Image Management Service

Best Practices

Issue   01

Date   2024-11-07

Security Declaration

Vulnerability

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

Contents

1 Overview

Common IMS practices are provided in this document to help you easily use IMS.

Table 1-1 IMS practices

Practice	Description
Creating a Linux Image Using VirtualBox and an ISO File	Describes how to create a Linux image using VirtualBox. To do so, you need to install VirtualBox, use it to create a VM from an ISO file, configure the VM, and generate a VHD image using the configured VM.
Cleaning Up the Disk Space of a Windows ECS	Describes how to clean up the disk space of a Windows ECS.
Converting the Image Format	Describes how to use qemu-img or Huawei qemu-img-hw to convert the image format. qemu-img can be used for mutual conversion of formats VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED but cannot be used for the format conversion to ZVHD or ZVHD2. To convert an image file to any of the two formats, use qemu-img-hw.
Creating a Private Image Using Packer	Describes how to create an Ubuntu 22.04 Server 64-bit private image from a CentOS 8.2 ECS using Packer and upload it to the cloud platform.
Configuring an ISO File as a Local Image Source	Describes how to configure a local image source by using the yum, apt, and zypper package managers and provides configuration examples of Debian 10.1.0 and CentOS 8.0.

Practice	Description
Migrating Service Data Across Accounts (Data Disks)	Generally, service data is stored on data disks. To migrate service data across accounts, you need to create data disk images and share them with the target account.

2 Creating a Linux Image Using VirtualBox and an ISO File

2.1 Introduction

VirtualBox

VirtualBox is free, open-source virtualization software. It was first offered by InnoTek GmbH from Germany and re-branded as Oracle VM VirtualBox when InnoTek was acquired by Oracle Corporation.

For more information about VirtualBox, visit the Oracle official website. Click **here** to see the guest OSs that can work with VirtualBox.

Scenarios

You can use a 32-bit or 64-bit Linux guest OS provided by VirtualBox to create an image file in VHD format.

Advantages

You can customize Linux image files.

Tools and Costs

Table 2-1 Tools and costs

Tool	Description	Costs
VirtualBox	VirtualBox is free, open-source virtualization software. Download it from: **https://www.virtualbox.org/wiki/Downloads**	Free

Tool	Description	Costs
ISO image file	It is used to install an OS on an empty VM. Download it from **https://ubuntu.com/download/server#downloads** For example, the file name is **Ubuntu-20.04-server.iso**.	Free
Cloud-Init	To ensure that you can inject initial custom information into ECSs created from the image, for example, setting the login password, you are advised to install Cloud-Init on the VM which is used as the image source. The method for obtaining Cloud-Init varies depending on the OS. For details, see **Installing Cloud-Init**.	Free
(Optional) One-click password reset plug-in	To ensure that you can reset the password of each ECS created from the image with a few clicks, you are advised to install the one-click password reset plug-in (CloudResetPwdAgent) on the VM which is used as the image source.	Free

Image Creation Process

The following figure shows how to use VirtualBox to create an image from an ISO file.

Figure 2-1 Image creation process
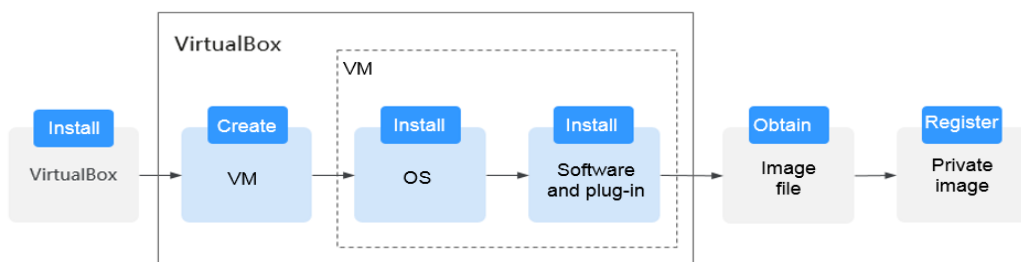
1. Install VirtualBox: Prepare a host (64-bit Windows is recommended) and install VirtualBox on the host. For details, see **Installing VirtualBox**.

2. Create a VM: Use VirtualBox to create an empty VM from which an image will be created. For details, see **Creating an Empty VM**.

3. Install an OS on the VM: Mount an ISO file to install an OS for the VM. The ISO file determines the OS of the image you want to create. For details, see **Installing a Linux OS on the VM**.

4. Install software and plug-ins on the VM: To ensure the proper running of ECSs that will be created from the image, install the required software and plug-ins on the VM, including native Xen and KVM drivers, Cloud-Init, and one-click password reset plug-in. For details, see **Configuring the VM**.

5. Obtain the image file: Export an image file in VHD format from VirtualBox. For details, see **Obtaining the Image File**.

6. Register a private image: Upload the exported VHD image file to an OBS bucket and register it as a private image. Then, you can use the private image to create ECSs. For details, see **Registering the Image File as a Private Image**.

2.2 Installing VirtualBox

Preparations

The host where VirtualBox is to be installed must meet the following requirements:

- A 64-bit Windows OS (recommended).

- At least 4 GB of memory and a dual-core processor. For example, the host specifications can be 8U16G.

- At least 20 GB of available disk space.

- Hardware virtualization (Intel VT-x or AMD-V). For how to enable this, see **Host CPU Settings (Hardware Virtualization)**.

  ◫ NOTE

  For details about how to install VirtualBox, see the VirtualBox user guide at **https://www.virtualbox.org/manual/UserManual.html**.

Host CPU Settings (Hardware Virtualization)

For an Intel host, perform the following operations to enable hardware virtualization:

◫ NOTE

The operations may differ depending on the CPU type. You can do it as prompted.

1. During the host startup, press the BIOS key set by the manufacturer to access the BIOS.

2. Choose **Configuration** > **Intel Virtual Technology**, and press **Enter**.

3. Select **Enabled** and press **Enter**. The value of **Intel Virtual Technology** will become **Enabled**.

4. Press **F10** to save the settings and exit.

Figure 2-2 Enabling hardware virtualization
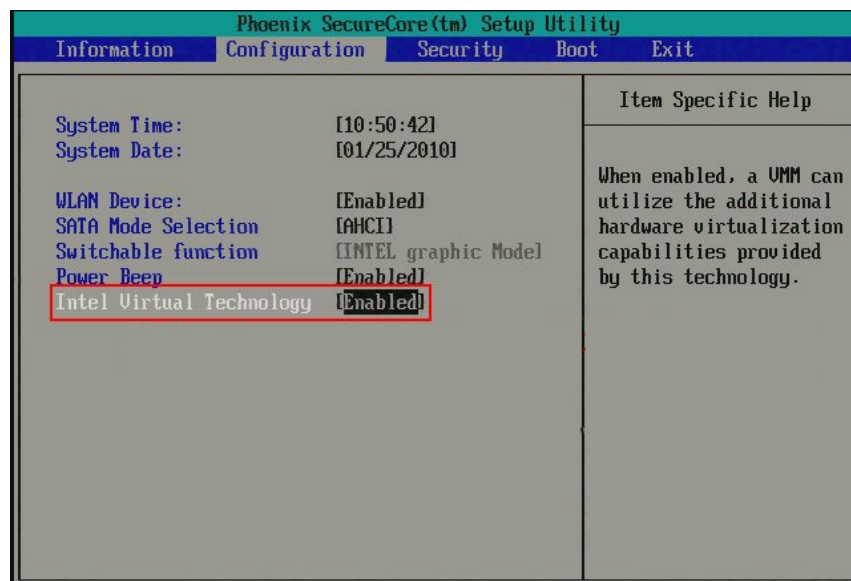
Procedure

1. Download the VirtualBox installation package. VirtualBox-5.2.0 is used as an example.

   Download it from **https://www.virtualbox.org/wiki/Downloads**.

2. Decompress the package. Right-click **VirtualBox-5.2.0-118431-Win.exe**, choose **Run as administrator**, and click **Next**.

Figure 2-3 Installing VirtualBox

3. Select the VirtualBox installation path and click **Next**.

Figure 2-4 Selecting an installation path
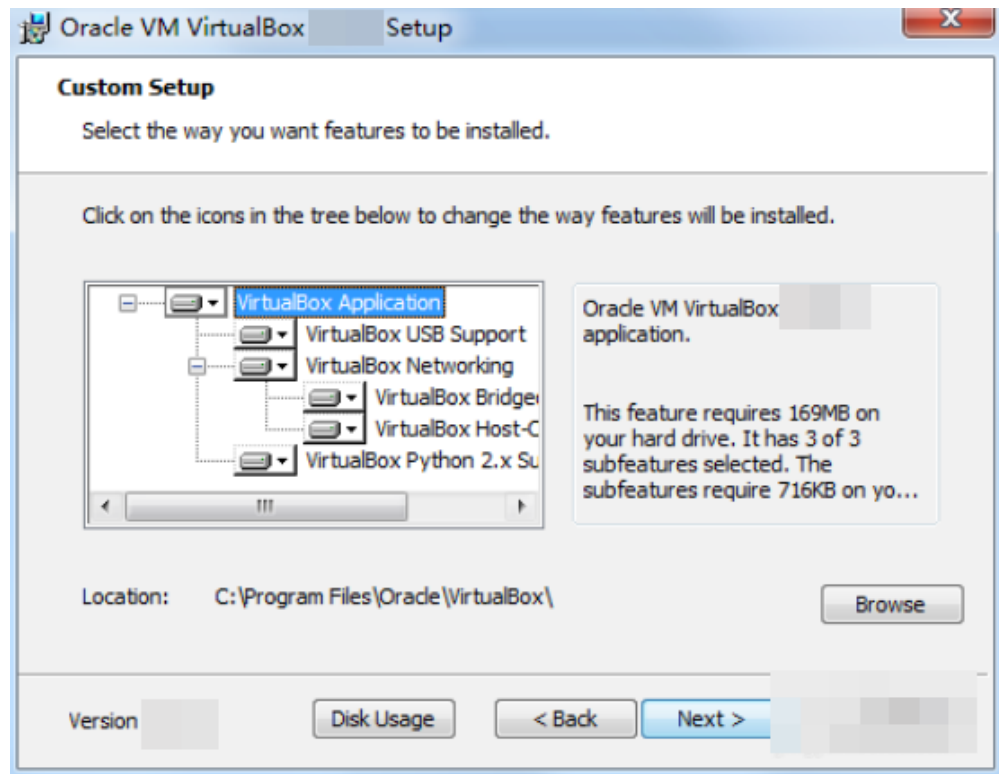
4. Personalize the settings and click **Next**.

Figure 2-5 Personalized settings
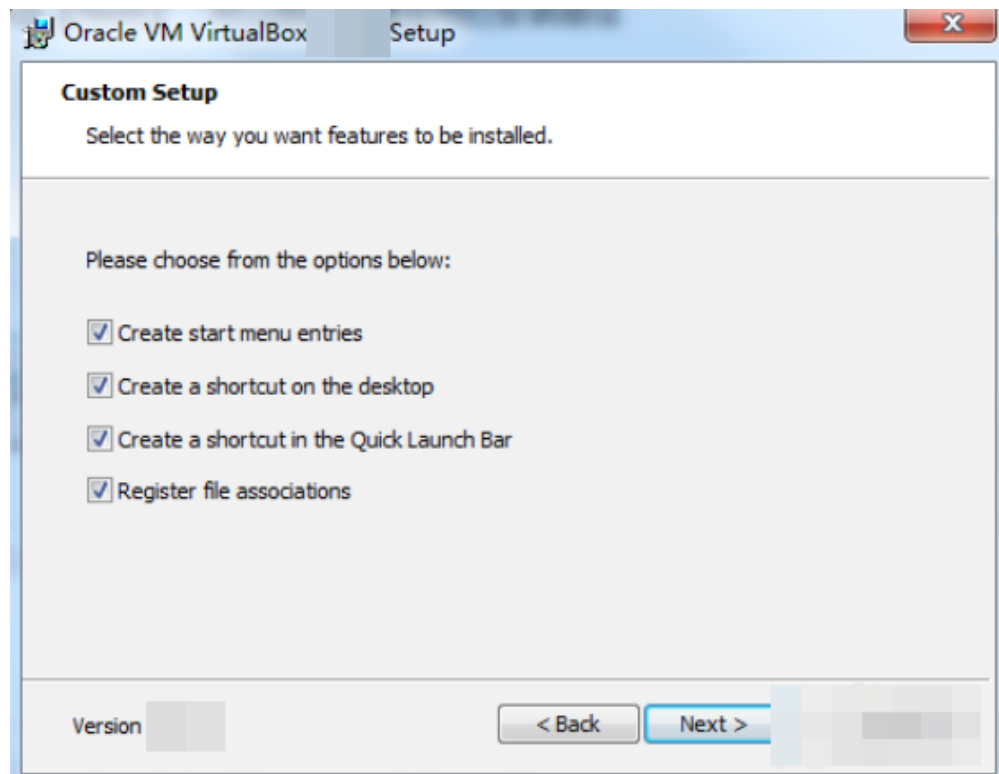
5.   Click **Finish**.

2.3 Creating a VM and Installing an OS

2.3.1 Creating an Empty VM

Prerequisites

VirtualBox has been installed.

Procedure

1.   Open VirtualBox and click **New**. In the displayed **Create Virtual Machine** dialog box, enter a VM name, select an OS type and version, and click **Next**.

     Take Ubuntu as an example. The type must be **Linux**.

     Ensure that the selected version is the same as that of the OS you want to install on the VM.

     Figure 2-6 Creating a VM
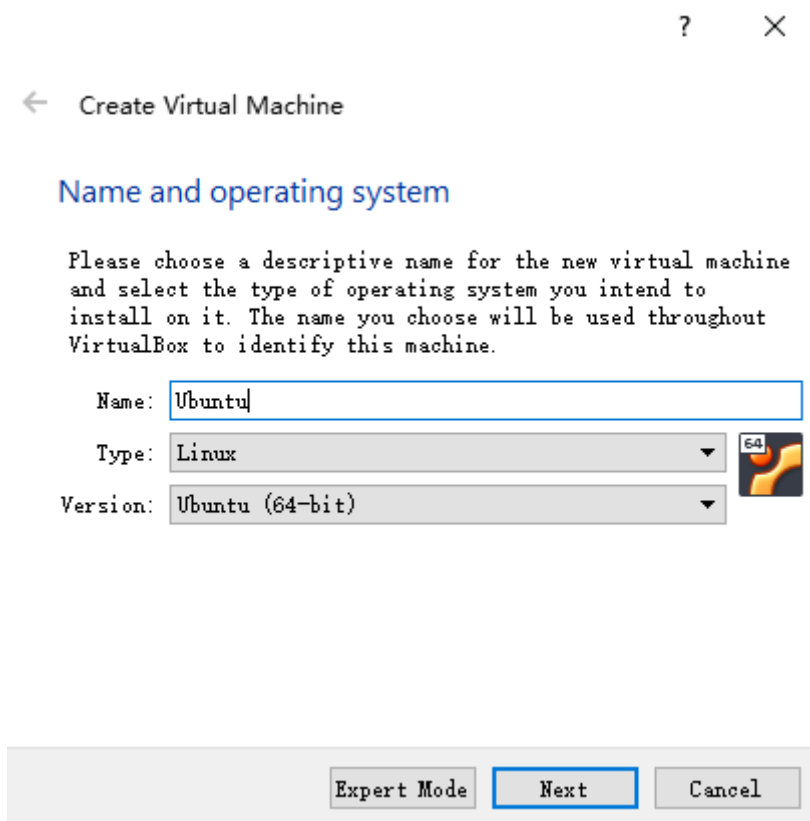
2.   In the **Memory size** dialog box, set a value and click **Next**.

     You can reference the VM specifications or official OS requirements. The minimum value is 256 MB. You can set the memory size to 512 MB as an example.

Figure 2-7 Setting the memory size
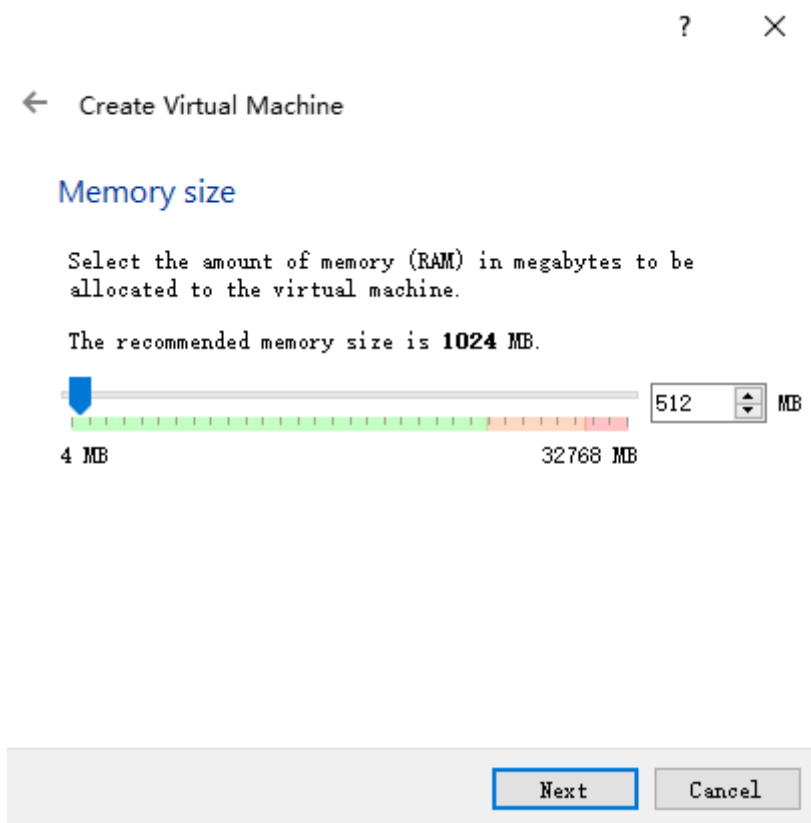
3. In the **Hard disk** dialog box, select **Create a virtual hard disk now** and click **Create**.

Figure 2-8 Creating a virtual hard disk
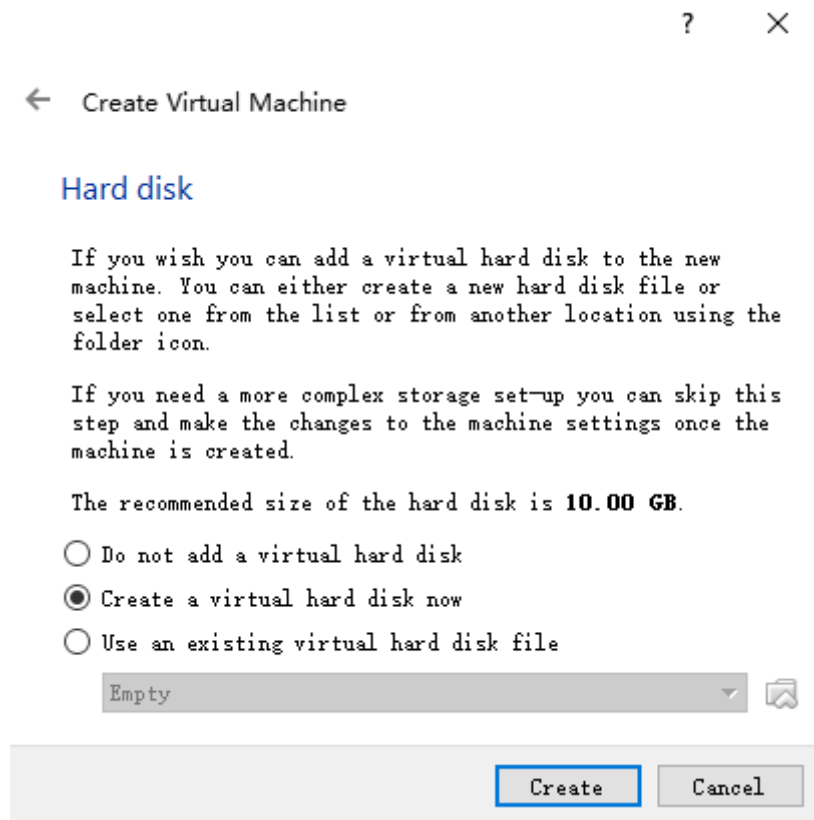
4. In the **Hard disk file type** dialog box, select **VHD** and click **Next**.

Figure 2-9 Setting the hard disk file type
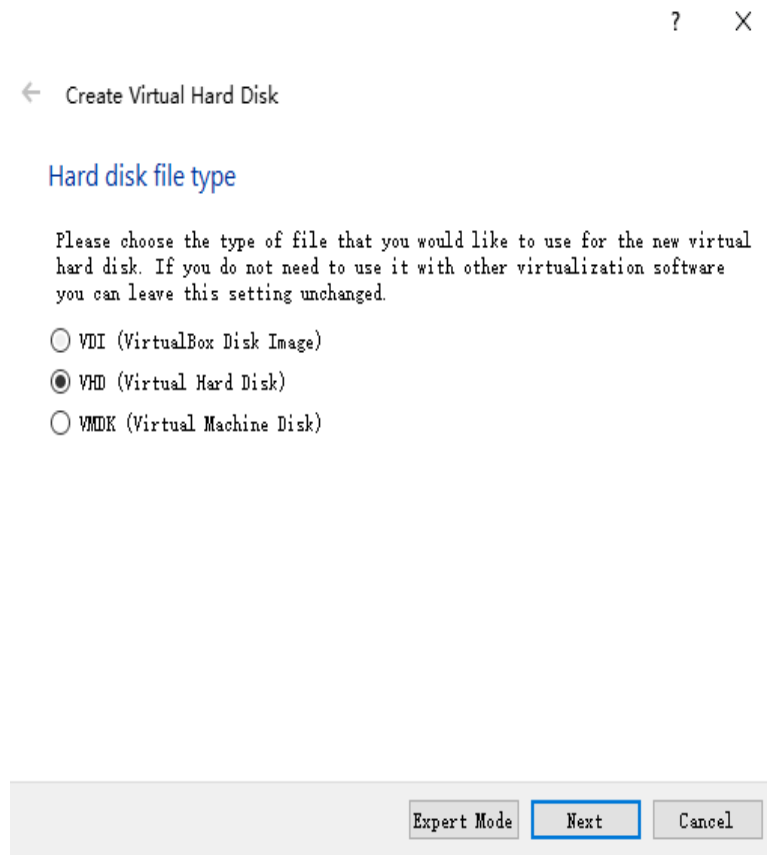
5. In the **Storage on physical hard disk** dialog box, select **Dynamically allocated** and click **Next**.

Figure 2-10 Selecting the disk allocation mode
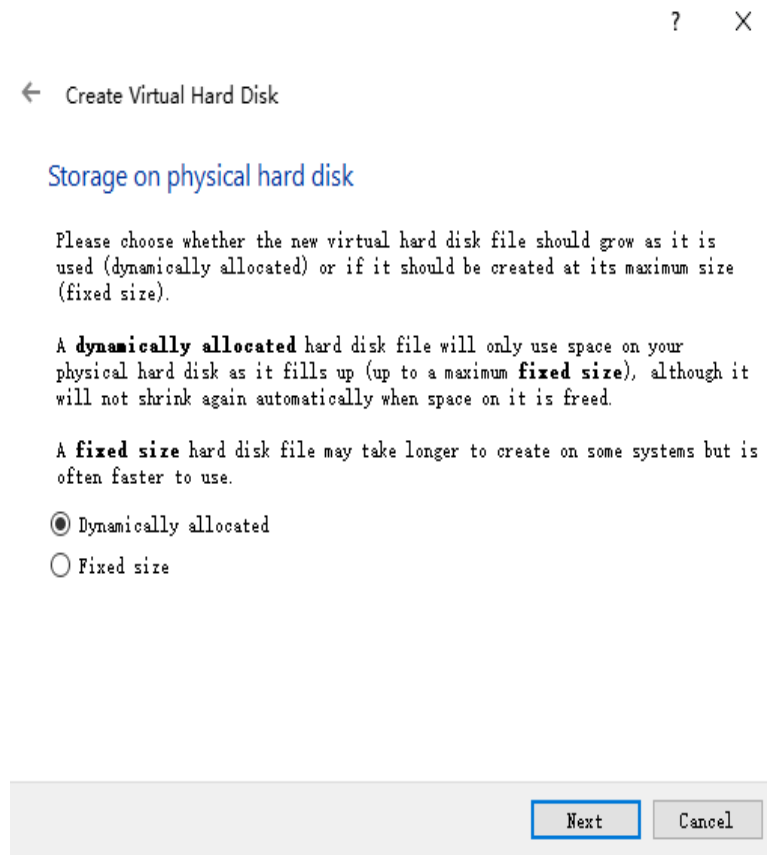
6. In the **File location and size** dialog box, set the disk size and storage location.

   For example, you can set the disk size to 20 GB.

Figure 2-11 Setting the disk location and size
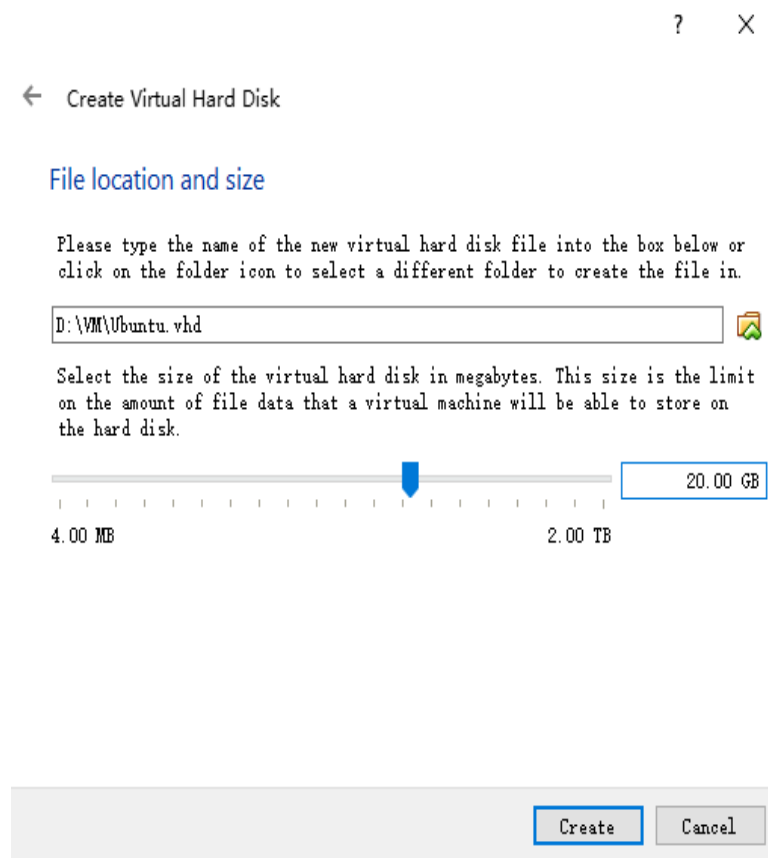
7.   Click **Create**.

2.3.2 Installing a Linux OS on the VM

The procedure varies depending on the image file you use. This section uses
Ubuntu 20.04 as an example to describe how to install a Linux OS on the VM.

Prerequisites

You have obtained the ISO image file, for example, **Ubuntu-20.04-server.iso**.

Procedure

Use the ISO file to install Linux for the empty VM.

1.   In VirtualBox Manager, select the new VM and click **Settings**.

Figure 2-12 Setting the VM
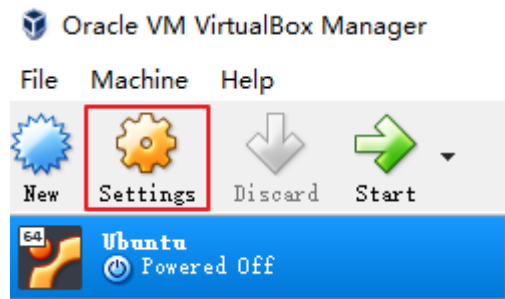
2. Choose **Storage** > **Empty**, click in the **Attributes** area, and select the ISO image file **Ubuntu-20.04-server.iso**.

Figure 2-13 Selecting the ISO file to be mounted
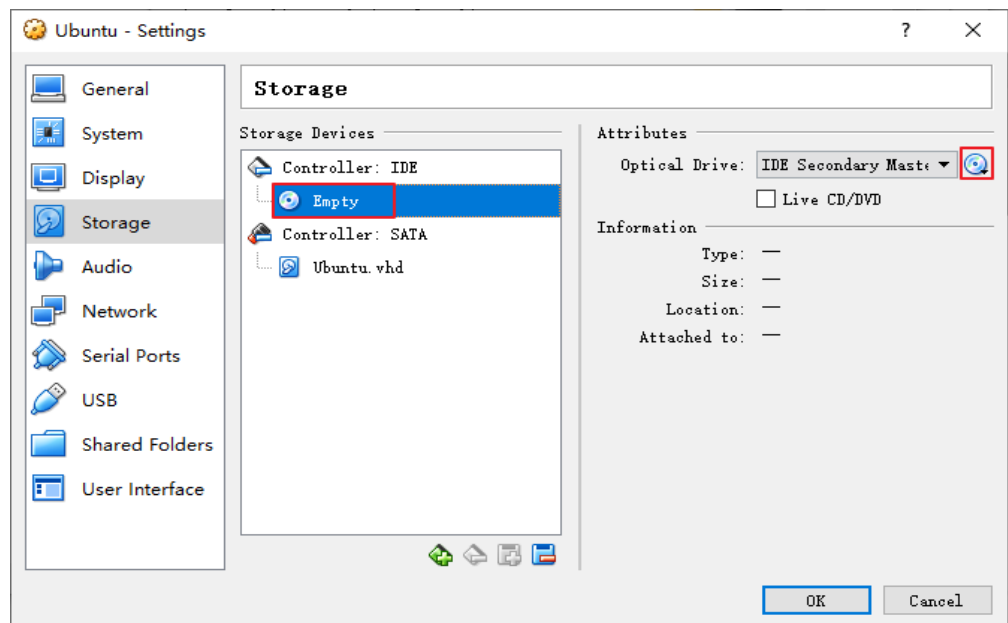
Figure 2-14 Mounted ISO file
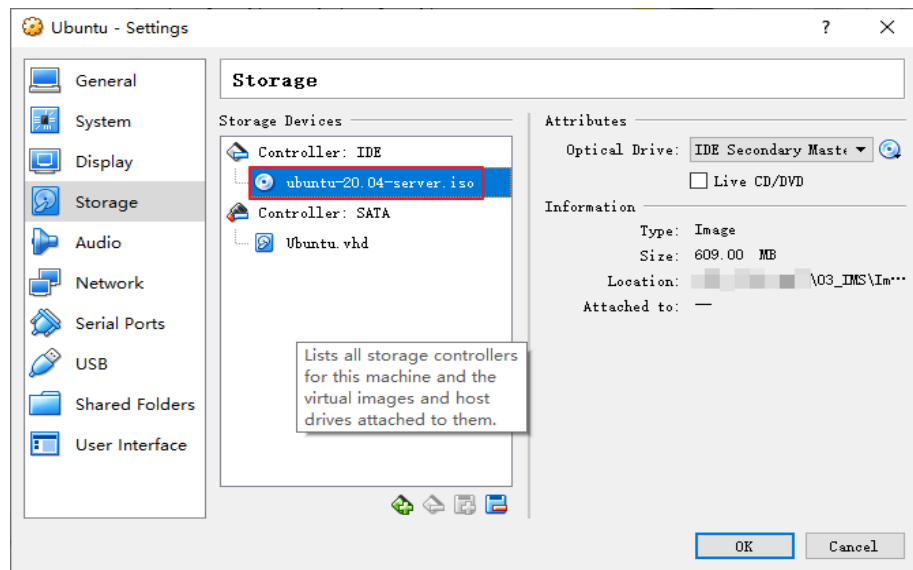
3. Click **OK**.

4. In VirtualBox Manager, select the new VM and click **Start**.

Figure 2-15 Starting the VM
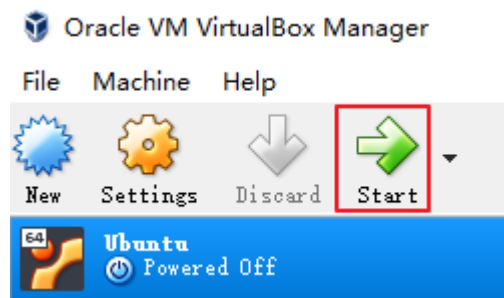
5. Install the OS.

   a. Select **English** and press **Enter**.
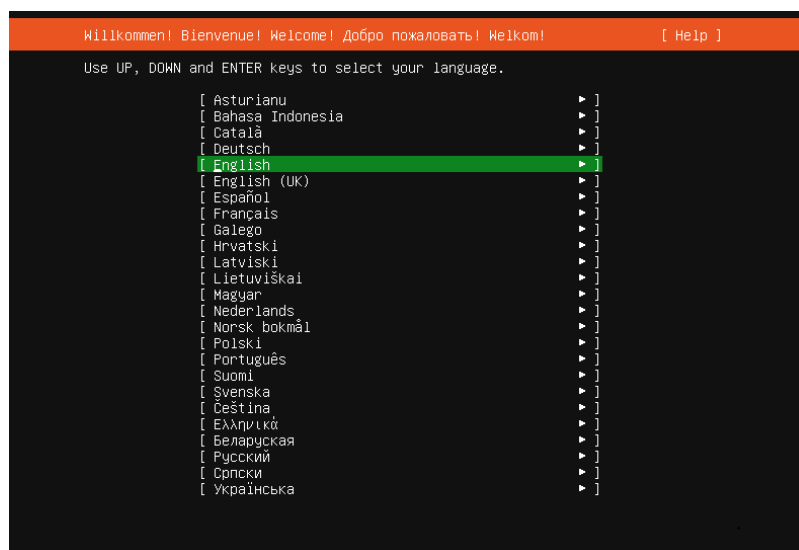
b. Select **Continue without updating**.
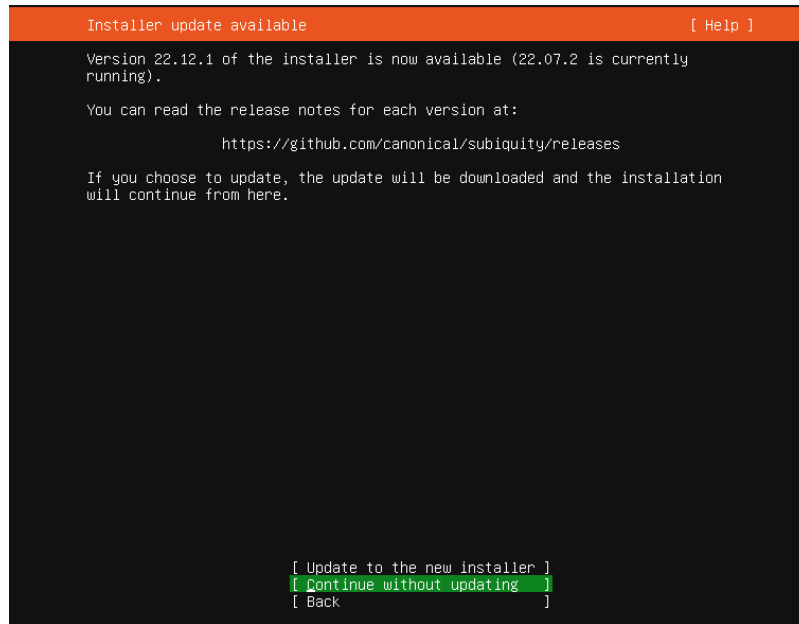
c. Retain the default settings for the keyboard. Select **Done**
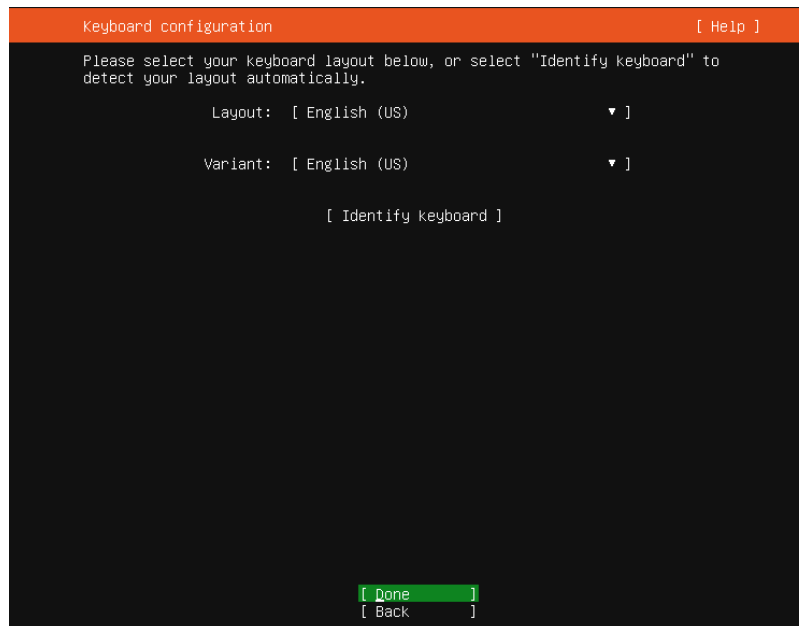
d. Retain the default settings for the installation base. Select **Done**.
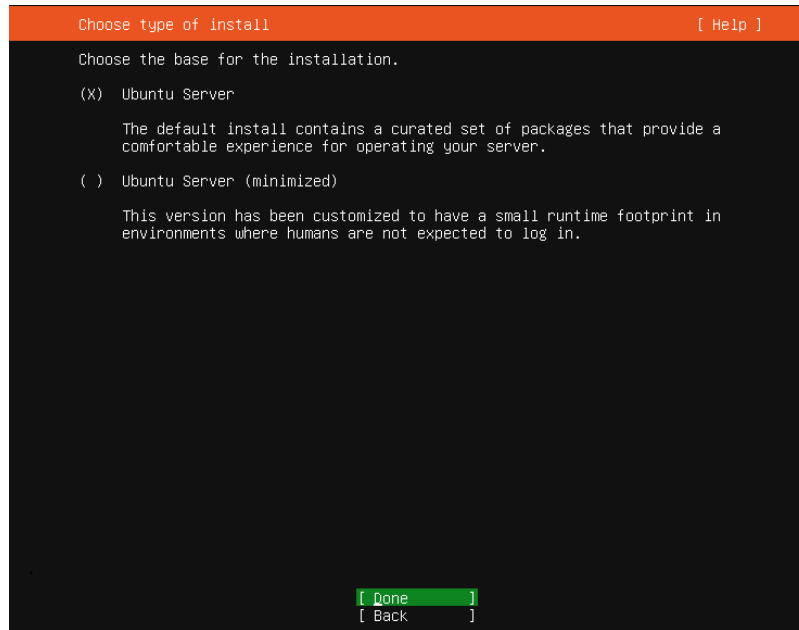
e. Retain the default settings for the network. Select **Done**.

The installation program will automatically identify the IP address. If the
network cannot be found, the installation program can still continue and
you can configure the network again after the installation is complete.
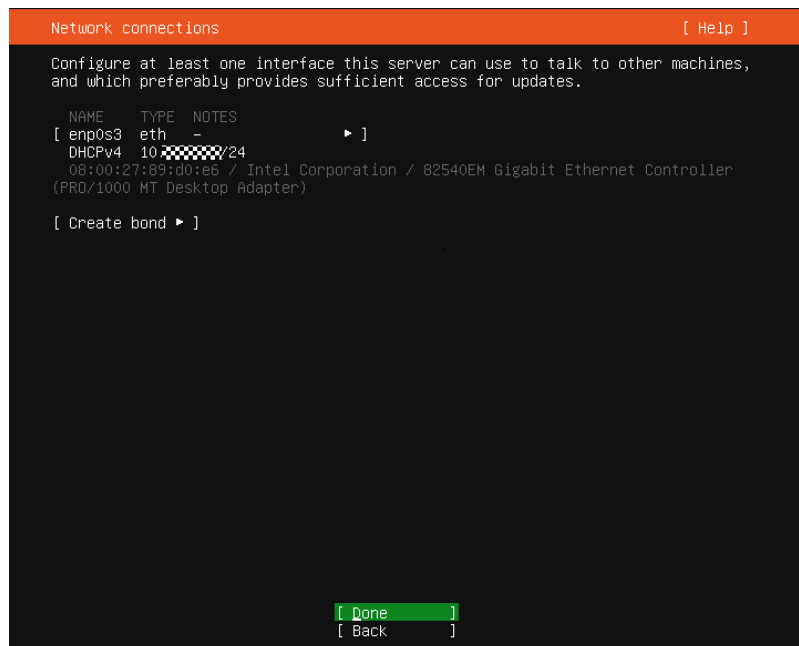
f. Retain the default settings for the proxy. Select **Done**.
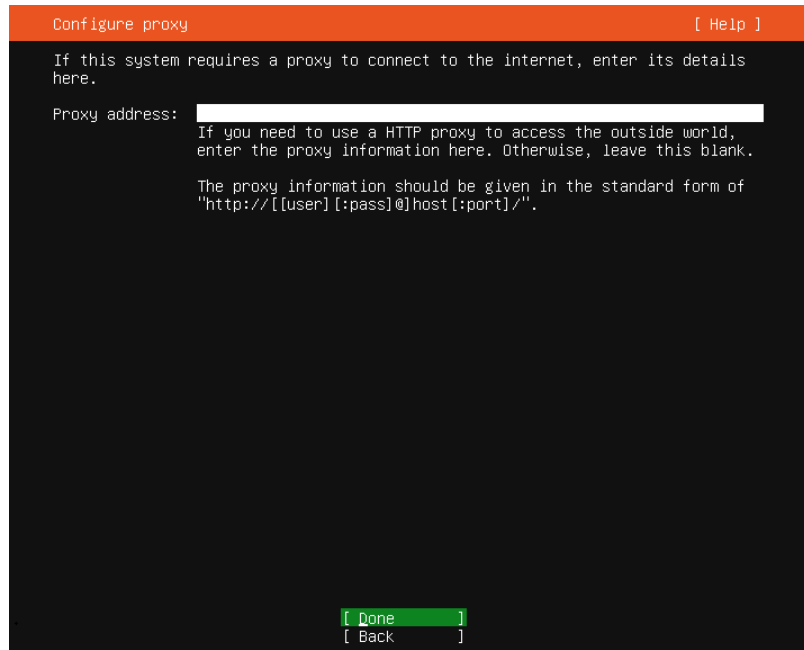
g.   Retain the default settings for the software source. Select **Done**.
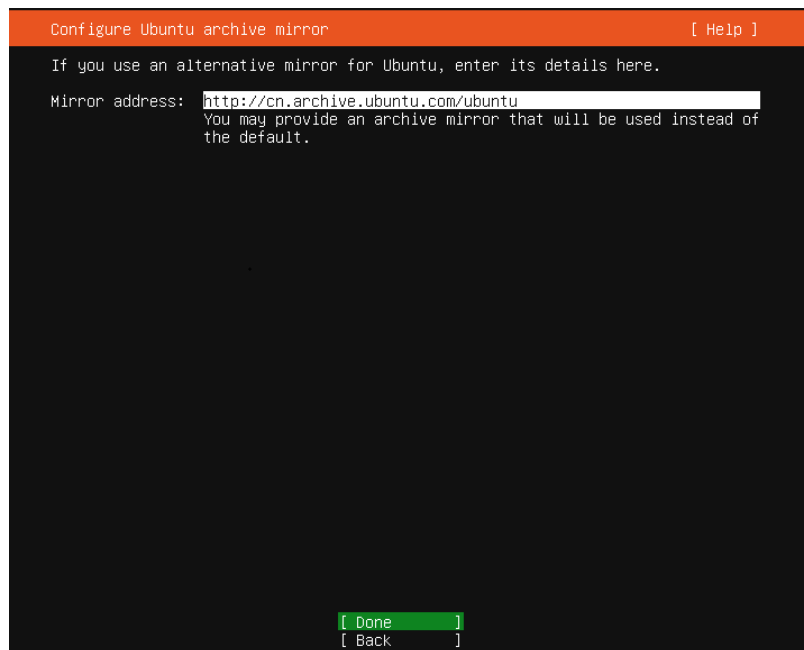
h.   Retain the default settings for disk partitioning (use an entire disk and set up this disk as an LVM group). Select **Done**.
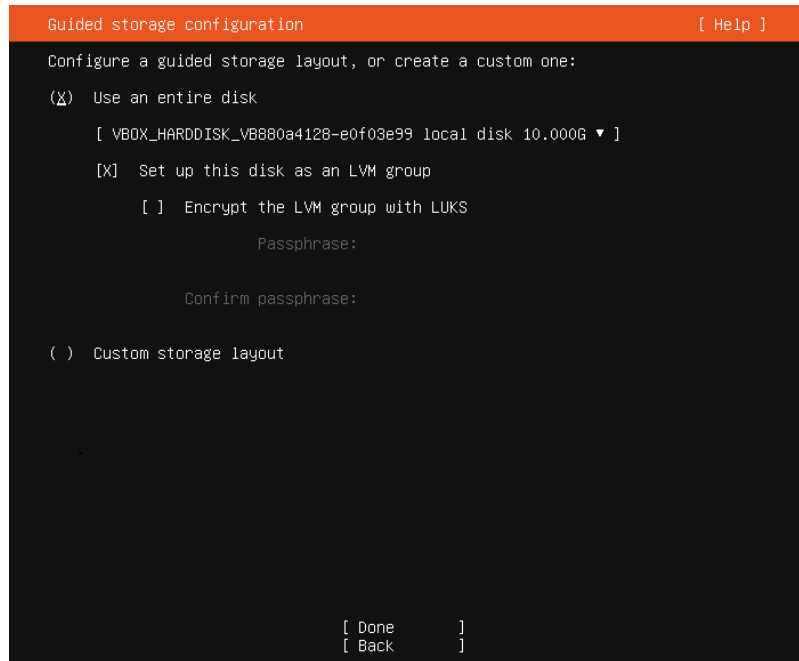
The file system information will be displayed. Check it and select **Done**.
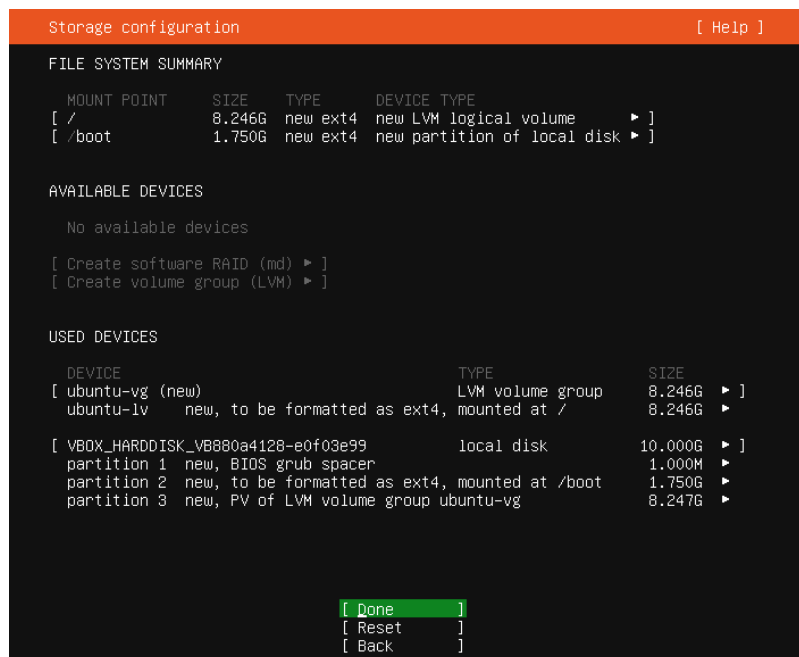
Confirm the destructive action and select **Continue**.
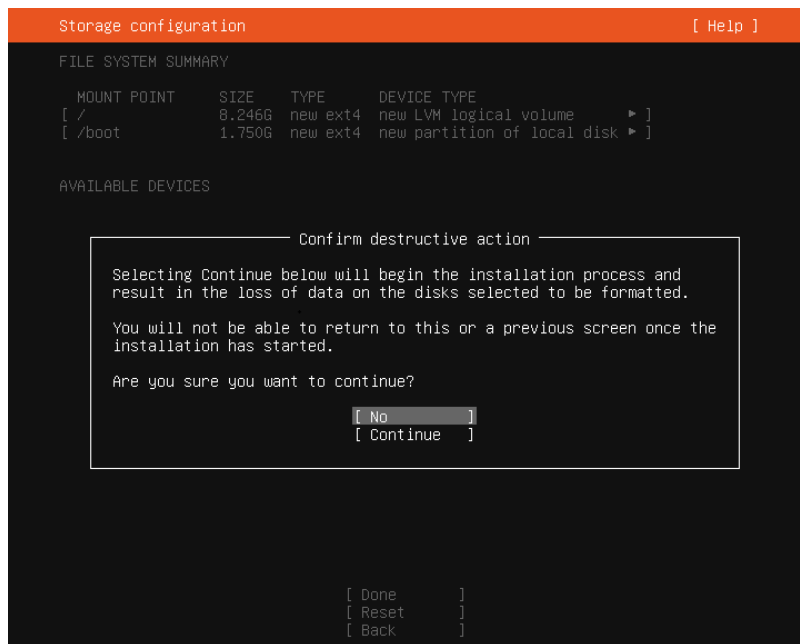
i. Configure the server name, username, and password. Select **Done**.

    Your name: It is not a username for logging in to the server. You can consider it as server description.

    Your Server's name: It is a unique server name on the same network. The name cannot contain uppercase letters.

    Pick a username: It is a username for logging in to the server. If you forget it or its password, you will not be allowed to log in to the server.

    Choose a password: It is the password for logging in to the server.

    Confirm your password: Enter your password again.
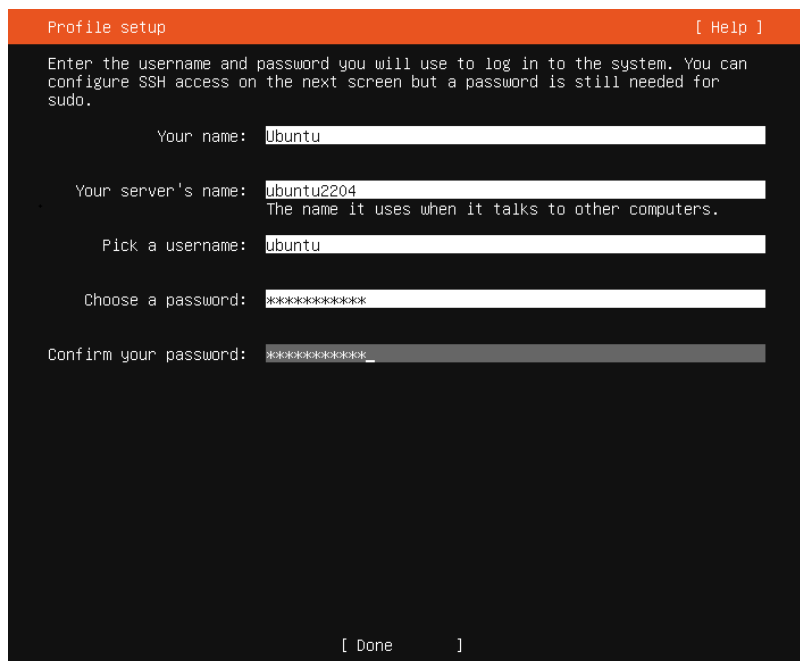
j. Install SSH so that you can remotely connect to the Linux server.

    Select **Install OpenSSH server**. Then, press **Tab** to select **Done**.
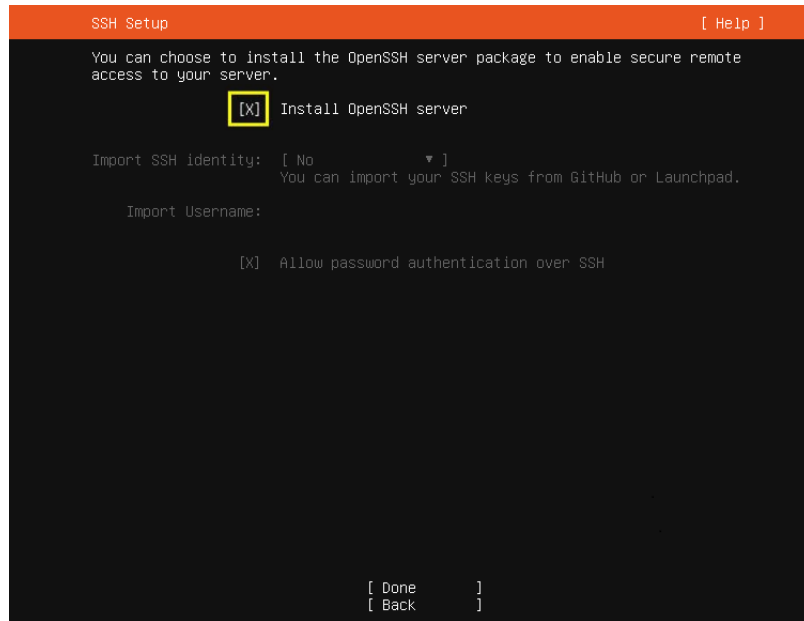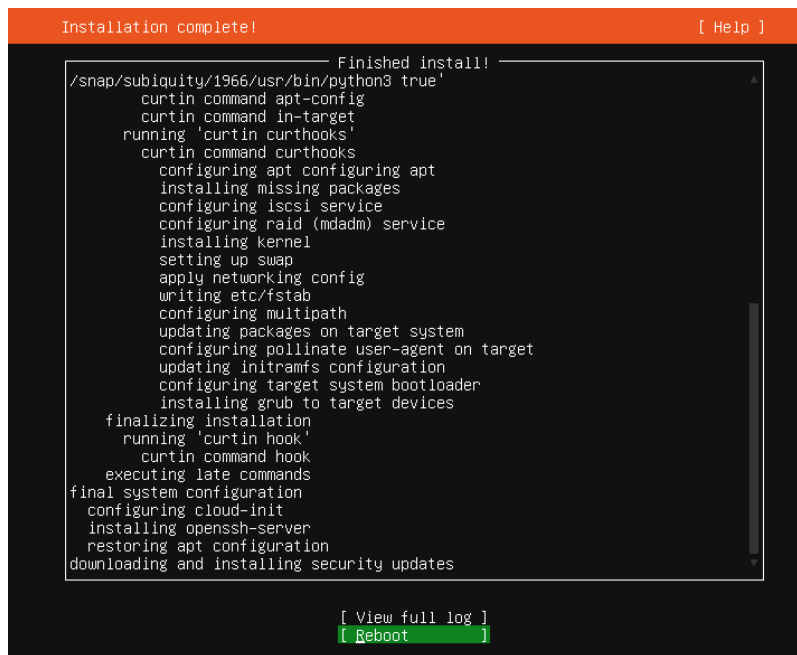
k.   Select **Done** to start the OS installation.
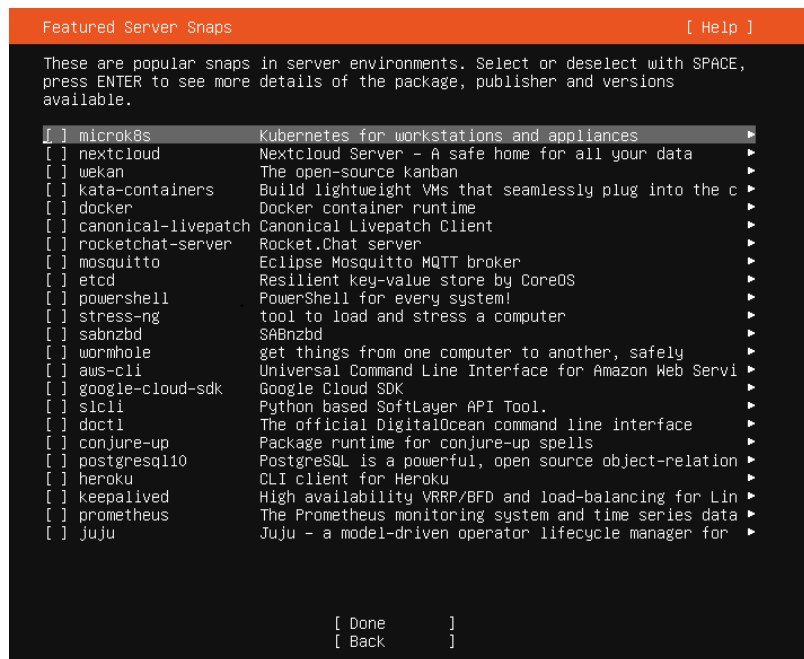
l.   After the installation is complete, select **Reboot** to restart the system.

2.4 Configuring the VM

2.4.1 Installing Drivers and Changing the Disk Identifiers to the UUID Format

To ensure that the ECSs created from the image support both Xen and KVM virtualization, install Native Xen and KVM drivers and change the disk identifiers to the UUID format for the VM which is used as the image source.

This section describes how to perform these operations on a Linux VM that runs Ubuntu 20.04. For other OSs, see **Optimization Process (Linux)**.

Install Native Xen and KVM Drivers

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 the native Xen (xen-pv) and KVM (virtio) drivers to the **/etc/initramfs-tools/modules** file (the format depends on the OS requirements).

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

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

4. Run the following command to generate initrd again:

   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, the VM will not be affected. This makes it easy to modify the drivers. However, the drivers cannot be shown by running the **lsinitrd** command. You can run the following commands to check whether the drivers are built-in ones in the kernel:

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

Change the Disk Identifier in the GRUB Configuration File to the UUID Format

Take Ubuntu 20.04 as an example. Run **blkid** to obtain the UUID of the root partition. Modify the **/boot/grub/grub.cfg** file and use the UUID of the root partition to configure the boot item. If the root partition already uses UUID, no modification is required. The procedure is as follows:

1. Log in to the newly created VM as user **root**.

2. Run the following command to query all types of mounted file systems and their 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"
   ```

3. 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
}
```

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

   – If **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34** is contained, the root partition is in the UUID format and no further action is required.

   – If **root=/dev/xvda1** is contained, the root partition is represented by a device name. Go to step **5**.

5. Obtain the UUID of the root partition based on **root=/dev/xvda1** and information obtained by running the **blkid** command.

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

   vi /boot/grub/grub.cfg

7. Press **i** to enter the editing mode. Change the identifier of the root partition to the UUID format. For example, change **root=/dev/xvda1** to **root=UUID=ec51d860-34bf-4374-ad46-a0c3e337fd34**.

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

9. 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
}
```

Change the Disk Identifiers in the fstab File to the UUID Format

Take Ubuntu 20.04 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. Run the following command to query all types of mounted file systems and their 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"
   ```

2. 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
   ```

3. Check whether the disk identifiers in the **fstab** file are device names or UUIDs.

   – If they are UUIDs, no further action is required.

   – If they are device names, go to step **4**.

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

   vi /etc/fstab

5. Press **i** to enter the editing mode and change the disk identifiers to the UUID format.

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

2.4.2 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 in any of the following ways: **(Recommended) Install Cloud-Init Using the Official Installation Package**, **Install Cloud-Init Using the Official Source Code Package and pip**, and **Install Cloud-Init Using the Official GitHub Source Code**.

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 2-2 lists the Cloud-Init installation paths for CentOS. Select the required installation package from the following addresses.

  Table 2-2 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 2-2**.
>
> ● 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

Cloud-Init-23.2.2 is used as an example to describe how to install Cloud-Init on CentOS, Fedora, Ubuntu, Debian, and SUSE.

Download the **cloud-init-23.2.2.tar.gz** source code package from **https://launchpad.net/cloud-init/trunk/23.2.2/+download/cloud-init-23.2.2.tar.gz**.

- **Centos 7/Fedora Server 36**

NOTICE

Ensure that Python 3 has been installed.

---

a. Check whether Cloud-Init has been installed. If any command output is displayed, Cloud-Init has been installed.
   cloud-init -v

b. Delete the cache directory of Cloud-Init.
   rm -rf /var/lib/cloud/*

c. Install dependency packages of Cloud-Init.
   yum install python3-pip -y
   yum install python3-devel

d. Download the Cloud-Init package.
   wget https://launchpad.net/cloud-init/trunk/23.2.2/+download/cloud-init-23.2.2.tar.gz

e. Decompress the Cloud-Init package.
   tar -zxvf cloud-init-23.2.2.tar.gz

f. Go to the **cloud-init-23.2.2** directory and install dependent libraries:
   cd cloud-init-23.2.2
   pip3 install -r requirements.txt

g. Install Cloud-Init.
   python3 setup.py build
   python3 setup.py install --init-system systemd

h. (Optional) Diable Cloud-Init's network configuration capability by modifying the **/etc/cloud/cloud.cfg** file.
   vi /etc/cloud/cloud.cfg

   Add the following content to the file:
   network:
   ** config: disabled**

i. Restart Cloud-Init and check its status.
   systemctl restart 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

j. Enable Cloud-Init related services to automatically start upon system boot.

> systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-
> final.service

    k.   Check whether Cloud-Init is running properly.
> cloud-init -v
> cloud-init init --local

```
Using /usr/local/lib/python3.6/site-packages
Finished processing dependencies for cloud-init==23.2.2
[root@localhost cloud-init-23.2.2]#
[root@localhost cloud-init-23.2.2]#
[root@localhost cloud-init-23.2.2]#
[root@localhost cloud-init-23.2.2]# cloud-init -v
/usr/local/bin/cloud-init 23.2.2
[root@localhost cloud-init-23.2.2]# _
```

● **Ubuntu 22.0.4/Debian 11**

NOTICE

Ensure that Python 3 has been installed.

---

    a.   Check and delete redundant Cloud-Init configuration files.
> rm -rf /var/lib/cloud/*
> rm -f /var/log/cloud-init*

        Delete all files except log-related configuration files from the **/etc/cloud/cloud.cfg.d/** directory.

    b.   Update your package list and check whether Wget is installed. If it is not, install it.
> sudo apt update
> sudo apt install wget

    c.   Install dependency packages.
> apt-get install cloud-guest-utils -y
> apt-get install python3-pip -y
> apt-get install python3-devel

    d.   Download the Cloud-Init package.
> wget https://launchpad.net/cloud-init/trunk/23.2.2/+download/cloud-init-23.2.2.tar.gz

    e.   Decompress the Cloud-Init package.
> tar -zxvf cloud-init-23.2.2.tar.gz

    f.   Go to the **cloud-init** directory.
> cd cloud-init-23.2.2

    g.   Install dependent libraries.
> pip3 install -r requirements.txt

    h.   Install Cloud-Init.
> python3 setup.py install

    i.   (Optional) Disable Cloud-Init's network configuration capability.

        You need to do so when the Cloud-Init version is 0.7.9 or later and you want to configure the network.

        📖 NOTE

            1. Open the **/etc/cloud/cloud.cfg** file.
> vi /etc/cloud/cloud.cfg

            2. Enter **i** and configure **network**. (If there is no such a configuration item, add it.)
> network:
>   config: disabled

    j.   Restart Cloud-Init and check its status.
> systemctl restart 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

k. Enable Cloud-Init related services to automatically start upon system boot.
    **systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-
    final.service**

l. Check whether Cloud-Init is running properly.
    cloud-init -v
    cloud-init init --local

- **SUSE Enterprise Linux Server 15**

NOTICE

Ensure that Python 3 has been installed.

---

a. View existing SUSE repositories.
    zypper lr

b. Delete the SUSE repositories.
    zypper rr *No. of repositories listed in 1*

c. Configure a SUSE repository.
    **zypper ar https://ftp5.gwdg.de/pub/opensuse/repositories/Cloud:/Tools/SLE_15_SP4/
    Cloud:Tools.repo**

d. Refresh the SUSE repository.
    zypper refresh

e. Install Cloud-Init.
    zypper install cloud-init

f. Run **cloud-init -v**. If error messages similar to the following are displayed, install the dependency packages.


```
.bash_history  .config  .esd_auth  .local  .ssh        .xauthKXyalg  .xsession-errors-:0  bin
linux-i941:~ # cloud-init -v
Traceback (most recent call last):
  File "/usr/bin/cloud-init", line 5, in <module>
    from pkg_resources import load_entry_point
  File "/etc/zypp/repos.d/distribute-0.7.3/pkg_resources.py", line 2989, in <module>
    working_set.require(__requires__)
  File "/etc/zypp/repos.d/distribute-0.7.3/pkg_resources.py", line 726, in require
    needed = self.resolve(parse_requirements(requirements))
  File "/etc/zypp/repos.d/distribute-0.7.3/pkg_resources.py", line 624, in resolve
    raise DistributionNotFound(req)
pkg_resources.DistributionNotFound: requests
linux-i941:~ # ll /usr/bin/
```

```
         100% |                              | 1.3MB 60kB/s
Collecting wheel
  Downloading https://files.pythonhosted.org/packages/1b/d2/22cde5ea9af055f81814
f9f2545f5ed8a053eb749c08d186b369959189a8/wheel-0.31.0-py2.py3-none-any.whl (41kB
)
         100% |                              | 51kB 65kB/s
cloud-init 18.1 requires configobj, which is not installed.
cloud-init 18.1 requires jinja2, which is not installed.
cloud-init 18.1 requires jsonpatch, which is not installed.
cloud-init 18.1 requires jsonschema, which is not installed.
cloud-init 18.1 requires oauthlib, which is not installed.
cloud-init 18.1 requires pyyaml, which is not installed.
cloud-init 18.1 requires requests, which is not installed.
Installing collected packages: pip, wheel
Successfully installed pip-10.0.1 wheel-0.31.0
linux-i941:~ # pip -V
pip 10.0.1 from /usr/lib/python2.7/site-packages/pip (python 2.7)
linux-i941:~ #
```

pip install requests pyyaml oauthlib jsonschema jsonpatch jinja2 configobj

g. Check whether Cloud-Init is successfully installed. If the following error message is displayed, configure **datasource_list** in **/etc/cloud/cloud.cfg**.

datasource_list: [OpenStack]
datasource:
  OpenStack:
    metadata_urls: ['http://169.254.169.254']
    max_wait: 120
    timeout: 5
    apply_network_config: false

h. Modify the configuration file, restart Cloud-Init, and check the Cloud-Init status.

systemctl restart 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

i. Enable Cloud-Init related services to automatically start upon system boot.

systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

j. Check whether Cloud-Init is running properly.

cloud-init -v
cloud-init init --local

Install Cloud-Init Using the Official Source Code Package and pip

The following operations use Cloud-Init 0.7.9 as an example to describe how to install Cloud-Init.

1. Download the **cloud-init-0.7.9.tar.gz** source code package (version 0.7.9 is recommended) and upload it to the **/home/** directory of the ECS.

   Download **cloud-init-0.7.9.tar.gz** from the following path:

   https://launchpad.net/cloud-init/trunk/0.7.9/+download/cloud-init-0.7.9.tar.gz

2. Create a **pip.conf** file in the **~/.pip/** directory and edit the following content:

   > 📖 **NOTE**
   >
   > If the **~/.pip/** directory does not exist, run the **mkdir ~/.pip** command to create it.

   [global]
   index-url = https://**<$mirror>**/simple/
   trusted-host = **<$mirror>**

   > 📖 **NOTE**
   >
   > Replace *<$mirror>* with a public network PyPI source.
   >
   > Public network PyPI source: **https://pypi.python.org/**

3. Run the following command to install the downloaded Cloud-Init source code package (select **--upgrade** as needed during installation):

pip install [--upgrade] /home/cloud-init-0.7.9.tar.gz

📖 NOTE

> For details about how to install a Cloud-Init source code package, see **Cloud-Init Documentation**

4. Run the **cloud-init -v** command. Cloud-Init is installed successfully if the following information is displayed:

   ```
   cloud-init 0.7.9
   ```

5. Enable Cloud-Init to automatically start upon system boot.

   – If the OS uses SysVinit to manage automatic start of services, run the following commands:

      chkconfig --add cloud-init-local; chkconfig --add cloud-init; chkconfig --add cloud-config; chkconfig --add cloud-final

      chkconfig cloud-init-local on; chkconfig cloud-init on; chkconfig cloud-config on; chkconfig cloud-final on

      service cloud-init-local status; service cloud-init status; service cloud-config status; service cloud-final status

   – If the OS uses Systemd to manage automatic start of services, run the following commands:

      systemctl enable cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

      systemctl status cloud-init-local.service cloud-init.service cloud-config.service cloud-final.service

---

⚠️ **CAUTION**

If you install Cloud-Init using the official source code package and pip, pay attention to the following:

1. Add user **syslog** to the **adm** group during the installation. If user **syslog** exists, add it to the **adm** group. For some OSs (such as CentOS and SUSE), user **syslog** may not exist. Run the following commands to create user **syslog** and add it to the **adm** group:

   useradd syslog

   groupadd adm

   usermod -g adm syslog

2. Change the value of **distro** in **system_info** in the **/etc/cloud/cloud.cfg** file based on the OS release version, such as **distro: ubuntu**, **distro: sles**, **distro: debian**, and **distro: fedora**.

Install Cloud-Init Using the Official GitHub Source Code

You can obtain the Cloud-Init source code from GitHub at **https://github.com/canonical/cloud-init/**

1. Run the following commands to download the source code package and copy it to the **/tmp/CLOUD-INIT** folder:

☐ NOTE

> Cloud-Init 0.7.6: **https://github.com/canonical/cloud-init/archive/refs/tags/0.7.6.zip**
> Cloud-Init 0.7.9: **https://github.com/canonical/cloud-init/archive/refs/tags/0.7.9.zip**
> All versions: **https://github.com/canonical/cloud-init/releases**

wget https://github.com/canonical/cloud-init/archive/refs/tags/0.7.6.zip

mkdir /tmp/CLOUD-INIT

cp cloud-init-0.7.6.zip /tmp/CLOUD-INIT

cd /tmp/CLOUD-INIT

2. Run the following command to decompress the package:

    unzip cloud-init-0.7.6.zip

3. Run the following command to enter the **cloud-init-0.7.6** folder:

    cd cloud-init-0.7.6

4. (Optional) If the Cloud-Init version is 18.3 to 22.3, run the following commands:

    **sed -i '/VALID_DMI_ASSET_TAGS =/a\VALID_DMI_ASSET_TAGS +=
    ["HUAWEICLOUD"]' cloudinit/sources/DataSourceOpenStack.py**

    **cat cloudinit/sources/DataSourceOpenStack.py | grep
    VALID_DMI_ASSET_TAGS**

    If the following information is displayed, the execution is successful.

    ```
    [root@ecs-cc9e cloud-init]# cat cloudinit/sources/DataSourceOpenStack.py | grep VALID_DMI_ASSET_TAGS
    VALID_DMI_ASSET_TAGS = VALID_DMI_PRODUCT_NAMES
    VALID_DMI_ASSET_TAGS += ["HUAWEICLOUD"]
    VALID_DMI_ASSET_TAGS += [DMI_ASSET_TAG_OPENTELEKOM, DMI_ASSET_TAG_SAPCCLOUD]
        elif dmi.read_dmi_data("chassis-asset-tag") in VALID_DMI_ASSET_TAGS:
    [root@ecs-cc9e cloud-init]#
    ```

5. Install Cloud-Init. The commands vary depending on the OS type.

    – For CentOS 6.x or SUSE 11.x, run the following commands:

       python setup.py build

       python setup.py install --init-system sysvinit

    – For CentOS 7.x or SUSE 12.x, run the following commands:

       python setup.py build

       python setup.py install --init-system systemd

    ☐ NOTE

    > 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**

6. 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

7. Run the following commands to check whether Cloud-Init has been installed:

   cloud-init -v

   cloud-init init --local

   Cloud-Init is successfully installed if the following information is displayed:

   ```
   cloud-init 0.7.6
   ```

2.4.3 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 select user **root**, enable the SSH permissions of user **root** and enable remote login to the ECS using a password.

   - If you inject a password, use it to log in to the ECS remotely using SSH or noVNC.
   - If you inject a key, use it to log in to the ECS remotely using SSH.

2. Enable remote login using the password of user **root** and enable the SSH permissions of user **root**.

   Take CentOS 6.7 as an example. If the value of **disable_root** in the **/etc/cloud/cloud.cfg** file is **0**, the permissions are enabled. (In some OSs, value
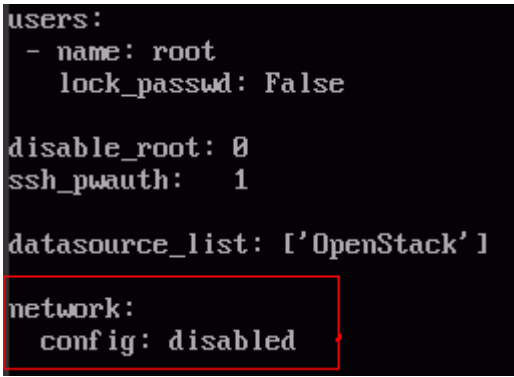
true indicates that the permissions are disabled, and **false** indicates that the permissions are enabled). Set **ssh_pwauth** to **1**, and **lock_passwd** to **False** (indicating that user passwords are not locked).

```
users:
 - name: root
   lock_passwd: False

disable_root: 0
ssh_pwauth: 1
```

3. 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**:

   Figure 2-16 Preventing Cloud-Init from taking over the network

   □ NOTE

   The added content must be in the YAML format.

4. 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
      apply_network_config: false
   ```

   □ NOTE

   - You can decide whether to set **max_wait** and **timeout**. The values of **max_wait** and **timeout** in the preceding command output 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
   - **apply_network_config: false** is only requited by users who use Cloud-Init 18.3 or later.

5. Add the following content to **/etc/cloud/cloud.cfg**:

   manage_etc_hosts: localhost

   This prevents the system from staying in the **Waiting for cloudResetPwdAgent** state for a long time during ECS startup.

Figure 2-17 Adding **manage_etc_hosts: localhost**

```
datasource_list: ['OpenStack']
manage_etc_hosts: localhost

datasource:
  OpenStack:
    # timeout: the timeout value for a request at metadata service
    timeout : 50
    # The length in seconds to wait before giving up on the metadata
    # service.  The actual total wait could be up to
    #    len(resolvable_metadata_urls)*timeout
    max_wait : 120
```

6. Run the **vi /etc/ssh/sshd_config** command to open the **/etc/ssh/sshd_config** file using the vi editor.

   Change the value of **PasswordAuthentication** in the **sshd_config** file to **yes**.

   📖 **NOTE**

   > For SUSE and openSUSE, change the values of the following parameters in the **sshd_config** file to **yes**:
   > - PasswordAuthentication
   > - ChallengeResponseAuthentication

7. Modify **cloud_init_modules** in the **cloud.cfg** configuration file.

   – Move **ssh** from the bottom to the top to speed up the SSH login.

   – Enable the hostname update. Do not comment out or delete the **- update_hostname** statement.

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

8. 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/python*.*/site-packages/cloudinit/sources/__init__.py

      The Python version varies depending on the OS.

      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.

b. Run the following command to switch to the **cloudinit/sources** folder:

cd /usr/lib/python*.*/site-packages/cloudinit/sources/

The Python version varies depending on the OS.

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*

9. 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

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

10. Delete user **linux** and the **/home/linux** directory from the image template.

userdel linux

rm -fr /home/linux

11. Ensure that Cloud-Init is enabled.

Delete the **/etc/cloud/cloud-init.disabled** file by running **cloud-init clean -- machine-id**. If there is no such a file, skip this step.

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**.

2.4.4 (Optional) Installing the One-Click Password Reset Plug-in

To ensure that you can reset the password of each ECS created from the image with a few clicks, you are advised to install the one-click password reset plug-in (CloudResetPwdAgent) on the VM which is used as the image source.

Procedure

1. Download the CloudResetPwdAgent software package.

   > **NOTE**
   >
   > The one-click password reset plug-in can be automatically updated only if an EIP is bound to the VM.

   You can download the **CloudResetPwdAgent.zip** package from:

   For 32-bit OSs: **http://ap-southeast-1-cloud-reset-pwd.obs.ap-southeast-1.myhuaweicloud.com/linux/32/reset_pwd_agent/CloudResetPwdAgent.zip**

   For 64-bit OSs: **http://ap-southeast-1-cloud-reset-pwd.obs.ap-southeast-1.myhuaweicloud.com/linux/64/reset_pwd_agent/CloudResetPwdAgent.zip**

2. Run the following command to decompress **CloudResetPwdAgent.zip**:

   unzip -o -d *Decompressed directory* **CloudResetPwdAgent.zip**

   There is no special requirement for the directory that stores the decompressed **CloudResetPwdAgent.zip**. You can choose a directory as you need. If the directory is **/home/PwdAgent/test**, the command is as follows:

   unzip -o -d /home/PwdAgent/test CloudResetPwdAgent.zip

3. Install the one-click password reset plug-in.

   a. Run the following command to open the **CloudResetPwdAgent.Linux** file:

      cd CloudResetPwdAgent/CloudResetPwdAgent.Linux

   b. Run the following command to add the execute permission for the **setup.sh** file:

      chmod +x setup.sh

   c. Run the following command to install the plug-in:

      sudo sh setup.sh

   d. Run the following command to check whether the installation is successful:

      service cloudResetPwdAgent status

      If the status of CloudResetPwdAgent is not **unrecognized service**, the installation is successful. Otherwise, the installation failed.

      > **NOTE**
      >
      > If the installation failed, check whether the installation environment meets requirements and install the plug-in again.

2.4.5 Configuring NetworkManager

Linux allows you to use NetworkManager to automatically configure the VM network. You are advised to use NetworkManager for new OS versions.

Alternatively, you can use the native network management service of the OS.

Red Hat, Oracle, CentOS 6.x, CentOS 7.x, EulerOS 2.x, Fedora 22, or Later

Install NetworkManager and use it for automatic network configuration.

1. Run the following command to install NetworkManager:

   yum install NetworkManager

2. Delete **ifcfg-eth1** to **ifcfg-eth11** from the **/etc/sysconfig/network-scripts/** directory and retain only **ifcfg-eth0**.

3. Run the following command to disable the network:

   service network stop

4. Run the following command to disable automatic startup of the network:

   chkconfig network off

5. Run the following commands to restart messagebus and NetworkManager and enable NetworkManager to start automatically at startup:

   service messagebus restart

   service NetworkManager restart

   chkconfig NetworkManager on

Debian 9.0 or Later

Install NetworkManager and use it for automatic network configuration.

1. Run the following command to install NetworkManager:

   apt-get install network-manager

2. Change the value of **managed** in the **/etc/NetworkManager/NetworkManager.conf** file to **true**.

3. Modify **/etc/network/interfaces** and retain only **eth0**.

4. Run the following commands to disable the network, restart messagebus and NetworkManager, and enable NetworkManager to start automatically at startup:

   service network-manager restart

   chkconfig network-manager on

   service networking stop

   service messagebus restart

   service network-manager restart

Ubuntu 14 or Later

Install NetworkManager and use it for automatic network configuration.

1. Run the following command to install NetworkManager:

apt-get install network-manager

2. Change the value of **managed** in the **/etc/NetworkManager/ NetworkManager.conf** file to **true**.

3. Modify **/etc/network/interfaces** and retain only **eth0**.

4. Run the following command to disable the network:

    service networking stop

5. Run the following command to disable automatic startup of the network:

    chkconfig network off

6. Run the following commands to restart D-Bus and NetworkManager:

    service dbus restart

    service network-manager restart

SUSE 11 SP3 and openSUSE 13 or Later

Install NetworkManager and use it for automatic network configuration.

1. Delete **ifcfg-eth1** to **ifcfg-eth11** from the **/etc/sysconfig/network-scripts/** directory and retain only **ifcfg-eth0**.

2. Run the following command to install NetworkManager:

    zypper install NetworkManager

3. Start YaST, choose **Network Devices** in the navigation pane on the left, and select **Network Settings** in the right pane. In the **Network Setup Method** area of the **Global Options** page, change **Traditional Method with ifup** to **User Controlled with NetworkManager**.

2.5 Obtaining the Image File

After the VM is configured, perform the following operations to generate and export a Linux image file:

1. Open VirtualBox, select the VM, choose **Settings** > **Storage**, and select **Linux.vhd**.

    Linux is the VM name.

2. On the right pane, view the image file location.

3. Go to the location to obtain the generated **Linux.vhd** image file.

2.6 Registering the Image File as a Private Image

Upload the image file to an OBS bucket and register it as a private image.

Constraints

-   Only an unencrypted image file or an image file encrypted using SSE-KMS can be uploaded to an OBS bucket.

-   When uploading an image file, you must select an OBS bucket with the storage class of Standard.

Procedure

1. Use OBS Browser+ to upload the image file. For details, see **OBS Browser+ Best Practices**.

   For how to download OBS Browser+, see **https:// support.huaweicloud.com/eu/browsertg-obs/obs_03_1003.html**.

2. Register the external image file as a private image. For details, see **Registering an Image File as a Private Image (Linux)**.

3 Cleaning Up the Disk Space of a Windows ECS

Scenarios

To clean up the disk space of a Windows ECS, you need to perform the following operations.

Disable Virtual Memory

Some disk space of an ECS serves as virtual memory which can be used when the ECS memory is exhausted. However, when the memory usage is high, frequent switching between the memory and virtual memory causes a large number of extra I/Os, which deteriorates the I/O performance. Therefore, you can disable virtual memory to free up disk space on Windows.

An ECS running Windows Server 2008 R2 Standard 64-bit is used as an example to describe how to disable virtual memory.

1. Right-click the **Computer** icon and choose **Properties**. In the displayed window, click **Advanced System Settings**.

2. In the **System Properties** dialog box, click the **Advanced** tab.

3. Click **Settings** in the **Performance** area.

4. In the displayed **Performance Options** dialog box, click the **Advanced** tab and then **Change**. The **Virtual Memory** dialog box is displayed.

5. Delete virtual memory.

   In the **Virtual Memory** dialog box, deselect **Automatically manage paging file size for all drives**, select the paging file of the disk whose virtual memory is to be deleted, select **No paging file**, and click **Set**.

> ⚠ **CAUTION**
>
> If the warning "If you disable the paging file or set the initial size to less than *xxx* megabytes and a system error occurs, Windows might not record details that could help identify the problem. Do you want to continue?" is displayed, set the initial size to *xxx* shown in the warning. The partition for storing paging files must have sufficient space. If the disk resources are insufficient, virtual memory will be insufficient. You only need to set virtual memory in one partition of the ECS.

6. Choose **Start** > **Control Panel** > **Appearance and Personalization** > **Show hidden files and folders**. The **Folder Options** dialog box is displayed. Click the **View** tab, deselect **Hide protected operating system files (Recommended)**, and select **Show hidden files, folders, and drives**. When you deselect **Hide protected operating system files (Recommended)**, a warning is displayed, as shown in **Figure 3-1**. Click **Yes** and then click **Apply**.

Figure 3-1 Warning
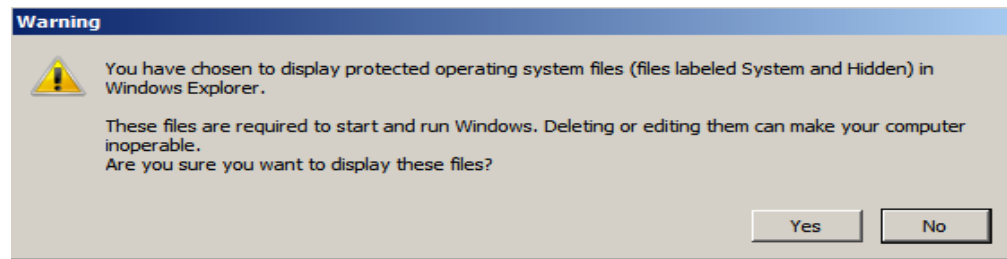
Figure 3-2 Showing hidden files
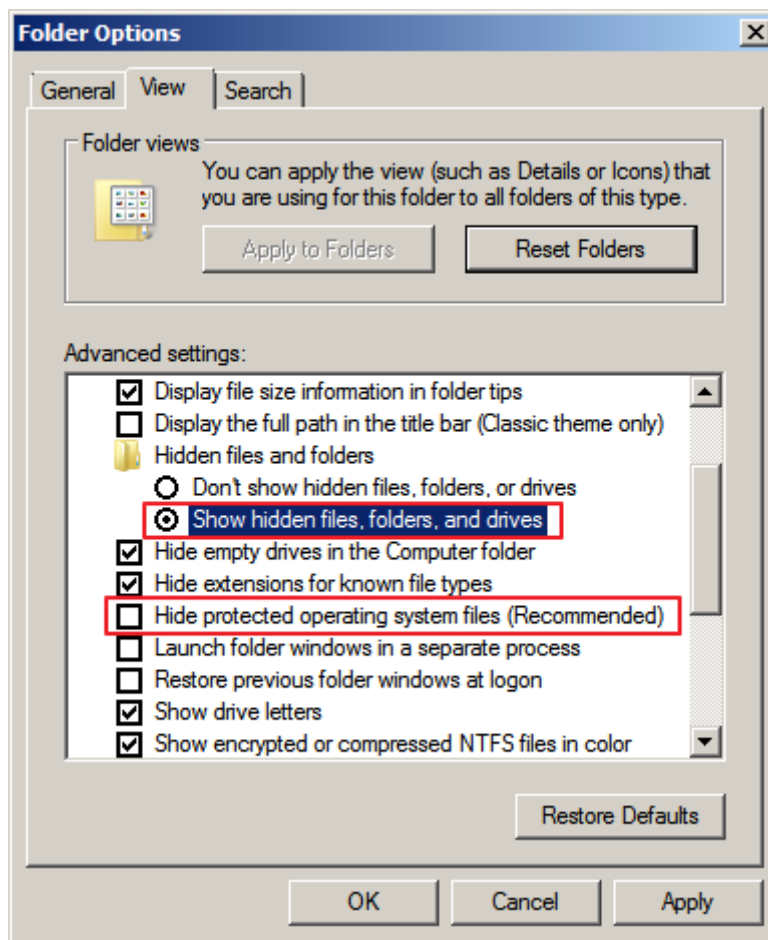
7. After file hiding is disabled, delete the hidden file **pagefile.sys** from disk C. After the OS is restarted, the available space increases.

Disable Hibernation

An ECS running Windows Server 2008 R2 Standard 64-bit is used as an example to describe how to disable hibernation.

Method 1: Delete the **Hiberfil.sys** file.

The **Hiberfil.sys** file is generated by Windows hibernation and occupies large system disk space. It saves memory data and sessions to disks so that the memory image file required by sessions can be quickly restored after the computer is restarted. Perform the following steps to delete the **Hiberfil.sys** file to free up some disk space:

1. Run **cmd.exe** as an administrator to open the command line interface (CLI).
2. Run the following command to disable hibernation (**Hiberfil.sys** file will be automatically deleted):

   powercfg -h off

Method 2: Disable hibernation.

1. Choose **Start** > **Control Panel** > **Appearance and Personalization** > **Personalization** > **Change screen saver** and click **Change plan settings**.

2. In the **Power Options** window, click **Change plan settings** on the right of **Balanced (recommended)**.

3. Set **Turn off the display** to **Never**.
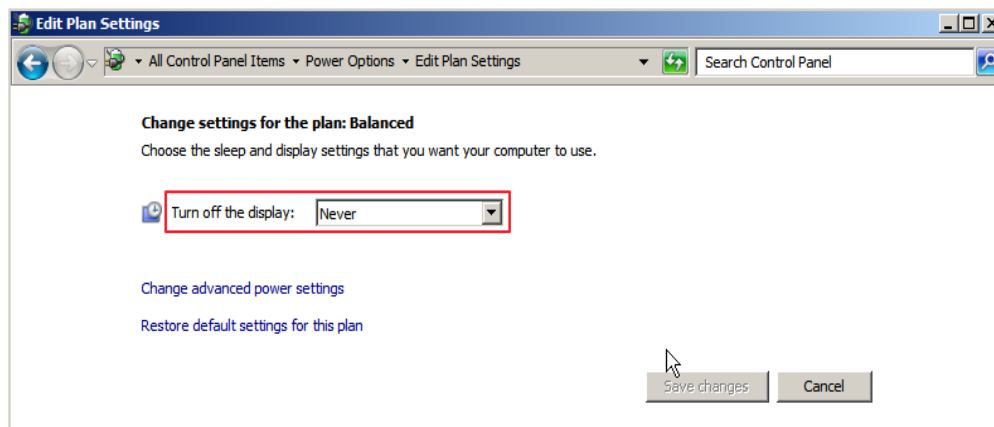
Delete Internet Explorer Temporary Files

An ECS running Windows Server 2008 R2 Standard 64bit is used as an example to describe how to delete temporary files generated by Internet Explorer.

Internet Explorer stores all information about Internet access in a specified folder, including Internet Explorer cache files, cookies files, browsing history, visited websites, URLs in the address bar, and Internet Explorer forms/passwords. Perform the following steps to delete temporary files generated by Internet Explorer:

1. Choose **Start** > **Control Panel** > **Network and Internet** > **Internet Options**. Click **Delete browsing history and cookies** to delete cookies and temporary files of Internet Explorer.
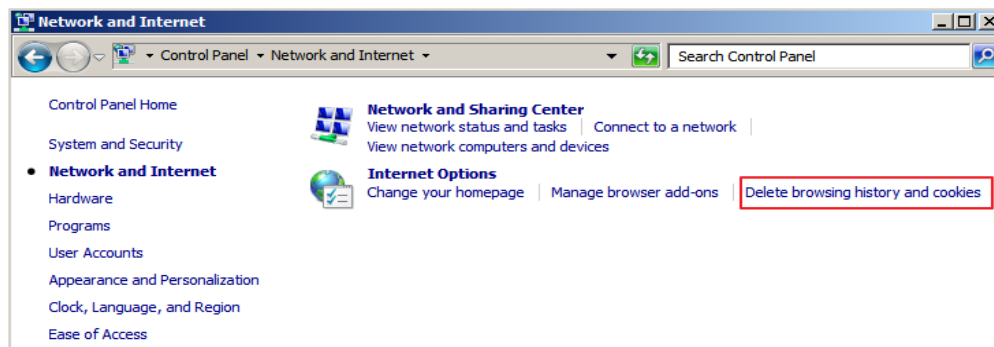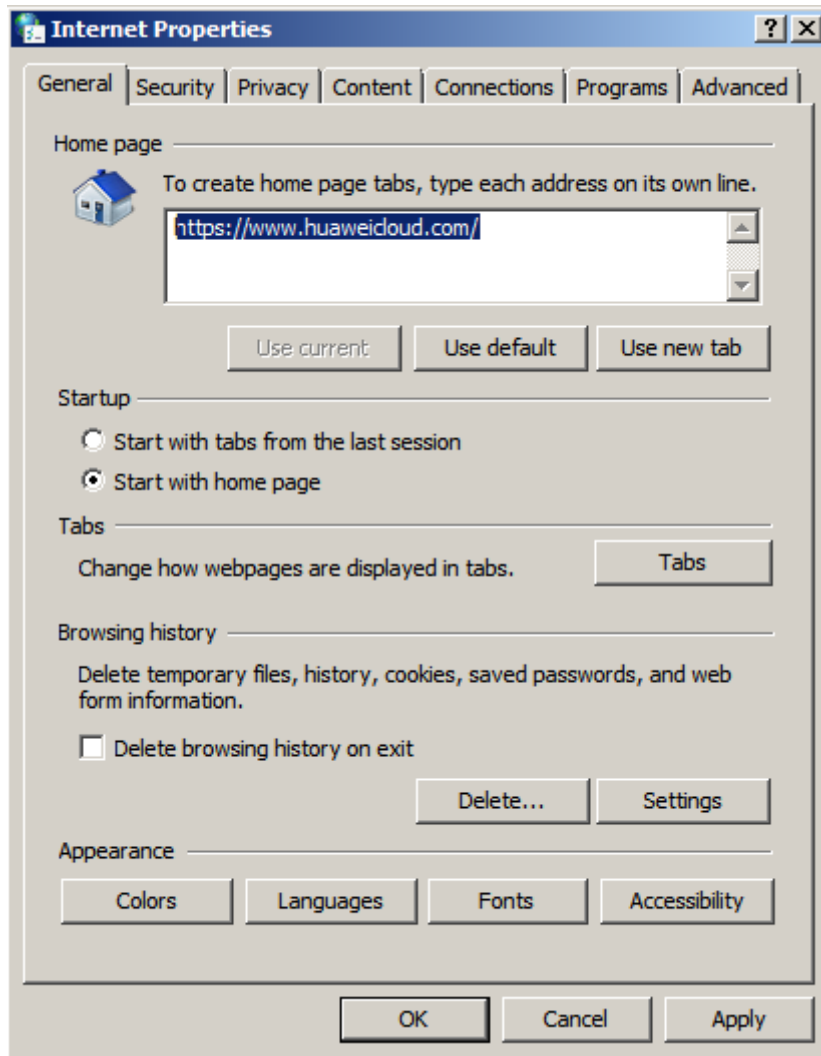
2. After the preceding step is performed, there may be residual files or records left. Open the **C:\Windows\temp** and **C:\Users**_Username_**\AppData\Local \Temp** folders and delete all files in them. (The folders store visited websites and other temporary information.)

Disable Data Execution Prevention (DEP)

DEP reserves a part of ECS memory for temporarily storing application data and another part of memory for temporarily storing application instructions. This protects ECSs from viruses and other security threats.

An ECS running Windows Server 2008 R2 Standard 64-bit is used as an example to describe how to disable DEP.

1. Right-click the **Computer** icon and choose **Properties**. In the displayed window, click **Advanced System Settings**.

2. In the **System Properties** dialog box, click the **Advanced** tab.

3. Click **Settings** in the **Performance** area.

4. In the **Performance Options** dialog box, click the **Data Execution Prevention** tab, select **Turn on DEP for essential Windows programs and services only**, and click **Apply**.
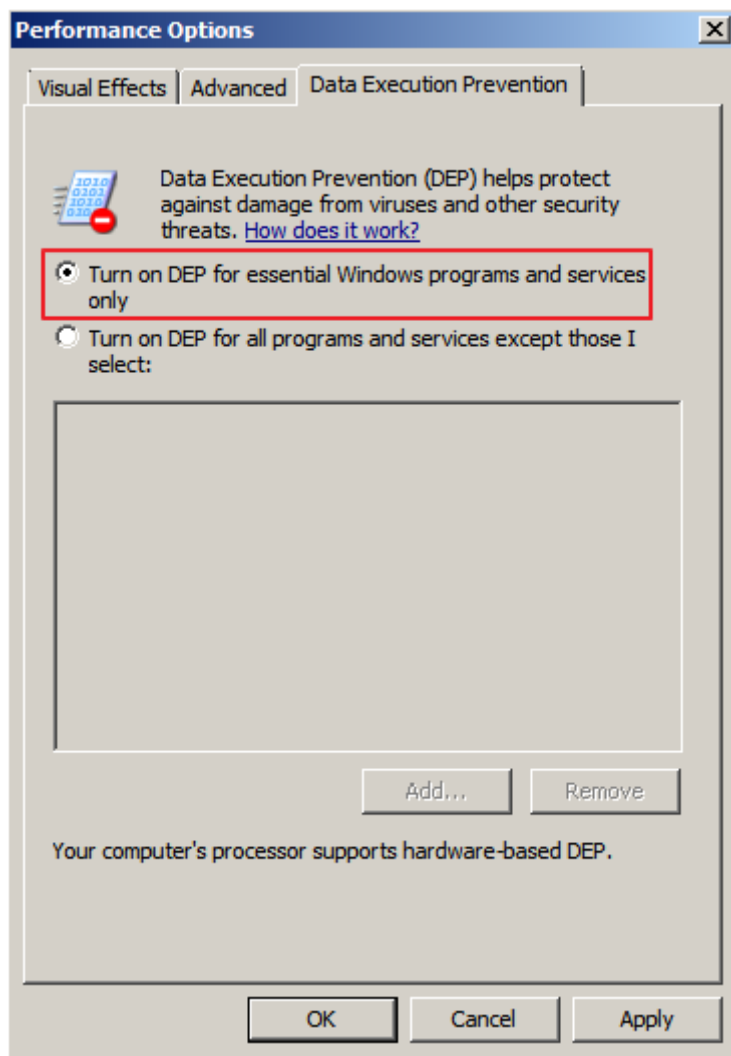
Delete Redundant Application Files

The **C:\Windows\prefetch** folder stores the index files generated by applications. The files are used to improve system performance and speed up system startup and file reading. However, the number of files increases with time. Deleting redundant files can free up disk space on Windows.

Delete all files in the **C:\Windows\prefetch** folder to delete redundant application files.

Clean Up Disks

You can use **Disk Cleanup** of ECSs to delete temporary files, empty the recycle bin, and delete redundant system files and other files.

On the ECS, click **Start**. In the search box, enter **Disk Cleanup**. Click the displayed **Disk Cleanup** to scan for the space that can be freed up. After the scan is complete, confirm the files to be deleted and click **OK** to start disk cleanup.

If **Disk Cleanup** is unavailable, you need to install Desktop Experience first. An ECS running Windows Server 2008 R2 Standard 64bit is used as an example to describe how to install Desktop Experience.

1. Choose **Start** > **All Programs** > **Administrative Tools** > **Server Manager**. In the navigation pane, click **Features**.

2. Click **Add Features**.

3. In the **Select Features** dialog box, select **Desktop Experience** and click **Next**.

Figure 3-3 Installing Desktop Experience
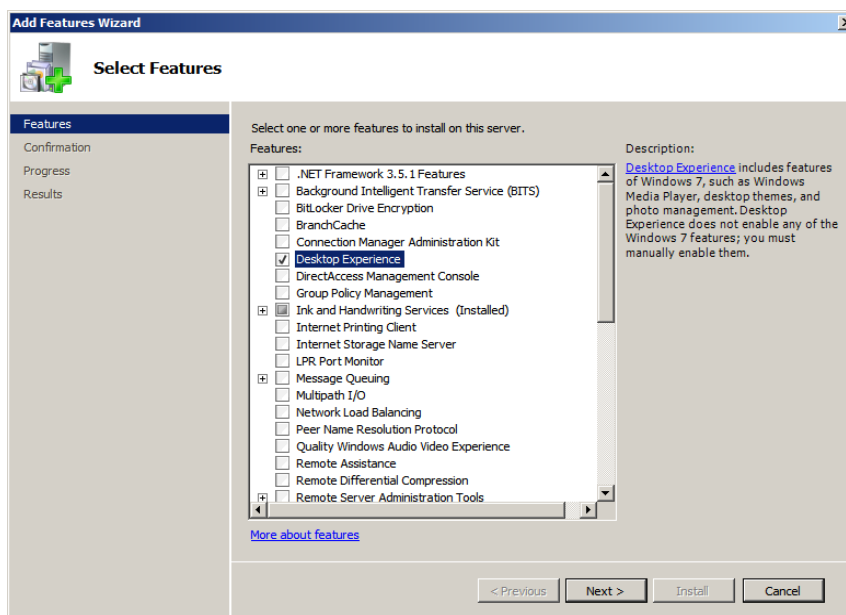
4. In the **Confirm Installation Selections** dialog box, confirm the installation of Desktop Experience, and click **Install**.

5. When the installation progress reaches 100% in the **Results** dialog box, the system prompts you to restart the server. Click **Close**, and then click **Yes** to restart the server.

6. After the server is restarted, start the server manager. In the **Function Summary** area, check whether Desktop Experience is installed.

Figure 3-4 Successful installation of Desktop Experience
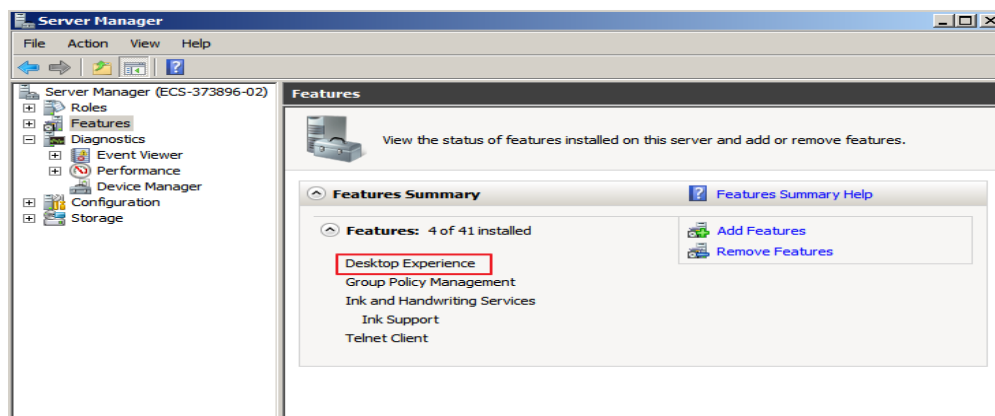
7. Choose **Start** > **All Programs** > **Accessories** > **System Tools** > **Disk Cleanup** to start the disk cleanup tool.

Defragment and Optimize Drives

Disk defragmentation is a process in which system software or professional disk defragmentation software is used to organize the fragments generated during the long-term use of disks into the smallest number of contiguous fragments, improving the performance and running speed of the server.

An ECS running Windows Server 2012 R2 Datacenter 64bit is used as an example to describe how to defragment and optimize drives.

1. Choose **Control Panel** > **All Control Panel Items** > **Administrative Tools**, and double-click **Defragment and Optimize Drives**.
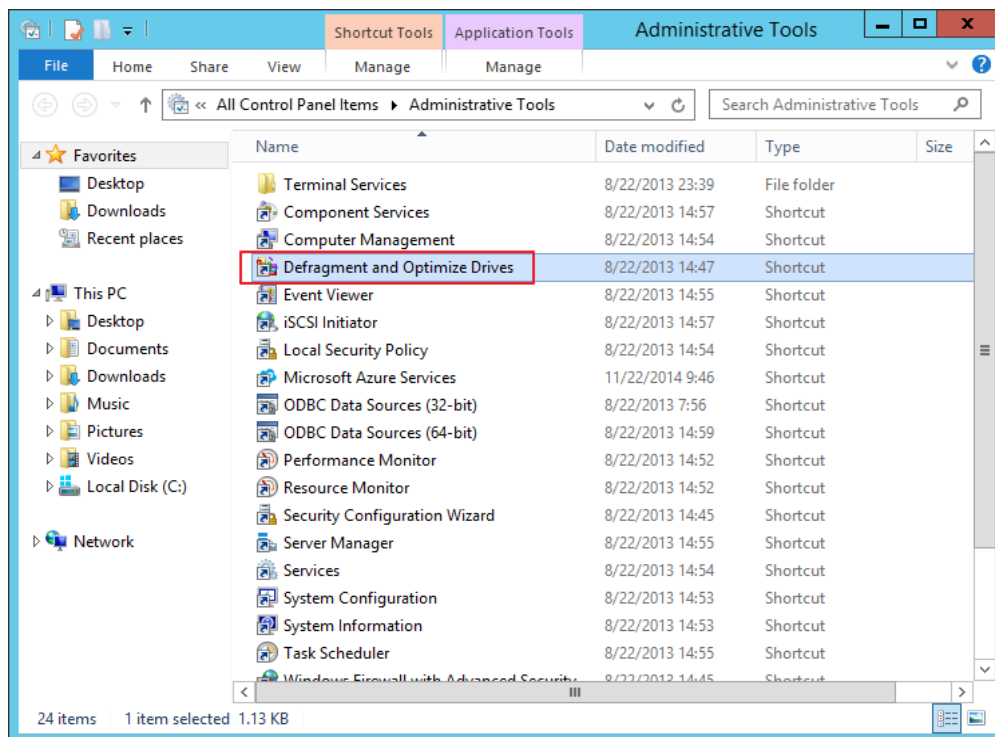
2. Select the disk partition to be optimized, and then click **Analyze** to analyze whether the disk partition needs to be optimized. If the analysis result indicates that optimization is necessary, click **Optimize**.
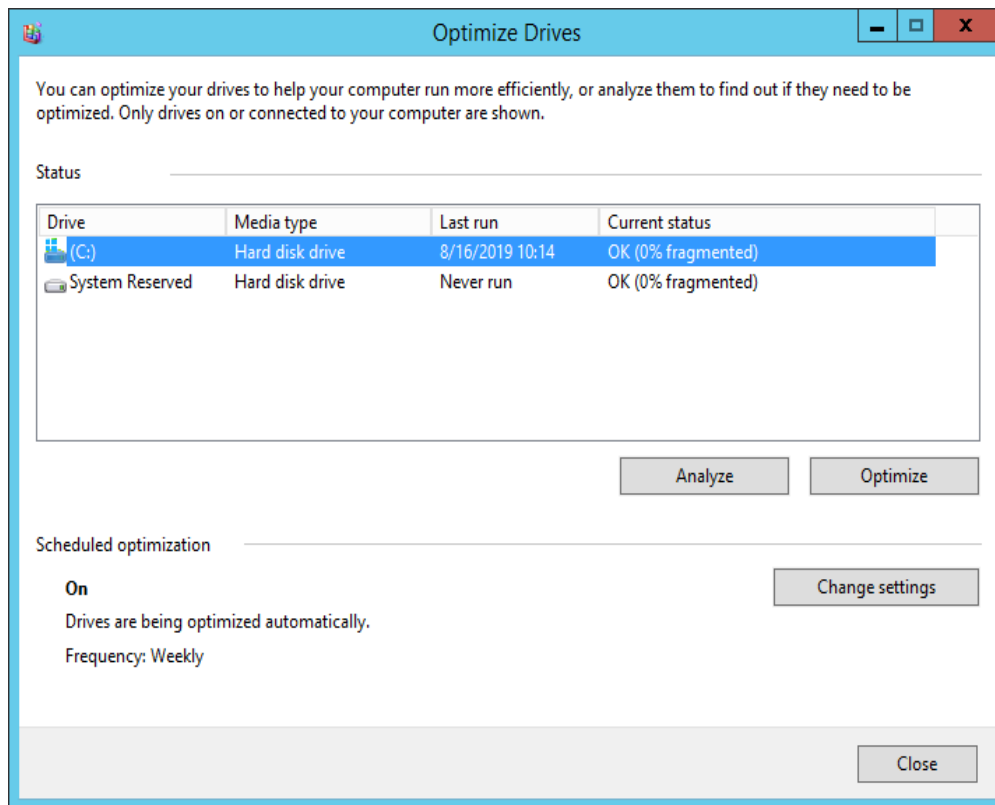
Delete .dmp Files

When a blue screen of death (BSOD) occurs on a Windows ECS, the system automatically generates a BSOD error file with the suffix .dmp. A .dmp file is a system error file in Windows, such as, **memory.dmp** and **minixxxx.dmp**.

You can manually delete .dmp files from disk C to free up system disk space.

⚠️ **CAUTION**

After the .dmp file is deleted, the BSOD cause on the ECS cannot be queried.

Clean Up Component Store

Windows Component Store stores all the files required for installing Windows. Updated installation files are also stored in Component Store, which causes the size of Component Store to increase as the update increases.

An ECS running Windows Server 2012 R2 Datacenter 64-bit is used as an example to describe how to clean up Component Store.

1. In Windows PowerShell, run the following command to delete the backup files generated during the Service Pack installation:

    dism /online /cleanup-image /spsuperseded

```
PS C:\Users\Administrator> dism /online /cleanup-image /spsuperseded

Deployment Image Servicing and Management tool
Version: 6.3.9600.19408

Image Version: 6.3.9600.19397

Service Pack Cleanup cannot proceed: No Service Pack backup files were found.
The operation completed successfully.
```

2. Run the following command to check the size of Component Store:

   Dism.exe /Online /Cleanup-Image /AnalyzeComponentStore

```
PS C:\Users\Administrator> Dism.exe /Online /Cleanup-Image /AnalyzeComponentStore

Deployment Image Servicing and Management tool
Version: 6.3.9600.19408

Image Version: 6.3.9600.19397

[==========================99.9%========================= ]

Component Store (WinSxS) information:

Windows Explorer Reported Size of Component Store : 7.90 GB

Actual Size of Component Store : 7.75 GB

    Shared with Windows : 4.12 GB
    Backups and Disabled Features : 3.33 GB
    Cache and Temporary Data : 297.92 MB

Date of Last Cleanup : 2019-08-16 11:00:48

Number of Reclaimable Packages : 3
Component Store Cleanup Recommended : Yes

The operation completed successfully.
```

3. Run the following command to clean up Component Store:

   Dism.exe /online /Cleanup-Image /StartComponentCleanup

```
PS C:\Users\Administrator> Dism.exe /online /Cleanup-Image /StartComponentCleanup

Deployment Image Servicing and Management tool
Version: 6.3.9600.19408

Image Version: 6.3.9600.19397

[==========================100.0%========================]
The operation completed successfully.
PS C:\Users\Administrator> _
```

Delete System Logs

System logs record hardware and software information, system problems, and system events. You can use the logs to locate error causes or track attacker actions. System logs include application logs, security logs, startup logs, and event forwarding logs. The **System32** > **LogFiles** folder on disk C of Windows stores the operation logs and event logs of Windows. Deleting the folder can free up the space of disk C.

An ECS running Windows Server 2008 R2 Standard 64-bit is used as an example to describe how to delete system logs.

1. Open the **C:\Windows\System32\LogFiles** folder, and delete all the files and folders in it.

   C:\Windows\System32\LogFiles

1. Choose **Start**, right-click **Computer**, and choose **Manage** in the shortcut menu.

2. In the displayed window, choose **Diagnostics** > **Event Viewer** > **Windows Logs** and delete logs of **Application**, **Security**, **Setup**, **System**, and **Forwarded Events**.
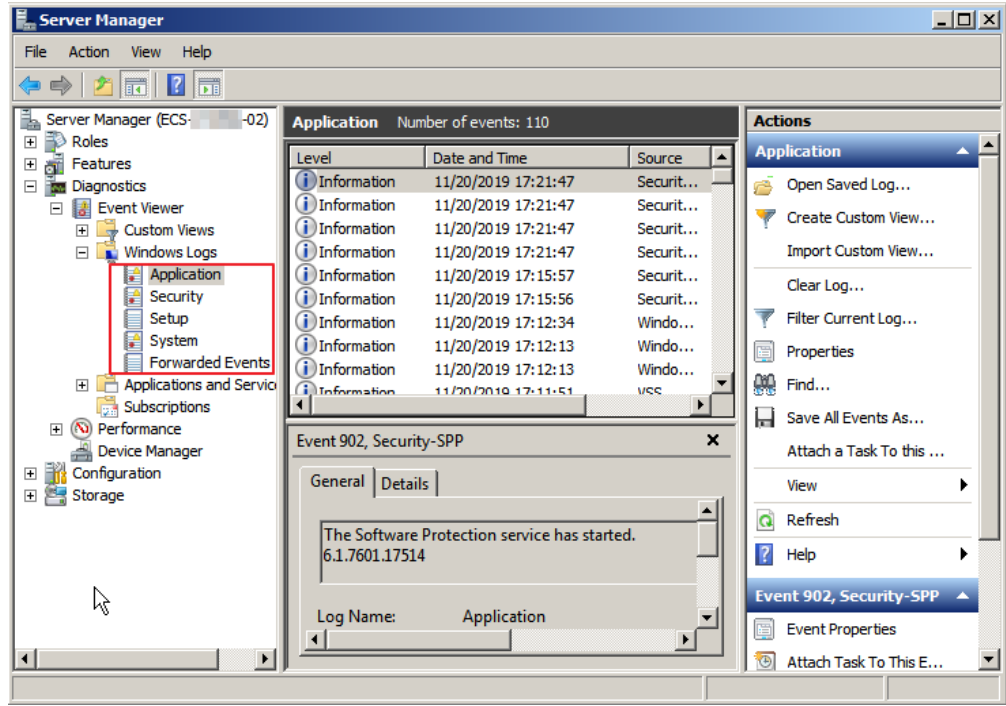
4 Converting the Image Format

4.1 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 Huawei Cloud. 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 4-1 Tool and costs

Tool	Description	Costs
qemu-img	**qemu-img** is an open-source tool for converting image formats. You can obtain it from: **https://qemu.weilnetz.de/w64/**	Free

Constraints

- **qemu-img** supports the mutual conversion of image formats VHD, VMDK, QCOW2, RAW, VHDX, QCOW, VDI, and QED.
- ZVHD and ZVHD2 are self-developed image file formats and cannot be identified by **qemu-img**. 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

  <p>📖 NOTE</p>

  > 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.

      <p>📖 NOTE</p>

      > 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

📖 **NOTE**

If any error occurs, see "Appendix 2: Common qemu-img-hw Errors" in **Converting the Image Format Using qemu-img-hw**.

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
```

4.2 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 Huawei Cloud. 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 4-2 Tool and costs

Tool	Description	Costs
qemu-img-hw	**qemu-img-hw** is developed by Huawei and 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 4-3 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/qemu-img-hw**.

cd /usr/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 qemu-img-hw Errors

- 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.

- Symptom:

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

  `./qemu-img-hu: error while loading shared libraries: libcrypto.so.10: cannot open shared object file: No such file or directory`

  Solution: Run **openssl version** to check whether the OpenSSL version is later than 1.0. If yes, install OpenSSL 1.0.

  Run the following commands to install OpenSSL 1.0:

  wget https://github.com/openssl/openssl/releases/download/OpenSSL_1_0_2k/openssl-1.0.2k.tar.gz

  tar -xvf openssl-1.0.2k.tar.gz

  cd openssl-1.0.2k

  ./config -d shared --prefix=/tmp/openssl (custom installation directory)

  make

  make install

cp /tmp/openssl/lib/libcrypto.so.1.0.0 /usr/lib64/libcrypto.so.10

Delete the **/usr/lib64/libcrypto.so.10** file if you do not need it any longer.

5 Creating a Private Image Using Packer

Packer is an open-source tool for creating custom images. Packer's behavior is determined by a template which is a HashiCorp Configuration Language (HCL2) or JSON file and mainly contains three component types: builder, provisioner, and post-processor. These components can be flexibly combined in the template to automatically create image files for multiple cloud platforms in parallel. Packer simplifies private image creation by using manageable code. In this way, you can flexibly customize images and easily use identical images between different cloud platforms.

This section describes how to use Packer to create an Ubuntu 22.04 Server 64-bit private image on a CentOS 8.2 64-bit ECS.

Constraints

Full-ECS images cannot be used as an image source when you use Packer to create private images.

Procedure

Install Packer

1. Log in to the management console, create an ECS (for example, an ECS running CentOS 8.2 64-bit), and bind an EIP to the ECS.

2. Log in to the ECS.

3. Visit **https://www.packer.io/downloads** and download the latest version of Packer that matches the ECS OS and architecture.

4. Run the following command to install Packer (**packer_1.9.1_linux_amd64.zip** as an example):

   wget --no-check-certificate https://releases.hashicorp.com/packer/1.9.1/packer_1.9.1_linux_amd64.zip

> **NOTE**
> - Ensure that you have bound an EIP to the ECS so that the ECS can access the Internet.
> - If message "command not found" is displayed, the wget tool is not installed. Run the **yum install wget** command to install it.

5. Run the following command to decompress the Packer installation package:

   unzip packer_1.9.1_linux_amd64.zip

6. Run the following command to move the Packer installation package to the **/usr/local/bin** directory:

   mv packer /usr/local/bin

   > **NOTE**
   > The **/usr/local/bin** directory has been added to environment variables. You can also move the Packer installation package to another directory that has been added to environment variables.

7. Run the following command to check whether Packer is installed successfully:

   packer -v

   – If the command output contains the Packer version number, Packer is installed successfully.

   – If "command not found" is displayed, the Packer installation failed. Check whether the directory where Packer resides has been added to environment variables.

   > **NOTE**
   > Run the **env | grep PATH** command to check whether the environment variable **PATH** contains the Packer installation directory.
   >
   > If **PATH** does not contain the Packer installation directory, perform the following operations to add it to **PATH**:
   >
   > 1. Run the following command to open the **profile** file:
   >
   >   **vim /etc/profile**
   >
   > 2. Press **i** to enter the editing mode and add **export PATH=$PATH:/usr/local/bin** to the end of the file.
   >
   >   If your Packer resides in a directory other than **/usr/local/bin**, replace **/usr/local/bin** in the command with the actual directory.
   >
   > 3. Press **Esc** to exit the editing mode. Enter **:wq** and press **Enter** to save the changes and exit.
   >
   > 4. Run the following command to make the change take effect:
   >
   >   **source /etc/profile**

Create a Packer Template

To create an image using Packer, you need a template in .pkr.hcl format. In the template, you need to configure **builders**, **provisioners**, and post-processors. In a provisioner, you can modify the source image as you need. For example, you can modify the configurations and specify the software to be pre-installed. In this example, a post-processor is used to redirect the output path of manifest. If your Packer template has multiple builders, you can locate the ID of the image created by each builder based on the manifest output. For more information about builders, provisioners, and post-processors, see the **official Packer documentation**.

This section takes a Shell provisioner as an example.

1. Run the following command to create a **hwcloud.pkr.hcl** file:

    touch hwcloud.pkr.hcl

2. Run the following command to open the **hwcloud.pkr.hcl** file:

    vim hwcloud.pkr.hcl

3. Press **i** to enter the editing mode and edit the template based on the site requirements. For parameter details, see **Table 5-1**. The following content is for reference only.

```
packer {
  required_plugins {
    huaweicloud = {
      version = ">= 1.0.0"
      source  = "github.com/huaweicloud/huaweicloud"
    }
  }
}

source "huaweicloud-ecs" "artifact" {
  region           = "xxx"
  availability_zone = "xxx"
  flavor           = "c6.large.2"
  source_image_name = "Ubuntu 22.04 server 64bit"
  image_name       = "Ubuntu-2204-image-powered-by-Packer"
  image_tags = {
    builder = "packer"
    os      = "Ubuntu-22.04-server"
  }

  ssh_username      = "root"
  eip_type          = "5_bgp"
  eip_bandwidth_size = 5
}

build {
  sources = ["source.huaweicloud-ecs.artifact"]

  provisioner "shell" {
    inline = ["apt-get update -y"]
  }

  post-processor "manifest" {
    strip_path = true
    output     = "packer-result.json"
  }
}
```

◻ **NOTE**

In **Table 5-1**, **region**, **availability_zone**, **flavor**, and **source_image_name** are the attributes of the temporary ECS that will be used to create the private image.

Table 5-1 Packer template parameters

Parameter	Description	Mandatory
region	Region name.	Yes
flavor	Flavor of the temporary ECS.	Yes
image_name	Name of the private image to be created.	Yes

Parameter	Description	Mandatory
image_tags	Tags of the private image to be created.	No
availability_zone	AZ.	No
source_image_name	Name of the source image. This is the image whose base OS you intend to build on top of. You can obtain the source image name from the public image list on the IMS console. **NOTE** To specify a source image, you can also use **source_image** to specify an image ID or use **source_image_filter** to filter a source image.	No
ssh_username	SSH login username of the private image to be created. This parameter is mandatory for SSH login.	No
eip_type	EIP line type.	No
eip_bandwidth_size	EIP bandwidth (Mbit/s). This parameter is mandatory if access to the Internet is required. **NOTE** **eip_type** and **eip_bandwidth_size** are configured to create a new EIP. To use an existing EIP, configure **reuse_ips** and **floating_ip**.	No
provisioner	Provisioner type. For details, see **Packer Provisioners**.	Yes
post-processor	Post-processor type.	No

📖 **NOTE**

For more parameters, see **https://github.com/huaweicloud/packer-plugin-huaweicloud/wiki**.

Create a Private Image Using Packer

1. After the Packer template is created, run the following command to import AK/SK:

   export HW_ACCESS_KEY=<AccessKey ID>

   export HW_SECRET_KEY=<AccessKey Secret>

2. Run the following command to create an image:

packer build hwcloud.pkr.hcl

```
huaweicloud-ecs.artifact: output will be in this color.

==> huaweicloud-ecs.artifact: Loading availability zones...
    huaweicloud-ecs.artifact: the specified availability_zone ap-southeast-1a is available
==> huaweicloud-ecs.artifact: Loading flavor: c6.large.2
==> huaweicloud-ecs.artifact: Creating temporary keypair: packer_64abc4fd-xxxx-xxxx-
xxxx-2139eee76819...
==> huaweicloud-ecs.artifact: Created temporary keypair: packer_64abc4fd-xxxx-xxxx-
xxxx-2139eee76819
    huaweicloud-ecs.artifact: Found Image ID: 19d9079e-xxxx-xxxx-xxxx-642116ad6557
==> huaweicloud-ecs.artifact: Creating temporary VPC...
    huaweicloud-ecs.artifact: temporary VPC ID: 6e309adc-xxxx-xxxx-xxxx-4c3356b972c7
==> huaweicloud-ecs.artifact: Creating temporary subnet...
    huaweicloud-ecs.artifact: temporary subnet ID: 66ab3bc3-xxxx-xxxx-xxxx-d8ca2d9378cc
    huaweicloud-ecs.artifact: the [default] security groups will be used ...
==> huaweicloud-ecs.artifact: Creating EIP ...
    huaweicloud-ecs.artifact: Created EIP: '0cd696e3-xxxx-xxxx-xxxx-220b8d277604' (159.xxx.xxx.180)
==> huaweicloud-ecs.artifact: Launching server in AZ ap-southeast-1a...
    huaweicloud-ecs.artifact: Waiting for server to become ready...
    huaweicloud-ecs.artifact: Server ID: 13f78f88-xxxx-xxxx-xxxx-7e8c27a9ad1f
==> huaweicloud-ecs.artifact: Using SSH communicator to connect: 159.138.141.180
==> huaweicloud-ecs.artifact: Waiting for SSH to become available...
==> huaweicloud-ecs.artifact: Connected to SSH!
==> huaweicloud-ecs.artifact: Provisioning with shell script: /tmp/packer-shell2456008753
huaweicloud-ecs.artifact: Hit:1 http://repo.huaweicloud.com/ubuntu jammy InRelease
        ......
    huaweicloud-ecs.artifact: Fetched 5,536 kB in 3s (2,006 kB/s)
    huaweicloud-ecs.artifact: Reading package lists...
==> huaweicloud-ecs.artifact: Stopping server: 13f78f88-xxxx-xxxx-xxxx-7e8c27a9ad1f ...
    huaweicloud-ecs.artifact: Waiting for server to stop: 13f78f88-xxxx-xxxx-xxxx-7e8c27a9ad1f ...
==> huaweicloud-ecs.artifact: Creating the system image: Ubuntu-2204-image-powered-by-Packer ...
    huaweicloud-ecs.artifact: Image: 62dc6e37-xxxx-xxxx-xxxx-a2a00a677f5b
==> huaweicloud-ecs.artifact: Terminating the source server: 13f78f88-xxxx-xxxx-xxxx-7e8c27a9ad1f...
==> huaweicloud-ecs.artifact: Deleted temporary public IP '0cd696e3-xxx-xxxx-xxxx-220b8d277604'
(159.xxx.xxx.180)
==> huaweicloud-ecs.artifact: Deleting temporary subnet: 66ab3bc3-xxxx-xxxx-xxxx-d8ca2d9378cc...
==> huaweicloud-ecs.artifact: Deleting temporary VPC: 6e309adc-xxxx-xxxx-xxxx-4c3356b972c7...
==> huaweicloud-ecs.artifact: Deleting temporary keypair: packer_64abc4fd-xxxx-xxxx-
xxxx-2139eee76819 ...
==> huaweicloud-ecs.artifact: Running post-processor:  (type manifest)
Build 'huaweicloud-ecs.artifact' finished after 5 minutes 48 seconds.

==> Wait completed after 5 minutes 48 seconds

==> Builds finished. The artifacts of successful builds are:
--> huaweicloud-ecs.artifact: An image was created: 62dc6e37-xxxx-xxxx-xxxx-a2a00a677f5b
```

3. Log in to the management console and click **Image Management Service** under **Compute**.

4. Click the **Private Images** tab and view the image created using Packer.

Helpful Links

Packer official guide: **https://www.packer.io/intro/getting-started/install.html**

6 Configuring an ISO File as a Local Image Source

Context

When you install software on a Linux ECS, the network may be disconnected or resources on the network may be invalid, resulting in software installation failures. In this case, you can configure an ISO file as a local image source to install the software.

Package Managers

Before configuring a local source, you need to determine the package manager to be used. Generally, there are three types of package managers: yum, apt, and zypper.

- yum is for RHEL-based OSs: RHEL, CentOS, EulerOS, and Fedora.
- apt is for Debian and Ubuntu.
- zypper is for SUSE and openSUSE.

Configuring a Local Image Source

Configure a local image source by following the instructions in **yum**, **apt**, or **zypper**.

- yum

  a. Upload the ISO file to the ECS and mount it to the **/mnt** directory.

  mount *XXX***.iso /mnt**

  b. Enter the **/etc/yum.repo.d** directory where the yum configuration file is stored and back up all **.repo** files. Then, create a **.repo** file, for example **local.repo**. Add the following information to the **local.repo** file:

  ```
  [rhel-local]
  name=local
  baseurl=file:///mnt
  enabled=1
  gpgcheck=0
  ```

> ◻ NOTE
>
> The **/mnt** directory specified in the configuration file must be the same as the mounting directory of the ISO file.

    c.   Clear yum.

        yum clean all

    d.   Generate a new cache.

        yum makecache

- apt

    a.   Upload the ISO file to the ECS and mount it to the **/mnt** directory.

        mount *XXX*.**iso /mnt**

    b.   Add the **apt cdrom** source.

        apt-cdrom -m -d /mnt/ add

    c.   View the added source in the configuration file.

        cat /etc/apt/sources.list

    d.   Update the source.

        apt-get update

- zypper

    a.   Upload the ISO file to the ECS.

    b.   Add the ISO file as the source.

        sudo zypper addrepo iso:/?iso=*/media/SOFTWARE/openSUSE-11.4-DVD-i586.iso DVDISO*

        In the preceding command:

        ▪   **/media/SOFTWARE/openSUSE-11.4-DVD-i586.iso** is the location of the ISO file.

        ▪   **DVDISO** is the source alias.

    c.   Check whether the source is successfully added.

        zypper repos

    d.   Refresh the source.

        zypper refresh

Examples

The operations in **Configuring a Local Image Source** may be different depending on the OS version. Basically, you need to add the source and refresh it. Debian 10.1.0 and CentOS 8.0 are used as examples to describe how to add a local source.

- Debian 10.1.0

Run the **cat /etc/apt/sources.list** command to check whether the **sources.list** file contains a default cdrom source.

Figure 6-1 Viewing the source

The source directs to the CD-ROM drive **/dev/cdrom**. Debian 10.1.0 provides a soft link to link the CD-ROM drive to **/media/cdrom**.

Figure 6-2 Checking the media directory

Therefore, mount the ISO file to **/media/cdrom**.

- CentOS 8.0

  a. Mount the ISO file to the **/mnt** directory.

  b. Rename all source files except **CentOS-Media.repo** in the **/etc/yum.repo.d** directory as .bak files or move them to another directory.

  c. Modify the **CentOS-Media.repo** file.

  Figure 6-3 Modifying the CentOS-Media.repo file

  The modifications in the red box are as follows:

  - Set **baseurl** to **file:///mnt/BaseOS** and **file:///mnt/AppStream**. **mnt** is the mounting directory of the ISO file. Delete invalid paths from the default configuration. Otherwise, a checksum error will occur.

  - Change the value of **gpgcheck** to **0**, indicating that the check is not performed.

- Change the value of **enabled** to **1** for the configurations to take effect.

d. Clear yum and generate a new cache.

yum clean all && yum makecache

# 7 Migrating Service Data Across Accounts (Data Disks)

Scenarios

Generally, service data is stored on data disks. To migrate the data across accounts, you need to create data disk images and share them with the target account. This section uses Linux as an example to describe how to migrate service data (only data disks) across accounts in the same region.

Cross-Account Service Data Migration (Data Disks)

To migrate service data stored on a data disk across accounts, create an image from the data disk, share the image with the target account. The target account accepts the shared image and attaches the new data disk created from the shared image to an existing or new ECS.

Figure 7-1 Migration process
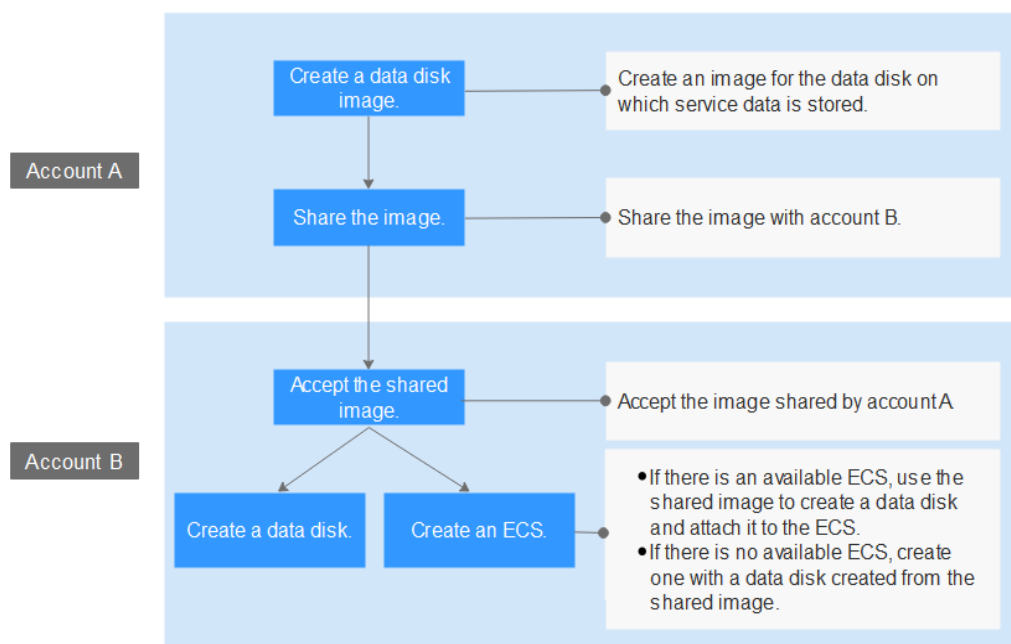
1. **Create a data disk image**
2. **Share the image with the target account**
3. **Accept the shared image**
4. **Create a data disk or an ECS**

Step 1: Create a Data Disk Image

Assume that **qianyi.txt** is stored on the data disk of your ECS and you want to migrate the file to another account.

```
[root@ecs-disk-image-test data]# cat qianyi.txt
This is a test.
[root@ecs-disk-image-test data]# df -TH
Filesystem      Type      Size  Used Avail Use% Mounted on
devtmpfs        devtmpfs  2.0G     0  2.0G   0% /dev
tmpfs           tmpfs     2.0G     0  2.0G   0% /dev/shm
tmpfs           tmpfs     2.0G  9.0M  2.0G   1% /run
tmpfs           tmpfs     2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/vda1       ext4       43G  2.2G   38G   6% /
tmpfs           tmpfs     398M     0  398M   0% /run/user/0
/dev/vdb1       ext4      106G   63M  101G   1% /mnt/data
[root@ecs-disk-image-test data]#
```

1. Log in to the management console and switch to the CN-Hong Kong region.
2. Under **Service List**, choose **Compute** > **Image Management Service**.

   The **Image Management Service** page is displayed.
3. In the upper right corner, click **Create Image**.

   The **Create Image** page is displayed.
4. Set parameters.

   Figure 7-2 Creating a data disk image
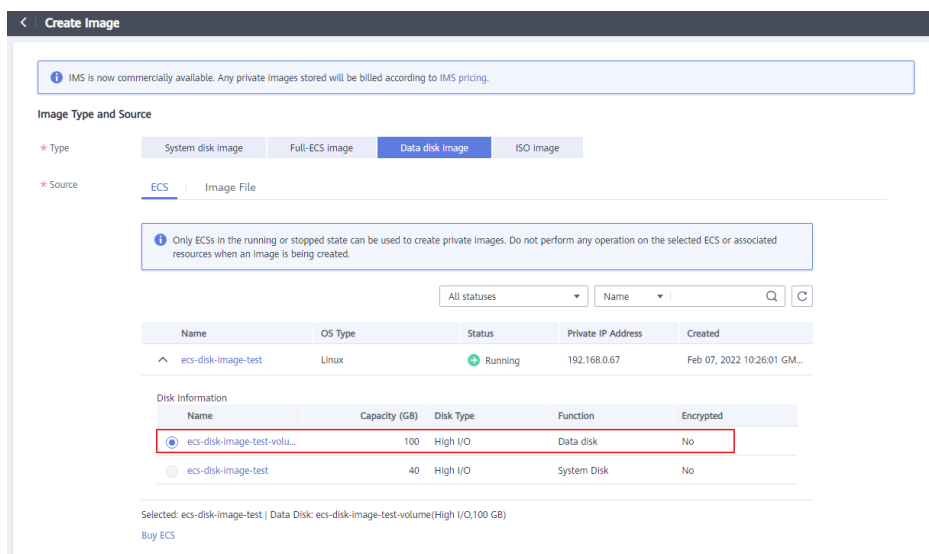
   – **Type**: Select **Create Image**.

- **Image Type**: Select **Data disk image**.
- **Source**: Select **ECS** and then select the **data disk ecs-disk-image-test-volume** data disk.
- **Name**: Enter a name for the data disk image, for example, **disk-image-test**.
- **Enterprise Project**: Select **default**.

5. Click **Next**.

6. Confirm the settings, read and agree to the agreement, and click **Submit**.

7. The system redirects to the private image list. Wait for several minutes and check whether the data disk image is successfully created.

Figure 7-3 Viewing private images

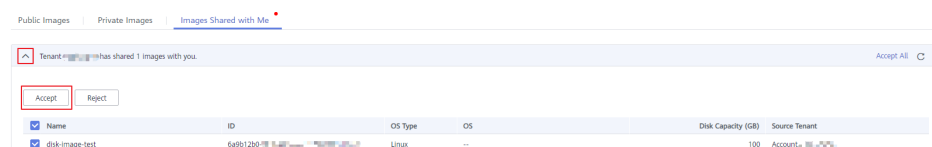|	Name ↓≡	Status	OS Type	OS	Image Type	Disk Capacity (GB)	Encrypted
☐	disk-image-test	✔ Normal	Linux	--	Data disk image	100	No

Step 2: Share the Image with the Target Account

Share the data disk image created in **Step 1: Create a Data Disk Image** with the target account. Before the image sharing, obtain the project ID of the target account. (You can obtain the project ID from **My Credentials**.)

1. Locate the row that contains the **disk-image-test** private image. Choose **More** > **Share** in the **Operation** column.

   The **Share Image** dialog box is displayed.

2. In the **Share Image** dialog box, enter the project ID of the target account.

3. Click **OK**.

Step 3: Accept the Shared Image

Accept the shared data disk image.

1. Log in to the management console using the account the image is shared with and switch to the EU-Dublin region.

2. Under **Service List**, choose **Compute** > **Image Management Service**. Then, click the **Images Shared with Me** tab.

3. Select **disk-image-test** and click **Accept**.

Figure 7-4 Accepting a shared image

Public Images	Private Images	Images Shared with Me

∧ Tenant ▓▓ has shared 1 images with you.						Accept All ⟳

Accept	Reject

☑	Name	ID	OS Type	OS	Disk Capacity (GB)	Source Tenant
☑	disk-image-test	6a9b12b0▓▓ ▓▓ ▓▓	Linux	--	100	Account▓ ▓ ▓▓

After the image is accepted, it is displayed in the shared image list.

Step 4: Create a Data Disk or an ECS

Use the shared image to create a new data disk and attach it to an existing ECS. Alternatively, create an ECS with a data disk created from the shared image. Then, check whether the service data is successfully migrated.

●   Create a new data disk and attach it to an existing ECS.

a.   Locate the row that contains the shared image **disk-image-test**, and click **Create Data Disk** in the **Operation** column.

Figure 7-5 Creating a data disk

The page for purchasing EVS disks is displayed.

b.   Configure the billing mode and disk specifications as needed. The AZ must be the same as that of the ECS to which the data disk will be attached. Click **Next**.

c.   Return to the EVS disk list. Wait for several minutes until the EVS disk is created successfully.

d.   Locate the row that contains the new EVS disk and click **Attach** in the **Operation** column to attach the data disk to the ECS.

e.   Log in to the ECS and check whether the service data is successfully migrated.

Run the **fdisk -l** command. The command output shows that the new data disk has been partitioned.
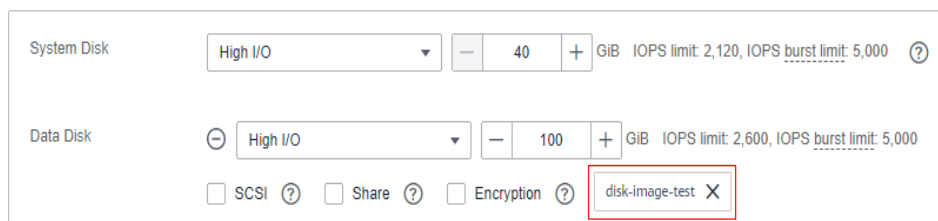

Mount the new partition to a directory of the ECS and check the **qianyi.txt** file. The file content is properly printed, which means that the service data migration is successful.

```
[root@ecs-f04d ~]# cd /mnt
[root@ecs-f04d mnt]# ls
[root@ecs-f04d mnt]# mkdir data
[root@ecs-f04d mnt]# ls
data
[root@ecs-f04d mnt]# mount /dev/vdb1 /mnt/data
[  170.467473] EXT4-fs (vdb1): recovery complete
[  170.468655] EXT4-fs (vdb1): mounted filesystem with ordered data mode. Opts: (null)
[root@ecs-f04d mnt]# cd data
[root@ecs-f04d data]# ls
lost+found  qianyi.txt
[root@ecs-f04d data]# cat qianyi.txt
This is a test.
```

- Create an ECS with a data disk attached.

    a. Under **Service List**, choose **Compute** > **Elastic Cloud Server**.

    b. In the upper right corner, click **Buy ECS**.

        The page for purchasing ECSs is displayed.

    c. Configure the billing mode, AZ, and specifications as needed and add a data disk which will be created from the shared data disk image. Complete the ECS creation as instructed.

        Figure 7-6 Adding a data disk

    d. Wait for several minutes and check whether the new ECS is displayed in the ECS list.

    e. Log in to the new ECS and check whether the service data is successfully migrated.

        Run the **fdisk -l** command. The command output shows that the new data disk has been partitioned. Mount the new partition to a directory of the ECS and check the **qianyi.txt** file. The file content is properly printed, which means that the service data migration is successful.

```
[root@ecs-38e4 ~]# fdisk -l

Disk /dev/vda: 42.9 GB, 42949672960 bytes, 83886080 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000f1217

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

Disk /dev/vdb: 107.4 GB, 107374182400 bytes, 209715200 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x7abde66f

   Device Boot      Start         End      Blocks   Id  System
/dev/vdb1            2048   209715199   104856576   83  Linux
[root@ecs-38e4 ~]# mkdir /mnt/data
[root@ecs-38e4 ~]# mount /dev/vdb1 /mnt/data
[ 1177.739741] EXT4-fs (vdb1): recovery complete
[ 1177.741032] EXT4-fs (vdb1): mounted filesystem with ordered data mode. Opts: (null)
[root@ecs-38e4 ~]# cd /mnt/data
[root@ecs-38e4 data]# ls
lost+found  qianyi.txt
[root@ecs-38e4 data]# cat qianyi.txt
This is a test.
```