

 Menu Menu

# LFCA: Learn User Account Management – Part 5

James Kiarie | Last Updated: March 26, 2021 | LFCA, Linux Certifications | 5 Comments

As a Linux system administrator, you will be tasked with ensuring the smooth flow of all IT operations in your organization. Given that some IT operations are intertwined, a systems administrator usually wears many hats including being a database or network administrator.

This article is **Part 5** of the [LFCA series](#), here in this part, you will acquaint yourself with the general system administration commands to create and manage users in a Linux system.

## User Account Management in Linux

One of the primary responsibilities of a Linux systems administrator is to create and manage users in a Linux system. Each user account has 2 unique identifiers: the **username** and the **User ID (UID)**.

Essentially, there are 3 main categories of users in Linux:

## Root User

---

The **root** user is the most powerful user in a Linux system and is usually created during the installation process. The root user has absolute power in the Linux system or any other UNIX-like OS. The user can access all the commands, files, and directories and modify the system to their preference.

The root user can update the system, install and uninstall packages, add or remove other users, grant or revoke permissions, and perform any other [system administration task](#) without any restrictions.

The root user can just about do anything on the system. The assumption by Linux and UNIX-like systems is that you know full well what you are doing with the system. That said, the root user can easily break the system. All it takes is for you to execute a fatal command, and the system will be up in smoke.

For this reason, running commands as the root user is highly discouraged. Instead, good practice demands that you should [configure a sudo user](#). That is grant sudo privileges to a regular user to perform certain administrative tasks and restrict some tasks only to the root user.

## Regular User

---

A regular user is a normal login user that can be created by a systems administrator. Usually, there is a provision to create one during the installation process. However, you can still create as many regular users as needed post-installation.

A regular user can only perform tasks and access files and directories for which they are authorized. If need be, a regular user can be granted elevated privilege



perform administrative-level tasks. Regular users can also be deleted or disabled when the need arises.

## Service Account

---

This is a non-login account that is created when a software package is installed. Such accounts are used by services to execute processes in the system. They are not designed or intended to carry out any routine or administrative tasks in the system.

## User Management Files

---

Information about users in a Linux system is stored in the following files:

- The **/etc/passwd** file
- The **/etc/group** file
- The **/etc/gshadow** file
- The **/etc/shadow** file

Let's understand each file and what it does:

### The **/etc/passwd** File

---

The **/etc/passwd** file contains quite a bit of information about users which is contained in various fields. To view the contents of the file, simply use the [cat](#) [command](#) as shown.

```
$ cat /etc/passwd
```

Here's a snippet of the output.

```
tecmint:x:1002:1002:tecmint,,,:/home/tecmint:/bin/bash
```

Let's focus on the first line and flesh out the various fields. Starting from the far left, we have the following:

- The **username**: This is the name of the user, in this case, tecmint.
- The **Password**: The second column represents the encrypted password of the user. The password is not printed in plain text, instead, a placeholder with an **x** sign is used.
- The **UID**: This is the User **ID**. It's a unique identifier for every user.
- The **GID**: This is the **Group ID**.
- A brief description or summary of the user.
- This is the path to the user's home directory. For tecmint user, we have **/home/tecmint**.
- This is the Login shell. For regular login users, this is usually represented as **/bin/bash**. For service accounts such as SSH or MySQL, this is usually represented as **/bin/false**.

## The /etc/group File

---

This file contains information about the user groups. When a user is created, the shell automatically creates a group that corresponds to the username of the user. This is known as the primary group. The user is added to the primary group upon creation.

For example, if you create a user called **bob**, the system automatically creates a group called **bob** and adds the user **bob** to the group.

```
$ cat /etc/group  
  
tecmint:x:1002:
```

The **/etc/group** file has 3 columns. From the far left, we have:

- Group name. Each group name must be unique.
- Group password. Usually represented by an **x** placeholder.
- Group ID (GID)



- Group members. These are members that belong to the group. This field is left blank if the user is the only member in the group.

**NOTE:** A user can be a member of multiple groups. Likewise, a group can have multiple members.

To confirm the groups that a user belongs to, run the command:

```
$ groups username
```

For example, to check the groups that the user **tecmint** belongs to, run the command:

```
$ groups tecmint
```

The output confirms that the user belongs to two groups: **tecmint** and **sudo**.

```
tecmint : tecmint sudo
```

## The /etc/gshadow File

---

This file contains encrypted or '**shadowed**' passwords for group accounts and, for security reasons, cannot be accessed by regular users. It's only readable by the root user and users with sudo privileges.

```
$ sudo cat /etc/gshadow
```

```
tecmint:!::
```



From the far left, the file contains the following fields:

- Group name
- Encrypted Group password
- Group admin
- Group members

## The /etc/shadow File

The **/etc/shadow** file stores the users actual passwords in a hashed or encrypted format. Again, the fields are colon-separated and take the format shown.

```
$ sudo cat /etc/shadow
```

```
tecmin:$6$iavr8PAxxnWmfh6J$iJeuHeo5drKWcXQ.BFGUruk4JWW7j4cwjX7ul
```

The file has 9 fields. Starting from the far left we have:

- The **username**: This is your login name.
- The **user's password**. This is presented in a hashed or encrypted format.
- The **last password change**. This is the date since the password was changed and is calculated since the epoch date. Epoch is the 1st January 1970.
- The **minimum password age**. This is the minimum number of days that must elapse before a password can be set.
- The **maximum password age**. This is the maximum number of days after which a password must be changed.
- The **warning period**. As the name suggests, this is the number of days shortly before a password expires that a user is notified of the impending password expiry.
- The **inactivity period**. The number of days after a password expires that a user account is disabled without the user changing the password.
- The **expiration date**. The date when the user account expired.
- **Reserved field**. – This is left blank.



## How to Add Users in a Linux System

For **Debian** and **Ubuntu** distributions, the **adduser** utility is used for adding users.

The syntax is quite simple and straightforward.

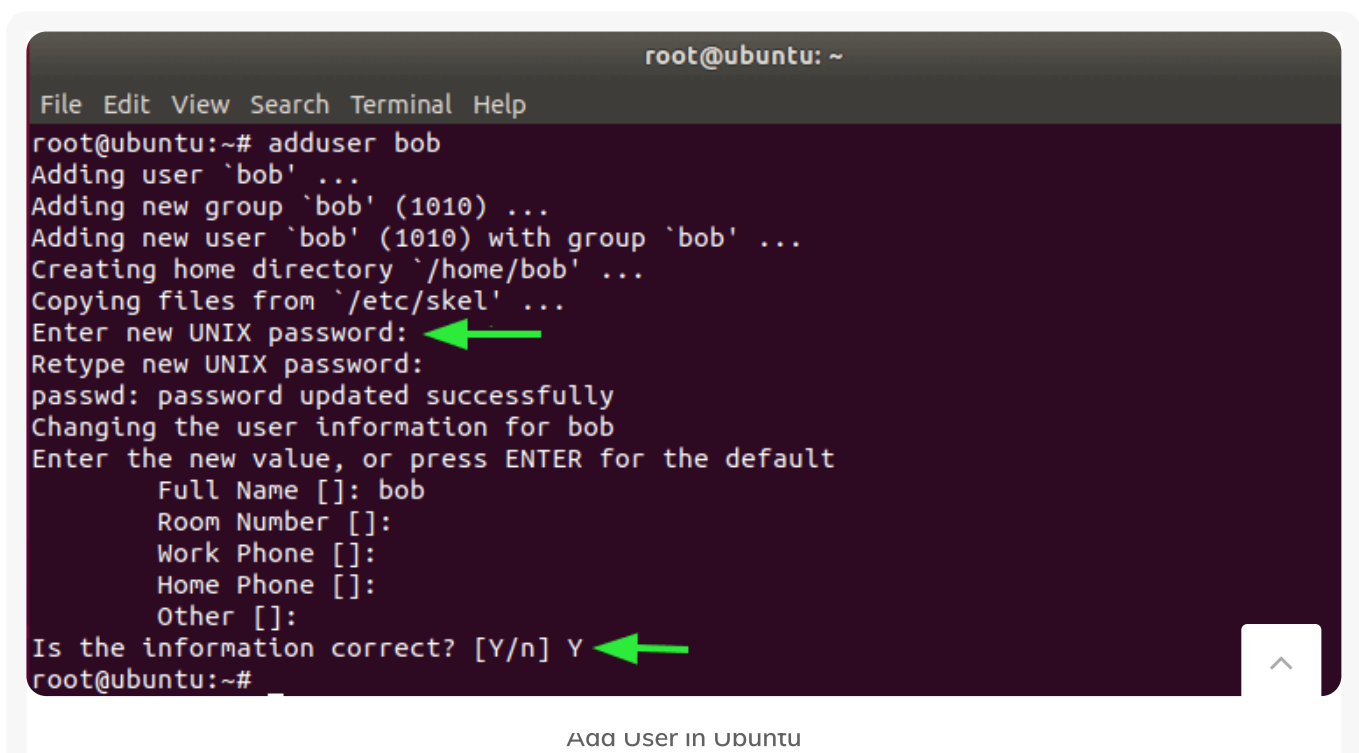
```
# adduser username
```



For example, to add a user called **bob**, run the command

```
# adduser bob
```

From the output, a user called '**bob**' is created and is added to a newly created group called '**bob**'. Additionally, the system also creates a home directory and copies configuration files into it.

Thereafter, you will be prompted for the new user's password and then confirm it. The shell will also prompt you for the user's full name and other optional information such as Room no and Work phone. This information is not really necessary, so it's safe to skip it. Finally, press '**Y**' to confirm that the information provided is correct.



```
root@ubuntu: ~  
File Edit View Search Terminal Help  
root@ubuntu:~# adduser bob  
Adding user `bob' ...  
Adding new group `bob' (1010) ...  
Adding new user `bob' (1010) with group `bob' ...  
Creating home directory `/home/bob' ...  
Copying files from `/etc/skel' ...  
Enter new UNIX password:   
Retype new UNIX password:  
passwd: password updated successfully  
Changing the user information for bob  
Enter the new value, or press ENTER for the default  
  Full Name []: bob  
  Room Number []:  
  Work Phone []:  
  Home Phone []:  
  Other []:  
Is the information correct? [Y/n] Y   
root@ubuntu:~#
```

Add User in Ubuntu

For **RHEL & CentOS-based** systems, use the [useradd command](#).

```
# useradd bob
```

Next, set the password for the user using the **passwd** command as follows.

```
# passwd bob
```

```
[root@centos-8 ~]#  
[root@centos-8 ~]# useradd bob  
[root@centos-8 ~]#  
[root@centos-8 ~]#  
[root@centos-8 ~]# passwd bob  
Changing password for user bob.
```

RedHat RHCSA and RHCE Certification Preparation Guide -

[Get This Book](#)

```
[root@centos-8 ~]#
```

Add User in CentOS

## How to Delete Users in a Linux System

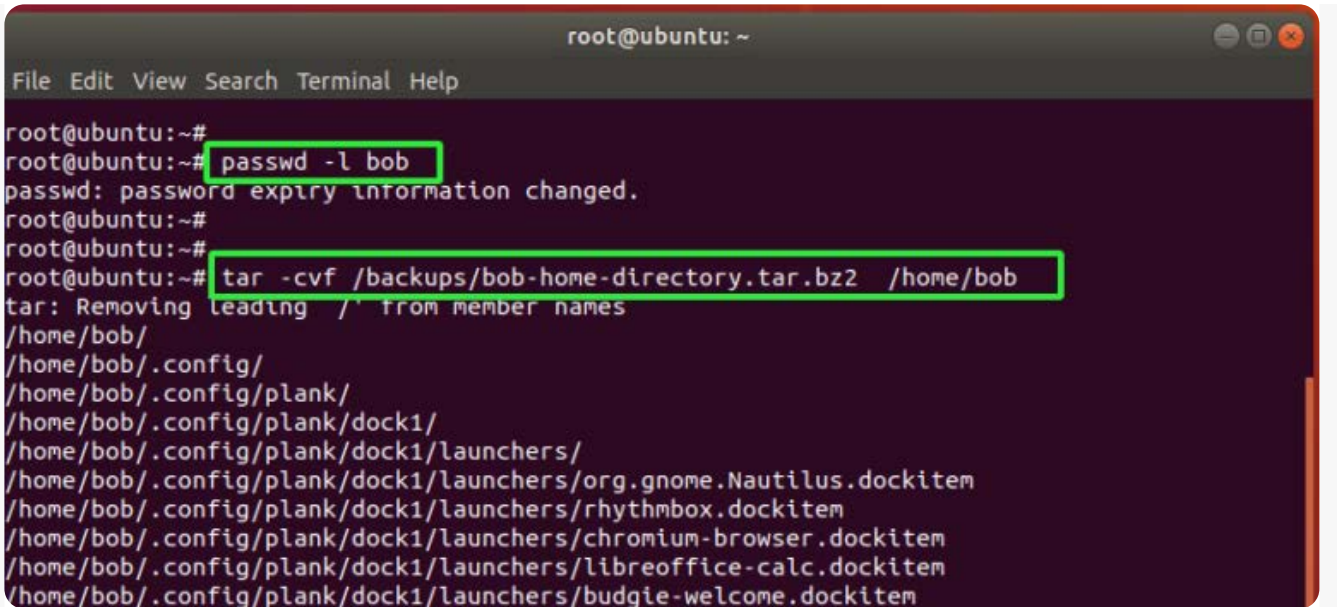
To delete a user from the system, it's advisable to first lock the user from logging into the system as shown.

```
# passwd -l bob
```

If you wish, you can backup the user's files using the [tar command](#).

```
# tar -cvf /backups/bob-home-directory.tar.bz2 /home/bob
```



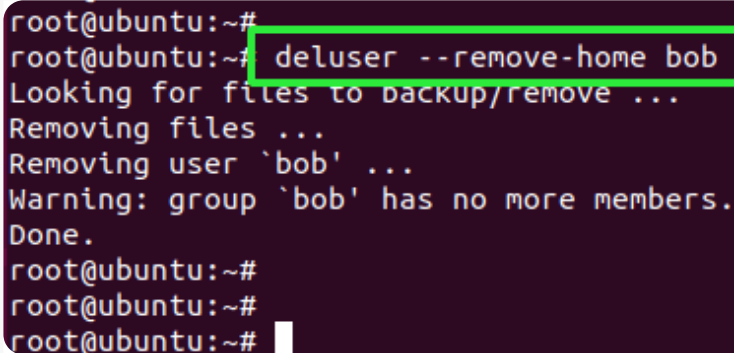


```
root@ubuntu: ~  
File Edit View Search Terminal Help  
root@ubuntu:~#  
root@ubuntu:~# passwd -l bob  
passwd: password expiry information changed.  
root@ubuntu:~#  
root@ubuntu:~#  
root@ubuntu:~# tar -cvf /backups/bob-home-directory.tar.bz2 /home/bob  
tar: Removing leading '/' from member names  
/home/bob/  
/home/bob/.config/  
/home/bob/.config/plank/  
/home/bob/.config/plank/dock1/  
/home/bob/.config/plank/dock1/launchers/  
/home/bob/.config/plank/dock1/launchers/org.gnome.Nautilus.dockitem  
/home/bob/.config/plank/dock1/launchers/rhythmbox.dockitem  
/home/bob/.config/plank/dock1/launchers/chromium-browser.dockitem  
/home/bob/.config/plank/dock1/launchers/libreoffice-calc.dockitem  
/home/bob/.config/plank/dock1/launchers/budgie-welcome.dockitem
```

Lock User Account in Linux

Finally, to delete the user together with the home directory use the **deluser** command as follows:

```
# deluser --remove-home bob
```



```
root@ubuntu:~#  
root@ubuntu:~# deluser --remove-home bob  
Looking for files to backup/remove ...  
Removing files ...  
Removing user 'bob' ...  
Warning: group 'bob' has no more members.  
Done.  
root@ubuntu:~#  
root@ubuntu:~#  
root@ubuntu:~#
```

Delete User in Linux

Additionally, you can use the **userdel** command as shown.

```
# userdel -r bob
```

The two commands completely remove the user alongside their home directory

^

## Conclusion

That was an overview of user management commands that will prove useful especially when managing user accounts in your office environment. Give them a try from time to time to sharpen your system administration skills.

Become a Linux Foundation Certified IT Associate (LFCA)

🔑 [LFCA Certification Exam](#)

< [10 Best Udemy Google Cloud Platform Courses in 2021](#)

[LFCA: Learn to Manage Time and Date in Linux – Part 6](#) >

If you liked this article, then do [subscribe to email alerts](#) for Linux tutorials. If you have any questions or doubts? do [ask for help in the comments](#) section.

### If You Appreciate What We Do Here On TecMint, You Should Consider:

TecMint is the fastest growing and most trusted community site for any kind of Linux Articles, Guides and Books on the web. Millions of people visit TecMint! to search or browse the thousands of published articles available FREELY to all.

If you like what you are reading, please consider buying us a coffee ( or 2 ) as a token of appreciation.

