scsi= small computer system interface

sdd,hdd,scsi(vmware),EBS (AWS)

HDD= Block storage device, 1 Block= 1 sector = 512 bytes

2 sectors=1kb

partitions:

primary

LVM

RAID: (redundant array of independent disks)

filesystems: TO STORE FILES in form of directory structure

windows:-FAT32,NTFS(New technology file system)

LINUX:-EXT2,EXT3,EXT4,XFS

What is disk, partitioning,file system,block,track?

Spindle motor,stepper motor(arm),r block

Primary:-  
bootable

/boot

We can create Upto 4 partitions per disk

1TB disk

P P P E

65535 LVM partitions

\*we cannot extend partition without downtime

To show the added disk, reboot server but we should not reboot production servers instead we can use we should use channel target LUN(logical Unit Number)

To scan the disk at voice level , command used is

Echo”- - -” >>/sys/class/scsi\_host/host0/scan

Host0= HBA card1

Host1= HBA card2 and so on (host bus adapter card)

Lsblk

Fdisk -l

Lsscsi

Df -hT

To create partition: fdisk /dev/sda,b,..

At last type **w** to save changes

To update partitioning in the kernel: partprobe /dev/sdb

To use directory, first need is to format the disk, **mkfs** a command used to format a block storage device with a specific file system

Mkfs.xfs /dev/sbd2

Now , mount

Mkdir /oradata

Mount /dev/sdb /oradata =temporary mounting

Check with df -hT to see the disk and file system total and available space.

So manually update in /etc/fstab

/dev/sdb1 /oradata xfs defaults 0(metadata backup) 0(filesystem bootup) save it

Mount -a = to validate whether file system added manually is correct or not

Mount to check partition done or not

Steps of LVM for adding new space

Install a new Hard Disk drive

Make a partition to use it

Designate physical volume (PV)

Manage Volume Group (VG)

Manage Logical Volume (LV)

Apply a filesystem Set a mount point

Steps for LVM

1. Install a new hard disk drive and scan

Echo”- - -” >>/sys/class/scsi\_host/host0/scan for host0,1,2,3

Pvcreate /dev/sdb

Vgcreate vg\_1 /dev/sdb

Lvcreate -L +30G -n oradata\_LV vg\_1

Mkfs.xfs /dev/vg\_1/oradata\_LV

Mkdir /oradata

Vim /etc/fstab

/dev/vg\_1/oradata\_LV /oradata xfs default 0 0

Mount -a

**Mount**

**Extend VG**

**(when space is available in VG)**

Lvextend -l +50%free /dev/vg\_1/oradata

Or

Lvextend -l +50G /dev/vg\_1/oradata **-r (-r means also extending file system)**

Command only to extend the file system

Xfx\_growfs /dev/vg\_1/oradata

**(when space is not available in VG)**

Add disk

Then

Pvcreate /dev/sdb

Vgextend vg\_1 /dev/sdb

Lvextend -l +50G /dev/vg\_1/oradata **-r**

Verify with df -hT ,lsblk or mount

To create LV partition in ext4 file system process is same but define file system explicitly

Command only to extend the file system

Resize2fs /dev/vg\_1/oradata

***File permissions***

Ls -l

8 columns

Default permission for users to access file: maximum 666 but by default it is 644 = rw- r-- r-- (default -umask=0022 so 666-22=644)

Now using umask to change permission:

Umask 0044

Then create a file

Then check with ll

We get 622=rw- -w- -w-

Now recommended to change umask to default

Another method=chmod

1. Chmod 666 file.txt
2. Chmod u+x file.txt g+rwx,o-x

2. What is LVM and it's example

Advantage of LVM

Possibilities of LVM

Real-time LVM Example

Adding New Space/disk using LVM Extending the space using LVM

Creating a partition using fdisk command

Commdands

fdisk -l

pvcreate

Pvdisplay

vgcreate

vgextend

Vgdisplay

Lvcreate

Lvextedn

Lvdisplay

Lvs

Mount

Mkfs

df -Th

Echo “- - -” >>/sys/class/scsi\_host/host0/scan

Echo “- - -” >>/sys/class/scsi\_host/host1/scan

Fdisk /dev/sda

N

1

+7g

W

Partprobe /dev/sda

Mkfs.xfs /dev/sda1

Mkdir /newdisk

Mount /dev/sda1 /newdisk

Vi /etc/fstab

Add entry

Mount -a

Mount

#this is primary partition

#logical volume

Add disk

Echo”- - -” >>/sys/class/scsi\_host/host0/scan

Pvcreate /dev/sda

Vgcreate vg1 /dev/sda

Lvcreate -l +50G -n lv1 vg1

Mkfs.xfs /dev/vg1/lv1

Mkdir /newmount

update lv in etc/fstab

/dev/vg1/lv1 /newmount xfs default 0 0

Mount -a (to refresh)

#to extend lvm (1- if space is available)

Lvextend -L +2G /dev/vg1/lv1 -r

Xfs\_growfs /dev/vg1/lv1

Refresh = mount -a

1. If space is not available

Add disk

Echo”- - -” >>/sys/class/scsi\_host/host0/scan

Pvcreate /dev/sdc

Vgextend oldvg /dev/sdc

Lvextend -l +2G /dev/oldvg/oldlv -r

Resize2fs /dev/oldvg/oldlv (for extended file system)

Mount -a

Xfsdump -f /tmp/bakup.dump /filesystem

Umount /filesystem

Lvremove /dev/vg1/lv1

Lvcreate -l +20G -n lv2 vg1

Mkfs.xfs /dev/vg/lv1

Mount -a

Xfsrestore -f /tmp/backup.dump /filesystem

extended

Umount /filesystem

#Arrange the sectors

E2fsck -f /dev/vg/lv

Resixe2fs /dev/vg/lv 2G

Lvreduce -l -1000M /dev/vg/lv

Mount -a

Umount /filesystem

Delete entry from etc/fstab

Lvremove lvpath

Vgremove bgname

Pvremove pv path

#disable disk

Echo offline > /sys/block/sdd/device/state

Echo 1 > /sys/block/sdd/device/delete

Delete from vm

/etc/logrotate.conf

Etc/logrotate.d

Cd /etc/logrotate.d/ touch myapp

/var/log/myapp/\*.var{

Daily

Size 10M

Olddir /var/log/myapp/archive

Cd /etc/var/log/ mkdir myapp touch myapp.log

**To count the lines in a file:**

$ awk 'END { print NR }' geeksforgeeks.txt

3

ghp\_FMvzjDKYQlZqANBa59kznkhmEobgct2ouF8J

GIT

Git add, git commit, git push

Git add . or git add filename

Git diff = to see the what changes are made

Git status

Git commit -m “comment or subject or tag ”

Git log = keeps tack of several commits like 1st ,second commit etc. (commit ID, author, Date, tag of commit)

Git log --oneline=to view changes in one line(only shows ID, branch and tag)

Git reset --hard commitID=To switch to previous versions of the code

Git checkout , git stash to undo the changes

Git clone (https- authenticate using password ,ssh- authenticate using key) ssh-keygen -t rsa= key is generated and add this key to github

Git remote add

Fork is only to copy the repository

Clone is to download the repository

Git checkout -b branch name=To create branch

Git branch branchname

Git branch= to view all branches

#commit command will apply the changes to the branch you are in

Git checkout branchname= To switch the branch

#to merge the branches

Git cherry-pick commitID= to merge to main branch

Git merge, git rebase to merge multiple times

Rebase organizes the changes in linear way like history of the merge(linear coomit history)

Merge will have the changes at the top

For git push --first clone and the git remote add

Use== git remote add origin repository link

Then

Git push -u origin name of the branch ,like git push -u origin master

Docker

Entrypoint- valuye or parameters cant change,

Cmd- can be changed according to the need

Multistage buid

From ubuntu as build

From openjdk:11

Copy from build

Entrypoint

\*Countless stages

Distroless image: minimalistic docker image which only have runtime environment, which provides security to the application so image is refrained from os vulnerabilities

(golang has do not runtime dependecies)

\*scratch (for golang application) is the minimal distroless image

Docker networking:

To connect container with host, default network created during docker build called bridge networking which is veth0 or docker0

Bridge networking

Host networking is not used which can compromise with the security because container and host are within the same subnet so vertual default docker networking is required

But to create network isolation(isolate containers in terms of networking), custom bridge nteworking using docker network command to communicate with the host

Docker exec -it login /bin/bash

Docker network create network\_name

Docker inspect container\_name= ipadress of network

Docker ps

Ping -V

Docker run -d --name finance --network=secure\_network nginx:latest

Bind mounts and volumes:

To get or backup the full details, data of the container when container is down:

1. Bind directories from host to container
2. Volumes; can manage lifescycle of volumes

Accessed using docker CLI

Can be attatched to any resource like ec2, bucket

Can be mounted on external devices

Docker -v source, destination, additional details

Docker --mount

Both commands are same but mount is more verbose(elaborative)

Docker volume create v\_name

Docker volume ls

Docker volume inspect v\_name

Docker run -d mount source v\_name, target=/app nginx:latest

For cloud Nat

Vpc- public subnet-VM (ephimeral)

-private subnet-VM (non ephimeral)

Login to private vm and ping google.com but request time out

So

1st create external static ip to attatch to nat gateway

Now hybrid connectivity- cloud router,select vpc and region

Now create cloud nat from network services ,select vpc, nat router,select manual ip.click create

For load balancer:

Access to the servers or VMs by using single ip is done using load balancer

It balances the traffic request to both the servers using a single ip

\*allow firewall rules to allow port tcp port 80 to access the internet on VMs

From network sevices, select http(s) type load balancer , add frontend configuration, http, tcp:80,ephimeral.add

Backend conf.- backend type-instance group

Add new backend, select instance group,port 80,request timeout 30s,max backend utilization

Select health check section

Create

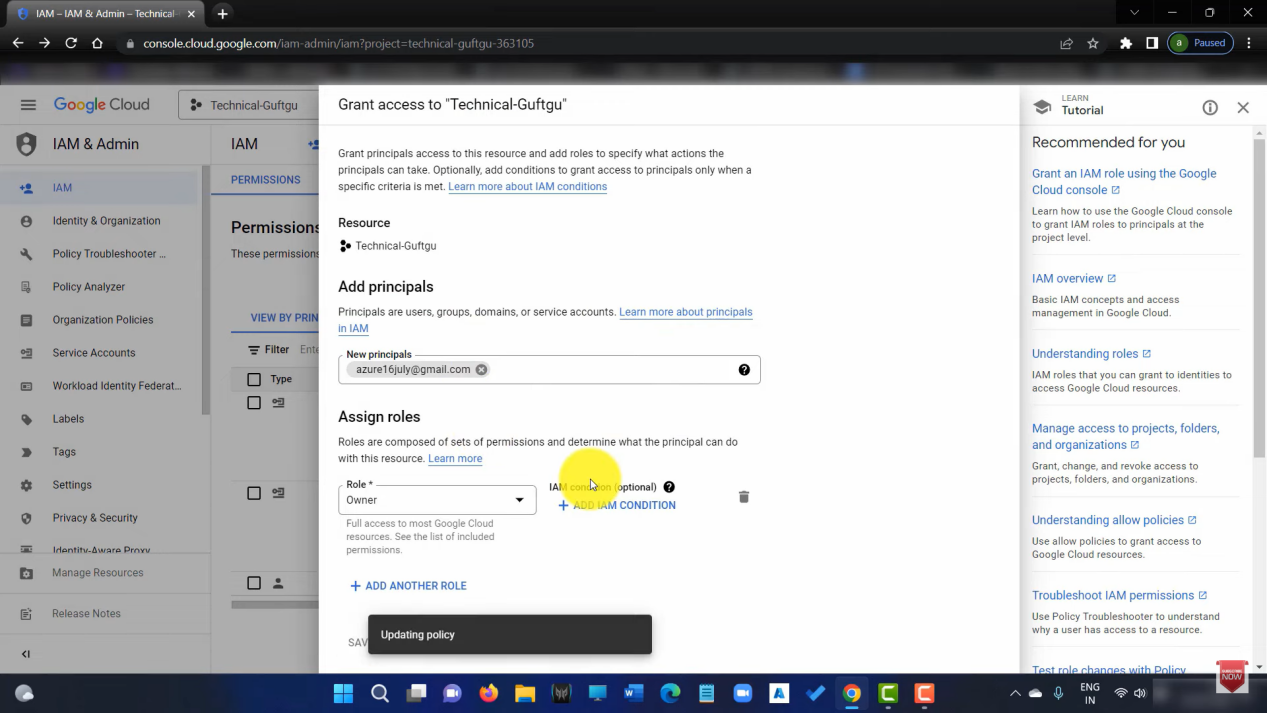
Check with load balancer’s ip to access the both the instances

Add auto scaling option , select instance group(specify the instance template), health check by specifying health criteria

IAM

Click grant access to create a new user

Basic roles-browser,editor,owner,view



#service account

Give names and details,tag for service account

Select role

Select users those who can access resources or services

While creating vm, specify the service account in Identity and API access, select service account to access VM for a particular service account

#custom roles for a user in a specified service account toaccess particular resources

#Identity aware proxy

To access the resources over private IP

Create user, assign roles, select IAP, select VM

