

REC

2-Tier Architecture

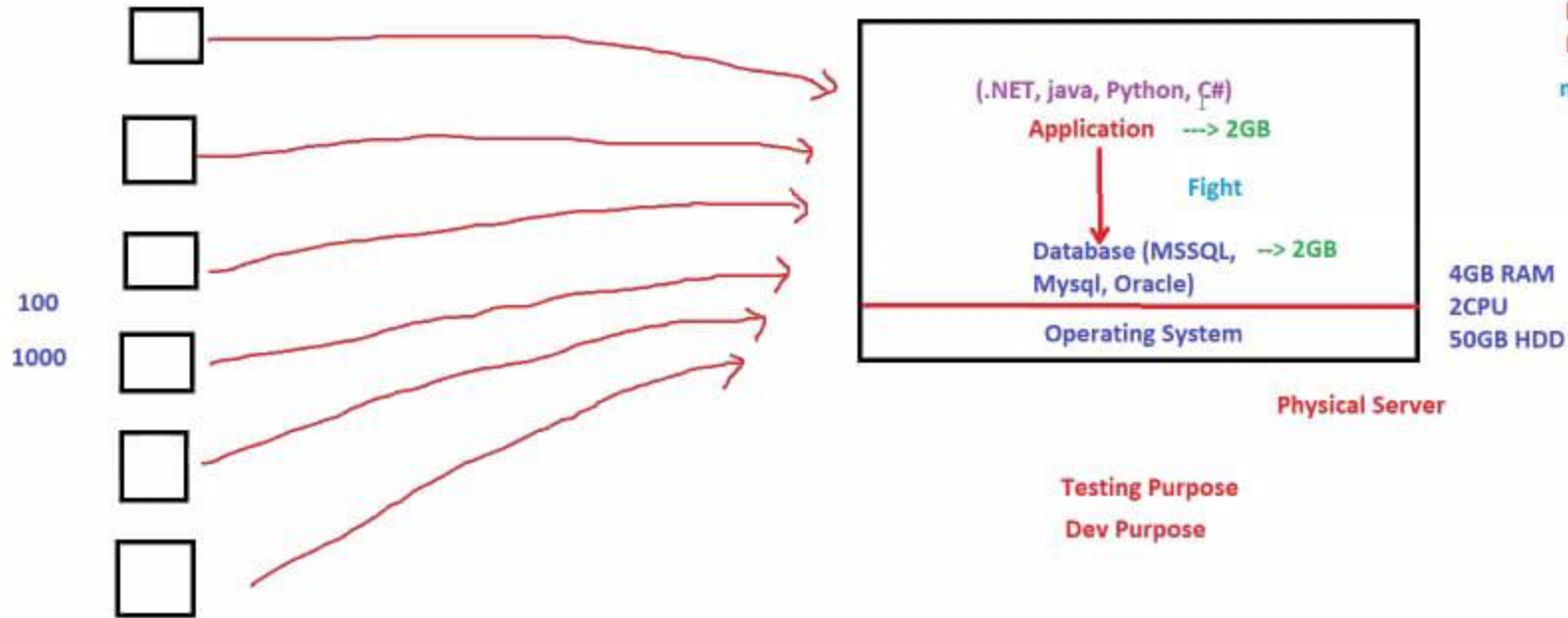
= BIG NO for Production

Client Layer

Server Layer

Slow
Die
Crash
bad performance
Hang

no security



Physical Server

Testing Purpose

Dev Purpose



Leave

REC

<http://192.168.10.20>
<http://192.168.10.21>

Jyostna



<http://jyostna.com>



Load Balancer
DNS Name



192.168.10.20



192.168.10.21



It is not recommended to use the website using IP address



Unmute



Start Video



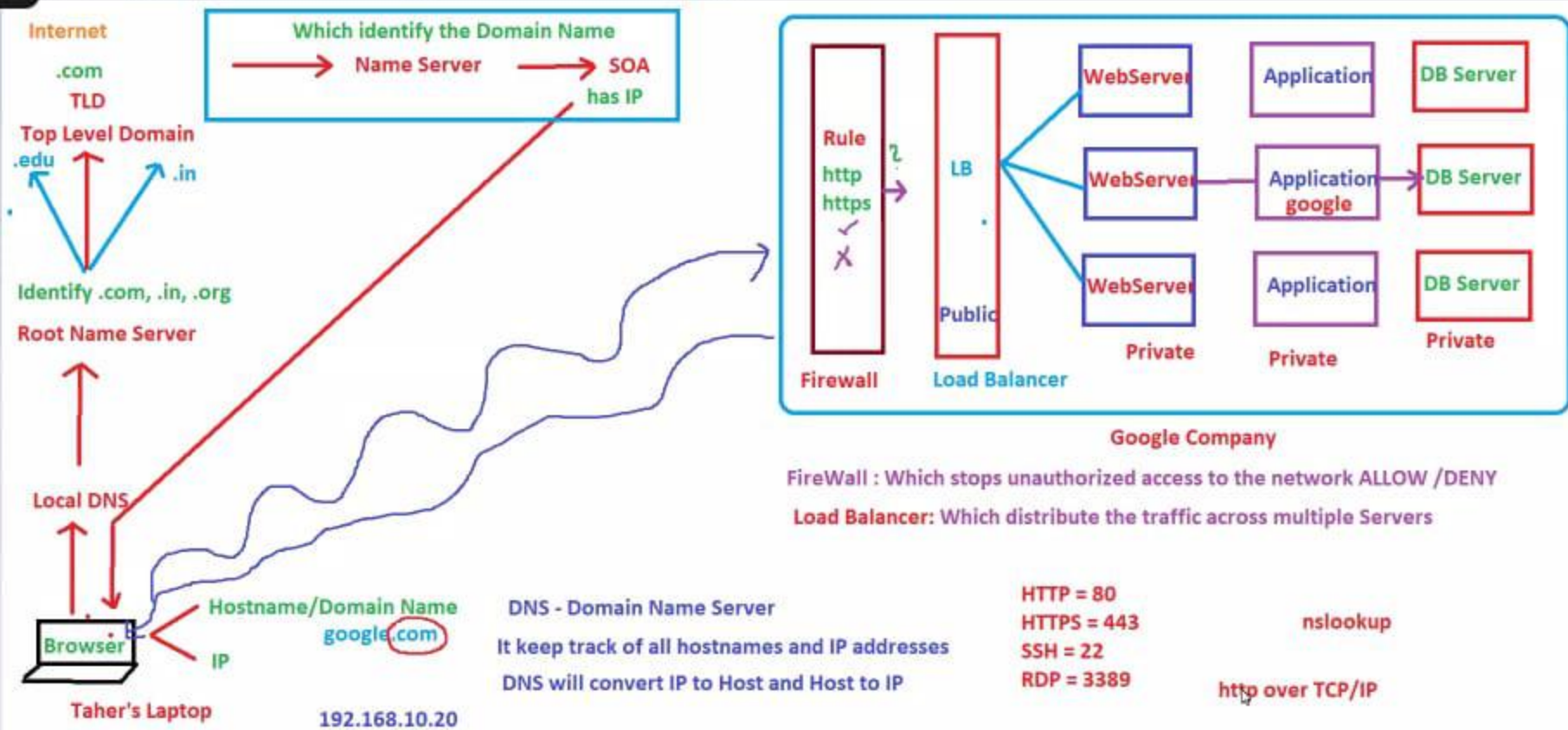
Share



Participants

More

REC



REC

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

<https://google.com>

URL

Protocol://DomainName:PortNumber

<http://google.com:80>

<http://192.168.10.20:80>

<http://192.168.10.20>

<http://192.168.10.20:8080>

Application

8080 ✓

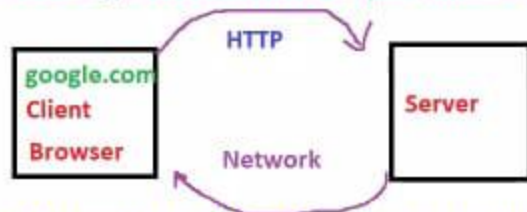
tomcat

IP

192.168.10.20

HTTP default Port Number is 80 ✓

HowEver, you can customize the port number on Application Level



HTTP transfers the data TO and FRO from the Browser to Server and Server to browser

HTTP Status Codes

404 = Page not found

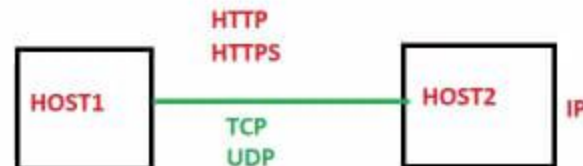
503 = Service Unavailable

401 = unauthorized

200 = Page Found

HTTP over TCP/IP

Transmission Control Protocol



IP

HTTPS : 443

encrypted

SSL

certificates

Packet

Src
Dest
protocol
Port
secure

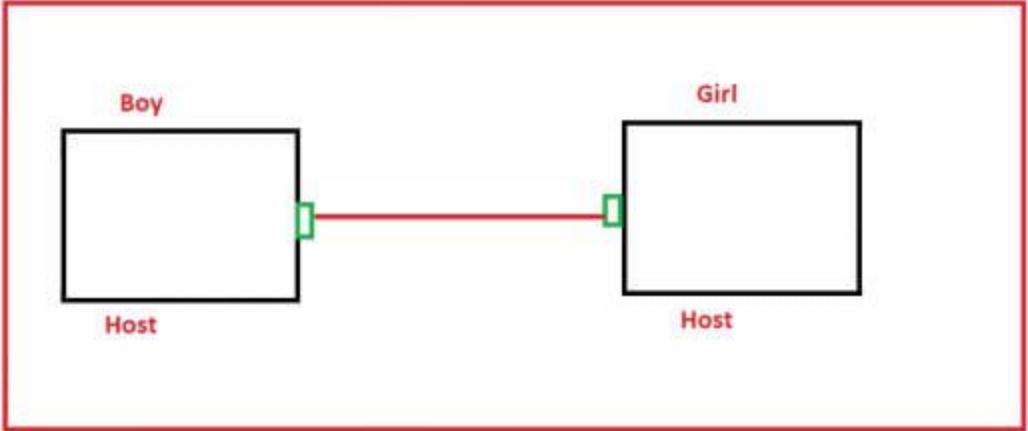
hacker

HTTP = not secure

HTTPS = secure

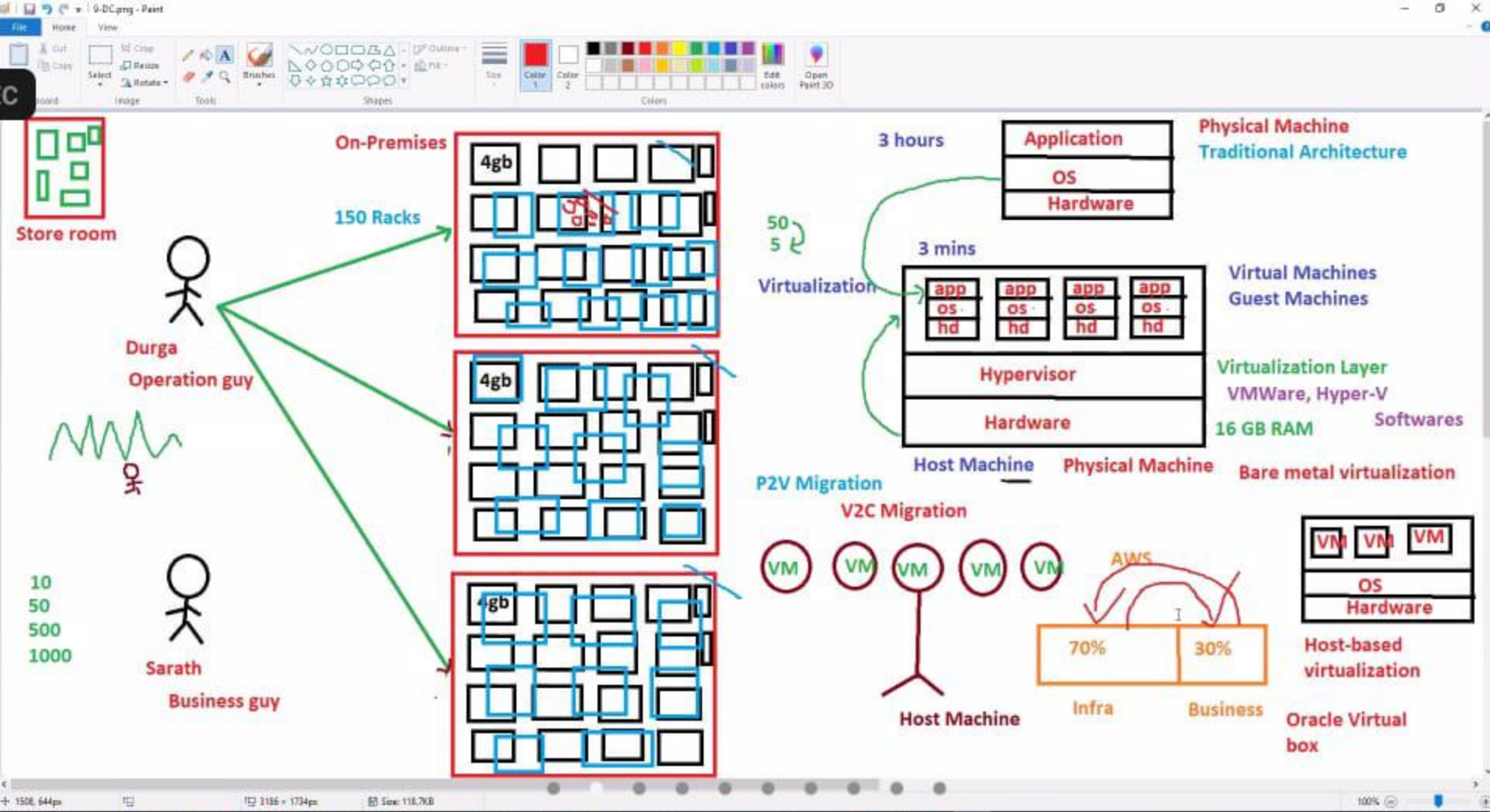
TCP establish the connection between the 2 hosts
TCP is like a messenger / bridge
TCP is reliable, UDP(user datagram protocol) is not reliable

REC



- Application Layer (HTTP)
- Presentation Layer (SSH)
- Session Layer
- Transport Layer (TCP)
- Network Layer (IP)
- Data Link Layer
- Physical Layer

REC



AWS has Global Infrastructure

AWS is Providing Infrastructure as a Service

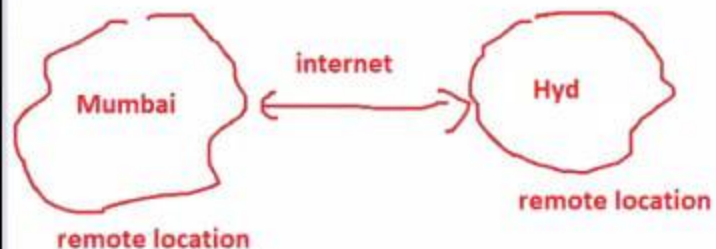
Cloud is present in the remote location

Remote location = DataCenters

We need internet to connect to the Cloud

AWS is a Cloud Provider which provides Infrastructure

AWS is a Public Cloud Provider which provides
Infrastructure as a Service



AWS = Amazon Web Services

Key-Words

Virtualization, Host machine, Virtual Machine, Infrastructure, Cloud,
Remote location, DataCenters, On-Premises, Load Balancer, Firewall, Protocols.

Physical DC --> Virtualization --> Cloud --> AWS (remote location(DataCenters))

Cloud Computing

Instead of doing computing on your local machines / local servers, you will now do the computing
on the remote location(cloud), this is called Cloud Computing

Deployment Models (Types of Cloud)

Public Cloud : The Services which are accessed by everyone like AWS

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

Hybrid Cloud : The combination of Public and Private Cloud

Community Cloud : It is same as private cloud but can be accessible from few organizations

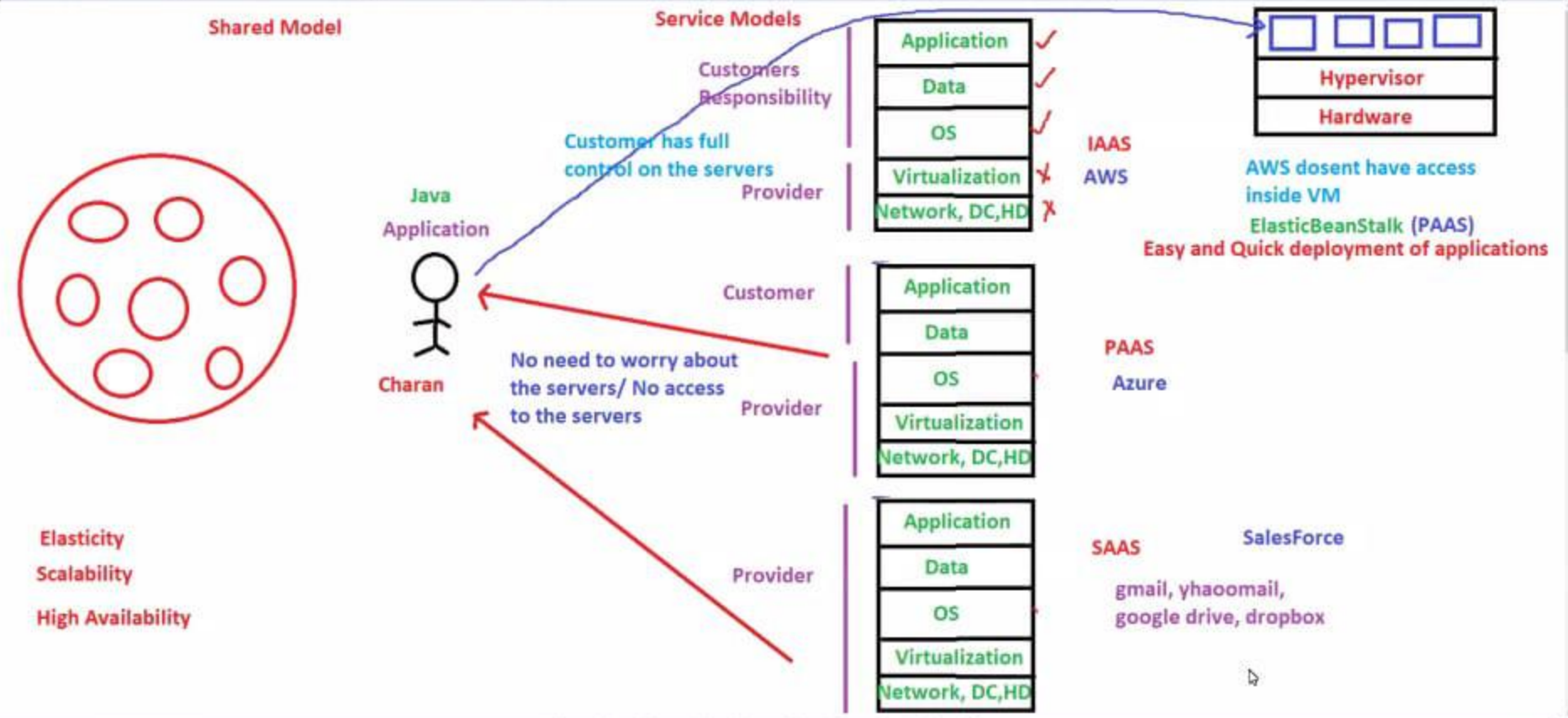
Service Models

Infrastructure As A Service (IAAS)

Platform As A Service (PAAS)

Software As A Service (SAAS)

REC



REC

Elasticity

Increasing and decreasing the capacity to meet increasing or decreasing work loads

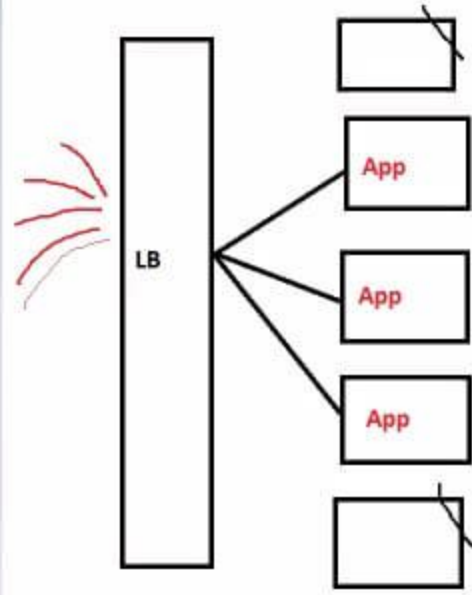
Elasticity is Short Term

Elasticity can be achieved in AWS with Auto-Scaling

Auto-Scaling = Scale Out and Scale IN

adding increasing removing decreasing

Elasticity is also called as Horizontal Scaling



Scalability



Increasing the capacity of the server is called scalability

Scalability = Scale UP and Scale DOWN

Scalability is Long term

Scalability is also called as Vertical Scaling

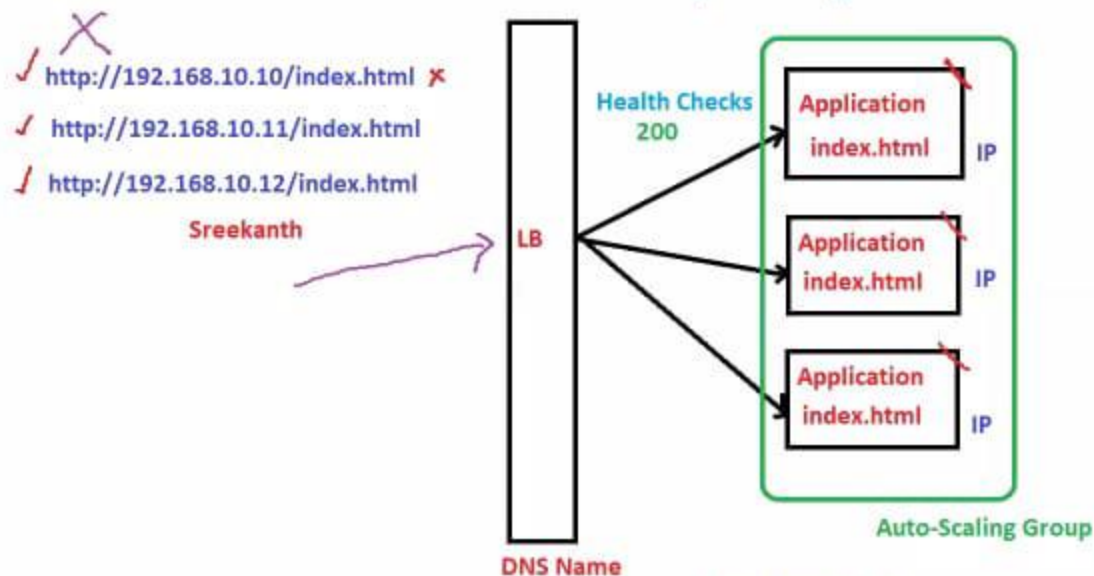
REC

Customer is not at all interested to access his application using IP, instead use LB DNS Name

High Availability

The Period of time the service is available to the customer is called HA

LB always monitor application not the server



- ✓ ~~http://192.168.10.10/index.html~~ ✗
- ✓ http://192.168.10.11/index.html
- ✓ http://192.168.10.12/index.html

Sreekanth

DNS Name

LB --> Monitoring

LB --> Failover

HA is measured in percentage

High Availability

Redundancy
Monitoring
Failover

REC

AWS has Global Infrastructure

A region has multiple DataCenters

A region has multiple AZ's

AZ's networks are interconnected

AZ's are in sync with each other(network) but not DATA.

Best Practise is to distribute the servers across multiple AZ's.

Very less chance that 1AZ goes down

1 AZ = 1 DC ✗

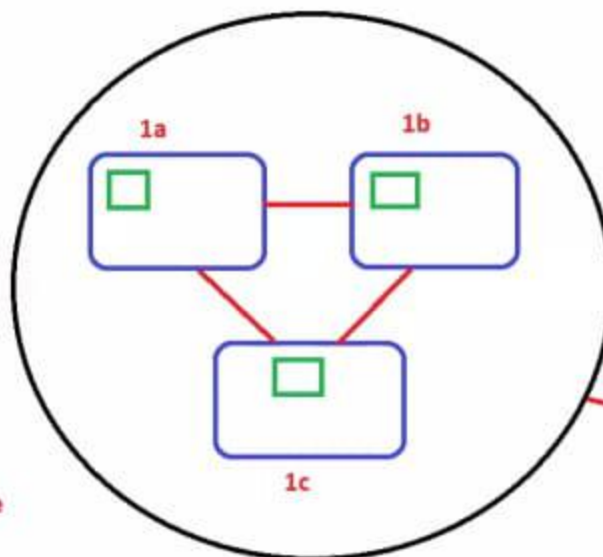
AZ is a group of DataCenters

Server = Instance

Regions and Availability Zones

Region: is a geo-location, Ex: AWS-region = Mumbai

Availability Zones: Simply a DataCenter(AZ)



AWS Region = Mumbai

1a

1b

1c

dadar, andheri, aroli | chane, bandra, ghatkopar | juhu, khar, dharavi

Every AWS region has a Code

AWS Mumbai Region = ap-south-1

AZ = ap-south-1a

ap-south-1b

ap-south-1c

Regions and AZ's are managed by AWS

AZ's can communicate with each other

Low Latency = Good

High Latency = Bad

google.com --> 1 sec = Low

google.com --> 5 secs = High

Another AWS region

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

LB can distribute the traffic to multiple instances across multiple AZ's

REC

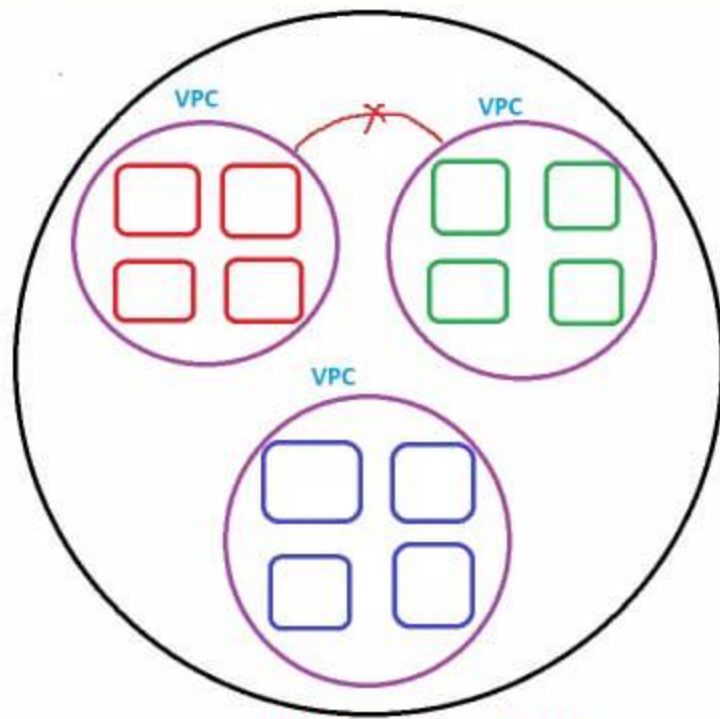
Venkatesh

Sreekanth

Kali

Naveen

Harsha



Virtual Private Cloud

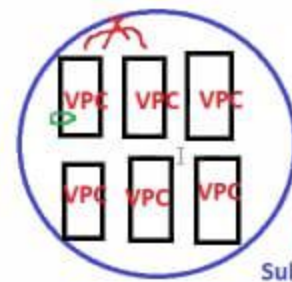
VPC

Virtual DataCenter on AWS

AWS Public Cloud Mumbai

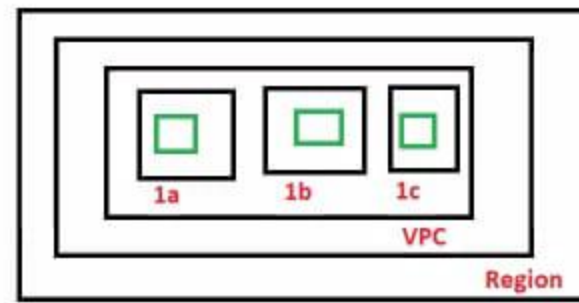
One region can have multiple VPC's

VPC is regional



AWS

Sulabh Complex Public



Region

AWS

AWS service can be Regional or Global

EC2 is regional

EC2 - Elastic Compute Cloud
Server = Instance = EC2 Instance (VM)

In EC2 Service, We can launch EC2 Instances

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, ASG, Scalability, Performance etc)

ELB is not a Server for us, It is a Service for us

You cannot Login to the ELB, but you can access ELB with DNS Name

ELB dosent have AZ's. it is created at regional level

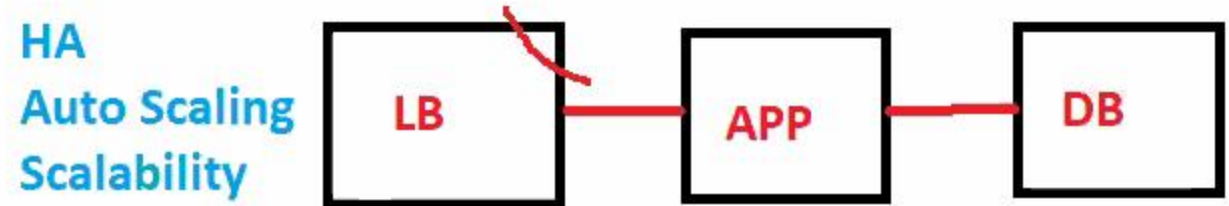
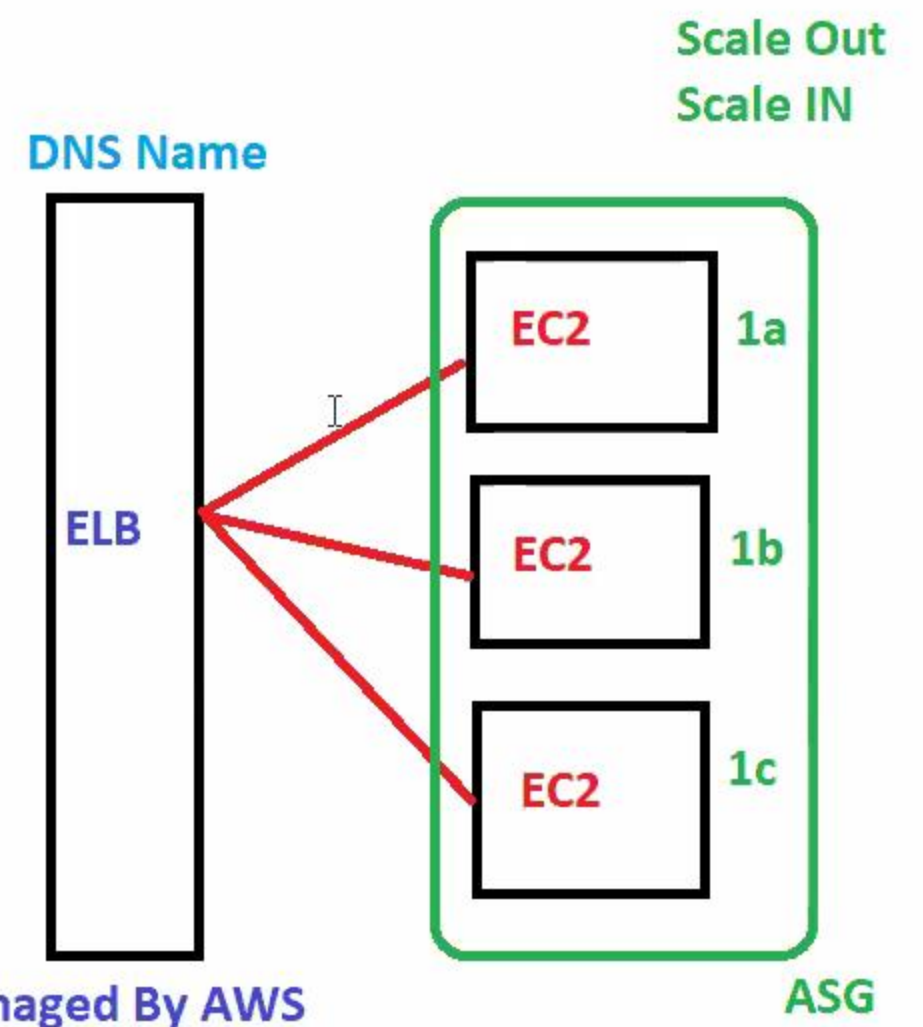
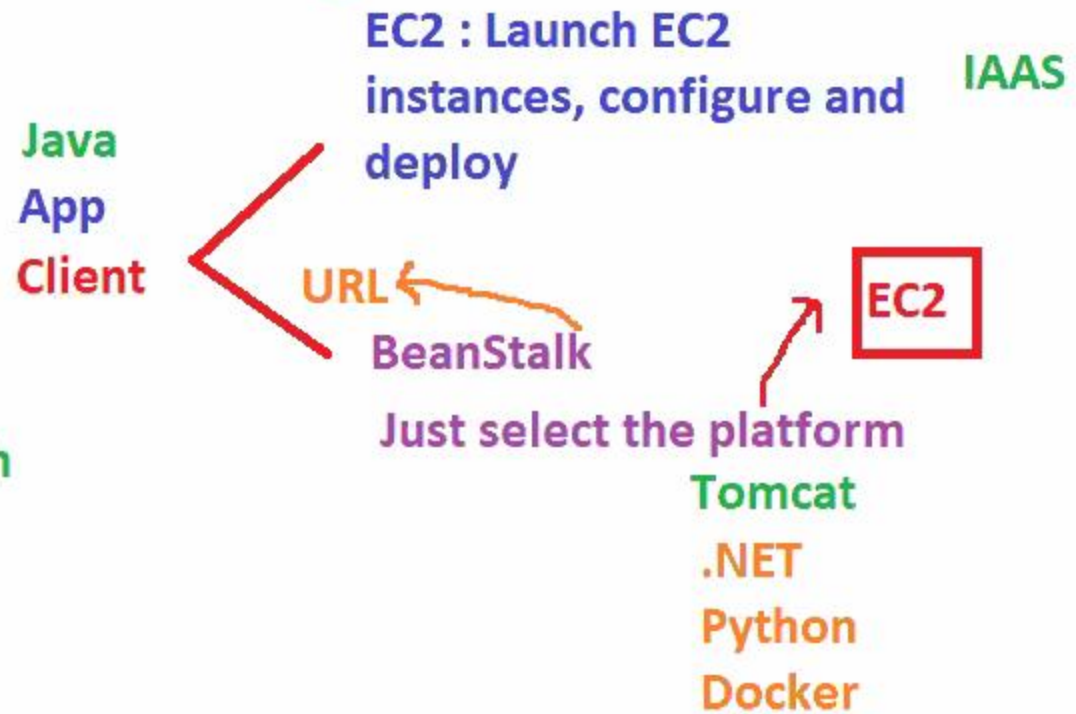
ElasticBeanStalk = Easy and quick deployment of applications in AWS

In General, in PAAS --> You dont have 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

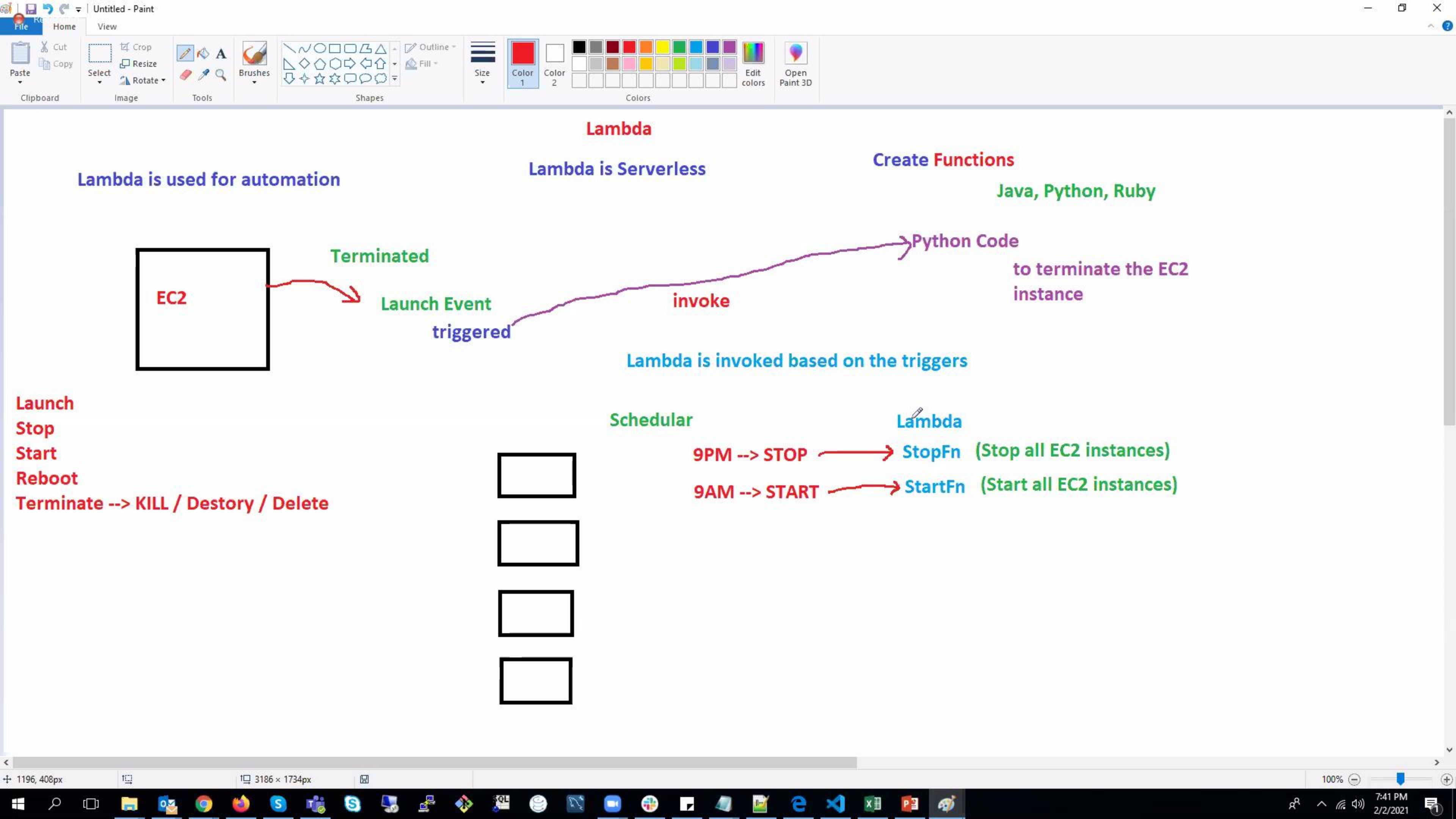
LightSail: If you want to setup and create a virtual private server which already has everything installed (WordPress, Gitlab, Joomla, Drupal, Nodejs, Django) --> No AutoScaling



On-Premises

CP From Chaitanya prasad to Everyone

yes sir



Lambda

Lambda is Serverless

Create Functions

Java, Python, Ruby

Lambda is used for automation



Terminated

Launch Event

triggered

invoke

Python Code

to terminate the EC2
instance

Lambda is invoked based on the triggers

Launch

Stop

Start

Reboot

Terminate --> KILL / Destory / Delete



Scheduler

Lambda

9PM --> STOP



StopFn

(Stop all EC2 instances)

9AM --> START



StartFn

(Start all EC2 instances)

100%

7:41 PM
2/2/2021



In AWS, all services will start with Simple and end with Service

SNS --> Simple Notification Service

S3 - Simple Storage Service

S3 is unlimited Storage by AWS

S3 is used to just store your files

S3 is used to store all FLAT files

With S3, you can upload, download, access your file

You cannot execute any files in S3

- Is it possible to install OS on S3? NO
- Is it possible to install DB on S3? NO
- It is possible to run .net, .exe, .py etc in S3 ? NO

S3 is Serverless

AWS handles HA, Performance, scalability etc for S3

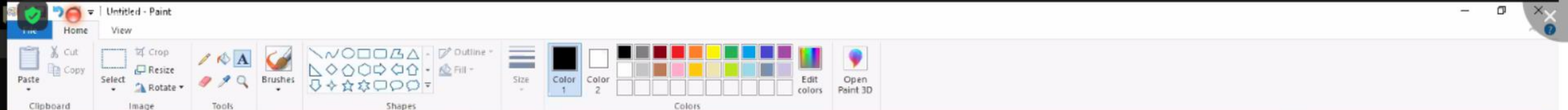
- Bucket is a Container for Objects
- Object is a File
- Key is the File Name / Name of the Object

- Floppy --> 2MB
- CD's --> 700MB
- DVD's --> 4.7 GB
- Pen Drives --> 128GB
- Hard Disk --> 2TB

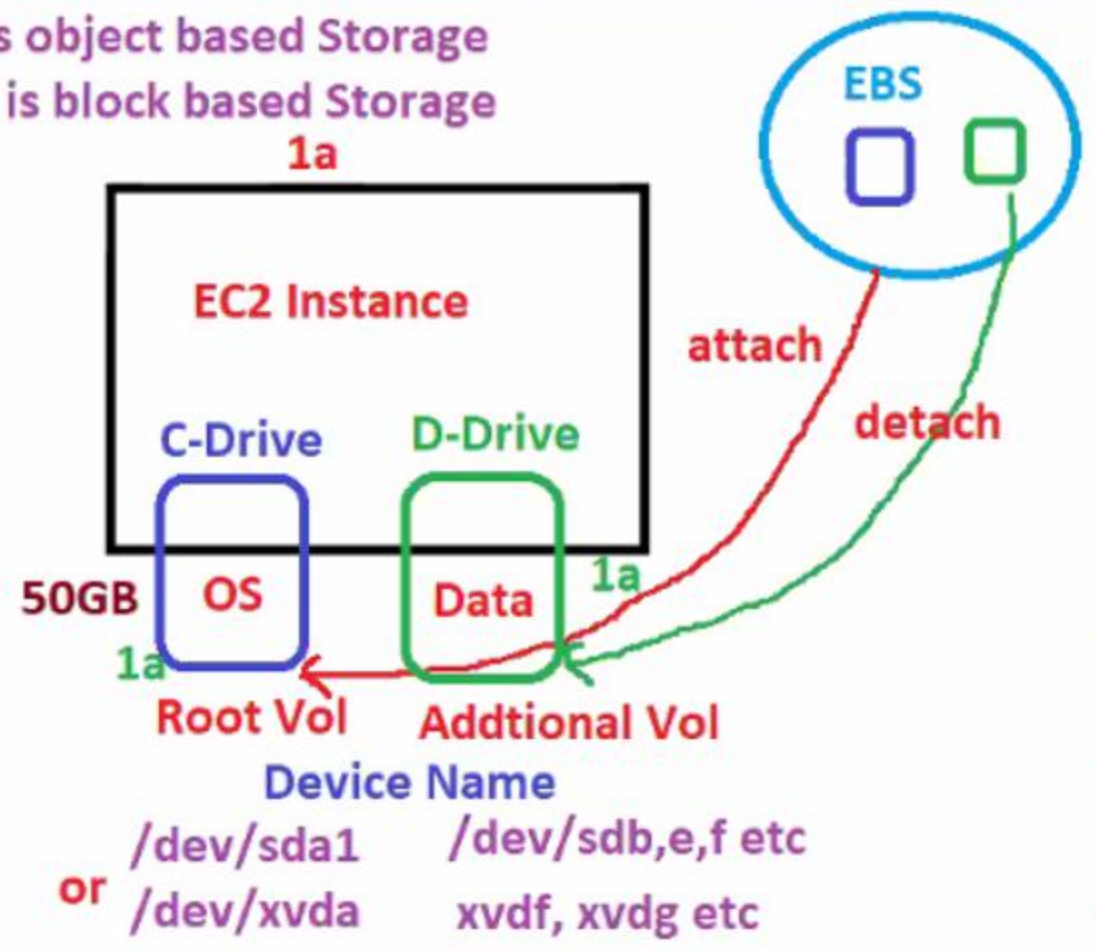
** S3 is Object Based Storage

Windows	S3
Folder	Bucket
Files	Objects
boom.mp3	Key

- ** S3 supports STATIC Website Hosting (HTML files)
- Create a bucket, upload all files/objects and enable static website hosting feature.
- HA, Performance, etc etc handled by S3
- YOu cannot attach S3 to EC2 instance,
- You can access S3 from Ec2 instance



****S3 is object based Storage**
****EBS is block based Storage**



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

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 the Additional volume while EC2 is running? **YES**, it is not recommended to detach while running, stop first

EC2 instance has AZ, Volumes also has AZ's

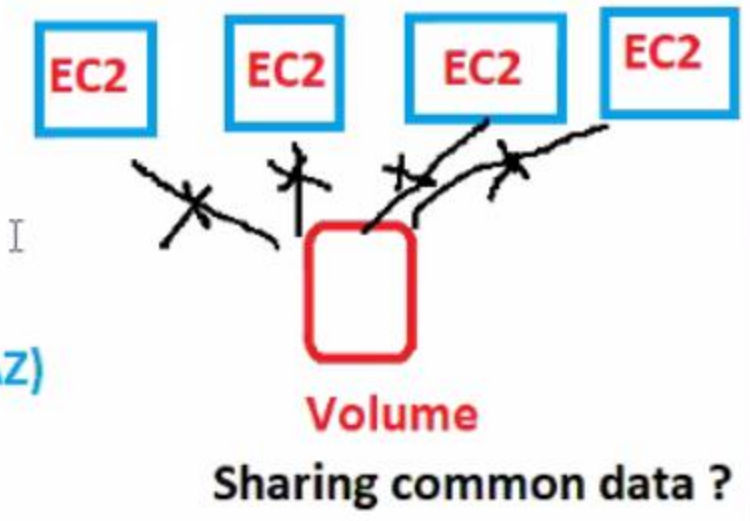
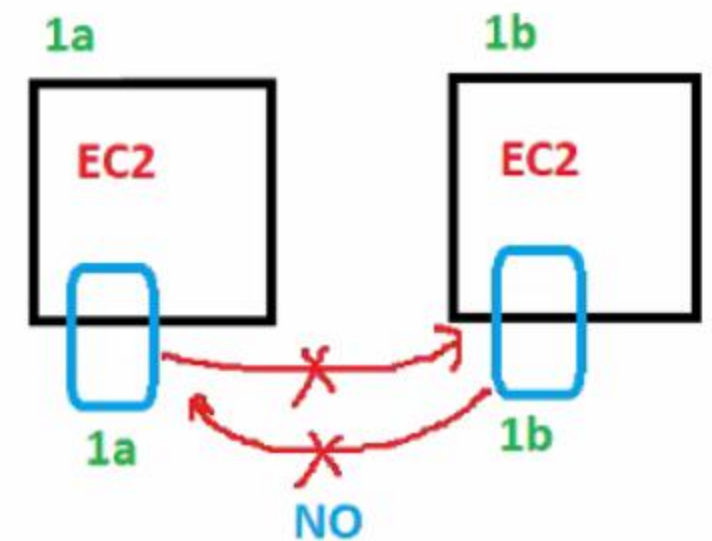
**** EC2 instance and Volume should be in the same AZ**
1a 1a

EBS - Elastic Block Storage

HardDisk = 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 has 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 can have only 1 ROOT Volume
EC2 can have multiple additional volumes
Max Size of the Volume is 16TB
Volumes should be pre-provisioned like 50GB, 100GB .. Max 16TB
You cannot attach a volume to multiple EC2 instances at the same time
Volumes size can be increased on FLY (no need to stop the EC2 instance)
Volume size cannot be decreased

Client side OS
Windows 10

Server Side OS
Windows 2016, 2019, Linux, ubuntu



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

Is it possible to attach single volume to multiple EC2 instances at the same time ? NO
Volumes cannot be shared across EC2 instances

Storage Gateway

On-Prem
 Synchronizing the data from On-prem to AWS

AWS
 S3
 EBS
 Glacier

EC2 Linux
 Root
 EFS

1a
1b
1c

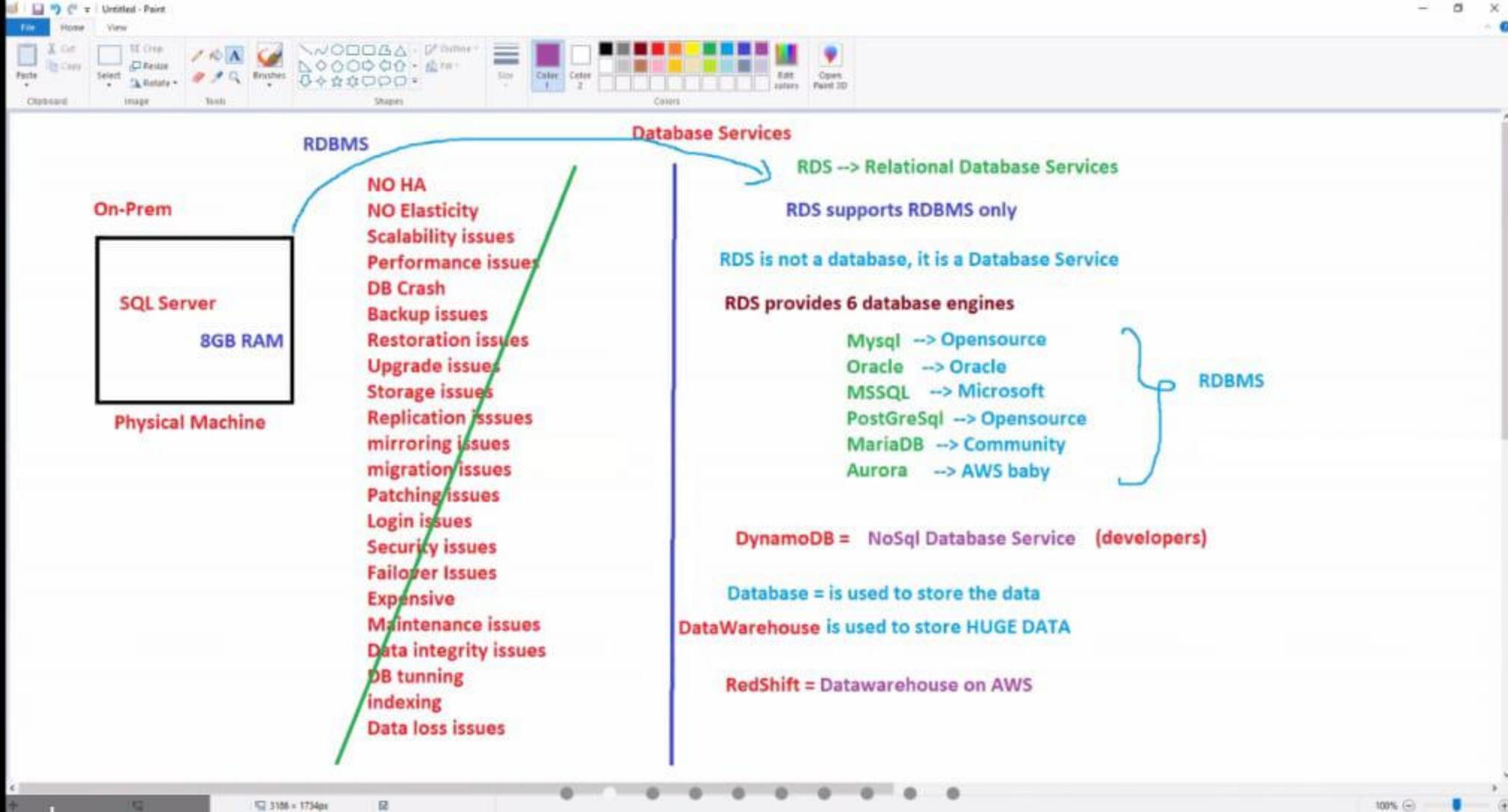
ABC
EFS

EFS - Elastic File System

EFS is completely managed by AWS
 EFS is only for Linux EC2 Instances
 Fsx is for Windows EC2 Instances
 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 it on EFS).
 EFS can be mounted to multiple EC2 instances at the same time across multiple AZ's.
 Glacier = Archiving Purpose
 Cheaper than S3

Snow Family
 SnowCone --> 8 TB
 SnowEdge --> 100 TB
 SnowMobile --> PB (Truck)
 SnowFamily is used to Physical transfer huge data from on-Prem to AWS and AWS to on-prem
 SnowFamily is a Physical Transfer



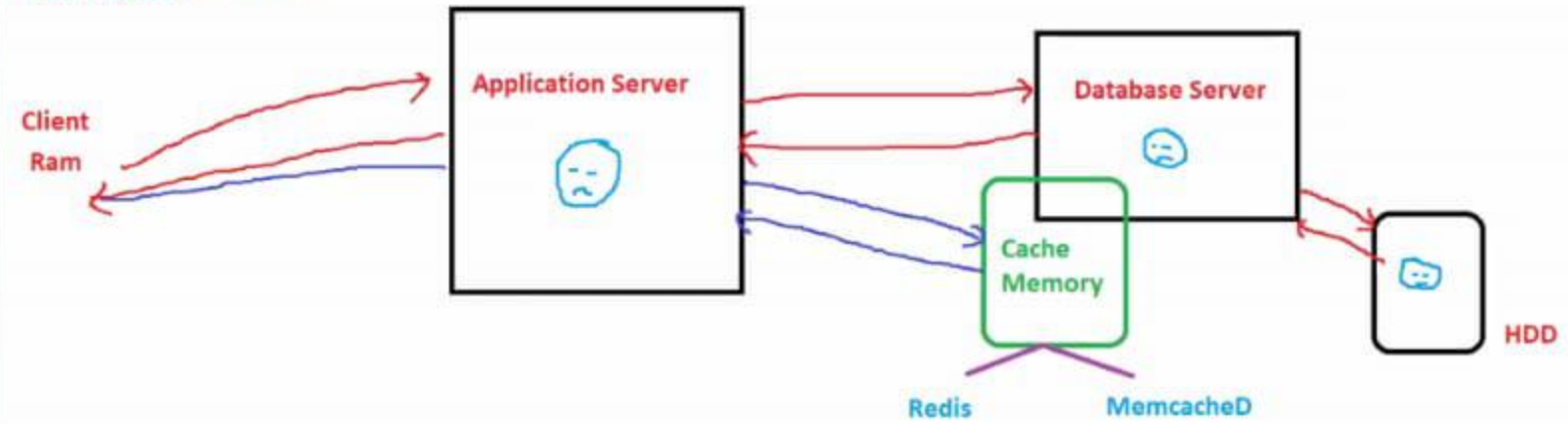


REC

ElastiCache = In memory database Caching Service
= Low Latency
= High Performance

Cache: Frequently access data is stored at this place

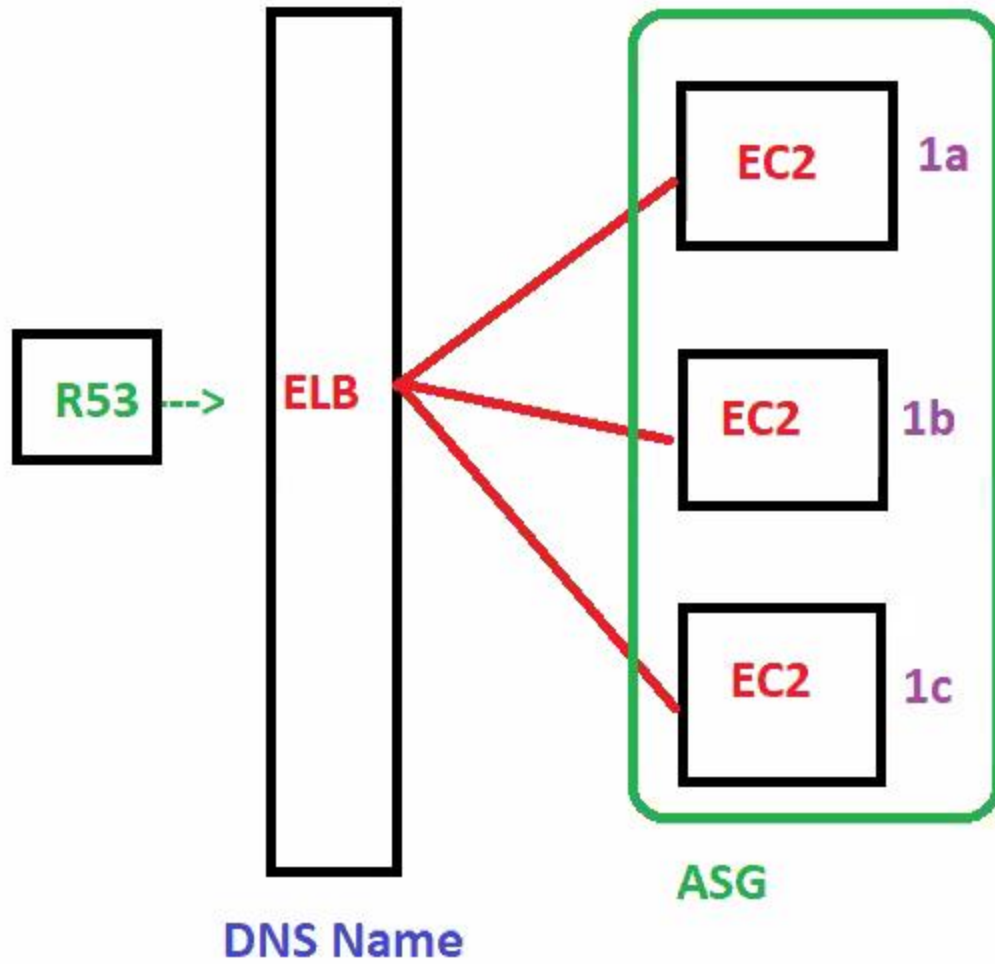
amazon.com mobiles



Redis
Memcached
~~Varnish~~



Route53
R53 is DNS Service in AWS
DNS Port Number is 53

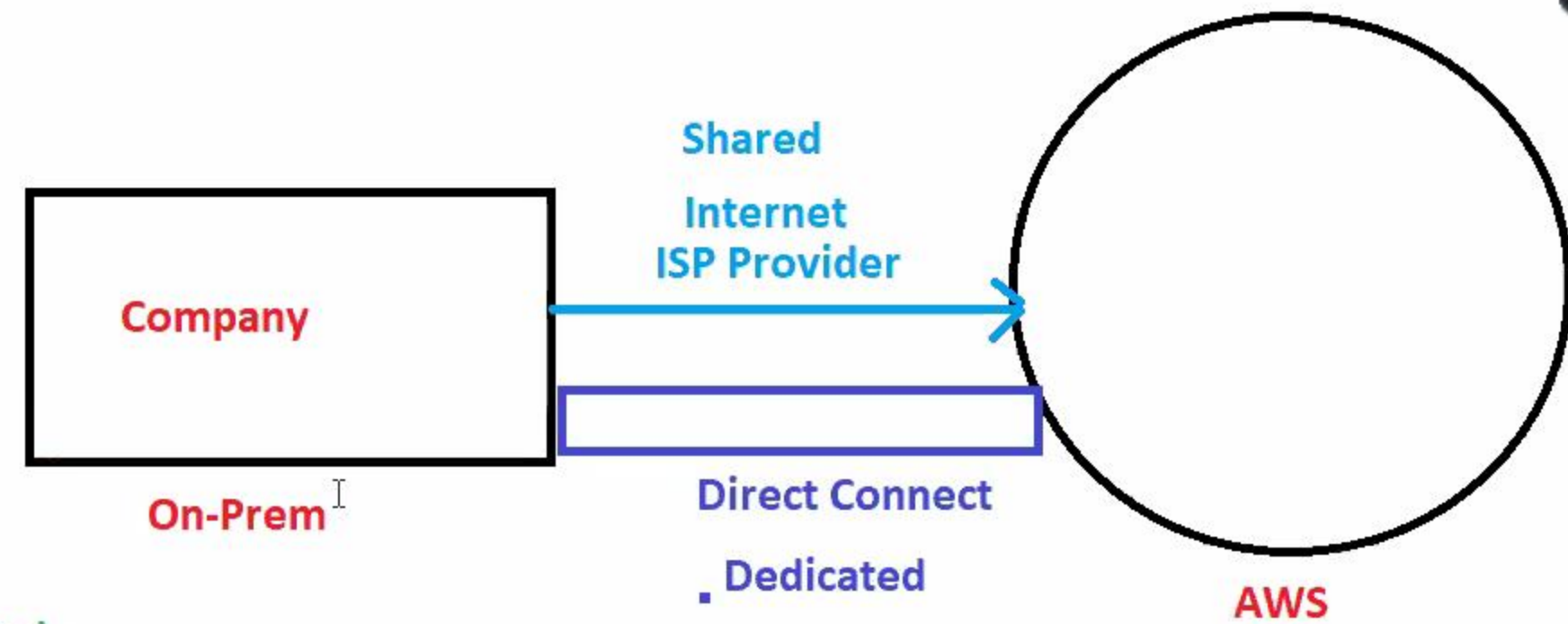
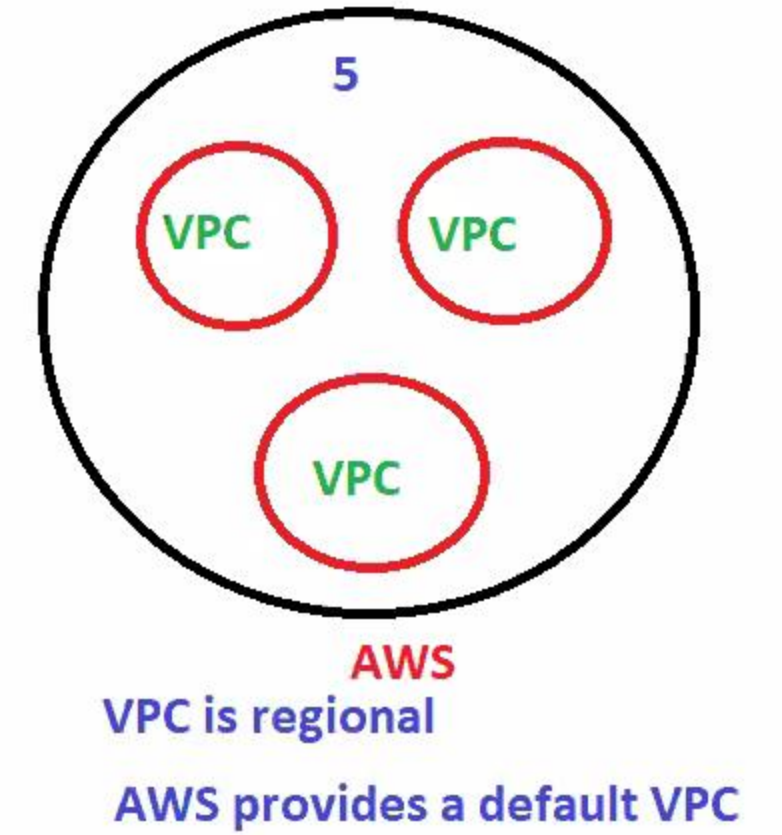
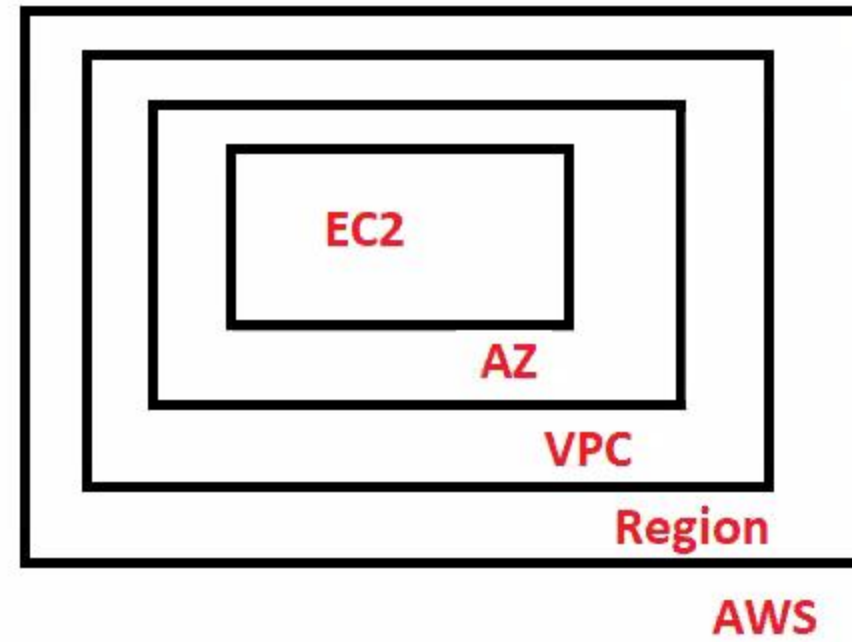


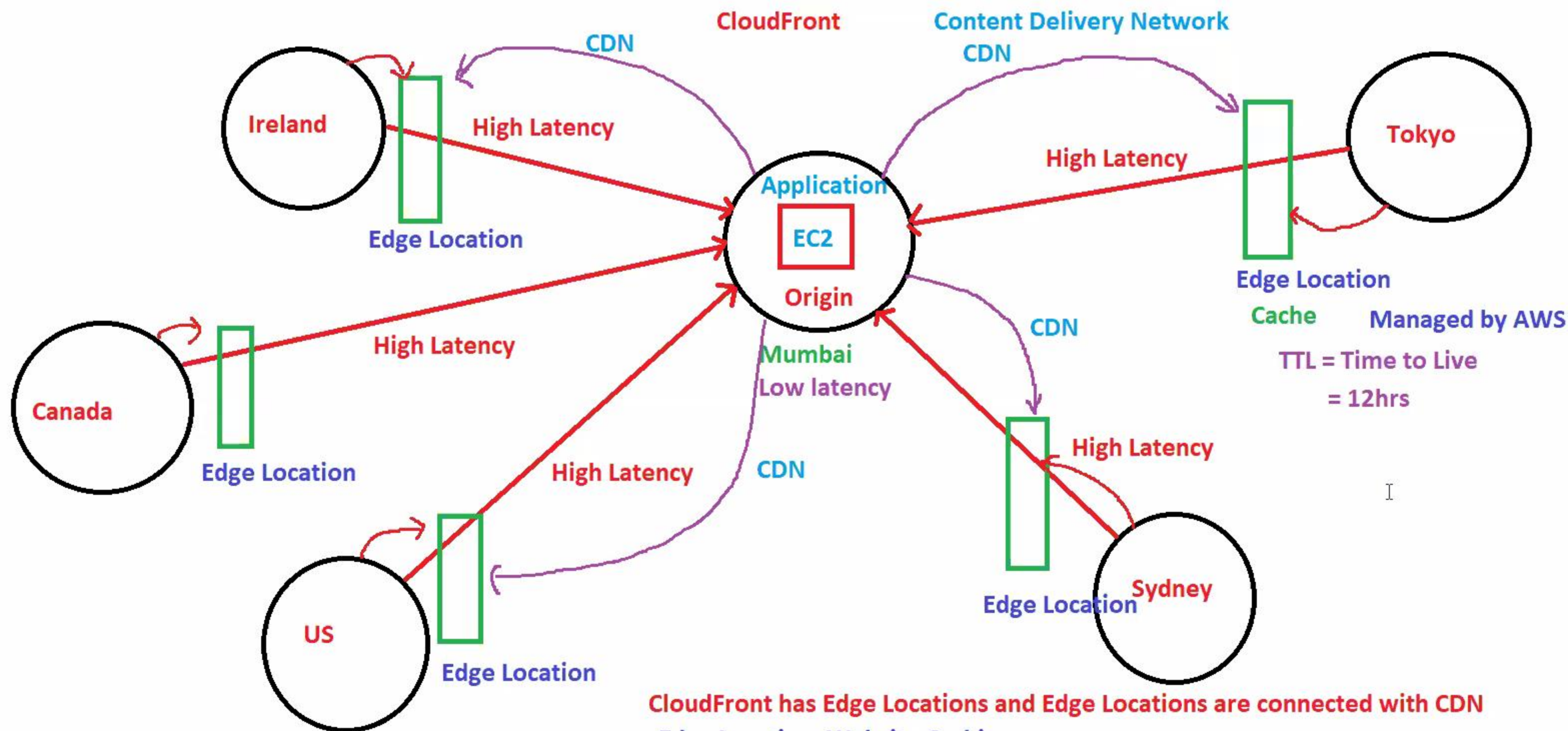
<http://myelb.39475934857.elb.amazonaws.com>

Mapping/routing records
Sreekanth.com

sreekanth.com --> Nasty URL (ELB DNS Name)

VPC : Virtual Private Cloud
like a virtual Datacenter on AWS Cloud





Cache Managed by AWS
TTL = Time to Live
= 12hrs

CloudFront has Edge Locations and Edge Locations are connected with CDN
Edge Location: Website Caching
It will be cahed based on TTL

Edge Locations are completely managed by AWS
We need to just setup/configure to use

IAM - Identity and Access Management

We can Control the entire AWS account using IAM by giving proper permissions to the IAM Users

Parent account (root account)

IAM Users
IAM Users
IAM Users
IAM Users

Users / Accounts

Root User

Root User has full permissions

Login with Email / PWD

IAM user

IAM user has limited Permissions

Login with username / pwd

Root Account

1 company

IAM Users

permissions/policies

Thirumal : EC2

Sreekanth : S3

Pavan : EFS

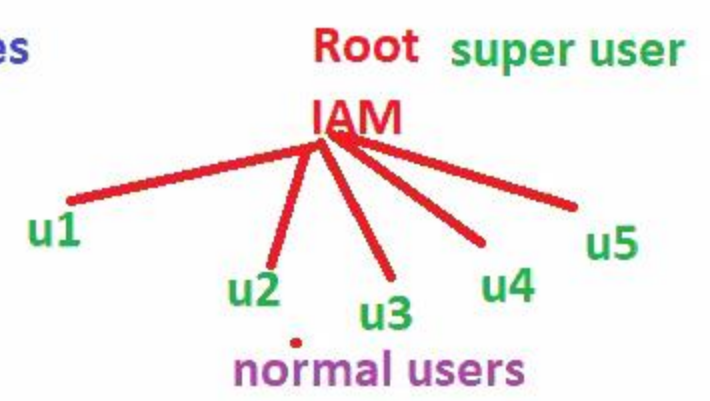
Rama : RDS

Shiva : CF

Pranav : Admin (Billing section)

The person who provide the card details he/she is the root user, the users who are using from that account they are called IAM users

IAM = Access control to AWS resources



Organizations is a Feature in IAM

Consolidated bill

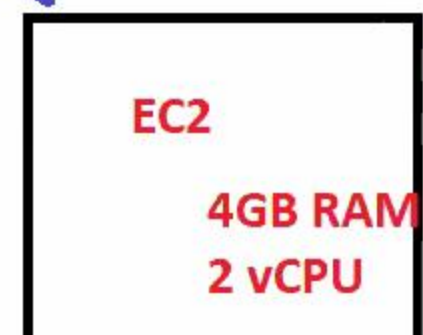
Organizations

(Owner)

Company (Root Account)



SES -> Simple Email Service
SNS --> Simple Notification Service
SQS -> Simple Queue Service



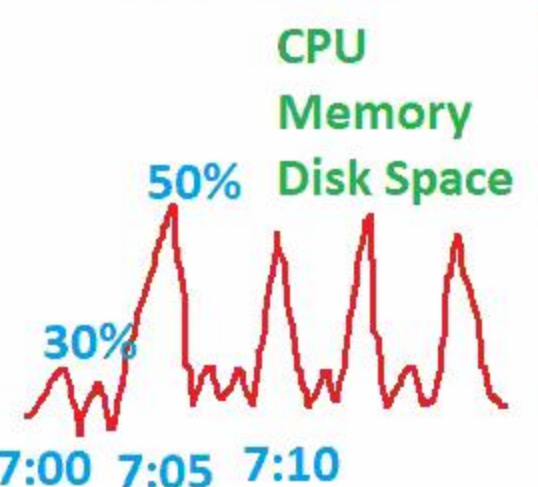
Cloudwatch

CPU > 90 %

SNS

email

SMS



Metrics

CloudWatch

CloudWatch is used to monitor AWS resources(EC2, ELB, S3, RDS, EFS, ASG etc)

CloudWatch Monitor Performance

Basic Monitoring : You will get the data points every 5 mins, FREE

Detailed Monitoring : You will get the data points every 1 min, Charged

Alarms

CPU > 90%

SNS

5AM

beep

CloudTrail : MOnitor entire AWS account

record, monitors, track, audit, logs

Config: Monitors the changed in AWS resources



GYM / AWS

AWS Support

Basic Support FREE

Developer Support --> 100 \$

Business Support

Enterprise Support --> \$ 15,000 / month

2

5

10

15

10

Leave Meeting