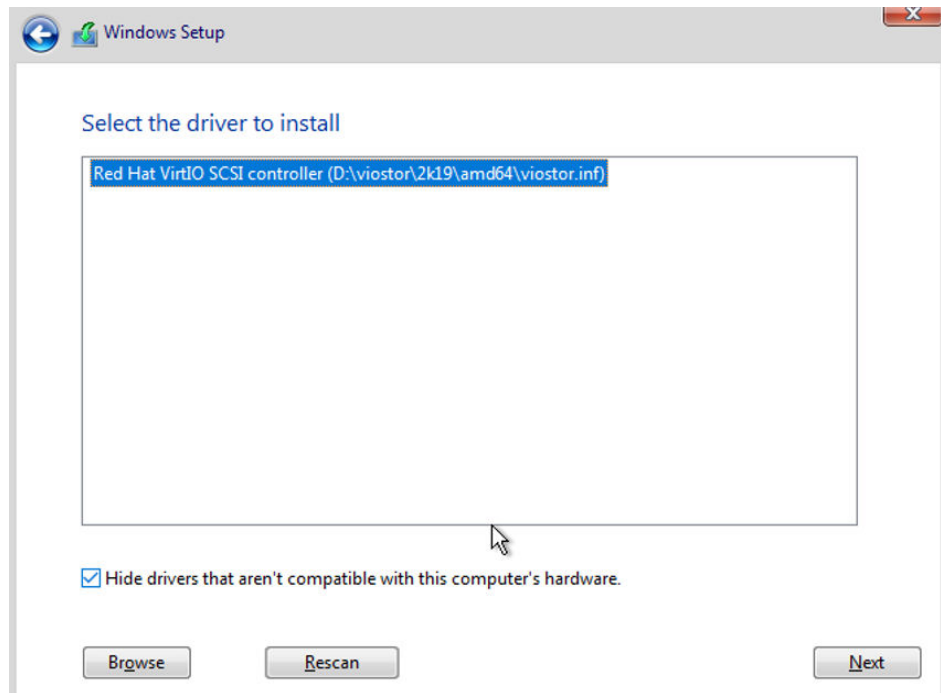
8. Select a driver based on the disk type.
 - If the disk type is VBD, choose **viostor > 2k19 > amd64** and click **OK**.

Figure 3-22 Browsing for a folder



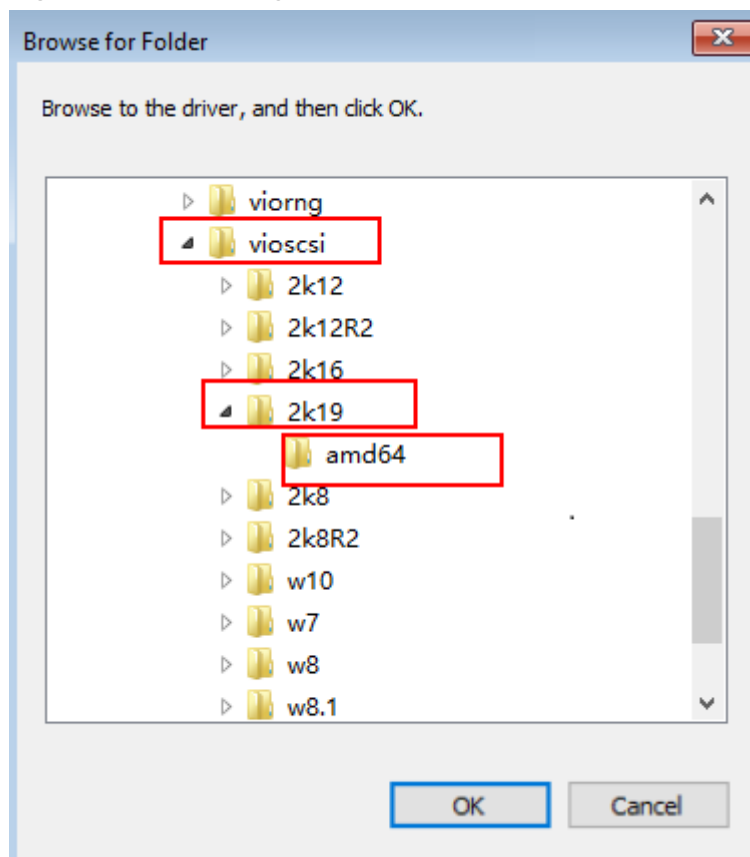
Select the **viostor.inf** driver and click **Next**.

Figure 3-23 Selecting the driver to install



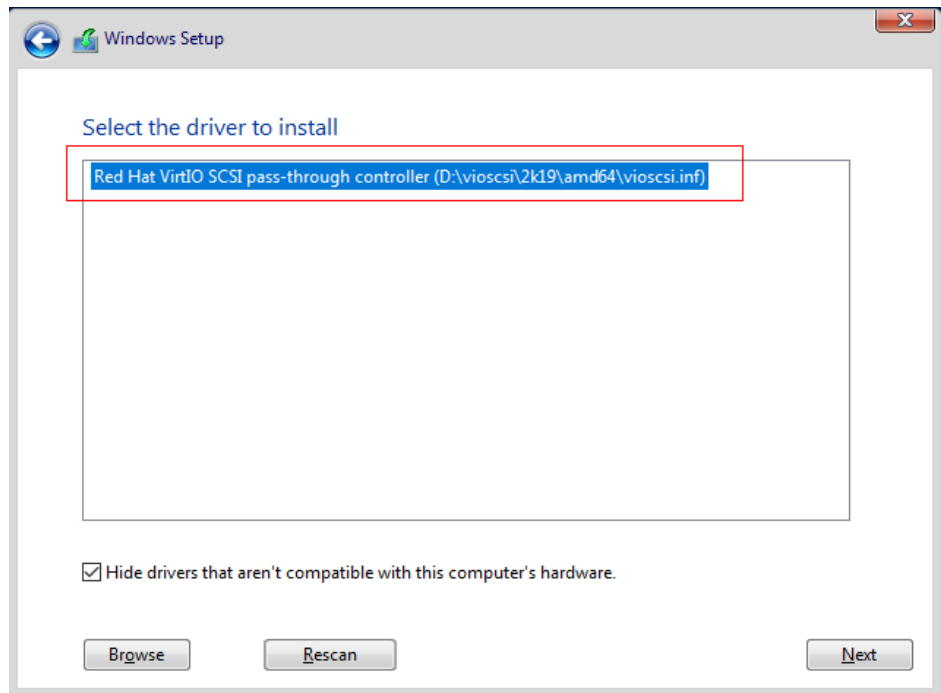
- If the disk type is SCSI, choose **vioscsi** > **2k19** > **amd64** and click **OK**.

Figure 3-24 Browsing for a folder



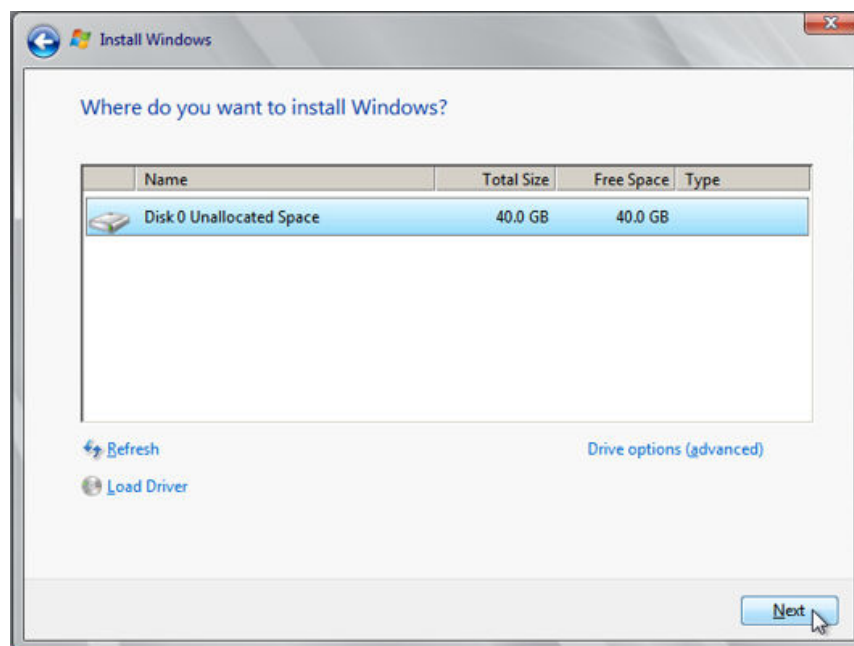
Select the **vioscsi.inf** driver and click **Next**.

Figure 3-25 Selecting the driver to install



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

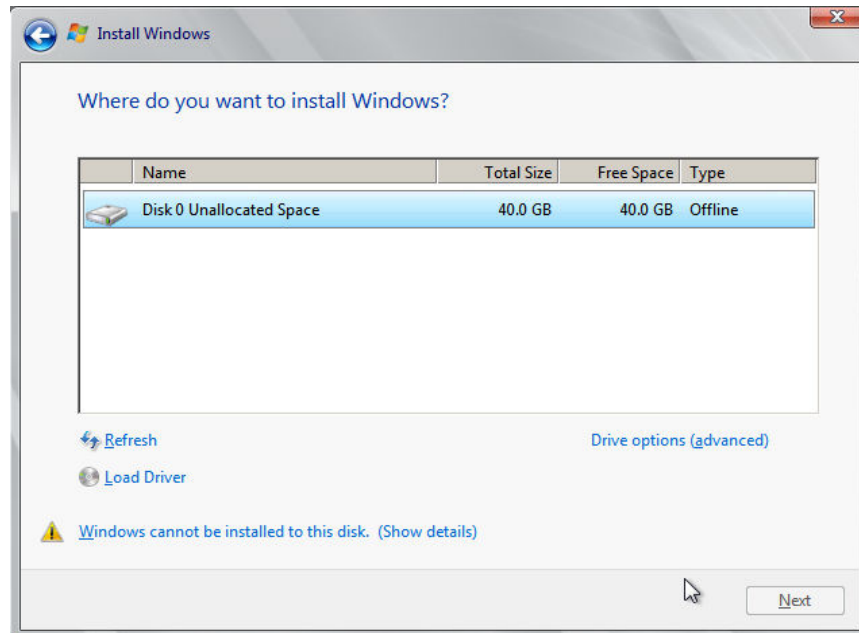
Figure 3-26 Installation path



NOTE

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

Figure 3-27 Offline disk



10. The **Installing Windows** dialog box is displayed, and the OS installation starts.

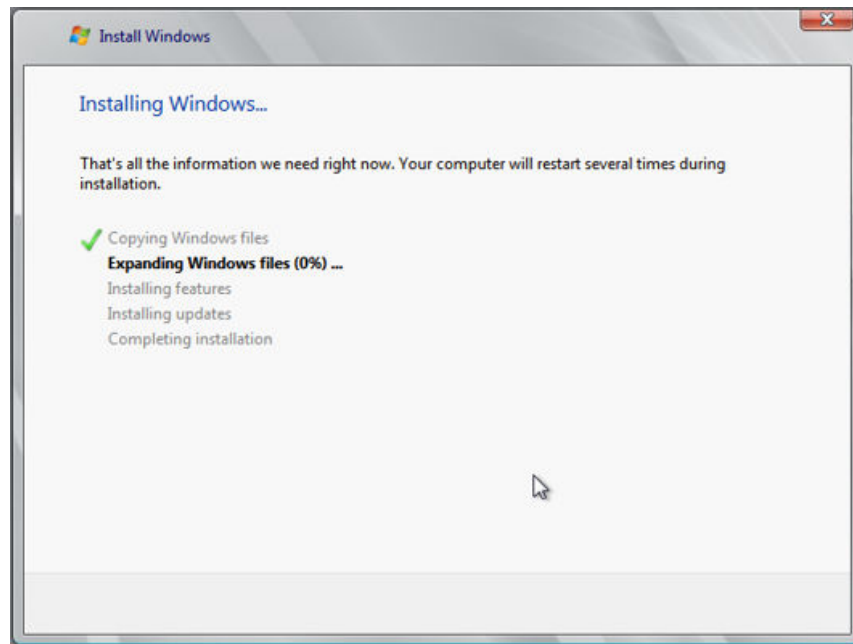
The installation takes about 50 minutes. The ECS restarts during the installation. After the ECS successfully restarts, log in to it again and configure the OS as prompted.

NOTE

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

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

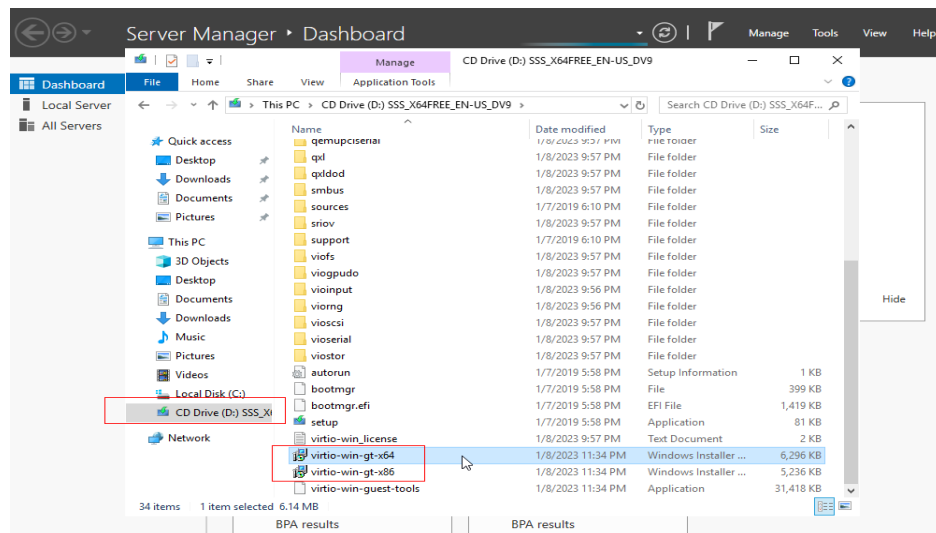
Figure 3-28 Installation progress



Step 2 Install drivers.

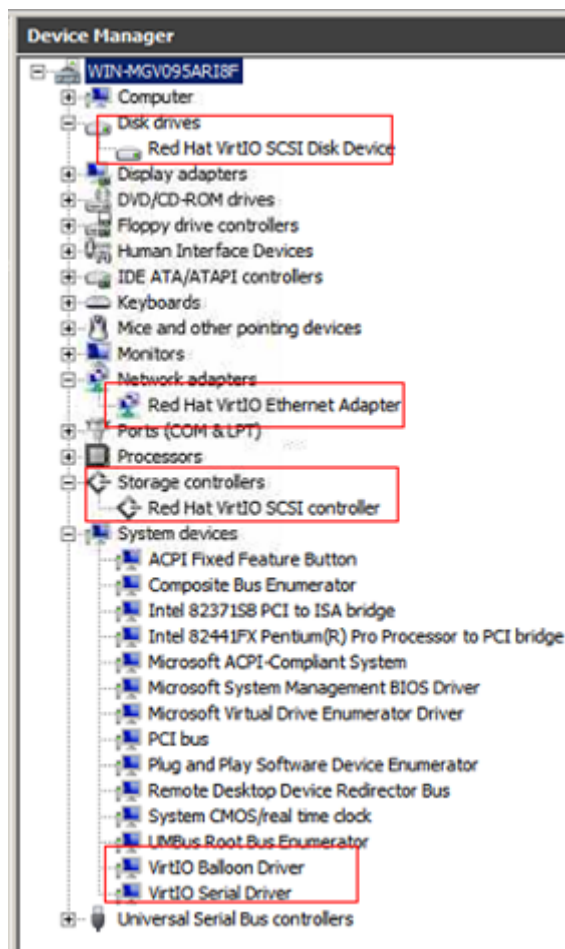
1. Open **Computer** and double-click the CD drive.

Figure 3-29 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 3-30 Device Manager



----End

3.10.6 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.

NOTE

The Guest OS drivers are VirtIO and PV drivers. VirtIO drivers have been installed on the ECS in the preceding section, so this section only describes how to install PV drivers.

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 DHCP is configured. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in [Configuring 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 PV drivers. For details, see [Installing PV Drivers](#).
After drivers are installed, you need to clear system logs. For details, see [Clearing System Logs](#).
4. 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](#).
5. (Optional) Configure value-added functions.
 - Enable NIC multi-queue. For details, see [How Do I Enable NIC Multi-Queue for an Image?](#)

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.

3.11 Creating a Linux System Disk Image from an ISO File

3.11.1 Overview

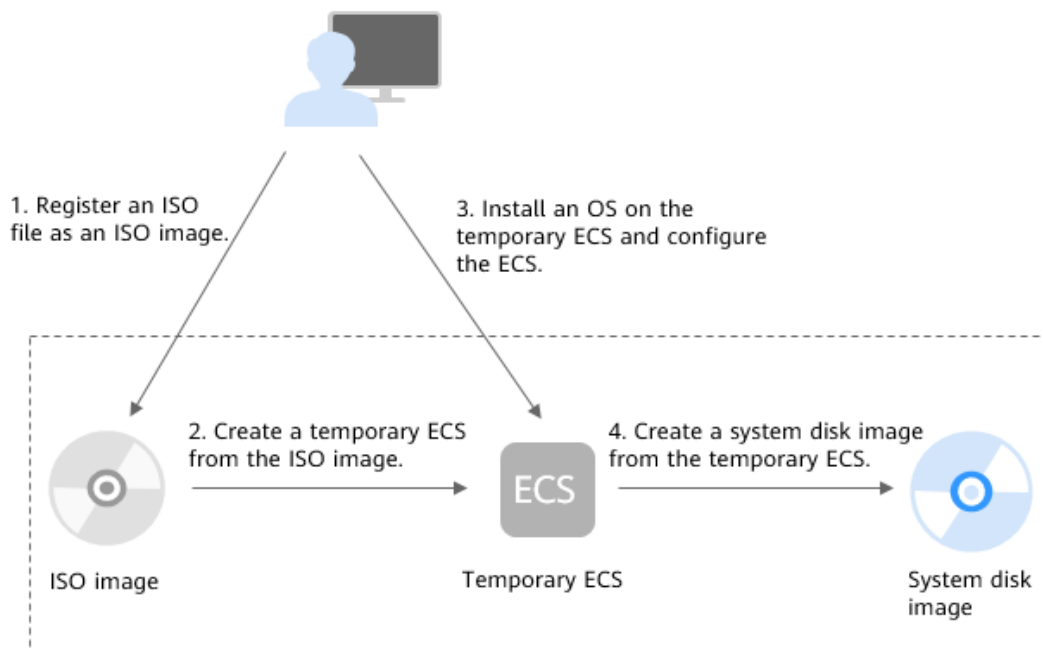
An ISO file is a disk image of an optical disc. A large number of data files can be compressed into a single ISO file. Likewise, to access the files stored in an ISO, the ISO file needs to be decompressed. For example, you can use a virtual CD-ROM to open an ISO file, or burn the ISO file to a CD or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

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

Creation Process

[Figure 3-31](#) shows the process of creating a Linux system disk image from an ISO file.

Figure 3-31 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 NICs, 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.

3.11.2 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, encrypted, or exported.

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

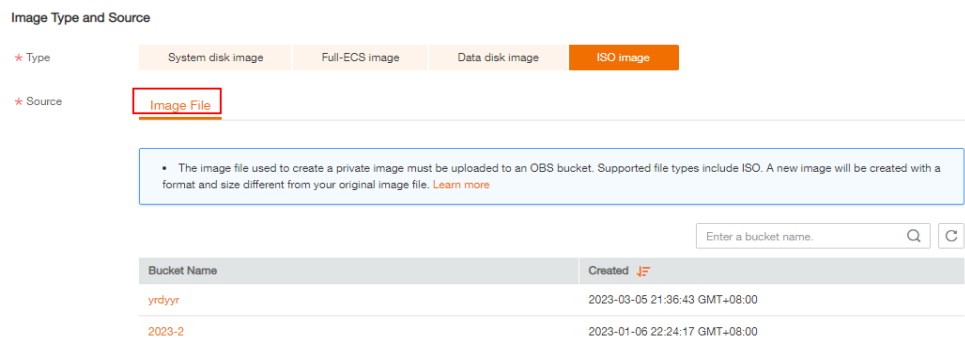
Step 1 Access the IMS console.

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 **ISO image** for **Type**.
3. In the image file list, select the bucket and then the image file.

Figure 3-32 Creating a private image from an ISO file



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

Figure 3-33 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.
- **Tag:** (Optional) Add a tag to the image to be created.
- **Description:** (Optional) Enter image description as needed.

5. Click **Create Now**.

6. Confirm the settings and click **Submit**.

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

3.11.3 Creating a Linux ECS from an ISO Image

Scenarios

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

Constraints

Dedicated Cloud (DeC) users cannot create ECSs from ISO images.

If the DeC service is enabled for a user in a specified region, the user will be a DeC user.

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.

NOTE

If you are a **DeC** user, the **Create ECS** button in the **Operation** column will be unavailable for you because a DeC user cannot use an ISO image to create an ECS.

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.

3.11.4 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.

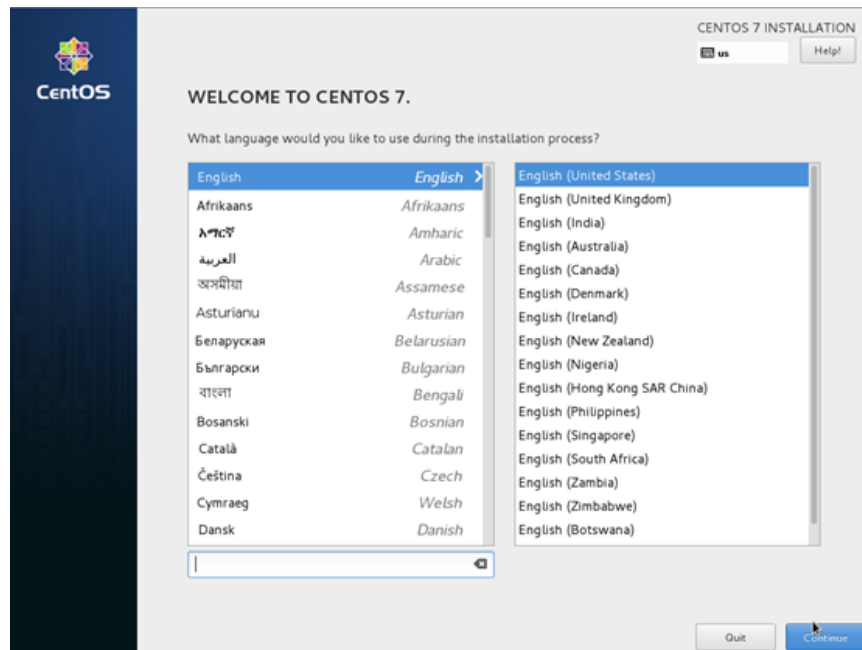
Procedure

CAUTION

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

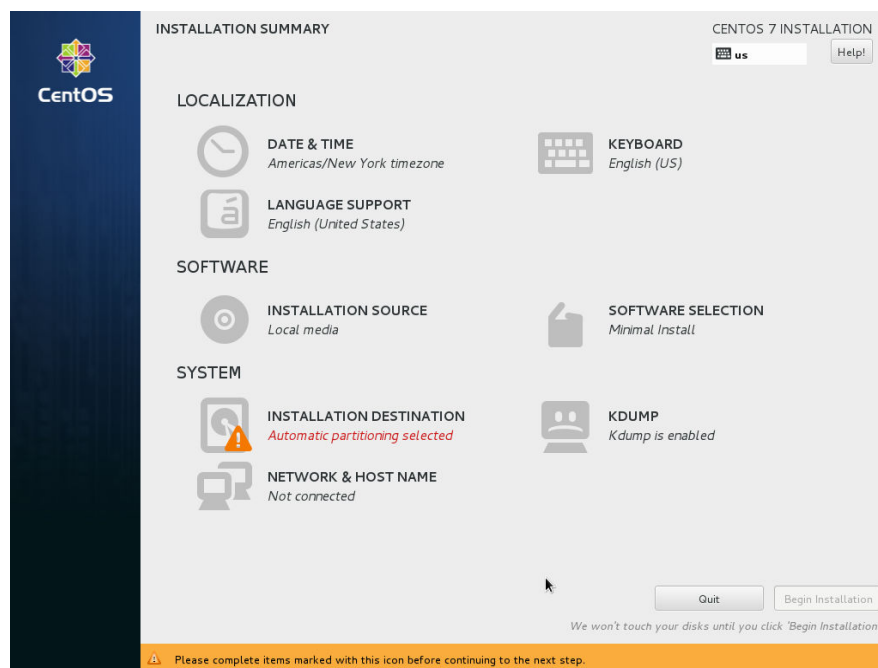
1. On the installation page, select the language and click **Continue**.

Figure 3-34 Installation page



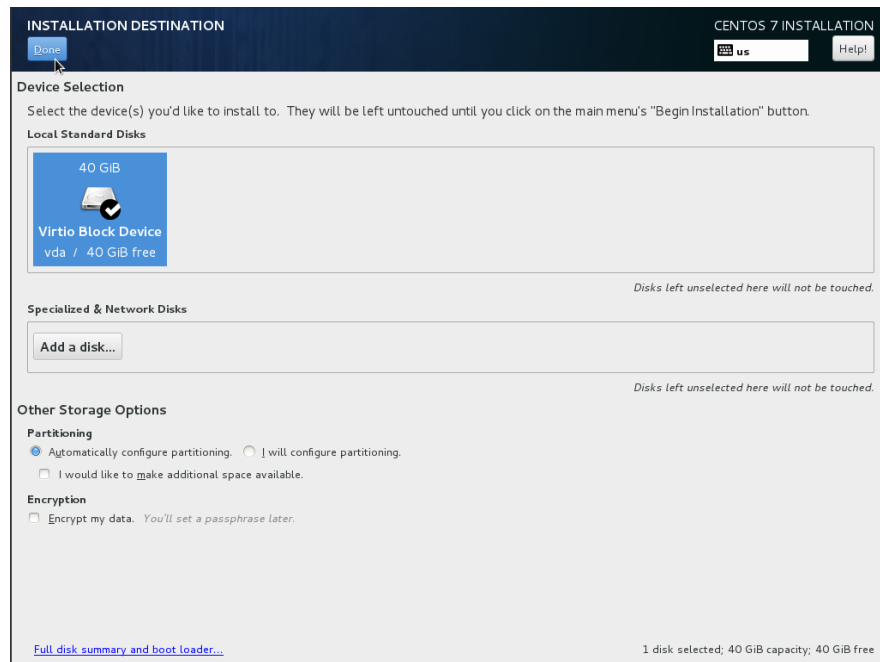
2. On the **INSTALLATION SUMMARY** page, choose **SYSTEM > INSTALLATION DESTINATION**.

Figure 3-35 INSTALLATION SUMMARY page



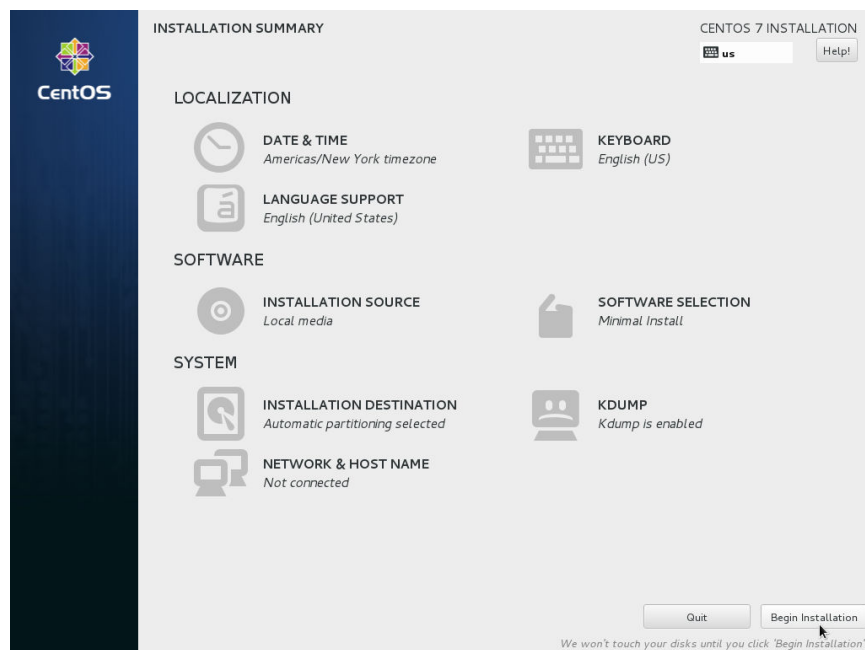
3. Select the target disk and click **Done**.

Figure 3-36 Installation location



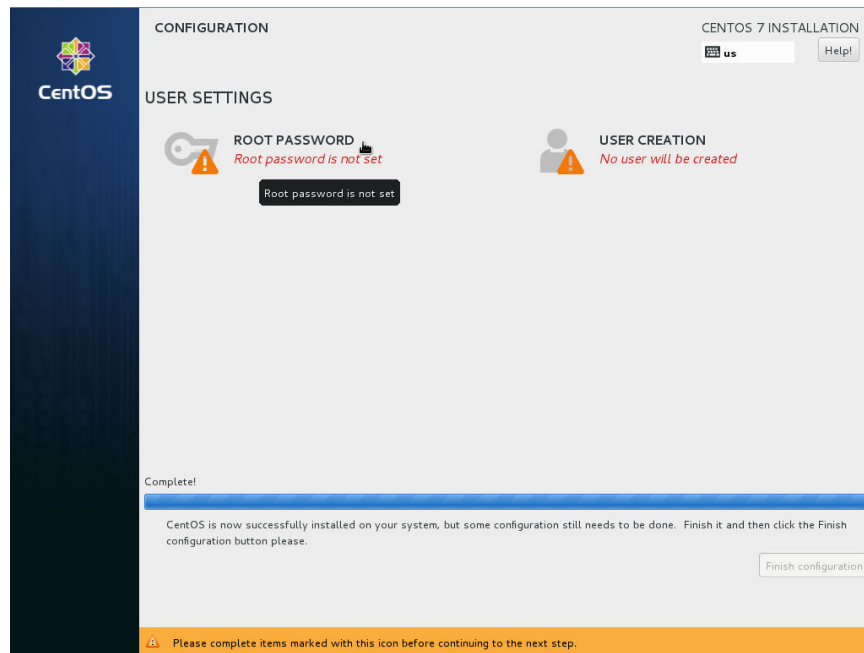
4. Click **Begin Installation**.

Figure 3-37 Starting installation



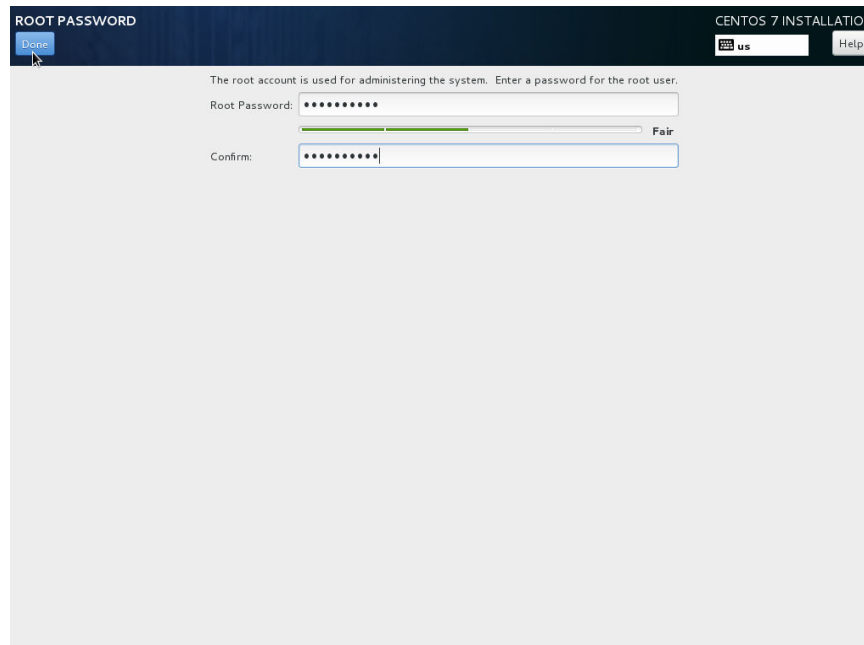
5. Wait for the automatic OS installation to complete. When the progress reaches 100%, CentOS is installed successfully.

Figure 3-38 Successful installation



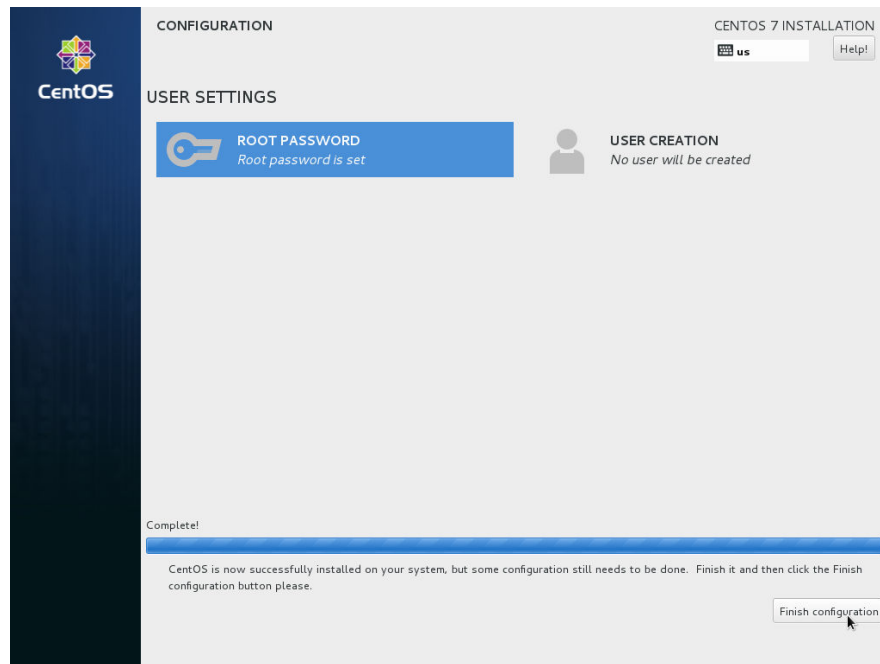
6. In the **USER SETTINGS** area, click **ROOT PASSWORD**.
The **ROOT PASSWORD** page is displayed.
7. Set a password for user **root** as prompted and click **Done**.

Figure 3-39 Setting a password for user root



8. Click **Finish configuration**.

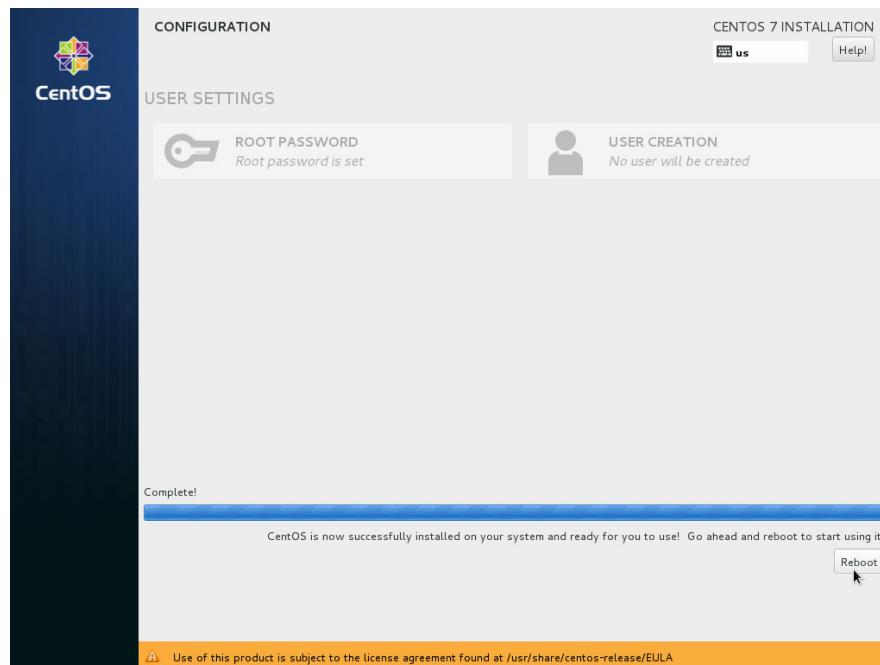
Figure 3-40 Completing configuration



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 3-41 Restarting the ECS



3.11.5 Configuring the ECS and Creating a Linux System Disk Image

Scenarios

After installing an OS for the temporary ECS, configure the ECS and install Xen and 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 DHCP is configured. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in [Configuring 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 native Xen and KVM drivers on the ECS used to create the image. Before installing native Xen and KVM drivers, uninstall PV drivers.

NOTE

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

- Uninstall PV drivers. For details, see [Uninstalling PV Drivers from a Linux ECS](#).
 - Install native Xen and KVM drivers. For details, see [How Do I Install Native Xen and 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 disk identifiers in the GRUB file to UUID. For details, see [Changing Disk Identifiers in the GRUB File to UUID](#).
 - Change disk identifiers in the fstab file to UUID. For details, see [Changing Disk Identifiers in the fstab File to UUID](#).
 - Clear the automatic mount configuration of non-system disks in the **/etc/fstab** file. 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?](#)

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.

3.12 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 alongside each RAW image file. The bitmap file will be uploaded together with the image file. For details, see [Fast Import of 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 3-14 Importing an image

Format	File Size	Reference
VMDK, VHD, QCOW2, VHDX, QED, VDI, QCOW, or ZVHD	Not larger than 128 GB	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• Creating a Data Disk Image from an External Image File
ISO	Not larger than 128 GB	<ul style="list-style-type: none">• Creating a Windows System Disk Image from an ISO File• Creating a Linux System Disk Image from an ISO File

3.13 Fast Import of an Image File

3.13.1 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**

- Fast import is used to quickly import large files. It depends on lazy loading which defers loading of file data until the data is needed. This reduces the initial loading time. However, RAW files do not support lazy loading. 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

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.

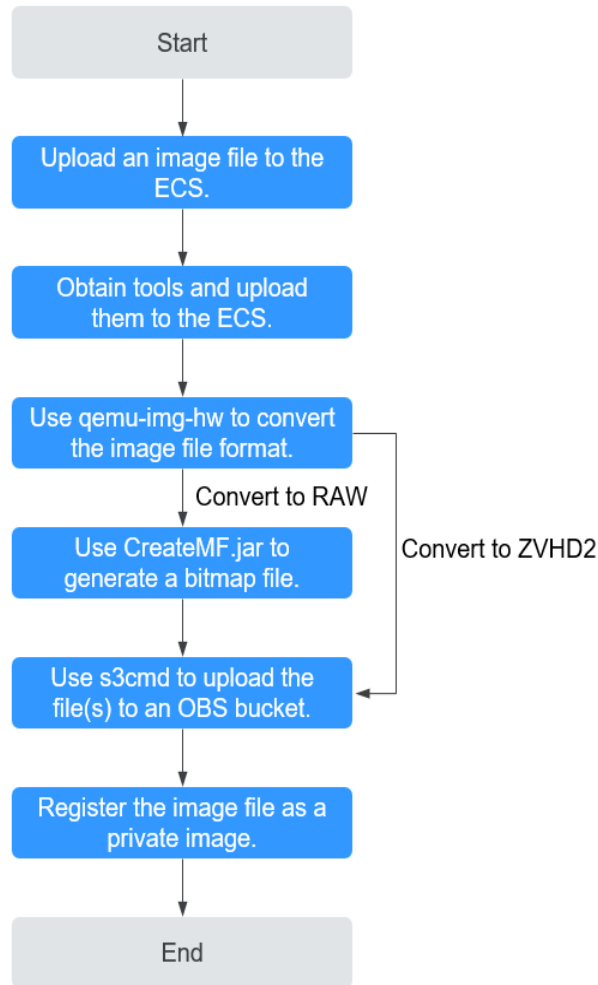
 **NOTE**

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 3-42 Import process (Linux)



For details, see [Fast Import in Linux](#).

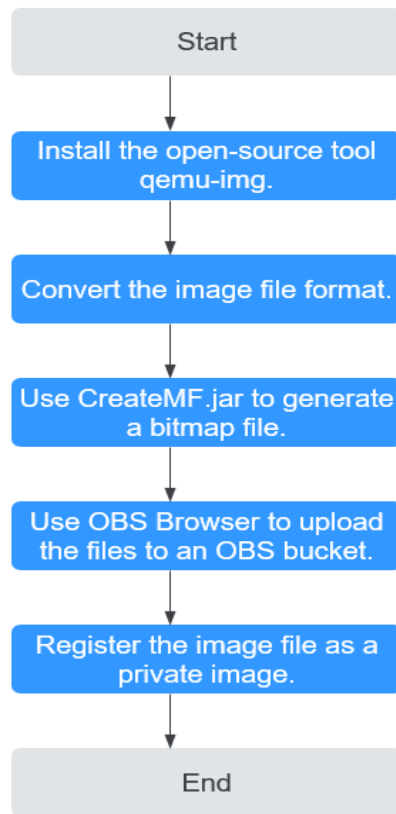
- Windows

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 3-43 Import process (Windows)



For details, see [Fast Import in Windows](#).

3.13.2 Fast Import in Linux

Scenarios

This section describes how to convert the format of a large 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 3-6 \(Windows\)](#) or [Table 3-10 \(Linux\)](#).

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 fast import tool package, upload it to the ECS, and then decompress the package.

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/quick-import-tools/qemu-img-hw**.

```
cd /usr/quick-import-tools/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/quick-import-tools/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 **System disk image** or **Data disk image** for **Type**.
4. Select **Image File** for **Source**. Select the bucket storing the ZVHD2 or RAW image file and then select the image 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.

The API is POST `/v2/cloudimages/quickimport/action`.

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.

3.13.3 Fast Import in 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 3-6 \(Windows\)](#) or [Table 3-10 \(Linux\)](#).

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

Table 3-15 CreateMF.jar package

Tool Package	How to Obtain
createMF.zip	https://cn-south-1-cloud-reset-pwd.obs.cn-south-1.myhuaweicloud.com/imageImportTools/createMF.zip

- 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.
- 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 **System disk image** or **Data disk image** for **Type**.
4. Select **Image File** for **Source**. Select the bucket storing the ZVHD2 or RAW image file and then select the image 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.

The API is POST /v2/cloudimages/quickimport/action.

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

----**End**

4 Managing Private Images

4.1 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 Server** 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.

4.2 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 interrupts 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 4-1 Modifying an image

Modify Image

* Name

Description

Minimum Memory

If the minimum memory size of an image has been increased, it must be set back to the original size before you reinstall OSs of the ECSs that were created using the image.

Unlimited 1 GB 2 GB 4 GB 8 GB

16 GB 32 GB 64 GB 128 GB

Maximum Memory

Unlimited 4 GB 32 GB 64 GB

128 GB

NIC Multi-Queue Supported Not supported ?

Boot Mode BIOS UEFI

The boot mode must be the same as that of the OS contained in the image file. Otherwise, ECSs created from this system disk image will fail to start.

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.

4.3 Exporting an Image

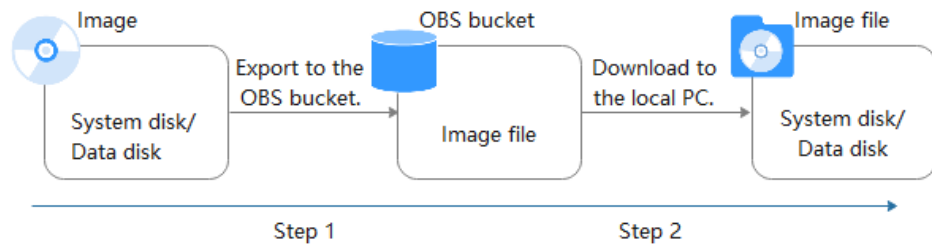
Scenarios

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-premises clusters or private clouds by exporting their images from the cloud platform. The following figure shows the process of exporting an image.

Figure 4-2 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 [Converting the Image Format Using qemu-img-hw](#).

 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+.

4.4 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, 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. On the **Private Images** tab, click **Export** above the image list and select what images to export.

The system will automatically export the list of selected private images in the current region under your account to a local directory.

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. On the **Public Images** tab, click **Export** above the image list, and select **Export all data to an XLSX file**.

The system will automatically export the list of all public images in the current region to a local directory.

4.5 Checking the Disk Capacity of an Image

Scenarios

You can check the disk capacity of a private image.

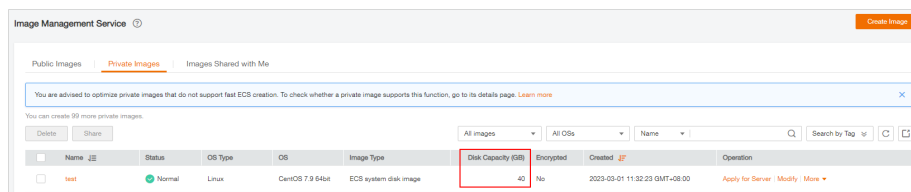
- To check the disk capacity of a system disk image, data disk image, or ISO image, see [Check the Disk Capacity of a System Disk Image, Data Disk Image, or ISO Image](#).
- To check the disk capacity of a full-ECS image, see [Check the Disk Capacity of a Full-ECS Image](#).

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 4-3 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 --.

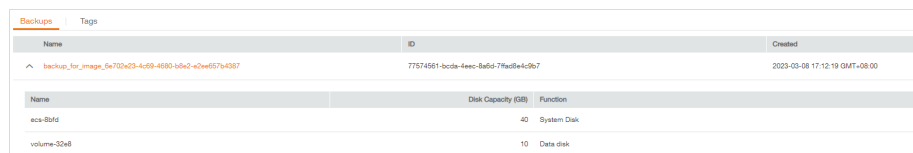
3. Click the full-ECS image name.
4. Click the **Backups** tab and view the capacities of the system disk and data disks in the backup.

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.
- 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 4-4 Checking backup details



Name	ID	Created
backup_for_image_5e702e23-4c59-4690-bba2-e2ee657e4387	77574561-bcda-4eec-8a56-7f8d8e4c9b7	2023-03-08 17:12:19 GMT+08:00

Name	Disk Capacity (GB)	Function
ecs-00d	40	System Disk
volume-52a8	10	Data disk

4.6 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.
- If a full-ECS image is still being created when you delete it, some intermediate backups may fail to be deleted. To avoid generating any unnecessary expenditures, you can delete them on the CBR console.

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 CSBS 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 CSBS backups of the full-ECS images.

NOTE

If CSBS 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**.

4.7 Sharing Images

4.7.1 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.
- Each image can be shared with a maximum of 128 tenants.
- Encrypted images cannot be shared.
- 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 project ID 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.

FAQ

If you have any questions, see [General Sharing FAQ](#).

4.7.2 Obtaining the Project ID

Scenarios

Before a tenant shares an image with you, you need to provide your project ID.

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 project ID.

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

4.7.3 Sharing Specified Images

Scenarios

After obtaining the project ID from a tenant, 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 project ID from the target tenant.
- Before sharing an image, ensure that any sensitive data has been deleted from the image.

Procedure

- Share multiple images.

- a. Access the IMS console.
 - i. Log in to the management console.
 - ii. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
- b. Click the **Private Images** tab.
- c. Select the private images to share and click **Share** above the image list.
- d. In the **Share Image** dialog box, enter the project ID of the target tenant.
To share images with more than one tenant, separate their project IDs with commas (,).

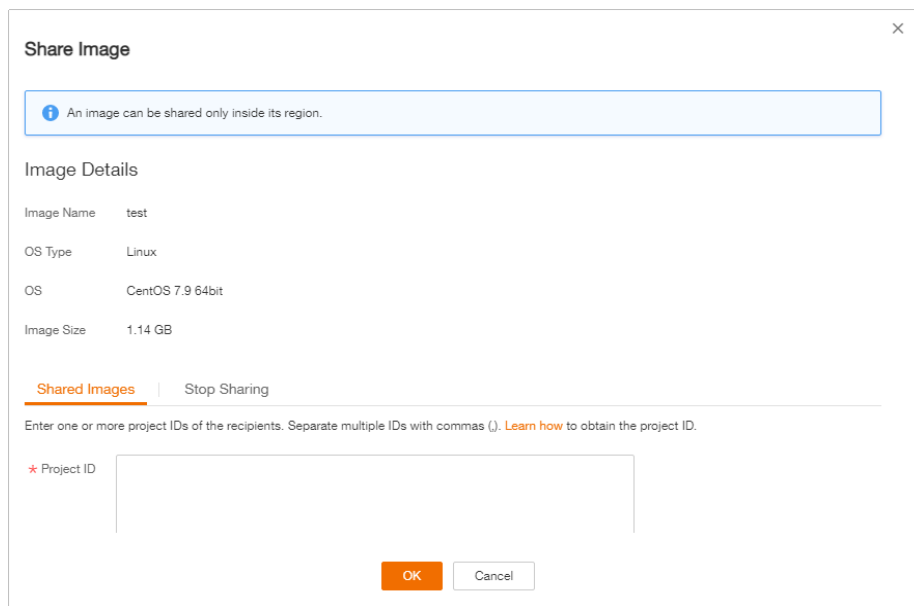
 **NOTE**

- You can enter a maximum of 100 project IDs at a time.
 - You can share images only within the region where they reside.
 - If the target tenant is a multi-project user, you can share images to any project of the tenant.
- e. Click **OK**.
- Share a single image.
 - a. Access the IMS console.
 - i. Log in to the management console.
 - ii. Under **Computing**, click **Image Management Service**.
The IMS console is displayed.
 - b. Click the **Private Images** tab.
 - c. 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.
 - d. In the **Share Image** dialog box, enter the project ID of the target tenant.
To share an image with more than one tenant, separate their project IDs with commas (,).

 **NOTE**

- You can enter a maximum of 100 project IDs at a time.
- You can share images only within the region where they reside.
- If the target tenant is a multi-project user, you can share images to any project of the tenant.

Figure 4-5 Sharing an image



- e. 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](#).

4.7.4 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. In the upper left corner, switch to the region where the target project is and then select the project.
 3. 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.

4. (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).

4.7.5 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 click **Reject** in the **Operation** column. In the displayed dialog box, click **Yes**.

4.7.6 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.

4.7.7 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 project for which you want to stop the image sharing and click **OK**.

4.7.8 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 project IDs of the tenants to be added.

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 project ID of the tenant to be added, and click **OK**.

To add multiple tenants, enter their project IDs and separate them with commas (,). Click **OK**.

 **NOTE**

- You can share images only within the region where they reside.
- A project ID uniquely identifies a tenant in a specific region. If you enter a project ID that belongs to a different region from the images, a message will display indicating that the tenant cannot be found.

4.7.9 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 the project IDs 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**.

4.8 Replicating Images Within a Region

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 image replication. 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 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 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.
 - **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.

4.9 Optimizing a Windows Private Image

4.9.1 Optimization Process

An ECS can run properly only after Xen Guest OS drivers (PV drivers) and KVM Guest OS drivers (VirtIO drivers) are installed on it. To ensure that ECSs support both Xen and KVM and to improve network performance, PV and 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 the latest version of PV drivers on the ECS.
For details, see [Installing PV Drivers](#).
3. Install VirtIO drivers that are needed to create KVM ECSs.
For details, see [Installing VirtIO Drivers](#).
4. 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.
5. Clear system logs and then stop the ECS.
For details, see [Clearing System Logs](#).
6. Create a Windows private image from the ECS.

4.9.2 Viewing the Virtualization Type of a Windows ECS

Open the cmd window and run the following command to query the virtualization type of the ECS:

systeminfo

If the values of **System Manufacturer** and **BIOS Version** are **Xen**, the ECS uses Xen. To make the Windows private image support KVM at the same time, perform operations in the following sections on the ECS.

 **NOTE**

If the ECS uses KVM, you are also advised to optimize the private image to prevent any exceptions with the ECSs created from the image.

Figure 4-6 Viewing the virtualization type of a Windows ECS

```

\systeminfo
Host Name: ECS-E5AF
OS Name: Microsoft Windows Server 2012 R2 Datacenter
OS Version: 6.3.9600 N/A Build 9600
OS Manufacturer: Microsoft Corporation
OS Configuration: Standalone Server
OS Build Type: Multiprocessor Free
Registered Owner: Windows User
Registered Organization:
Product ID: 00253-50000-00000-AA442
Original Install Date: 11/2/2015, 21:05:21
System Boot Time: 8/2/2018, 10:31:04
System Manufacturer: Xen
System Model: HVM domU
System Type: x64-based PC
Processor(s): 1 Processor(s) Installed.
[01]: Intel64 Family 6 Model 62 Stepping 4 GenuineInt
... 1 ~1000 Mhz
BIOS Version: Xen 4.1.2_115-908.762., 3/21/2018
Windows Directory: C:\Windows
System Directory: C:\Windows\system32
Boot Device: \Device\HarddiskVolume1
System Locale: en-us;English (United States)
Input Locale: en-us;English (United States)
Time Zone: (UTC+08:00) Beijing, Chongqing, Hong Kong, Urunqi
Total Physical Memory: 1,016 MB
Available Physical Memory: 226 MB
Virtual Memory: Max Size: 1,336 MB
Virtual Memory: Available: 476 MB
Virtual Memory: In Use: 860 MB
    
```

4.9.3 Obtaining Required Software Packages

PV Drivers

Table 4-1 lists the PV driver packages required for optimizing Windows private images.

Table 4-1 PV driver packages

Software Package	OS	How to Obtain
pvdriver-win2008R2-64bit.zip	Windows Server 2008 R2 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2008R2-64bit.zip
pvdriver-win2012-64bit.zip	Windows Server 2012 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2012-64bit.zip
pvdriver-win2012R2-64bit.zip	Windows Server 2012 R2 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2012R2-64bit.zip
pvdriver-win2016-64bit.zip	Windows Server 2016 64bit	https://ecs-instance-driver.obs.cn-north-1.myhuaweicloud.com/pvdriver-win2016-64bit.zip

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.

4.9.4 Installing PV Drivers

This section only applies to Xen ECSs, which have been discontinued and no longer been available for new users. If you are a new user or you are an existing user that will use this image to create a non-Xen ECS, skip this section.

Scenarios

Before using an ECS or external image file to create a private image, ensure that PV drivers have been installed in the OS so that ECSs created from this image can support Xen virtualization, the I/O performance can be improved, and advanced functions such as hardware monitoring can be available.

NOTICE

If you do not install PV drivers, the ECS network performance will be poor, and the security groups and firewall configured for the ECS will not take effect.

PV drivers have been installed by default when you use a public image to create ECSs. You can perform the following operations to verify the installation:

Open the **version** configuration file to check whether the PV drivers are the latest:

C:\Program Files (x86)\Xen PV Drivers\bin\version

Prerequisites

- An OS has been installed for the ECS, and an EIP has been bound to the ECS.
- The remaining capacity of the ECS system disk must be greater than 32 MB.
- If the ECS uses Windows 2008, you must install PV drivers as an administrator.
- The PV driver package has been downloaded on the ECS. For how to obtain the software package, see [Obtaining Required Software Packages](#).
- To avoid an installation failure, perform the following operations before starting the installation:
 - Uninstall third-party virtualization platform tools, such as Citrix Xen Tools and VMware Tools. For how to uninstall the tools, see the corresponding official documents of the tools.
 - Disable your anti-virus and intrusion detection software. You can enable them after PV drivers are installed.

Installing PV 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. On the ECS, choose **Start > Control Panel**.
3. Click **Uninstall a program**.
4. Uninstall **GPL PV drivers for Windows x.x.x.xx** as prompted.
5. Download PV drivers of the required version based on the ECS OS and [Obtaining Required Software Packages](#).
6. Decompress the PV driver package.
7. Right-click **GPL PV Drivers for Windows x.x.x.xx**, select **Run as administrator**, and complete the installation as prompted.
8. Restart the ECS as prompted to make the PV drivers take effect.
ECSs running Windows Server 2008 must be restarted twice.

NOTE

After the PV drivers are installed, the ECS NIC configuration will be lost. If you have configured NICs before, you need to configure them again.

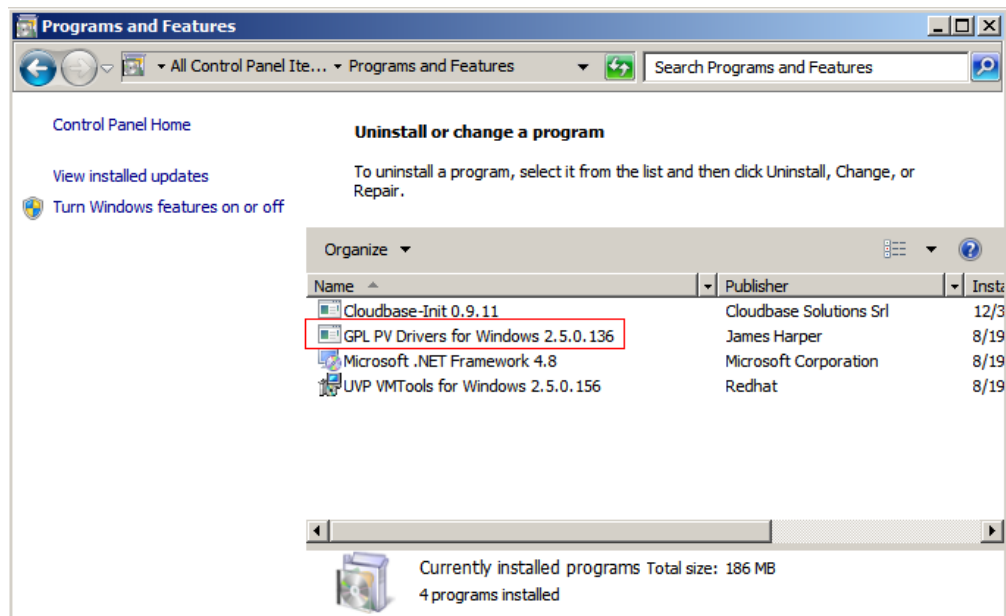
Verifying the Installation

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

1. Click **Start**. Choose **Control Panel > Programs and Features**.
2. Locate PV drivers for Windows.

If PV drivers for Windows exist, the installation is successful, as shown in [Figure 4-7](#).

Figure 4-7 Verifying the installation



4.9.5 Installing VirtIO Drivers

Scenarios

VirtIO is a standard interface for VMs to access host devices. It is used to improve the I/O performance between VMs and hosts. For details about VirtIO, see [VirtIO](#).

For details about open source code of virtio-win/kvm-guest-drivers-windows, see <https://github.com/virtio-win/kvm-guest-drivers-windows>.

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.

This section describes how to install VirtIO drivers on a KVM ECS.

NOTICE

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.

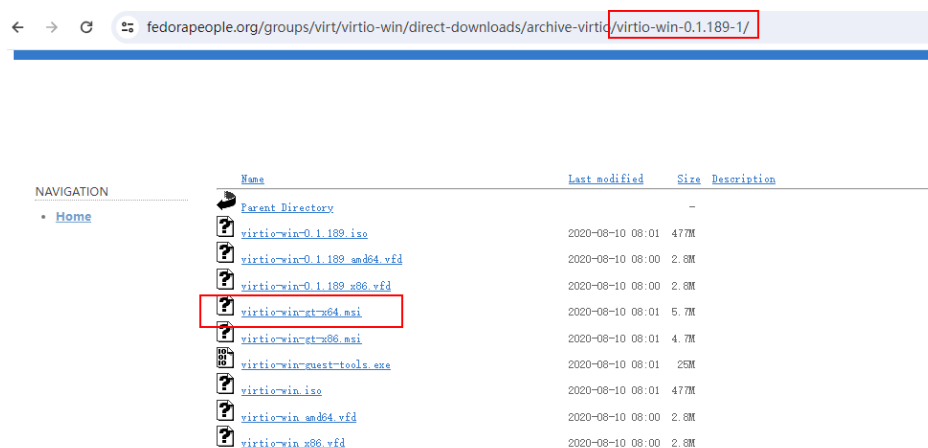
For details about how to log in to an ECS, see *Elastic Cloud Server User Guide*.

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 4-8 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 4-9 Starting the installation

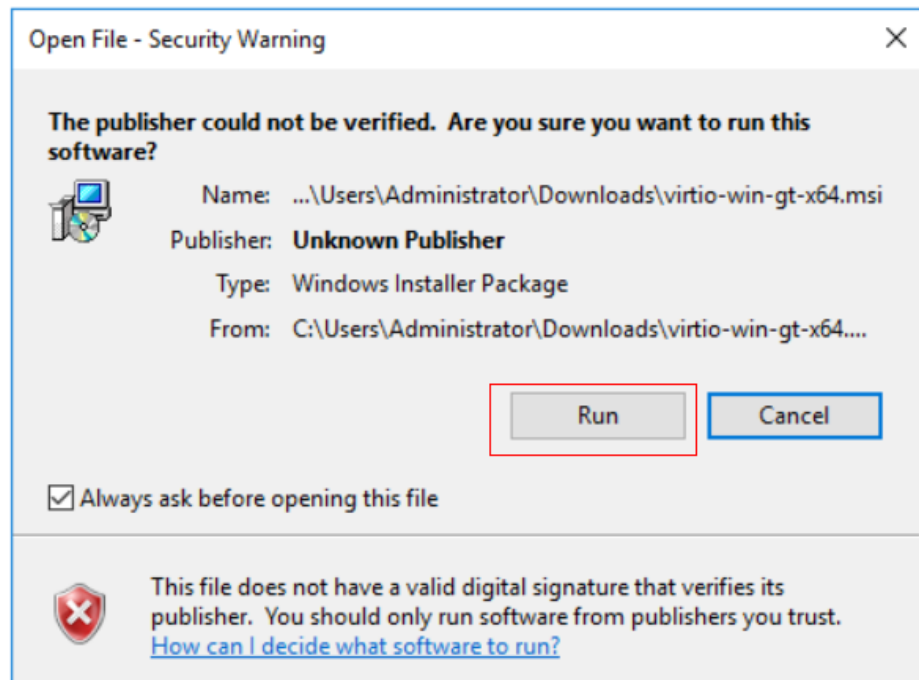


Figure 4-10 Installation wizard

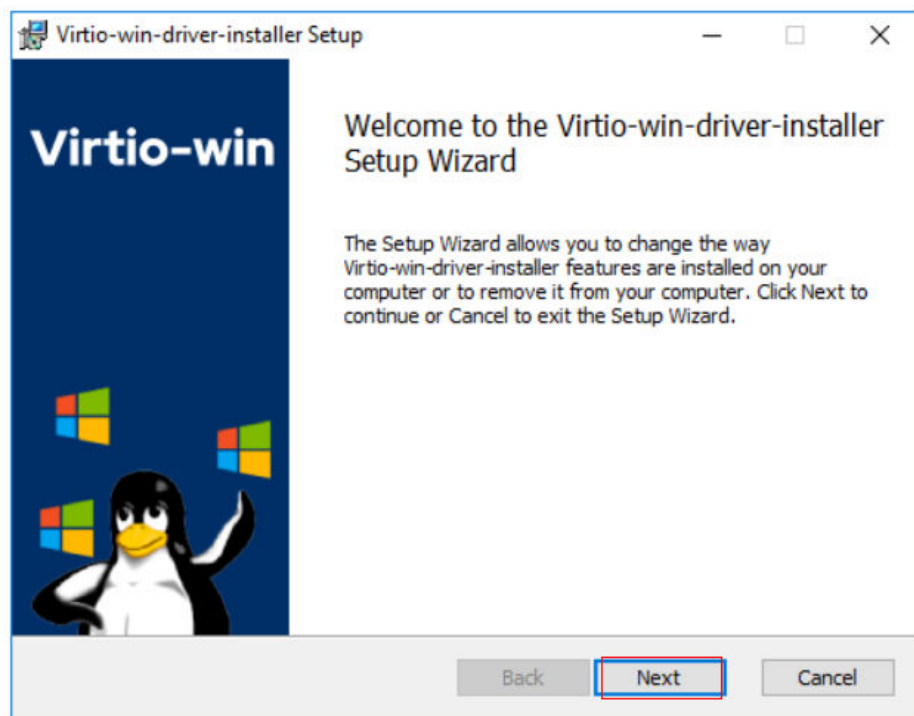
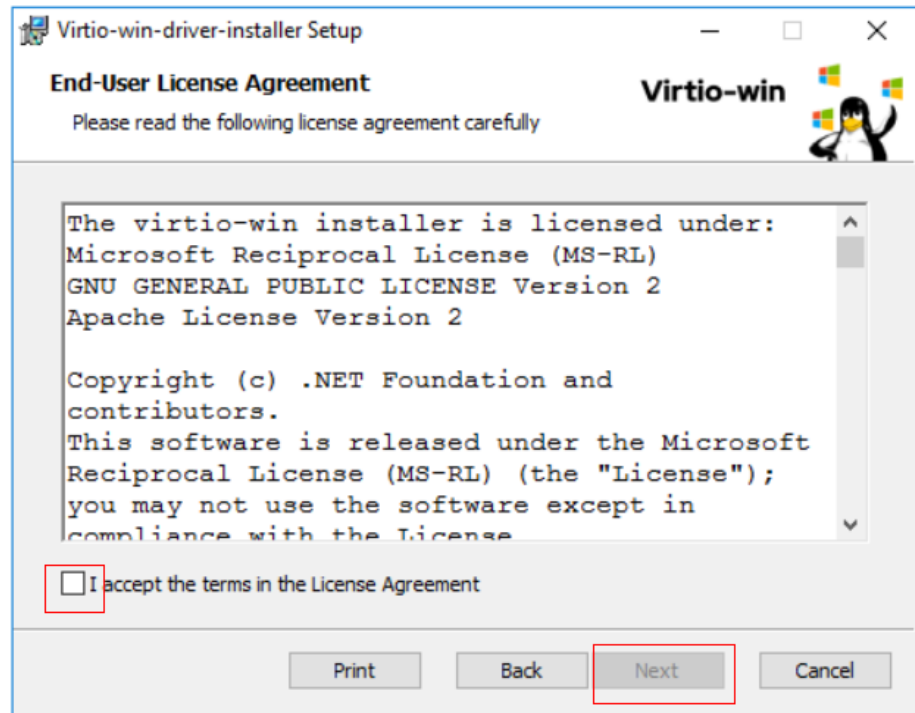


Figure 4-11 Accepting the agreement



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

Figure 4-12 Selecting VirtIO drivers to install

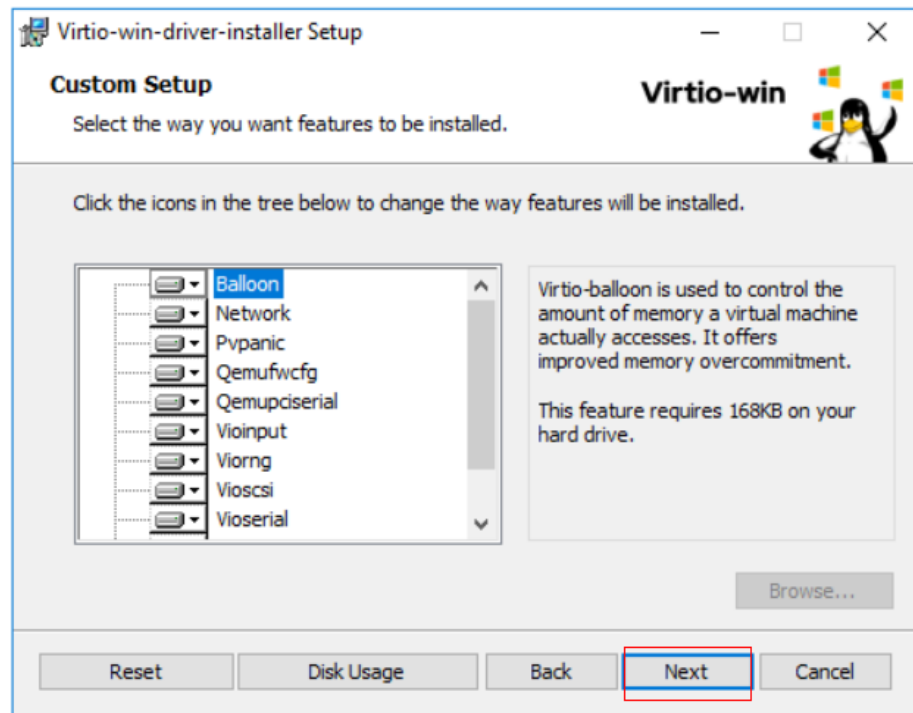
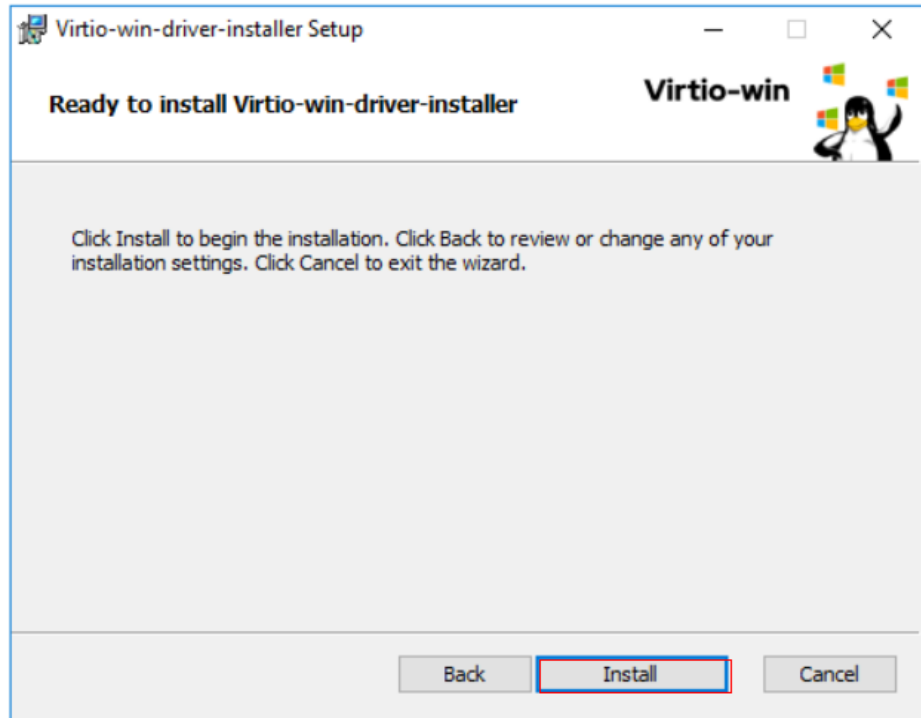
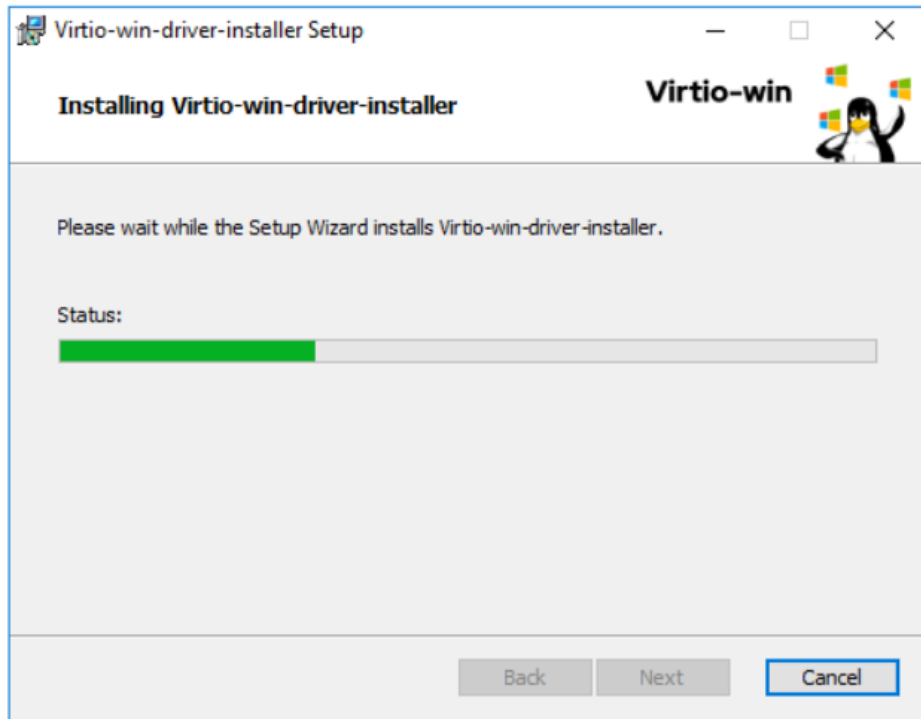


Figure 4-13 Proceeding with the installation.



4. Wait until the installation is complete.

Figure 4-14 Installation in process



5. Restart the ECS after the installation is complete.

Figure 4-15 Installation completed

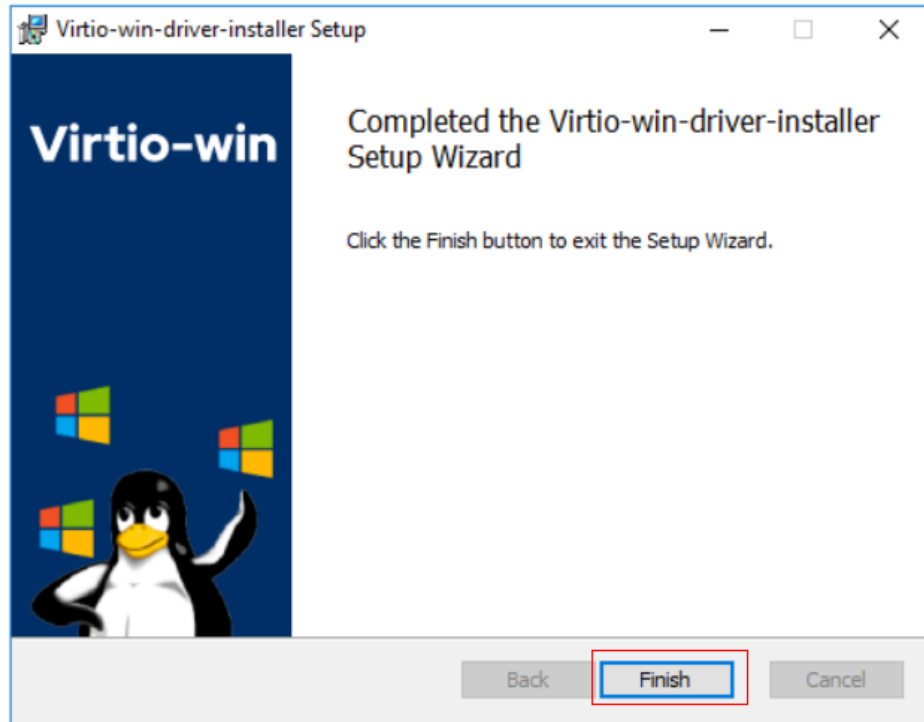
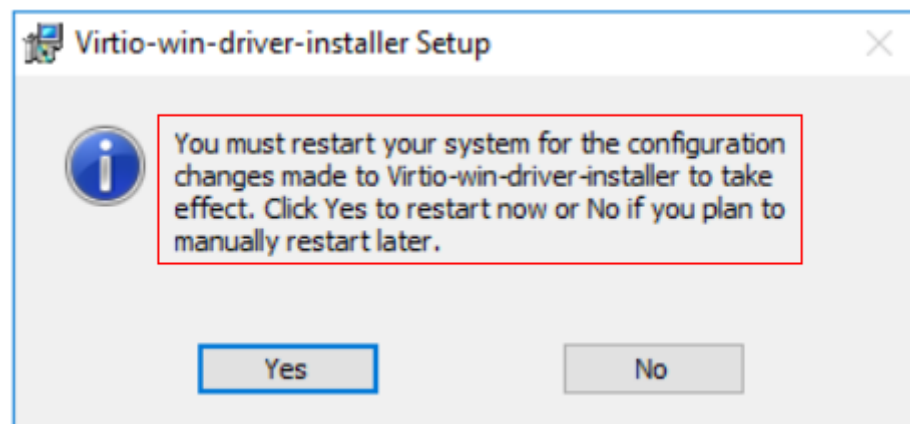


Figure 4-16 Restart prompt



6. After the restart, perform the operations in [Verifying the Installation](#) to verify that the VirtIO drivers have been successfully installed.

Verifying the Installation

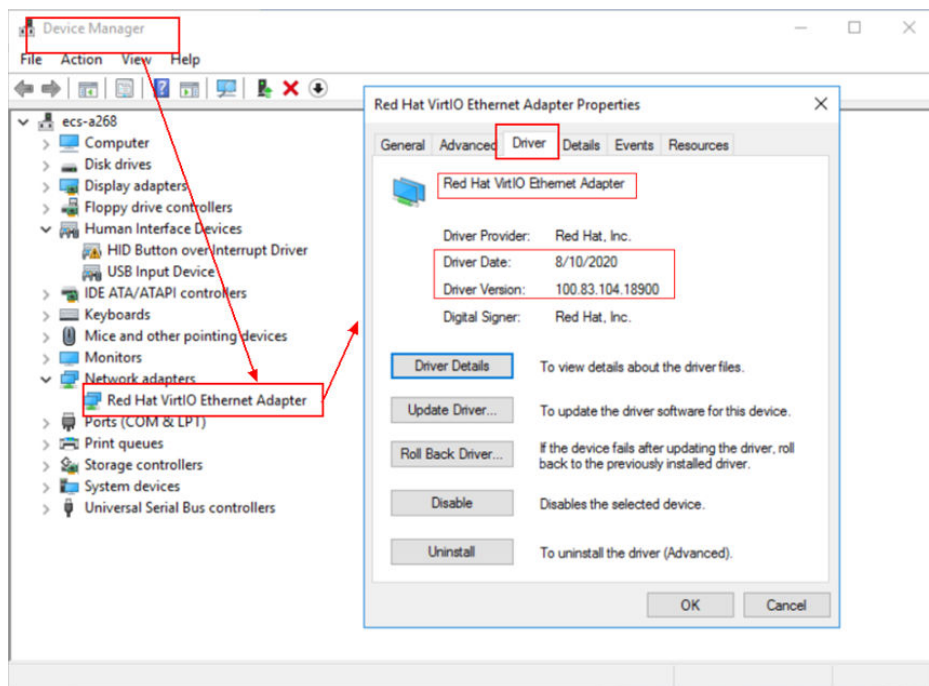
Perform the following steps to verify the installation of the VirtIO drivers:

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 4-17 Version and date of downloaded drivers

Name	Last modified	Size	Description
Parent Directory		-	
virtio-win-0.1.189.iso	2020-08-10 08:01	4.77M	
virtio-win-0.1.189_amd64.vfd	2020-08-10 08:00	2.8M	
virtio-win-0.1.189_x86.vfd	2020-08-10 08:00	2.8M	
virtio-win-gt-x64.msi	2020-08-10 08:01	5.7M	
virtio-win-gt-x86.msi	2020-08-10 08:01	4.7M	
virtio-win-guest-tools.exe	2020-08-10 08:01	25M	
virtio-win.iso	2020-08-10 08:01	4.77M	
virtio-win_amd64.vfd	2020-08-10 08:00	2.8M	
virtio-win_x86.vfd	2020-08-10 08:00	2.8M	

Figure 4-18 Version and date of drivers in Device Manager



4.9.6 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.

4.10 Optimizing a Linux Private Image

4.10.1 Optimization Process


The virtualization of ECSs is gradually changing from Xen to KVM. Therefore, private images need to support both Xen and KVM. To ensure that ECSs created from a private image can run properly, you are advised to optimize it no matter it is using Xen or KVM.

A Linux ECS can run properly only when native Xen (Xen PV) drivers and KVM (VirtIO) drivers have been installed on it and disk identifiers in its GRUB file and fstab file have 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](#).
The virtualization type may cause slice differences in an optimization process.

Process

1. Uninstall PV drivers from the ECS.
For details, see [Uninstalling PV Drivers from a Linux ECS](#).
-  **NOTE**
- If the ECS is using KVM virtualization, skip this step.
2. Change disk identifiers in the GRUB file to UUID.
For details, see [Changing Disk Identifiers in the GRUB File to UUID](#).
 3. Change disk identifiers in the fstab file to UUID.
For details, see [Changing Disk Identifiers in the fstab File to UUID](#).
 4. Install native virtualization drivers.
 - For Xen, install native Xen and KVM drivers. For details, see [Installing Native Xen and KVM Drivers](#).
 - For KVM, install native KVM drivers. For details, see [Installing Native KVM Drivers](#).
 5. Delete log files and historical records, and stop the ECS.
For details, see [Clearing System Logs](#).
 6. Create a Linux private image from the ECS.

4.10.2 Checking Whether a Private Image Needs to be Optimized

- If the virtualization type is Xen, optimization is required.

- 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. Run the following command to check the virtualization type of an ECS:

lscpu

- If the value of **Hypervisor vendor** is **Xen**, optimize the private image as instructed in [Process](#).
- If the value of **Hypervisor vendor** is **KVM**, go to the next step for further check.

Figure 4-19 Checking the virtualization type of a Linux ECS

```
# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
CPU(s):                 4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  62
Model name:             Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz
Stepping:               4
CPU MHz:                3000.079
BogoMIPS:               6000.15
Hypervisor vendor:     Xen
Virtualization type:   full
L1d cache:              32K
L1i cache:              32K
L2 cache:               256K
L3 cache:               25600K
NUMA node0 CPU(s):     0-3
You have new mail in /var/spool/mail/root
root@SZV-home1#
```

2. 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](#).

```
root@ip-10-0-348-71:~# lsinitrd /boot/initramfs-$(uname -r).img | grep virtio
-rw-r--r-- 1 root root 8984 Sep 30 2021 usr/lib/modules/4.18.0-348.7.1.el8_5.x86_64/kernel/drivers/block/virtio_b
lk.ko.xz
-rw-r--r-- 1 root root 14836 Sep 30 2021 usr/lib/modules/4.18.0-348.7.1.el8_5.x86_64/kernel/drivers/char/virtio_co
nsole.ko.xz
-rw-r--r-- 1 root root 25748 Sep 30 2021 usr/lib/modules/4.18.0-348.7.1.el8_5.x86_64/kernel/drivers/net/virtio_net
.ko.xz
-rw-r--r-- 1 root root 8684 Sep 30 2021 usr/lib/modules/4.18.0-348.7.1.el8_5.x86_64/kernel/drivers/scsi/virtio_sc
si.ko.xz
```

Otherwise, VirtIO drivers have not been installed. Optimize the private image as instructed in [Process](#).

4.10.3 Uninstalling PV Drivers from a Linux ECS

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.

To ensure that you can successfully install native Xen and KVM drivers, you must uninstall PV drivers from the ECS first.

Procedure

1. Log in to the ECS as user **root** using VNC.
2. Run the following command to check whether PV drivers are installed in the OS:

```
ps -ef | grep uvp-monitor
```

- If the following information is displayed, PV drivers have been installed.
- Otherwise, PV drivers are not installed. No further actions will be required.

```
root 4561 1 0 Jun29 ? 00:00:00 /usr/bin/uvp-monitor
root 4567 4561 0 Jun29 ? 00:00:00 /usr/bin/uvp-monitor
root 6185 6085 0 03:04 pts/2 00:00:00 grep uvp-monitor
```

3. In the VNC login window, open the CLI.
For how to open the CLI, see the OS manual.
4. Run the following command to uninstall PV drivers:

```
/etc/.uvp-monitor/uninstall
```

- PV drivers are uninstalled successfully if the following command output is displayed:

```
The PV driver is uninstalled successfully. Reboot the system for the uninstallation to take effect.
```

- If the command output indicates that **.uvp-monitor** is not found, go to [5](#).

```
-bash: /etc/.uvp-monitor/uninstall: No such file or directory
```

5. Perform the following operations to delete uvp-monitor that failed to take effect, preventing log overflow:
 - a. Run the following command to check whether UVP user-mode programs are installed in the OS:

```
rpm -qa | grep uvp
```


Information similar to the following is displayed:

```
libxenstore_uvp3_0-3.00-36.1.x86_64  
uvp-monitor-2.2.0.315-3.1.x86_64  
kmod-uvpmod-2.2.0.315-3.1.x86_64
```

- b. Run the following commands to delete the installation packages:

```
rpm -e kmod-uvpmod  
rpm -e uvp-monitor  
rpm -e libxenstore_uvp
```

4.10.4 Changing Disk Identifiers in the GRUB File to UUID

Scenarios

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

Modify the **menu.lst** or **grub.cfg** 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 a 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 change 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 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 change 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-9516-5a687fecdc6" 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=latacyrheb-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-9516-5a687fecdc6**.
- If **root=UUID=f382872b-eda6-43df-9516-5a687fecdc6** is contained, the root partition is in 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-9516-5a687fecdc6**.
- 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-9516-5a687fecdc6 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD  
SYSFONT=latacyrheb-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 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
}
```

4.10.5 Changing Disk Identifiers 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 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 a 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
```

4.10.6 Installing Native Xen and 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 Xen and KVM drivers.

CAUTION

If an ECS has no Xen drivers installed, the network performance of the ECS will be poor, and the security groups and firewall configured for the ECS will not take effect.

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 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 [CentOS and EulerOS](#).

- Ubuntu and Debian

Modify the `/etc/initramfs-tools/modules` file. Add the Xen PV and VirtIO 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/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@CTU10000xxxx ~]# 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 **dracut -f *Name of the initramfs or initrd file actually used***. You can obtain the actual initramfs or initrd file name 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is initramfs, run the following commands:

```
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:

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

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

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

```
[root@CTU10000xxxx 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@CTU10000xxxx 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@CTU10000xxxx ~]#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@ CTU10000xxxx 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@ CTU10000xxxx 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 install the drivers. For details, see [scenario 1](#).

If the OS version is SUSE 12 SP1, modify the `/etc/dracut.conf` file to install the 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 install the drivers. For details, see [scenario 3](#).

- Earlier than SUSE 12 SP1 or openSUSE 13:

 NOTE

Before installing the drivers, 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:

```
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 package 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 `dracut -f` command to regenerate `initrd`.

 NOTE

If the virtual file system is not the default `initramfs` or `initrd`, run `dracut -f Name of the initramfs or initrd file actually used`. 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 processor.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 is the **initrd** file actually used. If **/boot** is missing in the **initrd** file path, you need to add it when you run the **dracut -f** command. In this case, the command should be **dracut -f /boot/initramfs-XXX**.

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

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

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

```

SIA10000xxxx:~ # 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

```

```

SIA10000xxxx:~ # 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 to add **xen_platform_pci.dev_unplug=all** and change the root settings.

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 UUID format.
- **xen_platform_pci.dev_unplug=all** is used 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
```

```
SIA10000xxxx:~ # 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

SIA10000xxxx:~ # 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 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
```

- SUSE 12 SP1:

- 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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:

```
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:

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

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

- Later than SUSE 12 SP1 or openSUSE 13:

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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:

```
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:

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

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

Assume that the virtual file system is `initrd`. The command output will be:

```
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 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
```

4.10.7 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 4-2 Modifying configuration files for different OSs

OS	Configuration	Reference
CentOS/EulerOS	<p>Take CentOS 7.0 as an example.</p> <ol style="list-style-type: none"> In the <code>/etc/dracut.conf</code> file, add VirtIO drivers to add_drivers, including <code>virtio_blk</code>, <code>virtio_scsi</code>, <code>virtio_net</code>, <code>virtio_pci</code>, <code>virtio_ring</code>, and <code>virtio</code>. Separate driver names with spaces. Save and exit the <code>/etc/dracut.conf</code> file and run the dracut -f command to generate initrd again. 	CentOS and EulerOS
Ubuntu/Debian	<ol style="list-style-type: none"> In the <code>/etc/initramfs-tools/modules</code> file, add VirtIO drivers, including <code>virtio_blk</code>, <code>virtio_scsi</code>, <code>virtio_net</code>, <code>virtio_pci</code>, <code>virtio_ring</code>, and <code>virtio</code>. Separate driver names with spaces. Save and exit the <code>/etc/initramfs-tools/modules</code> file and run the update-initramfs -u command to generate initrd again. 	Ubuntu and Debian
SUSE and openSUSE	<p>If the OS version is earlier than SUSE 12 SP1 or openSUSE 13:</p> <ol style="list-style-type: none"> In the <code>/etc/sysconfig/kernel</code> file, add VirtIO drivers to INITRD_MODULES="". VirtIO drivers include <code>virtio_blk</code>, <code>virtio_scsi</code>, <code>virtio_net</code>, <code>virtio_pci</code>, <code>virtio_ring</code>, and <code>virtio</code>. Separate driver names with spaces. Run the mkinitrd command to generate initrd again. 	SUSE and openSUSE (Earlier than SUSE 12 SP1 or openSUSE 13)
	<p>If the OS version is SUSE 12 SP1:</p> <ol style="list-style-type: none"> In the <code>/etc/dracut.conf</code> file, add VirtIO drivers to add_drivers. VirtIO drivers include <code>virtio_blk</code>, <code>virtio_scsi</code>, <code>virtio_net</code>, <code>virtio_pci</code>, <code>virtio_ring</code>, and <code>virtio</code>. Separate driver names with spaces. Run the dracut -f command to generate initrd again. 	SUSE and openSUSE (SUSE 12 SP1)

OS	Configuration	Reference
	<p>If the OS version is later than SUSE 12 SP1 or openSUSE 13:</p> <ol style="list-style-type: none"> In the <code>/etc/dracut.conf</code> file, add VirtIO drivers to add_drivers, including <code>virtio_blk</code>, <code>virtio_scsi</code>, <code>virtio_net</code>, <code>virtio_pci</code>, <code>virtio_ring</code>, and <code>virtio</code>. Separate driver names with spaces. Save and exit the <code>/etc/dracut.conf</code> file and run the dracut -f command to generate initrd again. 	<p>SUSE and openSUSE (Later than SUSE 12 SP1 or openSUSE 13)</p>

CentOS and EulerOS

- Run the following command to open the `/etc/dracut.conf` file:
vi /etc/dracut.conf

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

```
[root@CTU10000xxxx ~]# vi /etc/dracut.conf
# additional kernel modules to the default
add_drivers+="virtio_blk virtio_scsi virtio_net virtio_pci virtio_ring virtio"
....
```

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

- 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.

- 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@CTU10000xxxx 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@CTU10000xxxx ~]#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@ CTU10000xxxx 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@ CTU10000xxxx 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 processor.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@CTU10000xxxx ~]# 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@CTU10000xxxx ~]# 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
```

4.10.8 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
```

4.11 Encrypting Images

4.11.1 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

- KMS 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.
- Images of multi-project users cannot be encrypted.

4.11.2 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](#).

4.12 Converting the Image Format

4.12.1 Converting the Image Format Using `qemu-img`

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 4-3 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 the cloud's self-developed image file formats and cannot be identified by **qemu-img**. To convert image files to any of the two formats, use the **qemu-img-hw** tool. For details, see [Converting the Image Format Using `qemu-img-hw`](#)
- When you run a command to convert the format of VHD image files, replace **vhd** with **vpc**.

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
```

 **NOTE**

If an error occurs, delete **-f vpc**. Then, qemu-img can identify the correct image format.

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
```

4.12.2 Converting the Image Format Using `qemu-img-hw`

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 into any of these formats using the open-source tool **qemu-img** before being imported. However, the **qemu-img** tool cannot convert image files to the ZVHD or ZVHD2 format. To convert image files to any of the two formats, use the self-developed tool **qemu-img-hw**. This section describes how to use **qemu-img-hw** to convert an image file to ZVHD2.

Tool and Costs

Table 4-4 Tool and costs

Tool	Description	Costs
qemu-img-hw	qemu-img-hw is used for converting image formats. You can obtain it from: https://cn-south-1-cloud-reset-pwd.obs.cn-south-1.myhuaweicloud.com/imageImportTools/qemu-img-hw.zip	Free

Constraints

qemu-img-hw can be used only in Linux. You can run it on a local Linux server or a Linux ECS on the cloud platform. The following procedure uses an EulerOS ECS as an example.

Procedure

1. Upload the image file to be converted to the ECS.
 - If the local host runs a Linux OS, 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 local host runs a Windows OS, use a file transfer tool, such as WinSCP, to upload the image file to the ECS.
2. Obtain the **qemu-img-hw** software package, upload it to the ECS, and then decompress the package.

Table 4-5 qemu-img-hw package

Tool Package	How to Obtain
qemu-img-hw.zip	https://cn-south-1-cloud-reset-pwd.obs.cn-south-1.myhuaweicloud.com/imageImportTools/qemu-img-hw.zip

 NOTE

This tool can be used only on x86 servers.

3. Convert the image format.
 - a. Go to the directory where **qemu-img-hw** is stored, for example, **/usr/quick-import-tools/qemu-img-hw**.
cd /usr/quick-import-tools/qemu-img-hw
 - b. Run the following command to change file permissions:
chmod +x qemu-img-hw
 - c. Run the **qemu-img-hw** command to convert the image file to the ZVHD2 format.

The command format of **qemu-img-hw** is as follows:

```
./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
```

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 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.

5 Windows Operations

5.1 Configuring 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 configure 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

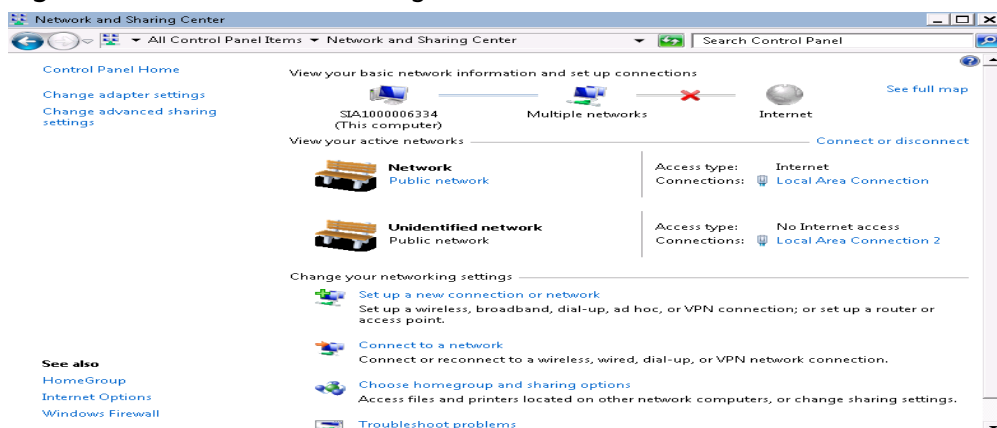
You have logged in to the ECS used to create a Windows private image.

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

Procedure

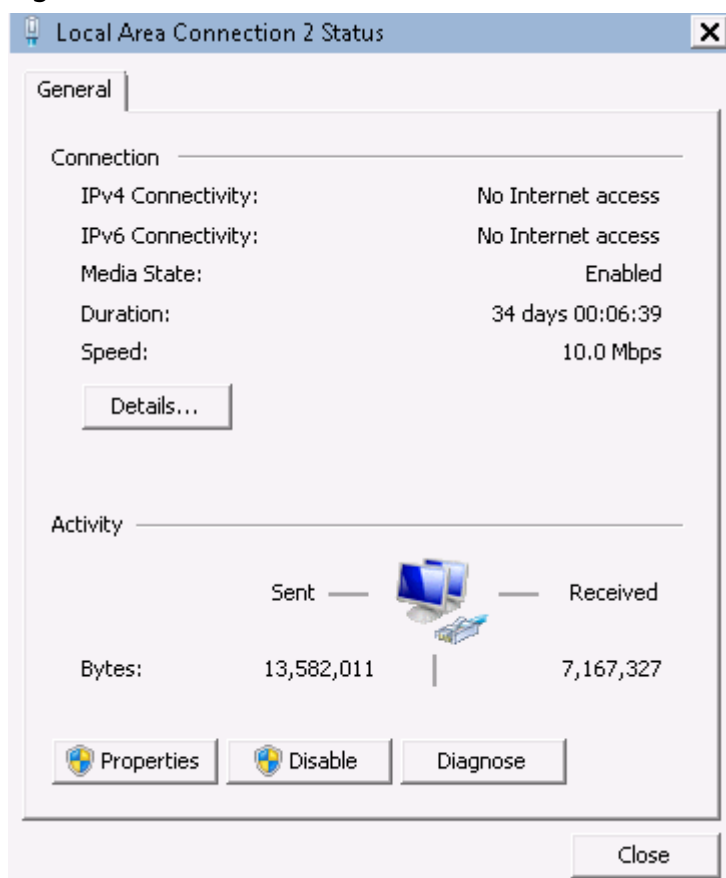
1. On the ECS, choose **Start > Control Panel**.
2. Click **Network and Internet Connections**.
3. Click **Network and Sharing Center**.

Figure 5-1 Network and Sharing Center



4. Select the connection configured with the static IP address. For example, click **Local Area Connection 2**.

Figure 5-2 Local Area Connection 2 Status

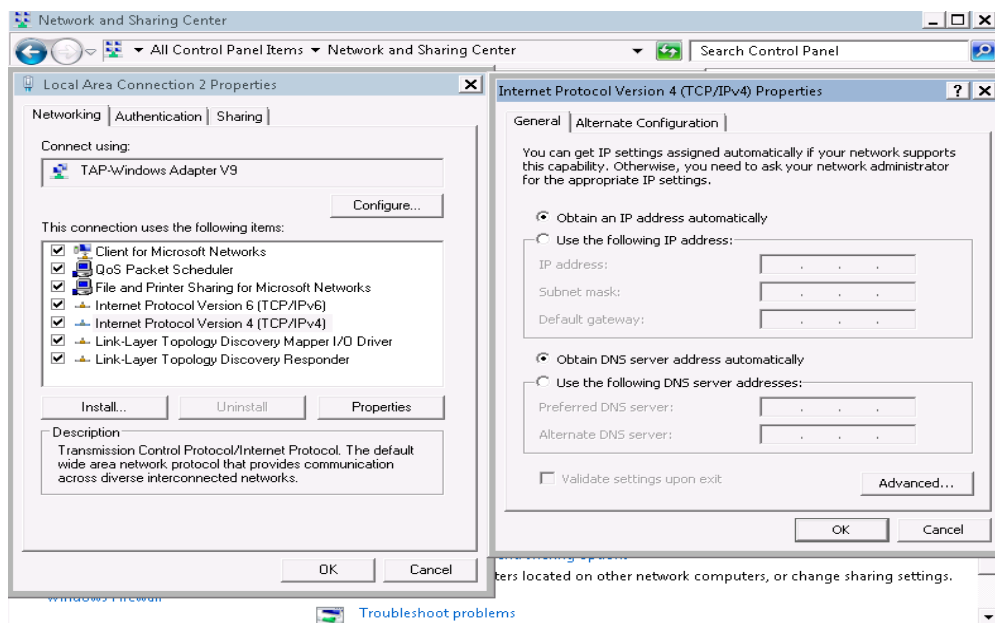


5. Click **Properties** and select the configured Internet protocol version.
6. On the **General** tab, select **Obtain an IP address automatically** and **Obtain DNS server address automatically** and click **OK**, as shown in Figure 5-3. The system will automatically obtain an IP address.

NOTE

You are advised to record the original network information so that you can restore the network if necessary.

Figure 5-3 Configuring DHCP



5.2 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

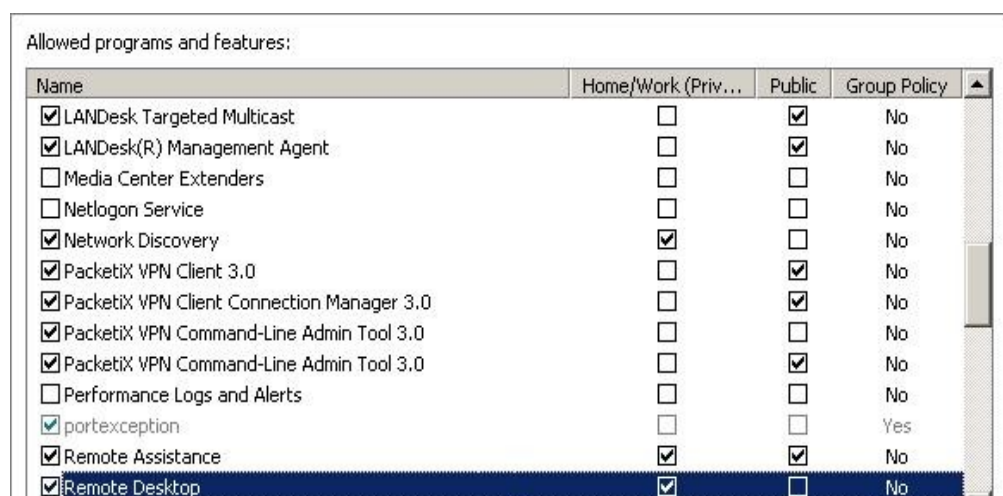
1. Before enabling this function, you are advised to set the resolution of the ECS to 1920×1080.

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 5-4 Configuring remote desktop



5.3 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 ECS uses DHCP to obtain IP addresses.

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/>).

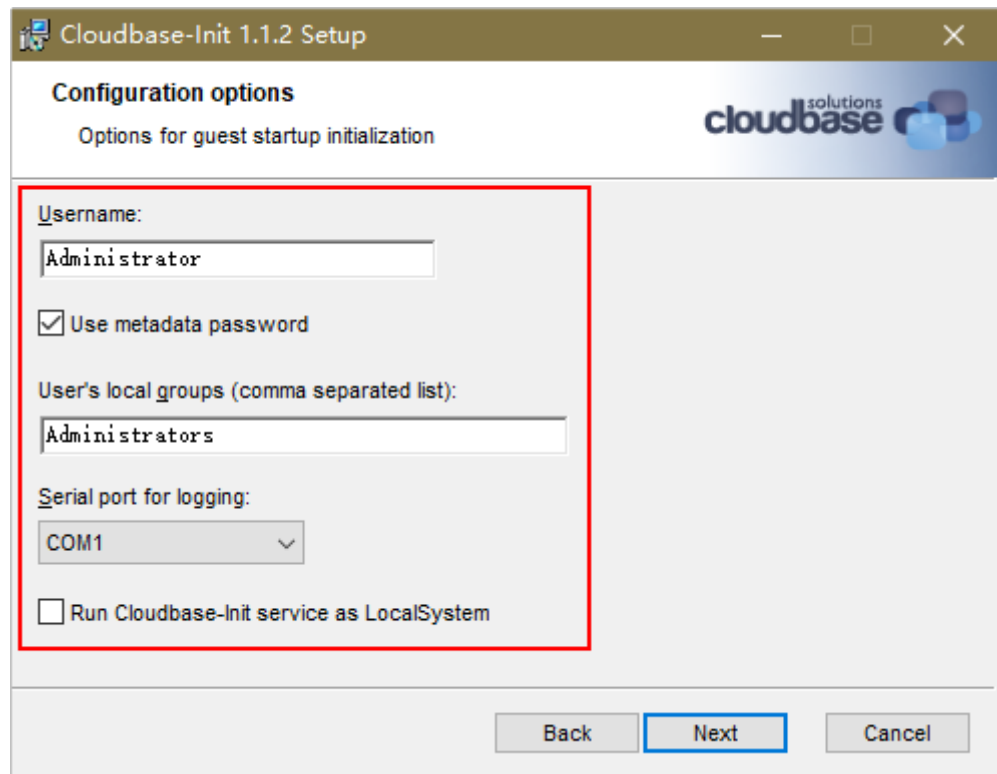
To obtain the stable version, visit the following paths:

 - 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

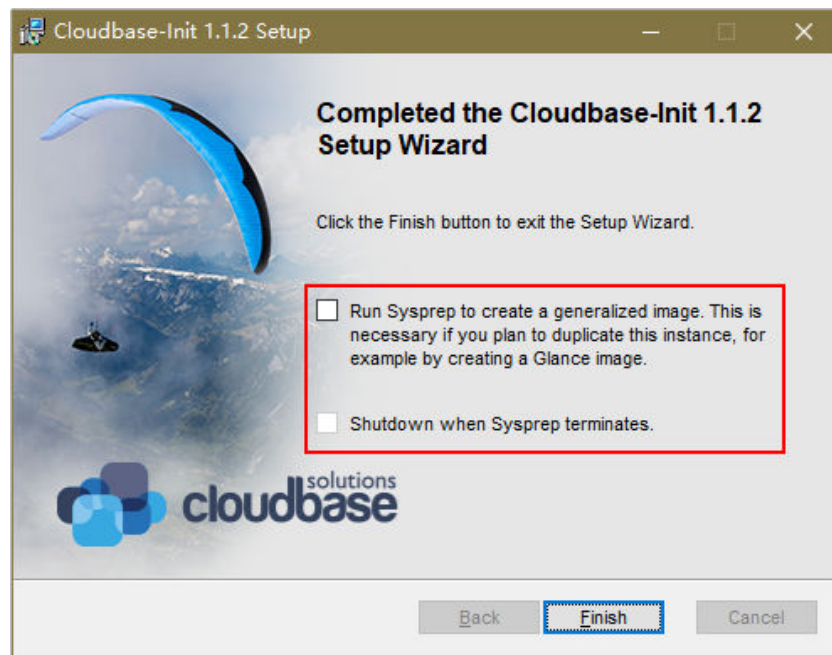
The version number shown in the figure is for reference only.

Figure 5-5 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 5-6 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 disable password changing upon first login:
first_logon_behaviour=no
- h. (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 5-1 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

5.4 Running Sysprep

Scenarios

Running Sysprep ensures that an ECS has a unique SID after it is joined 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

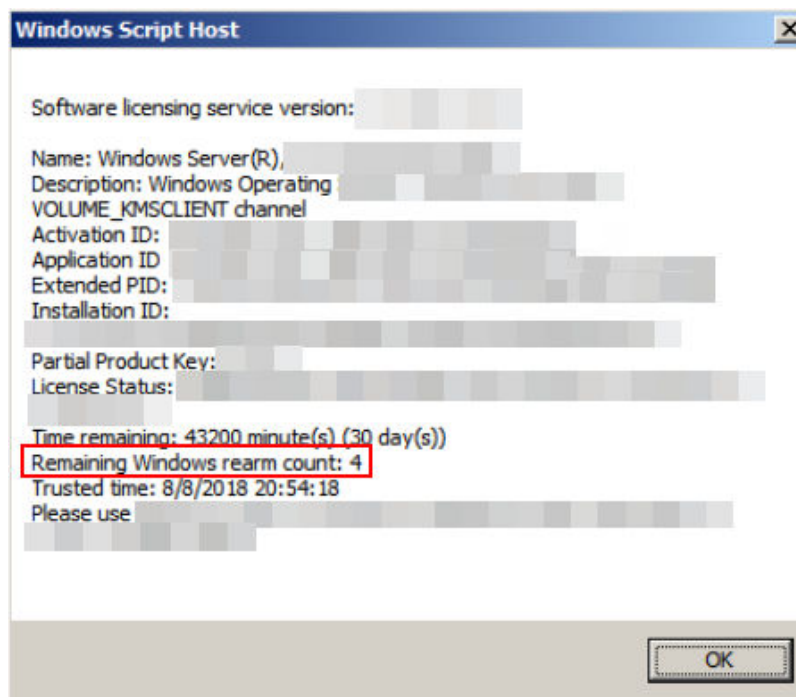
- Run Sysprep as the administrator.
- For a newly activated Windows ECS, you can run Sysprep only once at a time.
- If an ECS is created from an image file, only Sysprep provided by the image file can be used. In addition, Sysprep must always reside in the **%WINDIR%\system32\sysprep** directory.
- Windows 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 5-7 Windows Script Host



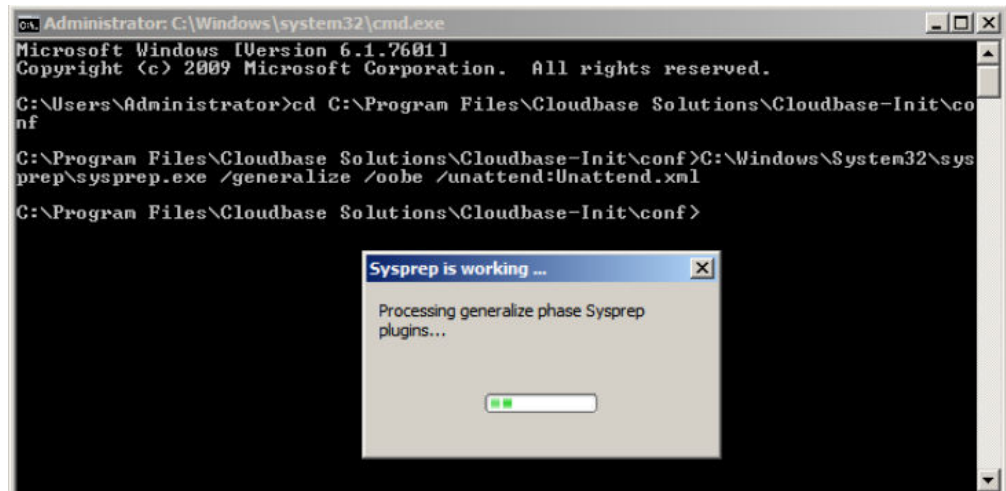
Procedure

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 5-8 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 5-9 ECS SID before Sysprep is executed

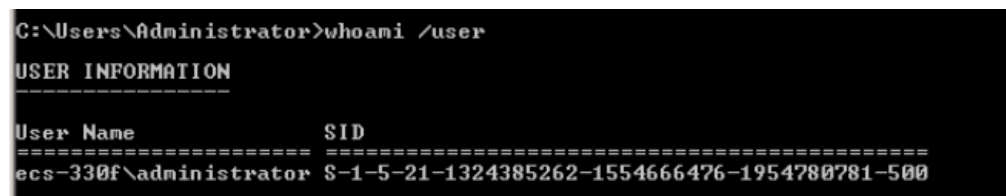
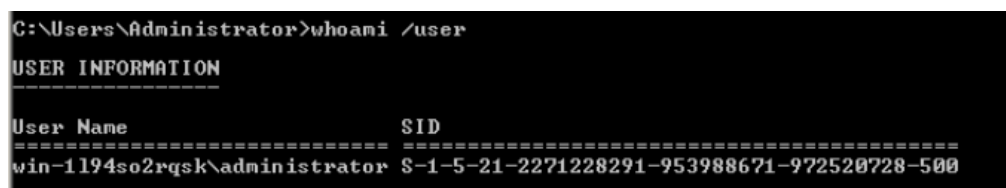


Figure 5-10 ECS SID after Sysprep is executed



5.5 Installing Special Windows Drivers

Scenarios

Before using some types of ECSs to create private images, you need to install special drivers on the ECSs.

GPU Driver

If you want to use the created private image to create GPU-accelerated ECSs, install a proper GPU driver for the image to enable GPU acceleration. There are two types of NVIDIA Tesla GPU drivers for GPU-accelerated ECSs, Tesla and GRID/vGPU drivers.

- To use graphics acceleration, such as OpenGL, DirectX, or Vulkan, install the GRID/vGPU driver and separately configure a GRID license. The GRID/vGPU driver with a vDWS license also supports CUDA for both computing and graphics acceleration.
- To use NVIDIA CUDA computing acceleration, install the Tesla driver.

Table 5-2 Installing the GRID driver

ECS Type	How to Install the Driver
G1	For details, see Instances > Installing a Driver and Toolkit > Installing a GRID Driver on a GPU-accelerated ECS > Downloading GRID Driver and Software License Packages in <i>Elastic Cloud Server User Guide (Paris Region)</i> .
G3	For details, see Instances > Installing a Driver and Toolkit > Installing a GRID Driver on a GPU-accelerated ECS > Downloading GRID Driver and Software License Packages in <i>Elastic Cloud Server User Guide (Paris Region)</i> .
G2	Log in at http://www.nvidia.com/Download/index.aspx?lang=en-us . You are advised to select the latest CUDA Toolkit version. NOTICE After the GPU driver is installed, run the following command to switch the GPU working mode and restart the ECS (for example, the GPU driver is installed in <code>C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi.exe</code>): <code>"C:\Program Files\NVIDIA Corporation\NVSMI\nvidia-smi.exe" -dm 0</code>

SR-IOV NIC Driver

If you want to use the created private image to create G2 ECSs, install the SR-IOV NIC driver for the image to improve performance and scalability.

To download the SR-IOV driver, log in at <https://downloadcenter.intel.com/search?keyword=Intel++Ethernet+Connections+CD>. You are advised to select version 20.4.1 or later.

If error "No Intel adapter found" occurs during the driver installation, refer to [What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?](#) for troubleshooting.

6 Linux Operations

6.1 Configuring 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 configure 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.

6.2 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
```

 - o If no, no further action is required.
 - o 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
```

 - o If no, no further action is required.
 - o 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
```

6.3 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 ECS uses DHCP to obtain IP addresses.

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 (Ubuntu 22.0.4 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 6-1 lists the Cloud-Init installation paths for CentOS. Select the required installation package from the following addresses.

Table 6-1 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/archive/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 6-1](#).

- 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/archive/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>
```

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**.
-

6.4 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 ECS uses DHCP to obtain IP addresses.

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 6-1 Example configuration

```
35 # max_wait: 10 # (defaults to 120 seconds)
36 +datasource_list: [ OpenStack ]
37 +datasource:
38 + OpenStack:
39 + metadata_urls: ['http://[REDACTED]']
40 + max_wait: 120
41 + timeout: 5
42 + apply_network_config: false
43
```

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.

Take Ubuntu as an example.


```
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
$Zh4lchiJR7NuZvtJHsYBQJlg5RoQCRLS1X2Hsgj2s5JwXI7KUO1we8WYcwbzeaS2VNPmNo28vmxx
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 6-2 Preventing Cloud-Init from taking over the network

```

users:
  - default

disable_root: 1
ssh_pwauth: 0

datasource_list: [ OpenStack ]
datasource:
  OpenStack:
    metadata_urls: ['http://[redacted]']
    max_wait: 120
    timeout: 50

network:
  config: disabled
  
```

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 6-3 Speeding up the SSH login to the ECS

```

cloud_init_modules:
- ssh
- migrator
- bootcmd
- write-files
- growpart
- resizefs
- set_hostname
- update_hostname
- update_etc_hosts
- rsyslog
- users-groups
  
```

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 6-4 Modifying the `__init__.py` file

```
192 # if there is an ipv4 address in 'local-hostname', then
193 # make up a hostname (LP: #475354) in format ip->xx.xx.xx.xx
194 lhost = self.metadata['local-hostname']
195 if util.is_ipv4(lhost):
196     toks = []
197     if resolve_ip:
198         toks = util.gethostbyaddr(lhost)
199
200     if toks:
201         toks = str(toks).split('.')
202     else:
203         toks = ["ip-%s" % lhost.replace(".", "-")]
204 else:
205     toks = lhost.split(".nova.local")
206
207 if len(toks) > 1:
208     hostname = toks[0]
209     #domain = '.'.join(toks[1:])
210 else:
211     hostname = toks[0]
212
213 if fqdn and domain != defdomain:
214     return "%s.%s" % (hostname, domain)
215 else:
216     return hostname
```

- 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 6-5 Setting the parameter value to `cloudLogHandler`

```
[logger_cloudinit]
level=DEBUG
qualname=cloudinit
handlers=cloudLogHandler
propagate=1
```

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**.

6.5 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 6-6 EVS disk configuration in the **fstab** file

```
[root@ecs-bf78 ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Wed Feb 27 06:58:16 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=4c2c090d-4228-49fc-9cbe-3920b3bf287c / ext4 defaults 1 1
UUID=9c29104b-31b8-4421-a207-102f86ec7ae5 /mnt/test ext4 defaults 1 1
```

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.

7 Managing Tags

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.


8 Managing Quotas

What Is Quota?

Quotas can limit the number or amount of resources available to users, such as the maximum number of ECS 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. In the upper right corner of the page, click  .
The **Quotas** page is displayed.
3. View the used and total quota of each type of resources on the displayed page.
If a quota cannot meet service requirements, apply for a higher quota.

How Do I Apply for a Higher Quota?

The system does not support online quota adjustment.

If you need to adjust a quota, contact the operations administrator.

9 Auditing Key Operations

9.1 IMS Operations Audited 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 9-1 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
Replicating an image	ims	copyImage
Exporting an image	ims	exportImage

Operation	Resource Type	Trace Name
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 9-2 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
Deleting a tenant from the group where the members can use a shared image	deleteMember	IMS	image	glance

9.2 Viewing Traces




Scenarios

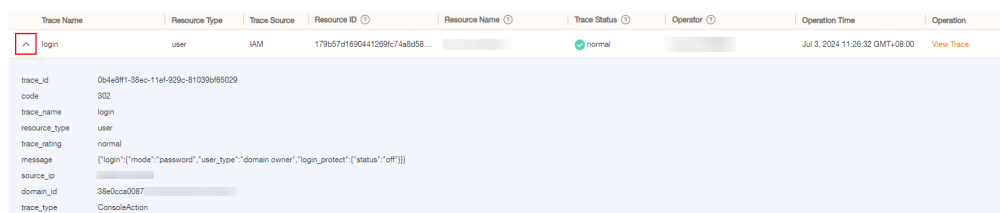
After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. Cloud Trace Service (CTS) stores operation records (traces) generated in the last seven days.

NOTE

These operation records are retained for seven days on the CTS console and are automatically deleted upon expiration. Manual deletion is not supported.

Viewing Real-Time Traces

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. 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:** Select **Last 1 hour, Last 1 day, or Last 1 week**, or specify a custom time range within the last seven days.
5. Click **Query**.
6. 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 5,000 records.
 - Click  to view the latest information about traces.
7. Click  on the left of a trace to expand its details.



Trace Name	Resource Type	Trace Source	Resource ID	Resource Name	Trace Status	Operator	Operation Time	Operation
login	user	IAM	179b67d169041299c74a8d58...		normal		Jul 3, 2024 11:26:32 GMT+08:00	View Trace

trace_id	0b4e8f11-38ac-11ef-929c-81039b65029
code	302
trace_name	login
resource_type	user
trace_status	normal
message	[{"login":{"mode":"password","user_type":"domain owner","login_protect":{"status":"off"}}}]
source_ip	
domain_id	38e0cca0087
trace_type	ConsoleAction

10 FAQs

10.1 Image Consulting

10.1.1 Basic Concepts

Images are classified as public, private, and shared.

Image Type	Description
Public	<p>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.</p>
Private	<p>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.</p> <p>A private image can be a system disk image, data disk image, ISO image, or full-ECS image.</p> <ul style="list-style-type: none">• 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, see "Sharing Images" in <i>Image Management Service User Guide</i> .

You can modify an image, share images, export images, encrypt images, replicate images, export an image list, and delete images.

Table 10-1 Managing private images

Feature	Description	Helpful Link
Modifying an image	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.	<ul style="list-style-type: none">• Sharing Images• Image Sharing
Exporting images	You can export private images to your OBS bucket and download them to your local PC for backup.	<ul style="list-style-type: none">• Exporting an Image• Image Export
Encrypting images	You can create encrypted images to improve data security. KMS envelope encryption is used. Encrypted images can be created from external image files or encrypted ECSs.	<ul style="list-style-type: none">• Encrypting Images
Replicating images	By replicating images, you can convert encrypted and unencrypted images into each other or enable some advanced features, for example, fast instance provisioning.	Replicating Images Within a Region
Tagging an image	You can tag your private images for easy management and search.	Managing Tags
Exporting image list	You can export the public or private image list in a given region as a CSV file for local maintenance and query.	Exporting Image List

Feature	Description	Helpful Link
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

10.1.2 How Do I Select an Image?

When creating an ECS or a BMS, 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	<ul style="list-style-type: none">• 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	<ul style="list-style-type: none">• 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 1 GB, and there must be at least 1 GB of memory.
Linux	<ul style="list-style-type: none">• High-performance server applications (for example, Web) and 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 1 GB, and there must be at least 512 MB of memory.

10.1.3 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 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](#).

10.1.4 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 10-2](#) lists the differences between them.

Table 10-2 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	<ul style="list-style-type: none"> ● 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 	<ul style="list-style-type: none"> ● 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	<ul style="list-style-type: none"> ● Data backup and restoration ● Rapid service deployment and migration 	<ul style="list-style-type: none"> ● 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.

10.1.5 Can I Tailor an Image?

Yes, but you'd better not.

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.

10.1.6 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 for the ECS. If the ECS fails, select a backup you want to use to restore the ECS, create a full-ECS image from the backup, and use the image to create a new ECS or to reinstall the OS of the ECS.
- Create a system disk image from the ECS. If the ECS fails, use the system disk image to create a new ECS or to reinstall the OS.

- Create a snapshot for the system disk of the ECS. If the ECS fails, you can restore the system disk data from the snapshot.

10.1.7 How Can I Apply a Private Image to an Existing ECS?

- You can use the image to change the ECS OS. When you change the OS, select that image. For details about how to change the OS, see "Changing the OS" in *Elastic Cloud Server User Guide*.

10.1.8 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.

10.1.9 Can I Use Private Images of Other Accounts?

Yes.

You can use the private images other accounts shared with you.

For details, see [Sharing Specified Images](#).

10.2 End-of-Support for OSs

10.2.1 What Do I Do If CentOS Linux Is No Longer Maintained?

CentOS has planned to stop maintaining CentOS Linux. The cloud platform will stop providing CentOS Linux public images. This section describes the impacts and tells you how to address the situation.

Background

On December 8, 2020, CentOS announced its plan to stop maintaining CentOS Linux and launched CentOS Stream. For more information, see [CentOS Project shifts focus to CentOS Stream](#).

CentOS Linux 8 ended on December 31, 2021, and CentOS Linux 7 will end on June 30, 2024. CentOS Linux 9 and later versions will not be released, and patches will no longer be updated.

Impacts

CenterOS Linux users will be affected as follows:

- After December 31, 2021, CentOS Linux 8 users will not be able to obtain any maintenance or support services, including problem fixing and function updates.
- After June 30, 2024, CentOS Linux 7 users will not be able to obtain any maintenance or support services, including problem fixing and function updates.

The cloud users will be affected as follows:

- CentOS Linux 8 public images will continue for a certain time. ECSs created from CentOS Linux 8 images will not be affected, but the images will no longer be updated.
- The cloud platform will synchronize with CentOS for the support of CentOS Linux. After December 31, 2021, support services will no longer be available for CentOS 8. The support for CentOS 7 will continue until June 30, 2024.

Solution

You can **change** the OS so that the services originally running in CentOS Linux can continue to run in other OSs.

For details about how to change to CentOS Stream or Rocky Linux, see "Instances > Managing ECSs > Changing the OS" in *Elastic Cloud Server User Guide*.

Table 10-3 Precautions for changing an OS

Item	Precautions for Changing an OS
Data backup	<ul style="list-style-type: none">• Data in all partitions of the system disk will be cleared, so you are advised to back up the system disk data prior to an OS change.• Data in data disks remains unchanged.
Custom settings	After the OS is changed, custom settings such as DNS and hostname will be reset and need to be reconfigured.

Table 10-4 Available OSs

OS	Description	Intended User
CentOS Stream	CentOS Stream is a continuous delivery distribution provided by CentOS.	Individuals or enterprises that are used to CentOS and desire continuous updates
Rocky Linux	Rocky Linux is a community-driven enterprise-class OS. It is a downstream release of Red Hat Enterprise Linux (RHEL). Rocky Linux is fully compatible with and as stable as CentOS.	Individuals or enterprises that want to continue to use free images in an open source community

OS	Description	Intended User
Debian and Ubuntu	They are Linux distributions that differ in use and compatibilities.	Individuals or enterprises that can afford the OS change costs

10.3 Image Creation

10.3.1 General Creation FAQ

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?

You can create up to 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. But the data that was written into the ECS during the image creation will not be included in the image.

Where Can I Check 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. Check the image creation progress in the **Status** column.

Creating an image may take a period of time because it requires to install Xen and KVM drivers, load the OS kernel, and configure GRUB boot. In addition, the network speed, image file type, and disk size all have impacts on how long the entire process takes.

Can I Use a Private Image of an IAM User Under My Account to Create an ECS?

Yes.

Private images created by an IAM user are visible to the account that the IAM user belongs to as well as all other IAM users (if any) under this account.

- You can use a system disk image or full-ECS image to create an ECS.
- You can use a data disk image to create an EVS disk.

In addition, private images created by an account are visible to IAM users under this account.

10.3.2 Full-ECS Image FAQ

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:

- 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 image and backup are stored. You need to pay for the vault.

When creating a full-ECS image from a CSBS backup, the space used to store the image and CSBS backup is not open to users but still need to be billed.

So, no matter which backup type you select, you need to pay for the storage. Selecting a vault does not mean that you need to pay extra fees.

Where Can I Check the Data Disk Details of a Full-ECS Image?

To check data disk details, you need to go to the CSBS or CBR console, depending on where the full-ECS image is created from. That is because only system disk information (**Disk Capacity**) is displayed in the image list and image details after a full-ECS image is created.

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 backup 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.
- A full-ECS image cannot be shared.
- A full-ECS image cannot be replicated within the same region.

10.3.3 How Can I Use a Backup to Create an EVS Disk or ECS?

You can use CSBS backups to create ECSs and use VBS backups to create EVS disks.

- CSBS backups cannot be directly used to create ECSs. You need to use a backup to create a private image and then use the private image to create ECSs.

For details about how to create a private image from a CSBS backup, see [Creating a Full-ECS Image from a CSBS Backup](#). For details about how to use a private image to create ECSs, see [Creating an ECS from an Image](#).

- VBS backups can be directly used to create EVS disks. For details, see "Using a Backup to Create a Disk" in *Cloud Backup and Recovery User Guide*.

10.3.4 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, a CSBS backup, 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.

10.3.5 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 is only an intermediate product and is not be available on the ECS console. The final product is a system disk image that has an OS and drivers installed.
- 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](#)

10.3.6 How Do I Create a Full-ECS Image Using an ECS That Has a Spanned Volume?

You are not advised to use a Windows ECS that has a spanned volume to create a full-ECS image. If you create such an image and then use it 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 the volume. Use the ECS to create a full-ECS image. You can then use the full-ECS image to create an ECS and 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.

10.3.7 Why Is Sysprep Required for Creating a Private Image from a Windows ECS?

Why Is Sysprep Required?

Sysprep is used to generalize images. It removes server-specific information, like the security identifier (SID), from an image so that ECSs created from this image can have unique SIDs in a domain. If your windows ECSs do not need to be joined to a domain, Sysprep is not required.

 CAUTION

Before running Sysprep, ensure that Windows is activated.

Restrictions on Running Sysprep

Sysprep can only be used to configure new installations of Windows and not to reconfigure an existing installation. You can run Sysprep as many times as required to build and to configure your installation of Windows. However, you can reset Windows activation only up to three times.

 NOTE

In the Windows command line, run the following command to check how many times you can run Sysprep:

```
slmgr /dlv
```

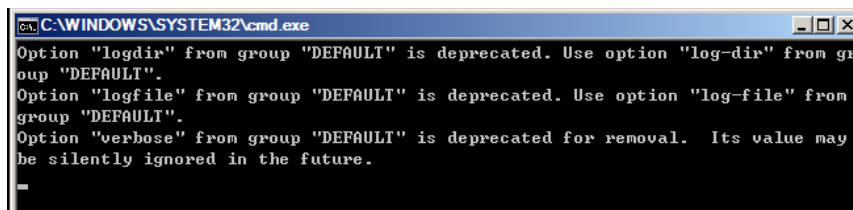
In the displayed **Windows Script Host** dialog box, if the value of **Remaining Windows rearm count** is 0, you cannot run Sysprep.

10.3.8 How Do I Handle the Startup Failure of a Windows ECS Created from a Windows Image Generalized by Sysprep?

Symptom

1. When you start the ECS, information similar to the following is displayed.

Figure 10-1 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 10-2 Viewing ECS logs

```

115 2018-08-28 23:43:15, Info [SETUPGDC.EXE] Running commands is already in progress.
116 2018-08-28 23:43:15, Info [0x000000] PANTER CRackboard:Open: C:\Windows\panther\command.exe(CommandExec) succeeded.
117 2018-08-28 23:44:33, Error [SETUPGDC.EXE] Hit an error (0x = 0x00000001) while running cmd.exe /c ""C:\Program Files\Cloudbase Solutions\Cloudbase-Init\Python
C:\Program Files\Cloudbase Solutions\Cloudbase-Init\conf\cloudbase-init-unattend.conf" %* exit 1 || exit 21
118 2018-08-28 23:44:33, Info [0x000000] PANTER CRackboard:Close: C:\Windows\panther\command.exe(CommandExec) succeeded.
119 2018-08-28 23:44:33, Info [SETUPGDC.EXE] SetupGDC returning with exit code (4).
120 2018-08-28 23:44:53, Info [windeploy.exe] Process exited with exit code (0x1f).
121 2018-08-28 23:44:53, Error [windeploy.exe] Setup.exe failed, returning exit code (0x1f).
122 2018-08-28 23:44:53, Error [windeploy.exe] Failure occurred during online installation. Online installation cannot complete at this time. hr = 0x80004003
123 2018-08-28 23:44:53, Info [windeploy.exe] Flushing registry to disk....
124 2018-08-28 23:44:53, Info [windeploy.exe] Flush took 16 ms..

```

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 10-3 Deleting the RunSynchronous snippet

```

- <settings pass="specialize">
- <component language="neutral" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:wcm="http://schemas.microsoft.com
versionScope="nonSxS" publicKeyToken="31bf3856ad364e35" processorArchitecture="amd64" name="Microsoft-Windows-Deployment">
  <RunSynchronous>
    - <RunSynchronousCommand wcm:action="add">
      <Order>1</Order>
      <Path>cmd.exe /c ""C:\Program Files\Cloudbase Solutions\Cloudbase-Init\Python\Scripts\cloudbase-init.exe" --config-file
Solutions\Cloudbase-Init\conf\cloudbase-init-unattend.conf" && exit 1 || exit 2"</Path>
      <Description>Run Cloudbase-Init to set the hostname</Description>
      <WillReboot>OnRequest</WillReboot>
    </RunSynchronousCommand>
  </RunSynchronous>
</component>
</settings>
</unattend>

```

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.

10.3.9 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_ild** role (**op_gated_ild** 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 customer service to ensure that the current environment supports lazy loading. Obtain a new token and use the new token to create an image.

10.4 Image Sharing

10.4.1 General Sharing FAQ

How Many Tenants Can I Share an Image With?

128

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, and 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 is 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 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.

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 create 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 Others?

You cannot directly share such images with other tenants. If you do need to do so, you can replicate a shared image as a private image and then share it.

Can I Use an Image I Have Shared with Others to Create an ECS?

Yes. After sharing an image with others, 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 it to create a new ECS or change the OS of an existing ECS.

The cloud platform does not ensure the integrity or security of images shared by other accounts. You are advised to choose only images shared by trusted accounts.

What Are the Risks of Sharing Images?

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 others, 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 ID. You cannot specify a region or an AZ. An image can only be shared within a given region. 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 a shared image to create a new data disk but not to restore your existing data disks. However, you can use the new data disk to restore data. For details, see [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 share the image with you again. For details, see [Adding Tenants Who Can Use Shared Images](#).
- Method 2
Accept the rejected image again. For details, see [Accepting Rejected Images](#).

10.4.2 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) by replicating an image.

The following table describes the details.

Scenario	Operation	Description	Helpful Link
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 is still owned by the user who shared it.	Sharing Specified Images
In-region replication under the same account	Replicate	This is used for conversion between encrypted and unencrypted images or for enabling advanced features (such as fast ECS creation from an image).	Replicating Images Within a Region

10.4.3 Why Can't I Share My Images?

Some images cannot be shared. Therefore, the **Share** option is not provided for them in the **Operation** column.

Images can only be shared within the same region.

For details about image sharing, see [Sharing Images](#).

10.5 OS

10.5.1 How Do I Select an OS?

- Windows
Used for development platforms or services that run Windows.
The system disk must be at least 40 GB, and there must be at least 1 GB of memory.
Internet Information Services (IIS) and SQL Server can be installed.
- Linux
Used for development platforms or services that run Linux. CentOS and Ubuntu are provided. CentOS is recommended.
The system disk must be at least 40 GB, and there must be at least 512 MB of memory.
- If your servers require more than 4 GB of memory, select a 64-bit OS because 4 GB is the maximum memory a 32-bit OS can access.

10.5.2 How Is BIOS Different from UEFI?

Table 10-5 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.

10.5.3 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

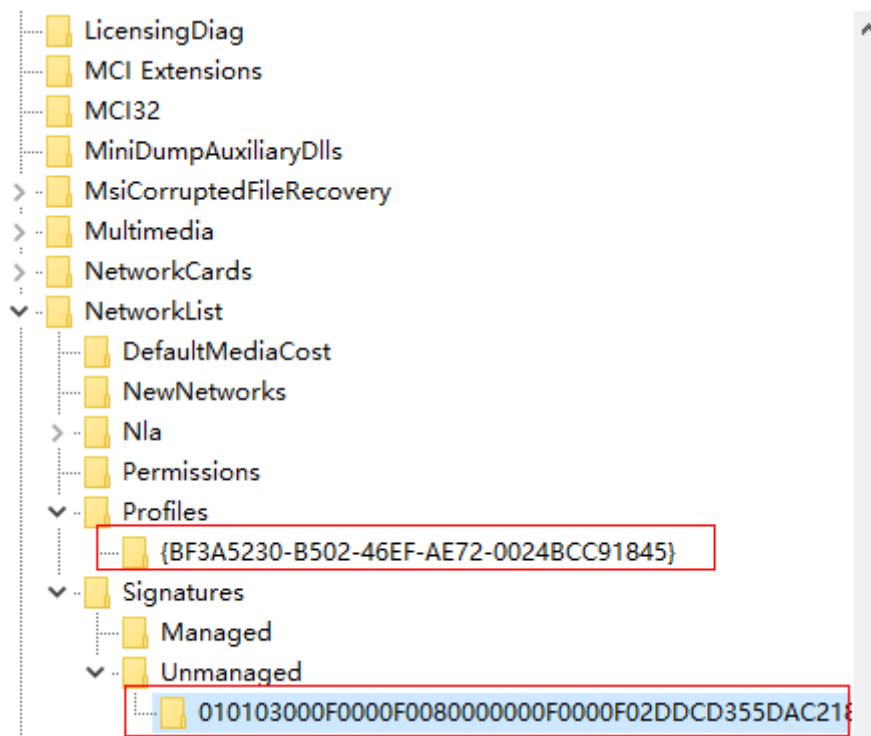
1. Press **Win+R**. In the displayed dialog box, enter **regedit** and press **Enter** to open the registry editor.

 **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 10-4 Registry directory



10.5.4 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 10-5 Querying the GRUB file version

```
[root@... ]# rpm -qa | grep grub  
grub2-2.02-0.44.el7.centos.x86_64
```

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 10-6 Modifying the timeout duration

```
#boot=/dev/sda
default=0
timeout=8
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-696.16.1.el6.x86_64)
  root (hd0,1)
  kernel /boot/vmlinuz-2.6.32-696.16.1.el6.x86_64 ro root=UUID=2bc0f5fd-e8
19-4ba5-8ce0-8fe12b6efc24 rd_NO_LUKS rd_NO_LVM LANG=en_US.UTF-8 rd_NO_MD SYSFONT
=latarecyrheb-sun16 crashkernel=auto KEYBOARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb q
  initrd /boot/initrd.img-2.6.32-696.16.1.el6.x86_64
```

10.5.5 What Do I Do If a Windows 7 ECS Equipped with an Intel 82599 NIC Reports an Error in SR-IOV Scenarios?

Symptom

When the 20.4.1 driver package downloaded at Intel website <https://downloadcenter.intel.com/search?keyword=Intel++Ethernet+Connections+CD> was installed in a Windows 7 64bit ECS with SR-IOV passthrough enabled, the system displayed the message "No Intel adapter found".

Cause Analysis

The OS identifies an Intel 82599 passthrough NIC without a driver installed as an Ethernet controller. When the 20.4.1 driver package was installed, the OS did not identify the Intel NIC, leading to the error.

Solution

Run **Autorun.exe** in the folder where the 20.4.1 driver package is stored. Install a driver on the NIC before installing the driver package so that the NIC can be identified as an Intel 82599 virtual function (VF) device by the OS. Use either of the following methods to install the driver:

- Method 1: Update the version.
 - a. Download the 18.6 driver package at the Intel website.
 - b. Run **Autorun.exe**.
 - c. Run **Autorun.exe** in the folder where the 20.4.1 driver package is stored to update the driver.
- Method 2: Use the device manager.
 - a. Start the Windows resource manager. Right-click **Computer** and choose **Manage** from the shortcut menu. In the **Device Manager** window, locate the NIC. When the NIC has no driver installed, the NIC locates in **Other devices** and is named **Ethernet Controller**.
 - b. Right-click **Ethernet Controller** and choose **Update Driver Software**.
 - c. Click **Browse**, select the path where the driver package is stored, and click **Next**.
 - d. Locate the NIC in **Network Adapter** of **Device Manager**.
 - e. Run **Autorun.exe** to install the 20.4.1 driver package.

10.5.6 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.

10.6 Image Import

10.6.1 Can I Use Images in Formats not Described in This Document?

No. Currently, only VMDK, VHD, RAW, QCOW2, VHDX, QED, VDI, QCOW, ZVHD2, ISO, and ZVHD formats are supported.

Images in -flat.vmdk format and image file packages containing snapshot 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/>

10.6.2 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 are some potential impacts:

1. If you do not delete network rule files from the **udev** directory, those rules will be applied to newly created ECSs. Also, if you do not configure DHCP, the NICs of newly created ECSs will not start from eth0. You need to remotely log in to the new ECSs to resolve these issues.

2. When creating Linux ECSs:
 - Custom passwords cannot be injected.
 - Certificates cannot be injected.
 - Other custom configurations cannot be applied to new ECSs.
3. If you do not delete the automatic mount configuration from the **fstab** file, new ECSs may fail to start.

10.6.3 What Do I Do If I Chose the Wrong OS or System Disk Capacity When Registering a Private Image?

If you selected the wrong OS, the ECS creation may fail when the image is used.

If the system disk capacity you configured is less than the one in the image file, image registration will fail.

In such cases, delete the image and create a new one using the correct settings.

10.6.4 Why Did My VHD Upload Fail? Why Does the System Say the System Disk in the VHD Image File Is Larger Than What I Specified on the Management Console?

The possible causes may be:

1. Too small a value was specified when registering the image. Check the system disk capacity in the VHD image file. Specify a value at least this large when you use the VHD image file to register an image.
2. The VHD's actual disk size is larger than its virtual size. This can happen if the VHD image file was generated using **qemu-img** or a similar tool. For details, see <https://bugs.launchpad.net/qemu/+bug/1490611>.

Run the following command to check the VHD image file information:

```
[xxxx@xxxx 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 always an integer number of GBs. As a result, if an actual size is, like in the example here, **2147991552 bytes (2.0004 GB)**, the virtual size will be only **2 GB**. In this example, you need to specify an integer larger than the actual size 2.0004 GB. The system disk capacity on the management console can only be an integer value, so you enter an integer larger than 2 GB.

10.7 Image Export

10.7.1 Can I Download My Private Images to a Local PC?

Yes. You can download private images as instructed in [Exporting an Image](#).

Currently, only images in VMDK, VHD, QCOW2, or ZVHD format can be exported.

The default format of a private image is ZVHD2. Images exported in different formats may vary in size.

10.7.2 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. So, it cannot be used on a BMS.

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

10.7.3 Why Is the Image Size in an OBS Bucket Different from That Displayed in IMS?

Symptom

When 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, but in an OBS bucket, the size is 2.91 GB.

Cause Analysis

The default format of a private image is ZVHD2, but it may be stored in a different format (VMDK, VHD, QCOW2, or ZVHD) in an OBS bucket after it is exported. The format conversion may lead to size changes.

10.7.4 Can I Download a Public Image to My PC?

You cannot directly download a public image. However, 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 then download the image.

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.

10.7.5 What Are the Differences Between Import/Export and Fast Import/Export?

Item	Description	Helpful Link
Import	<p>Import an external image file to the management console for creating a private image.</p> <p>External image files in the following formats can be imported: VMDK, VHD, QCOW2, RAW, VHDX, QED, VDI, QCOW, ZVHD2, and ZVHD.</p> <p>Maximum file size: 128 GB</p> <p>During the import, operations such as driver injection will be performed in the background. Therefore, the import takes a longer time than fast import.</p>	<ul style="list-style-type: none"> • Creating a Windows System Disk Image from an External Image File • Creating a Linux System Disk Image from an External Image File
Fast import	<p>When importing an external image file in the RAW or ZVHD2 format to the management console, you can select Enable Fast Create. The import is fast because the system does not perform any operations like driver injection.</p> <p>Verify that:</p> <ul style="list-style-type: none"> • 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. <p>Maximum file size: 1 TB</p>	<p>Fast Import of an Image File</p>
Export	<p>You can export private images to OBS buckets and download them to your local PC for further use on other cloud platforms.</p> <p>Maximum file size: 128 GB (If an image file is larger than 128 GB, use fast export.)</p> <p>You can specify the format of the exported image file. Currently, only QCOW2, VMDK, VHD, and ZVHD are supported.</p>	<p>Exporting an Image</p>

Item	Description	Helpful Link
Fast export	<p>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.</p> <p>The file size is not limited.</p> <p>Encrypted images do not support fast export.</p>	Exporting an Image

10.7.6 Why the Export Option Is Unavailable for My Image?

Some images cannot be exported. So, 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

10.8 Image Optimization

10.8.1 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 PV and VirtIO drivers on ECSs.
- Linux ECSs: Install Xen PV and VirtIO drivers and add them to initrd.

10.8.2 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.

- Windows does not have VirtIO drivers installed by default.
- 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 by the community can be eliminated from drivers as soon as possible.

What Do I Need to Do?

- Upgrade VirtIO drivers in Windows private images or running Windows ECSs.
- If you have any technical issues or questions, contact the customer service.

10.8.3 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 OS kernel boot parameters and change disk partition names to UUID.
- Change disk partition names in the **/etc/fstab** file to UUID.
- Check whether the **initrd** file has Xen and IDE drivers. If no, the system will load the Xen and IDE drivers.
- 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**.

NOTE

For the following image files, the system does not copy built-in VirtIO drivers 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.
- Restore the PV driver registry key to rectify driver installation failures and Xen driver conflicts.
- Restore DHCP so that the OS will dynamically obtain information such as the IP address based on the DHCP protocol.

10.8.4 How Do I Configure an ECS, a BMS, or an Image File Before I Use It to Create an Image?

ECS or Image File Configurations

Table 10-6 ECS configurations

OS	Configuration	Reference
Windows	<ul style="list-style-type: none"> • Configure DHCP. • Enable remote desktop connection. • (Optional) Install special Windows drivers. • (Optional) Install Cloudbase-Init. • Install Guest OS drivers (PV and VirtIO drivers). • Run Sysprep. 	Creating a System Disk Image from a Windows ECS
Linux	<ul style="list-style-type: none"> • Configure DHCP. • (Optional) Install Cloud-Init. • Delete files from the network rule directory. • Change disk identifiers in the GRUB file to UUID. • Change disk identifiers in the fstab file to UUID. • Install native Xen and KVM drivers. • Detach data disks from the ECS. 	Creating a System Disk Image from a Linux ECS

Table 10-7 Image file configurations

OS	Configuration	Reference
Windows	<ul style="list-style-type: none"> • Configure DHCP. • Enable remote desktop connection. • Install Guest OS drivers (PV and VirtIO drivers). • (Optional) Install Cloudbase-Init. • (Optional) Enable NIC multi-queue. 	Preparing an Image File

OS	Configuration	Reference
Linux	<ul style="list-style-type: none"> • Delete files from the network rule directory. • Configure DHCP. • Install native Xen and KVM drivers. • Change disk identifiers in the GRUB file to UUID. • Change disk identifiers in the fstab file to UUID. • Delete the automatic mount configuration of non-system disks from the /etc/fstab file. • (Optional) Install Cloud-Init. • (Optional) Enable NIC multi-queue. 	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.

BMS or Image File Configurations

Table 10-8 BMS configurations

OS	Configuration	Reference
Windows	<ul style="list-style-type: none"> • Install the bms-network-config package. • Install Cloudbase-Init. • Delete residual files from the OS. 	"Creating a Private Image from a BMS" in <i>Bare Metal Server User Guide</i>
Linux	<ul style="list-style-type: none"> • Install software in the bms-network-config package. • Install Cloud-Init. • Delete residual files from the OS. 	"Creating a Private Image from a BMS" in <i>Bare Metal Server User Guide</i>

Table 10-9 Image file configurations

OS	Configuration	Reference
Windows	<ul style="list-style-type: none">• Install drivers for x86 V5 BMSs.• Install Cloudbase-Init.• Install software in the bms-network-config package.• (Optional) Install the SDI iNIC driver.• Set the Windows time zone.• Set the virtual memory.• (Optional) Configure automatic Windows update.• Configure SID.	<i>Bare Metal Server Image Creation Guide</i>
Linux	<ul style="list-style-type: none">• Install and configure Cloud-Init.• Modify the hardware device driver that boots the OS.• Install software in the bms-network-config package.• (Optional) Install the SDI iNIC driver.• (Optional) Install the Hi1822 NIC driver.• (Optional) Install the IB driver.• (Optional) Install drivers for x86 V5 BMSs.• (Optional) Install the UltraPath software.• Perform security configuration.• Configure remote login to the BMS.• Configure automatic root partition expansion.	<i>Bare Metal Server Image Creation Guide</i>

10.8.5 What Do I Do If a Windows Image File Is Not Pre-Configured When I Use It to Register a Private Image?

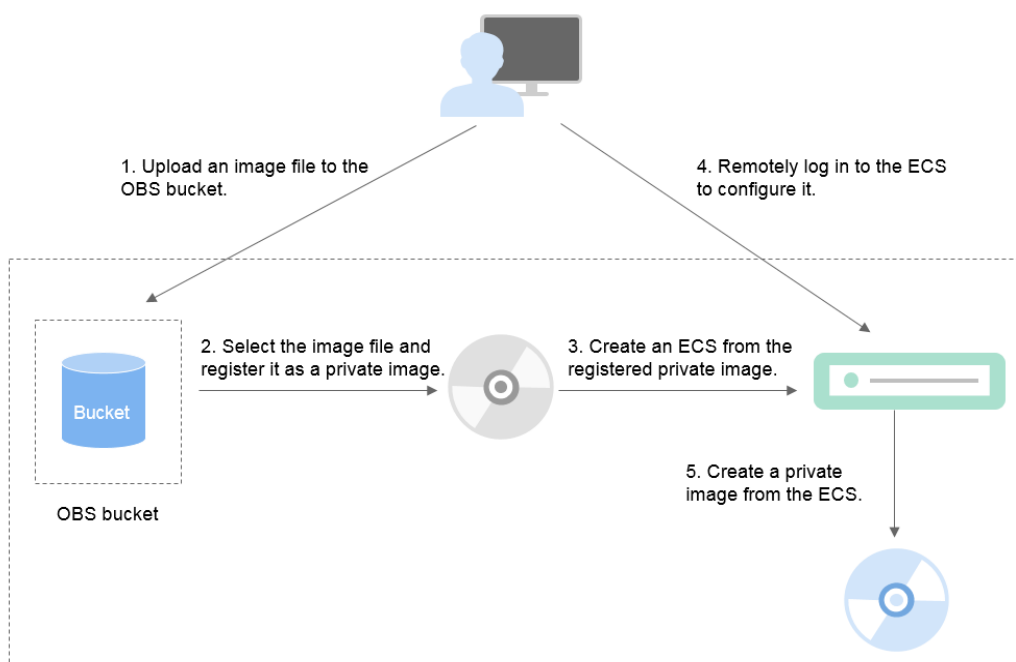
If an image file is not configured as instructed in [Table 3-5](#) before it is exported from the original platform, you can use it to create an ECS, configure the ECS, and use the ECS to create a private image. [Figure 10-7](#) shows the process.

CAUTION

An ECS can run properly only after Xen Guest OS drivers (PV drivers) and 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 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.

- To install PV drivers, see [Installing PV Drivers](#).
- To install VirtIO drivers, see [Installing VirtIO Drivers](#).

Figure 10-7 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 Server** in the **Operation** column.
4. Set parameters as promoted to create an ECS. Note that:
 - Bind an EIP to the ECS so that you can upload installation packages to the ECS or download them from the ECS.
 - You must add inbound rules for the ECS security groups 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 when 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. Check the ECS to see if the private image used to create the ECS has been pre-configured.
 - a. Check whether the ECS can be successfully started. If it can, 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 ECS cannot start up, install Guest OS drivers for the image file on the original platform and go back to **Step 1: Upload the Image File**.
 - 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 DHCP is configured by referring to **2** in **Step 4: Configure the ECS**.
 - d. Use MSTSC to log in to 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 DHCP is configured. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in **Configuring 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?](#)

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.

10.8.6 What Do I Do If a Linux Image File Is Not Pre-Configured When I Use It to Register a Private Image?

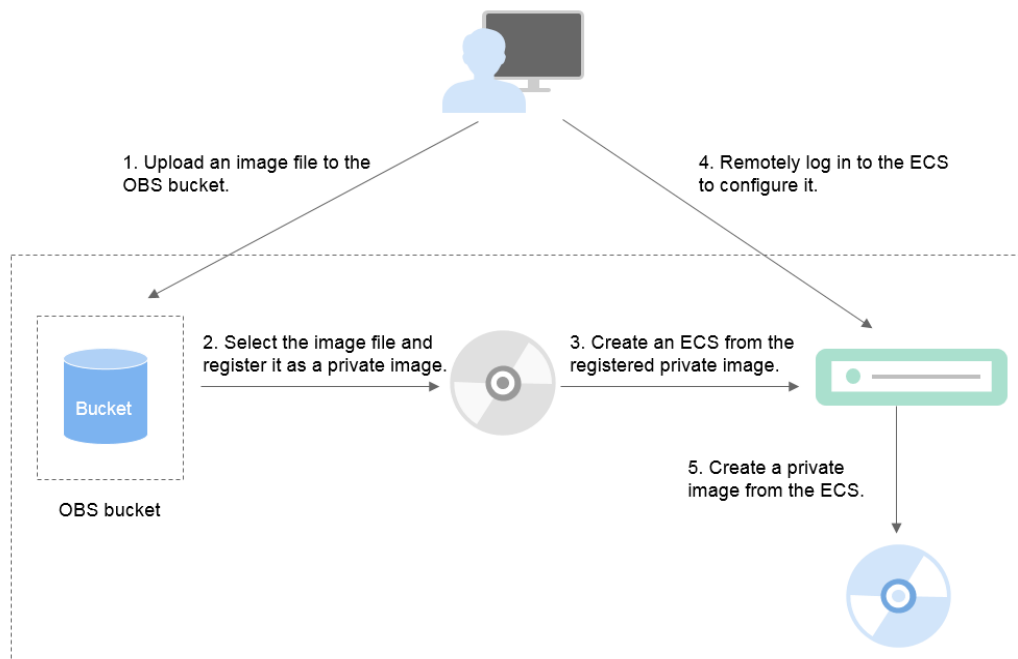
If an image file is not configured as instructed in [Table 3-9](#) before it is exported from the original platform, you can use it to create an ECS, configure the ECS, and use the ECS to create a private image. [Figure 10-8](#) shows the process.

CAUTION

An ECS can run properly only after Xen and 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 Xen, install native Xen and KVM drivers. For details, see [How Do I Install Native Xen and KVM Drivers?](#)
 - For KVM, install native KVM drivers. For details, see [Installing Native KVM Drivers](#).
-

Figure 10-8 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

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 Server** 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. Check the ECS to see if the private image used to create the ECS has been pre-configured.
 - a. Check whether the ECS can be successfully started. If the start succeeds, Xen and 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 Xen and 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 DHCP is configured. If the ECS is configured with a static IP address, change its IP address assignment mode to DHCP as instructed in [Configuring 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 disk identifiers in the GRUB file to UUID. For details, see [Changing Disk Identifiers in the GRUB File to UUID](#).
 - Change disk identifiers in the fstab file to UUID. For details, see [Changing Disk Identifiers in the fstab File to UUID](#).
 - Clear the automatic mount configuration of non-system disks in the **/etc/fstab** file. 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?](#)

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.

10.8.7 How Do I Enable NIC Multi-Queue for an Image?

Scenarios

Network I/O bandwidth can keep increasing to the point where a single vCPU cannot process all of the NIC interrupts. NIC multi-queue allows multiple vCPUs to process NIC interrupts, thereby improving network PPS and I/O performance.

ECSs Supporting NIC Multi-Queue

NIC multi-queue can only be enabled on an ECS with the specifications, image, and virtualization type 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 there are more than 1 NIC queue, NIC multi-queue is supported.

- Only KVM ECSs support NIC multi-queue.
- The Linux public images listed in [Table 10-10](#) support NIC multi-queue.

NOTE

- Windows public images have not supported NIC multi-queue. If you enable NIC multi-queue for a Windows public 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 10-10 KVM ECSs that support NIC multi-queue

OS	Image	Support for NIC Multi-Queue
Windows	Windows Server 2008 WEB R2 64bit	Yes (only supported by private images)
	Windows Server 2008 R2 Standard/Datacenter/Enterprise 64bit	Yes (only supported by private images)
	Windows Server 2012 R2 Standard/Datacenter 64bit	Yes (only supported by private images)
	Windows Server 2016 Standard/Datacenter 64bit	Yes (only supported by private images)
Linux	Ubuntu 14.04/16.04 Server 64bit	Yes
	openSUSE 42.2 64bit	Yes
	SUSE Enterprise 12 SP1/SP2 64bit	Yes
	CentOS 6.8/6.9/7.0/7.1/7.2/7.3/7.4/7.5/7.6 64bit	Yes
	Debian 8.0.0/8.8.0/8.9.0/9.0.0 64bit	Yes
	Fedora 24/25 64bit	Yes
	EulerOS 2.2 64bit	Yes

Operation Instructions

Assume that an ECS has the required specifications and virtualization type.

- If the ECS was created from a public image listed in [ECSs Supporting NIC Multi-Queue](#), NIC multi-queue is enabled on the ECS by default. You do not need to enable NIC multi-queue manually.
- If the ECS was created from an external image file with an OS listed in [ECSs Supporting NIC Multi-Queue](#), you may need to perform the following operations to enable NIC multi-queue:
 - a. [Register the External Image File as a Private Image](#).
 - b. [Enable NIC Multi-Queue for the Image](#).
 - c. [Create an ECS from the Private Image](#).
 - d. [Enable NIC Multi-Queue on the ECS](#).

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](#).

Enable NIC Multi-Queue for the Image

Windows has not commercially supported NIC multi-queue. If you enable NIC multi-queue for a Windows image, an ECS created from such an image may take longer than normal to start.

Use any of the following methods to enable NIC multi-queue for 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. On the displayed **Private Images** page, locate the row that contains the image and click **Modify** in the **Operation** column.
3. Enable 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 image.
3. In the upper right corner of the displayed image details page, click **Modify**. In the displayed **Modify Image** dialog box, enable 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.

Set **Content-Type** to `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
  }
]
```

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 on the ECS

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 interrupts 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 10-9** Enabled irqbalance

```
[root@localhost ~]# service irqbalance status
Redirecting to /bin/systemctl status irqbalance.service
● irqbalance.service - irqbalance daemon
 Loaded: loaded (/usr/lib/systemd/system/irqbalance.service; enabled; vendor preset: enabled)
 Active: active (running) since Wed 2018-08-15 10:27:30 CST; 4h 5min ago
 Main PID: 858 (irqbalance)
 CGroup: /system.slice/irqbalance.service
 └─858 /usr/sbin/irqbalance --foreground

Aug 15 10:27:30 localhost.localdomain systemd[1]: Started irqbalance daemon.
Aug 15 10:27:30 localhost.localdomain systemd[1]: Starting irqbalance daemon...
```



**Step 3** (Optional) Enable interrupt binding.

Enabling irqbalance allows the system to automatically allocate NIC interrupts, 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

## 10.8.8 How Do I Make a System Disk Image Support Fast ECS Creation?

### Scenarios

Fast Create greatly reduces the time required to create ECSs from a system disk image. Currently, this feature is supported for all newly created system disk images by default, but 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 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 Within a Region](#).
3. Wait for the replication to complete. Then, the new image can be used to quickly create ECSs.

## 10.8.9 Why Did 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.

## 10.8.10 How Do I Install Native Xen and 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 Xen and KVM drivers.

---

#### CAUTION

If an ECS has no Xen drivers installed, the network performance of the ECS will be poor, and the security groups and firewall configured for the ECS will not take effect.

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 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 [CentOS and EulerOS](#).

- Ubuntu and Debian

Modify the `/etc/initramfs-tools/modules` file. Add the Xen PV and VirtIO 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/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@CTU10000xxxx ~]# 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 **dracut -f *Name of the initramfs or initrd file actually used***. You can obtain the actual `initramfs` or `initrd` file name 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:

```
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:

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

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

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

```
[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_m0d
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 install the drivers. For details, see [scenario 1](#).

If the OS version is SUSE 12 SP1, modify the `/etc/dracut.conf` file to install the 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 install the drivers. For details, see [scenario 3](#).

- Earlier than SUSE 12 SP1 or openSUSE 13:

#### NOTE

Before installing the drivers, 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:

```
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 package 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).

```
SIA10000xxxx:~ # 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 `dracut -f` command to regenerate initrd.

#### NOTE

If the virtual file system is not the default `initramfs` or `initrd`, run `dracut -f Name of the initramfs or initrd file actually used`. 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 processor.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` is the `initrd` file actually used. If `/boot` is missing in the `initrd` file path, you need to add it when you run the `dracut -f` command. In this case, the command should be `dracut -f /boot/initramfs-xxx`.

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

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

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

```
SIA10000xxxx:~ # 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
```

```
SIA10000xxxx:~ # 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 to add `xen_platform_pci.dev_unplug=all` and change the root settings.

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 UUID format.
  - `xen_platform_pci.dev_unplug=all` is used 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
```

```
SIA10000xxxx:~ # 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
```

```
SIA10000xxxx:~ # 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 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
```



- SUSE 12 SP1:
  - 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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:  
**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:  
**lsinitrd /boot/initrd-`uname -r` | grep xen**  
**lsinitrd /boot/initrd-`uname -r` | grep virtio**
- Later than SUSE 12 SP1 or openSUSE 13:  
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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:  
**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:

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

Assume that the virtual file system is initrd. The command output will be:

```
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 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
```

## 10.9 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 details, see [Replicating Images Within a Region](#).

### 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.

### How Will I Be Billed for Replicating Images?

- 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.

## How Do I Replicate an Image Across Regions and Accounts?

Replicate the image to the target region and share it with the target account. Then, the image will be displayed in the shared image list of the target account.

## 10.10 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 ECSs does not affect the images.

### Can I Delete an Image I Shared with Others?

- If an image is shared with a project, you can delete the image without requiring any participation from the image recipients. After you delete the image, the image recipients cannot use it any longer. Inform the recipients to back up their data before you delete the image.

### How Do I Delete a Shared Image? Does the Deletion Affect Any ECS or EVS Disk Created from It?

Reject this image on the **Images Shared with Me** tab page. This does not affect any ECS or EVS disk created from it.

## 10.11 Image Encryption

### How Can I Change an Unencrypted Image to an Encrypted One?

You can select an encryption key when you replicate the image. Then, the system will generate an encrypted version of the unencrypted image.

## 10.12 Accounts and Permissions

### 10.12.1 What Do I Do If I Enabled EPS But Now I Cannot Find Private Images in My Enterprise Project?

#### 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.

### 10.12.2 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.

For details, see .

For details, see *Identity and Access Management User Guide*.

## 10.13 Cloud-Init

### 10.13.1 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, you run the following command to configure 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 10-10** Failed to enable Cloud-Init to start automatically

```
root@ecs-wjq-ubuntu14:~# systemctl enable cloud-init-local.service cloud-init.service 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 configure 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 10-11 Checking Cloud-Init status (1)

```
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 preset: enabled)
 Active: failed (Result: exit-code) since Fri 2018-08-17 07:12:20 UTC; 1min 25s ago
 Process: 4418 ExecStart=/usr/bin/cloud-init init --local (code=exited, status=203/EXEC)
 Main PID: 4418 (code=exited, status=203/EXEC)

Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Starting Initial cloud-init job (pre-networking)...
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Main process exited, code=exited, status=203/EXEC
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Failed to start Initial cloud-init job (pre-networking).
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Unit entered failed state.
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Failed with result 'exit-code'.
lines 1-11/11 (END)
```

Figure 10-12 Checking Cloud-Init status (2)

```
● cloud-init-local.service - Initial cloud-init job (pre-networking)
 Loaded: loaded (/lib/systemd/system/cloud-init-local.service; enabled; vendor preset: enabled)
 Active: failed (Result: exit-code) since Fri 2018-08-17 07:12:20 UTC; 59s ago
 Process: 4418 ExecStart=/usr/bin/cloud-init init --local (code=exited, status=203/EXEC)
 Main PID: 4418 (code=exited, status=203/EXEC)

Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Starting Initial cloud-init job (pre-networking)...
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Main process exited, code=exited, status=203/EXEC
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: Failed to start Initial cloud-init job (pre-networking).
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Unit entered failed state.
Aug 17 07:12:20 ecs-wjq-ubuntu14 systemd[1]: cloud-init-local.service: Failed with result 'exit-code'.
```

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 10-13 Restarting Cloud-Init

```
root@ecs-wjq-ubuntu14: # systemctl start cloud-init-local.service; systemctl sta
tus 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: active (exited) since Fri 2018-08-17 07:18:01 UTC; 4ms ago
 Process: 4491 ExecStart=/usr/bin/cloud-init init --local (code=exited, status=
 Main PID: 4491 (code=exited, status=0/SUCCESS)

Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] __init__.py[DEBUG
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: F
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] cloud-init[DEBUG]
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: R
Aug 17 07:18:01 ecs-wjq-ubuntu14 cloud-init[4491]: [CLOUDINIT] util.py[DEBUG]: c
lines 1-16/16 (END)
```

- 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  
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 Configure 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  
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 10-14 Information returned when Cloud-Init automatic start successfully set

```
root@ecs-debian-9:/tmp/CLOUD-INIT# cloud-init# cloud-init init --local
/usr/local/lib/python2.7/dist-packages/Cheetah-2.4.4-py2.7.egg/Cheetah/Compiler.py:1509: 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.
 "You don't have the C version of NameMapper installed!"
Cloud-Init v. 0.7.6 running 'init-local' at Mon, 20 Aug 2018 02:31:45 +0000. Up 704.40 seconds.
root@ecs-debian-9:/tmp/CLOUD-INIT# -cloud-init#
```

The compilation fails because GCC is not installed.

To solve this issue:

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 configure 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 10-15** Prompt indicating the failure to configure Cloud-Init automatic start

```
Failed to enable unit: Unit file /etc/systemd/system/cloud-init-local.service is masked.
```

To solve this issue:

- 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 10-16** Verifying the service status

```
■ cloud-init-local.service - Initial cloud-init job (pre-networking)
 Loaded: loaded (/lib/systemd/system/cloud-init-local.service; enabled; vendor preset: enabled)
 Active: active (exited) since Mon 2018-08-20 02:48:37 UTC; 6s ago
 Process: 1082 ExecStart=/usr/bin/cloud-init init --local (code=exited, status=0/SUCCESS)
 Main PID: 1082 (code=exited, status=0/SUCCESS)
 Tasks: 0 (limit: 4915)
 Group: /system.slice/cloud-init-local.service

Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Running command ['blkid', '-tLABEL=config-2', '-odevi
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] _init.py[DEBUG]: Seeing if we can get any data from <class 'cloudi
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Reading from /proc/mounts (quiet=False)
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Read 1947 bytes from /proc/mounts
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Fetched 'depts': {'mountpoint': '/dev/pts', 'opts':
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] cloud-init[DEBUG]: No local datasource found
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: Read 14 bytes from /proc/uptime
Aug 20 02:48:37 ecs-debian-9 cloud-init[1082]: [CLOUDINIT] util.py[DEBUG]: cloud-init mode 'init' took 0.104 seconds (0.10)
Aug 20 02:48:37 ecs-debian-9 systemd[1]: Started Initial cloud-init job (pre-networking).

■ cloud-init.service - Initial cloud-init job (metadata service crawler)
 Loaded: loaded (/lib/systemd/system/cloud-init.service; enabled; vendor preset: enabled)
 Active: active (exited) since Mon 2018-08-20 02:48:40 UTC; 3s ago
 Process: 1096 ExecStart=/usr/bin/cloud-init init (code=exited, status=0/SUCCESS)
 Main PID: 1096 (code=exited, status=0/SUCCESS)
 Tasks: 0 (limit: 4915)
 Group: /system.slice/cloud-init.service

Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] helpers.py[DEBUG]: config-ca-certs already ran (freq=once-per-instanc
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] stages.py[DEBUG]: Running module rsyslog (<module 'cloudinit.config.c
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] helpers.py[DEBUG]: config-rsyslog already ran (freq=once-per-instanc
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] stages.py[DEBUG]: Running module users-groups (<module 'cloudinit.con
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] cloud-init[DEBUG]: Ran 13 modules with 0 failures
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] util.py[DEBUG]: Reading from /proc/uptime (quiet=False)
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] util.py[DEBUG]: Read 14 bytes from /proc/uptime
Aug 20 02:48:40 ecs-debian-9 cloud-init[1096]: [CLOUDINIT] util.py[DEBUG]: cloud-init mode 'init' took 2.657 seconds (2.66)
Aug 20 02:48:40 ecs-debian-9 systemd[1]: Started Initial cloud-init job (metadata service crawler).

■ cloud-config.service - Apply the settings specified in cloud-config
 Loaded: loaded (/lib/systemd/system/cloud-config.service; enabled; vendor preset: enabled)
 Active: active (exited) since Mon 2018-08-20 02:48:41 UTC; 2s ago
 Process: 1140 ExecStart=/usr/bin/cloud-init modules --mode=config (code=exited, status=0/SUCCESS)
 Main PID: 1140 (code=exited, status=0/SUCCESS)
 Tasks: 0 (limit: 4915)
 Group: /system.slice/cloud-config.service
```

## CentOS 7/Fedora 28: Required C Compiler Not Installed

- Symptom

After Cloud-Init is successfully installed, you 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

You cannot use a new password to log in to an ECS created from an image with Cloud-Init installed. After logging in to the ECS using the old password, you find that NICs of the ECS are not started.

**Figure 10-17** NIC not started

```
root@ecs-fedora28-wjq-test ~]# ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
 inet 127.0.0.1 netmask 255.0.0.0
 inet6 ::1 prefixlen 128 scopeid 0x10<host>
 loop txqueuelen 1000 (Local Loopback)
 RX packets 0 bytes 0 (0.0 B)
 RX errors 0 dropped 0 overruns 0 frame 0
 TX packets 0 bytes 0 (0.0 B)
 TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Solution

Log in to the ECS used to create that image, open the DHCP configuration file `/etc/sysconfig/network-scripts/ifcfg-ethX`, and comment out `HWADDR`.

## 10.13.2 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 customize 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 these custom configurations to your ECSs, and 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 this private image can be customized.

- For Windows, download and install Cloudbase-Init.  
For details, see [Installing and Configuring Cloudbase-Init](#).
- For Linux, 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](#).

### 10.13.3 What Do I Do If Installed NetworkManager and Now I Can't Inject the Key or Password Using Cloud-Init?

#### Cause

One likely possibility 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](#).

### 10.13.4 How Do I Install growpart for SUSE 11 SP4?

#### Scenarios

For SUSE and openSUSE, growpart is an independent tool and is not included in a **cloud-\*** package. You need to install it separately.

#### Procedure

1. Check whether Cloud-Init and growpart are installed.

**rpm -qa | grep cloud-init**

If cloud-init is installed, the command output should be similar to the following:

```
cloud-init-0.7.8-39.2
```

**rpm -qa | grep growpart**

If growpart is installed, the command output should be similar to the following:

```
growpart-0.29-8.1
```

2. If they are installed, uninstall them.  
**zypper remove cloud-init growpart**
3. Delete residual files.  
**rm -fr /etc/cloud/\***  
**rm -fr /var/lib/cloud/\***
4. 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. 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. 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. Check whether growpart, python-oauth, and Cloud-Init are installed successfully.  
**rpm -qa | grep growpart**  
If growpart is installed, the command output should be similar to the following:  

```
growpart-0.27-1.1
```

**rpm -qa | grep python-oauth**  
If python-oauth is installed, the command output should be similar to the following:  

```
python-oauthlib-0.6.0-1.5
python-oauth-1.0.1-35.1
```

**rpm -qa | grep cloud-init**  
If Cloud-Init is installed, the command output should be similar to the following:  

```
cloud-init-0.7.6-27.19.1
```
8. Check the configurations.  
**chkconfig cloud-init-local on;chkconfig cloud-init on;chkconfig cloud-config on;chkconfig cloud-final on**

## 10.14 ECS Creation

### 10.14.1 Can I Change the Image of a Purchased ECS?

Yes.

If you have selected the wrong 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*.

## 10.14.2 Can I Change the Specifications Defined by a Private Image When I Use the Image to Create an ECS?

Yes. You can specify the CPU, memory, bandwidth, system and data disks of the ECS you are creating. The system disk must be smaller than 1,024 GB but no less than the system disk capacity 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.

For details, see [Creating an ECS from an Image](#).

## 10.14.3 Can I Specify the System Disk Capacity When I Create an ECS Using an 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.

For details, see [Creating an ECS from an Image](#).

Yes, but you cannot create a system disk smaller than the original and the maximum allowed is 30,768 GB.

### NOTE

Ensure that your ECS OS can support the system disk size you specified.

## 10.14.4 What Do I Do If a Partition Is Not Found During the Startup of an ECS Created from an Imported Private Image?

### Cause

Disk partition IDs are changed after the cross-platform image import. As a result, no partition can be found at startup. In this case, you need to change disk partition IDs in the image to UUID.

### Solution

openSUSE 13.2 is used as an example.

1. Check disk partition IDs.

**ls -l /dev/disk/by-id/**

An output similar to the following should be seen:

```
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 ECS disks are simulated using Quick EMUlator (QEMU). The content on the left of -> are disk partition IDs, and that on the right of -> are partition names.

2. Check disk partition UUIDs.

**ls -l /dev/disk/by-uuid/**

An output similar to the following should be seen:

```
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 -> are disk partition UUIDs, and that on the right of -> are partition names. Based on the outputs in [1](#) and this step, you can obtain the mappings between the partition names, IDs, and UUIDs.

3. Open the **/etc/fstab** file to check partition names.

**vi /etc/fstab**

An 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 each disk partition ID to a UUID based on the outputs in [1](#) and [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, or the ECS will be unable to start up normally.

5. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the editor.
6. Check the partition names in the system boot configuration file.  
The system boot configuration file varies depending on the OS. Confirm the boot configuration file of the current OS.

- GRUB 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 each 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
```

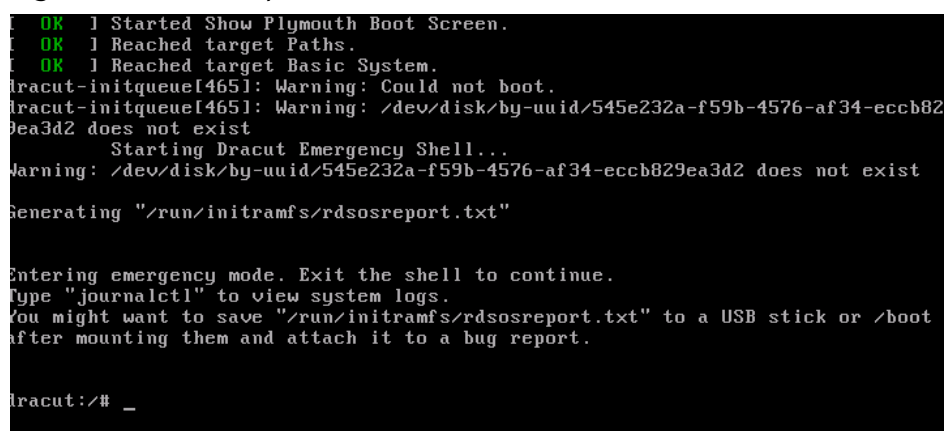
8. Press **Esc**, enter **:wq**, and press **Enter**. The system saves the configuration and exits the vi editor.

## 10.14.5 What Do I Do If the Disks of a CentOS ECS Created from an Image Cannot Be Found?

### Symptom

When you started a CentOS ECS, the system cannot find disks. Generally, this is because the `xen-blkfront.ko` module was not loaded during the startup. You need to modify OS kernel startup parameters.

Figure 10-18 Startup screen



```
OK] Started Show Plymouth Boot Screen.
OK] Reached target Paths.
OK] Reached target Basic System.
dracut-initqueue[465]: Warning: Could not boot.
dracut-initqueue[465]: Warning: /dev/disk/by-uuid/545e232a-f59b-4576-af34-eccb829ea3d2 does not exist
Starting Dracut Emergency Shell...
Warning: /dev/disk/by-uuid/545e232a-f59b-4576-af34-eccb829ea3d2 does not exist
Generating "/run/initramfs/rdsosreport.txt"

Entering emergency mode. Exit the shell to continue.
Type "journalctl" to view system logs.
You might want to save "/run/initramfs/rdsosreport.txt" to a USB stick or /boot
after mounting them and attach it to a bug report.

dracut:/# _
```

### Solution

Modify OS kernel boot parameters.

#### NOTE

These operations can only be performed after a normal OS startup. So, perform them in the source ECS of the image instead of the current ECS.

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 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 file (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, and then use the image to create a new ECS.

## 10.14.6 What Do I Do If I Enabled Automatic Configuration During Image Registration for an ECS Created from a Windows Image and Now It Won't Start?

### Cause

This may be caused by an issue with offline VirtIO driver injection.

### Solution

When you inject VirtIO drivers into the image for a Windows ECS, note that:

- If the boot mode in the image file is UEFI, the VirtIO drivers cannot be injected offline.
- Disable Group Policy Object (GPO) because some policies may cause offline VirtIO driver injection to fail.
- Stop any installed antivirus software. They may cause offline VirtIO driver injection to fail.

To install VirtIO drivers, see [Optimizing a Windows Private Image](#).

## 10.14.7 What Do I Do If an Exception Occurs When I Start an ECS Created from an Image Using UEFI Boot?

### Symptom

An ECS created from a private image booting to UEFI cannot start.

## Possible Causes

The image is configured for UEFI boot, but the uefi attribute was 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.

# 10.15 Driver Installation

## 10.15.1 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 PV and VirtIO drivers on ECSs.
- Linux ECSs: Install Xen PV and VirtIO drivers and add them to initrd.

## 10.15.2 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.

- Windows does not have VirtIO drivers installed by default.
- 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 by the community can be eliminated from drivers as soon as possible.

### What Do I Need to Do?

- Upgrade VirtIO drivers in Windows private images or running Windows ECSs.
- If you have any technical issues or questions, contact the customer service.



### 10.15.3 Why Did 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.

### 10.15.4 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](#).

### 10.15.5 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](#).

### 10.15.6 How Do I Install Native Xen and 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 Xen and KVM drivers.

---

#### CAUTION

If an ECS has no Xen drivers installed, the network performance of the ECS will be poor, and the security groups and firewall configured for the ECS will not take effect.

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 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 [CentOS and EulerOS](#).
- Ubuntu and Debian  
Modify the `/etc/initramfs-tools/modules` file. Add the Xen PV and VirtIO 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/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@CTU10000xxxx ~]# 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 **dracut -f *Name of the initramfs or initrd file actually used***. You can obtain the actual initramfs or initrd file name 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is initramfs, run the following commands:

```
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:

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

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

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

```
[root@CTU10000xxxx 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@CTU10000xxxx 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@CTU10000xxxx ~]#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@CTU10000xxxx 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@CTU10000xxxx 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 install the drivers. For details, see [scenario 1](#).

If the OS version is SUSE 12 SP1, modify the `/etc/dracut.conf` file to install the 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 install the drivers. For details, see [scenario 3](#).

- Earlier than SUSE 12 SP1 or openSUSE 13:

 NOTE

Before installing the drivers, 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:

```
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 package 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 `dracut -f` command to regenerate `initrd`.

 NOTE

If the virtual file system is not the default `initramfs` or `initrd`, run `dracut -f Name of the initramfs or initrd file actually used`. 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 processor.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** is the **initrd** file actually used. If **/boot** is missing in the **initrd** file path, you need to add it when you run the **dracut -f** command. In this case, the command should be **dracut -f /boot/initramfs-XXX**.

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

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

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

```

SIA10000xxxx:~ # 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

```

```

SIA10000xxxx:~ # 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 to add **xen\_platform\_pci.dev\_unplug=all** and change the root settings.

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 UUID format.
- **xen\_platform\_pci.dev\_unplug=all** is used 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
```

```
SIA10000xxxx:~ # 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

SIA10000xxxx:~ # 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 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
```

- SUSE 12 SP1:

- 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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:  
**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:  
**lsinitrd /boot/initrd-`uname -r` | grep xen**  
**lsinitrd /boot/initrd-`uname -r` | grep virtio**
  - Later than SUSE 12 SP1 or openSUSE 13:  
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@CTU10000xxxx ~]# 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. Check whether native Xen and KVM drivers have been installed. If the virtual file system is `initramfs`, run the following commands:  
**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:  
**lsinitrd /boot/initrd-`uname -r` | grep xen**  
**lsinitrd /boot/initrd-`uname -r` | grep virtio**  
Assume that the virtual file system is `initrd`. The command output will be:

```
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 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
```

## 10.16 Image Tags

### 10.16.1 How Many Tags Can I Add to an Image?

An image can have a maximum of 10 tags.

Each tag consists of a key and a value. The key contains a maximum of 36 characters, and the value contains a maximum of 43 characters. The key cannot be left blank or an empty string. The value cannot be left blank but can be an empty string.

For details, see [Managing Tags](#).

### 10.16.2 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.

### 10.16.3 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.