



ABC IT Education

WE'LL TAKE YOU FROM ZERO TO HERO IN A SNAP

Linux Systems Administration



Linux User Management

User Management

- In the user management lesson you'll learn how to manage users and groups, you'll learn where user and group information is stored at on a Linux system and you'll learn how to add, delete and modify accounts and groups.
- Linux is a multiuser operating system, so not only can multiple accounts exist on the system, but those accounts can be used at the same time.

Accounts have a:

- Username (or login ID).
- UID (user ID). This is a unique number.
- Default group.
- Comments.
- Shell.
- Home directory location.

User Management

- Each account consists of a user name and a unique number called the UID, short for user ID.
- Also each account has a default group to which it belongs, some comments associated with that account a shell to execute when the user logs into the system and a home directory.
- All this information is stored in the `/etc/passwd` file.

`/etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
```

The format of the `/etc/passwd`:

```
username:password:UID:GID:comments:  
home_dir:shell
```

User Management

- The first entry in the `/etc/passwd` file is the root account.
- The password file contains different fields of information, each separated by a colon (:).
- The first field contains a username and the next field contains the password.
- Next, you'll find the user ID, then the default group ID followed by comments.
- The sixth field in the password file is the location of the accounts home directory, while the last field in the password file is the shell to execute when that user logs into the system.
- In the example of the root account, you can see that the username is root.

User Management

- In the password field, there is an 'x', and this means that the encrypted password is stored in the /etc/shadow file.
- We will be talking about the shadow file in just a moment.
- The user ID for the root account is zero, the default group ID for the root account is also zero, the comment for the root account is simply the word root, the home directory is '/root', and the shell associated with the root account is bash.
- Let's look at another entry in the /etc/passwd file. This is for the Monica account.

/etc/passwd

```
msaks:x:1003:1003:Monica Saks:/home/msaks:/bin/bash
```

The format of the /etc/passwd:

```
username:password:UID:GID:comments:  
home_dir:shell
```

User Management

- In this case, the username is msaks, like the root account, the password field contains an 'x', the user ID or UID is 1003, the group ID is also 1003.
- The comment in this case is the full name of the person that's using the account, which is Monika Saks.
- The home directory is '/home/msaks' and the shell is '/bin/bash'.
- Note: Even though Linux supports usernames up to 32 characters in length, it is customary to keep usernames to 8 or fewer characters.
- When using usernames longer than 8 characters you will run in some situations where the UID is displayed in place of the username, for example, when looking at the output of the 'ps' command.

User Management

```
$ ps -fu joehenderson
UID          PID    PPID  C  STIME TTY          TIME CMD
joehend+  1182  1181   0  22:48 pts/1    00:00:00 -bash

OR

$ ps -fu joehenderson
UID          PID    PPID  C  STIME TTY          TIME CMD
1000         1182  1181   0  22:48 pts/1    00:00:00 -bash
```

- In addition, the usernames are case sensitive.
- Even though uppercase letters are allowed in user names, by convention usernames are in all lower case letters.
- Digits are also allowed in usernames but avoid using special characters.

User Management

- Historically encrypted password information was stored in the `/etc/passwd` file following the username.
- However the `/etc/passwd` file is readable by anyone on the system, so storing password information, even encrypted, is a security risk.
- Now, by default, the encrypted password information is stored in `/etc/shadow` which is only readable by the super user account.
- The UID is a unique number and the root account is always UID 0.
- Accounts meant to be used by the system typically have UIDs lower than 1000.
- This is configurable however by updating the `/etc/login.defs` file.

User Management

- The group ID, or GID, listed in the password file entry for the account is the account's default group.
- When a user creates a file, that file will belong to the user's default group.
- If a user wants to create files using another group they can use the '**newgrp**' command to change to a new group before creating the files.
- The comment field typically contains the user's real name or a description of what the account is used for.
- You'll sometimes hear this field referred to as the GECOS field.
- This is a historical hold over from the early years of Unix.

User Management

- When a user logs into the system they are placed into their home directory listed in the `/etc/passwd` file.
- If this directory does not exist they will be placed into the root directory.
- The shell will be executed when the user logs into the system with their account.
- You can see a list of installed shells on your Linux system by looking at the `'/etc/shells'` file.
- Whatever is listed in the shell field will be executed upon login even if the program is not actually a shell.
- You may see `'/usr/sbin/nologin'` or `'/bin/false'` in the shell field for certain accounts.
- This ensures that no one can use those accounts interactively.

User Management

- You can also use the shell field to execute a program when a user logs into the system.
- For example, you could force users into a menu driven application that only allows them access to certain actions.
- Like the '/etc/passwd' file, the '/etc/shadow' file contains a series of fields separated by a colon.

/etc/shadow

```
root:$6$9g1IC8AYzqoZP21:16502:0:99999:7:::
```

- The first field is the username, the second field contains the encrypted password, the third field is the number of days since January 1, 1970 since the password has been changed, the fourth field is the number of days before the password can be changed.

User Management

- The fifth field is the number of days after which the password must be changed, and if this field contains 99999, the user never has to change their password.
- The sixth field is the number of days to warn the user that their password will expire.
- The seventh field is the number of days after the password has expired that the account is disabled.
- The eighth field is the number of days since January 1, 1970 that an account has been disabled and the ninth field is reserved for future use.
- Now that you know where account information is stored, let's create an account using the '**useradd**' command.
- **Note:** Adding accounts requires superuser privileges, so make sure that you are using the root account or sudo.

User Management

useradd

`useradd [options] username`

<code>-c "COMMENT"</code>	Comments for the account.
<code>-m</code>	Create the home directory.
<code>-s /shell/path</code>	The path to the user's shell.

- The format of the 'useradd' command is 'useradd' followed by 'options' and finally the username.
- The most commonly used options for the useradd command include
 - '-c' followed by a comment,
 - '-m' to create a user's home directory,
 - '-s' to provide the path to the user's shell.
- `$ useradd -c 'Taku Masong' -s /bin/bash -m tmasong`

User Management

- In this example, an account is created for Taku Masong.
- His username is tmasong and his shell is bash.
- Next, let's assign the account a password.
- To do this, use the '**passwd**' command followed by the username. You'll be prompted to enter a password for the user and to confirm that password.
- Let's look at the entry for the account in the '/etc/passwd' file and the '/etc/shadow' file.

Account Information for "tmasong"

```
# tail -1 /etc/passwd
tmasong:x:1005:1007:Taku Masong:/home/tmasong:/bin/bash
# tail -1 /etc/shadow
tmasong:$6$iDDsgsPYtR8c2Uc.:16507:0:99999:7:::
```

User Management

- In this example, Taku's UID is 1005, his GID is 1007, his home directory is '/home/tmasong' and his shell is '/bin/bash'.
- Other options for the useradd command include the following.
 - '-g' to specify default group for the account.
 - '-G' followed by a list of groups.
- This adds the account to those additional groups.
- Let's create an account for Elias Harris.
- His login will be 'eharris' and his default group will be 'sales'.
- We will also make him a member of the 'students' group as well.

User Management

- Note that not every account on a Linux system is meant to be used by a person.
- Some accounts exist to run applications or perform system functions.
- Some common examples of this include accounts that run web server processes, database processes, or some other application processes.
- Let's create an account that will be used to run the Apache web server process.
- You'll notice that the shell was set to `'/usr/sbin/nologin'`.
- This is because we don't want someone to be able to log into the system using the account.

System or Application Accounts

```
# useradd -c "Apache Web Server  
User" -d /opt/apache -r -s  
/usr/sbin/nologin apache  
# tail -1 /etc/passwd  
apache:x:999:999:Apache Web Server  
User:/opt/apache:/usr/sbin/nologin  
#
```

- ➡ We also use the '-r' option which instructs useradd to create a system account.
- ➡ Effectively this means the account will receive a UID in the system account range as defined in '/etc/login.defs'.
- ➡ In this example, the user received UID 999.

User Management

- The home directory was specified using the '-d' option.
- By default, the home directory for a new account is created in the '/home' directory.
- The actual directory will be the name of the user account.
- So, by default the apache account's home directory will be '/home/apache'.
- However, since this account is used by an application we set the home directory to the directory where the application is installed.
- Notice that the '-m' option was not used in this case.
- When using the '-m' option the contents of the skeleton directory, '/etc/skel', are copied into the user's home directory.

User Management

- The contents of '/etc/skel' usually include shell configuration files which are not needed for application accounts in many instances.
- Here are the new options we used to create this account.
 - '-r' to create a system account.
 - '-d' to specify a path to the home directory.
- It's a common practice to use the same UID for an account across multiple systems.
- This makes syncing data or sharing data easier to do as Linux uses UID's to determine a file's ownership.
- The account name is really for the sake of us humans.
- We use the '-u' option to specify a UID when creating an account.

User Management

Use -u to specify the UID

```
# useradd -c "MySQL Server" -d  
/opt/mysql -u 97 -s /usr/sbin/nologin  
mysql
```

- In this example, we're creating an account to be used with the MySQL server by specifying the UID of 97 by using '-u 97' on the command line.
- After we create the account, you can see that the UID is indeed 97 in the etc password file.

```
# tail -1 /etc/passwd  
mysql:x:97:1003:MySQL Server:  
/opt/mysql:/usr/sbin/nologin
```

User Management

- To delete an account use the '**userdel**' command followed by the username.
- If you want to delete the account's home directory use the '-r' option. It also removes the user's mail spool file if it exists.
- In this example, we'll delete the eharris account, but leave his home directory intact since there are some files in there we want to use later.
- We'll also delete the tmasong account and remove his home directory.

User Management

`userdel [-r] username`

```
# ls /home
eharris tmasong
# userdel eharris
# ls /home
eharris tmasong
# userdel -r tmasong
# ls /home
eharris
```

- To update, or modify, an existing account use the **usermod** command.
- Here are the most commonly used options to the **usermod** command.
- You can use '-c' followed by a comment to change the comment associated with an account.

User Management

- You can change the primary group by specifying '-g' followed by the group name.
- If you want to change the additional groups the account belongs to, use 'G' followed by a list of groups separated by commas.
- To change the account shell, use '-s' followed by a path to that shell.

usermod

```
usermod [ options ] username
```

-c "COMMENT"	Comments account.
-g GROUP	Specify the default group.
-G GROUP1,GROUPN	Additional groups.
-s /shell/path	Path to the user's shell.

User Management

- Here is an example where we update the comment associated with the MySQL account.
- I've to re-run the user 'usermod' command we can see that the change takes effect in the '/etc/passwd' file.

usermod

```
# grep mysql /etc/passwd
mysql:x:97:1003:MySQL Server:
/opt/mysql:/usr/sbin/nologin
# usermod -c "MySQL User" mysql
# grep mysql /etc/passwd
mysql:x:97:1003:MySQL User:
/opt/mysql:/usr/sbin/nologin
```

User Groups Management

- Group details are stored in the '/etc/group' file.
- The first entry in the /etc/group file is the root group.
- Here is a sample entry from /etc/group.

/etc/group

```
root:x:0:
```

```
sales:x:1001:john,mary
```

The format of the /etc/group file:

```
group_name:password:GID:account1,accountN
```

- As you can see, the format of the group file is as follows.
- 'group_name' followed by password, followed by a GID, or group ID, then a list of accounts or members of that group.
- Like the password file, each field is separated by a colon.

User Groups Management

- The group name is the human readable name that you will see when group information is displayed by commands such as 'ls'.
- The password field is used for privileged groups, but that functionality is rarely ever used.
- When there is an 'x' in this field it means that shadow group passwords are being used and that information is stored in the '/etc/gshadow' file.
- The GID is the group ID, and it is simply a unique number which represents the group.
- The remaining field lists the members of the group separated by commas.
- You might have noticed that the root group did not contain a list of members.

User Groups Management

- Remember that the `/etc/passwd` file specifies an account's default group.
- In the case of the root user the default GID is 0.
- When an account's default GID is listed in the `/etc/passwd` file, that account is in that group even if it is not listed in the `members` field in the `/etc/group` file.

`groups [username]`

```
# groups root  
root
```

- To display the groups that a member belongs to, pass the username to the **`groups`** command.
- If you execute the **`groups`** command without supplying a username, then your group memberships are listed.

User Groups Management

- To confirm that the root user is in fact part of the root group, we can run the command '**groups root**'.
- To create a group, use the 'groupadd' command.
- The most commonly used option for the groupadd command is '-g' which allows you to specify the GID.

groupadd [options] group_name

```
# groupadd web
# tail -1 /etc/group
web:x:1003:
# groupadd -g 2500 db
# tail -1 /etc/group
db:x:2500:
```

- In the above example, we are creating two groups.

User Groups Management

- The first group will let the group command automatically select the GID.
- For the second group we'll specify the GID.
- You can see that the web group received GID 1003 and the 'db' group received GID 2500 since that's what we specified on the command line.
- To delete a group simply pass the group name to the **'groupdel'** command.
- To delete the 'db' group, run 'groupdel db'.

groupdel group_name

```
# groupdel db  
#
```

User Groups Management

- To change the properties of an existing group use the '**groupmod**' command.

groupmod

```
groupmod [options] group_name
```

```
-g GID          Change the group ID to GID.
```

```
-n GROUP        Rename the group to GROUP.
```

- To change the GID, use '-g' followed by the new GID.
- To change the name of the group, use '-n' followed by the new name of the group.

User Groups Management

groupmod

```
# grep web /etc/group
web:x:1003:
# groupmod -g 1234 web
# grep web /etc/group
web:x:1234:
# groupmod -n http web
# grep http /etc/group
http:x:1234:
```

- ➡ In this example we change the GID of the web group from 1003 to 1234.
- ➡ Next we change the name from web to http.

Class Practice

- Create the following groups
 - writers
 - tv
 - movie
- Create user accounts
 - All users will have the default group of writers.
 - Some will also be in the tv group.
 - Others will be in the movie group.

- In this class practice, we are going to create a writers group, a tv group, and a movie group.
- Let's create the writers group using '**groupadd writers**'.
- Then the same thing with 'tv' group. Finally 'movie' group.
- Check the /etc/group file and you can see that the last three entries are the three accounts we just created.

- You'll notice that the GID or group IDs are incremented by one automatically by the groupadd command.
- Let's create the first of our users, **John Stone**.
- We'll put the comment as this user's name, the primary group will be writers, so '-g writers', the additional group will be 'tv'.
- We'll use '-m' to make sure the home directory gets created, and will specify shell with '-s' in this case we'll give this user the /bin/bash, and finally we need to provide a username or login ID so we'll use first initial, last name in this example.
- So I can create a password for this user.
- Let's see what groups this user is in using the 'groups' command.

- You can see that the user is in writers and tv group, just like we specified in the useradd command.
- Go ahead and add another user, **Peter Bossman**. Put him in the writer's group as well. He is a tv writer, so put him in the television group or tv group.
- Create his home directory and give him the bash shell.
- Add a third writer, **Lanhan Clinton**, with an additional group of movie, meaning that he is in the writers and movie groups.
- Let's add one final writer, **Mark Twain**. Use '-g' for the primary group, let's use writers, '-G' for any additional groups, movie.
- Let's add this person to 2 groups, add to the movie and tv groups.

- Now if we look at the `/etc/group` file, we can see the list of members in the `tv` and `movie` groups, however we do not see any users and the members list for `writers`.
- We know that those users are in that group because we typed `groups <one of the users>` and you can see that he's in the `writer's` group.
- If we search for `writer's` group GID in the `/etc/passwd` file, we can confirm that each of these user's primary groups are the GID which corresponds to the GID of the `writers` group and that's why they are included in the `writer's` group even though their usernames are not listed in the `/etc/group` file.

Summary

- Account information is stored in the `/etc/passwd` and `/etc/shadow` files.
- In addition to a username, each account consists of a unique number, the UID, a default group, a comment, a home directory location, and a login shell.
- Accounts are created with the `'useradd'` command, deleted with the `'userdel'` command and modified by using the `'usermod'` command.
- Group information is stored in the `/etc/group` file.
- To create a group, use the `'groupadd'` command, to delete groups by using the `'groupdel'` command and to update an existing group use the `'groupmod'` command.
- To list group memberships for an account, use the `'groups'` command.