**\* \* \* AWS Session \* \* \***

**01-27-Aws-27-DEC-24**

------------------------------------------------------------------------------------------------------------------

1) What is IT infrastructure ?

2) Challenges with On-Prem infrastructure

3) What is Cloud Computing

4) Cloud Service Models

5) Cloud Providers

6) AWS Introduction

7) AWS Services Overview

**=============================**

**What is IT Infrastructure ?**

**=============================**

=> To run a software company then we need to purchase below resources

a) machines

b) network

c) power

d) storage

e) backup

f) security

=> The above resources are called as IT infrastructure.

=> We can maintain infrastructure in 2 ways

a) On-Prem infrastructure

b) Cloud infrastructure

**==================================**

**What is On-Prem Infrastructure ?**

**==================================**

=> On-Prem means we need to purchase and we need to maintain our resources to run our business.

=> We have several challenges with On-Prem infrastructure

1) Lot of money investment

2) Lot of man power

3) Scalability (increase / decrease)

4) Availability

5) Network issues

6) Security issues

=> To overcome the problems of on-prem infrastructure companies are preferring Cloud Infrastructure.

**===========================**

**What is Cloud Computing ?**

**===========================**

=> The process of delivering IT resources over the internet on demand basis is called as Cloud Computing.

=> We have below advantages with cloud computing.

- Pay as you go

- Less cost

- Scalability

- Availability

- Security

- Backup

Note: Cloud computing works based on pay as you go model.

**================**

**Cloud Providers**

**================**

=> The companies which are providing IT infrastructure based on "pay as you go" model are called as Cloud Providers.

1) Amazon (AWS)

2) Microsoft (Azure)

3) Google (GCP)

4) Salesforce

5) Ali Baba

6) Digital Ocean

**=======================**

**Cloud Service Models**

**=======================**

=> We have 3 types cloud service models

**1) IAAS**

**2) PAAS**

**3) SAAS**

**=============**

**What is IaaS**

**=============**

=> IAAS stands for infrastructure as a service

=> Provider will give infrastructure for us

Ex: Machines, Network, Storage

=> As a customer we need to prepare platform to run our application.

Ex: install required softwares + setup web servers + deploy application.

**=============**

**What is PaaS**

**=============**

=> PaaS stands for Platform as a service

=> Provider will give ready made platform to run our application directley.

=> As a customer we need to take care of only our application deployment.

**=================**

**What is SaaS**

**=================**

=> SaaS stands for software as a service

-> Cloud Provider will give their application to run our business.

Ex: zoom, google drive, dropbox, microsoft teams, jira ...

**===========**

**AWS Cloud**

**===========**

=> AWS stands for Amazon webservices.

=> AWS providing cloud services from 2006 onwards

=> AWS works based on Pay as you go model

=> 190+ countries using AWS cloud services to run their businesses

=> AWS having global infrastructure

34 - Regions

108 - Availability Zones

**===============**

**AWS Services**

**===============**

=> We have 200+ Services.....

1) EC2 : To Create Virtual Machines (Hourly Billing)

2) S3 : Unlimited storage

3) RDS : Relational Database service

4) EFS : Elastic File System (shared file system)

5) IAM : Identity & Access Management

6) VPC : Virtual Private Cloud

7) Elastic Beanstack : End to end web-application mgmt (PaaS)

8) Lambdas : Serverless computing

9) Route 53 : Domain Mapping (DNS)

10) ECS : Elastic Container Service

11) EKS : Elastic Kubernetes Service

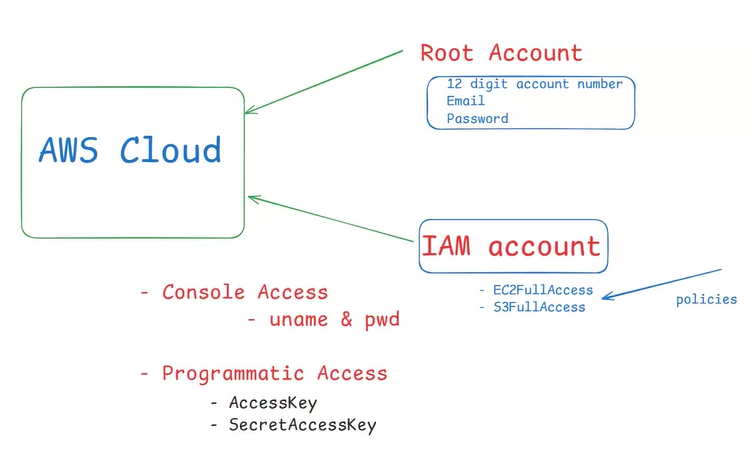
12) Cloud Watch : Monitoring

13) SNS : Simple Notification Service .....

**=====**

**IAM 02-27-Aws-30-DEC-24**

**=====**

****

**\* Identity and Access management**

=> It is used to manage users, groups, policies and roles

=> IAM is a free service

=> In AWS cloud platform we will have 2 types of accounts

1) Root Account

2) IAM Account

Note: When we signup in aws website then by default it will consider that as root account.

=> Root account is very powerfull account with no restrictions.

=> If we login with Root user credentials, we can access everything in AWS cloud.

Note-1 : We shouldn't use root account for day to day activities.

Note-2 : We shouldn't share root account credentials with anyone.

Note-3: Company will not provide root account credentials for team members.

Note-4: It is recommended to enable MFA for root account.

MFA : Multi Factor Authentication

**===================================**

**Multi Factor Authentication (MFA)**

**===================================**

-> It is used to provide additional security for root account.

-> Enable MFA for root account using Google Authenticator app.

-> After enabling MFA, logout and login into root account and check behaviour.

**============**

**IAM Account**

**============**

=> For team members IAM accounts will be created with limited access

=> For daily activities in aws cloud we should use IAM account only

=> For IAM user we can provide below types of accesses

1) Console Access (web login)

- uname & pwd

2) Programmatic Access

- AccessKey and SecretAccessKey

**Note: To communicate with AWS cloud using terraform then IAM user should have programmatic access**.

1) Create IAM account and attach policies (RDSFullAcces, S3FullAccess)

2) Login into IAM account and check EC2 service (can't access because no permission)

**=================**

**IAM User Group**

**=================**

**Note - Policies are only applicable for users and group level only.**

=> When we want to provide some permissions for multiple users then we can create IAM user group and we can add users to that group and we can attach policies to group.

**1) Create User Group**

**2) Attach Policies to group**

**3) Add Users to group**

**=========**

**IAM Role**

**=========**

**=> IAM role nothing but set of permissions**

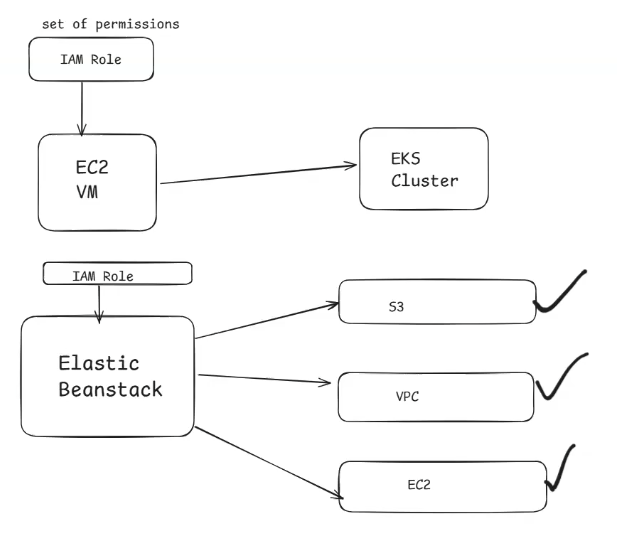
Ex-1: EC2 VM wants to create EKS cluster, Then EC2 VM should have IAM Role with EKS permissions.

1. create IAM role with all EKS policies

2) Attach IAM role to ec2 vm

**Note – AWS policies will be saved in the JSON format.**

**If one service want to communicate with the another service then IAM role comes into the picture.**

****

**============**

**IAM Summary**

**============**

1) What is IAM

2) What is Root Account

3) How to enable MFA

4) What is IAM account

5) Console Access Vs Programmatic Access

6) Users Creation

7) User Groups

8) Policies / Permissions

9) Roles

10) Working with Custom Policies

**=========**

**AWS RDS 03-27-Aws-02-JAN-25**

**=========**

1) What is Database

2) Why we need database

3) On-Prem Database & Challenges

4) What is RDS & Why

5) RDS Setup

6) Connecting with RDS DB Server using MySQL Workbench

**===================**

**What is Database ?**

**===================**

Database : It is a software which is used to store the data permanently.

Whatsapp Application -------------> Database

- Text msgs

- Audio files

- Video files

- Documents

=> Every s/w application will use database to store the data permanently.

=> Software applications will use SQL to communicate with databases.

SQL

Application --------------> Database

=> Using SQL we can perform CRUD operations in the database.

**C -> Create**

**R -> Retrieve**

**U -> Update**

**D -> Delete**

=> We have several databases in the market

- Oracle

- MySQL

- SQLServer

- PostGres ....

=> The above databases are called as Relational Databases.

=> Relational databases will store the data using tables.

=> Table represents data using Rows and columns.

**=================**

**Database Setup**

**=================**

=> We can setup database in 2 ways

1) On-Prem Database

2) Cloud Database

**==================================**

**Challenges with On-Prem Database**

**==================================**

You Download and Install database on your own.

1) Purchase DB server license

2) Install DB Server s/w

3) Security

4) Network

5) Availability

6) Scalability

7) Backup & Restore

8) Adminstration (DBA)

* To overcome above challenges it is highly recommended to use Cloud Databases.
* If we use cloud database then cloud provider will manage database server for us.
* **RDS is a service which is used to manage relational databases.**
* AWS RDS service providing cloud databases.
* **RDS stands for relational database service in AWS cloud.**
* RDS is used to create & manage relational databases.
* **RDS is a fully managed service in AWS cloud.**
* RDS works based on "pay as you go" model.

**=============**

**RDS Lab Task**

**=============**

**Step-1 :** Create MySQL DB Server using RDS

**Step-2 :** Enable "**MySQL : 3306**" port number in Security Group Inbound Rules

**Step-3 :** Test MySQL DB Connection using "**MySQL Workbench software (client s/w)**"

**Step-4 :** Execute some SQL queries for practice (optional)

**Step-5 :** Delete RDS instance to avoid billing.

**============================**

**MySQL DB Creation Steps**

**============================**

Creation method : Standard Create

Engine Type : MySQL

Templates: Free Tier

DB instance identifier : ashokit-db-instance

public access : Yes

Credentials : Self Managed

Additional Configuarations : Initial DB name : ashokitdb

**====================**

**Database Details**

**====================**

DB Endpoint - database-1.cnuuke2i0lav.eu-north-1.rds.amazonaws.com

DB username - admin

DB password - Masterpassword

DB port - 3306

Note : Using above details we can check database connectivity.

=> Once connectivity is successful then we will share database details with developmen team.

In realtime public access will be NO

**============================**

**SQL Queries For Practice**

**============================**

=> Execute below sql queries using workbench

show databases;

use ashokitdb;

show tables;

## table creation query

create table emp(

eid int(10),

ename varchar(100),

esal int(10)

);

## retrieve records query

select \* from emp;

## insert query

insert into emp values(1, 'john', 1000);

insert into emp values(2, 'smith', 2000);

## retrieve records query

select \* from emp;

===========

Assigment

===========

1. Connect with RDS DB Server using EC2 Linux VM and execute above sql queries.

* Create one RDS database and ec2 instance
* Connect RDs instance to the ec2 using command
* Command - mysql -h <rds-endpoint> -u <username> -p
* Then enter password and your database is connected.

**=================================**

**AWS S3 (Simple Storage Service) 04-27-Aws-03-JAN-25**

**=================================**

-> S3 is a storage service in AWS cloud.

-> S3 supports unlimited storage.

-> Using S3 we can store any amount of data from anywhere at anypoint of time.

-> S3 supports object based storage (files).

One Object = One file

**Note: We can store any type of file in s3**

**Ex:** .txt, .mp4, .png, .jpeg, .doc, .pdf, .xls ...

-> In S3, we need to create buckets to store objects (files)

**Note: In one bucket we can store group of objects.**

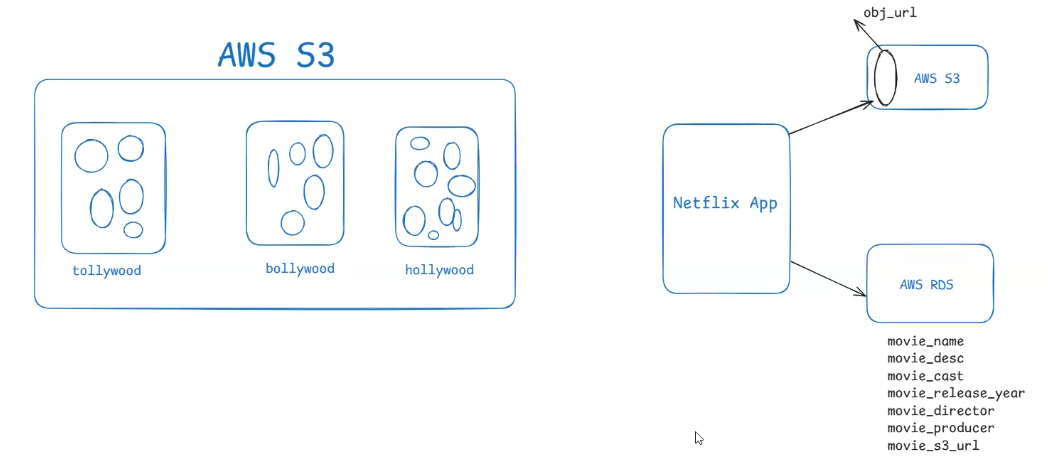
**Note: Every bucket should have unique name.**

-> When we create a bucket, end-point url will be generated to access bucket.

-> When we upload object into bucket, every object will get its own end-point url.

Note: By default, buckets and objects are private (we can them as public).

**In RDS we can store only txt format data. Ex. Urls**

****

**Where we will use s3 storage ?**

**Ans –** If our project is dealing with the some files , binary files (audio , video ,images) that is not recommended to store in the database , so that’s why file based storage is required for that s3 service comes into the picture

**Note - Our job is only to create a database and the buckets**

**=================================**

**Static Website Hosting using S3 05-27-Aws-06-JAN-25**

**=================================**

-> Website nothing but collection of web pages.

Ex: login page, register page, services, about us page, contact us page...

-> Websites are divided into 2 types

**1) static website**

**2) dynamic website**

-> The website which gives same response/content for all users is called as **static website.**

-> The website which gives response based on user is called as **Dynamic website.**

**=> Static websites we can deploy in 2 ways**

1) Create EC2 Linux VM and Install HTTPD Webserver and Deploy

2) Deploy using S3 bucket

Step-1: Create s3 bucket with unique name

Step-2: Upload website files & folders into bucket with public read-access

Step-3: Enable Static website hosting (in bucket properties)

index-document : index.html

error-document : error.html

Note: After enabling static website hosting it generates end-point URL for our website.

Step-4: Access our website using website endpoint url.

**===================**

**S3 Storage Classes**

**===================**

=> Storage classes are used to specify how frequently we want to access our objects from S3.

=> At the time of uploading object into S3 bucket we can select storage class for that object.

=> We have several storage classes in s3 like below..

1) Standard (default) : To access object more than once in a month

2) Intelligent - Tiering : Unknown access patterns

3) Standard-IA : Infrequent Accesed Data (Only once in month)

4) One Zone-IA : Stored in single availability zone (once in a month)

5) Glacier Instant Retrieval : Long Live Archieve Data

(once in quarter -> Milli Seconds)

6) Glacier Flexible Retrieval : Once in a year (Minutes to Hours)

7) Glacier Deep Archieve : Less Than once in a year (Hours to download)

8) Reduced Redundency : Not Cost Effective (Not recommended)

**============**

**Versioning**

**============**

=> It is used to maintain multiple variants of same file.

=> By default versioning will be disabled for S3 bucket.

=> As versioning is disabled, when we upload file again it will override old file.

=> If we don't want to replace old objects from bucket then we can enable Versioning.

=> Versioning we will enable at bucket level and it is applicable at object level

**================**

**Object Locking**

**================**

-> It is used to enable the feature WORM (Write once read many times) model.

-> We can enable object lock on **versioning enabled** buckets.

-> Object Lock will be enabled at bucket level and it is applicable at object level.

- created object cant be deleted from other users. They only can read .

Note: If we enable object locking then versioning will be enabled by default.

**=================================**

**What is Transfer Accelaration**

**=================================**

=> It is used to speed up data transfer process in s3 bucket.

=> When we enable Transfer Accelaration it provides endpoint url to upload the data to s3 bucket quickly.

Note: If we enable Transfer Accelaration bill will be generated.

Janabhoomi express (6 hrs, 300 Rs)

Hyd -----------------------------------------------------------> Vijaywada

Vande Bharat Express (3 hrs, 1000 Rs)

Hyd -----------------------------------------------------------> Vijaywada

**===============**

**AWS S3 Limits**

**===============**

**\* Interview Question - how many buckets we can create in an s3 ?**

**ANS -**

=> By default we can create upto 100 s3 buckets per aws account. However we can request AWS support team to increase the limit.

**\* Interview Question -** **what is the size of the object we can upload ?**

**Ans-**

=> Individual object size can be upto 5TB. For uploading large object we can use multipart upload.

For uploading large object we can use multipart upload

Note: In one bucket we can upload unlimited objects. ( Files )

**=========================**

**Realtime Usecases of S3**

**=========================**

1) Application files (images, audios, videos, docs...)

2) Database backup files ( Snapshots )

3) EBS volumes snapshots

4) Server log files

**===========**

**Summary**

**===========**

1) What is AWS S3 & Why ?

2) S3 Buckets & Objects

3) Static website Hosting using S3

4) S3 Storage Classes

5) Versioning

6) Bucket Locking

7) Transfer Accelaration

8) S3 Limits

9) Real-Time usecases

**========**

**AWS EC2 06-27-Aws-07-JAN-25**

**========**

=> Elastic Compute Cloud

=> It is Most demanded service in AWS

=> It is Used to create Virtual Machines in AWS cloud

=> EC2 VM is called as EC2 instance

EC2 Instance = Computer / Server / VM / Virtual Machine / V Box

=> EC2 instance is re-sizable (we can change configuration based on demand)

=> EC2 is a paid service (hourly billing)

=> EC2 VM Minimum billing period is 1 hour

9:00 AM to 9:15 AM => 15 mins => 1 hour billing

9:30 AM to 9:45 AM => 15 mins => 1 hour billing

Note: To encourage beginners, AWS provided t2.micro/t3.micro for 1 year free

(monthly 750 hours)

EC2 has storage with EBS service .

* **EBS stands for Elastic Block Store**

**Note: To encourage beginners, AWS provided t2.micro/t3.micro for 1 year free (monthly 750 hours)**

=> EC2 VM will have storage with EBS service

=> EBS stands for elastic block store.

**Ec2 VM with Windows OS : 30 GB (default)**

**EC2 VM with Linux OS : 8 GB (default)**

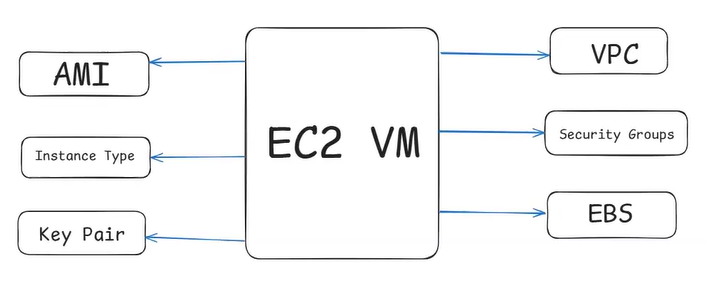
**EBS max capacity : 16 TB (16000 GB)**

**===================**

**Creating EC2 VM**

**===================**

=> To create EC2 instance we will use below components

****

**1.AMI**

* Ami stands for **Amazon Machine Image.**
* AMI represents operating system like **Windows, Linux, MAC etc...**

**2.Instance Type**

* Instance Type represents machine configuration ( Memory,RAM)
* Ex : t2.micro, t3.micro, t2.medium, t2.large etc....

**3. Keypair**

* Keypair is used to connect with EC2 VM securely
* **Ex: .pem file**

**4.VPC**

* VPC provides network required for the EC2 vm

**5.Security Group**

Outgoing

Incoming

* Security Group represents inbound and outbound rules to handle traffic.(Request)

**6. EBS (Elastic Block Store ) Storage / SSD**

* EBS Volume represents storage required for EC2 instance.

Note-1 : One Key Pair we can use for multiple EC2 instances

Note-2 : One Security Group (SG) we can use for multiple EC2 instances

**===============================**

**Types of IP's in AWS Cloud**

**===============================**

=> We have 3 types of IP's in AWS cloud

**1) private ip**

* Private IP is a fixed IP in AWS. It is used for internal communication (With in VPC).
* Even if we restart the EC2 instance, private IP will not change.

**2) public ip**

* public ip is a dynamic IP in aws. It is used to connect with EC2 VM from outside.
* When we restart our EC2 instance then public ip gets changed.
* If we want fixed public ip then we need to use Elastic IP.

**3) Elastic IP**

* Elastic IPs are commercial (bill will be generated).

**================================================================**

**Assignment : Create EC2 VM with Windows AMI and connect with that using RDP client.**

**================================================================**

Create 1 instance and select windows ami and create.

Click on connect 🡪 RDS Client 🡪 Get password🡪upload pem file🡪decrypt password.

Search in windows🡪Remote desktop connection🡪show option🡪enter url🡪username=Adminnistrator 🡪save and connect.

We can connect physical machine to the windows virtual machine using RDP protocall.

Remote Desktop Protocall. It is running on port 3389.

**=================================**

**What is user-data in EC2 VM ? 07-27-Aws-08-JAN-25**

**=================================**

- It is used to execute script while launching ec2 machine.

- User data will execute only once

- Create EC2 VM (amazon linux) with below user data

**Que – Why we use user data in the EC2 VM ?**

**Ans-** We will use user data at the creation of the machine for installing some application .

It is time saving .

#! /bin/bash

sudo su

yum install httpd -y

cd /var/www/html

echo "<html><h1>Life Insurance Server - 1</h1></html>" > index.html

service httpd start

**========================**

**What is Load Balancer ?**

**========================**

* When we run our application in single server then we have to face below challenges

1) One server should handle all request

2) Burden will increase on server

3) Response will be delayed for clients

4) Server can crash

5) Single Point Of Failure

* To avoid above problems, we will run our application in multiple servers.
* We will use Load Balancer to distribute load to all servers in the round robbin fashion.
* We have below advantages with Load Balancer

1) App will run in multiple severs

2) Load will be distributed

3) Burden will be reduces on servers

4) Fast Performance

5) High Availability

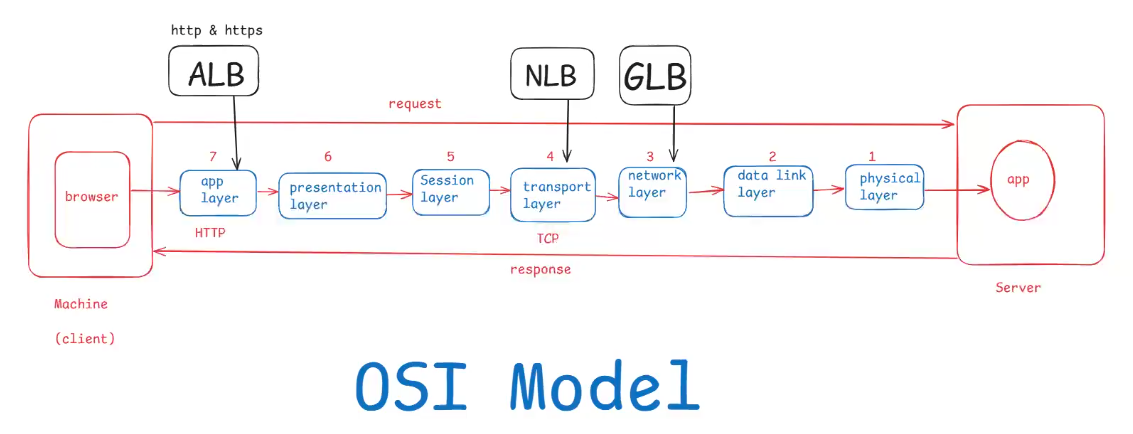
* In AWS we have 4 types of Load Balancers

1) Application Load Balancer (ALB)

2) Network Load Balancer (NLB)

3) Gateway Load Balancer (GLB)

4) Classic Load Balancer (previous generation) Not used



---------------------------------------------------------------------------------------------------------------

**Application Load Balancer (ALB)**

•ALB work at the Layer 7 (**Application Layer**) of the OSI model.

• Designed to route HTTP and HTTPS traffic based on content.

• Supports advanced routing features such as host-based routing, path-based routing, and routing based on HTTP headers.

• Ideal for modern web applications, microservices, and container-based applications.

--------------------------------------------------------------------------------------------------------------

**Network Load Balancer(NLB)**

• Works at Layer 4 (Transport Layer) of the OSI model.

Designed to handle high volumes of TCP, TLS, and UDP traffic.

• Provides ultra-high performance, low-latency load balancing. Suitable for applications requiring high scalability and low-latency communication, such as gaming applications, loT applications, and services requiring high-throughput TCP/UDP traffic.

--------------------------------------------------------------------------------------------------------------

**Gateway Load Balancer**

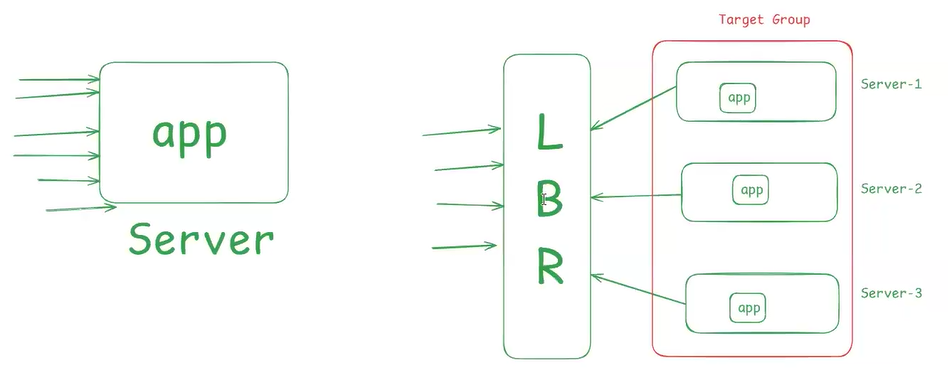
Gateway Load Balancer will work at the network layer.

* It will work at third party applications.

**=========================**

**Load Balancer Lab Task**

**=========================**

****

**Target Group = Set of servers.**

Step-1) Create EC2 VM-1 with below user data

#! /bin/bash

sudo su

yum install httpd -y

cd /var/www/html

echo "<html><h1>Life Insurance Server - 1</h1></html>" > index.html

service httpd start

Step-2) Create EC2 VM-2 with below user data

#! /bin/bash

sudo su

yum install httpd -y

cd /var/www/html

echo "<html><h1>Life Insurance Server - 2</h1></html>" > index.html

service httpd start

Step-3) Add these 2 instances to one "Target Group"

* Click on create target group 🡪 instances🡪Give name for target group🡪
* Select instances🡪include as pending below🡪create target group🡪register targets

Step-4) Create Load Balancer with Target Group (ALB)

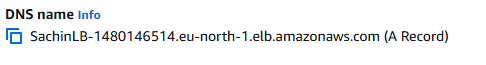
* Create load balancer🡪application load balancer🡪give load balancer name
* Select availability zones 🡪create load balancer.

**Listeners and Routing**

* **We request comes to the load balancer it should forward request to the target groups.**

Step-5) Access Load Balancer DNS in browser

* Before hitting url check security inbound rule and enable HTTP port 80.



**After Use Delete All target group ,load balancer , instances.**

**============**

**Assignment**

**============**

##### Microservices Load Balancing with multiple Target Groups : https://www.youtube.com/watch?v=QvEJ8--zneU ####

**===============**

**Auto Scaling**

**===============**

* **Increase or Decrease our resources**

=> It is used to adjust the capacity required to handle the load of our application.

Ex: For big billon day sale we can't guess traffic

=> If requests are increasing then servers should be increased and if requests are decreasing then servers should be reduced.

=> If we use Auto scaling then it will increase or decrease the no.of servers based on incoming traffic.

=> We have below advantages with Auto Scaling

**1) Cost Management**

**2) High Availability**

* If one server is damaged then it will replace with the another server.
* 24/7 server are available.

**3) Fault Tolerance**

* If any server is down / there is any problem with the server then that will replace with that server.

**Note – EKS cluster internally working with the auto scaling.**

**====================**

**EC2 Instance Types**

**====================**

=> Amazon EC2 (Elastic Compute Cloud) offers a variety of instance types to suit different use cases.

1) General Purpose

- T series : t2, t3, t4...

- M series : m6g, m5, m5a, m5n...

2) Compute Optimized

- C series : c7gm c6i, c6g, c5, c5a

3) Memory Optimized

- R series : r6g, r5, r5a, r5n

- X series : x2gd, x1e

4) Storage Optimized

- I series : i4i, i3, i3en

- D series : d2

5) Accelarated Computing

- P series : p4, p3, p2

- G series : g5, g4ad, g4dn

- inf series : inf1

6) High Performance Computing

- H series : hpc6id

**==============**

**What is EBS**

**==============**

* EBS stands for Elastic Block Store
* It is block level storage device (Hard Disc / SSD)
* When we create ec2 instance then "EBS Volume" gets created automatically.

**Note: If we remove EBS volume from EC2 instance then we can't start/use that EC2 instance.**

------------------------------------------------------------------------------------------------------------------

* In EBS we have 2 types of volumes (storages)

**1) Root Volume**

**2) Additional Volume**

**Note: When we launch EC2 instance by default we will get one Root volume.**

* Root volume is mandatory to launch EC2 instance.
* Additional EBS volumes are optional devices (we can add/remove)
* For windows VM, we will get 30 GB as default volume size.
* For Linux VM, we will get 8 GB as default volume size.

**Note: EBS volume can have upto 16 TB (16000 GB)**

* One EC2 VM can have multiple EBS volumes.
* One EBS volume can be attached to one EC2 VM at a time.
* EBS volumes are Availability Zone Specific

Mumbai : ap-south-1

ap-south-1a

ap-south-1b

ap-south-1c

Note: In which availability zone our EC2 VM got created, in same availability zone we have to create EBS volumes for attaching.

**==========**

**Snapshots**

**==========**

=> Snapshots are used for volumes backup

=> snapshots are region specific

=> From Volume we can create snapshot and from snapshot we can create volume

volume =====> snapshot ========> volume

=> Snapshot can't be attached to EC2 instance directley

(volumes can be attached to ec2 instances directley).

=========================================================

Q) How to copy the data from 1a zone VM to 1b zone VM ?

=========================================================

Step-1 : Create snapshot for 1a VM volume

Step-2 : From snapshot create volume in 1b zone

Step-3 : Attached newly created 1b volume to 1b VM.

=================

EBS Volume Types

=================

=> We have 5 types of EBS volumes

1) General Purpose Volume (Min: 1 GiB, Max: 16384 GiB)

2) Provisioned IOPS (io1) (Min: 4 GiB, Max: 16384 GiB)

3) Cold HDD (Min: 125 GiB, Max: 16384 GiB)

4) Throuput Optimized (Min: 125 GiB, Max: 16384 GiB)

5) Magnetic (standard) (Min: 1 GiB, Max: 1024 GiB)

===========================================

1) What is EC2 & Why

2) What is AMI

3) What is Key Pair

4) What is Security Group

5) Launch Windows VM and connect with RDP client

6) Launch Linux VM and connect with SSH Client

7) What is User Data

8) Types of IPs

9) Launching Static website using HTTPD webserver

10) EBS Volumes & Snaphosts

11) Load Balancers

12) OSI Model

13) Auto Scaling Group