What are a user and a group?

Linux is a multi-user operating system, several people may be logged in and actively working on a given machine at the same time. So instead of everyone logging in with the same account, each one of them can have their own account. This helps in avoiding sharing credentials among various users. But there will be some data that different users logged in should be able to share and use.

Creating Users

To create users, we will be making use of the **useradd** command. The basic syntax of the command is:

useradd [options] username

For example, we want to create a user jane so that she has a home directory and can log in. If you were to issue the command:

sudo useradd jane

The user would be created, without a home directory and would be locked out of logging in.

To create the user Jane with a home directory and login, use the command:

sudo useradd -m jane

When this command is executed, it will ask for a password to create the user. Once done this command creates the user and the user's home directory to match the username.

But about that lockout issue? There are two ways you can resolve this. If you have created the user, you could issue the command:

sudo passwd sley

You will be prompted to enter and verify the new password. At this point, the user account will be unlocked and they can log in. Once the password is updated successfully, Jane now has a password to log in.

To do this all in a single step, use:

sudo useradd -m sley -p PASSWORD

Where PASSWORD is the password you set for the user Jane.

Once the user logs in, they can change their password by using the **passwd** command, entering their current password, and then entering/verifying their new password.
In some systems, only a superuser called the root can change the passwords.

Creating a Group

To create the group training use the command
sudo groupadd training
Now we want to add a new user, sley, to the group training. For this, we will use the usermod command:
sudo usermod -a -G training sley
The -a option tells **usermod** that we are appending and the **-G** option tells usermod that we are appending to the group name that follows the option.
In some systems, the user Jane may not become a part of a new group with the above command. In such a case, use the command:
sudo usermod -g training sley

Checking if the home directory is created

Let us learn how to check if the user's home directory has been created.
ls /home
To know which users are already a member of a group, use command:
grep training /etc/group

Changing User su and sudo

Used to determine name of current user
who
or to know 'who have I logged in as' we can type:
whoami

Switching User using the Terminal

switch users the terminal way, then we use the command su. To switch to sley, type:
su sley

sudo

We have been using this term sudo, what does it really do? To understand this, we have to understand the concept of a superuser. Root gets created automatically when we install ubuntu and root is the superuser. It has unrestricted access to all commands, files, directories, and resources. It can also grant and remove any permissions for other users. It can peep into your file or anybody's files but we can not peep into root's files unless we are granted permission.
So root is the superuser, it is more like the administrator with super privileges. Only root has the ability to create users and groups.
Root's privileges are:

- Full read/write/ execute privileges
- Creating or installing files or software
- Modifying files and settings
- Creating or deleting users and data

Switching user to root

To switch user to root on the terminal type:
su root
It asks for a password, Suppose we give john's password, it fails.
So what we really need to type is:
sudo su root

Changing File Permissions

- **chmod o+w filename.txt** to add write permissions for others
- **chmod -w filename.txt** to remove write permissions for everyone
- **chmod g+w filename.txt** to add write permissions for group
- **chmod o-r filename.txt** to remove read permissions for others
- **chmod +x filename.txt** to add execute permissions for all

Chmod 664 represents changing the users, groups and other permissions to rw-rw-r(respectively)

Managing or Accessing the ACL in Linux

With “getfacl” and “setfacl” on the command line, one can manage the ACL in Linux.

bullet
Viewing or checking existing ACL permission:
With “getfacl“, one can view the existing ACL permission of a file or directory.
The syntax of the command is: **$getfacl myfolder1**
•Here, myfolder1 is the directory name.

- The first three output lines display the directory's name, owner, and group.
- The next three lines contain the three ACL entries owner owing group and others, and no additional ACL permissions are set.

Setting up the Access Control Lists (ACL)

To modify ACL, use “setfacl” command.
To add permissions use “setfacl -m“.

Add permissions to some users:
$ setfacl -m “u:username:permissions” or $ setfacl -m “u:uid:permissions“

Example:
$ setfacl -m u:user1:r-x mydata

Add permissions to some groups:
$ setfacl -m “g:groupname:permissions” or $ setfacl -m “g:gid:permissions”

Example:
$ setfacl -m g:sales:r–“

Remove all extended ACL permissions:
$ setfacl -b myfolder