DevOps Shack

User/Group & Permissions/Ownership

[Click Here To Enrol To Batch-6 | DevOps & Cloud DevOps]

User and Group Management in Linux

Linux is a multi-user operating system, allowing multiple users to access the system simultaneously. User and group management is essential for maintaining system security and controlling access to files and resources.

Users

Types of Users

1. **Root User:** The root user is the superuser with unrestricted access to the system. The root user's home directory is /root.
2. **Regular Users:** These are non-privileged users created by the root user or system administrators. Each regular user has a unique username and a home directory located in /home/username.

Creating Users

To create a new user, use the useradd command followed by the username:

sudo useradd john

After creating a user, set the password using the passwd command:

sudo passwd john

User Configuration Files

- **/etc/passwd:** Contains user account information. Each line represents a user.

plaintext

john:x:1001:1001:John Doe,,,:/home/john:/bin/bash

- **/etc/shadow:** Contains encrypted user passwords and account expiration information. Only readable by the root user.

john:6abcd1234$abcdefghijklmnopqrstu:/home/john:/bin/bash

- **/etc/group:** Contains group information.

developers:x:1002:john

Groups

Groups allow administrators to manage permissions for multiple users simultaneously. Each user can belong to one or more groups.

Creating Groups

To create a new group, use the groupadd command:

sudo groupadd developers

Adding Users to Groups

To add a user to a group, use the usermod command:

sudo usermod -aG developers john

User and Group Management Commands

- **useradd:** Adds a new user.

sudo useradd username

- **passwd:** Sets or changes a user's password.

sudo passwd username

- **usermod:** Modifies a user account. The -aG option adds the user to a group.

sudo usermod -aG groupname username

- **groupadd:** Adds a new group.

sudo groupadd groupname

- **groups:** Displays the groups a user belongs to.

groups username

- **deluser:** Deletes a user.

sudo deluser username

- **delgroup:** Deletes a group.

sudo delgroup groupname

Permissions and Ownership

Linux uses a permission model to control access to files and directories. Each file and directory has an owner, a group, and permissions set for the owner, group, and others.

File Ownership

- **Owner:** The user who owns the file.
- **Group:** The group that owns the file.

To change file ownership, use the chown command:

sudo chown user:group filename

File Permissions

Permissions are represented by a string of characters and divided into three groups: owner, group, and others.

plaintext

Copy code

-rwxr-xr--

- The first character indicates the file type (- for a regular file, d for a directory).
- The next three characters represent the owner's permissions.
- The following three characters represent the group's permissions.
- The last three characters represent the others' permissions.

Each set of permissions includes:

- **r:** Read
- **w:** Write
- **x:** Execute

Changing File Permissions

To change file permissions, use the chmod command. Permissions can be set using symbolic or numeric modes.

Symbolic Mode:

- **u:** Owner
- **g:** Group
- **o:** Others
- **a:** All (owner, group, and others)

- **+:** Add permission
- **-:** Remove permission
- **=:** Set permission

Examples:

chmod u+rwx filename # Add read, write, and execute permissions for the owner

chmod g-w filename # Remove write permission for the group

chmod o=rx filename # Set read and execute permissions for others

Numeric Mode: Permissions can also be represented using octal numbers:

- **r = 4**
- **w = 2**
- **x = 1**

Examples:

chmod 755 filename # Sets rwxr-xr-x (owner: rwx, group: r-x, others: r-x)

chmod 644 filename # Sets rw-r--r-- (owner: rw-, group: r--, others: r--)

Example Scenarios

Scenario 1: Create a New User and Assign to a Group

1. Create a new user named john:

sudo useradd john

2. Set the password for john:

sudo passwd john

3. Create a new group named developers:

sudo groupadd developers

4. Add john to the developers group:

sudo usermod -aG developers john

5. Verify that john is a member of the developers group:

groups john

Scenario 2: Change File Ownership and Permissions

1. Create a file named example.txt:

touch example.txt

2. Change the ownership of example.txt to john and the group to developers:

sudo chown john:developers example.txt

3. Set the permissions of example.txt to rwxr-xr-- (owner: rwx, group: r-x, others: r--):

chmod 754 example.txt

4. Verify the ownership and permissions of example.txt:

ls -l example.txt

Output:

-rwxr-xr-- 1 john developers 0 Aug 6 10:00 example.txt

Detailed Example with Commands

 # Create a new user named alice

sudo useradd alice

 # Set the password for alice

sudo passwd alice

 # Create a new group named engineers

sudo groupadd engineers

 # Add alice to the engineers group

sudo usermod -aG engineers alice

 # Verify that alice is a member of the engineers group

groups alice

 # Create a file named project.txt

touch project.txt

 # Change the ownership of project.txt to alice and the group to engineers

sudo chown alice:engineers project.txt

 # Set the permissions of project.txt to rw-rw-r--

chmod 664 project.txt

 # Verify the ownership and permissions of project.txt

ls -l project.txt

Output:

-rw-rw-r-- 1 alice engineers 0 Aug 6 10:00 project.txt

Summary

User and group management in Linux allows administrators to control access to system resources securely. The permission and ownership model provides a robust way to manage file access, ensuring that users can only perform actions they are authorized to do. Understanding these concepts and commands is essential for effective system administration.