Adding Users

If you are signed in as the root user, you can create a new user at any time by typing:

adduser username

If you are signed in as a non-root user who has been given sudo privileges, as demonstrated in the next section of this tutorial, you can add a new user by typing:

sudo adduser username

Next, you'll need to give your user a password so that they can log in. To do so, use the passwd command:

passwd username

**Note:** Remember to add sudo ahead of the command if you are signed in as a non-root user with sudoprivileges.

You will be prompted to type in the password twice to confirm it. Now your new user is set up and ready for use! You can now log in as that user, using the password that you set up.

Granting Sudo Privileges to a User

If your new user should have the ability to execute commands with root (administrative) privileges, you will need to give the new user access to sudo.

We can do this by adding the user to the wheel group (which gives sudo access to all of its members by default) through the gpasswd command. This is the safest and easiest way to manage sudo user rights.

If you are currently signed in as the root user, type:

gpasswd -a username wheel

If you are signed in using a non-root user with sudo privileges, type this instead:

sudo gpasswd -a username wheel

Now your new user is able to execute commands with administrative privileges. To do so, simply type sudo ahead of the command that you want to execute as an administrator:

sudo some\_command

You will be prompted to enter the password of the regular user account that you are signed in as. Once the correct password has been submitted, the command you entered will be executed with rootprivileges.

**Managing Users with Sudo Privileges**

While you can add and remove users from a group (such as wheel) with gpasswd, the command doesn't have a way to show which users are members of a group. In order to see which users are part of the wheel group (and thus have sudo privileges by default), you can use the lid function. lid is normally used to show which groups a user belongs to, but with the -g flag, you can reverse it and show which users belong in a group:

sudo lid -g wheel

The output will show you the usernames and UIDs that are associated with the group. This is a good way of confirming that your previous commands were successful, and that the user has the privileges that they need.

Deleting Users

If you have a user account that you no longer need, it's best to delete the old account. You have a couple of methods to do so, though the choice of which method to use depends on your own situation.

If you want to delete the user without deleting any of their files, type this command as root:

userdel username

If you want to delete the user's home directory along with the user account itself, type this command as root:

userdel -r username

create a file which contains all the user name. Something like this:

users.txt

nurealam

nayeem

mrahman

farid

rubi

sankar

Save the file as *userlist.txt*.

Now create the following bash file:

Creater\_user.sh

#!/bin/sh

for i in `more userlist.txt `

do

echo $i

adduser $i

done

Save the file and exit.

*$chmod 755 userlist.txt*

*$chmod 777 create\_user.sh*

*$./create\_user.sh : TO EXECUTE SHELL SCRIPT*

Now run the file:

*./userlist.txt*

This will add all the users to the system. Now we have to change the passwords. Let's say we want *username123* as password. So for user *nayeem*the password will be *nayeem123*, *rubi123* for user *rubi* and so on.

Create another bash file as follows:

#!/bin/sh

for i in `more userlist.txt `

do

echo $i

echo $i"123" | passwd –-stdin "$i"

echo; echo "User $username’s password changed!"

done

Run the file. All the passwords are changed.