User Management, Ownership, and Permissions Notes:

User Management

Types of Users

1. Super/Admin User
    - Super/admin user = most privileged user.
    - Not a normal user – most privileged user.
    - Non-restricted user.
    - UID = 0
    - GID = 0
    - Full name/comment: root
    - Home directory: /root
    - Shell: /bin/bash
2. System Users
    - Examples: bin, ping, daemon, system etc.
    - Interact with kernel of system.
    - UID = 1–999
    - GID = 1–999
    - Full name/comment = bin, ping, daemon, system
    - Home directory = /bin, /ping, /daemon, /system
    - Shell = /sbin/nologin
3. Regular Users
    - Examples: john, smith, tomcat, sonar etc.
    - Restricted user.
    - UID = 1000 and so on
    - GID = 1000 and so on
    - Full name/comment = John Ronald, Smith France, Tom Cat, Sonar Ate
    - Home directory = /home/john, /home/smith, /home/tomcat etc.
    - Shell = /bin/bash

Configuration Files

1. /etc/passwd → File contains all user details except the password details.
2. /etc/shadow → File contains the password details and password setting details.
3. /etc/group → File contains group details and user assigned to group.

Commands

1. Create a new user `useradd`

```
john
```

2.Create a password for user (set/reset)

```
passwd john
```

→ When you typing password it is not visible.

3. How to create a group

```
groupadd Blacksquad
```

4. Assign user to the group

```
usermod -G Blacksquad vikram
```

5. Create a customized user

useradd -d /home/amar -c "AMAR" -s /bin/sh amar

6. Modify the user details

```
usermod [options] username
```

7. How to set the password setting

- List all options for password setting:

```
chage -l username
```

- Set expiration warning days:

```
chage -W 7 username
```

- Set max days before password change:

```
chage -M 90 username
```

8. How to delete user & groups

```
userdel username
```

```
groupdel groupname
```

Delegating Sudo Privileges to Regular Users

- This file contains all the privileges to perform root level activities without needing the root password.
- To access /etc/sudoers, we need to use command:

```
visudo
```

Example

1. Switch to Amar account:

```
su - amar
```

2. Try useradd bejoy → shows error.
3. Try yum install git -y → shows error for privileges.

Here we need to give sudo privileges to Dilli from root user.

- Open visudo
- It's edit this line → Add new line below this → Add our new one.
- Example line:

```
dilli     ALL=(ALL)      ALL
```

- This line gives sudo privileges of all to the added user.

Save & quit with :wq

Testing sudo

1. Enter su - dilli.
2. Now try to install something or add new user.

⚠ It won't work without prefixing with sudo.

Here we need to add "sudo" before all the root level activity commands — even if we got root privilege also.

- Example:

```
sudo yum install git -y
```

- It won't work for others, because only the person added in visudo role can access the root level commands with prefix of sudo.

Permissions

Permissions are provided on any object to user, groups and others.

Types:

- read (r) → view content of files/directories
- write (w) → add/append/modify/delete content of file/directories
- execute (x) → run program/script

Permission Codes

- Read = r = 4 = 400
- Write = w = 2
- Execute = x = 1
- Highest permission = 777
- Machine takes permissions for sequential users → user, group, others.

Examples:

```
rw-    (6)
r--    (4)
r-x  (5)
```

- Default File permissions: 644
- Default Directory permissions: 755 Chmod Command

Syntax: `chmod [options] [permissions] [file/dir]`

Options:

- -R = recursive

CMDs:

```
chmod 400 sample.txt chmod
444 sample.txt chmod 666
sample.txt
chmod u=rw,g=rw,o=rw new-sample.txt chmod
u=r,g=rw,o=w new-sample.txt
chmod +x sample.sh chmod
-x sample.sh chmod -R 777
dir1 chmod -R 700 dir1
chmod -R 400 * chmod -R
400 /app/data chmod -R
400 /app/data/*
```

Note

- Use ls -ld for directories.
- -R stands for recursive.

Ownership

Change Ownership

- Change ownership is used for transferring the ownership to other user via chown command.

Syntax:

```
chown [options] [user]:[group] [file/directory]
```

CMDs:

```
chown john:john sample.txt
chown john:root new-sample.txt
chown root:root test.sh chown -
R john:john test/dir chown -R
john:john dir1 chown -R
john:john * chown -R john:john
```

```
/app/data chown -R john:john
/app/data/*
```

Notes on Ownership

- While checking permissions use ls -ld for directories.
- If we enter in vi editor, if that file has read-only permissions → use id !.
- If we type ls -lR we can see the directory tree (one inside another, recursive).