Wednesday

PAGE EDGE

DATE : 6/8/25

AWS

* Software : Dev, Devices, Server, Laptop, Desktop, Storage

* Resources :

1. Human Resource
2. IT Resource

1. Human Resource - The people who are involving in development of software

   Ex: Developer, Test Engineer

2. IT Resources - There are two types:

   1. Hardware Resources
   2. Software Resources

1. Hardware Resources :

* Physical Components

a. Computer (Laptops, Desktop, Services)

   memory - RAM (4GB, 8GB, 16GB)

   processor - CPU (intel i, i5, i6, i7)

   storage - SSD (Solid State Drive) - Faster

   HDD (Hard Disk Drive) - Slower

b. Storage Devices :

* SSD * HDD

c. Networking Equipments :

   Router, Switches, Cables

d. Data Centre

i. Cluster of Server, ii. House of Server

2. Software Resources:

* Software: It is a set of instruction to perform some/any task.

Types of software:

1. System Software

* It is a software which is use to manage system hardware and application software.

Ex: OS like Windows OS, Mac OS, Linux OS etc.

2. Application Software:

* It is a software which is use to perform specific task.

Ex: Calculator, Adobe, Paint... etc

Problem with Traditional IT Approach:

1. Initial investment is high or high capital.
2. Hardware maintenance is difficult

Because 24/7 → Power Supply
→ Temperature (AC)
→ Team

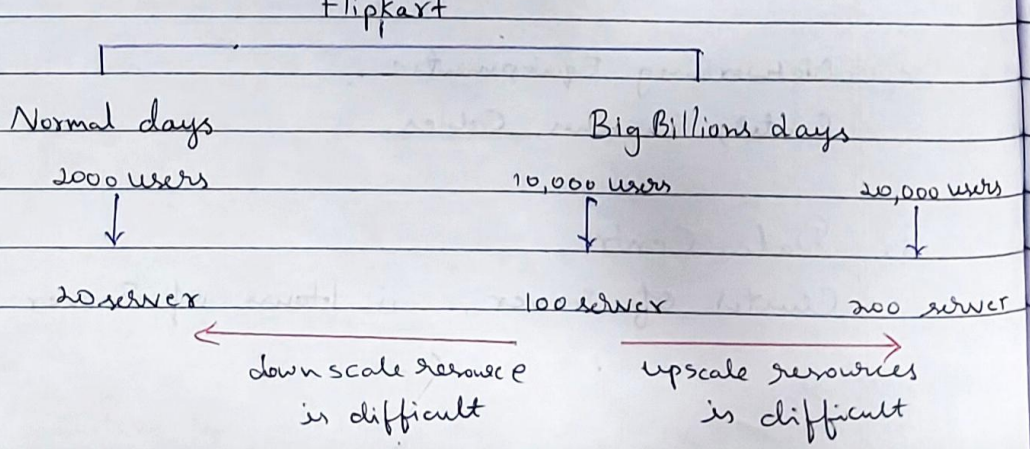
3. Less Scalable

Note: Scalable means increase or decrease in the resources.

* Increasing resources means upscale
* Decreasing resources means downscale

So upscale & downscale the resource is difficult immediately

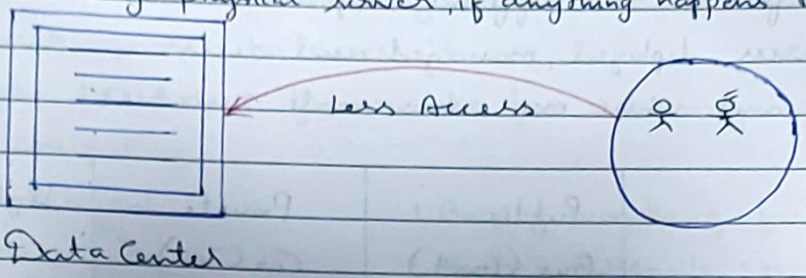
Ex:



4. Less Accessible:

* If your having physical server, if anything happens to the server.

Ex:



5. Less Security

* In On-premises data center, if server gets destroyed all the data will be lost
* As its physical data center someone can easily hack it.

Cloud Computing:

* Cloud is anything but over the internet
* Computing means IT resources (H/w, S/w Resources)

What is Cloud Computing?

* Cloud Computing is a technology to overcome problem with traditional IT approach were one can access IT resources over the internet.

Type of Cloud Computing:

Cloud Deployment Models	Cloud Services Models
Private Cloud	IaaS
Public Cloud	PaaS
Hybrid Cloud	SaaS

1. Cloud Deployment Model:

* It refers to the different ways in which cloud services and resources deployed, managed and access
* How and where one can create resources.

	Public Bus (Govt)	Private Car (You)	Hybrid Cab/Taxi (Someone)
Petrol	x	✓	x
Customization	x	✓	x
Maintenance	x	✓	x
Privacy	x	✓	✓/x
Restrictions	x	✓	✓/x
Road	x	✓	✓/x

a. Public Cloud:

* It is open for all the general public people to create resources
* It is provided by third party
  Ex: AWS, Azure, IBM etc

b. Private Cloud:

* It is open for only particular organisation or on-premises.
* It is provided by third party or it might be on-premises.
  Ex: AWS, Own org etc

c. Hybrid Cloud:

* It is the combination of public and private cloud.
* Creating resources in both public and private cloud.

DATE: 7/8/25

2. Cloud Service Model:

* How cloud provider are going to provide you the services and how your going to access it.

- It refers to various frameworks for delivering services in cloud computing.

Ex: Biriyani

Rice, veg — Cook

Salt, water

Utensils

Store

1. Buying everything for cooking
2. Renting - Utensils, Store } Cooking
   Ingrid - Buying
3. Renting - Utensils, Store } Cooking
   Ingrid - Pay for usage only
4. Directly consume it from the Hotel

Note: If want to develop an applications and to deploy it, the required resources are:

Traditional Approach:

1. Application
2. Data
3. Runtime
4. OS
5. Virtualization
6. Server
7. Storage
8. Networking

1. Infrastructures - as - a - Service (IaaS)

* In IaaS the cloud provider are taking care of hlw resource. and slw resources are taken care by users

User	Cloud Provider
1. Application	5. Virtualization
2. Data	6. Server
3. Runtime	7. Storage
4. OS	8. Networking

2. Platform-as-a-Service (PaaS)

* So, here cloud provider are going to take care of all hardware resources and some s/w resources

User

1. Application
2. Data

Cloud Provider

3. Runtime
4. OS
5. Virtualization
6. Server
7. Storage
8. Networking

3. Software-as-a-Service (SaaS)

* So all the resources are taken care by the cloud provider itself.

Cloud Provider

1. Application
2. Data
3. Runtime
4. OS
5. Virtualization
6. Server
7. Storage
8. Networking

Principles of Cloud Computing:

1. It follows on-demand-self-service

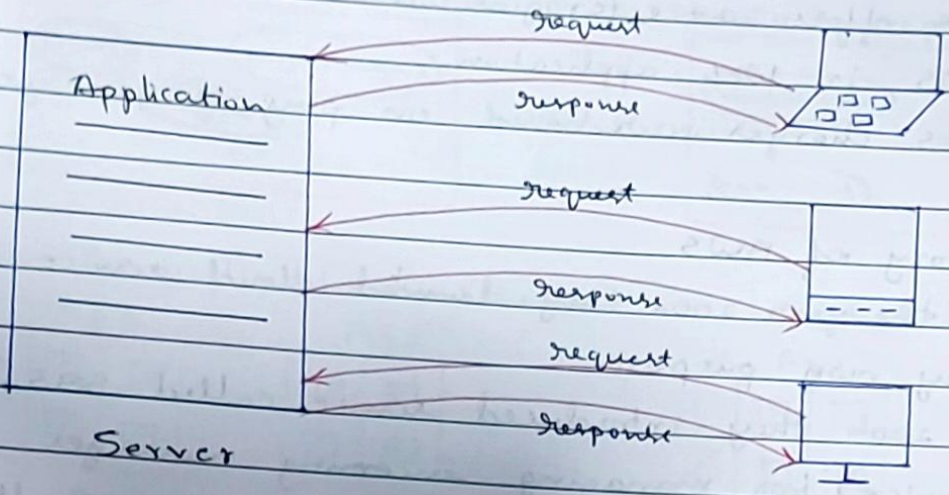
2. It follows pay-as-you-go model

* Pay for the resources you use

3. It follows Client-Server Model.

DATE : / /

* Client/User accessing the application over the internet is called Client-Server Model.



Advantages of Cloud Computing:

1. No initial investment
2. No need to maintain hardware components
3. High Scalable - It follows on-demand-self-service
4. High Accessible - It follows Client-Server Model
5. High Security
6. Cost-efficient - It follows pay-as-you-go

Top Cloud Providers:

1. AWS - Amazon Web Services
2. Microsoft Azure
3. GCP - Google Cloud Platform
4. Oracle Cloud
5. IBM

Friday

AWS

3/8/25

* AWS is a cloud service provider were one can create or one can get IT resources (h/w components / s/w components) over the internet
* AWS offers 20+ categories & 200+ services
* AWS is Web application
* AWS charges are based on pay-as-you-go

History of AWS

* In the year 2002 they launched cloud service initially for they own purpose
* In 2004 they introduced service called SQS (Simple Queue Service) for managing incoming messages
* In the year 2006 AWS was launched officially for public with the service S3 (Simple Storage Service), EC2 (Elastic Compute Cloud), SQS and Simple DB.

Features of AWS:

1. Cost efficient
2. High Scalable
3. High Accesible
4. High Security
5. High reliable

AWS Global Infrastructure:

* They designed AWS Global Infrastructure to provide they service with low latency, high accessibility, high scalable, high security and high reliable to the end user

Key Components

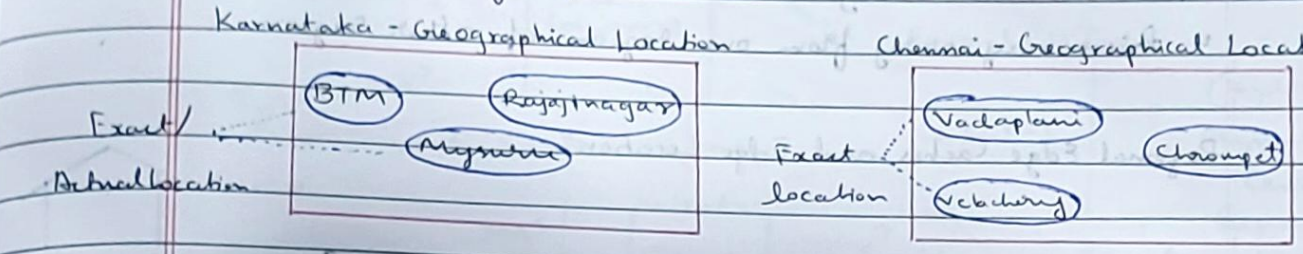
1. Regions
2. Availability zones
3. Edge Locations
4. Regional Edge Caches

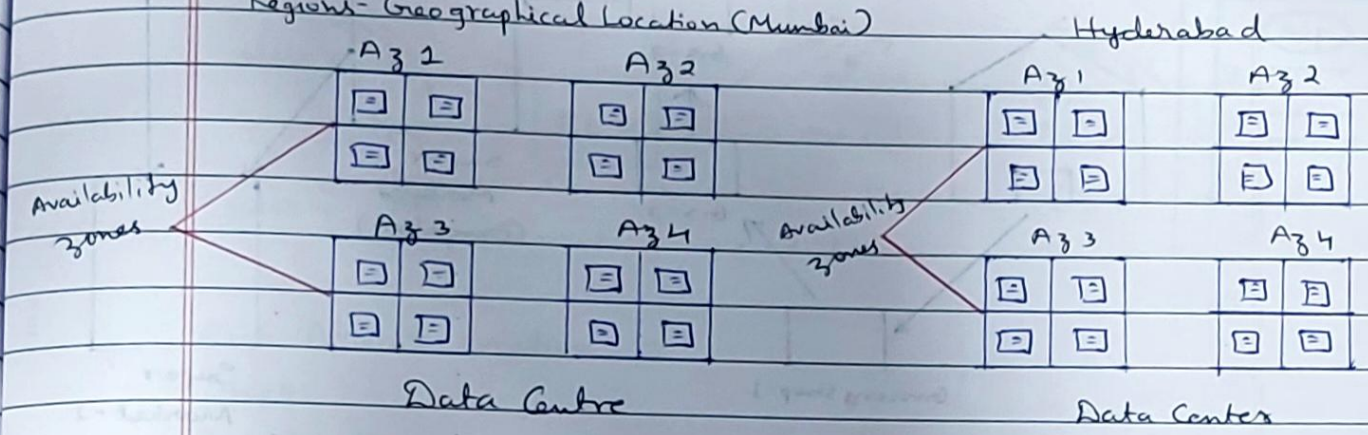
8/25

DATE: / /

1. The Regions and Availability Zone:

Ex: Branches of Ispider





Regions:

* Regions are nothing but geographical location
* Group of availability zones are called as regions
* AWS offers 30+ regions

Ex: In India : Mumbai and Hyderabad

Availability zones:

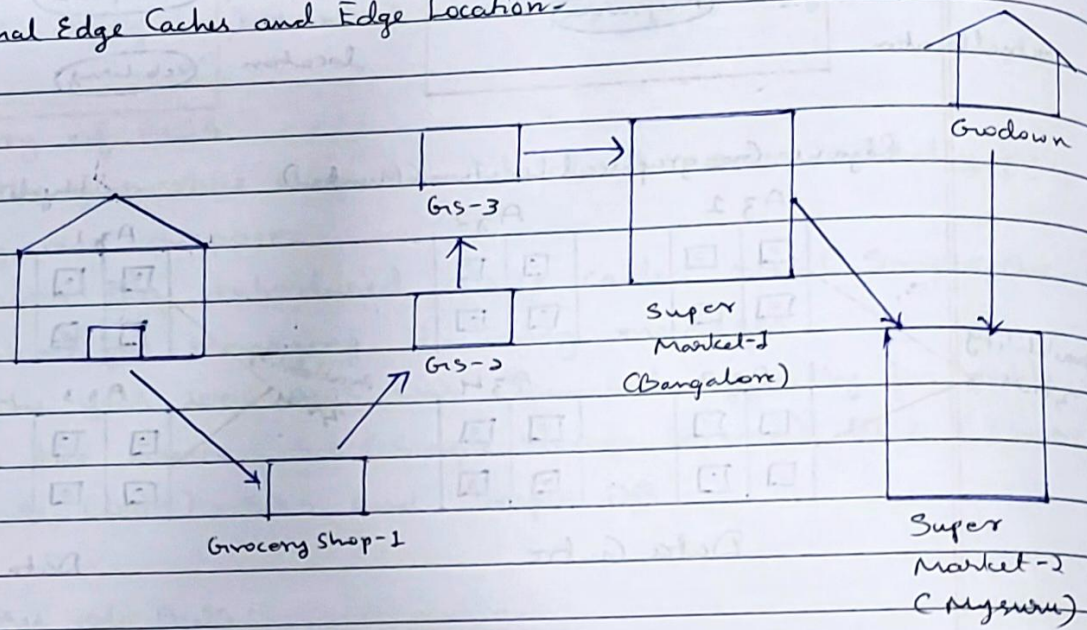
* It is the exact location of physical data centre
* Each regions will have minimum of two availability zones & maximum of six availability zones
* AWS offers 100+ availability zones

Note: How to select regions:

1. Low latency

2. Legal requirement and Data Governance (laws)
3. Service availability
4. Prices may vary from one region of other region.

Regional Edge Caches and Edge Location:

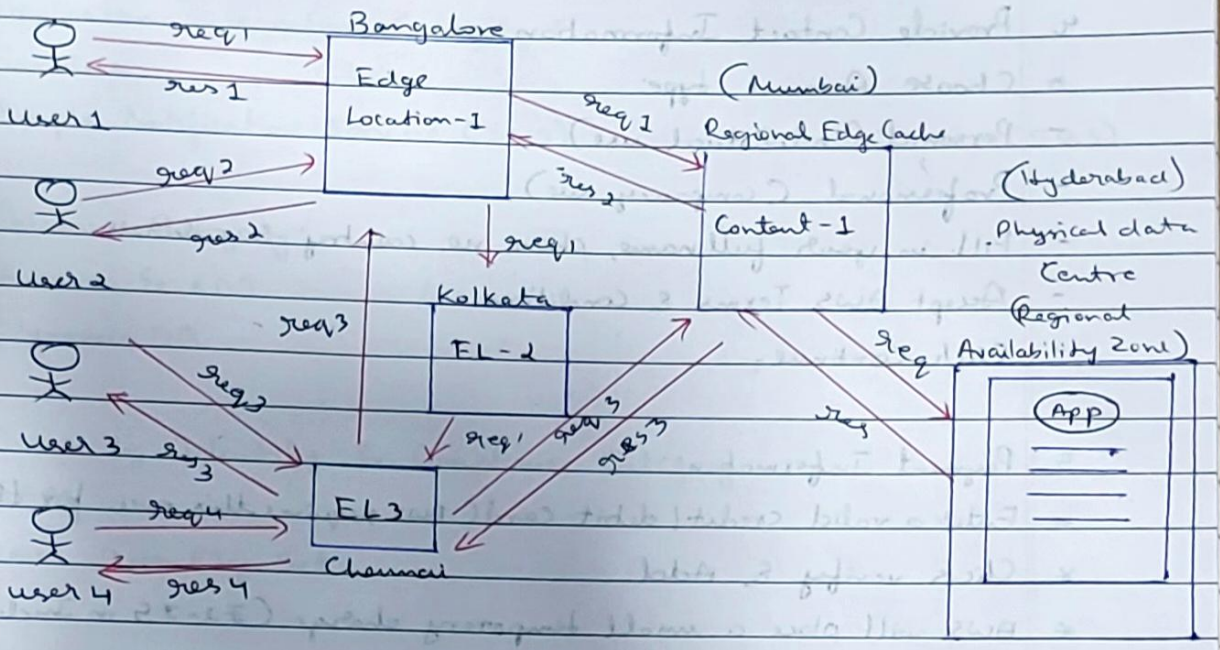


Edge Location:

* It is also a small data centers to deliver the content of the application with low latency
* So it will only caches the data which is accessed frequently.
* Edge location are also called as point of present there are present out of regions & availability zone.
* Location: Bangalore, Mumbai, Kolkata etc...
* There are 600+ edge location

Regional Edge Caches:

* So, it is also a big data center which will caches the content of the application.
* It store both frequently & infrequently accessed content
* There are only 10+ regional edge caches Ex: Mumbai



AWS Account Creation:

Steps:

1. Go to the AWS Website

* Open a browser & visit : https://aws.amazon.com
  Click "Create an AWS Account" (top-right corner)

2. Enter Account Information

* You'll see the sign-up page : Fill in.
* Root email address (use a personal or work email, not already used for AWS)
* AWS account name (this will be your accounts display name).
* Password (strong one)
* click continue

3. Verify email:

* AWS sends a verification code to your email.
* Check your inbox, enter the code and proceed.

4. Provide Contact Information:

* Choose Account type:
- Personal (individual use)
- Professional (company use)
- Fill in your full name, phone no. country & address.
- Accept AWS Terms & Condition.
- Click Continue.

5. Payment Information:

* Enter a valid credit/debit card (AWS requires this even for the free tier)
* Click verify & Add.
* AWS will place a small temporary charge (₹2-₹5 in India) to verify card.

6. Identity Verification:

* Choose Text message (SMS) or voice call.
* Enter your phone no. & verification code sent by AWS.

7. Choose a Support Plan

* Basic Plane (Free) → Recommended for most new users.
* You can upgrade later if needed.

8. Sign In to AWS Management Console

* Go to https://aws.amazon.com/console/
* Login as Root user using your registered email & password.
* You now have an active AWS account.

AWS Categories & Services

1. Compute Category

* EC2
* Lambda

DATE : / /

2. Application Integration Category
* Services
* Simple Notification Services (SNS) & Simple Queue Service (SQS)

3. Database Category:
* Aurora & RDS
* DynamoDB

4. Security Identity & Compliance Category
* IAM
* IAM Identity Set

5. Storage Category
* Service, S3 Glacier, S3

6. Networking & Content Delivery Category:
* CloudFront * Route 53 * VPC

7. Cloud Financial Management Category:
* Billing & Cost Management
* AWS Billing Conductor
* AWS Marketplace

8. Container Category:
* Elastic Container Service
* Elastic Container Registry
* Elastic Kubernetes Services

9. Customer Enablement Category:
* Manage Services
* Support

EC2

10. Developer Tools:
* Code Commit
* Code Deploy
* Code Pipeline
* Cloud Shell

11. Management and Governance Category
* Cloud Watch
* Cloud Formation
* Cloud Trail

EC2

* EC2:

Elastic Compute Cloud

Resizable IT Resources Over the internet

H/W Resources S/W Resources

* EC2 is one of the resizable AWS service were one can get IT resources over the internet
* It belong to compute category.
* It is a type of Infrastructure as a Service [IaaS]

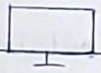
Components of EC2:

1. Instances:

Physical Machine

Laptop or others

Virtual Machine

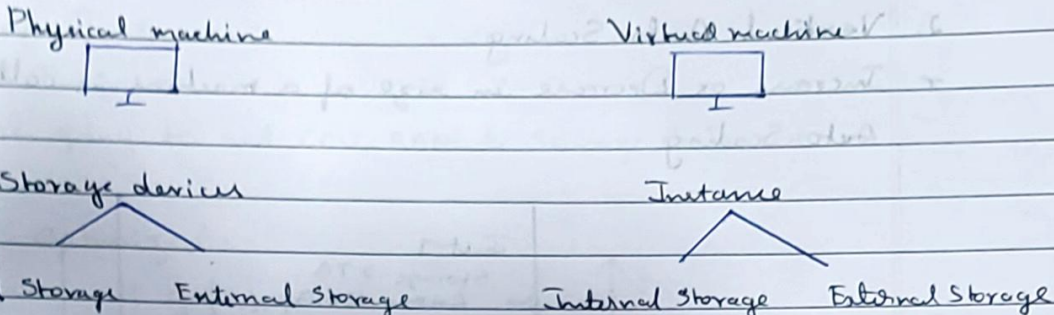


Instances

* Virtual machine / Server is called as Instances.

2 EBS Volume:
* Elastic Block Store.

DATE : / /



* Virtual drives are known as EBS volume.

3. Auto Scaling Group:

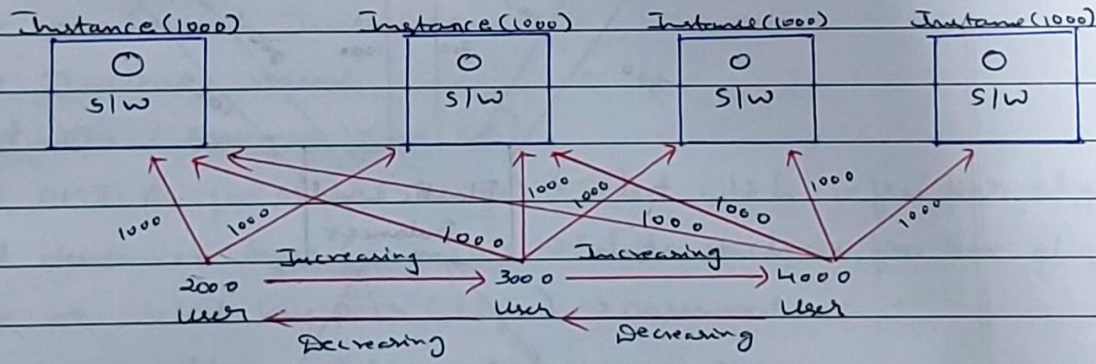
* Upscaling or Downscaling the resources automatically is called as Auto Scaling Group.
* There are two types of auto scaling.

1. Horizontal Auto Scaling
2. Vertical Auto Scaling

1. Horizontal Auto Scaling:

* Increasing or Decreasing no. of resources is called as Horizontal Auto Scaling.

(4GB-RAM
51 - Storage)

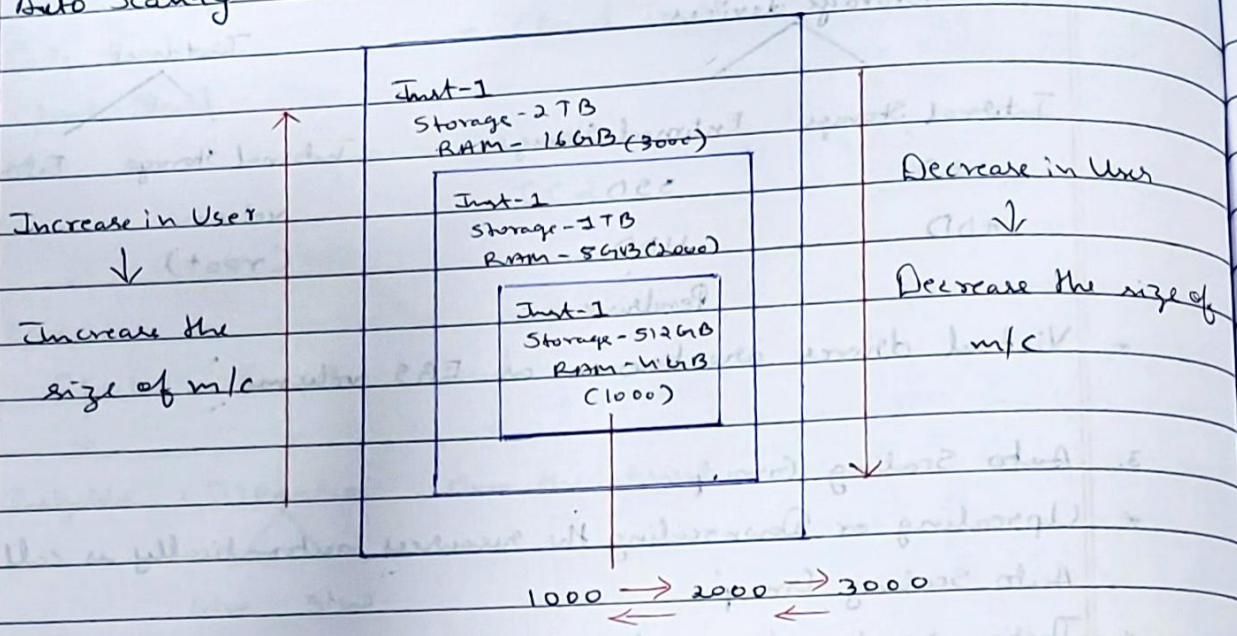


User Increasing - Increasing the no. of m/c's

User Decreasing - Decreasing the no. of m/c's

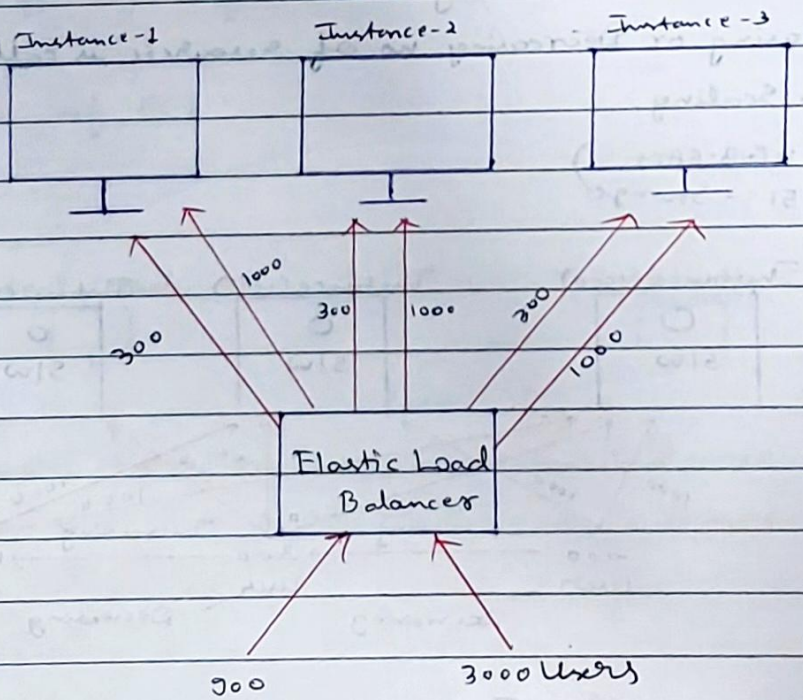
2. Vertical Auto Scaling:

* Increase or Decrease in size of a machine is called as Vertical Auto Scaling



4. Elastic Load Balancer:

* Elastic Load Balancer is used to distributing the load across the m/c's equally.



Monday

PAGE EDGE

DATE : 11 / 8 / 25

Types of Instances:

1. General Purpose:
* You are going to get CPU, RAM & Storage balanced
2.
2. Compute Optimized:
* Here you will get huge amount of CPU.

3. Memory Optimized:
* Here you will get huge amount of memory

4. Storage Optimized:
* Here you will get huge amount of storage

5. Accelerated Computing:
* If you want to develop any AI/ML application

6. HPC (High Performance Computing):
* Here you will get Graphics, CPU & memory.

Steps to Create Instance:

1. Sign into console
2. Go to EC2 dashboard
3. Click on Instances
4. Click on Launch Instances
5. Give Instance name
6. Select OS (Amazon Linux OS)
7. Select AMI (Amazon Machine Image) -> Select which is free tier eligible
8. Select Instances type (t2.micro) -> Select which is free tier eligible
9. Create or select key pair (login) -> password
10. Networking settings -> No need to worry now
11. Configuring storage (min 8 GB and more -> 16 TB for Linux OS)
-> Free tier (upto 30GB)

12. Click on launch Instance.

Instances Types:

* In Instance type section → click on info → click on Amazon EC2 instance type → click on Amazon EC2 Instances Type Details.
* By default t2.micro / t2.nano / t3.micro based on regions are available for free-tier eligible customer.
* 750 hrs of free usage per month

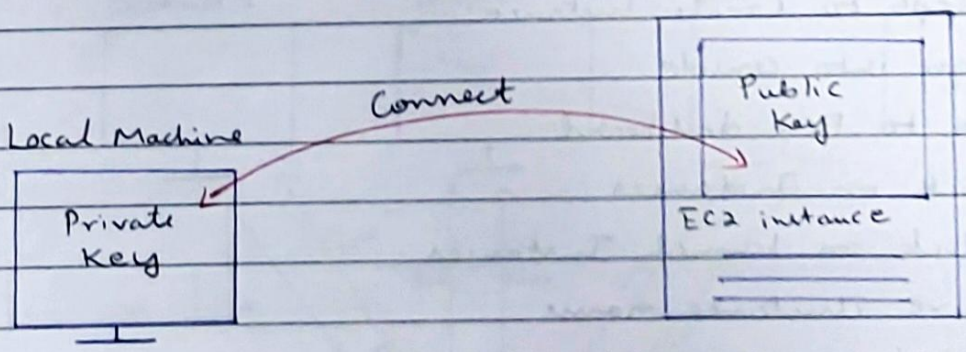
Key pair login:

* You can use a key pair to securely connect to your instance
* Ensures that you have access to the selected key pair before you launch the instance.

2 type of keypair

Public Key
* It is saved in your instances (AWS account)

Private Key
* It is saved in your local machine.



Key pair types:

1. RSA:

* This is the traditional & most widely used type of key pair
* It is named after the inventors - Ron Rivest, Adi Shamir,

Leonard Adleman.
* It is bigger & slower than ED25519

2. ED25519:
* This is the newer type of key pair that is considered to be more secure than RSA.
* This is smaller & faster than RSA.

Private key file format:
1. .pem (privacy enhanced mail)
* Windows >10, Linux, macOS
* For use with open SSH (Secure Shell)

2. .ppk (putty private key)
* Download putty software.

Steps to Connecting to Instance
1. EC2 Instance Connect → direct connect from EC2 console
2. SSH Client → Connect from your local machine using terminal/ Cand prompt.

Steps to connect to instance using SSH Client:
* Locate your private key file → go to downloads
* Open cmd prompt in downloads
* Copy command below example in SSH section
* Paste it in cmd prompt
* Confirm to connect by typing yes
* Click on enter

12/8/25
Tuesday

Basic Linux Commands.

1. touch <file_name> → to create file
2. touch <file1> <file2> → to create multiple files.
3. mkdir <directory_name> → to create directory
4. rmdir <directory_name> → to delete empty directory
5. rm-r <directory_name> → to delete directory which is not empty
6. cd <directory_name> → to change from one directory to other directory.
7. cd.. → to exit from current directory to previous / its parent directory
8. ls → to list all the file & directories
9. clear → to clear screen
10. nano <file_name> → to edit or add content to the files
11. pwd → to print the path of present working directory
12. rm <file_name> → to delete file
13. sudo yum install <s/w> → to install the s/w (Amazon Linux)
14. <s/w> --version → to check whether s/w is installed or not & its version
15. sudo yum remove <s/w> → to uninstall s/w (Amazon Linux)
16. history
17. cat <file_name> → to display the content of file
18. sudo apt install <s/w> → to install s/w in Ubuntu
19. sudo apt remove <s/w> → to uninstall s/w in Ubuntu
20. sudo apt update → to update apt package
21. mvn --version → to check maven version
22. sudo apt install openjdk-17-jdk → to install java in Ubuntu

Windows = folder
Linux = directory

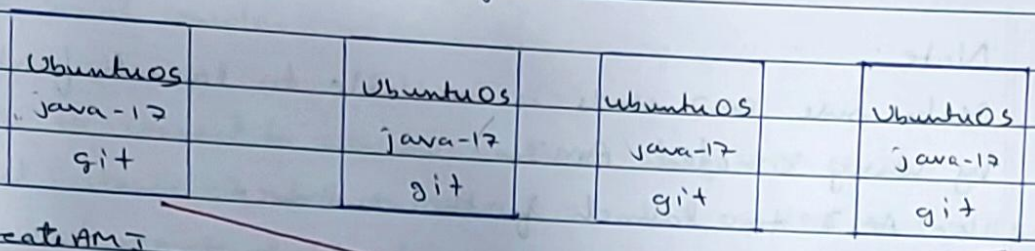
Nano editor
* navigate to editor
* add content
* to save → ctrl+s
* to exit → ctrl+x

DATE: / /

Task:

1. Create 4 instances
2. Install java & git in each instance
3. Create file sample.txt & add some content in all instance

AMI (Amazon Machine Image)



* Consists of OS + preinstalled software
* It is a blueprint or template consisting of OS + preinstalled s/w.

Note: So before creating AMI, Stop the instance, not mandatory but recommended.

How to create AMI & use it:

1. Launch instance (Instance 1)
2. Connect to instance & install required s/w in instances (instance 1)
3. Create AMI for this Instance
4. For remaining instances to be launchers use the created AMI.

Steps to create AMI:

Select instances → Click on Actions → Click on Image & Templates → Click on Create Image → Give Image name → Image description (optional) → Click on create Image

Step to use Created AMI to launch further instances

* Use AMI to create further instances → In applications & OS images section → choose My AMI's → Owned by me → select the Image

Note:

* Make sure AMI state is available to launch further instances by using created AMI
* Use AMI to launch further instance.
* Delete AMI

EC2 → Images → Click on AMI → Select the AMI → Click on Actions → click Deregister AMI → Click on Deregister AMI

* Delete Snapshots.

EC2 → EBS → Click on Snapshots → Click on Actions → Click on delete snapshots → type delete to confirm → click on delete

Steps to Create windows Instances and connecting it.

* While launching windows instances select OS as Windows

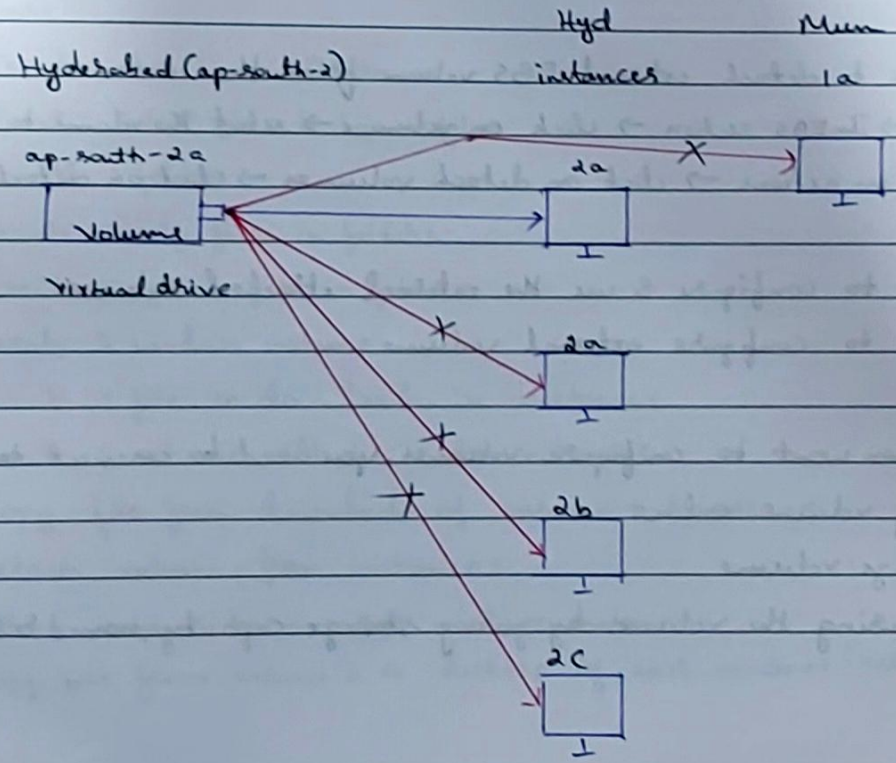
Connecting to Instance

1. Session Manager
2. RDP (Remote Desktop) Client
3. EC2 Serial console

Steps to Connect to windows instance using RDP client.

select instance → click on connect → click on RDP client → Click on Download remote desktop file → click on get passwd → click on upload private key file → choose correct key pair file → upload file → click on decrypt passwd → copy the passwd → go to downloads → click on RDP file → click on connect → past passwd → click on ok → click on yes →

EBS (Elastic Block Stores):



* We can achieve concept of virtual drives.
* We will create volumes which is nothing but virtual drives.
* It is availability zone specific
* You can attach one volume per instance at a time.

Note: * Volume present in 1a Az, can only use to instances of 1a Az but only for one instances at a time.

Steps to create EBS volumes:
EC2 → In EBS section → click on volumes → click on create volume → Select volume type (gp3) → Mention size of volume → Mention IOPS value → Mention throughput value → select the Az where volume should be created → click on create volume.

Steps to attach EBS volume to Instance:
EC2 → In EBS section → click on volumes → select the volume to the attach (verify the state of volume is available or not) → click on actions → click on Attach volume → select the instance which you want to attach the volume → select the device name → click on Attach volume.

Steps to detach external EBS volume from the instance:
EC2 → In EBS section → click on volumes → select the volume to be detached click on actions → click on detach volumes → click on detach.

Steps to configure & use the external attached volume:
* Steps to configure external volume:
Note:
If you want to configure volumes, you need to connect to instance first
1. Bring volume online
2. Initialize volume
3. Configuring the volume by giving storage capacity, name & drive letter

1. Bring volume online:
Server manager → wait → click on "file & storage service" → In volume → click on Disks → available volumes will reflect → select the volume which is offline → Right click → Bring online → click on Yes.

2. Initialize Volume:
Server manager → click on "file & storage service" → In volumes → click on Disks → available volumes will reflect → select the volume to initialize → Right click on that volume → click on Initialize → click on Yes.

3. Configure volume by give name, drive letter & giving volume storage capacity:
Server manager → Click on file & storage service → In volumes → click on disks → available volumes will reflect → select the volume to configure → right click on that volume → click on New volume → click on next → click on next → set the storage for volume to be create → click on next → give drive letter → click on next → give volume name → click on next → confirmation (details of volume) → click on create.

Task:

1. Create one Linux Instance
2. Create 5 text files & 5 java files
3. Create 2 folders
4. Create 3 pdf files in folder 1
5. Create 3 python files in folder 2
6. Create 2 windows instance
7. Create 1 file in downloads in instance 1
8. Create 1 volume attach it to instance 1
9. Copy file form downloads of instance 1 to volume 1
10. detach volume from instance 1
11. attach volume to another windows instance
12. Copy file from volume 1 to desktop of 2nd windows instance.

15/8/2025
Friday

Snapshots:

* Snapshots are used for backup purpose
* We can backup volumes & AMI image's

Step to achieve backup for EC2 instance:

1. Launch Instance
2. Create files & Install s/w's
3. Create Snapshot
4. Create Image for Snapshot
5. Launch Instance using image.

Steps to create Snapshot:

1. Go to EC2 service
2. Click on volumes under EBS
3. Select the volume to create snapshot
4. Click on Action
5. Click on create Snapshot

Note: Wait until Snapshot status is completed

→ Now, Delete instance

Step to create AMI using snapshot

1. Go to EC2 service
2. Click on snapshots under EBS
3. Select snapshot to create AMI
4. Click on Actions
5. Click on create Image.

→ Now, create one more instance using AMI, check whether installed s/w & created files are reflecting or not.

Saturday

S3

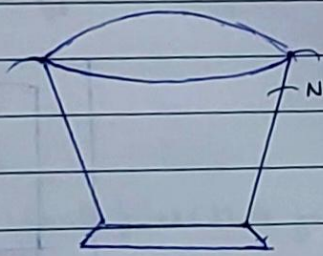
S3:

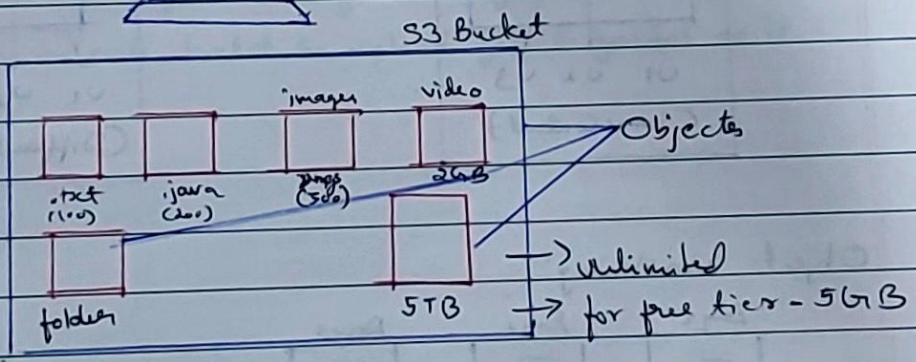
* S3 stands for Simple Storage Service
* It belongs to storage category.
* It is scalable, secure, durable & reliable cloud storage service
* Users can achieve unlimited storage.
* Components of S3 are
  - Buckets
  - Objects
* You can store any types of files.
* It is Global service.

Use Cases of S3:

1. Storage
2. Backup and restore files
3. Static Web Hosting
4. Big Data and AI/ML storage
5. Store your app data.

1. Buckets:





S3 Buckets:

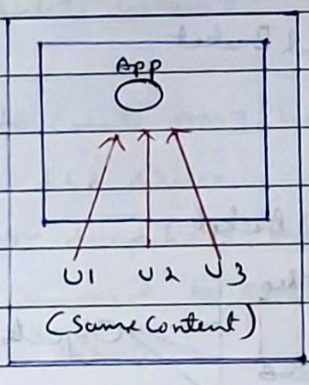
* Global service components.
* You can achieve unlimited storage.
* It is container to store objects (file)
* You can store any type of files (objects)
* For free tier user → only 5GB capacity is free
* Bucket name should be unique across the globe
* Can be accessed at anytime from anywhere
* It follows naming convention (lowercase, -, etc)

Objects:

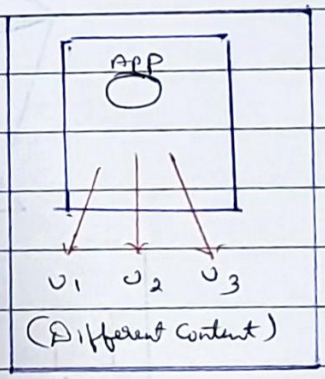
* Objects are nothing but files / folders
* Maximum single file size can be upto 5TB
* You can store any type of file
* You can do file/object versioning
* You can encrypt objects

Note:

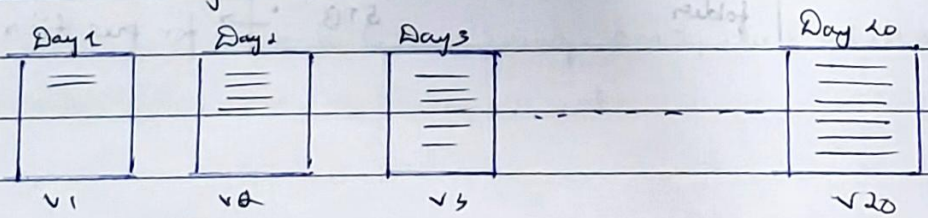
Static Web hosting



Dynamic Web hosting



Object Versioning:



* Saving the multiple different version of single file is called object versioning.

Steps to Create bucket:

1. Go to S3 service.
2. Click on Create bucket
3. Select bucket type as "General purpose"
4. Give bucket name (follow naming convention)
5. Select Object Ownership.
   - ACLs (Access Control List)
   - ACLs disable → Only owner is owned the objects
   - ACLs enable → Other AWS users can also owned the objects
6. Block public access (check it)
7. Bucket versioning (disable)
8. Click on create bucket

How to upload Objects:

* Click on the Bucket → click on upload → click on Add files / Add folder → select the files/folders to upload → click on open → click on upload.

Task:

1. Create 3 buckets
2. Upload your local files (2) & folder (2) foreach bucket
3. Create folder in each bucket & upload file & folder to it.

18/9/2023
Monday

Steps to Static Web Hosting:

1. Go to "template website" & download html templates & extract it.
2. Create Bucket
3. Upload file separately & each folder separately
4. Go to Properties (click on "Properties")

↓

* Search for Static Web hosting
- Click on edit
- Choose enable for static web hosting.
- Select Hosting type as Host a static website
- Index document ("index.html").
- Click on save changes.

5. Go to Permissions ("Click on permissions")

→ search "Block Public access"

* Check whether it is turn off or on.
* If it is on you need to turn it off.
* click on edit
- uncheck Block all Public access
- click on save changes
- type confirm
- click on confirm

→ search "Bucket Policy"

* Create bucket policy
- check for bucket policy
- click on edit
- click on Policy generator

→ AWS Policy Generator

Step 1: Select Policy Type

* S3 Bucket Policy

Step 2 : Add Statement (s)

Effect → Allow

Principal → *

Actions → Get Object

ARN → copy & paste bucket ARN followed by /*

(Amazon Resource Name)

→ Click on Add Statement

Step 3 : Generate Policy

→ Click on generate policy

* You will get auto generated policy, copy & paste it in edit bucket policy

→ click on save changes.

6. Now you achieved static web hosting using S3.

Task :

1 Do 3 static web hosting.

10/8/2025
Tuesday

IAM

IAM:

* IAM stands for Identity and Access Management in AWS.
* It is a service that helps you securely manage access to AWS services & resources.
* IAM allows you to control "who" (users, groups, roles) can do "what" (permissions, policies) on "which resources" (S3, EC2, RDS etc)
* It comes under security, Identity & Compliance category.
* It is Global Service.

Main Components of IAM

1. Users
2. User Groups
3. Roles
4. Permission policies

1. Users:

* Represents individual identities in AWS (e.g. developers, admins, testers).
* Each user has credentials:
  - Username + password → For AWS Management Console login
  - Access keys (Access Key ID + Secret Access Key) → For CLI / SDK / API

2. Groups:

* A collection of Users
* Helps apply the same set of permissions to multiple users.
* Example:
  - Developers group → EC2 + S3 access
  - Admin group → Full AWS access

3. Roles:

* A temporary identity with specific permissions.
* Unlike users, roles don't have permanent credentials.

* Used for:
  - AWS services to interact with each other (e.g. EC2 accessing S3)
  - Cross-account access
  - Federated access (Google, Active Directory, etc)
* Whoever is trying to access e.g. EC2 accessing S3 → Create role for EC2

v. Policies:

* A JSON document that defines permissions.
* Example policy: Allow s3:GetObject on arn:aws:s3:::mybucket*.
* Two types:
  1. AWS Managed Policies (predefined by AWS)
  2. Customer Manged Policies (created by you)
* Whoever is giving permissions
  - Eg, EC2 accessing S3 → Create Permissions Policies for S3.

→ Eg for Users and User Group:

User: Ur1, Ur2, Ur3, Ur4, Ur5, Ur6, Ur7, Ur8, Ur9, Ur10

Groups: Development, Testing, DevOps.

Groups	Permissions	Users
Developments	* EC2-Full access	* Ur1, Ur2, Ur3
	* S3-Full access	Ur4
	* IAM-Full access	
Testing	* EC2-Full access	* Ur5, Ur6
DevOps	* EC2-Full access	* Ur7, Ur8
	* S3-Full access	
	* Iam-Full access	
	* VFC-Full access	
—	* EC2-Full access	* Ur9
	* S3-Full access	

5. Access keys:

* Used by applications / CLI / SDK instead of passwords.
* Pair of: → Access key ID
  → Secret Access Key

Hands-on on IAM:

Steps to create users & user group:

Go to IAM service → click on users → click on create users → give user name → check "Provide user access to the AWS Management Console" → set passwd (autogenerated/custom) → check "User must create a new password at next sign-in" → click on next → select add user to group → you can select existing group you can create new group → click on create group → give group name → attach required permissions for the group → click on user group → select the group to add user → click on next → Review it → click on user → download user credentials → Access the AWS using user credential.

Steps to create Roles & Permission Policies:

Create Role:

1. Go to IAM service
2. Under Access Management --> click on Roles
3. Click on create role
4. Trusted entity type
   → AWS service
5. Select Use case
   → EC2
6. Click on Next
7. Add permission (Permission policies) → Don't select any policy
8. Click on Next
9. Give role name
10. Click on create role
11. Click on view role

Create Permissions Policies:

1. Go to IAM service
2. Click on Roles.

3. Search for role in which you want to create permissions policies.
4. Click on the role.
5. In Permissions → In Permissions Policies → click on Add permission → click on create Inline Policy.
6. Specify permission (Select the service to which we want to create permissions)
   → S3
7. Actions allowed (Select the actions to be performed on S3 bucket from EC2)
   (Don't select all sections)
   In Access level
   1. List
      → ListAllMyBuckets → List the buckets
      → ListBucket → List the objects
   2. Read
      → GetObject → to download objects & to access it
   3. Write
      → CreateBucket → to create the bucket
      → DeleteBucket → to delete the bucket
      → DeleteObject → to delete object
      → PutObject → to upload object

8. Resources:
   → All (To perform selected actions on all S3 buckets)
   → Specific (Need to choose specified bucket to perform selected actions
      * Buckets ARN (Specify bucket / Any)
      * Objects (Specify objects / Any)
9. Click on next
10. Give policy name
11. Click on create policy
12. Roles & policy are created

Steps to attach roles:
Go to EC2 → Select the instance to which you want to attach roles → Click on Actions → click on Security → Click on Modify IAM role → Select the role which you want to attach → Click on Update IAM role

Note:
* To install awscli in ubuntu instance
→ update apt package → sudo apt update
→ To install aws cli → sudo snap install aws-cli --classic

Commands to access S3 bucket from EC2.

1. To list all buckets (List All My Buckets) → aws s3 ls (ls=list)
2. To list the objects of the bucket (List Bucket) → aws s3 ls s3://bucket name
   * To download & upload the objects →
3. aws s3 cp <source_path> <destination_path> (cp=copy)
3. To download the objects from s3 to ec2 (Get Object) →
   aws s3 cp s3://bucket_name/file_name /path/to/ec2
   (/home/ec2-user)
4. To upload the objects from ec2 to s3 (Put Object) →
   aws s3 cp /path/to/ec2-user/file_name s3://bucket_name
5. To create bucket (Create Bucket) → aws s3 mb s3://new_bucket_name
   (mb=make bucket)
6. To delete bucket (Delete Bucket) → aws s3 rb s3://bucket_name
   (rb=remove bucket)
7. To delete objects/files from the bucket (Delete Object) →
   aws s3 rm s3://bucket_name/file_name (rm=remove)

PAGE EDGE

DATE : 21 / 8 / 25

Thursday

Task:

1. Create Linux Instance
2. Create S3 bucket
3. Upload local files to bucket directly
4. Try to access S3 bucket from ec2 instance
5. Create IAM roles & permission policies
6. Attach IAM Role to ec2 instance
7. Perform all operations (list bucket, list objects, mb, rb, download, upload & delete file)

Task:

1. Create 2 group (Attach policies)
2. Create 5 users
3. Assign 2 users to one group & 3 users to another group
4. And then login to aws account as a IAM user for 1 user in each group.

28/8/2025
Thursday

VPC

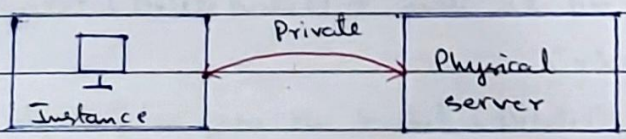
VPC:

* It belongs to the Networking & Content Delivery Category
* A Virtual Private Cloud (VPC) in AWS is a logically isolated virtual network in the cloud where you can run your resources securely.
* It gives you complete control over your networking environment, similar to having your own private data centre, but hosted on AWS infrastructure.
* It allows you to launch & manage AWS resources (like EC2 instance, RDS database etc) in a secure, customizable network environment
* VPC ensures isolation from other user's networks in AWS.
* You can control how resources communicate internally & externally.
* We can achieve the concept of Hybrid Cloud.

- AWS Direct Connect : AWS Direct Connect is a dedicated network connection between your on-premises data centre (or office) & AWS. Instead of using the public internet, it provides a private, high-speed, & low-latency link to AWS services.
- AWS VPN : AWS VPN is a service that creates a secure, encrypted connection between your on-premises network (or user devices) & your AWS VPC over the public internet.

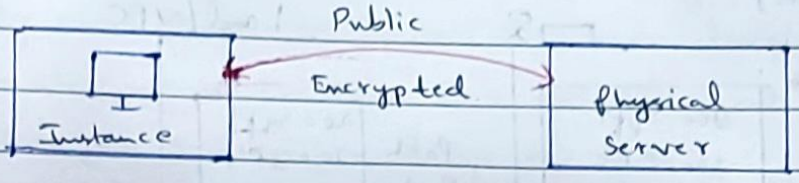
Note:

1. AWS Direct Connect



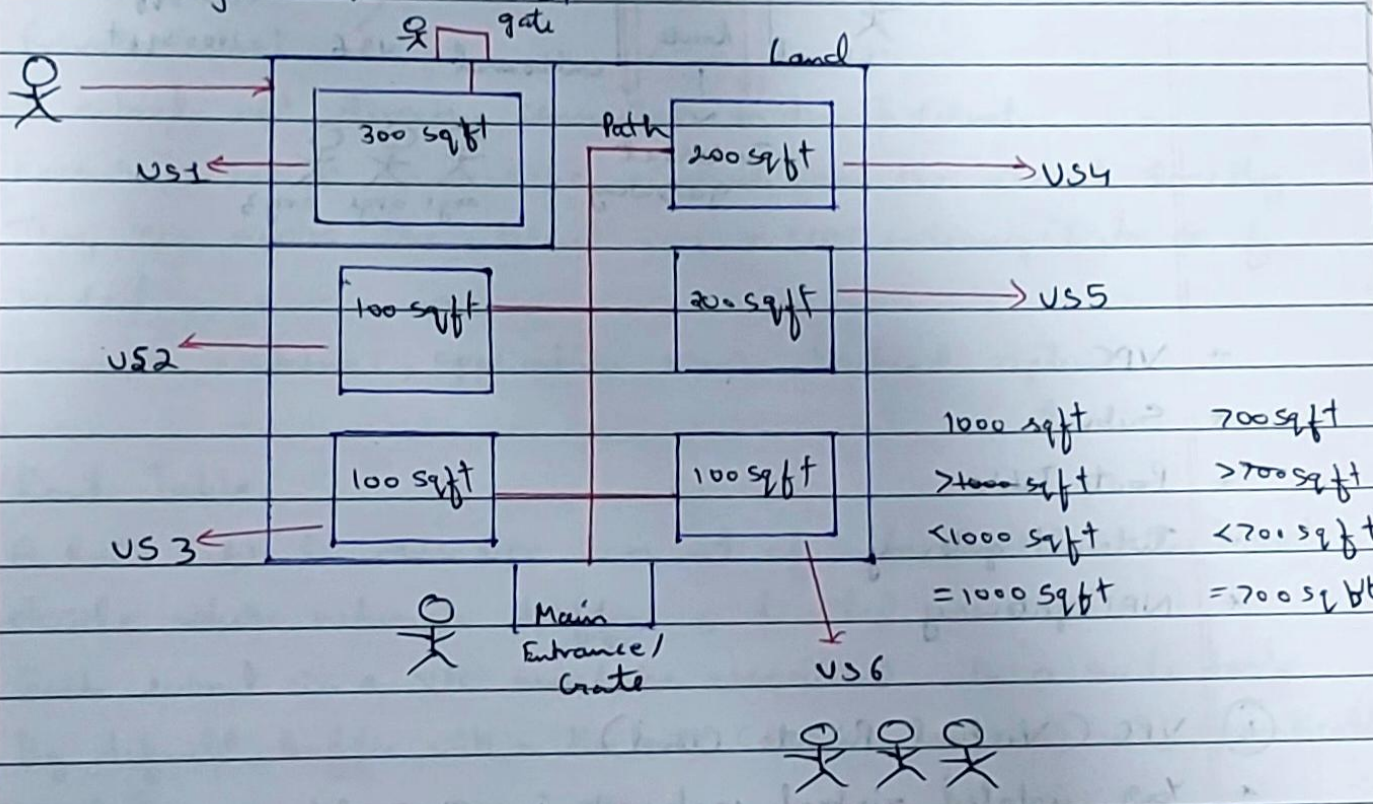
a. Hybrid over a dedicated private line.
b. Private dedicated connection between your on-premises data centre & AWS VPC, without using the public internet.

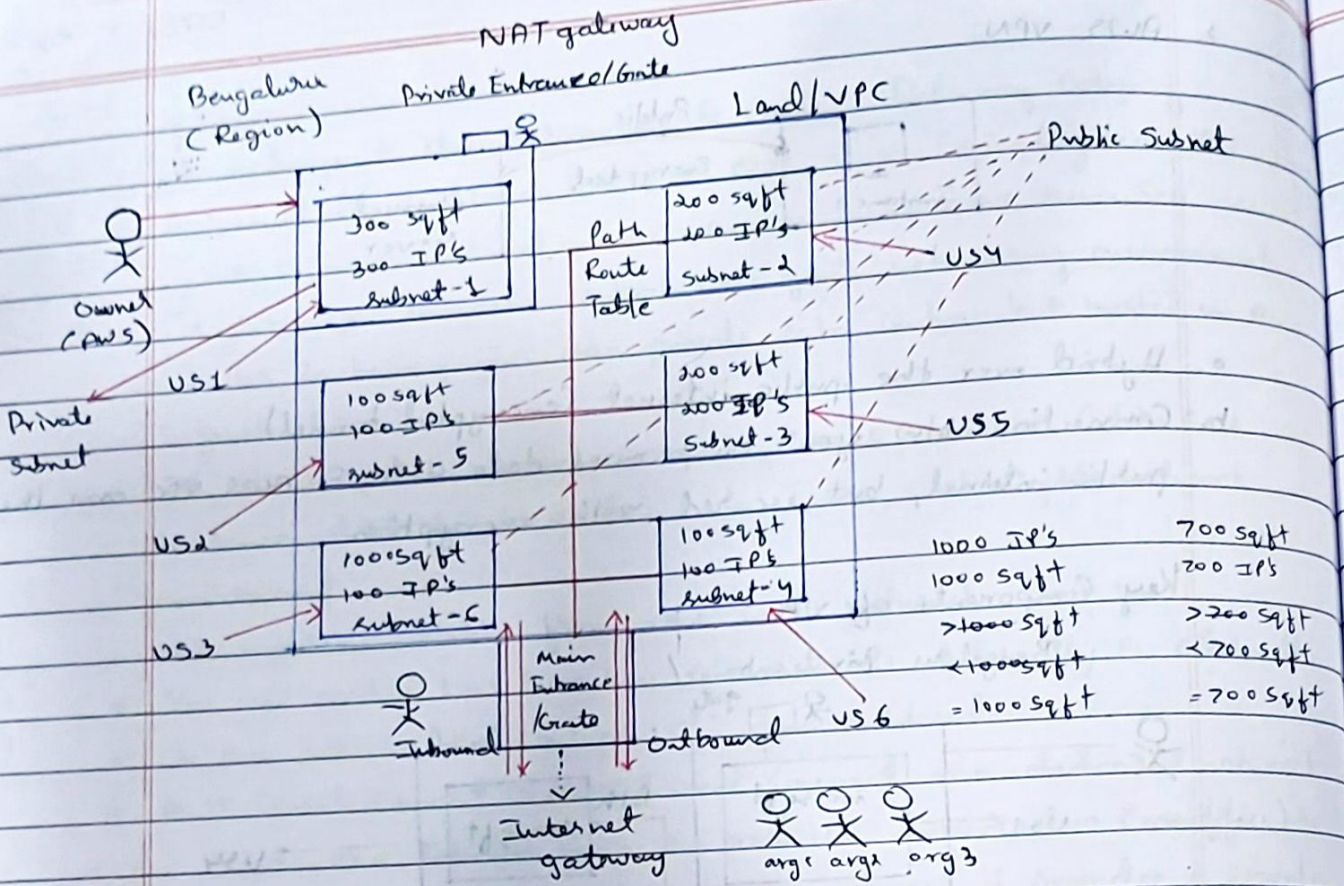
2. AWS VPN:



a. Hybrid over the public internet (encrypted tunnel)
b. Connection b/w your on-premises data centre & AWS VPC over the public internet, but secured with encryption.

Key Components of VPC:





* VPC
* Subnet
* Route Table
* Internet gateway
* NAT gateway

(1.) VPC (Virtual Private Cloud)

* Your isolated virtual network in AWS, like your own private data centre.

(2.) Subnet:

* A subnet is a segment of a VPC's IP address range where you can place your AWS resources (like EC2 instances, databases etc.)
* It helps organize & isolate resource inside a VPC.

* Each subnet exists within a single Availability zone (Az) for high availability and fault tolerance (It means designing resources (EC2, RDS etc) across multiple Az's, so that your applications keeps running even if one component or Az fails).

Type of Subnets:

1. Public Subnet:
* A subnet that is connected to an Internet Gateway (IGW)
* Resources inside can send & receive traffic from the internet
* Example: Web servers, load balancers.

2. Private Subnet
* A subnet not directly connected to the Internet.
* Resource inside cannot be accessed from the internet directly
* They can access the internet via a NAT Gateway/Instance if needed.
* Example: Databases, application servers, backend systems.

(3.) Route Table
* A Route Table in AWS VPC is a set of rules (called routes) that decide where network traffic is directed.
- Each subnet in a VPC must be associated with a route table.
- By default, traffic within the VPC (between subnets) is automatically routed
- To send traffic outside (e.g. to the internet, another VPC, or on-premises), you must add route.
- Eg: -
  - Public Subnet's route table → Has a route to the Internet Gateway (IGW)
  - Private Subnet's route table → Has a route to the NAT Gateway for internet access

NOTE:
Route tables control how traffic moves inside a VPC & how it exits to external networks.

④ Internet Gateway (IGW):

* It is a VPC component that allows resources in public subnets to communicate with the internet.
* It must be attached to a VPC & referenced in route table to enable inbound & outbound traffic
* It's the bridge b/w your VPC & the internet.

⑤ NAT (Network Address Translation) gateway:

* A NAT gateway is an AWS-managed service that allows resources in a private subnet to access the internet (for updates, downloads, and patches) while preventing the internet from initiating connections to them.
* It's a secure, scalable & managed way to give outbound internet access to private subnet resources.
* Outbound internet access for private subnet resources.
* Only outbound connections are allowed; inbound traffic is blocked.
* A NAT Gateway acts like a bridge so that private subnet instances can download, updates, install packages, or access external services without exposing them directly to the internet.

Friday

PAGE EDGE
DATE: 29/8/25

Address:

* We need address if we want to identify any place uniquely
* We need address, if we want to identify machines uniquely.

Types of address in m/c's:

1. Physical address
2. Logical address

1. Physical address

* It is the address of your local machine.
* It is constant & not going to change.
* If you want to identify physical address of your machine
  → cmd prompt → getmac
* Addresses you are getting is based on network cards that your machine consists.
* To get physical address we don't need any network.
  getmac → get media access control address
  → only for windows

2. Logical address:

* If you want to find your machine uniquely over the internet at that time you need this logical address.
* Whenever you connect to internet, then only you will get logical address
* For your m/c, if you want identify logical address of your m/c.
  → ipconfig

Types of IP address:

1. Based on Permanence:

Permanence → refers to how long something lasts or remains unchanged.

a. Static IP address (Permanent)
b. Dynamic IP address (Non-Permanent)

2. Based on Version:
a. IPv4
b. IPv6

3. Based on Functionality:
a. Public IP address
b. Private IP address

1a. Static IP Address
* IP address which is constant
* It is permanent IP address assigned to device
* It will not change over time

1b. Dynamic IP Address.
* IP address which is not constant
* It is non permanent IP address assigned to device.
* This IP is keep on changing while connecting to new networks.

2a. IPv4.
* It stands for "Internet Protocol version 4".
* If we are seeing any IP address with four set of number seperated by a dot is IPv4 address.

1st set . 2nd set . 3rd set . 4th set

* Each & every set is an octate number.

2^8 . 2^8 . 2^8 . 2^8

1st set	2nd set	3rd set	4th set
256	256	256	256
0	0	0	0
1	1	1	1
2	2	2	2
...	...	...	...
255	255	255	255

$2^{\wedge 8} = 256$

$256 * 256 * 256 * 256$

$= 4294967296$

$= 4.3$ billion possibilities

Initial IPv4 address 0.0.0.0

$2^8 = 256$

Last IPv4 address 255.255.255.255

0.0.0.0	0.0.1.0	0.0.2.0	
0.0.0.1	0.0.1.1	0.0.2.1	0.0.0.0
0.0.0.2	0.0.1.2	0.0.2.2	
0.0.0.3	0.0.1.3	0.0.2.3	
0.0.0.255	0.0.1.255	0.0.2.255	255.255.255.255

256 x 256 x 256 x 256
= 4294967296
= 4.3 billion possibilities

Classes of IPv4:

* IPv4 addresses are divided into 5 different classes
* They are class A, class B, class C, class D & class E
* Each class is having some range of IP addresses
* Each class is having different purpose

1. Class A
* Range 0.0.0.0 to 126.255.255.255
* These are used for layer networks

2. Class B
* Range 127.0.0.0 to 191.255.255.255
* These are used for medium networks

3. Class C
* Range 192.0.0.0 to 223.255.255.255
* These are used for smaller networks

4. Class D
* Range 224.0.0.0 to 239.255.255.255
* Multicast purpose (Radio, walkie-talkie etc.)

5. Class E
* 240.0.0.0 to 255.255.255.255
* Experimental purpose

Note:
* class A, B & C are seen in day to day life
* Whereas class D & E are reserved networks.

Instances	Private IPv4 addresses	Public IPv4 addres
Instance 1	172.31.30.28 (B)	54.185.208.135 (A)
	172.31.30.29	. . . . .
	172.31.30.29	35.89.76.158
Instance 2	172.31.20.173 (B)	35.94.121.93 (A)
	172.31.20.173	. . . . .
	172.31.20.173	35.88.121.8

3a. Public IPv4 address:
* While working over the internet
i.e, only if you are having public IP address, then only you will be able to communicate with anything over the internet.
* It is Dynamic IP address
* Can be duplicate n no. of times within an area that means people can get the same IP addresses.
* This is unique IP address.

36. Private IPv4 address.

* Associated to your machine.
* It is static IP address
* Using this you can uniquely identify your machine over the internet.

Exhaustion of IPv4:

max ipv4 → 4.3 billion

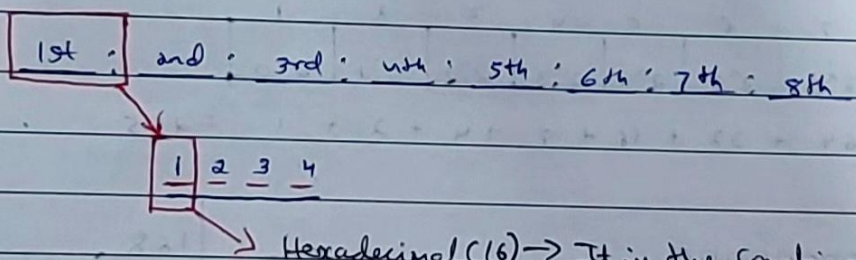
max population → 8 billion

3 billion people are using machines

3 billion * 2 = 6 billion devices

IPv6:

* It stands for Internet Protocol version 6.
* To overcome the exhaustion of IPv4 address
* It is having 8 sets separated by a colon & each set is having 4 digits
* Each digit is hexadecimal (16).
* It is alpha numeric (contains both alphabets & numbers)



* Each & every digit is having 16 option.
* Possibilities of each set is $16^4 = 16*16*16*16 = 65,536$

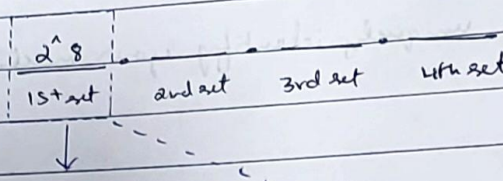
→ How many IPv6 address:

* $65536^8 = 65536 * 65536 * 65536 * 65536 * 65536 * 65536 * 65536 * 65536$.

01/09/2025
Monday

Why 2^8 ?

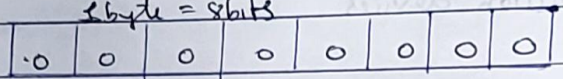
* Conversion of decimal IP to binary IP

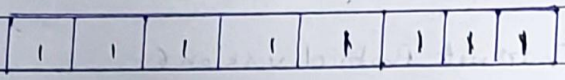


Base value is 2, because of binary numbers (1's & 0's)

1 set = 1 byte
1 byte = 8 bits
* Power is 8 due to 8 bits

1 byte = 8 bits





172 = 10101100

203 = 11001011

254 = 11111110

8	7	6	5	4	3	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255

0 → 00000000

1 → 00000001

2 → 00000010

3 → 00000011

168 → 128 + 32 + 8 → 10101000

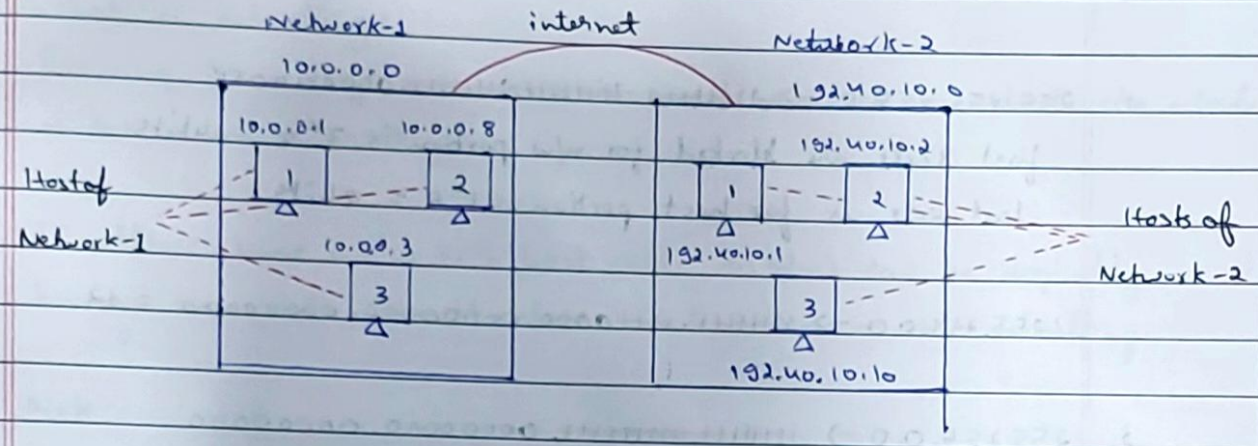
2	168
2	84 - 0
2	42 - 0
2	21 - 0
2	10 - 1
2	5 - 0
2	2 - 1
	1 - 0

168 = 10101000

DATE: / /

IPv4 address is divided into 2 parts:

1. Network Portion
2. Host Portion



10.0.0.3 (1st network & 3rd device)

192.40.10.2 (2nd network & 2nd device)

10.0.0.8 (1st n/w & 2nd device)

192.40.10.10 (2nd n/w & 3rd dev)

1. Network Portion:

* Identifies which network the IP belongs to
* It will define the network range
* Consecutive 1's from left

2. Host Portion

* Identifies a specific device within that network
* It will defines the specific device within that network
* Consecutive 0's after network portion.

Note:

* To identify & decide to which is host portion & which is network portion Subnet mask comes into picture.

Default subnet mask are:

1. 255.255.255.0
2. 255.255.0.0
3. 255.0.0.0

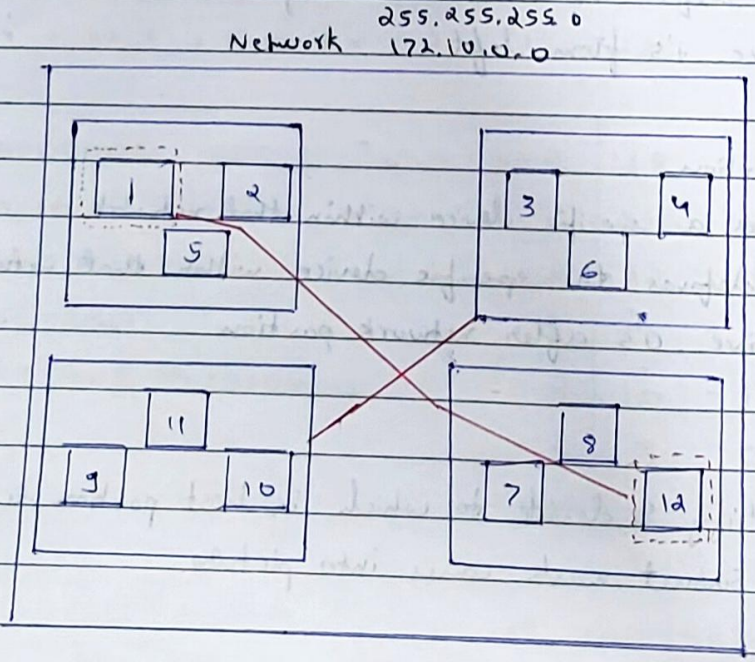
1. 255.255.255.0 -> 11111111.11111111.11111111.00000000
   first 3 sets are blocked for n/w portion = 3*8 = 24bits
   last set is for host portion = 1*8 = 8bits.

   255.224.0.0 -> 11111111.11100000.00000000.00000000 = 32

2. 255.255.0.0 -> 11111111.11111111.00000000.00000000
   first 2 sets are blocked for n/w portion = 2*8 = 16 bits
   last set is for host portion = 2*8 = 16 bits

3. 255.0.0.0 -> 11111111.00000000.00000000.00000000
   first set is blocked for n/w portion = 1*8 = 8bits
   last 3 sets are blocked for host portion = 3*8 = 24bits

Why Subnet?



1. 255.255.255.0 → 11111111.11111111.11111111.00000000 255.255.0.0
   10.0.0.0 → 00001010.00000000.00000000.00000000 172.16.0.0
   172.100.0.0 → 10101100.00001010.00000000.00000000

What is subnetting?

* Breaking down large networks into small networks or subnetworks which is known as subnetting.

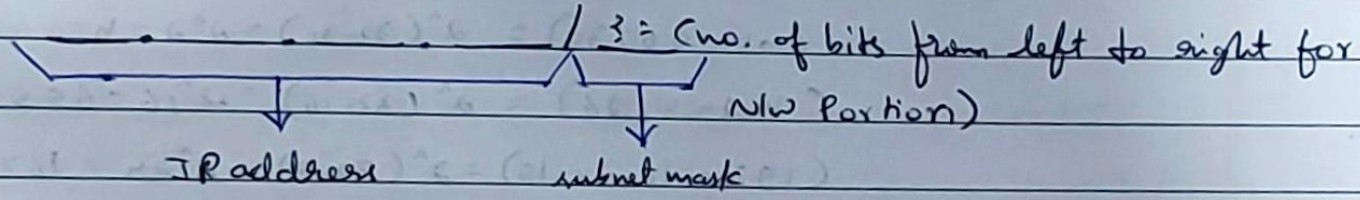
or

* IP address have a n/w & a host part, so n/w's can be logically broken down into small n/ws which is known as subnetting

Note:

* To indicate IP address & subnet mask together, where CIDR comes into pictures.
* CIDR is also called as slash notation.

Representation of CIDR:

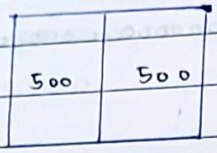


Examples:

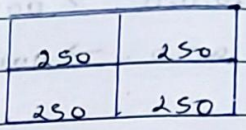
1. 192.168.10.1/8 → 8bits for n/w portion & 24 bits for host portion
2. 134.10.4.0/20 → 20 bits for n/w portion & 12 bits for host portion
3. 172.40.0.0/30 → 30 bits for n/w portion & 2 for host portion
4. ~~122.10.2.0/32 → 32 bits for NP & 0 for HP.~~

How to divide one network into subnetworks

Land



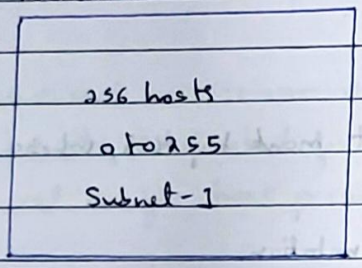
1000 sqft



Note:

* To divide the nlw we need to use host portion

Network (100.0.0.0/24)



How to calculate hosts?

Hosts IP's = 2^(Total no. of bits - No. of bits occupied for NP)

= 2^(32-24)

= 2^8

= 256

(100.0.0.0/30) = 2^(32-30) = 2^2 = 4

(10.24.8/16) = 2^(32-16) = 2^16 = 65536

(193.0.10.240/8) = 2^(32-8) = 2^24 = 16777216

Note:

* To divide the nlw we need to use host portion

Network -1 (100.0.0.0/24)

256 hosts 0 to 255

subnet-1	subnet-2
128 hosts	128 hosts
100.0.0.0/25	100.0.0.128/25
0 to 127	128 to 255

~~hosts = 2^(32-23) = 2^9 = 512~~ X

hosts = 2^(32-25) = 2^7 = 128 ✓

100.0.0.8 → Sn1

100.0.0.150 → Sn2

100.0.0.255 → Sn 2

100.0.0.100 → Sn 1

Network-1 (100.0.0.0/24)

256 hosts

subnet-1	subnet-2
64 hosts	64 hosts
100.0.0_/26	100.0.0_/26
0-63	64-127
Subnet-3	Subnet-4
64 hosts	64 hosts
100.0.0_/26	100.0.0_/26
128-191	192-255

120 → Sn2

30 → Sn1

200 → Sn4

160 → Sn3

~~120~~

Components of VPC

1. VPC
2. Subnets
3. Route Tables
4. Internet Gateway
5. NAT Gateway

1. Steps to create VPC:

1. Go to VPC dashboard
2. click on your VPC
   → VPC settings
   → Resources to create
   → select VPC only
   → Give name

→ IPv4 CIDR block
→ IPv4 CIDR manual input
→ IPv4 CIDR
(Allocate IPv4 address using CIDR)
→ IPv6 CIDR block
(No IPv6 CIDR block)
→ Tenancy
3. Click on create VPC

2. Steps to create Subnet:
1. Go to VPC dashboard
2. Click on Subnets
3. Click on Create subnet
→ VPC
→ select VPC ID
→ Subnet settings
→ Give subnet name
→ Select AZ
→ IPv4 VPC CIDR block (10.0.0.0/24)
→ It provides allocated VPC range
→ IPv4 subnet CIDR block (10.0.0.0/24)
→ Assign IP's to subnet
4. Click on Create subnet

3. Steps to create Route Tables:
→ Go to VPC dashboard
→ click on route tables
→ click on create route tables
Route table settings
Name - optional
VPC

→ select vpc where you want to use it
→ click on create route tables.

4. Steps to create Internet gateway
→ Go to vpc dashboard
→ click on internet gateway
→ click on create internet gateway
Internet gateway settings.
Name tag
→ give name
→ click on create internet gateway.

Attaching components:

1. Attaching subnet to route tables.
→ go to subnet.
→ select the subnet which you want to attach it
→ click on Actions
→ click on edit route table association
→ select the rt to which you needed to attach to the subnet
→ click on save

2. Attaching internet gateway to VPC:
→ go to IGW
→ select to igw need to be attach
→ click on Actions
→ click on Attach to vpc
→ select vpc to which igw need to be attach
→ click on Attach igw

3. Attaching route tables & internet gateway
→ go to route tables.

→ select the rt to be attached
→ click on actions
→ click on edit routes
→ click on add route
→ select the destination (0.0.0.0/0)
→ select target as Internet gateway
  → choose igw
→ click on save changes

Note: * After practicing, immediately delete VPC.

→ Don't delete default VPC & default components
  → click on your vpc
  → click on action
  → click on delete vpc
  → type delete to confirm
  → click on delete

Steps to create VPC & its components using 'VPC & more'

1. Go to vpc dashboard
2. click on Your VPC
3. click on create VPC

VPC settings.
  →select VPC & more
Give Project name
IPv4 CIDR block
  →allocate IP's to VPC
IPv6 CIDR block
  →select No IPv6 CIDR block
Tenancy
  →select Default

Choose "Number of Availability Zones (AZ's)".
Choose "Number of public subnets"
Choose "Number of private subnets"
NAT gateway ($)
→ choose Name
VPC end points
→ choose name
4. Click on create VPC
5. Click on view VPC

Steps to Launch EC2 instance in configured VPC:
→ Go to EC2 service
→ click on Launch Instance
→ give instance name
→ Select OS
→ Select instance type
→ select / create key pair

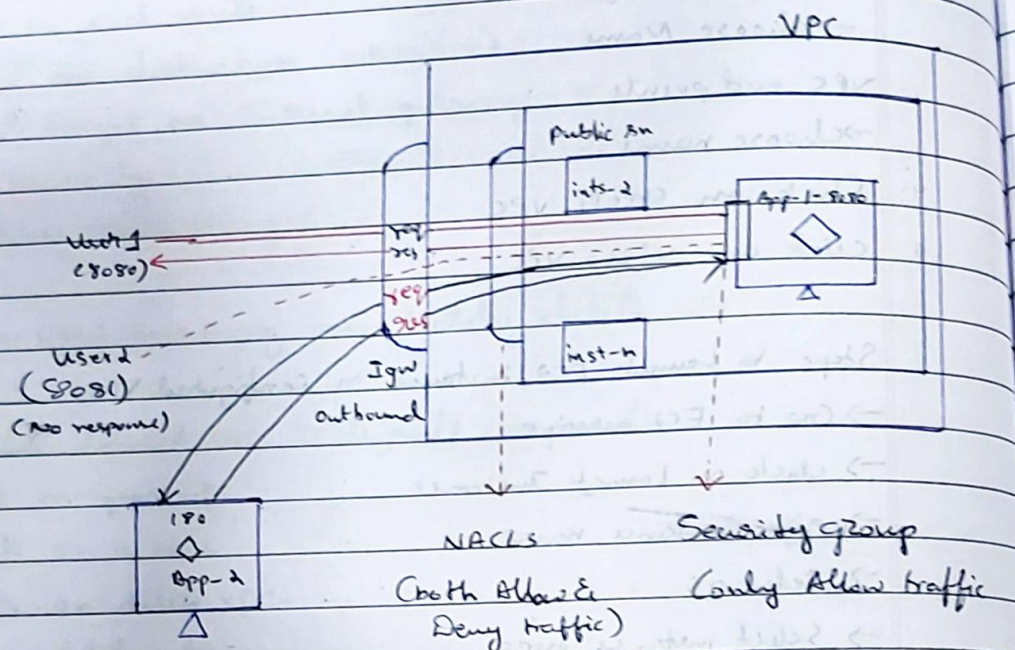
→ Network settings.
→ click on edit
→ choose created VPC
→ Choose created subnet (select public/private subnet)
→ Auto assign public IP
select enable
→ keep everything as it is
→ configure storage
→ click on launch instance

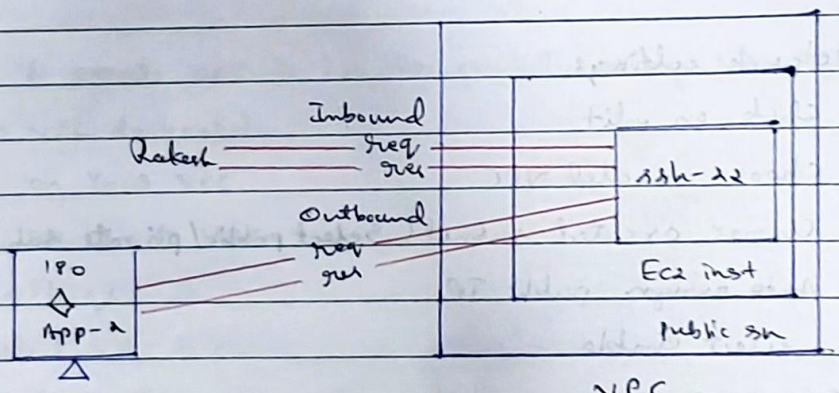
Note: * By deleting VPC the components associated with VPC i.e, subnet, rt, igw will also delete automatically.

04/09/25
Thursday

Security Groups & NACLs (Network Access Control Lists)

* Both sg & NACL are used to control n/w traffic in cloud environments (aws).





1. Security Groups:

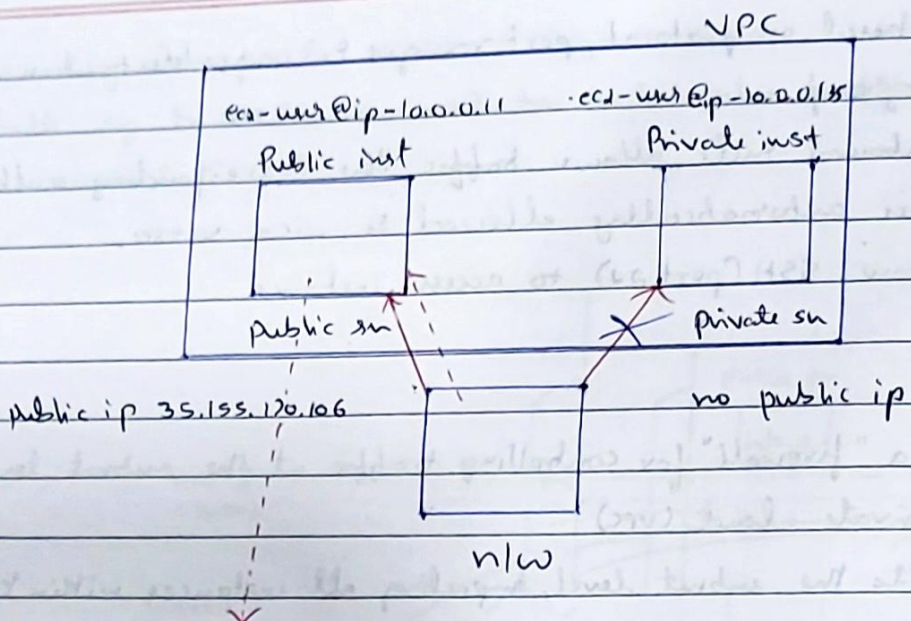
* Acts as a "virtual firewall" for instances to control inbound & outbound traffic
* Applied at the instance level
* Supports only allow rules

* Rules are based on protocol, port ranges & source/destination IP or security group.
* If an inbound rule allows traffic, the corresponding outbound response is automatically allowed & vice versa.
  Eg: Allow SSH (port 22) to access instance.

2. NACLs
* Acts as a "firewall" for controlling traffic at the subnet level in a virtual private cloud (VPC)
* Applied to the subnet level, impacting all instances within the subnet.
* Supports both allow & deny rules.
* Rules are evaluated in a numbered order (lowest rule number takes priority)
* Applies to all traffic entering & leaving the subnet.
* You must explicitly allow both inbound & outbound traffic for a connection.

Task:
1. Create VPC using vpc only
2. Create 2 subnets (one public and one private)
3. Create VPC using VPC & more
4. Create 2 subnets (one public & one private)
5. Launch 4 instances in each subnet.

Task:
1. Connect to private instance
2. Install java & git
3. Create 5 txt files.



sudo ssh -i "ms-keypair-2.pem" ec2-user@10.0.0.135

Steps to connect to Private instance

1. Create 2 instances, one in public sn & other one in private sn
2. Try to connect to private instance (not possible)
3. Now to connect to public instance
4. Create key pair file in public instance with same name.
5. go to downloads copy the content of keypair file & past it in this file
6. Select the private instance
7. Click on connect
8. Select ssh connection
9. Copy the cmd below example & paste it in public instance terminal.
10. Note → use sudo before pasting.
11. type yes to confirm connection
12. Now you connected to private instance through public instance

PAGE EDGE

DATE : 5/9/25

Friday

LAMBDA (2)

* It is one of the service in AWS
* It belongs to Compute Category
* Component of Lambda is Functions.

Different between EC2 and Lambda:

Application → Code
→ to run your code you need instance.

	EC2	Lambda
	Server/Instance	→ Run your code without thinking about any server or server configuration.
User	- OS - Type of instance - Configure storage - Scaling the resources	→ All these are taken care by AWS itself
	- It follows pay-per-hours	→ Serverless Compute Category → It follows pay-per-millisec

What is Lambda?

* Lambda is a serverless compute service, which is offered by AWS, where users are no need to worry about any server configuration and simply run their code.
* Run your code without thinking about any servers.

Feabures of Lambda.

1. Server less
2. Programming Language Support
3. It follows pay-per-millisecond

Programming Languages:
.NET, Java, Python, Ruby, Node.js, Custom Runtime

Pricing
2 Categories:
1. Request /month
2. GB-sec /month

1. Request /month
* First 1M request/month Free
* 1 program → create/run it for 1 time → 1 request
  1 program → Run it for 10 times → 10 request
  10 program → run 10 times each pgm → 10*10 = 100 requests

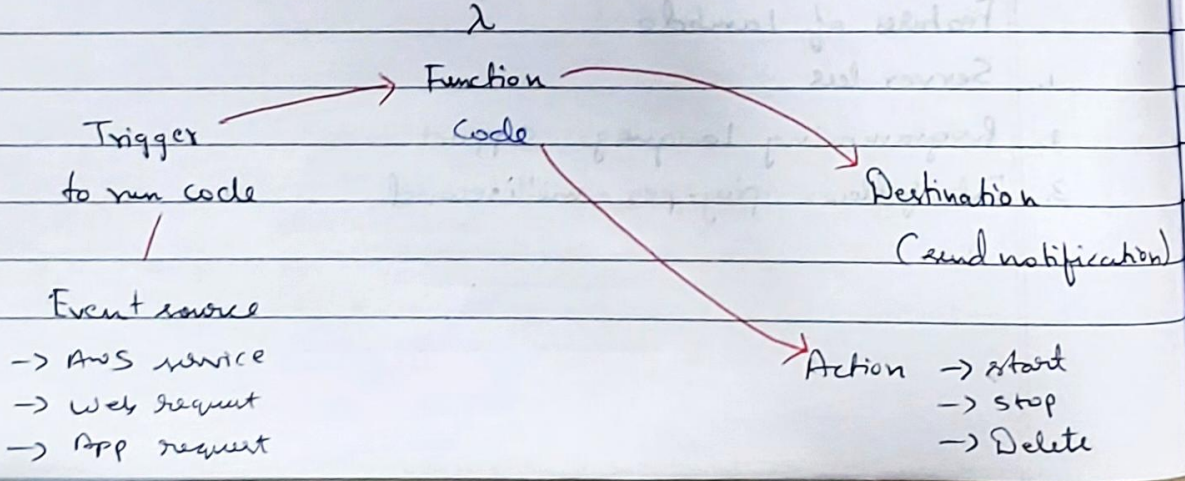
Note: $0.20 per 1M Requests/month

2. GB-sec/month
First 400K GB-sec/month Free
1 program:

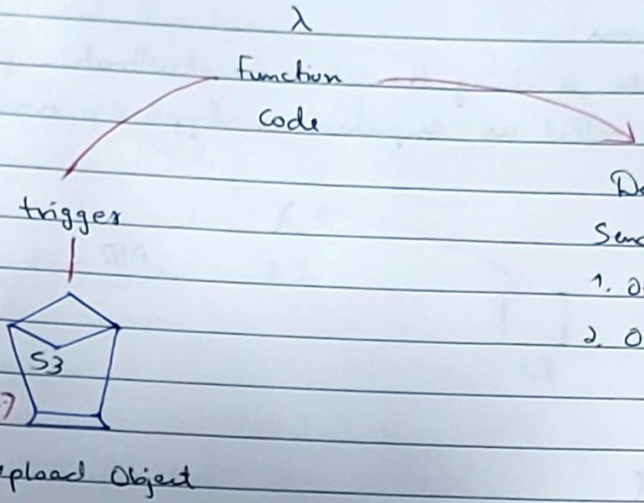
Storage Time
100mb * 10sec = 1000mb sec = 1GB-sec
400K GB-sec - 1GB-sec = 399K GB-sec

Note: $16.67 per 1M GB-sec/month

Structure of Lambda:



Eg



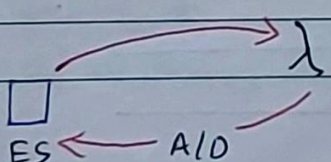
Ullas → Upload ✓
Kiran → Upload ✗

Type of event source
1. Push Based
2. Pull Based

1. Push Based
* Event source is triggering lambda function is called as Push Based

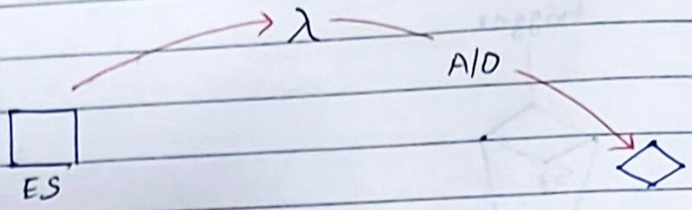
Types:
1. Synchronous
2. Asynchronous

1. Synchronous:
* If lambda is doing action or destination part on the same event source is called Synchronous type of Event source.



2. Asynchronous:

* If lambda is doing the action or destination part on the diff. thing is called as Asynchronous type of event source.



2. Pull Based

* Lambda will watch over the event source based on pattern.

B G B G B G B G

B G B G B B G G

Types of Pull Based

1. Queues
2. Streams.

1 Queues:

* Lambda pulls messages from queue like SQS, processing them as they arrive

1 2 3 4 5 6 7 8 9 ....

2. Streams:

* Lambda pulls records from a stream like Kinesis or DynamoDB streams, invoking the function with batch of records.

Saturday

Steps to Create Function:

Function 1: (Simple Code)

1. Navigate to lambda service
2. Click on Function
3. Click on create function
4. Choose Author from scratch
5. Give function name
6. Choose Runtime
   → Python
7. Click on create function

In Function

→ In code section → source code → click on test (to execute code)
→ Configure test event → click on create test event →
Give event name → Click on save

Note: If any modification is done to the code, you need to update it by clicking on deploy. (Deploy)

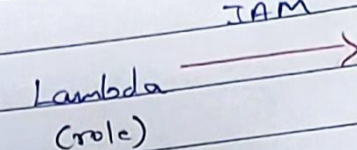
Function 2: (Stop instance)

1. Navigate to lambda service
2. Click on Function
3. Click on create function
4. Choose Author from scratch
5. Give function name
6. Choose Runtime
   → Python
7. Click on create function

In Code Section

→ You need to copy & paste the stop instance script (script 2) in lambda_function.py
→ Modify by mentioning region code & instance id.

→ Click on Deploy
→ Click on test (event)
→ error (unauthorized)
→ Attach IAM to the lambda function



Steps to create IAM roles and pp and also steps to attach it
→ Go to IAM service
→ Create Role for lambda
→ Create pp for EC2
  → Actions allowed
    → Stop Instance
  → Resources
    → All or specific by choosing specific instance.

Steps to Attach IAM role to Lambda Function:
1. Go to created function.
2. Click on Configure
  → General configuration
    → Click on edit
    → Keep everything as it is
    → set timeout if required
    → In Execution role
      → Choose use an existing role
    → In Existing role
      → select created role
  → Click on save.

Now test the code.

Function 3: Start Instance:
→ Copy & paste start instance script (script 2)
→ Modify by mentioning region code & instance id
→ While creating IAM permissions, select Action as 'Start Instances'

Function 4: Start_Stop_Instance
* It will describe whether instance is stopped / running state, if if is stopped state it will start instance & if it is running state it will stop instance

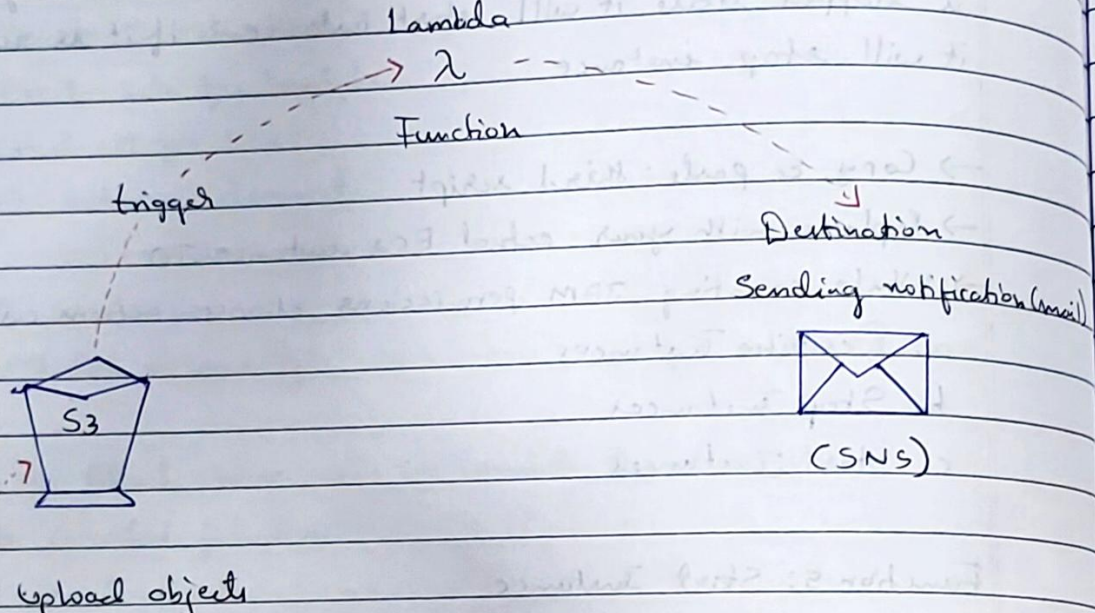
→ Copy & paste third script
→ Replace with your actual EC2 instance ID
→ While creating IAM permissions choose actions as
  a. Describe Instances
  b. Stop Instances
  c. Start Instances

Function 5: Start Instance
→ If multiple instances are in stopped state, lambda function will start those instances
→ Copy & paste fourth script
→ # Replace with your AWS region, e.g., 'us-east-1'
→ While creating IAM permissions choose actions as
  a. Describe Instances
  b. Stop Instances
  c. Start Instances

Function 6: Stop Instances
* If multiple instances are in running state, lambda function will stop those instances.

→ Copy & paste fifth script
→ # Replace with your AWS region, e.g., 'us-east-1'
→ While creating IAM permissions choose actions as
  a. Describe Instance
  b. Stop Instance
  c. Start Instance

Function 7: (Scenario 1)



Steps:
1. Create S3 bucket
2. Create Lambda function
3. Create SNS topic

SNS

SNS
* This is also one of the service in AWS
* It stands for Simple Notification Service
* It belongs to application integration category
* It is used to send notification

SNS

* The components of SNS are "Topic" & "Subscription".

→ Steps to create SNS Topic
  → Go to SNS service
  → Click on topics
  → Click on create topic
  → Select type as Standard
  → Give topic name
  → Click on create topic

→ Steps to create Subscription
  → Go to created topic
  → In subscription section
  → Click on Create subscription
  → Select the topic
  → Select email as protocol
  → enter mail id for endpoint
  → click on create subscription

Note: * Mail holder should confirm to add subscription.

→ We need to attach S3 and SNS to Lambda function
  → Go to created function
  Steps to Add trigger
  → click on Add trigger
  → Select source as S3 for trigger configuration
  → Choose bucket
  → Choose event type as PUT
  → Check box for Recursive invocation
  → Click on Add.

Steps to Add destination:

→ Click on Add destination
→ Select source as Asynchronous invocation
→ Select condition as On success
→ select Destination type as SNS topic
→ Choose destination ARN
→ Click on Save

Function 8:

Canary Function:

* It is nothing but blueprint
* It is used to visit and check the content of a website and also it is used for scheduling and event.

Steps:

1. Create Cloud Watch (Event Bridge) → configure it together using blueprint
2. Create function
3. Create Topic and add subscription

→ Go to Lambda Service
→ Click on create function
→ Choose "Use a blueprint" option

→ Search for blueprint name as Canary
→ Select the schedule a periodic check of any URL
→ Give Function name
→ In Execution role
→ Create a new role with basic Lambda permissions
→ In Event Bridge (CloudWatch Events) trigger
→ Rule

→ Choose create a new role
→ Give rule name
→ Rule description
→ Optional
→ Rule type
→ Select Schedule expression
→ Schedule expression
→ rate (1 minute) or rate (2 minutes)

→ In Environment variables
→ Enter key and value
→ Enter website link (value) in value place in front of site (key)
→ Enter content (value) of the website in front of expected (key)
→ Click on create function

Note:
→ While creating function itself we are configuring event source /trigger (Cloudwatch) by using blueprint.
→ But we need to add destination after creating function
—

→ Add destination as SNS topics to the function.

10/9/25
Wednesday

Cloud Watch

Cloud → over the internet
Watch → Monitoring the resources

* This is one of the service in AWS.
* Using this service we are going to monitor the AWS resources
* This service belongs to Management and Governance category
* Using this service we are not only monitoring resources and also we are monitoring applications.

Components of Cloud Watch:

1. Metrics
2. Dashboard
3. Alarms
4. Events

1. Metrics:

Milk → ml, litre (volumes)
Sugar → grams, kg
Eggs → numbers, dozens

* In AWS each and every resources are having their own metrics
* Using metrics in Cloud watch anyone can monitor any particular resources with the help of particular metrics.

2. Dashboard:

* In Dashboard we can see the multiple metrics of the resources in the Cloud Watch.
* It is the place where one can monitor the resources which they have created

Types of Dashboard:

1. Automatic Dashboard (AWS will provide dashboard)
2. Custom Dashboard (User can create their own dashboard)

* Whenever we are creating the dashboard, we have full customization for selecting on what metric of the resources you want to create
* Also you need to select widget type based on your need.
  Eg. Line, Pie, Bar, Gauage etc.,

Steps to create Custom Dashboard:

→ Go to CloudWatch service
→ Click on Dashboards
→ Choose Custom Dashboard
→ Click on Create Dashboard
→ Give name for Dashboard
→ again click on Create Dashboard
→ Select widget which you prefer
→ Click on next
→ Choose the resources based on metrics which you want to monitor
→ Click on create widget.

3. Alarms:

Clock alarms:

time (metric for clock alarms) → Actual time 5:00 am = Set Alarm 5:00 AM (Condition) → Start ringing (Action)

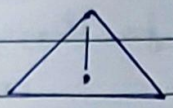
Eg for AWS services

→ Consider EC2

CPU utilization (metric for EC2) → <40% or >40% Condition → Stop Instance(s), Reboot Instance(s), Terminate Instance(s) Action

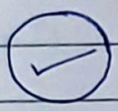
Alarm States:

1. In Alarm



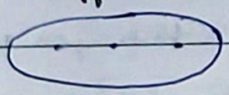
→ When your alarm is in active or is in working condition

2. OK / Success



→ Resources working perfectly

3. Insufficient data



→ If data is insufficient

Steps to Create Alarms

→ Go to CloudWatch Service

→ Under Alarms

→ Click on In Alarms

→ Click on Create Alarm

→ Click on Select metric

→ Search for resource (EC2 instance) based on metric (CPU utilization)

→ Click on select metric

→ In Metric Section

→ Select statistic

→ Set time period

DATE : / /

→ In condition section
  → Select threshold type as static
  → Whenever CPU Utilization is.. (Define alarm condition) → lower
  → than.. (mention value → 40)
→ Click on next
→ If you want to get notification you can do configuration or if not configuration is not required.
→ Search for EC2 action
  → Click on Add EC2 action
  → Choose alarm state as In Alarm
  → Select the EC2 action
→ Click on next
→ Give alarm name
→ Click on Next
→ Check preview and create for verifying metric, condition, action and name.
→ Click on create Alarm.
→ Click on view alarm to view created alarm.