**ELASTICITY**

* + Increasing and decreasing the capacity to meet increasing and decreasing work load
  + Elasticity is short term
  + Elasticity is also called as Horizontal Scaling
  + Elasticity can be achieved in AWS using Auto-Scaling (AG)
  + Auto-Scaling = Scale Out and Scale In

1. Scale Out : Adding/increasing servers(instances)
2. Scale In : Removing/decreasing servers(instances)

**SCLABILITY**

* Increasing the capacity of the server (i.e same machine) is called Scalability
* Scalability for long term
* Scalability = Scale Up and Scale Down
* Scalability is also called as Vertical Scaling
* Capacity means combination of both CPU + RAM

|  |
| --- |
| DB SERVER |

16 GB

8 GB

**HIGH AVAILABILITY:**

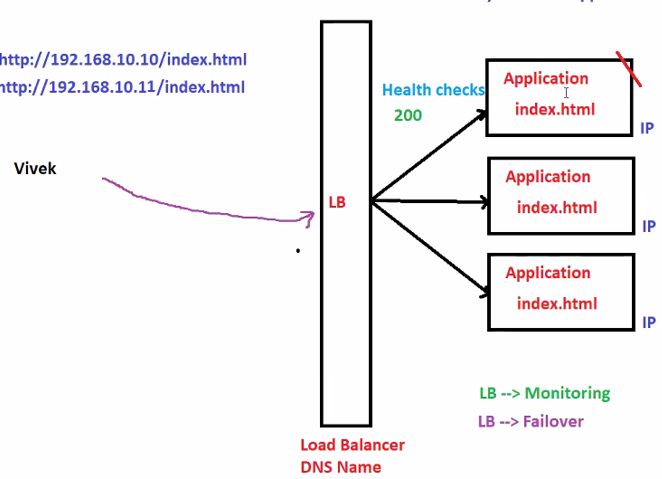
* The period of time service is available to the customer is called High Availability

**LOAD BALANCER:**

* Distribute the traffic to different server is called Loan Balancer (LB)
* Load balancer (LB) always monitor application not the server
* Redundancy: Keeping same application on different server
* Monitoring: Doing health check is called Monitoring (i.e LB will take care)
* Failover: One server goes down other server pick it up

**HIGH AVAILABILITY**

|  |
| --- |
| Redundancy  Monitoring  Failover |



Note: Customer is not interested to access his application using IP address, instead us LB DNS Name

**REGION:**

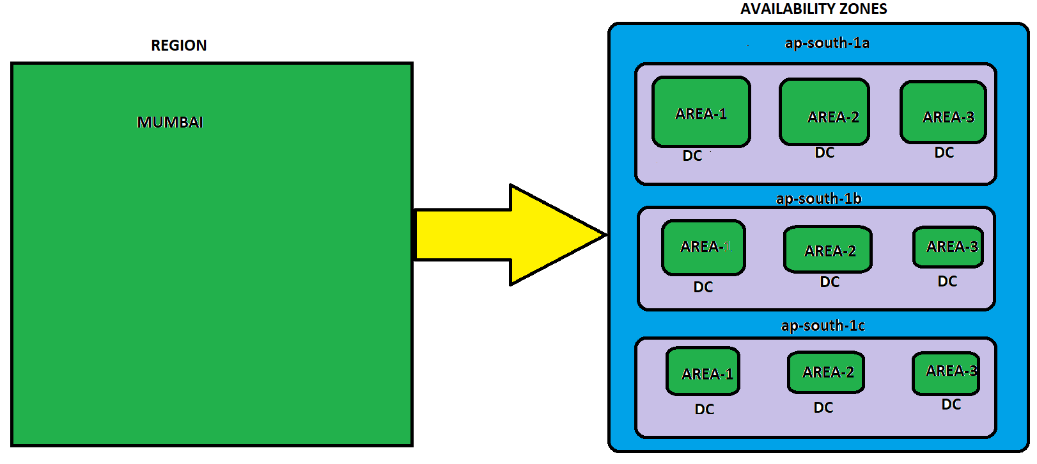
* Region is a geographical area

Example : Mumabai

* Region don’t communicate each other by default, if required yes.
* A region has multiple datacenters
* A region has multiple availability zones (AZ’s)

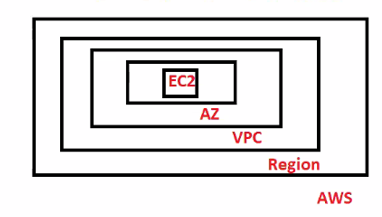
**AVAILABILITY ZONE:**

* Nothing but group of datacenters (DC)
* Availability zones (AZ’s) has instances/servers
* Availability zones(AZ’s) can communicate with each other
* Availability zones are sync with each other (i.e network) no the data
* Very less chance to that 1 AZ goes down
* Regions and Availability Zones (AZ’s) are managed by AWS



**VPC(Virtual Private Cloud)**

* VPC is reginal
* AWS will provide default VPC
* One region can have multiple VPC’s
* 2 VPC can’t talk to each other by default
* Per region we have 5 VPC’s by default
* It is like a virtual datacenter on cloud



**ELASTIC COMPUTE CLOUD (EC2)**

* In EC2 Service, we can launch EC2 instances
* EC2 is regional
* Service = Instance =EC Instance (VM)

**ELASTIC LOAD BALANCE:**

* ELB Distribute the traffic to multiple EC2 Instances across multiple AZ’s
* ELB is completely managed by AWS

(i.e HA, AS, Scalability, Performance etc)

* ELB is not a Server, it is a service for us
* We can’t login to ELB, but we can access ELB with DNS name
* ELB doesn’t have AZ’s,it is created at reginal level
* ELB is IAAS
* ELB has the IP address, these are dynamic
* AWS always recommended to use DNS name not ip address
* ELB has dynamic ip address, if you need static IP, contact AWS support center (but keep in mind we are compromising performance of ELB by getting static IP)

**TYPES OF LOAD BALANCER**

Application Load Balancer

* Latest Generation
* Default ELB
* We will use HTTP / HTTPS
* Best suitable for microservice

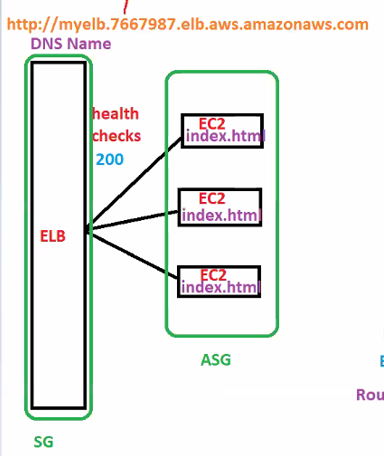
Network Load Balancer

* Latest Generation
* We will use for TCP
* Extreme high performance
* Will use static IP

Classic Load Balancer

* Previous Generation
* We will use HTTP, HTTPS and TCP

**New**Gateway Load Balancer



**ROUTING FEATURE**

* + Host based routing

Ex: https://www.suresh.com

* + Path based routing

Ex: https://www.suresh.com/admin

**ELASTIC BEANS STALK:**

* Easy and quick deployment of application in AWS

**Note:**

1. In general (other provider) in PAAS client don’t have control on the server
2. But, in AWS (i.e PAAS) Client have full control on the EC2 instances which is launched beanstalk
3. EBS handles EC2 instances (OS) behalf us
4. EBS is PAAS

**LIGHTSAIL:**

* If you want to setup and create a virtual private servers which already has everything installed

Example: wordpress, gitlab,node.js,joomla,drupal,Django

* No Auto-Scaling available

**LAMBDA:**

* Lambda is used for automation
* Lambda is Serverless
* Lambda is invoked based on the triggers

**SIMPLE STORAGE SERVER(S3)**

* In AWS, All services will start with SIMPLE and end with service.
* Ex: Simple Storage Service(S3)
* Simple Notification Service(SNS)
* S3 is unlimited storage from AWS
* S3 is used to store all FLAT file using S3 we can upload,download and access our files
* We can execute any files in S3
* Is it possible to install OS in S3? NO
* Is it possible to install DB in S3? NO
* Is it possible to run .net,java,.ext files ..etc in S3? NO
* S3 is serverless and AWS will handle HighAvailabilty,Performance,Scalability etc for S3
* S3 support Static Website Hosting (SWH)(HTML files)
  + Create bucket,upload all your files,enable satic website hosting
* **S3 is object based storage**
* S3 is GLOBAL and BUCKETS are reginal

**ELASTIC BLOCK STORAGE**

* **EBS is block based storage**
* HardDisk = Volume = EBS Volume
* Volume can attach and detached to instance
* We can attach multiple volumes to the EC2 instance
* EC2 has default volume and that is called ROOT VOLUME
* The ROOT Volume always has OS(Operating Syste)
* EC2 Support only Server-side OS not client site
* EC2 can have only 1 ROOT Volume
* EC3 can have multiple Additional Volumes
* Max size for EBS volume is 16TB per account
* We can’t attach same volume to multiple instances at the same time
* Volume can be pre-provisioned like (Ex 50GB,100GB..etc)
* Volume size can’t be decreased
* ROOT volume always mounted / attached as /dev/sda1
* Not possible to detach the ROOT Volume while EC2 Running status.
* Possible to detach the additional volume while EC2 Running status
* EC2 instance and volume should be same region
* We cant attache ap-south-1a(AZ) volume to EC2 instance different AZ(ap-south-1b)
* We cant attache ap-south-1a(AZ) volume to EC2 instance same AZ(ap-south-1a)

**ELASTIC FILE SYSTEM:**

* By using this service, we can attach single volume to multiple EC2 instance at the same time across multiple AZ’s (*i.e same REGION*)
* EFS is unlimited storage
* EFS will launch only LINUX machine and completely managed by AWS
* EFS is serverless for us
* EFS works with NFSv4 Protocol
* EFS is ***FILEBASE*** storage
* EFS is file sharing purpose, we can install any software but not recommended
* EFS doesn’t require any pre-provisioning (i.e it will automatically increase and shrink based on the data you put on EFS)
* Same EFS can be mounted(attached) multiple EC2 instance

Example: Pen drive, separate(device) mount point

ap-south-1a ap-south-1b ap-south-1c

EC2

EC2

EC2

root

root

root

**EFS(Elastic File System)**

* File share concept will use NFS protocol

**SNOW FAMILIY(Physical BOX)**

* *SnowFamily* is physical transfer and it is a service in AWS
* We can transfer data only S3
* *SnowFamily* is used to transfer HUGE data from on-premises to AWS and AWS to on-premises (vice versa)

Example: Like courier, Big physical server, Bigger HDD

Based on requirement we can order below device

* SnowCone 8TB
* SnowEdge 100TB
* SnowMobile PB(TRUCK)

**GLACIER:**

* If we are not using data frequently then we can move that data in to Glacier(archive/backup)
* Glacier is cheaper than S3

**STORAGE GATE WAY(SG)**

* It is used to synchronizing data from on-premises to AWS

SG

**ON-PREMISES**

2TB/8TB ..etc

**STORAGE GATE WAY**

**ROUTE53**

* Route is all about routing purpose
* Route53 is DNS service in AWS
* DNS port number is 53

EC2

ELB

EC2

R53

EC2

ASG

DNS Name

https://myE.352656.elb.aws.amazonaws.com

myhome.com

**DIRECT CONNECT**

* AWS will provide dedication connection from on-promises to AWS network(i.e between company and AWS)
* It is dedicated connection and more expensive (lease line connection)

**EDGE LOCATION:**

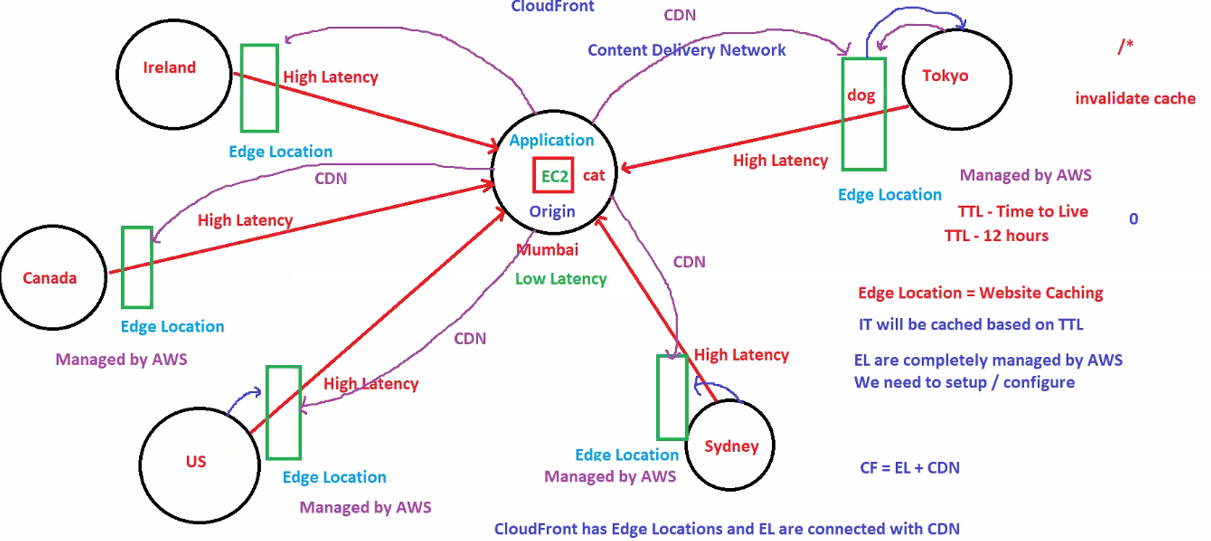
* Edge location is nothing but cache server/website Caching
* It will be cached based on TTL
* EL is completely managed by AWS, we need to setup/configure
* Each and every region has its own edge location and completely managed by AWS
* Inside CloudFront we will configure Edge location settings.
* While configure Edge location we will be configure **TTL(Time To Live)**
* We can break TTL value by using **invalidate cache**

**CONTENT DELIVERY NETWORK**

* It is AWS internal network
* Using this data will be copied into edge location
* Edge locations are connected to each other using CDN

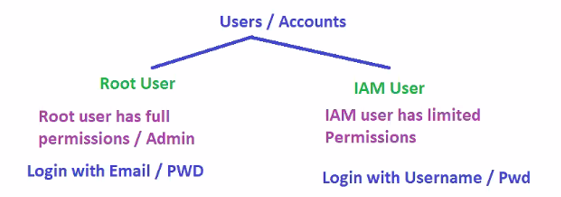
**CLOUDFRONT:**

* As name suggest, Cloud is Infront of you
* CloudFront has Edge locations and EL are connected with CDN
* CloudFront = Edge Location (EL)+ CDN

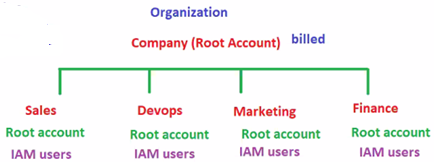


**IAM (IDENTITY AND ACCESS MANAGEMENT)**

* We can control the entire AWS account using IAM by giving proper permission to the IAM users
* The person who provides the card details he/she is the root user, the users how are using from that account they are called IAM users



* Organization is a service which handles multiple root accounts



**CLOUDWATCH**

* CloudWatch is used to monitor AWS services (EC2,ELB,RDS,ASG etc)
* CloudWatch Monitor performance
* CloudWatch monitors using alaram
* Two types of Monitoring
  + Basic Monitoring

We will get the data points every 5 mins

It is free service

Default monitoring always basic

* + Detailed Monitoring

We will get the data points every 1min,

It is chargeable

**CLOUDTRAIL**

* Monitor entire AWS environment

Records, monitor,track,audit,logs

**CONFIG**

* Monitors the changes in AWS resources

**METRICS**

* Metrics = CPU+Network+Memory

**AWS SUPPORT**

* Basic Support -- Free service
* Developer Support -- Cost
* Business Support -- Cost
* Enterprise Support -- High Costly