

Huawei Cloud EulerOS

FAQs

Issue 01
Date 2024-06-30



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

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

Trademarks and Permissions



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

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

Notice

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

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

Contents

1 What Do I Do If CentOS Linux Is No Longer Maintained?.....	1
2 Does Huawei Cloud Have a Migration Solution for CentOS?.....	4
3 How Do I Install the MLNX Driver?.....	6
4 How Do I Enable SELinux on an ECS Running HCE OS?.....	9
5 How Do I Change the OS Name on the Console After the OS Is Migrated?.....	11
6 How Are Huawei Cloud EulerOS, openEuler, and EulerOS Different from One Another?.....	14
7 How Do I Enable WireGuard in Kernel and Install wireguard-tools?.....	15
8 How Do I Save the User Credential Information for Logging In to Docker Like What Docker CE Does?.....	17
9 What Is OOM? Why Does OOM Occur?.....	18
10 How Do I Handle IPVS Errors?.....	26
11 Why Can't HCE OS Obtain an IPv6 Address After IPv6 Is Enabled for an ECS?...	28

1 What Do I Do If CentOS Linux Is No Longer Maintained?

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

Background

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

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

Impacts

CentOS Linux users will be affected as follows:

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

Huawei Cloud users will be affected as follows:

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

Solution

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

For details about how to change to Debian or Ubuntu, see [Changing the OS](#).

- Change CentOS Linux to one of those listed in [Table 1-2](#).
If you want to change the ECS OS and the software is loosely coupled with the OS, change it. This does not affect the ECS configurations (such as NICs, disks, and VPNs).
 - For details about how to change to Huawei Cloud EulerOS, see [Changing an OS to Huawei Cloud EulerOS](#).
 - For details about how to change to Debian or Ubuntu, see [Changing the OS](#).
- Migrate CentOS Linux to Huawei Cloud EulerOS.
If you want to change the OS but retain OS parameter settings, migrate the OS to Huawei Cloud EulerOS. This does not affect the ECS configurations (such as NICs, disks, and VPNs).
For details, see [Migrating an OS to Huawei Cloud EulerOS](#).

The following table describes the differences between the two methods.

Table 1-1 Differences between OS change and OS migration

Item	Changing the OS	Migrating the OS
Data backup	<ul style="list-style-type: none">• Data in all partitions of the system disk will be cleared, so you are advised to back up the system disk data prior to an OS change.• Data in data disks remains unchanged.	<ul style="list-style-type: none">• System disk data is not cleared, but you are still advised to back up the system disk data to prevent any exception in system software.• Data in data disks remains unchanged.
Custom settings	After the OS is changed, custom settings such as DNS and hostname will be reset and need to be reconfigured.	After the OS is migrated, custom settings such as DNS and hostname do not need to be reconfigured.

Table 1-2 Available OSs

OS	Description	Intended User
Huawei Cloud EulerOS	Huawei Cloud EulerOS (HCE OS) is an openEuler-based cloud operating system. HCE OS offers cloud native, high-performing, secure, and easy-to-migrate capabilities. This accelerates service migration to the cloud and promotes application innovation. You can use HCE OS to replace operating systems such as CentOS and EulerOS.	Individuals or enterprises that want to continue to use free images in an open source community
Debian and Ubuntu	They are Linux distributions that differ in use and compatibilities.	Individuals or enterprises that can afford the OS change costs

2 Does Huawei Cloud Have a Migration Solution for CentOS?

Background

CentOS Linux 8 ended on December 31, 2021, and CentOS Linux 7 will end on June 30, 2024. CentOS will no longer support new software and patch updates. CentOS services may be exposed to risks or even become unavailable. What's worse, they cannot be restored in a timely manner.

Huawei Cloud EulerOS can be a perfect alternative to CentOS. You can use our migration tool to easily migrate OSs such as CentOS or EulerOS to Huawei Cloud EulerOS. Additionally, you can benefit from professional services in cloud native hybrid deployment, security hardening, fast migration, efficient O&M, and professional certification.

Compatibility Evaluation

Technologically, Huawei Cloud EulerOS can replace CentOS. Huawei Cloud EulerOS is fully independent and controllable, and continuously evolves with astute contributions from the openEuler open-source community. Huawei Cloud EulerOS can work with 400 types of boards in the southbound direction, including mainstream compute products. In the northbound direction, Huawei Cloud EulerOS is 100% compatible with applications in mainstream application scenarios, such as cloud native, storage, database, big data, and web. More than 5,000 types of applications have passed the compatibility certification on Huawei Cloud EulerOS and can be alternatives to those running on CentOS.

To ensure the seamless migration from CentOS to Huawei Cloud EulerOS, you can use the compatibility evaluation tool to quickly scan the software to determine whether they are compatible with Huawei Cloud EulerOS.

For compatible software, the software configuration is not modified during the migration and does not need to be reconfigured after the migration. For incompatible software, the evaluation report provides workarounds for adaptation after the migration.

Migration Feasibility

Huawei Cloud has mature migration guides for the following types of applications:

- Distributed clustered applications, such as big data and distributed storage. CentOS-to-EulerOS migration does not interrupt services, thanks to distributed software scaling.
- Active/standby applications, such as databases. CentOS-to-EulerOS migration does not interrupt services. The standby application will be first migrated, followed by the active application. The active-to-standby synchronization allows for seamless migration.
- Standalone applications: Services need to be interrupted during CentOS migration. This migration solution is mature and proven. It works like re-deploying the application on Huawei Cloud EulerOS.

Contact Us

Professional Huawei Cloud engineers are available 24/7 to provide the help and support you need in the event that you experience an issue. If you encounter any issue while using Huawei Cloud, you can [submit a service ticket](#).

3 How Do I Install the MLNX Driver?

Install the MLNX driver on HCE OS 2.0 (including x86 and Arm).

Constraints

- The kernel of HCE OS 2.0 is Linux 5.10 or later.
- The CX6 NIC driver version is 23.10-1.1.9.0-LTS or later.

Prerequisites

HCE OS 2.0 with kernel 5.10 or later has been installed.

Installing the MLNX Driver for the x86 Architecture

1. Download the CX6 NIC driver installation package
[MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-x86_64.tgz](#).
2. Decompress the package and go to the working directory.

```
tar -xf MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-x86_64.tgz  
cd MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-x86_64
```
3. Install the CX6 NIC driver.

```
./mlnxofedinstall --basic --without-depcheck --distro OPENEULER22.03 --  
force --kernel 5.10.0-60.18.0.50.oe2203.x86_64 --kernel-sources /lib/  
modules/$(uname -r)/build
```

NOTE

`5.10.0-60.18.0.50.oe2203.x86_64` is the kernel version when the official MLNX_OFED package is compiled.

4. Create a link.

```
ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.x86_64/extra/mlnx-  
ofa_kernel /lib/modules/$(uname -r)/weak-updates/  
ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.x86_64/extra/kernel-mft /lib/  
modules/$(uname -r)/weak-updates/  
depmod -a
```
5. Run `reboot` to restart the OS.
6. Run the `/etc/init.d/openibd status` command to check the driver installation result.

If the following information is displayed, the driver is installed:

```
[root@lnmp ~]# /etc/init.d/openibd status

HCA driver loaded

The following OFED modules are loaded:

rdma_ucm
rdma_cm
ib_ipoib
mlx5_core
mlx5_ib
ib_uverbs
ib_umad
ib_cm
ib_core
mlx5fw

[root@lnmp ~]# █
```

Installing the MLNX Driver for the Arm Architecture

1. Download the CX6 NIC driver installation package
[MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-aarch64.tgz](#).
2. Decompress the package and go to the working directory.
tar -xf MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-aarch64.tgz
cd MLNX_OFED_LINUX-23.10-1.1.9.0-openeuler22.03-aarch64
3. Install the CX6 NIC driver.
./mlnxofedinstall --basic --without-depcheck --distro OPENEULER22.03 --force --kernel 5.10.0-60.18.0.50.oe2203.aarch64 --kernel-sources /lib/modules/\$(uname -r)/build

📖 NOTE

5.10.0-60.18.0.50.oe2203.aarch64 is the kernel version when the official MLNX_OFED package is compiled.

4. Create a link.
ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.aarch64/extra/mlnx-ofa_kernel /lib/modules/\$(uname -r)/weak-updates/
ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.aarch64/extra/kernel-mft /lib/modules/\$(uname -r)/weak-updates/
depmod -a
5. Run **reboot** to restart the OS.
6. Run the **/etc/init.d/openibd status** command to check the driver installation result.

If the following information is displayed, the driver is installed:

```
[root@lnmp ~]# /etc/init.d/openibd status  
  
HCA driver loaded  
  
The following OFED modules are loaded:  
  
rdma_ucm  
rdma_cm  
ib_ipoib  
mlx5_core  
mlx5_ib  
ib_uverbs  
ib_umad  
ib_cm  
ib_core  
mlxfw  
  
[root@lnmp ~]# █
```

4 How Do I Enable SELinux on an ECS Running HCE OS?

By default, SELinux is disabled on HCE OS. You can enable SELinux as needed.

⚠ CAUTION

Do not run the `/etc/selinux/config` command to enable SELinux. If you enable SELinux by running this command, login may fail.

Procedure

1. Open the configuration file `/boot/grub2/grub.cfg` and delete `selinux=0`.
2. Run the `touch /.autorelabel` command.

The `/.autorelabel` file triggers the OS to relabel all files on the disk during startup. This process may take several minutes. After the relabel operation is complete, the OS automatically restarts for the operation to take effect and deletes the `/.autorelabel` file to ensure that the relabel operation will not be performed again.

3. Open the configuration file `/etc/selinux/config`, set `SELINUX` to `permissive`, and run the `reboot` command to restart the OS.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

4. Open the configuration file `/etc/selinux/config`, set `SELINUX` to `enforcing`, and run the `reboot` command to restart the OS.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=enforcing
# SELINUXTYPE= can take one of three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

5. Run the **getenforce** command to check the SELinux status.
If **Enforcing** is displayed, SELinux is enabled.

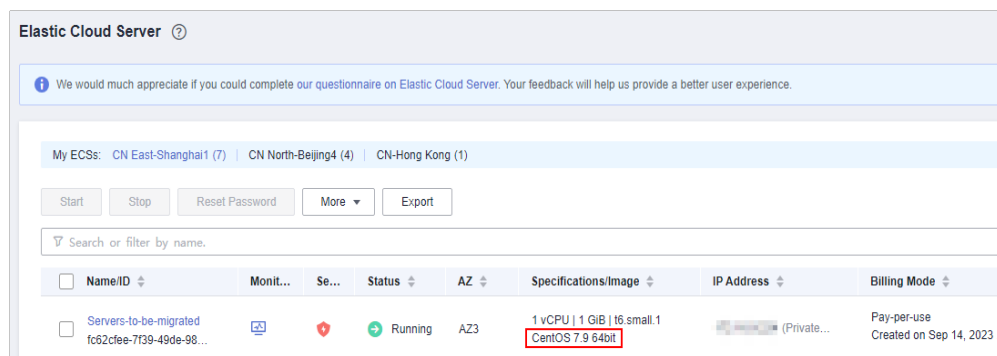
```
[root@localhost ~]#
[root@localhost ~]# getenforce
Enforcing
```

5 How Do I Change the OS Name on the Console After the OS Is Migrated?

Background

After the original OS (for example, CentOS 7.9) is migrated to Huawei Cloud EulerOS, the existing OS name CentOS 7.9 rather than Huawei Cloud EulerOS is displayed on the console.

You can create a private image and then switch to the created private image to change the OS name to Huawei Cloud EulerOS.



Procedure

1. Log in to the [ECS console](#).
2. Locate the ECS whose OS is to be migrated and choose **More > Manage Image > Create Image** in the **Operation** column.
3. On the **Create Image** page, configure the following parameters:
 - **Region:** Retain the default value.
 - **Type:** Retain the default value.
 - **Image Type:** Retain the default value.
 - **Source:** Retain the default value.
 - **Name:** Enter an image name that is easy to identify.
 - **Agreement:** Read the agreements and select the **Agreement** option.

i IMS is now in commercial use. Any private images stored will be billed according to IMS pricing. x

Image Type and Source

* Region: CN East-Shanghai1

Regions are geographic areas isolated from each other. Resources are region-specific and cannot be used across regions through internal network connections. For low network latency and quick resource access, select the nearest region.

* Type: Create Image Import Image

* Image Type: System disk image Full-ECS image Data disk image

* Source: ECS BMS

- Only ECSs in the running or stopped state can be used to create private images.
- Before creating an image, configure and optimize the ECS. Ensure Cloud-Init is installed if the ECS runs Linux and Cloudbase-Init is installed if the ECS runs Windows. [Learn more](#)
- Do not perform any operation on the selected ECS or associated resources when an image is being created.

All statuses ID fc62cfee-7f39-49de-985 x Q C

Name	OS	Status	Private IP Address	Created
v ● Servers-to-be-migrated	CentOS 7.9 64bit	Running	192.168.1.1	Sep 14, 2023 11:16:25 G...

Selected: Servers-to-be-migrated | OS: CentOS 7.9 64bit | System Disk: General Purpose SSD | 40 GiB
[Buy ECS](#)

Image Information

Encryption: Unencrypted ?

* Name:

* Enterprise Project: --Select-- C ?

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

You can add 10 more tags.

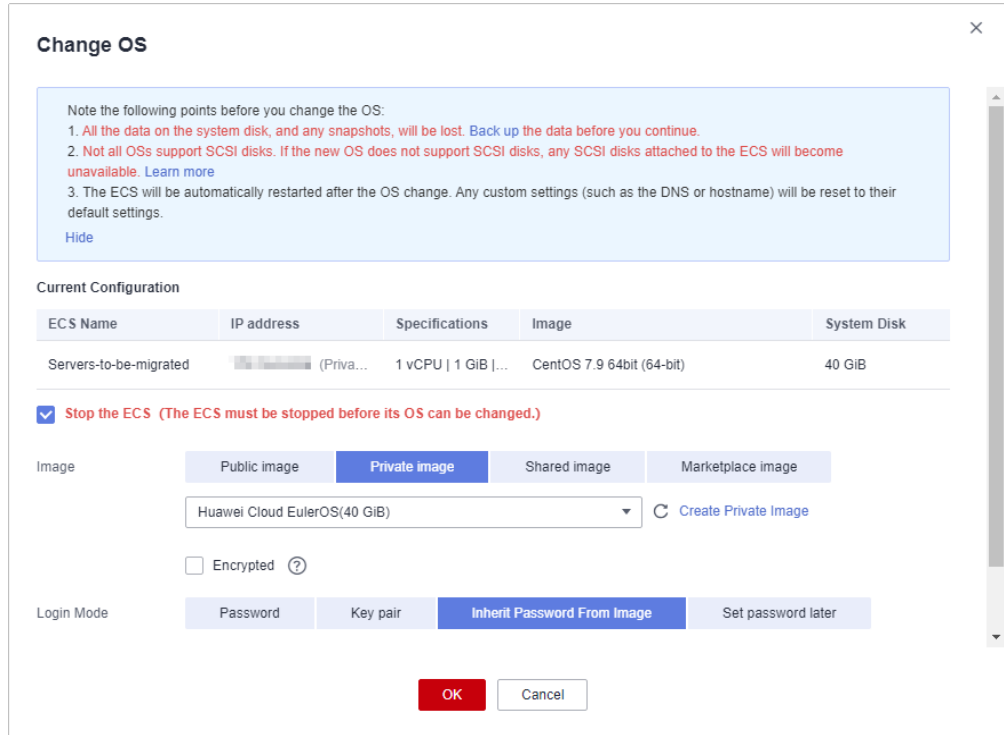
Description:

0/1,024

Agreement: I have read and agree to the [Statement of Commitment to Image Creation and Image Disclaimer](#).

Next

4. Click **Next**.
5. Confirm the image information and click **Submit**.
6. Switch back to the ECS console, locate the ECS whose OS is to be changed, and choose **More > Manage Image > Change OS** in the **Operation** column.
7. In the **Change OS** dialog, configure the following parameters:
 - Select **Stop the ECS**.
 - **Image**: Select **Private image**.
 - **Login Mode**: Select **Password**.



8. Click **OK** and complete the verification as prompted.

9. Read and select the statement, and click **OK**.

After the switchover, the OS name on the console is changed to Huawei Cloud EulerOS.

6 How Are Huawei Cloud EulerOS, openEuler, and EulerOS Different from One Another?

Huawei Cloud EulerOS, openEuler, and EulerOS are developed by Huawei. [Table 6-1](#) lists the differences among them.

Table 6-1 Differences among Huawei Cloud EulerOS, openEuler, and EulerOS

OS	Description
Huawei Cloud EulerOS	<p>Huawei Cloud EulerOS is a commercial Linux distribution developed based on openEuler. Huawei Cloud EulerOS can replace CentOS and EulerOS and provides professional maintenance and assurance. Currently, images are free of charge.</p> <p>NOTE Huawei Cloud EulerOS 2.0 is a cloud operating system developed based on openEuler 22.03 LTS.</p>
openEuler	<p>openEuler is a free, open-source operating system that does not provide commercial maintenance and assurance. openEuler was initially developed by Huawei and officially donated to the OpenAtom Foundation on November 9, 2021. Since then, the community has been providing technical support for openEuler.</p>
EulerOS	<p>EulerOS is an enterprise-class Linux distribution that offers high security, scalability, and performance, making it an ideal choice for customers seeking reliable IT infrastructure and cloud computing services.</p> <p>NOTE EulerOS is developed based on openEuler and is a Huawei internal operating system.</p>

7 How Do I Enable WireGuard in Kernel and Install wireguard-tools?

NOTE

wireguard-tools comes from the community. If you encounter any problems when using the tool, visit <https://github.com/WireGuard/wireguard-tools/pulls>.

Enabling WireGuard in Kernel

You can run the **modprobe wireguard** command to enable WireGuard.

Installing wireguard-tools

Step 1 Run the following command to install the dependency package:

```
dnf install gcc make
```

Step 2 Run the following command to download the wireguard-tools source code package:

```
wget https://git.zx2c4.com/wireguard-tools/snapshot/wireguard-tools-1.0.20210914.tar.xz
```

Step 3 Run the following command to decompress the obtained source code package:

```
tar -xf wireguard-tools-1.0.20210914.tar.xz
```

Step 4 Go to the **wireguard-tools-1.0.20210914/src** directory and run the following commands in sequence to compile and install the tool:

```
make
```

```
make install
```

Step 5 Check whether the tool is successfully installed.

You can run the **wg -h** and **wg-quick -h** commands to check whether the installation is successful.

```
[root@localhost ~]# wg -h
Usage: wg <cmd> [<args>]

Available subcommands:
  show: Shows the current configuration and device information
  showconf: Shows the current configuration of a given WireGuard interface, for use with `setconf`
  set: Change the current configuration, add peers, remove peers, or change peers
  setconf: Applies a configuration file to a WireGuard interface
  addconf: Appends a configuration file to a WireGuard interface
  syncconf: Synchronizes a configuration file to a WireGuard interface
  genkey: Generates a new private key and writes it to stdout
  genspk: Generates a new preshared key and writes it to stdout
  pubkey: Reads a private key from stdin and writes a public key to stdout
You may pass `--help' to any of these subcommands to view usage.
[root@localhost ~]# wg-quick -h
Usage: wg-quick [ up | down | save | strip ] [ CONFIG_FILE | INTERFACE ]

CONFIG_FILE is a configuration file, whose filename is the interface name
followed by `.conf'. Otherwise, INTERFACE is an interface name, with
configuration found at /etc/wireguard/INTERFACE.conf. It is to be readable
by wg(8)'s `setconf' sub-command, with the exception of the following additions
to the [Interface] section, which are handled by wg-quick:

- Address: may be specified one or more times and contains one or more
  IP addresses (with an optional CIDR mask) to be set for the interface.
- DNS: an optional DNS server to use while the device is up.
- MTU: an optional MTU for the interface; if unspecified, auto-calculated.
- Table: an optional routing table to which routes will be added; if
  unspecified or `auto', the default table is used. If `off', no routes
  are added.
- PreUp, PostUp, PreDown, PostDown: script snippets which will be executed
  by bash(1) at the corresponding phases of the link, most commonly used
  to configure DNS. The string `%i' is expanded to INTERFACE.
- SaveConfig: if set to `true', the configuration is saved from the current
  state of the interface upon shutdown.

See wg-quick(8) for more info and examples.
[root@localhost ~]#
```

----End

8 How Do I Save the User Credential Information for Logging In to Docker Like What Docker CE Does?

Background

When you run the **docker login** command to log in to Docker Community Edition (CE), data such as the username and password is saved in the user configuration file in Base64 format, which poses security risks. To ensure security, Docker included in HCE OS 2.0 encrypts the data by default. However, some Docker CE tools do not support this feature. You need to manually change the saving mode of Docker in Huawei Cloud EulerOS 2.0 like what Docker CE does.

Procedure

1. Configure the required environment variable.
export USE_DECRYPT_AUTH=true
2. Run the **docker login** command to log in to Docker again.
docker login

```
[root@ecs-7eca ~]# docker login [REDACTED]
WARNING: Error loading config file: /root/.docker/config.json: Invalid auth configuration file
Username: [REDACTED]
Password: [REDACTED]
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
```

3. Verify the data saving mode. It is recommended that you save the environment variable setting in a persistent file (such as **~/.bash_profile** or **/etc/profile**) so the setting will be applied upon system reboot.
echo "export USE_DECRYPT_AUTH=true" >> ~/.bash_profile

9 What Is OOM? Why Does OOM Occur?

OOM Concepts

Out of Memory (OOM) occurs when all available memory is exhausted and the system is unable to allocate memory for processes, which will trigger a kernel panic or OOM killer.

On Linux, OOM killer is a process that prevents other processes from collectively exhausting the host's memory. When the system is critically low on memory, the processes that use more memory than available will be killed to ensure the overall availability of the system.

OOM Parameters

Table 9-1 OOM parameters

Parameter	Description	Value	How to Change
panic_on_oom	<p>Controls how the system reacts when OOM occurs. When OOM occurs, the system has two options:</p> <ul style="list-style-type: none"> Triggers a kernel panic, during which the system may break down frequently. Selects one or more processes and triggers OOM killer to end the selected processes and to release the memory so that the system can be used normally. 	<p>You can run either of the following commands to view the parameter value:</p> <pre>cat /proc/sys/vm/panic_on_oom</pre> <pre>sysctl -a grep panic_on_oom</pre> <ul style="list-style-type: none"> If the value is set to 0, OOM killer is triggered when the memory is insufficient. If the value is set to 1, either OOM killer or kernel panic will be triggered. If the value is set to 2, a kernel panic will be forcibly triggered. As a result, the system restarts. <p>NOTE The default value of this parameter in Huawei Cloud EulerOS is 1.</p>	<p>For example, to set the value to 0, use either of the following methods:</p> <ul style="list-style-type: none"> Temporary configuration: The configuration takes effect immediately. However, after the system is restarted, the value changes to the default one. sysctl -w vm.panic_on_oom=0 Persistence configuration: The configuration still takes effect after the system is restarted. Run vim /etc/sysctl.conf, add vm.panic_on_oom =0 to the configuration file, and then run sysctl -p or restart the system for the configuration to take effect.

Parameter	Description	Value	How to Change
oom_kill_allocating_task	<p>Determines which processes are selected when the system triggers OOM killer and attempts to end some processes. The options are as follows:</p> <ul style="list-style-type: none"> Process that triggers OOM Process with the highest oom_score value 	<p>You can run either of the following commands to view the parameter value:</p> <pre>cat /proc/sys/vm/oom_kill_allocating_task</pre> <pre>sysctl -a grep oom_kill_allocating_task</pre> <ul style="list-style-type: none"> If the value is set to 0, the process with the highest oom_score value is selected. If the value is a non-zero value, the process that triggers OOM is selected. <p>NOTE The default value of this parameter in Huawei Cloud EulerOS is 0.</p>	<p>For example, to set the value to 1, use either of the following methods:</p> <ul style="list-style-type: none"> Temporary configuration: The configuration takes effect immediately. However, after the system is restarted, the value changes to the default one. sysctl -w vm.oom_kill_allocating_task=1 Persistence configuration: The configuration still takes effect after the system is restarted. Run vim /etc/sysctl.conf, add vm.oom_kill_allocating_task=1 to the configuration file, and then run sysctl -p or restart the system for the configuration to take effect.

Parameter	Description	Value	How to Change
oom_score	<p>Indicates the score of a process, which consists of two parts:</p> <ul style="list-style-type: none"> System score: The system automatically calculates the score based on the memory usage of the process. User score: This is the oom_score_adj score, which can be customized. 	<p>You can adjust the value of oom_score_adj to adjust the final score of a process. You can run the following command to view the parameter value:</p> <pre>cat /proc/Process ID/oom_score_adj</pre> <ul style="list-style-type: none"> If the value is set to 0, the oom_score value is not adjusted. If the value is negative, the probability of the process getting picked and terminated by OOM killer is reduced. If the value is positive, the probability of the process getting picked and terminated by OOM killer is increased. <p>NOTE The value of oom_score_adj ranges from -1000 to 1000. If this parameter is set to OOM_SCORE_ADJ_MIN or -1000, OOM killer is not allowed to end the process.</p>	<p>For example, to set oom_score_adj to 1000 for the process whose ID is 2939, run the following command:</p> <pre>echo 1000 > /proc/2939/oom_score_adj</pre>

Parameter	Description	Value	How to Change
oom_dump_tasks	Specifies whether to record the system process information and OOM killer information when OOM occurs, for example, dump information about all processes in the system like the process ID, memory used by the process, and page table information of the process. Such information helps you understand the cause of OOM.	<p>You can run either of the following commands to view the parameter value:</p> <pre>cat /proc/sys/vm/oom_dump_tasks</pre> <pre>sysctl -a grep oom_dump_tasks</pre> <ul style="list-style-type: none"> If the value is set to 0, related information is not printed when OOM occurs. If the value is not 0, a system-wide task dump is produced to print the memory usage of all tasks in the system in the following scenarios: <ul style="list-style-type: none"> A kernel panic is triggered due to OOM. The process to be terminated is not found. The process is found and terminated. <p>NOTE The default value of this parameter in Huawei Cloud EulerOS is 1.</p>	<p>For example, to set the value to 0, use either of the following methods:</p> <ul style="list-style-type: none"> Temporary configuration: The configuration takes effect immediately. However, after the system is restarted, the value changes to the default one. sysctl -w vm.oom_dump_tasks=0 Persistence configuration: The configuration still takes effect after the system is restarted. Run vim /etc/sysctl.conf, add vm.oom_dump_tasks=0 to the configuration file, and then run sysctl -p or restart the system for the configuration to take effect.

Example of OOM Killer

- Set Huawei Cloud EulerOS system parameters by referring to [Table 9-1](#). The following is an example configuration:

```
[root@localhost ~]# cat /proc/sys/vm/panic_on_oom
0
[root@localhost ~]# cat /proc/sys/vm/oom_kill_allocating_task
0
```

```
[root@localhost ~]# cat /proc/sys/vm/oom_dump_tasks  
1
```

- **panic_on_oom=0** indicates that OOM killer is triggered when OOM occurs.
- **oom_kill_allocating_task=0** indicates that the process with the highest **oom_score** value is preferentially terminated when OOM killer is triggered.
- **oom_dump_tasks=1** indicates that the process and OOM killer information is recorded when OOM occurs.

2. Start the process.

Start three same test processes (test, test1, and test2) in the system at the same time, continuously request new memory for the three processes, and set **oom_score_adj** of process **test1** to **1000** (indicating that OOM killer will terminate this process first), until the memory is used up and OOM occurs.

```
[root@localhost ~]# ps -ef | grep test  
root 2938 2783 0 19:08 pts/2 00:00:00 ./test  
root 2939 2822 0 19:08 pts/3 00:00:00 ./test1  
root 2940 2918 0 19:08 pts/5 00:00:00 ./test2  
[root@localhost ~]# echo 1000 > /proc/2939/oom_score_adj  
[root@localhost ~]# cat /proc/2939/oom_score_adj  
1000
```

3. View the OOM information.

After a period of time, OOM occurs in the system, and OOM killer is triggered. At the same time, the memory information of all processes in the system is printed in **/var/log/messages**, and process **test1** is terminated.

```
[root@localhost ~]# ps -ef | grep test  
root 2938 2783 0 19:08 pts/2 00:00:06 ./test  
root 2940 2918 0 19:08 pts/5 00:00:06 ./test2  
root 3383 2861 0 19:41 pts/4 00:00:00 grep --color=auto test  
[root@localhost ~]#
```

```
Mar 4 19:15:14 localhost kernel: 5865.355939] kernel fault(0x2) notification starting on CPU 10  
Mar 4 19:15:14 localhost kernel: 5865.355939] kernel fault(0x2) notification finished on CPU 10  
Mar 4 19:15:14 localhost kernel: 5865.355944] top invoked oom: gfp_mask=0x100cca(GFP_HIGHUSER_MOVABLE), order=0  
Mar 4 19:15:14 localhost kernel: 5865.355950] vcpu: 10 pcpu: 3000 comm: top xdump: loaded rained: 0 5.10.0-136.12.0.86.fc46_73.hco2_x86_64 #1  
Mar 4 19:15:14 localhost kernel: 5865.355950] hardware name: OpenStack Foundation OpenStack Nova, BIOS rel=1.12.1.0-g45cab58-20230319_150422-szxtsc10000 04/01/2014  
Mar 4 19:15:14 localhost kernel: 5865.355953] call trace:  
Mar 4 19:15:14 localhost kernel: 5865.355967] dump_stack+0x57/0x60  
Mar 4 19:15:14 localhost kernel: 5865.355970] oom_show_debug_info.part.0+0x4a/0x131  
Mar 4 19:15:14 localhost kernel: 5865.355973] ? printk+0x38/0x27  
Mar 4 19:15:14 localhost kernel: 5865.355978] out_of_memory.cold+0x65/0xab  
Mar 4 19:15:14 localhost kernel: 5865.355978] _alloc_pages+0xab9/0x1050  
Mar 4 19:15:14 localhost kernel: 5865.355980] pagecache_get_page+0x1c3/0x250  
Mar 4 19:15:14 localhost kernel: 5865.355982] filemap_fault+0x2/0x70  
Mar 4 19:15:14 localhost kernel: 5865.356000] ext4_filemap_fault+0x2d/0x50 [ext4]  
Mar 4 19:15:14 localhost kernel: 5865.356070] do_fault+0x27/0x40  
Mar 4 19:15:14 localhost kernel: 5865.356073] do_read_fault+0x31/0xf0  
Mar 4 19:15:14 localhost kernel: 5865.356074] handle_mm_fault+0x48/0x7b0  
Mar 4 19:15:14 localhost kernel: 5865.356076] handle_mm_fault+0x4c/0x2f0  
Mar 4 19:15:14 localhost kernel: 5865.356079] exec_page_fault+0x1d/0x40  
Mar 4 19:15:14 localhost kernel: 5865.356081] asm_exc_page_fault+0x1b/0x30  
Mar 4 19:15:14 localhost kernel: 5865.356082] asm_exc_page_fault+0x19/0x30  
Mar 4 19:15:14 localhost kernel: 5865.356086] RIP: 00007f4b7b08b060  
Mar 4 19:15:14 localhost kernel: 5865.356093] Code: Unable to access opcode bytes at RIP 0x7f4b7b08b060.  
Mar 4 19:15:14 localhost kernel: 5865.356094] RSP: 002b:00007f4b7b082340 EFLAGS: 00010202  
Mar 4 19:15:14 localhost kernel: 5865.356096] RAX: 00007f4b7b08e088 RDI: 0000000000000000 RCX: 0000000000000000  
Mar 4 19:15:14 localhost kernel: 5865.356096] RDX: 0000559f8de1c881 RSI: 00007f4b7b86e540 R0I: 000000000000000b  
Mar 4 19:15:14 localhost kernel: 5865.356097] RBP: 00007f4b7b097060 R08: 00007f4b7b082340 R09: 00007f4b7b082340  
Mar 4 19:15:14 localhost kernel: 5865.356098] R10: 0000000000000000 R11: 0000000000000240 R12: 00007f4b7b082340  
Mar 4 19:15:14 localhost kernel: 5865.356099] R13: 0000000000000001 R14: 0000000000000000 R15: 0000559f8de1d161  
Mar 4 19:15:14 localhost kernel: 5865.356099] CPU: 10 PID: 2939/3000 (top) Not tainted 5.10.0-136.12.0.86.fc46_73.hco2_x86_64 #1  
Mar 4 19:15:14 localhost kernel: 5865.356074] #012slab info:  
Mar 4 19:15:14 localhost kernel: 5865.369230] slabinfo - version: 2.1  
Mar 4 19:15:14 localhost kernel: 5865.369240] # name active_objs<num_objs> <objsize> <objperslab> <pagesperslab> : tunables <limits> <batchcount> <sharedfactor>  
Mar 4 19:15:14 localhost kernel: 5865.369241] nf_contrack_expect 0 0 0 240 33 2 : tunables 0 0 0 : slabdata 11 11 0  
Mar 4 19:15:14 localhost kernel: 5865.369244] nf_contrack 561 561 320 51 4 : tunables 0 0 0 : slabdata 11 11 0  
Mar 4 19:15:14 localhost kernel: 5865.369253] rpc_inode_cache 46 46 704 46 8 : tunables 0 0 0 : slabdata 1 1 0  
Mar 4 19:15:10 localhost kernel: 5865.376051] 2938 0 2938 5974 0 69942 076 0 bash  
Mar 4 19:15:15 localhost kernel: 5865.376051] 2938 0 2938 60135 20338 0 20744 26022 0 test  
Mar 4 19:15:15 localhost kernel: 5865.376051] 2939 0 2939 61030 29150 580256 33434 1000 test1  
Mar 4 19:15:15 localhost kernel: 5865.376051] 2940 0 2940 60402 29080 48786 30518 0 test2  
Mar 4 19:15:15 localhost kernel: 5865.376061] 2941 0 2941 655 40 6536 249 0 sshd  
Mar 4 19:15:15 localhost kernel: 5865.376064] 2945 0 2945 4640 141 6536 230 0 sshd  
Mar 4 19:15:15 localhost kernel: 5865.376065] 2946 0 2946 457 44 7378 241 0 sshd  
Mar 4 19:15:15 localhost kernel: 5865.376066] 2950 0 2950 457 44 7378 241 0 sshd  
Mar 4 19:15:15 localhost kernel: 5865.376069] 2951 0 2951 457 44 7378 241 0 sshd  
Mar 4 19:15:15 localhost kernel: 5865.376091] 2961 0 2961 5974 81 69632 324 0 sftp-server  
Mar 4 19:15:15 localhost kernel: 5865.376092] 3010 0 3010 6436 309 7378 62 0 top  
Mar 4 19:15:15 localhost kernel: 5865.376072] oom-kill:constraint=CONSTRAINT_NONE,memsk=(null),cpuset=/,mems_allowed=global,oom_task_memcg/system.slice/ssh.service(task=test1,pids=339) uid=0  
Mar 4 19:15:15 localhost kernel: 5865.376091] out of memory: Killed process 2939 (test1) total-vm:252212kB, mem-res:11627kB, file-rss:4kB, shmem-rss:4kB, UID:0 pgtables:69348 oom_score_adj:1000  
Mar 4 19:20:03 localhost systemd[1]: Starting system activity accounting tool...  
Mar 4 19:20:03 localhost systemd[1]: sysstat-collect.service: deactivated successfully.
```

Possible Causes

- The cgroup memory is insufficient.

The memory exceeds the value of **memory.limit_in_bytes** in cgroup. Suppose **memory.limit_in_bytes** is set to 80 MB and 100 MB of memory is allocated to memhog. As shown in the logs (stored in the **/var/log/messages**

directory), the memhog process (PID: 2021820) uses 81,920 KB of memory, which exceeds the memory specified by **memory.limit_in_bytes** and triggers OOM.

```
warning|kernel[-][2919920.414131] memhog invoked oom-killer: gfp_mask=0xcc(GFP_KERNEL), order=0, oom_score_adj=0
info|kernel[-][2919920.414220] memory: usage 81920kB, limit 81920kB, failcnt 30
err|kernel[-][2919920.414272] Memory cgroup out of memory: Killed process 2021820 (memhog) total-vm:105048kB, anon-rss:81884kB, file-rss:1544kB, shmem-rss:0kB, UID:0 pgtables:208kB oom_score_adj:0
```

- The parent cgroup memory is insufficient.

The memory of child cgroups is sufficient, but the memory of the parent cgroup is insufficient and exceeds the memory limit. In the following example, **memory.limit_in_bytes** is set to 80 MB for the parent cgroup and to 50 MB for the two child cgroups, respectively. A program is used to cyclically allocate memory in the two child cgroups to trigger OOM. Some logs in **/var/log/messages** are as follows:

```
warning|kernel[-][2925796.529231] main invoked oom-killer: gfp_mask=0xcc(GFP_KERNEL), order=0, oom_score_adj=0
info|kernel[-][2925796.529315] memory: usage 81920kB, limit 81920kB, failcnt 199
err|kernel[-][2925796.529366] Memory cgroup out of memory: Killed process 3238866 (main) total-vm:46792kB, anon-rss:44148kB, file-rss:1264kB, shmem-rss:0kB, UID:0 pgtables:124kB oom_score_adj:0
```

- The system memory is insufficient.

The free memory of the OS is insufficient, and some programs keep requesting memory. Even some memory can be reclaimed, the memory is still insufficient, and OOM is triggered. In the following example, a program is used to cyclically allocate memory in the OS to trigger OOM. The logs in **/var/log/messages** show that the free memory of Node 0 is lower than the minimum memory (the value of **low**), triggering OOM.

```
kernel: [ 1475.869152] main invoked oom: gfp_mask=0x100dca(GFP_HIGHUSER_MOVABLE|__GFP_ZERO), order=0
kernel: [ 1477.959960] Node 0 DMA32 free:22324kB min:44676kB low:55844kB high:67012kB reserved_highatomic:0kB active_anon:174212kB inactive_anon:1539340kB active_file:0kB inactive_file:64kB unevictable:0kB writepending:0kB present:2080636kB managed:1840628kB mlocked:0kB pagetables:7536kB bounce:0kB free_pcp:0kB local_pcp:0kB free_cma:0kB
kernel: [ 1477.960064] oom-kill:constraint=CONSTRAINT_NONE,nodemask=(null),cpuset=/,mems_allowed=0,global_oom,task_memcg=/system.slice/sshd.service,task=main,pid=1822,uid=0
kernel: [ 1477.960084] Out of memory: Killed process 1822 (main) total-vm:742748kB, anon-rss:397884kB, file-rss:4kB, shmem-rss:0kB, UID:0 pgtables:1492kB oom_score_adj:1000
```

- The memory of the memory nodes is insufficient.

In a NUMA system, an OS has multiple memory nodes. If a program uses up the memory of a specific memory node, OOM may be triggered even when the OS memory is sufficient. In the following example, there are two memory nodes, and a program is used to cyclically allocate memory on Node 1. As a result, the memory of Node 1 is insufficient, but the OS memory is sufficient. Some logs in **/var/log/messages** are as follows:

```
kernel: [ 465.863160] main invoked oom: gfp_mask=0x100dca(GFP_HIGHUSER_MOVABLE|__GFP_ZERO), order=0
kernel: [ 465.878286] active_anon:218 inactive_anon:202527 isolated_anon:0#012 active_file:5979 inactive_file:5231 isolated_file:0#012 unevictable:0 dirty:0 writeback:0#012 slab_reclaimable:6164 slab_unreclaimable:9671#012 mapped:4663 shmem:2556 pagetables:846 bounce:0#012 free:226231 free_pcp:36 free_cma:0
kernel: [ 465.878292] Node 1 DMA32 free:34068kB min:32016kB low:40020kB high:48024kB reserved_highatomic:0kB active_anon:188kB inactive_anon:778076kB active_file:20kB inactive_file:40kB unevictable:0kB writepending:0kB present:1048444kB managed:866920kB mlocked:0kB pagetables:2752kB bounce:0kB free_pcp:144kB local_pcp:0kB free_cma:0kB
kernel: [ 933.264779] oom-kill:constraint=CONSTRAINT_MEMORY_POLICY,nodemask=1,cpuset=/,mems_allowed=0-1,global_oom,
```

```
task_memcg=/system.slice/sshd.service,task=main,pid=1733,uid=0  
kernel: [ 465.878438] Out of memory: Killed process 1734 (main) total-vm:239028kB, anon-  
rss:236300kB, file-rss:200kB, shmem-rss:0kB, UID:0 pgtables:504kB oom_score_adj:1000
```

- Other possible cause
During memory allocation, if the memory of the buddy system is insufficient, OOM killer is triggered to release the memory to the buddy system.

Solutions

- Check if there is memory leak, which causes OOM.
- Check whether the **cgroup limit_in_bytes** configuration matches the memory plan. If any modification is required, run the following command:

```
echo <value> > /sys/fs/cgroup/memory/<cgroup_name>/memory.limit_in_bytes
```
- If more memory is required, upgrade the ECS flavors.

10 How Do I Handle IPVS Errors?

Background

An IP virtual server (IPVS) is used for load balancing and network forwarding. If you configure an IPVS but do not set up a real server in the system, error logs will be generated after you log in to the ECS using VNC.

Symptoms

If an IPVS is configured but no real server is set up, when a network request is sent to the virtual server address, an error log similar to the following is displayed in the CLI after you log in to the ECS using VNC.

```
[32264.645949][T268365] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
[32265.234919][T268366] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
[32265.954662][T268367] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
[32266.557032][T268368] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
[32267.166530][T268369] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
[32267.725920][T268370] IPVS: wlc: TCP 192.168.1.1:5000 - no destination available
```

Solution

1. Install `ipvsadm`.
2. Run the `ipvsadm -Ln` command to query the configuration of the current virtual server. Find the entry corresponding to the virtual server for which an error is reported.

Figure 10-1 No real server configured

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port[:Subnet] Scheduler Established(Sec.) Flags
  -> RemoteAddress:Port[:Of] Forward Weight ActiveConn InActConn UteppAddr:uteppport Unild Mac
TCP 192.168.1.1:5000 wlc
```

If no real server is configured, the configuration is incomplete, and an error is generated. Check your service process.

Figure 10-2 Real server configured

```
IP Virtual Server version 1.2.1 (size=4096)
Prot LocalAddress:Port[:Subnet] Scheduler Established(Sec.) Flags
  -> RemoteAddress:Port[:Of] Forward Weight ActiveConn InActConn UteppAddr:uteppport Unild Mac
TCP 192.168.1.1:5000 wlc
  -> 192.168.1.2:5000 Masq 1 0 0
```

As shown in the figure, a real server is configured.

3. To eliminate the interference of the IPVS error log on the VNC, perform either of the following operations:
 - Disable network requests sent by the service. The specific operations are determined by the service requirements.
 - Run the following command to adjust the print level of the kernel printk:
echo 3 4 1 7 > /proc/sys/kernel/printk

 **NOTE**

If the system configuration is modified temporarily, restore the system configuration at a proper time.

- Log in to the ECS using CloudShell.

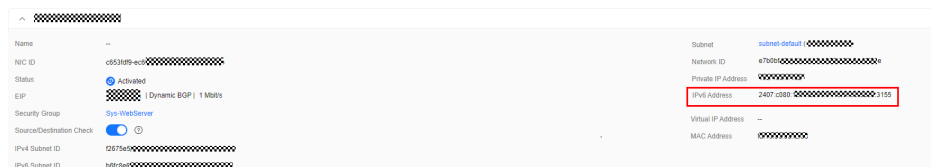
11 Why Can't HCE OS Obtain an IPv6 Address After IPv6 Is Enabled for an ECS?

Background

After IPv6 is enabled for the ECS NIC on the ECS console, HCE OS cannot obtain an IPv6 address because DHCP is not correctly configured for the OS.

Symptoms

IPv6 has been enabled on the ECS console, and an IPv6 address is displayed on the ECS details page.



However, the OS cannot obtain an IPv6 address.

```
root@ecs:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:00:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.24/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 86343sec preferred_lft 86343sec
    inet6 fe80::f816:3eff:fe9a:f2a8/64 scope link
        valid_lft forever preferred_lft forever
```

Solution

1. Configure DHCP to automatically obtain IPv6 addresses. Add the following content to the NIC configuration file `/etc/sysconfig/network-scripts/ifcfg-ethx`:

```
IPV6INIT="yes"
```

```
DHCPV6C="yes"
```

```
[root@huaweicloud ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
PERSISTENT_DHCLIENT="yes"
IPv6INIT="yes"
DHCPV6C="yes"
[root@huaweicloud ~]#
```

2. Run the following command to restart NetworkManager to obtain an IPv6 address:

systemctl restart NetworkManager

```
[root@huaweicloud ~]# systemctl restart NetworkManager
[root@huaweicloud ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:14:00:04 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.14/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 86394sec preferred_lft 86394sec
    inet6 2407::3155/128 scope global dynamic noprefixroute
        valid_lft 7195sec preferred_lft 7195sec
    inet6 fe80::f816:3eff:fe9a:f2a8/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@huaweicloud ~]#
```