**Day-6 Task: File Permissions and Access Control Lists**

#owner – vishal saxena  
Day 6 Challenge

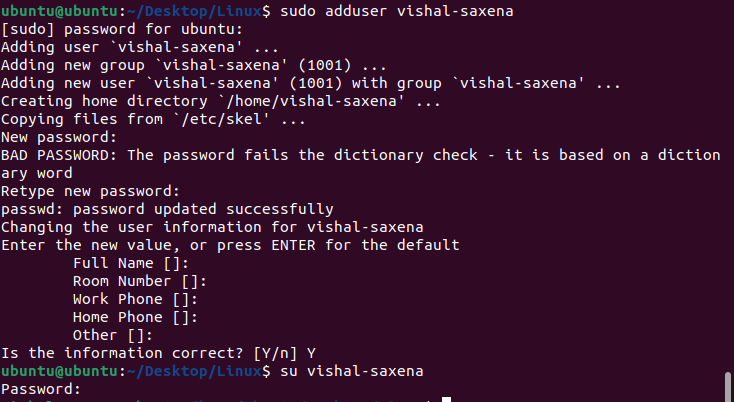
1. Create a simple file and do ls -ltr to see the details of the files.

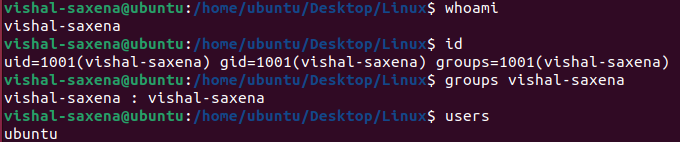
In the below Screen Shots I have created a user using the following command .

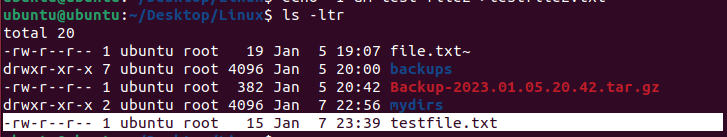
**sudo adduser vishal-saxena**   
   
# {I have mentioned some question related to the above commeand which comes in my mind and I have explained the same below

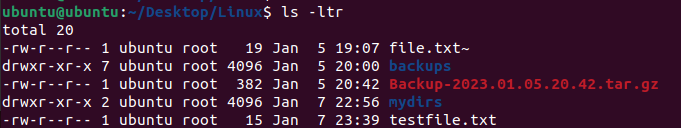
**adduser** and **useradd** commands that are used to create new user accounts in Linux. Both commands are similar, but there are some differences between them.

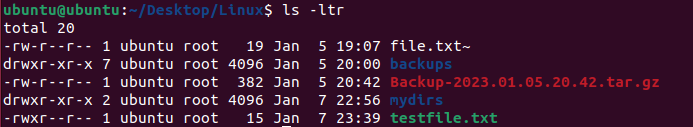
The **adduser** command is a user-friendly front-end to the **useradd** command, and is usually found in Debian-based systems such as Ubuntu. It is designed to be easier to use than **useradd**, and it has a simpler syntax.}











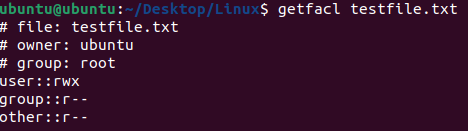
**ACL command**

Access Control Lists (ACLs) are a mechanism used in Linux to specify fine-grained permissions for files and directories beyond the standard read, write, and execute permissions. ACLs allow you to specify permissions for individual users and groups, and they can be used to grant or deny access to specific files or directories.

ACLs are specified using entries that specify a user or group and the permissions that are granted or denied to that user or group. There are several types of ACL entries, including user entries, group entries, mask entries, and default entries.

User entries specify the permissions for a specific user. Group entries specify the permissions for a specific group. Mask entries specify the maximum permissions that can be granted to a user or group. Default entries specify the default permissions for new files and directories in a directory.

Here is an example of an ACL entry:

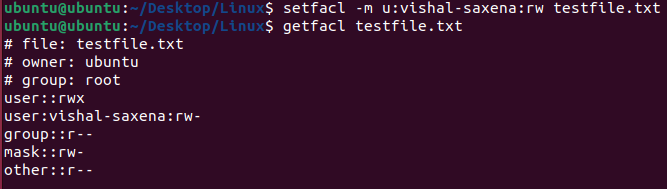


user:vishal-saxena:rw-

This entry grants the user **vishal-saxena** read and write permissions, but not execute permission.

**command below**

﻿**setfacl -m u:vishal2:rw testfile.txt**



**Q. Difference between passwd and gpasswd command**

The **passwd** command is used to change a user's password, while the **gpasswd** command is used to manage group passwords.

The **passwd** command is used to change the password of a user account, as well as to set or change the password of the **root** user (i.e., the superuser). The **passwd** command can also be used to check the aging information for a user account, or to lock or unlock a user account.

Here is the basic syntax for the **passwd** command:

**passwd [options] [user]**

Here is a description of the options used in the **passwd** command:

* **-a**: This option specifies the password hashing algorithm to use.
* **-d**: This option deletes the password for the user account, effectively disabling the account.
* **-e**: This option forces the password to expire, requiring the user to change the password at the next login.
* **-l**: This option locks the user account, preventing the user from logging in.
* **-u**: This option unlocks the user account, allowing the user to log in.
* **-S**: This option displays the status of the user account (e.g., whether the account is locked or has a password set).
* **user**: This is the name of the user whose password you want to change or manage. If no user is specified, the password of the current user is changed.

Here is an example of how to use the **passwd** command to change the password of the user **vishal-saxena2**:

**sudo passwd vishal-saxena2**

The system will prompt you to enter the new password for the user **vishal-saxena2** and confirm it by entering it again when prompted.

On the other hand, the **gpasswd** command is used to manage group passwords. It can be used to set or change the password for a group, or to add or remove users from a group that has a password.

Here is the basic syntax for the **gpasswd** command:

gpasswd [options] group

Here is a description of the options used in the **gpasswd** command:

* **-a user**: This option adds the specified user to the group.
* **-d user**: This option removes the specified user from the group.
* **-m user**[,user,...]: This option sets the specified users as members of the group.
* **-R:** This option removes all users from the group.
* **-r:** This option removes the password for the group, effectively disabling the group.
* **group:** This is the name of the group whose password you want to manage.

Here is an example of how to use the gpasswd command to set a password for the group developers:

**sudo gpasswd -A vishal-saxena2 developers**

This will set a password for the group developers and add the user vishal-saxena2 to the group. The system will prompt you to enter a new password for the group and confirm it by entering it again when prompted.

Keep in mind that you need to have administrative privileges (e.g., by using sudo)