OS : Debian

Latest Version : V 11 (Bullseye)

Features : Stable,Version Stay for Long time,ideal for Servers ,Rolling release option,support many pc Architectures ,DVD CD and ISO available ,java friendly

Most Stable Version :V 11 (Bullseye)

OS : CentOS

Latest Version : Centos 8.5-2111

Features : Very Stable,Free,Open Source ,Run by the community , Same like Red hat but free and no support

Most Stable Version :CentOs 8

OS : RedHat

Latest Version : RHEL 8.5

Features : Run by RedHat Org, Not free, long releases ,Stable ,easy configuration,contains development tools

Most Stable Version :RHEL 8

OS : Ubuntu

Latest Version : Ubuntu 20.04.3 LTS

Features : Ubuntu is free and an open-source Os, Secure ,it Runs without install ,Customizable,Resource Friendly ,can be used for PC ,can be used for servers too

Most Stable Version : Ubuntu Server 20.04

OS : Fedora

Latest Version : fedora 35

Features : Fedora OS offers many architectures ,very reliable and stable Os,it has unique security features ,provides a very powerful firewall,easy to use ,large community,actively developed

Most Stable Version : fedora 35

OS : suse

Latest Version : SLES 15 SP3

Features : Centralized administration and deployment tools,Support for network boot and diskless operation,Simplified installation process for installing and configuring the point of service environment,YaST graphical user interface for creating images

Most Stable Version : SLES 15 SP2

OS : kali Linux

Latest Version : 2021.4a

Features Full customisation of Kali ISOs,Live USB Boot,Kali Undercover,The industry standard,free

Most Stable Version : 2021.4a

Commands

cd -> used to navigate and go back to home

cd .. -> used to go back

cd / -> used to go back

cd /home/dev -> absolute path

cd dev -> relative path

ls -> is used to look at files in current directory

ls -l -> is used to look files with details

mkdir -> is used to create a directory

pwd ->is used to see current path

cat -> used to read file

whoami -> used to see who using the system

exit -> is used to exit

history -> is used to look at all the commands we used

sudo -i -> is used to switch to root user

mkdir name1 name2 -> can create multiple directories

touch -> is used to create a files like txt

touch test{1..4} -> is used to create a files with name test1 test2 test3 test4

cp filename path -> is used to copy file and paste it at given path

cp -r -> is used to copy directory

cp --help -> is used to see all the options available

mv filename path -> is used to move file to a path

mv oldname newname -> is used to change name of the file

mv -r -> is used to move directory

mv *.txt -> is used to move all files with .txt

rm filename -> is used to remove the file

rm -r -> is used to remove the directory

rm *.txt -> is used to remove all files with .txt

rm rf -> is used to force delete anything

file filename -> is used to see file type

mkdir -p /home/opt/dev/test -> used to create entire path if not exist before

ln -s /path/file /home/links -> is used to create link to directly access file content

ls -lt -> sort the list of items in directory

ls -ltr -> to reverse sort

grep firewall anaconda-ks.cfg -> used for the searching word firewall in cfg file

grep -i -> used to remove case sensitive

grep -i firewall * -> used to search all the files in directory

grep -iR firewall * -> is used to even look in directory inside a directory

grep -v ,grep vi -> is used to look for all other words except it

less filname.cfg -> used to read and search (/word)the content in file

more filename -> is used to read file and also show percentage of the file

head -10 filename.cfg -> is used to read first 10 lines of the file

tail -2 filename.cfg -> i used to see last 2 lines of the files

tail -f message -> is used to see dynamic log

cut -d: -f1 /etc/passwd -> used to see all the values in field 1 (first column)

awk -F':' '{print $1}' /etc/passwd -> advanced search

sed 's/Tarun/Tar/g' file.txt -> to change name but won't be changes in file

sed -i 's/Tarun/tar/g' file.txt -> to actually make changes in file

echo -> is used to print text

uptime > /temp/sysinfo.txt -> redirecting the output into a file sysinfo.txt

date >> /temp/sysinfo.txt -> used to append the output to sysinfo.txt

free -m -> used to see memory details

df -h ->hard disk details

date -> used to check date

path /dev/null -> its like a black hole anything redirected to this path will be gone

freee -m 2> /temp/error.txt -> is used to redirect the errors

date -m &> /temp/error.txt -> is used to redirect anything

wc -l file.txt -> is used to count no of lines in file

ls | wc-l -> used to count the files in directory

ls | grep host -> used to search the files

find /etc -name host * -> to find filename

id vagrant -> is used to look user details

useradd name -> used to add user

groupadd name -> used to add group

usermod -aG groupname username -> to assign user to group(secondary group)

usermod -ag groupname username -> to assign user to group(primary group)

passwd username -> to add password for user

su - username -> to login as a user

last -> to see login list

who -> to see current login user

userdel -r username -> to delete the user

groupdel -r groupname -> to delete group

chown user:group /opt/devopsdir -> to select the user

ls -ld /opt/devopsdir -> to see user permissions

chmod 640 /opt/devopsdir -> to change xwr for the file(numeric)

chmod rw- /opt/devopsdir -> to change xwr for the file (char)

visudo -> used to edit the sudo to an user

cd /etc/sudoers.d/ -> we can cp the exist file and change the sudo(for group use %group name)

Yum install vim -y -> for installing vim editor

Manual package install

We can't install this using yum

Go to google and type tree rpm

Click rpmfind.net

Look for centos 7.9 version since i'm using this

Copy the link by right clicking it

Go to vm root user

Packages like tree ,httpd

Package Commands

curl link -o link name ->to install

rpm -ivh -> command to install in human readable format

rpm -qa -> will list all the rpm

rpm -e rpm name -> to erase the rpm

cd /etc/yum.repos.d/ -> in this directory repositories will be there

yum search httpd -> to search for a packages

yum remove httpd -> to remove the packages

yum install wget

yum update -> it will upgrade all packages

yum reinstall package -> to reinstall

Services will be running in servers we can start and stop those services

systemctl status httpd -> used to look service status

systemctl start httpd -> used to start the service

systemctl restart httpd -> used to restart

systemctl reload service -> used to restart

sudo reboot -> will reboot the vm

systemctl enable httpd -> used to auto start the service when we restart to start the vm

systemctl is-active or is-enabled httpd -> used to check if active or not or enabled or not

Cat /etc/systemd/system/multi-user.target.wants/httpd.service

top -> is like task manager in windows(ps aux or ps -ef also same)

kill processNo -> used to stop process

Kill -9 processNo -> used to stop force

To kill all child processes

Ps -ef | grep httpd | grep -v 'grep' |awk '{print $2}' | xargs kill -9

cd /var/log/ -> to look at logs

tar czvf jenkines_06122020.tar.gz path of file -> c-create z-compress v-verbos f-file to archive the file

tar xzvf jenkines_06122020.tar.gz where to save it-> to extract the file

yum install zip unzip -y -> is used to install zip and unzip

zip -r zipfilename.zip file -> used to zip the file

unzip filename -> to unzip the file

UBUNTU COMMANDS

adduser -> to add a user / vim /etc/ssh/sshd_config -> go here and we can change pass yes

export EDITOR=vim -> to add vim editor

wget link -> to get package file

dpkg -i tree(filename.deb) -> to install the package

dpkg -l -> to list debian packages

dpkg -r filename.deb -> to uninstall the package

apt install package -> same like yum can do search reinstall remove

apt upgrade -> to upgrade the packages

apt purge package -> used to remove package and also its data(configuration files)

VAGRANT COMMANDS

vagrant global-status -> used to see details about vms (vagrant global-status --prune) to remove unwanted info and run this command again

vagrant status -> used to see status of vms

ls -a -> used to see all files even hidden (touch .file) use “.” to create the hidden files

We can edit vagrant file by notepad ++ and Inside vagrant file we can edit cpu ram or ips

```
#   vb.memory = "1024"
```

vb.cpus=2 -> used to set no of cpus

cd /vagrant/ -> in vm if we search this the directory will be same in vagrant it is synch directory

Where you can put ur vm data and you can access that data in your windows pc that way you can backup your data if ur vm get corrupt

Touch bash.sh -> to create a bash file

To change synch directory path go to vagrant file edit and change config windows path that will be on left user (\\) as a path for windows path

```
config.vm.synced_folder "../data", "/vagrant_data"
```

Provisioning is running a script when creating a vm so anything written between this shell will be executed provisioning will only work first time install if you want to execute them in already installed vm use (vagrant reload --provision)

```
config.vm.provision "shell", inline: <<-SHELL
  yum install vim -y
  mkdir -p /etc/test
  SHELL
```

HOSTING A WEBSITE IN LINUX

Go to tooplate.com and download a html code

Then create a new folder and get vagrant file and vagrant up it

Install unzip wget httpd packages

Go type ifconfig or ip addr show

Then select private or public ip

Pate it on chrome to see default centos page

cd /var/www/html/ -> this is where you should put your html files in linux

Go to that path and create vim index.html

Write something and save

systemctl restart httpd

Now check that private ip page you will see the text

Now go back to tooplate.com and hit f12 and go to network and click the download button of html page then u see the download link in network copy it

wget copied link -> to download the html file

unzip file .zip

Move this files to /var/www/html/

then restart the httpd and check ur private address

Vola you just created a server

WORDPRESS IN UBUNTU

Create a new ubuntu vm with private and public network enables

Go to https://ubuntu.com/tutorials/install-and-configure-wordpress#2-install-dependencies

Install all the required packages

cd /srv/www/wordpress/ -> here you can see all wordpress files

Then go to link and follow the configure apache for word

cd /etc/apache2/sites-enabled/ -> to check what are enables sites

At page 3 replace password with what u have set

AUTOMATE THE WEBSITE CREATION

We will use vagrant provisioning to automate the entire process this is called IAAC (Infrastructure as a code)

Just go to vagrant file and edit with notepad ++ change network configs put the commands instead of manually creating for every site we can just automate by changing wget link and uzip names

```
config.vm.provision "shell", inline: <<-SHELL

    yum install httpd wget unzip -y

        systemctl start httpd

        systemctl enable httpd

        cd /tmp/

        wget https://www.tooplate.com/zip-templates/2119_gymso_fitness.zip

        unzip -o 2119_gymso_fitness.zip

        cp -r 2119_gymso_fitness/* /var/www/html/

        systemctl restart httpd
```

SHELL

Go to https://www.vagrantup.com/docs and select getting started tutorials to see all vagrant commands

Multi vm vagrant file

Go to https://www.vagrantup.com/docs/multi-machine and copy this code

```
Vagrant.configure("2") do |config|

config.vm.provision "shell", inline: "echo Hello"


config.vm.define "web" do |web|

  web01.vm.box = ""vmname

 end


 config.vm.define "db" do |db|

  db01.vm.box = "vmname"

 end

End
```

For detailed info go to c/work/Devops udemy and multi vms see the file

VPROFILE PROJECT

Copy the git code url

Go to git bash

git clone url -> used to clone the repository

git checkout branch_name -> to check the branches

vagrant plugin install vagrant-hostmanager -> used to connect all vms in multivms to same host

vagrant ssh vmname ->for accessing multi vms

cat /etc/hosts -> to see host names

ping vmname -> used to ping vms in same host

To make a variable permanent create a variable DATABASE_PASS='admin123'

vi /etc/profile -> paste this variable at bottom

source /etc/profile -> to make it permanent

yum install git mariadb-server -y

mysql -u root -p -> to login to root of mariadb

show databases; -> used to see databases

use databasename; -> to switch to that database

Show tables; -> used to see tables

ss -tunlp | grep port -> used to check the ports

rabbitmqctl add_user name passwd -> to create a user in rabbitmq

rabbitmqctl set_user_tags username tagname -> used to give a tag for user

vi /etc/hostname -> to change host name

hostname chnaged_name -> to confirm and restart

hostname -> to check hostname

Check your pdf of vprofile project to understand steps

BASH SCRIPTING

vim name.sh -> to create a script file

In side script start with #! /bin/bash

Variablename =”string” -> used to store value in variable

$variablename -> used to access that variable

$1 -$9 used to pass command line argument

vim test.sh

In test.sh

echo “first arg”

echo $1

echo “second arg”

echo $2

Before execute change the chmod +x filename

Now to pass the values

Give path and scriptname

test.sh hello

test.sh world

DevOps (visualpath.in)

$0 -> shows name of the script

$#-> how many arguments are passed to bash script

$@ -> all the arguments passed

$? -> the exist status of most recent process if it works output will be 0 else some no

$$ -> process of current script

$USER $HOSTNAME $SECONDS $RANDOM $LINENO

“In this $variable” -> in double quotes variable works

‘ in this $variable”-> in single quotes variable doesn't work

“In this \variable”-> when we use \ the special character will be ignored

time=`uptime` -> it will take the output and store it in variable

time =$(uptime) -> this do same thing too

export variable name -> used to export the variable to child shell which mean inside scripts

ls -a -> used to see hidden files

.bashrc -> if u vi .bashrc and put variable there it will work even after re login for single user

vim /etc/profile -> if u put variable in this it will be available for all users

read variable -> used to take input for that variable

read -p username: variable -> used to show message before taking input and store the value in variable

read -sp password: variable -> to hide user input

:se nu -> used in vim editor to give line nos

/word -> used in vim to look for a word

CONDITIONAL STATEMENTS

If [$num -gt 100]

then

echo “we are in ture condition “

else

echo “we are in else condition”

fi

grep -ic name -> used to count no of times it occurred

gt ,eq,lt -> greater than equal to less than

If [condition]

then

Code

elif [condition]

then

code

else

code

fi

! -> used as not (!)

-n string ->if length of string is greater than zero

-z string -> true if length of string is zero

= -> equal to

!= -> not equal to

-d file -> file exist and its a directory

-e file -> file exist

-r file -> file exist and read permission is granted

-w file -> file exist and write permission is granted

-x file -> file exist and execute permission is granted

s file -> file exist and its size is greater than zero

0 is true in bash scripting and non 0 is false

cd /var/run/httpd/httpd.pid-> here the process running will have a pid

crontab -e -> here we have to set path of our script and we have to give time and everything

Min Hour Date Month DayOfWeek

30 20 * * 1-5 /path of script &>> /var/log/monit.log

sleep no-> used to make execution stop for that many seconds

LOOPS

for var1 in java .net python .ruby

do

echo “$var1”

done

```
counter=0
while [ $counter -lt 5 ]
do
   echo “value of counter is $counter“
   counter=$(( $counter + 1))
done
```

true and false also works //for host in `cat hosts.sh` ;do echo "$host";done -> for loop in vm

Remote access among multi vms

First create a multi vms

now give host names for each vm

vim /etc/hosts -> here add ip and hostname

From this vm use

ssh vagrant@web01 -> to login to a vm from this vm

now login to each centos vm and add user named devops

And also give user sudo permissions

For ubuntu we can’t login directly from vm so we have to manually open it and go to

vim /etc/ssh/sshd_config

:su nu -> to set serial no

/password-> used to search for the world in vim

Now find the password authentication and change it to yes and do systemctl restart ssh

Now you can login into the ubuntu from vms using password

Now also create user devops in ubuntu

Now try using ssh devops@web01 uptime

This will get the task done without even entering to the particular vm

ls .ssh/ -> here keys and locks will be stored to access use ls -a

ssh-keygen -> is used to create a key based login

ssh-copy-id devops@web01 -> used to put the lock into vms and use key to access them

scp filename devops@web01:/tmp/ -> used to push file from one vm to another vm

NOTE : you should use sudo commands or path in scripts because you can't access root

The communication between two or more network interface is called computer network

OSI(Open system interconnection) is model of communication developed by ISO

Label on Column	Service Name	UDP and TCP Port Numbers Included
DNS	Domain Name Service – UDP	UDP 53
DNS TCP	Domain Name Service – TCP	TCP 53
HTTP	Web	TCP 80
HTTPS	Secure Web (SSL)	TCP 443
SMTP	Simple Mail Transport	TCP 25
POP	Post Office Protocol	TCP 109, 110
SNMP	Simple Network Management	TCP 161,162 UDP 161,162
TELNET	Telnet Terminal	TCP 23
FTP	File Transfer Protocol	TCP 20,21
SSH	Secure Shell (terminal)	TCP 22
AFP IP	Apple File Protocol/IP	TCP 447, 548

NETWORK COMMANDS

ping ip -> to ping to that ip

vi /etc/hosts -> inside (192.168.10.12 (ip) name) this is used to create a name for an ip you can ping with that name

tracert www.google.in -> shows all the routes it takes to reach google server used to see latency

netstat -antp -> used to see all ports and processes that are using them

ss -tunlp -> same as above

nmap -y -> is used to install commands (it is illegal in some countries so use for troubleshooting purpose only)

nmap localhosts -> used to see opened ports

nmap ip or name -> to see what ports are open in other vms

dig www.google.in -> it shows the ip address of the website

nslookup www.google.in -> it shows same as dig

route -n -> it used to show gateways

arp -> is used to look and add it to kernel table

mtr www.google.com -> show all the hops in live

telnet ip port -> used to connect to that ip via port

Ifconfig

AWS EBS EXTRAS

fdisk -l -> is used to see all volumes

df -h -> is used to see which volume is mounted to which one

Partitioning the disk

fdisk /dev/xvdf -> used to go to partitioning menu click m for help

Press n for new partition

Then press p

Press 1

Click enter

Last sector you can specify the size like +3G or just enter again (p to print and w to confirm)

To format the disk

mkfs -> and click tab twice

You see the options select

mkfs.ext4 /dev/xvdf -> to select the volume to format

To mount the disk

mount /dev/xvdf /var/www/html/images/

df -h -> to check

umount /var/www/html/images/

To permanently mount the disk

vim /etc/fstab -> go in here and put /dev/xvdf /var/www/html/images ext4 defaults 0 0

Now mv the images to this /var/www/html/images/

If website images is missing too fix that go to

vi /etc/selinux/config -> here change the selinux enforcing to disbaled

To unmount the disk

umount /var/www/html/images/ (umount -l -> to force unmount not recommended)

If not stopped and it is being used by web service

yum install lsof -y

cd /var/www/html/images/

lsof /var/www/html/images/ -> to see the process using them and kill it

cd

lsof /var/www/html/images/

mount /var/www/html/images/

Once we unmount it we can detach it in aws