**Purpose:**

Often it is required to change the user to something with more or less privileges.

| **To become New User** | **Enter in Bash:** | **privileges and file/folder access** |
| --- | --- | --- |
| root | su | The su command substitutes the current user in use by the system in the shell. You can switch to any user by taking su and adding a username by it. This will tell the system to switch (and essentially log out of) the current user to the one specified. Alternatively, the su command can gain root access by entering su without specifying anything after the command.  The root user’s password has to be known and used to log in. |
| root | sudo su - | Though there isn’t very much difference from “su”, root given with sudo su is by requesting the current user’s password, not the root password |
| root | sudo -i | A much cleaner and preferred way of gaining root and a root environment without directly interacting with the root user. The environment of the current user is not transferred |
| root | sudo -s | The command gains root but will not change the user or the user environment  This command is best used when the user doesn’t want to touch root at all and just wants a root shell for easy command execution. |
| postgres | sudo su postgres | Switch to superuser postgres |
| jetty | sudo su jetty |  |
| etc |  |  |

The following is from <https://www.booleanworld.com/how-to-add-remove-and-modify-users-in-linux/> . In this article, we will see how to manage users in Linux by adding new users, modifying existing users and delete user accounts which are not required.

To execute most of the commands in this article, you need root access. The distribution we’re using to show these examples supports sudo, so we’ll add sudo before the command to execute the command as root. If you don’t have sudo installed, then you may either log in as the root user into your system, or you may run su and enter the root password to gain root access.

The commands we’ve discussed here work across all Linux distributions, including Debian, Ubuntu, CentOS and RHEL.

**Add a user in Linux**

To add users, run the useradd command, like so:

sudo useradd -m <name of the user>

For example, if you want to add the user named john, then the command will be like:

sudo useradd -m john

By default, useradd creates a user without creating a home directory. So, to make useradd create a home folder, we’ve used the -m switch.

If the command is successful, it won’t have any output, like so:

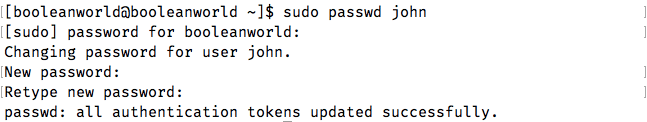
Adding a user in Linux

Behind the scenes, it automatically creates the user john by assigning a unique user ID for the user, and adding the user’s details to the /etc/passwd file. It also creates a home directory for the user under /home (so the full path is /home/john).

At this point, the user has been created, but they don’t have a password and can’t log in. So, to assign a password to the newly created user, run the passwd command like so:

sudo passwd <username>

The command will ask for the new password, and ask you to confirm it:



This command adds the user’s password in /etc/shadow in an encrypted format. After running this command, the new user should be able to login as usual.

You can view the new user’s ID by using id -u <username>. In our case, john was given an ID of 1001:

https://www.booleanworld.com/wp-content/uploads/2019/11/linux-users-3.png

**User Groups in Linux**

Before we move on creating and managing “groups”, let us review what a group is.

In Linux, groups are a collection of users. A group can have zero or more users assigned to it. Just like users, each group has its own “group name” and a unique “group ID”. Groups are used to assign users to a set of permissions, access or privileges.

There are two types of groups:

1. **Primary Group:** When a Linux user is created, it is automatically assigned to a single, default group, known as the “primary group”. Usually, the name of the primary group is the same as the user’s username, although you can change this if you want.
2. **Supplementary Group:** Apart from the primary group, you can add a user to other groups. These other groups to which a user belongs are called supplementary groups.

Let us understand this with an example. Previously, when we created the user john, a group named john was automatically created as well. This group is the primary group, and the user john was added to the group john.

Later, you can add john to another group, such as sales. Now, with respect to the user john, sales is a supplementary group.

Information about all groups on your system is stored in /etc/group. Groups can also have passwords, although they’re rarely used because it requires everyone in the group to know a common password. If a group has a password, it is stored in /etc/gshadow.

**Create a new Group**

To create a new group in Linux, run the groupadd command, like so:

sudo groupadd <name of the group>

For example, to add a group named sales to your system, use the command:

sudo groupadd sales

Just like the useradd command groupadd also doesn’t show any output if the user was successfully created:

https://www.booleanworld.com/wp-content/uploads/2019/11/linux-users-4.png

If you want to verify that the group was indeed created, you can view the file /etc/groups.

Previously, we’ve mentioned that group passwords are rarely ever used. However, if you do want to assign a group password, you can use the gpasswd command:

sudo gpasswd sales

Enter the group password and confirm it by typing it again. The group password is set once you complete this process.

**View a user’s groups and user ID**

To see a user’s information, such as a user’s ID and the groups they belong to, you can use the id command. To see your own user’s information, simply type:

id

The output shows your user’s ID (uid) and primary group’s ID (gid), as well as a list of primary and supplementary groups you belong to. For example, in the output below, the user booleanworld belongs to the groups booleanworld and wheel.

Viewing the user information of your own user account.

On the other hand, if you want to view the information for a different user, use the following command:

id <username>

It lists the user ID, primary group ID, as well as the associated names of the groups and their IDs:

Viewing a different user's user ID, group ID and groups.

Otherwise, if you only want to see the groups to which a user belongs, you can use the groups command. It’s similar to id, and by default it lists your own groups by default. For example, once again we can see that the current user, booleanworld belongs to the groups booleanworld and wheel.

Viewing groups with the

If you want to see the group of another user, use:

groups <username>

In the example below, we’re using it to list the groups john belongs to:

Viewing a different user's groups.

**Add a User to a Group**

Now that you have an idea about groups, we can now modify a user and assign them to groups. To add a user to a group, use the following command:

sudo usermod -a -G <group name> <user name>

For example, to assign the user John to the group sales, you should run:

sudo usermod -a -G sales john

Here, the -a flag “adds” the user to the group and the -G signifies that we’re adding them to a supplementary group (as opposed to changing their primary group).

If you want to change the user’s primary group instead, you can use the -g flag like so:

sudo usermod -g <primary group name> <username>

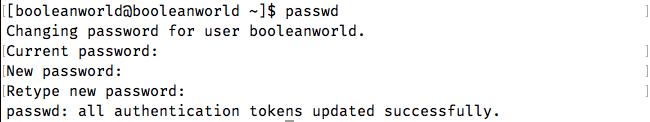
In the above command, notice that we aren’t using the -a append flag. This is because, we want to simply change the primary group of the user, and we’re not adding the user to a group. By its very definition, a primary group can only have one user.

**Change Password of a User**

Previously, when we created a new user, we’ve used the passwd command to assign a password to the new user. You can also use this to change passwords. If you want to change your own password, simply run:

passwd

When you change your own password, it’ll ask you for your current password. Once you enter it correctly, you will be asked to enter your new password twice.



You can also use it to change a different by providing the username, although you need to be root to change passwords for others. The syntax for changing another user’s password is:

sudo passwd <username>

When you are the root user, passwd doesn’t ask you for your current password — it’ll just ask you for the new password.

You can also use passwd to prevent a user from logging in (aka “locking out the user”), using the -l switch. For example, if you want to prevent john from logging in, you can use:

sudo passwd -l john

**Grant Sudo Permissions to Users**

sudo is a utility to allow users to execute commands as another user, usually the root user. In most distributions, only a certain set of users can execute sudo.

For example, in Debian and its derivatives (like Ubuntu), users in the sudo group can use the sudo command. Similarly, CentOS or RHEL has a wheel group that does the same thing.

If you want a user (say john) to be able to use sudo, you can use usermod to add them to the sudo group like so:

sudo usermod -a -G sudo john

For CentOS or RHEL use:

sudo usermod -a -G wheel john

What if you don’t use something based on Debian or CentOS? Although the default sudo configuration can vary a lot between distributions, the steps below should help you get started.

First, you should create your own group, such as sysadmins and add users to it, just like we’ve done previously. Then, you can edit the file /etc/sudoers as a root user to allow anyone who belongs to sysadmins to have sudo access. To edit the file, you can use an editor like nano or vi by running:

sudo nano /etc/sudoders # if you have 'nano' installed

sudo vi /etc/sudoers # if you have 'vi' installed

Now, go to the end of the file, and add the following text on its own line. This will allow anyone belonging to sysadmins to use sudo:

%sysadmins  ALL=(ALL)       ALL

Save the file and exit the editor. After this, any users in sysadmins would be able to use sudo to run commands.

**Delete a User in Linux**

To delete a user in Linux, you can use userdel command, like so:

sudo userdel <username>

By default, this command preserves the home directory and some other special files, such as the user’s list of cron jobs. If you want to delete these files as well, you should use the --remove-all-files flag.

For example, if you want to delete the user john along with the home directory of john, use the following command:

sudo userdel -r john