1. Linux User Management:

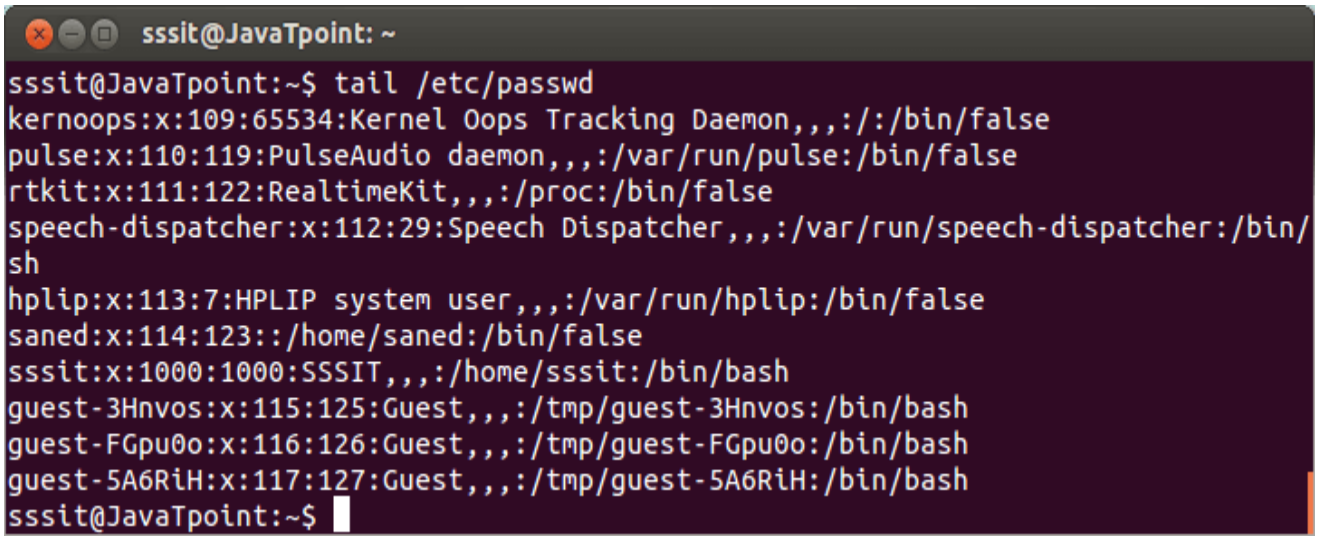
User management includes everything from creating a user to deleting a user on your system. User management can be done in three ways on a Linux system.

Graphical tools are easy and suitable for new users, as it makes sure you'll not run into any trouble.

Command line tools includes commands like useradd, userdel, passwd, etc. These are mostly used by the server administrators.

Third and very rare tool is to **edit the local configuration files** directly using vi.

1. /etc/passwd
   The local user database in Linux is /etc/passwd directory.

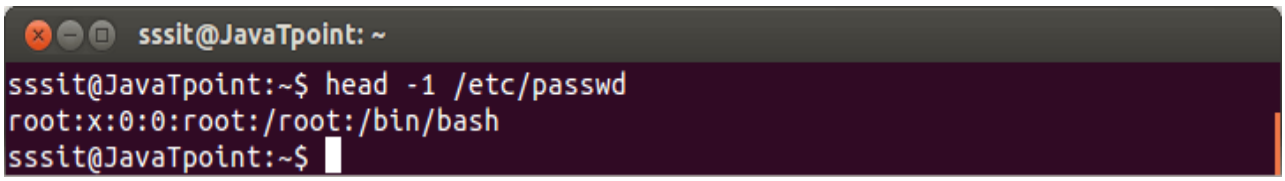


Look at the above snapshot, it has seven columns separated by a colon. Starting from the left columns denotes username, an x, user id, primary group id, a description, name of home directory and a login shell.

---

root

The root user is the superuser and have all the powers for creating a user, deleting a user and can even login with the other user's account. The root user always has userid 0.



---

useradd

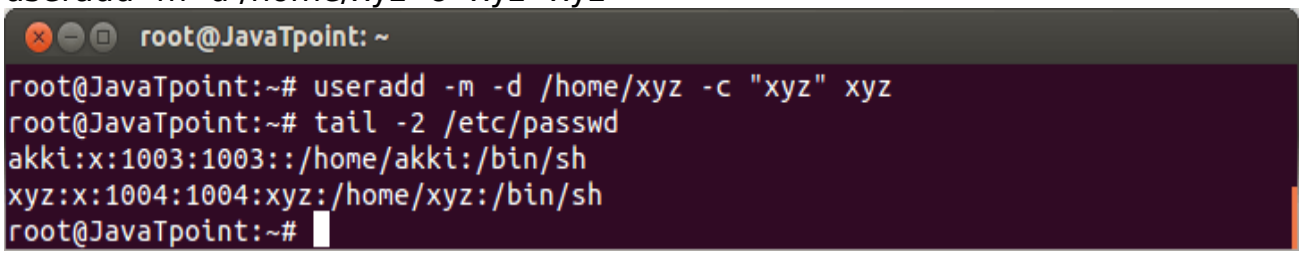
With useradd commands you can add a user.

Syntax:

1. useradd -m -d /home/**<userName>** -c "**<userName>**" **<userName>**

Example:

1. useradd -m -d /home/xyz -c "xyz" xyz



Look at the above snapshot, we have created a user **xyz** along with creating a home directory (-m), setting the name of home directory (-d), and a description (-c).

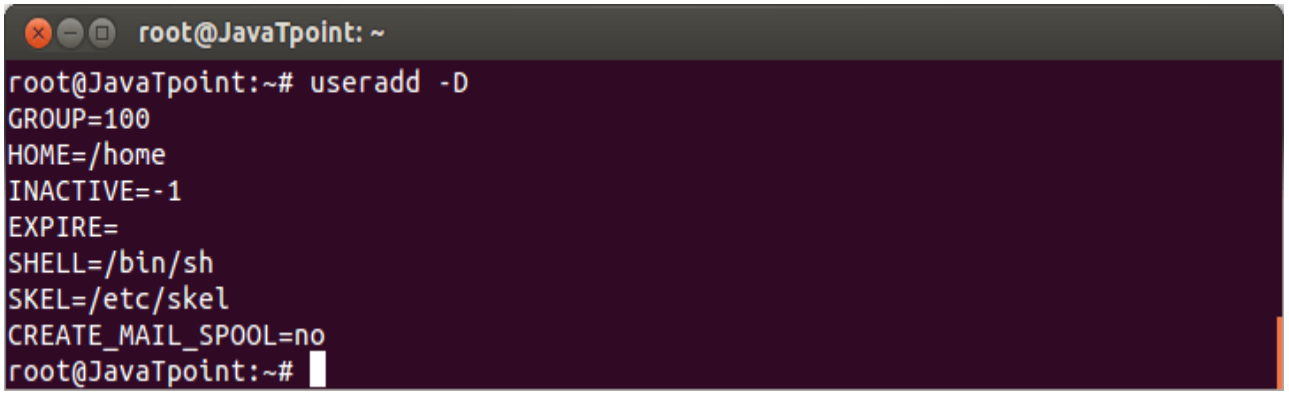
The 'xyz' received **userid** as 1004 and **primary group id** as 1004.

/etc/default/useradd

File /etc/default/useradd contains some user default options. The command **useradd -D** can be used to display this file.

Syntax:

1. useradd -D



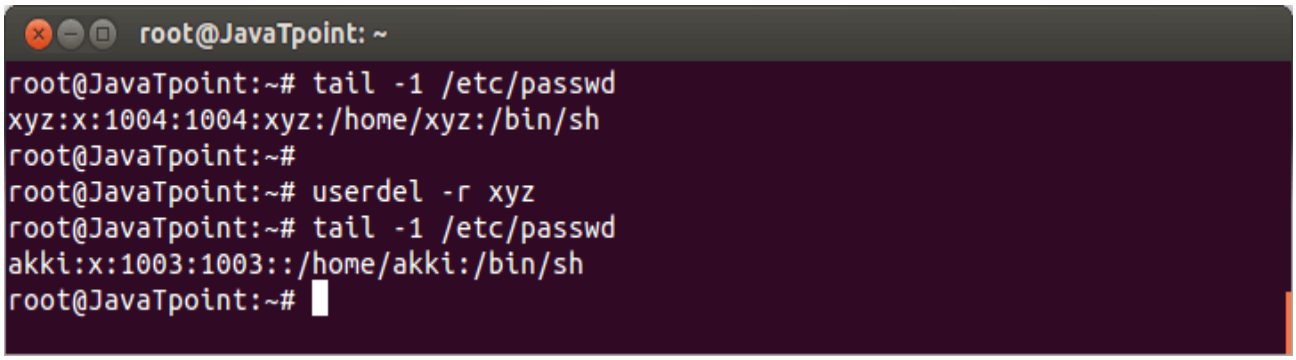
---

userdel

To delete a user account userdel command is used.

Syntax:

1. userdel -r **<userName>**



Example:

1. userdel -r xyz

   Look at the above snapshot, first we have shown the xyz user account with 'tail' command. To delete it, command **"userdel -r xyz"** is passed.

   To recheck, again 'tail' command is passed and as you can see no xyz user account is displayed.

   Hence, it is deleted.

---

usermod

The command usermod is used to modify the properties of an existing user.

Syntax:

1. usermod -c **<**'newName'**> <oldName>**

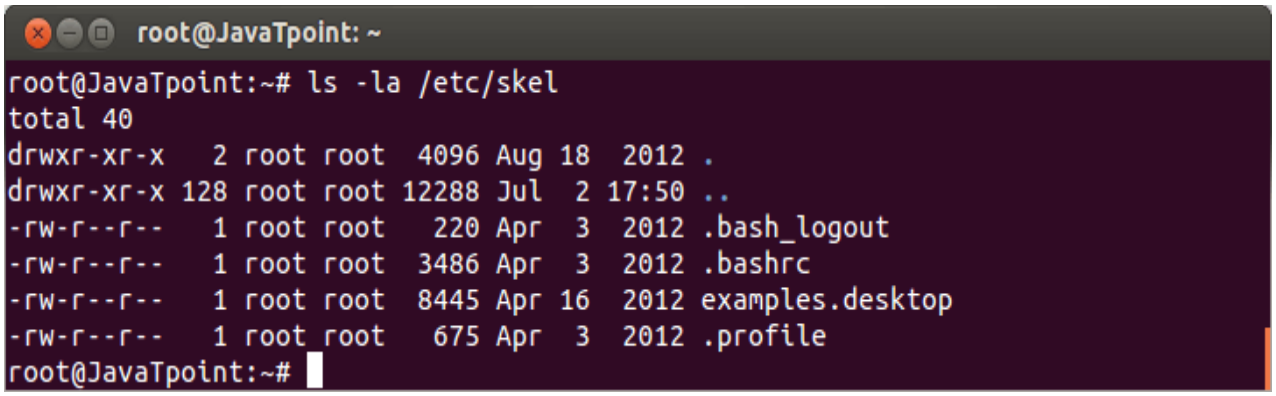
Example:

1. usermod -c 'jhonny' john

Look at the above snapshot, user name **john** is replaced by the new user name **jhonny**

/etc/skel/

The /etc/skel/ contains some hidden files which have profile settings and default values for applications. Hence, it serves as a default home directory and user profile. While using useradd -m option, the /etc/skel/ is copied to the newly created directory.



Look at the above snapshot, files of /etc/skel/ is listed.

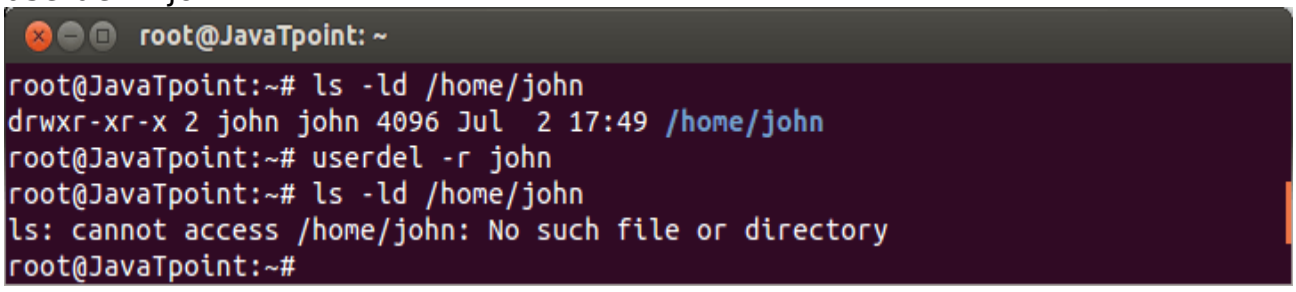
Deleting Home Directories

By using **userdel -r** option, you can delete home directory along with user account.

Syntax:

1. userdel -r **<userName>**
   Example:

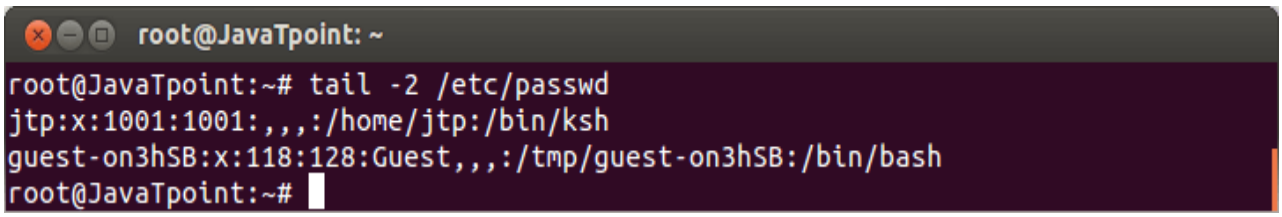
1. userdel -r john



Look at the above snapshot, both home directory as well as user account john is deleted.

Login Shell

The /etc/passwd file also tells about the login shell for the user.



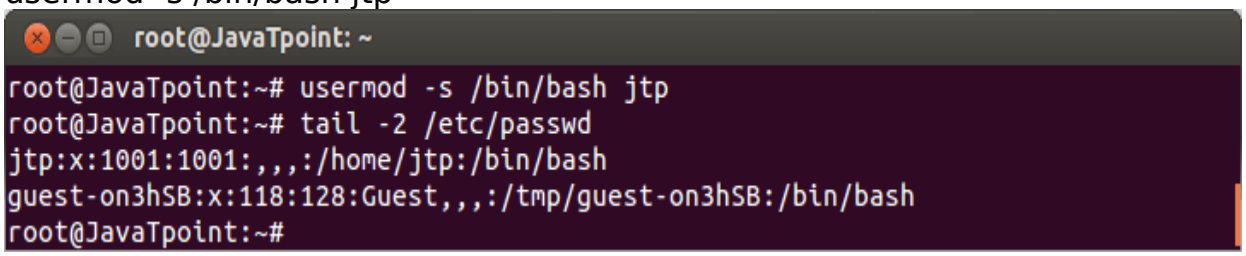
Look at the above snapshot, user guest will log in with **/bin/bash** shell and user jtp will log in with **/bin/ksh shell**.

You can change the shell mode with usermod command for a user.

Syntax:

1. usermod -s **<newShell> <userName>**
   Example:

1. usermod -s /bin/bash jtp



Look at the above snapshot, shell of jtp is changed to **/bin/bash** from **/bin/ksh.**

---

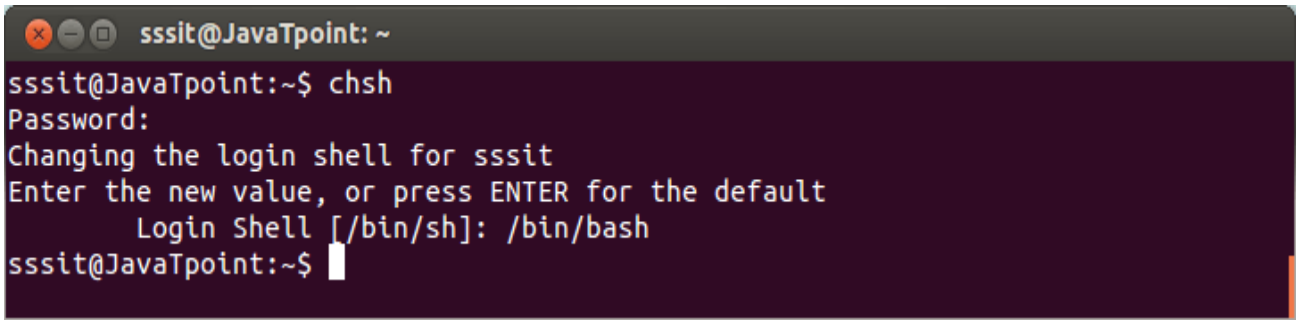
chsh

Users can change their login shell with chsh command.

Both the command **chsh** and **chsh -s** will work to change the shell.

Syntax:

1. chsh



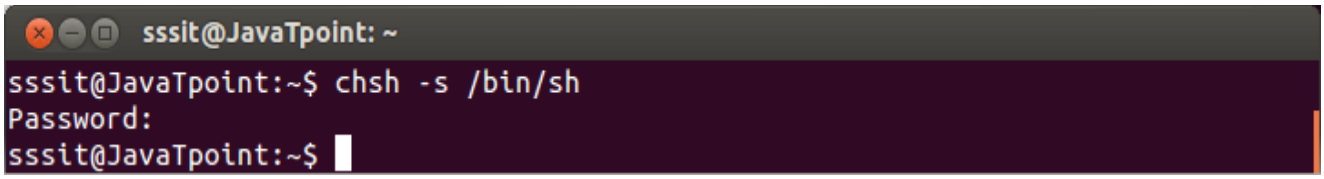
Look at the above snapshot, command chsh has changed the sssit login shell from **/bin/sh** to **/bin/bash**.

Syntax:

1. chsh -s **<newShell>**

Example:

1. chsh -s /bin/sh



Look at the above snapshot, login shell is changed into /bin/s.

2. Linux User Password:

This chapter tells you about the local users password. You will learn here to change the password, set the password using different methods.

First method is by using **passwd command**.

Second method is with **openssel passwd** command.

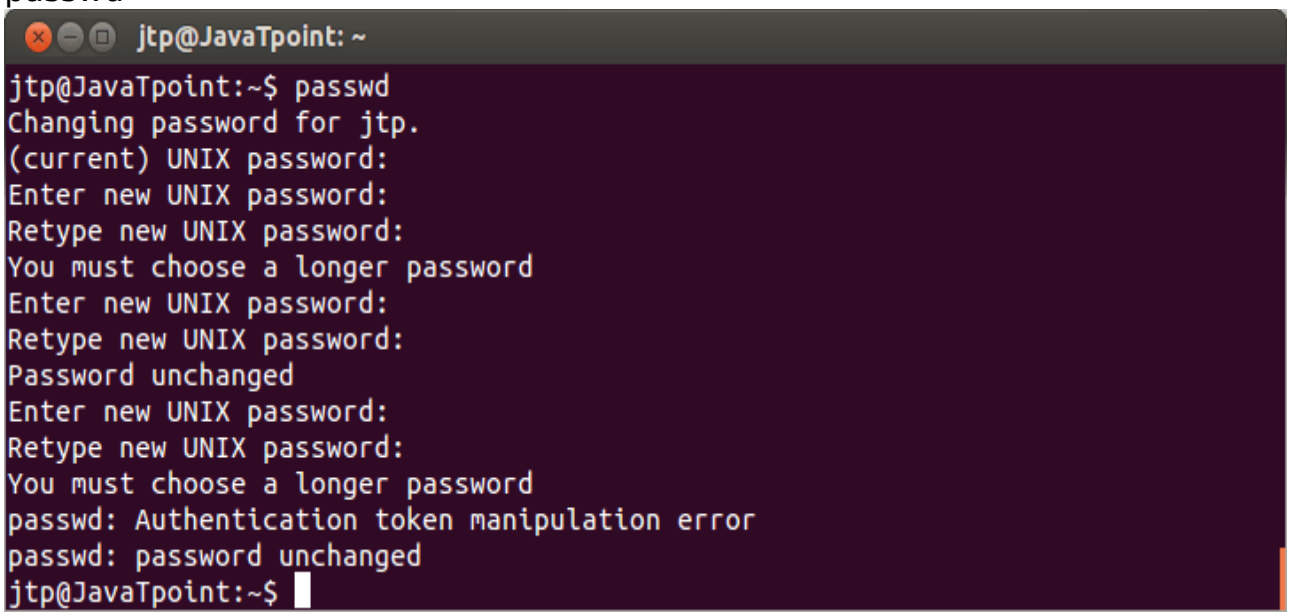
Using passwd command

passwd

A user can set the password with the command **passwd**. Old password has to be typed twice before entering the new one.

Syntax:

1. passwd



Look at the above snapshot, shell warns the user from creating a simple password. Ultimately, after two or three attempts if password is not changed then the command **passwd fails** and you have to pass the command again.

Although, these rules are not applied on the root user neither they need to type the old password. They can change the password directly.

Syntax:

1. passwd **<userName>**
   Example:

1. passwd jtp

```
root@JavaTpoint: ~
root@JavaTpoint:~# passwd jtp
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@JavaTpoint:~#
```

Look at the above snapshot, password is changed successfully without any warning.

Shadow File

Shadow files are the encrypted user passwords which are kept in **/etc/shadow**. This file is read-only directory and can be read only by root.

Syntax:

1. /etc/shadow

```
root@JavaTpoint: ~
root@JavaTpoint:~# tail -6 /etc/shadow
sssit:$6$Z36K8tNu$VopTtZ/q3V/4N/LFyusuM4h91HdFZbWT0F7eJWW6z/tGQpDAb7OGHlyVER1x0X
Otqu/klHQR3Lptq86oMCLMe0:16932:0:99999:7:::
guest-3Hnvos:*:16934:0:99999:7:::
guest-FGpu0o:*:16935:0:99999:7:::
guest-5A6RiH:*:16939:0:99999:7:::
jtp:$6$FJ3zTwe6$Yrs8nuZVz0MXBVbLWnlIGs9tG67vgVU0DR1MtUx2S9phPQGPJ7CFKawCaiJ.wmXR
nfhgf3CeOgK//14nP4Dyf1:16985:0:99999:7:::
guest-on3hSB:*:16984:0:99999:7:::
root@JavaTpoint:~#
```

Look at the above snapshot, the **/etc/shadow** file contains nine columns separeted by colons.

Starting from left to right, these nine columns contain username, encrypted password, last changed password day, number of days password must be left unchanged, password expiry day, warning number of days before password expiry, number of days after expiry before disabling the account, and the day account was disabled. Last column has no meaning yet.

Encryption With passwd

Passwords are always stored in encrypted format. Encryption is done with crypt function. The simplest way to add a user with a password is to add the user with the command **useradd -m** and then set the user's password with command **passwd**.

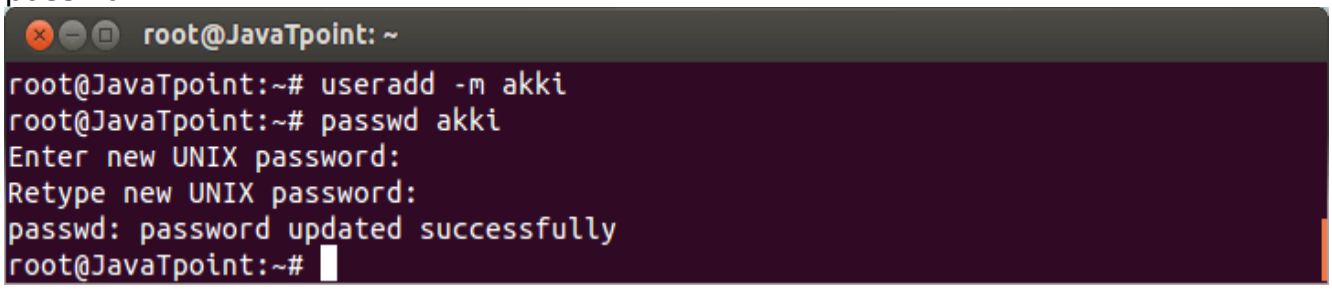
Syntax:

1. useradd -m **<userName>**
   Example:

1. useradd -m akki
   Syntax:

1. passwd **<typePassword>**
   Example:

1. passwd ****



Look at the above snapshot, user name akki is created with a password successfully.

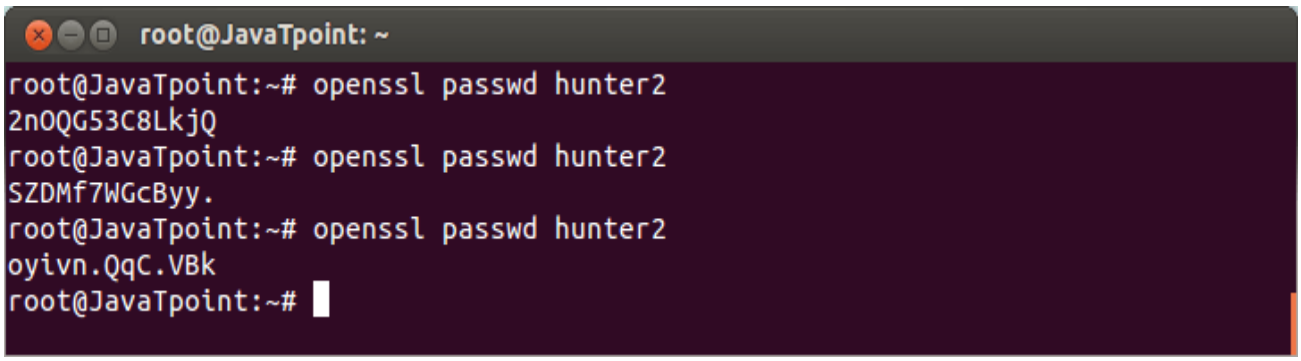
Using openssl passwd

Encryption With openssl

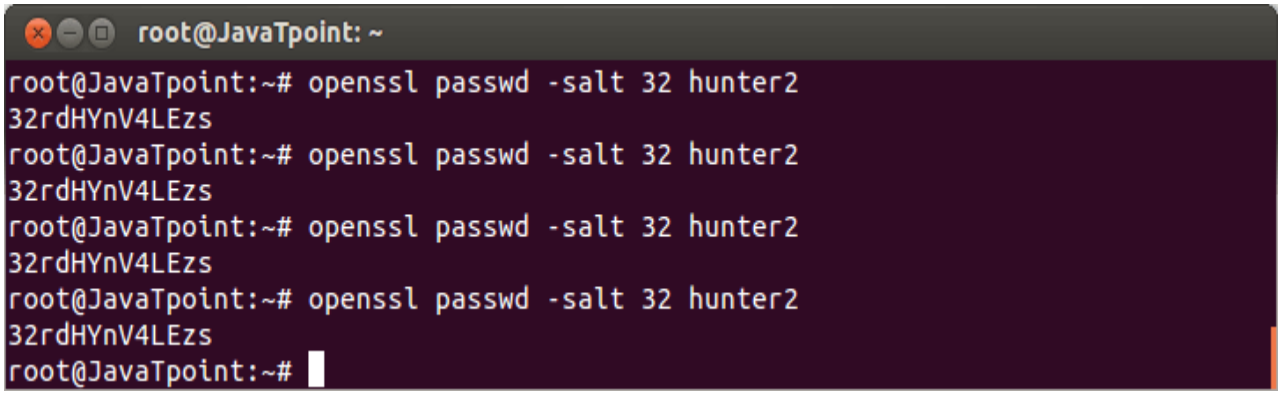
To create a user with a password **-p** option is also used, but that requires an encrypted password.

Thid encrypted password can be generated with openssl passwd command.

openssl passwd command can genereate several distinct hashes for the same pssword. To do this, it uses salt.



This salt can be chosen and is visible as the first two characters of the hash as shown below.



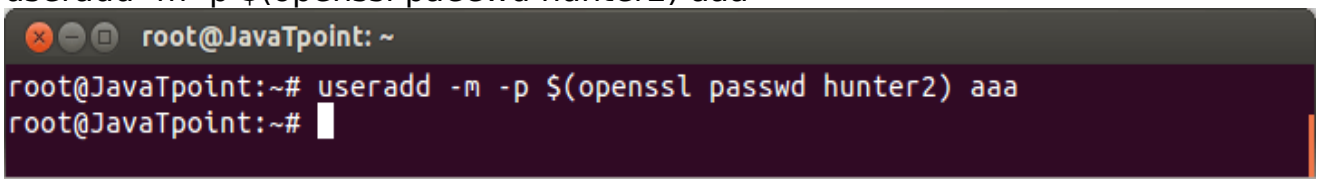
Look at the above snapshot, the first two characters start from the defined sale **'32'**.

To create a user with password using openssl command, following syntax is used.

Syntax:

1. useradd -m -p $(openssl paeewd hunter2) **<userName>**
Example:

1. useradd -m -p $(openssl paeewd hunter2) aaa



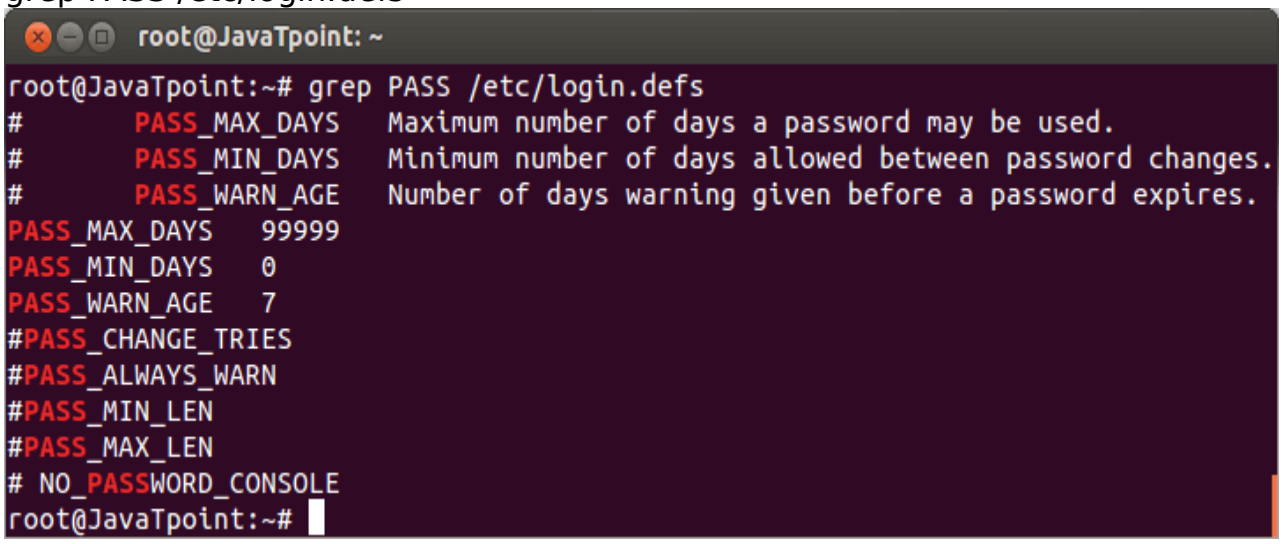
Look at the above snapshot, user aaa is created and its password is kept into command history.

/etc/login.defs

The /etc/login.defs file contains some default settings like password aging and length settings.,

Syntax:

1. grep PASS /etc/login.defs



---

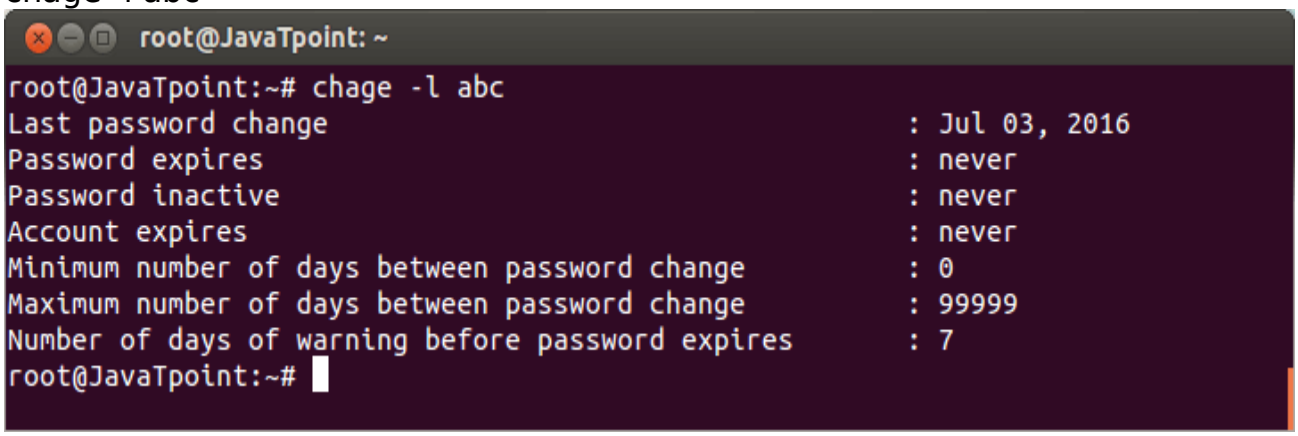
chage

The chage command can be used by a user to know the information about their password. The -l option is used to list the information.

Syntax:

1. chage -l **<userName>**
   Example:

1. chage -l abc



Disabling A Password

Passwords in /etc/shadow are not saved starting with exclamation mark **(!)**. If exclamation mark is present in starting then password can not be used.

This feature can be used to disable a password and the process is called **locking, disabling** and **suspending** a user account. It can be done in **vi** or with **usermod** command.

Here, we'll disable the password of akki with usermod command.

Syntax:

1. usermod -L **<userName>**
   Example:

1. usermod -L akki

```
root@JavaTpoint: ~
root@JavaTpoint:~# grep akki /etc/shadow
akki:$6$6eXsqsgr$0lOwua3L7FRPMYoMWkHlWx6Im5zfMGntyU3HfHfcSaVtcuiGD4624aCyZCYmAwZ
fRRLtsgs2HGMgx64kqk3NN/:16985:0:99999:7:::
root@JavaTpoint:~#
root@JavaTpoint:~# usermod -L akki
root@JavaTpoint:~#
```

Look at the above snapshot, first command shows hashed password of **akki**, and command **"usermod -L akki"** disables the password of akki. Now user akki can't authenticate using this password.

Look at the above snapshot, hashed password is preceded with **!**, which means it is disabled.

Please note that root user will be able to open the akki account as password is not needed here. And if user akki wouldn't have set password, then akki can also login.

You can **unlock** your account with **usermod -U**.

Syntax:

1. usermod -U **<userName>**
   Example:

1. usermod -U akki

```
root@JavaTpoint: ~
root@JavaTpoint:~# usermod -U akki
root@JavaTpoint:~# grep akki /etc/shadow
akki:$6$6eXsqsgr$0lOwua3L7FRPMYoMWkHlWx6Im5zfMGntyU3HfHfcSaVtcuiGD4624aCyZCYmAwZ
fRRLtsgs2HGMgx64kqk3NN/:16985:0:99999:7:::
root@JavaTpoint:~#
```

Look at the above snapshot, hashed password of akki is unlocked now as there is no **(!)** mark in starting.