## Redhat 8 Linux CMD

1. ip a => obtiene la ip que utilizaremos para realizar la coneccion por ssh.
Ejemplo: inet 192.168.2.7/24
2. set | grep Display => muestra el valor de una variable en este caso la variable Display
3. scp  music.zip  root@192.168.100.4:/home/infomarcy/Documents => Move a file through SSH
4. Ctl + a => go to the beginning of the line
5. tail -f /var/log/server.log => follow the log in real time
6. Init 6 => restart the machine
7. shutdown -h +60 =>  turn off the machine in 60 minutes
8. poweroff => turn off the machine
9. w or who => show the users that are logging in the machine at the moment, and what are they doing.
10. top : Monitor processes. Show which process is using the most cpu.
11. netstat -tupln => show what is happening on the network
12. mkdir -p dev/com/java => crea subdirectories
13. cp test.txt respaldo/ => copy command
14. mv text.txt mytext.txt => rename a file
15. echo "this should be in a file" > newfile.txt => create a file output with the text enter :: override the file
16. echo "This should be on line 2" >> mytexy.txt => append the text to the file
17. rm -rf <caperta eliminar> => elimina una carpeta y sus subcarpetas
18. cat file.txt | grep user => find the line that matches what you are looking for
19. grep someone ./* | uniq => find all the files in the directory which have the word someone and show the file just once.
20. grep someone ./* | uniq | cut -d: f1 find all the files in the directory which have the word someone and show the file just once and show just the files
21.  chmod 777 => grant all permisions
22. Whoami => show the owner or user
23. tail /etc/passwd => show the users in the OS
24. kill -l  => list all the process that can be terminated
25. kill -9 PID => terminate a process
26. Sudo pkill -u user =>  Terminate all processes running by an specific user
27. crontab => scheduling task using time
28. tar -zcvf docs.tar.gz Documents/ => zip and create a compressed .tar.gz file
29. mv file.zip Documents/ => mueve un archivo al directorio seleccionado
30. tar -zxf docs.tar.gz => extract the compressed content
31. wc -l  < testfile=> count the number of lines in a file. In this case our filename is testfile
32. cat testfile | wc -l => count the number of lines in a file. In this case our filename is testfile
33. (cat testfile | wc -l) && echo "Done!" => if the first part execute sucessfully then the second part will be executed, otherwise the second part won't get executed.
34. ls ldjjfkdkdhddh || echo "success!" => OR operator. Even if the first one fails the second one will get execute
35. myvar="This is so wonderful" => variables has to be declare with the equal sign(=) with no spaces inbetween. (some_number=10)

echo "This is my ${some_number}th beer.

36. echo "There are `wc -l < /etc/group` of lines in the /etc/group file" => we can declare commands using the ``.

(num_lines=`ls $HOME | wc -l`)

37. Ctrl + d => logout of the shell or get out of ssh
38. Ctrl + l => Clear the screen
39. Ctrl + e => go to the end of the line
40. Ctrl + a => go to the beginning of the line
41. chmod +x test.sh => add the excute bit
42. bash test.sh => run the test.sh file
43. ls -la => list with the owners properties
44. ls - al=> find all files including the hiddens ones.
45. echo $PATH => path variables
46. which sudo => shows the path of the sudo command
47. PATH=$PATH/somedir => asign a new value to a path
48. alias lr="ls -lra" => create an alias to call the commands more easily
49. lsof -i :8080 => which process is listening on port
50. systemctl status mysql => check the status od the service

## Install tomcat

51. wget url.zip => download a file using the url that contained that file
52. sudo yum install unzip => install unzip to be able to extract the zip file
53. unzip filename.zip => unzip a file using command line
54. sudo mkdir -p /opt/tomcat => create a directory tomcat inside the opt directory
55. sudo mv apache-tomcat /opt/tomcat/ => copy the tomcat folder inside the newly create directory for tomcat
56. cd opt/tomcat/apache-tomcat-folder/conf/ server.xml :: <Connector port="9090" => change the tomcat port to 9090
57. cd /opt/tomcat/apache-tomcat-folder/bin, ls -la, sudo chmod +x * => set the executable permission to all of the script that are present in the current directory
58. cd /opt/tomcat/apache-tomcat-folder/bin, ./startup.sh => start the tomcat server
59. cd opt/tomcat/apache-tomcat-folder/conf/ tomcat-users.xml

At the end of the file uncomment and change this

<role rolename="manager-script"/>

<role rolename="manager-gui"/>

<user username="tomcat" password="pass" roles="manager-script,manager-gui"/> => provide access and permision to jenkins server to start and stop the tomcat and deploy the artifact on tomcat

60. cd opt/tomcat/apache-tomcat-folder/webapps/manager/META-INF, context.xml, comment the <Valve className /> => configure tomcat to allow the users to deploy the war
61. cd bin/, ./shutdown.sh, ./startup.sh => restart the tomcat server to apply the changes
62. cd /opt/tomcat/, cp -r apache-tomcat-folder-prod=> copy the tomcat folder to run a second tomcat server on the same machine, change the port number on conf
63. sudo -iu <username> => change user using ssh

## Login in SSH without password

1. sudo -iu <username> => change user using ssh. sudo -iu mgarcia
2. ssh-keygen -t rsa => configure the ssh between the master and the node without the password, we generate the rsa certificate which uses public and private keys
3. ssh root@<slave node ip> => do ssh from the master to the slave node
4. sudo mkdir -p .ssh => create the .ssh directory which will hold the private and public keys
5. ctrl +d => go back to the master node
6. cat id_rsa.pub | ssh root@192.168.100.30 'cat >> .ssh/authorized_keys' => copy the public key of the master node on the slave node authorize key inside the .ssh directory
7. ssh root@<slave machine> => login without password

## User Account management

1. useradd -m -d /home/desarrollador -s /bin/bash desarrollador => create a new user
2. passwd desarrollador => will prompt to create a password
3. tail /etc/shadow => show the users with its logging permissions
4. usermod -L desarrollador => block the account
5. usermod -U desarrollador => unblock the account
6. userdel desarrollador => delete a user then the home directory needs to ne deleted rm -rf /home/desarrollador/

## ReadHat 8

1. sudo yun update => update all the packages on the system
2. sudo reboot => restart the machine
3. sudo yum groupinstall "Development Tools" => install the development tools for the Os
4. sudo yum install kernel-devel elfutils-libelf-devel

## VIM tutorial

1. vim prueba.txt => crea un nuevo archivo llamado prueba.txt
2. i (insert) => habilita el modo de edicion para empezar a escribir
3. Para salir del modo de edicion oprimimos la tecla Esc
4. Nos movemos al inicio con las fechas del teclado + V para seleccionar texto
5. Cortamos y pegamos presionando la tecla p
6. Esc :w guarda los cambios del documento
7. :q podemos salir del editor
8. :wq! => salir y guardar cambios
9. :set number => show the line in the document
10. Esc + dd => delete a line
11. Esc + u => undo changes
12. Esc + Ctrl + r => redo changes
13. /search_a_word => find all the matches
14. Esc + :%s/word_to_replace/replacement/g => search and replace all the ocurrences

Esc + /Marcello/Marce/g

# Install Java

1. sudo dnf localinstal javaFileName.rpm => install java from rpm file
2. JAVA_HOME=/usr/java/jdk1.8.0_201 export JAVA_HOME => set Java home variable

# Jenkins

1. systemctl start | stop | status | enablel jenkins => first start jenkins service, check the status and then enable the jenking sercices
2. firewall-cmd --permanent --add-port=8080/tcp => enable the firewall for jenkins in port 8080
3. firewall-cmd --reload
4. Sudo gitlab-ctl restart nginx
5. Install git on the machine where jenkins is running
6. Github integration plugin to add github to jenkins
7. sudo yum install maven => Install maven on the machine where jenkins is running (mvn --version)
8. which git => find the path instalation of got
9. Install the plugins Copy artifact & deploy containers => copy the artifacts from job A To job B, deploy containers deploy some artifacts on tomcat
10. Plugin jenkins Job DSL => help create the jenkins job on the fly with the help of the code file

# Build

1. Invoke top-level Maven targets

Maven Version => LocalMaven

2. Goals => test install
3. Goals => clean package
4. POM => maven-sample/single-module/pom.xml

# Archive artifacts

1. Post-build action => archive the artifacts
2. Files to archive => **/*.jar => ** means all available path in our workspace, *.jar means all the files which has the jar extension
3. Build others projects => Projects to build => job name and select Trigger only of build is stable

# Build Pipeline plugin

# Configure Jenkins Slave on the Master

1. Manage Jenkins => Manage Nodes and Clouds => New Node => <node name> JenkinsNode1 => Check Permanent Agent => OK
2. mkdir jenkinsNode1 => create a directory on the slave machine to save all the files that the jenkins slave will use

**3. Configure the slave node**

Name : JenkinsNode1

Description : Jenkins Slave 1

# of executors : 2
Remote root directory: /root/jenkinsNode1
Labels: CentOS8
Usage: Use this node as much as possible
Launch method: Launch agent via SSH
**Host:** <ip of the slave machine>
**Credentials**:  username =>  root,  password => you password for ssh
**Host Key Verification Strategy :** Manually trusted key Verification Strategy

## Configure Jenkins Job to run on an specific label

1. General => Restrict where this project can be run : <label name> CentOS8 => this job will run on the Jenkins Nodes which has the label CentOS8

## Install RPM file

1. sudo dnf install nodejs => install nodejs
2. sudo dnf localinstall filename.rpm => install an rpm aplication
3. hosname -i => show the hostname and ip address

## Permissions

1. ls  -al *.sh => find all the files with ends with .sh and show their permissions
2. chmod +x *.sh => assign executable permisions to the file ending in .sh
3. sudo chmod -R 777 homebrew/ => dar permisos a la carpeta
4. lsof -i:8080, kill -9 PID => kill a process