openEuler
20.03 LTS

Administrator Guide

Date 2020-04-10

Contents

Terms of Use

About This Document

Overview

This document provides common administrator operations of the openEuler system to help administrators better use the system.

Intended Audience

This document is intended for all administrators who use openEuler.

Symbol Conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
NOTICE	Indicates a potentially hazardous situation which, if not avoided, could result in equipment damage, data loss, performance deterioration, or unanticipated results. NOTICE is used to address practices not related to personal injury.
NOTE	Supplements the important information in the main text. NOTE is used to address information not related to personal injury, equipment damage, and environment deterioration.

1 Basic Configuration

1.1 Using Commands

1.1.1 Setting the System Locale

System locale settings are stored in the /etc/locale.conf file and can be modified by the localectl command. These settings are read at system boot by the systemd daemon.

Displaying the Current Locale Status

To display the current locale status, run the following command:

```
localectl status
```

Example command output:

```
$ localectl status
   System Locale: LANG=zh_CN.UTF-8
       VC Keymap: cn
      X11 Layout: cn
```

Listing Available Locales

To display available locales, run the following command:

```
localectl list-locales
```

You can check that by listing all Chinese locales with the following command:

```
$ localectl list-locales | grep zh
zh_CN.UTF-8
```

Setting the Locale

To set the language environment, run the following command as the user **root**. In the command, *locale* indicates the language type to be set. Run the **localectl list-locales** command to obtain the value range. Change the value based on the site requirements.

```
localectl set-locale LANG=locale
```

For example, if you want to use Simplified Chinese as the locale, run the following command as the user **root**:

```
# localectl set-locale LANG=zh_CN.UTF-8
```

NOTE

After the modification, log in again or run the following command to update the configuration file for the modification to take effect:

```
source /etc/locale.conf
```

1.1.2 Setting the Keyboard Layout

Keyboard layout settings are stored in the /etc/locale.conf file and can be modified by the localectl command. These settings are read at early boot by the systemd daemon.

Displaying the Current Settings

To display the current keyboard layout settings, run the following command:

```
localectl status
```

Example command output:

```
$ localectl status
   System Locale: LANG=zh_CN.UTF-8
       VC Keymap: cn
      X11 Layout: cn
```

Listing Available Keyboard Layouts

To list all available keyboard layouts that can be configured on openEuler, run the following command:

```
localectl list-keymaps
```

For example, the command output of the Chinese keyboard layout is as follows:

```
$ localectl list-keymaps | grep cn
cn
```

Setting the Keyboard Layout

To set the keyboard layout, run the following command as the user **root**. In the command, *map* indicates the keyboard layout to be set. Run the **localectl list-keymaps** command to obtain the value range. Change it based on the site requirements.

```
localectl set-keymap map
```

The keyboard layout will be equally applied to graphical user interfaces.

Then you can verify if your setting was successful by checking the status:

```
$ localectl status
   System Locale: LANG=zh CN.UTF-8
       VC Keymap: cn
      X11 Layout: us
```

1.1.3 Setting the Date and Time

This topic describes how to set the system date, time, and time zone by using timedatectl, date, and hwclock commands.

1.1.3.1 Using the timedatectl Command

Displaying the Current Date and Time

To display the current date and time, run the following command:

```
timedatectl
```

Example command output:

```
$ timedatectl
              Local time: Mon 2019-09-30 04:05:00 EDT
          Universal time: Mon 2019-09-30 08:05:00 UTC
                RTC time: Mon 2019-09-30 08:05:00
               Time zone: America/New York (EDT, -0400)
System clock synchronized: no
             NTP service: inactive
         RTC in local TZ: no
```

Synchronizing the System Clock with a Remote Server

Your system clock can be automatically synchronized with a remote server using the Network Time Protocol (NTP). Run the following command as the user **root** to enable or disable NTP. The value of *boolean* is **yes** or **no**, indicating that the NTP is enabled or disabled for automatic system clock synchronization. Change the value based on the site requirements.

NOTE

If the remote NTP server is enabled to automatically synchronize the system clock, you cannot manually change the date and time. If you need to manually change the date or time, ensure that automatic NTP system clock synchronization is disabled. You can run the **timedatectl set-ntp no** command to disable the NTP service.

```
timedatectl set-ntp boolean
```

For example, to enable automatic remote time synchronization, run the following command:

```
# timedatectl set-ntp yes
```

Changing the Current Date

NOTE

Before changing the date, ensure that automatic NTP system clock synchronization has been disabled.

Run the following command as the user **root** to change the current date. In the command, *YYYY* indicates the year, *MM* indicates the month, and *DD* indicates the day. Change them based on the site requirements.

```
timedatectl set-time YYYY-MM-DD
```

For example, to change the current date to August 14, 2019, run the following command as the user **root**:

```
# timedatectl set-time '2019-08-14'
```

Changing the Current Time

> **NOTE**
>
> Before changing the time, ensure that automatic NTP system clock synchronization has been disabled.

To change the current time, run the following command as the user **root**. In the command, *HH* indicates the hour, *MM* indicates the minute, and *SS* indicates the second. Change them based on the site requirements.

```
timedatectl set-time HH:MM:SS
```

For example, to change the current time to 15:57:24, run the following command:

```
# timedatectl set-time 15:57:24
```

Changing the Time Zone

To list all available time zones, run the following command:

```
timedatectl list-timezones
```

To change the current time zone, run the following command as the user **root**. In the command, *time_zone* indicates the time zone to be set. Change it based on the site requirements.

```
timedatectl set-timezone time_zone
```

Imagine you want to identify which time zone is closest to your present location while you are in Asia. You can check that by listing all available time zones in Asia with the following command:

```
# timedatectl list-timezones | grep Asia
Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Baghdad
Asia/Bahrain
......

Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Srednekolymsk
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Thimphu
Asia/Tokyo
```

To change the time zone to Asia/Shanghai, run the following command:

```
# timedatectl set-timezone Asia/Shanghai
```

1.1.3.2 Using the date Command

Displaying the Current Date and Time

To display the current date and time, run the following command:

```
date
```

By default, the **date** command displays the local time. To display the time in Coordinated Universal Time (UTC), run the command with the --utc or -u command line option:

```
date --utc
```

You can also customize the format of the displayed information by providing the + "format" option on the command line:

```
date +"format"
```

Table 1-1 Formatting options

Format Option	Description
%H	The hour in the HH format (for example, 17)
%M	The minute in the MM format (for example, 37)
%S	The second in the SS format (for example, 25)
%d	The day of the month in the DD format (for example, 15)
%m	The month in the MM format (for example, 07)
%Y	The year in the YYYY format (for example, 2019)
%Z	The time zone abbreviation (for example, CEST)
%F	The full date in the YYYY-MM-DD format (for example, 2019-7-15). This option is equal to %Y-%m-%d.
%T	The full time in the HH:MM:SS format (for example, 18:30:25). This option is equal to %H:%M:%S.

Example commands and outputs:

- To display the current date and time:

  ```
  $ date
  Sat Aug  17 17:26:34 CST 2019
  ```

- To display the current date and time in UTC:

  ```
  $ date --utc
  Sat Aug  17 09:26:18 UTC 2019
  ```

- To customize the output of the date command:

  ```
  $ date +"%Y-%m-%d %H:%M"
  2019-08-17 17:24
  ```

Changing the Current Time

To change the current time, run the date command with the --set or -s option as the root user: Run the following command as the user **root**. In the command, *HH* indicates the hour, *MM* indicates the minute, and *SS* indicates the second. Change them based on the site requirements.

```
date --set HH:MM:SS
```

By default, the date command sets the local time. To set the system clock in UTC instead, run the command with the --utc or -u command line option:

```
date --set HH:MM:SS --utc
```

For example, to change the current time to 23:26:00, run the following command as the user **root**:

```
# date --set 23:26:00
```

Changing the Current Date

To change the current date, run the command with the --set or -s command line option. Run the following command as the user **root**. In the command, *YYYY* indicates the year, *MM* indicates the month, and *DD* indicates the day. Change them based on the site requirements.

```
date --set YYYY-MM-DD
```

For example, to change the current date to November 2, 2019, run the following command as the user **root**:

```
# date --set 2019-11-02
```

1.1.3.3 Using the hwclock Command

You can run the hwclock command to set the real time clock (RTC).

Real-Time Clock and System Clock

Linux divides clocks into the following types:

- System clock: clock of the current Linux kernel.
- Hardware clock RTC: hardware clock of the mainboard powered by the battery. This clock can be set in the **Standard BIOS Feature** option of the BIOS.

When Linux starts, it reads the RTC and sets the system clock time based on the RTC time.

Displaying the Current Date and Time

To display the current RTC date and time, run the following command as the user **root**:

```
hwclock
```

Example command output:

```
# hwclock
2019-08-26 10:18:42.528948+08:00
```

Setting the Date and Time

Run the following command as the user **root** to change the date and time of the current hardware. In the command, *dd* indicates the day, *mm* indicates the month, *yyyy* indicates the year, *HH* indicates the hour, and *MM* indicates the minute. Change them based on the site requirements.

```
hwclock --set --date "dd mm yyyy HH:MM"
```

For example, to change the current time to 21:17 on October 21, 2019, run the following command:

```
# hwclock --set --date "21 Oct 2019 21:17" --utc
```

2 Viewing System Information

- Run the following command to view the system information:

  ```
  cat /etc/os-release
  ```

  For example, the command and output are as follows:

  ```
  # cat /etc/os-release
  NAME="openEuler"
  VERSION="20.03 (LTS)"
  ID="openEuler"
  VERSION_ID="20.03"
  PRETTY_NAME="openEuler 20.03 (LTS)"
  ANSI_COLOR="0;31"
  ```

- View system resource information.

  Run the following command to view the CPU information:

  ```
  lscpu
  ```

  Run the following command to view the memory information:

  ```
  free
  ```

  Run the following command to view the disk information:

  ```
  fdisk -l
  ```

3 User Management

In Linux, each common user has an account, including the user name, password, and home directory. There are also special users created for specific purposes, and the most important special user is the admin account whose default user name is root. In addition, Linux provides user groups so that each user belongs to at least one group, facilitating permission management.

The control of users and user groups is a core element of openEuler security management. This topic introduces the user and group management commands and explains how to assign privileges to common users in graphical user interface and on command lines.

3.1 Adding a User

useradd Command

Run the **useradd** command as the user **root** to add user information to the system. In the command, *options* indicates related parameters and *user_name* indicates the user name.

```
useradd [options] user_name
```

User Information Files

The following files contain user account information:

- /etc/passwd: user account information
- /etc/shadow file: user account encryption information
- /etc/group file: group information
- /etc/default/useradd: default configurations
- /etc/login.defs: system wide settings
- /etc/skel: default directory that holds initial configuration files

Example

For example, to create a user named user_example, run the following command as the user **root**:

```
# useradd user_example
```

NOTE

If no prompt is displayed, the user is successfully created. After the user is created, run the **passwd** command to assign a password to the user. A new account without a password will be banned.

To view information about the new user, run the **id** command:

```
# id user example
uid=502(user_example)    gid=502(user_example)    groups=502(user_example)
```

To change the password of the user_example, run the following command:

```
# passwd user_example
```

Then, enter the password and confirm it as prompted:

```
# passwd user example
Changing password for user user example.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

NOTE

If the command output contains **BAD PASSWORD: The password fails the dictionary check - it is too simplistic/sytematic**, the password is too simple and needs to be reset.

3.2 Modifying a User Account

Changing a Password

Common users can change their passwords using the **passwd** command. Only the admin is allowed to use the **passwd username** command to change passwords for other users.

Changing User's Login Shell

Common users can use the **chsh** command to change their login shell. Only the admin is allowed to run the **chsh username** command to change login shell for other users.

Users can also run the **usermod** command as the user **root** to modify the shell information. In the command, *new_shell_path* indicates the target shell path, and *username* indicates the user name to be modified. Change them based on the site requirements.

```
usermod -s new_shell_path username
```

For example, to change the shell of user_example to csh, run the following command:

```
# usermod -s /bin/csh user_example
```

Changing the Home Directory

- To change the home directory, run the following command as the user **root**. In the command, *new_home_directory* indicates the created target home directory, and *username* indicates the user name to be changed. Change them based on the site requirements.

  ```
  usermod -d new_home_directory username
  ```

- To move the content in the current home directory to a new one, run the usermod command with the -m option:

  ```
  usermod -d new_home_directory -m username
  ```

Changing a UID

To change the user ID, run the following command as the user **root**. In the command, *UID* indicates the target user ID and *username* indicates the user name. Change them based on the site requirements.

```
usermod -u UID username
```

The usermod command can change a user's UID in all files and directories under the user's home directory. However, for files outside the user's home directory, their owners can only be changed using the **chown** command.

Changing Account Expiry Date

If the shadow password is used, run the following command as the user **root** to change the validity period of an account. In the command, *MM*, *DD*, and *YY* indicate the month, day, and year, respectively, and *username* indicates the user name. Change them based on the site requirements.

```
usermod -e MM/DD/YY username
```

3.3 Deleting Users

Run the **userdel** command as the user **root** to delete an existing user.

For example, run the following command to delete user Test:

```
# userdel Test
```

If you also need to delete the user's home directory and all contents in the directory, run the **userdel** command with the -r option to delete them recursively.

> NOTE
>
> You are not advised to directly delete a user who has logged in to the system. To forcibly delete a user, run the **userdel -f** *Test* command.

3.4 Authorizing Administrator Accounts

3.4.1 Granting Rights to a Common User

The **sudo** command allows common users to execute commands that can be executed only by administrator accounts.

The **sudo** command allows the user specified in the **/etc/sudoers** file to execute the administrator account commands. For example, an authorized common user can run:

```
sudo /usr/sbin/useradd newuser1
```

The **sudo** command can specify a common user that has been added to the **/etc/sudoers** file to process tasks as required.

The information configured in the **/etc/sudoers** file is as follows:

- Blank lines or comment lines starting with **#**: Have no specific functions.
- Optional host alias lines: Create the name of a host list. The lines must start with **Host_Alias**. The host names in the list must be separated by commas (,). For example:

  ```
  Host_Alias  linux=ted1,ted2
  ```

  ted1 and **ted2** are two host names, which can be called **linux**.
- Optional user alias lines: Create the name of a user list. The lines must start with **User_Alias**. The user names in the list must be separated by commas (,). The user alias lines have the same format as the host alias lines.
- Optional command alias lines: Create the name of a command list. The lines must start with **Cmnd_Alias**. The commands in the list must be separated by commas (,).
- Optional running mode alias lines: Create the name of a user list. The difference is that such alias can enable a user in the list to run the **sudo** command.
- Necessary declaration lines for user access:

  The declaration syntax for user access is as follows:

  ```
  user host = [ run as user ] command list
  ```

  Set the user to a real user name or a defined user alias, and set the host to a real host name or a defined host alias. By default, all the commands executed by sudo are executed as user **root**. If you want to use another account, you can specify it. **command list** is either a command list separated by commas (,) or a defined command alias. For example:

  ```
  ted1   ted2=/sbin/shutdown
  ```

  In this example, ted1 can run the shutdown command on ted2.

  ```
  newuser1 ted1=(root) /usr/sbin/useradd,/usr/sbin/userdel
  ```

  This indicates that newuser1 on the ted1 host can run the **useradd** and **userdel** commands as the user **root**.

NOTE

- You can define multiple aliases in a line and separate them with colons (:).
- You can add an exclamation mark (!) before a command or a command alias to make the command or the command alias invalid.
- There are two keywords: ALL and NOPASSWD. ALL indicates all files, hosts, or commands, and NOPASSWD indicates that no password is required.
- By modifying user access, you can change the access permission of a common user to be the same as that of the user **root**. Then, you can grant rights to the common user.

The following is an example of the **sudoers** file:

```
#sudoers files
#User alias specification
User Alias ADMIN=ted1:POWERUSER=globus,ted2
#user privilege specification
ADMIN ALL=ALL
POWERUSER ALL=ALL,!/bin/su
```

In the preceding information:

- User_Alias ADMIN=ted1:POWERUSER=globus,ted2

  Two aliases ADMIN and POWERUSER are defined.
- ADMIN ALL=ALL

  ADMIN can run all commands as the user **root** on all hosts.
- POWERUSER ALL=ALL,!/bin/su

  POWERUSER can run all commands except the **su** command as the user **root** on all hosts.

4 Using the DNF to Manage Software Packages

DNF is a Linux software package management tool used to manage RPM software packages. The DNF can query software package information, obtain software packages from a specified software library, automatically process dependencies to install or uninstall software packages, and update the system to the latest available version.

NOTE

- DNF is fully compatible with YUM and provides YUM-compatible command lines and APIs for extensions and plug-ins.
- You must have the administrator rights to use the DNF. All commands in this chapter must be executed by the administrator.

4.1 Configuring the DNF

4.1.1 Modifying the Configuration File

The main configuration file of the DNF is /etc/dnf/dnf.conf. The **main** part in the file stores the global settings of the DNF. You can add one or more **repository** sections to the file to set the location of the software source to be installed.

In addition, the /etc/yum.repos.d directory stores one or more repo source files, which define different repositories.

You can configure a software source by either directly configuring the /etc/dnf/dnf.conf file or adding the .repo file to the /etc/yum.repos.d directory.

Modify the main Part

The /etc/dnf/dnf.conf file contains the **main** part. The following is an example of the configuration file:

```
[main]
gpgcheck=0
installonly_limit=3
clean_requirements_on_remove=True
best=True
```

Common options are as follows:

Table 4-1 main parameter description

Parameter	Description
cachedir	Cache directory for storing RPM packages and database files.
keepcache	The options are 1 and 0, indicating whether to cache the RPM packages and header files that have been successfully installed. The default value is 0, indicating that the RPM packages and header files are not cached.
debuglevel	Sets debugging information generated by the DNF. The value ranges from 0 to 10. A larger value indicates more detailed debugging information. The default value is 2. The value 0 indicates that the debug information is not displayed.
clean_requirements_on_remove	Deletes the dependency items that are no longer used during DNF removal. If the software package is installed through the DNF instead of the explicit user request, the software package can be deleted only through clean_requirements_on_remove, that is, the software package is introduced as a dependency item. The default value is **True**.
best	The system always attempts to install the latest version of the upgrade package. If the latest version cannot be installed, the system displays the cause and stops the installation. The default value is **True**.
obsoletes	The options are **1** and **0**, indicating whether to allow the update of outdated RPM packages. The default value is 1, indicating that the update is allowed.
gpgcheck	The options are **1** and **0**, indicating whether to perform GPG verification. The default value is **1**, indicating that verification is required.
plugins	The options are **1** and **0**, indicating that the DNF plug-in is enabled or disabled. The default value is **1**, indicating that the DNF plug-in is enabled.
installonly_limit	Sets the number of packages that can be installed at the same time by running the **installonlypkgs** command. The default value is 3. You are advised not to decrease the value.

Modify the repository Part

The repository part allows you to customize software source repositories. The name of each repository must be unique. Otherwise, conflicts may occur. The following is a minimum configuration example of the [repository] section:

```
[repository]
name=repository_name
baseurl=repository_url
```

Common options are as follows:

> NOTE
>
> openEuler provides an online image source at https://repo.openeuler.org/. For example, if the openEuler 20.03 version is aarch64, the **baseurl** can be set to https://repo.openeuler.org/openEuler-20.03-LTS/OS/aarch64/.

Table 4-2 repository parameter description

Parameter	Description
name=repository_name	Name string of a software repository.
baseurl=repository_url	Address of the software repository. • Network location using the HTTP protocol, for example, http://path/to/repo • Network location using the FTP protocol, for example, ftp://path/to/repo • Local path: for example, file:///path/to/local/repo

Displays the Current Configuration

- To display the current configuration information, run the following command:

  ```
  dnf config-manager --dump
  ```

- To display the configuration of a software source, query the repo id:

  ```
  dnf repolist
  ```

  Run the following command to display the software source configuration of the corresponding ID. In the command, *repository* indicates the repository ID.

  ```
  dnf config-manager --dump repository
  ```

- You can also use a global regular expression to display all matching configurations.

  ```
  dnf config-manager --dump glob_expression
  ```

4.1.2 Creating a Local Software Repository

To create a local repository of software sources, perform the following steps.

1. Install the createrepo software package. Run the following command as the root user:

   ```
   dnf install createrepo
   ```

2. Copy the required software packages to a directory, for example, /mnt/local_repo/.
3. Run the following command to create a software source:

```
createrepo --database /mnt/local_repo
```

4.1.3 Adding, Enabling, and Disabling Software Sources

This section describes how to add, enable, and disable the software source repository by running the **dnf config-manager** command.

Adding Software Source

To define a new software repository, you can add the repository part to the /etc/dnf/dnf.conf file or add the .repo file to the /etc/yum.repos.d/ directory. You are advised to add the .repo file. Each software source has its own .repo file. The following describes how to add the .repo file.

To add such a source to your system, run the following command as the user **root**. After the command is executed, the corresponding .repo file is generated in the **/etc/yum.repos.d/** directory. In the command, *repository_url* indicates the repo source address. For details, see Table 4-2.

```
dnf config-manager --add-repo repository_url
```

Enabling a Software Repository

To enable the software source, run the following command as the user **root**. In the command, *repository* indicates the repository ID in the new .repo file. You can run the **dnf repolist** command to query the repository ID.

```
dnf config-manager --set-enable repository
```

You can also use a global regular expression to enable all matching software sources. In the command, *glob_expression* indicates the regular expression used to match multiple repository IDs.

```
dnf config-manager --set-enable glob_expression
```

Disabling a Software Repository

To disable a software source, run the following command as the user **root**:

```
dnf config-manager --set-disable repository
```

You can also use a global regular expression to disable all matching software sources.

```
dnf config-manager --set-disable glob_expression
```

4.2 Managing Software Package

The DNF enables you to query, install, and delete software packages.

Searching for Software Packages

You can search for the required RPM package by its name, abbreviation, or description. The command is as follows:

```
dnf search term
```

The following is an example:

```
$  dnf search httpd
======================================== N/S matched: httpd
========================================
httpd.aarch64 : Apache HTTP Server
httpd-devel.aarch64 : Development interfaces for the Apache HTTP server
httpd-manual.noarch : Documentation for the Apache HTTP server
httpd-tools.aarch64  : Tools for use with the Apache HTTP Server
libmicrohttpd.aarch64  : Lightweight library for embedding a webserver in applications
mod_auth_mellon.aarch64  : A SAML 2.0 authentication module for the Apache Httpd Server
mod_dav_svn.aarch64  : Apache httpd module for Subversion server
```

Listing Software Packages

To list all installed and available RPM packages in the system, run the following command:

```
dnf list all
```

To list a specific RPM package in the system, run the following command:

```
dnf list glob_expression...
```

The following is an example:

```
$ dnf list httpd
Available Packages
httpd.aarch64            2.4.34-8.h5.oe1         Local
```

Displaying RPM Package Information

To view information about one or more RPM packages, run the following command:

```
dnf info package_name...
```

The following is a command example:

```
$ dnf info httpd
Available Packages
Name       : httpd
Version    : 2.4.34
Release    : 8.h5.oe1
Arch       : aarch64
Size       : 1.2 M
Repo       : Local
Summary    : Apache HTTP Server
URL        : http://httpd.apache.org/
License    : ASL 2.0
Description : The Apache HTTP Server is a powerful, efficient, and extensible
           : web server.
```

Installing an RPM Package

To install a software package and all its dependencies that have not been installed, run the following command as the user **root**:

```
dnf install package_name
```

You can also add software package names to install multiple software packages at the same time. Add the **strict=False** parameter to the /etc/dnf/dnf.conf configuration file and run the **dnf** command to add --setopt=strict=0. Run the following command as the user **root**:

```
dnf install package_name package_name... --setopt=strict=0
```

The following is an example:

```
# dnf install httpd
```

NOTE

If the RPM package fails to be installed, see 11.5 Installation Failure Caused by Software Package Conflict, File Conflict, or Missing Software Package.

Downloading Software Packages

To download the software package using the DNF, run the following command as the user **root**:

```
dnf download package_name
```

If you need to download the dependency packages that are not installed, add **--resolve**. The command is as follows:

```
dnf download --resolve package_name
```

The following is an example:

```
# dnf download --resolve httpd
```

Deleting a Software Package

To uninstall the software package and related dependent software packages, run the following command as the user **root**:

```
dnf remove package_name...
```

The following is an example:

```
# dnf remove totem
```

4.3 Managing Software Package Groups

A software package set is a group of software packages that serve a common purpose, for example, a system tool set. You can use the DNF to install or delete software package groups, improving operation efficiency.

Listing Software Package Groups

The summary parameter can be used to list the number of all installed software package groups, available groups, and available environment groups in the system. The command is as follows:

```
dnf groups summary
```

The following is an example:

```
# dnf groups summary
Last metadata expiration check: 0:11:56 ago on Sat 17 Aug 2019 07:45:14 PM CST.
Available Groups: 8
```

To list all software package groups and their group IDs, run the following command:

```
dnf group list
```

The following is an example:

```
# dnf group list
Last metadata expiration check: 0:10:32 ago on Sat 17 Aug 2019 07:45:14 PM CST.
Available Environment Groups:
   Minimal Install
   Custom Operating System
   Server
Available Groups:
   Development Tools
   Graphical Administration Tools
   Headless Management
   Legacy UNIX Compatibility
   Network Servers
   Scientific Support
   Security Tools
   System Tools
```

Displaying the Software Package Group Information

To list the mandatory and optional packages contained in a software package group, run the following command:

```
dnf group info glob_expression...
```

The following is an example of displaying the Development Tools information:

```
# dnf group info "Development Tools"
Last metadata expiration check: 0:14:54 ago on Wed 05 Jun 2019 08:38:02 PM CST.

Group: Development Tools
 Description: A basic development environment.
 Mandatory Packages:
   binutils
   glibc-devel
   make
   pkgconf
   pkgconf-m4
   pkgconf-pkg-config
   rpm-sign
 Optional Packages:
   expect
```

Installation Software Package Group

Each software package group has its own name and corresponding group ID. You can use the software package group name or its ID to install the software package.

To install a software package group, run the following command as the user **root**:

```
dnf group install group_name
dnf group install groupid
```

For example, to install the software package group of Development Tools, run the following command:

```
# dnf group install "Development Tools"
# dnf group install development
```

Deleting a Software Package Group

To uninstall a software package group, you can use the group name or ID to run the following command as the user **root**:

```
dnf group remove group_name
dnf group remove groupid
```

For example, to delete the software package group of Development Tools, run the following command:

```
# dnf group remove "Development Tools"
# dnf group remove development
```

4.4 Check and Update

You can use the DNF to check whether any software package in your system needs to be updated. You can use the DNF to list the software packages to be updated. You can choose to update all packages at a time or update only specified packages.

Checking For Update

To list all currently available updates, run the following command:

```
dnf check-update
```

The following is an example:

```
# dnf check-update
Last metadata expiration check: 0:02:10 ago on Sun 01 Sep 2019 11:28:07 PM  CST.

anaconda-core.aarch64      19.31.123-1.14           updates
anaconda-gui.aarch64       19.31.123-1.14           updates
anaconda-tui.aarch64       19.31.123-1.14           updates
anaconda-user-help.aarch64  19.31.123-1.14           updates
anaconda-widgets.aarch64    19.31.123-1.14           updates
bind-libs.aarch64         32:9.9.4-29.3            updates
bind-libs-lite.aarch64     32:9.9.4-29.3            updates
bind-license.noarch        32:9.9.4-29.3            updates
bind-utils.aarch64         32:9.9.4-29.3            updates
...
```

Upgrade

To upgrade a single software package, run the following command as the user **root**:

```
dnf update package_name
```

For example, to upgrade the RPM package, run the following command:

```
# dnf update anaconda-gui.aarch64
Last metadata expiration check: 0:02:10 ago on Sun 01 Sep 2019 11:30:27 PM  CST.
Dependencies Resolved
================================================================================
 Package               Arch        Version            Repository    Size
================================================================================
Updating:
 anaconda-gui           aarch64     19.31.123-1.14      updates      461 k
 anaconda-core          aarch64     19.31.123-1.14      updates      1.4 M
 anaconda-tui           aarch64     19.31.123-1.14      updates      274 k
 anaconda-user-help      aarch64     19.31.123-1.14      updates      315 k
 anaconda-widgets        aarch64     19.31.123-1.14      updates      748 k

Transaction Summary
================================================================================
Upgrade  5 Package

Total download size: 3.1 M
Is this ok [y/N]:
```

Similarly, to upgrade a software package group, run the following command as the user **root**:

```
dnf group update group_name
```

Updating All Packages and Their Dependencies

To update all packages and their dependencies, run the following command as the user **root**:

```
dnf update
```

5 Service Management

This topic describes how to manage your operating system and services using the systemd.

5.1 Introduction to systemd

The systemd is a system and service manager for Linux operating systems. It is designed to be backward compatible with SysV and LSB init scripts, and provides a number of features such as Socket & D-Bus based activation of services, on-demand activation of daemons, system state snapshots, and mount & automount point management. With systemd, the service control logic and parallelization are refined.

Systemd Units

In systemd, the targets of most actions are units, which are resources systemd know how to manage. Units are categorized by the type of resources they represent and defined in unit configuration files. For example, the avahi.service unit represents the Avahi daemon and is defined in the **avahi.service** file. Table 5-1 lists available types of systemd units.

Table 5-1 Available types of systemd units

Unit Type	File Extension	Description
Service unit	.service	A system service.
Target unit	.target	A group of systemd units.
Automount unit	.automount	A file system automount point.
Device unit	.device	A device file recognized by the kernel.
Mount unit	.mount	A file system mount point.

Unit Type	File Extension	Description
Path unit	.path	A file or directory in a file system.
Scope unit	.scope	An externally created process.
Slice unit	.slice	A group of hierarchically organized units that manage system processes.
Snapshot unit	.snapshot	A saved state of the systemd manager.
Socket unit	.socket	An inter-process communication socket.
Swap unit	.swap	A swap device or a swap file.
Timer unit	.timer	A systemd timer.

All available types of systemd units are located in one of the following directories listed in Table 5-2.

Table 5-2 Locations of available systemd units

Directory	Description
/usr/lib/systemd/system/	Systemd units distributed with installed RPM packages.
/run/systemd/system/	Systemd units created at runtime.
/etc/systemd/system/	Systemd units created and managed by the system administrator.

5.2 Features

Fast Activation

The systemd provides more aggressive parallelization than UpStart. The use of Socket- and D-Bus based activation reduces the time required to boot the operating system.

To accelerate system boot, systemd seeks to:

- Activate only the necessary processes
- Activate as many processes as possible in parallel

On-Demand Activation

During SysVinit initialization, it activates all the possible background service processes that might be used. Users can log in only after all these service processes are activated. The drawbacks in SysVinit are obvious: slow system boot and a waste of system resources.

Some services may rarely or even never be used during system runtime. For example, CUPS, printing services are rarely used on most servers. SSHD is rarely accessed on many servers. It is unnecessary to spend time on starting these services and system resources.

systemd can only be activated when a service is requested. If the service request is over, systemd stops.

Service Lifecycle Management by Cgroups

An important role of an init system is to track and manage the lifecycle of services. It can start and stop a service. However, it is more difficult than you could ever imagine to encode an init system into stopping services.

Service processes often run in background as daemons and sometimes fork twice. In UpStart, the expect stanza in the configuration file must be correctly configured. Otherwise, UpStart is unable to learn a daemon's PID by counting the number of forks.

Things are made simpler with Cgroups, which have long been used to manage system resource quotas. The ease of use comes largely from its file-system-like user interface. When a parent service creates a child service, the latter inherits all attributes of the Cgroup to which the parent service belongs. This means that all relevant services are put into the same Cgroup. The systemd can find the PIDs of all relevant services simply by traversing their control group and then stop them one by one.

Mount and Automount Point Management

In traditional Linux systems, users can use the **/etc/fstab** file to maintain fixed file system mount points. These mount points are automatically mounted during system startup. Once the startup is complete, these mount points are available. These mount points are file systems critical to system running, such as the **HOME** directory. Like SysVinit, systemd manages these mount points so that they can be automatically mounted at system startup. systemd is also compatible with the **/etc/fstab** file. You can continue to use this file to manage mount points.

There are times when you need to mount or unmount on demand. For example, a temporary mounting point is required for you to access the DVD content, and the mounting point is canceled (using the **umount** command) if you no longer need to access the content, thereby saving resources. This is traditionally achieved using the autofs service.

The systemd allows automatic mount without a need to install autofs.

Transactional Dependency Management

System boot involves a host of separate jobs, some of which may be dependent on each other. For example, a network file system (NFS) can be mounted only after network connectivity is activated. The systemd can run a large number of dependent jobs in parallel, but not all of them. Looking back to the NFS example, it is impossible to mount NFS and activate network at the same time. Before running a job, systemd calculates its dependencies, creates a temporary transaction, and verifies that this transaction is consistent (all relevant services can be activated without any dependency on each other).

Compatibility with SysVinit Scripts

Like UpStart, systemd introduces new configuration methods and has new requirements for application development. If you want to replace the currently running initialization system with systemd, systemd must be compatible with the existing program. It is difficult to modify all the service code in any Linux distribution in a short time for the purpose of using systemd.

The systemd provides features compatible with SysVinit and LSB initscripts. You do not need to modify the existing services and processes in the system. This reduces the cost of migrating

the system to systemd, making it possible for users to replace the existing initialization system with systemd.

System State Snapshots and System Restoration

The systemd can be started on demand. Therefore, the running status of the system changes dynamically, and you cannot know the specific services that are running in the system. systemd snapshots enable the current system running status to be saved and restored.

For example, if services A and B are running in the system, you can run the **systemd** command to create a snapshot for the current system running status. Then stop process A or make any other change to the system, for example, starting process C. After these changes, run the snapshot restoration command of systemd to restore the system to the point at which the snapshot was taken. That is, only services A and B are running. A possible application scenario is debugging. For example, when an exception occurs on the server, a user saves the current status as a snapshot for debugging, and then perform any operation, for example, stopping the service. After the debugging is complete, restore the snapshot.

5.3 Managing System Services

The systemd provides the systemctl command to start, stop, restart, view, enable, and disable system services.

Comparison Between SysVinit and systemd Commands

The **systemctl** command from the **systemd** command has the functions similar to the **SysVinit** command. Note that the **service** and **chkconfig** commands are supported in this version. For details, see Table 5-3. You are advised to manage system services by running the **systemctl** command.

Table 5-3 Comparison between SysVinit and systemd commands

SysVinit Command	systemd Command	Description
service *network* start	systemctl start *network*.service	Starts a service.
service *network* stop	systemctl stop *network*.service	Stops a service.
service *network* restart	systemctl restart *network*.service	Restarts a service.
service *network* reload	systemctl reload *network*.service	Reloads a configuration file without interrupting an operation.
service *network* condrestart	systemctl condrestart *network*.service	Restarts a service only if it is running.
service *network* status	systemctl status *network*.service	Checks the service running status.
chkconfig *network* on	systemctl enable *network*.service	Enables a service when the service activation time arrives or a trigger condition for enabling the service is

SysVinit Command	systemd Command	Description
		met.
chkconfig *network* off	systemctl disable *network*.service	Disables a service when the service activation time arrives or a trigger condition for disabling the service is met.
chkconfig *network*	systemctl is-enabled *network*.service	Checks whether a service is enabled.
chkconfig --list	systemctl list-unit-files --type=service	Lists all services in each runlevel and checks whether they are enabled.
chkconfig *network* --list	ls /etc/systemd/system/*.wants/*network*.service	Lists the runlevels in which a service is enabled and those in which the service is disabled.
chkconfig *network* --add	systemctl daemon-reload	Used when you need to create a service file or change settings.

Listing Services

To list all currently loaded services, run the following command:

```
systemctl list-units --type service
```

To list all services regardless of whether they are loaded, run the following command (with the all option):

```
systemctl list-units --type service --all
```

Example list of all currently loaded services:

```
$ systemctl list-units --type service
UNIT                     LOAD   ACTIVE    SUB     JOB   DESCRIPTION
atd.service               loaded active    running      Deferred execution scheduler
auditd.service            loaded active    running      Security Auditing Service
avahi-daemon.service       loaded active    running      Avahi mDNS/DNS-SD Stack
chronyd.service           loaded active    running      NTP client/server
crond.service             loaded active    running      Command Scheduler
dbus.service              loaded active    running      D-Bus System Message Bus
dracut-shutdown.service    loaded active    exited       Restore /run/initramfs on
shutdown
firewalld.service          loaded active    running      firewalld - dynamic firewall
daemon
getty@tty1.service         loaded active    running      Getty on tty1
gssproxy.service          loaded active    running      GSSAPI Proxy Daemon
irqbalance.service         loaded active    running      irqbalance daemon
iscsid.service            loaded activating start   start Open-iSCSI
```

Displaying Service Status

To display the status of a service, run the following command:

```
systemctl status name.service
```

Table 5-4 describes the parameters in the command output.

Table 5-4 Output parameters

Parameter	Description
Loaded	Information on whether the service has been loaded, the absolute path to the service file, and a note of whether the service is enabled.
Active	Information on whether the service is running and a time stamp.
Main PID	PID of the service.
CGroup	Additional information about related control groups.

To verify whether a particular service is running, run the following command:

```
systemctl is-active name.service
```

The output of the **is-active** command is as follows:

Table 5-5 Output of the is-active command

Status	Description
active(running)	One or more services are running in the system.
active(exited)	A service that ends properly after being executed only once. Currently, no program is running in the system. For example, the **quotaon** function is performed only when the program is started or mounted.
active(waiting)	The program needs to wait for other events to continue running. For example, the print queue service is being started, but it needs to be queued (print jobs) so that it can continue to wake up the printer service to perform the next print function.
inactive	The service is not running.

Similarly, to determine whether a particular service is enabled, run the following command:

```
systemctl is-enabled name.service
```

The output of the **is-enabled** command is as follows:

Table 5-6 Output of the is-enabled command

Status	Description
"enabled"	Has been permanently enabled through **Alias=** *Alias*, **.wants/**, or **.requires/** soft link in the **/etc/systemd/system/** directory.
"enabled-runtime"	Has been temporarily enabled through **Alias=** *Alias*, **.wants/**, or **.requires/** soft link in the **/run/systemd/system/** directory.
"linked"	Although the unit file is not in the standard unit directory, one or more soft links pointing to the unit file exist in the **/etc/systemd/system/** permanent directory.
"linked-runtime"	Although the unit file is not in the standard unit directory, one or more soft links pointing to the unit file exist in the **/run/systemd/system/** temporary directory.
"masked"	Has been masked permanently by the **/etc/systemd/system/** directory (soft link to **/dev/null**). Therefore, the **start** operation fails.
"masked-runtime"	Has been masked temporarily by the **/run/systemd/systemd/** directory (soft link to **/dev/null**). Therefore, the **start** operation fails.
"static"	Not enabled. There is no option available for the **enable** command in the [Install] section of the unit file.
"indirect"	Not enabled. But the list of values for the **Also=** option in the [Install] section of the unit file is not empty (that is, some units in the list may have been enabled), or the unit file has an alias soft link which is not in the **Also=** list. For a template unit, it indicates that an instance different from **DefaultInstance=** is enabled.
"disabled"	Not enabled. But the [Install] section of the unit file contains options available for the **enable** command.
"generated"	The unit file is dynamically generated by the unit generator. The generated unit file may not be directly enabled, but is implicitly enabled by the unit generator.
"transient"	The unit file is dynamically and temporarily generated by the **runtime** API. The temporary unit may not be enabled.
"bad"	The unit file is incorrect or other errors occur. **is-enabled** does not return this status, but displays an error message. The **list-unit-files** command may display this unit.

For example, to display the status of gdm.service, run the **systemctl status gdm.service** command.

```
# systemctl status gdm.service
gdm.service - GNOME Display Manager   Loaded: loaded
```

```
(/usr/lib/systemd/system/gdm.service; enabled)   Active: active (running) since Thu
2013-10-17 17:31:23 CEST; 5min ago
 Main PID: 1029 (gdm)
   CGroup: /system.slice/gdm.service
           ├─1029 /usr/sbin/gdm
           ├─1037 /usr/libexec/gdm-simple-slave --display-id /org/gno...
           └─1047 /usr/bin/Xorg :0 -background none -verbose -auth /r...Oct 17 17:31:23
localhost systemd[1]: Started GNOME Display Manager.
```

Starting a Service

To start a service, run the following command as the user **root**:

```
systemctl start name.service
```

For example, to start the httpd service, run the following command:

```
# systemctl start httpd.service
```

Stopping a Service

To stop a service, run the following command as the user **root**:

```
systemctl stop name.service
```

For example, to stop the Bluetooth service, run the following command:

```
# systemctl stop bluetooth.service
```

Restarting a Service

To restart a service, run the following command as the user **root**:

```
systemctl restart name.service
```

This command stops the selected service in the current session and immediately starts it again. If the selected service is not running, this command starts it too.

For example, to restart the Bluetooth service, run the following command:

```
# systemctl restart bluetooth.service
```

Enabling a Service

To configure a service to start automatically at system boot time, run the following command as the user **root**:

```
systemctl enable name.service
```

For example, to configure the httpd service to start automatically at system boot time, run the following command:

```
# systemctl enable httpd.service
ln -s '/usr/lib/systemd/system/httpd.service'
'/etc/systemd/system/multi-user.target.wants/httpd.service'
```

Disabling a Service

To prevent a service from starting automatically at system boot time, run the following command as the user **root**:

```
systemctl disable name.service
```

For example, to prevent the Bluetooth service from starting automatically at system boot time, run the following command:

```
# systemctl disable bluetooth.service
Removed /etc/systemd/system/bluetooth.target.wants/bluetooth.service.
Removed /etc/systemd/system/dbus-org.bluez.service.
```

5.4 Changing a Runlevel

Targets and Runlevels

In systemd, the concept of runlevels has been replaced with systemd targets to improve flexibility. For example, you can inherit an existing target and turn it into your own target by adding other services. Table 5-7 provides a complete list of runlevels and their corresponding systemd targets.

Table 5-7 Mapping between runlevels and targets

Runlevel	systemd Target	Description
0	runlevel0.target, poweroff.target	The operating system is powered off.
1, s, single	runlevel1.target, rescue.target	The operating system is in single user mode.
2, 4	runlevel2.target, runlevel4.target, multi-user.target	The operating system is in user-defined or domain-specific runlevel (by default, it is equivalent to runlevel 3).
3	runlevel3.target, multi-user.target	The operating system is in non-graphical multi-user mode, and can be accessed from multiple consoles or networks.
5	runlevel5.target, graphical.target	The operating system is in graphical multi-user mode. All the services running at level 3 can be accessed through graphical login.
6	runlevel6.target, reboot.target	The operating system is rebooted.
emergency	emergency.target	Emergency shell.

Viewing the Default Startup Target

Run the following command to view the default startup target of the system:

```
systemctl get-default
```

Viewing All Startup Targets

Run the following command to view all startup targets of the system:

```
systemctl list-units --type=target
```

Changing the Default Target

To change the default target, run the following command as the user **root**:

```
systemctl set-default name.target
```

Changing the Current Target

To change the current target, run the following command as the user **root**:

```
systemctl isolate name.target
```

Changing to Rescue Mode

To change the operating system to rescue mode, run the following command as the user **root**:

```
systemctl rescue
```

This command is similar to the **systemctl isolate rescue.target** command. After the command is executed, the following information is displayed on the serial port:

```
You are in rescue mode. After logging in, type "journalctl -xb" to viewsystem logs,
"systemctl reboot" to reboot, "systemctl default" or "exit"to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
```

NOTE

You need to restart the system to enter the normal working mode from the rescue mode.

Changing to Emergency Mode

To change the operating system to emergency mode, run the following command as the user **root**:

```
systemctl emergency
```

This command is similar to the **systemctl isolate emergency.target** command. After the command is executed, the following information is displayed on the serial port:

```
You are in emergency mode. After logging in, type "journalctl -xb" to viewsystem logs,
"systemctl reboot" to reboot, "systemctl default" or "exit"to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
```

NOTE

You need to restart the system to enter the normal working mode from the emergency mode.

5.5 Shutting Down, Suspending, and Hibernating the Operating System

systemctl Command

The systemd uses the systemctl command instead of old Linux system management commands to shut down, restart, suspend, and hibernate the operating system. Although previous Linux system management commands are still available in systemd for compatibility reasons, you are advised to use **systemctl** when possible. The mapping relationship is shown in Table 5-8.

Table 5-8 Mapping between old Linux system management commands and systemctl

Linux Management Command	systemctl Command	Description
halt	systemctl halt	Shuts down the operating system.
poweroff	systemctl poweroff	Powers off the operating system.
reboot	systemctl reboot	Reboots the operating system.

Shutting Down the Operating System

To shut down the system and power off the operating system, run the following command as the user **root**:

```
systemctl poweroff
```

To shut down the operating system without powering it off, run the following command as the user **root**:

```
systemctl halt
```

By default, running either of these commands causes systemd to send an informative message to all login users. To prevent systemd from sending this message, run this command with the **--no-wall** option. The command is as follows:

```
systemctl --no-wall poweroff
```

Restarting the Operating System

To restart the operating system, run the following command as the user **root**:

```
systemctl reboot
```

By default, running either of these commands causes systemd to send an informative message to all login users. To prevent systemd from sending this message, run this command with the **--no-wall** option. The command is as follows:

```
systemctl --no-wall reboot
```

Suspending the Operating System

To suspend the operating system, run the following command as the user **root**:

```
systemctl suspend
```

Hibernating the Operating System

To hibernate the operating system, run the following command as the user **root**:

```
systemctl hibernate
```

To suspend and hibernate the operating system, run the following command as the user **root**:

```
systemctl hybrid-sleep
```

6 Process Management

This topic explains how Linux kernel manages processes. It also provides examples to help you better understand common process control commands, at and cron services, as well as process query commands.

6.1 Managing System Processes

The operating system manages multiple user requests and tasks. In most cases, the operating system comes with only one CPU and one main memory, but it may have multiple tier-2 disks and input/output (I/O) devices. Therefore, users have to share resources, but it appears to users that they are exclusively occupying resources. The operating system places user tasks, OS tasks, emailing, print tasks, and other pending tasks in the queue and schedules the tasks according to predefined rules. In this topic, you will know how the operating system manages processes.

6.1.1 Scheduling a Process

The time-consuming and resource-demanding part of maintenance work is often performed at late night. You can arrange relevant processes to get started at the scheduled time instead of staying up all night. Here, we will explain the process scheduling commands.

6.1.1.1 Using the at Command to Run Processes at the Scheduled Time

Function

The at command is used to run a batch of processes (a series of commands) at the scheduled time or time+date.

Syntax of the at command:

```
at [-V] [-q queue] [-f filename] [-mldbv] time
at -c job [job...]
```

Time Format

The scheduled time can be in any of the following formats:

- hh:mm today: If hh:mm is earlier than the current time, the selected commands will be run at hh:mm the next day.
- midnight, noon, teatime (typically at 16:00), or the like
- 12-hour format followed by am or pm
- Time + date (month day, mm/dd/yy, or dd.mm.yy) The scheduled date must follow the scheduled time.

The scheduled time can also be relative time, which is suitable for scheduling commands that are going to be executed soon. For example, now+*N* minutes, hours, days, or weeks. *N* is time, which may be a few days or hours. Further, the scheduled time can be words like today, tomorrow, or the like. Here are some examples of the scheduled time.

Imagine the current time is 12:30 June 7 2019 and you want to run a command at 4:30 pm. The scheduled time in the at command can be any of the following:

```
at 4:30pm
at 16:30
at 16:30 today
at now+4 hours
at now+ 240 minutes
at 16:30 7.6.19
at 16:30 6/7/19
at 16:30 Jun 7
```

Although you can select any of the preceding examples according to your preference, absolute time in 24-hour format, such as at 16:30 6/7/19, is recommended.

Privileges

Only commands from standard input or from the file specified by the -f option can be scheduled by the at command to be executed. If the su command is executed to switch the operating system from user A to user B and then the at command is executed at the shell prompt of user B, the at command execution result is sent to user B. whereas emails (if any) are sent to user A.

For example, to run the slocate -u command at 10 am on June 8, 2019, perform the following steps:

```
# at  10:00  6/8/19
at> slocate -u
at>
[1]+  Stopped    at  10:00  6/8/19
```

When the at> prompt appears, type **slocate -u** and press Enter. Repeat substep 2 to add other commands that need to be run at 10 am on 8 June 2015. Then, press Ctrl+d to exit the at command.

The administrator is authorized to run the at command unconditionally. For other users, their privilege to run the at command is defined in /etc/at.allow and /etc/at.deny files.

6.1.1.2 Using the cron Service to Run Commands Periodically

The at command can run commands at the scheduled time but only once. It means that after the running command is specified, the system completes the task at the specified time. If you need to run commands repeatedly, the cron service is a good helper.

Cron Service

The **cron** service searches the **/var/spool/cron** directory for **crontab** files named by the user name in the /etc/passwd file and loads the search results into memory to execute the commands in the **crontab** files. Each user has a crontab file, with the file name being the same as the user name. For example, the **crontab** file of the **globus** user is **/var/spool/cron/globus**.

The **cron** service also reads the cron configuration file **/etc/crontab** every minute, which can be edited in various formats. If no crontab files are found, the **cron** service enters sleep mode and releases system resources. One minute later, the **cron** service is awoken to repeat the search work and command execution. Therefore, the background process occupies few resources and is wakened up every minute to check whether there are commands to be executed.

Command execution results are then mailed to users specified by the environment variable MAILTO in the /etc/crontab file. The **cron** service, once started, does not require manual intervention except when you need to replace periodic commands with new ones.

crontab Command

The crontab command is used to install, edit, remove, list, and perform other operations on crontab files. Each user has its own crontab files and can add commands to be executed to the files.

Here are common crontab command options:

- crontab -u //Set the **cron** service of a user. This option is required only when the **crontab** command is run by the **root** user.
- crontab -l //List details of the **cron** service of a user.
- crontab -r //Remove the **cron** service of a user.
- crontab -e //Edit the **cron** service of a user.

For example, to list cron service settings of the user **root**, run the following command:

```
crontab -u root -l
```

crontab Files

Enter the commands to be executed and time in crontab files. Each line in the files contains six fields. The first five fields are the time when the specified command is executed, and the last field is the command to be executed. Fields are separated by spaces or tabs. The format is as follows:

```
minute hour day-of-month month-of-year day-of-week commands
```

Each field is described as follows:

Table 6-1 Parameter description

Parameter	Description
minute	The minute of the hour at which commands will be executed. Value range: 0–59.
hour	The hour of the day at which periodic commands will be executed. Value range: 0–23.
day-of-month	The day of month at which periodic commands will be executed. Value range: 1–31.
month-of-year	The month of year at which periodic commands will be executed. Value range: 1–12.
day-of-week	The day of week at which periodic commands will be executed. Value range: 0–6.
commands	Periodic commands.

The fields cannot be left unspecified. In addition to numerical values, the following special symbols are allowed: Asterisk (*): a wildcard value. Forward slash (/): followed by a numeral N to indicate that commands will be executed at a regular interval of N. Hyphen (-): used with a range.Comma (,): used to separate discrete numbers. A complete path to the commands shall be provided.

For example, to allow the operating system to add sleepy to the /tmp/test.txt file every two hours from 18 pm to 22 pm, add the following line in a crontab file:

```
* 18-22/2 * * * echo "sleepy" >> /tmp/test.txt
```

Each time the cron service settings of a user are edited, the cron service generates in the /var/spool/cron directory a crontab file named after the user. The crontab file can be edited only using the crontab -e command. Alternatively, the user can create a file and run the crontab *filename* command to import its cron settings into the new file.

For example, to create a crontab file for the globus user, perform the following steps: The procedure is as follows:

1. Create a file using any text editor. Add the commands that need to be executed periodically and the command execution interval to the new file. In this example, the new file is **~/globus.cron**.
2. Run the following command to install the new file as the crontab file of the globus user: run the following command:

   ```
   crontab  globus. ~/globus.cron
   ```

After the new file is installed, you will find a file named globus in the **/var/spool/cron** directory. This file is the required crontab file.

NOTE

Do not restart the cron service after a crontab file is modified, because the cron service, once started, reads the crontab file every minute to check whether there are commands that need to be executed periodically. You do not need to restart the **cron** service after modifying the **crontab** file.

/etc/crontab File

The **cron** service reads all files in the **/var/spool/cron** directory and the **crontab** file in the **/etc/crontab** directory every minute. Therefore, you can use the **cron** service by configuring the **crontab** file. A crontab file contains user-specific commands, whereas the **/etc/crontab** file contains system-wide commands. Example /etc/crontab file

```
SHELL=/bin/sh
PATH=/usr/bin:/usr/sbin:/sbin:/bin:/usr/lib/news/bin
MAILTO=root //If an error occurs or data is output, the data is sent to the account by email.
HOME=/
#  run-parts
01 * * * * root run-parts /etc/cron.hourly //Run scripts in the /etc/cron.hourly directory once an hour.
02 4 * * *   root run-parts /etc/cron.daily    //Run scripts in the /etc/cron.daily directory once a day.
22 4 * * 0  root run-parts /etc/cron.weekly    //Run scripts in the /etc/cron.weekly directory once a week.
42 4 1  * *  root run-parts /etc/cron.monthly    //Run scripts in the /etc/cron.monthly directory once a month.
```

NOTE

If the **run-parts** parameter is deleted, a script name instead of a directory name is executed.

6.1.2 Suspending/Resuming a Process

A process can be suspended or resumed by job control, and the process will continue to work from the suspended point after being resumed. To suspend a foreground process, press Ctrl+Z. After you press Ctrl+Z, the cat command is suspended together with the foreground process you wish to suspend. You can use the jobs command instead to display a list of shell jobs, including their job names, IDs, and status.

To resume a process in foreground or background, run the fg or bg command, respectively. The process then starts from where it paused previously.

6.2 Viewing Processes

Linux is a multi-task system and needs to get process information during process management. To manage processes, you first need to know the number of processes and their statuses. Multiple commands are available to view processes.

who Command

The who command is used to display system user information. For example, before running the talk command to establish instant communication with another user, you need to run the who command to determine whether the target user is online. As another example, the system administrator can run the who command to learn what each login user is doing at the current time. The who command is widely seen in system administration since it is easy to use and can return a comprehensive set of accurate user information.

The following is an example output of the who command, where system users and their status are displayed: The use of the **who** command is as follows:

```
# who
admin    tty1        Jul 28 15:55
admin    pts/0       Aug  5 15:46 (192.168.0.110)
admin    pts/2       Jul 29 19:52 (192.168.0.110)
root    pts/3       Jul 30 12:07 (192.168.0.110)
root    pts/4       Jul 31 10:29 (192.168.0.144)
root    pts/5       Jul 31 14:52 (192.168.0.11)
root    pts/6       Aug  6 10:12 (192.168.0.234)
root    pts/8       Aug  6 11:34 (192.168.0.234)
```

ps Command

The **ps** command is the most basic and powerful command to view process information. The ps command is used to display process information, including which processes are running, terminated, resource-hungry, or stay as zombies.

A common scenario is using the ps command to monitor background processes, which do not interact with your screen, keyboard, and other I/O devices. Table 6-2 lists the common ps command options.

Table 6-2 Common ps command options

Option	Description
-e	Displays all processes.
-f	Full output format.
-h	Hides column headings in the listing of process information.
-l	Long output format.
-w	Wide output format.
-a	Lists all processes on a terminal, including those of other users.
-r	Lists only running processes.
-x	Lists all processes without controlling terminals.

For example, to list all processes on a terminal, run the following command:

```
# ps -a
  PID TTY          TIME CMD
12175 pts/6    00:00:00 bash
24526 pts/0    00:00:00 vsftpd
29478 pts/5    00:00:00 ps
32461 pts/0    1-01:58:33 sh
```

top Command

Both the top and the ps commands can display a list of currently running processes, but the top command allows you to update the displayed list of processes repeatedly with the press of a button. If the top command is executed in foreground, it exclusively occupies foreground until it is terminated. The top command provides real-time visibility into system processor status. You can sort the list of CPU tasks by CPU usage, memory usage, or task execution

time. Extensive customization of the display, such as choice of columns or sorting method, can be achieved using interactive commands or the customization file.

Figure 6-1 provides an example output of the top command.

Figure 6-1 Example command output

```
top - 19:04:08 up 9 days,  3:09,  8 users,  load average: 2.17, 2.08, 2.06
Tasks: 242 total,   8 running, 234 sleeping,   0 stopped,   0 zombie
Cpu(s):  8.3%us,  0.2%sy,  0.0%ni, 91.5%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:     19983M total,    19777M used,      206M free,      567M buffers
Swap:     2053M total,       10M used,     2043M free,    12326M cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
32757 root      20   0 4462m 3.0g 5440 S   100 15.3   1542:40 qemu-kvm
32461 root      20   0 11580 1380 1120 R   100  0.0   1563:47 sh
31437 root      20   0 4626m 2.4g 5436 R     4 12.1  14:36.89 qemu-kvm
29553 root      20   0 17256 1392  932 R     0  0.0   0:00.02 top
31438 root      20   0     0    0    0 S     0  0.0   0:12.80 vhost-31437
32758 root      20   0     0    0    0 S     0  0.0   0:25.21 vhost-32757
    1 root      20   0 10540  796  748 S     0  0.0   0:04.59 init
    2 root      20   0     0    0    0 S     0  0.0   0:00.00 kthreadd
    3 root      20   0     0    0    0 S     0  0.0   0:01.64 ksoftirqd/0
    6 root      RT   0     0    0    0 S     0  0.0   0:01.08 migration/0
    7 root      RT   0     0    0    0 S     0  0.0   0:01.66 watchdog/0
    8 root      RT   0     0    0    0 S     0  0.0   0:01.09 migration/1
    9 root      20   0     0    0    0 S     0  0.0   0:05.58 kworker/1:0
   10 root      20   0     0    0    0 S     0  0.0   0:01.31 ksoftirqd/1
   11 root      20   0     0    0    0 S     0  0.0   0:50.48 kworker/0:1
   12 root      RT   0     0    0    0 S     0  0.0   0:01.27 watchdog/1
   13 root      RT   0     0    0    0 S     0  0.0   0:01.64 migration/2
   14 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/2:0
   15 root      20   0     0    0    0 S     0  0.0   1:01.89 ksoftirqd/2
   16 root      RT   0     0    0    0 S     0  0.0   0:01.38 watchdog/2
   17 root      RT   0     0    0    0 R     0  0.0   0:01.12 migration/3
   18 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/3:0
   19 root      20   0     0    0    0 S     0  0.0   0:22.84 ksoftirqd/3
   20 root      RT   0     0    0    0 S     0  0.0   0:01.33 watchdog/3
   21 root      RT   0     0    0    0 S     0  0.0   0:01.56 migration/4
   22 root      20   0     0    0    0 S     0  0.0   0:00.00 kworker/4:0
   23 root      20   0     0    0    0 S     0  0.0   0:00.01 ksoftirqd/4
   24 root      RT   0     0    0    0 S     0  0.0   0:01.29 watchdog/4
```

kill Command

The **kill** command is used to terminate a process regardless of whether the process is running in foreground or background. It differs from the combo key **Ctrl+c**, which can terminate only foreground processes. The kill command is used to terminate a process regardless of whether the process is running in foreground or background. The reason for terminating a background process can be heavy use of CPU resources or deadlock.

The kill command sends a signal to terminate running processes. By default, the TERM signal is used. The TERM signal terminates all processes incapable of capturing the TERM signal. To terminate a process capable of capturing the TERM signal, use the KILL signal (signal ID: 9) instead.

Two types of syntax of the kill command:

```
kill [-s signal | -p] [-a] PID…
kill -l [signal]
```

The process ID is retrieved from the ps command. The **-s** option indicates the signal sent to specified program. The signal details can be viewed by running the **kill -l** command. The **-p** option indicates the specified process IDs.

For example, to terminate the process with ID 1409, run the following command:

```
# kill -9 1409
```

Example output of the kill command with the -l option

```
# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

7 Configuring the Network

7.1 Configuring an IP Address

7.1.1 Using the nmcli Command

NOTE

The network configuration configured by running the **nmcli** command takes effect immediately and will not be lost after the system restarts.

7.1.1.1 Introduction to nmcli

nmcli (NetworkManager Command Line Interface) is the command-line utility to configure networking through NetworkManager. The basic format of using **nmcli** is as follows:

```
nmcli [OPTIONS] OBJECT { COMMAND | help }
```

In the preceding command, **OBJECT** can be one of the following options: **general**, **networking**, **radio**, **connection**, and **device**. **OPTIONS** can be optional options, such as **-t**, **--terse** (for script processing), **-p**, **--pretty** (for human-readable output), **-h**, and **--help**. For more information, run the **nmcli help** command.

```
# nmcli help
```

Common commands are listed as follows:

- To display the general status of NetworkManager, run the following command:

  ```
  nmcli general status
  ```

- To display all connections, run the following command:

  ```
  nmcli connection show
  ```

- To display the current active connections only, add the **-a** or **--active** option as follows:

  ```
  nmcli connection show --active
  ```

- To display the device identified by NetworkManager and its connection status, run the following command:

```
nmcli device status
```

- To start or stop network interfaces, for example, run the nmcli commands:

```
nmcli connection up id enp3s0
nmcli device disconnect enp3s0
```

7.1.1.2 Setting Network Connections

Run the following command to display all the available network connections:

```
# nmcli con show


NAME    UUID                                  TYPE     DEVICE
enp4s0  5afce939-400e-42fd-91ee-55ff5b65deab  ethernet  enp4s0
enp3s0  c88d7b69-f529-35ca-81ab-aa729ac542fd  ethernet  enp3s0
virbr0  ba552da6-f014-49e3-91fa-ec9c388864fa  bridge    virbr0
```

NOTE

In the command output, **NAME** indicates the connection ID (name).

After a network connection is added, the corresponding configuration file is generated and associated with the corresponding device. To check for available devices, run the following command:

```
# nmcli dev status

DEVICE     TYPE     STATE     CONNECTION
enp3s0     ethernet  connected  enp3s0
enp4s0     ethernet  connected  enp4s0
virbr0     bridge    connected  virbr0
lo         loopback  unmanaged  --
virbr0-nic  tun      unmanaged  --
```

7.1.1.2.1 Configuring Dynamic IP Connections

Configuring IP Addresses

When DHCP is used to allocate a network, run the following command to add a network configuration file:

```
nmcli connection add type ethernet con-name connection-name ifname interface-name
```

For example, to create a dynamic connection configuration file named **net-test**, run the following command:

```
# nmcli connection add type ethernet con-name net-test ifname enp3s0
Connection 'net-test' (a771baa0-5064-4296-ac40-5dc8973967ab) successfully added.
```

The NetworkManager sets **connection.autoconnect** to **yes** and saves the setting to the **/etc/sysconfig/network-scripts/ifcfg-net-test** file. In the **/etc/sysconfig/network-scripts/ifcfg-net-test** file, **ONBOOT** is set to **yes**.

Activating a Connection and Checking Device Connection Status

Run the following command to activate a network connection:

```
# nmcli con up net-test
Connection successfully activated (D-Bus active
path:/org/freedesktop/NetworkManager/ActiveConnection/5)
```

Run the following command to check the connection status of devices:

```
# nmcli device status

DEVICE     TYPE     STATE     CONNECTION
enp4s0     ethernet  connected  enp4s0
enp3s0     ethernet  connected  net-test
virbr0     bridge    connected  virbr0
lo         loopback  unmanaged  --
virbr0-nic  tun      unmanaged  --
```

7.1.1.2.2 Configuring Static IP Connections

Configuring IP Addresses

To add a static IPv4 network connection, run the following command:

```
nmcli connection add type ethernet con-name connection-name ifname interface-name ip4
address gw4 address
```

NOTE

To add an IPv6 address and related gateway information, use the **ip6** and **gw6** options.

For example, to create a static connection configuration file named **net-static**, run the following command:

```
# nmcli con add type ethernet con-name net-static ifname enp3s0 ip4 192.168.0.10/24
gw4 192.168.0.254
```

You can also specify the IPv6 address and gateway for the device. The following is an example:

```
# nmcli con add type ethernet con-name test-lab ifname enp3s0 ip4 192.168.0.10/24 gw4
192.168.0.254 ip6 abbe::**** gw6 2001:***::*
Connection 'net-static' (63aa2036-8665-f54d-9a92-c3035bad03f7) successfully added.
```

The NetworkManager sets the internal parameter **ipv4.method** to **manual**, **connection.autoconnect** to **yes**, and writes the setting to the **/etc/sysconfig/network-scripts/ifcfg-my-office** file. In the file, **BOOTPROTO** is set to **none**, and **ONBOOT** is set to **yes**.

Run the following command to set IPv4 addresses of two DNS servers:

```
# nmcli con mod net-static ipv4.dns "*.*.*.* *.*.*.*"
```

Run the following command to set IPv6 addresses of two DNS servers:

```
# nmcli con mod net-static ipv6.dns "2001:4860:4860::**** 2001:4860:4860::****"
```

Activating a Connection and Checking Device Connection Status

Run the following command to activate a network connection:

```
# nmcli con up net-static ifname enp3s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/6)
```

Run the following command to check the connection status of devices:

```
# nmcli device status

DEVICE     TYPE     STATE     CONNECTION
enp4s0     ethernet  connected  enp4s0
enp3s0     ethernet  connected  net-static
virbr0     bridge   connected  virbr0
lo         loopback  unmanaged  --
virbr0-nic  tun      unmanaged  --
```

Run the following command to view the connection details (with the **-p** and **--pretty** options to add the title and segment to the output):

```
# nmcli -p con show net-static
===============================================================================
Connection profile details (net-static )
===============================================================================
connection.id:                       net-static
connection.uuid:                     b9f18801-6084-4aee-af28-c8f0598ff5e1
connection.stable-id:                 --
connection.type:                     802-3-ethernet
connection.interface-name:            enp3s0
connection.autoconnect:               yes
connection.autoconnect-priority:       0
connection.autoconnect-retries:        -1 (default)
connection.multi-connect:              0 (default)
connection.auth-retries:               -1
connection.timestamp:                 1578988781
connection.read-only:                 no
connection.permissions:                --
connection.zone:                     --
connection.master:                    --
connection.slave-type:                 --
connection.autoconnect-slaves:         -1 (default)
connection.secondaries:                --
connection.gateway-ping-timeout:       0
connection.metered:                   unknown
connection.lldp:                     default
connection.mdns:                     -1 (default)
connection.llmnr:                    -1 (default)
```

7.1.1.2.3 Adding a Wi-Fi Connection

Run the following command to check for available Wi-Fi access points:

```
# nmcli dev wifi list
```

Run the following command to generate a static IP address configuration that allows Wi-Fi connections automatically allocated by the DNS:

```
# nmcli con add con-name Wifi ifname wlan0 type wifi ssid MyWifi ip4 192.168.100.101/24
gw4 192.168.100.1
```

Run the following command to set a WPA2 password, for example, **answer**:

```
# nmcli con modify Wifi wifi-sec.key-mgmt wpa-psk
# nmcli con modify Wifi wifi-sec.psk answer
```

Run the following command to change the Wi-Fi status:

```
# nmcli radio wifi [ on | off ]
```

7.1.1.2.4 Modifying Attributes

Run the following command to check a specific attribute, for example, mtu:

```
# nmcli connection show id 'Wifi ' | grep mtu
802-11-wireless.mtu: auto
```

Run the following command to modify the attribute:

```
# nmcli connection modify id 'Wifi ' 802-11-wireless.mtu 1350
```

Run the following command to confirm the modification:

```
# nmcli connection show id 'Wifi ' | grep mtu
802-11-wireless.mtu: 1350
```

7.1.1.3 Configuring a Static Route

- Run the nmcli command to configure a static route for a network connection:

  ```
  # nmcli connection modify enp3s0 +ipv4.routes "192.168.122.0/24 10.10.10.1"
  ```

- Run the following command to configure the static route using the editor:

  ```
  # nmcli con edit type ethernet con-name enp3s0
  ===| nmcli interactive connection editor |===
  Adding a new '802-3-ethernet' connection
  Type 'help' or '?' for available commands.
  Type 'describe [<setting>.<prop>]' for detailed property description.
  You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x,
  ipv4, ipv6, dcb
  nmcli> set ipv4.routes 192.168.122.0/24 10.10.10.1
  nmcli>
  nmcli> save persistent
  Saving the connection with 'autoconnect=yes'. That might result in an immediate
  activation of the connection.
  Do you still want to save? [yes] yes
  Connection 'enp3s0' (1464ddb4-102a-4e79-874a-0a42e15cc3c0) successfully saved.
  nmcli> quit
  ```

7.1.2 Using the ip Command

> NOTE
>
> The network configuration configured using the **ip** command takes effect immediately, but the configuration will be lost after the system restarts.

7.1.2.1 Configuring IP Addresses

Run the **ip** command to configure an IP address for the interface. The command format is as follows, where *interface-name* indicates the NIC name.

```
ip addr [ add | del ] address dev interface-name
```

Configuring a Static IP Address

Run the following command as the user **root** to configure an IP address:

```
# ip address add 192.168.0.10/24 dev enp3s0
```

Run the following command to view the configuration result:

```
# ip addr show dev enp3s0
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq codel state UP group
default qlen 1000
   link/ether 52:54:00:aa:ad:4a brd ff:ff:ff:ff:ff:ff
   inet 192.168.202.248/16 brd 192.168.255.255 scope global dynamic noprefixroute
enp3s0
      valid lft 9547sec preferred lft 9547sec
   inet 192.168.0.10/24 scope global enp3s0
      valid lft forever preferred lft forever
   inet6 fe80::32e8:cc22:9db2:f4d4/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

Configuring Multiple IP Addresses

The **ip** command can be used to assign multiple IP addresses to an interface. You can run the **ip** command multiple times to assign IP addresses to an interface. The following is an example:

```
# ip address add 192.168.2.223/24 dev enp4s0
# ip address add 192.168.4.223/24 dev enp4s0
# ip addr

3: enp4s0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq codel state UP group
default qlen 1000
   link/ether 52:54:00:aa:da:e2 brd ff:ff:ff:ff:ff:ff
   inet 192.168.203.12/16 brd 192.168.255.255 scope global dynamic noprefixroute enp4s0
      valid lft 8389sec preferred lft 8389sec
   inet 192.168.2.223/24 scope global enp4s0
      valid_lft forever preferred_lft forever
   inet 192.168.4.223/24 scope global enp4s0
      valid_lft forever preferred_lft forever
   inet6 fe80::1eef:5e24:4b67:f07f/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

7.1.2.2 Configuring a Static Route

Configuring a Static Route

To add a static route to the routing table, run the **ip route add** command. To delete a route, run the **ip route del** command. The following shows the common format of the **ip route** command:

```
ip route [ add | del | change | append | replace ] destination-address
```

To display the current IP routing table, run the **ip route** command. The following is an example:

```
# ip route

default via 192.168.0.1 dev enp3s0 proto dhcp metric 100
default via 192.168.0.1 dev enp4s0 proto dhcp metric 101
192.168.0.0/16 dev enp3s0 proto kernel scope link src 192.168.202.248 metric 100
192.168.0.0/16 dev enp4s0 proto kernel scope link src 192.168.203.12 metric 101
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1 linkdown
```

To add a static route to the host address, run the following command as the user **root**:

```
ip route add 192.168.2.1 via 10.0.0.1 [dev interface-name]
```

In the preceding command, **192.168.2.1** is the IP address in the dot-decimal notation, **10.0.0.1** is the next hop, and *interface-name* is the exit interface for entering the next hop.

To add a static route to the network, that is, an IP address that represents an IP address range, run the following command as the user **root**:

```
ip route add 192.168.2.0/24 via 10.0.0.1 [dev interface-name]
```

In the preceding command, **192.168.2.1** is the IP address of the target network, *10.0.0.1* is the network prefix, and *interface-name* is the NIC name.

7.1.3 Configuring the Network Through the ifcfg File

NOTE

The network configured in the **ifcfg** file does not take effect immediately. You need to run the **systemctl reload NetworkManager** command to restart the network service for the configuration to take effect.

Configuring a Static Network

The following uses the **enp4s0** network interface as an example to describe how to configure a static network by modifying the **ifcfg** file. The **ifcfg-enp4s0** file is generated in the **/etc/sysconfig/network-scripts/** directory. Modify the following parameters in the file:

```
TYPE=Ethernet
PROXY METHOD=none
BROWSER ONLY=no
BOOTPROTO=none
IPADDR=192.168.0.10
PREFIX=24
DEFROUTE=yes
IPV4 FAILURE FATAL=no
IPV6INIT=yes
IPV6 AUTOCONF=yes
IPV6 DEFROUTE=yes
IPV6 FAILURE FATAL=no
IPV6 ADDR GEN MODE=stable-privacy
NAME=enp4s0static
UUID=08c3a30e-c5e2-4d7b-831f-26c3cdc29293
DEVICE=enp4s0
ONBOOT=yes
```

Configuring a Dynamic Network

The following uses the **em1** network interface as an example to describe how to configure a dynamic network by modifying the **ifcfg** file. The **ifcfg-em1** file is generated in the **/etc/sysconfig/network-scripts/** directory. Modify the following parameters in the file:

```
DEVICE=em1
BOOTPROTO=dhcp
ONBOOT=yes
```

To configure an interface to send different host names to the DHCP server, add the following content to the **ifcfg** file:

```
DHCP_HOSTNAME=hostname
```

To configure an interface to ignore the routes sent by the DHCP server to prevent network services from updating the /etc/resolv.conf file using the DNS server received from the DHCP server, add the following content to the **ifcfg** file:

```
PEERDNS=no
```

To configure an interface to use a specific DNS server, set the **PEERDNS** parameter to **no** and add the following content to the **ifcfg** file:

```
DNS1=ip-address
DNS2=ip-address
```

ip-address is the IP address of the DNS server. This allows the network service to update the **/etc/resolv.conf** file using the specified DNS server.

Default Gateway Configuration

When determining the default gateway, parse the **/etc/sysconfig/network** file and then the **ifcfg** file, and uses the value of **GATEWAY** that is read last as the default route in the routing table.

In a dynamic network environment, when the NetworkManager is used to manage hosts, you are advised to set the default gateway to DHCP assignment.

7.2 Configuring a Host Name

7.2.1 Introduction

There are three types of host names: **static**, **transient**, and **pretty**.

- **static**: Static host name, which can be set by users and saved in the **/etc/hostname** file.
- **transient**: Dynamic host name, which is maintained by the kernel. The initial value is a static host name. The default value is **localhost**. The value can be changed when the DHCP or mDNS server is running.
- **pretty**: Flexible host name, which can be set in any form (including special characters/blanks). Static and transient host names are subject to the general domain name restrictions.

> NOTE
>
> Static and transient host names can contain only letters (a–z and A–Z), digits (0–9), hyphens (-), underlines (_), and periods (.). The host names cannot start or end with a period (.) or contain two consecutive periods (.). The host name can contain a maximum of 64 characters.

7.2.2 Configuring a Host Name by Running the hostnamectl Command

Viewing All Host Names

Run the following command to view the current host name:

```
# hostnamectl status
```

> NOTE
>
> If no option is specified in the command, the **status** option is used by default.

Setting All Host Names

Run the following command as the **root** user to set all host names:

```
# hostnamectl set-hostname name
```

Setting a Specific Host Name

Run the following command as the **root** user to set a specific host name:

```
# hostnamectl set-hostname name [option...]
```

The option may be one or more of **--pretty**, **--static**, and **--transient**.

If **--static** or **--transient** is used together with **--pretty**, the host names of the **static** or **transient** type will be simplified to the host names of the **pretty** type with spaces replaced with hyphens (-) and special characters deleted.

When setting a host name of the **pretty** type, use quotation marks if the host name contains spaces or single quotation marks. An example is as follows:

```
# hostnamectl set-hostname "Stephen's notebook" --pretty
```

Clearing a Specific Host Name

To clear a specific host name and restore it to the default format, run the following command as the **root** user:

```
# hostnamectl set-hostname "" [option...]
```

In the preceding command, "" is a blank character string, and the *option* may be one or more of **--pretty**, **--static**, and **--transient**.

Remotely Changing a Host Name

To change the host name in a remote system, run the **hostnamectl** command with the **-H** or **--host** option.

```
# hostnamectl set-hostname -H [username]@hostname new_hostname
```

In the preceding command, *hostname* indicates the name of the remote host to be configured, *username* indicates the user-defined name, and *new_hostname* indicates the new host name. **hostnamectl** is used to connect to the remote system through SSH.

7.2.3 Configuring a Host Name by Running the nmcli Command

To query a static host name, run the following command:

```
# nmcli general hostname
```

To name a static host as **host-server**, run the following command as user root:

```
# nmcli general hostname host-server
```

To enable the system to detect the change of the static host name, run the following command as the **root** user to restart the hostnamed service:

```
# systemctl restart systemd-hostnamed
```

7.3 Configuring Network Bonding

7.3.1 Running the nmcli Command

- To create a bond named **mybond0**, run the following command:

  ```
  # nmcli con add type bond con-name mybond0 ifname mybond0 mode active-backup
  ```

- To add a slave interface, run the following command:

  ```
  # nmcli con add type bond-slave ifname enp3s0 master mybond0
  ```

  To add another slave interface, repeat the preceding command with the new interface name:

  ```
  # nmcli con add type bond-slave ifname enp4s0 master mybond0
  Connection 'bond-slave-enp4s0' (05e56afc-b953-41a9-b3f9-0791eb49f7d3)
  successfully added.
  ```

- To enable a bond, run the following command to enable the slave interface first:

  ```
  # nmcli con up bond-slave-enp3s0
  Connection successfully activated (D-Bus active path:
  /org/freedesktop/NetworkManager/ActiveConnection/14)
  # nmcli con up bond-slave-enp4s0
  Connection successfully activated (D-Bus active path:
  /org/freedesktop/NetworkManager/ActiveConnection/15)
  ```

  Then, run the following command to enable the bond:

  ```
  # nmcli con up bond-mybond0
  Connection successfully activated (D-Bus active path:
  /org/freedesktop/NetworkManager/ActiveConnection/16)
  ```

7.3.2 Configuring Network Bonding by Using a Command Line

7.3.2.1 Checking Whether the Bonding Kernel Module Is Installed

By default, the bonding kernel module is loaded. To load this module, run the following command as the **root** user:

```
# modprobe --first-time bonding
```

Run the following command to display the information about the module:

```
# modinfo bonding
```

For more commands, run the modprobe --help command.

7.3.2.2 Creating a Channel Bonding Interface

To create a channel bonding interface, you can create a file named **ifcfg-bondN** in the **/etc/sysconfig/network-scripts/** directory (replacing N with the actual interface number, for example, 0).

Write the corresponding content to the configuration file according to the type of the interface to be bonded, for example, network interface. An example of the interface configuration file is as follows:

```
DEVICE=bond0
NAME=bond0
TYPE=Bond
BONDING_MASTER=yes
IPADDR=192.168.1.1
PREFIX=24
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="bonding parameters separated by spaces"
```

7.3.2.3 Creating a Slave Interface

After creating a channel bonding interface, you must add the **MASTER** and **SLAVE** instructions to the configuration file of the slave interface.

For example, to bind the two network interfaces enp3s0 and enp4s0 in channel mode, the configuration files are as follows:

```
TYPE=Ethernet
NAME=bond-slave-enp3s0
UUID=3b7601d1-b373-4fdf-a996-9d267d1cac40
DEVICE=enp3s0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
TYPE=Ethernet
NAME=bond-slave-enp4s0
UUID=00f0482c-824f-478f-9479-abf947f01c4a
DEVICE=enp4s0
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

7.3.2.4 Activating Channel Bonding

To activate channel bonding, you need to enable all the slave interfaces. Run the following command as the **root** user:

```
# ifup enp3s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/7)
```

```
# ifup enp4s0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/8)
```

NOTE

If an interface is in **up** state, run the **ifdown** *enp3s0* command to change the state to **down**. In the command, *enp3s0* indicates the actual NIC name.

After that, enable all the slave interfaces to enable the bonding (do not set them to **Down**).

To enable the NetworkManager to detect the modifications made by the system, run the following command as user **root** user after each modification:

```
# nmcli con load /etc/sysconfig/network-scripts/ifcfg-device
```

Run the following command to check the status of the bonded interface:

```
# ip link show

1: lo: <LOOPBACK,UP,LOWER UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq codel state UP mode
DEFAULT group default qlen 1000
   link/ether 52:54:00:aa:ad:4a brd ff:ff:ff:ff:ff:ff
3: enp4s0: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq codel state UP mode
DEFAULT group default qlen 1000
   link/ether 52:54:00:aa:da:e2 brd ff:ff:ff:ff:ff:ff
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode
DEFAULT group default qlen 1000
   link/ether 86:a1:10:fb:ef:07 brd ff:ff:ff:ff:ff:ff
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq codel master virbr0 state DOWN
mode DEFAULT group default qlen 1000
   link/ether 52:54:00:29:35:4c brd ff:ff:ff:ff:ff:ff
```

7.3.2.5 Creating Multiple Bondings

The system creates a channel bonding interface for each bonding, including the **BONDING_OPTS** instruction. This configuration method allows multiple bonded devices to use different configurations. Perform the following operations to create multiple channel bonding interfaces:

- Create multiple **ifcfg-bondN** files that contain the **BONDING_OPTS** instruction so that network scripts can create bonding interfaces as required.
- Create or edit the existing interface configuration file to be bonded, and add the **SLAVE** instruction.
- Use the MASTER instruction to assign the interface to be bonded, that is, the slave interface, to the channel bonding interface.

The following is an example of the configuration file of a channel bonding interface:

```
DEVICE=bondN
NAME=bondN
TYPE=Bond
BONDING MASTER=yes
IPADDR=192.168.1.1
PREFIX=24
ONBOOT=yes
```

```
BOOTPROTO=none
BONDING_OPTS="bonding parameters separated by spaces"
```

In this example, replace N with the number of the bonded interface. For example, to create two interfaces, you need to create two configuration files **ifcfg-bond0** and **ifcfg-bond1** with correct IP addresses.

7.4 IPv6 Differences (vs IPv4)

7.4.1 Restrictions

- chrony supports global addresses but not link-local addresses.
- Firefox supports the access to the global address through HTTP or HTTPS, but does not support the access to the link-local address.

7.4.2 Configuration Description

7.4.2.1 Setting the MTU of an Interface Device

Overview

In an IPv6 scenario, the minimum MTU value of the entire routing path is used as the PMTU value of the current link. The source end determines whether to fragment packets based on the PMTU value. Other devices on the entire path do not need to fragment packets. This reduces the load of intermediate routing devices. The minimum value of IPv6 PMTU is 1280.

Setting the MTU of the Interface Device

If the MTU of an interface configured with an IPv6 address is set to a value smaller than **1280** (the minimum value of the IPv6 PMTU), the IPv6 address of the interface will be deleted and cannot be added again. Therefore, in IPv6 scenarios, the MTU of the interface device must be greater than or equal to 1280. The details are as follows:

```
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
   link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
   inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic enp3s0
      valid lft 38663sec preferred lft 38663sec
   inet6 2001:222::2/64 scope global
      valid lft forever preferred lft forever
# ip link set dev enp3s0 mtu 1200
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1200 qdisc pfifo_fast state UP group
default qlen 1000
   link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
   inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic enp3s0
      valid lft 38642sec preferred lft 38642sec
# ip addr add 2001:222::2/64 dev enp3s0
RTNETLINK answers: No buffer space available
# ip link set dev enp3s0 mtu 1500
# ip addr show enp3s0
```

```
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
   link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
   inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic enp3s0
      valid_lft 38538sec preferred_lft 38538sec
# ip addr add 2001:222::2/64 dev enp3s0
# ip addr show enp3s0
3: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
   link/ether 52:54:00:62:xx:xx brd ff:ff:ff:ff:xx:xx
   inet 10.41.125.236/16 brd 10.41.255.255 scope global noprefixroute dynamic enp3s0
      valid_lft 38531sec preferred_lft 38531sec
   inet6 2001:222::2/64 scope global
      valid_lft forever preferred_lft forever
```

7.4.2.2 Stateful IPv6 Address Autoconfiguration

Overview

Both IPv6 and IPv4 addresses can be obtained through DHCP. There are configuration methods for IPv6 address: stateless autoconfiguration and stateful autoconfiguration.

- Stateless autoconfiguration

  The DHCP server is not required for management. The device obtains the network prefix according to the router advertisement (RA), or the prefix of a link-local address is fixed to fe80::. The interface ID is automatically obtained based on the value of IPV6_ADDR_GEN_MODE in the ifcfg file.

  a. If the value of IPv6_ADDR_GEN_MODE is stable-privacy, the device determines a random interface ID based on the device and network environment.

  b. If the value of IPv6_ADDR_GEN_MODE is EUI64, the device determines the interface ID based on the device MAC address.

- Stateful autoconfiguration: The DHCP server manages and leases IPv6 addresses from the DHCPv6 server base on the DHCPv6 protocol.

  In stateful autoconfiguration, the DHCPv6 server can classify clients based on the vendor class configured on the clients and assign IPv6 addresses in different address segments to different types of clients. In IPv4 scenarios, the client can use the -V option of the dhclient command to set the vendor-class-identifier field. The DHCP server classifies clients based on the vendor-class-identifier field in the configuration file. In IPv6 scenarios, if the same method is used to classify clients, the classification does not take effect.

  ```
  dhclient -6 <interface> -V <vendor-class-identifier string> <interface>
  ```

  This is because DHCPv6 differs greatly from DHCP. The vendor-class-option in DHCPv6 replaces the vendor-class-identifier in DHCP. However, the -V option of dhclient cannot be set to vendor-class-option.

Setting the vendor class for dhclient in Stateful IPv6 Address Autoconfiguration

- On the client, add the setting of vendor class by using the configuration file.

  Client configuration file (/etc/dhcp/dhclient6.conf): The file location can be customized. You need to specify the configuration file using the dhclient -cf option.

  ```
  option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};
  interface "enp3s0" {
  ```

```
      send dhcp6.vendor-class <Enterprise-ID number> <vendor class string length>
<vendor class string>;
}
```

NOTE

- <Enterprise-ID number>: a 32-digit integer, indicating the enterprise ID. The enterprise is registered through the IANA.
- <vendor class string length>: a 16-digit integer, indicating the length of the vendor class string.
- <vendor class string>: character string of the vendor class to be set, for example, HWHW.

On the client:

```
dhclient -6 <interface> -cf /etc/dhcp/dhclient6.conf
```

- The DHCPv6 server configuration file (/etc/dhcp/dhcpd6.conf) needs to be specified by the dhcpd -cf option.

```
option dhcp6.vendor-class code 16 = {integer 32, integer 16, string};
subnet6 fc00:4:12:ffff::/64 {
      class "hw" {
            match if substring ( option dhcp6.vendor-class, 6, 10 ) = "HWHW";
      }
      pool6 {
            allow members of "hw";
            range6 fc00:4:12:ffff::ff10 fc00:4:12:ffff::ff20;
      }
      pool6 {
            allow unknown clients;
            range6 fc00:4:12:ffff::100 fc00:4:12:ffff::120;
      }
}
```

NOTE

In substring (option dhcp6.vendor-class, 6, 10), the start position of the substring is 6, because the substring contains four bytes of <Enterprise-ID number> and two bytes of <string length>. The end position of the substring is 6+<vendor class string length>. In this example, the vendor class string is HWHW, and the length of the string is 4. Therefore, the end position of the substring is 6 + 4 = 10. You can specify <vendor class string> and <vendor class string length> as required.

On the server:

```
dhcpd -6 -cf /etc/dhcp/dhcpd6.conf <interface>
```

7.4.2.3 Kernel Supporting Socket-Related System Calls

Overview

The length of an IPv6 address is extended to 128 bits, indicating that there are sufficient IPv6 addresses for allocation. Compared with the IPv4 header, the IPv6 header is simplified, and the IPv6 automatic configuration function is enhanced. IPv6 addresses are classified into unicast addresses, multicast addresses, and anycast addresses. Common unicast addresses include link-local addresses, unique local addresses, and global addresses. As there are sufficient global IPv6 addresses, unique local addresses are not used. (formerly known as site-local addresses, which were discarded in 2004.) Currently, the mainstream unicast addresses are link-local address and global address. The current kernel supports socket system invoking. The link-local address and global address using unicast addresses are different.

Differences Between the link-local Address and global Address During Socket Invoking

RFC 2553: Basic Socket Interface Extensions for IPv6 defines the sockaddr_in6 data structure as follows:

```
struct sockaddr in6 {
   uint8 t        sin6 len;      /* length of this struct */
   sa family t     sin6 family;    /* AF INET6 */
   in port t      sin6 port;     /* transport layer port # */
   uint32 t       sin6 flowinfo;  /* IPv6 flow information */
   struct in6 addr sin6 addr;     /* IPv6 address */
   uint32 t       sin6 scope id;  /* set of interfaces for a scope */
};
```

NOTE

sin6_scope_id: a 32-bit integer. For the link-local address, it identifies the index of the specified interface. For the link-range sin6_addr, it identifies the index of the specified interface. For the site-range sin6_addr, it is used as the site identifier (the site-local address has been discarded).

When the link-local address is used for socket communication, the interface index corresponding to the address needs to be specified when the destination address is constructed. Generally, you can use the if_nametoindex function to convert an interface name into an interface index number. Details are as follows:

```
int port = 1234;
int sk fd;
int iff index = 0;
char iff name[100] = "enp3s0";
char * ll addr[100] = "fe80::123:456:789";
struct sockaddr in6 server addr;

memset(&server addr,0,sizeof(structsockaddr in6));
iff_index=if_nametoindex(iff_name);

server addr.sin6 family=AF INET6;
server addr.sin6 port=htons(port);
server_addr.sin6_scope_id=iff_index;
inet pton(AF INET6, ll addr, &(server addr.sin6 addr));

sk fd=socket(AF INET6, SOCK STREAM, IPPROTO TCP);
connect(sk_fd, (struct sockaddr *)&server_addr, sizeof(struct sockaddr_in6));
```

7.4.2.4 Persistency Configuration of the IPv4 dhclient Daemon Process

Overview

When the NetworkManager service is used to manage network services, if the ifcfg-<interface-name> configuration file of an interface is configured to obtain an IP address in DHCP mode, the NetworkManager service starts the dhclient daemon process to obtain an IP address from the DHCP server.

The dhclient provides the -1 option to determine whether the dhclient process persistently attempts to request an IP address or exits after the request times out before receiving a response from the DHCP server. For the IPv4 dhclient daemon process, you can set

PERSISTENT_DHCLIENT in the ifcfg-<interface-name> configuration file to determine whether to set the persistence of the IPv4 dhclient process.

Restrictions

1. If the ongoing dhclient process is killed, the network service cannot automatically start it. Therefore, you need to ensure the reliability.
2. If PERSISTENT_DHCLIENT is configured, ensure that the corresponding DHCP server exists. If no DHCP server is available when the network service is started and the dhclient process continuously attempts to send request packets but does not receive any response, the network service is suspended until the network service times out. The network service starts the IPv4 dhclient processes of multiple NICs in serial mode. If persistency is configured for a NIC but the DHCP server is not ready, the network service will be suspended when obtaining an IPv4 address for the NIC. As a result, the NIC cannot obtain an IPv4 or IPv6 address.

The preceding restrictions apply to special scenarios. You need to ensure reliability.

Configuration Differences Between IPv4 DHCP and IPv6 DHCPv6

You can configure the ifcfg-<interface-name> parameter on an interface to enable IPv4 and IPv6 to dynamically obtain IP addresses using DHCP or DHCPv6. The configuration is as follows:

```
BOOTPROTO=none|bootp|dhcp
DHCPV6C=yes|no
PERSISTENT_DHCLIENT=yes|no|1|0
```

- BOOTPROTO: **none** indicates that an IPv4 address is statically configured. bootp|dhcp enables DHCP dhclient to dynamically obtain an IPv4 address.
- DHCPV6C: **no** indicates that an IPv6 address is statically configured, and **yes** indicates that the DHCPv6 dhclient is enabled to dynamically obtain the IPv6 address.
- PERSISTENT_DHCLIENT: **no|0** indicates that the IPv4 dhclient process is configured as nonpersistent. If the dhclient sends a request packet to the DHCP server but does not receive any response, the dhclient exits after a period of time and the exit value is 2. **yes|1** indicates that the IPv4 dhclient process is configured to be persistent. The dhclient process repeatedly sends request packets to the DHCP server. **If PERSISTENT_DHCLIENT is not configured, dhclient of IPv4 is set to yes|1 by default.**

NOTE

The PERSISTENT_DHCLIENT configuration takes effect only for IPv4 and does not take effect for IPv6-related dhclient -6 processes. By default, the persistence configuration is not performed for IPv6.

7.4.2.5 Differences Between IPv4 and IPv6 Configuration Using the iproute Command

Overview

IPv4 and IPv6 are two different protocol standards. Therefore, the iproute commands are different in usage. This section describes the differences between IPv4 and IPv6 commands in the iproute package.

Lifecycle of an IPv6 Address

IPv6 status	Description
tentative	Temporary state: The newly added address is still in the DAD process.
preferred	Preferred state: The DAD process is complete, but no NA packet is received, indicating that the address does not conflict.
deprecated	Deprecated state: An address has a validity period (valid_lft or preferred_lft). After preferred_lft expires, the address changes to the deprecated state. The address in this state cannot be used to create a new connection, but the original connection can still be used.
invalid	Invalid state: If the lease renewal fails after the preferred_lft time expires, the address status is set to invalid after the valid_lft time expires, indicating that the address cannot be used again.

Remarks:

- preferred_lft: preferred lifetime. The preferred_lft address has not expired and can be used for normal communication. If there are multiple preferred addresses, the address is selected based on the kernel mechanism.
- valid_lft: valid lifetime. The address cannot be used for creating new connections within the period of [preferred_lft, valid_lft]. The existing connections are still valid.

ip link Command

The commands are as follows:

```
ip link set IFNAME mtu MTU
```

The minimum PMTU of IPv6 is 1280. If the MTU is set to a value smaller than 1280, IPv6 addresses will be lost. Other devices cannot ping the IPv6 address.

ip addr Command

1. The commands are as follows:

   ```
   ip [-6] addr add IFADDR dev IFNAME
   ```

   You can choose to add the -6 option or not to add the IPv6 address. The ip addr command determines whether the address is an IPv4 address or an IPv6 address based on the address type.

   If the -6 option is specified but IFADDR is an IPv4 address, an error message is returned.

2. The commands are as follows:

   ```
   ip [-6] addr add IFADDR  dev IFNAME [home|nodad]
   ```

   [home|nodad] is valid only for IPv6 addresses.

   - home: specifies the home address defined in RFC 6275. (This address is obtained by the mobile node from the home link, and is a permanent address of the mobile node. If the mobile node remains in the same home link, communication between various entities is performed normally.)

- nodad: indicates that DAD is not performed when this IPv6 address is added. (RFC 4862) If multiple interfaces on a device are configured with the same IPv6 address through nodad, the IPv6 address is used in the interface sequence. An IPv6 address with both nodad and non-nodad cannot be added the same interface because the two IP addresses are the same. Otherwise, the message "RTNETLINK answers: File exists" is displayed.

3. The commands are as follows:

```
ip [-6] addr del IFADDR dev IFNAME
```

You can choose to add the -6 option or not to delete an IPv6 address. The ip addr del command determines whether an IPv4 address or an IPv6 address is used based on the address type.

4. The commands are as follows:

```
ip [-6] addr show dev IFNAME
[tentative|-tentative|deprecated|-deprecated|dadfailed|-dadfailed|temporary]
```

- If the -6 option is not specified, both IPv4 and IPv6 addresses are displayed. If the -6 option is specified, only IPv6 addresses are displayed.
- [tentative|-tentative|deprecated|-deprecated|dadfailed|-dadfailed|temporary]. These options are only for IPv6. You can filter and view addresses based on the IPv6 address status.
  i. tentative: (only for IPv6) lists only the addresses that have not passed duplicate address detection (DAD).
  ii. -tentative: (only for IPv6) lists only the addresses that are not in the DAD process.
  iii. deprecated: (only for IPv6) lists only the deprecated addresses.
  iv. -deprecated: (only for IPv6) lists only the addresses that are not deprecated.
  v. dadfailed: (only for IPv6) lists only the addresses that fail the DAD.
  vi. -dadfailed: (only for IPv6) lists only the addresses that do not encounter DAD failures.
  vii. temporary: (only for IPv6) lists only the temporary addresses.

ip route Command

1. The commands are as follows:

```
ip [-6] route add ROUTE [mtu lock MTU]
```

- -6 option: You can add the -6 option or not when adding an IPv6 route. The ip route command determines whether an IPv4 or IPv6 address is used based on the address type.
- mtu lock MTU: specifies the MTU of the locked route. If the MTU is not locked, the MTU value may be changed by the kernel during the PMTUD process. If the MTU is locked, PMTUD is not attempted. All IPv4 packets are not set with the DF bit and IPv6 packets are segmented based on the MTU.

2. The commands are as follows:

```
ip [-6] route del ROUTE
```

You can choose whether to add the -6 option when deleting an IPv6 route. The ip route command determines whether an IPv4 address or an IPv6 address is used based on the address type.

ip rule command

1. The commands are as follows:

   ```
   ip [-6] rule list
   ```

   -6 option: If the -6 option is set, IPv6 policy-based routes are printed. If the -6 option is not set, IPv4 policy-based routes are printed. Therefore, you need to configure the -6 option according to the specific protocol type.

2. The commands are as follows:

   ```
   ip [-6] rule [add|del] [from|to] ADDR table TABLE pref PREF
   ```

   -6 option: IPv6-related policy routing entries need to be configured with the -6 option. Otherwise, the error message "Error: Invalid source address." is displayed. Accordingly, the -6 option cannot be set for IPv4-related policy routing entries. Otherwise, the error message "Error: Invalid source address." is displayed.

7.4.2.6 Configuration Differences of the NetworkManager Service

Overview

The NetworkManager service uses the ifup/ifdown logical interface definition to perform advanced network settings. Most of the parameters are set in the /etc/sysconfig/network and /etc/sysconfig/network-scripts/ifcfg-<interface-name> configuration files. The former is a global setting, and the latter is a setting of a specified NIC. When the two settings conflict, the latter takes effect.

Configuration Differences

The configuration differences in /etc/sysconfig/network are as follows:

IPv4	IPv6	Description
NA	IPV6FORWARDING=yes|no	IPv6 forwarding. By default, IPv6 packets are not forwarded.
NA	IPV6_AUTOCONF=yes|no	If IPv6 forwarding is enabled, the value is **no**. Otherwise, the value is **yes**.
NA	IPV6_ROUTER=yes|no	If IPv6 forwarding is enabled, the value is **yes**. Otherwise, the value is **no**.
NA	IPV6_AUTOTUNNEL=yes|no	Indicates the automatic tunnel mode. The default value is **no**.
GATEWAY	IPV6_DEFAULTGW=<IPv6 address[%interface]> (optional)	Indicates the default gateway in IPv6.
NA	IPV6_DEFAULTDEV=<interface> (optional)	Specifies the default forwarding NIC.
NA	IPV6_RADVD_PIDFILE=<pid-file> (optional)	The default path of ipv6_radvd_pid is /var/run/radvd/radvd.pid.

IPv4	IPv6	Description
NA	IPV6_RADVD_TRIGGER_ACTION=startstop|reload|restart|SIGHUP (optional)	Default radvd trigger action.

The differences in /etc/sysconfig/network-scripts/ifcfg-<interface-name> are as follows:

IPv4	IPv6	Description
IPADDRn	IPV6ADDR=<IPv6 address>[/<prefix length>]	indicates the IP address.
PREFIXn	NA	The network prefix, network alias, and PPP are invalid. The priority is higher than that of NETMASK.
NETMASKn	NA	Indicates the subnet mask. It is used only for the alias and PPP.
GATEWAY	IPV6_DEFAULTGW=<IPv6 address[%interface]> (optional)	Default gateway
MTU	IPV6_MTU=<MTU of link> (optional)	Default MTU
IPV4_FAILURE_FATAL=yes|no	IPV6_FAILURE_FATAL	The default value is **no**. If this parameter is set to **yes**, ifup-eth exits when dhclient fails.
NA	IPV6_PRIVACY=rfc3041	Disabled by default.
NA	IPV6INIT=yes|no	IPv6 is enabled by default.
NA	IPV6FORWARDING=yes|no	This function is disabled by default and has been discarded.

7.4.3 FAQ

7.4.3.1 The iscsi-initiator-utils Does Not Support the fe80 IPv6 Address.

Symptom

When a client uses an IPv6 address to log in to the iSCSI server, run the iscsiadm -m node -p ipv6address -l command. If the global address is used, replace ipv6address in the command example with the global address. However, the link-local address (IPv6 address starting with fe80) cannot be used because the current mechanism of iscsi-initiator-utils does not support the link-local address to log in to the iSCSI server.

Possible Cause

If you log in to the system using the iscsiadm -m node -p fe80::xxxx -l format, a login timeout error is returned. This is because you must specify an interface when using the link-local address. Otherwise, the iscsi_io_tcp_connect function fails to invoke the connect function, and the standard error code 22 is generated.

If you use the iscsiadm -m node -p fe80::xxxx%enp3s0 -l format for login, the iscsi_addr_match function will compare the address fe80::xxxx%enp3s0 with the address fe80::xxxx in the node information returned by the server. The comparison result does not match, causing the login failure.

Therefore, **the current mechanism of iscsi-initiator-utils does not support login to the iSCSI server using a link-local address.**

7.4.3.2 The IPv6 Address Is Lost After the NIC Is Down.

Symptom

Run the ip link down+up NIC or ifconfig down+up NIC command to disable the NIC and then enable it to go online. Check the IP address configured on the NIC. It is found that the IPv4 address is not lost but the configured IPv6 address is lost.

Possible Cause

According to the processing logic in the kernel, if the NIC is set to the down state, all IPv4 and IPv6 addresses will be cleared. After the NIC is set to the up state, the IPv4 address is automatically restored, and the automatically configured IPv6 link-local address on the NIC is also restored. However, other IPv6 addresses are lost by default. To retain these IPv6 addresses, run the **sysctl -w net.ipv6.conf.<*NIC name*>.keep_addr_on_down=1** command.

7.4.3.3 Taking a Long Time to Add or Delete an IPv6 Address for a Bond Interface with Multiple IPv6 Addresses

Symptom

When users run the following command to add or delete (including flush) an IPv6 address, the waiting time increases linearly along with the number of IPv6 addresses configured on a bond interface. **X** is the least significant 16 bits that dynamically change. For example, it takes about five minutes to add 3000 IPv6 address to or delete them from a bond interface that already has four physical NICs using a single thread, while for a common physical NIC, it takes less than 10 seconds.

```
ip a add/del 192:168::18:X/64 dev DEVICE
```

Possible Cause

When an IPv6 address is added to a bond interface, the IPv6 multicast address is generated and synchronized to all physical NICs. The time required increases with the number of IPv6 addresses. As a result, it takes a too long time.

Solution

The IPv6 multicast address is generated by combining the least significant 24 bits of the IPv6 address and 33-33-ff. If there are too many multicast addresses, it takes a long time to add or delete the address. If there are a few multicast addresses, the time required is not affected.

It is recommended that you set the least significant 24 bits of the IPv6 address to be the same as the most significant 24 bits of the IPv6 address. In this way, a single NIC can communicate with external devices using only one IP address in a network segment.

7.4.3.4 Rsyslog Log Transmission Is Delayed in the Scenario Where Both IPv4 and IPv6 Are Used

Symptom

When both IPv4 and IPv6 addresses are configured in the configuration file of the rsyslog client and the port configurations are the same, there is a possibility that log output is delayed when the server collects logs.

Possible Cause

The delay is caused by the buffer queue mechanism of rsyslog. By default, rsyslog writes data to a file only when the number of buffer queues reaches a specified value.

Solution

You can disable the buffer queue mechanism by configuring the Direct mode. Add the following information at the beginning of the new remote transmission configuration file in the /etc/rsyslog.d directory on the rsyslog remote transmission server:

```
$ActionQueueType Direct
$MainMsgQueueType Direct
```

NOTE

- In direct mode, the queue size is reduced by 1. Therefore, one log is reserved in the queue for the next log output.
- The direct mode degrades the rsyslog performance of the server.

8 Managing Hard Disks Through LVM

8.1 LVM Overview

Logical Volume Manager (LVM) is a mechanism used for managing disk partitions in Linux. By adding a logical layer between disks and file systems, LVM shields the disk partition layout for file systems, thereby improving flexibility in managing disk partitions.

The procedure of managing a disk through LVM is as follows:

1. Create physical volumes for a disk.
2. Combine several physical volumes into a volume group.
3. Create logical volumes in the volume group.
4. Create file systems on logical volumes.

When disks are managed using LVM, file systems are distributed on multiple disks and can be easily resized as needed. Therefore, file system space will no longer be limited by disk capacities.

Basic Terms

- Physical media: refers to physical storage devices in the system, such as hard disks (**/dev/hda** and **/dev/sda**). It is the storage unit at the lowest layer of the storage system.
- Physical volume (PV): refers to a disk partition or device (such as a RAID) that has the same logical functions as a disk partition. PVs are basic logical storage blocks of LVM. A PV contains a special label that is stored in the second 512-byte sector by default. It can also be stored in one of the first four sectors. A label contains the universal unique identifier (UUID) of the PV, size of the block device, and the storage location of LVM metadata in the device.

- Volume group (VG): consists of PVs and shields the details of underlying PVs. You can create one or more logical volumes within a VG without considering detailed PV information.
- Logical volume (LV): A VG cannot be used directly. It can be used only after being partitioned into LVs. LVs can be formatted into different file systems and can be directly used after being mounted.
- Physical extent (PE): A PE is a small storage unit in a PV. The PE size is the same as the size of the logical extent in the VG.
- Logical extent (LE): An LE is a small storage unit in an LV. In one VG, the LEs of all the LVs have the same size.

8.2 Installing the LVM

NOTE

The LVM has been installed on the openEuler OS by default. You can run the **rpm -qa | grep lvm2** command to check whether it is installed. If the command output contains "lvm2", the LVM has been installed. In this case, skip this section. If no information is output, the LVM is not installed. Install it by referring to this section.

Step 1 Configure the local yum source. For details, see 10.1 Configuring the Repo Server.

Step 2 Clear the cache.

```
#dnf clean all
```

Step 3 Create a cache.

```
#dnf makecache
```

Step 4 Install the LVM.

```
#dnf install lvm2
```

Step 5 Check the installed RPM package.

```
#rpm -qa | grep lvm2
```

----End

8.3 Managing PVs

Creating a PV

Run the **pvcreate** command to create a PV.

```
pvcreate [option] devname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-f**: forcibly creates a PV without user confirmation.
  - **-u**: specifies the UUID of the device.
  - **-y**: answers yes to all questions.

- *devname*: specifies the name of the device corresponding to the PV to be created. If multiple PVs need to be created in batches, set this option to multiple device names and separate the names with spaces.

Example 1: Create PVs based on **/dev/sdb** and **/dev/sdc**.

```
pvcreate /dev/sdb /dev/sdc
```

Example 2: Create PVs based on **/dev/sdb1** and **/dev/sdb2**.

```
pvcreate /dev/sdb1 /dev/sdb2
```

Viewing a PV

Run the **pvdisplay** command to view PV information, including PV name, VG to which the PV belongs, PV size, PE size, total number of PEs, number of available PEs, number of allocated PEs, and UUID.

```
pvdisplay [option] devname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-s**: outputs information in short format.
  - **-m**: displays the mapping from PEs to LEs.
- *devname*: indicates the device corresponding to the PV to be viewed. If no PVs are specified, information about all PVs is displayed.

Example: Run the following command to display the basic information about the PV **/dev/sdb**:

```
pvdisplay /dev/sdb
```

Modifying PV Attributes

Run the **pvchange** command to modify the attributes of a PV.

```
pvchange [option] pvname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-u**: generates a new UUID.
  - **-x**: indicates whether PE allocation is allowed.
- *pvname*: specifies the name of the device corresponding to the PV to be modified. If multiple PVs need to be modified in batches, set this option to multiple device names and separate the names with spaces.

Example: Run the following command to prohibit PEs on the PV **/dev/sdb** from being allocated.

```
pvchange -x n /dev/sdb
```

Deleting a PV

Run the **pvremove** command to delete a PV.

```
pvremove [option] pvname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-f**: forcibly deletes a PV without user confirmation.
  - **-y**: answers yes to all questions.
- *pvname*: specifies the name of the device corresponding to the PV to be deleted. If multiple PVs need to be deleted in batches, set this option to multiple device names and separate the names with spaces.

Example: Run the following command to delete the PV **/dev/sdb**:

```
pvremove /dev/sdb
```

8.4 Managing VGs

Creating a VG

Run the **vgcreate** command to create a VG.

```
vgcreate [option] vgname pvname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-l**: specifies the maximum number of LVs that can be created on the VG.
  - **-p**: specifies the maximum number of PVs that can be added to the VG.
  - **-s**: specifies the PE size of a PV in the VG.
- *vgname*: name of the VG to be created.
- *pvname*: name of the PV to be added to the VG.

Example: Run the following command to create VG **vg1** and add the PVs **/dev/sdb** and **/dev/sdc** to the VG.

```
vgcreate vg1 /dev/sdb /dev/sdc
```

Viewing a VG

Run the **vgdisplay** command to view VG information.

```
vgdisplay [option] [vgname]
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-s**: outputs information in short format.
  - **-A**: displays only attributes of active VGs.
- *vgname*: name of the VG to be viewed. If no VGs are specified, information about all VGs is displayed.

Example: Run the following command to display the basic information about VG **vg1**:

```
vgdisplay vg1
```

Modifying VG Attributes

Run the **vgchange** command to modify the attributes of a VG.

```
vgchange [option] vgname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-a**: sets the active status of the VG.
- *vgname*: name of the VG whose attributes are to be modified.

Example: Run the following command to change the status of **vg1** to active.

```
vgchange -ay vg1
```

Extending a VG

Run the **vgextend** command to dynamically extend a VG. In this way, the VG size is extended by adding PVs to the VG.

```
vgextend [option] vgname pvname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **dev**: debugging mode.
  - **-t**: test only.
- *vgname*: name of the VG whose size is to be extended.
- *pvname*: name of the PV to be added to the VG.

Example: Run the following command to add PV **/dev/sdb** to VG **vg1**:

```
vgextend vg1 /dev/sdb
```

Shrinking a VG

Run the **vgreduce** command to delete PVs from a VG to reduce the VG size. A VG must contain at least one PV.

```
vgreduce [option] vgname pvname ...
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-a**: If no PVs are specified in the command, all empty PVs are deleted.
  - **--removemissing**: deletes lost PVs in the VG to restore the VG to the normal state.
- *vgname*: name of the VG to be shrunk.
- *pvname*: name of the PV to be deleted from the VG.

Example: Run the following command to remove PV **/dev/sdb2** from VG **vg1**:

```
vgreduce vg1 /dev/sdb2
```

Deleting a VG

Run the **vgremove** command to delete a VG.

```
vgremove [option] vgname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-f**: forcibly deletes a VG without user confirmation.
- *vgname*: name of the VG to be deleted.

Example: Run the following command to delete VG **vg1**.

```
vgremove vg1
```

8.5 Managing LVs

Creating an LV

Run the **lvcreate** command to create an LV.

```
lvcreate [option] vgname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-L**: specifies the size of the LV in kKmMgGtT.
  - **-l**: specifies the size of the LV (number of LEs).
  - **-n**: specifies the name of the LV to be created.
  - **-s**: creates a snapshot.
- *vgname*: name of the VG to be created.

Example 1: Run the following command to create a 10 GB LV in VG **vg1**.

```
lvcreate -L 10G vg1
```

Example 1: Run the following command to create a 200 MB LV in VG **vg1** and name the LV **lv1**.

```
lvcreate -L 200M -n lv1 vg1
```

Viewing an LV

Run the **lvdisplay** command to view the LV information, including the size of the LV, its read and write status, and snapshot information.

```
lvdisplay [option] [lvname]
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-v**: displays the mapping from LEs to PEs.
- *lvname*: device file corresponding to the LV whose attributes are to be displayed. If this option is not set, attributes of all LVs are displayed.

> **NOTE**
>
> Device files corresponding to LVs are stored in the VG directory. For example, if LV **lv1** is created in VG **vg1**, the device file corresponding to **lv1** is **/dev/vg1/lv1**.

Example: Run the following command to display the basic information about LV **lv1**:

```
lvdisplay /dev/vg1/lv1
```

Adjusting the LV Size

Run the **lvresize** command to increase or reduce the size of an LVM LV. This may cause data loss. Therefore, exercise caution when running this command.

```
lvresize [option] vgname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-L**: specifies the size of the LV in kKmMgGtT.
  - **-l**: specifies the size of the LV (number of LEs).
  - **-f**: forcibly adjusts the size of the LV without user confirmation.
- *lvname*: name of the LV to be adjusted.

Example 1: Run the following command to increase the size of LV **/dev/vg1/lv1** by 200 MB.

```
lvresize -L +200 /dev/vg1/lv1
```

Example 2: Run the following command to reduce the size of LV **/dev/vg1/lv1** by 200 MB.

```
lvresize -L -200 /dev/vg1/lv1
```

Extending an LV

Run the **lvextend** command to dynamically extend the size of an LV online without interrupting the access of applications to the LV.

```
lvextend [option] lvname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-L**: specifies the size of the LV in kKmMgGtT.
  - **-l**: specifies the size of the LV (number of LEs).
  - **-f**: forcibly adjusts the size of the LV without user confirmation.
- *lvname*: device file of the LV whose size is to be extended.

Example: Run the following command to increase the size of LV **/dev/vg1/lv1** by 100 MB.

```
lvextend -L +100M /dev/vg1/lv1
```

Shrinking an LV

Run the **lvreduce** command to reduce the size of an LV. This may delete existing data on the LV. Therefore, confirm whether the data can be deleted before running the command.

```
lvreduce [option] lvname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-L**: specifies the size of the LV in kKmMgGtT.
  - **-l**: specifies the size of the LV (number of LEs).
  - **-f**: forcibly adjusts the size of the LV without user confirmation.
- *lvname*: device file of the LV whose size is to be extended.

Example: Run the following command to reduce the space of LV **/dev/vg1/lvl** by 100 MB:

```
lvreduce -L -100M /dev/vg1/lv1
```

Deleting an LV

Run the **lvremove** command to delete an LV. If the LV has been mounted by running the **mount** command, you need to run the **umount** command to unmount the LV before running the **lvremove** command.

```
lvremove [option] vgname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-f**: forcibly deletes an LV without user confirmation.
- *vgname*: name of the LV to be deleted.

Example: Run the following command to delete LV **/dev/vg1/lv1**.

```
lvremove /dev/vg1/lv1
```

8.6 Creating and Mounting a File System

After creating an LV, you need to create a file system on the LV and mount the file system to the corresponding directory.

Creating a File System

Run the **mkfs** command to create a file system.

```
mkfs [option] lvname
```

In the preceding information:

- *option*: command parameter options. Common parameter options are as follows:
  - **-t**: specifies the type of the Linux system to be created, such as **ext2**, **ext3**, and **ext4**. The default type is **ext2**.
- *lvname*: name of the LV device file corresponding to the file system to be created.

Example: Run the following command to create the **ext4** file system on LV **/dev/vg1/lv1**:

```
mkfs -t ext4 /dev/vg1/lv1
```

Manually Mounting a File System

The file system that is manually mounted is not valid permanently. It does not exist after the OS is restarted.

Run the **mount** command to mount a file system.

```
mount lvname mntpath
```

In the preceding information:

- *lvname*: name of the LV device file corresponding to the file system to be mounted.
- *mntpath*: mount path.

Example: Run the following command to mount LV **/dev/vg1/lv1** to the directory **/mnt/data**.

```
mount /dev/vg1/lv1 /mnt/data
```

Automatically Mounting a File System

A file system that is automatically mounted does not exist after the OS is restarted. You need to manually mount the file system again. If you perform the following steps after manually mounting the file system, the file system can be automatically mounted after the OS is restarted.

Step 1 Run the **blkid** command to query the UUID of an LV. The following uses LV **/dev/vg1/lv1** as an example:

```
blkid /dev/vg1/lv1
```

Check the command output. It contains the following information in which *uuidnumber* is a string of digits, indicating the UUID, and *fstype* indicates the file system type.

/dev/vg1/lv1: UUID=" *uuidnumber* " TYPE=" *fstype* "

Step 2 Run the **vi /etc/fstab** command to edit the **fstab** file and add the following content to the end of the file:

```
UUID=uuidnumber  mntpath                fstype    defaults       0 0
```

In the preceding information:

- Column 1: indicates the UUID. Enter *uuidnumber* obtained in Step 1.
- Column 2: indicates the mount directory of the file system. Replace *mntpath* with the actual value.
- Column 3: indicates the file system format. Enter *fstype* obtained in Step 1.
- Column 4: indicates the mount option. In this example, **defaults** is used.
- Column 5: indicates the backup option. Enter either **1** (the system automatically backs up the file system) or **0** (the system does not back up the file system). In this example, **0** is used.
- Column 6: indicates the scanning option. Enter either **1** (the system automatically scans the file system during startup) or **0** (the system does not scan the file system). In this example, **0** is used.

Step 3 Verify the automatic mounting function.

1. Run the **umount** command to unmount the file system. The following uses LV **/dev/vg1/lv1** as an example:

```
umount /dev/vg1/lv1
```

2. Run the following command to reload all content in the **/etc/fstab** file:

   ```
   mount -a
   ```

3. Run the following command to query the file system mounting information (**/mnt/data** is used as an example):

   ```
   mount | grep /mnt/data
   ```

   Check the command output. If the command output contains the following information, the automatic mounting function takes effect:

   /dev/vg1/lv1 on /mnt/data

----End

9 Using the KAE

9.1 Overview

Kunpeng Accelerator Engine (KAE) is a software acceleration library of openEuler, which provides hardware acceleration engine function on the Kunpeng 920 processor. The engine supports symmetric encryption, asymmetric encryption, and digital signature. It is ideal for accelerating SSL/TLS applications, and can significantly reduce processor consumption and improve processor efficiency. In addition, users can quickly migrate existing services through the standard OpenSSL interface.

The KAE supports the following algorithms:

- Digest algorithm SM3, which supports the asynchronous mode.
- Symmetric encryption algorithm SM4, which supports asynchronous, CTR, XTS, and CBC modes.
- Symmetric encryption algorithm AES, which supports asynchronous, ECB, CTR, XTS, and CBC modes.
- Asymmetric algorithm RSA, which supports asynchronous mode, and key sizes 1024, 2048, 3072, and 4096.
- Key negotiation algorithm DH, which supports asynchronous mode, and key sizes 768, 1024, 1536, 2048, 3072, and 4096.

9.2 Application Scenarios

The KAE applies to the following scenarios, as shown in Table 9-1.

Table 9-1 Application scenarios

Scenario	Data
Big data	Stream data
Data encryption	Block data
Intelligent security protection	Video stream data
Web service	Handshake connections

9.3 Installing, Running, and Uninstalling the KAE

9.3.1 Installing the Accelerator Software Packages

9.3.1.1 Preparing for Installation

Environment Requirements

- The accelerator engine is enabled on TaiShan 200 servers.

  NOTE

  - You need to import the accelerator license. For details, see section "License Management" in the TaiShan Rack Server iBMC (V500 or Later) User Guide.
  - If the accelerator is used in the physical machine scenario, the SMMU must be disabled. For details, see the TaiShan 200 Server BIOS Parameter Reference.
- CPU: Kunpeng 920
- OS: openEuler-20.03-LTS-aarch64-dvd.iso

KAE Software Description

Table 9-2 RPM software packages of the KAE

Software Package	Description
kae_driver-*version number-1.OS type*.aarch64.rpm	Accelerator driver, including the uacce.ko, hisi_qm.ko, hisi_sec2.ko, and hisi_hpre.ko kernel modules. Support: SM3, SM4, AES, RSA, and DH algorithms.
libwd-*version number-1.OS type*.aarch64.rpm	Coverage: **libwd.so** dynamic link library. It provides interfaces for the KAE.
libkae-*version number-1.OS type*.aarch64.rpm	Dependency: libwd RPM package. Coverage: **libkae.so** dynamic library. Support: SM3, SM4, AES, RSA, and DH algorithms.

9.3.1.2 Installing the Accelerator Software Package

Prerequisites

- The remote SSH login tool has been installed on the local PC.
- The openEuler OS has been installed.
- The RPM tool is running properly.
- OpenSSL 1.1.1a or a later version has been installed.
  You can run the following commands to query the version number of OpenSSL:
  - openssl version

Procedure

Step 1 Log in to the openEuler OS CLI as user **root**.

Step 2 Create a directory for storing accelerator engine software packages.

Step 3 Use SSH to copy all accelerator engine software package to the created directory.

Step 4 In the directory, run the **rpm -ivh** command to install the accelerator engine software packages.

NOTE

Install the **libwd** package first because the **libkae** package installation depends on the **libwd** package.

```
rpm -ivh uacce*.rpm hisi*.rpm libwd-*.rpm libkae*.rpm
Verifying...                       ################################# [100%]
Preparing...                       ################################# [100%]
checking installed modules
uacce modules start to install
Updating / installing...
   1:uacce-1.2.10-4.oe1             ################################# [ 14%]
uacce modules installed
   2:libwd-1.2.10-3.oe1             ################################# [ 29%]
   3:libkae-1.2.10-3.oe1            ################################# [ 43%]
checking installed modules
hisi hpre modules start to install
   4:hisi hpre-1.2.10-4.oe1         ################################# [ 57%]
hisi hpre modules installed
checking installed modules
hisi rde modules start to install
   5:hisi rde-1.2.10-4.oe1          ################################# [ 71%]
hisi rde modules installed
checking installed modules
hisi sec2 modules start to install
   6:hisi sec2-1.2.10-4.oe1         ################################# [ 86%]
hisi sec2 modules installed
checking installed modules
hisi zip modules start to install
   7:hisi zip-1.2.10-4.oe1          ################################# [100%]
hisi_zip modules installed
```

Step 5 Run the **rpm -qa** command to check whether the accelerator software packages have been installed properly. Run the **rpm -ql** command to check whether files in the software packages are correct. The following is an example:

```
rpm -qa|grep -E "hisi|uacce|libwd|libkae"
hisi_rde-1.2.10-4.oe1.aarch64
hisi_sec2-1.2.10-4.oe1.aarch64
libkae-1.2.10-3.oe1.aarch64
hisi_hpre-1.2.10-4.oe1.aarch64
uacce-1.2.10-4.oe1.aarch64
libwd-1.2.10-3.oe1.aarch64
hisi_zip-1.2.10-4.oe1.aarch64
rpm -ql uacce hisi* libwd* libkae
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/hisi_qm.ko
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/uacce.ko
/etc/modprobe.d/hisi hpre.conf
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/hisi hpre.ko
/etc/modprobe.d/hisi rde.conf
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/hisi rde.ko
/etc/modprobe.d/hisi sec2.conf
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/hisi sec2.ko
/etc/modprobe.d/hisi zip.conf
/lib/modules/4.19.90-2003.4.0.0036.oe1.aarch64/extra/hisi zip.ko
/usr/include/warpdrive/config.h
/usr/include/warpdrive/include/uacce.h
/usr/include/warpdrive/smm.h
/usr/include/warpdrive/wd.h
/usr/include/warpdrive/wd bmm.h
/usr/include/warpdrive/wd cipher.h
/usr/include/warpdrive/wd comp.h
/usr/include/warpdrive/wd dh.h
/usr/include/warpdrive/wd digest.h
/usr/include/warpdrive/wd rsa.h
/usr/lib64/libwd.so.1.2.10
/usr/local/lib/engines-1.1/libkae.so.1.2.10
```

Step 6 Restart the system or run commands to manually load the accelerator engine drivers to the kernel in sequence, and check whether the drivers are successfully loaded.

```
# modprobe uacce
# lsmod | grep uacce
# modprobe hisi_qm
# lsmod | grep hisi_qm
# modprobe hisi_qm
# modprobe hisi_sec2 # Loads the hisi_sec2 driver to the kernel based on the configuration file in /etc/modprobe.d/hisi_sec2.conf.
# modprobe hisi_hpre # Loads the hisi_hpre driver to the kernel based on the configuration file in /etc/modprobe.d/hisi_hpre.conf.
```

----End

Setting Environment Variables

Run the following command to export the environment variable (If you have specified the installation directory, use the actual installation directory instead of **/usr/local**):

```
export OPENSSL_ENGINES=/usr/local/lib/engines-1.1
```

Performing the Post-Installation Check

Run the **rpm -qa** command to check whether the accelerator engine software packages are successfully installed.

If the command output contains *software package name-version number-*, the software package is successfully installed. The following is an example:

```
rpm -qa|grep -E "hisi|uacce|libwd|libkae"
hisi_rde-1.2.10-4.oe1.aarch64
hisi_sec2-1.2.10-4.oe1.aarch64
libkae-1.2.10-3.oe1.aarch64
hisi_hpre-1.2.10-4.oe1.aarch64
uacce-1.2.10-4.oe1.aarch64
libwd-1.2.10-3.oe1.aarch64
hisi_zip-1.2.10-4.oe1.aarch64
```

9.3.1.3 Performing Required Operations After Installation

9.3.1.3.1 Testing the OpenSSL Accelerator Engine

You can run the following commands to test some accelerator functions.

- Use the OpenSSL software algorithm to test the RSA performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed rsa2048
...
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.001384s 0.000035s   724.1  28365.8.
```

- Use the KAE to test the RSA performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae rsa2048
....
                  sign    verify    sign/s verify/s
rsa 2048 bits 0.000355s 0.000022s   2819.0  45478.4
```

NOTE

#After KAE acceleration, the signature performance is improved from 724.1 sign/s to 2819 sign/s.

- Use the OpenSSL software algorithm to test the asynchronous RSA performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -async_jobs 36 rsa2048
....
                   sign    verify    sign/s verify/s
rsa 2048 bits 0.001318s 0.000032s    735.7  28555
```

- Use the KAE to test the asynchronous RSA performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -engine kae -elapsed -async_jobs
36 rsa2048
```

```
....
                sign    verify    sign/s verify/s
rsa 2048 bits 0.000018s 0.000009s  54384.1 105317.0
```

NOTE

#After KAE acceleration, the asynchronous RSA signature performance is improved from 735.7 sign/s to 54384.1 sign/s.

- Use the OpenSSL software algorithm to test the performance of the SM4 CBC mode.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp sm4-cbc
You have chosen to measure elapsed time instead of user CPU time.
....
Doing sm4-cbc for 3s on 10240 size blocks: 2196 sm4-cbc's in 3.00s  ....
type         51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304
bytes8388608 bytes
sm4-cbc         82312.53k    85196.80k    85284.18k    85000.85k
85284.18k    85261.26k
```

- Use the KAE to test the SM4 CBC mode performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae -evp sm4-cbc
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
...
Doing sm4-cbc for 3s on 1048576 size blocks: 11409 sm4-cbc's in 3.00s
...
type         51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304
bytes8388608 bytes
sm4-cbc        383317.33k   389427.20k   395313.15k   392954.73k
394264.58k   394264.58k
```

NOTE

After KAE acceleration, the SM4 CBC mode performance is improved from 82312.53 kbit/s to 383317.33 kbit/s when the input data block size is 8 MB.

- Use the OpenSSL software algorithm to test the SM3 mode performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp sm3
You have chosen to measure elapsed time instead of user CPU time.
Doing sm3 for 3s on 102400 size blocks: 1536 sm3's in 3.00s
....
type         51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304
bytes8388608 bytes
sm3            50568.53k    52428.80k    52428.80k    52428.80k    52428.80k
52428.80k
```

- Use the KAE to test the SM3 mode performance.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -engine kae -evp sm3
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
Doing sm3 for 3s on 102400 size blocks: 19540 sm3's in 3.00s
```

```
....
type          51200 bytes  102400 bytes  1048576 bytes 2097152 bytes 4194304
bytes 8388608 bytes
sm3           648243.20k   666965.33k   677030.57k   678778.20k
676681.05k   668292.44k
```

> **NOTE**
>
> After KAE acceleration, the SM3 algorithm performance is improved from 52428.80 kbit/s to 668292.44 kbit/s when the input data block size is 8 MB.

- Use the OpenSSL software algorithm to test the asynchronous performance of the AES algorithm in CBC mode.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp aes-128-cbc
-async_jobs 4
You have chosen to measure elapsed time instead of user CPU time.
Doing aes-128-cbc for 3s on 51200 size blocks: 65773 aes-128-cbc's in 3.00s
Doing aes-128-cbc for 3s on 102400 size blocks: 32910 aes-128-cbc's in 3.00s
....
type         51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304
bytes8388608 bytes
aes-128-cbc   1122525.87k  1123328.00k  1120578.22k  1121277.27k
1119879.17k  1115684.86k
```

- Use the KEA engine to test the asynchronous performance of the AES algorithm in CBC mode.

```
linux-rmw4:/usr/local/bin # ./openssl speed -elapsed -evp aes-128-cbc
-async_jobs 4 -engine kae
engine "kae" set.
You have chosen to measure elapsed time instead of user CPU time.
Doing aes-128-cbc for 3s on 51200 size blocks: 219553 aes-128-cbc's in 3.00s
Doing aes-128-cbc for 3s on 102400 size blocks: 117093 aes-128-cbc's in 3.00s
....
type         51200 bytes 102400 bytes1048576 bytes2097152 bytes4194304
bytes8388608 bytes
aes-128-cbc   3747037.87k  3996774.40k  1189085.18k  1196774.74k
1196979.11k  1199570.94k
```

> **NOTE**
>
> - The AES algorithm supports only asynchronous usage when the data length is 256 KB or less.
> - After KAE acceleration, the AES algorithm performance is improved from 1123328.00 kbit/s to 3996774.40 kbit/s when the input data block size is 100 KB.

9.3.2 Upgrading the Accelerator Software Packages

Scenario

You can run the **rpm -Uvh** command to upgrade the accelerator software.

Procedure

Step 1 Download the latest accelerator engine software packages from the openEuler community.

Step 2 Use SSH to log in to the Linux CLI as user **root**.

Step 3 Save the downloaded software packages to a directory.

Step 4 In the directory, run the **rpm -Uvh** command to upgrade the accelerator driver package and engine library package. The following is an example:

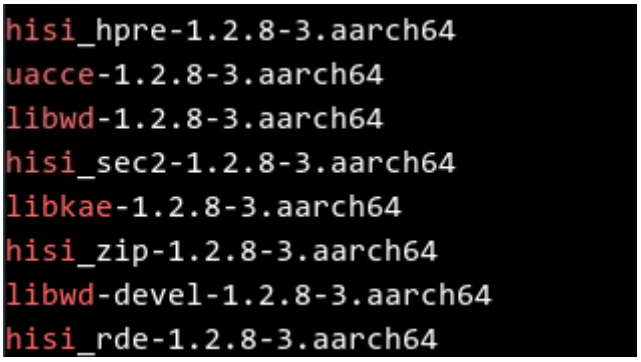
The command and output are as follows:

```
rpm -Uvh uacce*.rpm hisi*.rpm libwd*.rpm libkae*.rpm
```

```
warning: uacce-1.2.8-3.aarch64.rpm: Header V3 RSA/SHA1 Signature, key ID b25e7f66: NOKEY
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
checking installed modules
uacce modules start to install
Updating / installing...
   1:uacce-1.2.8-3                    ################################# [  6%]
uacce modules installed
   2:libwd-1.2.8-3                    ################################# [ 13%]
   3:libkae-1.2.8-3                   ################################# [ 19%]
checking installed modules
hisi_hpre modules start to install
   4:hisi_hpre-1.2.8-3                ################################# [ 25%]
hisi_hpre modules installed
checking installed modules
hisi_rde modules start to install
   5:hisi_rde-1.2.8-3                 ################################# [ 31%]
hisi_rde modules installed
checking installed modules
hisi_sec2 modules start to install
   6:hisi_sec2-1.2.8-3                ################################# [ 38%]
hisi_sec2 modules installed
checking installed modules
hisi_zip modules start to install
   7:hisi_zip-1.2.8-3                 ################################# [ 44%]
hisi_zip modules installed
   8:libwd-devel-1.2.8-3              ################################# [ 50%]
Cleaning up / removing...
   9:hisi_zip-1.2.8-2                 ################################# [ 56%]
hisi_zip modules uninstalled
  10:hisi_sec2-1.2.8-2                ################################# [ 63%]
hisi_sec2 modules uninstalled
  11:hisi_rde-1.2.8-2                 ################################# [ 69%]
hisi_rde modules uninstalled
  12:hisi_hpre-1.2.8-2                ################################# [ 75%]
hisi_hpre modules uninstalled
  13:uacce-1.2.8-2                    ################################# [ 81%]
uacce modules uninstalled
  14:libwd-devel-1.2.8-2              ################################# [ 88%]
  15:libkae-1.2.8-2                   ################################# [ 94%]
  16:libwd-1.2.8-2                    ################################# [100%]
```

Step 5 Run the **rpm -qa** command to check whether the upgrade is successful. Ensure that the queried version is the latest version.

```
rpm -qa | grep -E "hisi|uacce|libwd|libkae"
```



Step 6 Restart the system or run the following commands to manually uninstall the drivers of the earlier version, load the drivers of the latest version, and check whether the new drivers are successfully loaded.

```
Uninstall the existing drivers.
# lsmod | grep uacce
uacce              262144  3 hisi_hpre,hisi_sec2,hisi_qm
#
# rmmod hisi_hpre
# rmmod hisi_sec2
# rmmod hisi_qm
# rmmod uacce
# lsmod | grep uacce
#
Load the new drivers.# modprobe uacce
# modprobe hisi_qm# modprobe hisi_sec2 #Loads the hisi_sec2 driver to the kernel based on the configuration file in /etc/modprobe.d/hisi_sec2.conf.
# modprobe hisi_hpre  #Loads the hisi_hpre driver to the kernel based on the configuration file in /etc/modprobe.d/hisi_hpre.conf.
# lsmod | grep uacce
uacce             36864  3 hisi_sec2,hisi_qm,hisi_hpre
```

----End

9.3.3 Uninstalling the Accelerator Software Packages

Scenario

You do not need the accelerator engine software or you want to install new accelerator engine software.

Procedure

Step 1 Use SSH to log in to the Linux CLI as user **root**.

Step 2 Restart the system or run commands to manually uninstall the accelerator drivers loaded to the kernel, and check whether the drivers are successfully uninstalled.

```
# lsmod | grep uacce
```

```
uacce              36864  3 hisi_sec2,hisi_qm,hisi_hpre
# rmmod hisi_hpre
# rmmod hisi_sec2
# rmmod hisi_qm
# rmmod uacce
# lsmod | grep uacce
#
```

Step 3 Run the **rpm -e** command to uninstall the accelerator engine software packages. The following is an example:

NOTE

Due to the dependency relationships, the **libkae** package must be uninstalled before the **libwd** package.

```
rpm -e libkae libwd-devel libwd hisi_hpre hisi_sec2 hisi_zip hisi_rde uacce
```

```
hisi_rde modules uninstalling
hisi_rde modules uninstalled
hisi_zip modules uninstalling
hisi_zip modules uninstalled
hisi_sec2 modules uninstalling
hisi_sec2 modules uninstalled
hisi_hpre modules uninstalling
hisi_hpre modules uninstalled
uacce modules uninstalling
uacce modules uninstalled
```

Step 4 Run the **rpm -qa |grep** command to check whether the uninstallation is successful.

```
rpm -qa | grep -E "hisi|uacce|libwd|libkae"
```

----End

9.4 Querying Logs

Table 9-3 lists log information related to the accelerator engine.

Table 9-3 Log information

Directory	File	Description
/var/log/	kae.log	By default, the log level of the OpenSSL engine log is **error**. To set the log level, perform the following procedure: 1. Run **export KAE_CONF_ENV=/var/log/**. 2. Create the **kae.cnf** file in **/var/log/**.

Directory	File	Description
		3. In the **kae.cnf** file, configure the content as follows: [LogSection] debug_level=error #Value: **none**, **error**, **info**, **warning** or **debug** **NOTE** In normal cases, you are advised not to enable the **info** or **debug** log level. Otherwise, the accelerator performance will deteriorate.
/var/log/	message/syslog	• Kernel logs are stored in the **/var/log/message** directory. **NOTE** Alternatively, you can run the **dmesg > /var/log/dmesg.log** command to collect driver and kernel logs.

9.5 Application Cases

9.5.1 Acceleration Engine Application

NOTE

If you have not purchased the engine license, you are advised not to use the KAE engine to invoke the corresponding algorithms. Otherwise, the performance of the OpenSSL encryption algorithm may be affected.

9.5.1.1 Example Code for the KAE

```
#include <stdio.h>
#include <stdlib.h>
/* OpenSSL headers */
#include <openssl/bio.h>
#include <openssl/ssl.h>
#include <openssl/err.h>
#include <openssl/engine.h>
int main(int argc, char **argv)
{
    /* Initializing OpenSSL */
    SSL_load_error_strings();
    ERR_load_BIO_strings();
    OpenSSL_add_all_algorithms();
    /*You can use ENGINE_by_id Function to get the handle of the Huawei
Accelerator Engine*/
```

```
    ENGINE *e = ENGINE_by_id("kae");
    /* Enable the accelerator asynchronization function. This parameter is
optional. The value 0 indicates disabled, and the value 1 indicates enabled.
The asynchronous function is enabled by default. */
    ENGINE_ctrl_cmd_string(e, "KAE_CMD_ENABLE_ASYNC", "1", 0)
    ENGINE_init(e);

    RSA*rsa=RSA_new_method(e);#Specify the engine for RSA encryption and
decryption.
    /*The user code*/
    ......
;
    ENGINE_free(e);
;
}
```

9.5.1.2 Using the KAE in the OpenSSL Configuration File openssl.cnf

Create the **openssl.cnf** file and add the following configuration information to the file:

```
openssl_conf=openssl_def
[openssl_def]
engines=engine_section
[engine_section]
kae=kae_section
[kae_section]
engine_id=kae
dynamic_path=/usr/local/lib/engines-1.1/kae.so
KAE_CMD_ENABLE_ASYNC=1 #The value 0 indicates that the asynchronous function
is disabled. The value 1 indicates that the asynchronous function is enabled.
The asynchronous function is enabled by default.
default_algorithms=ALL
init=1
```

Export the environment variable **OPENSSL_CONF**.

```
export OPENSSL_CONF=/home/app/openssl.cnf #Path for storing the openssl.cnf
file
```

The following is an example of the OpenSSL configuration file:

```
#include <stdio.h>
#include <stdlib.h>
/* OpenSSL headers */
#include <openssl/bio.h>
#include <openssl/ssl.h>
#include <openssl/err.h>
#include <openssl/engine.h>
```

```
int main(int argc, char **argv)
{
    /* Initializing OpenSSL */
    SSL_load_error_strings();
    ERR_load_BIO_strings();
#Load openssl configure
OPENSSL_init_crypto(OPENSSL_INIT_LOAD_CONFIG, NULL);
OpenSSL_add_all_algorithms();
    /*You can use ENGINE_by_id Function to get the handle of the Huawei
Accelerator Engine*/
    ENGINE *e = ENGINE_by_id("kae");
    /*The user code*/
    ......
;
    ENGINE_free(e);
;
```

9.6 Troubleshooting

9.6.1 Initialization Failure

Symptom

The accelerator engine is not completely loaded due to an initialization failure.

Solution

Step 1 Check whether the accelerator drivers are loaded successfully. Specifically, run the **lsmod** command to check whether uacce.ko, qm.ko, sgl.ko, hisi_sec2.ko, hisi_hpre.ko, hisi_zip.ko, and hisi_rde.ko exist.

```
# lsmod | grep uacce
uacce               262144  2
hisi_hpre,hisi_qm,hisi_sec2,hisi_zip,hisi_rde
```

Step 2 Check whether the accelerator engine library exists in **/usr/lib64** (directory for RPM installation) or **/usr/local/lib** (directory for source code installation) and the OpenSSL installation directory, and check whether the correct soft link is established.

```
 [root@localhost home]# ll /usr/local/lib/engines-1.1/ |grep kae
#Check whether the KAE has been correctly installed and whether a soft link
has been established. If yes, the displayed information is as follows:
# ll /usr/local/lib/engines-1.1/ |grep kae
lrwxrwxrwx. 1 root root    22 Nov 12 02:33 kae.so -> kae.so.1.0.1
lrwxrwxrwx. 1 root root    22 Nov 12 02:33 kae.so.0 -> kae.so.1.0.1
-rwxr-xr-x. 1 root root 112632 May 25  2019 kae.so.1.0.1
```

```
[[root@localhost home]#
 [root@localhost home]# ll /usr/lib64/ | grep libwd
#Check whether libwd has been correctly installed and whether a soft link has
been established. If yes, the displayed information is as follows:
lrwxrwxrwx.  1 root root      14 Nov 12 02:33 libwd.so -> libwd.so.1.0.1
lrwxrwxrwx.  1 root root      14 Nov 12 02:33 libwd.so.0 -> libwd.so.1.0.1
-rwxr-xr-x.  1 root root   137120 May 25  2019 libwd.so.1.0.1
[root@localhost home]#
```

Step 3 Check whether the path of the OpenSSL engine library can be exported by running the **export** command.

```
# echo $OPENSSL_ENGINES
# export OPENSSL_ENGINES=/usr/local/lib/engines-1.1
#  echo $OPENSSL_ENGINES
/usr/local/lib/engines-1.1
```

----End

9.6.2 Failed to Identify Accelerator Devices After the Acceleration Engine Is Installed

Symptom

After the acceleration engine is installed, the accelerator devices cannot be identified.

Solution

Step 1 Check whether the device exists in the virtual file system. Normally, the following accelerator devices are displayed:

```
# ls -al /sys/class/uacce/
total 0
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-2
-> ../../devices/pci0000:78/0000:78:00.0/0000:79:00.0/uacce/hisi_hpre-2
lrwxrwxrwx. 1 root root 0 Nov 14 03:45 hisi_hpre-3
-> ../../devices/pci0000:b8/0000:b8:00.0/0000:b9:00.0/uacce/hisi_hpre-3
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-4
-> ../../devices/pci0000:78/0000:78:01.0/uacce/hisi_rde-4
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_rde-5
-> ../../devices/pci0000:b8/0000:b8:01.0/uacce/hisi_rde-5
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-0
-> ../../devices/pci0000:74/0000:74:01.0/0000:76:00.0/uacce/hisi_sec-0
lrwxrwxrwx. 1 root root 0 Nov 14 08:39 hisi_sec-1
-> ../../devices/pci0000:b4/0000:b4:01.0/0000:b6:00.0/uacce/hisi_sec-1
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-6
-> ../../devices/pci0000:74/0000:74:00.0/0000:75:00.0/uacce/hisi_zip-6
```

```
lrwxrwxrwx. 1 root root 0 Nov 17 22:09 hisi_zip-7
-> ../../devices/pci0000:b4/0000:b4:00.0/0000:b5:00.0/uacce/hisi_zip-7
```

Step 2 If you want to use the HPRE device but the device is not found in Step 1, check whether the accelerator software is correctly installed by referring to 9.6.3 Failed to Upgrade the Accelerator Drivers.

Step 3 If the accelerator software is correctly installed, run the **lspci** command to check whether the physical device exists.

```
# lspci | grep HPRE
79:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd.
HiSilicon HPRE Engine (rev 21)
b9:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd.
HiSilicon HPRE Engine (rev 21)
## lspci | grep SEC
76:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd.
HiSilicon SEC Engine (rev 21)
b6:00.0 Network and computing encryption device: Huawei Technologies Co., Ltd.
HiSilicon SEC Engine (rev 21)
## lspci | grep RDE
78:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine
(rev 21)
b8:01.0 RAID bus controller: Huawei Technologies Co., Ltd. HiSilicon RDE Engine
(rev 21)
## lspci | grep ZIP
75:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP
Engine (rev 21)
b5:00.0 Processing accelerators: Huawei Technologies Co., Ltd. HiSilicon ZIP
Engine (rev 21)
#
```

Step 4 If no physical device is found in Step 3, perform the following operations:

- Check whether the accelerator license has been imported. If no, import the accelerator license. For details, see "License Management" in the TaiShan Rack Server iBMC (V500 or Later) User Guide. After the accelerator license is imported, power off and restart the BMC to enable the license.
- Check whether the BMC and BIOS versions support the accelerator feature.

----End

9.6.3 Failed to Upgrade the Accelerator Drivers

Symptom

After the accelerator drivers are upgraded, the driver version is not changed after the system is restarted.

Possible Cause

Before the accelerator drivers are upgraded, the system upgrades other driver packages. These driver packages may update the boot file system initramfs, and update the accelerator drivers to initramfs before upgrade. For example, if the NIC driver is updated or initramfs is manually updated, the system loads the accelerator drivers from initramfs first during restart.

Solution

After the accelerator drivers are upgraded, run the **dracut --force** command to update initramfs again.

10 Configuring Services

10.1 Configuring the Repo Server

NOTE

This section uses the **openEuler-20.03-LTS-aarch64-dvd.iso** image file as an example. Modify the image file as required.

10.1.1 Overview

Create the **openEuler-20.03-LTS-aarch64-dvd.iso** image provided by openEuler as the repo source. The following uses Nginx as an example to describe how to deploy the repo source and provide the HTTP service.

10.1.2 Creating or Updating a Local Repo Source

Mount the openEuler image **openEuler-20.03-LTS-aarch64-dvd.iso** to create and update a repo source.

10.1.2.1 Obtaining the ISO Image File

Obtaining the Software Package

Obtain the openEuler software package from the following website:

https://openeuler.org/zh/download.html

10.1.2.2 Mounting an ISO File to Create a Repo Source

Run the mount command to mount the image file.

The following is an example:

```
mount /home/openEuler/openEuler-20.03-LTS-aarch64-dvd.iso  /mnt/
```

The mounted mnt directory is as follows:

```
.
|── boot.catalog
|── docs
|── EFI
|── images
|── Packages
|── repodata
|── TRANS.TBL
└── RPM-GPG-KEY-openEuler
```

In the preceding command, **Packages** indicates the directory where the RPM package is stored, **repodata** indicates the directory where the repo source metadata is stored, and **RPM-GPG-KEY-openEuler** indicates the public key for signing openEuler.

10.1.2.3 Creating a Local Repo Source

You can copy related files in the image to a local directory to create a local repo source. The following is an example:

```
mount /home/openEuler/openEuler-20.03-LTS-aarch64-dvd.iso  /mnt/
mkdir -p /srv/repo/
cp -r /mnt/Packages /srv/repo/
cp -r /mnt/repodata /srv/repo/
cp -r /mnt/RPM-GPG-KEY-openEuler /srv/repo/
```

The local repo directory is as follows:

```
.
|── Packages
|── repodata
└── RPM-GPG-KEY-openEuler
```

Packages indicates the directory where the RPM package is stored, **repodata** indicates the directory where the repo source metadata is stored, and **RPM-GPG-KEY-openEuler** indicates the public key for signing openEuler.

10.1.2.4 Updating the Repo Source

You can update the repo source in either of the following ways:

- Use the ISO file of the new version to update the existing repo source. The method is the same as that for creating a repo source. That is, mount the image or copy the image to the local directory.
- Add rpm packages to the Packages directory of the repo source and update the repo source. You can run the createrepo command to update the repo source.

  ```
  dnf install createrepo
  createrepo --update --workers=10 /srv/repo
  ```

  In this command, --update indicates the update, and --workers indicates the number of threads, which can be customized.

10.1.3 Deploying the Remote Repo Source

Install openEuler OS and deploy the repo source using Nginx on openEuler OS.

10.1.3.1 Installing and Configuring Nginx

1. Download the Nginx tool and install it.
2. After installing Nginx, configure /etc/nginx/nginx.conf.

NOTE

The configuration content in this document is for reference only. You can configure the content based on the site requirements (for example, security hardening requirements).

```
user  root;
worker processes auto;                          # You are advised to set this
parameter to core-1.
error_log /var/log/nginx/error.log warn;          # log storage location
pid       /var/run/nginx.pid;

events {
   worker_connections  1024;
}

http {
   include      /etc/nginx/mime.types;
   default_type  application/octet-stream;

   log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';

   access_log  /var/log/nginx/access.log  main;
   sendfile       on;
   keepalive timeout  65;

   server {
      listen      80;
server name localhost;                      #Server name (URL)
      client max body size 4G;
root /srv/repo;                          #Default service directory

      location / {
autoindex on;                               # Enable the access to lower-layer files
in the directory.
         autoindex exact size on;
         autoindex localtime  on;
      }

   }

}
```

10.1.3.2 Starting Nginx

1. Run the systemd command to start the Nginx service.

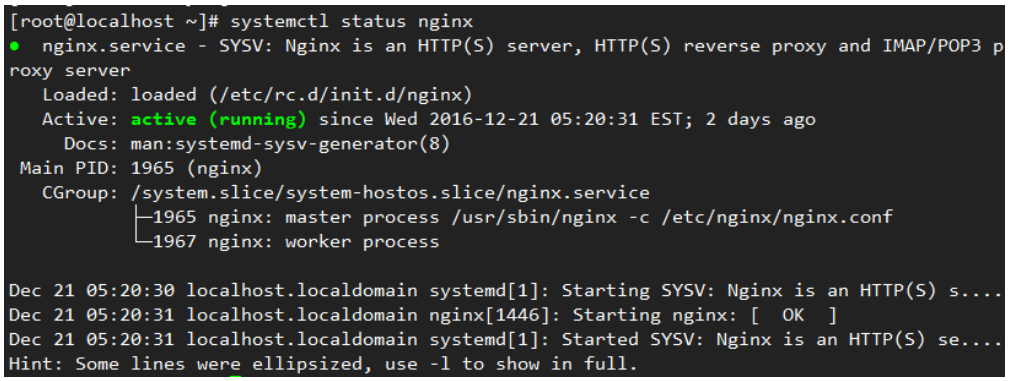
```
systemctl enable nginx
systemctl start nginx
```

2. You can run the following command to check whether the Nginx is started successfully:

```
systemctl status nginx
```

– Figure 10-1 indicates that the Nginx service is started successfully.

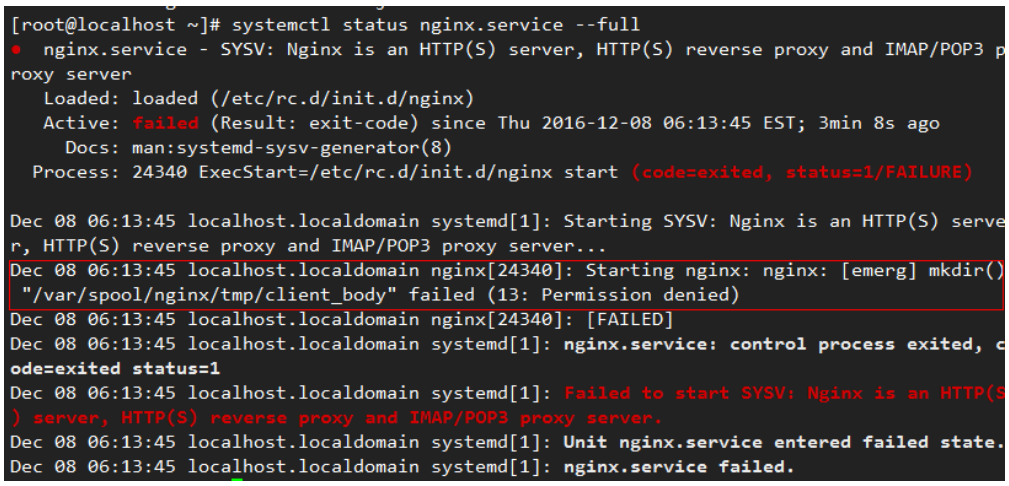
Figure 10-1 The Nginx service is successfully started.



– If the Nginx service fails to be started, view the error information.

```
systemctl status nginx.service --full
```

Figure 10-2 Nginx startup failure



As shown in Figure 10-2, the Nginx service fails to be created because the /var/spool/nginx/tmp/client_body directory fails to be created. You need to manually create the directory. Similar problems are solved as follows:

```
mkdir -p /var/spool/nginx/tmp/client body
mkdir -p /var/spool/nginx/tmp/proxy
mkdir -p /var/spool/nginx/tmp/fastcgi
mkdir -p /usr/share/nginx/uwsgi temp
mkdir -p /usr/share/nginx/scgi_temp
```

10.1.3.3 Deploying the Repo Source

1. Run the following command to create the /srv/repo directory specified in the Nginx configuration file /etc/nginx/nginx.conf:

```
mkdir -p /srv/repo
```

2. Set the SELinux working mode to the permissive mode:

```
setenforce permissive
```

> **NOTE**
>
> After the repo server is restarted, you need to configure the repo server again.

3. Configure firewall rules to enable the port (port 80) configured for Nginx.

```
firewall-cmd --add-port=80/tcp --permanent
firewall-cmd --reload
```

Check whether port 80 is enabled. If the output is **yes**, port 80 is enabled.

```
firewall-cmd --query-port=80/tcp
```

You can also enable port 80 using iptables.

```
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
```

4. After the Nginx service is configured, you can use the IP address to access the web page, as shown in Figure 10-3.

Figure 10-3 Nginx deployment succeeded

5. Use either of the following methods to add the repo source to the /srv/repo directory:
   - Copy related files in the image to the /srv/repo directory.

     ```
     mount /home/openEuler/openEuler-20.03-LTS-aarch64-dvd.iso  /mnt/
     cp -r /mnt/Packages /srv/repo/
     cp -r /mnt/repodata /srv/repo/
     cp -r /mnt/RPM-GPG-KEY-openEuler /srv/repo/
     ```

     The **openEuler-20.03-LTS-aarch64-dvd.iso** file is stored in the **/home/openEuler** directory.
   - Create a soft link for the repo source in the /srv/repo directory.

     ```
     ln -s /home/openEuler/os /srv/repo/os
     ```

     /home/openEuler/os is the created repo source, and /srv/repo/os points to /home/openEuler/os.

10.1.4 Using the repo Source

The repo source can be configured as a yum source. Yellow dog Updater,Modified (yum for short) is a shell front-end software package manager. Based on the Redhat package manager (RPM), YUM can automatically download the rpm package from the specified server, install the package, and process dependent relationship. It supports one-off installation for all dependent software packages.

10.1.4.1 Configuring repo as the yum Source

You can configure the built repo as the yum source and create the ***.repo configuration file (the extension .repo is mandatory) in the /etc/yum.repos.d/ directory. You can configure the yum source on the local host or HTTP server.

- Configuring the local yum source.

  Create the **openEuler.repo** file in the **/etc/yum.repos.d** directory and use the local repository as the yum source. The content of the **openEuler.repo** file is as follows:

  ```
  [base]
  name=base
  baseurl=file:///srv/repo
  enabled=1
  gpgcheck=1
  gpgkey=file:///srv/repo/RPM-GPG-KEY-openEuler
  ```

  NOTE

  gpgcheck indicates whether to enable the GNU privacy guard (GPG) to check the validity and security of sources of RPM packages. **1** indicates GPG check is enabled. **0** indicates the GPG check is disabled. If this option is not specified, the GPG check is enabled by default.

  gpgkey is the storage path of the signature public key.

- Configuring the yum source for the HTTP server

  Create the **openEuler.repo** file in the **/etc/yum.repos.d** directory and use the repository on the HTTP server as the yum source. The content of the **openEuler.repo** file is as follows:

  ```
  [base]
  name=base
  baseurl=http://192.168.1.2/
  enabled=1
  gpgcheck=1
  gpgkey=http://192.168.1.2/RPM-GPG-KEY-openEuler
  ```

  NOTE

  192.168.1.2 is an example. Replace it with the actual IP address.

10.1.4.2 repo Priority

If there are multiple repo sources, you can set the repo priority in the .repo file. If the priority is not set, the default priority is 99. If the same RPM package exists in the sources with the same priority, the latest version is installed. **1** indicates the highest priority and **99** indicates the lowest priority. For example, set the priority of **openEuler.repo** to **2**.

```
[base]
name=base
baseurl=http://192.168.1.2/
enabled=1
priority=2
gpgcheck=1
gpgkey=http://192.168.1.2/RPM-GPG-KEY-openEuler
```

NOTE

gpgcheck indicates whether to enable the GNU private guard (GPG) to check the validity and security of sources of RPM packages. **1** indicates GPG check is enabled. **0** indicates the GPG check is disabled. If this option is not specified, the GPG check is enabled by default.

gpgkey is the storage path of the signature public key.

10.1.4.3 Related Commands of dnf

The dnf command can automatically parse the dependency between packages during installation and upgrade. The common usage method is as follows:

```
dnf <command> <packages name>
```

Common commands are as follows:

- Installation

  ```
  dnf install <packages name>
  ```

- Upgrade

  ```
  dnf update <packages name>
  ```

- Rollback

  ```
  dnf downgrade <packages name>
  ```

- Checking for update

  ```
  dnf check-update
  ```

- Uninstallation

  ```
  dnf remove <packages name>
  ```

- Query

  ```
  dnf search <packages name>
  ```

- Local installation

  ```
  dnf localinstall <absolute path to package name>
  ```

- Viewing historical records

  ```
  dnf history
  ```

- Clearing cache records

  ```
  dnf clean all
  ```

- Updating cache

  ```
  dnf makecache
  ```

10.2 Configuring the FTP Server

10.2.1 General Introduction

FTP Overview

File Transfer Protocol (FTP) is one of the earliest transmission protocols on the Internet. It is used to transfer files between the server and client. FTP allows users to access files on a remote system using a set of standard commands without logging in to the remote system. In addition, the FTP server provides the following functions:

- Subscriber classification

  By default, the FTP server classifies users into real users, guest users, and anonymous users based on the login status. The three types of users have different access permissions. Real users have complete access permissions, while anonymous users have only the permission to downloading resources.

- Command records and log file records

FTP can use the syslogd to record data, including historical commands and user transmission data (such as the transmission time and file size). Users can obtain log information from the /var/log/ directory.

- Restricting the access scope of users

  FTP can limit the work scope of a user to the home directory of the user. After a user logs in to the system through FTP, the root directory displayed by the system is the home directory of the user. This environment is called change root (chroot for short). In this way, users can access only the main directory, but not important directories such as /etc, /home, and /usr/local. This protects the system and keeps the system secure.

Port Used by the FTP Server

The FTP service requires multiple network ports. The server uses the following ports:

- Command channel. The default port number is 21.
- Data channel. The default port number is 20.

Port 21 is used to receive connection requests from the FTP client, and port 20 is used by the FTP server to proactively connect to the FTP client.

Introduction to vsftpd

FTP has a long history and uses the unencrypted transmission mode, and is therefore considered insecure. This section describes the Very Secure FTP Daemon (vsftpd), to use FTP in a more secure way.

The vsftpd is introduced to build a security-centric FTP server. The vsftpd is designed with the following features:

- The startup user of the vsftpd service is a common user who has low system permission. In addition, the vsftpd service uses chroot to change the root directory, preventing the risk of misusing system tools.
- Any vsftpd command that requires high execution permission is controlled by a special upper-layer program. The upper-layer program has low permission and does not affect the system.
- vsftpd integrates most of the extra commands (such as dir, ls, and cd) used by FTP. Generally, the system does not need to provide extra commands, which are secure for the system.

10.2.2 Using vsftpd

Installing vsftpd

To use the vsftpd service, you need to install the vsftpd software. If the yum source has been configured, run the following command as the root user to install the vsftpd service:

```
# dnf install vsftpd
```

Service Management

To start, stop, or restart the vsftpd service, run the corresponding command as the root user.

- Starting vsftpd services

  ```
  # systemctl start vsftpd
  ```

You can run the netstat command to check whether communication port 21 is enabled. If the following information is displayed, the vsftpd service has been enabled.

```
# netstat -tulnp | grep 21
tcp6       0      0 :::21                   :::*                    LISTEN
19716/vsftpd
```

NOTE

If the netstat command does not exist, run the following command to install the netstat command and then run the netstat command:

```
dnf install net-tools
```

- Stopping the vsftpd services

```
# systemctl stop vsftpd
```

- Restarting the vsftpd service

```
# systemctl restart vsftpd
```

10.2.3 Configuring vsftpd

10.2.3.1 vsftpd Configuration Files

You can modify the vsftpd configuration file to control user permissions. Table 10-1 describes the vsftpd configuration files. You can modify the configuration files as required. You can run the man command to view more parameter meanings.

Table 10-1 vsftpd configuration files

Configuration File	Description
/etc/vsftpd/vsftpd.conf	Main configuration file of the vsftpd process. The configuration format is Parameter=Parameter value. The parameter and parameter value cannot be empty. You can run the following command to view details about the vsftpd.conf file: man 5 vsftpd.conf
/etc/pam.d/vsftpd	Pluggable authentication modules (PAMs) are used for identity authentication and restrict some user operations.
/etc/vsftpd/ftpusers	List of users who are not allowed to use the vsftpd. By default, the system account is also in this file. Therefore, the system account cannot use vsftpd by default.
/etc/vsftpd/user_list	List of users who are allowed or not allowed to log in to the vsftpd server. Whether the file takes effect depends on the following parameters in the main configuration file vsftpd.conf: userlist_enable: indicates whether to enable the userlist mechanism. The value YES indicates that the userlist mechanism is enabled. In this case, the userlist_deny configuration is valid. The value NO indicates that the userlist mechanism is disabled. userlist_deny: indicates whether to forbid users in the user list to log in. YES indicates that users in the user list are forbidden to log in. NO indicates that users in the command are allowed to log in. For example, if userlist_enable is set to YES and userlist_deny is

Configuration File	Description
	set to NO, all users in the user list cannot log in.
/etc/vsftpd/chroot_list	Whether to restrict the user list in the home directory. By default, this file does not exist. You need to create it manually. It is the value of chroot_list_file in the vsftpd.conf file. The function of this parameter is determined by the following parameters in the vsftpd.conf file: • chroot_local_user: indicates whether to restrict all users to the home directory. The value YES indicates that all users are restricted to the home directory, and the value NO indicates that all users are not restricted to the home directory. • chroot_list_enable: indicates whether to enable the list of restricted users. The value YES indicates that the list is enabled, and the value NO indicates that the list is disabled. For example, if chroot_local_user is set to YES, chroot_list_enable is set to YES, and chroot_list_file is set to /etc/vsftpd/chroot_list, all users are restricted to their home directories, and users in chroot_list are not restricted.
/usr/sbin/vsftpd	Unique execution file of vsftpd.
/var/ftp/	Default root directory for anonymous users to log in. The root directory is related to the home directory of the ftp user.

10.2.3.2 Default Configuration Description

> NOTE
>
> The configuration content in this document is for reference only. You can modify the content based on the site requirements (for example, security hardening requirements).

In the openEuler system, vsftpd does not open to anonymous users by default. Run the vim command to view the main configuration file. The content is as follows:

```
# vim /etc/vsftpd/vsftpd.conf
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
listen=NO
listen_ipv6=YES
pam_service_name=vsftpd
userlist_enable=YES
```

Table 10-2 describes the parameters.

Table 10-2 Parameter description

Parameter	Description
anonymous_enable	Indicates whether to allow anonymous users to log in. YES indicates that anonymous users are allowed to log in; NO indicates that anonymous users are not allowed to log in.
local_enable	Whether to allow local users to log in. YES indicates that local users are allowed to log in. NO indicates that local users are not allowed to log in.
write_enable	Whether to allow the login user to have the write permission. YES indicates that the upload and write function is enabled, and NO indicates that the function is disabled.
local_umask	Indicates the umask value when a local user adds a profile.
dirmessage_enable	Indicates whether to display the contents that users need to pay attention to when a user accesses a directory. The options are YES (yes) and NO (no).
xferlog_enable	Indicates whether to record file upload and download operations. The options are YES (record operations) and NO (not record operations).
connect_from_port_20	Indicates whether port 20 is used for data transmission in port mode. YES indicates that port 20 is used, and NO indicates that port 20 is not used.
xferlog_std_format	Indicates whether the transfer log file is written in the standard xferlog format. The options are YES (yes) and NO (no).
listen	Indicates whether the vsftpd service is started in standalone mode. The options are YES (yes) and NO (no).
pam_service_name	Support for PAM management. The value is a service name, for example, vsftpd.
userlist_enable	Indicates whether to support account login control in the /etc/vsftpd/user_list file. The options are YES (yes) and NO (no).
tcp_wrappers	Indicates whether to support the firewall mechanism of the TCP Wrappers. The options are YES (yes) and NO (no).
listen_ipv6	Indicates whether to listen to IPv6 FTP requests. The options are YES (yes) and NO (no). listen and listen_ipv6 cannot be enabled at the same time.

10.2.3.3 Setting the Local Time

Overview

In the openEuler system, vsftpd uses the Greenwich Mean Time (GMT) time by default, which may be different from the local time. For example, the GMT time is 8 hours later than the Beijing time. You need to change the GMT time to the local time. Otherwise, the server

time and client time are inconsistent, which may cause errors during file upload and download.

Setting Method

To set the vsftpd time to the local time, perform the following steps:

Step 1 Open the vsftpd.conf file and change the value of use_localtime to **YES**. Run the following command:

```
# vim /etc/vsftpd/vsftpd.conf
```

Modify the file contents as follows:

```
use_localtime=YES
```

Step 2 Restart the vsftpd service.

```
# systemctl restart vsftpd
```

Step 3 Set the vsftpd service to start automatically upon power-on.

```
# systemctl enable vsftpd
```

----End

10.2.3.4 Configuring Welcome Information

To use the vsftpd service normally, the welcome information file must exist. To configure the welcome.txt file of the vsftp service, perform the following steps:

Step 1 Open the vsftpd.conf configuration file, add the welcome information to the file, save the file, and exit.

```
# vim /etc/vsftpd/vsftpd.conf
```

The following configuration lines need to be added:

```
banner_file=/etc/vsftpd/welcome.txt
```

Step 2 Create welcome information. Specifically, open the welcome.txt file, write the welcome information, save the file, and exit.

```
# vim /etc/vsftpd/welcome.txt
```

The following is an example:

```
Welcome to this FTP server!
```

----End

10.2.3.5 Configuring the Login Permission of a System Account

Generally, users need to restrict the login permission of some accounts. You can set the restriction as required.

Two files are used to restrict the login of system accounts. The default files are as follows:

- /etc/vsftpd/ftpusers: This file is managed by the PAM module and is determined by the settings of the /etc/pam.d/vsftpd file.

- /etc/vsftpd/user_list: This file is set by userlist_file in vsftpd.conf and is provided by vsftpd.

Both files must exist and have the same content. You can write the accounts whose UIDs are smaller than 500 to the two files by referring to the /etc/passwd. Each line indicates an account.

To restrict the login of system accounts, add the accounts to /etc/vsftpd/ftpusers and /etc/vsftpd/user_list.

Open the user_list file to view the account information in the current file. The command and output are as follows:

```
# vim /etc/vsftpd/user_list
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

10.2.4 Verifying Whether the FTP Service Is Successfully Set Up

You can use the FTP client provided by openEuler for verification. The command and output are as follows. Enter the user name (an existing user in the system) and password as prompted. If the message "Login successful" is displayed, the FTP server is successfully set up.

```
# dnf install ftp
# ftp localhost
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
220-Welcome to this FTP server!
220
Name (localhost:root): USERNAME
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
```

10.2.5 Configuring a Firewall

To open the FTP service to the Internet, you need to configure the firewall and SElinux.

```
# firewall-cmd --add-service=ftp --permanent
success
# firewall-cmd --reload
```

```
success
# setsebool -P ftpd_full_access on
```

10.2.6 File Transmission

Overview

This section describes how to transfer files after the vsftpd service is started.

Connecting to the Server

Command Format

ftp [*hostname* | *ip-address*]

hostname indicates the name of the server, and **ip-address** indicates the IP address of the server.

Requirements

Run the following command on the command-line interface (CLI) of the openEuler OS:

```
ftp ip-address
```

Enter the user name and password as prompted. If the following information is displayed after the authentication is successful, the FTP connection is successful. In this case, you have accessed the directory of the connected server.

```
ftp>
```

At this prompt, you can enter different commands to perform related operations.

- Display the current IP address of the server.

  ```
  ftp>pwd
  ```

- Display the local path. You can upload the files in this path to the corresponding location on the FTP server.

  ```
  ftp>lcd
  ```

- Exit the current window and return to the local Linux terminal.

  ```
  ftp>!
  ```

Downloading a File

Generally, the get or mget command is used to download files.

How to use get

- Function description: Transfers files from a remote host to a local host.
- Command format: **get** [*remote-file*] [*local-file*]

  remote-file indicates a remote file, and *local-file* indicates a local file.
- For example, to obtain the /usr/your/openEuler.htm file on the remote server, run the following command:

  ```
  ftp> get /usr/your/openEuler.htm
  ```

How to use mget

- Function description: Receives a batch of files from the remote host to the local host.

- Command format: **mget** [*remote-file*]

  remote-file indicates a remote file.

- For example, to obtain all files in the /usr/your/ directory on the server, run the following command:

```
ftp> cd /usr/your/
ftp> mget *.*
```

NOTE

- In this case, a message is displayed each time a file is downloaded. To block the prompt information, run the **prompt off** command before running the **mget *.*** command.
- The files are downloaded to the current directory on the Linux host. For example, if you run the ftp command in /usr/my/, all files are downloaded to /usr/my/.

Uploading a file

Generally, the put or mput command is used to upload files.

How to use put

- Function: Transfers a local file to a remote host.
- Command format: **put** [*local-file*] [*remote-file*]

  remote-file indicates a remote file, and *local-file* indicates a local file.

- For example, run the following command to transfer the local Euler.htm file to the remote host /usr/your/ and change the file name to openEuler.htm:

```
ftp> put Euler.htm /usr/your/openEuler.htm
```

How to use mput

- Function: Transfers a batch of files from the local host to a remote host.
- Command format: **mput** [*local-file*]

  local-file indicates a local file.

- For example, run the following command to upload all HTM files in the local directory to the /usr/your/ directory on the server:

```
ftp> cd /usr/your
ftp> mput *.htm
```

Deleting a File

Generally, the **delete** or **mdelete** command is used to delete a file.

How to use delete

- Function description: Deletes one or more files from the remote server.
- Command format: **delete** [*remote-file*]

  remote-file indicates a remote file.

- For example, to delete the openFile from the remote server, run the following command:

```
ftp> delete openFile
```

How to use mdelete

- Function description: Deletes files from a remote server. This function is used to delete files in batches.

- Command format: **mdelete** [*remote-file*]

  remote-file indicates a remote file.
- For example, to delete all files whose names start with **a**, run the following command:

```
ftp> mdelete a*
```

Disconnecting from the Server

Run the bye command to disconnect from the server.

```
ftp> bye
```

10.3 Configuring the Web Server

10.3.1 Apache Server

10.3.1.1 Overview

World Wide Web (Web) is one of the most commonly used Internet protocols. At present, the web server in the Unix-Like system is mainly implemented through the Apache server software. To operate dynamic websites, LAMP (Linux + Apache + MySQL + PHP) is developed. Web services can be combined with multimedia such as text, graphics, images, and audio, and support information transmission through hyperlinks.

The web server version in the openEuler system is Apache HTTP server 2.4, that is, httpd, which is an open-source web server developed by the Apache Software Foundation.

10.3.1.2 Managing httpd

Overview

You can use the systemctl tool to manage the httpd service, including starting, stopping, and restarting the service, and viewing the service status. This section describes how to manage the Apache HTTP service.

Prerequisites

- To use the Apache HTTP service, ensure that the rpm package of the httpd service has been installed in your system. The installation command is as follows:

  ```
  # dnf install httpd
  ```

  For more information about service management, see 5 Service Management.
- To start, stop, and restart the httpd service, you must have the root permission.

Starting a Service

- Run the following command to start and run the httpd service:

  ```
  # systemctl start httpd
  ```
- If you want the httpd service to automatically start when the system starts, the command and output are as follows:

```
# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service →
/usr/lib/systemd/system/httpd.service.
```

NOTE

If the running Apache HTTP server functions as a secure server, a password is required after the system is started. The password is an encrypted private SSL key.

Stopping the Service

- Run the following command to stop the httpd service:

```
# systemctl stop httpd
```

- If you want to prevent the service from automatically starting during system startup, the command and output are as follows:

```
# systemctl disable httpd
Removed /etc/systemd/system/multi-user.target.wants/httpd.service.
```

Restarting a Service

You can restart the service in any of the following ways:

- Restart the service by running the restart command:

```
# systemctl restart httpd
```

  This command stops the ongoing httpd service and restarts it immediately. This command is generally used after a service is installed or when a dynamically loaded module (such as PHP) is removed.

- Reload the configuration.

```
# systemctl reload httpd
```

  This command causes the running httpd service to reload its configuration file. Any requests that are currently being processed will be interrupted, causing the client browser to display an error message or re-render some pages.

- Re-load the configuration without affecting the activation request.

```
# apachectl graceful
```

  This command causes the running httpd service to reload its configuration file. Any requests that are currently being processed will continue to use the old configuration file.

Verifying the Service Status

Check whether the httpd service is running.

```
# systemctl is-active httpd
```

If active is displayed in the command output, the service is running.

10.3.1.3 Configuration File Description

After the httpd service is started, it reads the configuration file shown in Table 10-3 by default.

Table 10-3 Configuration file description

File	Description

File	Description
/etc/httpd/conf/httpd.conf	Main configuration files.
/etc/httpd/conf.d	Secondary directory of configuration files, which are also contained in the main configuration file. The secondary directory of a configuration file is contained in the main configuration file.

Although the default configuration can be used in most cases, you need to be familiar with some important configuration items. After the configuration file is modified, run the following command to check the syntax errors that may occur in the configuration file:

```
# apachectl configtest
```

If the following information is displayed, the syntax of the configuration file is correct:

```
Syntax OK
```

NOTE

- Before modifying the configuration file, back up the original file so that the configuration file can be quickly restored if a fault occurs.
- The modified configuration file takes effect only after the web service is restarted.

10.3.1.4 Management Module and SSL

Overview

The httpd service is a modular application that is distributed with many Dynamic Shared Objects (DSOs). DSOs can be dynamically loaded or unloaded when running if necessary. These modules are located in the /usr/lib64/httpd/modules/ directory of the server operating system. This section describes how to load and write a module.

Loading a Module

To load a special DSO module, you can use the load module indication in the configuration file. The modules provided by the independent software package have their own configuration files in the /etc/httpd/conf.modules.d directory.

For example, to load the asis DSO module, perform the following steps:

Step 1 In the /etc/httpd/conf.modules.d/00-optional.conf file, uncomment the following configuration line:

```
LoadModule asis_module modules/mod_asis.so
```

Step 2 After the loading is complete, restart the httpd service to reload the configuration file.

```
# systemctl restart httpd
```

Step 3 After the loading is complete, run the httpd -M command to check whether the asis DSO module is loaded.

```
# httpd -M | grep asis
```

If the following information is displayed, the asis DSO module is successfully loaded:

```
asis_module (shared)
```

----End

> **NOTE**
>
> **Common httpd commands**
>
> - httpd -v: views the httpd version number.
> - httpd -l: views the static modules compiled into the httpd program.
> - httpd -M: views the static modules and loaded dynamic modules that have been compiled into the httpd program.

Introduction to SSL

Secure Sockets Layer (SSL) is an encryption protocol that allows secure communication between the server and client. The Transport Layer Security (TLS) protocol ensures security and data integrity for network communication. openEuler supports Mozilla Network Security Services (NSS) as the security protocol TLS. To load the SSL, perform the following steps:

Step 1 Install the **mod_ssl** RPM package.

```
# dnf install mod_ssl
```

Step 2 After the loading is complete, restart the httpd service to reload the configuration file.

```
# systemctl restart httpd
```

Step 3 After the loading is complete, run the **httpd -M** command to check whether the SSL is loaded.

```
# httpd -M | grep ssl
```

If the following information is displayed, the SSL has been loaded successfully.

```
ssl_module (shared)
```

----End

10.3.1.5 Verifying Whether the Web Service Is Successfully Set Up

After the web server is set up, perform the following operations to check whether the web server is set up successfully:

Step 1 Run the following command to check the IP address of the server:

```
# ifconfig
```

If the following information is displayed, the IP address of the server is 192.168.1.60.

```
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.1.60  netmask 255.255.255.0  broadcast 192.168.1.255
inet6 fe80::5054:ff:fe95:499f  prefixlen 64  scopeid 0x20<link>
ether 52:54:00:95:49:9f  txqueuelen 1000  (Ethernet)
RX packets 150713207  bytes 49333673733 (45.9 GiB)
RX errors 0  dropped 43  overruns 0  frame 0
TX packets 2246438  bytes 203186675 (193.7 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
ether 52:54:00:7d:80:9e  txqueuelen 1000  (Ethernet)
RX packets 149937274  bytes 44652889185 (41.5 GiB)
RX errors 0  dropped 1102561  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
inet 127.0.0.1  netmask 255.0.0.0
inet6 ::1  prefixlen 128  scopeid 0x10<host>
loop  txqueuelen 1000  (Local Loopback)
RX packets 37096  bytes 3447369 (3.2 MiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 37096  bytes 3447369 (3.2 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2 Configure the firewall.

```
# firewall-cmd --add-service=http --permanent
success
# firewall-cmd --reload
success
```

Step 3 Verify whether the web server is successfully set up. You can select the Linux or Windows operating system for verification.

- Using the Linux OS

  Run the following command to check whether the web page can be accessed. If the service is successfully set up, the web page can be accessed.

  ```
  curl https://192.168.1.60
  ```

  Run the following command to check whether the command output is 0. If the command output is 0, the httpd server is successfully set up.

  ```
  echo $?
  ```

- Using the Windows OS

  Open the browser and enter the following address in the address box. If the web page can be accessed, the httpd server is successfully set up.

  https://*192.168.1.60*

  If the port number is changed, enter the address in the following format:

  https://*192.168.1.60: port number*

----End

10.3.2 Nginx Server

10.3.2.1 Overview

Nginx is a lightweight web server which also acts as a reverse proxy server and email (IMAP/POP3) proxy server. It features low memory usage and strong concurrency capability. Nginx supports FastCGI, SSL, virtual hosts, URL rewrite, Gzip, and extension of many third-party modules.

10.3.2.2 Installing Nginx

Step 1 Configure the local yum source. For details, see 10.1 Configuring the Repo Server.

Step 2 Clear the cache.

```
#dnf clean all
```

Step 3 Create a cache.

```
#dnf makecache
```

Step 4 Install the MariaDB server.

```
#dnf install nginx
```

Step 5 Check the installed RPM package.

```
dnf list all | grep nginx
```

----End

10.3.2.3 Managing Nginx

Overview

You can use the systemctl tool to manage the Nginx service, including starting, stopping, and restarting the service, and viewing the service status. This section describes how to manage the Nginx service.

Prerequisites

- Ensure that the Nginx service has been installed. If not, install it by referring to 10.3.2.2 Installing Nginx.

  For more information about service management, see 5 Service Management.
- To start, stop, and restart the httpd service, you must have the **root** permission.

Starting a Service

- Run the following command to start and run the Nginx service:

  ```
  # systemctl start nginx
  ```
- If you want the Nginx service to automatically start when the system starts, the command and output are as follows:

  ```
  # systemctl enable nginx
  Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service →
  /usr/lib/systemd/system/nginx.service.
  ```

  NOTE

  If the running Nginx server functions as a secure server, a password is required after the system is started. The password is an encrypted private SSL key.

Stopping the Service

- Run the following command to stop the httpd service:

  ```
  # systemctl stop nginx
  ```

- If you want to prevent the service from automatically starting during system startup, the command and output are as follows:

```
# systemctl disable nginx
Removed /etc/systemd/system/multi-user.target.wants/nginx.service.
```

Restarting a Service

You can restart the service in any of the following ways:

- Restart the service.

```
# systemctl restart nginx
```

  This command stops the ongoing Nginx service and restarts it immediately. This command is generally used after a service is installed or when a dynamically loaded module (such as PHP) is removed.

- Reload the configuration.

```
# systemctl reload httpd
```

  This command causes the running Nginx service to reload its configuration file. Any requests that are currently being processed will be interrupted, causing the client browser to display an error message or re-render some pages.

- Smoothly restart Nginx.

```
# kill -HUP PID
```

  This command causes the running Nginx service to reload its configuration file. Any requests that are currently being processed will continue to use the old configuration file.

Verifying the Service Status

Check whether the httpd service is running.

```
# systemctl is-active nginx
```

If **active** is displayed in the command output, the service is running.

10.3.2.4 Configuration File Description

After the Nginx service is started, it reads the configuration file shown in Table 10-4 by default.

Table 10-4 Configuration file description

File	Description
/etc/nginx/nginx.conf	Main configuration files.
/etc/nginx/conf.d	Secondary directory of configuration files, which are also contained in the main configuration file. The secondary directory of a configuration file is contained in the main configuration file.

Although the default configuration can be used in most cases, you need to be familiar with some important configuration items. After the configuration file is modified, run the following command to check the syntax errors that may occur in the configuration file:

```
# /usr/sbin/nginx -t
```

If the command output contains **syntax is ok**, the syntax of the configuration file is correct.

NOTE

- Before modifying the configuration file, back up the original file so that the configuration file can be quickly restored if a fault occurs.
- The modified configuration file takes effect only after the web service is restarted.

10.3.2.5 Management Modules

Overview

The Nginx service is a modular application that is distributed with many Dynamic Shared Objects (DSOs). DSOs can be dynamically loaded or unloaded when running if necessary. These modules are located in the **/usr/lib64/nginx/modules/** directory of the server operating system. This section describes how to load and write a module.

Loading a Module

To load a special DSO module, you can use the load module indication in the configuration file. Generally, the modules provided by independent software packages have their own configuration files in the **/usr/share/nginx/modules** directory.

The DSO is automatically loaded when the **dnf install nginx** command is used to install the Nginx in the openEuler operating system.

10.3.2.6 Verifying Whether the Web Service Is Successfully Set Up

After the web server is set up, perform the following operations to check whether the web server is set up successfully:

Step 1 Run the following command to check the IP address of the server:

```
# ifconfig
```

If the following information is displayed, the IP address of the server is **192.168.1.60**.

```
enp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
inet 192.168.1.60  netmask 255.255.255.0  broadcast 192.168.1.255
inet6 fe80::5054:ff:fe95:499f  prefixlen 64  scopeid 0x20<link>
ether 52:54:00:95:49:9f  txqueuelen 1000  (Ethernet)
RX packets 150713207  bytes 49333673733 (45.9 GiB)
RX errors 0  dropped 43  overruns 0  frame 0
TX packets 2246438  bytes 203186675 (193.7 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp4s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
ether 52:54:00:7d:80:9e  txqueuelen 1000  (Ethernet)
RX packets 149937274  bytes 44652889185 (41.5 GiB)
RX errors 0  dropped 1102561  overruns 0  frame 0
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
inet 127.0.0.1  netmask 255.0.0.0
inet6 ::1  prefixlen 128  scopeid 0x10<host>
loop  txqueuelen 1000  (Local Loopback)
RX packets 37096  bytes 3447369 (3.2 MiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 37096  bytes 3447369 (3.2 MiB)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 2 Configure the firewall.

```
# firewall-cmd --add-service=http --permanent
success
# firewall-cmd --reload
success
```

Step 3 Verify whether the web server is successfully set up. You can select the Linux or Windows operating system for verification.

- Using the Linux OS

  Run the following command to check whether the web page can be accessed. If the service is successfully set up, the web page can be accessed.

  ```
  curl http://192.168.1.60
  ```

  Run the following command to check whether the command output is **0**. If the command output is **0**, the Nginx server is successfully set up.

  ```
  echo $?
  ```

- Using the Windows OS

  Open the browser and enter the following address in the address box. If the web page can be accessed, the Nginx server is successfully set up.

  http://*192.168.1.60*

  If the port number is changed, enter the address in the following format:

  http://*192.168.1.60: port number*

----End

10.4 Setting Up the Database Server

10.4.1 PostgreSQL Server

10.4.1.1 Software Description

Overview

Figure 10-4 shows the PostgreSQL architecture and Table 10-5 describes the main processes.

Figure 10-4 PostgreSQL architecture

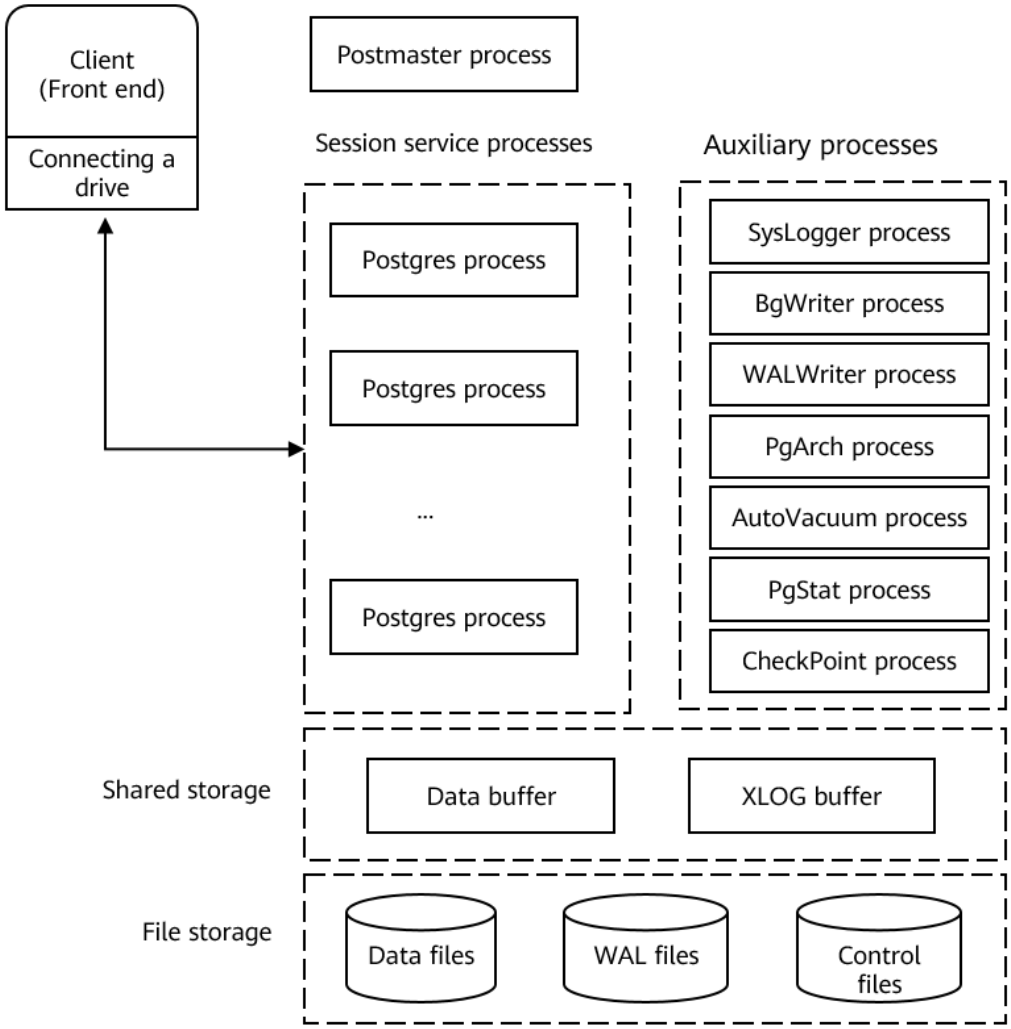


Table 10-5 Main processes in PostgreSQL

Process Type	Process Name	Description
Main process	Postmaster	Postmaster process controls all database instances in general and is responsible for starting and stopping database instances.
Resident pro	Postgres (resident process)	This process manages backend resident processes and is also called postmaster. By default, this process listens Unix domain sockets and the 5432 port of TCP/IP and waits for the front end to process the connections. You can change the listening port

Process Type	Process Name	Description
cess		number in the **postgresql.conf** file of PostgreSQL.
Subprocess	Postgres (subprocess)	The subprocess determines whether to allow the connection according to the security policy defined by the **pg_hba.conf** file. According to the security policy, the subprocess rejects certain IP addresses and networks, allows only certain users to connect to the databases, or allows only certain databases to be connected. Postgres receives the query from the front end, searches the database, and returns the results. Sometimes, it also updates the database. The updated data is recorded in transaction logs (WAL logs for PostgreSQL). This method is used when the system is powered off, the server breaks down, or the server is restarted. In addition, the logs can also be used for data recovery in other scenarios. In PostgreSQL 9.0 or later, WAL logs can be transferred to other PostgreSQL systems to replicate database in real-time.
Auxiliary processes	SysLogger (system log)	The main process starts the Syslogger auxiliary process only when **logging_collection** in the **Postgres.conf** file is set to **on**.
	BgWriter (background write)	This process writes dirty pages from the shared memory to the drive. The purpose is to improve the performance of inserting, updating, and deleting data.
	WALWriter (write-ahead log)	This process writes modification operations into drives before data is modified so that the data does not need to be persisted into files in subsequent real-time data updates.
	PgArch (archive)	write-ahead logs (WALs) are recycled. The PgArch process backs up WALs before archiving them. After the entire database is backed up, the Point in Time Recovery (PITR) technology can be used to archive WALs. The database can be restored to any point after the full backup by using the full backup data and the subsequently archived WALs.
	AutoVacuum (automatic cleanup)	In the PostgreSQL database, after a DELETE operation is performed on a table, old data is not immediately deleted. When new data is added, the system creates a data row instead of overwriting the old data. The old data is only marked as deleted and will be cleared only when no other concurrent transactions are reading the data. In this case, the data is cleared by the AutoVacuum process.
	PgStat (statistics collection)	This process collects data statistics. It is used to estimate the cost during query optimization, including the number of insertions update, and deletion operations performed on a table or index, the number of drive block read and write operations, and the number of row read operations. **pg_statistic** stores the

Process Type	Process Name	Description
		information collected by the PgStat.
	CheckPoint (checkpoint)	A checkpoint is a transaction sequence point set by the system. It is used to ensure that log information before a checkpoint written into the drives.

10.4.1.2 Configuring the Environment

NOTE

The following environment configuration is for reference only. Configure the environment based on the site requirements.

10.4.1.2.1 Disabling the Firewall and Automatic Startup

NOTE

It is recommended that firewall be disabled in the test environment to prevent network impact. Configure the firewall based on actual requirements.

Step 1 Stop the firewall service.

```
#systemctl stop firewalld
```

Step 2 Disable the firewall service.

```
#systemctl disable firewalld
```

NOTE

The automatic startup is automatically disabled as the firewall is disabled.

----End

10.4.1.2.2 Disabling SELinux

Step 1 Modify the configuration file.

```
#sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

----End

10.4.1.2.3 Creating a User Group and a User

NOTE

In the server environment, independent users are assigned to each process to implement permission isolation for security purposes. The user group and user are created for the OS, not for the database.

Step 1 Create a PostgreSQL user or user group.

```
#groupadd  postgres
```

```
#useradd  -g postgres postgres
```

Step 2 Set the postgres user password. (Enter the password twice for confirmation.)

```
#passwd postgres
```

----End

10.4.1.2.4 Creating Data Drives

NOTE

- When testing the ultimate performance, you are advised to attach NVMe SSDs with better I/O performance to create PostgreSQL test instances to avoid the impact of disk I/O on the performance test result. This section uses NVMe SSDs as an example. For details, see Step 1 to Step 4.
- In a non-performance test, run the following command to create a data directory. Then skip this section.

  #mkdir /data

Step 1 Create a file system (xfs is used as an example. Create the file system based on the site requirements.). If a file system has been created for a disk, an error will be reported when you run this command. You can use the **-f** parameter to forcibly create a file system.

```
#mkfs.xfs /dev/nvme0n1
```

Step 2 Create a data directory.

```
#mkdir /data
```

Step 3 Mount disks.

```
#mount -o noatime,nobarrier /dev/nvme0n1 /data
```

----End

10.4.1.2.5 Data Directory Authorization

Step 1 Modify the directory permission.

```
#chown -R postgres:postgres /data/
```

----End

10.4.1.3 Installing, Running, and Uninstalling PostgreSQL

10.4.1.3.1 Installing PostgreSQL

Step 1 Configure the local yum source. For details, see 10.1 Configuring the Repo Server.

Step 2 Clear the cache.

```
#dnf clean all
```

Step 3 Create a cache.

```
#dnf makecache
```

Step 4 Install the PostgreSQL server.

```
#dnf install postgresql-server
```

Step 5 Check the installed RPM package.

```
#rpm -qa | grep postgresql
```

----End

10.4.1.3.2 Running PostgreSQL

10.4.1.3.2.1 Initializing the Database

> **NOTICE**
>
> Perform this step as the postgres user.

Step 1 Switch to the created PostgreSQL user.

```
#su - postgres
```

Step 2 Initialize the database. In the command, **/usr/bin** is the directory where the **initdb** command is located.

```
$/usr/bin/initdb -D /data/
```

----End

10.4.1.3.2.2 Starting the Database

Step 1 Enable the PostgreSQL database.

```
$/usr/bin/pg_ctl -D /data/ -l /data/logfile start
```

Step 2 Check whether the PostgreSQL database process is started properly.

```
$ps -ef | grep postgres
```

If the following information is displayed, the PostgreSQL processes have been started.

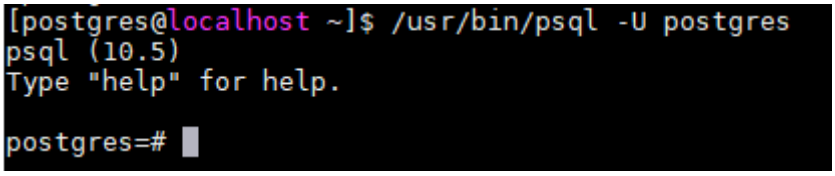
```
[root@localhost ~]# ps -ef | grep postgres
root       11232    2230  0 10:09 pts/0    00:00:00 su - postgres
postgres   11233   11232  0 10:09 pts/0    00:00:00 -bash
postgres   12319       1  0 11:22 pts/0    00:00:00 /usr/bin/postgres -D /data
postgres   12321   12319  0 11:22 ?        00:00:00 postgres: checkpointer process
postgres   12322   12319  0 11:22 ?        00:00:00 postgres: writer process
postgres   12323   12319  0 11:22 ?        00:00:00 postgres: wal writer process
postgres   12324   12319  0 11:22 ?        00:00:00 postgres: autovacuum launcher process
postgres   12325   12319  0 11:22 ?        00:00:00 postgres: stats collector process
postgres   12326   12319  0 11:22 ?        00:00:00 postgres: bgworker: logical replication launcher
root       12348    7094  0 11:29 pts/1    00:00:00 grep postgres
[root@localhost ~]#
```

----End

10.4.1.3.2.3 Logging In to the Database

Step 1 Log in to the database.

```
$/usr/bin/psql -U postgres
```



NOTE

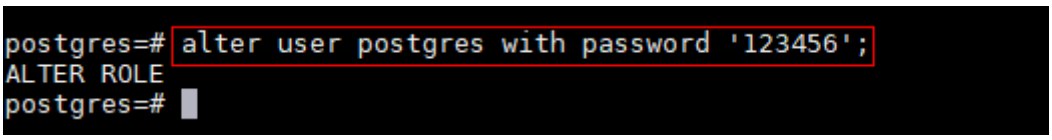
You do not need to enter a password when logging in to the database for the first time.

----End

10.4.1.3.2.4 Configuring the Database Accounts and Passwords

Step 1 After login, set the postgres user password.

```
postgres=#alter user postgres with password '123456';
```



----End

10.4.1.3.2.5 Exiting the Database

Step 1 Run **\q** to exit from the database.

```
postgres=#\q
```

----End

10.4.1.3.2.6 Stopping the Database

Step 1 Stop the PostgreSQL database.

```
$/usr/bin/pg_ctl -D /data/ -l /data/logfile stop
```

----End

10.4.1.3.3 Uninstalling PostgreSQL

Step 1 Stop the database as the postgres user.

```
$/usr/bin/pg_ctl -D /data/ -l /data/logfile stop
```

Step 2 Run the **dnf remove postgresql-server** command as the user **root** to uninstall the PostgreSQL database.

```
#dnf remove postgresql-server
```

----End

10.4.1.4 Managing Database Roles

10.4.1.4.1 Creating a Role

You can use the **CREATE ROLE** statement or **createuser** command to create a role. The **createuser** command encapsulates the **CREATE ROLE** statement and needs to be executed on the shell GUI instead of the database GUI.

```
CREATE ROLE rolename [ [ WITH ] option [ ... ] ];
createuser rolename
```

In the preceding information:

- **rolename**: indicates a role name.
- Parameters of the *option* are as follows:
  - **SUPERUSER | NOSUPERUSER**: determines whether a new role is a superuser. If this parameter is not specified, the default value **NOSUPERUSER** is used, indicating that the role is not a superuser.
  - **CREATEDB | NOCREATEDB**: specifies whether a role can create a database. If this parameter is not specified, the default value **NOCREATEDB** is used, indicating that the role cannot create a database.
  - **CREATEROLE | NOCREATEROLE**: determines whether a role can create roles. If this parameter is not specified, the default value **NOCREATEROLE** is used, indicating that the role cannot create roles.
  - **INHERIT | NOINHERIT**: determines whether a role inherits the other roles' permissions in the group to which the role belongs. A role with the INHERIT attribute can automatically use any permissions that have been assigned to its direct or indirect group. If this parameter is not specified, the default value **INHERIT** is used.
  - **LOGIN | NOLOGIN**: determines whether a role can log in. A role with the LOGIN attribute can be considered as a user. A role without this attribute can be used to manage database permissions but is not a user. If this attribute is not specified, the default value **NOLOGIN** is used. However, if **CREATE USER** instead of **CREATE ROLE** is used to create a role, the LOGIN attribute is used by default.
  - **[ENCRYPTED | UNENCRYPTED] PASSWORD'password'**: password of a role. The password is valid only for roles with the LOGIN attribute. **ENCRYPTED | UNENCRYPTED**: determines whether to encrypt the password. If this parameter is not specified, the value **ENCRYPTED** is used, that is, the password is encrypted.
  - **VALID UNTIL'timestamp'**: specifies the timestamp when the password of a role expires. If this parameter is not specified, the password is permanently valid.
  - **IN ROLE rolename1**: lists one or more existing roles. The new role *rolename* will be added to and become a member of **rolename1**.
  - **ROLE rolename2**: lists one or more existing roles. These roles will be automatically added as members of the new role *rolename*. That is, the new role is a user group.

To run this command, you must have the CREATEROLE permission or is the database superuser.

Example

#Create a role **roleexample1** who can log in.

```
postgres=# CREATE ROLE roleexample1 LOGIN;
```

#Create a role **roleexample2** with the password **123456**.

```
postgres=# CREATE ROLE roleexample2 WITH LOGIN PASSWORD '123456';
```

#Create a role named **roleexample3**.

```
[postgres@localhost ~]$ createuser roleexample3
```

10.4.1.4.2 Viewing Roles

You can run the **SELECT** statement or the PostgreSQL meta-command **\du** to view the role.

```
SELECT rolename FROM pg_roles;
\du
```

In the preceding command, *rolename* indicates the role name.

Example

#View the **roleexample1** role.

```
postgres=# SELECT roleexample1 from pg_roles;
```

#View the existing roles.

```
postgres=# \du
```

10.4.1.4.3 Modifying a Role

Modifying a Username

Use the **ALTER ROLE** statement to modify an existing role name.

```
ALTER ROLE oldrolername RENAME TO newrolename;
```

In the preceding information:

- *oldrolername*: original role name.
- *newrolename*: new role name.

Example of Modifying a User

#Change the role name **roleexample1** to **roleexapme2**.

```
# ALTER ROLE roleexample1 RENAME TO roleexample2;
```

Modifying a User Password

Use the **ALTER ROLE** statement to modify the login password of a role.

```
ALTER ROLE rolename PASSWORD 'password'
```

In the preceding information:

- *rolename*: indicates a role name.
- *password*: password.

Example of Modifying the Password of a Role

#Modify the password of **roleexample1** to **456789**.

```
# ALTER ROLE roleexample1 WITH PASSWORD '456789';
```

10.4.1.4.4 Deleting a Role

You can use the **DROP ROLE** statement or **dropuser** command to delete a role. The **dropuser** command encapsulates the **DROP ROLE** statement and needs to be executed on the shell GUI instead of the database GUI.

```
DROP ROLE rolename;
dropuser rolename
```

In the preceding command, *rolename* indicates the role name.

Example

#Delete the **userexample1** role.

```
postgres=# DROP ROLE userexample1;
```

#Delete the **userexample2** role.

```
[postgres@localhost ~]$ dropuser userexample2
```

10.4.1.4.5 Role Permissions

You can use the **GRANT** statement to grant permissions to a role.

Grant the table operation permission to a role.

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | REFERENCES | TRIGGER } [,...] | ALL
[ PRIVILEGES ] } ON [ TABLE ] tablename [, ...] TO { rolename | GROUP groupname | PUBLIC }
[, ...] [ WITH GRANT OPTION ]
```

Grant the sequence operation permission to a role.

```
GRANT { { USAGE | SELECT | UPDATE } [,...] | ALL [ PRIVILEGES ] } ON SEQUENCE sequencename
[, ...] TO { rolename | GROUP groupname | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Grant the database operation permission to a role.

```
GRANT { { CREATE | CONNECT | TEMPORARY | TEMP } [,...] | ALL [ PRIVILEGES ] } ON DATABASE
databasename [, ...] TO { rolename | GROUP groupname | PUBLIC } [, ...] [ WITH GRANT
OPTION ]
```

Grant the function operation permission to a role.

```
GRANT { EXECUTE | ALL [ PRIVILEGES ] } ON FUNCTION funcname ( [ [ argmode ] [ argname ]
argtype [, ...] ] ) [, ...] TO { rolename | GROUP groupname | PUBLIC } [, ...] [ WITH
GRANT OPTION ]
```

Grant the operation permission of the procedural language to a role.

```
GRANT { USAGE | ALL [ PRIVILEGES ] } ON LANGUAGE langname [, ...] TO { rolename | GROUP
groupname | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Grant the schema operation permission to a role.

```
GRANT { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] } ON SCHEMA schemaname [, ...]
TO { rolename | GROUP groupname | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Grant the tablespace operation permission to a role.

```
GRANT { CREATE | ALL [ PRIVILEGES ] } ON TABLESPACE tablespacename [, ...] TO { rolename
| GROUP groupname | PUBLIC } [, ...] [ WITH GRANT OPTION ]
```

Assign the member relationship of rolename1 to rolename2.

```
GRANT rolename1 [, ...] TO rolename2 [, ...] [ WITH ADMIN OPTION ]
```

In the preceding information:

- **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **REFERENCES**, **TRIGGER**, **USAGE**, **CREATE**, **CONNECT**, **TEMPORARY**, **TEMP**, **EXECUTE**, and **ALL [***PRIVILEGES***]** indicate user operation permissions. **ALL [***PRIVILEGES***]** indicates all permissions, the *PRIVILEGES* keyword is optional in PostgreSQL, but it is required in strict SQL statements.
- **ON** clause: specifies the object on which the permission is granted.
- **tablename**: table name.
- **TO** clause: specifies the role to which the permission is granted.
- **rolename**, **rolename1**, and **rolename2**: role names.
- **groupname**: name of a role group.
- **PUBLIC**: indicates that the permission is granted to all roles, including users who may be created later.
- **WITH GRANT OPTION**: indicates that the recipient of a permission can grant the permission to others. This option cannot be assigned to PUBLIC.
- **sequencename**: sequence name.
- **databasename**: database name.
- **funcname ([[argmode] [argname] argtype [, ...]])**: function name and its parameters.
- **langname**: procedural language name.
- **schemaname**: schema name.
- **tablespacename**: tablespace name.
- **WITH ADMIN OPTION**: A member can assign the member relationship of a role to other roles and cancel the member relationship of other roles.

Example

#Grant the CREATE permission on database1 to userexample.

```
# GRANT CREATE ON DATABASE database1 TO userexample;
```

#Grant all permissions on table1 to all users.

```
# GRANT ALL PRIVILEGES ON TABLE table1 TO PUBLIC;
```

10.4.1.4.6 Deleting User Permissions

You can use the **REVOKE** statement to revoke the permissions previously granted to one or more roles.

Revoke the table operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { { SELECT | INSERT | UPDATE | DELETE | REFERENCES | TRIGGER }
[,...] | ALL [ PRIVILEGES ] } ON [ TABLE ] tablename [, ...] FROM { rolename | GROUP
groupname | PUBLIC } [, ...]
```

Revoke the sequence operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { { USAGE | SELECT | UPDATE } [,...] | ALL [ PRIVILEGES ] }
ON SEQUENCE sequencename [, ...] FROM { rolename | GROUP groupname | PUBLIC } [, ...]
[ CASCADE | RESTRICT ]
```

Revoke the database operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { { CREATE | CONNECT | TEMPORARY | TEMP } [,...] | ALL
[ PRIVILEGES ] } ON DATABASE databasename [, ...] FROM { rolename | GROUP groupname
| PUBLIC } [, ...] [ CASCADE | RESTRICT ]
```

Revoke the function operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { EXECUTE | ALL [ PRIVILEGES ] } ON FUNCTION funcname
( [ [ argmode ] [ argname ] argtype [, ...] ] ) [, ...] FROM { rolename | GROUP groupname
| PUBLIC } [, ...] [ CASCADE | RESTRICT ]
```

Revoke the procedural language operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { USAGE | ALL [ PRIVILEGES ] } ON LANGUAGE langname [, ...]
FROM { rolename | GROUP groupname | PUBLIC } [, ...] [ CASCADE | RESTRICT ]
```

Revoke the schema operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { { CREATE | USAGE } [,...] | ALL [ PRIVILEGES ] } ON SCHEMA
schemaname [, ...] FROM { rolename | GROUP groupname | PUBLIC } [, ...] [ CASCADE |
RESTRICT ]
```

Revoke the tablespace operation permission from a role.

```
REVOKE [ GRANT OPTION FOR ] { CREATE | ALL [ PRIVILEGES ] } ON TABLESPACE tablespacename
[, ...] FROM { rolename | GROUP groupname | PUBLIC } [, ...] [ CASCADE | RESTRICT ]
```

Revoke the member relationship of rolename1 from rolename2.

```
REVOKE [ ADMIN OPTION FOR ] rolename1 [, ...] FROM rolename2 [, ...] [ CASCADE | RESTRICT ]
```

In the preceding information:

- **GRANT OPTION FOR**: The permission cannot be granted to others, but permission itself is not revoked.
- **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **REFERENCES**, **TRIGGER**, **USAGE**, **CREATE**, **CONNECT**, **TEMPORARY**, **TEMP**, **EXECUTE**, and **ALL [***PRIVILEGES***]** indicate user operation permissions. **ALL [***PRIVILEGES***]** indicates all permissions, the *PRIVILEGES* keyword is optional in PostgreSQL, but it is required in strict SQL statements.
- **ON** clause: specifies the object on which the permission is revoked.
- *tablename*: table name.
- **FROM** clause: specifies the role whose permission is revoked.
- *rolename*, *rolename1*, and *rolename2*: role names.
- *groupname*: name of a role group.

- **PUBLIC**: revokes the implicitly defined groups that have all roles. However, this does not mean that all roles lose the permissions. The permissions directly obtained and the permissions obtained through a group are still valid.
- *sequencename*: sequence name.
- **CASCADE**: revokes all dependent permissions.
- **RESTRICT**: does not revoke all dependent permissions.
- *databasename*: database name.
- **funcname (***[[argmode] [argname] argtype [, ...]]***)**: function name and its parameters.
- *langname*: procedural language name.
- *schemaname*: schema name.
- *tablespacename*: tablespace name.
- **ADMIN OPTION FOR**: The transferred authorization is not automatically revoked.

Example

#Grant the CREATE permission on database1 to userexample.

```
# GRANT CREATE ON DATABASE database1 TO userexample;
```

#Grant all permissions on table1 to all users.

```
# GRANT ALL PRIVILEGES ON TABLE table1 TO PUBLIC;
```

10.4.1.5 Managing Databases

10.4.1.5.1 Creating a Database

You can use the **CREATE DATABASE** statement or the **createdb** command to create a role. The **createdb** command encapsulates the **CREATE DATABASE** statement and needs to be executed on the shell GUI instead of the database GUI.

```
CREATE DATABASE databasename;
createdb databasename
```

In the preceding command, **databasename** indicates the database name.

To use this command, you must have the CREATEDB permission.

Example

Create a database named **database1**.

```
postgres=# CREATE DATABASE database1;
```

10.4.1.5.2 Selecting a Database

Use the **\c** statement to select a database.

```
\c databasename;
```

In the preceding command, **databasename** indicates the database name.

Example

#Select the **databaseexample** database.

```
# \c databaseexample;
```

10.4.1.5.3 Viewing a Database

Use the **\l** statement to view the database.

```
\l;
```

Example

#View all databases.

```
# \l;
```

10.4.1.5.4 Deleting a Database

You can run the **DROP DATABASE** statement or **dropdb** command to delete a database. The **dropdb** command encapsulates the **DROP DATABASE** statement and needs to be executed on the shell GUI instead of the database GUI.

CAUTION

Exercise caution when deleting a database. Once a database is deleted, all tables and data in the database will be deleted.

```
DROP DATABASE databasename;
dropdb databasename
```

In the preceding command, **databasename** indicates the database name.

The **DROP DATABASE** statement deletes the system directory items of the database and the file directories that contain data.

DROP DATABASE can be executed only by the super administrator or database owner.

Example

#Delete the **databaseexample** database.

```
# DROP DATABASE databaseexample;
```

10.4.1.5.5 Backing Up a Database

Run the **pg_dump** command to back up the database and dump the database to a script file or another archive file.

```
pg_dump [option]... [databasename] > outfile
```

In the preceding information:

- *databasename*: database name. If this parameter is not specified, the environment variable **PGDATABASE** is used. If that environment variable is not specified, use the username that initiates the connection.
- *outfile*: database backup file.
- *option*: parameter option of the **pg_dump** command. Multiple parameters can be separated by spaces. The common parameters of the **pg_dump** command are as follows:

  - **-f** *filename*, **--file=***filename*: specified output file. If this parameter is ignored, the standard output is used.
  - **-d, --dbname=***databasename*: database to be dumped.
  - **-h, --host=***hostname*: specifies the hostname.
  - **-p, --port=***portnumber*: port number.
  - **-U, --username=***username*: username of the connection.
  - **-W, --password**: forces PostgreSQL to prompt for a password before connecting to a database.

Example

#Back up the database1 database of user **postgres** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1.sql** file.

```
$ pg_dump -h 192.168.202.144 -p 3306 -U postgres -W database1 > db1.sql
```

10.4.1.5.6 Restoring a Database

Run the **psql** command to restore the database.

```
psql [option]... [databasename [username]] < infile
```

In the preceding information:

- *databasename*: database name. If this parameter is not specified, the environment variable **PGDATABASE** is used. If that environment variable is not specified, use the username that initiates the connection.
- *username*: name of a user.
- *infile*: **outfile** parameter in the **pg_dump** command.
- *option*: parameter option of the **psql** command. Multiple parameters can be separated by spaces. The common parameters of the **psql** command are as follows:
  - **-f** *filename*, **--file=***filename*: specified output file. If this parameter is ignored, the standard output is used.
  - **-d, --dbname=***databasename*: database to be dumped.
  - **-h, --host=***hostname*: specifies the hostname.
  - **-p, --port=***portnumber*: port number.
  - **-U, --username=***username*: username of the connection.
  - **-W, --password**: forces PostgreSQL to prompt for a password before connecting to a database.

The **psql** command cannot be used to automatically create the **databasename** database. Therefore, you need to create the **databasename** database before running the **psql** command to restore the database.

Example

#Import the **db1.sql** script file to the newdb database of the postgres user on the host **192.168.202.144** through port **3306**.

```
$ createdb newdb
$ psql -h 192.168.202.144 -p 3306 -U postgres -W -d newdb < db1.sql
```

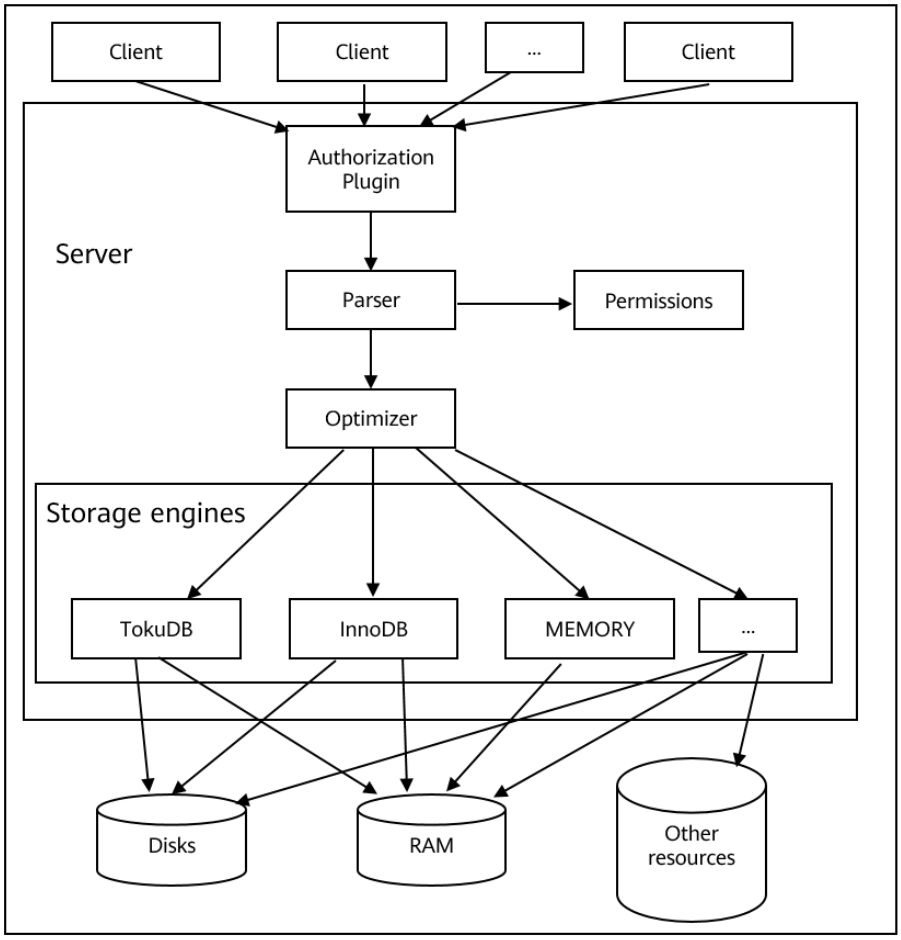
10.4.2 MariaDB Server

10.4.2.1 Software Description

The MariaDB database management system is a branch of MySQL and is maintained by the open-source community. The MariaDB database management system uses the General Public License (GPL). MariaDB is designed to be fully compatible with MySQL, including APIs and command lines, so that it can easily replace MySQL. MariaDB also provides many new features.

Figure 10-5 shows the MariaDB architecture.

Figure 10-5 MariaDB logical architecture



When MariaDB receives a SQL statement, the execution process is as follows:

1. When a client connects to MariaDB, the hostname, username, and password of the client are authenticated. The authentication function can be implemented as a plug-in.

2. If the login is successful, the client sends SQL commands to the server. The parser parses the SQL statements.
3. The server checks whether the client has the permission to obtain the required resources.
4. If the query has been stored in the query cache, the result is returned immediately.
5. The optimizer will find the fastest execution policy or plan. That is, the optimizer can determine which tables will be read, which indexes will be accessed, and which temporary tables will be used. A good policy can reduce a large number of disk access and sorting operations.
6. Storage engines read and write data and index files. Caches are used to accelerate these operations. Other features such as transactions and foreign keys are processed at the storage engine layer.

Storage engines manage and control data at the physical layer. They manage data files, data, indexes, and caches, making data management and reading more efficient. Each table has a .frm file that contains table definitions.

Each storage engine manages and stores data in different ways, and supports different features and performance. For example:

- MyISAM: suitable for environments with more reads and fewer writes. It does not support transactions and supports full-text indexes.
- noDB: supports transactions, row locks, and foreign keys.
- MEMORY: stores data in the memory.
- CSV: stores data in CSV format.

10.4.2.2 Configuring the Environment

NOTE

The following environment configuration is for reference only. Configure the environment based on the site requirements.

10.4.2.2.1 Disabling the Firewall and Automatic Startup

NOTE

It is recommended that firewall be disabled in the test environment to prevent network impact. Configure the firewall based on actual requirements.

Step 1 Stop the firewall service.

```
#systemctl stop firewalld
```

Step 2 Disable the firewall service.

```
#systemctl disable firewalld
```

NOTE

The automatic startup is automatically disabled as the firewall is disabled.

----End

10.4.2.2.2 Disabling SELinux

Step 1 Modify the configuration file.

```
#sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

----End

10.4.2.2.3 Creating a User Group and a User

NOTE

In the server environment, independent users are assigned to each process to implement permission isolation for security purposes. The user group and user are created for the OS, not for the database.

Step 1 Create a MySQL user or user group.

```
#groupadd mysql
#useradd -g mysql mysql
```

Step 2 Set the user password.

```
#passwd mysql
```

Enter the password twice for confirmation.

----End

10.4.2.2.4 Creating Data Drives

NOTE

- If a performance test needs to be performed, an independent drive is required for the data directory. You need to format and mount the drive. For details, see Method 1 or Method 2.
- In a non-performance test, run the following command to create a data directory. Then skip this section.

  #mkdir /data

Method 1: Using fdisk for Drive Management

Step 1 Create a partition, for example, **/dev/sdb**.

```
#fdisk /dev/sdb
```

Step 2 Enter **n** and press **Enter**.

Step 3 Enter **p** and press **Enter**.

Step 4 Enter **1** and press **Enter**.

Step 5 Retain the default settings and press **Enter**.

Step 6 Retain the default settings and press **Enter**.

Step 7 Enter **w** and press **Enter**.

Step 8 Create a file system, for example, **xfs**.

```
#mkfs.xfs /dev/sdb1
```

Step 9 Mount the partition to **/data** for the OS.

```
#mkdir /data
#mount /dev/sdb1 /data
```

Step 10 Run the **vi /etc/fstab** command and edit the **/etc/fstab** file to enable the data drive to be automatically mounted after the system is restarted. For example, add the content in the last line, as shown in the following figure.

In the last line, **/dev/nvme0n1p1** is only an example.

```
#
# /etc/fstab
# Created by anaconda on Tue Nov  5 16:17:11 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root /                       xfs     defaults        0 0
UUID=fa02453a-0d4c-4a30-bf6c-c4d01cbd2c86 /boot                   xfs     defaults        0 0
UUID=006C-8EE2          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
/dev/mapper/centos-home /home                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
/dev/nvme0n1p1 /data xfs defaults 1 2
~
```

----End

Method 2: Using LVM for Drive Management

NOTE

Install the LVM2 package in the image as follows:

1. Configure the local yum source. For details, see 10.1 Configuring the Repo Server. If the repository has been configured, skip this step.
2. Install LVM2.

   #yum install lvm2

Step 1 Create a physical volume, for example, **sdb**.

```
#pvcreate /dev/sdb
```

Step 2 Create a physical volume group, for example, **datavg**.

```
#vgcreate datavg  /dev/sdb
```

Step 3 Create a logical volume, for example, **datalv** of 600 GB.

```
#lvcreate -L 600G -n datalv datavg
```

Step 4 Create a file system.

```
#mkfs.xfs /dev/datavg/datalv
```

Step 5 Create a data directory and mount it.

```
#mkdir /data
#mount /dev/datavg/datalv /data
```

Step 6 Run the **vi /etc/fstab** command and edit the **/etc/fstab** file to enable the data drive to be automatically mounted after the system is restarted. For example, add the content in the last line, as shown in the following figure.

In the last line, **/dev/datavg/datalv** is only an example.

```
# /etc/fstab
# Created by anaconda on Wed Sep  4 15:55:37 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=cff874ec-7337-4c7f-941d-9110356c268f /                       ext4    defaults        1 1
UUID=e058b556-6001-42df-ba8f-676fb2fabf74 /boot                   ext4    defaults        1 2
UUID=DEC7-ECFC          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
UUID=5dc6fd0a-ba47-4d7c-afe7-2df8d7d084d0 /home                   ext4    defaults        1 2
UUID=0543b4b6-1f64-4c63-8684-c7b7014d9e21 swap                    swap    defaults        0 0

/dev/datavg/datalv  /data  xfs defaults 1 2
```

----End

10.4.2.2.5 Creating a Database Directory and Granting Permissions

Step 1 In the created data directory **/data**, create directories for processes and grant permissions to the MySQL group or user created.

```
#mkdir -p /data/mariadb
#cd /data/mariadb
#mkdir data tmp run log
#chown -R mysql:mysql /data
```

----End

10.4.2.3 Installing, Running, and Uninstalling MariaDB Server

10.4.2.3.1 Installing MariaDB

Step 1 Configure the local yum source. For details, see 10.1 Configuring the Repo Server.

Step 2 Clear the cache.

```
#dnf clean all
```

Step 3 Create a cache.

```
#dnf makecache
```

Step 4 Install the MariaDB server.

```
#dnf install mariadb-server
```

Step 5 Check the installed RPM package.

```
#rpm -qa | grep mariadb
```

----End

10.4.2.3.2 Running MariaDB Server

Step 1 Start the MariaDB server.

```
#systemctl start mariadb
```

Step 2 Initialize the database.

```
#/usr/bin/mysql_secure_installation
```

During the command execution, you need to enter the password of the database user **root**. If no password is set, press **Enter**. Then, set the password as prompted.

Step 3 Log in to the database.

```
# mysql -u root -p
```

After the command is executed, the system prompts you to enter the password. The password is the one set in Step 2.

NOTE

Run the **\q** or **exit** command to exit the database.

----End

10.4.2.3.3 Uninstalling MariaDB

Step 1 Stop the database process.

```
#ps -ef | grep mysql
#kill -9 PID
```

Step 2 Run the **dnf remove mariadb-server** command to uninstall MariaDB.

```
#dnf remove mariadb-server
```

----End

10.4.2.4 Managing Database Users

10.4.2.4.1 Creating Users

Run the **CREATE USER** statement to create one or more users and set corresponding passwords.

```
CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';
```

In the preceding information:

- *username*: name of a user.
- *host*: hostname, that is, the name of the host where the user connects to the database. As a local user, you can set the parameter to **localhost**. If the host name is not specified during user creation, the host name is **%** by default, indicating a group of hosts.
- *password*: password for logging in to the server. The password can be null. If the password is null, the user can log in to the server without entering the password. This method, however, is not recommended because it provides low security.

To use the **CREATE USER** statement, you must have the INSERT permission on the database or the global CREATE USER permission.

After a user account is created using the **CREATE USER** statement, a record is added to the user table in the database. If the account to be created exists, an error will occur during statement execution.

A new user has few permissions and can perform only operations that do not require permissions. For example, a user can run the **SHOW** statement to query the list of all storage engines and character sets.

Example

#Create a local user whose password is 123456 and username is userexample1.

```
> CREATE USER 'userexample1'@'localhost' IDENTIFIED BY '123456';
```

#Create a user whose password is 123456, username is userexample2, and hostname is 192.168.1.100.

```
> CREATE USER 'userexample2'@'192.168.1.100' IDENDIFIED BY '123456';
```

10.4.2.4.2 Viewing Users

Run the **SHOW GRANTS** or **SELECT** statement to view one or more users.

View a specific user:

```
SHOW GRANTS [FOR 'username'@'hostname'];
SELECT USER,HOST,PASSWORD FROM mysql.user WHERE USER='username';
```

View all users:

```
SELECT USER,HOST,PASSWORD FROM mysql.user;
```

In the preceding information:

- *username*: name of a user.
- *hostname*: host name.

Example

#View the user userexample1.

```
> SHOW GRANTS FOR 'userexample1'@'localhost';
```

#View all users in the MySQL database.

```
> SELECT USER,HOST,PASSWORD FROM mysql.user;
```

10.4.2.4.3 Modifying Users

Modifying a Username

Run the **RENAME USER** statement to change one or more existing usernames.

```
RENAME USER 'oldusername'@'hostname' TO 'newusername'@'hostname';
```

In the preceding information:

- *oldusername*: original username.
- *newusername*: new username.
- *hostname*: host name.

The **RENAME USER** statement is used to rename an existing account. If the original account does not exist in the system or the new account exists, an error will occur when the statement is executed.

To use the **RENAME USER** statement, you must have the UPDATE permission on the database or the global CREATE USER permission.

Example of Modifying a User

Change the username **userexample1** to **userexample2** and change the hostname to **locahost**.

```
> RENAME USER 'userexample1'@'localhost' TO 'userexample2'@'localhost';
```

Modifying a User Password

Use the **SET PASSWORD** statement to modify the login password of a user.

```
SET PASSWORD FOR 'username'@'hostname' = PASSWORD('newpassword');
```

In the preceding information:

- **FOR 'username'@'hostname'**: specifies the username and hostname whose password is to be changed. This parameter is optional.
- **PASSWORD('newpassword')**: indicates that the **PASSWORD()** function is used to set a new password. That is, the new password must be transferred to the **PASSWORD()** function for encryption.

CAUTION

The **PASSWORD()** function is a unidirectional encryption function. Once encrypted, the original plaintext cannot be decrypted.

If the **FOR** clause is not added to the **SET PASSWORD** statement, the password of the current user is changed.

The **FOR** clause must be given in the format of '*username*'**@**'*hostname*', where *username* indicates the username of the account and *hostname* indicates the hostname of the account.

The account whose password is to be changed must exist in the system. Otherwise, an error occurs when the statement is executed.

Example of Changing a User Password

#Change the password of user **userexample** whose hostname is **locahost** to **0123456**.

```
> SET PASSWORD FOR 'userexample'@'localhost' = PASSWORD('0123456') ;
```

10.4.2.4.4 Deleting Users

Use the **DROP USER** statement to delete one or more user accounts and related permissions.

```
DROP USER 'username1'@'hostname1' [,'username2'@'hostname2']...;
```

CAUTION

The deletion of users does not affect the tables, indexes, or other database objects that they have created, because the database does not record the accounts that have created these objects.

The **DROP USER** statement can be used to delete one or more database accounts and their original permissions.

To use the **DROP USER** statement, you must have the DELETE permission on the database or the global CREATE USER permission.

In the **DROP USER** statement, if the hostname of an account is not specified, the hostname is **%** by default.

Example

#Delete the local user **userexample**.

```
> DROP USER 'userexample'@'localhost';
```

10.4.2.4.5 Granting Permissions to a User

Run the **GRANT** statement to grant permissions to a new user.

```
GRANT privileges ON databasename.tablename TO 'username'@'hostname';
```

In the preceding information:

- **ON** clause: specifies the object and its level on which the permission is granted.
- **privileges**: indicates the operation permissions of a user, such as **SELECT**, INSERT, and **UPDATE**. To grant all permissions to a user, use **ALL**.
- *databasename*: database name.
- *tablename*: table name.
- **TO** clause: sets the user password and specifies the user to whom the permission is granted.
- *username*: name of a user.
- *hostname*: host name.

To grant the user the permission to operate all databases and tables, use asterisks (*), for example, ***.***.

If you specify a password for an existing user in the **TO** clause, the new password will overwrite the original password.

If the permission is granted to a non-existent user, a **CREATE USER** statement is automatically executed to create the user, but the password must be specified for the user.

Example

#Grant the SELECT and INSERT permissions to local user userexample.

```
> GRANT SELECT,INSERT ON *.* TO 'userexample'@'localhost';
```

10.4.2.4.6 Deleting User Permissions

Run the **REVOKE** statement to delete the permissions of a user, but the user will not be deleted.

```
REVOKE privilege ON databasename.tablename FROM 'username'@'hostname';
```

The parameters in the **REVOKE** statement are the same as those in the **GRANT** statement.

To use the **REVOKE** statement, you must have the global CREATE USER or UPDATE permission for the database.

Example

#Delete the INSERT permission of local user userexample.

```
> REVOKE INSERT ON *.* FROM 'userexample'@'localhost';
```

10.4.2.5 Managing Databases

10.4.2.5.1 Creating a Database

Run the **CREATE DATABASE** statement to create a database.

```
CREATE DATABASE databasename;
```

In the preceding command, *databasename* can be replaced with the database name, which is case insensitive.

Example

#Create a database named **databaseexample**.

```
> CREATE DATABASE databaseexample;
```

10.4.2.5.2 Viewing a Database

Run the **SHOW DATABASES** statement to view a database.

```
SHOW DATABASES;
```

Example

#View all databases.

```
> SHOW DATABASES;
```

10.4.2.5.3 Selecting a Database

Generally, you need to select a target database before creating or querying a table. Use the **USE** statement to select a database.

```
USE databasename;
```

In the preceding command, **databasename** indicates the database name.

Example

#Select the **databaseexample** database.

```
> USE databaseexample;
```

10.4.2.5.4 Deleting a Database

You can run the **DROP DATABASE** statement to delete a database.

CAUTION

Exercise caution when deleting a database. Once a database is deleted, all tables and data in the database will be deleted.

```
DROP DATABASE databasename;
```

In the preceding command, **databasename** indicates the database name.

The **DROP DATABASE** command is used to delete an existing database. After this command is executed, all tables in the database are deleted, but the user permissions of the database are not automatically deleted.

To use **DROP DATABASE**, you need the **DROP** permission on the database.

DROP SCHEMA is a synonym of **DROP DATABASE**.

Example

#Delete the **databaseexample** database.

```
> DROP DATABASE databaseexample;
```

10.4.2.5.5 Backing Up a Database

Run the **mysqldump** command to back up the database.

Back up one or more tables.

```
mysqldump [options] databasename [tablename ...] > outfile
```

Back up one or more databases:

```
mysqldump [options] -databases databasename ... > outfile
```

Back up all databases:

```
mysqldump [options] -all-databases > outputfile
```

In the preceding information:

- *databasename*: database name.
- *tablename*: name of a data table.
- *outfile*: database backup file.
- *options*: parameter option of the **mysqldump** command. Multiple parameters can be separated by spaces. The common parameters of the **mysqldump** command are as follows:
  - **-u, --user=***username*: specifies the username.
  - **-p, --password[=***password***]**: specifies the password.
  - **-P, --port=***portnumber*: specifies the port number.
  - **-h, --host=***hostname*: specifies the hostname.
  - **-r, --result-file=***filename*: saves the export result to a specified file, which is equivalent to >.
  - **-t**: backs up data only.
  - **-d**: backs up the table structure only.

Example

#Back up all the databases of the user **root** on the host **192.168.202.144** through port **3306** to the **alldb.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 --all-databases > alldb.sql
```

#Back up the db1 database of the user **root** on the host **192.168.202.144** through port **3306** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 --databases db1 > db1.sql
```

#Back up the tb1 table of the db1 database of the user **root** on the host **192.168.202.144** through port **3306** to the **db1tb1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 db1 tb1 > db1tb1.sql
```

#Back up only the table structure of the db1 database of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 -d db1 > db1.sql
```

#Back up only the data of the db1 database of the user **root** on the host **192.168.202.144** through port **3306** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 -t db1 > db1.sq
```

10.4.2.5.6 Restoring a Database

Run the **mysqldump** command to restore the database.

Back up one or more tables:

```
mysql -h hostname -P portnumber -u username -ppassword databasename < infile
```

In the preceding information:

- *hostname*: host name.
- *portnumber*: port number.
- *username*: name of a user.
- *password*: password.
- *databasename*: database name.
- *infile*: **outfile** parameter in the **mysqldump** command.

Example

#Restore a database.

```
# mysql -h 192.168.202.144 -P 3306 -uroot -p123456 -t db1 < db1.sql
```

10.4.3 MySQL Server

10.4.3.1 Software Description

MySQL is a relational database management system (RDBMS) developed by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle). It is one of the most popular Relational Database Management Systems (RDBMSs) in the industry, especially for web applications.

A relational database stores data in different tables instead of in a large data warehouse to improve efficiency and flexibility.

The Structured Query Language (SQL) used by MySQL is the most common standard language for accessing databases. MySQL uses dual-licensing distribution and is available in two editions: Community Edition and Commercial Edition. MySQL is optimal for small or medium-sized websites because of its small size, fast speed, low cost, and especially the open source code.

10.4.3.2 Configuring the Environment

NOTE

The following environment configuration is for reference only. Configure the environment based on the site requirements.

10.4.3.2.1 Disabling the Firewall and Automatic Startup

NOTE

It is recommended that firewall be disabled in the test environment to prevent network impact. Configure the firewall based on actual requirements.

Step 1 Stop the firewall service.

```
#systemctl stop firewalld
```

Step 2 Disable the firewall service.

```
#systemctl disable firewalld
```

NOTE

The automatic startup is automatically disabled as the firewall is disabled.

----End

10.4.3.2.2 Disabling SELinux

Step 1 Modify the configuration file.

```
#sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/sysconfig/selinux
```

----End

10.4.3.2.3 Creating a User Group and a User

NOTE

In the server environment, independent users are assigned to each process to implement permission isolation for security purposes. The user group and user are created for the OS, not for the database.

Step 1 Create a MySQL user or user group.

```
#groupadd mysql
#useradd -g mysql mysql
```

Step 2 Set the user password.

```
#passwd mysql
```

Enter the password twice for confirmation.

----End

10.4.3.2.4 Creating Data Drives

NOTE

- If a performance test needs to be performed, an independent drive is required for the data directory. You need to format and mount the drive. For details, see Method 1 or Method 2.
- In a non-performance test, run the following command to create a data directory. Then skip this section.

  #mkdir /data

Method 1: Using fdisk for Drive Management

Step 1 Create a partition, for example, **/dev/sdb**.

```
#fdisk /dev/sdb
```

Step 2 Enter **n** and press **Enter**.

Step 3 Enter **p** and press **Enter**.

Step 4 Enter **1** and press **Enter**.

Step 5 Retain the default settings and press **Enter**.

Step 6 Retain the default settings and press **Enter**.

Step 7 Enter **w** and press **Enter**.

Step 8 Create a file system, for example, **xfs**.

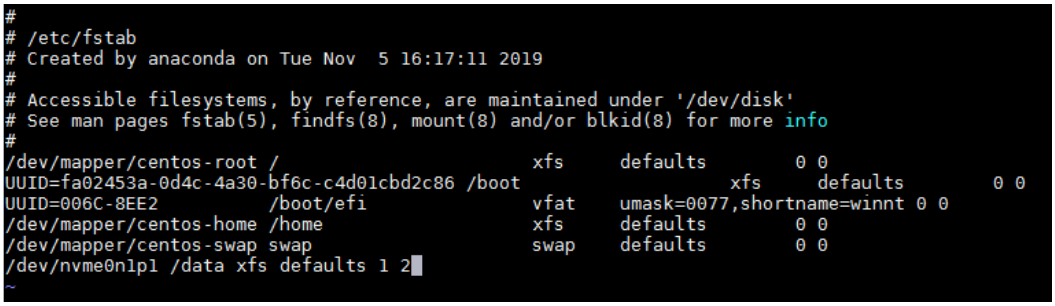
```
#mkfs.xfs /dev/sdb1
```

Step 9 Mount the partition to **/data** for the OS.

```
#mkdir /data
#mount /dev/sdb1 /data
```

Step 10 Run the **vi /etc/fstab** command and edit the **/etc/fstab** file to enable the data drive to be automatically mounted after the system is restarted. For example, add the content in the last line, as shown in the following figure.

In the last line, **/dev/nvme0n1p1** is only an example.



----End

Method 2: Using LVM for Drive Management

NOTE

Install the LVM2 package in the image as follows:

1. Configure the local yum source. For details, see 10.1 Configuring the Repo Server. If the repository has been configured, skip this step.
2. Install LVM2.

   #yum install lvm2

Step 1 Create a PV, for example, **sdb**.

```
#pvcreate /dev/sdb
```

Step 2 Create a physical VG, for example, **datavg**.

```
#vgcreate  datavg  /dev/sdb
```

Step 3 Create an LV, for example, **datalv** of 600 GB.

```
#lvcreate -L 600G -n datalv datavg
```

Step 4 Create a file system.

```
#mkfs.xfs /dev/datavg/datalv
```

Step 5 Create a data directory and mount it.

```
#mkdir /data
#mount /dev/datavg/datalv /data
```

Step 6 Run the **vi /etc/fstab** command and edit the **/etc/fstab** file to enable the data drive to be automatically mounted after the system is restarted. For example, add the content in the last line, as shown in the following figure.

In the last line, **/dev/datavg/datalv** is only an example.

```
# /etc/fstab
# Created by anaconda on Wed Sep  4 15:55:37 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=cff874ec-7337-4c7f-941d-9110356c268f /                       ext4    defaults        1 1
UUID=e058b556-6001-42df-ba8f-676fb2fabf74 /boot                   ext4    defaults        1 2
UUID=DEC7-ECFC          /boot/efi               vfat    umask=0077,shortname=winnt 0 0
UUID=5dc6fd0a-ba47-4d7c-afe7-2df8d7d084d0 /home                   ext4    defaults        1 2
UUID=0543b4b6-1f64-4c63-8684-c7b7014d9e21 swap                    swap    defaults        0 0

/dev/datavg/datalv  /data  xfs defaults 1 2
```

----End

10.4.3.2.5 Creating a Database Directory and Granting Permissions

Step 1 In the created data directory **/data**, create directories for processes and grant permissions to the MySQL group or user created.

```
#mkdir -p /data/mysql
#cd /data/mysql
#mkdir data tmp run log
#chown -R mysql:mysql /data
```

----End

10.4.3.3 Installing, Running, and Uninstalling MySQL

10.4.3.3.1 Installing MySQL

Step 1 Configure the local yum source. For details, see 10.1 Configuring the Repo Server.

Step 2 Clear the cache.

```
#dnf clean all
```

Step 3 Create a cache.

```
#dnf makecache
```

Step 4 Install the MySQL server.

```
#dnf install mysql
```

Step 5 Check the installed RPM package.

```
#rpm -qa | grep mysql
```

----End

10.4.3.3.2 Running MySQL

Step 1 Modify the configuration file.

1. Create the **my.cnf** file and change the file paths (including the software installation path **basedir** and data path **datadir**) based on the actual situation.

   ```
   #vi /etc/my.cnf
   ```

   Edit the **my.cnf** file as follows:

   ```
   [mysqld safe]
   log-error=/data/mysql/log/mysql.log
   pid-file=/data/mysql/run/mysqld.pid
   [mysqldump]
   quick
   [mysql]
   no-auto-rehash
   [client]
   default-character-set=utf8
   [mysqld]
   basedir=/usr/local/mysql
   socket=/data/mysql/run/mysql.sock
   tmpdir=/data/mysql/tmp
   datadir=/data/mysql/data
   default authentication plugin=mysql native password
   port=3306
   user=mysql
   ```

2. Ensure that the **my.cnf** file is correctly modified.

   ```
   #cat /etc/my.cnf
   ```

```
[root@localhost mysql]# cat /etc/my.cnf
[mysqld_safe]
log-error=/data/mysql/log/mysql.log
pid-file=/data/mysql/run/mysqld.pid
[mysqldump]
quick
[mysql]
no-auto-rehash
[client]
default-character-set=utf8
[mysqld]
basedir=/usr/local/mysql
socket=/data/mysql/run/mysql.sock
tmpdir=/data/mysql/tmp
datadir=/data/mysql/data
default_authentication_plugin=mysql_native_password
port=3306
user=mysql
```

CAUTION

In the configuration file, **basedir** specifies the software installation path. Change it based on actual situation.

3. Change the group and user of the **/etc/my.cnf** file to **mysql:mysql**.

```
#chown mysql:mysql /etc/my.cnf
```

Step 2 Configure environment variables.

1. Add the path of the MySQL binary files to the **PATH** parameter.

```
#echo export  PATH=$PATH:/usr/local/mysql/bin  >> /etc/profile
```

CAUTION

In the command, **/usr/local/mysql/bin** is the absolute path of the **bin** files in the MySQL software installation directory. Change it based on actual situation.

2. Run the following command to make the environment variables take effect:

```
#source /etc/profile
```

Step 3 Initialize the database.

NOTE

The second line from the bottom contains the initial password, which will be used when you log in to the database.

```
#mysqld --defaults-file=/etc/my.cnf --initialize
2020-03-18T03:27:13.702385Z 0 [System] [MY-013169] [Server]
/usr/local/mysql/bin/mysqld (mysqld 8.0.17) initializing of server in progress as
process 34014
2020-03-18T03:27:24.112453Z 5 [Note] [MY-010454] [Server] A temporary password is
generated for root@localhost: iNat=)#V2tZu
2020-03-18T03:27:28.576003Z 0 [System] [MY-013170] [Server]
/usr/local/mysql/bin/mysqld (mysqld 8.0.17) initializing of server has completed
```

If the command output contains "initializing of server has completed", the database has been initialized. In the command output, "iNat=)#V2tZu" in "A temporary password is generated for root@localhost: iNat=)#V2tZu" is the initial password.

Step 4 Start the database.

CAUTION

Start MySQL as user **mysql** if it is the first time to start the database service. If you start MySQL as user **root**, a message will be displayed indicating that the **mysql.log** file is missing. If you start MySQL as user **mysql**, the **mysql.log** file will be generated in the **/data/mysql/log** directory. No error will be displayed if you start the database as user **root** again.

1. Modify the file permission.

```
#chmod 777 /usr/local/mysql/support-files/mysql.server
```

2. Start MySQL.

```
#cp /usr/local/mysql/support-files/mysql.server /etc/init.d/mysql
#chkconfig mysql on
```

Start MySQL as user **mysql**.

```
#su - mysql
$service mysql start
```

Step 5 Log in to the database.

NOTE

- Enter the initial password generated during database initialization (Step 3).
- If MySQL is installed by using an RPM package obtained from the official website, the **mysqld** file is located in the **/usr/sbin** directory. Ensure that the directory specified in the command is correct.

```
$/usr/local/mysql/bin/mysql -uroot -p  -S /data/mysql/run/mysql.sock
```

```
[mysql@localhost ~]$ /usr/local/mysql/bin/mysql -uroot -p  -S /data/mysql/run/mysql.sock
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.17

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Step 6 Configure the database accounts and passwords.

1. After logging in to the database, change the password of user **root** for logging in to the database.

```
mysql>alter user 'root'@'localhost' identified by "123456";
```

2. Create a user **root** for all the other hosts in the domain.

```
mysql>create user 'root'@'%' identified by '123456';
```

3. Grant permissions to the user **root**.

```
mysql>grant all privileges on *.* to 'root'@'%';
mysql>flush privileges;
```

```
mysql> alter user 'root'@'localhost' identified by "123456";
Query OK, 0 rows affected (0.01 sec)

mysql> create user 'root'@'%' identified by '123456';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all privileges on *.* to 'root'@'%';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

Step 7 Exit the database.

Run the **\q** or **exit** command to exit the database.

```
mysql>exit
```

```
mysql> exit
Bye
```

----End

10.4.3.3.3 Uninstalling MySQL

Step 1 Stop the database process.

```
#ps -ef | grep mysql
#kill -9 PID
```

Step 2 Run the **dnf remove mysql** command to uninstall MySQL.

```
#dnf remove mysql
```

----End

10.4.3.4 Managing Database Users

10.4.3.4.1 Creating Users

Run the **CREATE USER** statement to create one or more users and set corresponding passwords.

```
CREATE USER 'username'@'hostname' IDENTIFIED BY 'password';
```

In the preceding information:

- *username*: name of a user.
- *host*: hostname, that is, the name of the host where the user connects to the database. As a local user, you can set the parameter to **localhost**. If the host name is not specified during user creation, the host name is **%** by default, indicating a group of hosts.
- *password*: password for logging in to the server. The password can be null. If the password is null, the user can log in to the server without entering the password. This method, however, is not recommended because it provides low security.

To use the **CREATE USER** statement, you must have the **INSERT** permission on the database or the global **CREATE USER** permission.

After a user account is created using the **CREATE USER** statement, a record is added to the user table in the database. If the account to be created exists, an error will occur during statement execution.

A new user has few permissions and can perform only operations that do not require permissions. For example, a user can run the **SHOW** statement to query the list of all storage engines and character sets.

Example

#Create a local user whose password is **123456** and username is **userexample1**.

```
> CREATE USER 'userexample1'@'localhost' IDENTIFIED BY '123456';
```

#Create a user whose password is **123456**, username is **userexample2**, and hostname is **192.168.1.100**.

```
> CREATE USER 'userexample2'@'192.168.1.100' IDENDIFIED BY '123456';
```

10.4.3.4.2 Viewing Users

Run the **SHOW GRANTS** or **SELECT** statement to view one or more users.

View a specific user:

```
SHOW GRANTS [FOR 'username'@'hostname'];
SELECT USER,HOST,PASSWORD FROM mysql.user WHERE USER='username';
```

View all users:

```
SELECT USER,HOST FROM mysql.user;
```

In the preceding information:

- *username*: name of a user.
- *hostname*: host name.

Example

#View the user **userexample1**.

```
> SHOW GRANTS FOR 'userexample1'@'localhost';
```

#View all users in the MySQL database.

```
> SELECT USER,HOST FROM mysql.user;
```

10.4.3.4.3 Modifying Users

Modifying a Username

Run the **RENAME USER** statement to change one or more existing usernames.

```
RENAME USER 'oldusername'@'hostname' TO 'newusername'@'hostname';
```

In the preceding information:

- *oldusername*: original username.
- *newusername*: new username.
- *hostname*: host name.

The **RENAME USER** statement is used to rename an existing account. If the original account does not exist in the system or the new account exists, an error will occur when the statement is executed.

To use the **RENAME USER** statement, you must have the **UPDATE** permission on the database or the global **CREATE USER** permission.

Example of Modifying a User

Change the username **userexample1** to **userexample2** and change the hostname to **locahost**.

```
> RENAME USER 'userexample1'@'localhost' TO 'userexample2'@'localhost';
```

Modifying a User Password

Use the **SET PASSWORD** statement to modify the login password of a user.

```
SET PASSWORD FOR 'username'@'hostname' = 'newpassword';
```

In the preceding information:

- **FOR**'*username*'**@**'*hostname*': specifies the username and hostname whose password is to be changed. This parameter is optional.
- *newpassword*: new password.

If the **FOR** clause is not added to the **SET PASSWORD** statement, the password of the current user is changed.

The **FOR** clause must be given in the format of '*username*'**@**'*hostname*', where *username* indicates the username of the account and *hostname* indicates the hostname of the account.

The account whose password is to be changed must exist in the system. Otherwise, an error occurs when the statement is executed.

Example of Changing a User Password

#Change the password of user **userexample** whose hostname is **locahost** to **0123456**.

```
> SET PASSWORD FOR 'userexample'@'localhost' = '0123456';
```

10.4.3.4.4 Deleting Users

Use the **DROP USER** statement to delete one or more user accounts and related permissions.

```
DROP USER 'username1'@'hostname1' [,'username2'@'hostname2']...;
```

CAUTION

The deletion of users does not affect the tables, indexes, or other database objects that they have created, because the database does not record the accounts that have created these objects.

The **DROP USER** statement can be used to delete one or more database accounts and their original permissions.

To use the **DROP USER** statement, you must have the **DELETE** permission on the database or the global **CREATE USER** permission.

In the **DROP USER** statement, if the hostname of an account is not specified, the hostname is **%** by default.

Example

#Delete the local user **userexample**.

```
> DROP USER 'userexample'@'localhost';
```

10.4.3.4.5 Granting Permissions to a User

Run the **GRANT** statement to grant permissions to a new user.

```
GRANT privileges ON databasename.tablename TO 'username'@'hostname';
```

In the preceding information:

- **ON** clause: specifies the object and level on which the permission is granted.
- *privileges*: indicates the operation permissions of a user, such as **SELECT**, INSERT, and **UPDATE**. To grant all permissions to a user, use **ALL**.
- *databasename*: database name.
- *tablename*: table name.
- **TO** clause: sets the user password and specifies the user to whom the permission is granted.
- *username*: name of a user.
- *hostname*: host name.

To grant the user the permission to operate all databases and tables, use asterisks (*), for example, ***.***.

If you specify a password for an existing user in the **TO** clause, the new password will overwrite the original password.

If the permission is granted to a non-existent user, a **CREATE USER** statement is automatically executed to create the user, but the password must be specified for the user.

Example

#Grant the **SELECT** and **INSERT** permissions to local user **userexample**.

```
> GRANT SELECT,INSERT ON *.* TO 'userexample'@'localhost';
```

10.4.3.4.6 Deleting User Permissions

Run the **REVOKE** statement to delete the permissions of a user, but the user will not be deleted.

```
REVOKE privilege ON databasename.tablename FROM 'username'@'hostname';
```

The parameters in the **REVOKE** statement are the same as those in the **GRANT** statement.

To use the **REVOKE** statement, you must have the global **CREATE USER** or **UPDATE** permission for the database.

Example

#Delete the **INSERT** permission of local user **userexample**.

```
> REVOKE INSERT ON *.* FROM 'userexample'@'localhost';
```

10.4.3.5 Managing Databases

10.4.3.5.1 Creating a Database

Run the **CREATE DATABASE** statement to create a database.

```
CREATE DATABASE databasename;
```

In the preceding command, *databasename* can be replaced with the database name, which is case insensitive.

Example

#Create a database named **databaseexample**.

```
> CREATE DATABASE databaseexample;
```

10.4.3.5.2 Viewing a Database

Run the **SHOW DATABASES** statement to view a database.

```
SHOW DATABASES;
```

Example

#View all databases.

```
> SHOW DATABASES;
```

10.4.3.5.3 Selecting a Database

Generally, you need to select a target database before creating or querying a table. Use the **USE** statement to select a database.

```
USE databasename;
```

In the preceding command, *databasename* indicates the database name.

Example

#Select the **databaseexample** database.

```
> USE databaseexample;
```

10.4.3.5.4 Deleting a Database

Run the **DROP DATABASE** statement to delete a database.

CAUTION

Exercise caution when deleting a database. Once a database is deleted, all tables and data in the database will be deleted.

```
DROP DATABASE databasename;
```

In the preceding command, *databasename* indicates the database name.

The **DROP DATABASE** command is used to delete an existing database. After this command is executed, all tables in the database are deleted, but the user permissions of the database are not automatically deleted.

To use **DROP DATABASE**, you need the **DROP** permission on the database.

DROP SCHEMA is a synonym of **DROP DATABASE**.

Example

#Delete the **databaseexample** database.

```
> DROP DATABASE databaseexample;
```

10.4.3.5.5 Backing Up a Database

Run the **mysqldump** command to back up the database.

Back up one or more tables:

```
mysqldump [options] databasename [tablename ...] > outfile
```

Back up one or more databases:

```
mysqldump [options] -databases databasename ... > outfile
```

Back up all databases:

```
mysqldump [options] -all-databases > outputfile
```

In the preceding information:

- *databasename*: database name.
- *tablename*: name of a data table.
- *outfile*: database backup file.
- *options*: parameter option of the **mysqldump** command. Multiple parameters can be separated by spaces. The common parameters of the **mysqldump** command are as follows:
  - **-u, --user=***username*: specifies the username.
  - **-p, --password[=***password***]**: specifies the password.
  - **-P, --port=***portnumber*: specifies the port number.
  - **-h, --host=***hostname*: specifies the hostname.
  - **-r, --result-file=***filename*: saves the export result to a specified file, which is equivalent to >.
  - **-t**: backs up data only.
  - **-d**: backs up the table structure only.

Example

#Back up all the databases of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **alldb.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 --all-databases > alldb.sql
```

#Back up the db1 database of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 --databases db1 > db1.sql
```

#Back up the tb1 table of the db1 database of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1tb1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 db1 tb1 > db1tb1.sql
```

#Back up only the table structure of the db1 database of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 -d db1 > db1.sql
```

#Back up only the table structure of the db1 database of user **root** on port **3306** of the host whose IP address is **192.168.202.144** to the **db1.sql** file.

```
# mysqldump -h 192.168.202.144 -P 3306 -uroot -p123456 -t db1 > db1.sq
```

10.4.3.5.6 Restoring a Database

Run the **mysqldump** command to restore the database.

Back up one or more tables:

```
mysql -h hostname -P portnumber -u username -ppassword databasename < infile
```

In the preceding information:

- *hostname*: host name.
- *portnumber*: port number.
- *username*: name of a user.
- *password*: password.
- *databasename*: database name.
- *infile*: **outfile** parameter in the **mysqldump** command.

Example

#Restore a database.

```
# mysql -h 192.168.202.144 -P 3306 -uroot -p123456 -t db1 < db1.sql
```

11 FAQs

11.1 Why Is the Memory Usage of the libvirtd Service Queried by Running the systemctl and top Commands Different?

Symptom

The output of the **systemctl** and **systemd-cgtop** commands shows that the libvirtd service occupies more than 1.5 GB memory, but the output of the **top** command shows that the libvirtd service occupies about 70 MB memory.

Possible Cause

The memory displayed in the services (including systemctl and systemd-cgtop) managed by systemd can be obtained from **memory.usage_in_bytes** in Cgroup. Running the **top** command is to query the memory information in the **/proc** directory. The query results are different because the statistical method varies.

Generally, the memory used by service processes has the following types:

- anon_rss: anonymous pages in user mode address spaces, for example, memory allocated by calling the malloc function or the mmap function with configured **MAP_ANONYMOUS**. When the system memory is insufficient, this type of memory can be swapped by the kernel.
- file_rss: mapped pages in user mode address spaces, including map file (such as mmap of a specified file) and map tmpfs (such as IPC shared memory). When the system

memory is insufficient, the kernel can reclaim these pages. Data may need to be synchronized between the kernel and map file before reclamation.

- file_cache: file cache (page in page cache of disk file), which is generated when a file is read or written. When the system memory is insufficient, the kernel can reclaim these pages. Data may need to be synchronized between the kernel and map file before reclamation.
- buffer pages: belongs to page cache, for example, cache generated when block device files are read.

anon_rss and file_rss belong to the resident set size (RSS) of processes, and file_cache and buffer pages belong to page cache. In brief:

RSS in the output of the **top** command = anon_rss + file_rss; Shared memory (SHR) = file_rss

memory.usage_in_bytes in Cgroup = cache + RSS + swap

In conclusion, the definition of memory usage obtained by running the **systemd** command is different from that obtained by running the **top** command. Therefore, the query results are different.

11.2 An Error Occurs When stripsize Is Set to 4 During RAID 0 Volume Configuration

Symptom

An error occurs when the **stripsize** parameter is set to **4** during RAID 0 volume configuration.

Possible Cause

The 64 KB page table can be enabled only in the scenario where **stripsize** is set to **64**.

Solution

You do not need to modify the configuration file. When running the **lvcreate** command on openEuler, set **stripesize** to **64** because the minimum supported stripe size is 64 KB.

11.3 Failed to Compile MariaDB Using rpmbuild

Symptom

When you log in to the system as user **root** and run the **rpmbuild** command to compile the MariaDB source code, the compilation fails and the following information is displayed:

```
+ echo 'mysql can'\''t run test as root'
mysql can't run test as root
+ exit 1
```

Possible Cause

The MariaDB does not allow user **root** to execute test cases. However, test cases are automatically executed during compilation. As a result, the compilation process is blocked.

Solution

Use a text editor, such as vi, to modify the value of the **runtest** variable in the **mariadb.spec** file.

Before the modification:

```
%global runtest 1
```

After the modification:

```
%global runtest 0
```

The modification disables the function of executing test cases during compilation, which does not affect the compilation and the RPM package content after compilation.

11.4 Failed to Start the SNTP Service Using the Default Configuration

Symptom

The SNTP service fails to be started with the default configuration.

Possible Cause

The domain name of the NTP server is not added to the default configuration.

Solution

Modify the **/etc/sysconfig/sntp** file and add the domain name of the NTP server in China: **0.generic.pool.ntp.org**.

11.5 Installation Failure Caused by Software Package Conflict, File Conflict, or Missing Software Package

Symptom

Software package conflict, file conflict, or missing software packages may occur during software package installation. As a result, the upgrade is interrupted and the installation fails. The error information about software package conflict, file conflict, and missing software packages is as follows:

The following is an example of software package conflict error information (the conflict between **libev-libevent-devel-4.24-11.oe1.aarch64** and **libevent-devel-2.1.11-2.oe1.aarch64** is used as an example):

```
package libev-libevent-devel-4.24-11.oe1.aarch64 conflicts with libevent-devel
provided by libevent-devel-2.1.11-2.oe1.aarch64
 - cannot install the best candidate for the job
 - conflicting requests
```

The following is an example of file conflict error information (the **/usr/bin/containerd** file conflict is used as an example):

```
Error: Transaction test error:
 file /usr/bin/containerd from install of containerd-1.2.0-101.oe1.aarch64 conflicts
with file from package docker-engine-18.09.0-100.aarch64
 file /usr/bin/containerd-shim from install of containerd-1.2.0-101.oe1.aarch64
conflicts with file from package docker-engine-18.09.0-100.aarch64
```

The following is an example of the error message indicating that the **blivet-data** software package is missing:

```
Error:
  Problem: cannot install both blivet-data-1:3.1.1-6.oe1.noarch and
blivet-data-1:3.1.1-5.noarch
   - package python2-blivet-1:3.1.1-5.noarch requires blivet-data = 1:3.1.1-5, but none
of the providers can be installed
   - cannot install the best update candidate for package blivet-data-1:3.1.1-5.noarch
   - problem with installed package python2-blivet-1:3.1.1-5.noarch(try to add
'--allowerasing' to command line to replace conflicting packages or '--skip-broken'
to skip uninstallable packages or '--nobest' to use not only best candidate packages)
```

Possible Cause

- In the software packages provided by openEuler, some software packages have different names but the same functions. As a result, the software packages cannot be installed at the same time.
- In the software packages provided by openEuler, some software packages have different names but the same functions. As a result, the files after installation are the same, causing file conflict.
- Some software packages are depended on by other software packages before the upgrade. After the software packages are upgraded, the software packages that depend on them may fail to be installed due to lack of software packages.

Solution

If a software package conflict occurs, perform the following steps (the software package conflict in "Symptom" is used as an example):

1. According to the error message displayed during the installation, the software package that conflicts with the to-be-installed software package **libev-libevent-devel-4.24-11.oe1.aarch64** is **libevent-devel-2.1.11-2.oe1.aarch64**.
2. Run the **dnf remove** command to uninstall the software package that conflicts with the software package to be installed.

   ```
   # dnf remove libevent-devel-2.1.11-2.oe1.aarch64
   ```

3. Perform the installation again.

If a file conflict occurs, perform the following steps (the file conflict in "Symptom" is used as an example):

1. According to the error message displayed during the installation, the names of the software packages that cause the file conflict are **containerd-1.2.0-101.oe1.aarch64** and **docker-engine-18.09.0-100.aarch64**.
2. Record the names of the software packages that do not need to be installed. The following uses **docker-engine-18.09.0-100.aarch64** as an example.
3. Run the **dnf remove** command to uninstall the software package that does not need to be installed.

   ```
   # dnf remove docker-engine-18.09.0-100.aarch64
   ```

4. Perform the installation again.

If a software package is missing, perform the following steps (the missed software package in "Symptom" is used as an example):

1. Determine the name of the software package to be upgraded (**blivet-data-1:3.1.1-5.noarch**) and the name of the dependent software package (**python2-blivet-1:3.1.1-5.noarch**) based on the error information displayed during the upgrade.
2. Run the **dnf remove** command to uninstall the software package that depends on the upgrade package or add the **--allowerasing** parameter when upgrading the software package.
   - Run the **dnf remove** command to uninstall the software package that depends on the **blivet-data-1:3.1.1-5.noarch** software package.

     ```
     # dnf remove python2-blivet-1:3.1.1-5.noarch
     ```

   - Add the **--allowerasing** parameter when upgrading the software package.

     ```
     # yum update blivet-data-1:3.1.1-5.noarch -y --allowerasing
     ```

3. Perform the upgrade again.