Useradd

useradd is a command in Linux that is used to add user accounts to your system. It is just a symbolic link to adduser command in Linux and the difference between both of them is that useradd is a native binary compiled with the system whereas adduser is a Perl script that uses useradd binary in the background. It makes changes to the following files:

/etc/passwd
/etc/shadow
/etc/group
/etc/gshadow
creates a directory for new user in /home

Syntax: useradd [options] [User_name]

Options

1) -c “comment” → Add a comment to the user.
2) -d “Home Dir” → To give a home directory path for new users, we use the following command in Linux.
3) -e “yyyy-mm-dd” → To give an expiry date for the user account.
4) -g gid → To make it belong to a specific group.
5) -G gids → Multiple Groups
6) -m → Create user’s home directory, if it doesn't Exist.
7) -M → To create a user without a home directory
8) -p “password” → To specify a password.
9) -s “path to shell” → users login shell
10) -u UID → To create a new user with a custom UID.
11) -k → create a User's Home directory using a custom skeleton.
12) -N → Create a user without a Group.

Note:

1) passwd UserName → To change a user's password.
2) Once a new user is created, its entry is automatically added to the **‘/etc/passwd‘** file. There are seven fields in an entry. These fields are:
    a) Username – The user login name is used to log into the system. It should be between 1 and 32 characters long.
    b) Password – The user password (or 'x' character) is stored in the ‘/etc/shadow‘ file in an encrypted format.
    c) User ID (UID) – Every user must have a User ID (UID), which stands for User Identification Number. By default, UID 0 is reserved for the root user, and UIDs ranging from 1 to 99 are reserved for other predefined accounts. Additionally, UIDs ranging from 100 to 999 are reserved for system accounts and groups.

d) Group ID (GID) – The primary Group ID (GID), which stands for Group Identification Number, is stored in the '/etc/group' file.
e) User Info – This field is optional and allows you to define extra information about the user, such as the user's full name. This information can be filled in using the finger command.
f) Home Directory – The absolute location of the user's home directory.
g) Shell – The absolute location of a user's shell i.e. /bin/bash.

3) By default, the 'useradd' command creates a user's home directory under the '/home' directory with the username.

adduser

- The `adduser` command is a high-level interface to `useradd`. It is more user-friendly and interactively prompts for information such as the password and user details.
- It automatically creates a home directory and sets up the user environment.

groupadd

Groups in Linux refer to the user groups. In Linux, there can be many users of a single system, (a normal user can take uid from 1000 to 60000, and one root user (uid 0) and 999 system users (uid 1 to 999)). In a scenario where there are many users, there might be some privileges that some users have and some don't, and it becomes difficult to manage all the permissions at the individual user level. So, using groups, we can group together a number of users, and set privileges and permissions for the entire group.

Syntax: groupadd [option] group_name

Note:

1) Every new group created is registered in the file "/etc/group". To verify that the group has been created, enter the command: sudo tail /etc/group. The four fields are as follows: group_name : password : group-id : list-of-members

Options:

1) -f, –force

This option forces the command to silently abort if the group with the given name already exists. If used with the -g or –gid option and the specified group id already exists, the command forcefully ignores the given group id and assigns a new and unique group id.

2) -g GID, –gid GID

This option assigns a specific numeric group id to the newly created group.

The group id (GID) should be non-negative and unique, unless explicitly created to be non-unique using the -o or –non-unique option.
If not specified, the command assigns a default group id greater than any existing group id.

3) -h, –help

Displays a help message, providing information about the groupadd command and its available options.
Useful for quickly accessing command documentation.

4) -K KEY=VALUE, –key KEY=VALUE

Overrides the default values set in the /etc/login.defs file.
Multiple -K options can be specified.
Parameters like GID_MIN and GID_MAX, defined in /etc/login.defs, can be modified using this option.

5) -o, –non-unique

Permits adding a group with a non-unique group id (GID).
Useful when you want to create groups with duplicate GIDs.

6) -p PASSWORD, –password PASSWORD

Sets an encrypted password for the group.
The password, returned by crypt(3), is visible to users and is stored in the /etc/gshadow file.
By default, the password is disabled, and it is crucial to ensure it adheres to the system's password policy.

7) -r, –system

Creates a system group.
System groups have numeric identifiers chosen within the SYS_GID_MIN-SYS_GID_MAX range, as defined in the /etc/login.defs file, instead of GID_MIN and GID_MAX.
-R CHROOT_DIR, –root CHROOT_DIR
Applies changes in the specified CHROOT_DIR directory and uses the configuration files from that directory.
Useful when managing groups within a chroot environment.

usermod

usermod command or modify user is a command in Linux that is used to change the properties of a user in Linux through the command line. After creating a user we have to sometimes change their attributes like password or login directory etc. so in order to do that we use the Usermod command. The information of a user is stored in the following files:

/etc/passwd
/etc/group
/etc/shadow
/etc/login.defs
/etc/gshadow
/etc/login.defs
When we execute the usermod command in the terminal the command makes the changes in these files itself.

Syntax: usermod [options] username

Options:

1) -c → We can add a comment field for the user account.
2) -d → To modify the directory for any existing user account.
3) -e → Using this option we can make the account expire in a specific period.
4) -g → Change the primary group for a User.
5) -G → To add a supplementary group.
6) -a → To add anyone of the group to a secondary group.
7) -l → To change the login name from tecmint to tecmint_admin.
8) -L → To lock the user account. This will lock the password so we can't use the account.
9) -m → moving the contents of the home directory from the existing home dir to the new dir.
10) -p → Use an un-encrypted password for the new password. (NOT Secured).
11) -s → Create a Specified shell for new accounts.
12) -u → Used to Assign UID for the user account between 0 to 999.
13) -U → To unlock the user accounts. This will remove the password lock and allow us to use the user account.

userdel

userdel command in Linux system is used to delete a user account and related files. This command basically modifies the system account files, deleting all the entries which refer to the username LOGIN. It is a low-level utility for removing the users.
Syntax: userdel [options] UserName

Options:

1) -f → This option forces the removal of the specified user account. It doesn't matter that the user is still logged in. It also forces the userdel to remove the user's home directory and mail spool, even if another user is using the same home directory or even if the mail spool is not owned by the specified user.
2) -r → Whenever we are deleting a user using this option then the files in the user's home directory will be removed along with the home directory itself and the user's mail spool. All the files located in other file systems will have to be searched for and deleted manually.
3) -h → This option displays a help message and exit.
4) -R → This option applies changes in the CHROOT_DIR directory and use the configuration files from the CHROOT_DIR directory.

groupdel

groupdel command is used to delete an existing group. It will delete all entries that refer to the group, modify the system account files, and it is handled by superuser or root user.
Syntax: groupdel [options] GROUP

Files:
/etc/group : It contains the account information of the Group.
/etc/gshadow : It contains the secure group account information.

Options:

1) -f –force: It is used to delete a group even if it is the primary group of a user.
2) -h –help: It displays the help message and exit.
3) -R –root: It applies the changes in the CHROOT_DIR directory. Also, it uses the configuration files from the CHROOT_DIR directory.

Switch users and sudo access:

su (Switch User):

Allows switching to another user account.
-: Switches to the root user.
-l: Simulates a full login.
Example:
su - username
su -

Difference	**su**	**su –**
Environment Variables	Retains the current user's environment variables	Resets the environment variables to those of the target user

Working Directory	Keeps the current working directory.	Changes the working directory to the target user's home directory.
Shell Settings	Retains the current user's shell settings.	Resets the shell settings to those of the target user.
Path Variable	The target user's PATH variable is not updated	The target user's PATH variable is updated to include the user-specific directories.

sudo (Superuser Do):

Execute a command with elevated privileges.
-u: Run the command as a specific user.
-s: Run a shell as another user.
Example:
sudo command
sudo -u username command
sudo -s -u username

Monitor User Activity:-

who command is used to find out the following information :
1. Time of last system boot
2. Current run level of the system
3. List of logged in users and more.

Description : The who command is used to get information about currently logged in user on to system.

Syntax : $who [options] [filename]

The who command displays the following information for each user currently logged in to the system if no option is provided :

1. Login name of the users
2. Terminal line numbers
3. Login time of the users in to system
4. Remote host name of the user

Options:

1) a → all info
2) H → prints the header
3) b → last boot time of the system
4) l → system login process
5) m → hostname and user associated with stdin
6) r → print current run level.

last

The last command in Linux is used to display the list of all the users logged in and out since the file `/var/log/wtmp` was created. One or more usernames can be given as an argument to display their login in (and out) time and their host-name.

Syntax: last [options] [username...] [tty...]

Options:

1) -n → last n entries
2) -R → To hide hostField Name
3) -F → The `-F` option can be used to display the login and logout times with their corresponding dates.
4) -s -t → The `-s` (since) and `-t` (until) options allow us to display login entries within a specific time period.
   EX: last -s yesterday -t today

w

The ‘w’ command in Linux gives us important information about who is currently using the computer, how much the computer is being used, and what programs are running. It’s a handy tool for people who take care of computer systems, as it helps them keep an eye on what users are doing, how much of the computer’s power is being used, and how to make everything run smoothly.
Syntax: w [options] user [...]

Columns	Description
USER	Displays the username of the logged-in user.
TTY	Shows the terminal device associated with the user session.
FROM	Indicates the remote host or IP address the user is connected to
LOGIN@	Displays the time at which the user logged in.
IDLE	Shows the duration of inactivity since the user's last interaction.
JCPU	Represents the CPU time used by all processes attached to the user's session.
PCPU	Displays the percentage of CPU time used by the user's current process.
WHAT	Provides information about the command or process running in the user's session.

id

The id command in Linux is used to find out user and group names and numeric IDs (UID or group ID) of the current user or any other user in the server. This command is useful to find out the following information as listed below:

User name and real user id.
Find out the specific Users UID.
Show the UID and all groups associated with a user.
List out all the groups a user belongs to.
Display security context of the current user.

Syntax: id [OPTION]… [USER]

Options:

-g : Print only the effective group id.
-G : Print all Group ID's.
-n : Prints name instead of number.
-r : Prints real ID instead of numbers.
-u : Prints only the effective user ID.
–help : Display help messages and exit.
–version : Display the version information and exit.

Some Questions related to Users and groups permissions

1) How to add a new group?
2) How to add a new user?
3) Switching from one user to another user
4) How to get information about a particular user?
5) How to delete users?
6) How to delete a group ?
7) How to modify user info
8) How to modify group info
7) How to change ownership of a file?
8) How to change group membership of a file?
9) How to change the group of a user ?
10) Add a User to Multiple Groups
11) How to check available groups?
12) How to change the password of a user?
13) What is the difference between adduser and useradd in Linux?
14)sudo command
15)sudoers file
16)sudo command
17)sudoers file
18) How to check the allowed commands by sudo for a particular user ?