

 Menu Menu

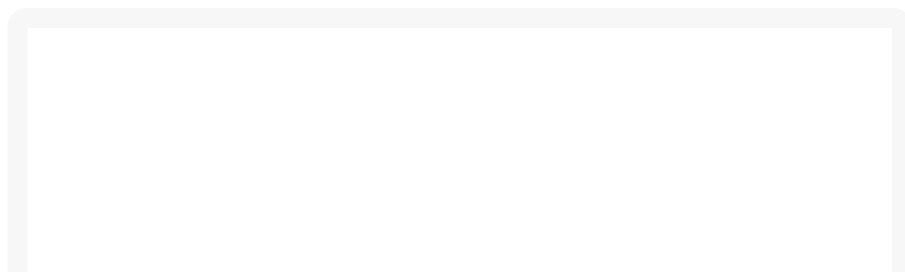
The Complete Guide to “useradd” Command in Linux – 15 Practical Examples

Ravi Saive | Last Updated: August 11, 2021 | Linux Commands | 119 Comments

We all are aware of the most popular command called ‘**useradd**’ or ‘**adduser**’ in Linux. There are times when a Linux System Administrator is asked to create user accounts on Linux with some specific properties, limitations, or comments.

[You might also like: [How to Create a Shared Directory for All Users in Linux](#)]

In Linux, a ‘**useradd**’ command is a low-level utility that is used for adding/creating user accounts in **Linux** and other **Unix-like** operating systems. The ‘**adduser**’ is much similar to the **useradd** command because it is just a symbolic link to it.





In some other Linux distributions, the **useradd** command may come with a slightly different version. I suggest you read your documentation, before using our instructions to create new user accounts in **Linux**.

When we run the '**useradd**' command in the Linux terminal, it performs the following major things:

- It edits **/etc/passwd**, **/etc/shadow**, **/etc/group** and **/etc/gshadow** files for the newly created user accounts.
- Creates and populates a home directory for the new user.
- Sets permissions and ownerships to the home directory.

Useradd Command Syntax

The Basic syntax of the **useradd** command is:

```
# useradd [options] username
```



In this article, we will show you the most used **15 useradd commands** with their practical examples in **Linux**. We have divided the section into two parts from **Basic** to **Advance** usage of the command.

- **Part I:** Basic Useradd Commands with **10** examples
- **Part II:** Advance Useradd Commands with **5** examples

Part I – 10 Basic Usage of useradd Commands

1. How to Add a New User in Linux

To add/create a new user, you've to follow the command '**useradd**' or '**adduser**' with '**username**'. The '**username**' is a user login name, that is used by a user to login into the system.

Only one user can be added and that username must be unique (different from other usernames already exists on the system).

For example, to add a new user called '**tecmint**', use the following command.

```
[root@tecmint ~]# useradd tecmint
```

When we add a new user in Linux with the '**useradd**' command it gets created in a locked state and to unlock that user account, we need to set a password for that account with the '**passwd**' command.

```
[root@tecmint ~]# passwd tecmint
Changing password for user tecmint.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```



```
[root@tecmint:~]# useradd tecmint
[root@tecmint:~]# passwd tecmint
Changing password for user tecmint.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@tecmint:~]# _
```

Create User in Linux

Once a new user is created, its entry is automatically added to the `/etc/passwd` file. The file is used to store the user's information and the entry should be.

```
tecmint:x:1000:1000:tecmint:/home/tecmint:/bin/bash
```

```
[root@tecmint:~]# cat /etc/passwd | grep tecmint
tecmint:x:1000:1000::/home/tecmint:/bin/bash
[root@tecmint:~]# _
```

view User info in Linux

The above entry contains a set of seven colon-separated fields, each field has its own meaning. Let's see what are these fields:

- **Username:** User login name used to login into the system. It should be between 1 to 32 characters long.
- **Password:** User password (or x character) stored in `/etc/shadow` file in encrypted format.
- **User ID (UID):** Every user must have a User ID (UID) User Identification Number. By default, **UID 0** is reserved for the root user and UID's ranging from **1-99** are reserved for other predefined accounts. Further UID's ranging from **100-999** are reserved for system accounts and groups.
- **Group ID (GID):** The primary Group ID (GID) Group Identification Number stored in the `/etc/group` file.
- **User Info:** This field is optional and allows you to define extra information about the user. For example, user full name. This field is filled by the 'finger' command.
- **Home Directory:** The absolute location of the user's home directory.
- **Shell:** The absolute location of a user's shell i.e. `/bin/bash`.

^

2. Create a User with Different Home Directory

By default **'useradd'** command creates a user's home directory under **/home** directory with a username. Thus, for example, we've seen above the default home directory for the user **'tecmint'** is **'/home/tecmint'**.

However, this action can be changed by using the **'-d'** option along with the location of the new home directory (i.e. **/data/projects**). For example, the following command will create a user **'anusha'** with a home directory **'/data/projects'**.

```
[root@tecmint ~]# useradd -d /data/projects anusha
[root@tecmint ~]# passwd anusha
```

You can see the user home directory and [other user-related information](#) like user id, group id, shell, and comments.

```
[root@tecmint ~]# cat /etc/passwd | grep anusha

anusha:x:1001:1001:./data/projects:/bin/bash
```

```
[root@tecmint:~]# useradd -d /data/projects anusha
[root@tecmint:~]# passwd anusha
Changing password for user anusha.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@tecmint:~]#
[root@tecmint:~]# cat /etc/passwd | grep anusha
anusha:x:1001:1001:./data/projects:/bin/bash
[root@tecmint:~]# _
```

Create User with Home Directory in Linux

3. Create a User with a Specific User ID



In Linux, every user has its own **UID (Unique Identification Number)**. By default, whenever we create a new user account in **Linux**, it assigns userid **500, 501, 502**, and so on...

But, we can create users with custom userid with the **'-u'** option. For example, the following command will create a user **'navin'** with custom userid **'1002'**.

```
[root@tecmint ~]# useradd -u 1002 navin
```

Now, let's verify that the user created with a defined userid (**1002**) using the following command.

```
[root@tecmint ~]# cat /etc/passwd | grep navin  
  
navin:x:1002:1002::/home/navin:/bin/bash
```

```
[root@tecmint:~]# useradd -u 1002 navin  
[root@tecmint:~]# passwd navin  
Changing password for user navin.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@tecmint:~]#  
[root@tecmint:~]# cat /etc/passwd | grep navin  
navin:x:1002:1002::/home/navin:/bin/bash  
[root@tecmint:~]# _
```

Create User with the User ID in Linux

NOTE: Make sure the value of a user ID must be unique from any other already created users on the system.

4. Create a User with a Specific Group ID

Similarly, every user has their own **GID (Group Identifier)**. We can create users with specific group IDs as well with the **-g** option. ^

Here in this example, we will add a user '**tarunika**' with a specific **UID** and **GID** simultaneously with the help of '**-u**' and '**-g**' options.

```
[root@tecmint ~]# useradd -u 1005 -g tecmint tarunika
```

Now, see the assigned user id and group id in '**/etc/passwd**' file.

```
[root@tecmint ~]# cat /etc/passwd | grep tarunika  
  
tarunika:x:1005:1000:~/home/tarunika:/bin/bash
```

To verify the user's GID, use the **id** command:

```
[root@tecmint ~]# id -gn tarunika
```

```
[root@tecmint:~]# useradd -u 1005 -g tecmint tarunika  
[root@tecmint:~]# passwd tarunika  
Changing password for user tarunika.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@tecmint:~]#  
[root@tecmint:~]# cat /etc/passwd | grep tarunika  
tarunika:x:1005:1000:~/home/tarunika:/bin/bash  
[root@tecmint:~]# id -gn tarunika  
tecmint  
[root@tecmint:~]# _
```

Create User with Group ID in Linux

5. Add a User to Multiple Groups

The '**-G**' option is used to add a user to additional groups. Each group name is separated by a comma, with no intervening spaces.



Here in this example, we are adding a user '**tecmint**' into multiple groups like **admins**, **webadmin**, and **developer**.

```
[root@tecmint:~]# groupadd admins
[root@tecmint:~]# groupadd webadmin
[root@tecmint:~]# groupadd developers
[root@tecmint:~]# usermod -a -G admins,webadmin,developers tecmint
[root@tecmint:~]# useradd -G admins,webadmin,developers paddy
```

Next, verify that the multiple groups are assigned to the user with the id command.

```
[root@tecmint ~]# id tecmint

uid=1000(tecmint) gid=1000(tecmint)
groups=1000(tecmint),1007(admins),1008(webadmin),1009(developers)
context=root:system_r:unconfined_t:SystemLow-SystemHigh
```



A terminal window showing a sequence of Linux commands to create groups, add users to groups, and verify the setup. Red arrows point from descriptive text labels to specific commands in the terminal output.

```
[root@tecmint:~]# groupadd admins
[root@tecmint:~]# groupadd webadmin
[root@tecmint:~]# groupadd developers
[root@tecmint:~]# usermod -a -G admins,webadmin,developers tecmint
[root@tecmint:~]# useradd -G admins,webadmin,developers paddy
[root@tecmint:~]# id tecmint
uid=1000(tecmint) gid=1000(tecmint) groups=1000(tecmint),1007(admins),1008(webadmin),1009(developers)
[root@tecmint:~]# id paddy
uid=1007(paddy) gid=1010(paddy) groups=1010(paddy),1007(admins),1008(webadmin),1009(developers)
[root@tecmint:~]# _
```

Annotations in the image:

- Create New Groups** points to the three `groupadd` commands.
- Add Existing User to Groups** points to the `usermod` command.
- Add New User to Groups** points to the `useradd` command.
- Check User Groups** points to the two `id` commands.

Add User to Group in Linux

[You might also like: [How to Add or Remove a User from a Group in Linux](#)]

6. Add a User without Home Directory

In some situations, where we don't want to assign home directories for a user, due to some security reasons. In such a situation, when a user logs into a system th



has just restarted, its home directory will be root. When such a user uses the [su command](#), its login directory will be the previous user's home directory.

To create users without their home directories, '-M' is used. For example, the following command will create a user 'shilpi' without a home directory.

```
[root@tecmint ~]# useradd -M shilpi
```

Now, let's verify that the user is created without a home directory, using the [ls command](#).

```
[root@tecmint ~]# ls -l /home/shilpi
```

```
ls: cannot access /home/shilpi: No such file or directory
```

```
[root@tecmint:~]# useradd -M shilpi
[root@tecmint:~]# passwd shilpi
Changing password for user shilpi.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@tecmint:~]#
[root@tecmint:~]# ls -l /home/shilpi
ls: cannot access '/home/shilpi': No such file or directory
[root@tecmint:~]# _
```

Create User Without Home Directory in Linux

7. Create a User with Account Expiry Date

By default, when we add user's with the 'useradd' command user account never get expires i.e their expiry date is set to 0 (means never expired).

However, we can set the expiry date using the '-e' option, which sets the date in YYYY-MM-DD format. This is helpful for creating temporary accounts for a specific period of time.



[You might also like: [How to Manage User Password Expiration and Aging in Linux](#)]

Here in this example, we create a user '**aparna**' with account expiry date i.e. **27th August 2021** in YYYY-MM-DD format.

```
[root@tecmint ~]# useradd -e 2021-08-27 aparna
```

Next, verify the age of the account and password with the '**chage**' command for user '**aparna**' after setting the account expiry date.

```
[root@tecmint ~]# chage -l aparna
```

Last password change	: Jun 25, 2021
Password expires	: never
Password inactive	: never
Account expires	: Aug 27, 2021
Minimum number of days between password change	: 0
Maximum number of days between password change	: 99999
Number of days of warning before password expires	: 7

```
[root@tecmint:~]# useradd -e 2021-08-27 aparna
[root@tecmint:~]# chage -l aparna
Last password change           : Jun 25, 2021
Password expires               : never
Password inactive              : never
Account expires                : Aug 27, 2021
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
[root@tecmint:~]# _
```

Create User with Account Expiry Date

8. Create a User with Password Expiry Date



The **'-f'** argument is used to define the number of days after a password expires. A value of **0** inactive the user account as soon as the password has expired. By default, the password expiry value set to **-1** means never expire.

Here in this example, we will set an account password expiry date i.e. **45 days** on a user **'mansi'** using **'-e'** and **'-f'** options.

```
[root@tecmint ~]# useradd -e 2014-04-27 -f 45 mansi
```

```
[root@tecmint:~]# useradd -e 2021-08-27 -f 45 mansi
[root@tecmint:~]# chage -l mansi
Last password change           : Jun 25, 2021
Password expires               : never
Password inactive              : never
Account expires                : Aug 27, 2021
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
[root@tecmint:~]# _
```

Create User with Password Expiry Date

9. Add a User with Custom Comments

The **'-c'** option allows you to add custom comments, such as the user's **full name**, **phone number**, etc to **/etc/passwd** file. The comment can be added as a single line without any spaces.

For example, the following command will add a user **'mansi'** and would insert that user's full name, **Manis Khurana**, into the comment field.

```
[root@tecmint ~]# useradd -c "Manis Khurana" mansi
```

You can see your comments in the **'/etc/passwd'** file in the comments section.

```
[root@tecmint ~]# tail -1 /etc/passwd
```

```
mansi:x:1010:1013:Manis Khurana:/home/mansi:/bin/sh
```

```
[root@tecmint:~]# useradd -c "Manis Khurana" mansi
[root@tecmint:~]#
[root@tecmint:~]# tail -1 /etc/passwd
mansi:x:1010:1013:Manis Khurana:/home/mansi:/bin/bash
[root@tecmint:~]# _
```

Create User with Full Name

10. Create User Login Shell in Linux

Sometimes, we add users who have nothing to do with the login shell or sometimes we require to assign [different shells to our users](#). We can assign different login shells to each user with the '-s' option.

Here in this example, will add a user 'tecmint' without login shell i.e. '/sbin/nologin' shell.

```
[root@tecmint ~]# useradd -s /sbin/nologin tecmint
```

You can check the assigned shell to the user in the '/etc/passwd' file.

```
[root@tecmint ~]# tail -1 /etc/passwd

tecmint:x:1011:1014:./home/tecmint:/sbin/nologin
```

```
[root@tecmint:~]# useradd -s /sbin/nologin tecmint
[root@tecmint:~]# tail -1 /etc/passwd
tecmint:x:1011:1014:./home/tecmint:/sbin/nologin
[root@tecmint:~]# _
```

Create User with Login Shell



Part II – 5 Advance Usage of useradd Commands

11. Add a User with Specific Home Directory, Default Shell, and Custom Comment

The following command will create a user '**ravi**' with home directory '**/var/www/tecmint**', default shell **/bin/bash** and adds extra information about the user.

```
[root@tecmint ~]# useradd -m -d /var/www/ravi -s /bin/bash -c "TecMint Owner" -U ravi
```

```
[root@tecmint:~]# useradd -m -d /var/www/ravi -s /bin/bash -c "TecMint Owner" -U ravi
[root@tecmint:~]# passwd ravi
Changing password for user ravi.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@tecmint:~]# cat /etc/passwd | grep ravi
ravi:x:1012:1015:TecMint Owner:/var/www/ravi:/bin/bash
[root@tecmint:~]# _
```

Create User with Home Directory and Login Shell

In the above command '**-m -d**' option creates a user with a specified home directory and the '**-s**' option sets the user's default shell i.e. **/bin/bash**. The '**-c**' option adds the extra information about the user and the '**-U**' argument creates/adds a group with the same name as the user.

12. Add a User with Home Directory, Custom Shell, Custom Comment, and UID/GID

The command is very similar to above, but here we defining shell as '**/bin/zsh**' and custom **UID** and **GID** to a user '**tarunika**'. Where '**-u**' defines the new user's **UID** (i.e. **100**) and whereas '**-g**' defines **GID** (i.e. **1000**).

```
[root@tecmint ~]# useradd -m -d /var/www/tarunika -s /bin/zsh -c "Tarunika Owner" -u 100 -g 1000
```

```
[root@tecmint:~]# useradd -m -d /var/www/tarunika -s /bin/zsh -c "TecMint Technical Writer" -u 1000 -g 100 tarunika
[root@tecmint:~]# cat /etc/passwd | grep tarunika
tarunika:x:1000:100:TecMint Technical Writer:/var/www/tarunika:/bin/zsh
[root@tecmint:~]# _
```

Create User with UID and GID

13. Add a User with Home Directory, No Shell, Custom Comment, and User ID

The following command is very much similar to the above two commands, the only difference is here, that we disabling the login shell to a user called '**avishek**' with a custom **User ID** (i.e. **1019**).

Here '**-s**' option adds the default shell **/bin/bash**, but in this case we set login to '**/usr/sbin/nologin**'. That means user '**avishek**' will not able to login into the system.

```
[root@tecmint ~]# useradd -m -d /var/www/avishek -s /usr/sbin/nologin
```

```
[root@tecmint:~]# useradd -m -d /var/www/avishek -s /usr/sbin/nologin -c "TecMint Sr. Technical Writer" -u 1019 avishek
[root@tecmint:~]# cat /etc/passwd | grep avishek
avishek:x:1019:1019:TecMint Sr. Technical Writer:/var/www/avishek:/usr/sbin/nologin
[root@tecmint:~]# _
```

Create User with UID and Nologin

14. Add a User with Home Directory, Shell, Custom Skell/Comment, and User ID

The only change in this command is, we used '**-k**' option to set the custom skeleton directory i.e. **/etc/custom.skel**, not the default one **/etc/skel**. We also used '**-s**' option to define different shell i.e. **/bin/tcsh** to user '**navin**'.

```
[root@tecmint ~]# useradd -m -d /var/www/navin -k /etc/custom.skel
```

```
[root@tecmint:~]# useradd -m -d /var/www/navin -k /etc/custom.skel -s /bin/tcsh -c "No Active Member of TecMint" -u 1027 navin
[root@tecmint:~]# cat /etc/passwd | grep navin
navin:x:1027:1027:No Active Member of TecMint:/var/www/navin:/bin/tcsh
[root@tecmint:~]# _
```

Create User with Shell and UID

15. Add a User without Home Directory, No Shell, No Group, and Custom Comment

The following command is very different than the other commands explained above. Here we used the ‘-M’ option to create a user without the user’s home directory and the ‘-N’ argument is used that tells the system to only create a

Linux Foundation LFCS and LFCE Certification Preparation Guide -

[Get This Book](#)

```
[root@tecmint ~]# useradd -M -N -r -s /bin/false -c "Disabled TecMint Member" clayton
```

```
[root@tecmint:~]# useradd -M -N -r -s /bin/false -c "Disabled TecMint Member" clayton
[root@tecmint:~]# cat /etc/passwd | grep clayton
clayton:x:993:100:Disabled TecMint Member:/home/clayton:/bin/false
[root@tecmint:~]# _
```

Create User with NoLogin and Group

For more information and options about useradd, run the ‘useradd’ command on the terminal to see available options.

```
# useradd
```

[You might also like: [15 Useful Usermod Command Examples in Linux](#)]

🔗 [Adduser](#) , [Linux Users](#) , [Useradd](#)



< [12 Useful “df” Commands to Check](#)