Understanding file permissions and ownership

Topics : Centos Server
Written on March 05, 2024

Understanding file permissions and ownership is crucial for managing access to files and directories in CentOS. In Linux, each file and directory has associated permissions that specify who can read, write, and execute them, as well as ownership information that determines which user and group own the file. Here's an overview of file permissions and ownership in CentOS:

1. File Permissions:

- File permissions consist of three sets of permissions: read (r), write (w), and execute (x).
- Permissions are set separately for three types of users: the file owner, the group owner, and others.
- The `ls -l` command can be used to display file permissions in long format. For example:

  ```
  ls -l filename
  ```

- The output will look like this:

  ```
  -rw-r--r-- 1 owner group size date filename
  ```

- In this example, the file has read and write permissions for the owner (`rw-`), read-only permissions for the group (`r--`), and read-only permissions for others (`r--`).

2. File Ownership:

- Every file and directory in CentOS is owned by a user and a group.
- The user who creates the file is usually set as the owner, and the group ownership defaults to the user's primary group.
- The `ls -l` command also displays ownership information. For example:

  ```
  ls -l filename
  ```

- The output includes the owner and group of the file:

  ```
  -rw-r--r-- 1 owner group size date filename
  ```

- In this example, `owner` is the user who owns the file, and `group` is the group that owns the file.

3. Modifying File Permissions and Ownership:

- File permissions can be modified using the `chmod` command. For example, to give read and write permissions to the owner of a file:

  ```
  chmod u+rw filename
  ```

- Similarly, you can use `g` for group and `o` for others to modify permissions for the group and others, respectively.
- File ownership can be modified using the `chown` command. For example, to change the owner of a file:

  ```
  chown newowner filename
  ```

- You can also change both the owner and group simultaneously using the `chown` command. For example:

  ```
  chown newowner:newgroup filename
  ```

4. Special Permissions:

- In addition to the basic read, write, and execute permissions, there are special permissions like setuid, setgid, and sticky bit.
- The setuid (`s`) and setgid (`s`) permissions allow a user who runs an executable file to temporarily gain the privileges of the file's owner or group, respectively.
- The sticky bit (`t`) permission is commonly used on directories to restrict the deletion of files by users other than the file owner.