How to create AWS account ???

email address , mobile number , Debit card (master / visa)

Google ===>> aws console login ==>> AWs management console ===>> sign into console ==>>> create new account ==>> email : xyz , password 1243 , conform password : 1234 ==>> Aws account name : ==>> continue..

AWS free-tier (select) ===>> personal (select) or proffesional/ business acount ===>>> address details ==> d.no , streect , pincode , state ,near landmark..

==>>> debit card details ==>> 16 digits number ==>> CVV ===>>> OTP (2/-) ===>>> do you have pancrad : no ===>>

AWS ==> verification ==>> mobile number ===>> voice message / text message (select) ==>> 4 digits (5896)

my role is : student

you are intestrd in : other..

AWs console login ==>> sign in ==>> email id with password ===>>

AWS account ==>> imediate / 24 hours..

==========

Linux servers : we will create linux servers in AWS account

Linux servers to connect ===>> we need to two softwares or applications install in our laptop.

1. gitbash

2. putty

==========

Linux : it is an operating system like windows..

Linux is process oriented operating system..

Datacenter : physical linux servers ===>> hardware ===>> o.s install (linux) ===>> application and databse installl ==>> APP ===>> EU.

AWS ==>> Cloud ===>>> AMI (amazon machine image) ==>> Ec2 instance ==>>> application and databse installl ==>> APP ===>> EU.

Unix : operating system.

Unix 4 types of flavours :

1. sun solaris 2. Redhat Linux (open source and free of cost) 3. IBM-AIX 4. HP-UNIX..

Other remaining these three are enterpraise versions ===>> License purhase ===>>>> manadatory..

windows :

C:/ : operating system install ===>> Admin user.

GUI mode operations (Graphical user interface) ===>> clicks.

files and folders

NTFS filesystem (new technology file system)

Linux :

/ ==>>> root ===>>> operating system install ===>>> rootuser / parent user / super user / Admin user.

CLI mode operations : (Command Line interface) ===>>> commands to type.

files and directories

ext2 , ext3 , ext4 (latest) ==>> file systems.

ext2 ===>> second extended file system.

ext3 ===>> third extended file system.

ext4 ===>> fourth extended file system.

==========

ec2 : elastic cloud compute : ec2 ==>>> virtal machine. ==>>> ec2 instance.

AMI ==>> Amazon machine image

Every AMI has their own identification number ===>> AMI ID.

Every operarting sysrem has their own AMI. ==>> o.s install.

security group :

It is a virtual firewall at ec2 instance level..

it contains set of rules..

every application has their own port number..

all ports in between ==>>> 0 to 65535

ec2-user (default user) ===>> able to login we need to add a rule ==>> ssh (mandatory)

ssh ==>> port number ==>>> 22.

http ==>> port number ==>> 80

mysql ==>>> database ==>> 3306

==========

when ever you created a ec2 intsance ==>> then automatically two Ip addresses will come.

1. public IP address : ec2 instance to login and application enduser to access.

this is visible only in AWs console dashboard.

2. private Ip address : it is used to internal communication.

this is visible in both AWS console dashboard and ec2 instance.

==========

keypair ===>>> ramakrishna ===>> download ===>>> extension ====>>> ramakrishna.pem ==>> pemfile.

pemfile conatins privatekey

afetr launching the ec2 instance ===>> defaultly create publickey key .

privatekey and publickey -->>> match --->>> default user able to login into ec2 instance (ec2-user).

==========

==========

Linux basic commands :

Files and directory operations :

Files operations :

cat > filename

I am new to linux ..

ctrl + d ==> save.

ex: cat > ramakrishna

I am new to linux , devops , aws

ctrl + d ==>> save..

==>>>> list of files ==>> ls

file identification ==>> ls ==> file ==>>> white color.

ls -l ==>> first field ==>> - (hyphen)

2. How to append the data ina file

cat >> filename

S3 and RDS ..

ctrl + d ==>> save

3. How to view inside data in a file.

cat filename

cat ramakrishna

4. How to copy file from one location to another location.

cp source destionation

note : destination must be a directory..

mkdir directoryname

mkdir sachin

ex: cp ramakrishna sachin

4. How to move file from one location to another location.

mv source destionation

note : destination must be a directory..

mkdir directoryname

mkdir yuvi

ex: mv ramakrishna yuvi

5. How to rename a file.

mv oldname newname

mv ramakrishna srinivas

6. How to create empty files.

touch filename

touch abc

touch {a..m}

7. file1 ==>>>data and file2 empty file ==>> file1 data copy to file2. ==>>> redirect.

cat file1 > file2

cat ramakrishna > abc

8. How to delete a file .

rm -rf filename

rm -rf ramakrishna.

==========
=

==========
=

directory operations :

how to create a directory ??

mkdir directoryname

mkdir sachin

ls ==>> directory color ==>> blue.

ls -l ==>> first field (d)

pwd ==>> present working directory..

cd ==>> change directory..

cd sachin

pwd

/home/ec2-user/sachin

mkdir rahul

cd rahul

pwd

/home/ec2-user/sachin/rahul

mkdir hardik

cd hardik

pwd

/home/ec2-user/sachin/rahul/hardik

mkdir lara

cd lara

pwd

/home/ec2-user/sachin/rahul/hardik/lara

cd ..

/home/ec2-user/sachin/rahul/hardik

cd ..

/home/ec2-user/sachin/rahul

cd ..

/home/ec2-user/sachin/

cd ..

/home/ec2-user/

mkdir -p /home/ec2-user/sachin/rahul/hardik/lara/ponting

cd /home/ec2-user/sachin/rahul/hardik/lara

pwd

/home/ec2-user/sachin/rahul/hardik/lara

cd ../../../../

pwd

/home/ec2-user/

==========

how to rename a directory

mv oldname newname

mv sachin dhoni

how to delete a directory

rm -rf directryname

rm -rf sachin

==========
==========

filter commands :

files / directories / users / groups

useradd ramakrishna

useradd bhargavi

user related information ==>>> /etc/passwd

head : top 10 users to display ==>> head /etc/passwd

tail : below 10 users to display ==>> tail /etc/passwd

more : page by page ===>> more /etc/passwd ==>>> space button press ==>> last page ==>> automatically exit..

less : page by page ==>> less /etc/passwd. ==>>> space button press ==>> last page ==>> not exit ==>>> q button press ==> quit

==========

vi editor :

files ===>>> create , with in the files ==>> data ==>>> modify and delete by using vi editor..

vi editor has 3 types of modes..

1. CLI mode.

2. Insert mode.

3. Extended mode.

vi ramakrisha ====>> CLI mode

press " i " key ==>> insert mode.

I am new to Linux..

escape shift:wq! ===>>> save ===>>> extended mode.

cat ramakrishna

==========

grep and find :

10 files

ramakrishna

ls -l | grep ramkrishna

ls -l | grep 123

ls -l | grep abc

ls -l | grep a

ls -l | grep A

i ==>> ignore case sensitive

ls -l | grep -i A

find :

find / -optins keyword

options :

1. files

2. directories

3. users

4. groups

5. inum ==>>> inode number ==>> 4 digit number.

find / -name ramakrishna

find / -name sachin

find / -user pavan

find / -group aws

find / -inum 1234

==========

files and directory permissions : ===>> security

security ===>> userlevel , grouplevel , otherlevel..

ls -l

- ==>> file

d ==>> directory

c ==>> charecter file

b ==>> block file

l ==>> link file.

rw- (userlevel) r-- (grouplevel) r-- (otherslevel)

r ==>>> read ===>> 4

w ==>> write ===>> 2

x ==>> execute ==>> 1

By using this command ==>> chmod command ==>> change modification.

2 types methods to giving the files and directory permissions.

1. symbolic method.

2. Absolute method.

==========

1. symbolic method.

file ==>>> bhargavi

userlevel 6 , grouplevel 3 , otherslevel ==>> 5

chmod u=rw,g=wx,o=rx bhargavi

sachin ==>>> 7 (userlevel) 6 (group level) 4 (otherlevel)

chmod u=rwx,g=rw,o=r sachin

==========

2. Absolute method.

yuvi ==>> 655

chmod 655 yuvi

chiru ==>> 666

chmod 666 chiru

abc ==>> only userlevel full permissions..

chmod 700 abc

xyz ==>> group level full permissions..

chmod 070 xyz

chmod 007 ponting..

==========

file full permissions : 666

directory full permissions : 777

default file permissions : 644

default directory permissions : 755

umask ==>> 022 / 0022

666 - 022 ==>> 644

777 - 022 ==>> 755

==========

Booting process :

ex: windows ==>> power on button ==>> press ==>> password ====>> in between poweron button and passowrd process ==>> Booting process..

Linux ==> power on button ==>> press then booting process will starts.

Booting process has 6 stages :

1. BIOS : Basic input output system.

2. MBR : Master boot record.

3. GRUB : Grand unified bootloader.

4. KERNEL :

5.INIT : initialization.

6. RUNLEVELS :

1. BIOS : Basic input out system.

It will checks the system integrity check .

system integrity check ==>> system's hradware check ==>> motherboard , cpu , ram , harddisk ==>> properly working or not ??

2. MBR : Master boot record:

It contains the bootables files ..

MBR has 3 components

1. Primary bootloader. ==>> 446 bytes.

2. Partition table information. ==>> 64 bytes.

3. MBR validation check. ==>> 2 bytes.

MBR size ==>> 512 bytes.

3. GRUB : Grand unified bootloader.

GRUB contains the information

Root device inforamtion ===>> /dev/xvda

multiple kernel images ==>> 5 , 6 , 7 , 8 , 9

default time ===>> ???

timedout ===>> ???

grub contains one configuration file ===>> /boot/grub/grub.conf

vi /boot/grub/grub.conf

/boot/grub/grub.conf ==>>> this configuration file link to /etc/grub.conf.

4. KERNEL :

It is the mediator between o.s and hardware.

it is the heart of the operating.

It will manages devices information , multitasking , filesystem information.

5. INIT :

It is parent of all process.

each process has their own unique identification number.

process ==>> unique id ==>> process id ==>>> PID

init ==>> pid ==>> 1

root ==>> pid ==>> 0

init 0 ===>> Hung state. (danger command.)

init 1 ===>> single user mode (trouble shoot)

init 2 ===>> multiuser mode with out network (networking related commands are not working)

init 3 ===>> multiuser mode with network (networking related commands are working here) ==>> default init level

init 4 ===>> un used.

init 5 ==>>> X11 (GUI mode)

init 6 ===>>> reboot ==>> danger command ===>> with respective people ==>> approval.

vi /etc/inittab

/etc/init.d ==>> scripts..

6. RUNLEVELS :

shell scripts ==>>> application install or backup ==>> scripts to put inside inside runlevels.

/etc/rc.d/rc0.d ==>> runlevel 0

/etc/rc.d/rc1.d ==>> runlevel 1

/etc/rc.d/rc2.d ==>> runlevel 2

/etc/rc.d/rc3.d ==>> runlevel 3 ==>>> default runlevel..

/etc/rc.d/rc4.d ==>> runlevel 4

/etc/rc.d/rc5.d ==>> runlevel 5

/etc/rc.d/rc6.d ==>> runlevel 6

vi /etc/rc.d/rc3.d/.backup.sh ==>> reboot ==>> you will get complete backup of linux server.

/etc/init.d ==>> scripts.. ==>>> app ==>> service ==>> manage.

==========

AWS ==>> runlevels ==>> alternative ==>> userdata ==>> script.

==========

Partitiong / filesystem creation :

deviding the hard disk into the no .of partitions..

500gb harddsik ===>> 10 parttions ==>> each partition has the size ==>> 50 gb..

Physical servers point of view :

device naming convensions :

/dev ==>> devices information.

/dev/sda ==>> SCSI

/dev/hda ==>> IDE

/dev/vda ==>> virtual disk..

4 , 8 , 12 , 16.

Each physical linux servers ==>> 16 hard disks attached one linux servers..

/dev/sda to /dev/sdp

/dev/sda to /dev/sde ==>>>> o.s internally used..

extranal we will attached to the physical linux server ==>>> /dev/sdf to /dev/sdp..

LInux ==>> file system types ==>> ext2 , ext3 , ext4 (latest)

senario :

Application team ==>> request raise to linux admin team ==>> 500 gb ==>> disk space (hard disk) ==>>>file sysyem ==>> app5 ==>> mount point ==>> application install.

Linux admin team ==>> request raise SAN (storage area network) team ==>> please attach 500 gb hard disk to lx123 (linux server name).

SAN team request raise to data center people (field engineers) ==>>> lax123 ==>attach to 500gb hard disk. ==>>> they will attach 500gb hard disk to linux

server.

Linux admin team follows below steps..

1. fdisk -l (o.s control)

2. partprobe /dev/sdf ==>>> kernel identification.

3. mkfs.ext4 /dev/sdf ==>> creating the file system.

4. mkdir app5

5. mounting : attaching a directory to the file system. it is called mount point.

mount -t ext4 /dev/sdf app5

6. cat /etc/mtab ==>> temparary mount points.

7. How to make permanate mount ??

vi /etc/fstab

devicename mountpoint typeoffilesystem defaults 0 (dump) 0 (check sequence)

/dev/sdf /home/ec2-user/app5 ext4 defaults 0 0

esc shift:wq!

8. cd app5

ls

lost + found ==>> directory..

touch {a..e}

reboot

==========

AWS cloud : EBS ==>> elastic block storage.

disk space ===>> volume

Application team ==>> request raise to linux admin team ==>> 500 gb ==>> volume ==>>> filesystem ==>>> app5 ==>> mount point ==>> application install.

EBS thumbrule :

Ec2 instance and volume should be in same availability zone.

Ec2 instance ==>> 1a ==>> AZ

volume ==>> same AZ (1a) ==> 500 gb

we will attach this volume to ec2 instance

volumes ==>> 16 volumes to create one ec2 instance.

/dev/sda to /dev/sdp.

/dev/sda to /dev/sde ==>> o.s internelly used.

volume attach to ec2 instance ==>>> /dev/sdf to /dev/sdp. (11)

After login into the ec2 instance ==>> device naming convension to display diffrent. ==>>> /dev/xvdf to /dev/xvdp.

Linux admin team follows below steps..

1. fdisk -l (o.s control)

2. lsblk ==>>> kernel identification.

3. mkfs.ext4 /dev/xvdf ==>> creating the file system.

4. mkdir app5

5. mounting : attaching a directory to the file system. it is called mount point.

mount -t ext4 /dev/xvdf app5

6. cat /etc/mtab ==>> temparary mount points.

7. How to make permanate mount ??

vi /etc/fstab

devicename mountpoint typeoffilesystem defaults 0 (dump) 0 (check sequence)

/dev/xvdf /home/ec2-user/app5 ext4 defaults 0 0

esc shift:wq!

8. cd app5

ls

lost + found ==>> directory..

touch {a..e}

reboot

==========

ebs history :

```
[root@ip-172-31-32-34 ec2-user]# history
    1  fdisk -l
    2  lsblk
    3  mkfs.ext4 /dev/xvdf
    4  mkdir app5
    5  mount -t ext4 /dev/xvdf app5
    6  cat /etc/mtab
    7  vi /etc/fstab
    8  df -h
    9  cd app5/
   10  ls
```

```
   11  touch {a..z}
   12  ls
   13  cd ..
   14  fdisk -l
   15  lsblk
   16  mkfs.ext4 /dev/xvdg
   17  mkdir app6
   18  mount -t ext4 /dev/xvdg app6
   19  cat /etc/mtab
   20  vi /etc/fstab
   21  ls
   22  cd app6
   23  ls
   24  touch {1..20}
   25  ls
   26  cd ..
   27  history
[root@ip-172-31-32-34 ec2-user]#

[root@ip-172-31-32-34 ec2-user]# cat /etc/fstab
#
UUID=26620198-186a-404b-b9a1-12d957d7c826     /           xfs    defaults,noatime
1   1
/dev/xvdf /home/ec2-user/app5 ext4 defaults 0 0
/dev/xvdg /home/ec2-user/app6 ext4 defaults 0 0
[root@ip-172-31-32-34 ec2-user]#
[root@ip-172-31-32-34 ec2-user]#
```

==========

Networking :

Two or more systems connected each other ==>> networking

systems ==>>>nothing but servers.

Physical servers point of view ==>> data center ===>> onpremise infrastructure.

Two servers are in same network ==>>> minimum requirements..

1. Two servers must be cabled with other.

2. Each servers has at least one NIC card (Network interface card / controller..)

3. Each NIC card has one IPaddress and subnetmask..

4. After login into physical servers==>> eth0 ===>> logic nic name ==>> 192.168.0.1 (IP address) and subnetmask ==>> 255.255.255.0.

NIC1 ==>> eth0

NIC2 ==>> eth1

NIC3 ==>> eth2

Based on the hardware ==>> nic slots..==> NICs will attach.

5. Then these two systems in same network.. and these systems communicate with each other..

6. server1 ==>>> login ==>> ping server2IPaddress ==>> ping sequence..

7. server2 ==>>> login ==>> ping server1IPaddress ==>> ping sequence..

==========

Networking advantages..

1. files transfer ==>>> from one server to another server.

2. Remoteuserly login ==>> from one server to another server. ==>>> applications install..

==========

The above requirements to do tasks..==>> we configure the ssh configuration.

ssh : secure shell ==>>>> port number ==>>> 22

ssh : secure sheel

1. server1 to server2 ==>>> files trasfer ==>> encrypted format.

server2 to server1 ==>>> files trasfer ==>> decrypted format.

SSH ==>> no one will hack.

2. ssh : password less authentication.

server1 to server2 ==>>> connect ==>> with out passowrd asking.

server2 to server1 ==>>> connect ==>> with out passowrd asking.

How to configure ssh configuration ??

central.pem ==>> privatekey.

server1 : central.pem ==>> privatekey ==>> copy.

1. vi /tmp/central.pem

 paste the privatekey ==>> save

2. chmod 700 /tmp/central.pem

server2 : central.pem ==>> privatekey ==>> copy.

1. vi /tmp/central.pem

 paste the privatekey ==>> save

2. chmod 700 /tmp/central.pem

==========

1. How to transfer files from one server to another server.

server1 to server2 ==>> files transfer

scp : secure copy

touch bhargavi

scp -i /tmp/central.pem filename ec2-user@server2IPaddress(public / private Ip):/home/ec2-user

scp -i /tmp/central.pem bhargavi ec2-user@50.20.10.5:/home/ec2-user

server2 to server1 ==>> files transfer

scp : secure copy

touch ramakrishna

scp -i /tmp/central.pem filename ec2-user@server1IPaddress(public / private Ip):/home/ec2-user

scp -i /tmp/central.pem ramakrishna ec2-user@60.20.10.5:/home/ec2-user

==========

2. How to login remote userly from one server to another server.

server1 to server2 ==>>> remoteuserly login.

ssh : secure shell

ssh -i /tmp/central.pem ec2-user@server2IPaddress(public / private Ip)

ssh -i /tmp/central.pem ec2-user@50.20.10.5 ==>> enter ==>> now you are in server2.

server2 to server1 ==>>> remoteuserly login.

ssh : secure shell

ssh -i /tmp/central.pem ec2-user@server1IPaddress(public / private Ip)

ssh -i /tmp/central.pem ec2-user@60.20.10.5 ==>> enter ==>> now you are in server1.

==========

ifconfig -a ==>> command

nic card logical name , up , running ,mtu (memory tranfer unit)==>>9001

nIC ==>>> mac address , IPaddress and subnetmask..

lo : loop back address ==>> self ping ===>> 127(series)

Ipaddress ==>> privateIP.

==>> How to change / assign the IPaddress of linux server ??

cd /etc/sysconfig/network-scripts

ls

ifcfg-eth0 ifcfg-eth1

vi ifcfg-eth0

IPADDR=192.168.20.5

save

service network start

==>> How to change / assign the hostname of the linux server ??

vi /etc/sysconfig/network

hostame = xyz.com

save

service network start

==========
hostname

xyz.com

==========

```
[root@ip-172-31-46-139 network-scripts]# history
    1  ping 54.250.156.121
    2  ifconfig -a
    3  ping 172.31.46.139
    4  vi /tmp/kalpana123.pem
    5  chmod 700 /tmp/kalpana123.pem
    6  touch jyothsna
    7  scp -i /tmp/kalpana123.pem jyothsna ec2-user@54.250.156.121:/home/ec2-user
    8  ls
    9  ifconfig -a
   10  ssh -i /tmp/kalpana123.pem ec2-user@54.250.156.121
   11  ifconfig -a
   12  git --version
   13  cd /etc/sysconfig/network-scripts/
   14  ls
   15  vi ifcfg-eth0
   16  hostname
   17  cat /etc/sysconfig/network
   18  vi /etc/sysconfig/network
   19  hostname
   20  history
[root@ip-172-31-46-139 network-scripts]#
```

==========

==========

Each Linux servers has one Ipaddress along with one subnetmask..

An Ip address is an 4 digit octal number

octal number ==>> 8.

example Ipaddress ==>> 192.168.5.10

Each ==>> digit / bit.

4 * 8 = 32 bits..

each bit or digit ==>> 2 power some thing.

Each bit ==>> binaray format ==>> 010110

We will decide Ipaddress ==>>> which class it will be avaible based on the first bit.

Ipaddress class types:

CLASS A : 0 to 127 ===>> 255.0.0.0 ==>> subnetmask ====>>> CIDR block ==>> /8

CLASS B : 128 to 191 ===>> 255.255.0.0 ==>> sunbetmask ==>> CIDR block ==>> /16 ===>> VPC

CLASS C : 192 to 223 ===>> 255.255.255.0 ==>> subnetmask ==>> CIDR block ==>> /24 ==>> subnet.

CLASS D : R&D

CLASS E : unused.

127 + 64 ===>> 191

191 + 32 ===>> 223

CIDR block / notation ==>> we will decide the cidr notation based on the subnetmask..

CIDR : classless interdomain route.

An Ipaddress can be devided into two portions.

1. Network portion (static / constant)

2. host portion (dynamic and change)

1. Network portion (static / constant) ===>> first 2 bits or 3 bits.

2. host portion (dynamic and change) ==>> last 2 bits or 1 bit.

our own network ==>>> how many Ipaddresses will relases and In this network ==>>

how many ec2 instances will create.??

ex: 30.50.10.40 ==>> SBI network

1. Network portion (static / constant) ===>> first 2 bits ==>> 2 power 16

2. host portion (dynamic and change) ==>> last 2 bits. ==>> 2 power 16 ===>> 500.

30.50.10.40 ==>> SBI network ==>>> in this network 500 Ipaddresse release ==>> 500 ec2 instances will create in this SBI network..

30.50.11.40

30.50.12.40

30.50.13.40

30.50.14.40

.
.
.

30.50.300.40

30.50.300.41

30.50.300.42

30.50.300.43
.
.
.

==========

ex: 90.50.40.25 ==>> HDFC network

1. Network portion (static / constant) ===>> first 3 bits ==>> 2 power 24

2. host portion (dynamic and change) ==>> last 1 bits. ==>> 2 power 8 ===>> 256

90.50.40.25 ==>> HDFC network ==>>> in this network 256 Ipaddresse release ==>> 256 ec2 instances will create in this HDFC network..

90.50.40.26

90.50.40.27

90.50.40.28

90.50.40.29

90.50.40.30

.
.
.

90.50.40.281

==========

Package Administration / software management / package management.

windows ==>>> softwares like ==>> vlc media player , pdf , msooffice..

Linux ==>>> packages..

Package Adminsitration ==>> LINUX ==>> two types utilities..

1. RPM : Redhat package manager

2. YUM : Yellow dog update modifier.

LINUX : RPM and YUM ==>> packages ==>> install , uninstall , verify , information , update , upgrade.

update ==>>> linux version 5.2 ===>>> linux version 5.5 ===>> patching.

upgrade ==>>> Linux version 5 ===>> linux version 6 ==>> upgrade.

Physical servers point of view :

1. RPM : Redhat package manager

step 1 :Physical linux server ===>> cd / dvd disk ===>>> group of packages copied into cd / dvd disk.

Physical linux server ===>> cd / dvd disk ===>>> insert ==>>> all packages ==>> copy to any location of the physical server.

location ==>> /var/ftp/pub/packages.

step 2 : go to the exact path of the available packages.

cd /var/ftp/pub/packages ===>> mandatory.

rpm -ivh packagename

i ==>> install , v ==>> verbose , h ==>> hash prompt.

rpm -ivh httpd

rpm -uvh packagename

rpm -uvh httpd

rpm -qa packagename

rpm -qa httpd

rpm info packagename

rpm info httpd

rpm update

rpm upgrade

Key point : It will check the dependencys..

httpd install ==>> dependent ==>> java ==>> first you need to install java and after that you need to install httpd.

RPM : drawbacks ==>> 1. path 2. dependency checking.

To overcome the above drawbacks in RPM then YUM came into the picture.

1. YUM : Yellow dog update modifier

step 1 :Physical linux server ===>> cd / dvd disk ===>>> group of packages copied into cd / dvd disk.

Physical linux server ===>> cd / dvd disk ===>>> insert ==>>> all packages ==>> copy to any location of the physical server.

location ==>> /var/ftp/pub/packages.

Repositories ===>> group of packages managed place .

we will create our own repositories.

/etc/repos.d ===>> we will create repositories here.

repository extension must be name.repo

vi /etc/repos.d/bhargavi.repo

[bhargavi]

base url : http:///var/ftp/pub/packages

gpgcheck = 0

enabled =1

esc shift:wq!

==>> yum install packagename

yum install httpd ==>> y/d/n ===>> type y.

yum install -y httpd

yum remove packagename

yum remove httpd

yum list

yum info packagename

yum info httpd

yum update -y

yum upgrade -y

==========

AWS ==>> cloud.

1. cd / dvd disk ==>> no need insert. ==>> these instances are virtual instances.

2. No need to create repositories.

yum install -y httpd ==>> online ==>> httpd site.

yum install -y git ==>> gitsite

yum install -y maven ==>> maven site

yum install -y docker ==>> docker site.

yum install -y tomcat ===>> tomcat site.

==========
===

Managing installed packages..

service packagename status

service packagename start

service packagename stop

service packagename restart

service packagename reload
==========

service httpd status

service httpd start

service httpd stop

service httpd restart

service httpd reload.

restart ===>> service ==>> stop and start

reload ===>> service ===>> httpd ==>>> install ==>>> internet issue ==>> 80 %
install==>> remaining 20 % install ==>> stop and start.

==========

The above only for one session.

chkconfig httpd on ==>>> application will always close to enduser.

chkconfig httpd off

```
================================================================================

[root@ip-172-31-4-161 ec2-user]# history
    1  yum install httpd
    2  service httpd status
    3  service httpd start
    4  service httpd status
    5  cd /var/www/html/
    6  ls
    7  vi index.html
    8  cd /home/ec2-user/
    9  yum install -y docker
   10  yum install -y git
   11  yum list | grep jdk
   12  yum install -y java-1.8.0-openjdk-devel.x86_64
   13  yum install -y ansible
   14  sudo amazon-linux-extras install ansible2 -y
   15  history
   16  yum install -y httpd
   17  history
[root@ip-172-31-4-161 ec2-user]#
[root@ip-172-31-4-161 ec2-user]#

================================================================================
======
================================================================================
======

Job automation / job scheduling ..

Job ==>> task ==> perticular interval of time  schedule ==>> job scheduling or job
automation.

job scheduling ==>> two types of methos or jobs..

1. at job.

2. cron job.

1. at job : It is used to only once at a specified time.

at  task of time.

step of task

ctrl + d ==>> save.

at now

mkdir sachin

ctrl + d ==>>> save.


at 10:30 am

ifconfig -a
```

ctrl + d ==>> save.

==>> when ever you created a job then automatically linux operating system gives a one unique id ==>>> job id.

list of jobs ==>> atq

at rm jobid ==>> delete the atjob

at rm 1234

===============

/etc/at.deny ==>>> bhargavi , pavan

/etc/at.allow ===>. ramakrishna , pavan

at , cron jobs are follows round robin algorithem.==>>> first in first out.

==

cron jobs : It is used to repetative taks.. ====>> poll scm , build peridically ==>> jenkins.

crontab -e ==>>here we will create cron jobs and cronjobs has the fields..

min hours dayofmonth month dayofweek command / task / script.

*   *   2   3   0   ./backup.sh

* ==>> all

*/2 ==>> every 2minits

*/5 ==>>> every 5 hours

*/4 ==>> evry 4 days

*/3 ==>> evry 3 months

*/0 ==>> evry sunday..

*/2-4

*/ 2,4,6

crontab -l

crontab -r

crontab -u

/etc/cron.deny ==>> vamsi , shekar

/etc/cron.allow ==>> rajendra , shekar

==
=

Troubleshooting commands / performance tuning / health checkup commands.

1. ps ==>> how many processes currently running your system.

2. ps -elf ==>> it displys all processes..

3. bg ==>>> background running processes to display.

4. fg ==>>> foreground running processes to display.

5. ps -ef | grep smon ==>> currently running application in linux server.

6. ps -ef | grep pmon ==>> currently running database in linux server.

7. top ==>> process running , stop , uptime , load average , cpu , memory , swap ...etc..==>> exit ==>> press q button.

8. iostat ==>>> disk related information.

9. vmstat ==>> virtual memory statistics information.; free -m

10. uptime ==>> load average ===>> 3 fields ==>> 1m 5m 15m

11. netstat ==>> networking statistics information ; netstat -nr ==>>> routing table information.

12. sar ==>> system activity report.

==========

```
[root@ip-172-31-14-39 ec2-user]#
[root@ip-172-31-14-39 ec2-user]# history
    1  ps
    2  ps -elf
    3  bg
    4  fg
    5  ps -ef | grep smon
    6  ps -ef | grep pmon
    7  top
    8  top
    9  iostat
   10  vmstat
   11  free -m
   12  netstat
   13  netstat -nr
   14  sar
   15  uptime
   16  history
[root@ip-172-31-14-39 ec2-user]#
[root@ip-172-31-14-39 ec2-user]#
```

==========