Huawei Cloud EulerOS (HCE)

FAQs

Issue	01
Date	2025-12-08

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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.

Huawei Cloud Computing Technologies Co., Ltd.

Address: Huawei Cloud Data Center Jiaoxinggong Road
Qianzhong Avenue
Gui'an New District
Gui Zhou 550029
People's Republic of China

Website: https://www.huaweicloud.com/intl/en-us/

Contents

1 What Do I Do If CentOS Linux Is No Longer Maintained?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

CenterOS Linux users will be affected as follows:

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

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.

- 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 CentOS Stream or Rocky Linux, 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 an OS	Migrating an OS
Data backup	• 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.	• 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) is an openEuler-based cloud operating system. HCE 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 it 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
CentOS Stream	CentOS Stream is a continuous delivery distribution provided by CentOS.	Individuals or enterprises that are used to CentOS and desire continuous updates
Rocky Linux	Rocky Linux is a community-driven enterprise-class OS. It is a downstream release of Red Hat Enterprise Linux (RHEL). Rocky Linux is fully compatible with and as stable as CentOS.	Individuals or enterprises that want to continue to use free images in an open source community
AlmaLinux	AlmaLinux is an open-source, community-driven Linux distribution developed by the CloudLinux team. It fills the gap left by the discontinuation of the CentOS Linux stable release. AlmaLinux can remain 1:1 compatibility with upstream RHEL. You can use it for OS changes without stopping your server.	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 ended 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.

HCE can be a perfect alternative to CentOS. You can use our migration tool to easily migrate OSs such as CentOS or EulerOS to HCE. 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, HCE can replace CentOS. HCE 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 HCE, you can use the compatibility evaluation tool to quickly scan the software to determine whether they are compatible with HCE.

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 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 if 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 in HCE (x86 and Arm).

Constraints

- The installation is only available for HCE 2.0.
- The CX6 NIC driver version is 23.10-1.1.9.0-LTS or later.

Prerequisites

HCE with kernel 5.10 or later has been installed.

Installing the MLNX Driver in x86

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.

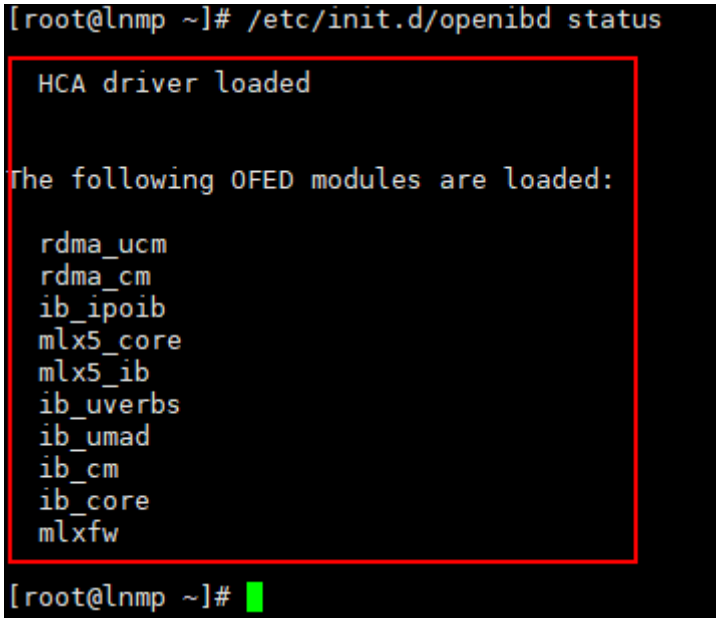
   ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.x86_64/extra/mlnx-ofa_kernel /lib/modules/$(uname -r)/weak-updates/

   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:

Figure 3-1 Successful driver installation



Installing the MLNX Driver in Arm

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.

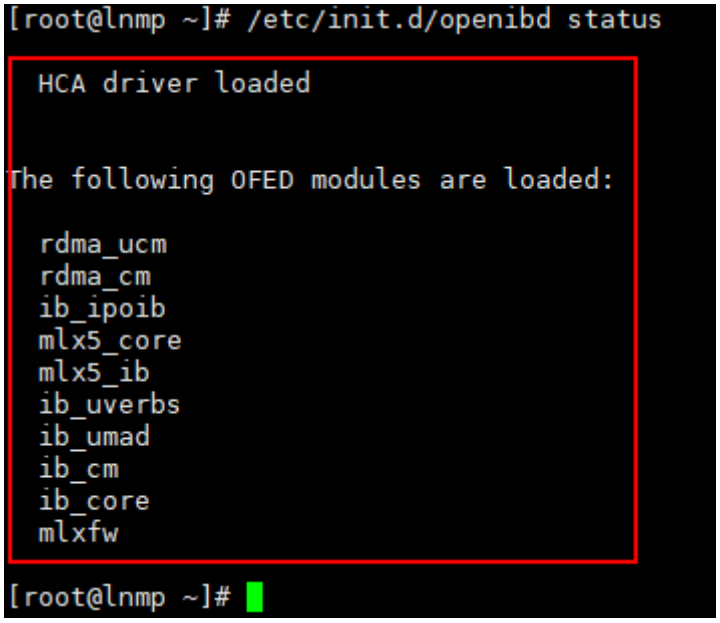
   ln -s /lib/modules/5.10.0-60.18.0.50.oe2203.aarch64/extra/mlnx-ofa_kernel /lib/modules/$(uname -r)/weak-updates/

   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 successfully.

Figure 3-2 Successful driver installation



4 How Do I Enable SELinux of HCE?

SELinux is disabled by default on HCE. 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. Modify the configuration file.

   EFI: Delete **selinux=0** from **/boot/efi/EFI/hce/grub.cfg**.

   BIOS: Delete **selinux=0** from **/boot/grub2/grub.cfg**.

   NOTE

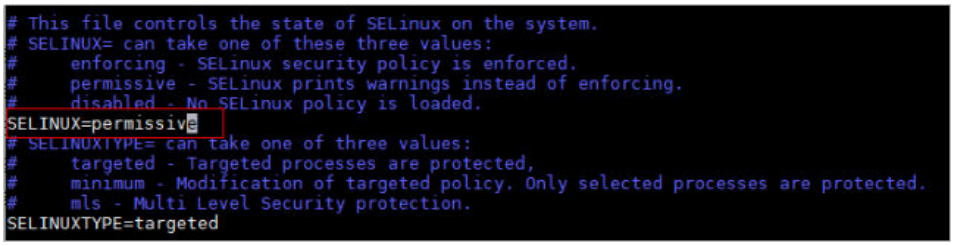
   If **selinux=0** cannot be found, ignore it.

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.

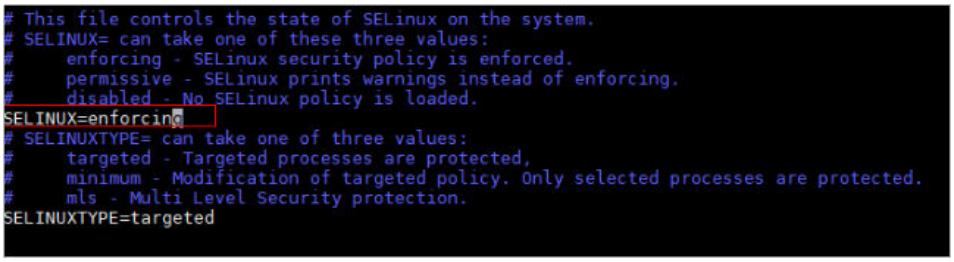
3. Open the configuration file **/etc/selinux/config**, set **SELINUX** to **permissive**, and run the **reboot** command to restart the OS.

   Figure 4-1 Setting SELINUX=permissive



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

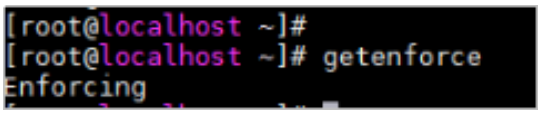
   Figure 4-2 Setting SELINUX=enforcing

5. Run the **getenforce** command to check the SELinux status.

   If **Enforcing** is displayed, SELinux is enabled.

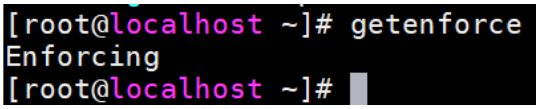
   Figure 4-3 Checking the SELinux status

5 How Do I Disable SELinux of HCE?

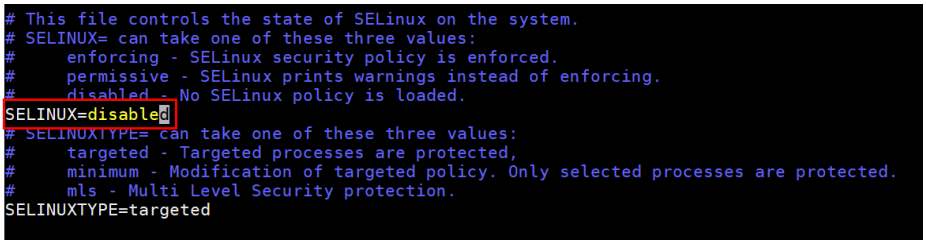
1. Run **getenforce** to check the SELinux status. **Enforcing** indicates SELinux is enabled.

Figure 5-1 Example of enabled SELinux



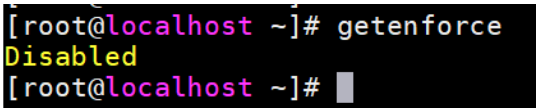
2. Open the **/etc/selinux/config** configuration file, set **SELINUX=disabled**, save the file, and exit. Run **reboot** to restart the OS.

Figure 5-2 Example of disabling SELINUX



3. After the restart, run **getenforce** to check the SELinux status. If **Disabled** is displayed, SELinux is disabled.

Figure 5-3 Example of disabled SELinux



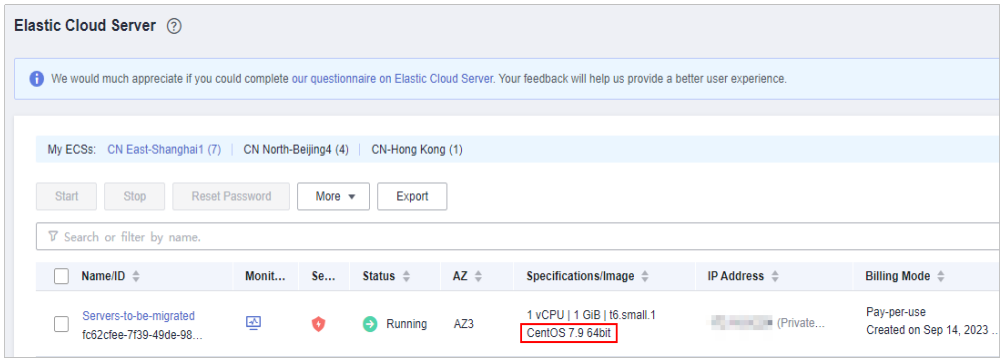
6 How Do I Change the OS Name on the Console After the OS Is Migrated?

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.

Figure 6-1 Changing the OS name

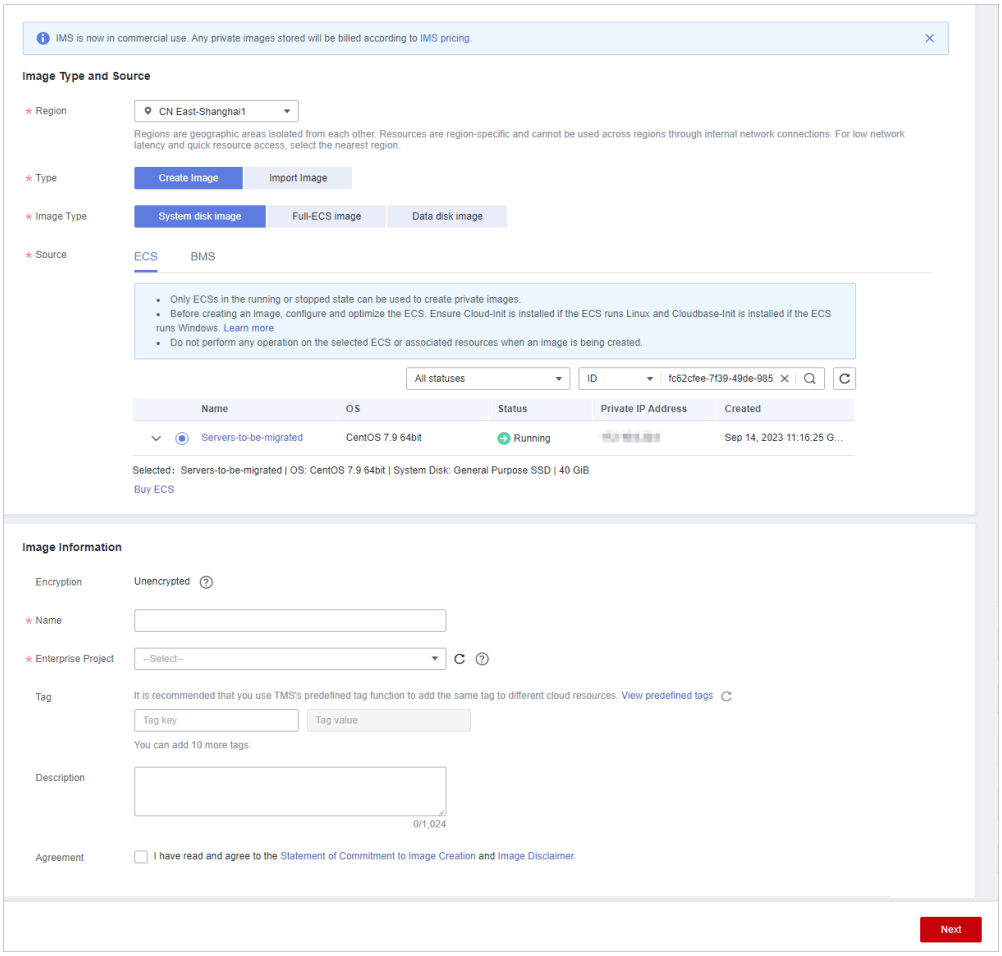


Procedure

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.

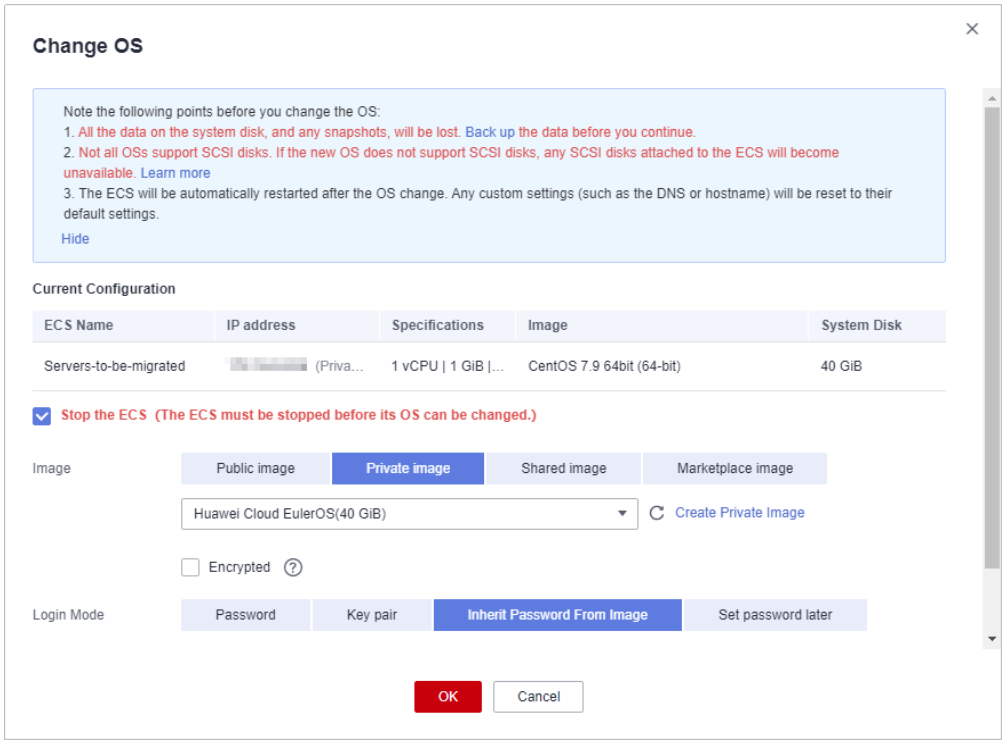
- **Agreement**: Read the agreements and select the **Agreement** option.

Figure 6-2 Configuring image information



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**.

Figure 6-3 Configuring parameters



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.

7 What Are the Differences Between HCE, EulerOS, and openEuler?

HCE, EulerOS, and openEuler are all developed by Huawei. **Table 7-1** describes the differences between them.

Table 7-1 Differences between HCE, EulerOS, and openEuler

OS	Description
HCE	HCE is a commercial Linux distribution developed based on openEuler. It can replace CentOS and EulerOS and provide professional maintenance and assurance. Currently, their images are free of charge.
openEuler	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.
EulerOS	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.

8 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.

Figure 8-1 Verifying the installation

```
[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
  genpsk: 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

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

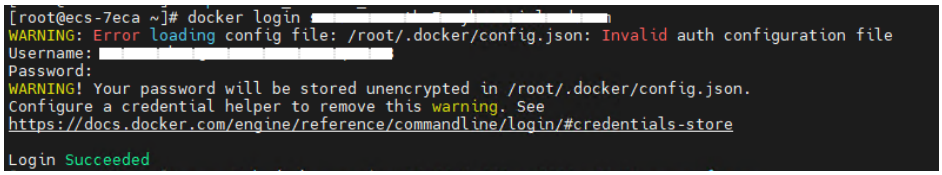
Background

When you run **docker login** 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 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 HCE 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
   ```

   Figure 9-1 Running docker login to log in to Docker again

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


10 What Are MemAvailable and MemAvailable?

When running **cat /proc/meminfo**, people tend to focus on the **MemFree** and **MemAvailable** fields in the output. The following describes the two fields.

1. **MemFree**: Physical memory that is not used.
   - It is the number of unallocated pages in the buddy allocator.
   - **MemFree** does not include the memory used as a page cache, slab cache, or buffer.
2. **MemAvailable**: Memory that can be used by applications, estimated by the kernel depending on the possibility of cache reclamation. **MemAvailable** includes:
   - MemFree
   - Most page caches (reclaimable)
   - Reclaimable slab cache
   - min_free_kbytes (minimum amount of free memory)

> **CAUTION**
>
> A small **MemFree** value does not mean the system is about to run out of memory. As long as there is sufficient reclaimable memory and swap, OOM errors will not occur even if the value of **MemFree** is small. The OOM risk depends on the value of **MemAvailable**.

11 How Does the System Reclaim Memory?

Linux Memory Reclamation Triggers

1. kswapd reclamation (background, asynchronous)
   - The kernel daemon kswapd continuously checks for the following memory levels:
     - **high**: The memory is normal and reclamation is not required.
     - **low**: kswapd is triggered to reclaim some pages.
     - **min**: The memory is insufficient even when kswapd has been triggered for reclamation.
   - This is gentle memory reclamation that is executed in the background and does not block processes.
2. Direct Reclaim (foreground, synchronous)
   - When a process requests memory but the system does not have enough memory available, the process directly reclaims memory by itself.
   - The reclamation is synchronous and may block processes.
3. Out of Memory (OOM) events
   - If the free physical memory is still insufficient for applications after Direct Reclaim is executed, the kernel triggers an OOM system to kill processes or enter a panic state.

Linux Memory Reclamation Policy

1. Reclaim page caches (file pages) first.
   - For a clean page, discard it directly. (the lowest cost option)
   - For a dirty page, write it back to disks and then discard it.
2. Reclaim slab caches (kernel objects such as dentry and inode).
   - Most of the kernel's object caches (memory managed by the slab or slub allocator), such as dentry caches and inode caches, can be reclaimed.
   - The kernel releases some of them using a shrinker (for example, **shrink_slab()**).

3. Swap anonymous pages (heaps/stacks).
   - Anonymous pages are created by processes for heap allocations, stack memory, and malloc calls. Anonymous pages are not backed by files on a disk. To reclaim them, you must write them to the swap partition. The kernel writes inactive anonymous pages to the swap partition and then releases the physical pages.
   - The cost is much higher than that of reclaiming caches.
4. Trigger an OOM kill.
   - If page caches have been reclaimed and the swap space is full, Linux triggers the OOM killer.

12 Why Is OS Memory Reserved?

In the context of kdump, a crash kernel is used to take over the system when the main kernel crashes. To ensure that the crash kernel can start normally, reserve a physical memory for it during system startup. **crashkernel** determines how much memory needs to be reserved for the crash kernel.

If **crashkernel** is set to **auto**, the system automatically determines how much memory is reserved for the crash kernel based on how much physical memory the server has. For **crashkernl=auto**, memory is reserved depending on the CPU architecture. In x86 or Arm, the reserved size is based on the memory specifications.

13 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 13-1 OOM parameters

Parameter	Description	Value	How to Change
panic_on_oom	Controls how the system reacts when OOM occurs. When OOM occurs, the system has two options: • 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.	You can run either of the following commands to view the parameter value: **cat /proc/sys/vm/panic_on_oom** **sysctl -a | grep panic_on_oom** • 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. **NOTE** The default value of this parameter in HCE is **0**.	For example, to set the value to **0**, use either of the following methods: • 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	Determines which processes are selected when the system triggers OOM killer and attempts to end some processes. The options are as follows: • Process that triggers OOM • Process with the highest **oom_score** value	You can run either of the following commands to view the parameter value: **cat /proc/sys/vm/oom_kill_allocating_task** **sysctl -a | grep oom_kill_allocating_task** • 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. **NOTE** The default value of this parameter in HCE is **0**.	For example, to set the value to **1**, use either of the following methods: • 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	Indicates the score of a process, which consists of two parts: • 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.	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: **cat /proc/***Process ID***/oom_score_adj** • 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. **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.	For example, to set **oom_score_adj** to **1000** for the process whose ID is **2939**, run the following command: **echo 1000 > /proc/2939/oom_score_adj**

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.	You can run either of the following commands to view the parameter value: **cat /proc/sys/vm/oom_dump_tasks** **sysctl -a | grep oom_dump_tasks** • 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: – A kernel panic is triggered due to OOM. – The process to be terminated is not found. – The process is found and terminated. **NOTE** The default value of this parameter in HCE is **1**.	For example, to set the value to **0**, use either of the following methods: • 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

1. Set HCE system parameters by referring to **Table 13-1**. The following is an example:

```
[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.

Figure 13-1 Checking process information

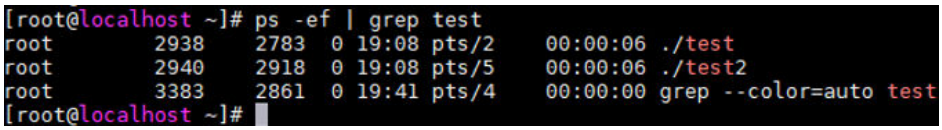


Figure 13-2 Error logs for OOM

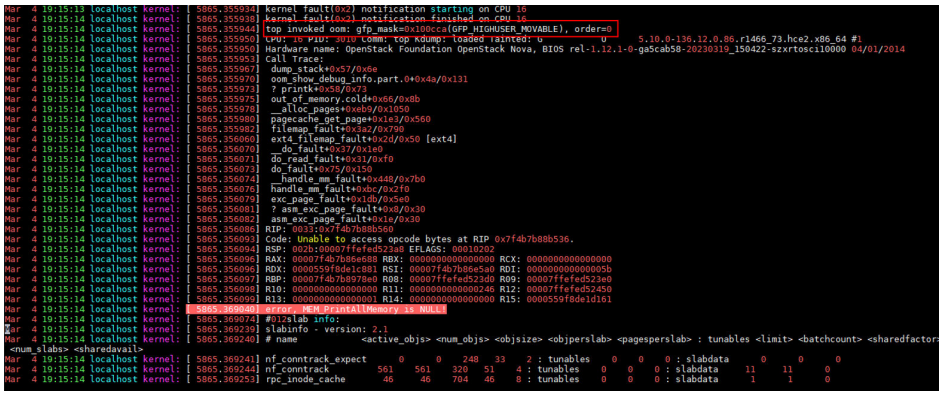
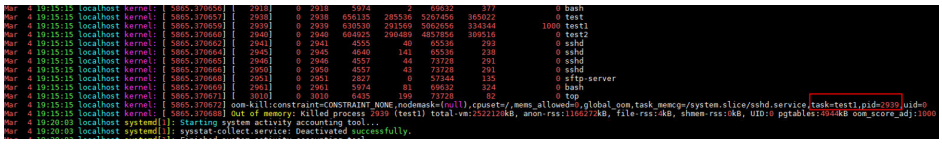


Figure 13-3 Processes that triggered OOM



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=0xcc0(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=0xcc0(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.

14 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.

Figure 14-1 Error logs

```
[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 14-2 No real server configured

   ```
   IP Virtual Server version 1.2.1 (size=4096)
   Prot LocalAddress:Port[:Subnet] Scheduler Established(Sec.) Flags
     -> RemoteAddress:Port[:Oif]          Forward Weight ActiveConn InActConn VtepAddr:vtepport VniId 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 14-3 Real server configured

   ```
   IP Virtual Server version 1.2.1 (size=4096)
   Prot LocalAddress:Port[:Subnet] Scheduler Established(Sec.) Flags
     -> RemoteAddress:Port[:Oif]          Forward Weight ActiveConn InActConn VtepAddr:vtepport VniId 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.

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

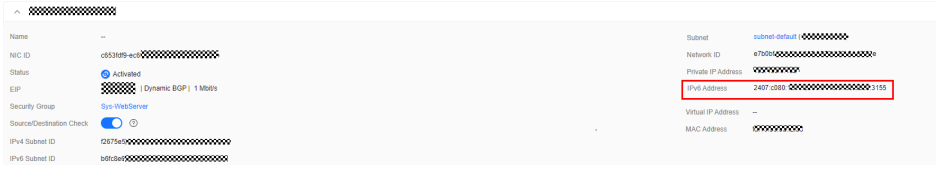
Background

After IPv6 is enabled for ECS NICs on the ECS console, HCE 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.

Figure 15-1 ECS IPv6 address



However, the OS cannot obtain an IPv6 address.

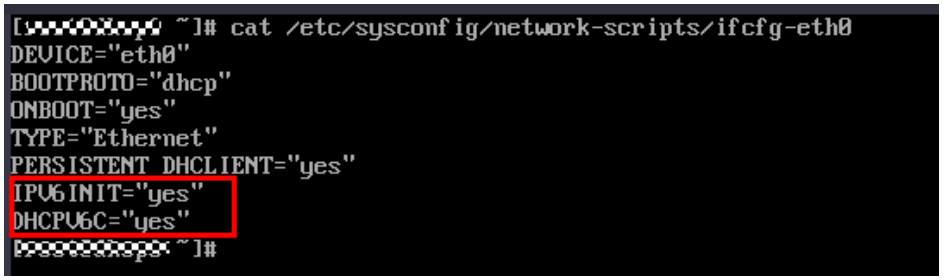
Figure 15-2 OS settings

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"
```

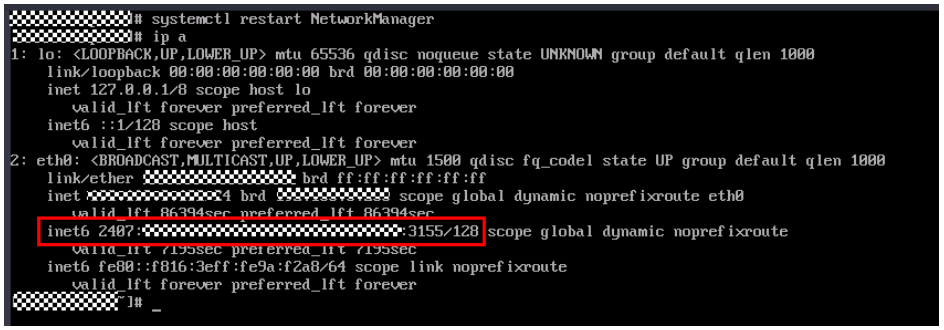
Figure 15-3 Adding parameters



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

```
systemctl restart NetworkManager
```

Figure 15-4 Obtaining an IPv6 address



16 How Do I Set Auto Logout Using TMOUT?

Scenarios

To ensure system security and reduce resource waste, users must log out they are not actively using the system. This can be achieved by setting an automatic logout timer using *TMOUT*.

TMOUT is an environment variable in Linux shell that defines number of seconds a shell session can be idle before it is automatically logged out. When this variable is set, shell will terminate session if there is no input activity for set time. If this variable is not set or is set to **0**, automatic logout is disabled, and users are not disconnected due to long-time inactivity.

Temporary Auto Logout Time

1. Check the automatic logout time (the value of *TMOUT*):
   ```
   echo $TMOUT
   ```
   If no value is returned, *TMOUT* is not configured.
2. Configure the automatic logout time for the current session.
   ```
   export TMOUT=seconds
   ```

Permanent Auto Logout Time

Method 1

Run the following command to modify the **/etc/profile** file. If the modification does not take effect, modify the **/etc/bashrc** file. The modification steps are the same. In this way, the automatic logout time will be configured for all users who use the configuration file.

```
vim /etc/profile
```

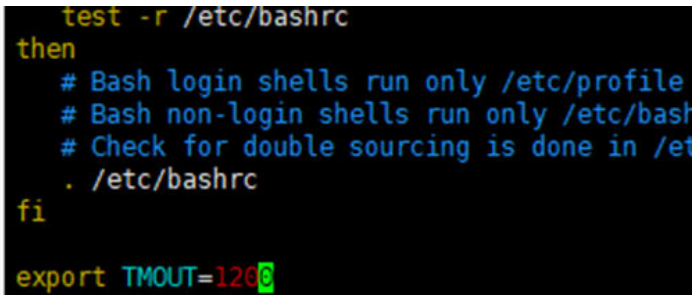
Or

```
vim /etc/bashrc
```

Add the following command to the end of the file: For example, you can set the automatic logout time to 1,200 seconds. If the value is set to **0**, automatic logout will be disabled.


```
export TMOUT=1200
```

Figure 16-1 Configuring the automatic logout time



Save the file and run the following command to refresh the file:

```
source /etc/profile
```

Method 2

Run the following commands in sequence to change the automatic logout time:

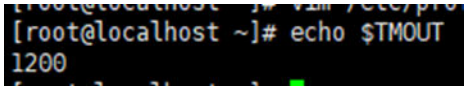
```
sed -i '$a\export TMOUT=1200' /etc/profile
source /etc/profile
```

Check the automatic logout time:

```
echo $TMOUT
```

If the defined value is displayed, the automatic logout is configured successfully.

Figure 16-2 Successful configuration of automatic logout



17 How Do I Understand the Value of panic_on_oom?

Description

panic_on_oom controls how the system reacts when OOM occurs. When OOM occurs, the system has two choices:

- Trigger a kernel panic, during which the system may break down frequently.
- Trigger OOM killer to terminate certain processes and release the memory so that the system can be used normally.

You can run either of the following commands to view the parameter value:

cat /proc/sys/vm/panic_on_oom

sysctl -a | grep panic_on_oom

- 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.

Change Description

In HCE 2.0.2503 or earlier versions, the default value of **panic_on_oom** is **1**. In versions later than HCE 2.0.2503, the default value of **panic_on_oom** is **0**.

After an upgrade to HCE 2.0.2503 and then a rollback to the previous version, the default value of **panic_on_oom** is still **0**. If you need to change the value after a rollback, perform the following operations:

- Temporary configuration: The configuration takes effect immediately. However, after the system is restarted, the value changes to the default one.

  For example, to set **panic_on_oom** to **1**, run the following command:

  ```
  sysctl -w vm.panic_on_oom=1
  ```

- Persistent configuration: The configuration still takes effect after the system is restarted.

  For example, to set **panic_on_oom** to **1**, run the following command:


```
vim /etc/sysctl.conf
```

Add **vm.panic_on_oom = 1** to the file, and then run **sysctl -p** or restart the system to make the configuration persistent.

18 How Do I Add a Character Set?

By default, only some common character sets are installed in HCE. If you need other character sets, perform the following steps to install them:

CAUTION

Before the installation, ensure that the repository is correctly configured. For details, see **Configuring Repositories and Installing Software for HCE**.

Step 1 Run **dnf install glibc-all-langpacks glibc-locale-archive -y**.

Figure 18-1 Installing character sets

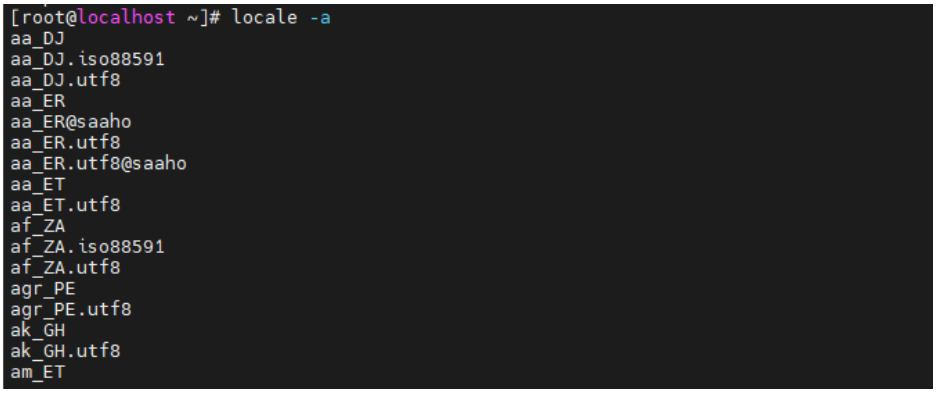
```
Last metadata expiration check: 0:46:51 ago on Sat 01 Mar 2025 10:50:12 AM CST.
Dependencies resolved.
================================================================================
 Package                  Architecture   Version                 Repository        Size
================================================================================
Installing:
 glibc-all-langpacks      aarch64        2.34-70.r88.hce2        hceversion        28 M
 glibc-locale-archive     aarch64        2.34-70.r88.hce2        hceversion        26 M

Transaction Summary
================================================================================
Install  2 Packages

Total download size: 54 M
Installed size: 431 M
Downloading Packages:
(1/2): glibc-locale-archive-2.34-70.r88.hce2.aarch64.rpm       47 MB/s |  26 MB     00:00
(2/2): glibc-all-langpacks-2.34-70.r88.hce2.aarch64.rpm        44 MB/s |  28 MB     00:00
--------------------------------------------------------------------------------
Total                                                          85 MB/s |  54 MB     00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
```

Step 2 Run **locale -a** to check whether the required character sets have been installed.

Figure 18-2 Checking the installed character sets



Step 3 To switch the language, run the following command (Polish is used as an example):

```
localectl set-locale LANG=pl_PL.UTF-8
```

Step 4 Restart HCE for the changes to take effect.

----End

19 How Do I Handle Secure Boot Failures Caused by Certificate Changes?

Background

Before March 2025, in the RPM package **hce-sign-certificate-1.0-1.hce2** released in HCE, two certificates were provided for secure boot: **HCE_Secure_Boot_RSA_Code-Signing_Authority_1.cer** and **Huawei_Code-Signing_Authority_CA_3.der.cer**. The two certificates were used to verify the signatures of Shim, GRUB, and vmlinux components during the secure boot of an OS. Either of the two certificates can be imported to BIOS as a signature verification certificate. The **HCE_Secure_Boot_RSA_Code-Signing_Authority_1.cer** certificate expired in April 2025 and could no longer be used.

If HCE released before March 2025 is installed on your server with secure boot enabled, and only the **HCE_Secure_Boot_RSA_Code-Signing_Authority_1.cer certificate** is imported, the OS cannot be booted from a restart after your OS is upgraded to HCE released after May 2025.

Detecting the Issue

Step 1 Check whether your OS version is HCE 2.0 released before March 2025. If yes, go to step 2. If no, your OS does not have this issue.

```
[root@localhost ~]# cat /etc/hce-latest
hceversion=HCE-2.0.2412.1_aarch64
compiletime=2025-02-28-10-00-01
```

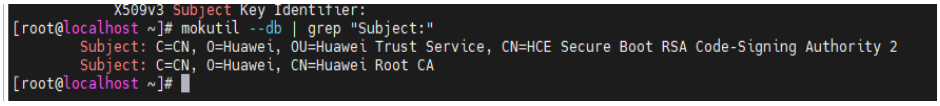
Step 2 Run **mokutil --sb** to check whether secure boot is enabled. If "SecureBoot enabled" is displayed, go to step 3. If not, your OS does not have this issue.

```
[root@localhost ~]# mokutil --sb
SecureBoot enabled
```

Step 3 Check the certificate imported to BIOS.

```
[root@localhost ~]# mokutil --db | grep "Subject:"
```

Figure 19-1 Checking the imported certificate



- If the command output contains "Huawei Root CA", your OS does not have this issue. No further action is required.
- If the command output does not contain "Huawei Root CA" but contains "Huawei Code-Signing Authority CA 3", your OS does not have this issue after it is upgraded to HCE 2.0 released after May 2025.
- If the command output contains neither "Huawei Root CA" nor "Huawei Code-Signing Authority CA 3" but contains "HCE Secure Boot RSA Code-Signing Authority 1", your OS will encounter this issue.

----End

Solution

Obtain the updated certificate in the HCE 2.0 image repository, decompress or install **hce-sign-certificate-1.0-2.hce2.x86_64.rpm** in the **https://repo.huaweicloud.com/hce/2.0/updates/x86_64/Packages/** directory, and import the new certificate **Huawei_Root_CA.cer** to BIOS.

- Reference for importing a certificate to BIOS:

  Kunpeng: https://support.huawei.com/enterprise/en/doc/EDOC1100088647/97a0d5a0

  2288H V5: https://support.huawei.com/enterprise/en/doc/EDOC1000163372/afc5c7f8?idPath=23710424|251364409|21782478|21872244

  2288H V6: https://support.huawei.com/enterprise/en/doc/EDOC1100195299/fdb56216?idPath=23710424|251364409|21782478|23692812

20 How Do I Bind NIC Interrupts to Cores?

With NIC interrupts bound to cores, data processing requests from each NIC can be allocated to their specified CPU core. This makes network data processing more efficient and improves the data throughput. This section describes how to configure NIC interrupt binding in HCE 2.0.

Prerequisites

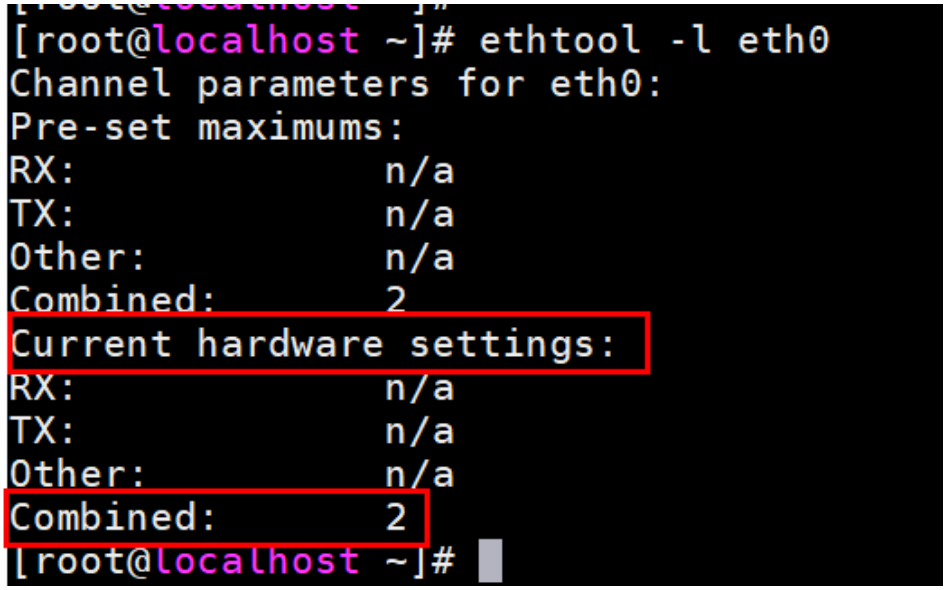
The irqbalance service must be disabled. You can run **systemctl stop irqbalance** to disable it. To disable the irqbalance service from automatically starting upon system startup, run **systemctl disable irqbalance**.

Procedure

1. Run **ethtool -l eth0** to check the number of NIC queues. In this example, there are two NIC queues.

   NOTE

   Replace **eth0** with your actual NIC name. The following steps use **eth0** as an example.



2. Run **lscpu** to check the number of CPU cores. In this example, there are four CPU cores.

```
[root@localhost ~]# lscpu
Architecture:                    x86_64
  CPU op-mode(s):                32-bit, 64-bit
  Address sizes:                 46 bits physical, 48 bits virtual
  Byte Order:                    Little Endian
CPU(s):                          4
  On-line CPU(s) list:           0-3
Vendor ID:                       GenuineIntel
```

3. Run the following command to check the interrupts of the current NIC queue (in this example, the IDs of interrupts for data input of eth0 are 25 and 27, and those for data output are 26 and 28):

```
cat /proc/interrupts | grep virtio0 | awk '{print $1 $(NF)}'
```

```
[root@localhost ~]# cat /proc/interrupts | grep virtio0 | awk '{print $1 $(NF)}'
24:virtio0-config
25:virtio0-input.0
26:virtio0-output.0
27:virtio0-input.1
28:virtio0-output.1
[root@localhost ~]#
```

NOTE

You can run the following command to view the mapping between NICs and VirtIO:

```
ethtool -i eth0 | grep bus-info | awk -F "bus-info:" '{print $2}' | xargs -I {} ls /sys/bus/pci/drivers/virtio-pci/{} | grep virtio
```

4. Check the existing binding.

   Run **cat /proc/irq/{25,26,27,28}/smp_affinity_list** to check the binding. In this example, interrupts 25, 26, 27, and 28 are bound to CPU3.

```
[root@localhost ~]# cat /proc/irq/{25,26,27,28}/smp_affinity_list
3
3
3
3
[root@localhost ~]#
```

5. Manually bind NIC interrupts.

   Bind data input interrupts of eth0 to CPU0 and CPU1.

```
echo 0 > /proc/irq/25/smp_affinity_list
echo 1 > /proc/irq/27/smp_affinity_list
```

   Bind data output interrupts of eth0 to CPU2 and CPU3.

```
echo 2 > /proc/irq/26/smp_affinity_list
echo 3 > /proc/irq/28/smp_affinity_list
```

6. Check the binding result.

   Run **cat /proc/irq/{25,26,27,28}/smp_affinity_list**. The command output shows that the binding is successful.

```
[root@localhost ~]# cat /proc/irq/{25,26,27,28}/smp_affinity_list
0
2
1
3
[root@localhost ~]#
```


21 How Do I Adjust the Threshold of memcpy in x86_64?

Background

The threshold of glibc's **memcpy** is determined by the parameter **x86_non_temporal_threshold**. It has a great impact on the memory bandwidth. You can adjust the threshold as needed to achieve better memory copy performance.

Method

The following setting is recommended by the glibc community:

```
export GLIBC_TUNABLES=glibc.cpu.x86_non_temporal_threshold=$(($(getconf LEVEL3_CACHE_SIZE) * 3 / 4))
```

memcpy

In glibc-2.34, **memcpy** and **memmove** are implemented in the similar way which is described in the glibc source code.

```
sysdeps/x86_64/multiarch/memmove-vec-unaligned-erms.S

/* memmove/memcpy/mempcpy is implemented as:
   1. Use overlapping load and store to avoid branch.
   2. Load all sources into registers and store them together to avoid
      possible address overlap between source and destination.
   3. If size is 8 * VEC_SIZE or less, load all sources into registers
      and store them together.
   4. If address of destination > address of source, backward copy
      4 * VEC_SIZE at a time with unaligned load and aligned store.
      Load the first 4 * VEC and last VEC before the loop and store
      them after the loop to support overlapping addresses.
   5. Otherwise, forward copy 4 * VEC_SIZE at a time with unaligned
      load and aligned store.  Load the last 4 * VEC and first VEC
      before the loop and store them after the loop to support
      overlapping addresses.
   6. If size >= __x86_shared_non_temporal_threshold and there is no
      overlap between destination and source, use non-temporal store
      instead of aligned store.  */
```

As described in item 6 above, if **__x86_shared_non_temporal_threshold** is exceeded, non-temporal stores instead of aligned stores will be used. Non-temporal stores use the **movntdq** instruction to bypass the CPU L3 cache and

directly access the memory. In this cache missing case, non-temporal stores omit the cache read and write and are more suitable for large memory copies than aligned stores.

22 What Do I Do If the resolv.conf File Is Modified by a Network Service Restart?

Symptom

After the system or the network service is restarted, the value of **nameserver** (DNS server IP addresses) in **/etc/resolv.conf** is changed.

Possible Cause

This may be caused by the **PEERDNS** and **RESOLV_MODS** parameters in the network interface configuration file.

During the restart of the network service, the **/etc/sysconfig/network-scripts/ifup-post** and **/etc/sysconfig/network-scripts/ifdown-post** scripts check for **RESOLV_MODS=no** and **PEERDNS=no** in the network interface configuration file (for example, **/etc/sysconfig/network-scripts/ifcfg-***). If the two parameters or either of them is not found, the scripts modify **/etc/resolv.conf**.

Parameter description:

1. **PEERDNS**: whether DNS servers can be changed in **/etc/resolv.conf**. If DHCP is used, the default value is **yes**.
   - **yes**: If DNS configurations exist in **/etc/resolv.conf**, change DNS servers in the file.
   - **no**: DNS servers in **/etc/resolv.conf** cannot be changed.
2. **RESOLV_MODS**: whether to write DNS servers into the configuration file.
   - **yes**: Write the values of **MS_DNS1** and **MS_DNS2** into **/etc/resolv.conf**.
   - **no**: DNS servers in **/etc/resolv.conf** cannot be changed.

Solution

Add the following content to the network interface configuration file **/etc/sysconfig/network-scripts/ifcfg-*** and restart the network service:

```
PEERDNS=no
RESOLV_MODS=no
```


23 How Do I Prevent the nohup Background Process from Being Killed After a VNC Session Ends?

Symptom

The nohup background process started in a VNC session is automatically killed by the system after the VNC session ends (for example, after you run **exit** to log out or a login times out).

Possible Cause

For security purposes, the upstream community modified the getty service and deleted **KillMode=process**. As a result, all processes are killed after the VNC login session ends. For details, see https://github.com/systemd/systemd/commit/021acbc188a53fa528161578305406c5c9c808b2.

Solution

You can modify the getty settings to retain the nohup background process.

Step 1 Run **vim /usr/lib/systemd/system/getty\@.service** to modify the getty service unit.

Step 2 Add **KillMode=process** to the **[Service]** section, and save the file.

```
[Service]
# the VT is cleared by TTYVTDisallocate
# The '-o' option value tells agetty to replace 'login' arguments with an
# option to preserve environment (-p), followed by '--' for safety, and then
# the entered username.
ExecStart=-/sbin/agetty -o '-p -- \\u' --noclear %I $TERM
Type=idle
Restart=always
RestartSec=0
UtmpIdentifier=%I
TTYPath=/dev/%I
TTYReset=yes
TTYVHangup=yes
TTYVTDisallocate=yes
IgnoreSIGPIPE=no
SendSIGHUP=yes
KillMode=process
```


Step 3 Run **systemctl daemon-reload** to reload services.

Step 4 Run **systemctl restart getty@`basename $(tty)`** to restart the getty service.

CAUTION

Restarting the getty service will exit the current session.

----End

24 How Do I Configure XPS? Does XPS Affect the System?

What Is XPS

Transmit Packet Steering (XPS) is designed to send data packets to specific transmit queues in a system with multi-queue NICs. It maps CPUs to transmit queues so that the kernel can automatically select transmit queues associated with a specific CPU. The kernel records the transmit queue selected for the first packet of a data flow and uses the queue to transmit subsequent packets. This reduces the overhead of selecting transmit queues for each packet.

XPS has the following advantages:

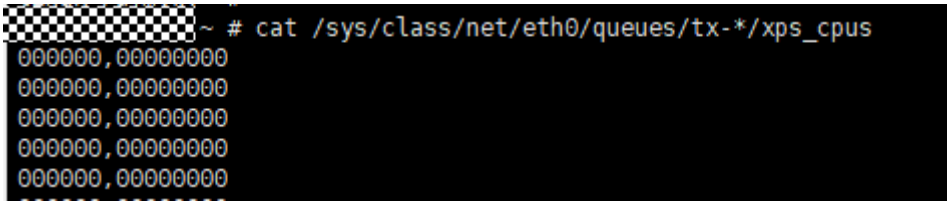
- CPUs do not have to compete as much for the same transmit queue, which reduces lock conflicts when NIC queues send data. Transmission is more efficient.
- The mapping between transmit queues and CPUs is consistent with the queue affinity of VirtIO NICs. This reduces cache misses during packet sending and cache invalidation caused by lock contention and thereby improves network transmission performance.

Configuring XPS

1. Check whether XPS is configured for your instance. (Ensure that **CONFIG_XPS** is enabled in the kernel.)

   NIC eth0 is used as an example.

   ```
   cat /sys/class/net/eth0/queues/tx-*/xps_cpus
   ```

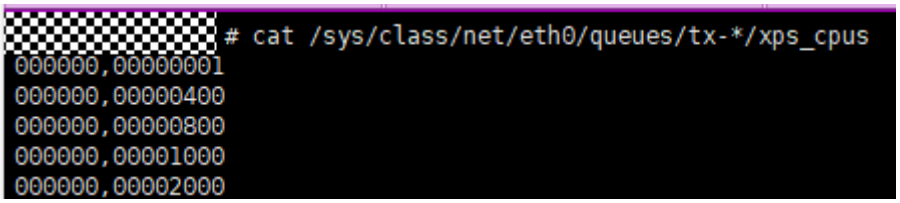
   

2. Configure XPS based on the number of CPUs and transmit queues. For details, see **Enabling NIC Multi-Queue**.
3. (Optional) Check whether XPS is configured.

   NIC eth0 is used as an example.


```
cat /sys/class/net/eth0/queues/tx-*/xps_cpus
```

The output depends on how many CPUs and queues there are. In the output shown here, XPS has been configured.



Clearing XPS

Although XPS is configured to improve network performance, network performance may be affected by XPS. If this problem occurs, run the command below to delete the XPS configuration. eth0 is the NIC that XPS was configured for.

```
sudo sh -c 'for txq in /sys/class/net/eth0/queues/tx-*; do echo 0 > $txq/xps_cpus; done'
```


25 How Do I Handle Unexpected Restarts?

Possible Cause 1: Out of Memory (OOM) Errors

- Symptom

  One or more processes used up all physical memory and swap space. To prevent a system hang, the kernel's OOM killer forcibly terminates the process that occupies the most memory. This may cause a chain reaction and the system restarts.

- Check method

  Check whether **/var/log/messages** contains keywords like "Out of memory: Killed process" or "invoked oom-killer".

- Solution

  a. Check service processes to see if there is memory leak, which causes OOM errors.

  b. If more memory is required by services, upgrade the ECS memory specifications.

Possible Cause 2: A Kernel Panic

- Symptom

  The kernel detects a severe, unrecoverable error (such as a driver bug, kernel bug, and hardware communication problem). The system will be frozen and may be automatically restarted.

- Check method

Step 1 Check whether **/var/log/messages** contains keywords like "Kernel panic - not syncing" or "Oops".

Step 2 Check the automatic restart settings.

```
cat /proc/sys/kernel/panic
# If the output is 0, the system is suspended after a panic occurs and waits for manual processing.
# If the output is N, the system automatically restarts N seconds after a panic occurs.
```

----End

- Solution

a. Roll back the kernel or drivers. If the problem occurs after a recent update, roll back the kernel or related drivers (such as graphics and network drivers) to the previous stable version.

b. Capture onsite information. To preserve the scene if the fault occurs again, you can temporarily disable automatic restart.

```
echo 0 > /proc/sys/kernel/panic # Temporarily disable automatic restart.
```

Possible Cause 3: Scheduled Restart

- Symptom

  A system restart is scheduled.

- Check method

  Check scheduled tasks.

  ```
  sudo grep -r "reboot\|shutdown" /etc/cron.d/ /etc/cron.hourly/ /etc/cron.daily/ /etc/cron.weekly/ /etc/cron.monthly/ /var/spool/cron/
  ```

- Solution

  Locate the scheduled restart task and delete it or comment it out.

26 What Do I Do If IMA Uses Too Much Memory?

Symptom

The slab memory usage keeps increasing. As a result, some services cannot be started. The **slabtop** command shows that kmalloc has taken up a lot of memory, and memory usage is still increasing.

Possible Cause

Integrity Measurement Architecture (IMA) measurement may use excessive kernel memory depending on policy settings. IMA, as part of the Linux open-source kernel, always stores measurement records in the kernel memory to ensure the reliability of measurement results.

Solution

Modify the cmdline to remove **integrity=1** and **ima_policy=tcb**.

