utrain

Users and Groups in Linux

utrains.org

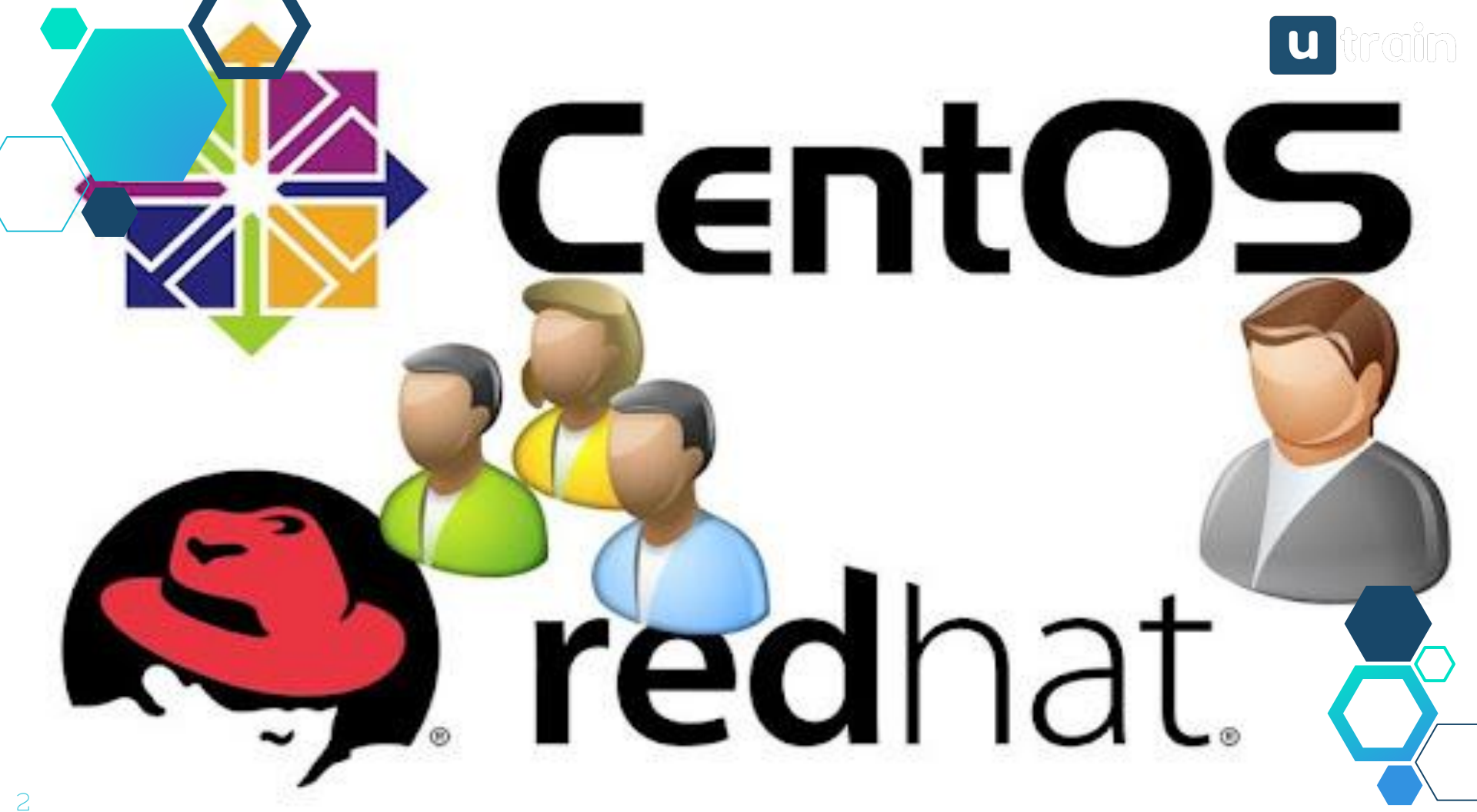

CentOS
redhat

Table of content

Introduction

Why do we need users and groups?

Introduction

- In companies, we have many groups of employees: managers, CEOs, security agents, etc.
- For employees to log in on the same computers but access different content and resources, something must be done.
- The companies can manage that by creating groups and adding each user to a specific group depending on its functions.

1

Creation of groups

How to create groups on a linux server

Important note!

Before starting this lesson,

- Launch your **Visual studio code**,
- Open a **terminal** or use the one that is opened
- Check the VMs on your computer: **vagrant global-status**
- Copy the **ID** of a **Centos 7 server**. If you don't have one, **please install it now**
- Resume or start a **Centos 7 server**: **vagrant resume ID** or **vagrant up** (this works just fine)
- Connect remotely to a **Centos 7 server**

That said and done, Let's get started!

Creation of groups

- To manage users on a linux server, we need to have root privileges
- That is, we must connect as the root user: $ **su root** then enter the password **vagrant**
- To create a group on the server, we use the command:
  - # **groupadd groupName**
- **Example:** to create a group called manager, we will run the command:
  - # **groupadd manager**

Creation of groups

- To verify that the group was successfully created, you can open a file **/etc/group.**
- This file contains all the groups created on the system.
  - **# cat /etc/group**
- At the end of the list of the various groups, you will see the name of the created group

```
postfix:x:89:
chrony:x:995:
vagrant:x:1000:vagrant
manager:x:1001:
[root@localhost vagrant]#
```

Creation of groups

- Since the file **/etc/group** is generally long, you can use the **tail** command to display just the **last 3 lines** of the file: **# tail -3 /etc/group**

 # cat /etc/group

```
postfix:x:89:
chrony:x:995:
vagrant:x:1000:vagrant
manager:x:1001:
[root@localhost vagrant]#
```

 # tail -3 /etc/group

```
[root@localhost vagrant]# tail -3 /etc/group
chrony:x:995:
vagrant:x:1000:vagrant
manager:x:1001:
[root@localhost vagrant]#
```

Creation of groups

- Let's create another group for **security agents**
  - **# groupadd security**
  - **# tail -3 /etc/group** to check if the group is created

```
[root@localhost vagrant]# tail -3 /etc/group
vagrant:x:1000:vagrant
manager:x:1001:
security:x:1002:
```

- Let's go to another section. How to add a user account to a group!

2

Creation of users

How to create users on a linux server

Add User to Group

Remove User from Group

Creation of users

- You may want to create a user account with specific options like :
  - A user in a particular user group
  - A user with a specific user id
  - A user with a specific shell

A **shell** is a program on the system that interprets the commands you run. There are many shells: **bash, korn, c shell etc.**

The shell we have been using so far is the **bash shell**

Creation of users

- To create a user on the system, we use the command: **useradd userName**
- Each user created on the system must belong to a group
- By default, when you create a new user account without specifying the group, **the system creates a new group with the user name provided for the account.**
- To check that, let's create a new user **# useradd serge**
- When you check the **/etc/group** file (**# tail -3 /etc/group**), you realise that a new group is also created with the name of the user.
- Then the user **serge** is added to the group **serge**.

Creation of users

- To check the group to which the user **serge** we have created belong, run the command: # **groups serge**

```
[root@localhost vagrant]# tail -3 /etc/group
manager:x:1001:
security:x:1002:
serge:x:1003:
[root@localhost vagrant]# groups serge
serge : serge
```

- This means a group called **serge** was created automatically

The useradd command options

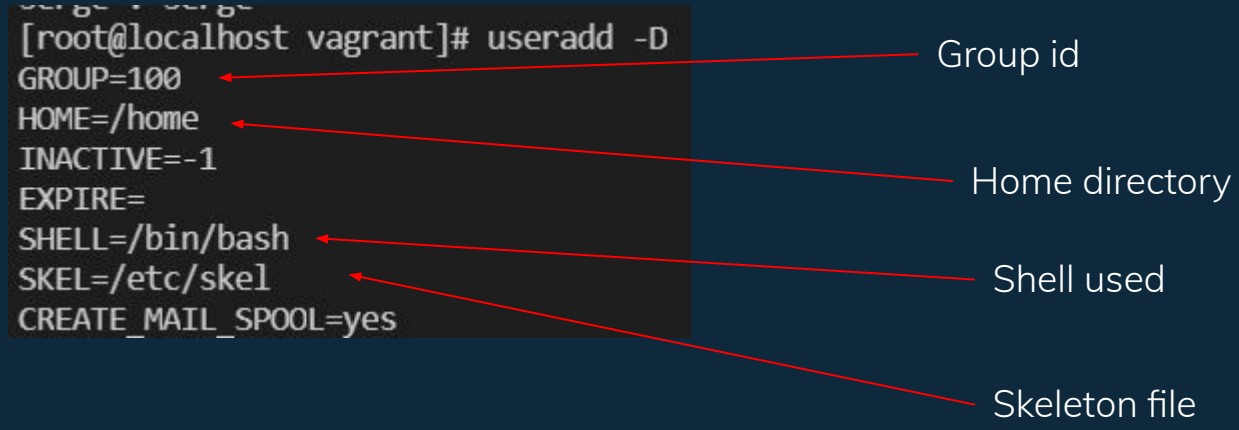
Creation of users

- Let's do some research on the # useradd command: # **man useradd**
- You can check the use of each option, but here we will describe only few options we will use

Options	Function
-c	Help us to give a description to the account (Ex:user's full name)
-g	Help us to specify the user's primary group
-G	Help us to specify secondary groups to which the user belongs
-m	To specify the user's home directory
-D	To know the default setup of the useradd command

Creation of users

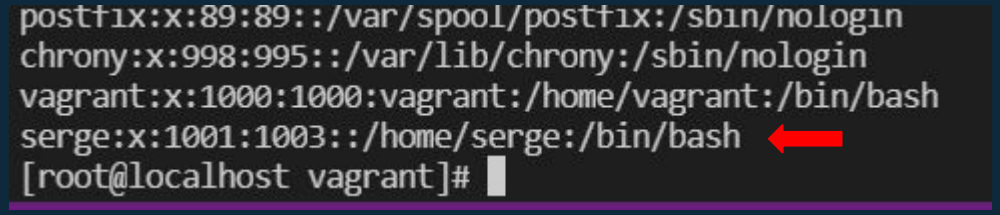
- Let's check the default setup of the useradd command: # **useradd -D**



The /etc/passwd file

Creation of users

- The **/etc/passwd** file contains the list of all the user accounts created on the system
- Thus, to check if an account was successfully created on the system, you can display the content of this file
  - # **cat /etc/passwd**



Creation of users

- Each line of the **/etc/passwd** file is made up of seven fields (separated by the :)

Fields	content
Field 1	The user name
Field 2	The password is stored in a different file (**/etc/shadow**)
Field 3	The user account id
Field 4	The user group id
Field 5	The user account description
Field 6	The user account home directory
Field 7	The shell used in the account (Ex: /bin/bash)

Types of user accounts on a system

Creation of users

- On our system, we have three types of accounts:
  - The system account
  - The root account
  - The regular user account
- If you go back in the **/etc/passwd** file, you will realise that there are many accounts differents from the root account and from the accounts that we created previously
- **Those are system accounts!**

Creation of users

- These accounts are **created by the system to run the server**
- You can notice that the shell used here is **/sbin/nologin**: **this means you cannot log in into these accounts**
- So if you create an account with the shell **/sbin/nologin**, you won't be able to login into that account. **Example:**

```
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

- By default, when you install the system, these accounts are created and blocked. Hence, **a hacker can't use such accounts to break into your system**

Users with specific options

Creation of users

- Now, let's create some users with specific options:

  # useradd john -c "John Jenkins" -g manager -G security -u 34567

- The command creates a user with the name **john**, description **John Jenkins**, belonging to the group **manager** (primary) and to the group **security**(secondary), and the **user id** is 34567
- The option **-u** helps us to **specify the user id**
- Run **# id john** and **# groups john** to check the id and the groups of the user account john

Creation of users

- Let's create another user with a **specific user id**
  - **# useradd manola -u 35678**
  - **# tail -4 /etc/passwd**
- Create another user with a **specific shell**
  - **# useradd john1 -s /bin/sh**
  - **# tail -4 /etc/passwd**
- You can use:
  - **# id** to check the id of the current user
  - **# id john1** to check the id of **john1**

Creation of users

- When we create user accounts on our system, a folder is created in the /home directory for each user
- When we list the content of the **/home directory**, we have:
  - **# cd /home**
  - **# ls**

```
[root@localhost vagrant]# ls /home
john  john1  manola  serge  vagrant
```

- All the users we created have a folder in the **/home** directory
- When you log in as a user and type **# pwd**, the result will always contain the **/home** directory in the path: **/home/username**

Creation of users

- **Example:** Switch to user **serge** and run the commands
  - **# pwd**
  - **# su serge**
  - **$ cd ~**
  - **$ pwd**
- Switch back to the root user
  - **# su** and in the **/home** directory (**cd ..**)

3

Delete a user account

How to delete users on a linux server

Delete users

- When you don't need a user account on your server anymore, you can delete it
- Let us say we created a user called student on our system and now we do not need it anymore: # **useradd student**
- To delete a user account, we use the command: **# userdel username**

Example: Let's delete the user **student** from our server: **# userdel student**

- After this, you can't login as a **student** anymore **but the folder student is still present in the /home directory** (**# ls /home** to check)
- The command **# userdel** deletes a user account, **but keeps its /home directory**

Delete users

- To delete a user as well as its home directory folder, we will add the option **-r** to the **userdel** command

Example: Let's delete the user **john1** from our server as well as its home directory: **# userdel -r john1**

- Check the **/home directory** (**# ls /home**)
- To delete a user account even when the user is still logged in, we use the option **-f** (for force)
- To get more informations on the userdel command, you can check its manual: # **man userdel**

Always remember this!

While learning, when you get stuck on something, don't panic.

- When you don't understand how a command works, always think of reading its manual: man commandname
- You can also make some research on your own either on **Google** or on **Youtube** to find tutorials and videos giving more explanations that will allow you to better understand these notions!

4

Modify a user account

How to modify a user account on a linux server

Modify user accounts

- When you realise that you have made a mistake while creating a user account, you can still modify it.
- To modify a user account, we use the command
  - **# usermod parameterOption username**
- This command has similar option with the **# useradd** command
- **Example:** Let's modify the description of the user **john** we created previously (The description was the user's full name **John Jenkins**)
  - **# usermod -c "John Kamga" john**

Modify user accounts

- The description was the user's full name John Jenkins
  - **# tail -5 /etc/passwd**
- Let's change it to John Kamga
  - **# usermod -c "John Kamga" john**
  - **# tail -5 /etc/passwd**

```
[root@localhost vagrant]# usermod -c "John Kamga" john
[root@localhost vagrant]# tail -5 /etc/passwd
vagrant:x:1000:1000:vagrant:/home/vagrant:/bin/bash
serge:x:1001:1003::/home/serge:/bin/bash
john:x:34567:1001:John Kamga:/home/john:/bin/bash
john1:x:35679:35679::/home/john1:/bin/sh
manola:x:35680:35680::/home/manola:/bin/bash
```

Modify user accounts

- Now, let's modify the **user shell**
  - **# tail -5 /etc/passwd**

```
serge:x:1001:1003::/home/serge:/bin/bash
john:x:34567:1001:John Kamga:/home/john:/bin/bash
john1:x:35679:35679::/home/john1:/bin/sh
manola:x:35680:35680::/home/manola:/bin/bash
```

- Let's change it to **/sbin/nologin: # usermod -s /sbin/nologin john**
  - **# tail -5 /etc/passwd**

```
serge:x:1001:1003::/home/serge:/bin/bash
john:x:34567:1001:John Kamga:/home/john:/sbin/nologin
john1:x:35679:35679::/home/john1:/bin/sh
manola:x:35680:35680::/home/manola:/bin/bash
```

- you can no more login into this account if this shell is not changed

Modify user accounts

- To change the user name of the user account **manola** we created previously
  - **# usermod -l manolo manola**
  - **# tail -5 /etc/passwd**
- The name changes from **manola** to **manolo**

```
serge:x:1001:1003::/home/serge:/bin/bash
john:x:34567:1001:John Kamga:/home/john:/sbin/nologin
john1:x:35679:35679::/home/john1:/bin/sh
manolo:x:35680:35680::/home/manola:/bin/bash
```

5

Set user password

How to set user password on a linux server

u train

Modify user accounts

- **Exercice:** we want to modify the password for the users **serge**, and **john**
- To do that, we use the command **# passwd username**
  - **# passwd serge**
  - Enter the new password: **school1**
- Retype the password for confirmation: **school1**
- Do the same with the user **john**(**# passwd john**)
- Login as user John with # su **John** and enter the password you specified

Modify user accounts

- You can check other options in the manual with **# man passwd**
- We have interesting options here:

Option	Function
-l	To lock a user account
-u	To unlock a user account
-d	To delete a password for an account

“ Creating groups is an efficient way to manage employees in a company’s system

Users can have access to specific content on the system, depending on the group to which they belong!. ”

“ Play around with these commands. You don’t need to memorise all the options for a command. Just practise them and make sure you read the manual when you get stuck on a command

”

See you guys in the next lesson!

Thanks!

Any questions?

You can find us at:

website: http://utrains.org/

Phone: +1 (302) 689 3440

Email: contact@utrains.org

Click on the link below to contact the support team for any issue!

utrains.org/support/

Create a ticket for your problem and we will get back to you soon!