AWS CSA & DEVELOPER

Date: 19/09/2020 Day1
Date: 20/09/2020 Day2
Date: 2/010/2020 Day3

AWS CSA

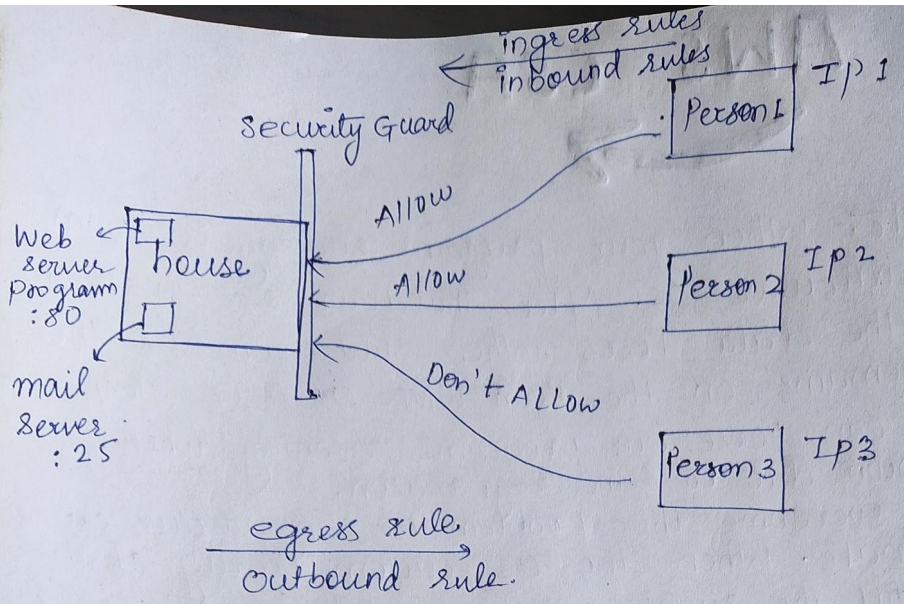
Hibernate:- When your programs running on Ram of a particular OS. After shutdown or on sleep mode the data loses when shutdown and the data remains on the Ram when OS is on sleep mode.

- So we can loose our programs running when OS shuts down due to some ~~reg~~ reason.
- So to overcome this problem we can put our OS on hibernate. When the OS automatically shuts down all the running program will be transfered from Ram to Hard disk.
- When you hibernate your OS then programs will be running and OS will shuts down.
- Region to region pricing changes.

Remote login in linux ⟶ ssh Protocol
22

Remote login in windows ⟶ RDP protocol
Remote Desktop protocol
3389

Private key, ⟶ We can launch also through Password but to maintain Authentication we use private key.



- When someone is comming to your house that is known as ingress / inbound rules.

  When you goes to someone's house then that is known as egress / outbound rules.

- Who can connect is called source. And which program you allowing you to connect then that is known as destination.

- security group never exists by its own we need need to create security group.

- If you delete all the rules in security groups then No one will connect.

- If you try to connect through putty (works on ssh) then you will get blocked.

Date :03/10/2020. #Day4

- Your system doesn't know public IP. Your system only knows private IP.

- ON AWS, All the Instances are connected each other through public IP using NATING

- Private IP never goes to Internet world.

Ps -aux ⟶ all the programs running.

- Storage types in AWS

  - object (https)
  - Block (IISCS)
  - file.

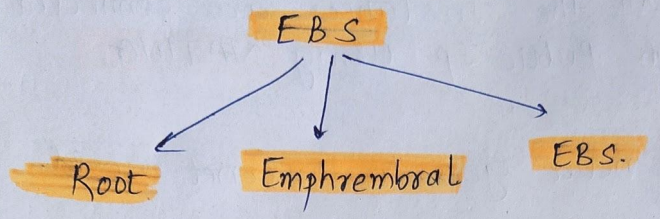
- **object storage** :- Data is permanent and there will no operating system involved. (S3)

- **Block storage** :- But when you want to install operating system and it is only possible to install OS when you have hard disk. (EBS)

With EC2 (Ami, Ram, cpu) it is impossible to install OS. Therefore EBS service comes under EC2. Every time you launch a EC2 instance the EBS comes with that.

- final requirement is too install the operating System. and EBS will help you in that.

EBS has three subtypes.

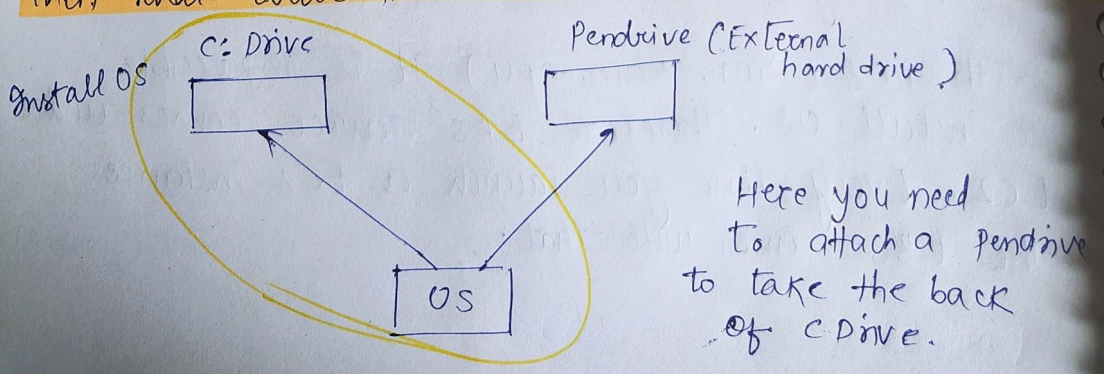


In every Hard disk you have partition. or if you want to instlau the operating system then you ~~can~~ need to create partition.

If you create three partition then you can install three OS at a time.

eg. if you have a hard disk and you have created two partition. C & Drive.
Data in C Drive will used for installing OS. where as Data in D drive will used for Backup of operating system.
By change, if OS ~~crach~~ crashes then data from C drive only goes and Back will be remain in D drive.

The data in a drive from which ~~so~~ OS installed then that drive is known root drive or root volume.



Here you need to attach a pendrive to take the back of C Drive.

Partition Concepts 👇👇
Date: 05/10/2020 Day5

- Same facility Amazon provides. you can attach the pendrive (External storage) in EC2 instance. And here in AWS that pendrive is known as EBS. Here EBS is third type of EBS.

- you can connect ulimited EBS storage to operating system.
- But attaching root drive is compulsory. root is the drive in which you have to install O.S.

fdisk ⟶ will show you no. of hard disk connected.

fdisk -l ⟶ list of all the hard disk.

mkfs.ext4 /dev/xvdf1 ⟶ create partition.

Volume (EBS) is a virtual pendrive. i.e you can't touch.

- Some services are global service like S3 where as some services are regional service (EBS).

unmont -l /myd

⟶ lazy unmount or force unmount.

First we have to unmount the volume and then detach that OS.

- Root is not same as emphrembral.

In S3, durability is 99.999999999 11 9's

Avalability is 99.999 5 9's.

Most powerful durability and Avalability in the world.

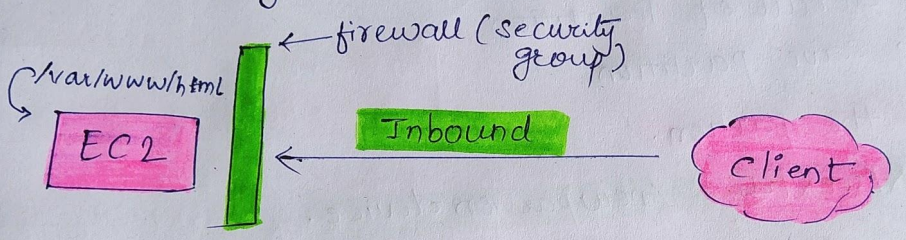
Date: 11/10/2020 Day 6 AWS CLI CONCEPTS
No Notes of that day.

Date:17/10/2020 Day 7 👇👇

CONTENT DELIVERY NETWORK

CLOUDFRONT

- cloudfront is used for CDN.
- Systemctl start httpd → will start Apache Webserver
- Sytemctl enable httpd → will enable Apache Webserver.
- Apache Web server will fetch from /var/www/html location only.



We need to set the security group with particular port number. As we are fetching webserver and Apache Web server works on port no 80. so we need to set the port 80 in security group.

- In ec2, We get root hard disk/device by default. If my ec2 linux instance get curupted then it will effect the root device (/). As we have Apache Web pages inside (/var/www/html) and it can be risky if the instance get terminated or curupted.
- 99.99% We have dynamic Website Now days. We can change the photos, data on the runtime.
- We can give seperate storage to /var/www/html. So if linux instances get terminated or curupted then our data inside /var/www/html will be safe.

- We can put /var/www/html inside EBS volume.
- ensure that we need to launch EBS in the same region in which we have launched our EC2 instance.
- Attach the volume in the instance

check the volume availability via,

```
fdisk -l
```

Create partition

```
fdisk /harddisk name/
```

n ⟶ new partition
P ⟶ detail of partition
w ⟶ save partition.

Format the partion.

```
mkfs.ext4 /new partion device.
```

mount with the /var/www/html.

```
mount /dev/xvdf1 /var/www/html
```

Now, we can check via fdisk -l

inside /var/www/html
you will find a file name lost + found.
inside lost + found file. The details of partition is stored.

- The durability of EBS is 98 to 99% and hence there is no good practice to store data inside EBS. so we need to switch S3 for storing data. Data means images and videos required for the Webpages.

The drawback of EBS is you can only retrieve the data if and only if you attach to the hardware. so, the Availability of EBS is not secure.

- EBS is not great for availability and durability

- S3 storage
  - → Object (static) storage
    - Images
    - Videos
    - Text file.

Durability of S3 is eleven 9's (99.999999999)
Avalability of S3 is five 9's (99.999%)

- Cloudfront is a service in AWS which provides content Delivery as a service.

When your clients are spreaded world-wide. and you want your client take the services provided by you in terms of website or any services. your website is hosted in one of the data center in Mumbai. Now your clients are all over the world Then they need to ~~visit~~ visit every time to mumbai data centers. To avaid this, AWS has created local data centers. In which when clients hits to the web server then they need to face latancy at first time and the cache of that website is stored in local datacenters. If any of the clients from that region hits then latency decreases. Because Now your content or data of website is stored neare clients region. so they can access easily.

So AWS, provide a service of content delivery and known as Cloudfront.

Date: 24/10/2020 Day: 8

- Akamaii is one of the company which provide CDN.
  15% to 30% of Network traffic goes from Akamaii.

The local data centers is known as edge locations
Amazon Aws have seperate edge locations.

Regional edges caches are same as edge locations
They are some ~~some~~ different.

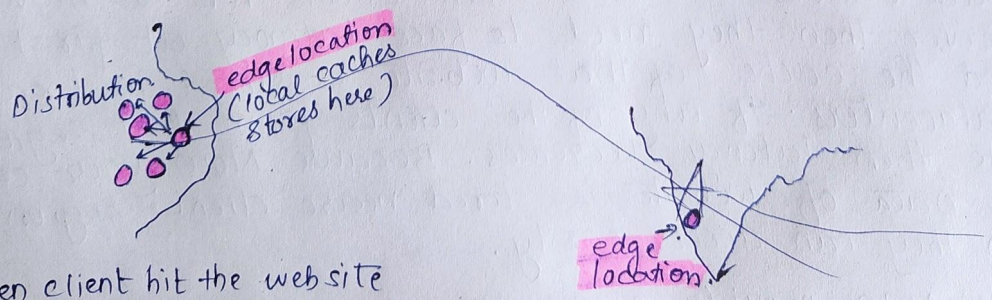
Aws has total 217 edge locations also known Point of presence (POP).

205 → edge location
12 → Regional edges cache

User/client automatically knows its nearest edge locations. This is because of DNS Route 53

Because of high speed optic fibre cables they used to transfer all the packets (Network packets)

At the first when client hit the website ~~th~~ the required data travel through global accelators (high speed optic fibres).
And Next time Cloud front will plays a vital role.



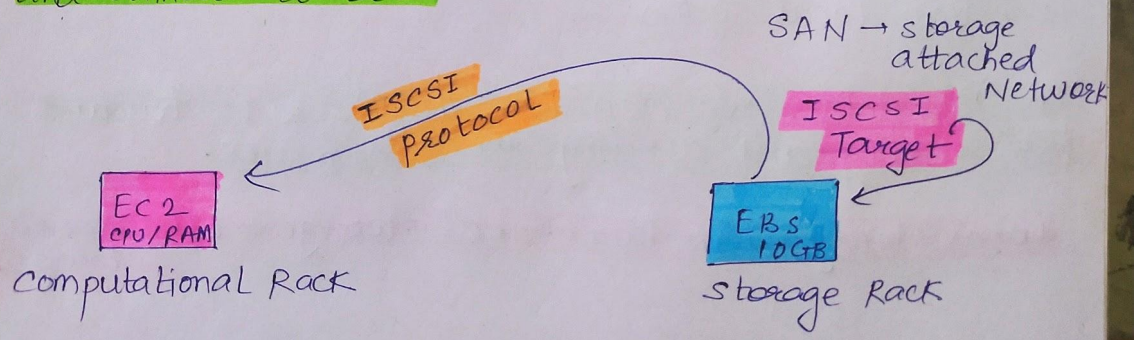
When client hit the website the data downloads in the edgelocation and distributes to their regional edge location

Date : 31/10/2020 Day 9

CLOUD TRIAL

- AWS, cloud trail is a service that enables governance, compliance, optional auditing and risk auditing of your AWS account.
- With Cloud Trail, you can log, continuously monitor and retain account activity related to actions across your AWS infrastructure.

- Without ec2 there is no meaning of EBS attachment.
- EBS is a regional service.
- EC2 Behind the scene contacts to computational Rack and give services to users/tenant.
- There are two types of Racks
  1.) Computing Rack ⇝ EC2
  2.) Storage Rack. ⇝ EBS
- Public cloud never provides the source code of the services. But we can learn the private cloud and also create our own source code of the services.
- Without creating partition, we can cannot install OS. First we need to create partition.
- If you launch OS, EBS automatically connects to EC2. If you launch EC2 in ap-south-1a then the storage Rack (EBS) automatically launches in ap-south-1a and connects to EC2.



ISCSI ⟶ Internet SCSI

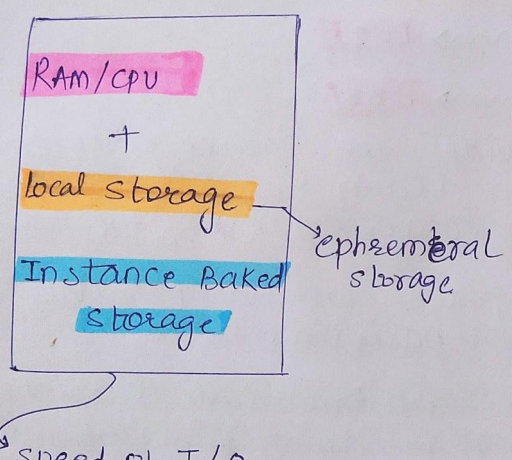
Whenever storage Rack connects to computational Rack. This can be connected using ISCSI protocol.

- We can use two types of protocols,
  1.) ISCSI
  2.) FC → Fibre channel

- Local hard disk → your compute services coming from the system and your hard disk comes from same system.

- SAN → Storage Attached Network.
  Computing services comes from the same system and harddisk (storge) comes from different system.

- Rack 1



In Big data, We can use the Rack 1 structure for reading and writing the data faster.

Local storage = InstanceBaked storage = Ephemeral storage

Types of Blockstorage

- Root
- Persistent Permanent (PD)
- Empheremral (Temporary)

ISCSI
Networking

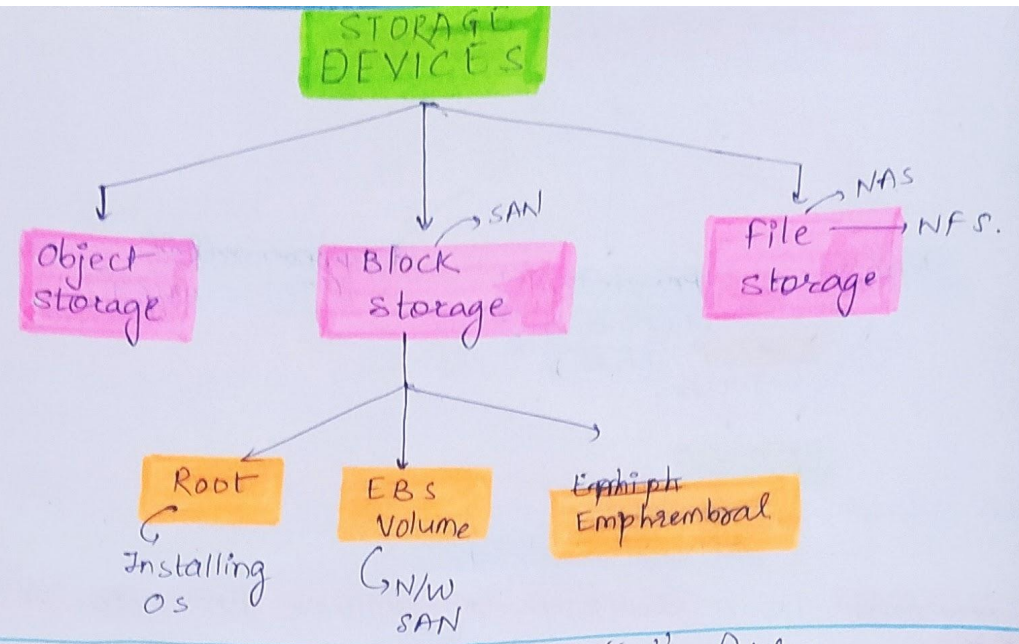
EBS

SNAPSHOTS

- snapshot is a backup of volume connected with EC2.
- We have to create a volume ~~of~~ and make partition ~~of~~ inside that volume
- And we can create a snapshot of that volume.
- If you want to update the file inside the volume then you can again create a volume ~~through~~.
- But if you want to ~~back~~ read the deleted files then you can create a volume of your snapshots.

HAPPY DIWALI ✌✌💣💣

Date: 28/11/2020
Day: 9



- All the software install in "/" - folder. If "/" drive fails or crashes then all the software will crash. so to tackle this problem.

- If you can increase the size of the volume without loosing the data its known as Vertical scaling. We can change dynamically the storage size and Performance.

- you can transfer the Whole hard disk to ~~or~~ other region or to other Availability zone.

> launch OS
> make the configuration in that OS.
> ~~Create~~ attach the Volume.
> Create the snapshot and make the partition
> Create the Volume in different region.
> you can copy that snapshot to different Availability zone and there you can create Volume and attach to your respective OS.

Creating Ami's.

> launch os

> make the configuration.

> create image by going into actions and create image.

> you can give permissions to your custom Am's.

Date: 29/11/2020
Day: 10

Creating Ami's.

> launch os
> make the configuration.
> create image by going into actions and create image.
> you can give permissions to your custom Am's.

EFS
ELASTIC FILE SYSTEM

- EBS, will give you Block storage. But the only challenge is that you can attach ebs volume to one instance only.
- If you had a requirement, that you need to provide the same data in more than one instance. Then EBS won't help you.
- S3 is a global service. Here you can use S3 to share the data into multiple instances. But there are some challenges.
- like

When you have code inside S3, and you can share to multiple instances. But if you had requirement that you need to run that code. then you cannot run in S3.

either you can copy that code to local hard disk.

[like when you have code in google drive and you want to run that code inside windows.
that can only possible due to after ~~running that~~ downloading that code to windows system.]

If you want to run the code then code must be part ~~lab~~ of inode table.

S3 never share the inode table. But NFS protocol share the inode table.

~~S3~~ NAS works on NFS protocol. NAS is a concept

EFS is the service provided by the AWS that is based on NFS concept.

EFS → NFS as a service

We can mount the storage to multiple instances at a time. The user can feel that ~~give fee~~ this is user's storage.

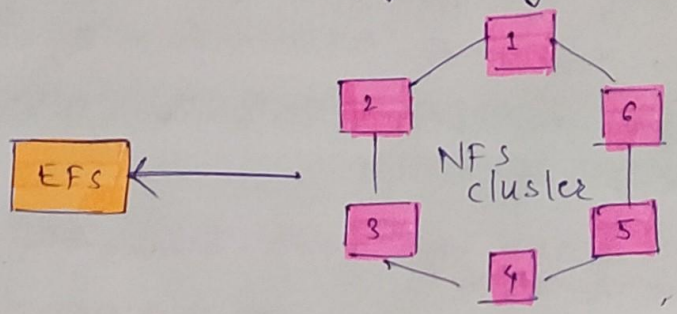
If you share the 10 GB file from S3 to different system. They need to download the file to read the file. For reading the 10 lines of that 10 GB file they need to download first.

But when you share the file of 10 GB then user's can read that file without downloading.

EFS is a serverless service.

Behind the EFS. AWS creates NFS cluster. eg Pacemaker

[In that cluster, client need to mount the nfs. therefore in EFS, you see, the mount targets. Internally, if you create EFS in AWS then AWS behind the scene launches 6 NFS servers. If any of the server's from that 6 NFS servers goes down then the ~~data co~~ other five manages the EFS. you can use any of the IP of the cluster to connect. ensure that you had configured your security groups.]



mkdir /data ← create a folder.

mount 172.31.86.200:/ /mydata

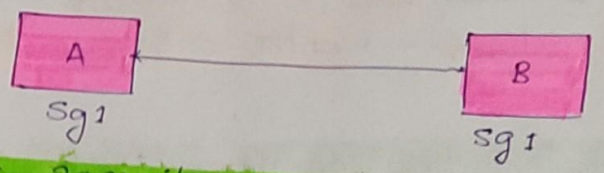
IP of nfs server.

storage of one nfs server.

Type	protocol	port range	source
ALL Traffic	ALL	ALL	sg-dc5b57e0 (default)

Rather than Allowing particular IP we will allow ~~sg~~ by security group name.

The guy who will have this security group can only connect.



If the security group is same in both the system then they can connect to each other.

Inside the nfs server might have their own server rule.

The size of one nfs server is 8E (exabite)

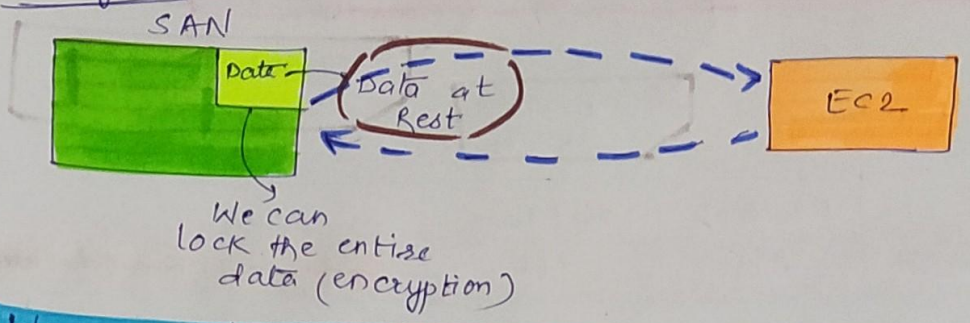
check selinux rule of a particular file.

ls -l -z

We can lock the entire data in EBS volume.

The data inside EBS known as data at rest. at resting only we can can encrypt that EBS.

Storage term



We need to use key. AWS gives you a key pre-created known as KMS.

Go into Volume, → ~~mod~~ create a volume → then encrypt the volume → enter a key → default key (aws/ebs).

if you have created the Volume already and want to encrypt that volume. then you need to create a snapshot first and then create Volume from that snapshot.

And when you already encrypted the Volume and create the snapshot then when you will create the Volume from the snapshot then the Volume will be automatically encrypted.

EFS is a regional service.

Date: 5/12/2020

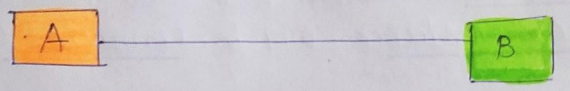
Day: 11

VPC

VIRTUAL PRIVATE NETWORK

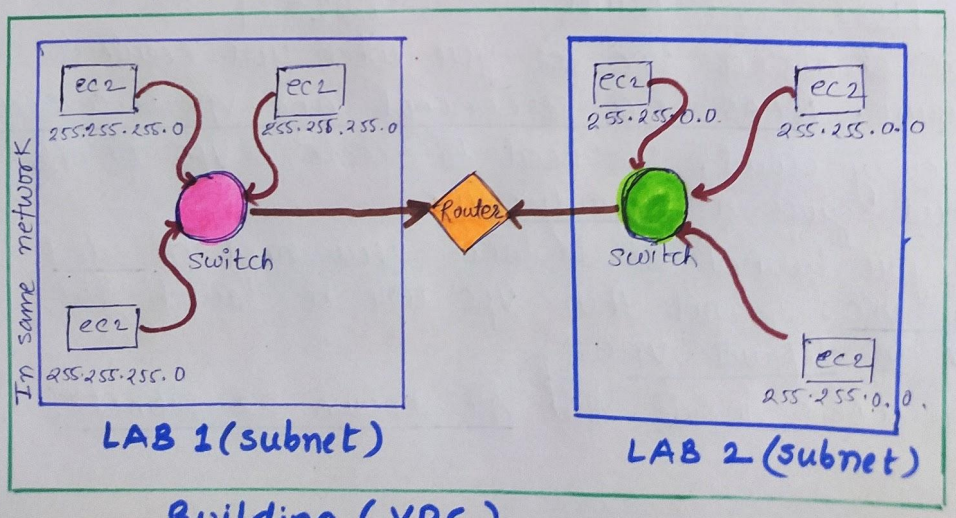
- vpc provides you a multi tenancy service
- vpc provides you a NAAS (Network As a service).

Requirements



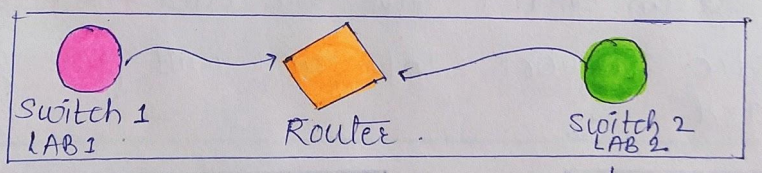
1. A and B system must have physical connectivity
2. A and B system must have IP.
3. If you ~~we~~ have same network then there we have switch.
4. If you have different network then we need Router.
5. Private network —Ping→ private network.
   Public Network —Ping→ public network.
   Private Network —not ping→ Public network.
6. A and B system must have nic card (Network card)

ENI → elastic network Interface will give you network card.



- As, you can see in previous Network diagram. We have LAB 1 and LAB 2. In LAB 1 we have three ec2 instances of same network. As they have same Netmask i.e (255.255.255.0). so Ip will be in same network.
- But to connect ec2 instance to the internet world we need to accumulate all the ec2 instance with a device known as switch. switch will also have its own Ip. so every ec2 instance will be connected to switch.
- In the LAB2, the same scenario we can see. the instances can connect to each other in the same LAB only.
- But If the instances of LAB1 wanted to ping/connect to the instances of LAB 2 then Router will help to do that.
- Router is meant for having connection to different Network.

The Ip of both the switches connected to Routers



- Aws create a vpc for you when you create a account. NAAS is a concept and vpc is a service. In every cloud, We need to create a vpc or the cloud gives a default vpc.
- If you launch an instance you need to select the VPC. If not then vpc will be launch that ec2 in default vpc.
- The LABS inside the vpc known as subnets.

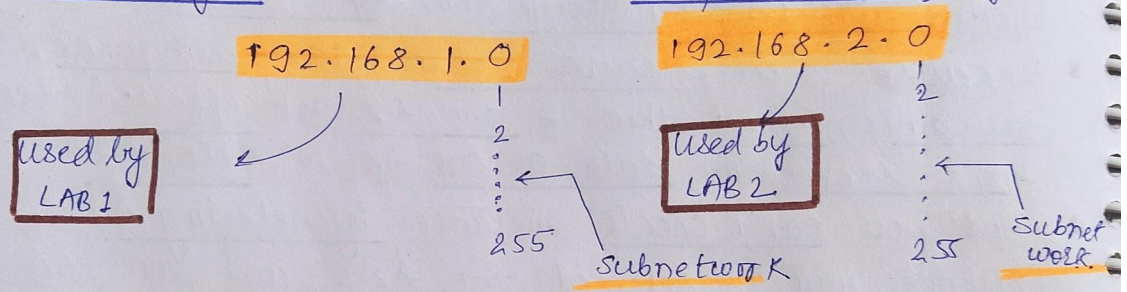
- In every, region there is a default building (vpc), and every building (vpc) has its own LABS (subnets).
- vpc not only gives you NAAS. They provide you entire infrastructure.
- We can create our own customize vpc and subnets.
- When you will create subnet, then you need to mention Which Availability zone (AZ) you want to use.
- AZ will give you Ram and cpu to the ec2 instance.
- In every region you have multiple AZ, eg. ap-south-1a, ap-south-1b, ap-south-1c.
- Because when disaster occurs like earthquake, tsunamic, flood then if one AZ got effected then AWS stores the data of one AZ to other AZ's.
- But you can't create instance by selecting AZ's. subnet will be selected the AZ's and you can launch the instance in that subnet.
- The connectivity between switch and ec2 instances in a lab done automatically. Cloud generates switch automatically behind the scene.
- Normally, We have switches with 24 no. of ports. But in AWS. ports created dynamically. Behind the scence ports created from some software in AWS.
- There is two way ~~way~~ to give the Ip address.
  1.) static → manually go inside os and give fixed Ip.
  2.) Dynamic → Dynamically IP allocated by DHCP.
- Whenever you launch ec2 instance, behind the scene AWS already created DHCP server for you. and DHCP is connected to switches directly.

- Aws don't know how many instances you want to launch or which range of Ip you want to give to your respective instance.

- In the networking world,
  DHCP allocates Ip, you need to give range.
  We use netmask to know how many hosts we have to give Ip.

  total Ip's = 256 i.e 0 to 255.
  $\hookrightarrow 256 = 2^8$.

Per LAB you ~~have~~ Aws create DHCP, switch for you.



∴ 192.168.1.0/24
↳ each lab you need to specify the range of Ip and blocks.

If you want to count all the Ip then it will be $2^{16} = 65536$.

∴ you need to give network name of the vpc.

192.168.0.0/16
↳ 16 means all the count of Ip's $2^{16} = 65536$.

Or with the netmass.

255.255.0.0/24.

- If the vpc has the range of 192.168.0.0/16 then the subnet range must consits between 192.168.0.0/ range.
- Network name is also named as cidr block.
- After creating vpc you need to create your own LABs (subnets).
- What you will be doing inside the LABs (subnets) the services will be taken from respective AZ
- VPC gives you isolated environment.
- DHCP allocates the range of IP automatically.

Date: 06/12/2020
Day: 12

- If the vpc has the range of 192.168.0.0/16 then the subnet range must consists between 192.168.0.0/ range.
- Network name is also named as cidr block.
- After creating vpc you need to create your own LABS (subnets).
- What you will be doing inside the LABS (subnets) the services will be taken from respective AZ

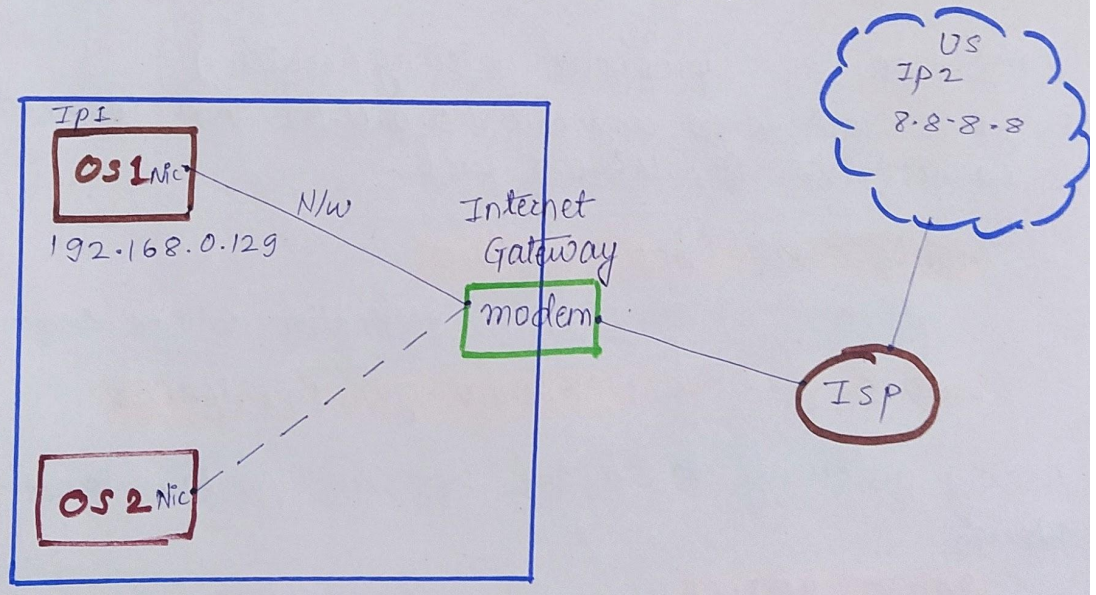
VPC gives you isolated enviornment.

DHCP allocates the range of Ip automatically.

ISP → Internet service provider.

Isp will provide the Internet to all the local systems.

See the architecture below.



- As you have different Network name and google has different network name. so to connect different network we need Router (modem).
- In mobile phone, when you enable the Hotspot the mobile phone automatically becomes Router.
- If you want to take your system to internet ~~wa~~ then there is a way to gate known as ~~as~~ gateway.
- Gateway (Router) has its own ip address.
- So if you want your system goes to internet world you need to go first to gateway.

route -n
↳ gateway description.

Destination	Gateway
0.0.0.0	192.168.0.1
any where	ip of your gateway (router)

If you want to go anywhere (0.0.0.0) then you need to connect to 192.168.0.1 first.

route del -net 0.0.0.0
↳ Anywhere destination will be removed.

route add -net 0.0.0.0 gw 192.168.0.1

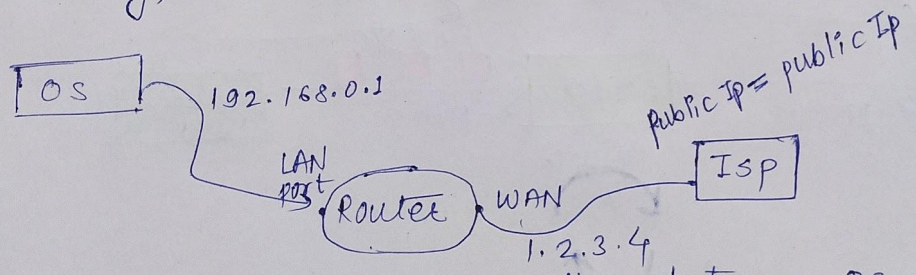
Adding gateway ~~to~~ so the we can go to internet world.

route PRINT
↳ routing table of windows.

Ip's

Public | Private

- private Ip comes free. But you need to pay for public Ip.
- using the private Ip. you can't go to internet. But using public Ip you can.
- **Range of private Ip's** :- 10.0.... , 172.17.... , 192.168.... } private Ip
- The rule of Networking ~~com~~ made by the some community and your respective Isp follows these rules.
- Where from you sending the packets known as source Ip and where you want to go known as Destination Ip.
- Isp not permit you to go outside and connect to internet. your private Ip will contact to Isp and Isp will ask to private Ip that from where are you coming and where you want to go. Isp will not permit to go free Ip (private Ip) to internet world. so Isp drops the ~~Int.~~ private Ip's.
- Normally, Router comes with two Network card.



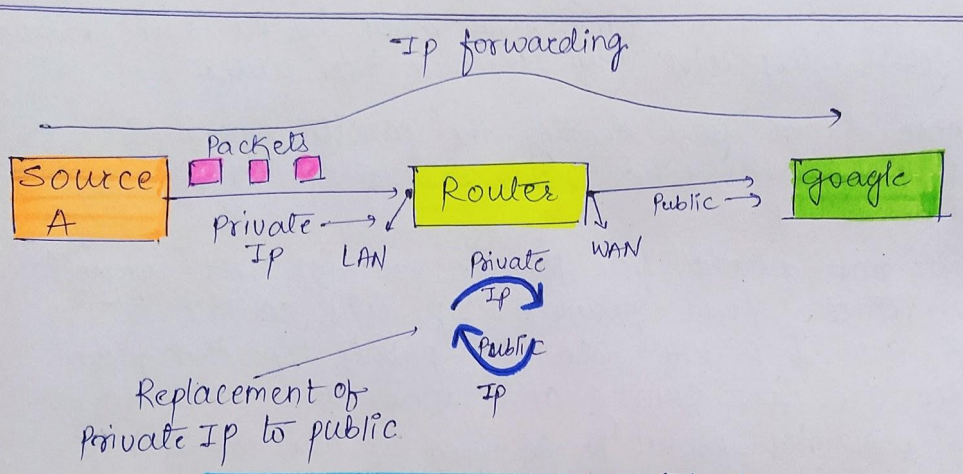
192.168.0.1 ⟶ connection between os and router

1.2.3.4 ⟶ connection between Router and Isp.

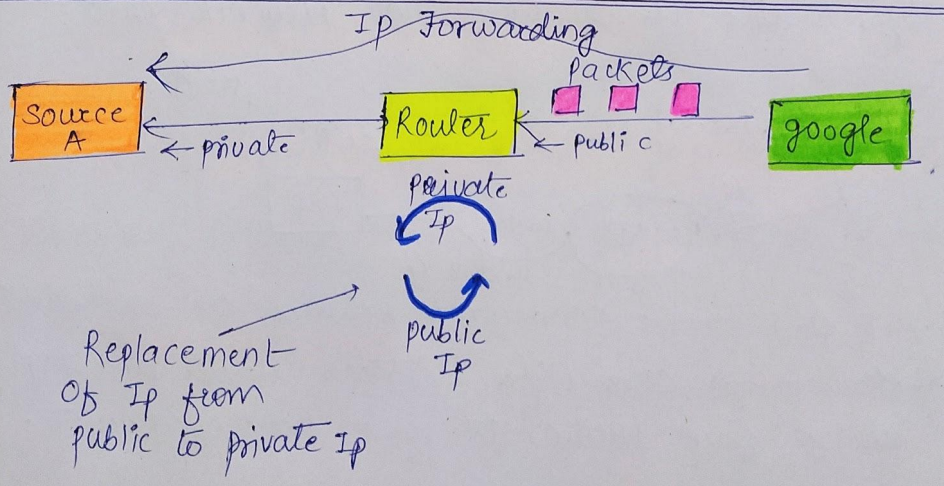
Isp always give public to right port of Router.

Private Network $\xrightarrow{\text{NATING}}$ Public Network.

- On the left port of Router have private Ip connected to OS and on the right port connected to Isp.
- Router when recieve the packets from the OS. it replaces the OS. Ip with its own public Ip. Now the replaced Ip from the router goes to internet.
- Note :- your OS Ip Never goes to the public Internet world.



Source NAT Architecture · SNAT



Destination NAT Architecture DNAT.

- Private IP gives you advanced security. The Operating system which ~~have pri~~ is not connected to the network that OS is the most secured OS in the world.
- By default, SNAT is enabled in your system.
- Destination NAT, DNAT is the reverse of SNAT.

DNAT is used for ~~ti~~ port forwarding.

Internet gateway is equal to DNATING.
DNating is a concept. Internet Gateway is a service provided by Aws.

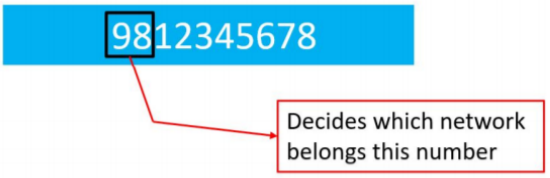
SNating is a concept. Nat gateway is a service provided by Aws.

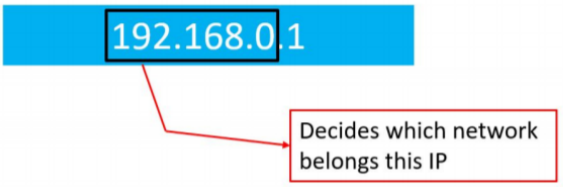
When you done SNating, router remember the session and ~~the~~ Dnating enable by default.

Requirements for connectivity between OS

- Both the system must have operating system
- Both the side must have network card (NIC card)
- Both the network card must have IP address IPv4
- IP of one OS must be in the same network of another OS.
- Physical Connectivity between one OS and another OS

How Network is decided to IP

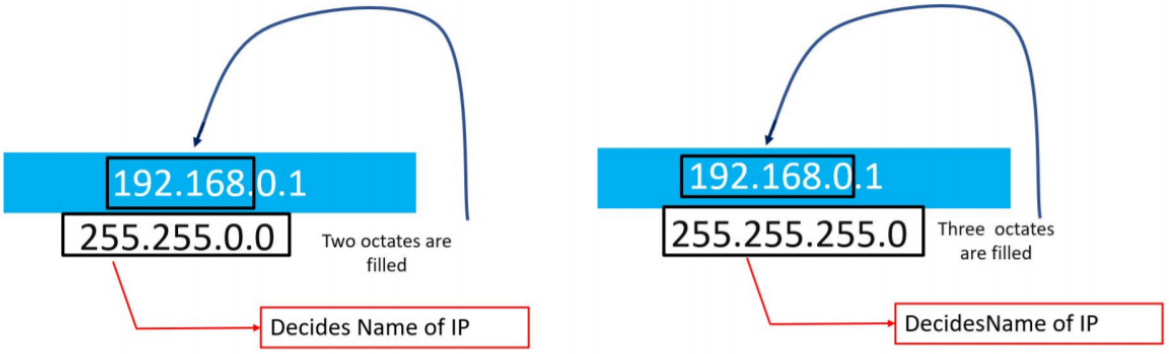




How name is given to IP

- IP name is decided by netmask range.

- If your netmask is of range 255.255.0.0 then only two octates of IP will be name of IP
  If your netmask is of range 255.255.255.0 then only third octates of IP will be name of IP



Nating Concepts

- **DHCP**(Dynamic Host Configuration Protocol Server) is the one who gives
  - IP address
  - Netmask address
  - Gateway address

- A system with no network connected is known as most secured system in the world

  Private IP —connects→ Private IP

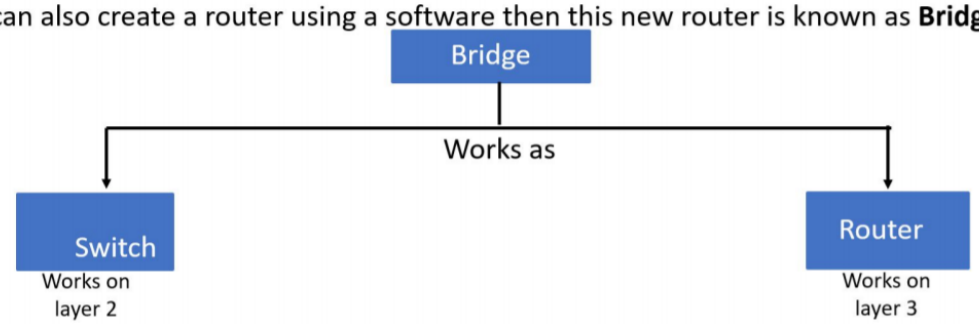
  Public IP —connects→ Public IP

- If you want to connect private IP to public IP then this can be possible via Nating

  Public IP —Nating→ Public IP

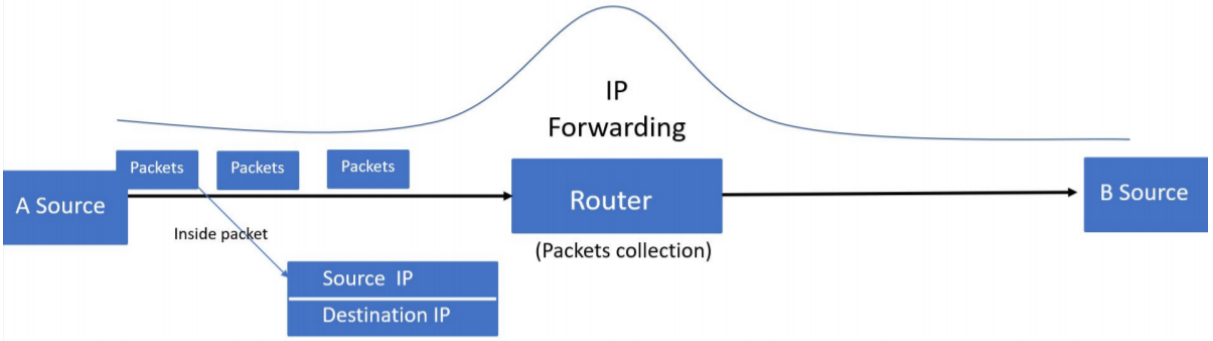
Bridge Network

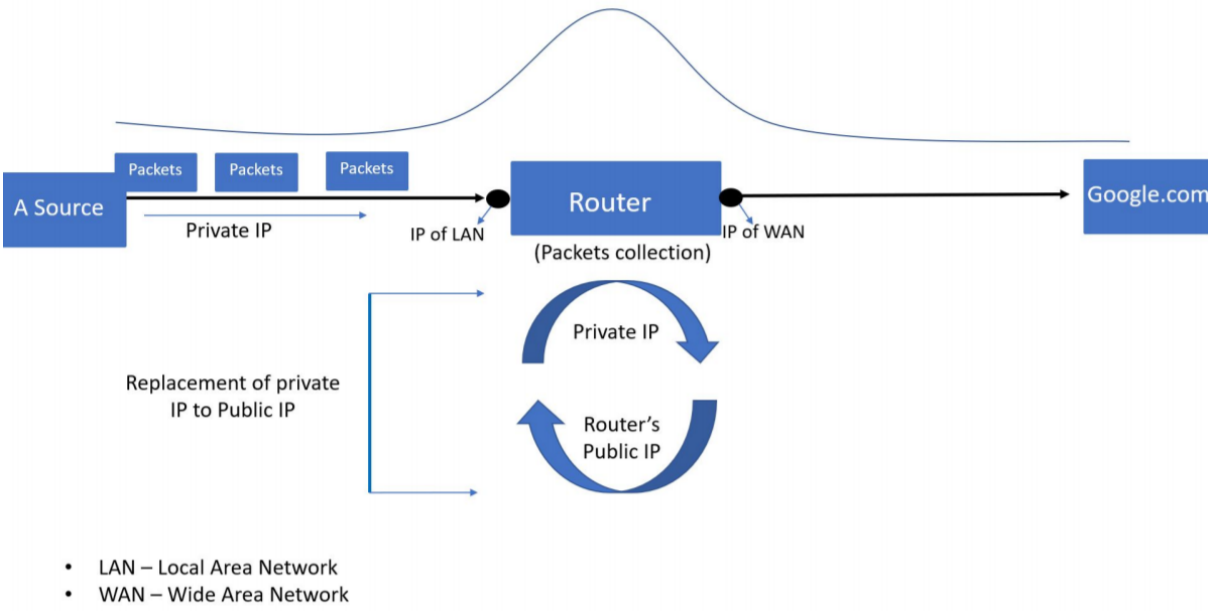
- All the **ISP** (Internet Service Provider) in the world internally connected to router.
- **Router** is the Device that will connect IP's of different network to the internet.
- **Switch** is a device that will connect the IP's in same network.
- Router rule's says if any private IP comes to router it won't allow that connects to internet.
- We can also create a router using a software then this new router is known as **Bridge**.



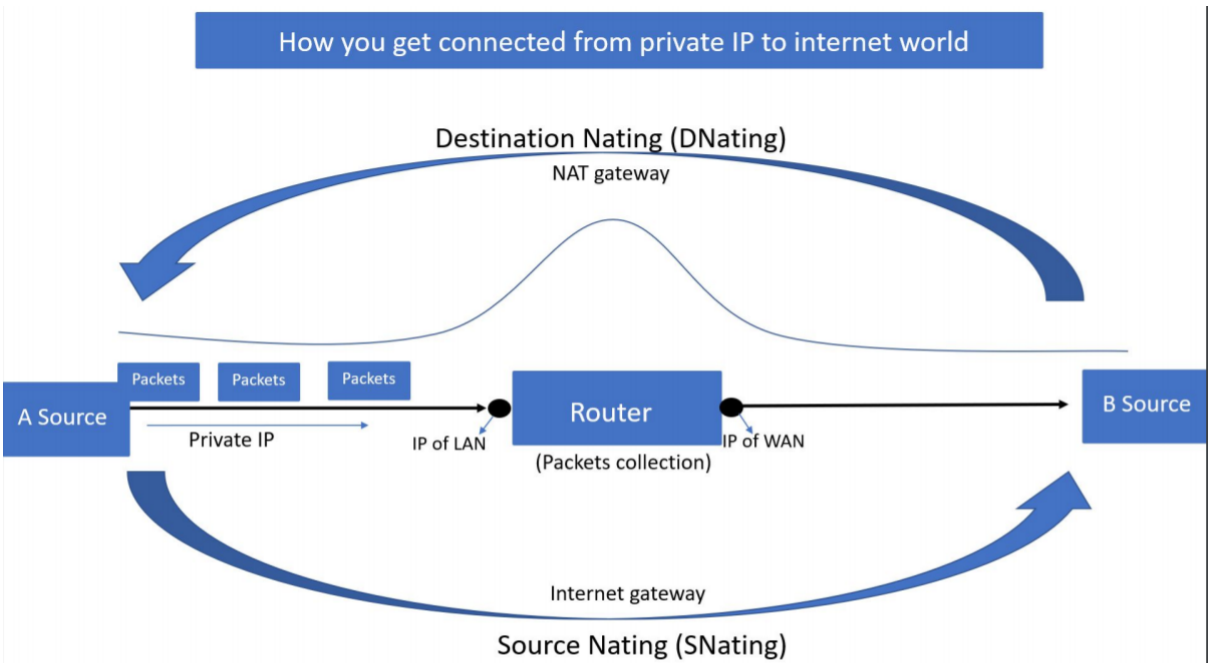
IP Forwarding

- Routers duty is to collects network packets and send it to destination





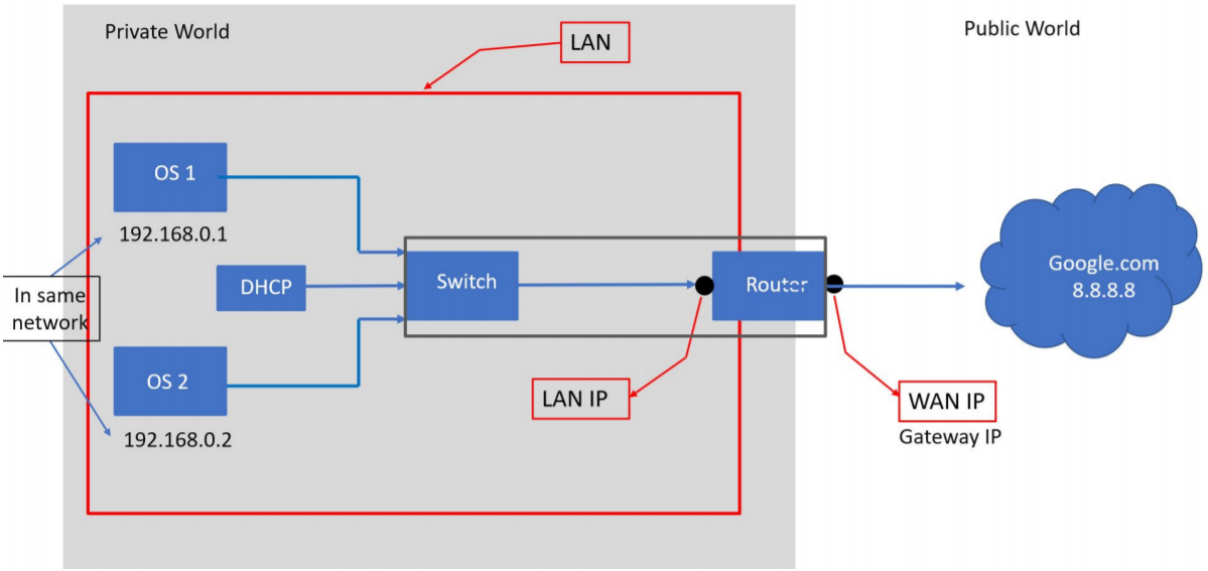
- LAN – Local Area Network
- WAN – Wide Area Network



Types of Nating

- If you want to connect from A source to destination B then **source nating** must be enabled.
- If you want to connect from B source to destination A, then **destination nating** must be enabled.
- **Source Nating** is by default is enabled in your system whereas **Destination Nating** in not enabled in your system for security reasons.
- **D Nating** is also known as **PAT** (port address translation)
- **S Nating** provide a program name as Internet gateway and **D Nating** provides a program known as NAT gateway.
- All your ec2 instances in AWS are running on LAN and connected to our laptop. If you want to login inside ec2 then you logged in successfully because whenever we launch ec2, AWS provide one router and in that router **D Nating** is already enabled.

NETWORKING INSIDE CONTAINERS



- LAN is collection of systems of same network. In this system, all the IP are in the same network. All same network IP is connected to switch and switch is connected to router as you can see in the figure.

- When new IP is allocated by DHCP, if it is not in the same network then it get connected to router directly.

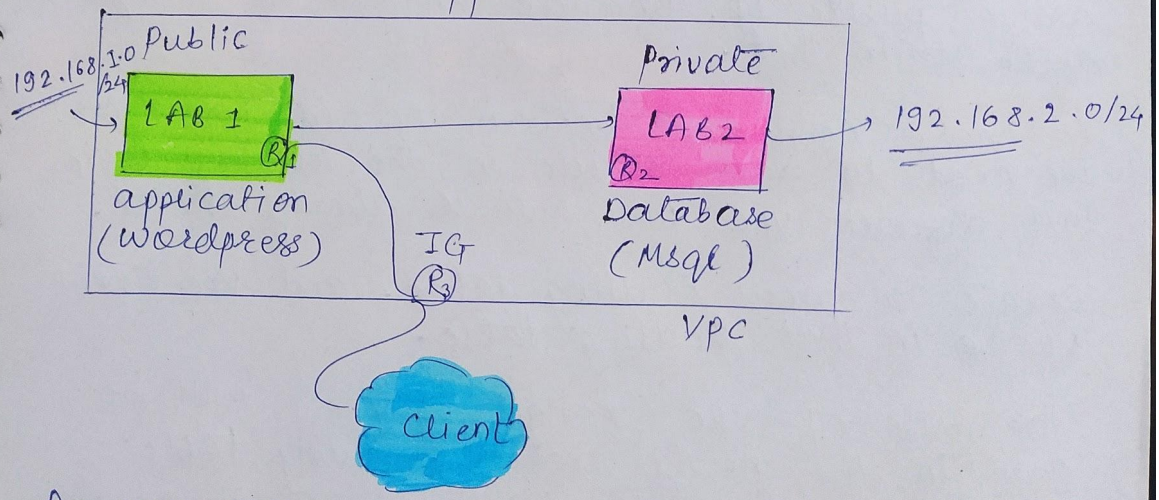
- Router collects all the network packets and sends to destination of the user. In router replacement of private IP with routers IP take place. Therefore, Users able to locate connect to destination that is to public internet world.

- The IP of each OS is isolated. To launch a OS in same network we must launch OS in same network so we can connect to switch. Docker have the capability to launch the OS in the same network.

Date: 12/12/2020
Day: 14

As you can see in the figure above, there are two subnets inside a VPC.
The two Respective Routers R_1 and R_2 are associated to respective LAB 1 and LAB2. R_1 and R_2 Routers are there for connection between the Labs (Subnets). Whereas the LAB 1 Router R_1 is connected R_3 (IG) Internet gateway. So R_3 will take LAB 1 out to public world. This lab (subnet) is known as public subnet. Whereas the LAB 2's ~~is~~ Router ~~R2~~ R2 is not connected to R3 (IG). Therefore, R3's LAB i.e LAB 2 is known as private subnet.

LAB 2 i.e. Subnet 2 is the most secure LAB. Because it's not connected to internet world. So We can store the Data or we can use LAB 2 can be used as a Database. It will be hidden to the public world. Whereas we can launch our application in the public subnet (LAB 1) so everyone can access the application.



As you can see the Range of LAB 1 and LAB 2 are different. So we have two Router to connect these two LABS.

- Internet gateway is comes in default vpc when you create account.
- If you want to create a new vpc. then you need to create net Internet gateway.
- After creating IG, attach to the newly created vpc.
- In one vpc you can attach one IG only. The IG will you have created is managed by Aws internally. so if the newly IG created fails then Aws created a new for you.
- Auto Assign public Ip will enable the DNAT rule in the IG.
- OS always have private Ip only. the public Ip you assign is the IG public Ip.
- On launching OS, in LAB 2 disable the auto assign public Ip. Because we will be having databases in that.

To make connection between IG and LAB 1 OS. we need to add a rule in the routing table. But defamet vpc these rule is already created.

Create a new routing table. and vpc (new vpc) in that routing table.

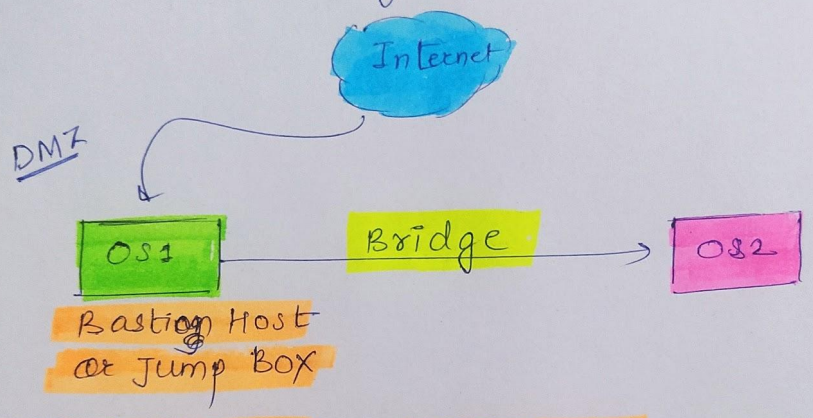
The rule of vpc (range of vpc) will be given to the newly created routing table. so, add a route in the newly create routing table.

Destination	Target
0.0.0.0/0	Internet Gateway.

After create the new route. We need to associate with the Lab 1 (subnet 1). so edit the Route table association.

Destination	Target
192.168.0.0/0	local
0.0.0.0/0	new router

Now you can ping to the public world.
But we cannot able to OS2 launched in Lab2.
so we need to go via OS1.



Bastion host or Jump box is used to configure the private subnet or we can say Bastion host provides you a Bridge to go to private subnet

---

Date: 13/12/2020
Day: 15

Nic card.

ifconfig enpos3 : 1 1.2.3.5

↳ creating virtual network card.

The mac address of virtual network card is same with original network card.

ssh support .pem key to login into remote instance.

the key should be in executable mode

chmod 400 keyname.pem

The instance launched in private subnet wouldn't have internet connection. so in that instance you not able to download any software. Because the private subnet don't connected to internet gateway.

so we need to create a router (Nat gateway) and connect to private subnet.

elastic IP ⟶ public IP

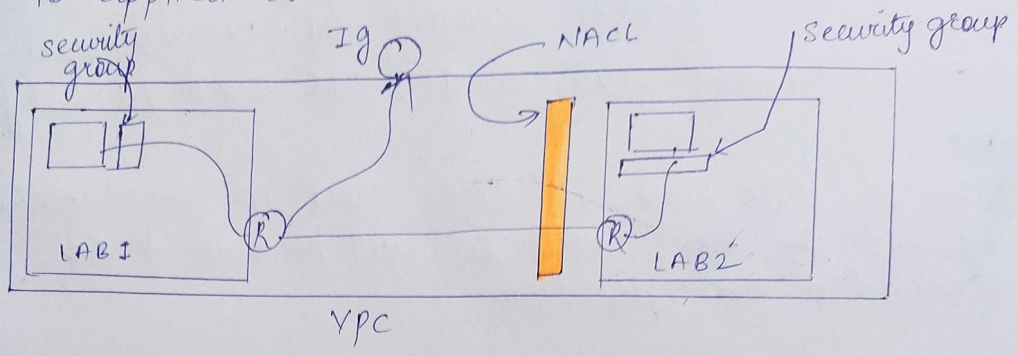
Both are same.

If one router wants to go to next Router then its known as Next Hop.

Wizard is a service which is used to launch entire vpc.

~~Neck~~ NACL is same like a firewall.

security group is applied per os. where as NACL is applied over all the Network.



By default the NACL is create for both the LAB. so if you create a new NAC the previous NACL remove.

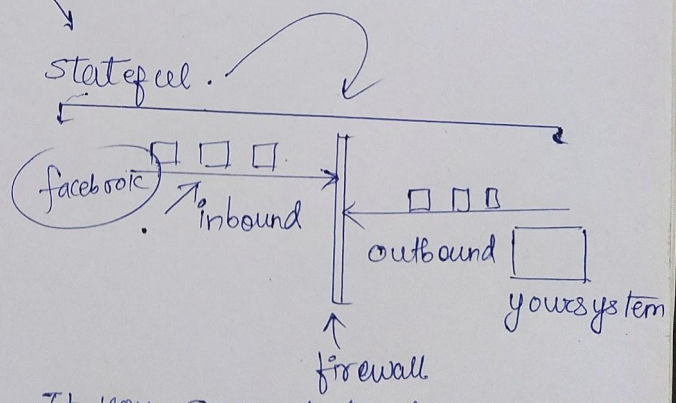
By default in security group when you filter the rule then that Rule will be allowing that protocol. where as in NACL you can Allow/Deny a protocol after filtering also.

firewall

stateless

In bound and outbound you have to build.

stateful.



If you connect to facebook the fire wall will allow the out bound. Then for inbound, then dynamically create a new rule to block All.

In NACL, the Rule has priority.

Rule	Type	protocol	Port Range	Source	Allow/Deny
10	All Icmp-IPv4	Icmp(1)	All	1.2.3.4/32	Allow
25	All Icmp-IPv4	Icmp(1)	ALL	0.0.0.0/0	Deny

Rule 10 will have to take first in consideration and the next rule 25 will have to take second in consideration.

Date: 26/12/2020
Day: 16

ELASTIC LOAD BALANCER (ELB)

- The whole business running on ~~an~~ particular instance so, We need to secure the instances and we have to give better uptime of that instances.
- So, We need to setup or Balance the instances. If your Website connected with many users. So there we need to balance the load of the application The We need to balance the ec2 instances running behind the scene.
- Load balancer will balance these requests and We ~~hav~~ can get disaster security of ec2.
- Like all the ec2 instances (target nodes) will be launched in different subnets. and AWS make a replica of the instance launched in a particular subnet. So if the one subnet collapse then ~~ed~~ Load balancer will ~~launche~~ connect the user's request to other target nodes (ec2-instances)
- By shifting the user's requests from one ec2 instance to another ec2 instance then users might face down time in seeing the application deployed in that instances or end points running.
- We have different types of load balancers from Aws. Depends on the requirements and their use cases We can use the respective services of Aws.
- Load balancer is a ~~servi~~ concept which will take all the instances Ip's and loadbalancer will give its own single Ip or domain name to the users ~~or~~ clients. So client/users don't have to go to see all the instances or Application deployed in the ~~sun~~ instances running behind. They will take the Ip or domain name of the Load balancer and they can see the Application deployed in the instances.

- users by these don't have to remember the Ip's of all the instances or backend servers.
- so, using the Loadbalancer's Ip / domain name only they can ~~can~~ view the application.
- But if your users are increasing day by day then load balancer will launches a seperate ~~ccc~~ instances behind the scene and that instance will now register into the load balancer and now users can easily see the Website or application deployed in the Backend servers.
- One of the common stratergies We use is round robin.
- Load balancer uses ~~Hap~~ Reverse proxy to get the applications contents to display on the user side.
- load balancer's Ip ~~ch~~ keep on checking the status of Backend servers. ~~They~~ They keep on checking the health of application running in the backend server. If any of the server is dead the load balancer deregister the dead servers. And if any servers got connected to the ~~s~~ load balancer then it registers to the load balancer.
- If your instances ~~be~~ goes down then it should be seamless. means your user shouldn't any trouble in seeing the applications.
- ELB (elastic load Balancer) ~~will~~ is the service provided by AWS which gives you facility of load balancer, haproxy, health check, etc.
- ELB also provides you the facility of cross zone means different instances launched in different subnets.

- ELB should be made on public subnet so users can get the Ip or domain name of that Load balancer.
- ELB first takes the requests from client and then it recreates that request and send to the backend servers. and using reverse proxy elb gives the results on the user side.
- We will give the private Ip of instances to the elb. There is no use of giving public Ip as user will not directly connects to the backend servers. It will be connecting to the elb only.
- We can configure the security group. We will be using least privileges. i.e will give the only port or Ip in the security group configuration of each instance. This will makes the backend servers highly secured.
- launch your instances with the help of your configured custom AMI. so we will discard the ssh port Because the ec2 is already configured. So we will not give the key and this make the ec2 instances will be highly secured.
- If your application is healthy then the success code is 200. Whereas if your application corrected then the success code will be 404 or else.
- so elb checks the healths of application runing in your instances. ~~if the ap~~ They check the health through success code of the Application.
- To check the health, We need to provide the page name in the URL or with Ip so ELB will ensure which backend server is active or which one is dead.

- Aws will gives you three types of ELB

ELB

- CLB (classic load Balancer)
- ALB (Application load Balancer)
- NLB (Network load Balancer)

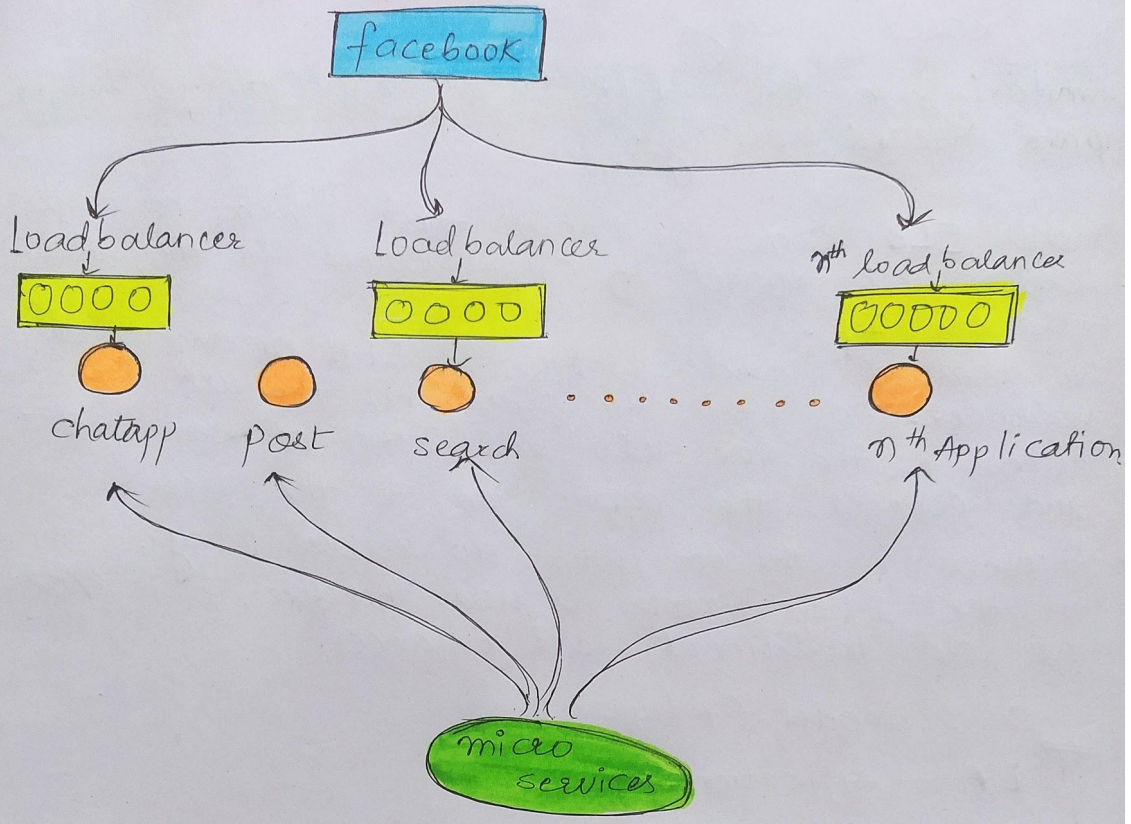
- ELB needs a Ip and the port no. you can give any port no. to elb. so user's request will be taken to the backend servers through the elb port.
- so we need to give elb port on which user will connect and another port of your servers i.e on which server your application running on.

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	8080	HTTP	80

- HTTP. work on a concept of TCP. so we can add the rule of tcp only in the security group.
- you need to give the health check information with their right ping path page. If you given the wrong location about the page in the load balancer then that respective instance will be removed by the load balancer.
- Available Health check timer will check the health ~~to~~ whether the application is live or dead. means if you set health check time to 5 min then the ELB will comes after every 5min to see the health of the application.
- When you haven't registered the ec2 instances or backend servers. so the Domain name of elastic load balancer will not show you the applications output Because there's no server behind.

Date: 27/12/2020
Day: 17

APPLICATION LOAD BALANCER



The combination of the micro (small) services to give the user a complete Application is done by creating different micro services.

Ip with the application (route) are connected to some target load balancer.

Ip/chat → route → Connected to one of load balancer of the backend server

Ip/search → route → connected to one of load balancer of the backend server.

- We don't need to give the route part with the IP. or we don't need to give the routing path.
- The next gen load balancer will have the feature of path based routing.
- Similarly, in AWS. ELB ~~is~~ will have the feature of path based routing.

- How ELB come to know that which backend servers will connect if client comes to the load Balancer IP.
  We need to provide Listner in ELB or any load balancer. so when the client hit the url the listner is the one who will listen on some port and connect the application to the user.
- Listener is the one who do reverse proxy.. in the ELB listner is used. Therefore, it is known as old traditional load balancer.

OSI model. (7 layers)

L7 → Application layer

L4 → Transport layer

L3 → N/W layer

If the program checking Application layer then that is Application layer program. similarly if the Program checks Transport layer then that is known as Transport layer program.

all the port no comes under transport layer. The old classic load balancer will be checking only port no. There fore they comes under transpor layer and known as transport layer load balancer.

Whereas there are different load balancer which follows with Network layer or Application layer and known as Network load balancer or Application load Balancer.

The program which goes till seventh layer (Application layer) then that program takes time to execute. There are the situations which can be faced in this program like more latency etc. They can't directly jump to layer 7. They have to follow the sequence

Network load balancer is high in latency, that ~~is~~ works on 3rd layer of OSI model. Therefore Network load balancer takes less time compare to Application load balancer.

Application load balancer goes till the 7th layer of the OSI model. Therefore, they takes time.

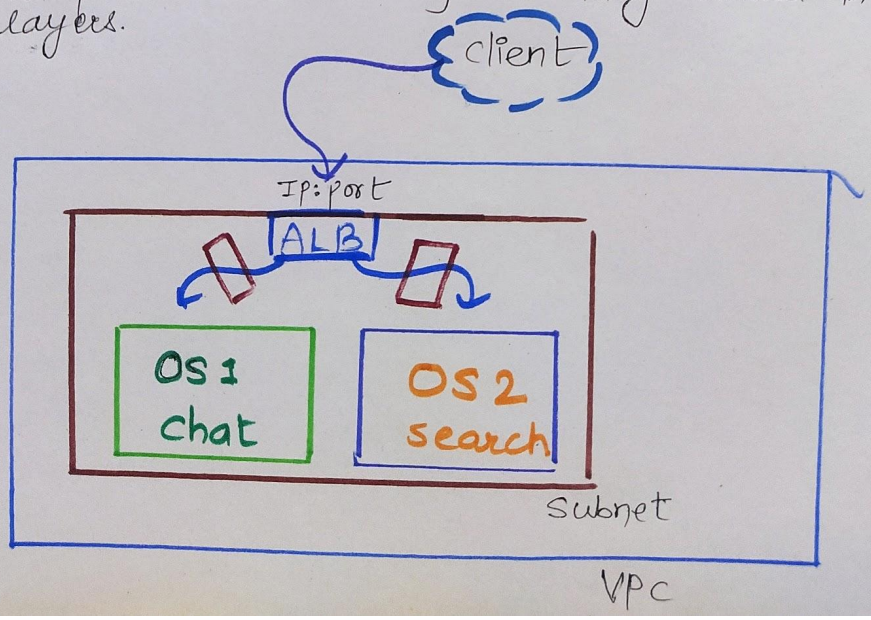
Application load balancer is great application for application architecture including microservices and containers (Kubernetes).

also they used in cookies and query string.

path based routing is done by ALB. and they have check the seventh layer (Application layer) from OSI model.

Therefore, Application load balancer will have all the layers in contact.

Application will be doing the port routing as well as path based routing as they crossed ~~the~~ all the layers.



- In the CLB, we only able to do port routing like ~~wh~~ client will be able to do IP: port only and all the backend servers will be connected to the client.
- Where as in ALB, along with port based routing we can now able to use path based routing also. like if your chat application running on port 8080 and the other application let say webserver running on same 8080 port so client will be be able to connect to chat application using IP: 8080/chat url where as if client wants to connect with webserver then they need to use IP: 8080/web.html.

```
systemct enable httpd --now
```

restart, enable.

Date: 03/01/2021
Day: 18

NETWORK LOAD BALANCER

- http works on layer 7 of OSI model.
- ELB checks the port, do reverse prony, works on layer 4, and takes more time as compare to Network load balancer.
- ALB is more slower than classic load balancer.

~~All~~ All the load Balancer goes in depth of the packes (decapsulate the packets). Therefore, CLB takes more time. This takes more time to connect with backend servers and to display or connect with the client. so to reduce the latency Aws provides service in ELB is NLB → Network load Balancer or Layer 4 load Balancer.

- If you want less latency, and more speed so that millions of client can ~~bar~~ handle NLB.
- NLB works on layer 4 not on layer 7 so it doesn't understand http protocol.
- NLB is very high in the price ~~wit~~ where as ALB and CLB is less in price.
- interval means → ~~How~~ When the load balancer will check the instances ~~or~~ is connected or not.
- Healthy ~~Thread~~ Threshold → How many times load balan will comes to cheet the status

Cross zone Means:

If one particular is in one subnet (ap-south-1a) and the other instance is in (ap-south-1b). Then Whether the load balancer has the capability to check the instances in different (cross) zone.

We launch different instances in different subnets for data disaster recovery.

By default in classic load balancer cross zone is by default enabled.

We don't have cross-zone option in ALB and NLB. But we can enable the cross-zone option in NLB.

When you connect to the first instance through load balancer then it takes time to load the application.

And when you again refresh the load balancer url, then it got connected to different instance and it takes again extra time to load the application. This will have wastage of time.

So, if we can do like if the same client wants to connect again and again or refreshing again. so the client should not go to different instance. that is we want that same client is stick to the same instance with its respective application. This concept is known as stickness. so by default, the stickness is disabled in CLB.

We can also use cookies, to memorise all the work done by the client on the respective application in respective instance.

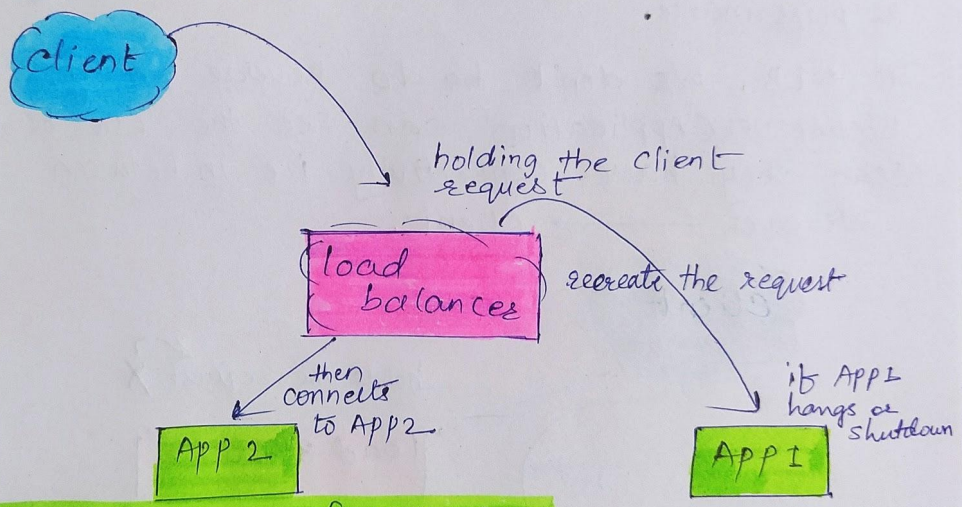
After enabling the stickness they ask how many seconds you want that the client will refresh in which time zone.

like if you give 120s that is 2 min, then the loadbalancer will connect you to the same instance for 2 min.

After the time out they will connect the client with different instances. and stickness will be remoud. This happens due to cookies. as the cookies will expire the clients will able to connect the different instances.

Enable connection Drainage:-

If your application running in the particular instance and you want the output or see the application then output will be holds by the load balancer. Client's request is holded by load balancer and the output of the ~~in~~ application is retrieve by load balancer and sends to the client's request.



Architecture of CLB, ALB

In CLB, ALB the AWS gives the DNS by its own where as in NLB we can assign the public Address IPV4 (using elastic IP) so the IP will never changes and it will gives more feasibility to work with Application using the one single/static IP.

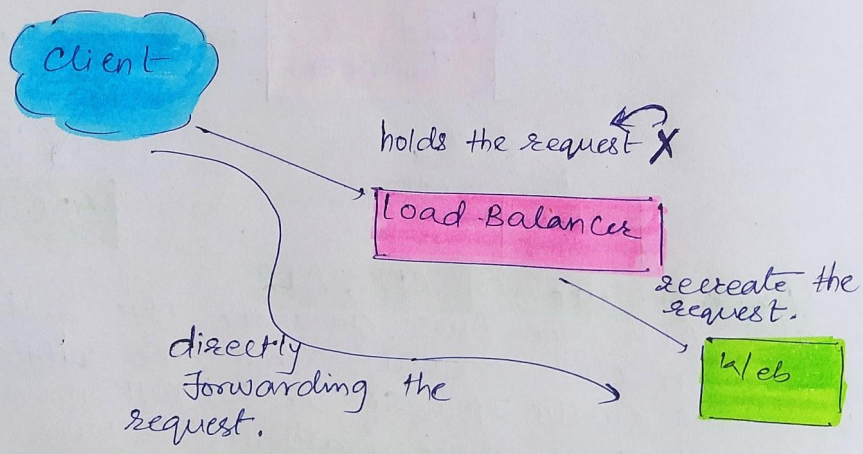
* Main features of NLB.

- High performance
- static IP (elastic IP choosen or Assigned by by AWS while configuring NLB).

NLB works on till 4th layer of OSI model so while configuring NLB you will not find the http protocol.

- Cross zone is by default disable in NLB.
- Behind the domain Name of NLB there is a fixed IP. and it will never change. that IP can be used to set the firewall rules in egress.
- ab tool is one of the tool to measure the load of the Webserver and performance.
- qperf is another tool like ab tool to check the network bandwidth and check the lateney in terms of performance.

In NLB, ~~tate~~ don't ~~to~~ do Reverse proxy. Webserver (Application can see the clients IP, Mean they have connectivity i.e in between Wleserver ⟶ Client.



Architecture of NLB

We have passthrough from http protocol, same in NLB will have bypassing the request to the Application from client.

NLB, will never ~~req~~ holds the request they directly forward.

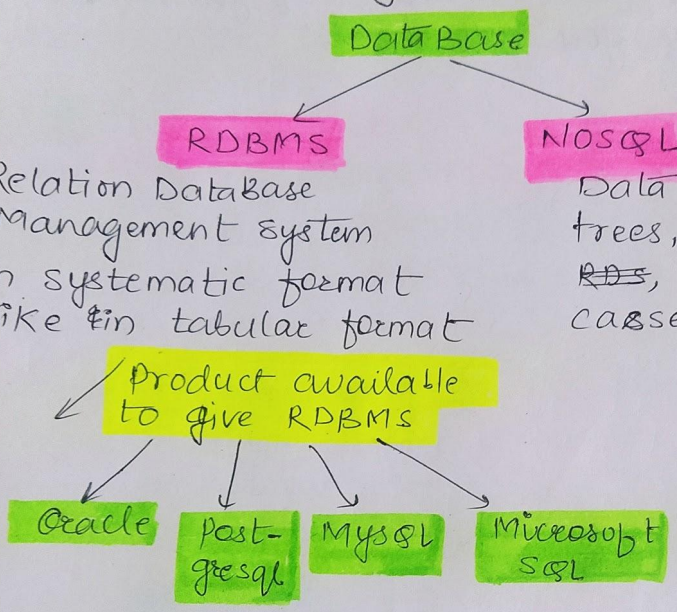
- you see the loadbalancer's log in s3 bucket. you can enable or disable the access log.
- All the elastic load balancer is secure its depends on how you configure respective ELB.

Date: 09/01/2021
Day: 19

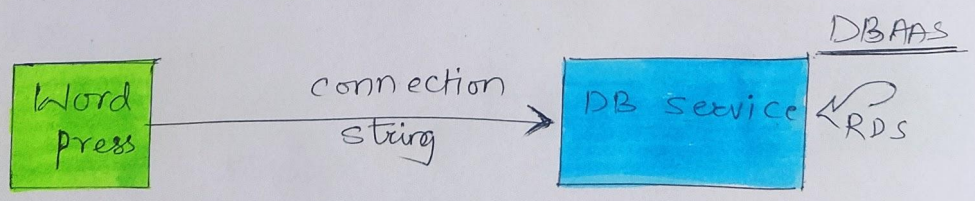
RDS
RELATIONAL DATABASE SERVICES

Database is one kind of program that used for storing data in particular format.

Database Normally classified into two types



RDS is a managed service. you need to use that simply. So, we don't need to install and manage the Databases.



Amazon has created pre-tunned instances. Depends on the historical data they pre-tunned ~~inst~~ for their workload.

every db type in computation have different rack and system. From these we get computational power.

DB instance size

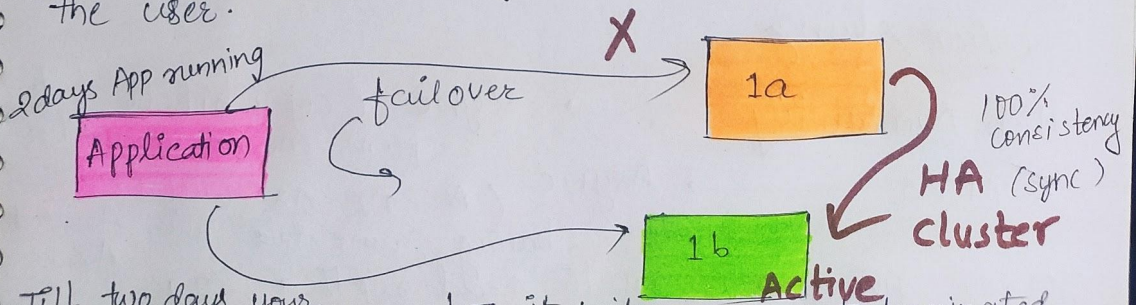
- standard class (includes m classes)
- Memory Optimized classes (includes r and x classes) → more optimized
- Bustable classes (includes t classes) → highly optimized

standard class ⟶ after launch database then storage comes from EBS.

Auto scaling groups for storaged is enable. Means if you have 20 GB storage and if your database got filled the Aws automatically increase storage ~~the~~ depends on the data.

Aws Manages MySQL services through HA (High Availability) cluster.

If your db instances goes down then its HA cluster's responsability to launch new db for the user.



Till two days your App connected with 1a db | If it fails or 1a got terminated the HA cluster will make 1b db with all data from 1a db

HA cluster updates main hard disk of 1b db, if 1a db got failed to connect with your Application. HA cluster plays a vital role in providing consistency.

RDMS meant for consistency. about 100% Consistency.

HA cluster in Mysql gives you 100% consistency. Or 100% got sync all the data of 1a db to 1b db.

Mysql works on 3306, but you can change the respective port no.

Point-in-time Backups will be maintained by AWS and AWS will create EBS with snapshot within time you ~~prod~~ provided ~~to~~ for backups.

MysQL is launched on ec2-instances. But they don't shows in the ec2-dashboard.

yum install mysql → install mysql.

host, user, password

mysql -h hostname -u username -p

connection string.

High Availability → Sync (copy bits to bits from one AZ to another)

replica → Async. (one AZ will be having same data with Another AZ.

you can give public IP to read replica so, some BI Analyst can get updates from that public IP.

Date: 10/01/2021

Day: 20

AUTO SCALING GROUPS

- An Auto scaling group contains a collection of Amazon EC2 instances that are treated as a logical grouping for the purposes of automatic scaling and Management.
- An Auto EC2 scaling features such as health check, replacements and scaling policies.
- you can launch auto scaling group with load balancers or without load balancers.
- Before launching Auto scaling group launch Templates. templates launching and templates configuration launching is one and the same. launching templates gives you extra facility.
- Auto scaling group, also gives you a facility of cloudwatch. They use cloudwatch to let you know the graphs after collecting metrics of the respective desired instances launched.
- When you hit the application running inside the ec2 scaling group then after specific about of load they start scaling out (launching) other instances with respect to your desired capacity.
- once all the instances reached to desired capacity then they won't scale out again.
- you can launch auto scaling group without load balancers also.
- once the metrics (info about compute services) ~~lan~~ gathered by Auto scaling groups they visualizes them into graphs.

Date: 16/01/2021
Day: 21

ROUTE 53

- Route 53 is just a Managed services of DNS.
- DNS is not only the one who converts the name into IP address.
- you can launch the multiple load balancers So if the one Load balancer goes down. The client can connects to otherload balancers.
  So, Now clients need to remember the names of other load balancer.
- So, to avoid this AWS has introduced a new program between the load balancers and the client.
- But what if this program goes down. This program never goes down. if all over the internet goes down then only this program will goes down.
- The name of the program is DNS → Domain Name System.
- DNS can be used for load balancing also.
  We can place the ip and set the aliases.

  8.8.8.8 lw a → aliases

The ip is placed inside /etc/hosts. file.
But if you have Broder/Big network then ~~this~~ We cannot create local database of IP in /etc/hosts.

Always try to give the names that uniquely identifies the systems. i.e Always tries to give unique host names.

you can use hostname with your domain name. eg. (.com, .in, .org)

The domain like aws.com is unique and no one can use this domain except ~~do~~ aws. So, aws is the owner of aws.com domain.

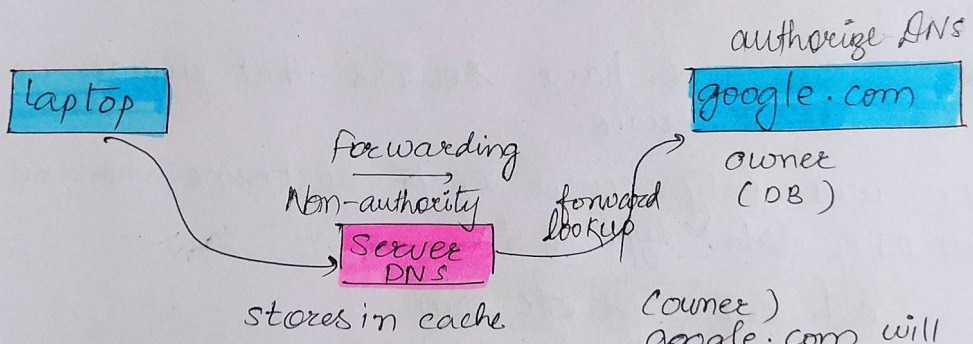
Aws gives you a facility of creating a database of hosts.

www → is ~~the~~ just a name.

WWW.google.com

← FQDN →

The DNS, also comes with cache facility. if you connect to destination using DNS. eg. Hotspot, Broadband then DNS uses cache facility.



(owner) google.com will decide if a respective system connected. And own will decide to respond and store the IP in the cache of the server. Till what time the ip will be stored in the cache of the server that is known as time to leave (TTL)

Resolving into Name in ip this known as forward lookup.

name $\xrightarrow[\text{A type}]{\text{forward}}$ IP

IP $\xrightarrow[\text{PTR type}]{}$ hostname

Every company maintains their own database of hosts.

The one who maintains their DNS Databases are known as Authority server or master server.

In internet, thousands of ip connected to each other. Therefore internet never goes down. Same as we have millions of DNS connected to each other. Therefore, DNS never goes down.

Bind is a product in linux which provides you DNS.

Same in AWS we have Route53 that provides you a DNS as a service.

DNS will only convert IP to hostname when you have PTR ~~data~~ type of data.

```
nslookup google.com
nslookup -debug google.com
```

Aliases is one of the way to give your host name of instances or loadbalancer to your host name.

hostname $\xrightarrow[\text{(cannonical name)}]{\text{CNAME}}$ hostname

www.google.com ⟶ mail.google.com ⟶ ~~google.mail.l.com~~ googlemail.l.google.com ⟶ IP.

CNAme = Nickname = Aliases.

But in perspective of AWS.
~~As~~ CNAme ~~CNAME~~ we can use with any of the name. where as Aliases can be used ~~by~~ for the only services present in AWS.

DNS works on port no 53. therefore they have given route53.

Date: 17/01/2021
Day: 21

CNAME = Nickname = Aliases.

But in perspective of AWS.
~~As CNAME~~ CNAME we can use with any of the name. where as Aliases can be used ~~by~~ for the only services present in AWS.

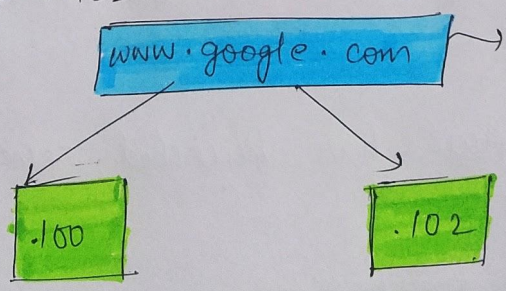
DNS works on port no 53. therefore they have given route53.

DNS uses UDP protocol. Every second DNS recieves many millions of the request. To ~~to~~ deal with the connections faster UDP protocol is the best way for the ~~be~~ faster and better connections.

Inside /etc/resolv.conf is the location where we put all the DNS Name servers.

When you have one DNS Name eg. www.googl.com and Behind that you have multiple Records of IP. ~~of all the~~ Then DNS do ~~ro~~ load balancing in the round robin fashion.

DNS → www.google.com
Ip1 → 192.168.99.100
Ip2 → 192.168.99.102



→ DNS does load balancing with the IP's.

Route53 provides DNS such that they can do load balancing and the health check also.
In routing policy we need to mention about the health check. By default, Route53 doesn't uses health check. We need to do that in routing policy. The name of Routing policy is Multi-Value.

```
nslookup  Ip or dnsname

Server : 192.168.0.128 # 53
Address : 13.232.206.42.
```
↳ port 53

The main dns is known as root dns and it is shown by . (dot)

```
dig  Ip or dnsname
```

When you will hit above command then you will get extra info out your dnsname.

The main info was

AUTHORITY SECTION:-

_____
_____ ← No. of AUTHORITIES (MASTER)
_____
_____

Behind the scene Aws has launched several Authorities (master).

```
host -t ns google.com
```

will show you the servers running behind that.

In simple routing, AWS don't provide Health checks.
If you give multiple Ip in the DNS servers.
Or when you give records of Ip. Then by using nslook ~~to~~ DNSName then you will have multiple Ip's in the ANSWER SECTION. But when you use ping command with the DNS Name then they ~~sho~~ shows you only one Ip. Behind they are balancing the load of the clients.

But when you change the TTL time. eg. 5 minutes then when you connect using ping command with the hostname then local DNS stores cliches ordered by the Master DNS with respect to 5 minutes. ~~In th~~ you will notice that you will be getting only one Ip. Because the caches stores in the local DNS for 5 minutes. After 5 minutes they can associate to other Ip.

Single routing, doesn't have health check property. So if one of Ip is not working the routes with single routing will not helps you. you need to switch to different routing. eg. Multi value routing.

One ~~pro~~ issue in the single routing is. when you have 5 mintues wait for the local caches But if the ip's behind the DNS goes down then till 5 minutes will show you the same Ip which is down. After 5 minutes when you connect to dns name Master DNS gives you the Ip's of other instances due to round robin. ~~some~~

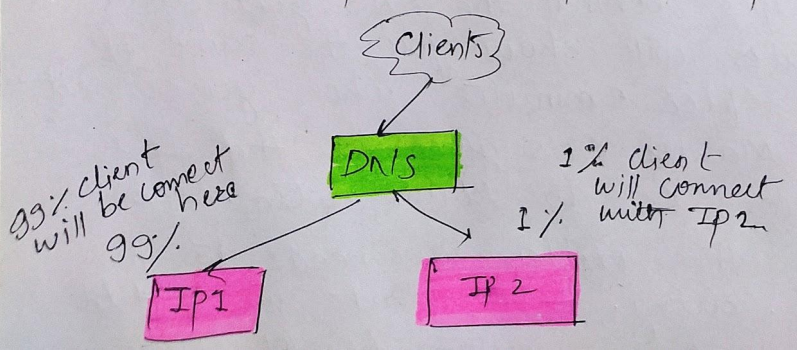
so to avoid this problem, we needs to go to Multivalue routing. Before that we need to Create health checkers per instance / load balancer basis.

In the creation of health checkers. we need to give the path of the application which is running inside the instance.
Aws internally checks the application multiple times to avoid the problem of going application off.
you can choose your regions. means from where you wanted to check healths.
How many tries have done to check the Ip or the application is healthy or not is denoted by **failure Threshold.**

you need give your respective records of Ip when Creating a Multi value Answer routing policy ~~with~~ by giving the health checker you created for your respective Ip.

Now after creating **Multi value Answer** routing then you don't have to worry that DNS Name will be connected to the ip's which is down. This issue you will not face in the Multi value Answer routing.
The DNS Name will connect to that Ip which ~~have~~ is up and running.

We have **weighted routing** policy. we use use this policy when we have to transfer the load to respective Ip's in specific ratio.



Failover is another routing policy. This routing policy states that when there is a running application the client will be connected to that application till it not go fail. If the application got fails or crashes then failover routing policy will do round robin and switch to next application running in other application.

nslookup hostname dnsName Masterdns

→ if you don't want to ~~do~~ use caches then you can use.

Global accelator, Cloudfront uses Geolocation and latency routing table to fetch the application from different edge locations.

Date: 23/01/2021

Day: 22

Metadata Server: The data about the server and the services of Aws is known as metadata server.

- Metadata server running in the private world of Aws. EC2 can connect to metadata server.

Linklocal Ip → Ip that available between range of **169.254.0.0** to **169.254.255.255**. are known as linklocal Ip's. EC2 instances uses linklocal Ip.

api → metadata of ec2

curl http://169.254.169.254/latest/metadata/public-ipv4

- Inside the above file public-ipv4 you will be ~~showing~~ seeing data about public Ip's of ec2 instances
- Above url is one of the API through which we can see the metadata of ec2 instances.
- format of Images on cloud → **AMI** / **qcow2**
- All the images are pre-installed images. We just wanted to ~~yo~~ use them.
- we can use Boot process logs to debug the issue if you face the issue while Launching the ~~image~~ OS.
- **cloud init** is a program that used by Aws to launch the OS. Cloud init is a program available in user-data which is available in meta-data of the api.
- you can create your own data inside userdata while launching the OS. cloud init will take those user data and run that while ~~can~~ launching the OS.
- Just you need to put shebang of the shell you will be using. eg. **!#/bin/bash**

- After launching OS you can go inside
  curl http://169.254.169.254/latest/metadata/user-data
  and there you will be seeing all the commands you have entered while launching the OS.

- you can use DNS to set the routing policy as latency and geolocation.
  when you use routing policy latency then you need to provide the IP. let say you have launched ec2 instance in Singapore region and other instances launched in US region then after creating record of the respective ~~host~~ rout53, then if you hit that url of rout53 with the record at the starting then you will be landing or accessing the page from Singapore region. Because accessed the url from India. If you hit that same url from USA then you will be accessing that page from US region. This is Basically due to latency routing policy.

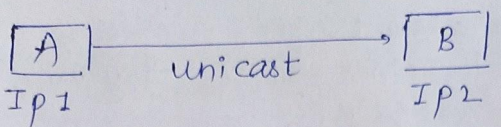
  But if you want that all the indians from India can access the site then they will connect and the ~~regio~~ people from USA region then they will not able to access. Therefore, to do like this setup then you need to use geolocation routing policy.

Date: 24/01/2021
Day: 23

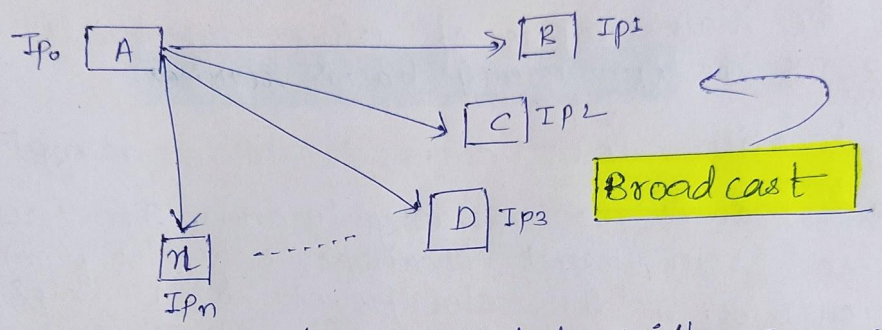
GLOBAL ACCELATOR

- When you use cloudfront for accessing application from ~~web~~ ~~loco~~ edge locations. Then they use Route 53 with latency routing policy.
  Due to Latency routing policy they gives us a facility of edge locations. so we can access the application instantly.
- One of the main feature of using edge locations is they provide low latency based services.
- edge location stores caches data of an application
- If you wanted to you edge location. Then you can use from nearest locations only. Not from other countries. In this cloudfrount won't helps you if you ~~don't~~ wanted to get the application from other countries. i.e you cannot create a cache in in local countries.
- Global Accelator don't creates a cache in the edge location. They directly connect the destination
- cloudfront creates a cache in the edge location every time you hit the destination, they fetch from near edge location. In cloudfront's we need to hoop the request from different network.
- Around 60% up performance you will get if you use global accelator.
- Global Accelator provides you two public static IP address. If one goes down by chance then they have another. so you will not loose the connection.

If you connect from one Ip with another Ip then it is known as **unicast**.



But if you ~~use~~ connect one Ip with ~~an~~ multiple Ip then it is known as **Broadcast**

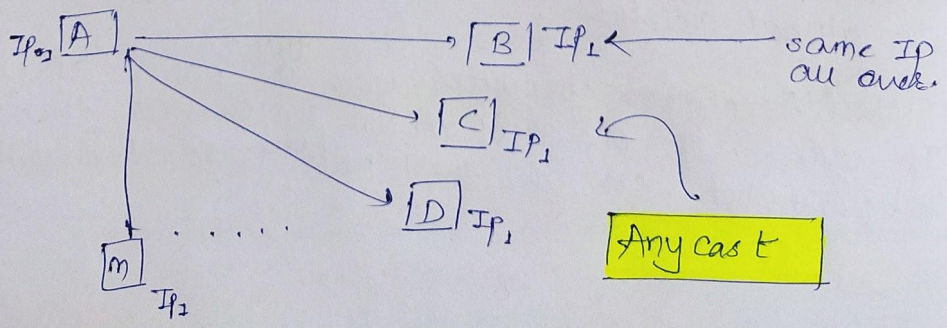


if you wanted to connected with any of the same Ip then it is known as Anycast.

Anycast came from ~~Ip~~ Networking world and DNS world.

Like when you wanted to connect any system ~~the~~ all the system uses same Ip so when you wanted to connect to any destination/system then they connect to the nearest location. Because all the Ip is same. But this is only possible in public world.

Global Accelator uses **Anycast Ip** Based connection



Global Accelator is only available in Oregon region.

- Similarly, Billing service is only available in North Virginia region.
- while creating global accelator you need to give ports on which your Application is running.
- Global Accelator will create load balancer for you and you just need to give the Subnets name.

Difference Between Global Accelator and CloudFront

Directly access web applications

Without AWS Global Accelerator

It can take many networks to reach the application. Paths to and from the application may differ. Each hop impacts performance and can introduce risks.

With AWS Global Accelerator

Adding AWS Global Accelerator removes these inefficiencies. It leverages the Global AWS Network, resulting in improved performance.

Day: 25 NO NOTES

Date: 06/02/2021
Day: 26

ELASTIC BEANSTALK

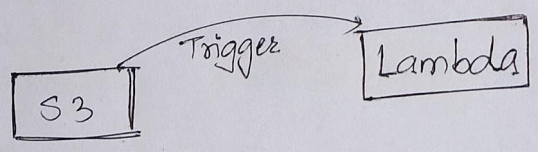
- W3 school is an example of PAAS. all the inner infrastructure will be hidden out.
- If you want to use PAAS in AWS then it will be provided by EB (Elastic BEANSTALK).
- Pipeline is don't managed by PAAS i.e ~~B~~ don't managed by Elastic Beanstalk. But Openshift can manage pipeline.
- Elastic Beanstalk support on WebApp support.
- EB is a free service.
- EB has all Deployment support eg. Blue/green, Canary deployments, etc.

Date: 20/02/2021
Day: 27

AWS LAMBDA

- We need some servers to run the servers eg from Webserver (apache, nginx, tomcat, Jboss).
- some time We need to deploy apps that is depended on Multiple functions. so we don't wanted to run all the functions. This is todays demand. So Aws provide you a service function As a service provided by Lambda.
- When you deploy an App using Beanstalk then Behind the scene they launch ec2-instances.
- Serverless means ~~not~~ there will be no servers running into your System.
  i.e Run code without thinking about servers.
- if you are using Beanstalk then you get a url of your application. if you don't use or use that url then also Aws will charge you. because the ec2-instances, loadbalancers are running behind.
- But in Lambda, we don't have this scenario like how much you use the function that much you have pay.

We can integrate all the services to lambda. eg. S3 can be integrate with lambda and can ~~get~~ get and put events using triggers.

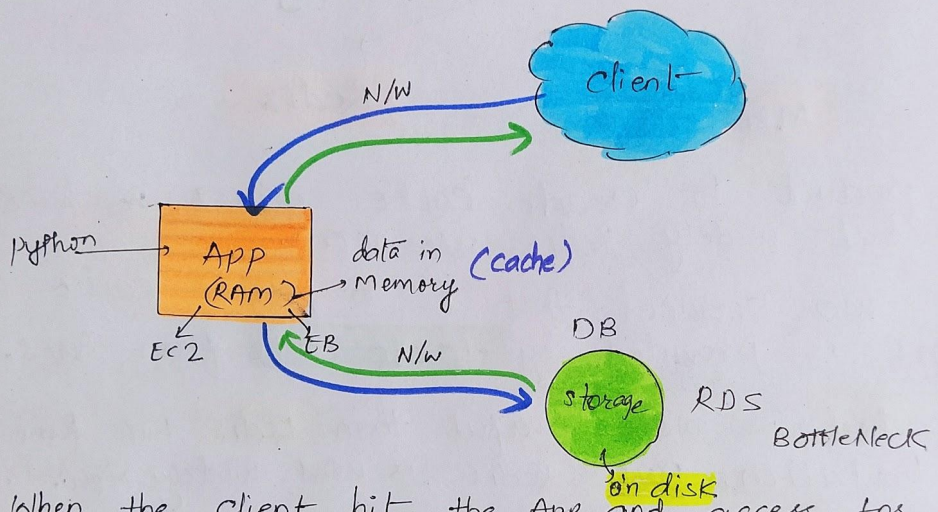


- In the free tier account you have 1 million Invocations.

Date: 21/02/2021
Day: 28

ELASTIC CACHE

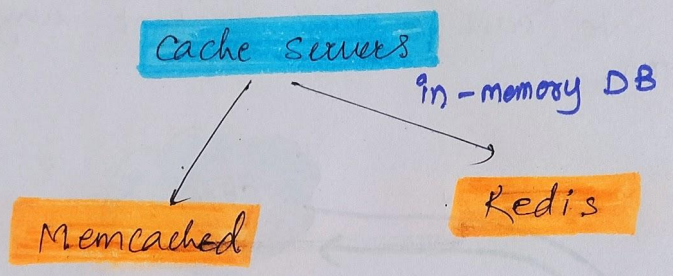
- storage is always slow compare to ram.
  storage ⟶ DataBase.
  storage is known as **Bottle Neck**.
- As soon as client click on particular button eg. search, Myaccount, exit, signout, cart, et so how we can provide fastest way to open the button.



- When the client hit the App and access for something and if its not there in Ram so it searches in Database and then it loads on the Ram and then serves to client. But also it updates in the in memory data base. so if next client comes for the same data then it will be given from Ram only. same thing we came across in **cloudfront**. These data are known as **session data** or **cache**.
- Cart is your just like session data or cache. some of the data is stored in harddisk (database) and some of the stored as a cache in Ram. eg. Cart data, login details etc.

- Don't create inmemory database in your localram. Normally, cache servers works in cluster. that if one ~~cluster~~ cache server goes down then we have different cache servers from the cluster.

- So we need to create cache servers into in memory database.



- two product to create cache servers. we have more but widely Redis use more.
- These two services that is to provide cache servers is provided by elastic cache from Aws.
- Memcached is more powerful than Redis but have less limitation. But Redis is used widely Now.

What ever you will do into reddis you need a connector/driver to connect your system with reddis.

driver ——→ redis

yum install python-pip

pip3 install redis

redis works on 6379

```
import redis

r = redis.StrictRedis(host='endpoint',
                      port= 6379,
                      password = " ",  → Null password.

print(r)
```

~~set(~~

If you want to send some data from system to Redis database. we need to send the data into Key.

let say cart = "mobile".

```
r.set("x", cart)
>> True
```

→ in redis it stores like x = "mobile"

```
r.get("x")
>> b'mobile'
```

→ this will pull the data from redis

- Even your system goes down and if you connected from other system it will be available only you need to connect to redis once.
- But if you reebot the redis cache servers then you will loose the data.
- We can create a cluster in Readis. But in Memcached this feature of creating (multi AZ) is not there.
- Sharding is one operation that makes facility on write operation on multiple partition.
- Same memcached support the facility of sharding. In memcached we don't have Backup support.

In reeddis, values stored in Key value type. In mysql or oracle the data stored in tabulas format and known as RDBMS. Therefor Redis is known as Key value data base and comes under NOSQL Data bases.

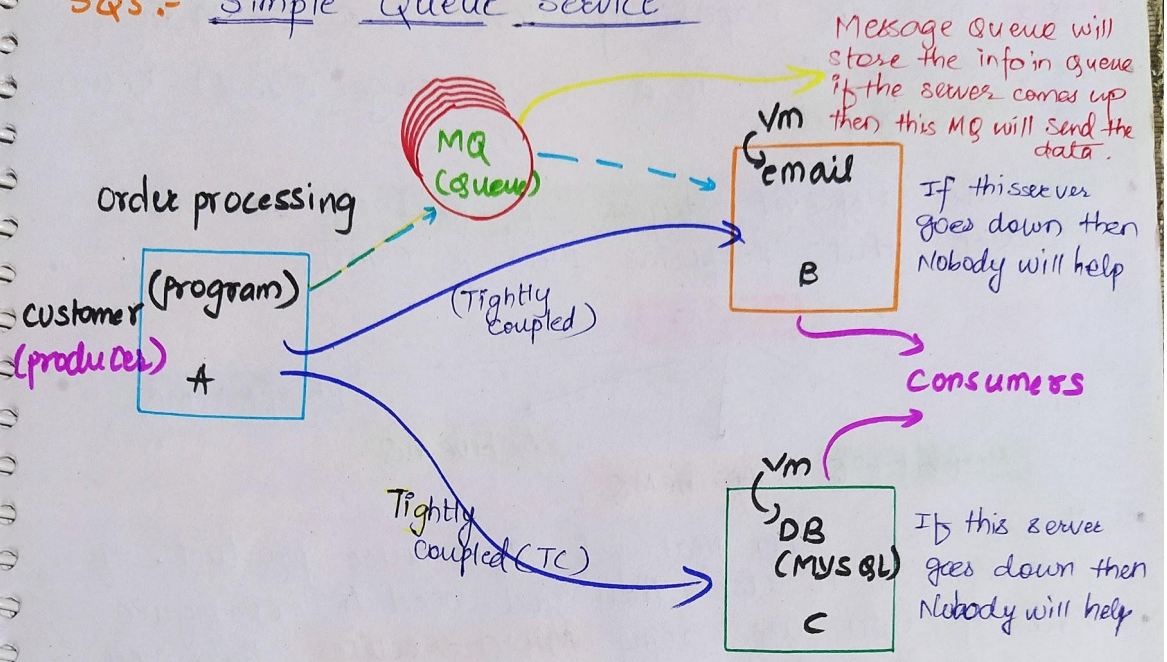
To connect to memcached you need to install driver.

```
pip3 install ~~pymemcache~~ pymemcache
cart = 'telescope'
~~import~~
from pymemcache.client.base import Client
c = Client('endpoint')
c.set('x', cart)
c.get('x')
>>> b'telescope'
```

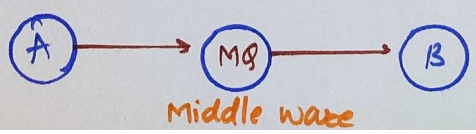
Date: 28/02/2021
Day: 29

SQS SNS MQS

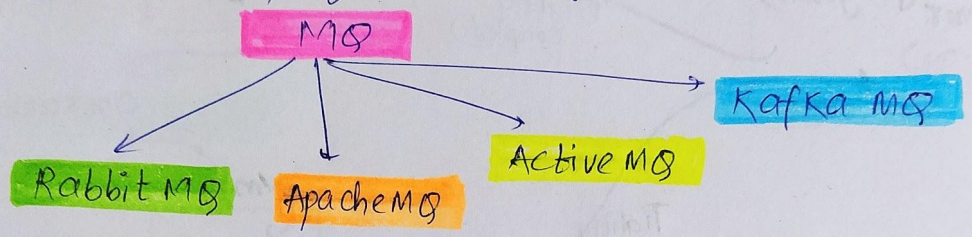
SQS :- Simple Queue service



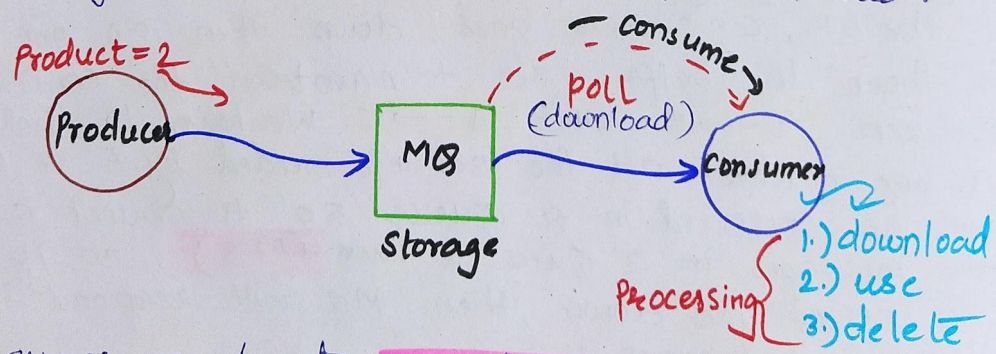
- When server (A) (producer) will sends the data to server B (email server) and server C (DB server) then they are tightly coupled with A server. so sometimes if the B, C servers goes down then No one ~~one~~ is there to help. so to maintain the continuality Between A → B and A → C. We have to include a box in which all the ~~se~~ info sended by A to B, C will be collected in a queue. so to collect all the messages in a queue we have (MQ). so if the server goes down then MQ will respond if the
- New servers comes up.
- As MQ is in the middle of server A and server B therefore, it is known as Middle ware.



- Message queue is just a concept. in which they provide to store data for 14 days. in AWS we have SQS that do Message Queue.
SQS has the capability store data to 14 days.
- The storage time in a SQS (message queue) known as through put.
- We have diffecnt types of products in the market that performs Message Queue.



- These are non managed opensource product. If you wanted to use then you need to configure.
- But if you are doing Micro-services then you need to used Manage message queue and AWS gives us SQS (simple Queue service)
- Message queue is not like a Database.



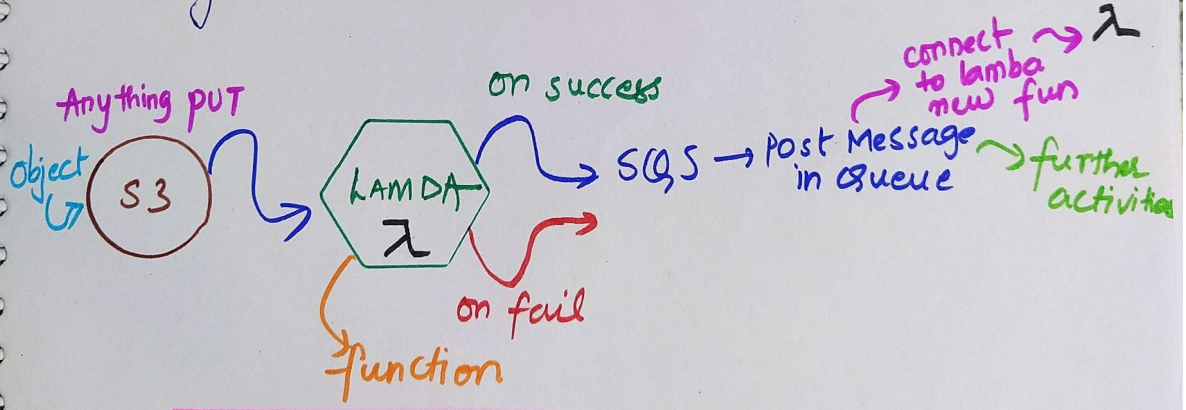
- consumer needs to poll (download) the data in the queue and use it. if they used then consumer needs to delete that data.

The size of the message minimum is 1Kb and Maximum size is 256 kb

- Data sending in Network packets is known as In-transit data.
- SQS is the serveless service.

If you poll the data from the Queue multiple times if the data is same then the same data is not get replicated in the database.

- eg. if customer purchase the product and paid the amount the he/she will purchase that product only. Because the data is only available in the database.
- As the purchased done then that will be deleted from the database. If not deleted then customer will recieve the same product again and again.
- To delete the message, We api for deleting the messages.



FIFO:- first in first out.

What messages in queue goes first will poll first.

The default visibility of a message in a queue is 30 seconds. i.e ~~the~~ if one customer poll the message from the Queue the other customer will not fetch (pol) the same message for atleast for 30 seconds.

Date: 06/03/2021

Day: 30

CLOUD FORMATION

- IAAC (Infrastructure As a code) → cloud formation
  We can directly automate the Aws services.

Manual
- webui
- cli

Automation
- Api
  - python (setup)
  - (Not intelligent)
  - (Based on Imperative language)
- Declarative language
  - (works on Idempotance)
  - Intelligent
  - cloudformation

- Cloudformation will create entire infrastructure by code.
- Cloudformation will automate the aws services only.
- Terraform can do the same thing as cloudformation doing. Terraform can provision the ~~coto~~ cloud resource

```
Resources:
    MyFirstOS byCF:
        Type: Aws::EC2::Instance
        Properties:
            ImageId: ami-05db93d568bb6c05ce
            InstanceType: t2.micro
            AvailabilityZone: ap-south-1a
```

Name of Resource

Type of Resource

Properties of Resource

Parameters:
  X: → input value
    Type: string
    Description: this is variable for my instance type.
Resources:
  Myfirst OS by CF:
    Type: AWS::EC2::Instance
    ~~Parameter~~
    Properties:
      ImageId: ami—"—
      InstanceType: !Ref X ← Refer X
      AvailabilityZone: ap-south-1a

variable

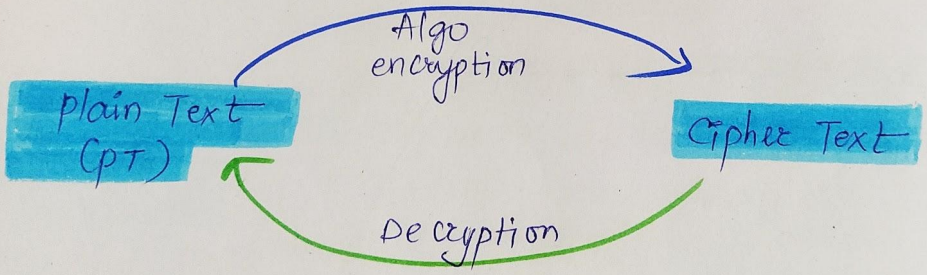
Date: 12/03/2021
Day: 32

KMS
KEY MANAGEMENT SERVICES

- If you want to encrypt → EBS, EFS, HTTPS, RDS, Lambda → Then that is managed by KMS.

- Cryptography → security, writing

In computer world, locking is known as encryption. the data which is locked (encrypted) is known as Cipher ~~data~~ text. and whenever you unlock the data then that is known as decode.



- if your plain text is encrypted and decrypted with a same keys then that is known as symmetric keys.

There are different types of algorithms that used in symmetric keys.

1.) AES : this algo is slow but more powerful.

2.) DES : this algo is fast but not powerful as AES

- Almost every symmetric key algorithm used in their respective use cases.
- But there is also Assymmetric Keys.
One key is enough to lock and unlock. In this case symmetric keys are enough to give security.
- By chance, if you locked (encrypt) the data and you lost the keys then it is not possible to unlock that data.
- Key is also like a data store in a file known as Key file. So it is most important to secure the keys of respective encryption.
- There are many companies who created real hardware that will save guard the keys. special kind of Appliances eg. HSM Appliances (Hardware security module).
- So, these Appliances are the one who in which keys are securily managed. But these ~~are~~ appliances are more expensive.
- Normally, the hardware is not safe, it might be crashes. If the the hardware crashes then it will also crashes the keys. So companies usually buyies 2 hardwares. If one hardware crashes or fails then extra hardware will be there to manage your keys
- So, we need some Managed services, so that we shouldn't ~~by~~ buy HSM model. and we can use pay as we go model. So AWS provides you a service that will helps you to pay as we go model: i.e Cloud HSM.

- Cloud HSM is created per VPC. so every system, databases (subnets) in that VPC will be encrypted.
- But there are also services in cloud that will create multiple HSM. to keep your keys securily.
- you don't have to create any cloud HSM. they will create behind the scene.
- Cloud will give you a hardware that dedicated to a particular user/tenant. that is cloud will not share that hardware to others. This will make keys highly secured.
- In KMS →
  - → Keys created by KMS
  - → Keys stored by KMS in cloud HMS cluster
  - → Keys managed securely by KMS.
- AES and openssl are the same algorithm.
- The main ~~is~~ thing is after creating keys from different algorithms where we keep the keys.
- so, KSM also provide ~~CKS~~ CKS → Custom Key store. where we have to keep the keys after keys creation.

There are three types of key:-

1.) KMS → KMS will create key automatically
2.) External (import your keys) → you can import your keys
3.) Cloud HSM (Custom) → will take keys from cloud HSM

aws kms encrypt --plaintext fileb://path of file
--key-id key-id

this will encrypt your data

aws kms encrypt --plain text fileb://path of file
--key-id key-id --query ciphertextblob --output table.

only this will be in output

output will be in tabular format

- The output will will be encrypted.

aws kms encrypt --plain text fileb://path of file
--key-id key-id --query ciphertext-Blob ~~text >~~
--output text > secure.txt

output of encrypted data in secure.txt

- To decode you need same key but also you need base64 decode

aws kms decrypt --ciphertextblob file://path --key-id
key-id ✗ gives you error.

this will provide you Base 64 format

Base 64 encryption ends with "=" at the end.

previous file

certutil -decode secure.txt secured.txt

this will create a binary file.

Binary file

aws kms decrypt ~~--cipher~~ ciphertext blob file://path
--key-id key-id secured.txt

this will give you in decoded format.