

AWS JOBS YOU CAN GET WITH AN AWS CERTIFICATION



- **#1. Operational Support Engineer**
- **#2. Cloud Software Engineer**
- **#3. System Integrator — Cloud**
- **#4. Cloud Developer**
- **#5. Cloud DevOps Engineer**
- **#6. AWS Solutions Architect**
- **#7. AWS SysOps Administrator**
- **#8. Senior AWS Cloud Architect**
- **#9. AWS Consultant**

Activate Windows
Go to Settings to activate Windows.

18

COURSE OUTLINE



Client Server Architecture

Network Basic Concepts

Virtualization Overview

Cloud Computing

Amazon Web Services Overview

High Availability Architecture

AWS Sign UP Procedure

MFA Configuration

AWS CLI

Compute (EC2, Elastic Beanstalk, Light Sail, Lambda, GA)

Storage (S3, EBS, EFS, Storage Gateway, Glacier, Snowball)

Databases(RDS, Redshift, DynamoDB, ElasticCache)

Network and Content Delivery(VPC, Route53, CloudFront, Direct Connect, Global Accelerator)

Management Tools(CloudWatch, CloudTrail)

Security(IAM, Trusted Advisor and Inspector)

Application Services(SWF, Transcoder)

Messaging (SNS, SQS and SES)

Overview on Developer Tools(Code Commit, CodeBuild, CodeDeploy, Code Pipeline)

Route53 Failover Project

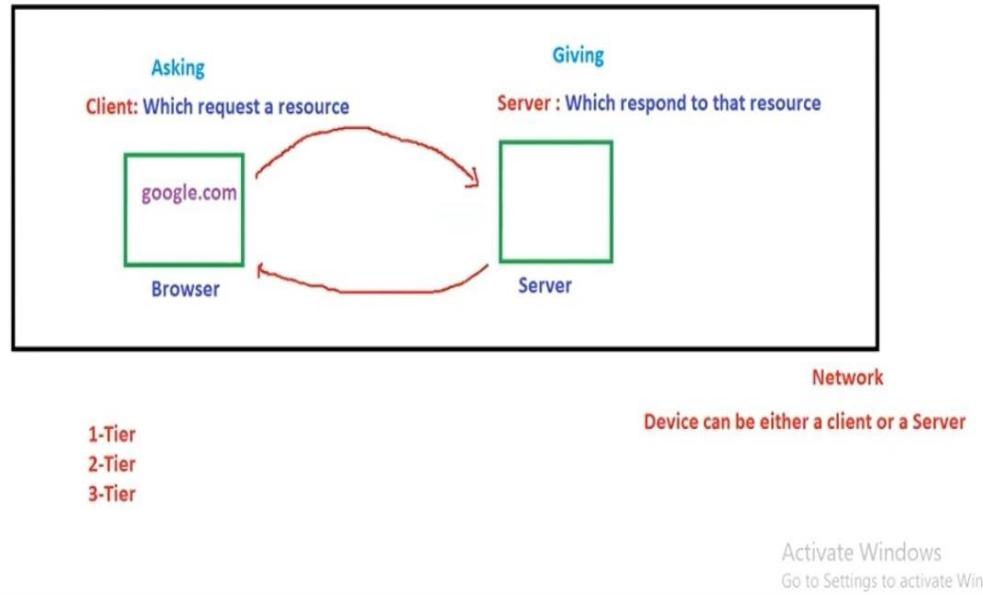
Architecting VPC from scratch(Project)

Final AWS project including multiple services

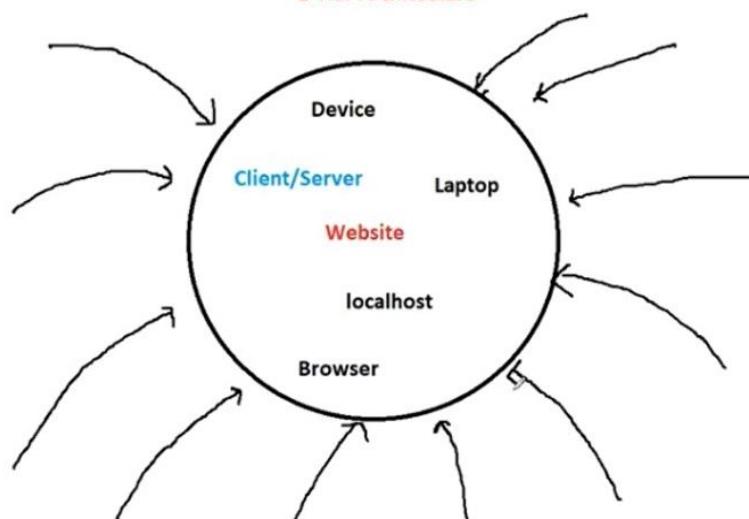
Activate Windows
Go to Settings to activate Wind

6

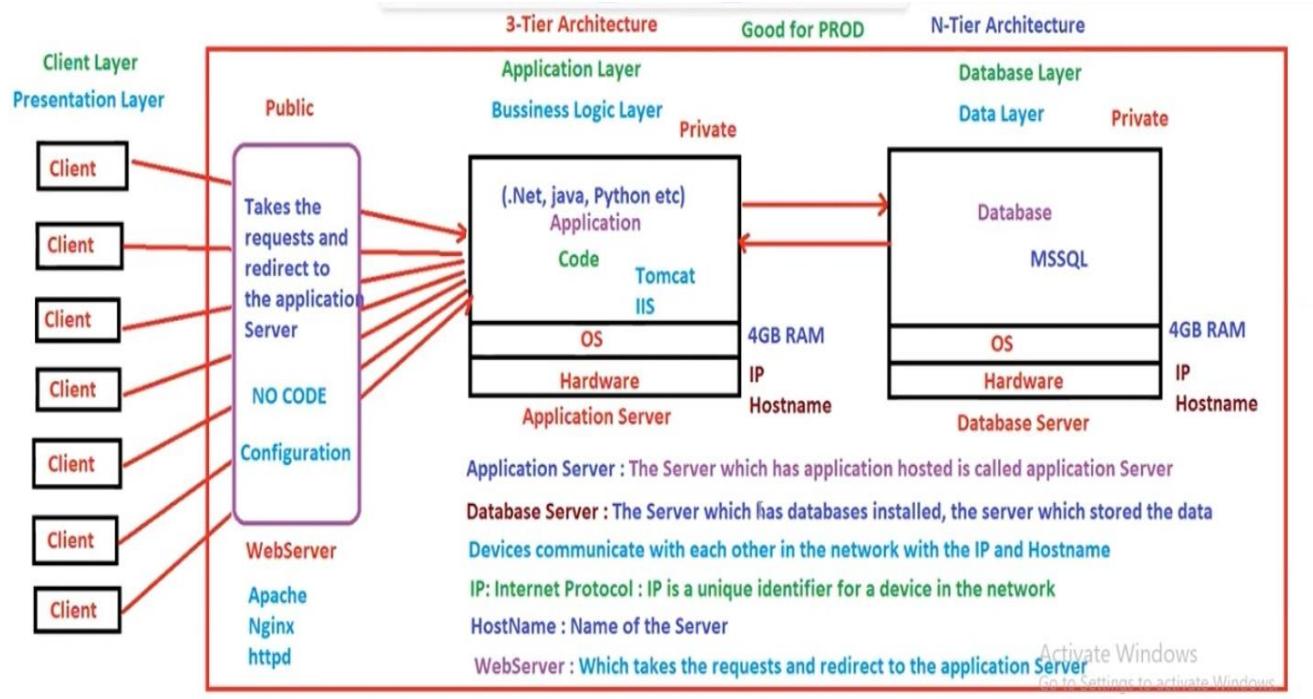
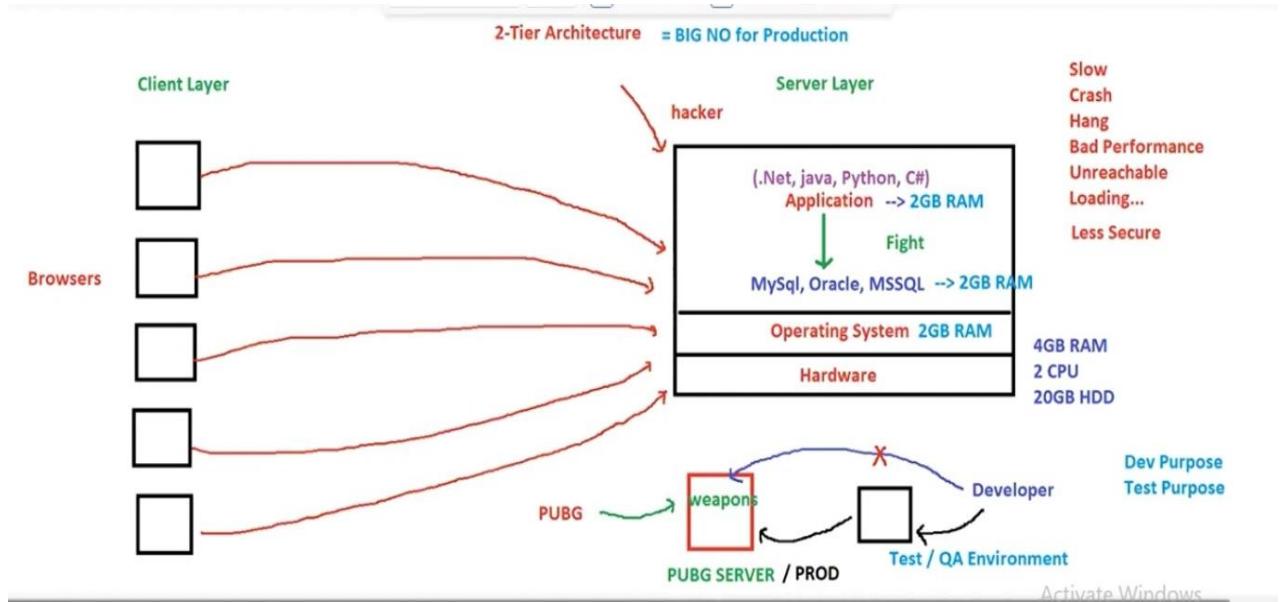
Client Server Architecture

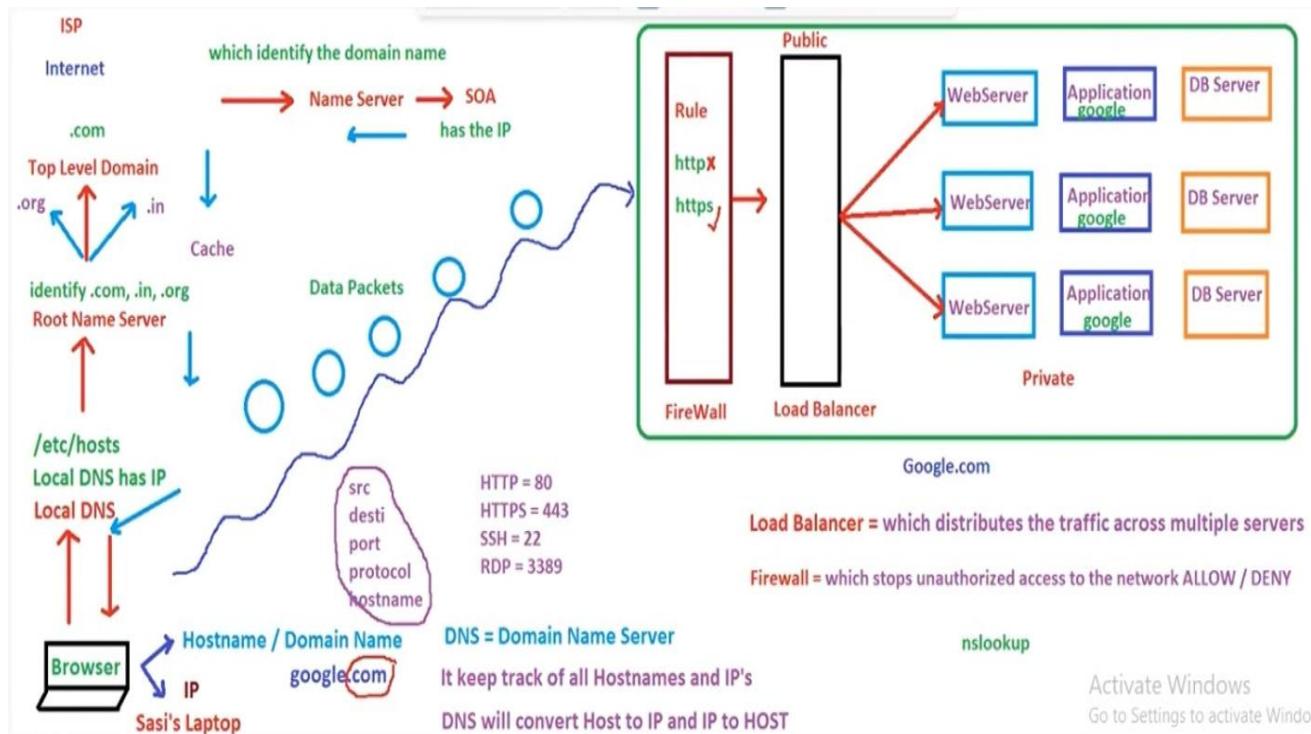
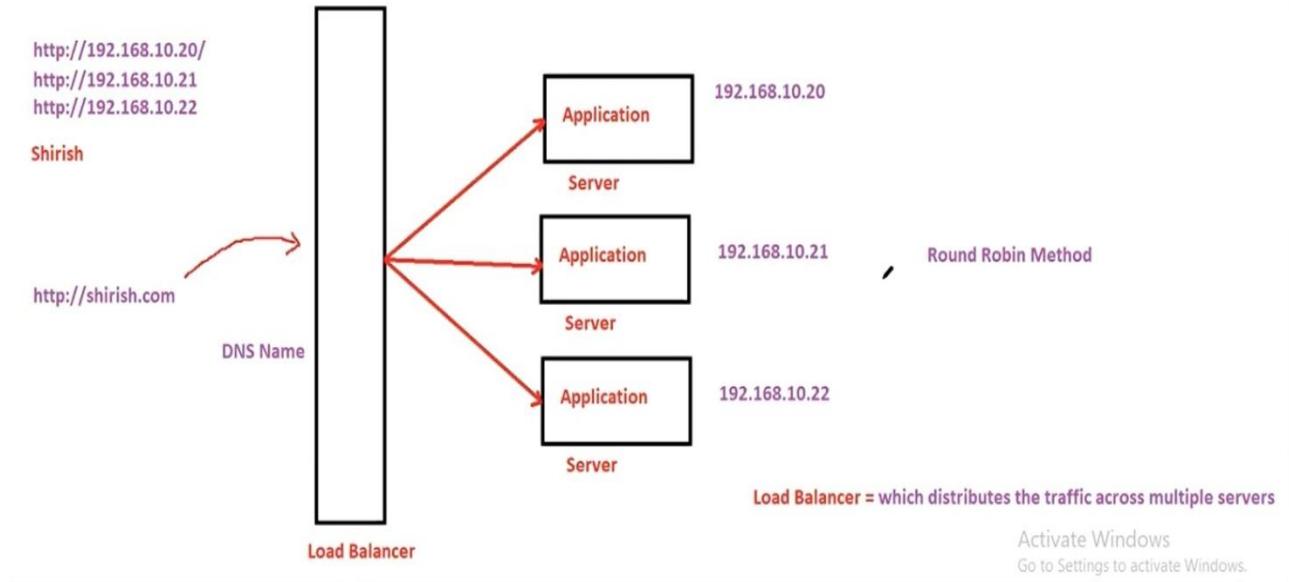


1-Tier Architecture



Act





Browser --> Local DNS --> Root Name Server --> Top Level Domain --> Name Server --> SOA
 Browser --> Firewall --> Load Balancer --> WebServer --> Application Server --> DB Server

HTTP = HyperText Transfer Protocol

<http://google.com>

HTTP default Port Number is 80

Protocol://Domainname:PortNumber

You can customize the port number on application level

<http://192.168.10.10:80>

<http://192.168.10.10:8080> ✓

<http://192.168.10.10>

Application

Tomcat

8080

Server 192.168.10.10

For customer its always recommend to expose port 80 only
[https - 443](https://www.google.com)

Status Code

404 = Page Not Found

503 = Service Unavailable

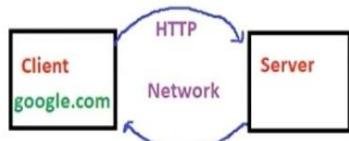
401 = Unauthorized

500 = Internal Server Error

200 = Success Code

HTTP Over TCP/IP

Transmission Control Protocol



HTTP transfers the data to and fro from the browser to Server

HTTPS - 443 (Secure)

Certificates SSL

Encrypted

SSL / TLS / HTTPS

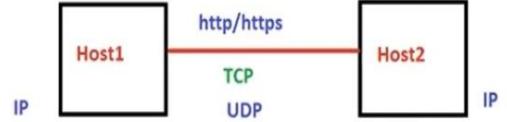


http (Not Secure)

hacker

Plain Text

Host = Client/Server

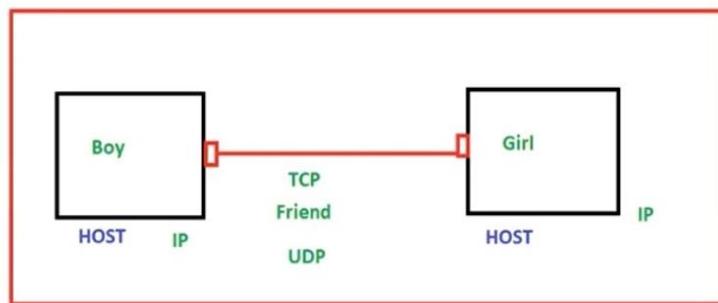


TCP establish the connection between the 2 hosts

TCP is like a messenger / bridge

Activate Windows
 Go to Settings to activate Windows.

OSI Layers



7 Layers

Application Layer ([http](http://www.google.com))

Presentation Layer ([ssh, rdp](ssh://www.google.com))

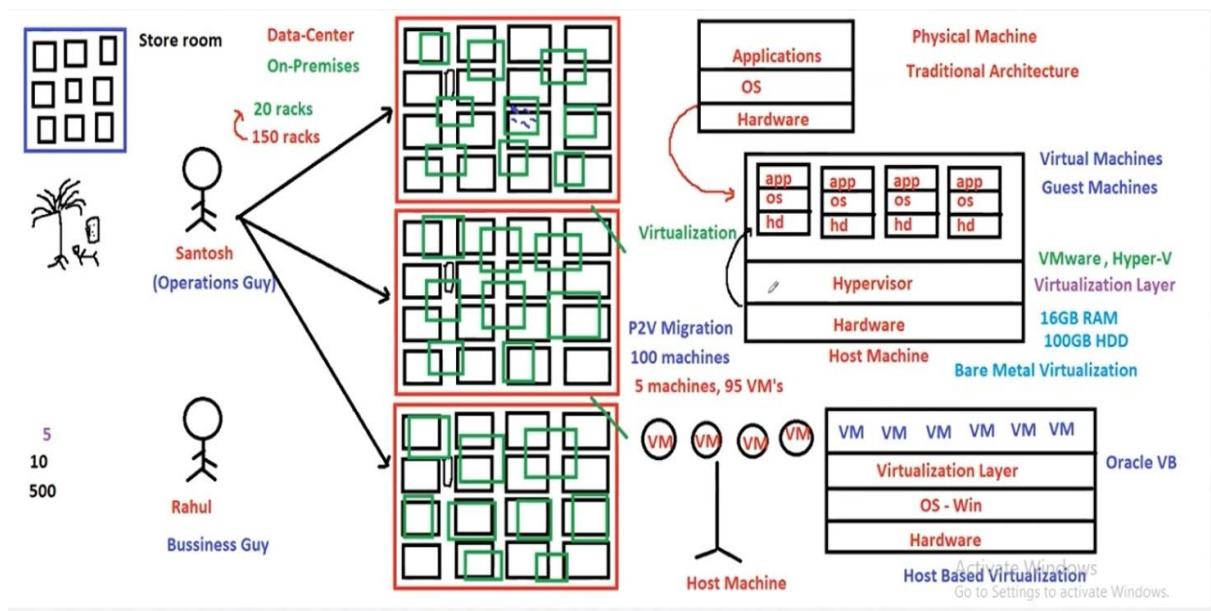
Session Layer

Transport Layer ([TCP](tcp://www.google.com))

Network Layer ([IP](ip://www.google.com))

Data Link Layer

Physical Layer



AWS has Global Infrastructure

AWS is providing Infrastructure as a Service

AWS has DataCenters

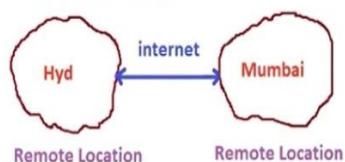
Cloud is present in the remote location

Remote Location = DataCenter

We need internet to connect to the Cloud

AWS is a Cloud Provider, who provides infrastructure as a Service

Amazon Web Services



AWS is Public Cloud Provider who provides Infrastructure as a Service

Key-Words

Virtualization, Host Machine, VM's, Infrastructure, DataCenter, Load Balancer, FireWall Protocols, Cloud, Remote Location

Physical DC → Virtualization → Cloud → AWS (remote location(DataCenters))

Cloud Computing

Instead of doing computing on local machine / on-prem, you will be now doing computing on remote location(Cloud) that is called Cloud Computing

Deployment Models (Types of Cloud)

Public Cloud : The Provider Services which are accessed by everyone Like EX: AWS , Azure, GCP

Private Cloud : The Services which are accessed with in the organization like Ex: IBM

Hybrid Cloud : The Combination of Public and Private Cloud

Community Cloud : It is same as Private Cloud, but can be accessible from few organizations

Service Models

Infrastructure as a Service (IAAS)

Platform as a Service (PAAS)

Software as a Service (SaaS)

Activate Windows
Go to Settings to activate Windows.

AWS is a group of Services

Virtual Machine = Server

EC2 = Elastic Compute Cloud

Server = Instance

EC2 instances

Shared Model
Customer has full control on the Servers

URL
java, .net
Application
Sagar
Client

Customer's Responsibility

Provider

Application
Data
OS
Virtualization
Network, DC

Console
Provider
IAAS
AWS

VM	VM	VM	VM
Hypervisor			
Hardware			
Physical Host Machine			

AWS doesn't have any access inside VM
ElasticBeanstalk (PAAS)

Easy and Quick deployment of application in AWS

PAAS

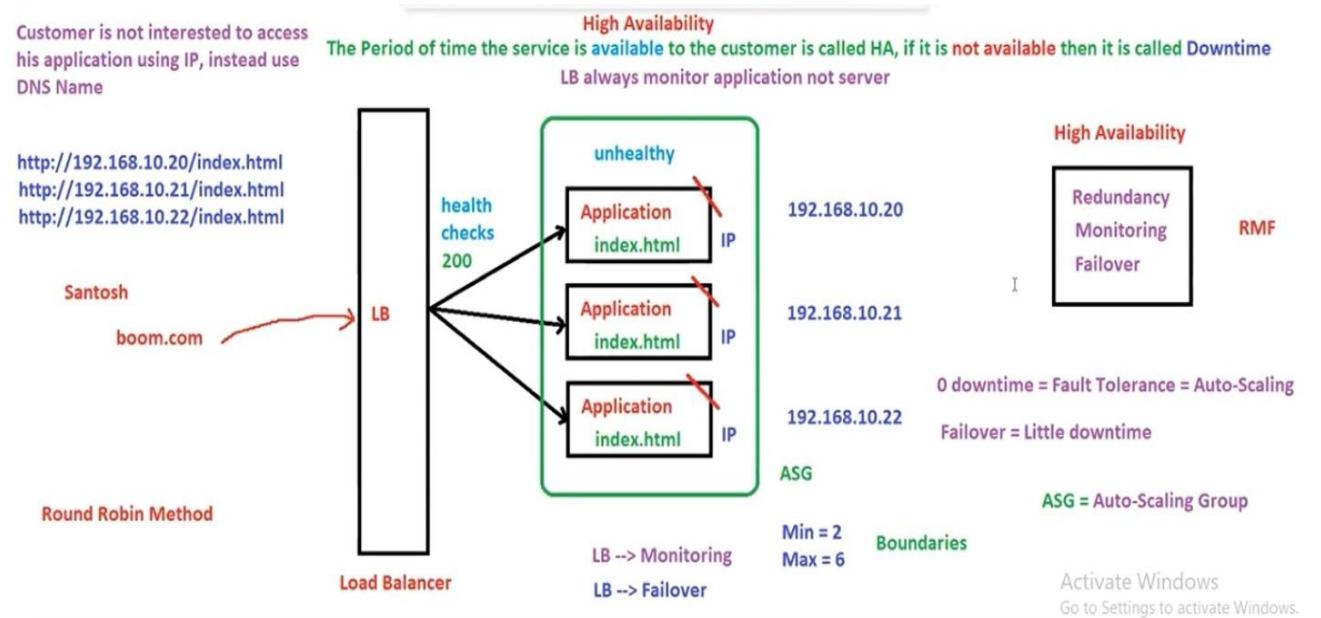
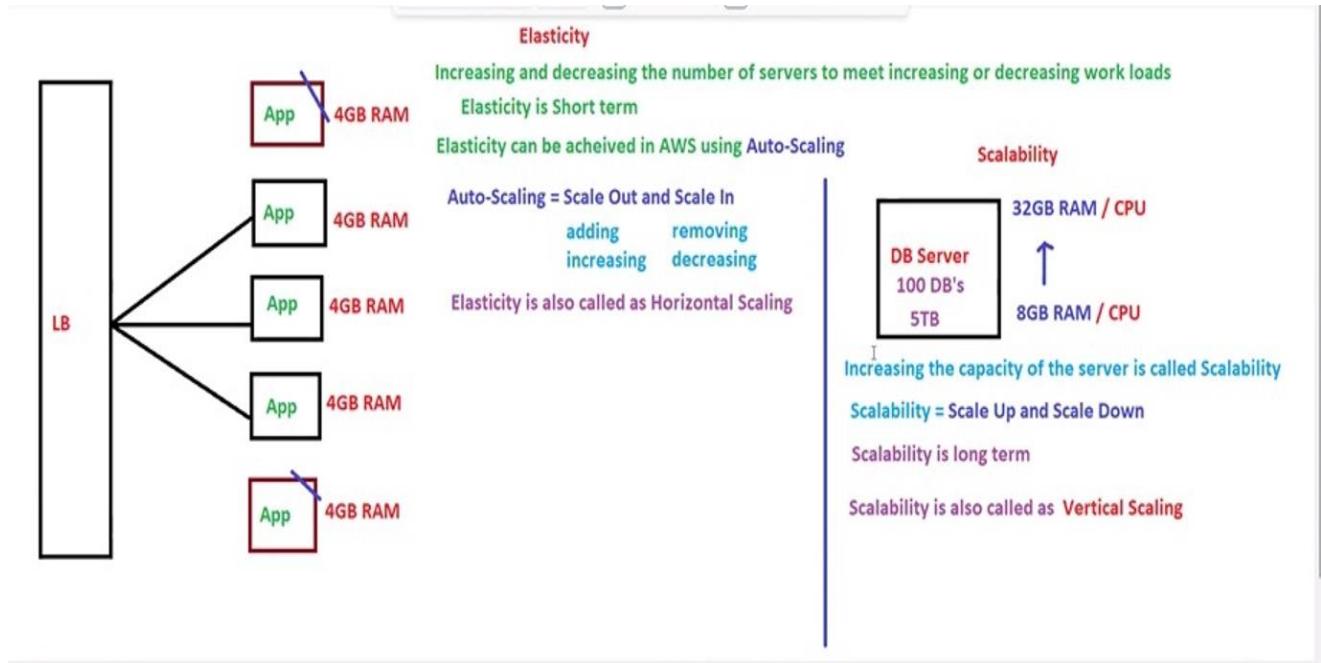
Azure

Application
Data
OS
Virtualization
Network, DC

Gmail, zoom, dropbox

SAAS

SalesForce



AWS has Global Infrastructure

Servers = Instances

Region is a place where AWS has its own infrastructure

A Region has multiple DataCenter

A Region has multiple AZ's

Servers/Instances are placed in AZ's

AZ's are in sync with each other(network), but not data

Best practise is to distribute the instances across multiple AZ's

Very very less chance that 1 AZ goes down

AZ = group of DC's

1a

1b

andheri,bandra,dadar thane, kurla,panvel

Regions and Availability Zone

Region = Its a geo-graphical area , Ex : AWS-Region =Mumbai

Availability Zone = Its a simply a DataCenter (AZ)

Mumbai = ap-south-1

AZ's = ap-south-1a

= ap-south-1b

= ap-south-1c

Regions and AZ's are managed by AWS

AZ's can communicate with each other by default

AZ's networks are inter-connected

Low Latency = Good

High Latency = Bad

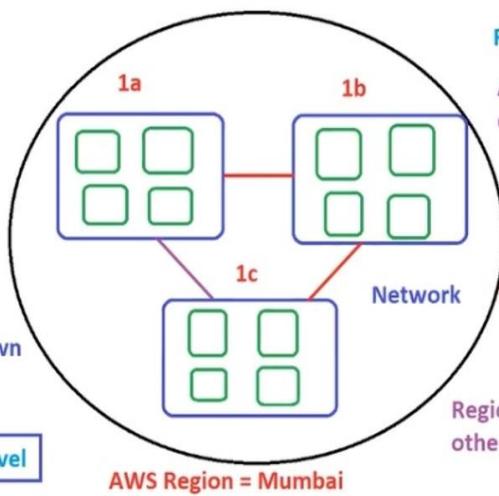
Network

default

Ireland

Regions dont communicate with each other by default, if required yes

Activate Windows
Go to Settings to activate Wi-Fi



Mahesh

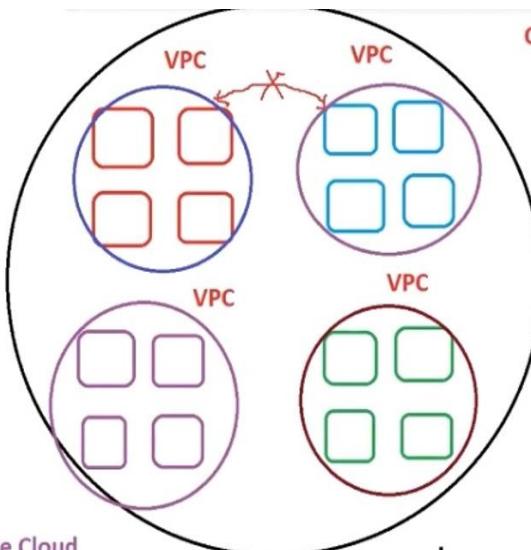
Srinu

Santosh

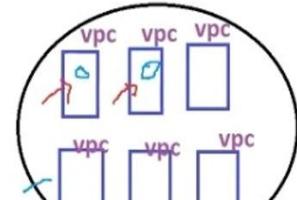
Sagar

Virtual Private Cloud

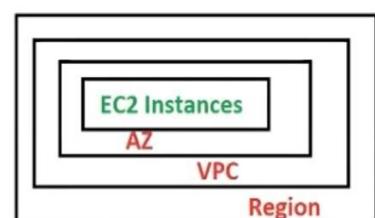
VPC



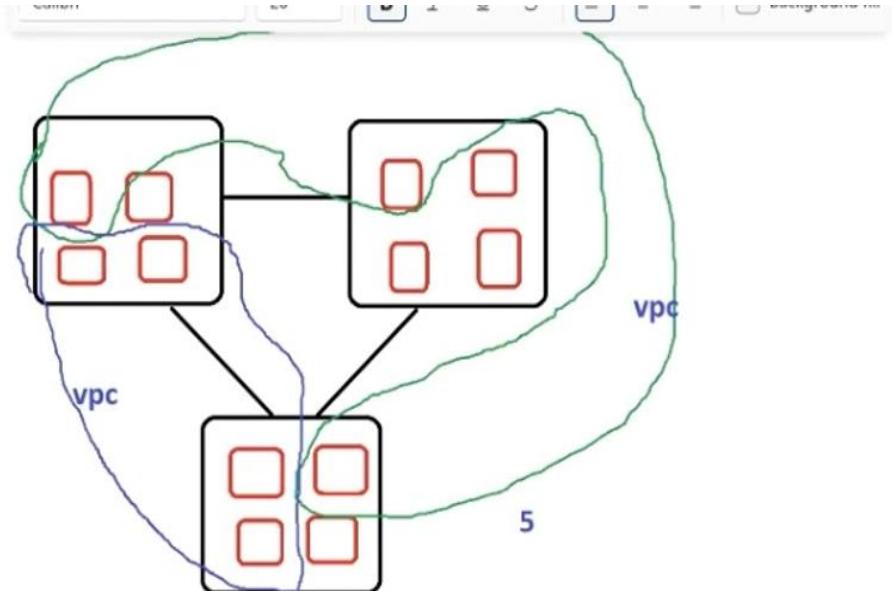
One Region can have multiple VPC's (Max 5 VPC's)



Sulabh Complex



Activate Wi-Fi
Go to Settings to activate Wi-Fi



VPC spans across AZ's

Load Balancer distributes the traffic to multiple EC2 instances across multiple AZ's

<p>AWS Services can be Regional or Global</p> <p>EC2 is Regional</p>	<p>EC2 = Elastic Compute Cloud Servers = Instances / EC2 Instances (VM) Load Balancer = Which distribute the traffic to the Servers Elastic Load Balancer(ELB) = ELB distribute the traffic to multiple EC2 instances across multiple AZ's ELB is completely managed by AWS (HA, AS, Scalability, Performance etc) ELB is not a Server, it is a Service for us You cannot login to ELB, you can access ELB with the DNS Name ELB doesn't have AZ's. It is created regional level</p>	<p>In EC2 Service, We can launch EC2 instances</p>
---------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------

elasticBeanstalk = Easy and Quick deployment of applications in AWS
In General, in PaaS --> You don't have any control on the Servers
In AWS Beanstalk --> You have full control on the EC2 instances launched by Beanstalk
Beanstalk handles EC2 instances(OS) behalf of us

lightSail = If you want to setup and create a virtual private server which already has everything installed (Wordpress, Gitlab, node.js, joomla, drupal, Django, Ghost, Nginx, Redmine etc) --> No Auto-Scaling

HA
Auto-Scaling
Scalability

LB → App → DB
On-Premises

App Client
java

EC2: Launch EC2 instances
Configure and deploy application
Beanstalk
Just Select the Platform

Tomcat
.NET
Docker
Python
GO

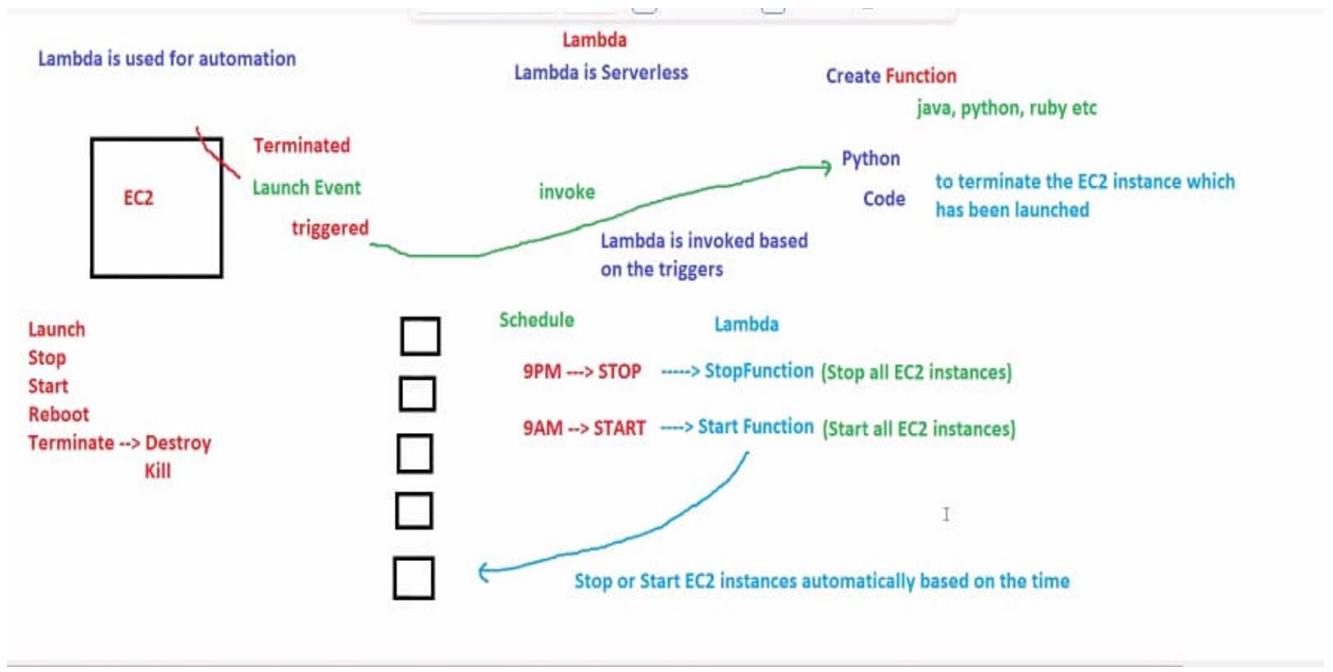
App
EC2

DNS Name

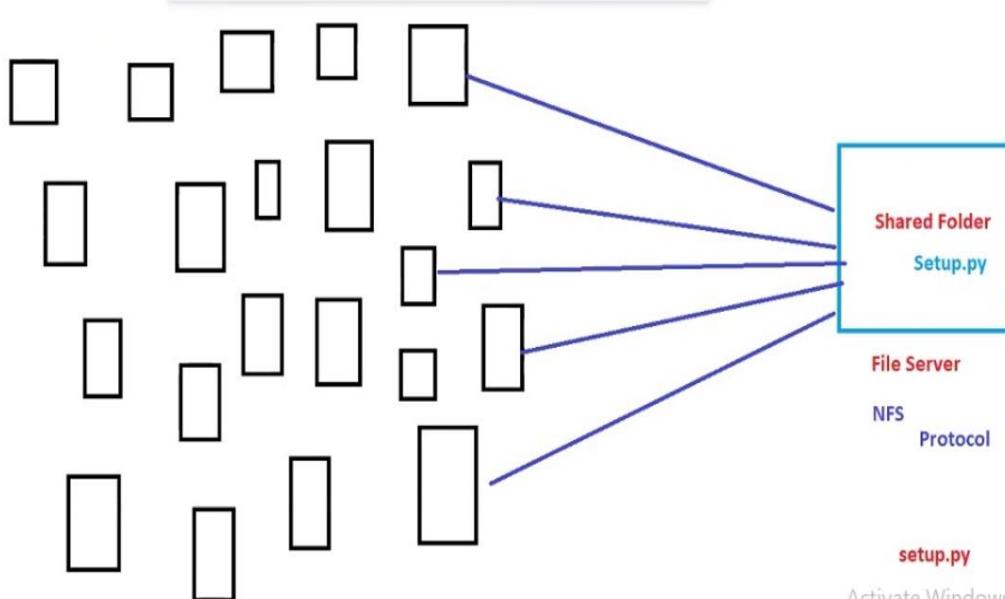
Scale OUT
Scale IN

ELB

EC2 1a
EC2 1b
EC2 1c



In AWS, all services will start with Simple and end with Service	S3 -Simple Storage Service S3 is unlimited Storage by AWS S3 is used to just store your files S3 can store all FLAT files With S3, you can upload, download and access your files	Floppy --> 2MB CD's --> 700MB DVD's --> 4.7GB Pen Drives --> 128GB Hard Disks --> 2TB
SNS --> Simple Notification Service SES --> Simple Email Service		
S3 is a Object Based Storage	You cannot execute any files in S3 Is it possible to install OS in S3? NO Is it possible to install DB in S3? NO Is it possible to run .net, .py etc in S3 ? NO	S3 is Global Buckets are regional
Windows	S3	AWS handles HA, Performance, Scalability etc for S3
Folder	Bucket	Bucket is a container for Objects
Files	Objects	Object is a File Key is the File name / Name of the Object
boom.mp4	KEY	S3 supports STATIC website hosting (HTML files)[Create a bucket, upload all files, enable static website hosting] (No need to worry about HA, performance, etc handled by S3)



Activate Windows
Go to Settings to activate Windows.

S3 is Object Based Storage
EBS is Block Based Storage

1a

EC2 Instance

C-Drive D-Drive

1a OS 1a Additional vol

Root Vol Additional vol
Device Name
/dev/sda1 /dev/sdb, e, f, g etc
/dev/xvda xvdf, xvdg , etc

Root Volume is always mounted / attached as /dev/sda1

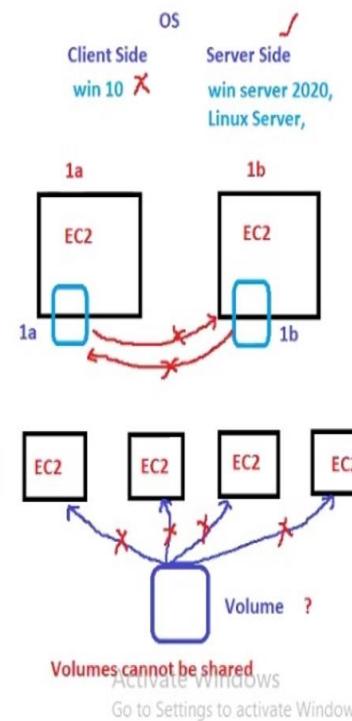
EBS - Elastic Block Storage
Hard Disk = Volume = EBS Volume
Volumes can be attached and detached
You can attach multiple volumes to the EC2 instance
EC2 instance has a default volume and that is called ROOT Volume
The Root Volume always contain OS (Win, Linux, Mac)
EC2 supports only Server side OS not Client side OS
If you have OS on the Volume, that volume is called ROOT Volume
EC2 instance can have only 1 ROOT Volume
EC2 instance can have multiple additional Volumes
Max size of each EBS Volume is 16TB
You cannot attach a volume to multiple EC2 instances at the same time
Volumes should be pre-provisioned like 50GB, 100GB ... max 16TB
Volume size can be increased on FLY (no need to stop the EC2 instance)
Volume size cannot be decreased

Is it possible to detach the ROOT Volume while EC2 is running? NO: Stop the EC2 instance first and then detach the ROOT Volume
Is it possible to detach ADDITIONAL Volume while EC2 is running? YES: it is not recommended to detach while running, stop first

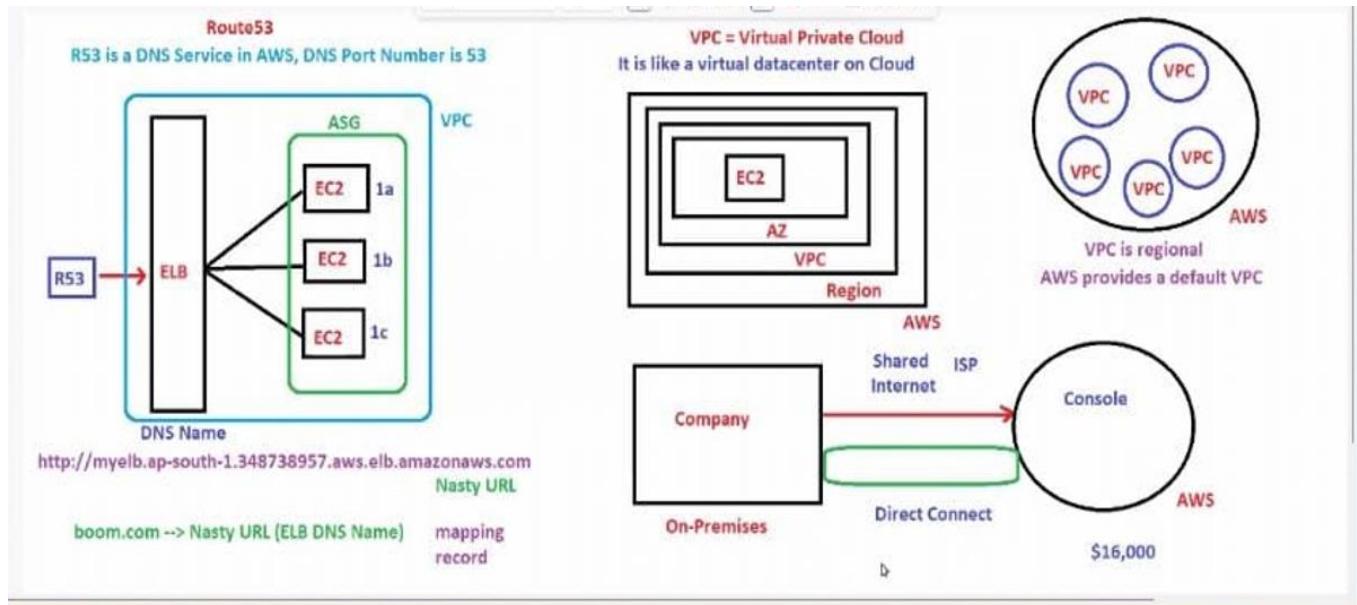
EC2 instance has AZ, Volume also have AZ
EC2 instance and Volume should be in same AZ

We cannot attach 1a volume to 1b EC2 instance (diff AZ)
We can attach 1a volume to 1a EC2 instance (same AZ)

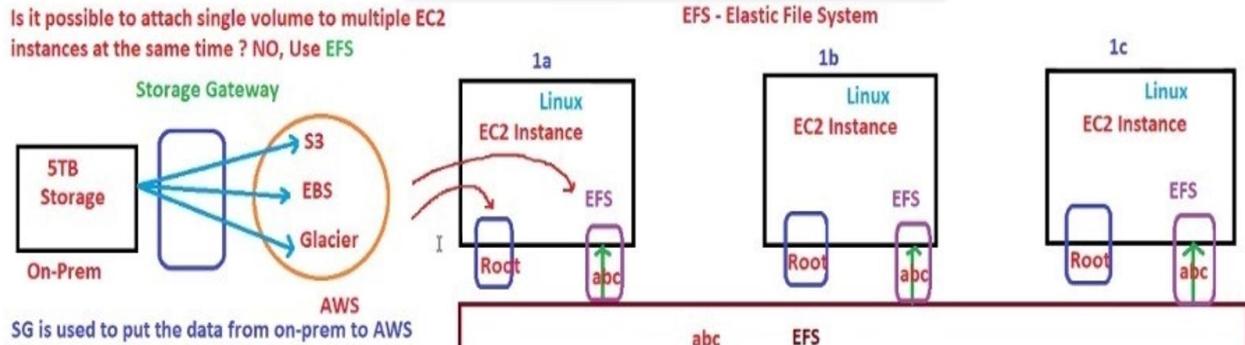
Activate Windows
Go to Settings to activate Windows.



Route 53



Elastic File System



EFS is completely managed by AWS

EFS is only for Linux EC2 instance

Fsx is for Windows EC2 instance

EFS works with NFSv4 Protocol

EFS is File based Storage

EFS is unlimited Storage

EFS doesn't require any pre-provisioning (it will automatically increase and shrink based on the data you put in EFS)

EFS can be mounted to multiple EC2 instances at the same time across AZ's

Glacier = Archive Purpose

Glacier is cheaper than S3

SNOW FAMILY

SnowCone --> 8TB

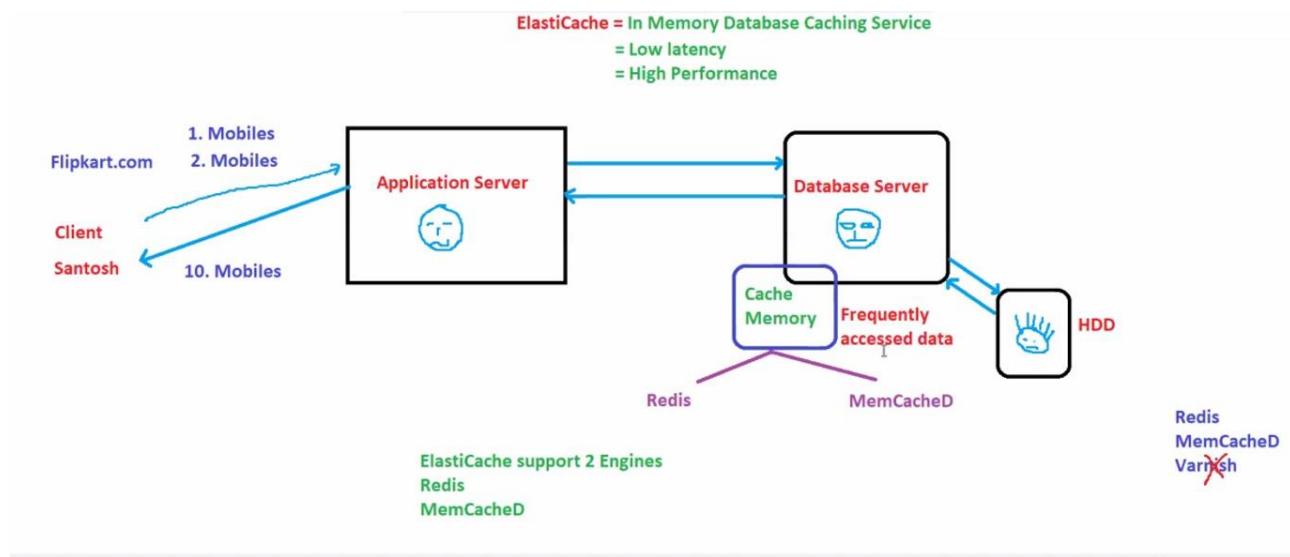
SnowEdge --> 100TB ---> S3

SnowMobile --> PB's (Truck)

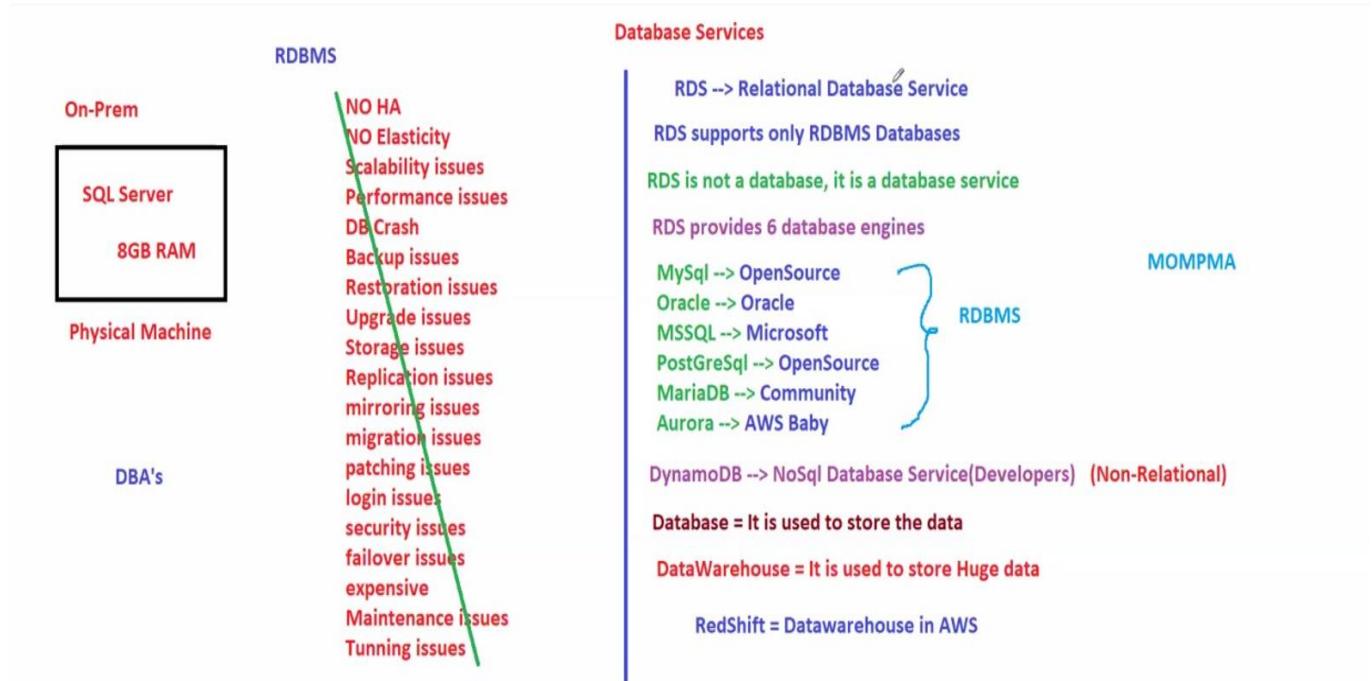
Snow Family is used to transfer huge data from On-prem to AWS and Vice versa

SnowFamily is a Physical Transfer

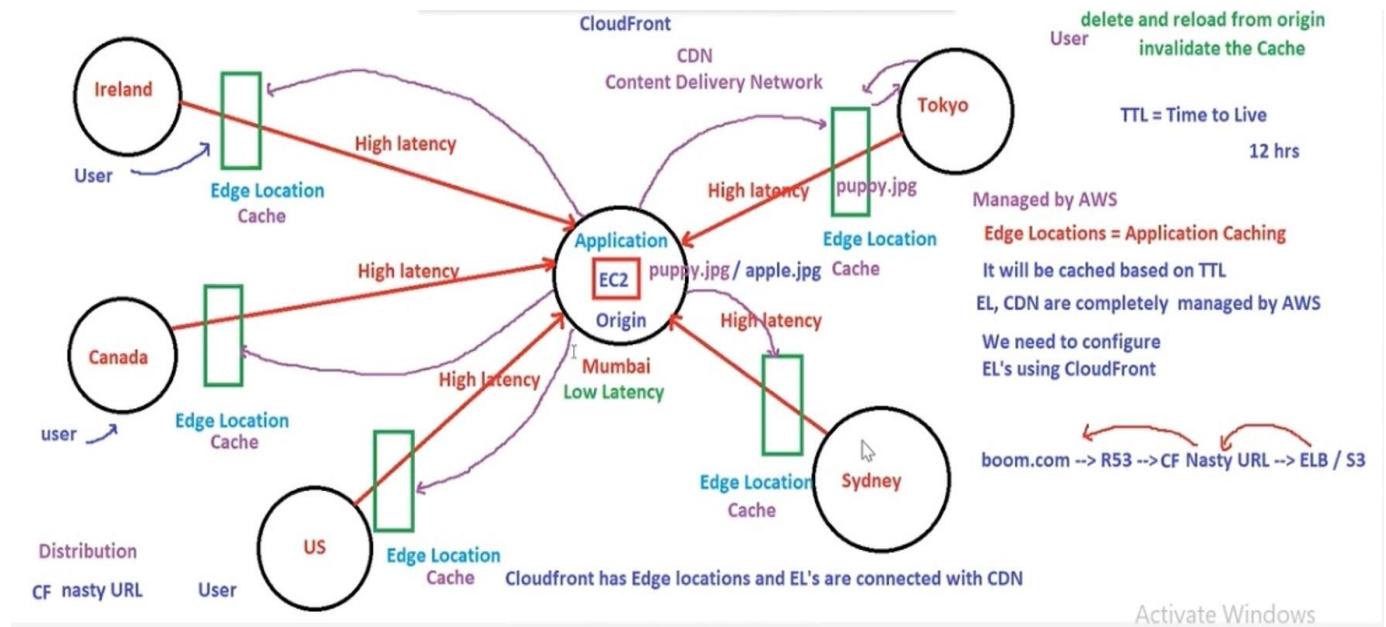
ElastiCache



Relational Database Service :

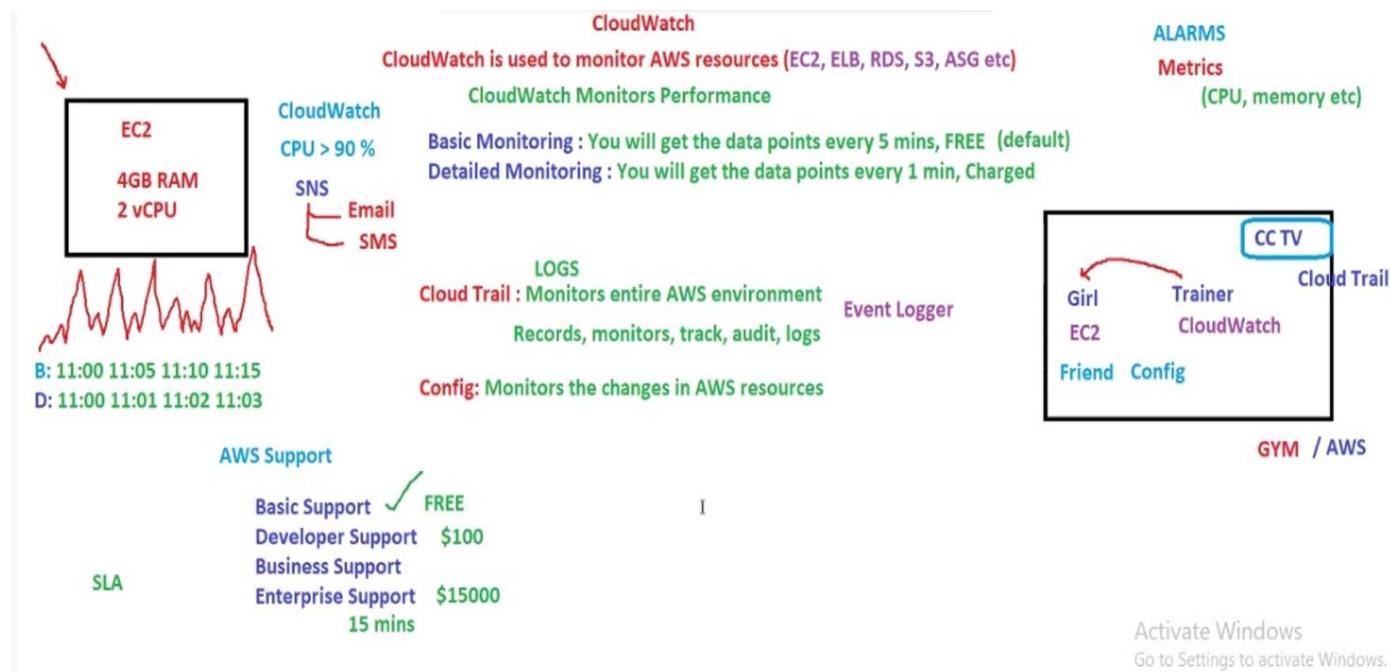


CloudFront :

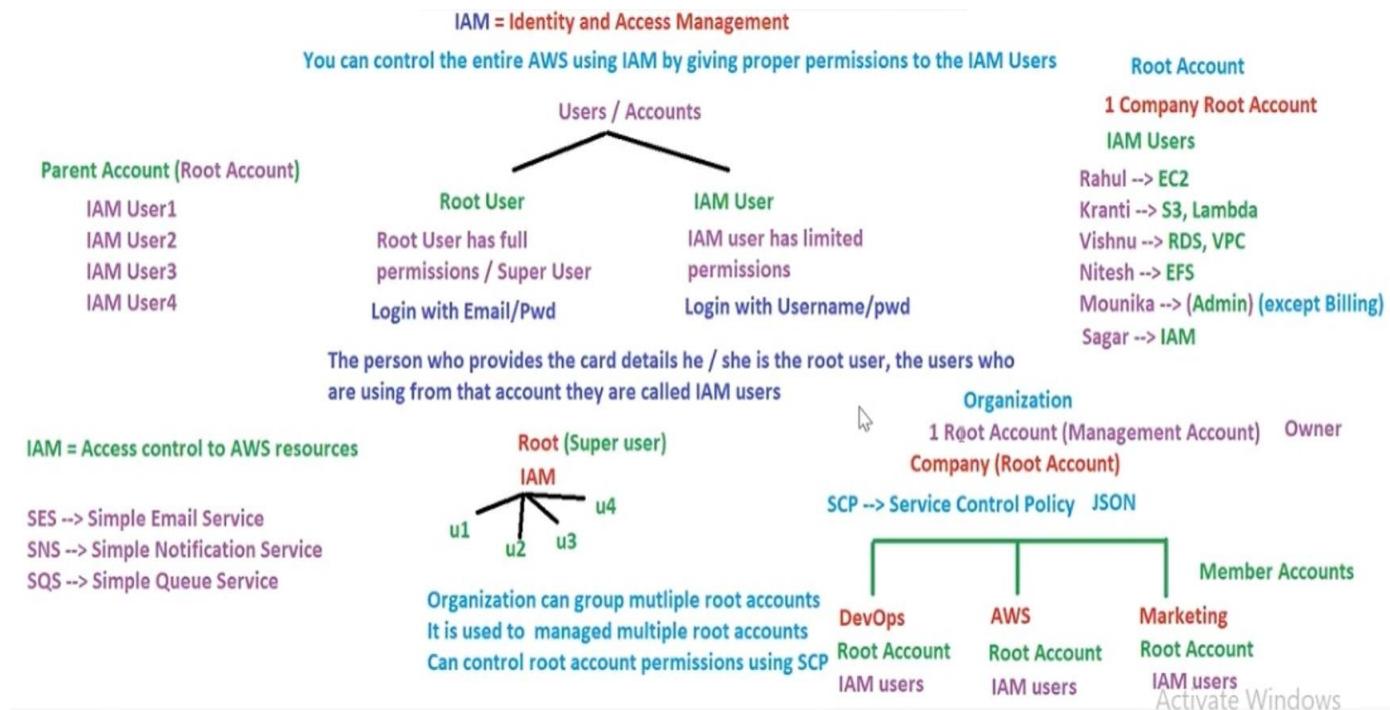


Activate Windows

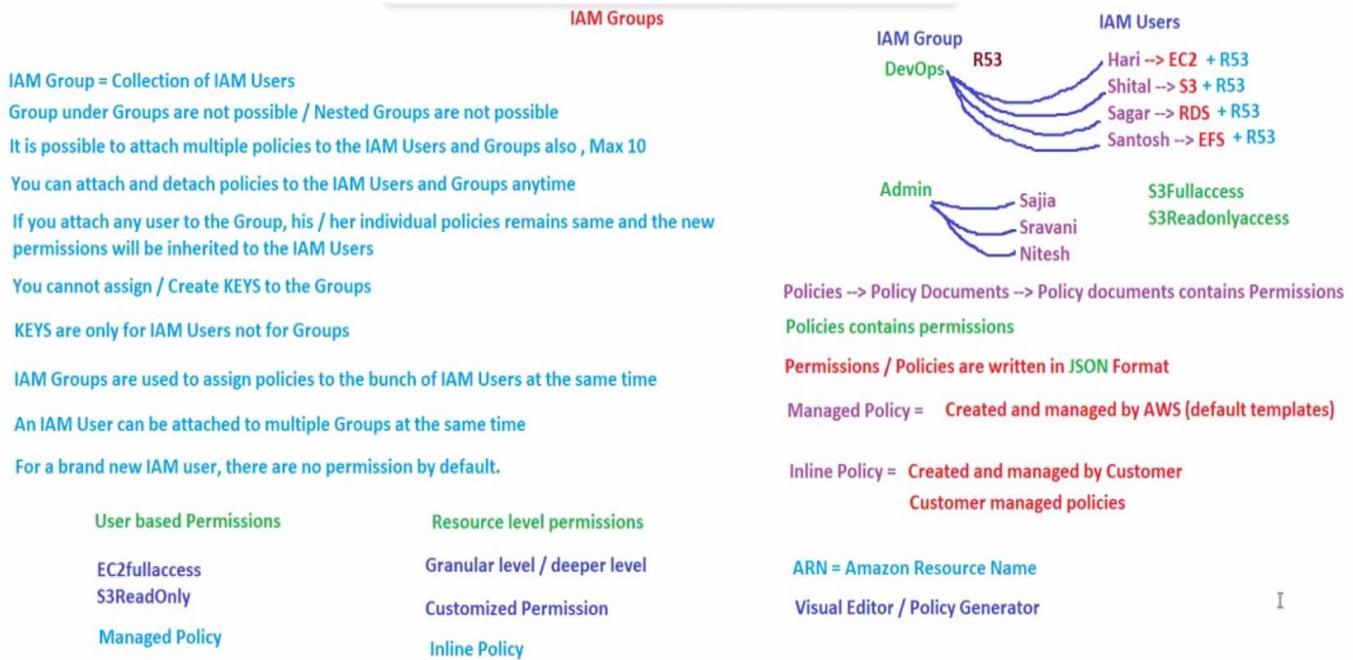
CloudWatch :

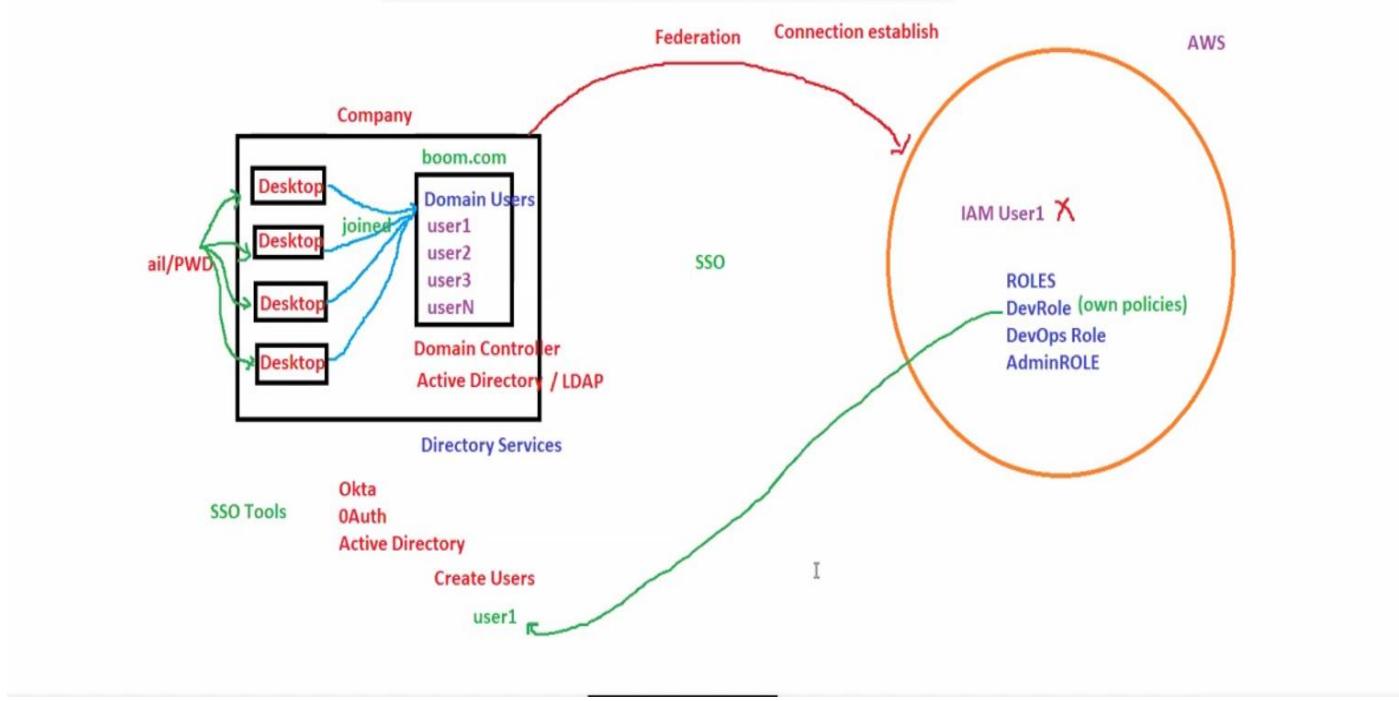


IAM

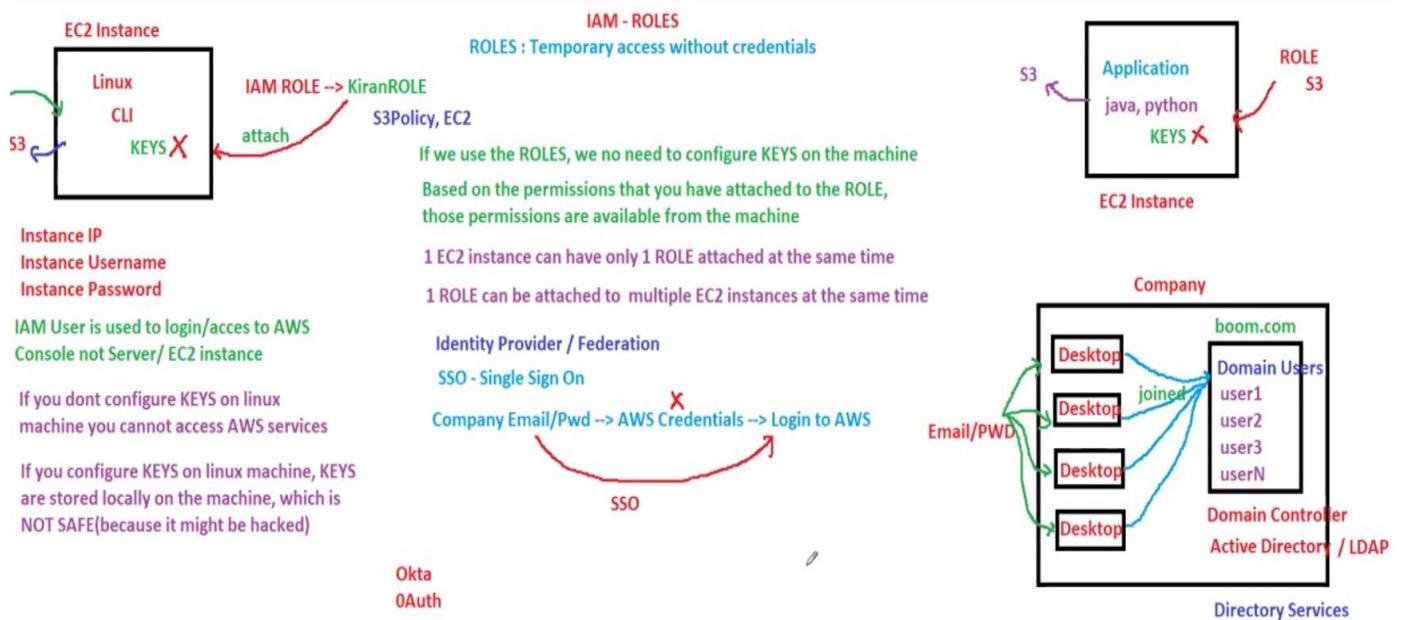


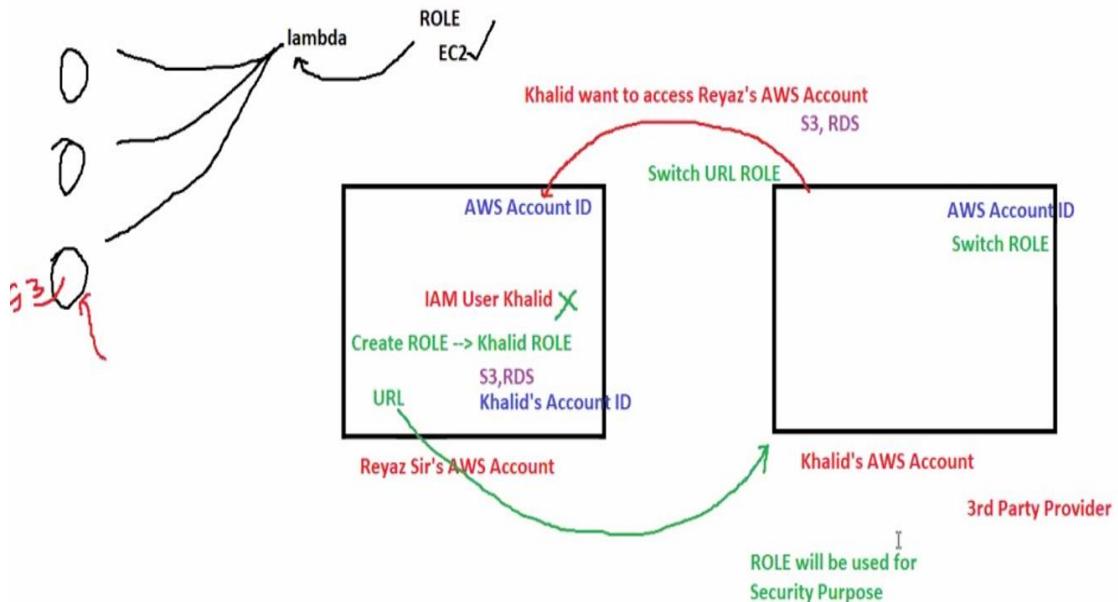
IAM Groups





IAM-ROLES





Elastic Compute Cloud

EC2 is Regional

Servers = Instances

750 hours / per month

1 Year FREE

On-Demand Instances

Fixed Price (Hourly)

Pay for what you have used

No Commitments

No Upfront payments

No Predictable Usage

Reserved Instances

Long Term Commitment

1 or 3 years

Upfront Payment(Full, partial)

75% discount on hourly price

Predictable Usage

Standard RI = 75% discount

Convertible RI = To change the capacity of the instance anytime

Scheduled RI = Reserve it for fraction of a day , a week , or a month

Savings Plan

EC2 - Elastic Compute Cloud

EC2 is a webservice from AWS that provides resizable compute services in the Cloud

Resizable = Scale Up / Scale Down , Scale Out / Scale In

Scalability

Elasticity

Pricing Models in AWS

SPOT Instances

Bidding

Auctioning

Huge Capacity for cheaper price

90% discount

Dedicated Host

If any customer need dedicated physical machine

Launch

Running --> Bill

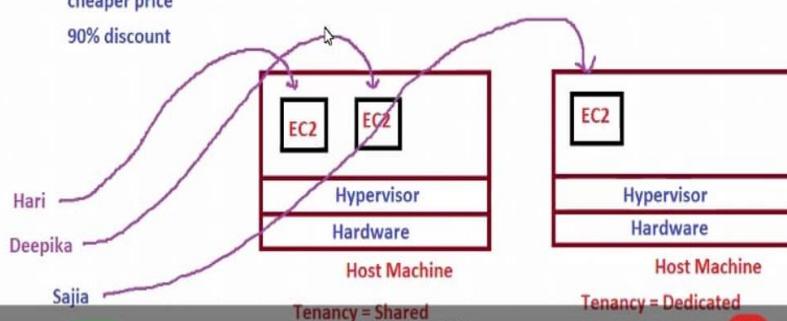
Stop --> No Bill

Start

Reboot --> No Bill

Terminate --> Kill / Destroy

No Bill



EC2-Families

General Instances = For General Purpose	EC2 - Families / Instance Types	Scalability = Scale Up and Scale Down
Memory Instances = If you need more memory for your applications	General Instances	Scalability can be achieved by changing the instance type
CPU Instances = More CPU	Family Members = Instance Types	Anytime = Scale Up → No Data Lost
Storage Instances = For More Storage	t2.nano = 0.5GB RAM, 1 vCPU	Anytime = Scale Down → No Data Lost
GPU Instances = Graphics, Heavy Machines	t2.micro = 1GB RAM, 1 vCPU	You should STOP the EC2 instance to change the instance type (downtime)
Every family has their own different instance types	t2.small = 2GB RAM, 2vCPU	If you change the instance type, data in the instance will NOT BE LOST
	t2.medium = 4GB RAM, 4vCPU	
	t2.large = 8GB RAM	
	t2.xlarge = 16GB RAM	

EC2
2GB RAM
t2.small
t2.large

Burstable Performance Instances
Billable

Application
1GB RAM
1 vCPU
6 CPU Credits

EC2 instance will enter into burstable mode and give high performance for limited period of time only

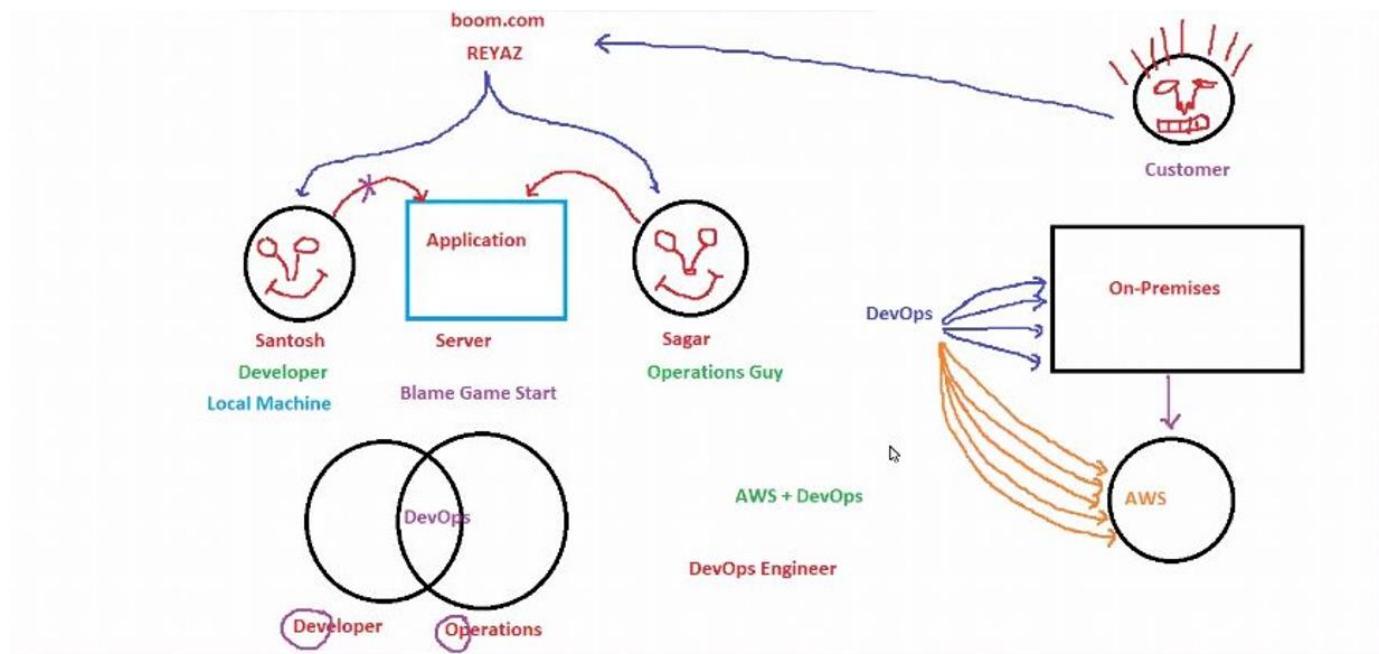
CPU credits will happen based on instance type by AWS

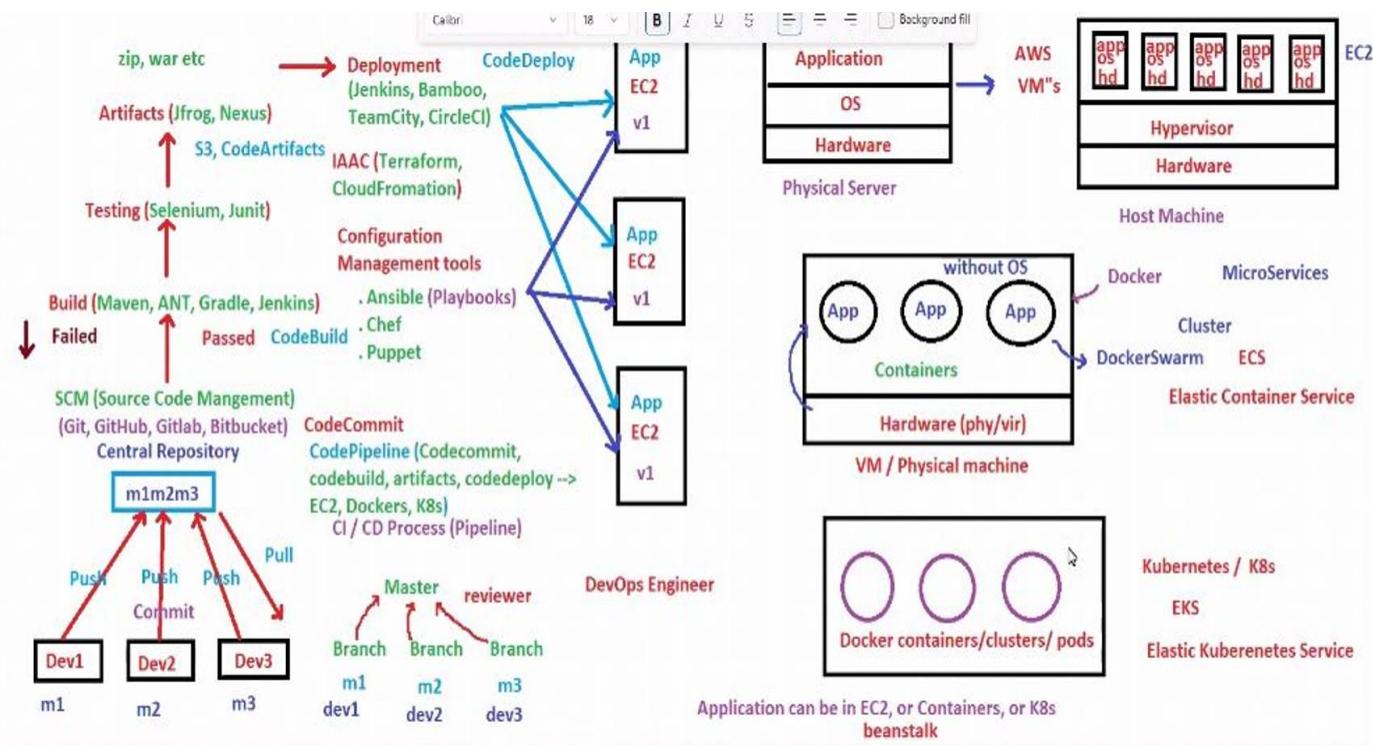
t2 and t3 instance type

All instance families

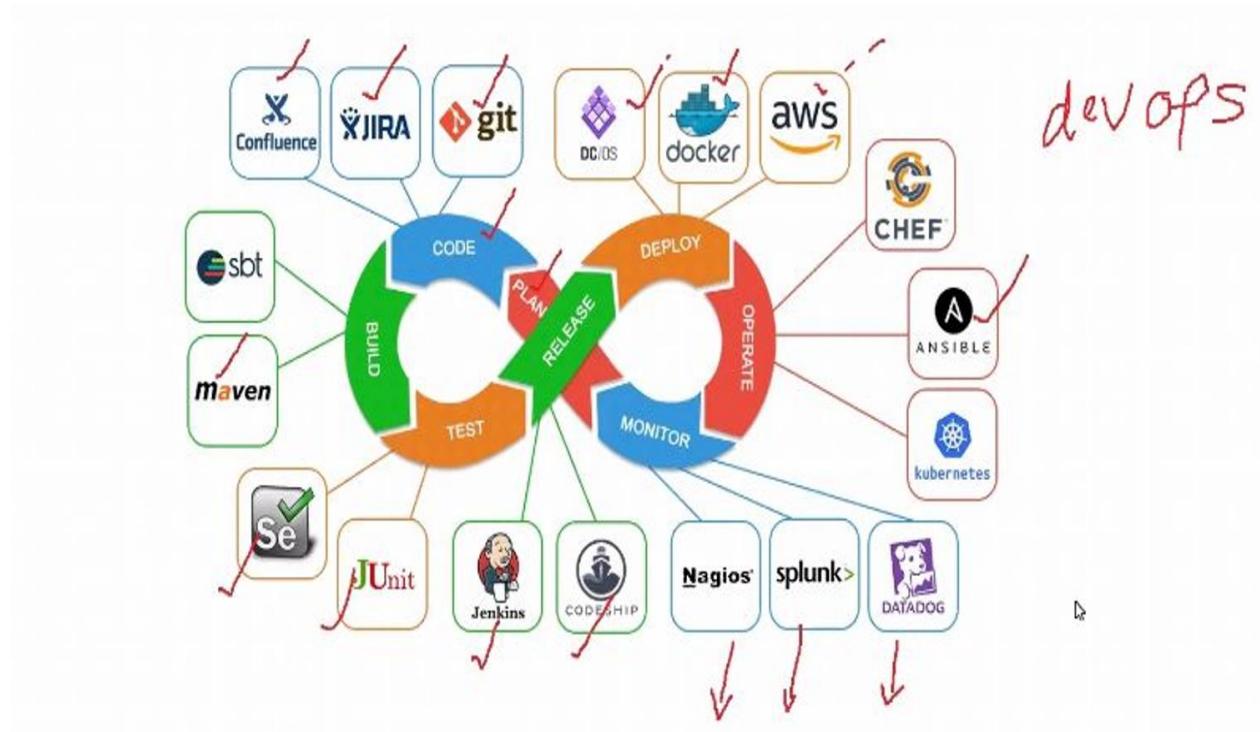
	c5	d3en	inf1	m5n	r5	z1d
t1	c5a	f1	m1	m5zn	r5a	
t2	c5ad	g2	m2	m6g	r5ad	
t3	c5d	g3	m3	m6gd	r5d	
t3a	c5n	g3s	m4	mac1	r5dn	
t4g	c6g	g4dn	m5	p2	r5n	
a1	c6gd	h1	m5a	p3	r6g	
c1	cc2	i2	m5ad	p3dn	r6gd	
c3	d2	i3	m5d	r3	x1	
c4	d3	i3en	m5dn	r4	x1e	

DevOps





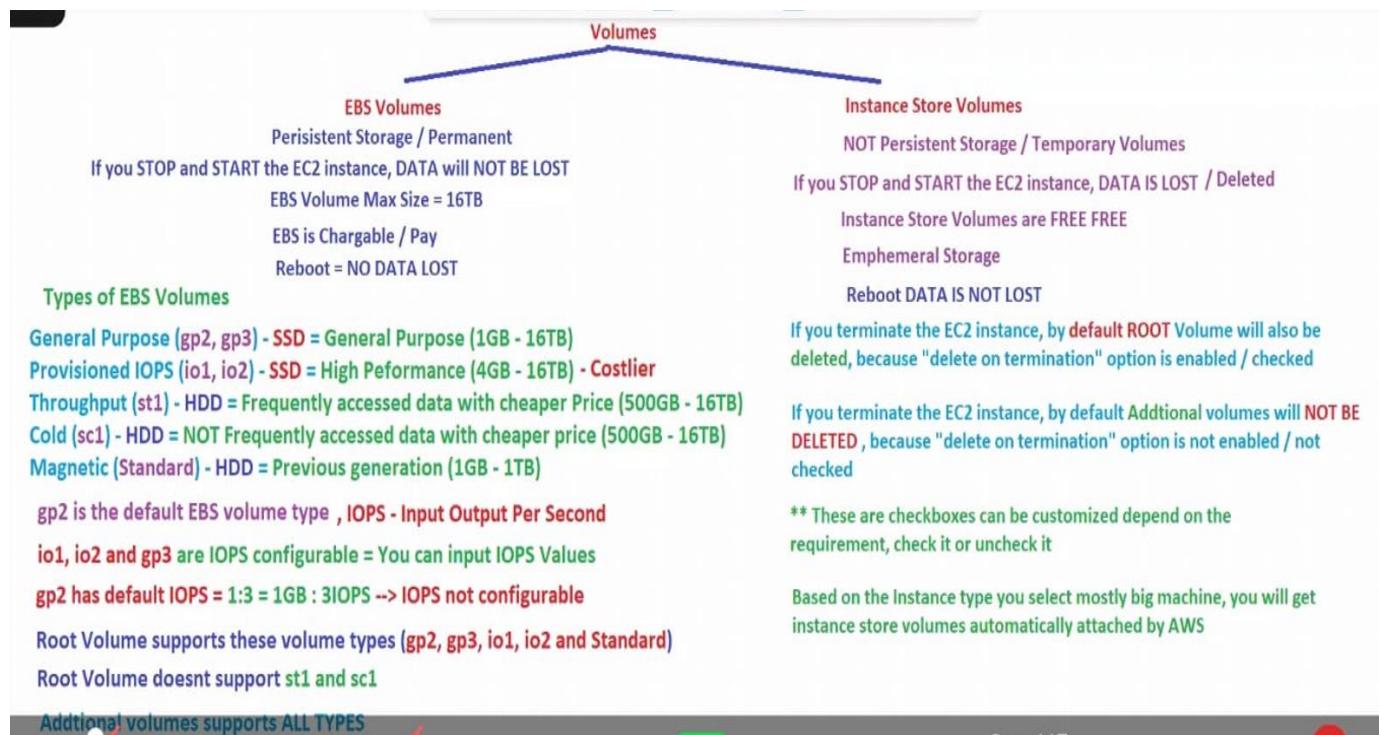
DevOps Tools

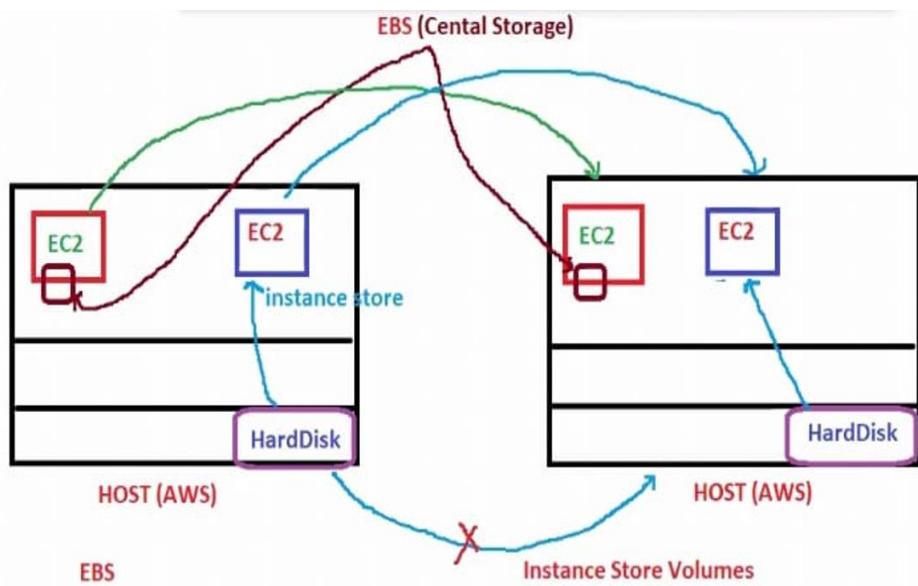


IAM Tags



Volumes





STOP and START --> JUMP --> DATA IS NOT LOST

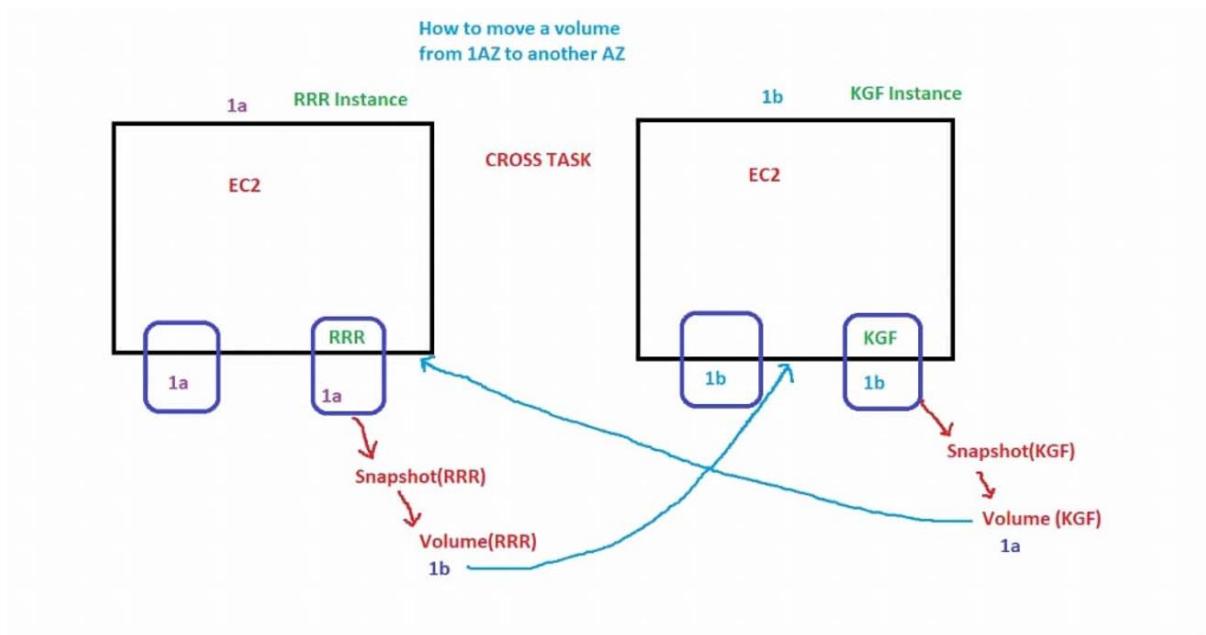
Reboot --> NO Jump --> DATA IS NOT LOST

Terminate --> DATA IS LOST

STOP and START --> Jump --> DATA IS LOST

Reboot --> No Jump --> DATA IS NOT LOST

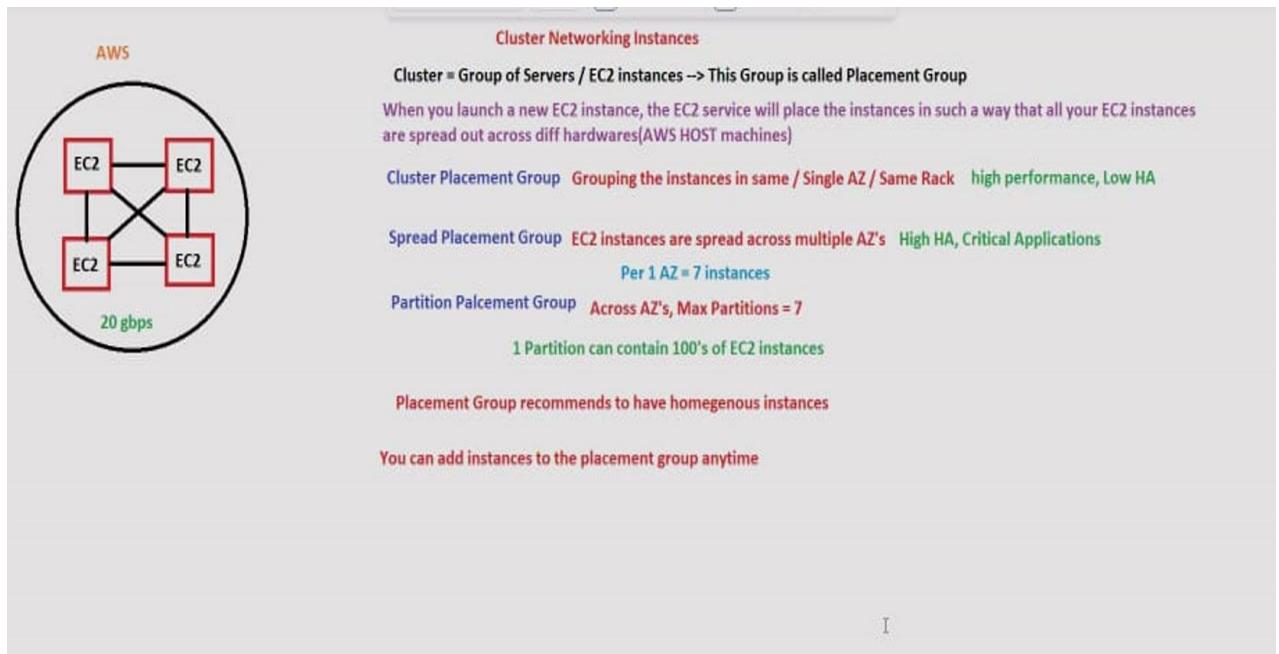
<p>1a</p> <p>EC2</p> <p>50GB</p> <p>C-Drive D-Drive</p> <p>1a 1a</p> <p>OS Data</p> <p>Root Additional</p> <p>Vol 50GB --> Snapshot(50GB) --> abc --> Snapshot (50GB) --> Xabc + def --> Snapshot --> def</p> <p>Incremental backup</p>	<p>SNAPSHOTS</p> <ol style="list-style-type: none"> 1. Snapshot is a point in time copy of the volume 2. Backup of the volume is also called as Snapshot 3. EBS Snapshots are created from EBS Volumes 4. You can create a Snapshot from the Volumes 5. EBS Volume --> EBS SNAPSHOT --> EBS VOLUME 6. You cannot attach snapshots directly to the EC2 instance, you have to create a volume out of snapshot first and then attach volume to the EC2 instance 7. Is it possible to login or use snapshots directly ? NO 8. Snapshots are stored in S3 [Provider's S3] 9. Snapshots doesn't have any AZ's 10. Snapshots are Regional 11. By default, Snapshots are Private, if required you can make it public 12. You can copy the snapshots from one region to another region in the same account 13. Snapshots can be shared from one AWS account to another AWS account using AWS account ID (Private) 14. EBS Volumes cannot be moved directly to any other region, use Snapshots <p>EBS Volumes are created from EBS Snapshots</p> <p>Instance Store volumes are created from a template stored in S3</p> <p>To create a snapshot we no need to stop the EC2 instance</p>	<p>By default, Volumes, snapshots are not Encrypted</p> <p>NOT Encrypted --> NOT Encrypted</p> <p>Encrypted --> Encrypted</p> <p>NOT Encrypted --> Encrypt (Copy Option)</p> <p>All Encryption Keys are stored in KMS (Key Management Service)</p> <p>Encryption and Decryption are completely managed by AWS</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



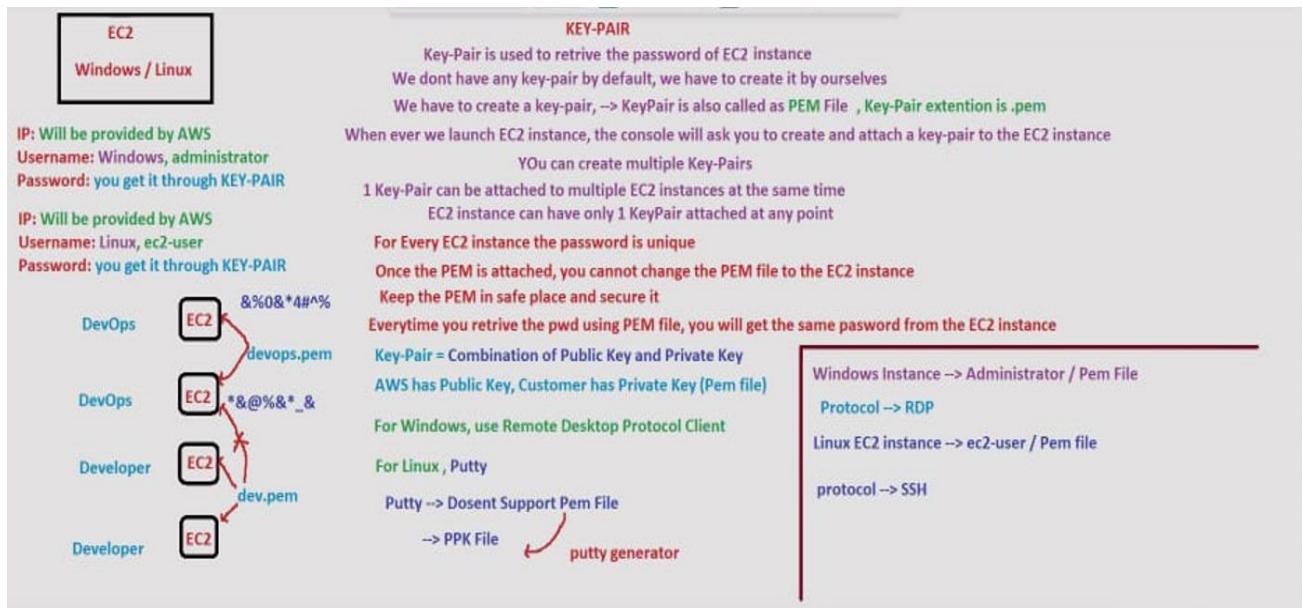
IMAGES

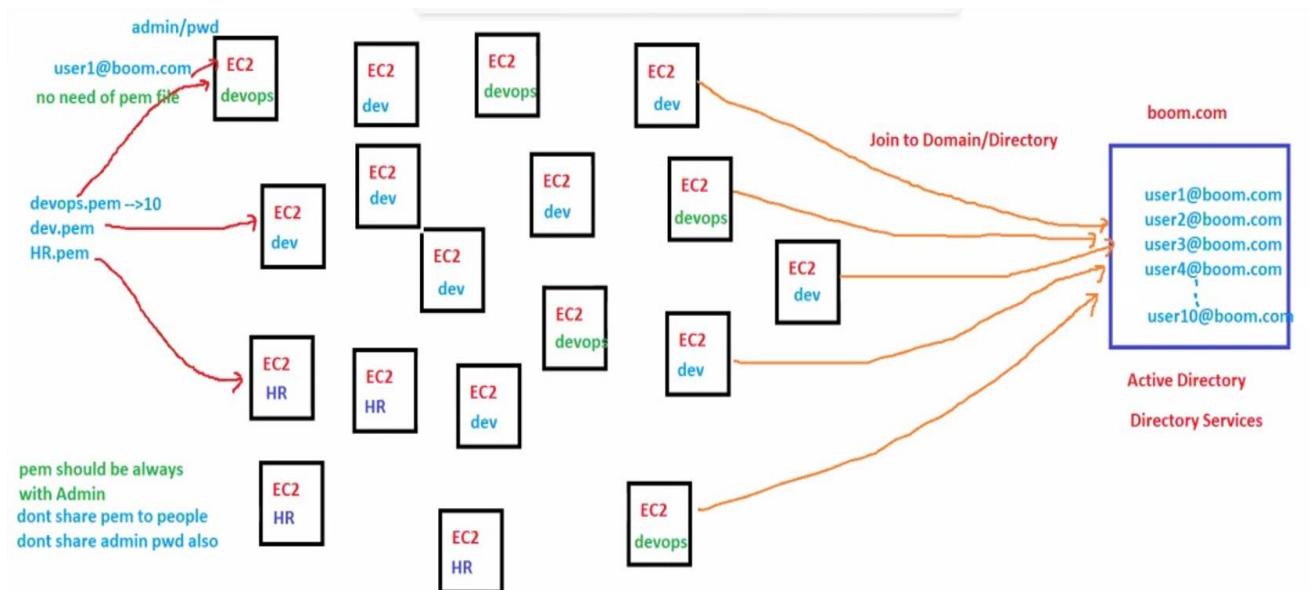
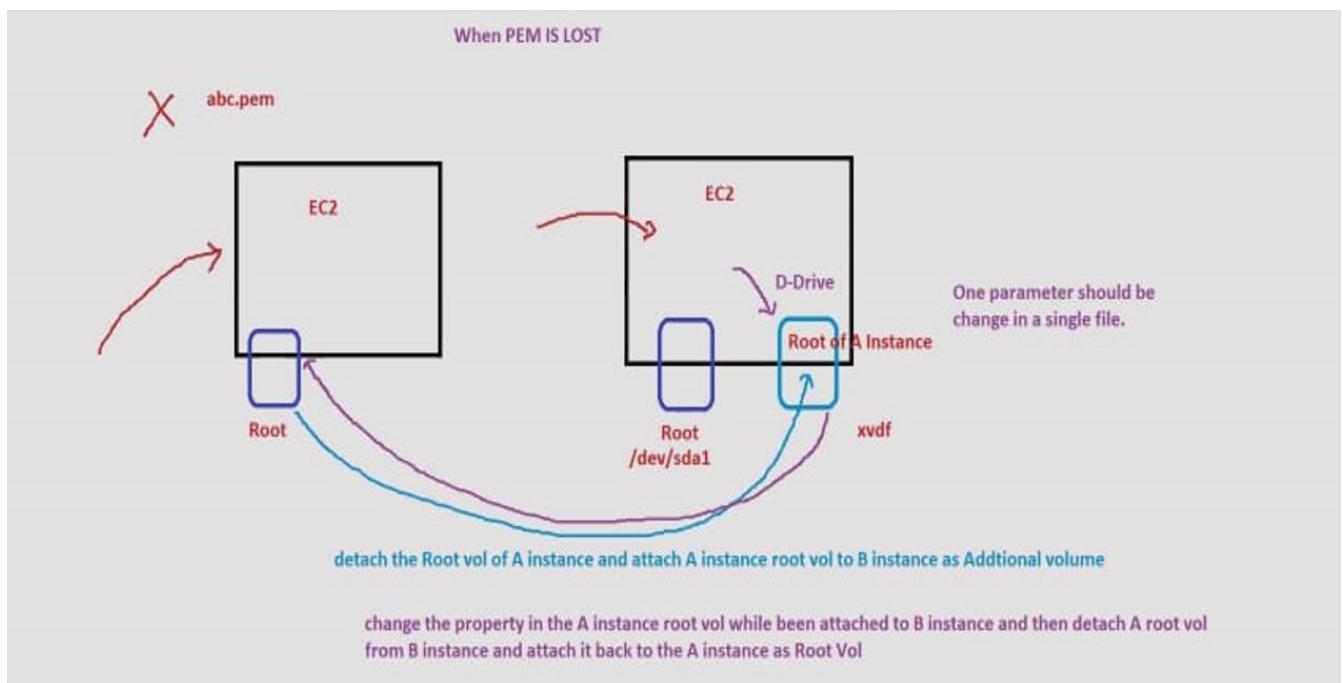
<p>Snapshot → Copy of the Volume AMI → Copy of entire EC2(includes Volumes)</p> <p>AWS AMI → Linux OS → AMI Linux EC2 Tomcat Custom AMI</p> <p>AMI's are stored in S3</p>	<p>IMAGES</p> <p>Copy of the OS is called Image Image = AMI = Amazon Machine Image Template of the OS is called AMI</p> <p>Copy of image includes all configurations that we did on original instance EC2 Instance → Image(AMI) → EC2 instance</p> <p>1 AMI, can be used multiple times to launch multiple EC2 instances AMI's are re-usable, AMI doesn't have any AZ's You cannot directly use AMI to login, instead launch EC2 instance from the image and then login to the EC2 instance By default, AMI's are Private, if required you can make it public AMI's are regional AMI's can be copied from one region to another region AMI can be shared from one AWS account to another AWS Account</p> <p>All public images are located at AWS Market Place</p> <p>Image Contains OS → Root Volume (EBS) → Root Volume (Instance Store)</p> <p>Images are backed by either EBS volume or instance store volume</p> <p>EC2 instance 2 Volumes → Image (2 volumes) Root Additional ↓ 2 Snapshots</p> <p>Snapshot (OS) → Image</p> <p>NO need to stop EC2 instance to create a image (up to you)</p>										
<p>OS</p> <table border="0"> <tbody> <tr> <td>Windows</td> <td>Linux</td> </tr> <tr> <td>win server</td> <td>RedHat</td> </tr> <tr> <td>2012, 2021, 2022</td> <td>CentOS</td> </tr> <tr> <td>Plain OS</td> <td>ubuntu</td> </tr> <tr> <td>Plain OS + Few Apps</td> <td></td> </tr> </tbody> </table>		Windows	Linux	win server	RedHat	2012, 2021, 2022	CentOS	Plain OS	ubuntu	Plain OS + Few Apps	
Windows	Linux										
win server	RedHat										
2012, 2021, 2022	CentOS										
Plain OS	ubuntu										
Plain OS + Few Apps											

Cluster Networking Instances

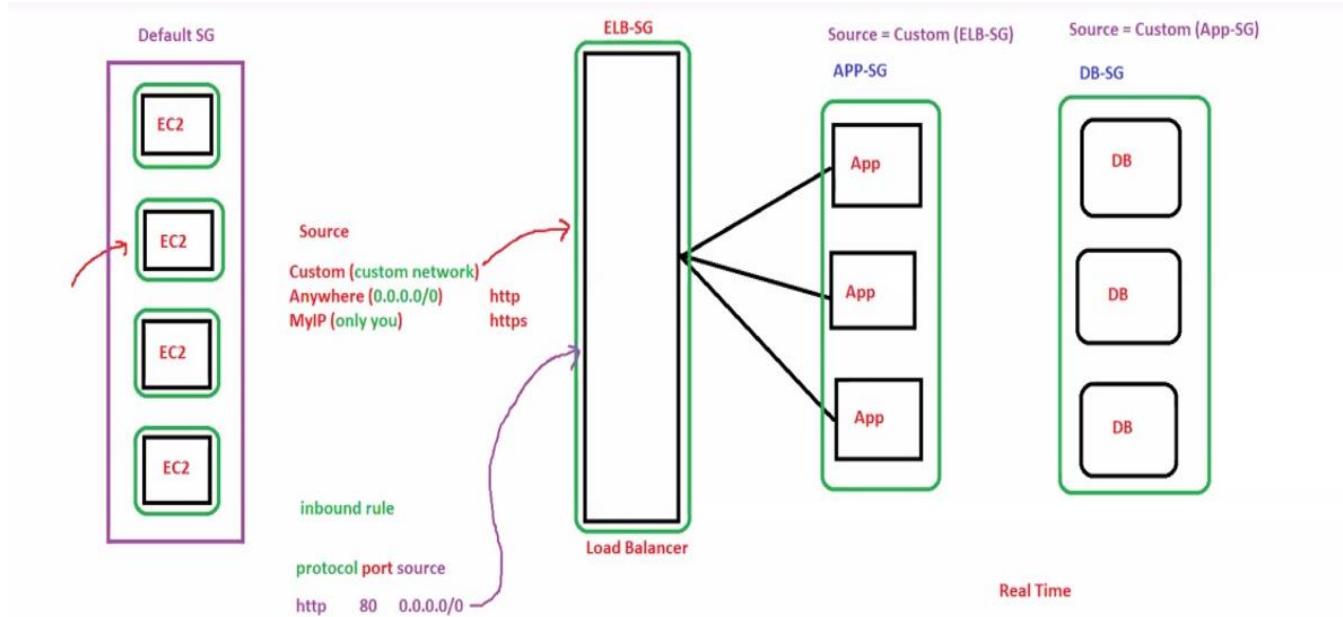
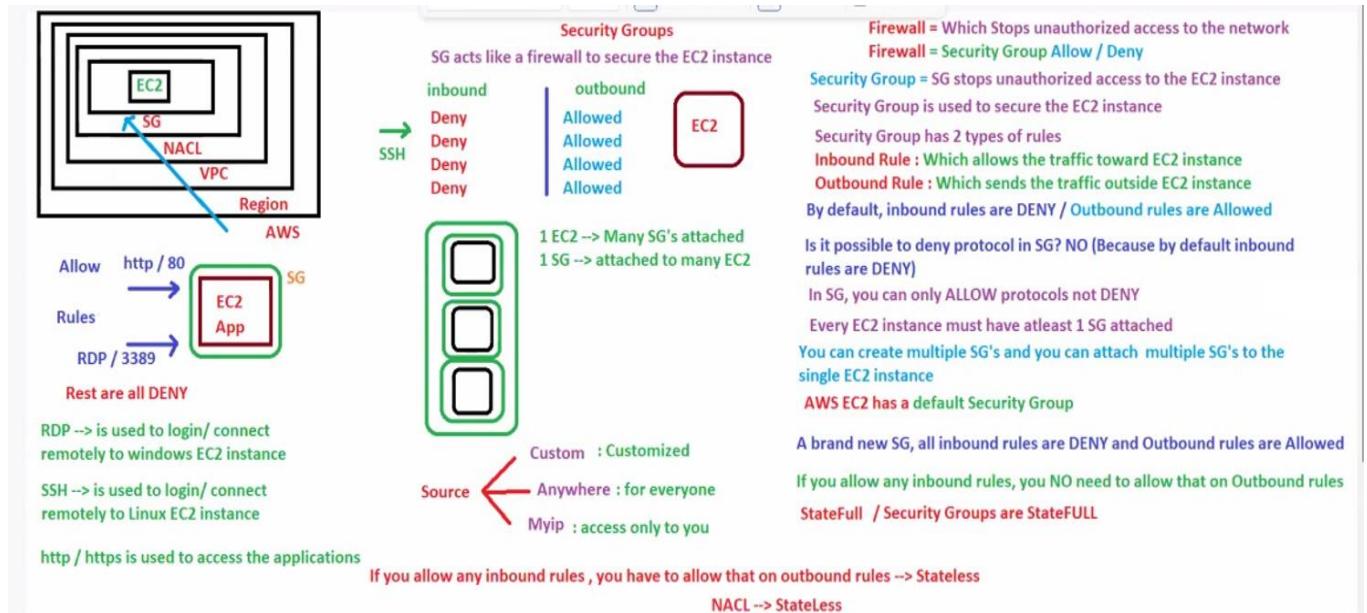


KEY-PAIR

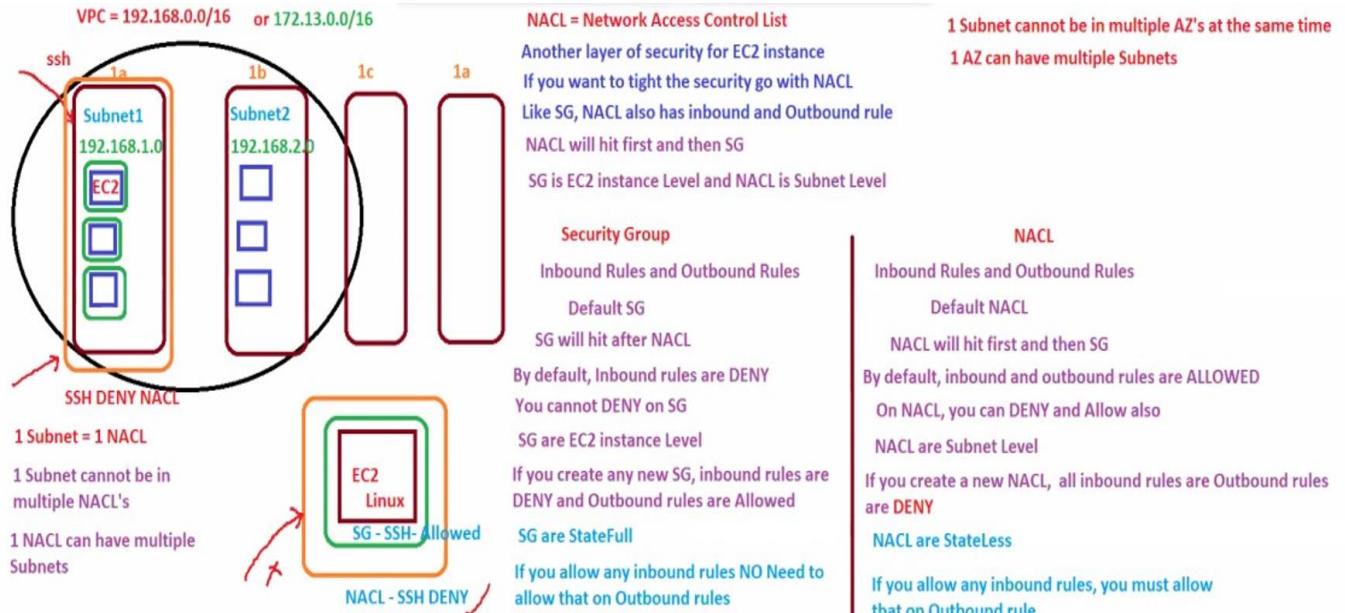




Security Groups



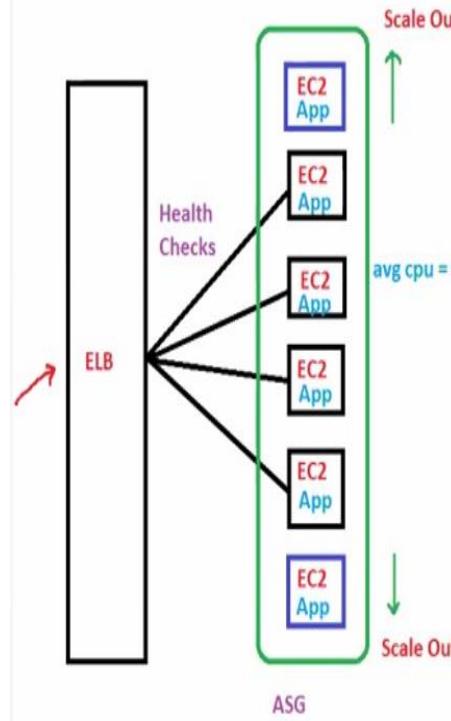
NACL



Auto-Scaling :

ELB does the Health Checks for the Application
CloudWatch monitors EC2 instance

Auto-Scaling = Scale Out and Scale In EC2 instances based on the load
adding removing



MIN = The min number of EC2 instances that the ASG should have Ex Min = 2 } Boundaries
MAX = The max number of EC2 instances that the ASG should have Ex Max = 6
Desired Capacity = The number of EC2 instances that you wish / desired to launch initially

3 Types of Scaling Options

Manual Scaling = If you are manually modifying DC, Min, Max

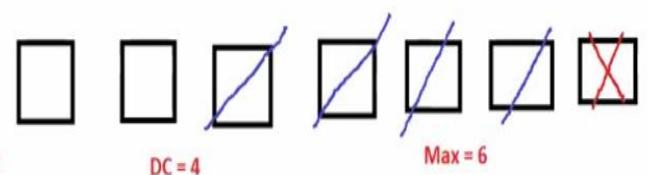
Scheduled Scaling = Based on the time period or a day

Dynamic Scaling = Based on the load (CPU, network, ELB request count) → Metrics (CloudWatch)

Launch configuration (AMI(app), Volumes, SG, Key-Pair, Tags, Instance types etc)

Launch Template

ASG = ELB + EC2 instances + Launch Configs + SNS



Elastic Load Balancer :

