DATE : 19 Sept
PAGE NO. :

SESSION-6 AWS

- Do not have to buy servers (hardware) They get their resources from AWS.
- AWS - Service provider (Cloud Provider).
- 51% market share in 2019.
- Azure (Microsoft) leading cloud provider after AWS. followed by ~~Azu~~ Alibaba, (GCP) ; google cloud.
- compute as a service (EC2)
  - RAM / CPU
- Storage as a service (EBS)
  - H.DD
  - Elastic block storage
- Security / ML / DB / Firewall / IOT etc service.
- Data center (DC) = Availability zone (A.Z)
- Delay = latency = waiting time
- Usually launch atleast 2-3 AZ in with Min 150km dist for D.R (disaster recovery).
- aws.amazon.com/free → ~~free~~ complimentary services

DATE 20 Sept
PAGE NO. :

SESSION - 7 AWS

- Compute service == EC2.
- region → Mumbai / N. Virginia
- AMI (Amazon Machine Image) == Bootable Img.

↳ Amazon Linux 2 AMI [64-bit (x86)]

↳ Instance Type → t2.micro
↳ Provides only 1 CPU. We can only launch only 1 Os.

Laptop

↳ Instance Details → 1
↳ Subnet → aps1a
↳ ~~Stop - Hibernate behaviour~~

↳ Add Storage → 8

↳ Add Tags → Os1 (Value)

↳ Configure Security Gp
↳ Review & Launch

↳ Review
↳ Launch.

↳ Create a new pair
↳ AWSclass2020key

DATE : / /
PAGE NO. :

- AWS provides linux with cli.
  & win with GUI.

- AWS do not provide MAC OS | win10 services
  ↳ generally made for industrial usage & industry generally uses servers.

- Select OS1 → ACTIONS → CONNECT
  connect ← EC2 Instance Connect ↵

lscpu
  ↳ shows no of cpu in OS.

free -m
  ↳ RAM

↳ Dashboard → limits (L corner).
  pg4.
  (request limit instance)
  ① NU St (T.) →

  New limit value → 4

↳ Action → Instance State → stop.

- Persistent storage == H.D

- Hypernate enable ; saves all the data of RAM in HD.

DATE : / /
PAGE NO. :

uptime
↳ how much time the OS is running or last reboot time.

- remote login

ssh ip

Putty does not support .pem it supports .ppk.

- puttygen ↓ changes .pem to .ppk
  Save private key

sudo su - root

DATE : 26/ Sept
PAGE NO. :

SESSION - 11 AWS

- We can connect to win using RDP.
- Win does not support private key. We need to decrypt the key first.
- TO connect with RDP we require 3 things :
  (i) User : Administrator
  (ii) Ip Address : Public
  (iii) Password : Decrypt .pem file
- Launch == Provision
- ⇒ what is my ip
  ↳ will give your public ip
- In S.G we have to give our public ip not private ip.

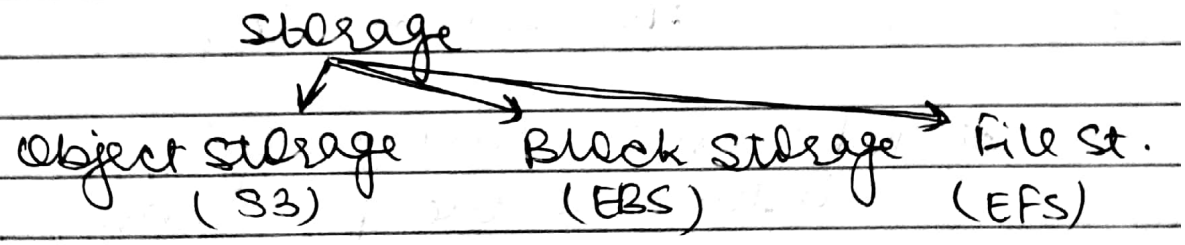
DATE : 3/Oct.
PAGE NO. :

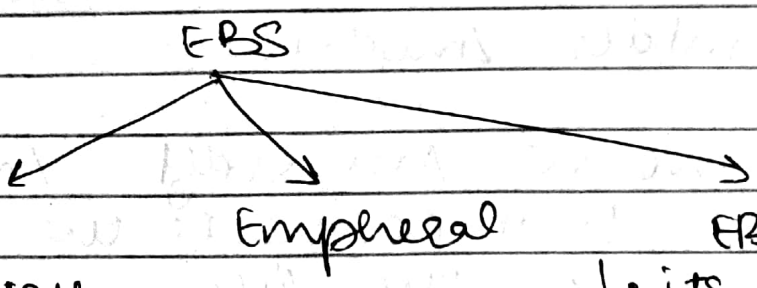
SESSION - 16 AWS

- RAM is volatile memory. Storage is persistant.
- S3 - Simple Storage Service (AWS).
  - Object storage (used to store data).
- Without a block device it is impossible to install an OS. ↳ HardDisk
- EBS → Elastic Block Storage/Store
  - Block storage (we can also install an OS)

But - this is not available.
Because we also require RAM, CPU & AMI.

- ∴ Instead of EBS, AWS provides EC2 service.
- EBS is sub-service of EC2.





↳ we install the OS in root block storage.

↳ its like an external HDD, pd we can share it with other instance

DATE : / /
PAGE NO. :

fdisk -l
↳ displays all the hard disk attached

df -h

- We can create new EBC by going to Volume tab in UFC.

HD
↳ partition
↳ Format
↳ mount
↳ Data.

We only have to do this once.

031 # fdisk /dev/xvdf
:n
Enter x4
→ Create the partition.
:w → save it.

mkfs.ext4 /dev/xvdf1
↳ formats the partition

mkdir /mydrive

mount /dev/xvdf1 /mydrive
↳ it connects the partition to the file

DATE : / /
PAGE NO. :

- xvda → root harddisk
  xvdf → EBS

~~umount~~ unmount /mydisine

- Then detach the volume from AWS.

OS2

mkdir /myd
mount /dev/xvdf1 /myd
ls
  ↳ It will show all the files in EBS.

TASK: Use EBS in dif. A.Z, will it work?
- NO It won't. It is a zonal service.

- we can't create partition in object storage (S3). Therefore, we can't install an OS.

- S3 (simple ~~st~~ cloud storage.
  ↳ fully managed
    ↳ already formated.
    ↳ no need of partition.

DATE : 4/ Oct.
PAGE NO. :

SESSION - 17 AWS

- Fully managed services are also known as serverless (S3)
  - ex: g.drive
- In S3 instead of folder we use bucket
  file ⟶ object

S3

- By default no one can download file
- To make the file downloadable change the settings to make public
- we can change settings for individual objects/bucket.
- No two bucket can have same name in one AZ.
  Bec. it creates an URL and they can't be same.

URL : protocol → BucketName → service N. → AZ → Domain → Object

- IAM : Identity Access Management
  - pre configured by AWS.
- S3 – Global service
  - we do not have to select a region.
- Root User
  - Unrestricted user
  - Acc. through which credit card is attached.

DATE : / /
PAGE NO. :

- we can add users and set IAM.
- Policy => Access

DATE: 11 / OCT
PAGE NO.:

SESSION-21 AWS

CLI FOR AWS

• Download the soft from: aws cli download install
(soft available in drive)
(.msi file)

aws --version

• Access key → user name
Secret key → password

• IAM: Identity Access Management.

I create an user. AWS M.C.A
II Access Type: Programmatic Access
III Power User Access (Policy)
↳ ✗ IAM services ✗
↳ ✗ Billing Dashboard ✗

login # aws configure
Region: ap-south-1
o/p format: ↵

aws help

aws ec2 help

aws ec2 describe-instances
↳ displays all the instances

DATE : / /
PAGE NO. :

aws ec2 start-instances --instance-ids <value>

aws ec2 run-instances [redacted]

--image-id <value>
--instance-type <value>
--key-name <value>
--security-group-ids <value>
--count <value>
--subnet-id <value>

#All in one single line

TASK
② . SG
③ . EC2 - Launch Instance
① . Key .
④ . EBS --create-volume

Attach

DATE: 17 Oct
PAGE NO.:

SESSION: 25 AWS

yum install httpd.
systemctl start httpd
↳ enable (persistent)

cd /var/www/html

URL: <public_ip>/lw.html

- Change the SG_inbound
  ↳ edit / add
  HTTP P:80 0.0.0.0/0

- We will mount the storage from EBS to /var/www/html to make this data persistent.

& connect it

fdisk /dev/xvdf
(+1G) /dev
mkfs.ext4 /xvdf1
mount /dev/xvdf1 /var/www/html

cp virat.jpg /var/www/html

Availability depends on EC2
Durability depends on EBS

DATE : / /
PAGE NO. :

Launch S3.
Create Bucket.
↳ Change the permission to public

Upload the image

<img src = "URL of img" >

- All the content is stored in S3.

- To make the delay low AWS copies the data to the nearest AZ when the client connects to the webpage for the first time. Then, the second time client visits the webpage they can directly access the data from nearest AZ.

This service is called CDN
(content delivery Network)
(Global service)

- CDN as a service == cloud front

- Akamai is the pioneer in CDN as a service

- Edge location are AZ for cloud front (CDN). ↓
POP (Points of Presence)

DATE : / /
PAGE NO. :

- They have their own metro optic fibers below ocean as their private network.
  - These cables help to low the latency/delay.

DATE: 24 Oct
PAGE NO.:

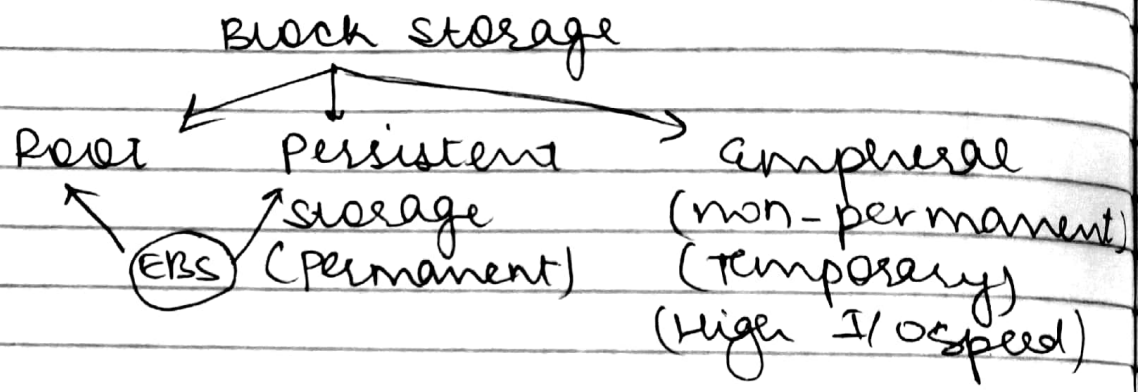
SESSION: 29 AWS

- Global network provided by AWS:
  (i) Low latency
  (ii) Security / Privacy
- Web page in EBS.
  Static content (images, videos) in S3.
- Main use of CloudFront is to create cache in different Edge location.
  ∴ CDN network.
- CloudFront
  Object Domain Name → S3 bucket
- TTL : After how much time to clear cache. (Time To Live)
- Blacklist / whitelist / ...
- Can see the graphical representation of the file.
- Invalidation : When we change the file in C.F we have to remove its cache.

DATE: 31/0Q.
PAGE NO.:

SESSION: 33 AWS

- Cloud Trail : It will audit all the IAM as well as root user.
- ISCSI is the protocol used to connect EC2 with EBS.
- In local block we have fast I/O speed. As compared to EBS ~~HDD~~ via networking



- Snapshot : (Backup)
- Replication and Backup are different concepts. We can replicate data using RAID and Hadoop.
- The part that is changed / updated is known as Delta.
- SnapShot creates a notes for the changes made.
- Snapshot is Point in Time backup (PIT)

DATE : 5/Dec
PAGE NO. :

SESSION : 47 AWS

- Any client of cloud is known as Tenant.
- Subnet : 255.255.255.0 or /24
  - Name of the network.
- VPC - Virtual Private cloud
  LAN - Subnet
- When we create a cloud acc. they create an VPC for us and the room is known as subnet. When we use any service, let's say, ec2 then it is stored in a subnet.
- VPC == NAAS → Network as a Service
  - IAAS → Infrastructure as a service

Launching VPC

- Create a VPC
  Name tag : lwvpc
  IPv4 CIDR block : 192.168.0.0/16

IP

Static / manual — Dynamic / Software / Auto

DHCP
dynamic host control protocol

DATE : / /
PAGE NO. :

- Create a vpc <<click>>/Button

Now we have to create a subnet.

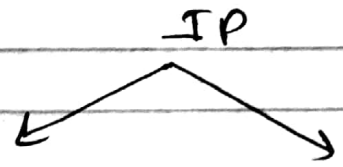
- Select a vpc. (myvpc)
- subnet name: mylab1
- IPv4 CIDR block : 192.168.1.0/24
- AZ. : ap-south-1a

DATE: 6/Dec
PAGE NO.:

SESSION: 48 AWS

route -n
↳ gives the gateway

route del -net 0.0.0.0 → gateway
route add -n 0.0.0.0 gw 192.168.0.1
↳ adds the gateway



- If we use public ip the private ip will be free. For public ip we need to go to an isp and buy it
  ↳ BSNL, jio
- Using Private ip it is impossible to go to internet. To ~~ssh~~ ~~go~~ ping to Internet we need a public ip.
- 192.168.0.0 - 192.168.255.255, 172.16.0.0 - 172.16.255.255, These are the private ip. Unable to go to internet.
- NAT : Network Address ~~Translation~~ Translation
- Router has 2 NIC's. When we connect to internet the sc ip is changed from private to public.

DATE : / /
PAGE NO. :

- SNAT (Source NAT) & DNAT (Destination NAT)

DATE: 12 Dec
PAGE NO.:

SESSION: 53 AWS

- In networking only 2 ips are reserved. network, Broadcast But in AWS, 5 ips are reserved; router, DHCP, network, broadcast, future use ip
- The router which is connected with the ISP is known as internetgateway (Igw). This router is at vpc or far office premises. This router is connected with SN1 & SN2 router then they have internet connectivity
- Subnet1 : 192.168.1.0/24 } subset of vpc
  Subnet2 : 192.168.2.0/24 }
  VPC : 192.168.0.0/16 superset of subnet
  FLSM: fixed length.
- Create a new VPC name: lw
  IPv4 : 192.168.0.0/16
  > Create.
- Subnet1:
  name: lwlab1
  AZ : us-east-1a
  IPv4 : 192.168.1.0/24
  > Create
- Subnet2:
  name : lwlab2
  AZ : us-east-1b

APTH 07

DATE : / /
PAGE NO. :

IPV4 : 192.168.2.0/24
>Create

- ★ Internet Gateway :
  name: lwigw
  >Create
  Attach to VPC : lwvpc

- Launch ec2. (in subnet 1)
  Network : lwvpc
  Subnet : lwlab1
  Auto-assign public ip : Enable
  ↳ Automatically assign pub. ip and create DNATing table.

- Launch ec2 (in Subnet 2)
  Network : lwvpc
  Subnet : lwlab2
  Auto-asign public ip: Disable
  ↳ Isolates the system. No connection from internet.

- We have public ip for ec2 lab1 but even also we are not able to ping it.
  ↳ Because, the instance and router are not physically connected we have to do it manually.

DATE : / /
PAGE NO. :

~~Routing Table~~

- Subnet → Routing Table:

  192.168.0.0 /16
  ↳ that means the OS can connect to any OS inside the vpc.

- If we want to enable internet connectivity we need to update the routing table.

- Now both have same routing table, so we need to create a new routing table.
  ↳ But the only want connectivity with OS1.

- Routing Table > Create
  VPC: lwvpc
  name: lwnewrt
  > Create

↳ > Edit route
  add route
  0.0.0.0/0 Internet gw
  > save route

DATE : / /
PAGE NO. :

- RT > Subnet Associations
  Subnet label
  > Routing Table
    ↳ They are using old one
  > edit
  Change the Routing Table id.
  > same.

- Now we have connectivity in Subnet1 but not in Subnet2.

- In this type of connection, the intermediate as or in this case Subnet1, it is known as Basking host or ~~Stepping host~~ Jump Box.

DATE : 13 Dec
PAGE NO. :

SESSION: 54 AWS

ifconfig enp0s3 1.2.3.4

ifconfig enp0s3:1 1.2.3.5
↳ Virtual NIC
By this we can add ips.

- Actually they all are the same; mac address is same.

ssh 192.168.2.199 -i sshkey.pem

first we have to make it read file. by changeing its permission.

- If someone from outside wants to connect to you, its DNAT & if you want to connect it is SNAT.

- NAT Gateway:

- If we want one way connectivity to private subnet we introduce a device in public subnet, it will take the request from private and forward it through IGW. This device has a ability of SNAT and the device is known as NAT Gateway.

DATE: / /
PAGE NO.:

- NAT Gateway launches only in public subnet.

- VPC > NAT Gateway > Create NAT gw
  name: lwnatgw
  Subnet: lwlab1 (Public Subnet)
  Elastic ip allocation: <<Allocate EIP>>
  b public

  > Create NAT gw

- Still we do not have internet connection so we need to update the routing table.

- Create Routing Table
  name: lwpvtnatgw
  vpc: lwvpc

  > Routes
  Edit routes.
  0.0.0.0/0 NAT Gateway
  > Save routes

DATE : / /
PAGE NO. :

- Subnet > Routing Table
  > Edit
  in for natgw.
  > same

- without public ip we can achieve internet connection, through NAT gateway.

- VPC dashboard
  ↳ Launch VPC wizard
  ↳ it automatically creates a VPC

- SG =: Firewall =: NACL

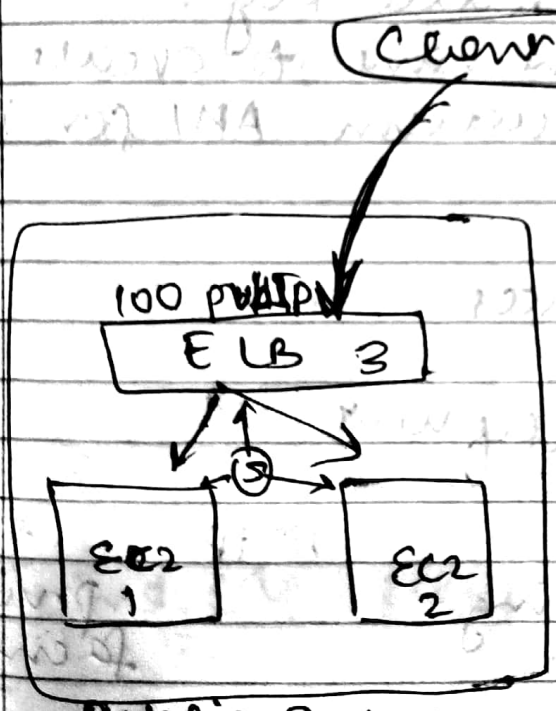
- The one common firewall for entire subnet is NACL. SG apply per os BUT NACL is for the entire network.

- NACL : Network Access Control List.

DATE: 26 Dec
PAGE NO.:

SESSION: 61 AWS py

- Load Balancer is a frontend server which divides the requests equally within the backend servers.
- Big Ip load balancer, Haproxy, Engine x. example of load balancer software
- Manage service for load Balancer: ELB (Elastic load Balancers).
- ELB (load Balancer) is a sub service of EC2
  Pre-requisite: VPC, EC2 instances
  Atleast 2 Subnets.



DATE: / /
PAGE NO.:

ARTICLE

- LB recieves a ~~req~~ request from the client & they recreates the request & sends them to the EC2 instance.

Launch 2 AMI's
VPC → default
subnet → 1a
Pub IP → Disable

SG:
All Traffic 0.0.0.0/0, ::/0
>> Launch
Proceed without
private key.
***. But we have to create
the custom AMI for it.

~~Load~~ Load Balancer
Name: _
Scheme: default.
LB Port

HTTP	8080 pub facing port.	HTTP	80 private facing port

Assign security gp for LB

Type	Port	Source
HTTP	80	Anywhere

DATE : / /

PAGE NO. :

Health check:

Ping Protocol : HTTP

" Port : :80

Path : /index.php

Add EC2 instance:

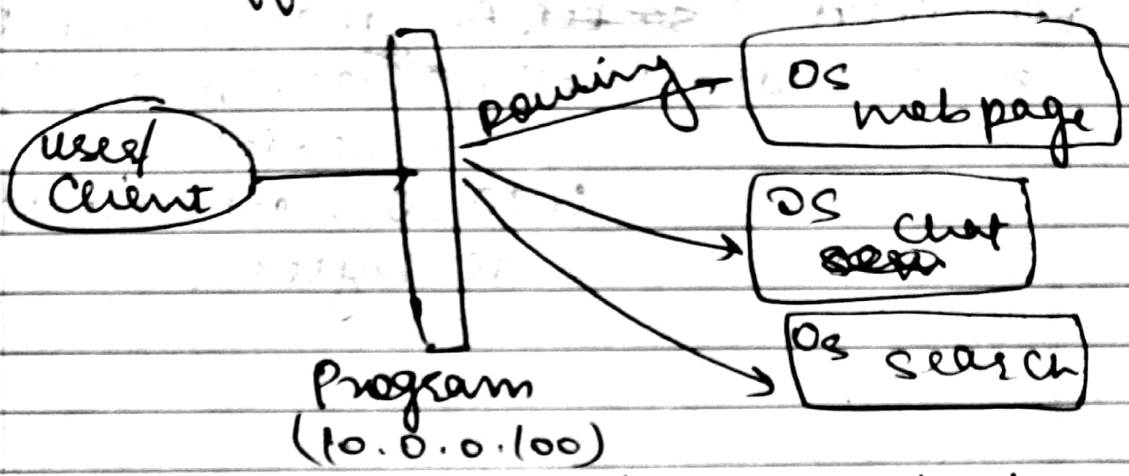
Add the instances.

* Instead of web check on cmd with curl cmd.

DATE : 27 Dec
PAGE NO. :

SESSION: 62 AWS Py

- Microservices are the sub services of a major service (web page).
- All the microservices are present on the different servers



for webpage : 10.0.0.100/webpage
for chat : 10.0.0.100/chat
: " /search

↳ path based routing (PBR)

- Three types of LBs : Application Based, Network Based, Classic

ALB → PBR
NLB → checks L7 (Application layer)
NBB → checks L4 (Network layer)

DATE : / /
PAGE NO. :

Create EC2 x2
SG : All.
Net : dy
Sub : (Same) any
Pub : Yes
Tag :- Chat
- Search

Chat os
yum install httpd php
cd /var/www/html
mkdir chat
cd chat
index.php
< pre >
< ?php
print 'welcome to chat app!';
print 'ifconfig';
↳ or /usr/sbin/ifconfig
?> </pre>

systemctl enable httpd --now

Search os
mkdir /var/www/html/search
index.php

DATE : / /
PAGE NO. :

- # . Load Balancing → Target Group

• ngrok gives us public ip for private ip.

~~2nd~~ . Instances

health check : /chat/index.php

Register Target Node : chat .

≫ Create Target Grp.

Load Balancer

Name : lwalb

Sch : int - facing

Ip : Ipv 4

Az : 1a (same zone)

: 1b (select 2 for high availability)

New SG.

TCP 8080 0.0.0.0/0

SG:

existing name : /chat

DATE : / /
PAGE NO. :

ALB → Listener
Rules → Add {+}

If	Then
/chat/*	w/chat
/search/*	w/search

SESSION 64 AWS

PAGE NO.:
DATE: 3/Jan/21

- ELB is generally slow, because it decapsulates the packets, thus latency arises.

- NLB is generally faster; works on layer 3. They do not check layer 7. Therefore cannot able to read the http header.

- CLB is cheap.

- ELB & ALB is free for some time but NLB is NOT FREE. in free tier.

Configure the CLB (existing)

- Cross zone LB: means it can direct the request in the instances located in different zones (AZ).
- when we visit a webpage for the first time the create a session; known as stickiness
  - ↳ we can enable stickiness in our LB.
  - ↳ It keeps all the info about the webpage. concept it known as cookies.

curl <<URL>> -I
  - ↳ gives the header of the packet.

PAGE NO. :
DATE : / /

- Connection draining: When the web app doesn't reply in the given time it will direct the packet to another instance. But sometimes, the LB holds the client for sometime in order to connect to the same node. This time where LB holds the client is known as connection draining.

- CLB the service is known as ~~Stickiness~~ connection draining
  In ALB it is known as debalance

- ~~Its~~ NLB gives very high performance.
  We can provide one elastic IPv4 with the help of EIP
- Backend servers are also known as Target groups.

- By default, cross zone is disabled in NLB.

cat var/log/httpd/access_log.
  L Tells from which IP, at what time the request comes.

- In NLB there is direct connectivity bet the client & the web page. But in another LB there is connectivity bet client & LB and LB and webpage.

PAGE NO. :
DATE : / /

- Here in the webpage we have to allow all the connectivity (ip) in SG.

PAGE NO.:
DATE: 9 Jan 21

SESSION: 69 AWS

- RDS : Relational Database ~~System~~ Service
  - ↳ Managed service

- DBAAS : DataBase As A Service
  - ↳ For creating a DB we do not have to launch an EC2 instance.

DB
- ↳ Standard Database

TASK : Create EC2 instance
- ↳ Apache web server
- ↳ php : wordpress (webapp)
  - search: wordpress github
  - (download php code)

Engine option
- ↳ MySql
- version: MySql 5.7.31
  - (or any v. compatable with php code)

Templates
- ↳ Freetier

Settings
- ↳ mysql (name)
- ↳ Master uname : nimal
  - passwd: ******
- ↳ db instance size : db.t2.micro

PAGE NO. :
DATE : / /

- Performance Tuning: For different use cases we have same hardware but our requirements are different. So we tune them for optimization.
- rh442 is a program for performance tuning

Storage
- SSD
  - 20 / Enable autoscaling
- max storage threshold: 1000

Availability & durability
- Launches db in multiple AZ. And the second will be on standdown. It will only become active when the first goes down.
- only available in production.

Connectivity
- VPC
  - default
- Subnet
  - default
- Public Access
  - Yes
- SG
  - create new: sgformysql
- AZ
  - ap-south-1a

PAGE NO. :
DATE : / /

#Additional config
↳ port no
↳ 3306 (default)

#Db auth
↳ passwd auth (default)

#Db options
↳ name : mydb
· param. gp: default. mysql5.7
option gp: default: mysql-5-7

>> create db.

#rpm -q mysql
or
yum install mysql

~~# mysql -h <host name> -u -p~~

#mysql -h <<host_name>> -u himal -p

mysql> show databases;
> use mydb
> show tables;
> create table user(int id, varchar(255) name,
varchar remarks);
> create table user(id int(200),
name varchar(255), remarks
varchar(255));

11

PAGE NO. :
DATE : / /

> select * from user;

> exit;

Databases
↳ Action
↳ Create read Replica

settings
~~↳ Replica~~
~~↳~~

PAGE NO. :
DATE : 10/Jan21

SESSION: 70 AWS

RDS
↳ Databases
↳ # Read Replica → Singapore

localhost
EM

#yum install mysql
#mysql -h <<mysql_rep_URL>> -u vimal -p

mysql> show databases;
mysql> use mydb;
> show tables;
> desc ~~cables~~ list;
Mumbai > insert into list values (1, "vimal", "ok");
original > select * from list;
Singapore > select * from list;

- Exact same copy will come in both the db.

AWS
- To make the read replica a stand alone we ~~read~~ change it to edit it.

Actions
↳ Promote
>> Continue >> Promote Read Replica

cloudwatch: ~~Monitor~~ monitors the ~~instances~~ metrics (events, logs, etc)

PAGE NO.:
DATE: / /

- Auto-scaling Group: It will automatically launch instances when required. For ex in ALB.

Autoscaling
- Autoscaling group.
  - Create a launch template
    - >> myec2launchtemplate
    - >> v1 desc
  - Launch template content
    - >> AMI : my-vimal-aws-webserver-php
    - >> t2.micro
    - >> key
    - >> VPC
    - >> SG
      - ↓
      - Inbound All Traffic
    - >> launch template
  - Create AS group
    - Name : myASGweb
    - launch temp: myec2 launchtemplate
  - Config settings
    - VPC : default
  - Load Balancer
    - Attach to an ELB.
    - Add Target gp: Classic LB

PAGE NO. :
DATE : / /

- AP size : 10 (Max)
- Scaling Policies
  - Target Tracking S.P.

PAGE NO.:
DATE: 16/Jan/21

SESSION: 73 AWS

- DNS is the facility that enables one name/ip for many ips. lets say if one ELB goes down, the client doesn't have to enter ip of second ELB DNS will automatically transfer the request.

- For DNS, we have a file, /etc/hosts to create local hosts.
- DNS assigns names for the ip address which is easier to remember.

vim /etc/hosts

8.8.8.8 vimal cw a
↳ It has multiple names

- cw.vimal.in ← FQDN / Host Name
  Domain Name
  ↳ unique
  ↳ we can use vimal.com vimal.org but not vimal.in. we can purchase them.

- TLD: Top level Domain
- FQDN: Fully Qualified Domain Name
- TTL: Time To Live
- lookup when we want ip for the domain name.
- Reverse lookup when we want the name for ip.

nslookup -debug www.google.com
↳ also gives TTL ↳ gives ip from DNS db

- A type when we want to connect dns to ip (Host → IP)
- PTR type when (IP → Host)

- DNS as a service : Route 53

#Route 53
↳ Domain
↳ Registered Domain
↳ Register Domain
Windia123.com / Ourlinuxworld.com
↳ Add to cart
≫ Continue

↳ Hosted zones (DNS page)
↳ They automatically create 2 records for us.
↳ Ourlw.com
↳ Create Record
↳ www.Ourlw.com (R.Name)
↳ A (R.Type)

#Launch EC2 instance
↳ myjirmalos
↳ 3 instances.
↳ Ourlw-assign pub ip : Enable
↳ Tag value : Osdns

Give the ip to DNS.

PAGE NO. :
DATE : / /

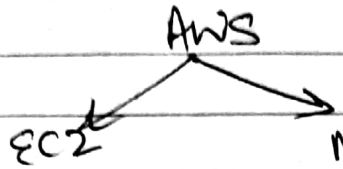
~~dsloop~~
nslookup www.google.com
↳ gives the ip from DNS db.
↳ -debug gives additional info like TTL.

- when the ip changes we select Alias option
↳ <<choose endpt>>
↳ ELB
↳ <<Region>>
↳ <<choose LB>>

- If we want to give a bigger DNS a small DNS we use CNAME as Record type.

PAGE NO. :
DATE : 17/Jan/21

SESSION: 74 AWS



↳ DNS

↳ DNSAAS
(Route 53)
↳ Route Policy

- H.N $\xrightarrow{A}$ IPV4
  H.N $\xrightarrow{AAAA}$ IPV6
  I.P $\xrightarrow{PTR}$ H.N
  H.N $\xrightarrow{CNAN/ALIAS}$ H.N

cat /etc/resolv.conf
↳ DNS conf file. (DNS server)

- www.lw.com $\xrightarrow{A}$ 100, $\xrightarrow{A}$ 200
  DNS
  ↳ LB → RR (Round Robin)
  ↳ when has more than one ip then automatically ~~Apache~~ acts as LB.

dig www.ourlw.com
↳ details of host name.

PAGE NO.:
DATE: 23/Jan/21

SESSION: 78 AWS

- 169.254 IP address are only for the communication in a network. These are known as ~~loop~~ link local ip.

curl http://169.254.169.254/latest/metada
↳ Gives the metadata of an instance.

Action → System&Monitor → Get system logs
↳ gives all the processes. eg. ~~log~~ boot proc.

Launch instance
↳ step: 3
↳ Advanced Details
↳ user data → #!/bin/bash
yum install httpd -y

- This cmd will be picked by cloud init and run at the boot time.
- Shebang (#!) is mandatory.
- By this method we can update the metadata.

- Geolocation: DNS based on location.

PAGE NO.:
DATE 24/Jan/21

SESSION: 79 AWS

- Edge locations are the smaller data centers (AZ) used for cache servers. Cloud front is a way by which we can create a cache in these locations.
- Global Accelerator (GTA): Amazon provides their high speed connectivity from their private network but not creates any cache in edge locations.

- Unicast: One to One connectivity
- Broadcast: One to many "
- Anycast: We provide same ip to multiple devices

Global Accelerator
- Create Accelerator
  - <Acc name>
  - < type> standard
  - <IP type> IPV4
  - >> Next
- Add listeners
  - Ports: 80
  - protocol: TCP
  - >> Next
- Add End Points group
  - Region: us-east-1
- Add endpoints
  - End pt type: EC2, (generally ALB, ELB)
  - End pt: Automatically pick running instance in-that region

>> Create Accelerator

PAGE NO. :
DATE 31/Jan/21

SESSION: 84 AWS

- IAM (Identity & Access Management): It checks the id & pass and provides the access policy for all the users.

H Create a bucket x2
  L tw-test-123 (name), tw-my-test-123456
  L Block all pub access
    > Create.

- Add files

- whenever we create a resource, AWS gives an unique name known as ARN (Amazon Resource Name).

- arn:partition:service:region:account-id:resource-type:resource-id
  L ARN syntax.

- IAM Dashboard
  > Create multi-factor authentication for root account.

L Users
  L Add user
  L User (username)
  L Access type
    L ~~Programmable Access~~, ~~Console~~ Access.
    L Console Access
  L custom password.
    > Next: Permissions

PAGE NO.:
DATE: / /

> Next: Tag
> Next: Review
> Create User.

- This user has no access policy. Therefore it cannot do anything

↳ Policies
↳ Create Policy
↳ JSON

```
"Statement": [
    {"Action": "S3:ListAllMyBuckets",
     "Effect": "allow",
     "Resource": "S3"
    }
]
```

> Review Policy

OR

↳ Visual Editor
- Service: S3
- Action: ListAllMyBuckets

> Review Policy
↳ Name: s3-policy
Descp: ——
> Create policy

↳ Attach existing Policy (S3-policy)

- Now this user has the power only to list the buckets in S3.

PAGE NO. :
DATE : / /

1. Create a role
   - EC2 (use case)
     - > Next
   - S3FullAccess (policy)
     - > Next
     - > Next
   - myrole (role name)
     - > Create role.

- EC2 Action (select an instance)
  - Security
    - modify
      - IAM role
        - myrole (IAM role)

PAGE NO.:
DATE 6Feb21

SESSION: 88 AWS

- EB: Elastic Beanstalk
- PAAS: Platform As A service
  1. One click and everything is configured. They are highly intelligent service which will test, create LB, autoscaling gp etc.

#EB

>> Create Application
↳ App name: mytestapp
↳ Tag: env: dev
↳ Platform
  ↳ PHP
↳ App code: upload your code
↳ source code origin: Local file
  ↳ version label: my-testapp-v1
  (very important)

<sample / index.php in win OS>

```
# index.php
<? php
print "App for AWS EB!!";

?>
```

>> Configure more option
(lot of advance options are provided)

>> Create App

PAGE NO. :
DATE : / /

- They in the back will create various things like EC2 instance and other services that you selected.

PAGE NO.:
DATE: 20/Feb/21

SESSION: 95 AWS

- FAAS : Function As A service
- Lambda is a service which will ~~test~~ run the algo or function.
- Whereas in elasticbeanstalk we deploy an app and behind the scene it will launch an ec2 instance
- While uploading the file in eb we ~~use~~ first need to compress it (zip file) and then upload it.

Lambda
- Create function
  - Author from scratch
    - fn name : mylwf1
    - Python3.8 runtime
- It behind the scene will create a us and will charge us on the basis of the no. of time the func is run.

mylwf1 ~~app~~
- prg1.py

```
def f1():
    print("this is first test fm f1 fnc")
def f2():
    print("2nd test fm f2 fn")
```

PAGE NO. :
DATE : / /

#. Runtime settings
handler
↳ ~~lambda_handler(event)~~
↳ prog1.f2()

- To test the code we can test it by clicking on Test button.

- When we create a fn in ~~lambda~~ lambda it is compulsory to give two arguments otherwise it'll fail.
↳ (events, contexts)
- One of the capability of cloudwatch is that they keeps all the logs.

x → checks from where the data comes.
↳ stored in json file
↳ def f2(events, contexts):
↳ x = events (by default, var)

- We can only run a function with lambda :
(i) Max runtime : 15 min
(ii) Memory : 128 - 10240 MB

PAGE NO. :
DATE : / /

mys3fl
↳ lambda_fn.py
def wsl(x, y):
    return ("s3 comes to me..")

*. Create a s3 bucket.
↳ Properties
↳ Event notification
↳ Create E.N
↳ E. Name: myevents31
↳ ☑ Put
↳ Destination
↳ ☑ Lambda fn.
lambda fn : mys3fl.
(REGIONAL SVC)

*. Lambda
(+) Add Trigger
↳ S3

PAGE NO.:
DATE: 21/Feb/21

SESSION: 96 AWS

- ~~Elastic cache~~:
- In memory db: When we provide some RAM as a db then it is known as in-memory db. First time they will fetch the data from db. Then they store it on the RAM. Sometimes it is also known as session database.

- Backup is the feature of redis. Not available in memcached.

cache server
↙ ↘
memcached Redis

service > Database > Elastic cache
Launch an EC2 instance
↳ Redis
↳ Create
↳ cluster Engine: Redis
↳ Amazon cloud
↳ name: myredis
↳ port: 6379 (by default)
↳ Node type: cache.t2.micro
↳ Multi-AZ [X]
↳ or Replica: 0
↳ Subnet gp
↳ ~~Manage~~ myredissubnet
↳ vpc: ~~default~~ same as EC2
↳ subnet: ap-south-1a
~~Backups~~
↳
>> Create.

PAGE NO. :
DATE : / /

ec2

yum install python3-pip -y
pip3 install redis
↳ driver for redis

# python3
>>> cast = "mobile"
>>> import redis
>>> r=redis.StrictRedis(host='—', port='6379', password="")
↳ IP of redis server

>>> r.set("x", cast)
↳ value
↳ variable name (key)
>>> r.get("x")
↳ gives the value of x.

- Facility of Read-replica and Multi-AZ are available in redis

memcached
↳ No of nodes: 1
>> Create.

- After a rebooting our server we loose the data because these are cache servers and the data is stored on the RAM.

PAGE NO. :
DATE : / /

memcached

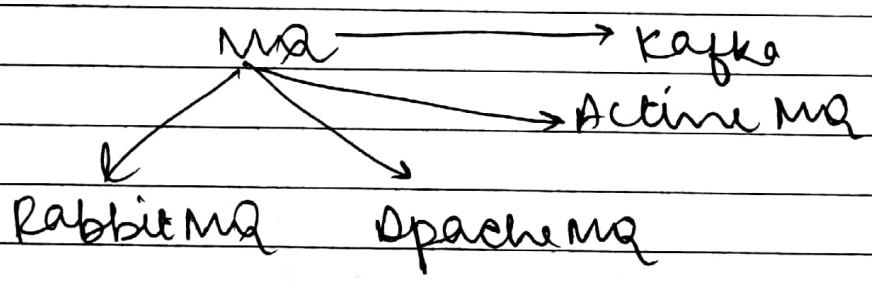
```
# pip3 install pymemcache
# python3

>>> cart = 'telescope'
    from pymemcache.client.base import Client
    c = Client(('localhost', 11211))
    c.set('x', cart)
    c.get('x')
```

PAGE NO.:
DATE: 28/Feb/24

SESSION: 101 Aws

- Middle ware: Programs act as middle man in ~~two programs~~ microservices, which facilitates some services preventing data loss.
  - SQS (Simple Queue Service)
    - ↳ managed Messaged Queuing
    - ↳ MQ add two programs through SQS through with loss of data doesn't happens in the case of down time.
  - Tightly coupled (Sync): when two program are properly/rightly connected.
    - ↳ directly.
    - ↳ There is no middle man.



SQS → Processes
- ↳ ~~send~~ receive/download
- ↳ use
- ↳ delete

> Create queue
- ↳ name: myq1
- ↳ standard

>> Create
- ↳ Default visibility timeout
  - ↳ To avoid ~~see~~ replication of data in VB.

PAGE NO. :
DATE : / /

lambda
- s3pegs
- Py 3.8

- we can also make event notification from lambda.

- Add Trigger
  - S3 (svc)
    - testmyew1234567 (bucket name)
    - PUT (event type)

```
# def lambda_handler(event, context):
    print("lambda test for sqs")
```

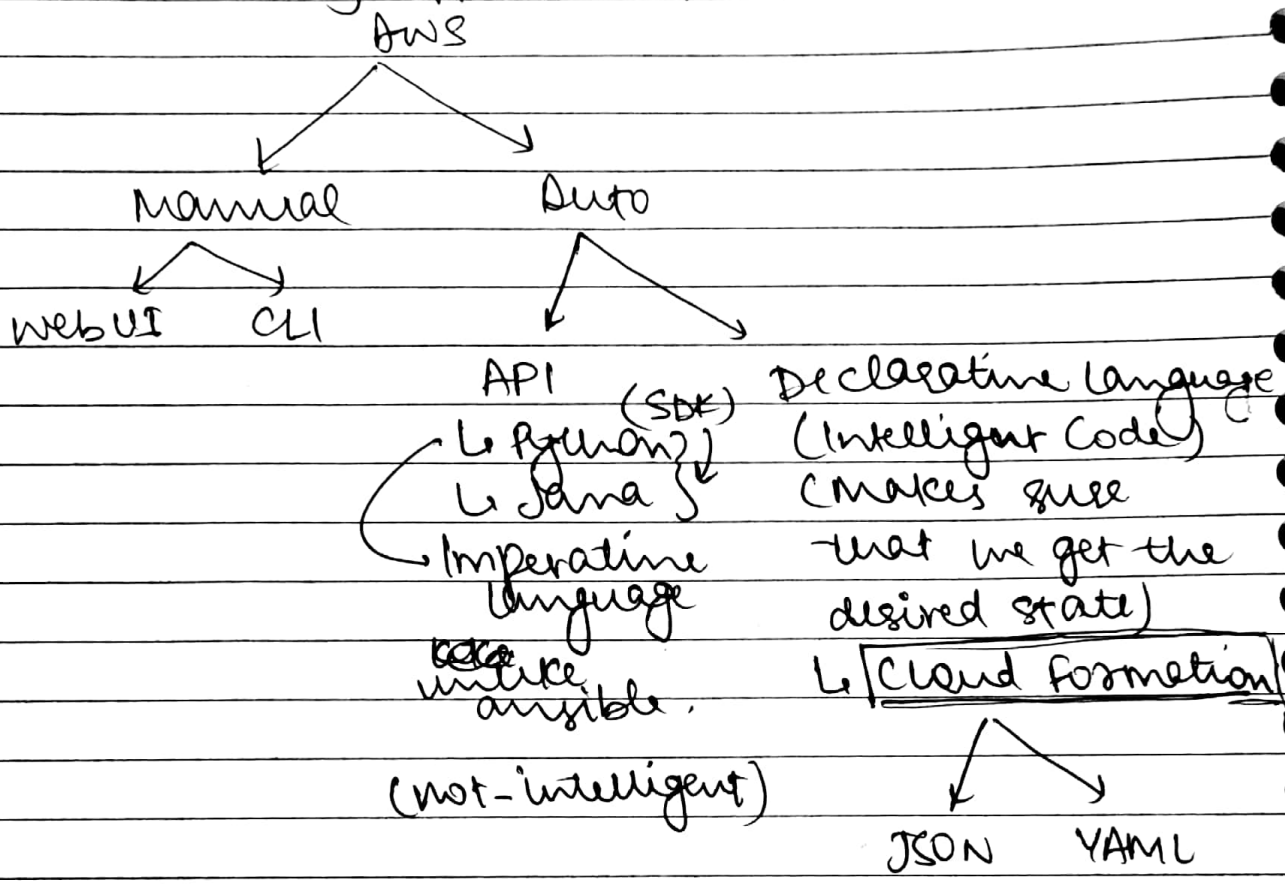
- Add destination
  - Condition: On Success
  - destination: SQS queue
    - lambdaSuccess

SQS queue (new)
- name: lambdaSuccess
  - >create.

PAGE NO.:
DATE 6 Mar 21

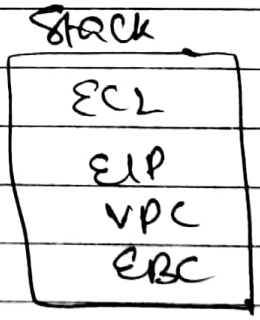
SESSION: 108 AWS

- Cloud Formation: (IAAS) (IAC) ← Infra as a Code
  - we automate the whole infrastructure
  - Only applicable for AWS.



- Terraform: (IAAS)
  - Multi-cloud automation possible.
  - declarative language.

- CF's used for big infrastructure



Cloud Formation
> Create Stack
↳ Template is ready
↳ Upload a template file

web: aws cloud formation key: EC2

- DSL: Domain specific language
  ↳ have their own key value pairs, syntax, etc.

CF

#"CF.yml" →
Type: AWS::EC2::Instance
Properties:
ImageId: ami-
InstanceType: t2.micro
AvailabilityZone: ap-south-1a

Resources:
MyFirstOSbyCF:
Type:
Properties:
Im

CF
↳ <<upload the file>>
>> next

↳ Stack name: myfirst stack
↳ Parameters
↳ we can create dynamic code

>> next.
>> Create stack

PAGE NO. :
DATE : / /

InstanceType : !Ref x ← dynamically creates a var (parameter)

cf.yml
parameters:
  x:
    Type: String
    description: This is for InstanceType
Resources:
  my—
    Type:
    prop:
      InstanceType : !Ref x

cf.
↳ select the cf.
  ↳ click on update
    (to update the .yml file)
↳ Replace current template
↳ <Upload the file>
    >> Next.

• Now they ask us the value of the parameter.
    >> Next
    >> Next
    >> ~~Create~~ update Stack

• Replacement : True
  ↳ means that the last instance will be terminated and new will be launched.

PAGE NO.:
DATE: 7 Mar 21

SESSION: 109 AWS

Cloud Formation

mkdir all
cd all
notepad aws.yml

*. For automation first create a diagram and a plan.

Plan
I : create vpc
II : create I.gw
III : attach Igw to vpc

Resources: ——→ Parameters:

VPC:
Type: AWS::EC2::VPC
Properties:
CidrBlock: '10.0.0.0/16'
EnableDnsHostnames: True
EnableDnsSupport: True
InstanceTenancy: default
Tags:
- Key: Name
Value: LW VPC CF

AMI:
Type: String
Default: ami-xx

>> Create stack
L Template is ready
L Upload the template
L << choose file >>
> next
L Give a name (stack name)

> Next
> Create Stack

Resources:
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: LW Igw CF

> Update the stack (unsafe)
  ↳ Replace Curr
  ↳ Upload the file.

  VPCGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      VpcId:
        Ref: VPC ← (Resource name)
      InternetGatewayId:
        Ref: InternetGateway

      (or)

      VpcId: !Ref VPC
      InternetGatewayId: !Ref InternetGateway

> Update the stack.

PAGE NO. :
DATE : / /

IV : Subnet.
- AZ
- VPC
- cidr block

{ !Ref or fn::Ref }

!Select [0, !GetAZs]

SubnetA:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    CidrBlock: 10.0.1.0/24
    AvailabilityZone: ~~fn::GetAZs~~
    Tags:
      - Key: Name
        Value: mysubnet1b1

>> Update stack

V : Routing Table.
VI : Create Route
VII : RT → Subnet (attach/update)

RouteTable
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: LW RT CF

>> update the stack

InternetRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGateway
  Properties:
    RouteTableId: !Ref RouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway

>> update the stack

S—R—T—A—
Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
  SubnetId: !Ref SubnetA
  RouteTableId: !Ref RouteTable

>> update

Instance
Type: AWS::EC2::Instance
Properties:
  ImageId: * ~~AMI-ID~~ * !Ref AMI
  KeyName: !Ref keyname
  SubnetId: !Ref SubnetA
  Tags:
    - Key: Name
      Value: LW_DS_CF

```
securityGroupIds:
  - sg-xxxxxx

MyEIP:
  Type: _____ :: EIP
  Properties:
    InstanceId: Instance


SecurityGp:
  Type: AWS::EC2::SecurityGroup
  Properties:
    SecurityGroupIngress:
      Protocol: -1  ← for all ports/protocol
      CidrBlock: 0.0.0.0/0

Outputs:
  PublicIp:
    Value:
      Fn::GetAtt: [MyEIP, PublicIp]
```

PAGE NO.:
DATE 13/Mar/21

SESSION: 114 AWS

- KMS → Encryption → EBS, EFS, Lambda, LB, RDS
  - Key management service

- ~~Cry~~ Cryptography → Greek word
  - Secure + writing

- Plain Text + key → Algo (DES, AES) encryption → Cipher Text + key
  Cipher Text + key → decryption → Plain Text

- In this we use ~~only~~ same key for encryption & decryption.
  This is known as symmetric key

- AES is more powerful. Used by all the banks, etc
  DES is faster. ∴ easier to break.

- Asymmetric key algorithm where encryption & decryption keys are different.
  - Public key
  - Private key

- Companies often use HSM appliances for the safeguard of their keys.

PAGE NO. :
DATE : / /

- Dedicated Tenancy: Hardware only provided to only one tenent.

KMS
- Configure key
  - Symmetric
    - > Create key
- Add labels
  - mykmskey1
    - > Next.
- Key Administrator

de-kem

aws configure
↑ we login with the IAM user.
mkdir mysecurefolder
cd
notepad s.txt
credit card
redhat.
aws kms encrypt --plaintext fileb://s.txt --keyid <——> --query ciphertextBlob --output text > secure.txt

aws kms decrypt --cipher-blob fileb://secure.txt --keyid <——>

- To add extra security we can also encode it.

PAGE NO.:
DATE: / /

- The data encoded by base64 always ends with an equals to (=) symbol.

certutil -decode secure.txt secure_d.txt
  ↳ base64 decode

aws kms decrypt --cyphertext-blob fileb://secure_d.txt
  ↳ decrypts the data in ~~decod~~ encoded form.

lambda
  ↳ fsecure (fn name)
    #
    def lambda_handler(event, context):
        ~~x = "hello"~~
        ~~return x~~
        db = "redhat"
             ↳ ~~encr~~ cipher text

  ↳ edit env. var.
    key : MYNAME
    value : nirmal

  ↳ fsecure
    *
    import os
    def lambda_handler(—):
        y = os.environ["MYNAME"]
        return y

PAGE NO. :
DATE : / /

↳ Edit Environmental var
↳ ☑ Enable encryption at rest/ in transit

AWS KMS key
↳ use a customer master key
3. <arn of mynewkey>

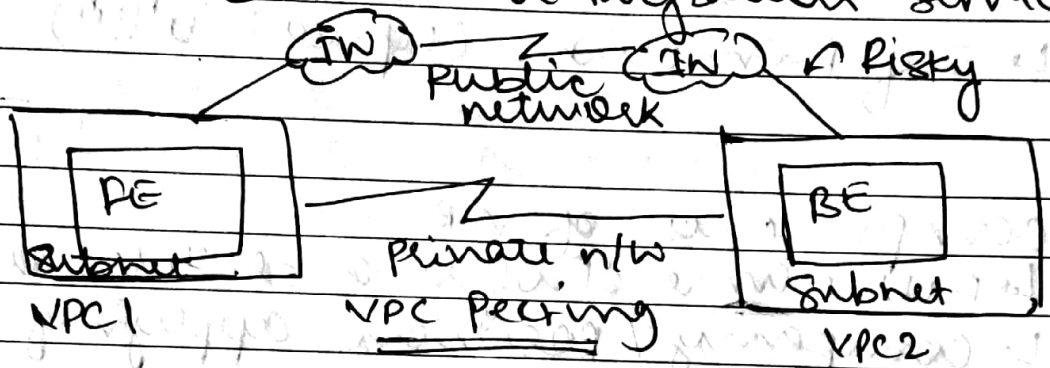
KMS : Policy
↳ Action : Decrypt
↳ Resources : Specific
↳ <arn of mykey1>
↳ name : my_lambda_kms
> Create policy

PAGE NO. :
DATE 27 Mar 21

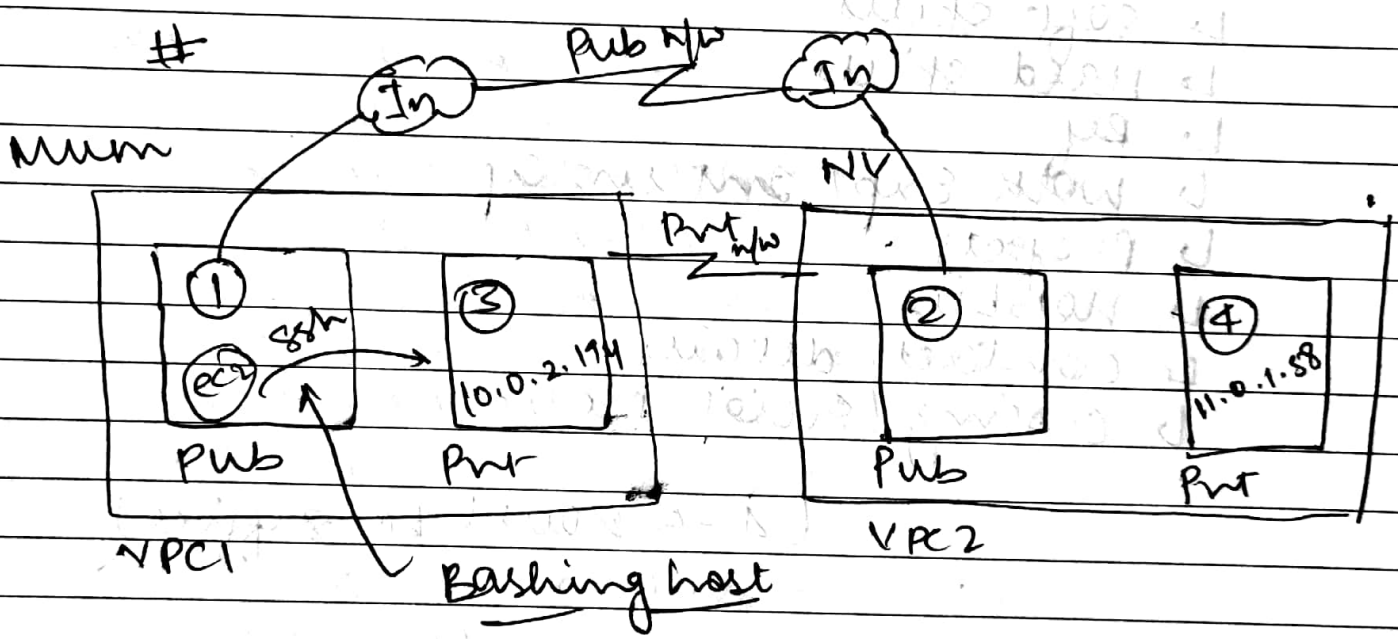
SESSION: 126 AWS

DMS

↳ Database Migration Service



#



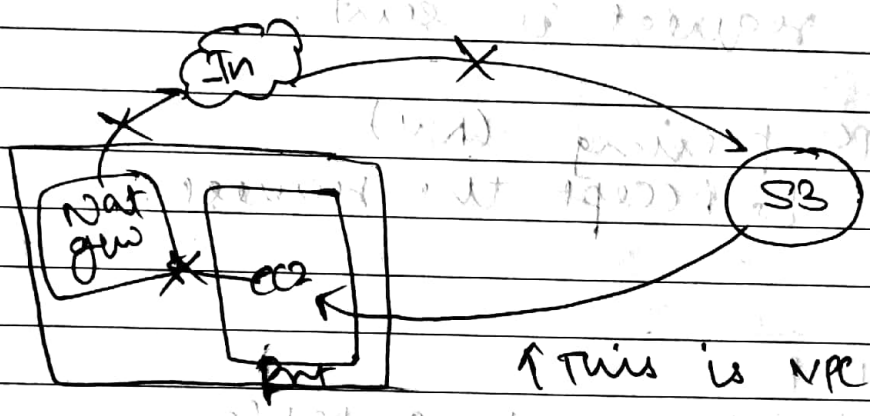
• VPC peering will give us a illusion that both VPCs are same and we can connect bet. a VPC.

EC2 instance (Mumm)
↳ VPC1 - prvt subnet
↳ VPC2 - prvt subnet (NV)
#

PAGE NO. :
DATE : / /

Bashing host

VPC endpoint



↑ This is VPC endpoint

- When we connect two ec2 instances in pvt. Subnets then the connection is VPC peering.
- In VPC peering & VPC endpoint the connect via aws private network.

PAGE NO. :
DATE : / /

VPC (Mumbai)
- Peering connection
  - peering tag: mypeer_NV
  - VPC : VPC1 (requester)
  - ~~VPC (Accepter): VPC2~~
  - Acc: my account
  - Region : Another region
  - VPC (Accepter): VPC2

request is sent.

VPC peering (NV)
- Accept the request.

VPC1
- Main routing table
  - Routes
    - (edit route)

(Destination)	(Target)
10.0.0.0/16	Peering Connection

↑ (VPC2 cidr)

> Same routes

VPC2
- main routing table
  - Routes
    - (edit route)

10.0.0.0/16	peering connection

↑ (VPC1 cidr)

PAGE NO. :
DATE : / /

VPC

↳ Endpoint

↳ service category: Aws service

↳ (select service) (S3) (type: gateway)

PAGE NO.:
DATE: 3 Apr 21

SESSION: 134 AWS

Storage

- AWS → Storage → Block st → EBS
  - Throughput optimized HDD
    - st1
  - Max throughput 500MiB/s

- EBS
  - gp2 ; throughput : not app.
  - st1 ; th ——— : 20/13
  - sc1 ; th ——— : 6/40
    - Throughput also depends on the size of harddisk we purchase

- IOPS : how much operations (input/op) an SSD can perform in one second.
- ~~Not~~ SSDs are not suitable for big data.
- But best for smaller ~~good~~ files & great in number.

PAGE NO.:
DATE: 10/Apr/21

SESSION: 139 AWS

Athena
- Glue (service)
  (Glue Data catalog)

- In logs we do not store metadata or column name. but only the data. This kind of data is raw data.
  ex: 1, nimal 100 OK
  2, jack 200 good
  3, pep 500 OK
  } raw data ← Big Data.
  - Hadoop, hdfs

- Data lake is the big storage where all the raw data is stored

- Through hdfs we can create data lake. But AWS has their own data lake service known as S3.

- Athena is a tool mainly for analytics of big data. It is tightly coupled with S3.
  For athena S3 is the data source/store.

- For analytics we first need to create a seperate metadata/schema and attach it to the raw data. This schema is created by glue.

PAGE NO.:
DATE: / /

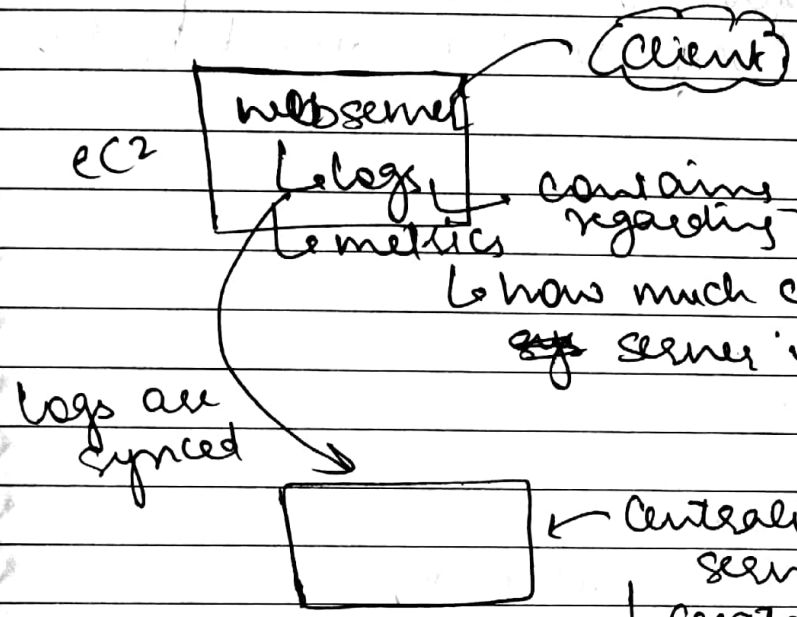
I. Upload raw data on S3
↳ (given in prev page)

↳ bucket (awsathena)
↳ folder : april
↳ folder: may

upload data in
awsathena/april/data.csv.

II. Athena
↳ Query data in Amazons3
↳ Add a table & enter schema manually.
↳ Database
↳ db: lwdb
↳ location of ip data: S3UR
↳ table name: emptable
↳ Dataformat : CSV
↳ Column name: id
↳ ——→ type : int
Add a column
↳ ——→ name: name
↳ ——→ type : string
Add
↳ name : salary
↳ type : int
Add
↳ name: remarks
↳ type: string
> Create table

- We need to create a dir in S3 awsathena/output/ and add this path as output location.
- Now, we can run all the SQL queries



- rsyslog
  - Creates personal localized logging server.
- AWS centralized logging service is cloud watch log.

PAGE NO. :
DATE : / /

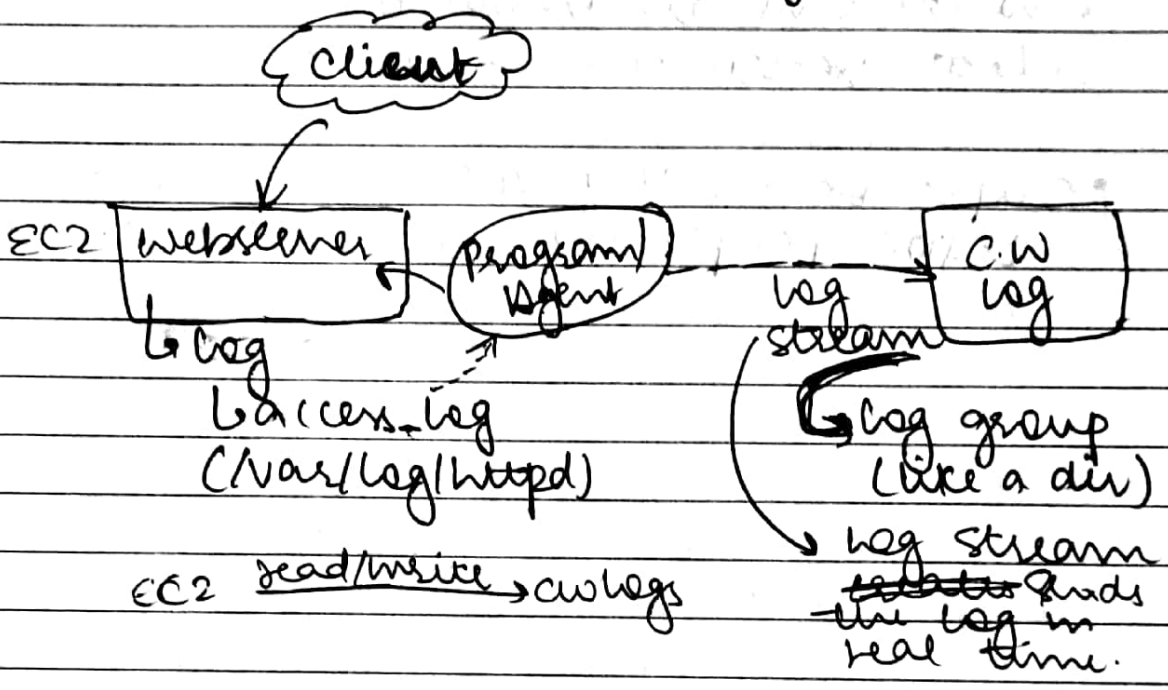
CloudWatch { cloud watch logs
Logs }

- To copy the logs to cloud watch we need agents, who will upload the logs to cloud watch.
  We the agent we can also upload our local logs to the cloud watch.

PAGE NO.:
DATE: 11 Apr 21

SESSION: 14 AWS

- For analytics we need to centralize all the local logs* and do some analysis on it, so AWS provides a service cloud watch logs.



Create httpd server.

vim /var/log/httpd/access_log
↳ file containing all the logs

cloud watch
↳ logs
↳ Create log gp
↳ name: weblg
↳ Retention setting: 1 day

PAGE NO. :
DATE : / /

webLg
↳ Log stream (Create)
↳ name : webls

ec2 webserver
yum install awslogs -y
↳ agent for cloud watch logs.

# cd /etc/awslogs/
# vim awscli.conf
region = <region_name>

vim awslogs.conf
[/var/log/messages] ← name or [webblockname]
log_group_name = webLg
log_stream_name = webls
file = /var/log/httpd/access_logs

systemctl restart awslogsd

IAM (Create role)
↳ EC2
↳ Cloud watch logs Full Access
> Create role
role name : ec2cwrole

EC2
↳ webl (instance)
↳ Action > Security > Modify Role
add ec2cwrole.

PAGE NO. :
DATE : / /

- cloud watch also provides the service of metrics
- Metrics : Monitors all the logs
  Eg: CPU utilization, n/w, storage, etc.

cloud watch
  - Metrics
    - All Metrics
      (lists all metrics that we can monitor)

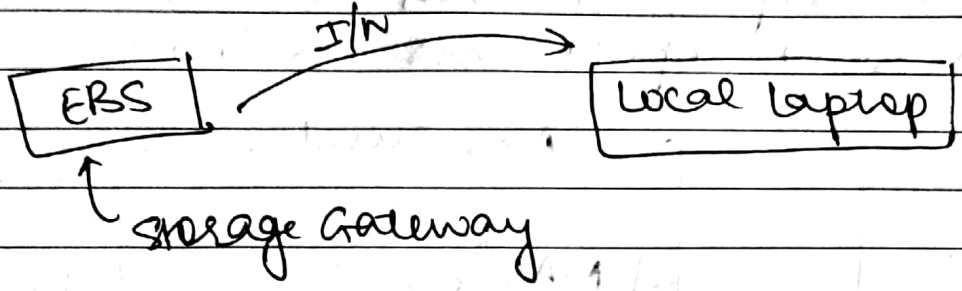
- SNS : Simple Notification Service
- When we use the alert service in cloud watch it alerts us by using SNS and SNS further notify us either via e-mail, sms, etc.
- Dashboard in cloudwatch is globally available in AWS.
  (all regions)

SNS
  - Topic
    (Create)
    - Create subscription
      - email, sms, —— etc

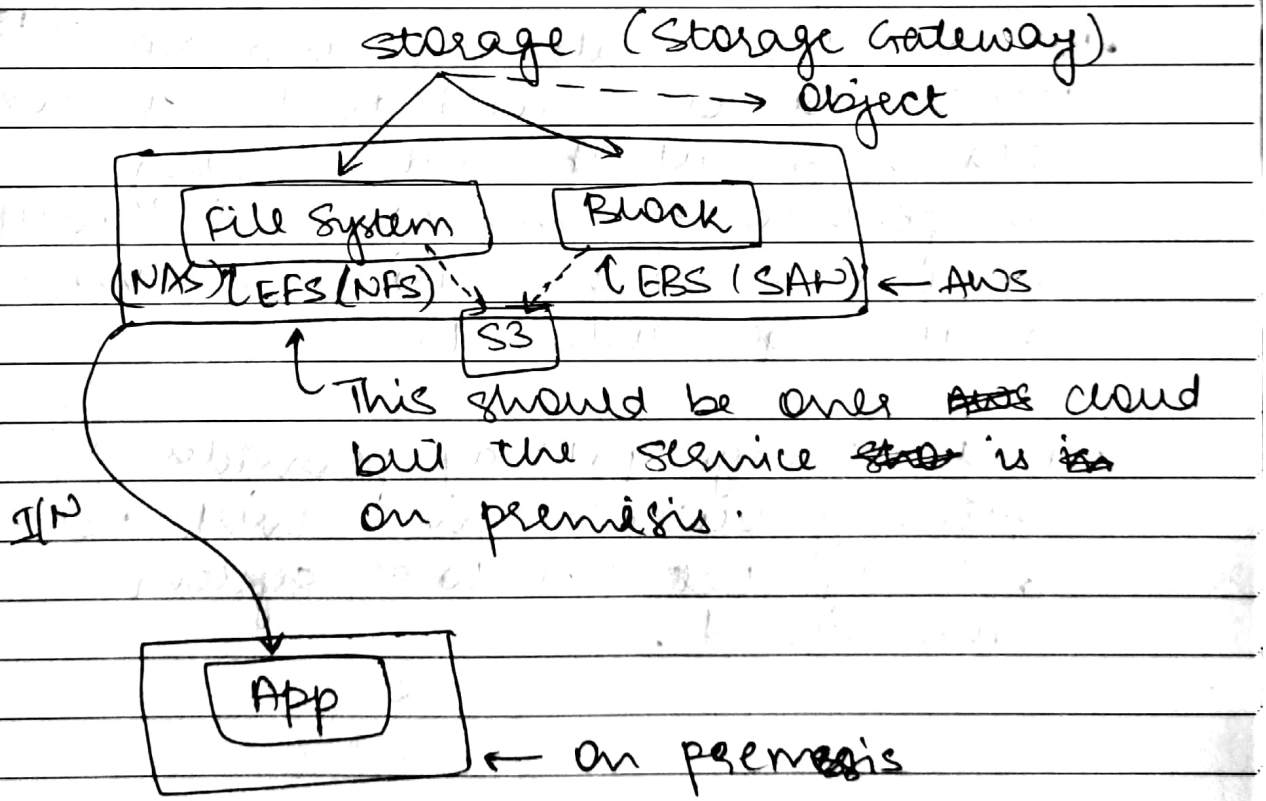
PAGE NO.:
DATE 17 Apr 21

Session: 146 AWS

Storage Gateway ; FSx ; Spot

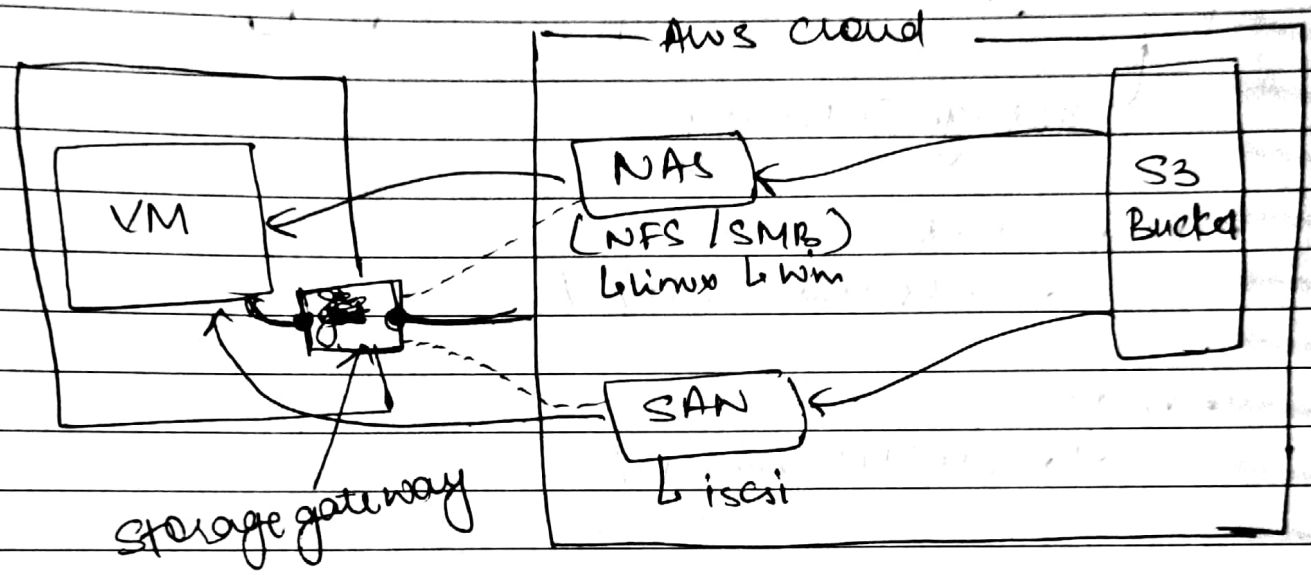


- webserver
  - /var/www/html/
- mysql
  - /var/lib/mysql/



- We can use NFS for storage but internally all the files are stored in S3

- We can also put logs to s3 through a storage gateway and with the help of Athenas we can also do some operations on it.



- We can also create our own NFS & iscsi SAN servers on aws cloud that will store data in s3 but it will be very complex for management & will be less secure bec. the files have to go through internet.

- But storage gateway provides a gateway/router which will act as a bridge bet our local system & aws cloud.

PAGE NO. :
DATE : / /

Storage Gateway
- Get Started
  - File gw
  - Host Platform: ~~Microsoft Hyper-V~~ EC2
    - Launch instance
      - m4.xlarge
      - PublicIP : enabled
      - Storage : EBS (extra) : 200GiB
      - Name: SG_ec2
      - SG: all traffic
  - End point type : Public
    (If ec2 we can also use VPC)
  - Connect to gw : <public ip of ec2>
    - gw_name : SG_NAS
    - gw_time zone : mum—
      - >> configure
  - config. local disk
    - /dev/sdb(ebs) : Cache
  - gw log : Disable logging

SG_NAS
- ~~Create volume~~
- Create file share
  - S3 bucket : lwstgw (bucket should be created in S3
  - File share name : lwstgw
  - Access obj : NFS
  - gw : SG_NAS
  - Storage class : S3 standard
    - >> create file share

PAGE NO. :
DATE : / /

SG_NAS → shared file
↳ link will provided.

local VM
<~~root~~ cmd fm SG_NAS (mount)> (with public ip) ec2
/var/www/html/
↳ This cmd will ~~exec~~ mount the storage gw ~~ssd~~ to the dir (/var/www/html/).

df -hT
↳ St is mounted; size is 8.0E
↙
{ exabyte
more than petabyte

Storage Gateway
↳ Create gw
↳ volume gw : Cached volume
↳ Host platform : ec2
↳ m4.xlarge
↳ Storage
↳ EBS : 150 GiB (for buffer)
↳ EBS : 150 GiB (for Cache)
↳ Name : SG_EC2-SAN
↳ SG : all traffic
↳ Endpoint type : Public
↳ <ip addr - pub - ec2 gw>

PAGE NO. :
DATE : / /

- Launch one more ec2 instance as a client (t2.micro)

↳ gw_name: SG_SAN
>Activate GW
↳ Local disks
↳ EBS (buffer)
↳ EBS (cache)
>configure logging
↳ Disable logging

SG_SAN → Create volume
↳ Capacity : 10 GiB
↳ iscsi target name: LW_st_target
~~Tags~~
↳ Config CHAP auth : skip

SG_SAN > Volume > volume > <link>

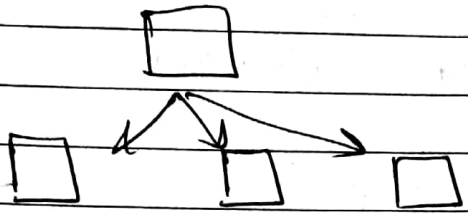
ec2-client
sudo su -
fdisk -l
↳ only one hd available
sudo yum install iscsi ——
sudo ~~vim~~ service iscsid start

[cmds avail in docs
AWS storage Gateway]

PAGE NO. :
DATE : / /

FSx → managed service

→ win DFS
→ lustre (linux cluster)



(similar to hadoop (hdfs) ~~but for win~~)

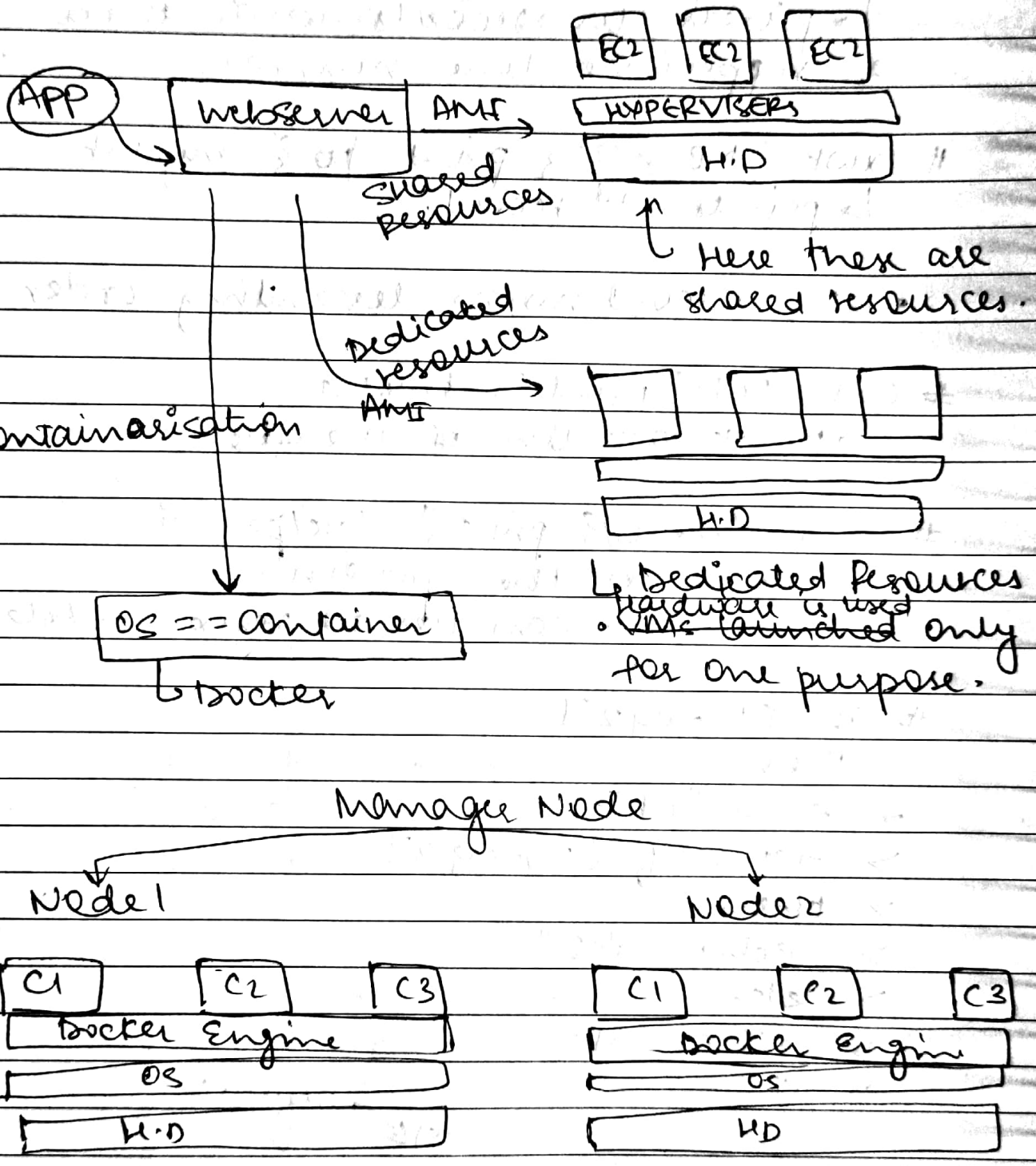
FSx

- file sys.
  - for win (win DFS)
- name : fsx1
- storage capacity : 100 GiB
- ~~win auth~~

PAGE NO. :
DATE: 25 Apr 24

SESSION: 151 AWS

ECS Cluster



- AWS creates this type of infra using cloudformation script.
- It is a fully managed service where we also have extra facilities of auto-upgradation, Scaling etc.

PAGE NO. :
DATE : / /

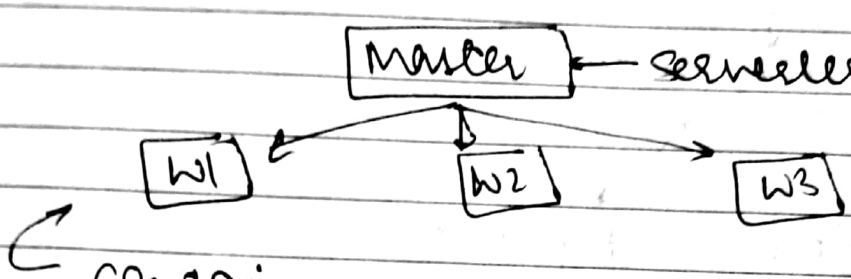
ECS
- EC2 Linux + Networking
- Config Custer
  - cluster : name
  - Provisioning Model : on-demand
  - instance type : t2.micro
  - no. of inst : 1
  - Ami : Ami2
  - VPC : default
  - subnet : 1a
    - we can create multiple subnets
  - pub ip : enabled
  - SG : ——
    - Inbound rules : 22 (port), 0.0.0.0 (ip)
  - IAM Roles : ecsInstanceRole
  - ☑ Enable container Insights

cloudwatch
- ECS instance
- ~~Edit~~ TASKS
  - Create new Task defination
    - EC2
      - configure task & container
        - name : taskecs
      - ~~Edit~~ Add container
        - name
        - img : nimal8/apache-webserver-php

PAGE NO.:
DATE: 2 May 21

Session: 157 AWS

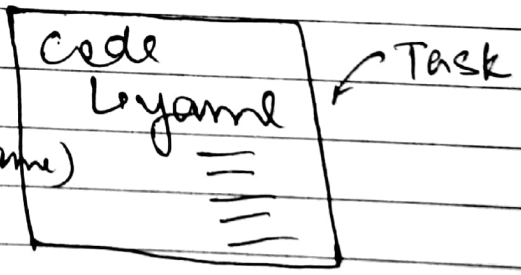
ECS : Amazon Container services



Container cluster
↳ ECS
↳ serverless
↳ Fargate

ECS
↳ cluster
↳ Networking only (Fargate)
↳ mycluster (cluster name)
↳ ☑ Create Container Insights
>>Create

>New Task
↳ Fargate
↳ Task def (Task name)
↳ Memory: 1GiB
↳ CPU: 0.5V
↳ Add Container
↳ container name: mycluweb
↳ image: cusbum img from ecs.
↳ Memory limit: 256
↳ port mapping:



PAGE NO.:
DATE: 15/May/21

Session: 167 AWS

S3 - Advance Features

S3
- Create new bucket
  - bucket name
  - ap-south-1 : az (region)
  - Block all public access
    >> Create Bucket

bucket (bucket)
- upload some files

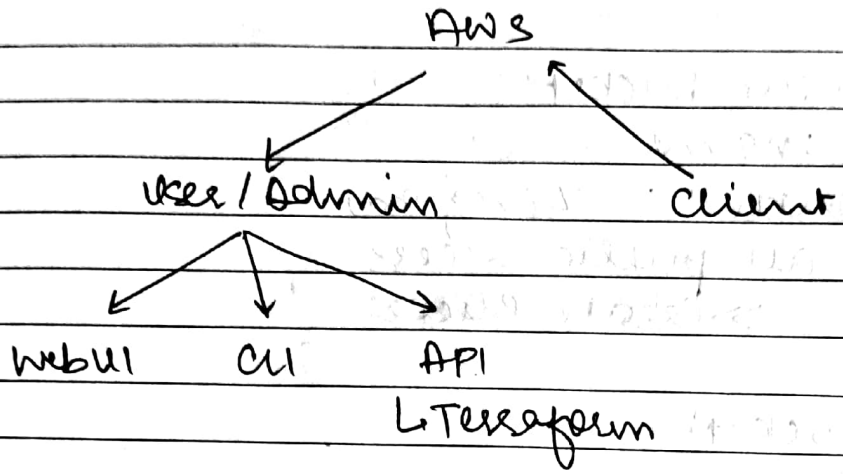
- Presigned url we can create tokens that will help to share public guys to view private files/objects

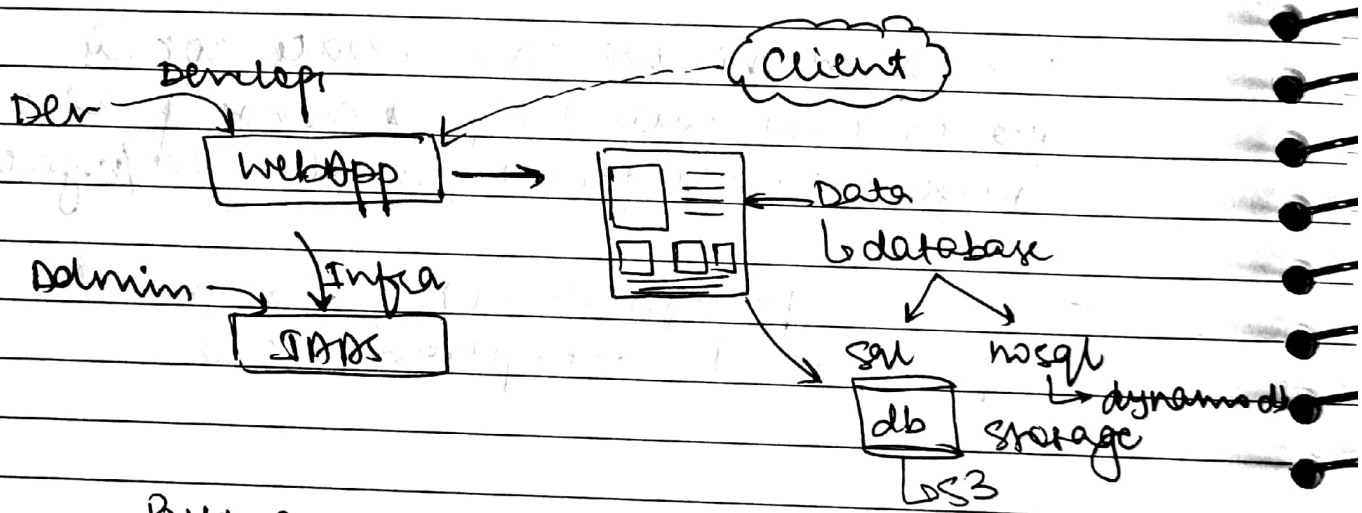
aws s3 presign <s3 uri> --region ap-south-1 --expires-in 300

PAGE NO. :
DATE : 16 May 21

Session: 168 AWS

DEVELOPER TRACK





- Python SDK (library)
  - boto3

cmd

pip install boto3

jupyter notebook

#.py

```
import boto3
s3c = boto3.resource('s3') | bname='iw123123'
s3c.create_bucket(
        Bucket=bname,
        CreateBucketConfiguration={
        'LocationConstraint': 'ap-south-1'
        })
for i in s3c.buckets.all():
    print(i.name)
        #prints all the bucket names.

bucket=s3c.bucket(bname) | bucket = s3c.bucket(bname)
bucket.put_object

data = open('nimal.jpg', 'rb')
bucket.put_object(key='nimal_photo.jpg',
                  Body=data)

for i in buckets.objects.all():
    print(i)  # prints all the objects

myobj = bucket.Object('nimal1.jpg')
myobj.upload_file('nimal.jpg')

#If we have file locally.

myobj.storage_class
    -> If nothing shows up by def.
       SC is standard.
```

for webpages

PAGE NO. :
DATE : / /

```
myobj.download_file (Key='himalt.jpg',
                     Filename = 'h.jpg')

myobj.delete()
    b deletes the file myobj

s3cl = boto3.client ('s3')
url=s3cl.generate_presigned_url(
                ClientMethod='get_object',
                Params={
                    'Bucket': bname,
                    'Key': myObj,
                },
                ExpiresIn=3600)
                HttpMethod=None)

print(url)
```