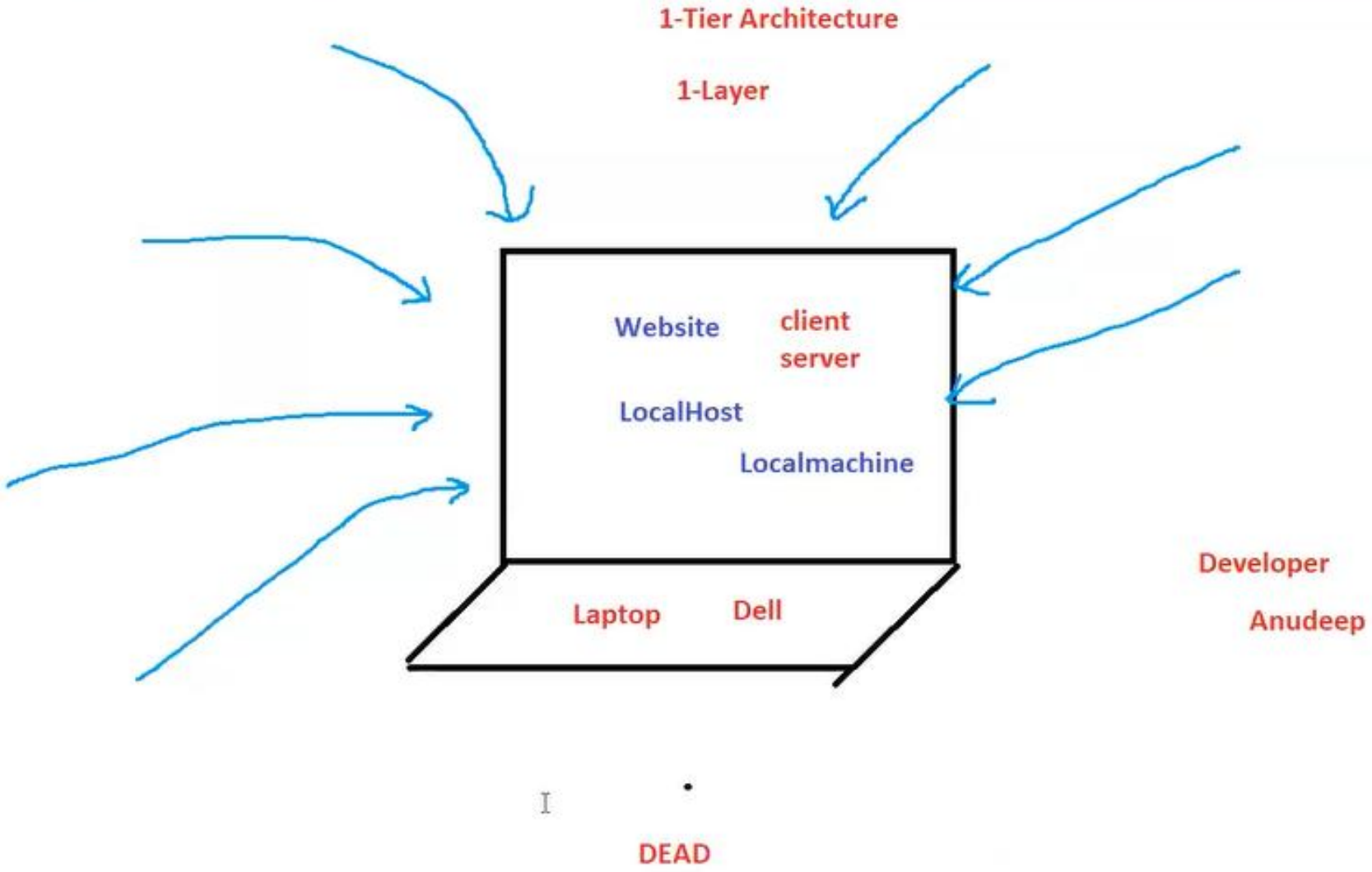
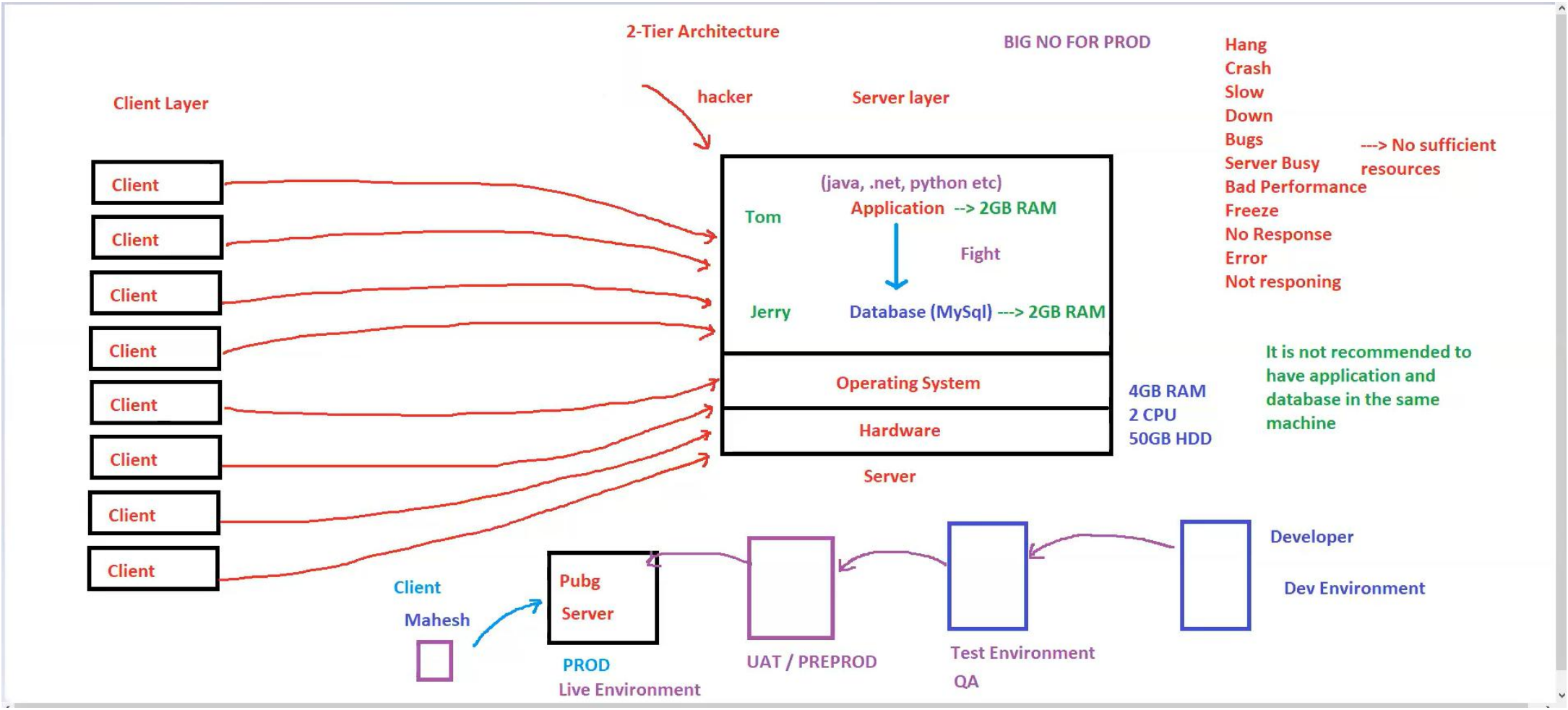
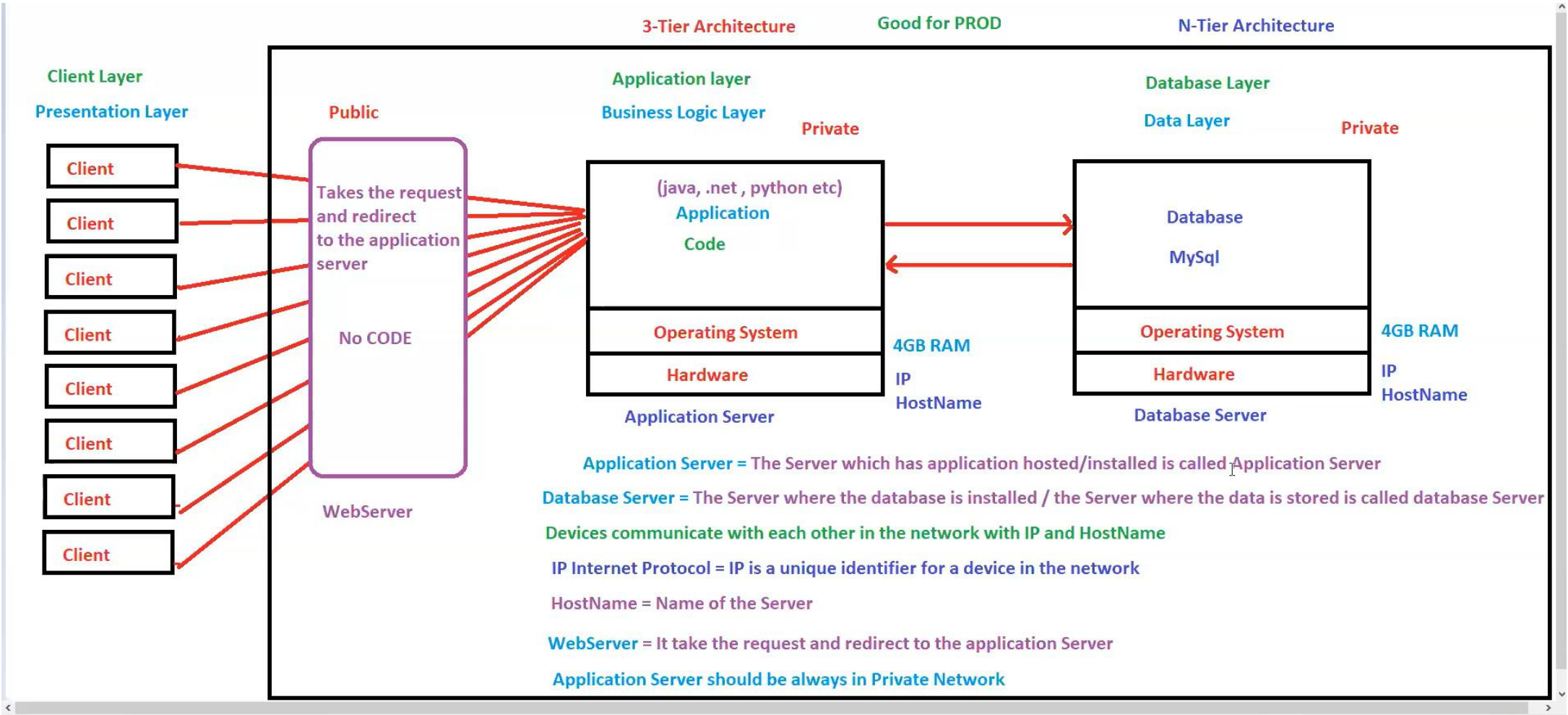
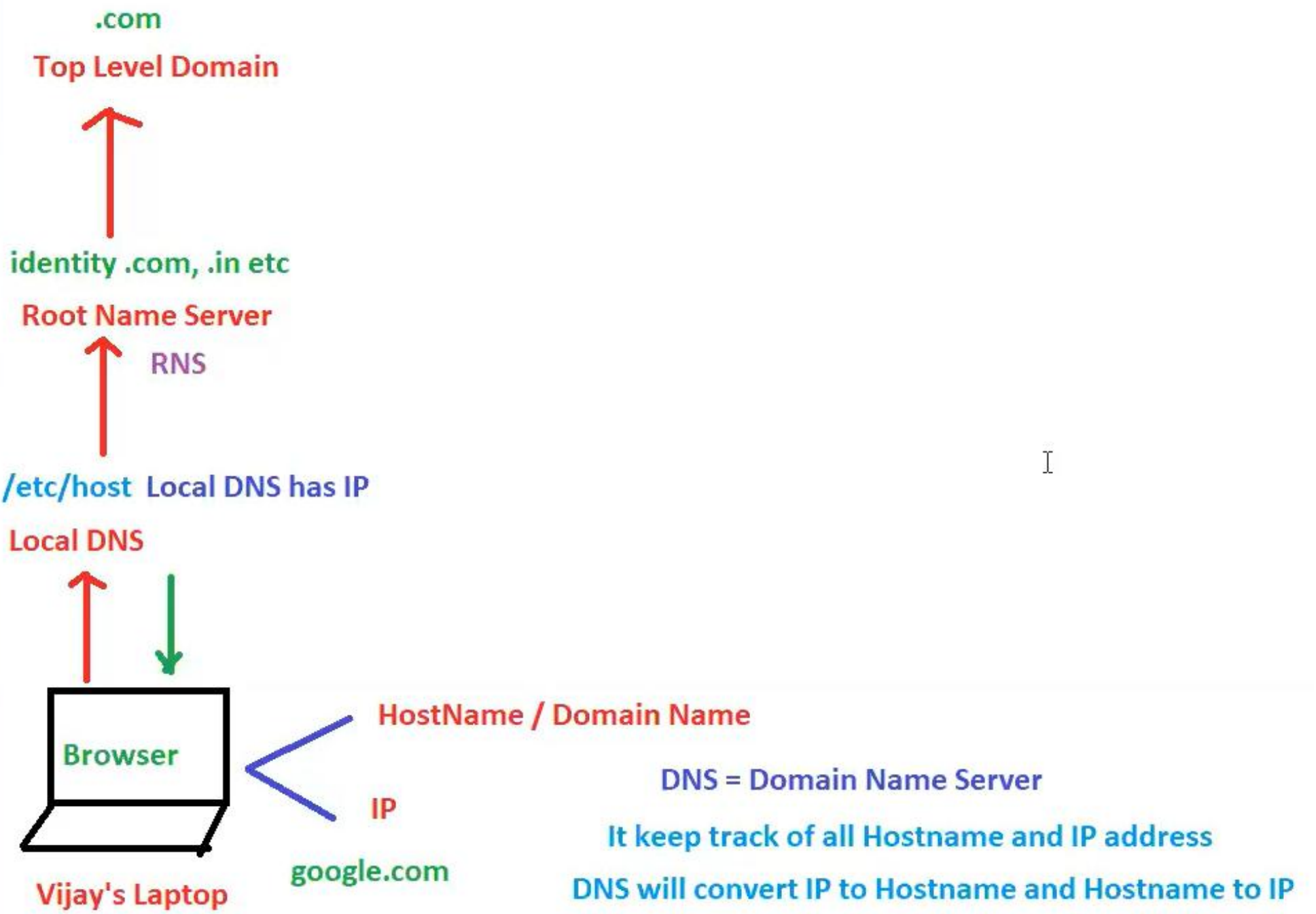
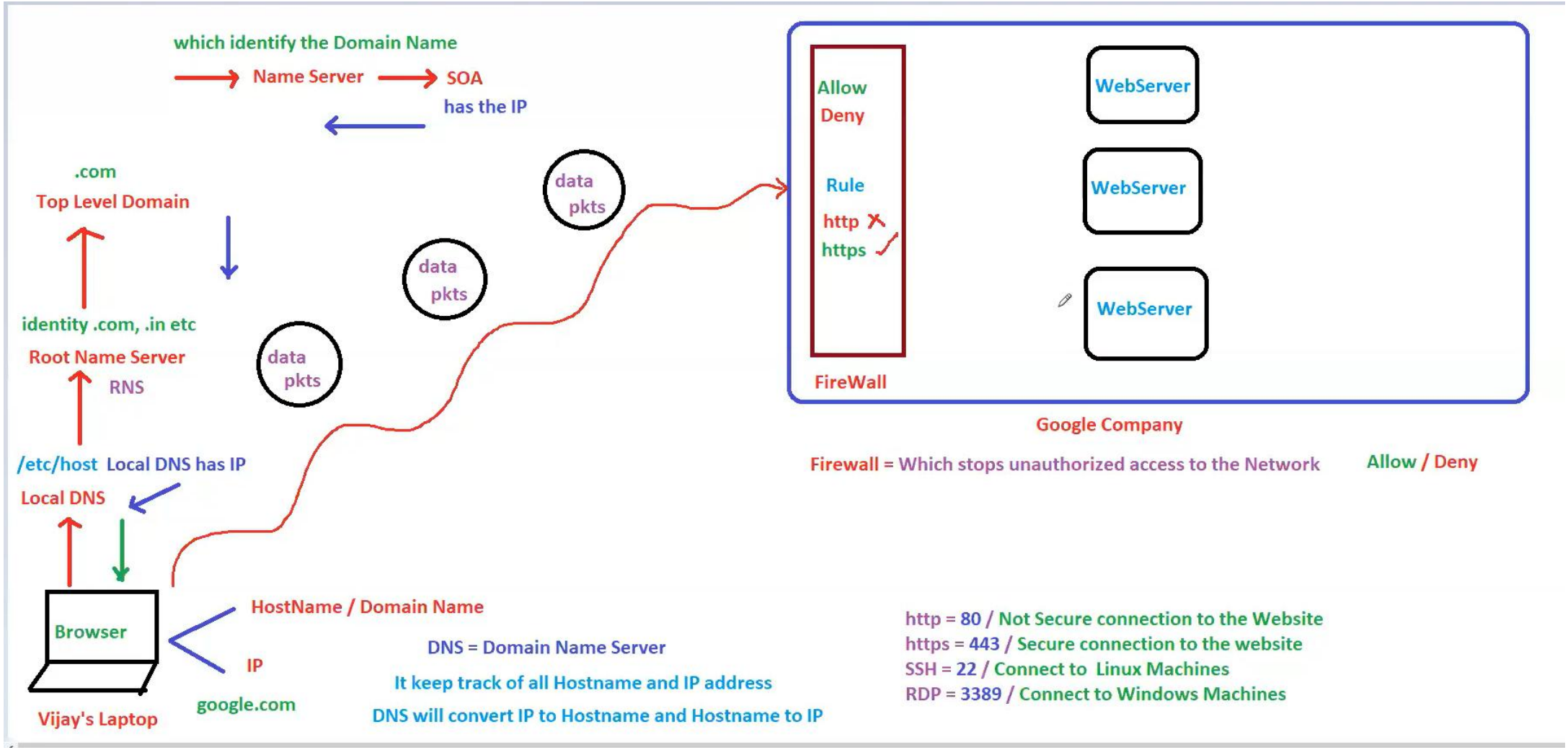
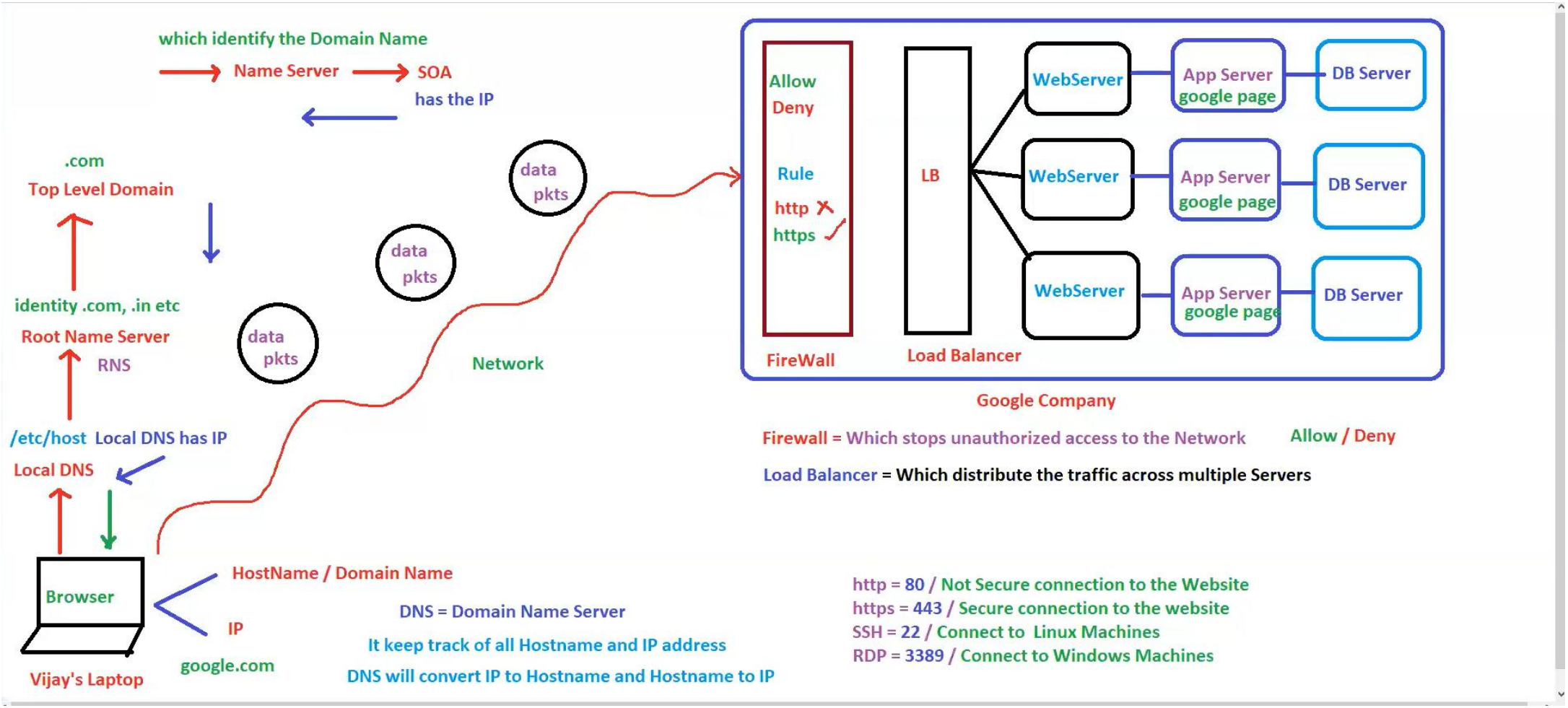

1-Tier Architecture
1-Layer
Website
client
server
LocalHost
Localmachine
Laptop
Dell
Developer
Anudeep
DEAD


2-Tier Architecture
BIG NO FOR PROD
Hang
Crash
Slow
Down
Bugs
Server Busy
Bad Performance
Freeze
No Response
Error
Not responing
---> No sufficient resources
Client Layer
hacker
Server layer
Client
Client
Client
Client
Client
Client
Client
Client
(java, .net, python etc)
Tom
Application --> 2GB RAM
Fight
Jerry
Database (MySql) ---> 2GB RAM
Operating System
Hardware
Server
4GB RAM
2 CPU
50GB HDD
It is not recommended to have application and database in the same machine
Developer
Dev Environment
Test Environment
QA
UAT / PREPROD
Client
Mahesh
Pubg
Server
PROD
Live Environment


3-Tier Architecture
Good for PROD
N-Tier Architecture
Client Layer
Presentation Layer
Public
Application layer
Business Logic Layer
Private
Database Layer
Data Layer
Private
Client
Client
Client
Client
Client
Client
Client
Client
Takes the request and redirect to the application server
No CODE
WebServer
(java, .net , python etc) Application Code
Operating System
Hardware
4GB RAM
IP
HostName
Application Server
Database
MySql
Operating System
Hardware
4GB RAM
IP
HostName
Database Server
Application Server = The Server which has application hosted/installed is called Application Server
Database Server = The Server where the database is installed / the Server where the data is stored is called database Server
Devices communicate with each other in the network with IP and HostName
IP Internet Protocol = IP is a unique identifier for a device in the network
HostName = Name of the Server
WebServer = It take the request and redirect to the application Server
Application Server should be always in Private Network


.com
Top Level Domain
identity .com, .in etc
Root Name Server
RNS
/etc/host Local DNS has IP
Local DNS
Browser
HostName / Domain Name
IP
google.com
Vijay's Laptop
DNS = Domain Name Server
It keep track of all Hostname and IP address
DNS will convert IP to Hostname and Hostname to IP


which identify the Domain Name
Name Server
SOA
has the IP
.com
Top Level Domain
identity .com, .in etc
Root Name Server
RNS
/etc/host
Local DNS has IP
Local DNS
data pkts
data pkts
data pkts
Allow
Deny
Rule
http
https
FireWall
WebServer
WebServer
WebServer
Google Company
Firewall = Which stops unauthorized access to the Network
Allow / Deny
Browser
HostName / Domain Name
IP
google.com
Vijay's Laptop
DNS = Domain Name Server
It keep track of all Hostname and IP address
DNS will convert IP to Hostname and Hostname to IP
http = 80 / Not Secure connection to the Website
https = 443 / Secure connection to the website
SSH = 22 / Connect to Linux Machines
RDP = 3389 / Connect to Windows Machines


which identify the Domain Name
Name Server
SOA
has the IP
.com
Top Level Domain
identity .com, .in etc
Root Name Server
RNS
/etc/host Local DNS has IP
Local DNS
Browser
Vijay's Laptop
HostName / Domain Name
IP
google.com
data pkts
data pkts
data pkts
Network
Allow
Deny
Rule
http
https
FireWall
LB
Load Balancer
WebServer
WebServer
WebServer
App Server google page
App Server google page
App Server google page
DB Server
DB Server
DB Server
Google Company
Firewall = Which stops unauthorized access to the Network
Allow / Deny
Load Balancer = Which distribute the traffic across multiple Servers
DNS = Domain Name Server
It keep track of all Hostname and IP address
DNS will convert IP to Hostname and Hostname to IP
http = 80 / Not Secure connection to the Website
https = 443 / Secure connection to the website
SSH = 22 / Connect to Linux Machines
RDP = 3389 / Connect to Windows Machines

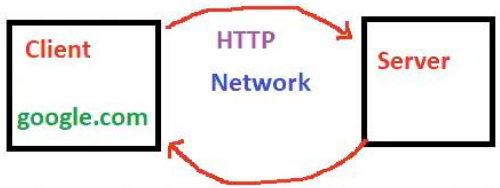
Browser ---> Local DNS ---> Root Name Server ---> Top Level Domain ----> Name Server ---> SOA
Browser ---> Firewall ---> Load Balancer ---> WebServer ---> App Server ---> DB Server

HTTP = HyperText Transfer Protocol

HTTP default Port Number is 80

For Customer it should be always 80 or 443

http://google.com
protocol://domainname:Port



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

Status Codes

404 = Page Not Found
503 = Service Unavailable
500 = Internal Server Error
200 = Page Found / Success

HTTPS = 443

certificates

SSL / TLS / HTTPS

Data is Encrypted

HTTP is Not Secure
HTTPS is Secure

Data Packet

HTTP

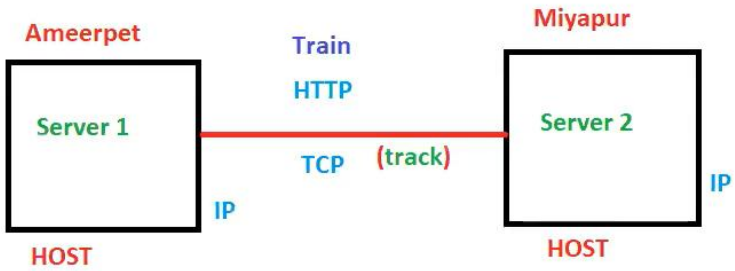
Plain Text / Clear Text

http://192.168.10.20:80
http://192.168.10.20:8080
http://192.168.10.20

Server 192.168.10.20

HTTP Over TCP/IP

Transmission Control Protocol

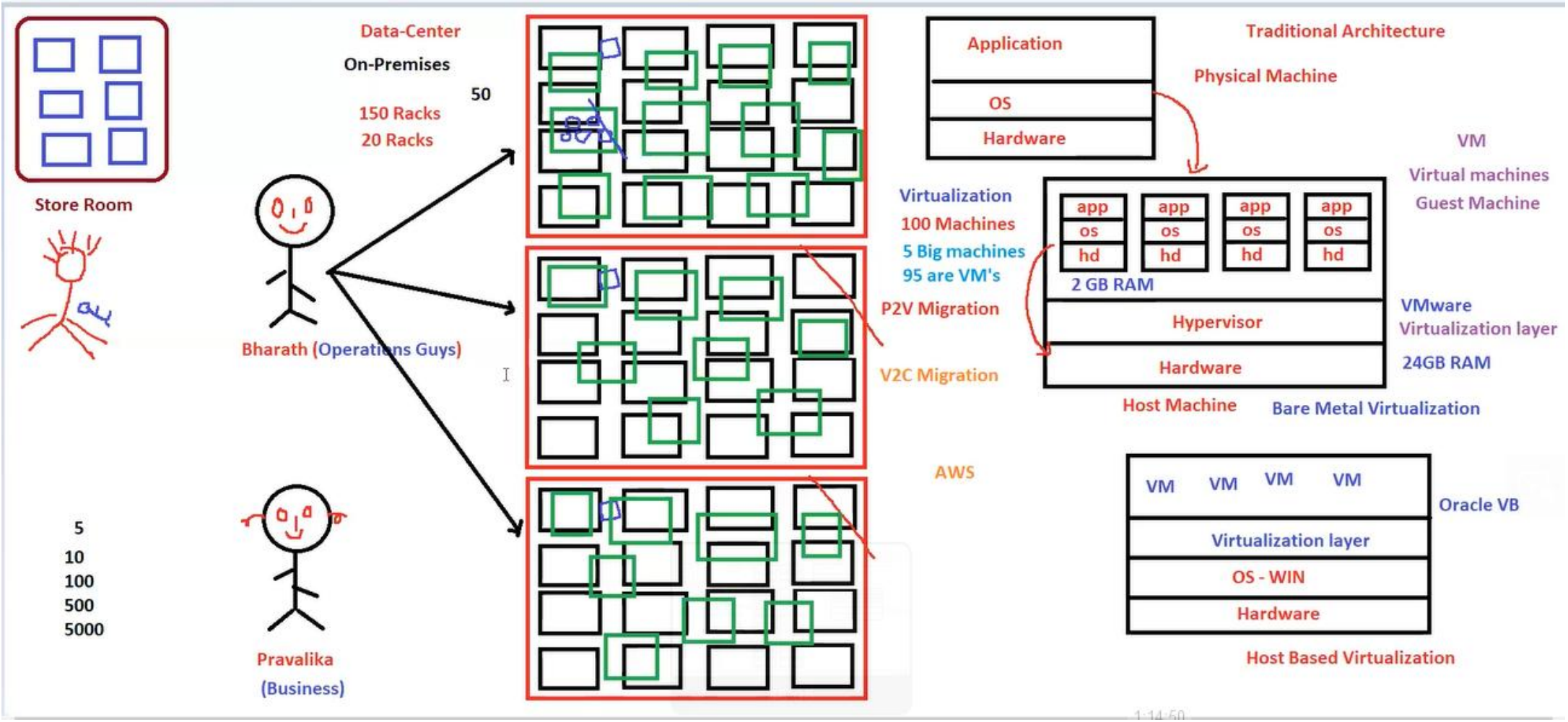


TCP establish the connection between 2 hosts

TCP is like a bridge / Messenger

TCP is reliable
UDP is not reliable

User DataGram Protocol


Store Room
Data-Center
On-Premises
50
150 Racks
20 Racks
Bharath (Operations Guys)
5
10
100
500
5000
Pravalika
(Business)
Application
OS
Hardware
Traditional Architecture
Physical Machine
VM
Virtual machines
Guest Machine
Virtualization
100 Machines
5 Big machines
95 are VM's
P2V Migration
V2C Migration
app
os
hd
2 GB RAM
Hypervisor
Hardware
VMware
Virtualization layer
24GB RAM
Host Machine
Bare Metal Virtualization
AWS
VM
Oracle VB
Virtualization layer
OS - WIN
Hardware
Host Based Virtualization

AWS has Global Infrastructure

AWS is providing Infrastructure as a Service

Cloud is present in the Remote Location

Remote Location = DataCenters

DataCenter = Infrastructure

Infrastructure = Servers, DB's, Network etc

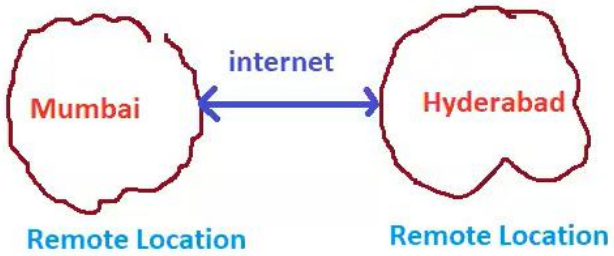
We need internet to connect to the Cloud

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

Amazon Web Services

(AWS Management Console)

AWS is a Group of Services



AWS is Public Cloud Provider, who provides infrastrucutre as a Service

Key-Words

Virtualization, Host Machine, DNS, VM's, Infrastructure, DataCenters, Protocols, Load Balancer, Firewall, Hypervisor, Cloud, Remote Location.

Physical DC ----> Virtualization ----> Cloud --> AWS (Remote Location(DataCenter))

Cloud Computing

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

Deployment Models (Types of Cloud)

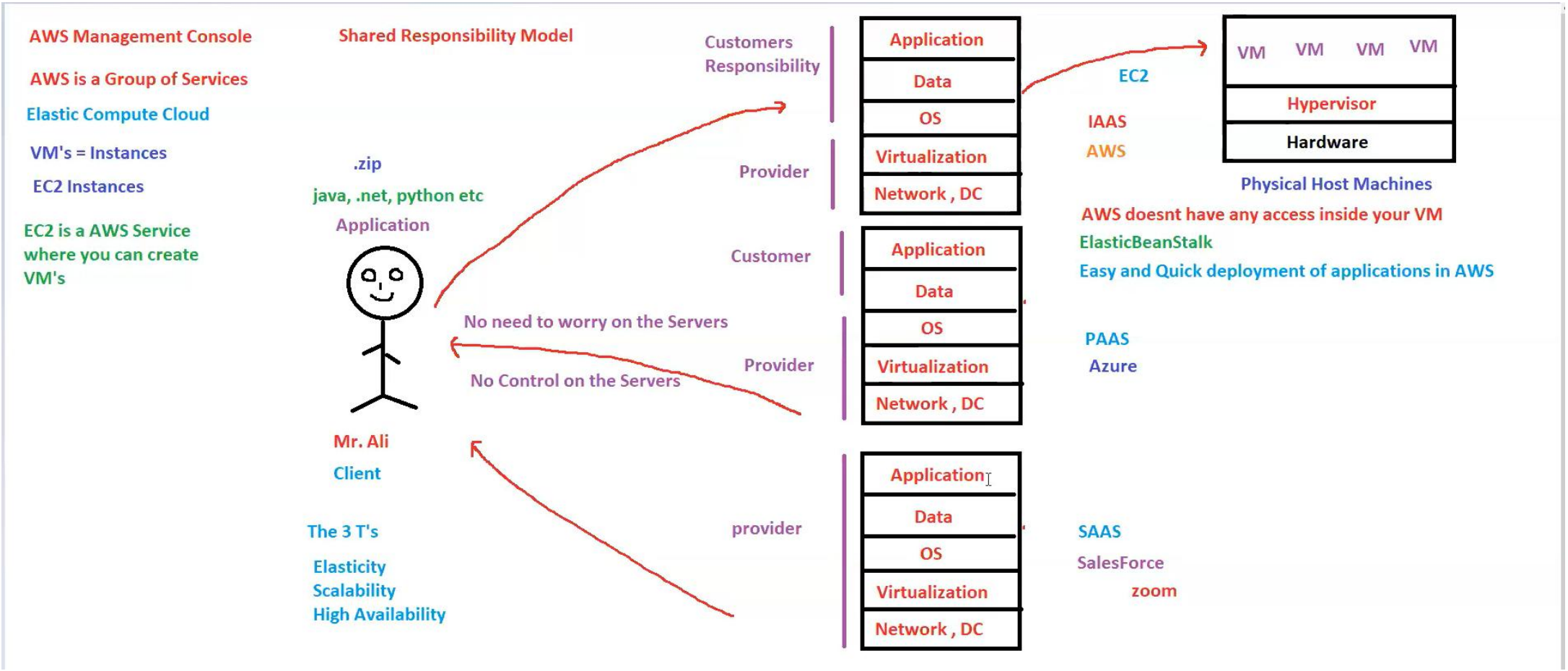
Public Cloud : The Providers Services which are accessed by everyone like AWS, Azure, GCP
Private Cloud : The Providers services which are accessed within the organization like Oracle
Hybrid Cloud : The combination of Public and Private Cloud

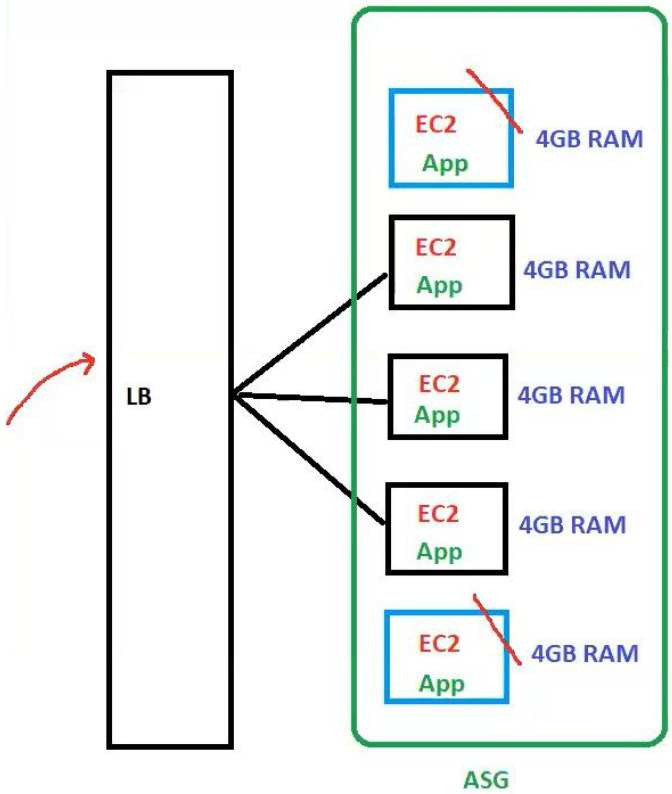
Service Models

Infrastructure as a Service (IAAS)

Platform as a Service (PAAS)

Software as a Service (SAAS)


AWS Management Console
AWS is a Group of Services
Elastic Compute Cloud
VM's = Instances
EC2 Instances
EC2 is a AWS Service where you can create VM's
Shared Responsibility Model
.zip
java, .net, python etc
Application
Mr. Ali
Client
The 3 T's
Elasticity
Scalability
High Availability
Customers Responsibility
Provider
Customer
Provider
provider
Application
Data
OS
Virtualization
Network , DC
Application
Data
OS
Virtualization
Network , DC
Application
Data
OS
Virtualization
Network , DC
No need to worry on the Servers
No Control on the Servers
EC2
IAAS
AWS
VM VM VM VM
Hypervisor
Hardware
Physical Host Machines
AWS doesnt have any access inside your VM
ElasticBeanStalk
Easy and Quick deployment of applications in AWS
PAAS
Azure
SAAS
SalesForce
zoom



Elasticity

Increasing or decreasing the number of servers based on the demand

Elasticity is Short Term

Elasticity can be acheived in AWS using Auto-Scaling

Auto-Scaling = Scale Out and Scale In

increasing adding

decreasing removing

Elasticity is also called as Horizontal Scaling

Use the same capacity of the Servers in AutoScaling Group

Scalability

Increasing the Capacity of the Servers is called Scalability

Scalability = Scale Up and Scale Down

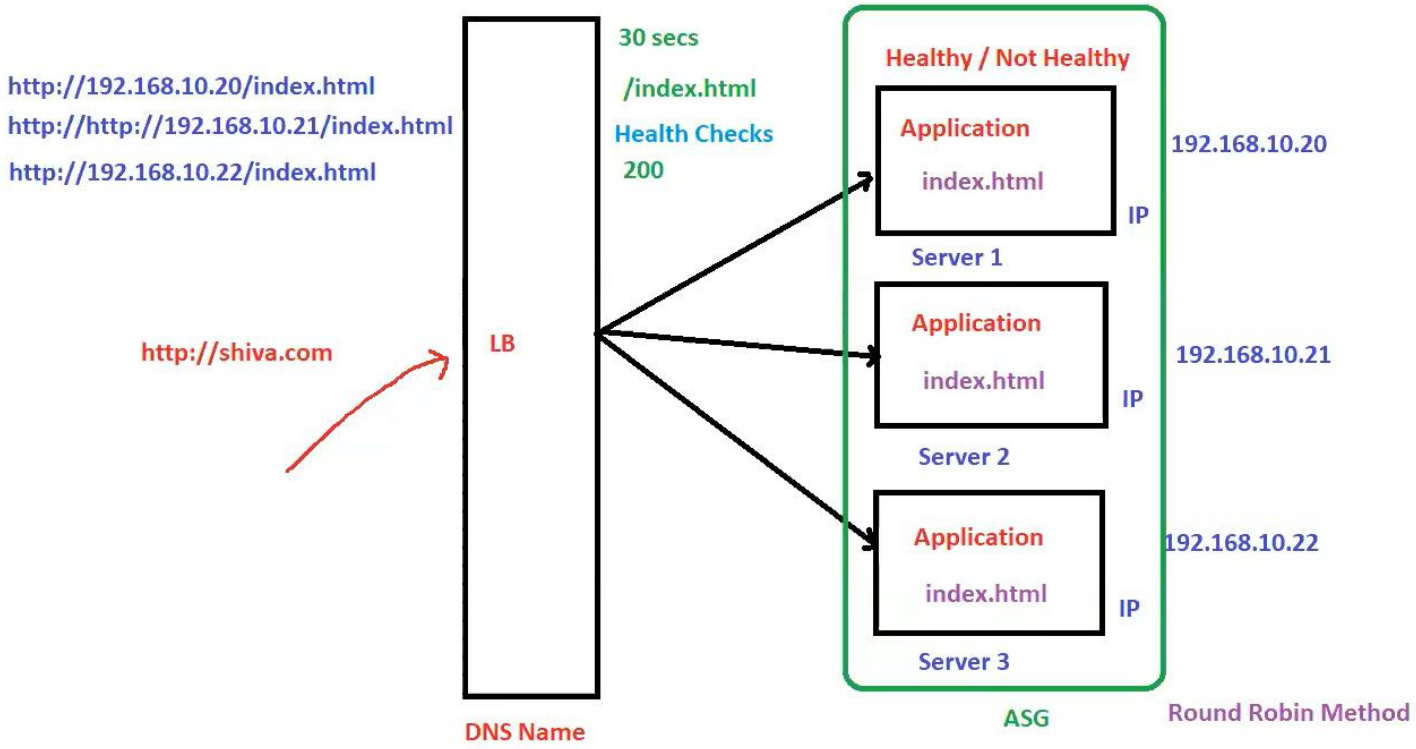
Scalability is Long Term

Scalability is also called as Vertical Scaling

High Availability

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

The period of time the service is NOT Available is called Downtime



LB will do the health checks to the application not Servers

LB is doing Monitoring using Health Checks

LB is doing the Failover

High Availability

RMF

Redundancy : Duplicate / having same application on diff machines

Monitoring : LB will check if application is reachable or not using Health Checks

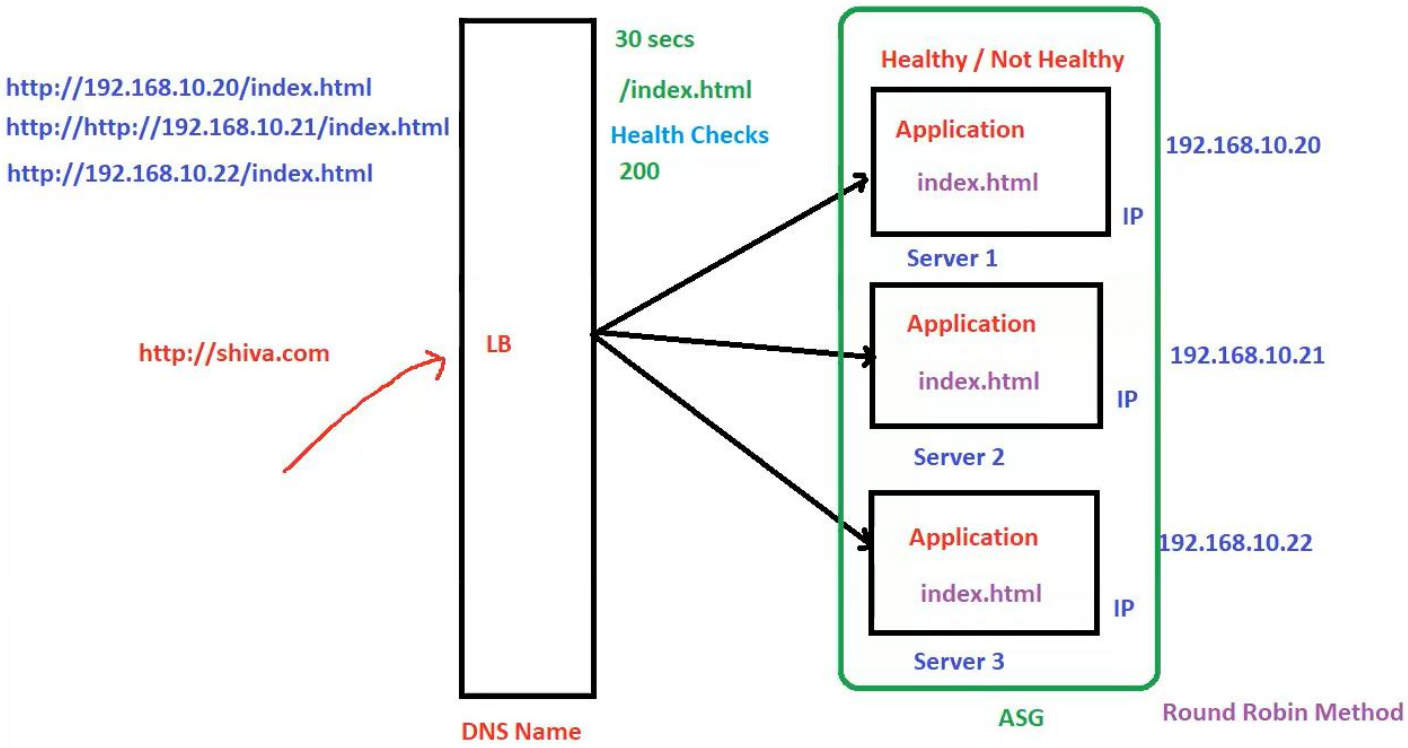
Failover : If one server goes down, other server will take the requests sent by LB

0 downtime = Auto-Scaling = Fault Tolerance

High Availability

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

The period of time the service is NOT Available is called Downtime



LB will do the health checks to the application not Servers

LB is doing Monitoring using Health Checks

LB is doing the Failover

High Availability

RMF

Redundancy : Duplicate / having same application on diff machines

Monitoring : LB will check if application is reachable or not using Health Checks

Failover : If one server goes down, other server will take the requests sent by LB

0 downtime = Auto-Scaling = Fault Tolerance

AWS has Global Infrastructure

A Region is a place where AWS has its Infrastructure

A Region Contains multiple DataCenters

A Region has multiple AZ's

Servers = Instances

Servers / Instances are placed in AZ's

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

very very less chance that 1 AZ does down

1a or 1b or 1c = Group of DC's

1AZ is a group of DC's

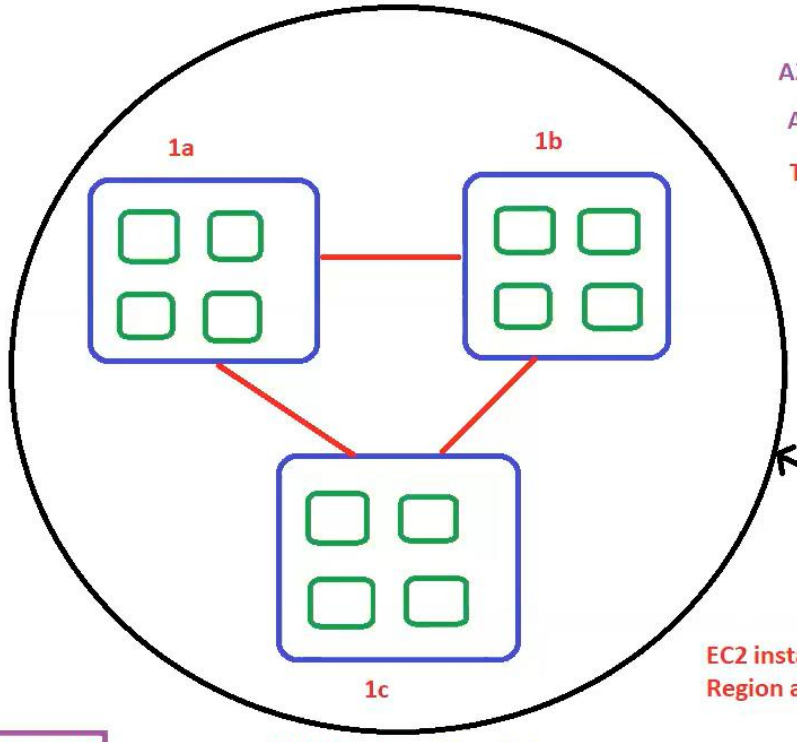
Instances across AZ's can share the data if required as AZ"s network are inter-connected

1a	1b	1c
dadar, andheri, aroli	thane, bandra, kalyan	juhu, kurla, vashi

Regions and Availability Zones

Region = Its a Geo-Graphical area, Ex AWS-Region = Mumbai

Availability Zone = Simply a DataCenter (AZ)



AWS - Region = Mumbai

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 communicate with each other by default

AZ's network are inter-connected

There are 32 Regions in AWS at the moment

Low Latency = Good

High Latency = Bad

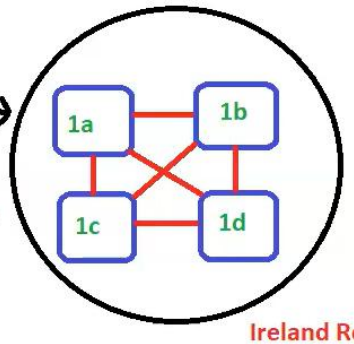
Latency is depend upon the Response time

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

Network

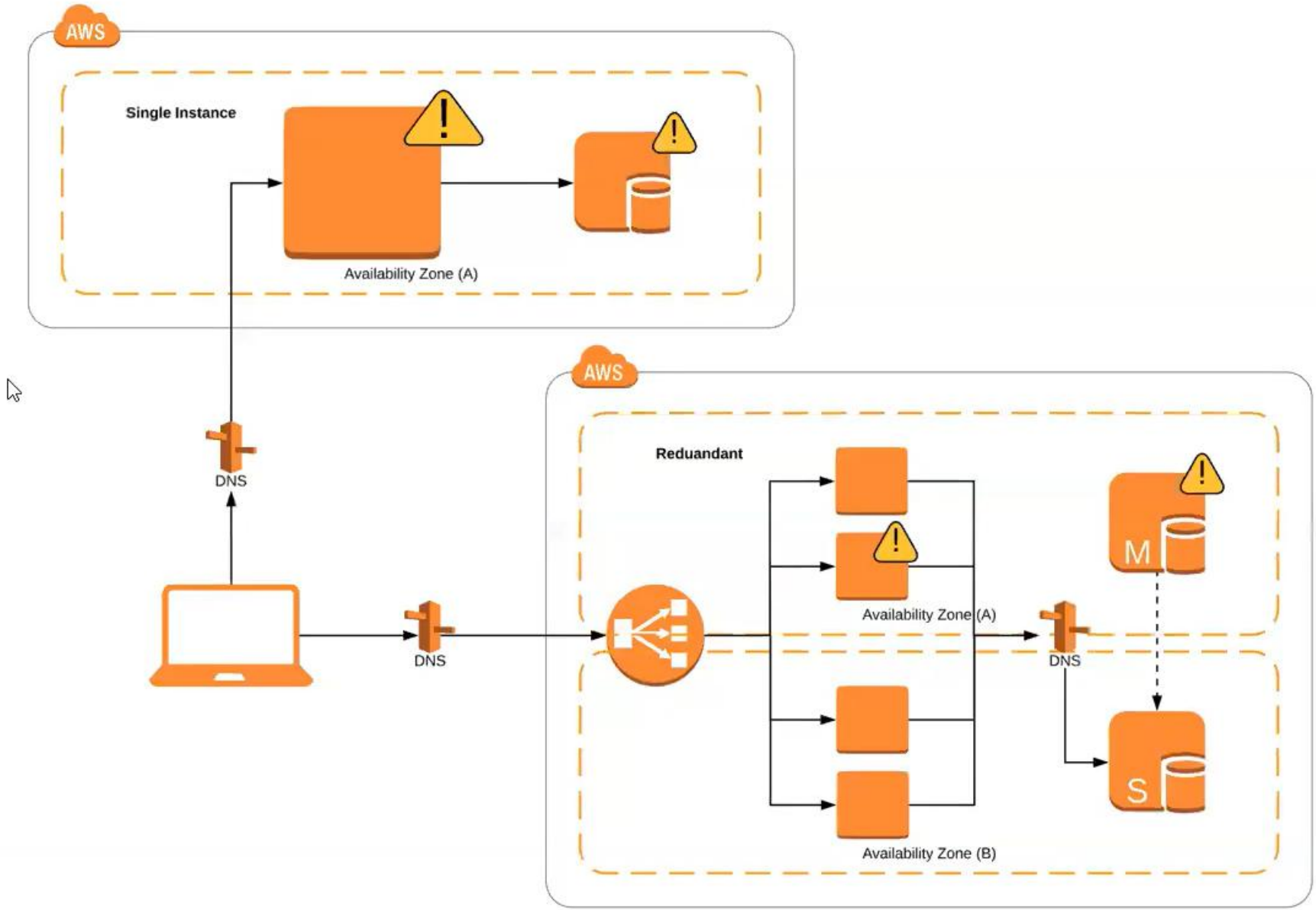
NO

default



Ireland Region

EC2 instance is specific to Region and AZ.


AWS
Single Instance
Availability Zone (A)
DNS
AWS
Reduandant
DNS
Availability Zone (A)
M
DNS
S
Availability Zone (B)

AWS Services can be either Regional or Global

EC2 is Regional Service

EC2 = Elastic Compute Cloud

In EC2 Service, we can launch EC2 instances

Servers = Instances / EC2 instances (VM's)

Load Balancer = Which distribute the traffic to the servers

Elastic Load Balancer (ELB) = Which distribute the traffic to multiple EC2 instances across AZ's

ELB is completely managed by AWS (HA, AS, Scalability, Performance etc)

ELB is a Service from AWS, not Server

You cannot login to ELB, you can access ELB with DNS Name

ELB doesnt have AZ's, it is created at Regional Level

ElasticBeanStalk = Easy and Quick deployment of applications in AWS

In General, PAAS ---> you dont have access and control on the Servers

In AWS BeanStalk --> You have full control on EC2 instances launched by BeanStalk

BeanStalk Handles EC2 instances (OS) behalf of us

The backbone of BeanStalk is EC2

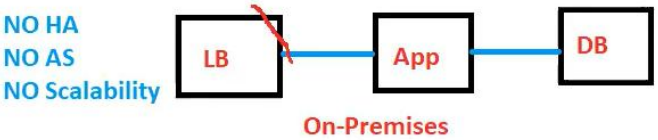
LightSail = If you want to setup and create a virtual lightsail instance which already has everything installed (Wordpress, Gitlab, Nodejs, Joomla, Drupal, Redmine, nginx, Cpanel etc)

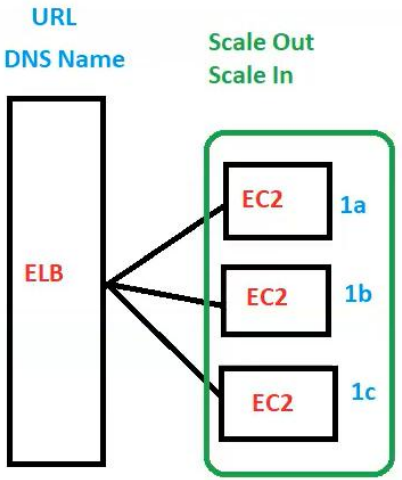
It doesnt support Auto-Scaling

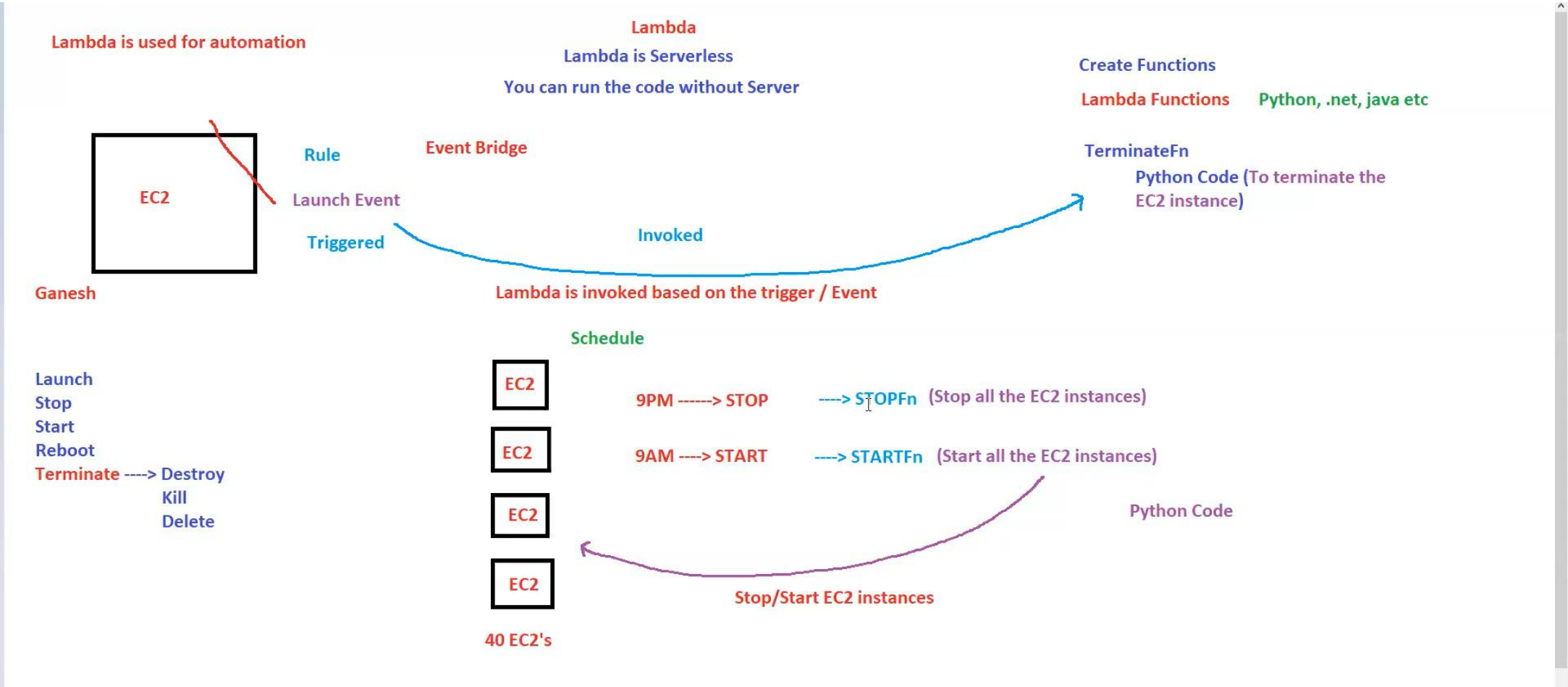
Give it some configurations

Upload the application

select the Platform






Lambda is used for automation
Lambda
Lambda is Serverless
You can run the code without Server
Create Functions
Lambda Functions
Python, .net, java etc
EC2
Rule
Event Bridge
Launch Event
Triggered
Invoked
TerminateFn
Python Code (To terminate the EC2 instance)
Ganesh
Lambda is invoked based on the trigger / Event
Schedule
Launch
Stop
Start
Reboot
Terminate ----> Destroy
Kill
Delete
EC2
EC2
EC2
EC2
9PM ------> STOP
----> STOPFn (Stop all the EC2 instances)
9AM ----> START
----> STARTFn (Start all the EC2 instances)
Python Code
Stop/Start EC2 instances
40 EC2's

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

SNS ---> Simple Notification Service

SES ---> Simple Email Service

S3 = Simple Storage Service

S3 is unlimited Storage

S3 is used to just store the files

S3 can store all FLAT files

With S3, we can upload, download and access files

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 install, run, execute any files in S3 ? NO

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

S3 is Object Based Storage

S3 is Serverless

AWS handles HA, Performance, Scalability etc for S3

S3 Support Static Website Hosting

Create a bucket, put all your files inside the bucket and enable static website hosting

No need to worry about HA, Performance, Scalability etc because S3 handles it

Laptop	
Windows	S3
Folder	Bucket
Files	Objects
jawan.mp4	KEY

Bucket is a container of Objects

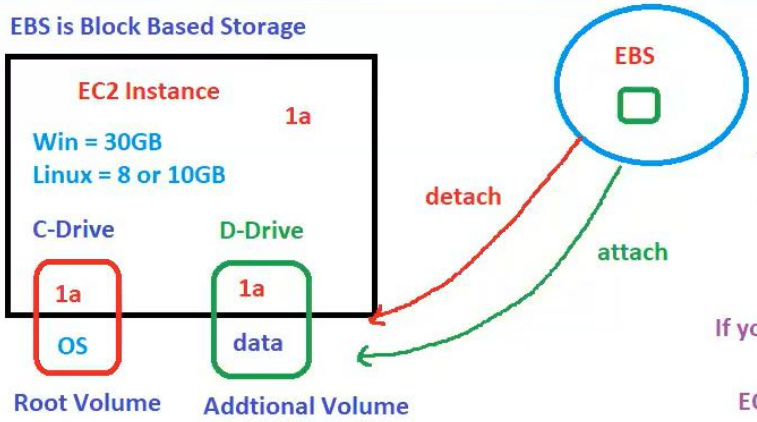
Object is File

Name of the file / Object is a KEY

S3 is Global

Buckets are Regional

EBS is Block Based Storage



device names

For Win / Linux - /dev/sda1 /dev/sdb,e,f,g etc

unbuntu - /dev/xvda xvdf, xvdg etc

Root Volume is always mounted 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 default volume, that volume is called ROOT Volume

The ROOT volume always has OS (Windows, Linux)

EC2 supports only Server Side OS not Client Side OS

If you have OS on the Volume, the Volume is called ROOT Volume

EC2 instance can have only 1 Root Volume

EC2 instance can have multiple addtional volumes

Max Size of the EBS 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 *

Volume Size can be increased on FLY (NO need to stop the EC2 instance, no downtime)

Volume size cannot be decreased (delete the vol and re-create it based on your size requirement)

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 Addtional volume while EC2 is running ? YES : It is not recommended to detach while running, Stop first

You cannot delete the volume while it is attached, detach first and then delete

EC2 instance has AZ, Volume also have AZ

EC2 instance and Volumes should be in the same AZ

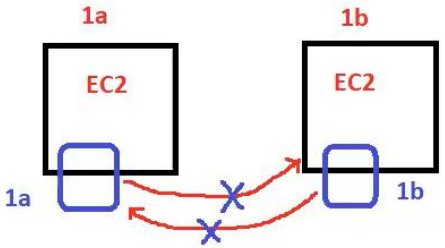
OS

Client Side

win 10, 11 ✗

Server Side OS ✓

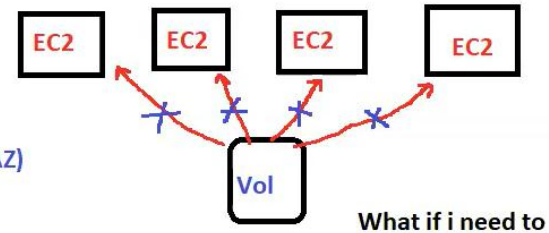
win server 2020, 2021, 2022, 2021 RedHat, CentOS, SUSE etc



1a vol can be attached to 1b EC2 instance ? NO (diff AZ)

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

We can attach 1a volume to 1a another EC2 instance (Same AZ)

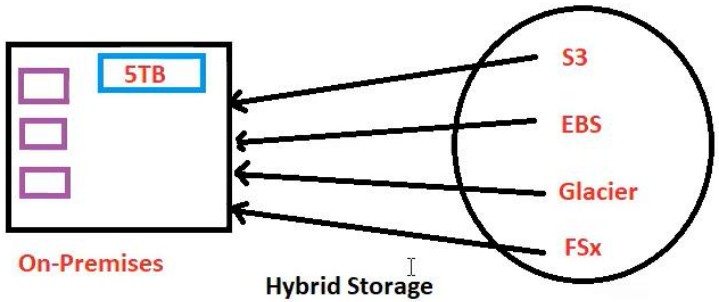


You cannot share the volumes among EC2 instances

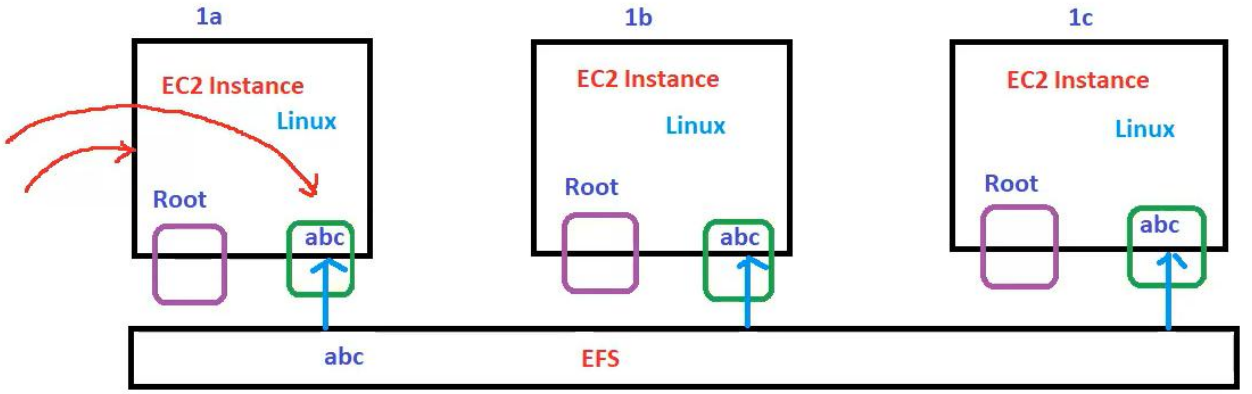
What if i need to have shared storage ?

Is it possible to attach a single Volume to multiple EC2 instances at the same time ? NO

Storage Gateway



Elastic File System



SNOW Family

Import/Export Service
Snowball

SnowCone ----> 8TB
SnowEdge ----> 100TB
SnowMobile ---> PB's

------> S3

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 doesnt require any pre-provisioning (it will automatically increase and decrease 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

Not Frequently used data

Cheaper than S3

SnowFamily is used to transfer huge data from on-prem to AWS and vice-versa

SnowFamily is a physical data transfer using devices

On-Premises

Physical Server

DBA

NO HA
NO Elasticity
Scalability issues
Performance issues
DB Crash
Backup issues
Restoration issues
Storage issues
Upgrade issues
Migration issues
replication issues
mirroring issues
loogin issues
security issues
failover issues
data issues netowrking issues
maitnenace issues
cost issues
patching issues
Headache issues

Database Services

RDS = Relational Database Services

RDS is a Service where you can setup, configure, maintain , secure the RDBMS Databases

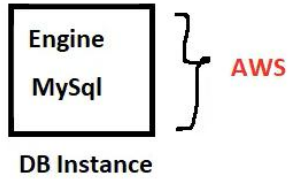
RDS is a Database Service, not database

RDS DB isntance, RDS Support only RDBMS Databases

RDS Supports 6 Engines

MOMPMA

- MySql ---> Opensource
- Oracle ---> Oracle
- MSSQL ---> Microsoft
- PostGreSql ---> Opensource
- MariaDB ---> Community
- Aurora ---> AWS



DB Instance

Can we do RDP, SSH to RDS DB Instance ? NO

Platform is managed by AWS

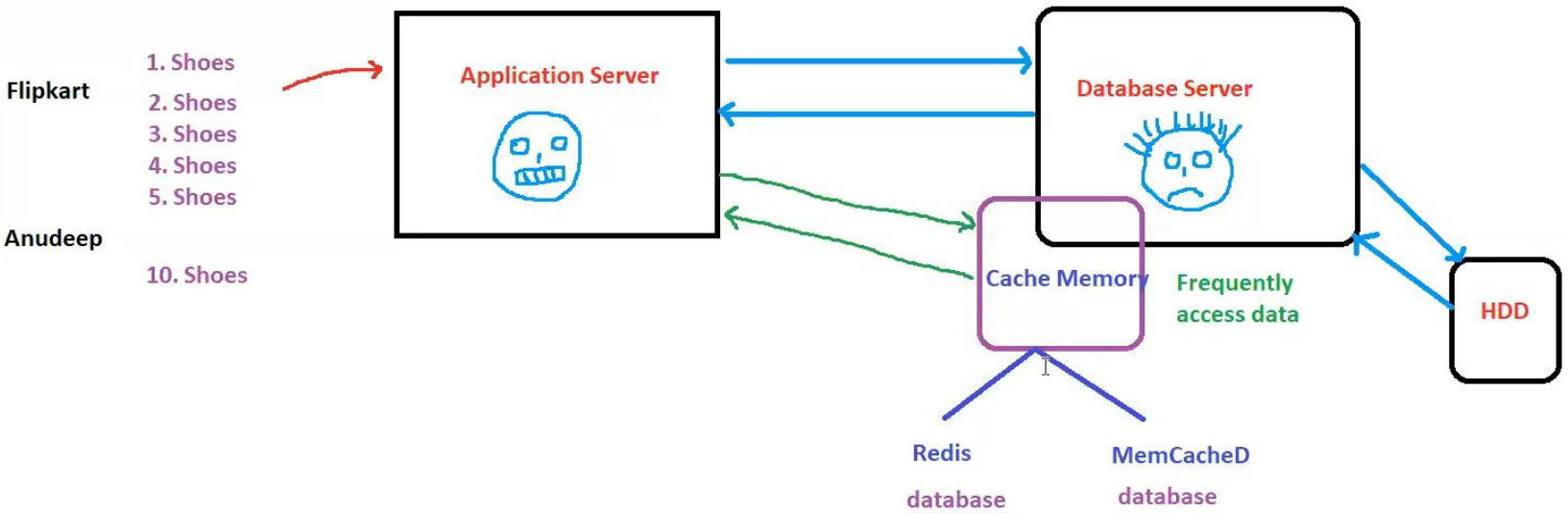
DynamoDB ---> NoSql Database Service (Developers) [Non-Relational]

Database = It is used to store the data

Datawarehouse = It is used to store the HUGE Data

RedShift = Datawarehouse in AWS

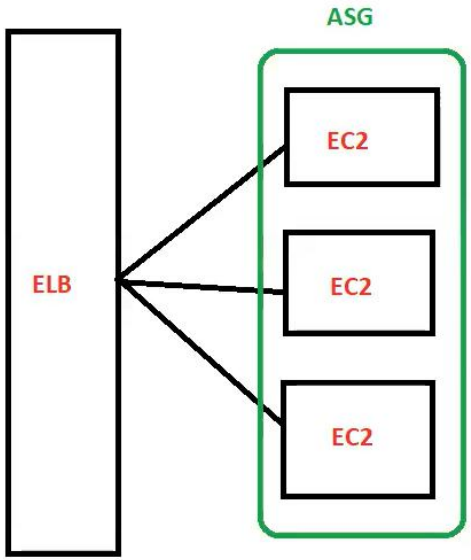
ElastiCache = In-Memory Database Caching Service
= Low Latency
= High Performance



Route53

R53 is a DNS Service from AWS, DNS Port Number is 53

R53 contains Records

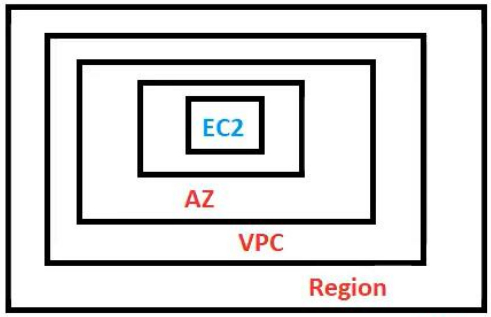


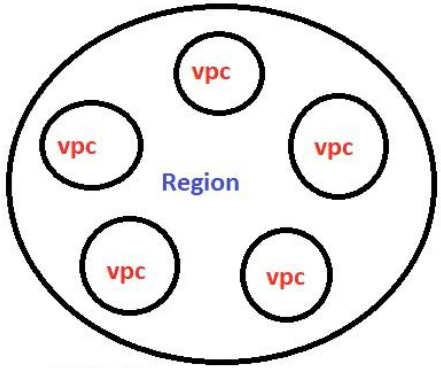
http://myelb.876786876.elb.ap-south-1.amazonaws.com

boom.com ----> Nasty URL mapping

VPC = Virtual Private Cloud

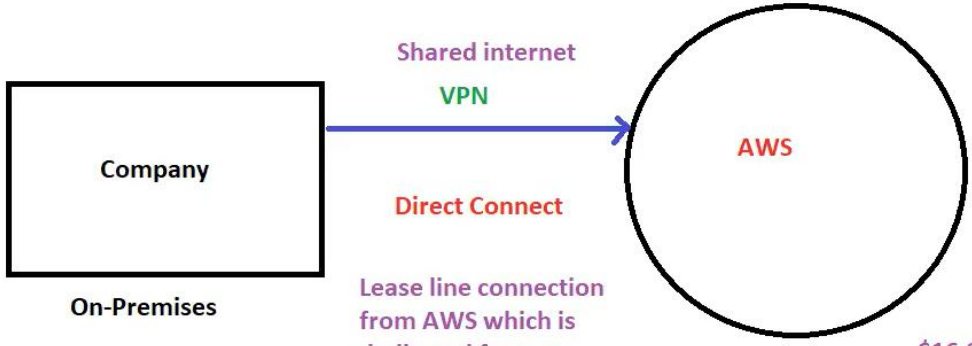
It is like a virtual datacenter on Cloud





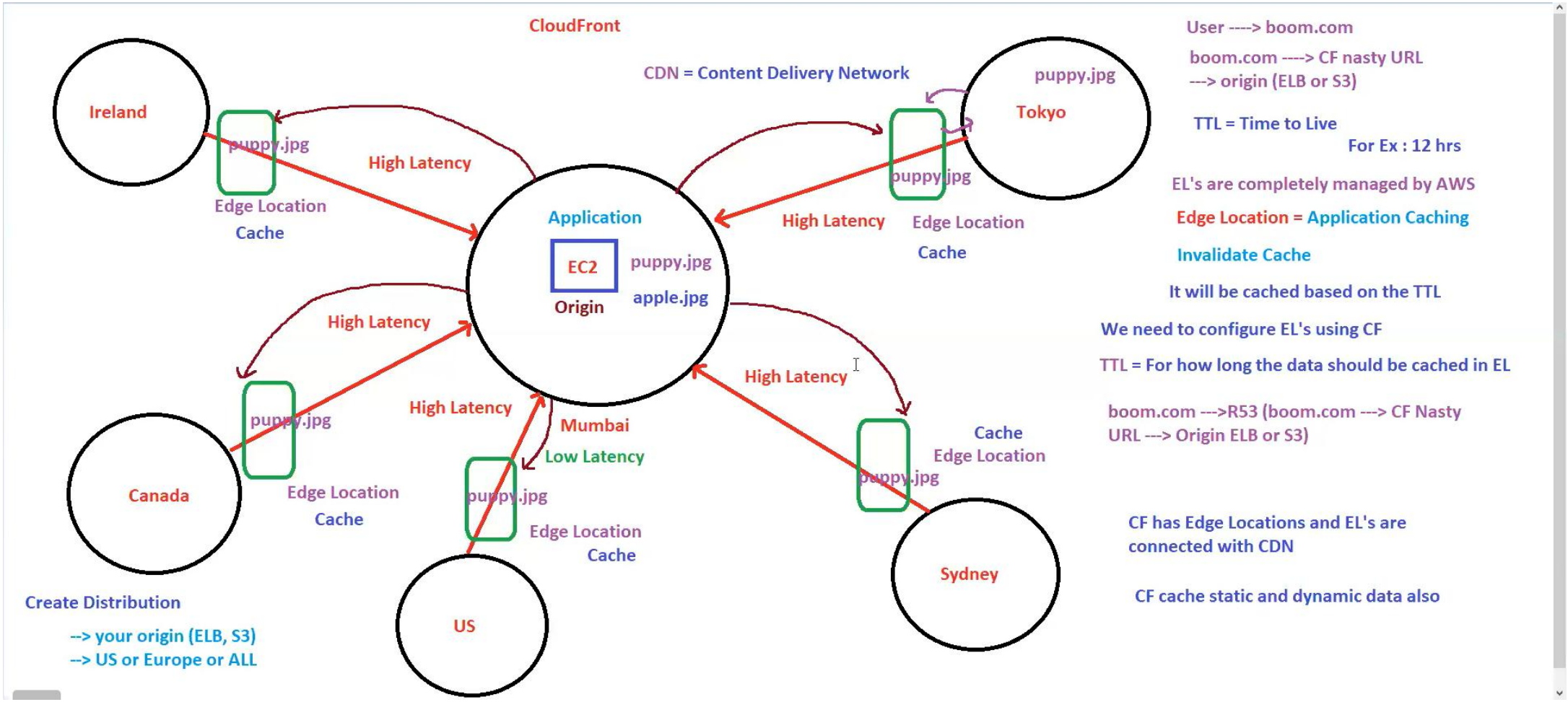
VPC is Regional

AWS provides a default VPC



Lease line connection from AWS which is dedicated for you

$16,000


CloudFront
CDN = Content Delivery Network
Ireland
puppy.jpg
High Latency
Edge Location
Cache
Application
EC2
Origin
puppy.jpg
apple.jpg
Tokyo
puppy.jpg
puppy.jpg
High Latency
Edge Location
Cache
High Latency
puppy.jpg
Canada
Edge Location
Cache
High Latency
Mumbai
Low Latency
puppy.jpg
Edge Location
Cache
US
High Latency
Cache
Edge Location
puppy.jpg
Sydney
Create Distribution
--> your origin (ELB, S3)
--> US or Europe or ALL
User ----> boom.com
boom.com ----> CF nasty URL
---> origin (ELB or S3)
TTL = Time to Live
For Ex : 12 hrs
EL's are completely managed by AWS
Edge Location = Application Caching
Invalidate Cache
It will be cached based on the TTL
We need to configure EL's using CF
TTL = For how long the data should be cached in EL
boom.com --->R53 (boom.com ---> CF Nasty
URL ---> Origin ELB or S3)
CF has Edge Locations and EL's are
connected with CDN
CF cache static and dynamic data also

IAM = Identity and Access Management

You can control the entire AWS using IAM by providing proper permissions to the IAM Users

Users / Accounts

Root User

Root User has Full Permissions / Super user

Login with Email / Pwd

IAM User

IAM user has limited Permissions

Login with Username / Pwd

Root Account

1 Company Account

IAM Users

Bharat ---> EC2
Harshi --> S3
Mahesh --> VPC
Nirnitha --> R53
Bhaskar --> Admin (Except Billing)

Parent Account (Root Account)

IAM User 1
IAM User 2
IAM User 3
....
....

IAM = Access Control on AWS Resources to the Users

Organizations

1 Root Account

Management Account

Company (Root Account)

SCP ---> Service Control Policy

Member Accounts

DevOps	Developer	Support	Testing
Root Account	Root Account	Root Account	Root Account
IAM	lambda	Admin	R53

Organization handles / controls multiple ROOT accounts for better management

SNS ---> Simple Notification Service

SES ---> Simple Email Service

SQS ---> Simple Queue Service
(First Service launched by AWS)

Trusted Advisor

AWS Inspector

EC2 Instance

4GB RAM
2 CPUs

B: 6:00 6:05 6:10 6:15
D: 6:00 6:01 6:02 6:03

CloudWatch

CPU > 90%

SNS
---> Email
---> SMS

CloudWatch

CloudWatch is used to monitor all AWS Resources (EC2, ELB , S3, RDS etc)

In CloudWatch we create Alarms to monitor AWS resources

CloudWatch monitors the performance

Basic Monitoring : You will get the data points every 5 mins, FREE [default]
Detailed Monitoring : You will get the data points every 1 min, Billable

CloudTrail : Records all activities happening in AWS account

Event Logger / Investigation Purpose

Config : Monitors the changes in AWS Resources

ALARMS

Metrics (CPU, Network, Disk, Requests etc)

AWS Support

Basic Support ----> FREE
Developer Support
Business Support
EnterPrise Support
$15,000

IAM - Deep Dive

IAM is Free

IAM is Global

Using IAM User, we can access the AWS Console

IAM is used for Security Purpose

With IAM, you can control entire AWS resources centrally

Dont share your email/pwd to others

You can share the ROOT account by creating IAM user

It is not recommended to use the ROOT account for daily activities or work, Instead use IAM User

MFA ---> Multi Factor Authentication (Google Authenticator)

MFA is highly recommended for ROOT and IAM users as well

We need to setup MFA for every individual IAM user

Open AWS Page ---> Login with Email/Pwd ---> MFA Code ---> Login to AWS Console

Root User ---> Full Permissions

IAM User ---> Limited Permissions

Permissions ---> Policies

IAM Users

Sowmya ---> EC2

Venky ---> S3 , IAM

Surya ---> VPC

Rizwan --> RDS

Mahesh --> Admin (Except Billing)

2 Ways to Access AWS

IAM User
Ganesh
EC2

MFA
(IAM User/Pwd)

Console Access
AWS Console Website (GUI
(Email/Pwd , Username/Pwd)
Root IAM

Console

Programmatical
KEYS NO MFA

Console access can also be Disabed and Enabled

Programmatical Access
(CLI, SDK's, Developer Tools)

Access Key : AKJDFHAJKDHFSJKDHF
Secret Key : S^D&*SFD^*%S*D%FD
} 1 Set

Login with KEYS (Access Key and Secret Key)

KEYS are user specific, individual IAM User have their own KEYS

It is not recommended to share the KEYS to anyone

Create the KEYS based on the requirement, dont create it unnecessarily

KEYS also have same permissions like Console

Every IAM User can have max 2 Set of KEYS

We create the KEYS and AWS provides the KEYS

We can make the KEYS inactive based on the requirement

Once Keys are lost , it is lost , you cannot get the same keys back

But you can re-generate N Number of times

If you re-generate, you will get the new KEYS, you cannot get the old keys back

DONT Create KEYS and use the KEYS of ROOT Account

IAM Groups

IAM Group = Collection of IAM Users

Group under Group is not possible / Nested Groups are not possible

It is possible to attach multiple policies to the IAM user and IAM Groups also, Max 10

You can add/attach and remove/detach policies to the IAM users and Groups any time

If you attach any IAM user to the IAM Group, IAM user individual polices remains same, and the new policies will be inherited to the IAM User.

You cannot assign / Create KEYS to the Groups

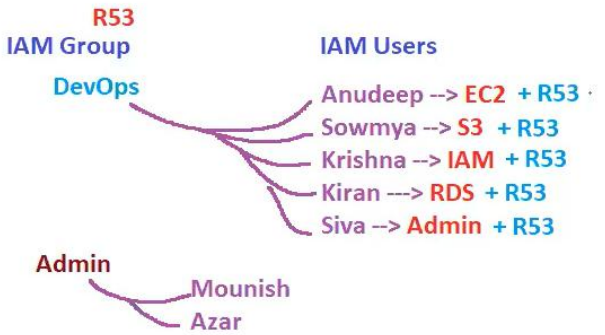
KEYS are only for IAM users and not for IAM Groups

An IAM user can be attached to mutiple IAM Groups at the same time

IAM groups are used to assign policies to the bunch of IAM user at the same time

A Brand new IAM user will not have any Permissions / Policies attached by default

User Based Permissions	Resource Level Permissions
EC2fullaccess S3readaccess	Granular Level / Deeper Level
Managed	Customized Permissions
	Inline



Policy ---> Policy contains Permissions

Permissions / Policies are written in JSON Format

Managed Policy : Created and managed by AWS (Predefined Policies)

EC2Fullaccess
EC2Readaccess

Inline Policy : Created and managed by Customer (Customized Policies)

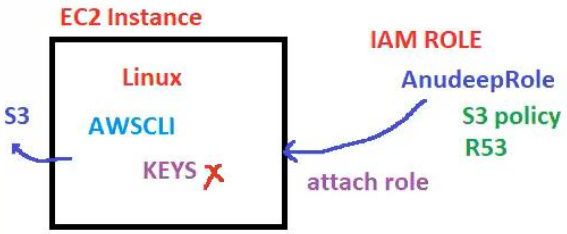
Customer Managed Policy

Visual Editor / Policy Generator

--> Json code generated automatically

ARN = Amazon Resource Name

To identify a AWS Resource to put in IAM Policy



Instance IP
Instance Username
Instance Password

IAM user is used to access AWS Console Not Servers / EC2 instance

If you dont configure the KEYS on Linux EC2 instance , you cannot access AWS Services

If you configure the KEYS on EC2 instance, KEYS are stored locally on the EC2 instance which is not SAFE (Because might be hacked)

IAM ROLES

ROLE = Temporary access without Credentials

If you use the ROLES, we no need to configure the KEYS On the EC2 instance

Based on permissions, that you have attached to the ROLE, those permissions are available from the EC2 instance

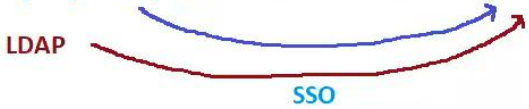
1 EC2 instance can have only 1 ROLE attached at the same time

1 ROLE can be attached to multiple EC2 instances at the same time

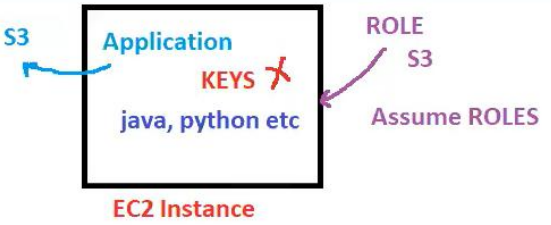
Identity Provider / Federation / Identity Center

SSO - Single Sign On no need to create IAM user

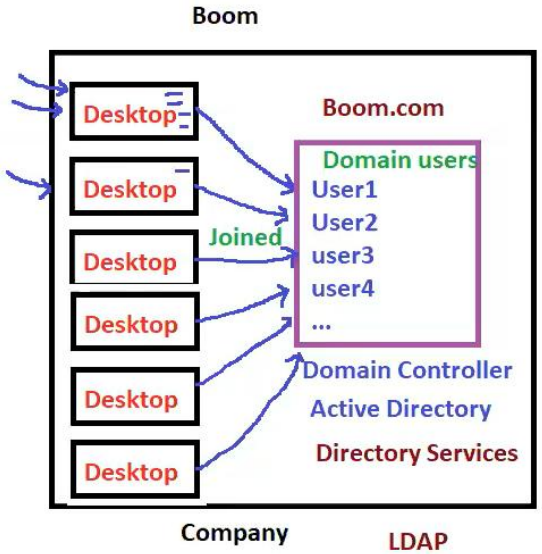
Company Email / Pwd ---> AWS Credentials ---> Login to AWS

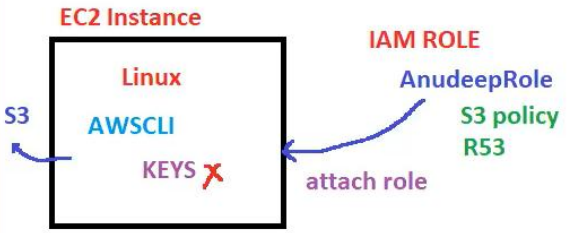


LDAP
OAuth
Okta



Some AWS Services will need to perform actions on your behalf





Instance IP
Instance Username
Instance Password

IAM user is used to access AWS Console Not Servers / EC2 instance

If you dont configure the KEYS on Linux EC2 instance , you cannot access AWS Services

If you configure the KEYS on EC2 instance, KEYS are stored locally on the EC2 instance which is not SAFE (Because might be hacked)

IAM ROLES

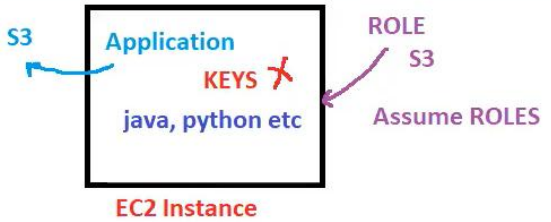
ROLE = Temporary access without Credentials

If you use the ROLES, we no need to configure the KEYS On the EC2 instance

Based on permissions, that you have attached to the ROLE, those permissions are available from the EC2 instance

1 EC2 instance can have only 1 ROLE attached at the same time

1 ROLE can be attached to multiple EC2 instances at the same time



Some AWS Services will need to perform actions on your behalf

Identity Provider / Federation / Identity Center

SSO - Single Sign On no need to create IAM user

Company Email / Pwd ---> AWS Credentials ---> Login to AWS

LDAP

SSO

Federation to AWS (Identity Provider)

(Connection From LDAP to AWS)

LDAP
OAuth
Okta

LDAP

Username/Pwd

IAM ROLES

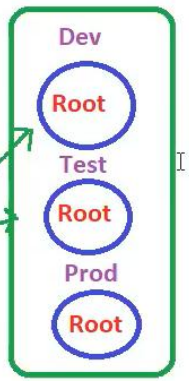
Developer (Lambda)
Admin
DevOps (EC2 etc)

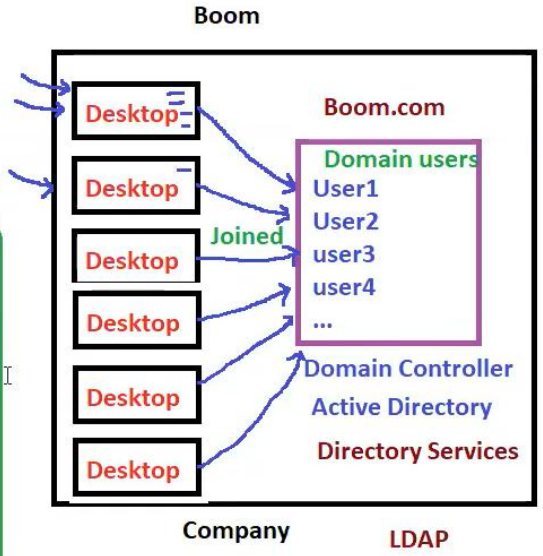
Identity Center

Rahul
User

SSO

Has a URL





EC2

Name = WebServer

Key = value

IAM - TAGS

TAGS are Key-Value Pair

TAGS are used for identification purpose

TAGS are not IAM Specific, It is through out AWS

TAGS are very important, but it is Optional

TAGS are helpful for doing automations in AWS

Jean

TAG

Price = 10k
Brand = Killer
Size =

Key = Value

500 IAM Users

DevOps

Admins

Security

DBA ✗

Hardworkers

Manually
500 users
delete one by one

Smart Workers

delete all IAM users where
Key = Name, Value = DBA

Python

Access Analyzer

Access analyzer analyze the access for all IAM users and gives actions to act on it

It is a best practise to change/rotate the Passwords and Keys for the IAM users for security purpose

Access Advisor

. It shows the service permissions granted to the users and when those were last used
. It is used to revise your policies to the Users

100 EC2 instances

50 DevOps
Name = DevOps

TAGS are helpful for doing Cost Optimization also

Max Tags per resource is 50

IAM Credentials Report

A report that list all your accounts users and the status of their various credentails

IAM Service

IAM Users
IAM Groups
IAM Policies
IAM ROLES
Federation/ IC

Organizations

For managing multiple Root accounts

Root Account

Management Account

SCP

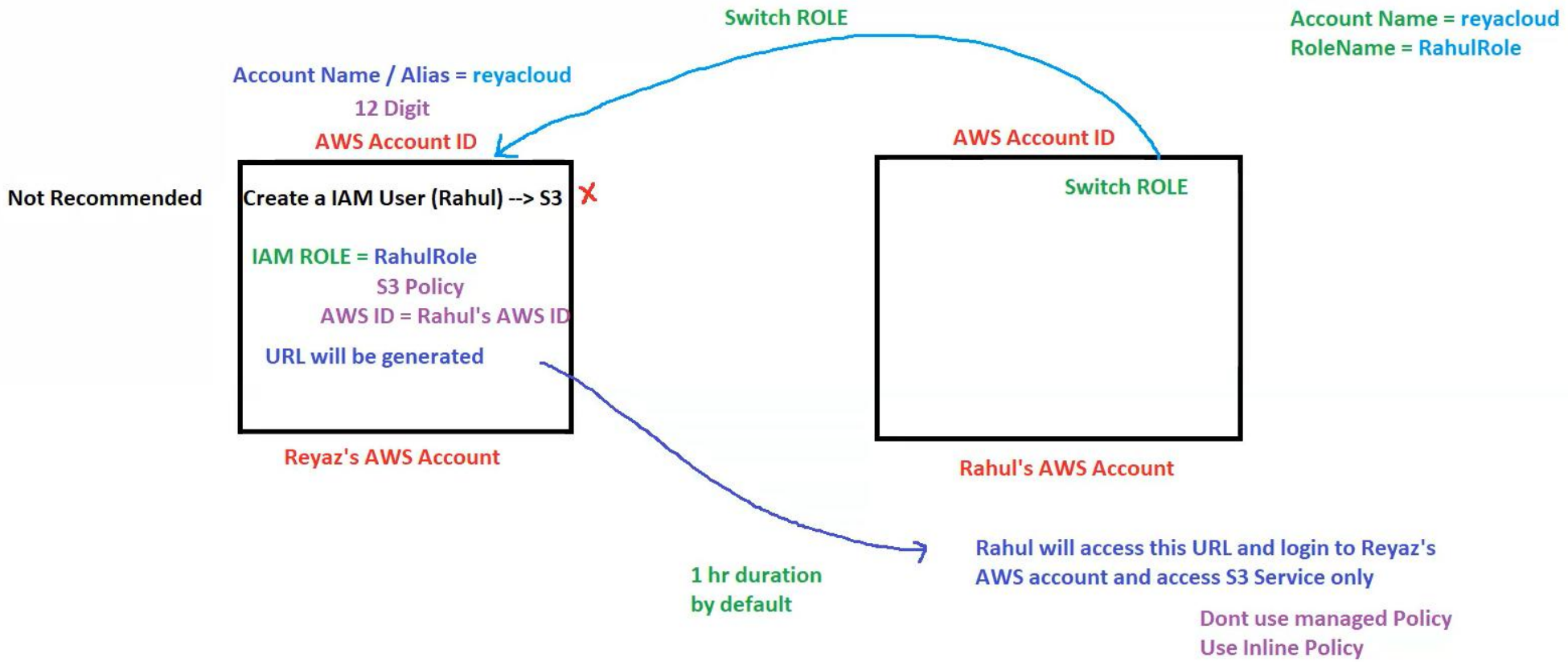
Root

Root

Root

Root

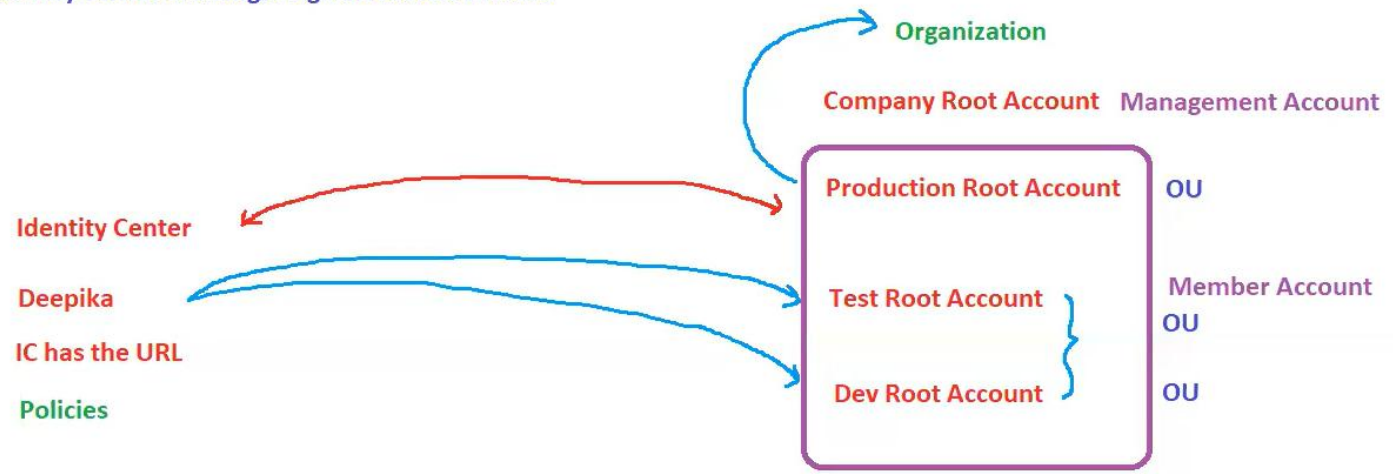
Member Accounts


Switch ROLE
Account Name = reyacloud
RoleName = RahulRole
Account Name / Alias = reyacloud
12 Digit
AWS Account ID
AWS Account ID
Switch ROLE
Not Recommended
Create a IAM User (Rahul) --> S3
IAM ROLE = RahulRole
S3 Policy
AWS ID = Rahul's AWS ID
URL will be generated
Reyaz's AWS Account
Rahul's AWS Account
1 hr duration
by default
Rahul will access this URL and login to Reyaz's
AWS account and access S3 Service only
Dont use managed Policy
Use Inline Policy

Identity Center

Single Sign On

Identity Center is a Single Sign on Service in AWS



Employee

Deepika

By default Organizations and Identity Center are not enabled

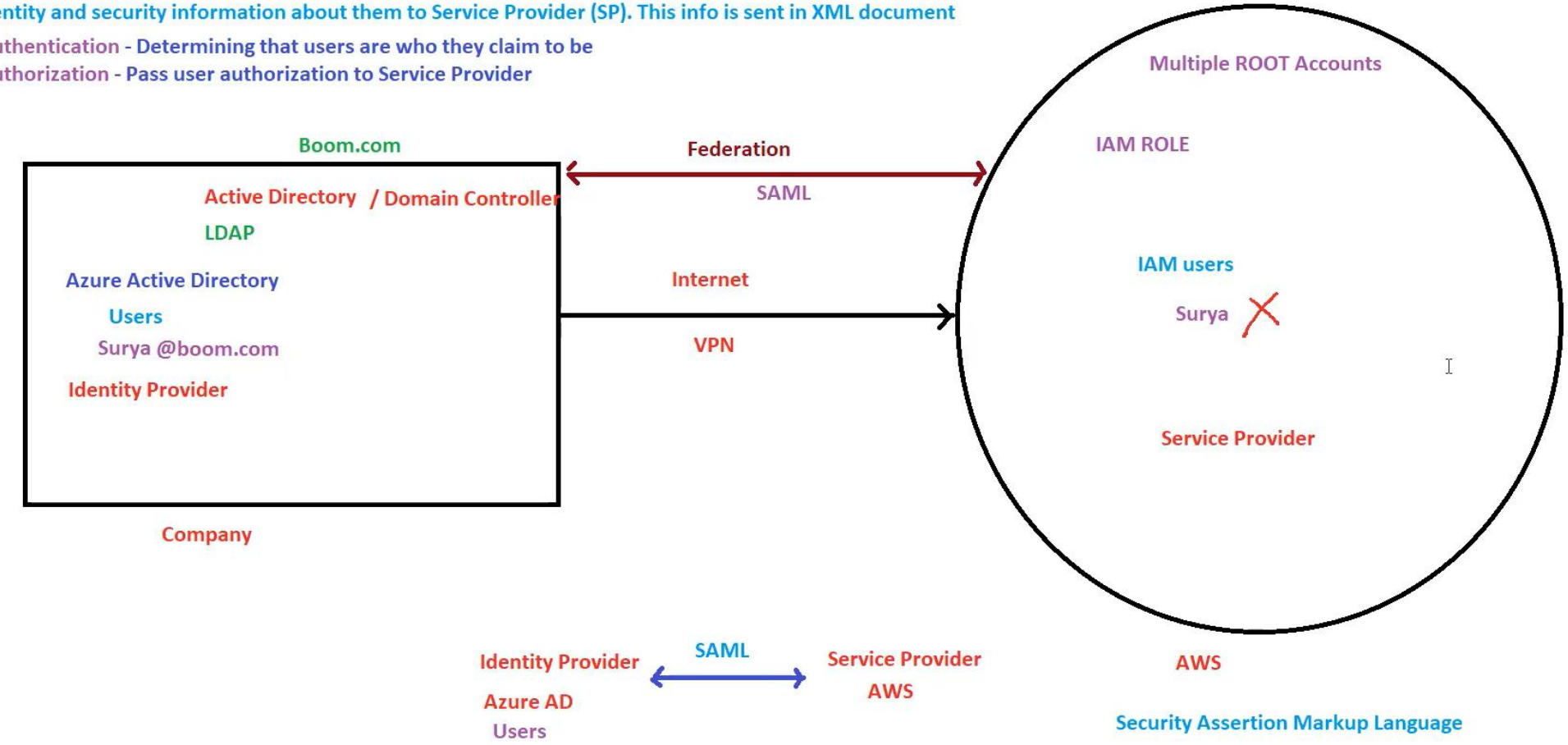
Organization Service is Global

1. YOu should have min 2 Root accounts
   1. management account 2. Memeber account
2. Enable Organizations and Identity Center
3. Go to Organization and Add AWS account (invite)
4. Go to IC, Create a Permission Set
5. Go to IC, Create User and assing Permission Set
6. Assing User to the Organization Units
7. Access AWS Access Portal URL and login with the user that you have created
   Access athe AWS account

SAML is an Open Federation standard that allows an identity Provider (IdP) to authenticate users and pass identity and security information about them to Service Provider (SP). This info is sent in XML document

Authentication - Determining that users are who they claim to be

Authorization - Pass user authorization to Service Provider



Identity Provider — SAML — Service Provider

Azure AD Users — AWS

Security Assertion Markup Language

EC2 is Regional

Servers = EC2 Instances

750 hours free per month

Price = Pay as you Go

Pay every hour

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

5 mins ---> Terminated
5 mins ---> Terminated
---> 2 hours

Launch
Running ---> Bill
Stop --> No Bill
Start
Reboot
Terminate ---> No Bill

Pricing Models

On-Demand Instances

Fixed Price (Hourly)

Pay for what you have used

Pay per Hour

No Commitment

No Upfornt Payment

No Predictable Usgae

Reserved Instances

Long Term commitment

1 or 3 years

Upfront Payment (Full, Partial)

72% discount on hourly price

Predictable Usage

Standard RI : 72% discount, Longer Workloads

Convertible RI : Long Workloads with flexible instances , 66% discount

(Change instance type, OS, Tenancy etc)

SPOT Instances

Bidding
Auctioning

Huge Capacity for cheaper price

90% Discount

Can loose EC2 instances

Most cost effective instances in AWS

Dedicated HOSTS

Book entire Physical Machine

Control instance placements

Dedicated Instances

No other customer will share your hardware.

No Control on Instance Placements

Capacity Reservation

Reserve capacity in a specific AZ for any duration .

No Time Commitment

Savings Plans

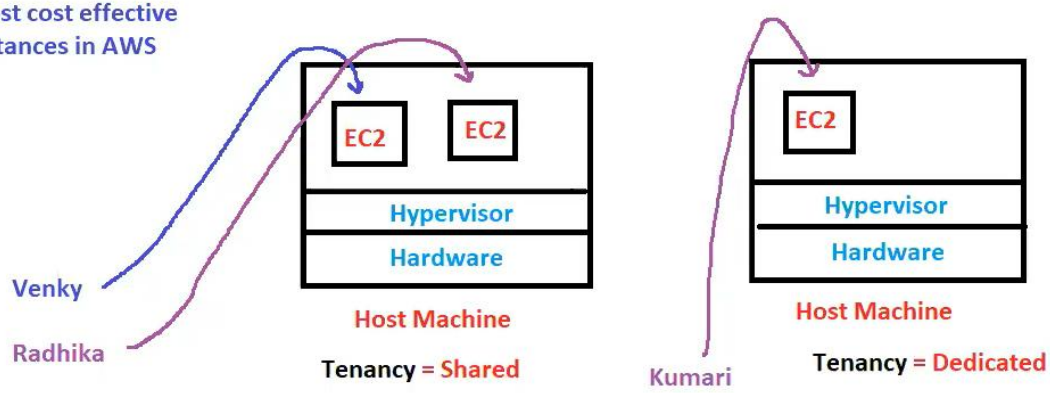
Same as RI but it has diff strategy

1 or 3 year , commitment to the amount of usage, Long Workloads

Get discount based on long term usage

Locked to specific instance family & Region

Felxible across instance Size, OS, Tenancy



EC2 Instance Families

General Instances

Memory Instances

CPU Instances

Storage Instances

GPU Instances

Instance Types

Instance Type = CPU + Memory

t2.nano = 0.5GB RAM , 1 vCPU
t2.micro = 1GB RAM, 1 vCPU
t2.small = 2GB RAM , 2 vCPU
t2.medium = 4GB RAM
t2.large = 8GB RAM
t2.xlarge = 16GB RAM

Scalability = Scale Up and Scale Down

Scalability can be acheived by changing the instance type

Anytime Scale Up ---> No Data Lost
Anytime Scale Down ---> No Data Lost

You need to STOP the EC2 instance to change the instance type
(Downtime Required)

If you want to change the instance type, data in the instance will NOT BE LOST

☑ Burstable Performance Instances

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

CPU Credits

2 vCPU + 6 vCPU

t2.small

CPU Credits depend upon
the instance type

Volumes

EBS Volumes

Persistent Storage / Permanent

If you STOP and START the EC2 instance, DATA is not LOST

EBS Volume max size = 16TB

EBS is Billable

Reboot = Data is NOT Lost

Types of EBS Volume

General Purpose (gp2, gp3) - SSD = General Purpose
Provisioned IOPS (io1, io2) - SSD = High Performance - Costlier
ThroughPut (st1) - HDD = Frequently access data with cheaper price
Cold (sc1) - HDD = Not Frequently access data with Cheaper Price
Magnetic (Standard) - HDD = Previous Generation

gp2 is a defautl EBS volume type, IOPS = Input Output Per Second

io1, io2 and gp3 are IOPS configurable = You can input IOPS values

More you input IOPS value the more performance and bill you get

gp2 has default IOPS = 1:3 = 1GB : 3IOPS ---> IOPS not configurable

Root volume support these volume types (gp2, gp3, io1 , io2 and standard)

Root volume doesnt support st1 and sc1

Addtional volumes support ALL types

io1/io2 can be multi-attached

Attach the same EBS volume to multiple EC2 instances at the same time in same AZ

Upto 16 EC2 instances at a time can be attached

Instance Store Volumes (ISV)

Not Persistent Storage / Temporary Volumes

If you STOP and START the EC2 instance, DATA is LOST / Gone / Deleted

instance store volumes are FREE

Emphemeral Storage

Reboot = Data is NOT Lost

If you terminate the EC2 instance, by default ROOT volume will also be deleted because "delete on termination" is enabled

If you want the ROOT volume to be retained, disable "delete on termination" while launching the EC2 instance

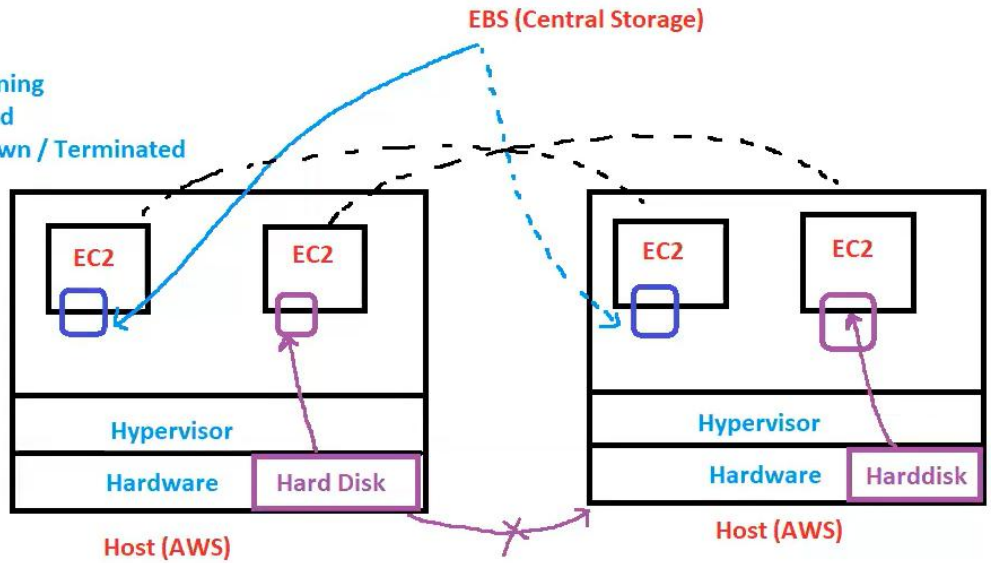
If you terminate the EC2 instance, by default Addtional volume will NOT BE deleted because "delete on termination" is Disabled

These are options Enabled/Disabled, anytime you can change based on requirement while launching the instance

Based on the instance type you select mostly big instance type, you will get the instance store volumes automatically attached by AWS for FREE

Instance 6 States

Launch ---> Pending / Running
Stop --> Stopping / Stopped
Terminate ---> Shuttingdown / Terminated



2 Types of Status Checks

Instance Status Check
System Status Check
2 Checks

1st hard check
2nd software check

2/2 checks are passed --> can login to EC2

1/2 checks are passed --> you cannot login to EC2

0/2 checks are passed --> you cannot login to EC2

AWS will do the Status Checks for Us

Solution : Stop and Start (jump) to new host machine and your EC2 will start working with 2/2 checks are passed

Solution 2 : Terminate the EC2 if you dont have any data, not recommended and launch new EC2 instance

EBS

Stop and Start = JUMP = Data is not Lost

Reboot = No Jump = Data is not LOST

Terminate = Data is lost

Instance Store Volume

Stop and Start = Jump = Data is Lost

Reboot = No Jump = Data is NOT LOST

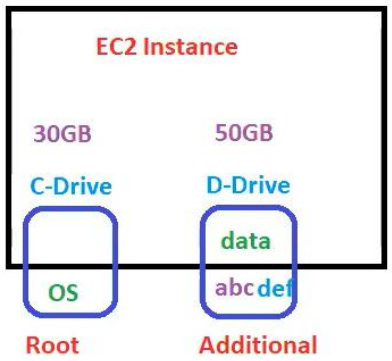
. EBS Volumes are network drives with good but limited performance
. If you need a high performance hard disk, use instance store volumes
. But Risk of data loss if hardware fails
. Backups is our responsibility not AWS

EC2 Hibernate

. In Memory (RAM) is Preserved
. The instance boot is much faster (the OS is not Stopped / Restarted)
. The RAM state is written to a file in the ROOT EBS Volume
. The Root EBS Volume must be Encrypted

Use Cases
. For Long Running Process
. Save RAM State
. Services that take time to initialize

. Instance RAM Size = Must be less than 150GB
. Root volume must be EBS and Encrypted , NOT ISV
. An instance cannot be hibernated for more than 60 days



Vol 50GB ---> Snapshot (50GB) ---> abc

--> Snapshot (50GB) ---> ~~abc~~def

Incremental Backup

SNAPSHOTS

1. Snapshot is a point in time copy of the Volume
2. Backup of the Volume is called Snapshot
3. EBS Snapshots are Created from EBS Volumes
4. You can create Snapshots from volumes
5. EBS Volumes ------> EBS Snapshots -----> EBS Volumes
6. You cannot attach a snapshot directly to the EC2 instance, you have to create a volume out of it first and then attach it to the EC2 instance
7. It is not possible to login or use the snapshot directly
8. Snapshots are Stored in S3 (Provider's S3)
9. Snapshots are visible from the EC2 Console
10. Snapshots doesnt have any AZ's
11. Snapshots are Regional
12. By default Snapshots are Private, if required we can make it public
13. You can copy the snapshot from one region to another region in the same account
14. Snapshots can be shared from one AWS account to another AWS account (AWS ID, Private)
15. EBS Volumes cannot be moved directly to any other regions, accounts ---> use Snapshots

EBS Snapshot Tiers - 2 Types

Standard Tier, Archive Tier

Default is Standard Tier

. Move the snapshots to an Archive Tier that is 75% Cheaper

. Takes 24 to 72 hours for restoring the archive ---> Billable

RecycleBin

. Setup rules to retain deleted snapshots so you can recove them after accidental deletions (Retention period 1 day to 1 yr)

. Selected Snapshots can also be sent to recyclebin using TAGS

Fast Snapshot Restore (FSR)

. Forcefull initilization of snapshot to have no latency for first use

EBS Volumes are created from EBS Snapshots

Instance Store Volumes are created from a template stored in S3

By default, Volumes are Not Encrypted

Encryption/Decryption is managed by AWS

Not Encrypted ---> Not Encrypted

Encrypted ---> Encrypted

Not Encrypted ---> Encrypted (Copy Option)

To create a snapshot we no need to STOP the EC2 instance

Data Life Cycle Manager: It is used to take the snapshots automatically / Schedule volumes that should get snapshots will be identified by TAGS

Life Cycle Policy/Rule should be created

Retention Period = 7 days

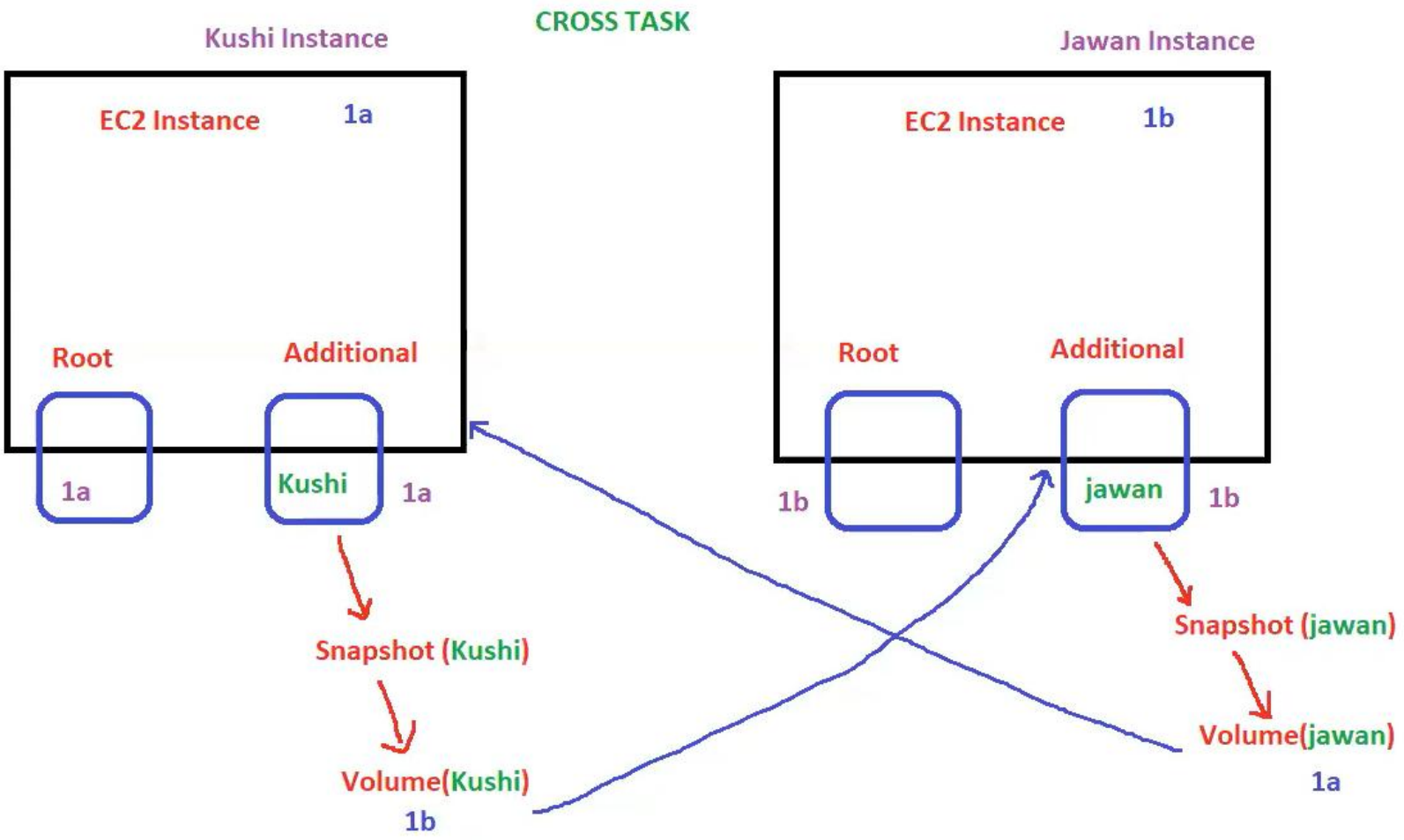
Example

All Encryption KEYS are Stored in KMS (Key Management Service)

Access and Secret Keys are for access purpose / Encryption Keys are for Security Purpose

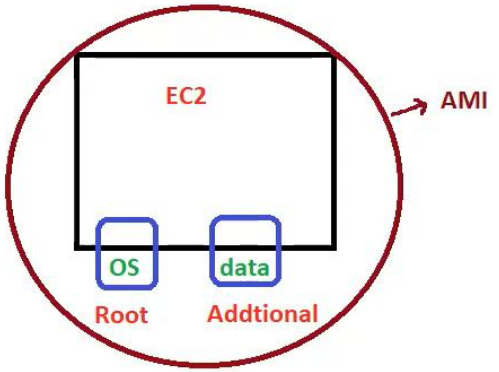
Encryption Snapshots cannot be Shared

How to move the volume from 1 AZ to another AZ



Snapshot ---> Copy of the Volume

AMI --> Copy of the entire EC2 instance (includes volumes also)



IMAGES

Copy of the OS is called Image

Image = AMI = Amazon Machine Image

Template of the OS is called AMI

AMI contains OS or OS + Few Apps

Copy of the image includes all configurations that we did on original instance

EC2 instance ----> Image(AMI) ---> EC2 instance

1 AMI, can be used multiple tiles to launch multiple EC2 instances

AMI's are re-usable, AMI doesnt 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 make it public

AMI's are Regional

AMI's can be copied from one region to another region in the same account

AMI's can be shared from one account to another AWS account using AWS ID

All public images are located at AWS Market Place

Image Contain OS ----> Root Volume (EBS)
---> Root Volume (ISV)

OS

Windows
Win Server
2021, 2022, 2023

Linux
RedHat, CentOS
Ubuntu / SUSE

Plain OS
OS + Few Apps

If you customize the app on OS, and then take the image ---> Custom AMI
Manually (Or) Golden AMI
Automatically

Creating images can be automated using
EC2 image builder ---> Golden AMI's

EC2 instance
2 Volumes
Root
Addtional
→ AMI(Image) (2 Volumes)
↓
2 Snapshots

When ever you are creating a image, Snapshots will also get created based on how many volumes you have in the EC2 instance

AWS
AMI
Linux
→ Linux OS (EC2 Tomcat) → Customer
AMI
Linux
Tomcat
Custom AMI
Created by Customer

AMI's are stored in S3

Images are backed by either EBS volumes or ISV

NO need to stop the EC2 instance to create image (but recommended is to STOP)

EC2

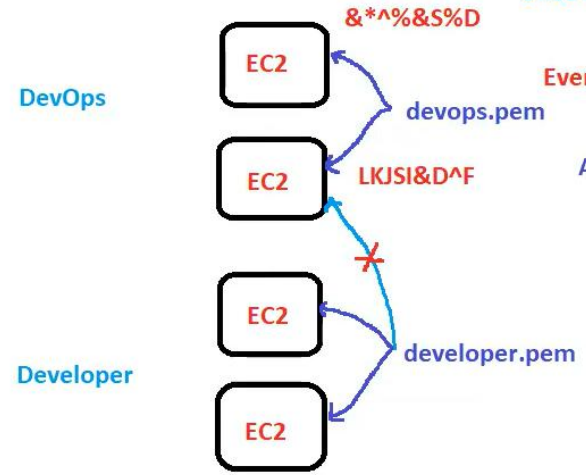
Windows / Linux

IP: will be provided by AWS
Username : Windows - Administrator
Password : You will get it through KEY-PAIR

IP: will be provided by AWS
Username : Linux - ec2-user
Password : You will get it through KEY-PAIR

KEY-PAIR

Key-Pair is used to retrive the password of the EC2 instance
We dont have any key-pair by default, we have to create it
We have to create a key-Pair ---> Key-Pair is also called as PEM file
Key-Pair extention is .PEM
When ever we launch EC2 instance, the console will ask you to create and attach a key-pair to the EC2 instance
You can create multiple key-Pairs
1 Key-Pair can be attached to multiple EC2 instances at the same time
EC2 instance can have only 1 Key-Pair attached at any point
For Every EC2 instance the password is unique / Different
Once the PEM file is attached, you cannot change the PEM file to the EC2 instance
Keep the PEM file in safe and secure it
Every time you retrive the password usign pem file, you will get the same password from the EC2 instance
Key-Pair = Combination of Public Key + Private Key
AWS has public key, Customer has private key(Pem file)



For Windows use Remote Desktop Protocol Client

For Linux

Putty ---> doesnt support pem file

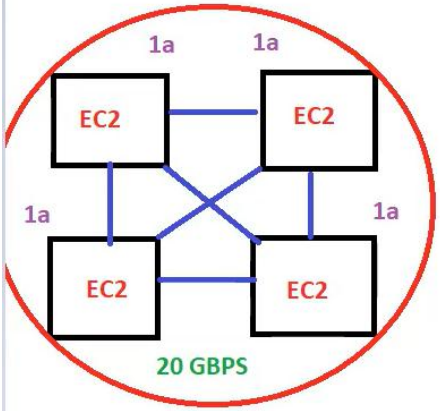
---> ppk file puttygen

Windows EC2 instance ---> administrator / Pem file

Protocol ---> RDP / 3389

Linux EC2 instance ---> ec2-user / pem file

Protocol ---> SSH / 22



Cluster Networking Instances

Cluster = Group of Servers / EC2 instances ---> This group is called Placement Group

When you launch a new EC2 instance, the EC2 service will place the instance such a way that all your instances are spread out across dif hardwares

Cluster Placement Group : Grouping the instances in same / single AZ / Same Rack, High Performance, Low HA

Spread Placement Group : EC2 instances are spread across AZ's. High HA, Critical Applications

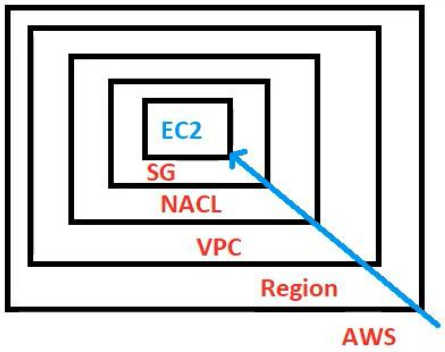
Per 1 AZ = 7 instances

Partition Placement Group : Across AZ's, Max Partitions = 7

1 Partition can contains 100's of EC2 instances

Placement Group recommended to have same / homogenous instance type

When you are launching the EC2 instance, you can select which placement group you want to keep the instance



SG

Allow http / 80

inbound rules

Allow SSH / 22

Rest are all DENY

RDP ---> is used to login / connect remotely to the Windows EC2 instance

SSH ---> is used to login / connect remotely to the Linux EC2 instance

http / https ---> is used to access the applications

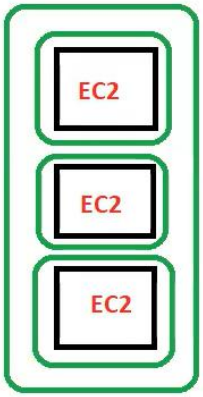
Security Groups

SG acts like a firewall to secure the EC2 instance

ssh

Security Group

Inbound	Outbound
Deny	Allowed
Deny	Allowed
Deny	Allowed



1 EC2 --> many SG's attached

1 SG --> attached to many EC2's

Source

Custom

Anywhere

MyIP

If you allow any inbound rules, you MUST allow on outbound rule ---> STATELESS

NACL's are STATELESS

Firewall = Which stops unauthorized access to the network

Firewall = Security Group Allow / Deny

Security Group = SG stops unauthorized access to the EC2 instance

Security Group is used to secure the EC2 instance

Security Group has 2 Rules

Inbound Rules --> Which allows the traffic towards EC2 instance

Outbound Rules --> Which send the traffic outside EC2 instance

By default, Inbound rules are DENY / Outbound rules are ALLOWED

Is it possible to DENY protocol in SG ? NO (Because by default inbound rules are already DENY)

In SG, you can only ALLOW, you cannot DENY

Every EC2 instance must have atleast 1 SG attached

You can create multiple SG's and you can attach multiple SG's to a single EC2 instance

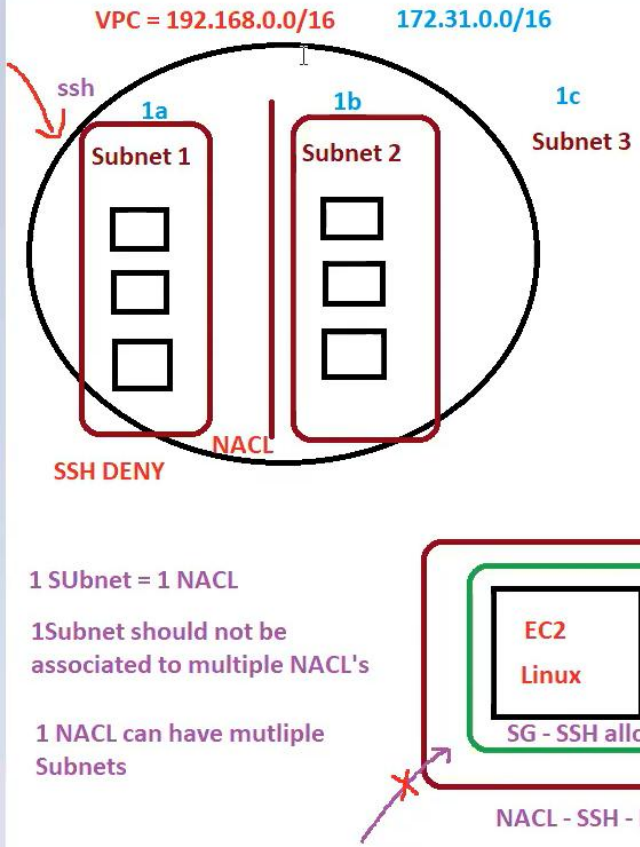
AWS EC2 has a default Security Group

A brand new SG , all inbound rules DENY and outbound rules are ALLOWED

If you allow any inbound rules, you no need to allow on outbound rules

STATEFUL

Security Groups are STATEFULL



1 SUbnet = 1 NACL

1Subnet should not be associated to multiple NACL's

1 NACL can have mutliple Subnets

NACL = Network Access Control List

Another layer of Security to the EC2 instance

If you want to tight security then go with NACL

Like SG, NACL also has inbound rule and Outbound rules

NACL will hit first and then SG

SG is EC2 instance level, NACL is Subnet Level

NACL's are Optional

1 Subnet is associated to 1 AZ

1 Subnet cannot be associated to multiple AZ's at the same time

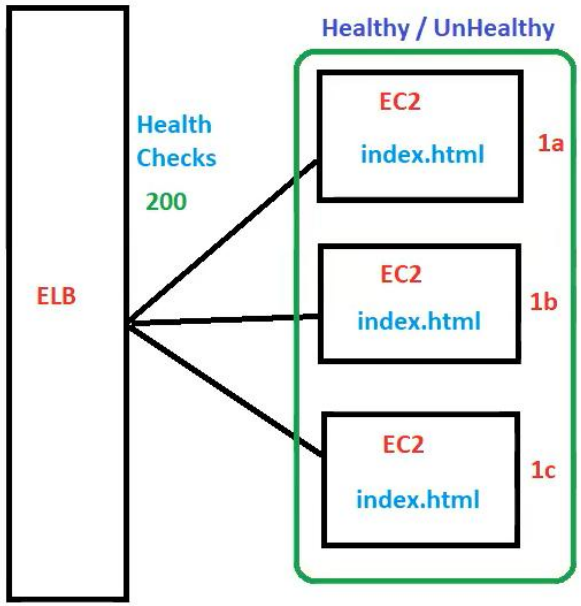
1 AZ can have multiple Subnets

Security Group	NACL
Inbound rules and Outbound rules	Inbound rules and Outbound rules
Default SG	Default NACL
SG will hit after NACL	NACL will hit first and then SG
By default , inbound rules are DENY	By default , inbound and outbound rules are ALLOWED
You cannot DENY on SG	On NACL, we can DENY and Allow Also
If you create any new SG, inbound rules are DENY and Outbound rules are ALLOWED	If you create any new NACL, inbound rules are DENY and Outbound rules are DENY
SG;s are Instance level	NACL's are Subnet Level
SG's are STATEFULL	NACL's are STATELESS
If you allow any inbound rules, you no need to allow on outbound rules	If you allow any inbound rules, you MUST allow also on outbound rule

Route53

http://boom.com

http://myelb.84758475.ap-south-1.elb.amazonaws.com



DNS Name

Nasty URL

Elastic Load Balancer

ELB which distribute the traffic to multiple EC2 instances across AZ's

ELB is completely managed by AWS

ELB is a Service by AWS, not Server for us

ELB can be accessed by URL (DNS Name), you cannot login to ELB

ELB has the IP address, but these IP's are Dynamic

AWS always recommend to use the ELB DNS Name not IP address

Types of Elastic Load Balancers

Application Load Balancer

ALB

Latest Generation

default choose is ALB

HTTP and HTTPS

Best for micro-services

Routing Features

Host based routing
Path based routing
String parameter routing
etc

like?course=aws

Network Load Balancer

NLB

Latest Generation

TCP and UDP

Extreme High Performance

Network Level

1 Static IP per AZ

Classic Load Balancer

CLB

HTTP, HTTPS and TCP

Previous Generation

Gateway Load Balancer

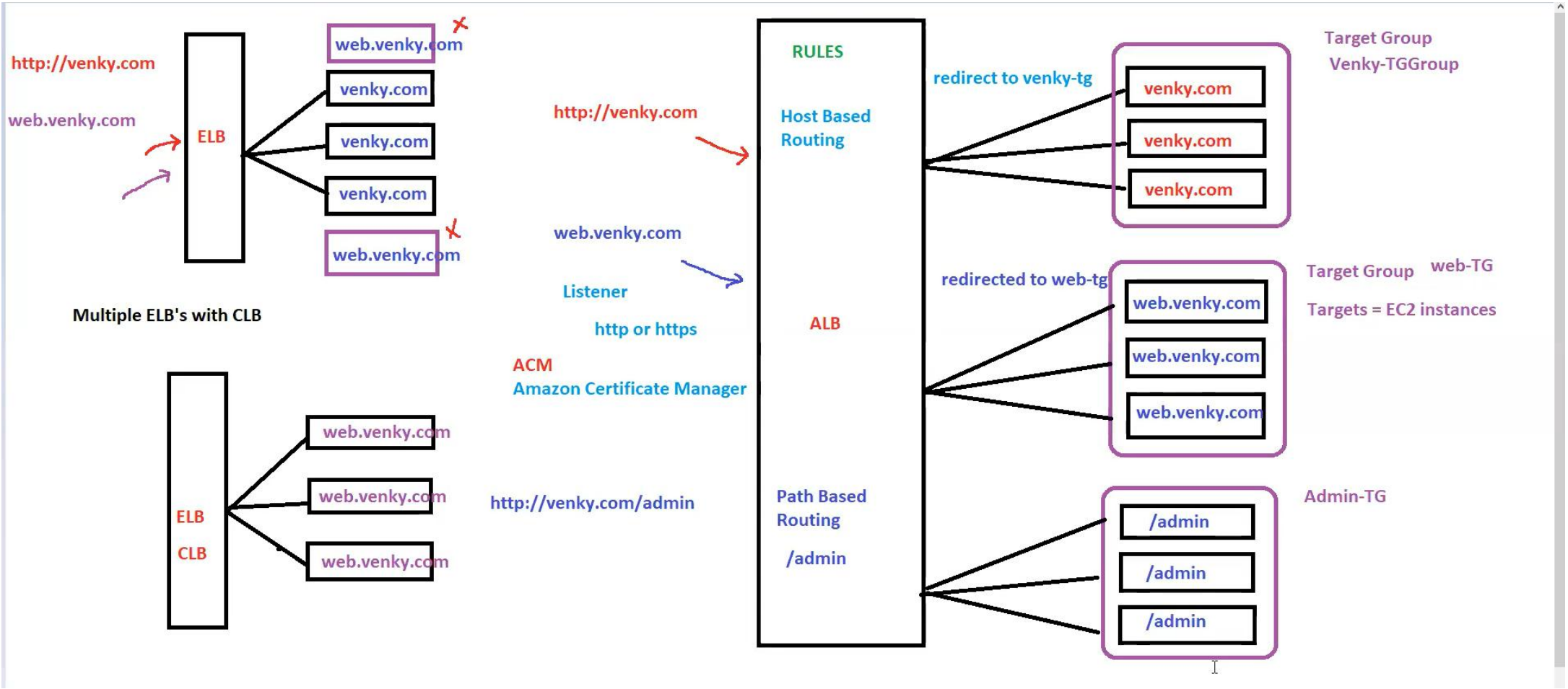
GLB

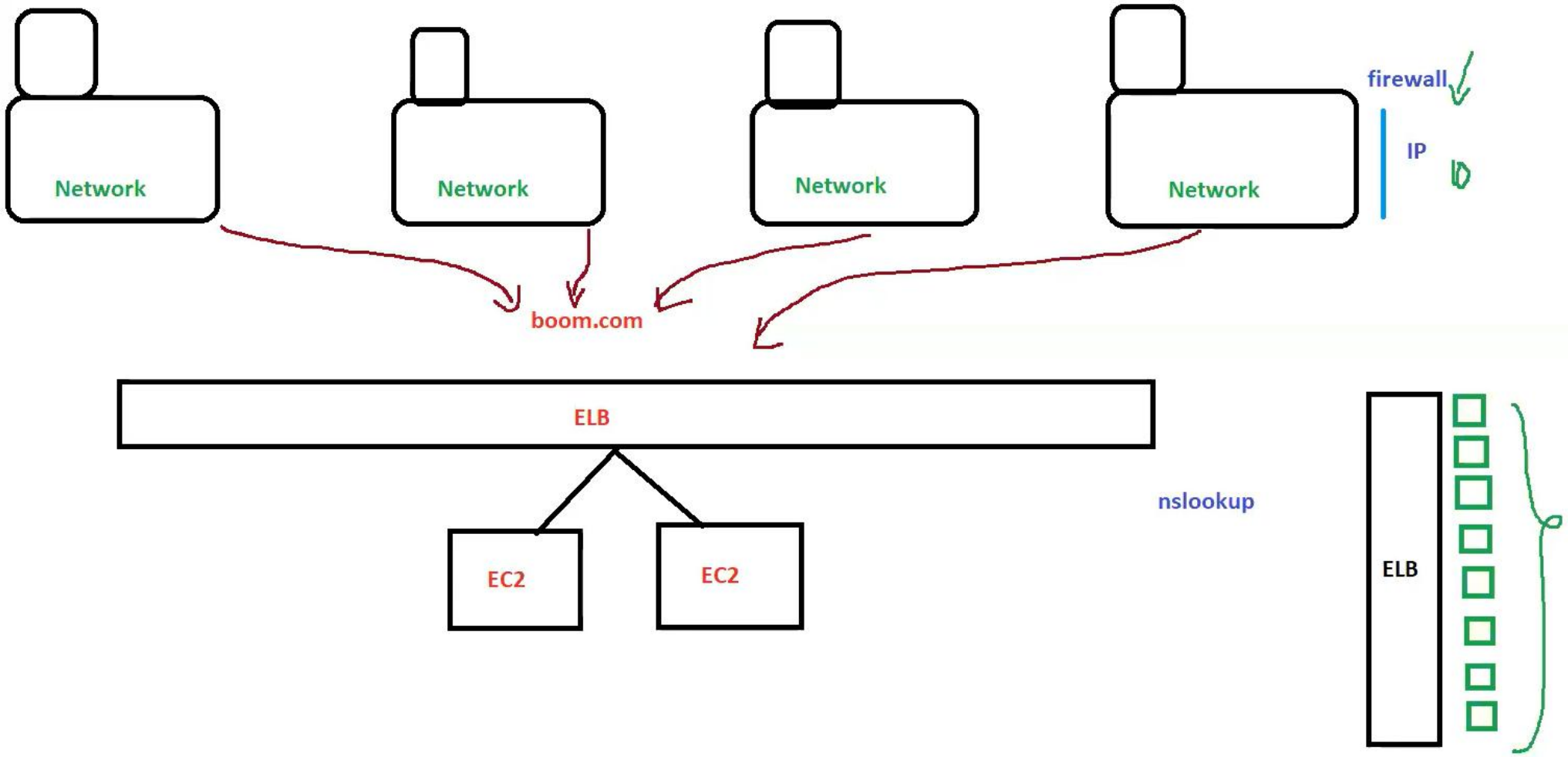
Latest Generation

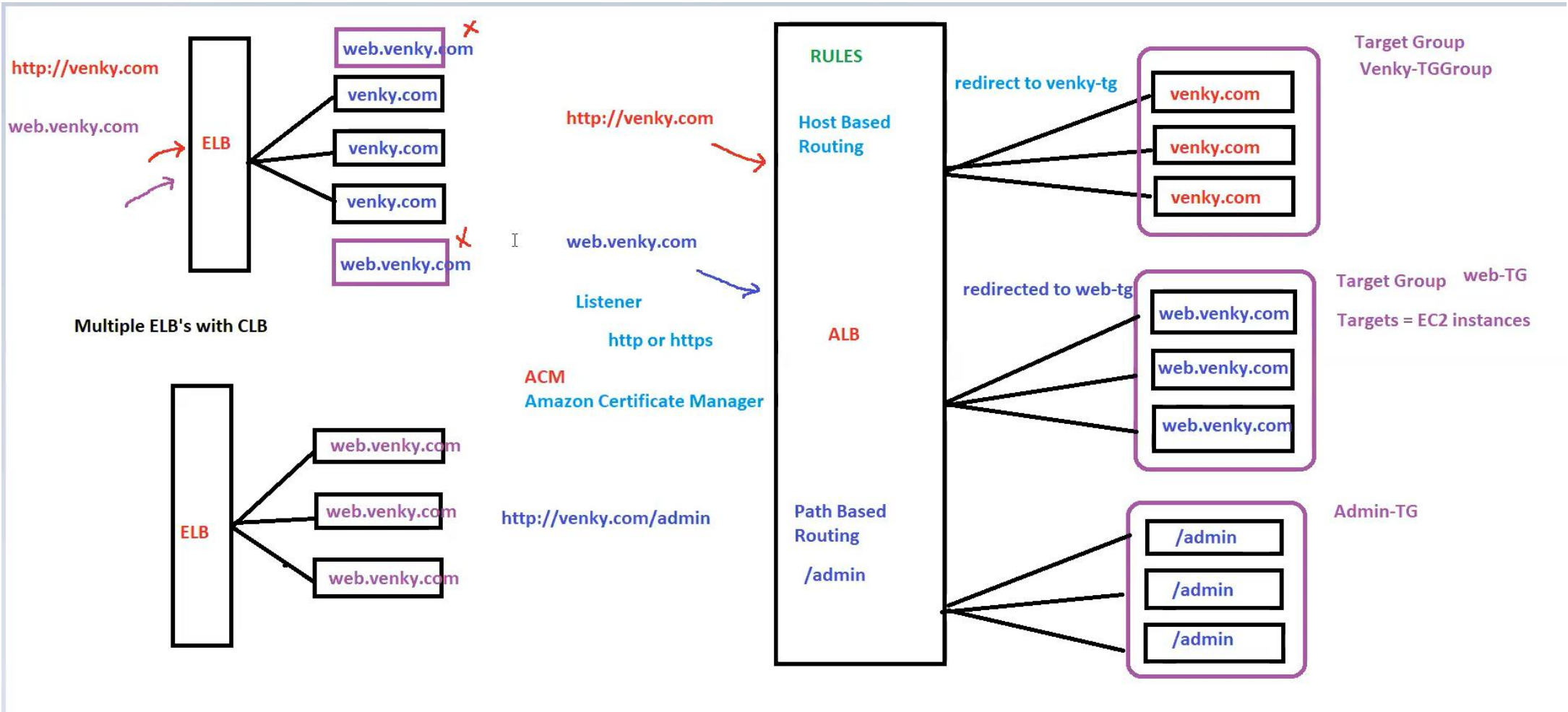
Deploy, manage, and scale a fleet of 3rd party network virtual appliance in AWS

. Firewall, Prevention Systems etc

GENEVE protocol on 6081


http://venky.com
web.venky.com
ELB
web.venky.com
venky.com
venky.com
venky.com
web.venky.com
Multiple ELB's with CLB
ELB
CLB
web.venky.com
web.venky.com
web.venky.com
http://venky.com
web.venky.com
Listener
http or https
ACM
Amazon Certificate Manager
http://venky.com/admin
RULES
Host Based
Routing
ALB
Path Based
Routing
/admin
redirect to venky-tg
redirected to web-tg
venky.com
venky.com
venky.com
web.venky.com
web.venky.com
web.venky.com
/admin
/admin
/admin
Target Group
Venky-TGGroup
Target Group
web-TG
Targets = EC2 instances
Admin-TG


Network
Network
Network
Network
firewall
IP
boom.com
ELB
EC2
EC2
nslookup
ELB


http://venky.com
web.venky.com
ELB
web.venky.com
venky.com
venky.com
venky.com
web.venky.com
Multiple ELB's with CLB
ELB
web.venky.com
web.venky.com
web.venky.com
http://venky.com/admin
http://venky.com
web.venky.com
Listener
http or https
ACM
Amazon Certificate Manager
RULES
Host Based
Routing
ALB
Path Based
Routing
/admin
redirect to venky-tg
redirected to web-tg
/admin
Target Group
Venky-TGGroup
venky.com
venky.com
venky.com
Target Group web-TG
Targets = EC2 instances
web.venky.com
web.venky.com
web.venky.com
Admin-TG
/admin
/admin
/admin

Types of IP's

Public IP

Which can be accessed from internet

Public IP is optional

Public IP is Dynamic

If you STOP and START the EC2 instance, Public IP will be Changed

AWS assign Public IP to the EC2 instance

Private IP

Which cannot be accessed from the internet

Private IP is mandatory

Private IP are mostly Static

With in the VPC and through VPN

Elastic IP (EIP)

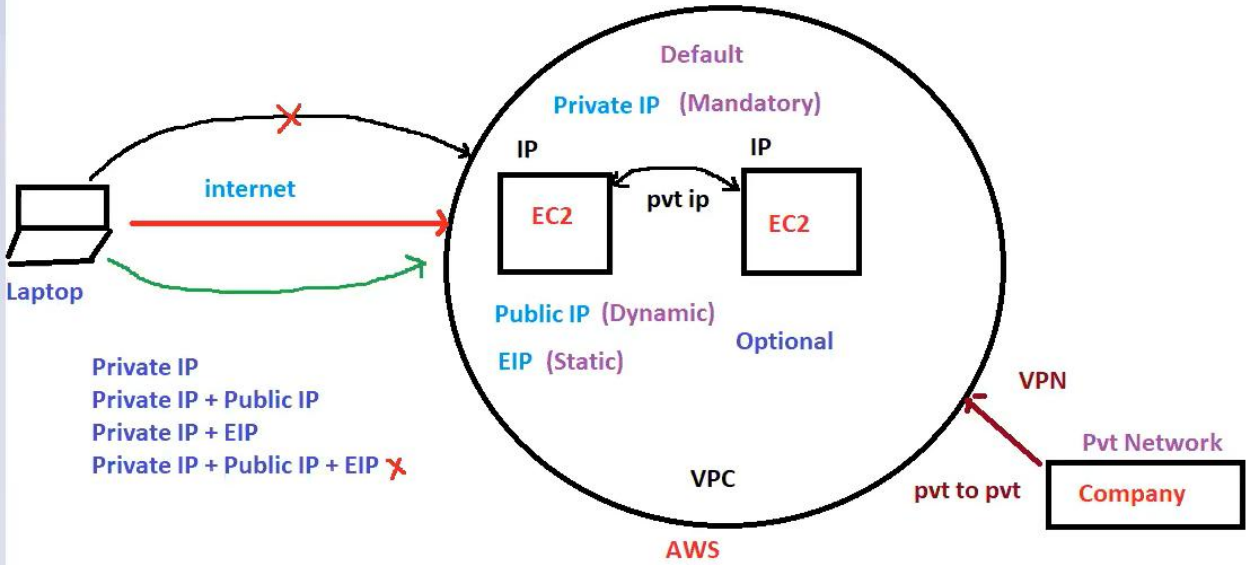
Same as Public IP, EIP is Static

If you STOP and START the EC2 instance EIP will not be changed

5 EIP's are FREE

EIP has to be associated to the EC2 instance

If you have not associated / Not used EIP to any instance, it will be charges if those EIP's are below 5 also, Dont keep it idle, Idle EIP's are Charged



Private IP
Private IP + Public IP
Private IP + EIP
Private IP + Public IP + EIP ✗

Instance Meta-Data = Data about instance

From Console, you get the instance meta-data information

From CLI, follow this below URL

http://169.254.169.254/latest/meta-data/ (use this inside EC2 instance only on CLI)

User-Data : BootStrap Script

The Script which you provide will run at the boot time of the EC2 instance

User-Data will run only for the very first time of launching EC2 instance

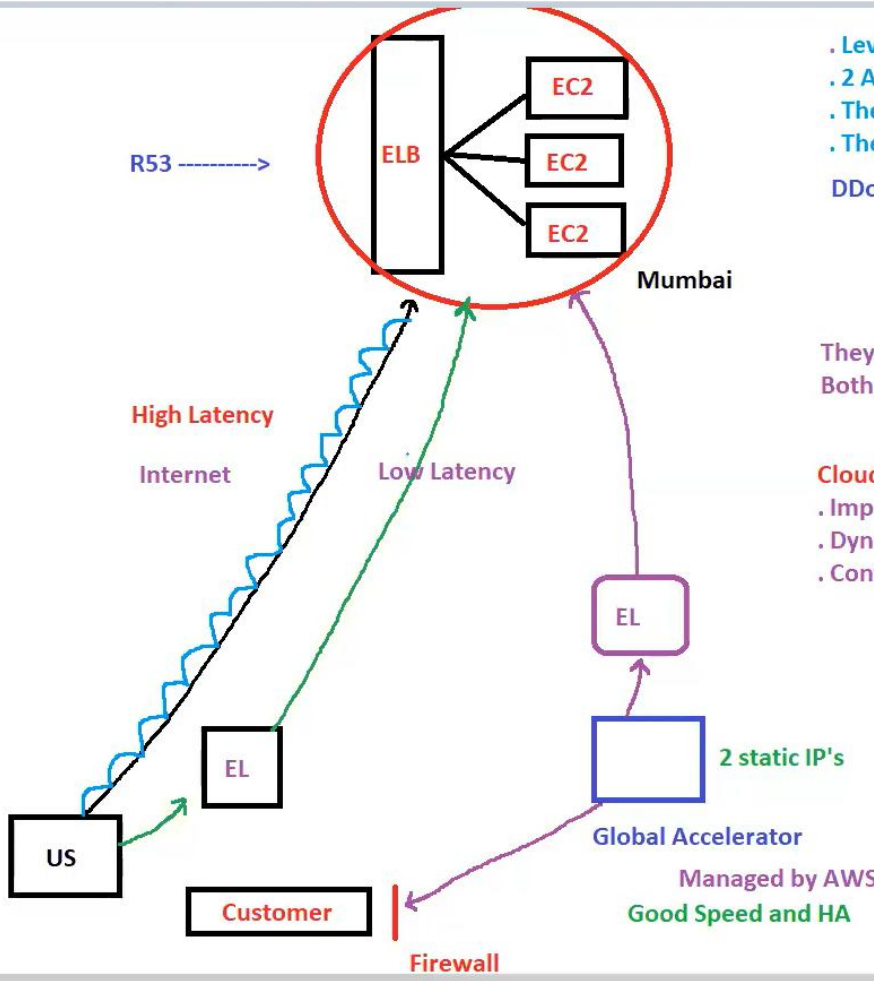
Linux = Shell Script
Windows = PowerShell

The 7 Steps

Select AMI	Select Instance Type	Instance Configuration	Select Storage	Select Security Group	Add Tags	Attach a PEM File	Launch
Linux Windows	t2.micro	How many instances ? Public IP user-data all configurations	EBS Volumes Root (30GB, 10GB) Addtional Volumes gp2	Default SG	Name = LinuxInstance	Create it Or use it	

Unicast IP : One Server holds on IP address

AnyCast IP : All Servers holds same IP address and the client is routed to the nearest one



. Leverage the AWS internal network to route to your application
. 2 Anycast IP are created for your application
. The Anycase IP send traffic directly to the Edge Locations
. The Edge Locations send traffic to application

DDoS protection through WAF and AWS Shield

Billable

GA vs CloudFront

They both use the AWS global network and its Edge Locations around the world
Both Services are integrated with AWS Shield for DDoS protection

CloudFront

. Improves performance for both Cache content (images and videos)
. Dynamic Content and Static also
. Content is Served from Edge Locations

Global Accelerator

. Improve performance for a wide range of applications
. It supports 2 Anycase Static IP
. Whitelisting IP and Disaster Recovery

User ---> boom.com ---> R53 ---> ELB
User ---> boom.com ---> R53 ---> CF ---> ELB
user ---> boom.com ---> R53 ----> GA ----> EL ----> ELB

CloudWatch is all about Alarms, Events and Logs

CloudWatch

Cloudwatch is used to monitor performance of AWS Resources

CloudWatch can monitor all AWS Services

CloudWatch can monitor only Host level metrics(Default Metrics)

CPU, Network, Disk and Status Checks are AWS Provided Default Metrics

Custom Metrics (Memory)

Basic Monitoring = Every 5 mins interval data points, FREE

Detailed Monitoring = Every 1 min interval data points, Billable

EC2

Application

OS

Host Level Metrics

CPU

Network

Disk

Status Check

B: 6:00 6:05 6:10 6:15

D: 6:00 6:01 6:02 6:03

Alarms

EC2 ---> CPU > 90% ---> Notification (SNS)

Alarms can also do some actions

(Terminate, STOP, Reboot, Recover)

Alarm has 3 States In Alarm, OK , Insufficient

Composite Alarms

. CW alarms are on single metrics

. Composite alarms are monitoring the states of multiple alarms

. AND or OR Conditions can be used

EVENTS / Event Bridge

Pending/Running, Stopping/Stopped, Shuttingdown/Terminated

Instance state change notification

EC2

App Server

Instance id

Stopped → EVENT RULE → Route to the target → SNS (Topic) Target

Pending/Running → RULE → Route to the target → Lambda (TerminateFN) Target

Schedule / Cron Job

EventBridge

9PM ----> STOPPED → Stop LambdaFn

9AM ----> START → Start LambdaFn

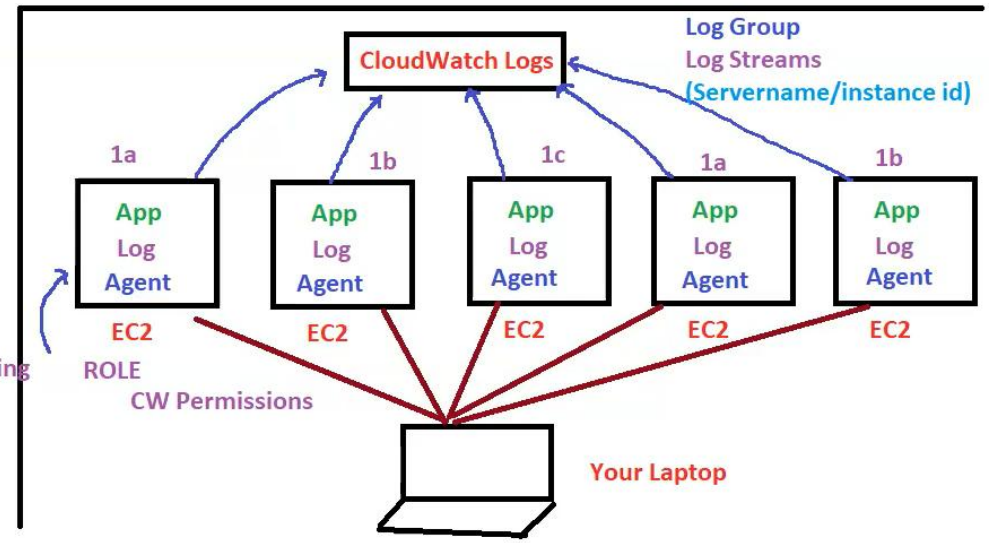
Cannary / Cannaries

Application Monitoring

Website Endpoint

NameSpace = Group of Metrics/ Collection of related metrics

CloudWatch can monitor Lambda Fns, Microservices, containers, K8s etc



CloudWatch Logs

Launch EC2 instance (Amazon Linux2)
Create a ROLE (permissions - CW)
and attach to the EC2 instance
Login to the EC2 instance
Install cloudwatch agent
configure the agent
start the agent

FYI: "File" parameter is related to the log path. Change the log path as per the requirement

```
sudo -s
yum install -y awslogs
cd /etc/awslogs
ls
cat awscli.conf
vi awscli.conf
 --> Press i --> change the region
 ---> Press Esc , :wq! --> Enter
cat awscli.conf
cat awslogs.conf
systemctl start awslogsd
```

see the logs in cloudwatchlogs

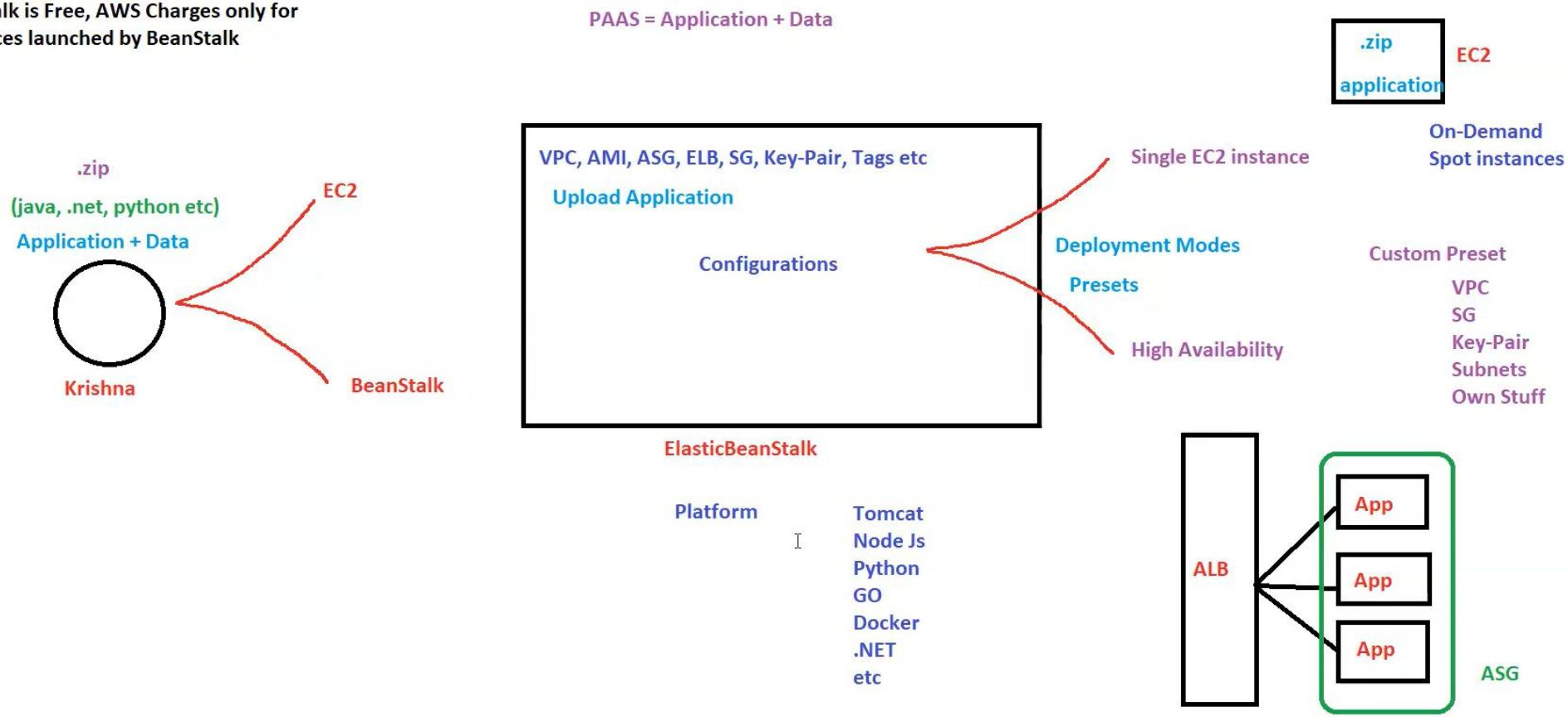
Backbone of BeanStalk is EC2

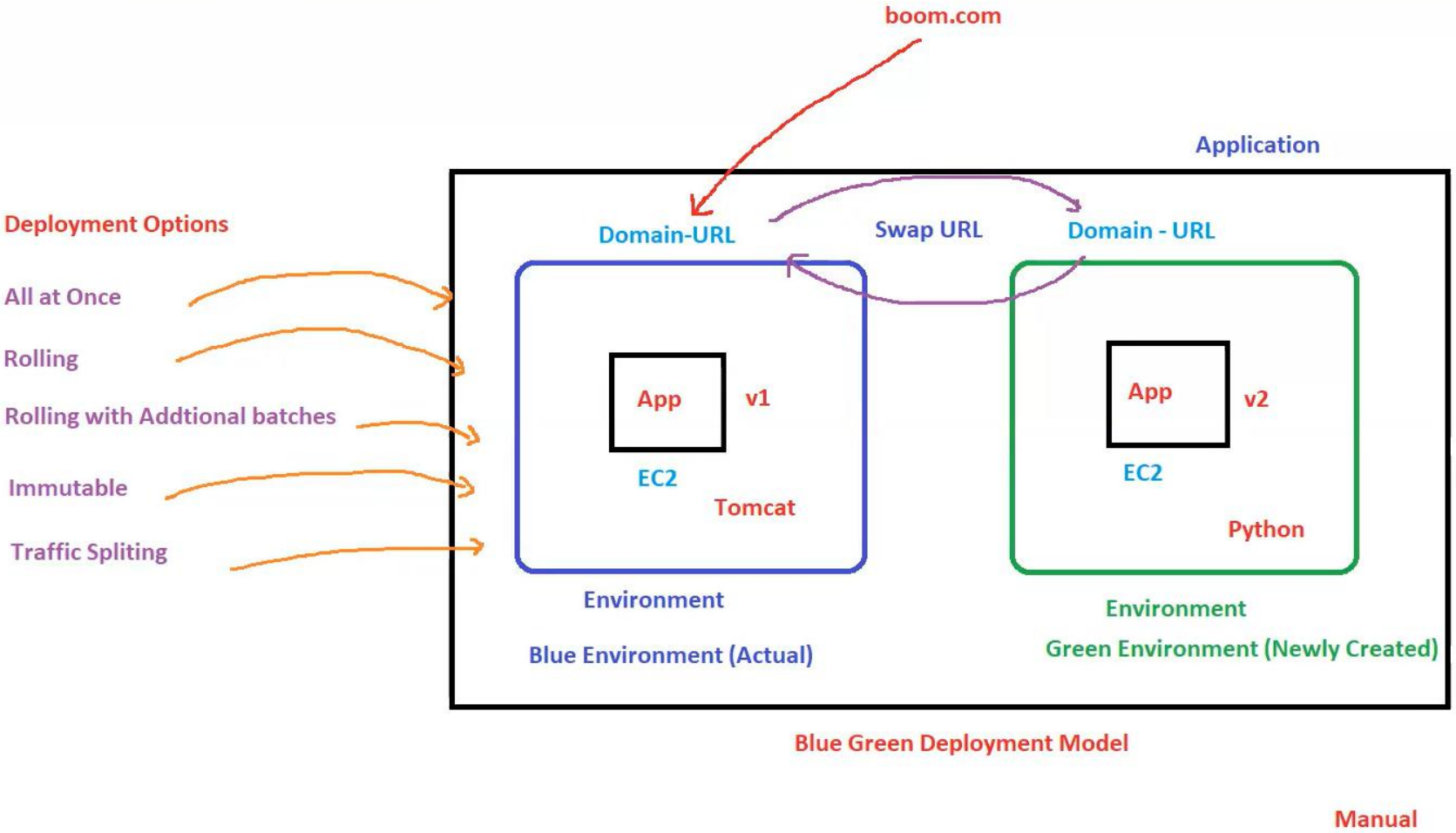
BeanStalk is Free, AWS Charges only for Resources launched by BeanStalk

ElasticBeanStalk

ElasticBeanStalk is used for easy and quick deployment of applications in AWS

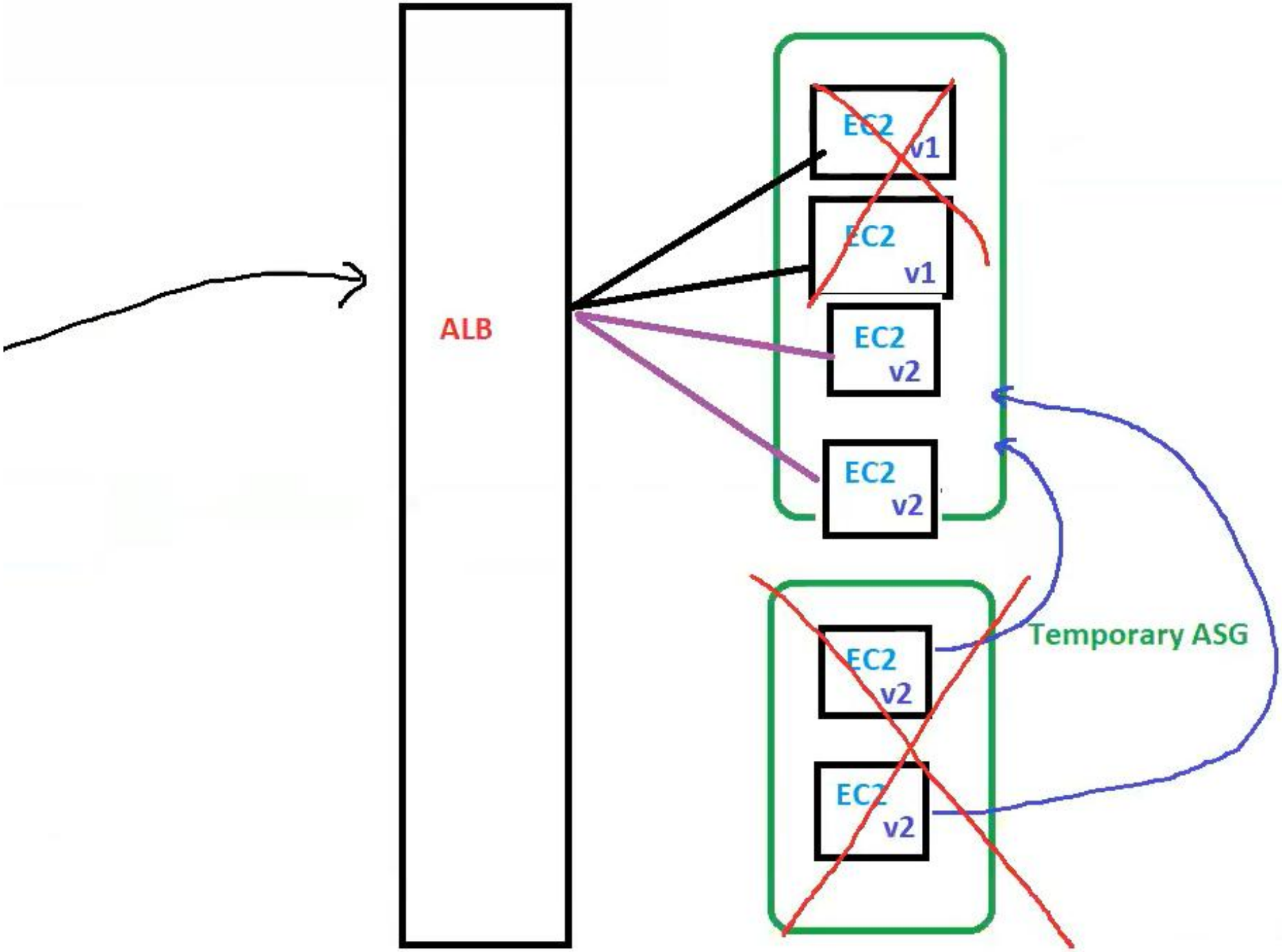
PAAS = Application + Data





Blue Green Deployment Model

Manual


ALB
EC2
v1
EC2
v1
EC2
v2
EC2
v2
EC2
v2
EC2
v2
Temporary ASG

S3 = Simple Storage Service

S3 is Global

S3 is Object Based Storage

In S3, we can store all FLAT Files

You can only upload, download and access files from S3

The Files in S3 cannot be executed

You cannot install OS, DB etc in S3

S3 is unlimited storage

S3 support Static Website Hosting

S3 is cheaper than EC2

S3 is Serverless

Bucket = Container of Objects

Object = File

KEY = Name of Object

Bkt1 ---> Mumbai
Bkt2 ---> Ireland

6pmbucket

Photos

puppy.jpg

Object URL / EndPoint

Mumbai Region / Private

Bucket

Bucket is Regional

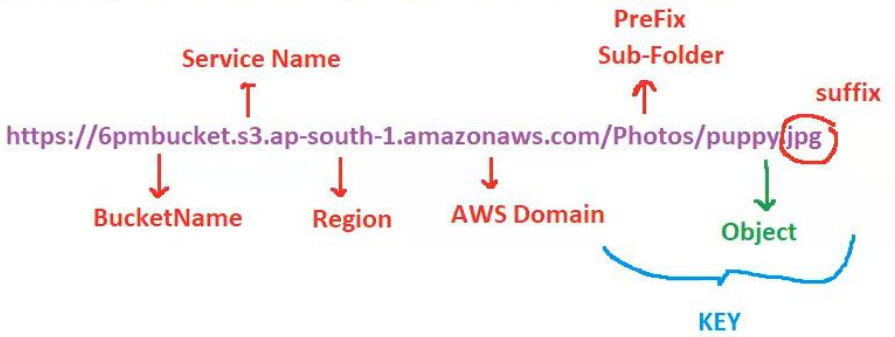
Bucket Names are universal / Unique

No nested buckets / Bucket under bucket cannot be created

You can create the buckets in different Regions

Max Number of Buckets you can created in your AWS account is 100 (Soft limit)

By default , buckets are Private, if required we can make it public

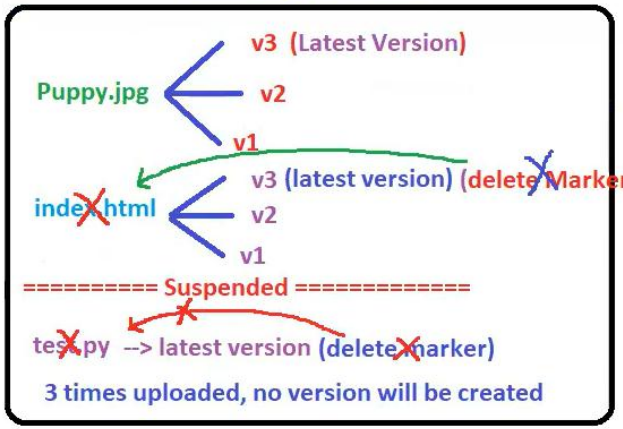


S3 is a WORM Model

Write Once and Read Many

S3 is mostly used for Read Purpose

Versioning is Enabled



6pmversionbucket

If you upload a object after versioning is suspended, the latest version will be created asusual
But, if you update the original object, versioning files are NOT CREATED

If you delete the original object, Delete marker is applied
If you delete the delete marker, Object will not be restored

In Suspended state, exisitng versioning files will not have any impact

S3 Versioning

Versioning is like a Backup tool

By Default, Versioning is not enabled on the bucket

Versioning is enabled on bucket level, but applied on Object Level

Version ID is always unique

Versioning files can be downloaded anytime

If you delete the original object, delete marker is applied on the latest version

If you want the object to be restored, delete the delete marker and your object is restored

If you want to restore the old version, download and upload it back to the bucket

Delete marker is applied only on the latest version, not for old/previous version

You cannot download the delete marker, you can only delete it

Based on the requirement we need to enable versioning, becareful on larger objects

Once you have enable versioning , you cannot disable it, you can SUSPEND it

S3 is unlimited storage

Min object size = 0 Bytes, Max Object Size = 5TB

You can have unlimited number of objects having 5TB each in single bucket

For Single PUT, you can upload max 5GB

Multi-Part Upload (MPU): Break the files into multiple chunks and upload it chunk by chunk : it can be done through CLI

AWS recommend, if you have object > 100MB use MPU

While uploading the object into S3, Selecting Storage class is mandatory

Storage Classes

Availability = Anytime

Durability = Longtime

Standard Frequently Access (FA)

This is used for frequently access data

Default Storage Class

Regular Purpose

No Retrival Charges

Availability = 99.99%
Durability = 11 9's

Min Object Size = 0 Bytes

Standard Infrequently Access (IA)

This is used for infrequently access data

Cheaper than FA

Retrival Charges apply

Once in a month

Availability = 99.9%
Durability = 11 9's

Min Object Size = 128KB

Min Duration = 30 days

Reduced Redundancy Storage (RRS)

Frequently access but NOT CRITICAL

No Retirval charges

AWS doesnt recommend to use this Storage Class

Cheaper than others

Availability = 99.99%
Durability = 99.99%

One-Zone IA

Infrequently access but NOT CRITICAL

Retrival charges apply

Availability = 99.5%
Durability = 11 9's

Min Object Size = 128KB

Min Duration = 30 days

Intelligent Tier

Unknown Access Patterns

Availability = 99.9%
Durability = 11 9's

Min Duration = 30 days

Glacier

Infrequently access data

Archiving Purpose

Vault = container of Archives

Archive = zip/object

1 Archive can be upto 40TB

Unlimited number or archives

1000 Vaults

Retrival Charges apply

Glacier Retrival Options

Expedited = 1 to 5 mins
Standard = 3 to 5 hours
Bulk = 5 to 12 hours

Availability = 99.99%
Durability = 11 9's

Min Duration = 90 days

Deep Glacier

Min Duration = 180 days

Life Cycle Management

Life Cycle Rules

It is possible to move the objects from one storage class to another storage class automatically

LCM is created on bucket level,
but applied on Object Level

Transistion and Expiration

You can setup S3 features for entire bucket
or for a prefix (sub-folder)

LCM Rule Current Versions Previous Versions

Transistion

FA ---> IA (30 days) ---> Glacier (60 days)

--> Delete after 365 days Expiration

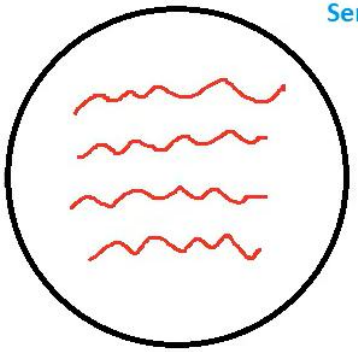
Life Cycle Management

Object Lock

Permanently

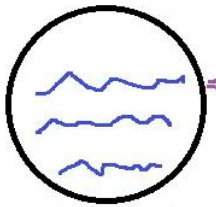
Certain period of time

Object Level Logs = CloudTrail



Bucket Name = Movies

Server Access Logs



moviesaccesslogs

Who is accessing your buckets

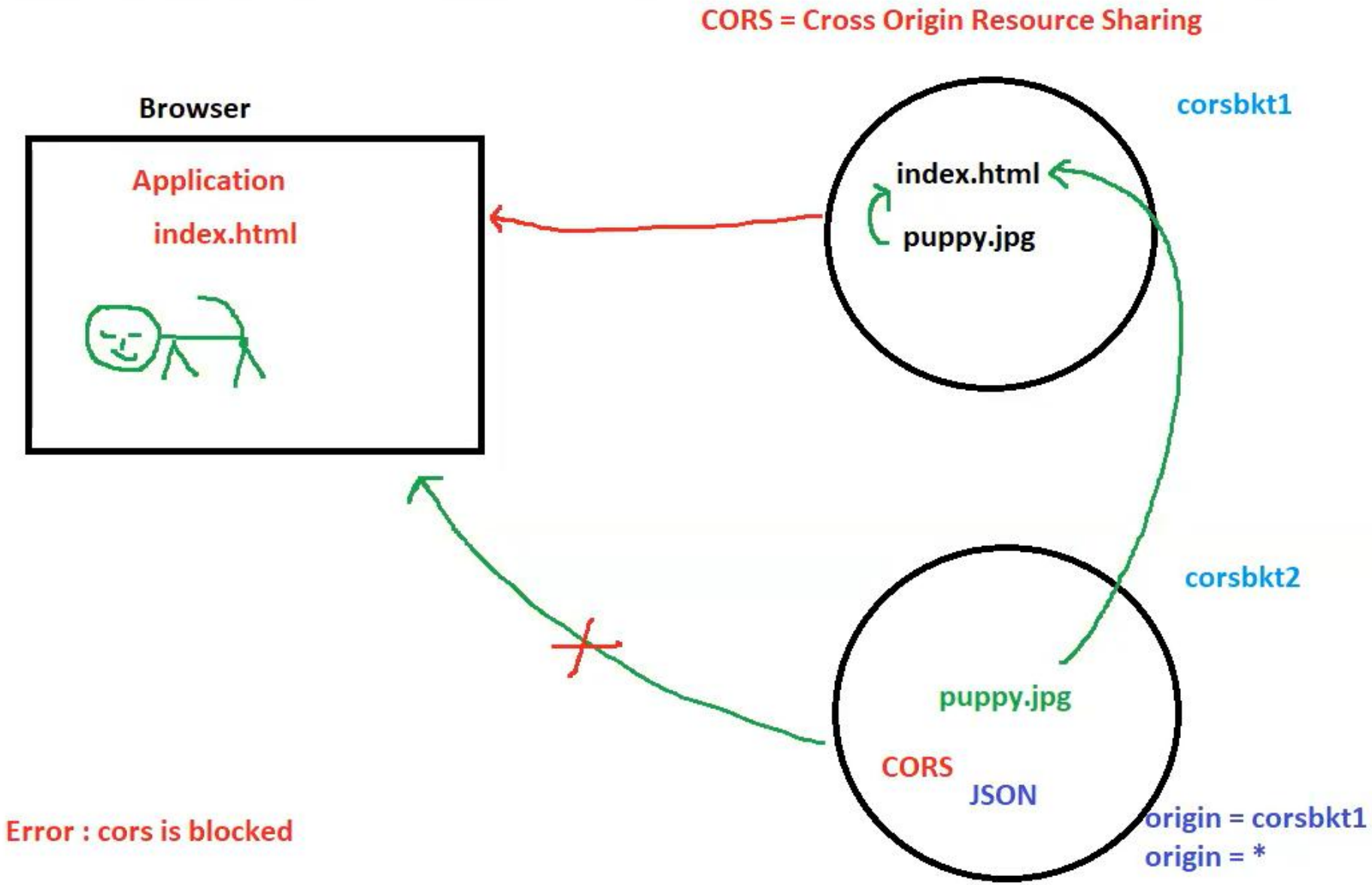
Server access logs are bucket level

192.168.10.106ambucketpuppy.jpgGETSuccess200

Athena : Analyze the logs directly from S3

IP	Src	Dest	Object	method	Time

SQL Query


CORS = Cross Origin Resource Sharing
Browser
Application
index.html
corsbkt1
index.html
puppy.jpg
corsbkt2
puppy.jpg
CORS
JSON
origin = corsbkt1
origin = *
Error : cors is blocked

CRR = Cross Region Replication

SRR = Same Region Replication

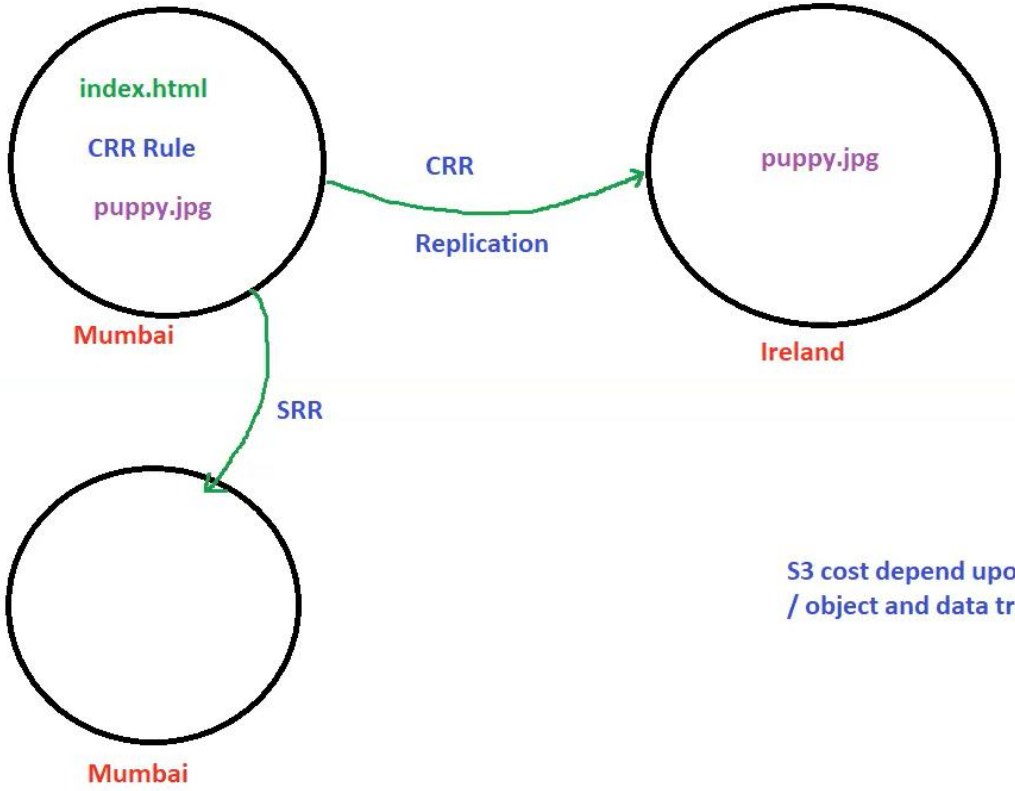
CRR/SRR is not enabled by default

CRR/SRR is enabled on bucket level

CRR/SRR can be for entire bucket or for a Prefix

Versoning is mandatory to have CRR/SRR

Existing objects can also be replicated at the time of creatign CRR Rule with Batch Operation Process



CRR buckets can be in same AWS account or different AWS account as well

S3 cost depend upon the size of the bucket / object and data transfer

Encryption can be done in 2 ways

In-Transit Encryption : Encryption while data is moving / Transfering HTTPS

Data at Rest Encryption : Encryption while data is at Rest , KMS

Amazon Certificate Manager (ACM)

is where you can generate HTTPS certificates (In-Transit)

Key Management Service (KMS)

is where you can create encryption keys for data at rest

Encryption

Amazon S3 has 3 types of Encryption

Server Side Encryption

SSE - S3 = (AWS Managed Key)

SSE - KMS = (AWS KMS Key)

SSE- C = (Customer Provided Key)

Client Side Encryption : Should be handled by Customer

In-Transit Encryption : HTTPS

S3 Data Consistency Models - 2 Types

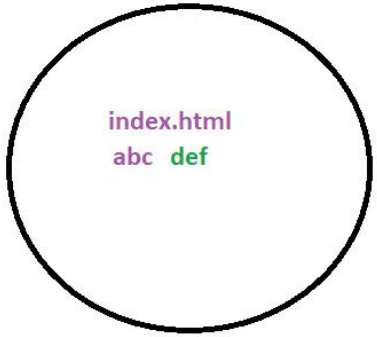
Read after write consistency for PUTS of New Objects

Eventually consistency for OVERWRITES of PUTS and DELETES

AES - 256

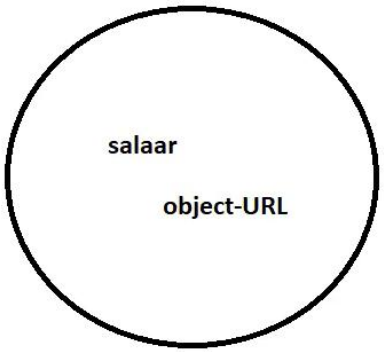
Advance Encryption Standard

By default, Bucket Encryption is Enabled



Bucket

Pre-Signed URL



Bucket

Endpoint

Endpoint is valid only for certain period of time

Ex : 5 mins

Afgter 5 mins, URL will expire

Temporary Purpose

Transfer Acceleration

Ireland Virginia Tokyo Sydney

internet 15 mins

CDN

Faster Upload

Hyderabad

ACL = Access Control List

S3 - Requester Pays

. In General, bucket owners pay for all S3 storage and data transfer cost associated with their bucket

. With Requester Pays Buckets, the requester instead of the bucket owner pays the costs of the requests and the data download from the bucket

. Helpful if you want to share the large data sets

. The requester must be authenticated in AWS so that AWS knows where to charge (in their AWS account). Cannot be anonymous

S3 Event Notifications

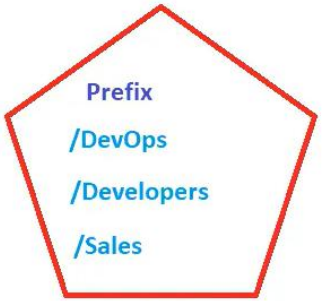
S3 Batch Operations

. Perform bulk operations on existing S3 objects with a single request

S3 Access Points

Access Point simplify security management for S3 buckets

Users	Access Points
DevOps Users	DevOps AP
Dev Users	Dev AP
Sales Users	Sales AP



S3 Bucket

Instead of writing critical bucket policies, you can create access points to each prefix and give the DNS Name to the users to acces their respective folders in Buckets

Access points can be Public (internet) or Private(VPC)

By default Object Nature is Private

If you create a Private Bucket, Objects are Private
If you create a Public BUcket, Objects are Still Private (If required make the object Public)

Encryption can be done in 2 ways

In-Transit Encryption : Encryption while data is moving / Transfering HTTPS

Data at Rest Encryption : Encryption while data is at Rest , KMS

Amazon Certificate Manager (ACM)

is where you can generate HTTPS certificates (In-Transit)

Key Management Service (KMS)

is where you can create encryption keys for data at rest

Encryption

Amazon S3 has 3 types of Encryption

Server Side Encryption

SSE - S3 = (AWS Managed Key)

SSE - KMS = (AWS KMS Key)

SSE- C = (Customer Provided Key)

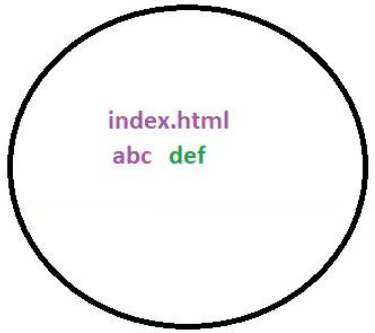
Client Side Encryption : Should be handled by Customer

In-Transit Encryption : HTTPS

AES - 256

Advance Encryption Standard

By default, Bucket Encryption is Enabled



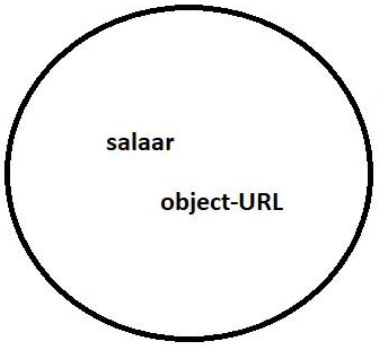
Bucket

S3 Data Consistency Models - 2 Types

Read after write consistency for PUTS of New Objects

Eventually consistency for OVERWRITES of PUTS and DELETES

Pre-Signed URL



Bucket

Endpoint

Endpoint is valid only for certain period of time

Ex : 5 mins

Afgter 5 mins, URL will expire

Temporary Purpose

Transfer Acceleration

Ireland Virginia Tokyo Sydney

internet 15 mins

CDN

Faster Upload

Hyderabad

ACL = Access Control List

CloudTrail

Provides goverence, compliance and audit for your AWS account
. CloudTrail is enabled by Default
. Get an history of events / API calls made in your AWS account
. Console
. SDK's
. CLI
. AWS Services
. Can put logs from Cloudtrail into Cloudwatch logs or S3
. A Trail can be applied to all regions or a single region
. If resource is deleted in AWS, investigate CloudTrail First

CloudTrail Events

Management Events

Operations performed on resources in AWS accounts

Ex: attachpolicy, create vpc, etc setting up logs etc

Data Events

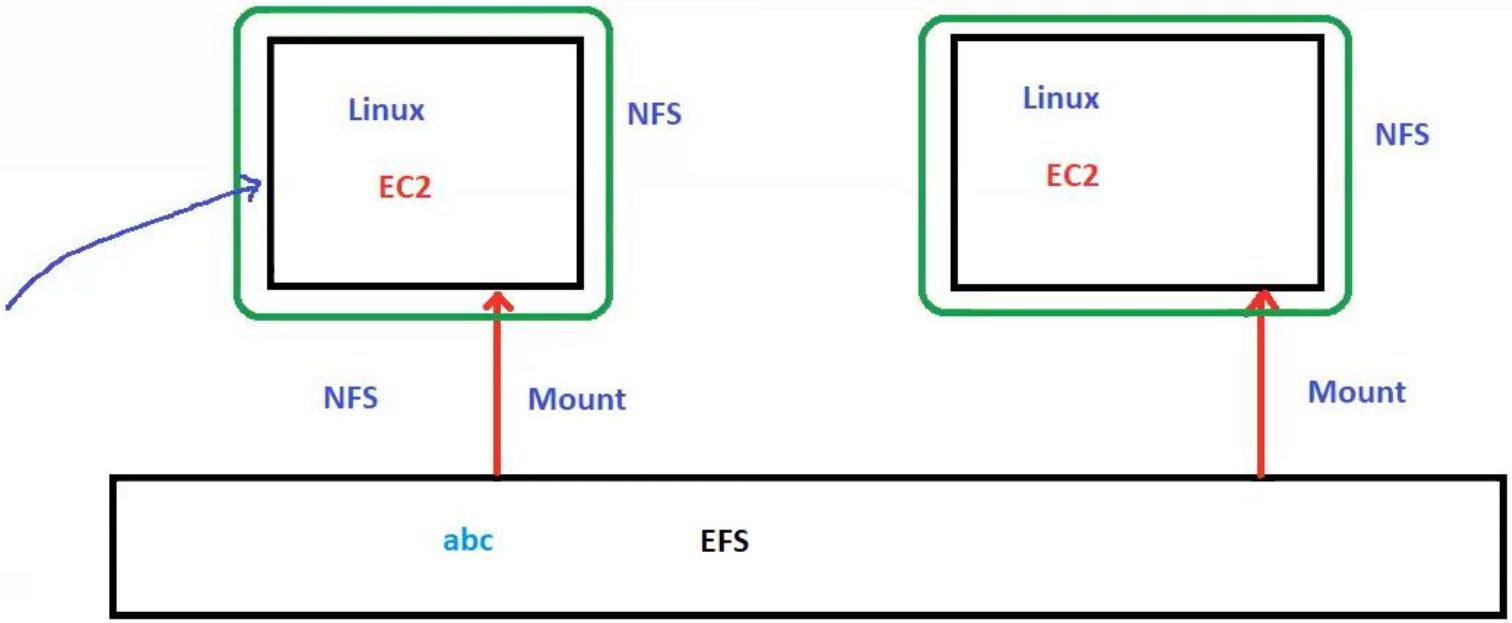
By default, data events are not logges (because of high volume operations)

Ex: getobject, deleteobject, putobject etc

Insight Events

. Enable CLoudTrail insights to detect unusual activities
. inaccurate provisioning
. hitting service limits
. bursts of IAM actions
CloudTrail insights analyze normal management events to create a baseline
. It detects write events to detect unusual patterns

Events are stored for 90 days in CloudTrail
To keep beyond this period, log them to S3 and use Athena



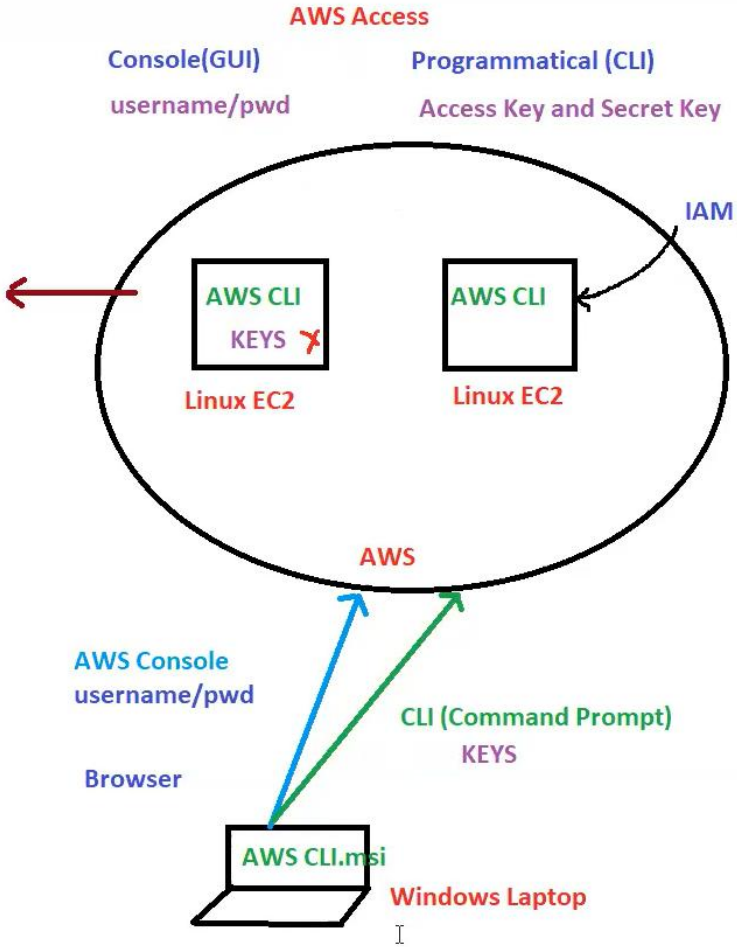
1. Create EFS
2. Launch EC2 instance and mount EFS and create some sample files
3. Launch another EC2 instance and mount the same EFS to the EC2 instance and check the files

yum install -y nfs-utils
mkdir efs
mount -t nfs4 filesystemdnsname:/ efs/
cd efs
create sample files

AWS CLI can be installed on Windows and Linux as well

For Windows ---> AWSCLI.msi

Linux ---> Use commands to install AWS CLI



IAM User
EC2
- Console
- CLI

BAD WAY : Launch Linux EC2 instance and install AWS CLI and configure KEYS and run sample commands

GOOD WAY : Launch Linux EC2 instance and install AWS CLI and attach ROLE and run sample commands

Install AWS CLI

Open Command Prompt

aws configure
access key
secret key
region : ap-south-1
output = table/json/xml

AWS CLI can be installed on Windows and Linux as well

For Windows ---> AWSCLI.msi

Linux ---> Use commands to install AWS CLI

AWS Access

Console(GUI)

username/pwd

Programmatical (CLI)

Access Key and Secret Key

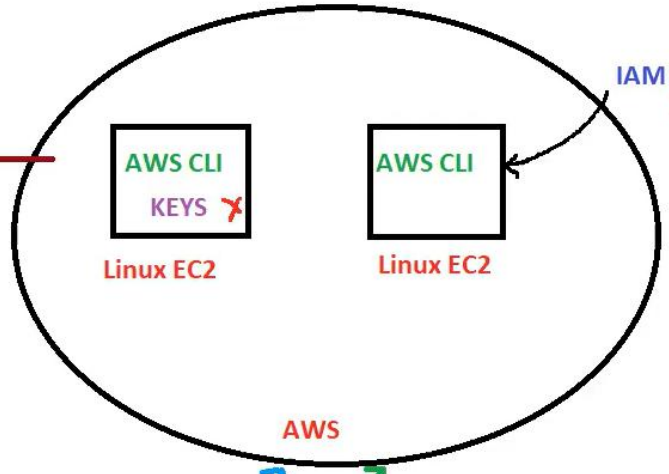
IAM User
EC2

Console

CLI

IAM ROLE
S3, EC2

Configuring the KEYS on the EC2 instance is BAD WAY



BAD WAY = Configure KEYS
GOOD WAY = USE ROLES

BAD WAY : Launch Linux EC2 instance and install AWS CLI and configure KEYS and run sample commands

GOOD WAY : Launch Linux EC2 instance and install AWS CLI and attach ROLE and run sample commands

1. Launch EC2 instance
2. Login and install AWS CLI
3. Create IAM ROLE and attach ec2 and s3 policies
4. Attach this IAM ROLE to the EC2 instance
5. Login to the EC2 instance and run the sample commands

Install AWS CLI

Open Command Prompt

aws configure
access key
secret key
region : ap-south-1
output = table/json/xml

cd ~/.aws
cat credentials

AWS Console
username/pwd

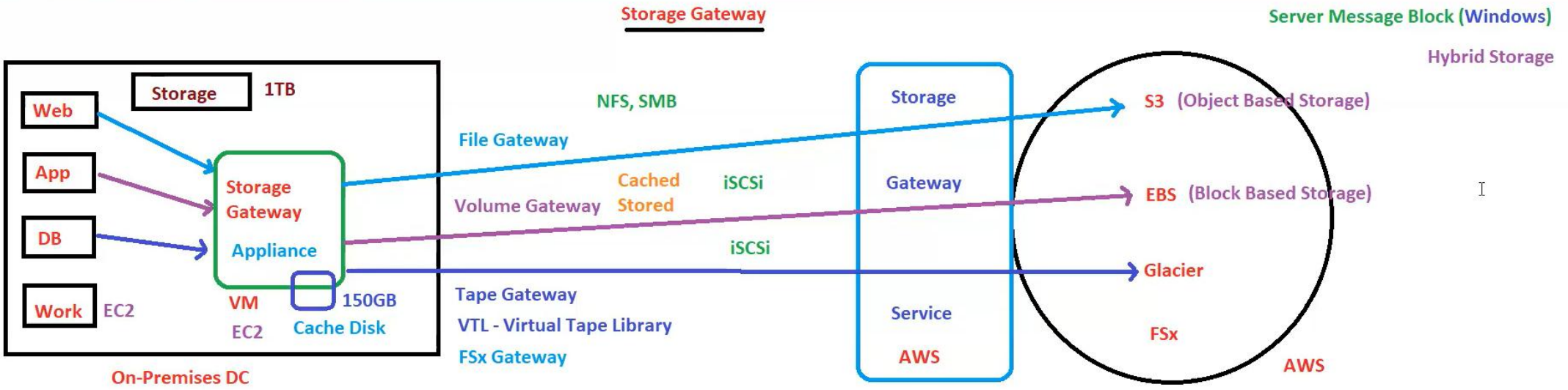
Browser

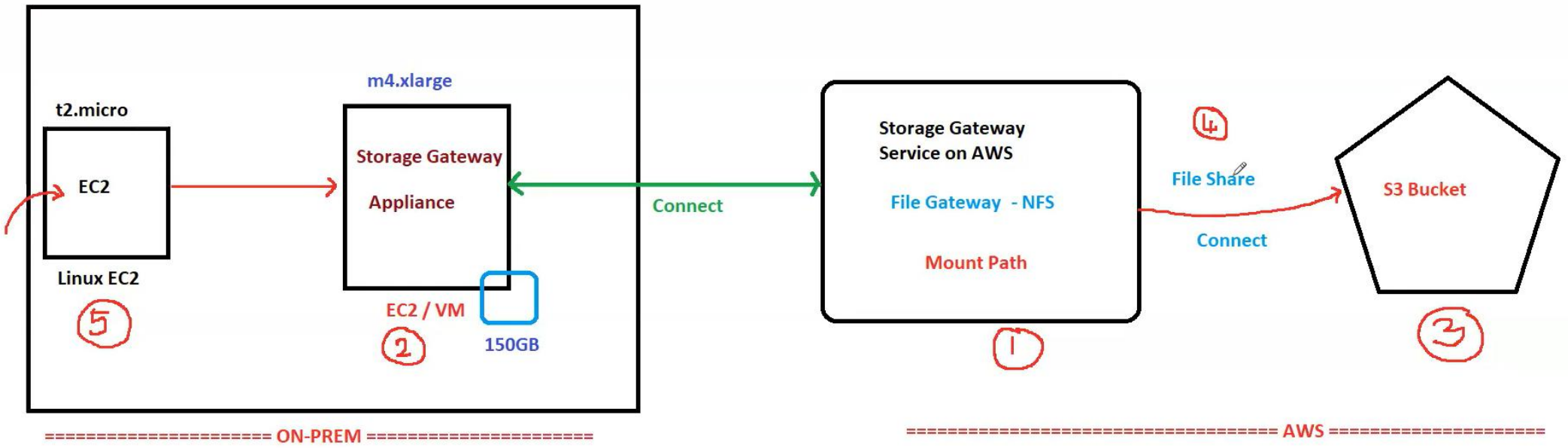
CLI (Command Prompt)
KEYS

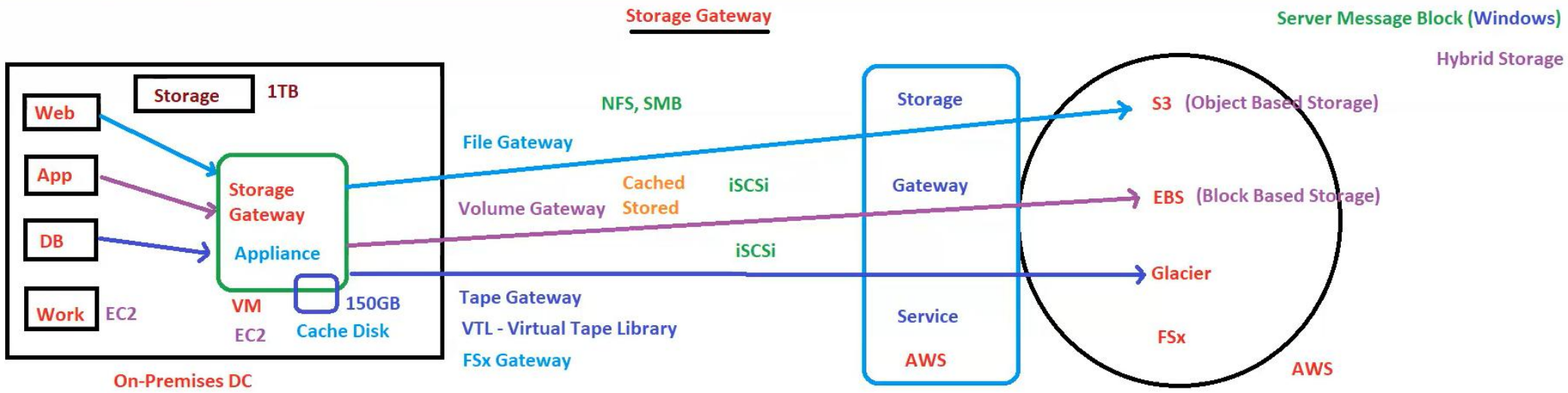
AWS CLI.msi

Windows Laptop
BAD WAY

TASKS: BAD WAY, GOOD WAY, and from your laptop(windows)


Storage Gateway
Server Message Block (Windows)
Hybrid Storage
Storage
1TB
Web
App
DB
Work
EC2
Storage Gateway
Appliance
VM
EC2
150GB
Cache Disk
On-Premises DC
NFS, SMB
File Gateway
Volume Gateway
Cached
Stored
iSCSi
iSCSi
Tape Gateway
VTL - Virtual Tape Library
FSx Gateway
Storage
Gateway
Service
AWS
S3 (Object Based Storage)
EBS (Block Based Storage)
Glacier
FSx
AWS


t2.micro
EC2
Linux EC2
5
m4.xlarge
Storage Gateway
Appliance
EC2 / VM
2
150GB
Connect
Storage Gateway
Service on AWS
File Gateway - NFS
Mount Path
1
4
File Share
Connect
S3 Bucket
3
========== ON-PREM ==========
========== AWS ==========

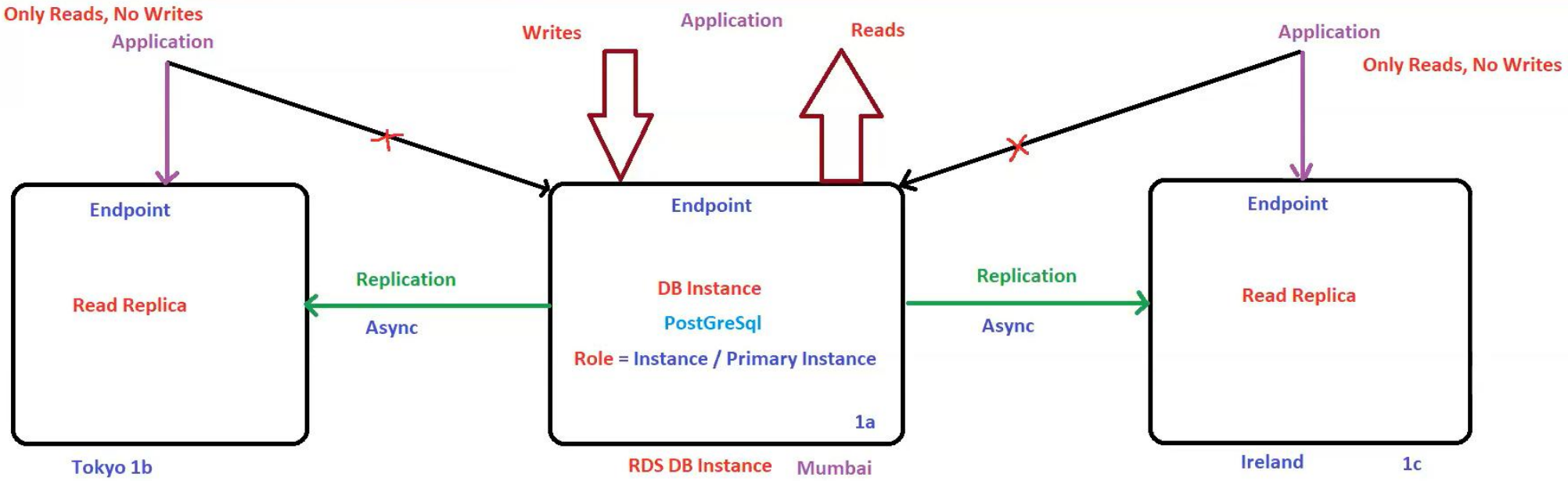


1. Go to AWS Storage Gateway Console and Create File Gateway
2. Launch EC2 instance with m5.xlarge having Root vol 80GB, add vol with 150GB
3. Enable All Traffic for default SG(Temporary, in real tiem use necessary protocols)
4. Continue on Storage Gateway Service Console, Connect to the Gateway: Provide EC2 instance Public IP (In real time, use EIP and VPC Hosted)
5. Wait until to get gateway active and allocate addtional 150GB as Cache Storage (no log group, no alarms)
6. Create a S3 Private Bucket
7. In Storage Gateway Service Console, Create a File Share and while creating file share choose NFS, Allow Client 0.0.0.0/0, and rest keep it defaults

8. Launch RedHat EC2 instance t2.micro for testing

```
sudo -s
yum install -y nfs-utils
mkdir gateway
mount -t nfs -o nolock,hard 172.31.26.106:/6pm-filegateway-bkt gateway/
cd gateway
touch hello.txt
```

Check hello.txt in S3



Application

HostName / Endpoint
Username
Password
Port Number

Endpoints are provided by AWS

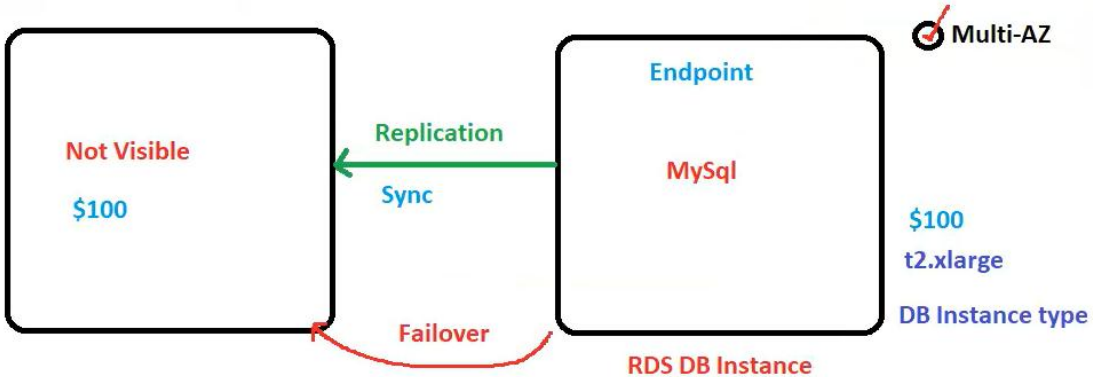
RDS is Regional
Read Replicas are used to increase the Performance
Max 15 Read Replicas
RR can be in diff AZ and cross region too
RR can be used for Read Purpose not Write Pupose
RR has its own Endpoints

It is possible to promote Read Replica to normal standalone instance

Multi-AZ

Multi-AZ is for high availability Purpose

Endpoint will remain same after failover



DB Operations will not have any failover , anything related to network, servers etc will have failover.

In Multi-AZ endpoints will not change after failover

You can enable Multi-AZ for RR also

For Multi-AZ charges are doubled

RDP, SSH are not possible to the RDS DB Instance

RDS DB Instances can be reserved

Parameter Group

Subnet Group

Backups ---> Snapshots

- Manual
- Automatic (Schedule)

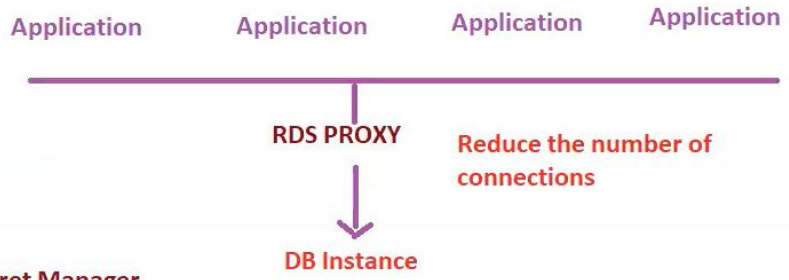
Performance Insights (Dashboard) for monitoring

Snapshots can be exported to S3 / Restore it from S3

Snapshots can be copied from one Region to another and accounts also

You can Scale Up DB instance anytime but NOT ON FLY (Stop and DB instance and change the instance type)

Auto-Scaling on Storage level



AWS Secret Manager

All keys, credentials, etc

ElastiCache

In-Memory database caching service

Cache: All frequently accessed data is stored at this place

Redis

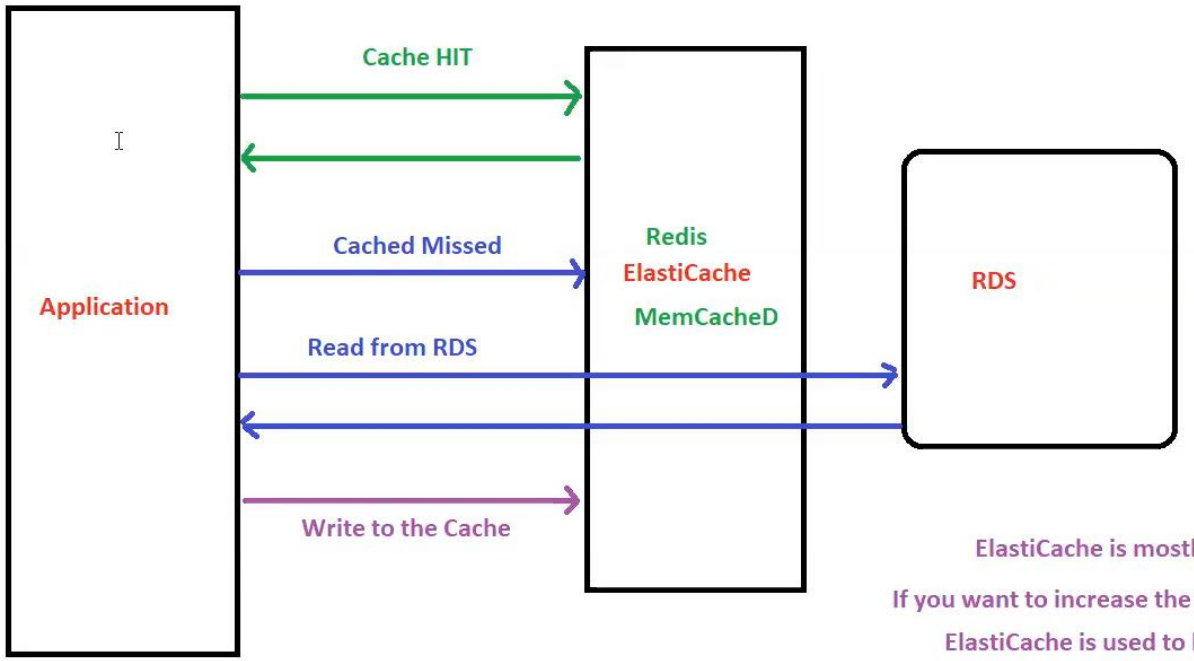
. Support HA, Failover
. Backups
. Data is Persistent

Developer Strategies

Lazy Loading : Load the data when ever necessary

Write Through : Parallely writing on DB and Cache

100% similar data on both db and cache



MemCacheD

. It doesnt support HA
No Failover
No Backups
Data is not Persistent

ElastiCache is mostly used for Read Purpose

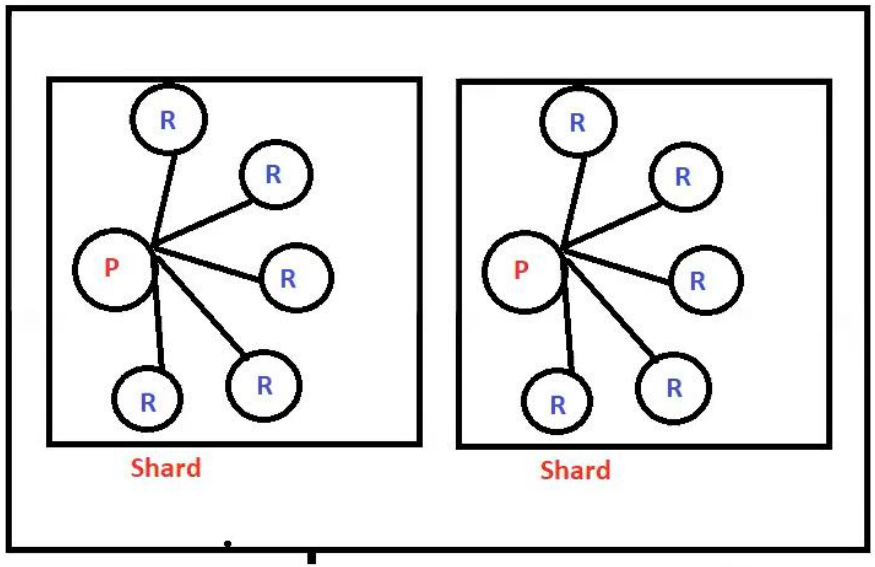
If you want to increase the performance choose ElastiCache

ElastiCache is used to handle session data

ElastiCahce support Encryption (In-Transit and Data at Rest)

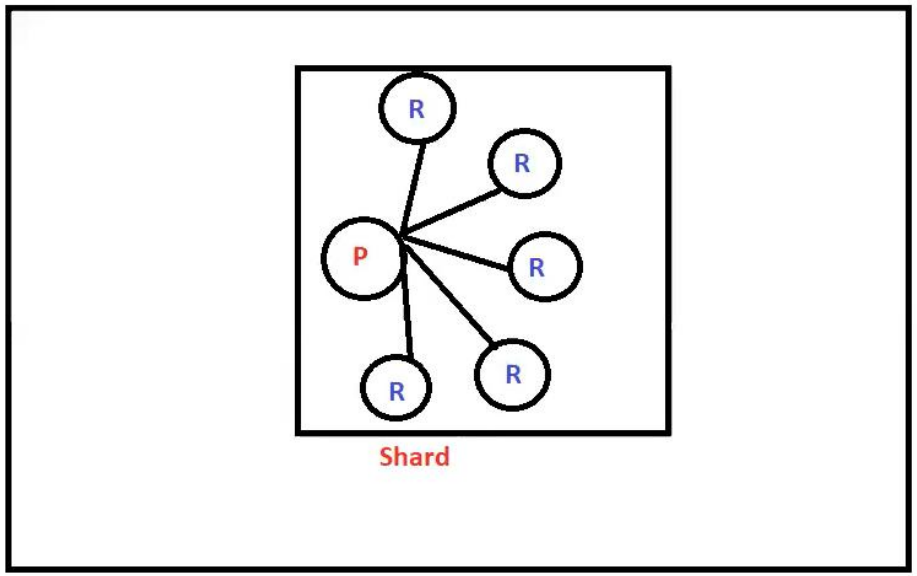
Redis

Cluster Mode Enabled



~ 500 Shards

Cluster Mode Disabled



1 Primary , 5 Replica Nodes

Shard = Collection of Nodes / Servers

Cluster = Collection of Shards

Each shard has 6 Nodes = 1 Primary , 5 Replica Nodes

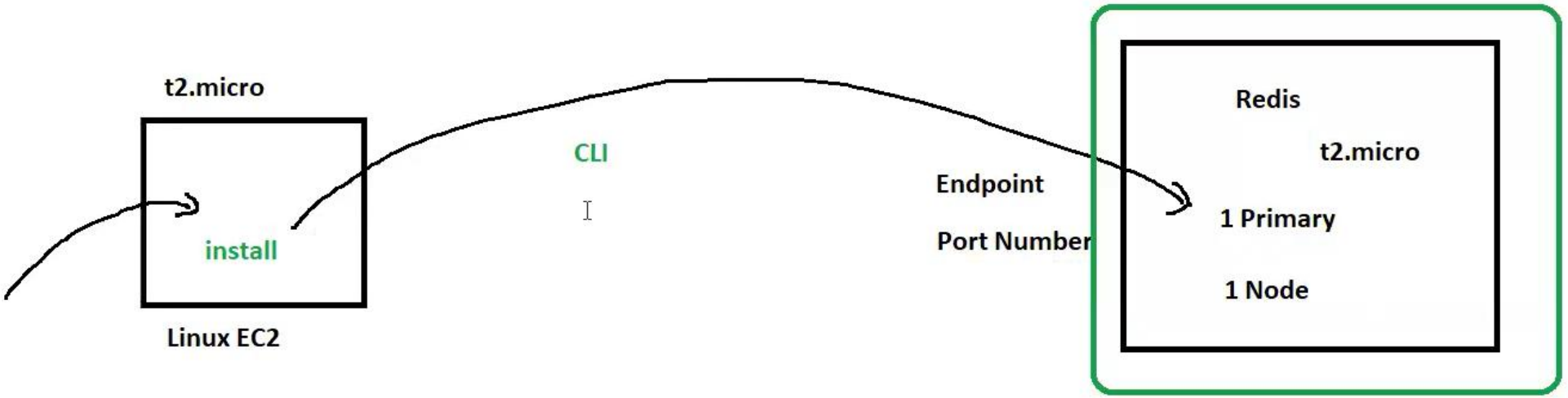
```
# App code

save_user(17, {"name": "Kevin McGehee"})
user = get_user(17)

# Write Through

def save_user(user_id, values):
    record = db.query("update users ... where id = ?", user_id, values)
    cache.set(user_id, record, 300)  # TTL
    return record

# Lazy Load

def get_user(user_id):
    record = cache.get(user_id)
    if record is None:
        record = db.query("select * from users where id = ?", user_id)
        cache.set(user_id, record, 300)  # TTL
    return record
```

Write Through

1. Updated DB
2. SET in Cache

Lazy Load

1. GET from cache.
2. If MISS get from DB
3. Then SET in Cache


t2.micro
install
Linux EC2
CLI
Endpoint
Port Number
Redis
t2.micro
1 Primary
1 Node

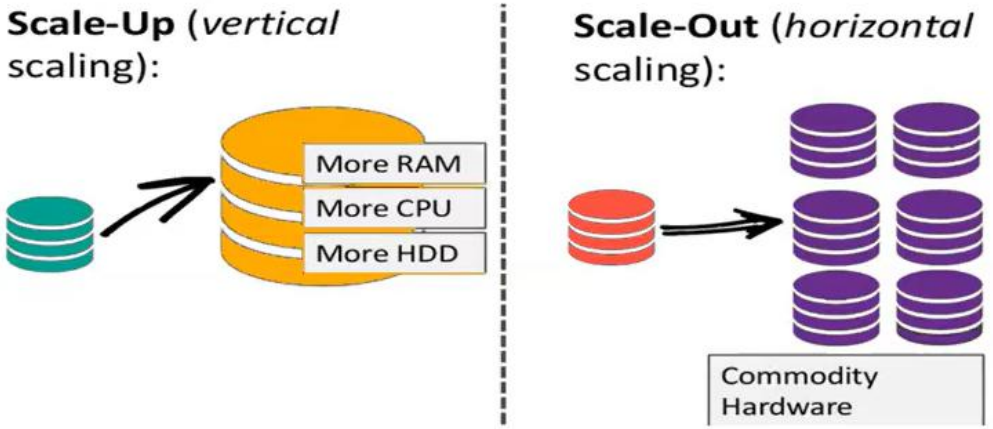
AWS DynamoDB

Traditional Architecture

- Traditional application leverage RDBMS databases
- These databases have the SQL query language
- Strong requirements about how the data should be modeled(tables, rows, columns, joins, indexes, etc)
- Ability to do join, aggregations, computations
- Vertical scaling (means usually getting a more powerful CPU/RAM/IO)

NoSQL databases

- The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data.
- The system response time becomes slow when you use RDBMS for massive volumes of data.
- To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive.
- The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."



NoSQL databases

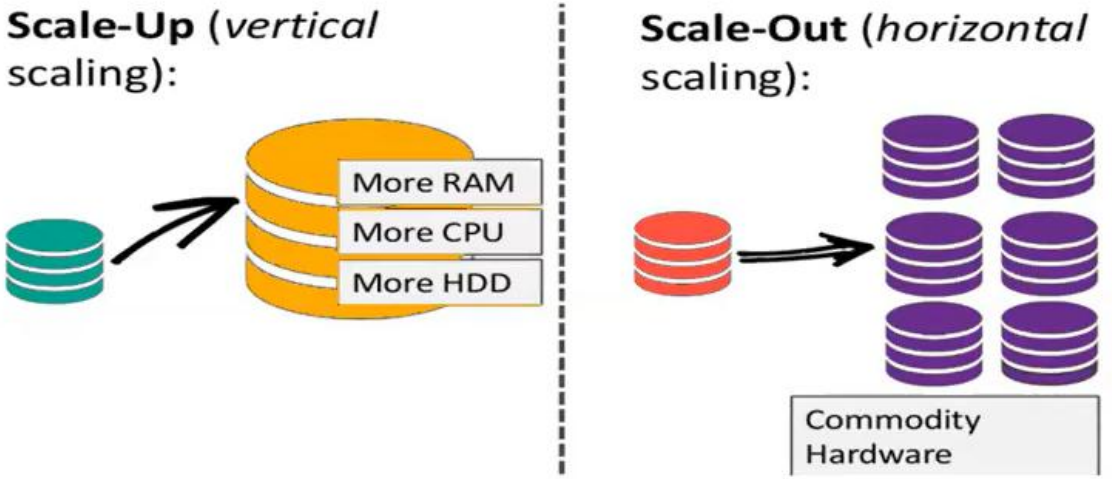
- NoSQL means, Not Only SQL.
- NoSQL databases are non-relational databases(no fixed columns etc) and are distributed(horizontal scaling).
- NoSQL databases include MongoDB, DynamoDB etc
- NoSQL databases do not support joins
- All the data that is needed for a query is present in one row
- NoSQL databases don’t perform aggregations such as “SUM”.
- NoSQL databases scale horizontally

SQL vs NOSQL databases

SQL	NoSQL
SQL is generally used in RDBMS	NoSQL is used for Non-Relational database system
Structured data can be stored in tables	Using JSON data, unstructured data can be stored
The Schemas are static	The Schemas are dynamic
Schemas are rigid and bound to relationships	Schemas are non-rigid, they are flexible
Helpful to design complex queries	No interface to prepare complex queries
Here we call tables, rows and columns	Here we call collections, collections has documents
MySql, Oracle, Sqlite, Postgres and MS-SQL	MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb

NoSQL databases

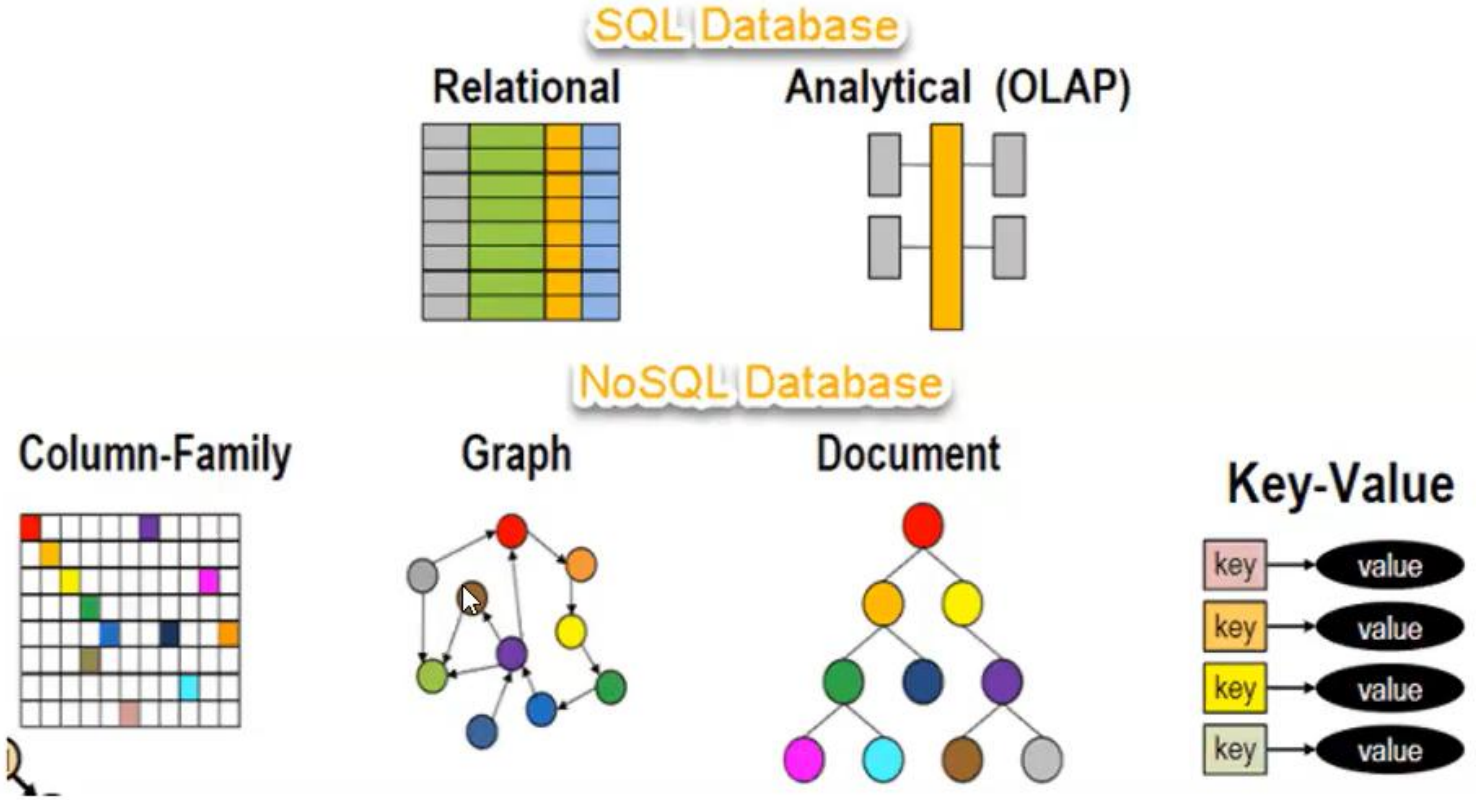
- The concept of NoSQL databases became popular with Internet giants like Google, Facebook, Amazon, etc. who deal with huge volumes of data.
- The system response time becomes slow when you use RDBMS for massive volumes of data.
- To resolve this problem, we could "scale up" our systems by upgrading our existing hardware. This process is expensive.
- The alternative for this issue is to distribute database load on multiple hosts whenever the load increases. This method is known as "scaling out."



SQL vs NOSQL databases

SQL	NoSQL
SQL is generally used in RDBMS	NoSQL is used for Non-Relational database system
Structured data can be stored in tables	Using JSON data, unstructured data can be stored
The Schemas are static	The Schemas are dynamic
Schemas are rigid and bound to relationships	Schemas are non-rigid, they are flexible
Helpful to design complex queries	No interface to prepare complex queries
Here we call tables, rows and columns	Here we call collections, collections has documents
MySql, Oracle, Sqlite, Postgres and MS-SQL	MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j and CouchDb

NoSQL databases



DynamoDB vs RDS

- DynamoDB offers “push button” scaling , meaning that you can scale your database on the fly, without any down time.
- DynamoDB is Serverless
- RDS is not so easy and you usually have to use a bigger instance size or to add a read replica.

Basics of DynamoDB

- Stored on SSD storage
- Spread across 3 geographically distinct data centers
- Eventual Consistent **Reads**(Default)
- Strongly Consistent **Reads**

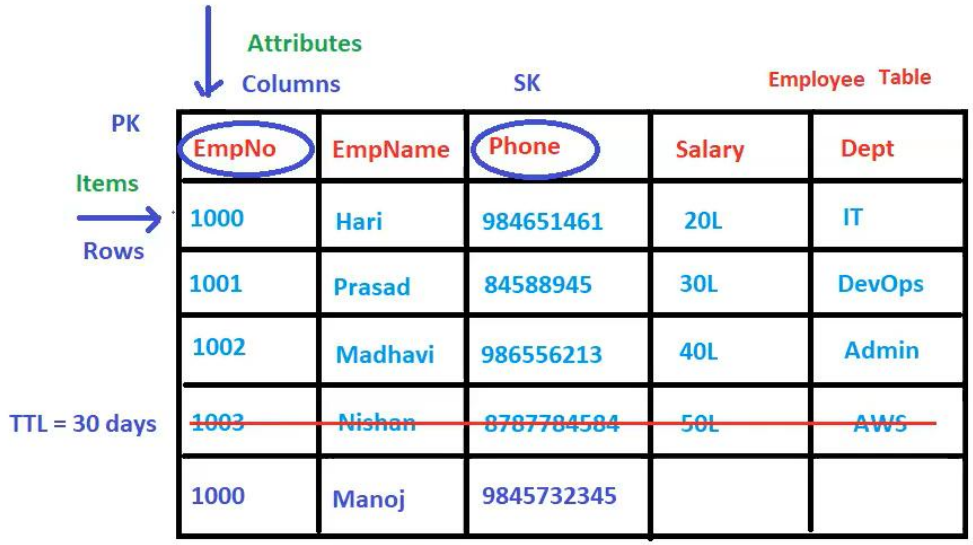


Table = Attributes + Items

Primary Key = The value should be always unique and not null

Primary Key = Partition Key

In DynamoDB, initially you create table with only one single column, and that column should be Partition Key

Primary Key is mandatory in DynamoDB while creating a table

Composite Key = Partition Key + Sort Key

Sort Key is Optional, PK is Mandatory

If you have a primary key for a single column = Duplicate values are NOT allowed

If you have a PK + SK = Primary Key duplicate values are Allowed

EmployeeNumber = Primary Key / Hash Attribute

Phone Number = Sort Key / Range Attruibute

Document
Key-Value

DynamoDB data is stored in JSON format

Consistency Models

READ

WRITE

ECR

default

SCR

Eventually Consistency Reads

Strongly Consistency Reads

DynamoDB Streams = Change Logs

Kinesis Data Streams

Captures item level changes

Indexes

It helps to increase the performance on the table / Retriving the data

DynamoDB performance depend upon RCU and WCU

LSI (Local Secondary Index) = Partition key + Any Column as SK

GSI (Gobal Secondary Index) = Any Column as Partition key + Any Column as SK

5 RCU, 5 WCU --> default

1 RCU = 5.2 million reads
1 WCU = 2.5 million writes

LSI can be created ony at the time of creating dynamodb table, later you cannot modity or delete it

GSI can be created , modified and delete anytime

Provisioned Capacity Units

Read Capacity Units

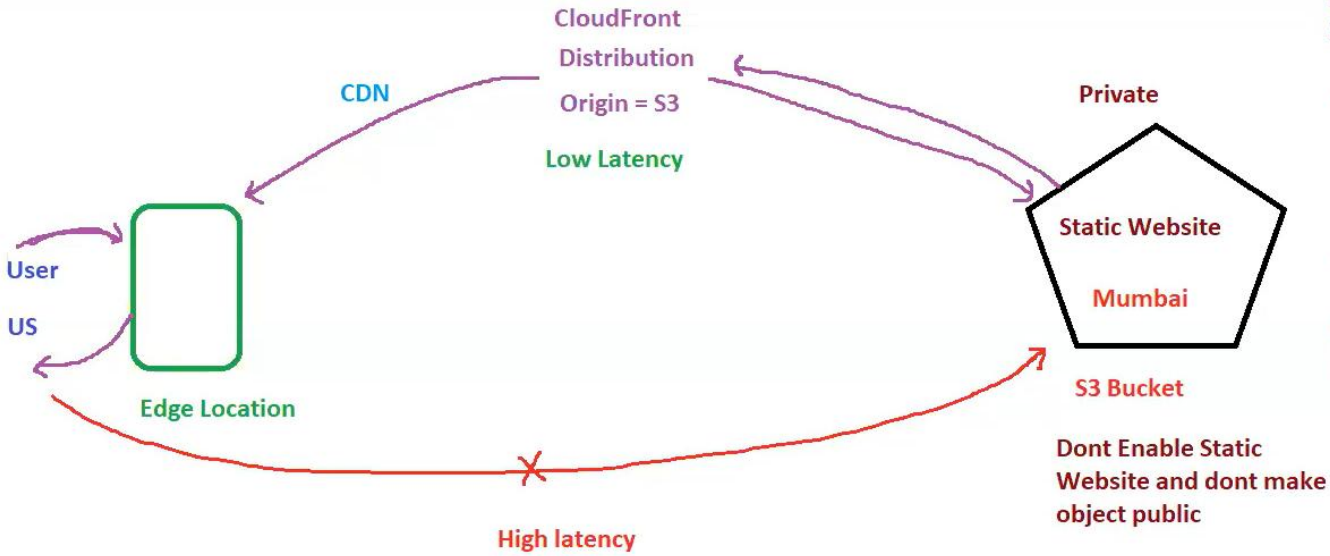
RCU

For ECR, 1 RCU = 2 Reads Per Second for 4KB Size
For SCR, 1 RCU = 1 Reads Per Second for 4KB Size

Write Capacity Units

WCU

1 WCU = 1 Write Per Second for 1KB Size



1. Create a Private S3 Bucket
2. Upload a Static Websiste into S3
3. Dont enable static website option and dont make the objects public

4. Create CloudFront Distribution
   Origin ---> S3 Bucket
   OAC --> Create origin access setting
   root object = index.html

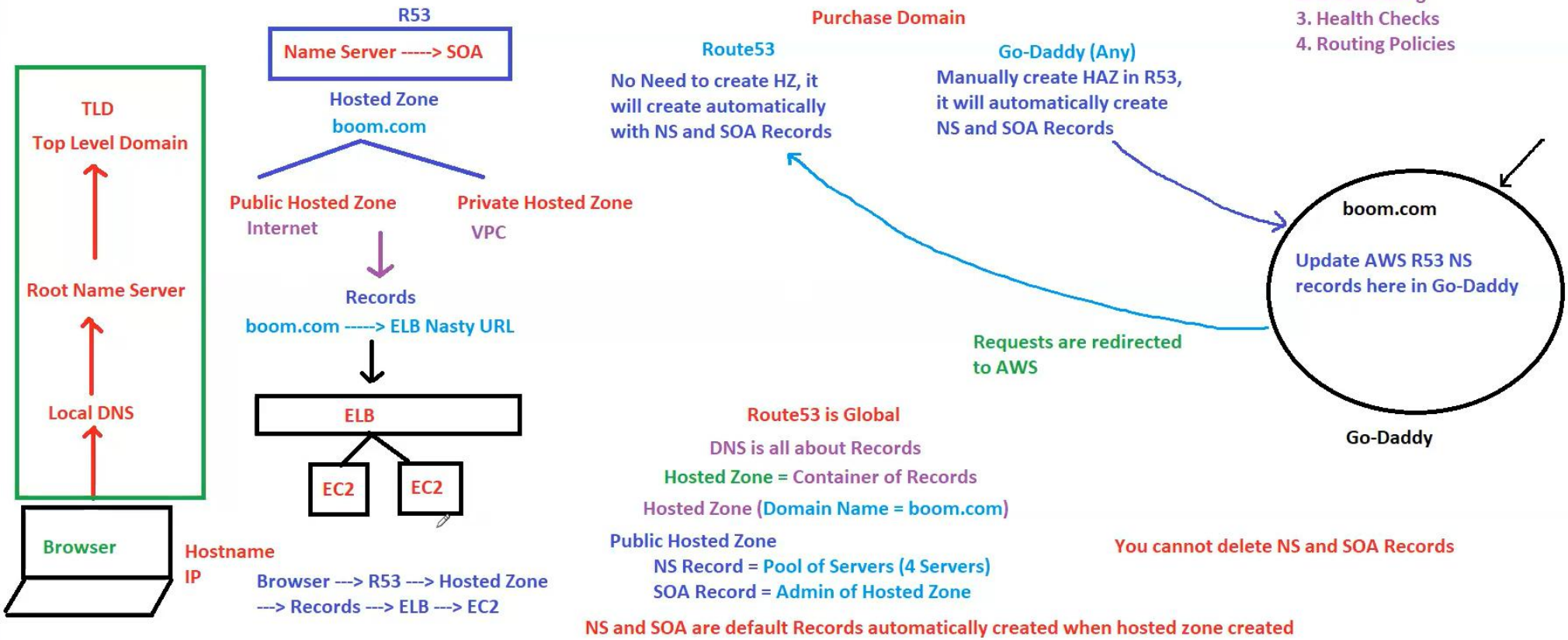
rest everything leave it defaults

Copy the policy and update in S3 bucket policy
Access the website with CloudFront URL

DNS = Keep tracks of all hostnames and IP address
= It converts hostname to ip and ip to hostname

Route53
DNS Service from AWS, DNS Port Number is 53

Route53 Features
1. Domain Registration
2. DNS Routing
3. Health Checks
4. Routing Policies



Route53 is Global

DNS is all about Records

Hosted Zone = Container of Records

Hosted Zone (Domain Name = boom.com)

Public Hosted Zone

NS Record = Pool of Servers (4 Servers)

SOA Record = Admin of Hosted Zone

You cannot delete NS and SOA Records

NS and SOA are default Records automatically created when hosted zone created

Route53 Records

A Record = URL to IPV4

AAAA Record = URL to IPV6

CName Record = URL to URL

Alias Record = URL to Any Resource

MX Record = Emails

A Record

http://raj.com

↓

Route53

↓

Hosted Zone (Raj.com)

↓

raj.com ---> EC2 instance IP

URL ---> IPV4

CNAME Record

http://raj.com

↓

Route53

↓

Hosted Zone (Raj.com)

↓

raj.com ---> ELB DNS URL

URL ---> URL

Alias Record

http://raj.com

↓

Route53

↓

Hosted Zone (Raj.com)

↓

raj.com ---> EC2 instance IP / ELB DNS Name

URL ---> IPV4 / ELB

URL ---> Any Resource

http://raj.com

Main Domain

Naked Domain

Zone Apex Record

admin.raj.com

hello.raj.com ---> Sub-Domains

photos.raj.com

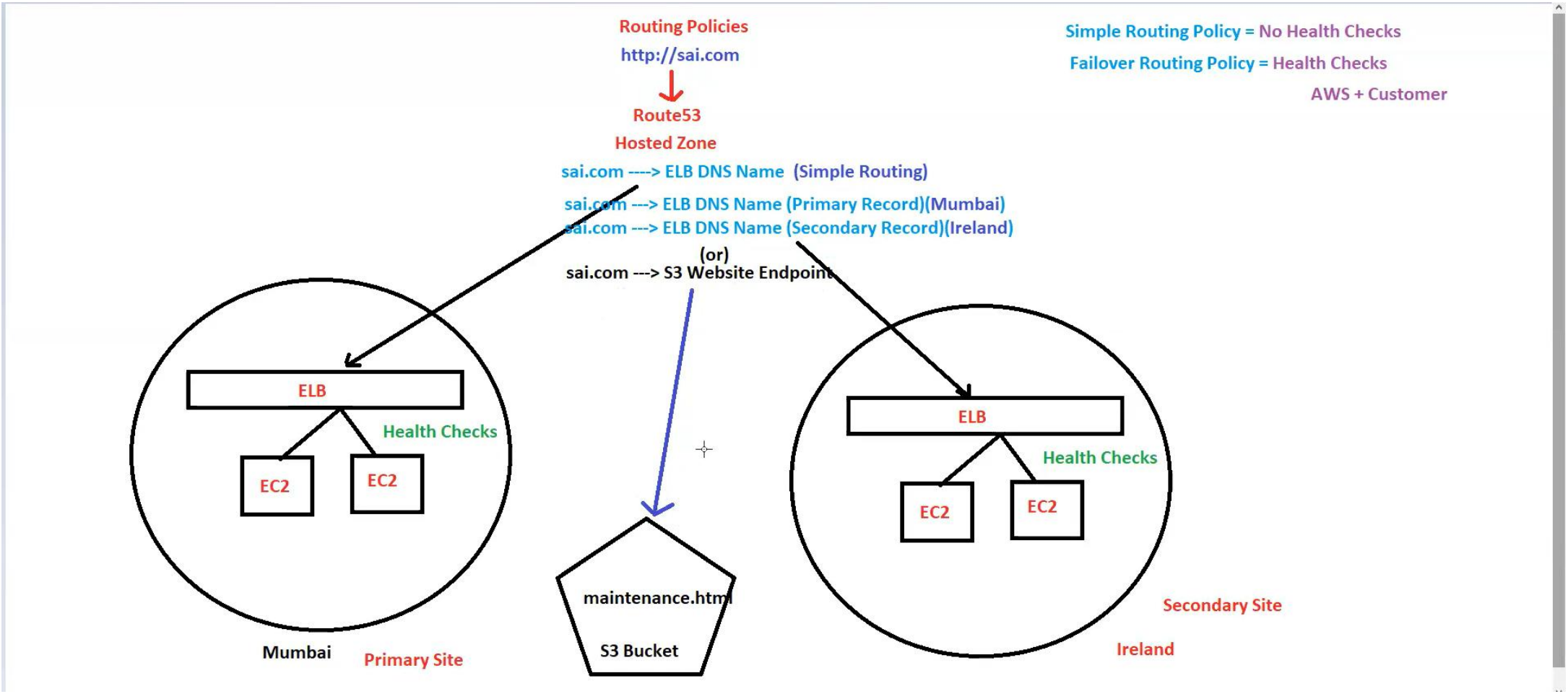
CNAME Records are Billable, Alias Records are FREE

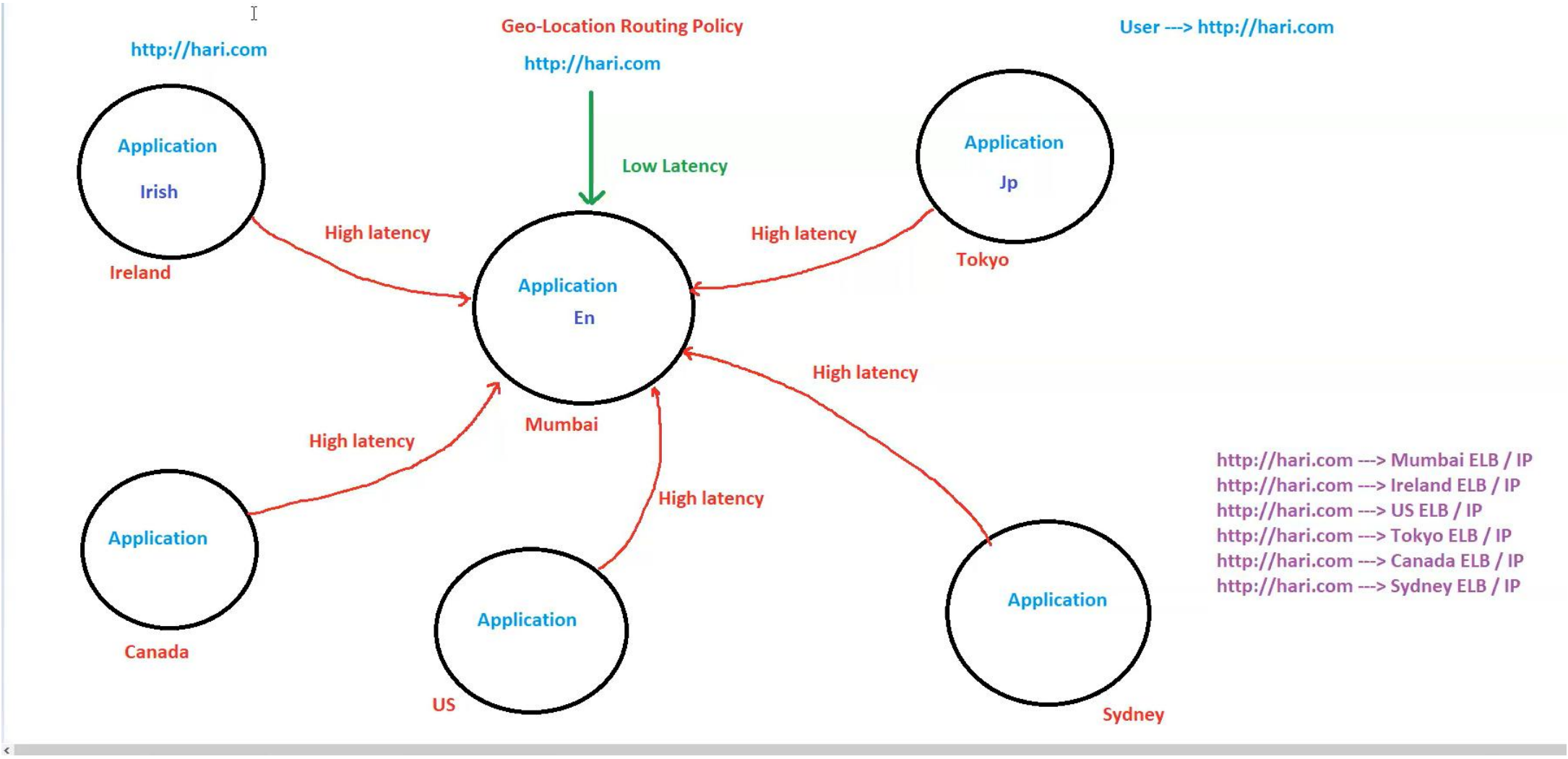
For Naked Domain, raj.com cannot use CNAME records, instead use Alias

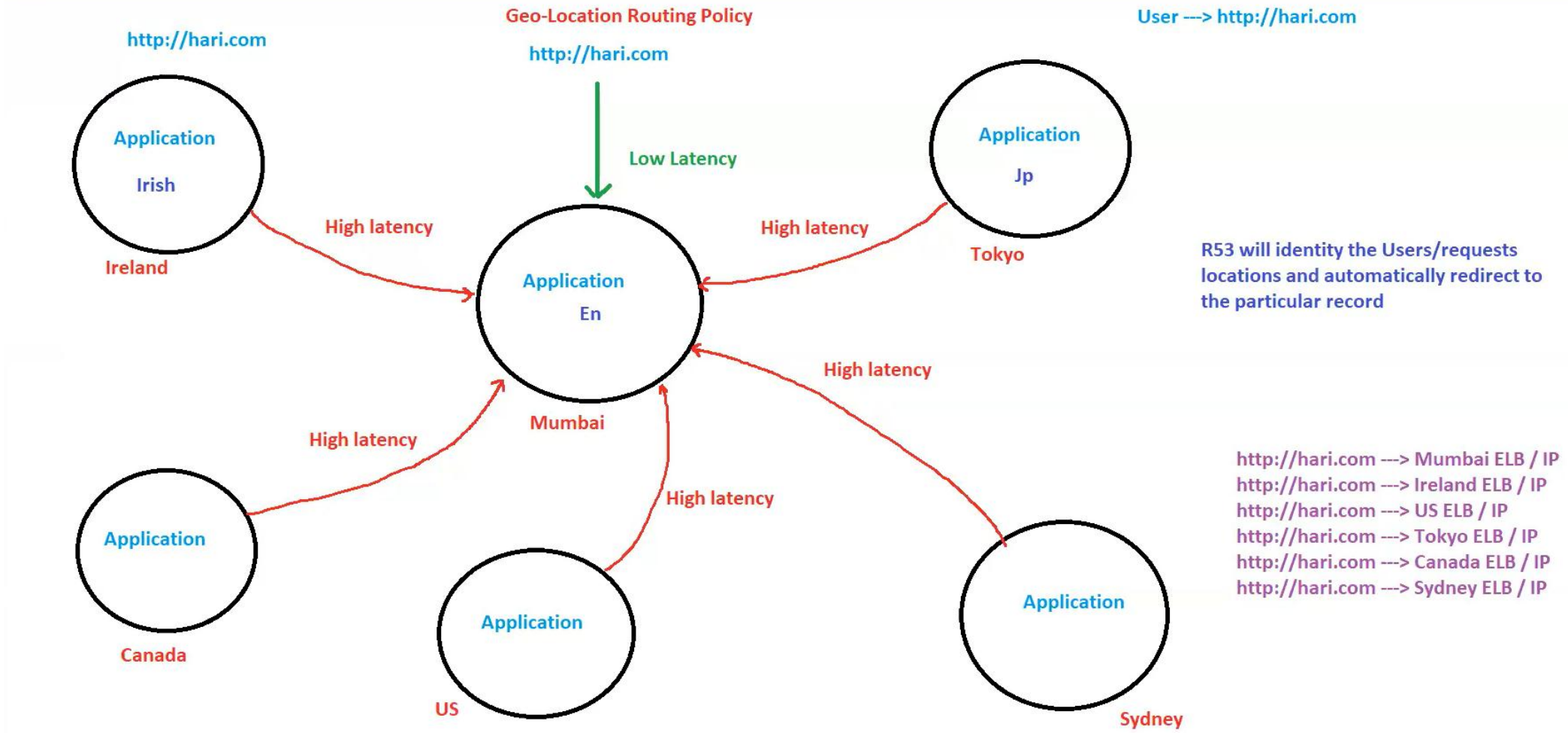
For all Sub-Domains you can use CNAME

Always choose Alias over CNAME

A Record + Alias Record


Routing Policies
http://sai.com
Route53
Hosted Zone
sai.com ----> ELB DNS Name (Simple Routing)
sai.com ---> ELB DNS Name (Primary Record)(Mumbai)
sai.com ---> ELB DNS Name (Secondary Record)(Ireland)
(or)
sai.com ---> S3 Website Endpoint
Simple Routing Policy = No Health Checks
Failover Routing Policy = Health Checks
AWS + Customer
ELB
Health Checks
EC2
EC2
Mumbai
Primary Site
maintenance.html
S3 Bucket
ELB
Health Checks
EC2
EC2
Secondary Site
Ireland


http://hari.com
Geo-Location Routing Policy
http://hari.com
User ---> http://hari.com
Application
Irish
Ireland
Low Latency
Application
Jp
Tokyo
High latency
High latency
Application
En
Mumbai
High latency
High latency
High latency
Application
Canada
Application
US
Application
Sydney
http://hari.com ---> Mumbai ELB / IP
http://hari.com ---> Ireland ELB / IP
http://hari.com ---> US ELB / IP
http://hari.com ---> Tokyo ELB / IP
http://hari.com ---> Canada ELB / IP
http://hari.com ---> Sydney ELB / IP


Geo-Location Routing Policy
http://hari.com
http://hari.com
Low Latency
User ---> http://hari.com
Application
Irish
Ireland
High latency
Application
En
Mumbai
High latency
Application
Jp
Tokyo
R53 will identity the Users/requests locations and automatically redirect to the particular record
High latency
High latency
Application
Canada
High latency
Application
US
Application
Sydney
http://hari.com ---> Mumbai ELB / IP
http://hari.com ---> Ireland ELB / IP
http://hari.com ---> US ELB / IP
http://hari.com ---> Tokyo ELB / IP
http://hari.com ---> Canada ELB / IP
http://hari.com ---> Sydney ELB / IP

VPC
Internet Gateway
Public and Private Subnet
NAT Gateway
Router with Routing tables

IGW ---> Provides internet to VPC

Public Subnet --> which is exposed to internet

Private Subnet --> which is NOT exposed to internet

Public Subnet traffic is routed to IGW

Private Subnet traffic is routed to NAT Gateway

NAT Gateway should be in Public Subnet

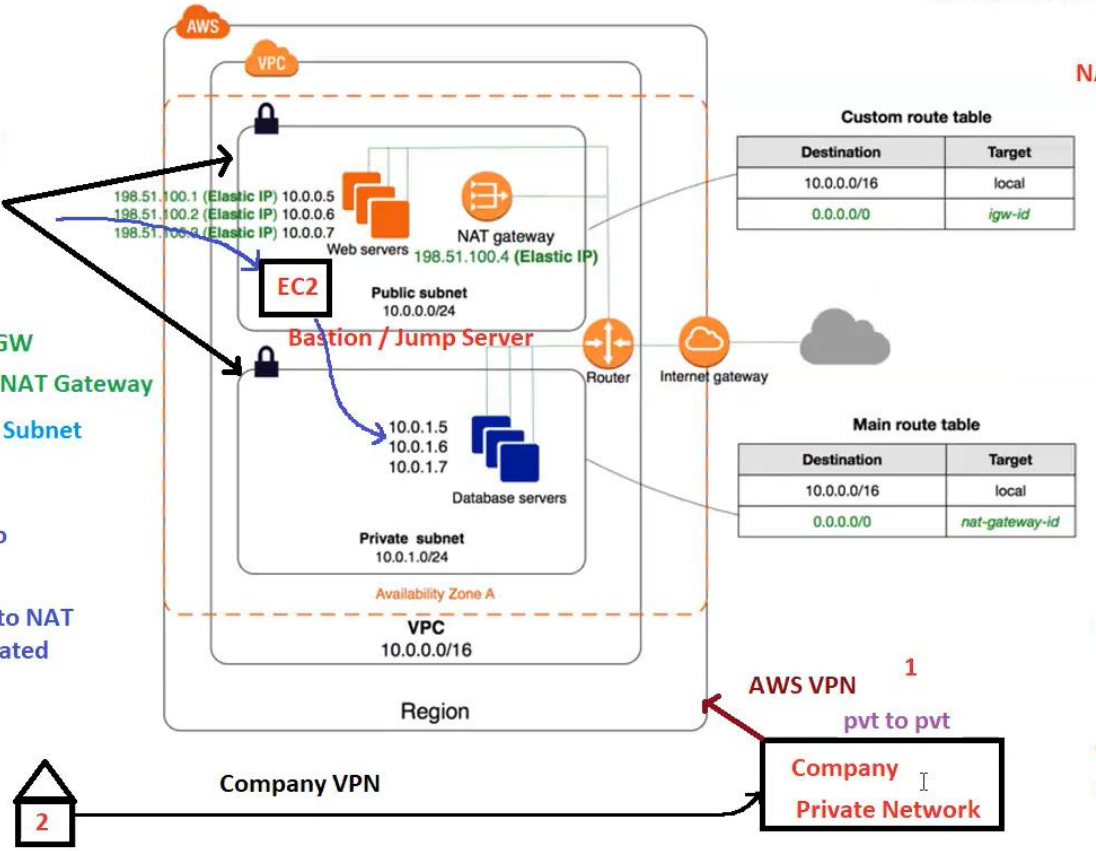
Router

Public RT --> All traffic is routed to IGW, Public Subnet is associated

Private RT --> All traffic is routed to NAT Gateway, Private Subnet is associated

1. Actual Real time
2. Work from home
3. Learning

VPC = Virtual Private Cloud



Custom route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	igw-id

Main route table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	nat-gateway-id

VPC is Regional, Max 5 VPC's per Region
We already have a default VPC provided by AWS

NAT = Network Address Translator

NAT is used to provide internet access to Private Subnet

NAT will convert Private Ip to Public IP

1. Create VPC
2. Create IGW and attach it to the VPC
3. Create Public Subnet
4. Create Private SUbnet
5. Create NAT Gateway (Public Subnet, EIP)
6. Create Public RT --> All Traffic is routed to IGW, Public Subnet is associated
7. Create Private RT --> All Traffic is routed to NAT Gateway, Private Subnet is associated
8. Create a Security Group
9. Launch EC2 instance in Public Subnet (Bastion Server) and another in Private subnet (PrivateServer)
10. Connect to Bastion Server and then to Private Server

NAT and IGW are Services not Servers

NAT Instance ---> Outdated
NAT Gateway --> Managed by AWS

VPC Endpoints: It is used to access only AWS Services without NAT

NAT = Full internet access

S3 VPCE
DynamoDB VPCE
etc

Private Purpose

192
172
10

Rest are all used for Public IP's

1 Subnet should be in 1AZ at the same time

1 AZ can have multiple Subnets

2^32-n

2^32-24 = 2^8 = 256 - 5 = 251

2^32-20 = 2^12 = 4096 - 5 = 4091

2^32-16 = 2^16 = 65536 - 5 = 65531

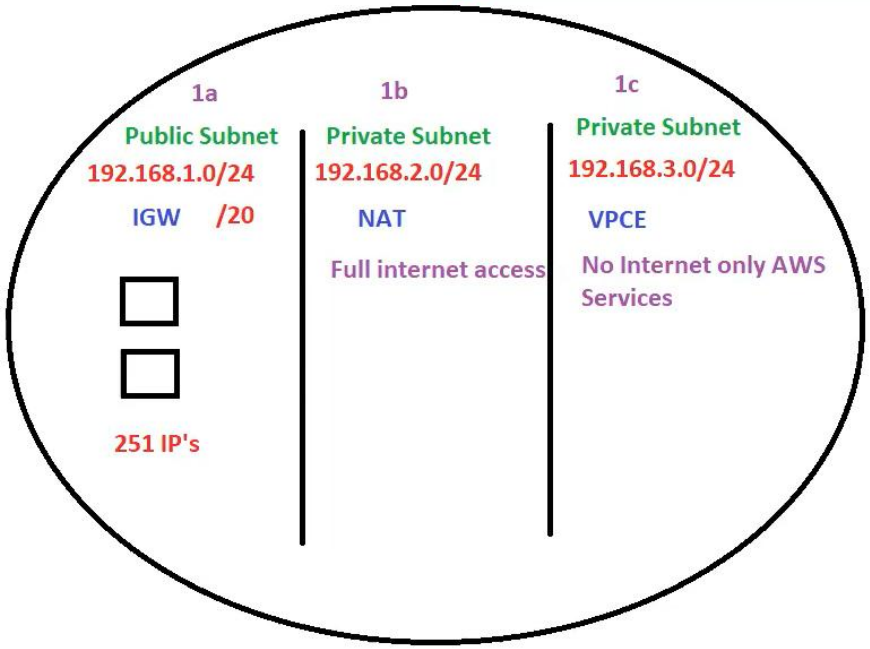
2^32-32 = 2^0 = 1

CIDR = Classless InterDomain Routing

Base IP Subnet Mask

VPC (192.168.0.0/16)

/8

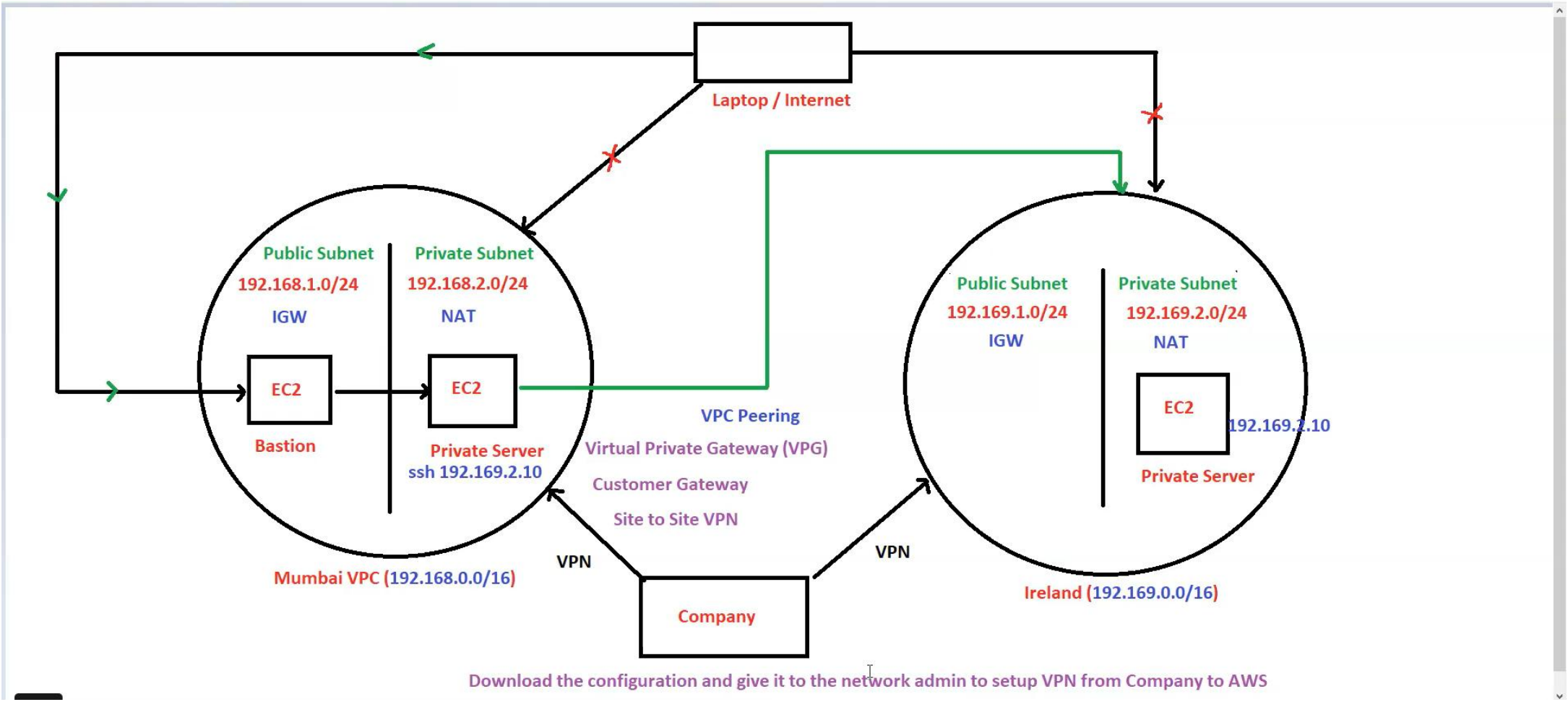


5 IP's are Reserved for each Subnet

.0 = Network
.1 = Router
.2 = DNS
.3 = Future
.255 = BroadCasting

1. Create VPC (192.168.0.0/16)
2. Create IGW and attach it to the VPC
3. Create Public Subnet (192.168.1.0/24)
4. Create Private Subnet (192.168.2.0/24)
5. Create NAT Gateway (Public Subnet, EIP)
6. Create Public RT ---> 0.0.0.0/0 routed to IGW, Public Subnet is associated
7. Create Private RT ---> 0.0.0.0/0 is routed to NAT Gateway, Private Subnet is associated
8. Create a Security Group
9. Launch EC2 instance in Public SUbnet (Bastion) and another in Private Subnet (Private Server)
10. Connect to bastion and then to Private Server

```
In Bastion Server

sudo -s
vi 6PMBATCHNEW.pem
 ---> press i
 ---> copy paste the pem file data
 ---> press esc, :wq!

chmod 400 6PMBATCHNEW.pem
ssh -i "6PMBATCHNEW.pem" ec2-user@192.168.2.11

from here you are in mumbaiprivate server

sudo -s
Try internet is working or not by installing a sample command yum install -y wget
Switch  OFF internet by removing the NAT entry in Private RT
Switch  ON internet by adding the NAT entry in Private RT
Install AWS CLI
Create a IAM ROLE with S3 Permissions and attach ROLE to the EC2 instance
Create a sample S3 bucket from AWS CLI (MumbaiPrivate Server) [this got created because we have NAT entry]
Remove NAT entry in Private RT
List S3 bucket from AWS CLI --> cannot access because no internet as you removed NAT entry
Create a VPCE
And List S3 bucket [this will work because we have now VPCE]



There are three types of VPC endpoints – Interface endpoints, Gateway Load Balancer endpoints, and Gateway endpoints.
```


Laptop / Internet
Public Subnet
192.168.1.0/24
IGW
Private Subnet
192.168.2.0/24
NAT
EC2
Bastion
EC2
Private Server
ssh 192.169.2.10
Mumbai VPC (192.168.0.0/16)
VPC Peering
Virtual Private Gateway (VPG)
Customer Gateway
Site to Site VPN
VPN
VPN
Company
Public Subnet
192.169.1.0/24
IGW
Private Subnet
192.169.2.0/24
NAT
EC2
192.169.2.10
Private Server
Ireland (192.169.0.0/16)
Download the configuration and give it to the network admin to setup VPN from Company to AWS