AWS - CLOUD

* Cloud - Network or internet, which is present at remote location & can provide the Services over network.
  - A way to store & access data or programs over the internet instead of your computer's hard drive

* Cloud Computing - It is like renting compute power & storage online. Instead of owning & maintaining your own hardware

* Deployment - putting your application or website on the AWS cloud so people can use it online

* Types of cloud models.
  1) public cloud (Shared resources open to Every-one)
  2) private " (dedicated resource of organization)
  3) Hybrid " (a mix of public & private)

* Types of cloud Service models
  1) Iaas → pizza made at home from Scratch
  2) paas → Ready made pizza add toppings
  3) Saas → order & eat

Iaas (learning servers)
- Application
- Data
- runtime
- Middleware
- O/s
- Virtualization
- Servers
- Storage
- Networking.

Paas (code)

Saas

* Virtualization = It allows multiple Operating System or applications to run on a single physical machine, improving Efficiency, resource utilization & flexibility.

Note: Virtualization can exist without cloud computing but cloud computing cannot exist without virtualization.

* Hypervisor :→ Software that Enables multiple Operating System to run on a single physical machine

* Types :
  1. Type 1 (Bare metal) → Installs directly on Hardware.
  2. Type 2 (Hosted) → Installs on top of an existing Operating System

- AWS achieves virtualization using the XEN hypervisor
- Every AWS EMR uses the bare metal Xen hypervisor.
  ↓
  2-types
  1. Hardware Virtual Machine (HVM)
  2. Paravirtualization (PV)

* AWS Suite Consists of
  1. Storage
  2. Databases
  3. Networking
  4. Compute
  5. Monitoring

* ~~Regions~~

* Module-2

1. Regions & Availability Zone

2. EC2 (Elastic = AMI)
   - Launch instance → Connect
   - AMI
     - free 3 micro
     - key value pair
     - Create
   - EC2 instance Connect
     - Connect (another method)
     - EC2 ⇒ copy public IP address = paste cmd = & type some commands
   - Linux = ssh As key pair download one file
     - cd download
     - ssh -i Server.pem ec2-user paste public IP
   - Windows = RDP

Exercise → Launch & connect to windows EC2 instance

Same

↓

AMI (Windows)

Connect → RDP client → remote desktop protocol

↓

download desktop (if required)

↓

Get password

↓

upload the .pem file

↓

decrypt password

↓

open desktop file, now paste the password

- While creating a key pair it downloads the automatically

* Key pair = used to securely connect to your instance

* Amazon Machine Image (AMI)
  It is a template that contains the software configuration (operating System, application server, applications) required to launch your instance.

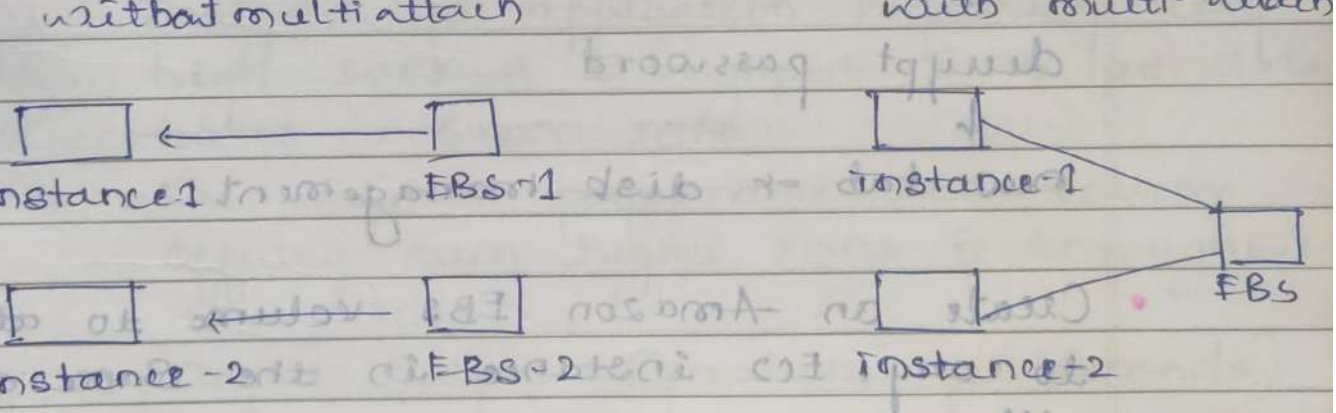
* Amazon EC2 = allows you to create virtual machines, or instances, that run on the AWS cloud

Regions = The geographical location where AWS data-centers are present
↳ mumbai - ap-south-1, -2, -3

* Instance type = t3.micro
↳ Select an instance type that meets your computing, memory, networking & storage needs

* Firewall (Security groups)
- A Security group is a set of firewall rules that control the traffic of your instance
- Add Specific rules to allow Specific traffic to reach your instance.
- SSH traffic = helps to connect to your instance
- HTTPS = when creating a web-server
- HTTP = " "

* Elastic Block Store =



* EBS Snapshots = enable backup & restoration

Elastic block Store is a Scalable, high performance block Storage Service in AWS. It is primarily used with Amazon EC2 instances to provide persistent storage for applications

↳ create storage volume & Attach them to EC2

Focus: Creating, Attaching & Managing EBS Vol

Create Instance
↓
Volume
↓
Create
↓
Select → attach (az - of EC2 & volume) should be same eunorth-1a - "1a

Go to instance = Select → connect → termin
- command lsblk
↓
10G xvdb

- While modifying volume = GB can be increased not decreased

- Windows
↓
decrypt password
↓
new tab → disk management

- Create an Amazon EBS volume to attach to any EC2 instance in the same availability zone

* Tags :→

- A tag is a label that you assign to an AWS resource
- tag consists of a key & an optional value

local storage solution for Amazon EC2

* EBS Volume :

durable, block-level storage device that you can attach to your instances.

- you can attach multiple EBS to single instance but it should have same availability zone

* Types of volumes :→ solid state drive

1. General purpose SSD (gp2 & gp3)
2. provisioned IOPS SSD (io1 & io2)
3. throughput optimized HDD(st1), Cold HDD(sc1), magnetic (Standard)

→ grand prix 2

- gp2 = provides a balance of price & performance for a wide range of workloads
- gp3 = provides 20% cost savings compared to gp2

- io1 = Designed for application requiring very high IOPS & throughput, such as database - bases = 64000 IOPS
- io2 = The highest performance EBS volume, offering even higher IOPS & throughput than io1

IOPS (Input Output Operations per Seconds)

Create & Manage -

Create instance = connect

terminal = sudo yum install nginx -y

sudo systemctl status nginx

" " start "

enable "

nginx

* instance → actions → template & image →
Create → provide name → Done

* Go to AMI → Create → Select → launch EC2
with AMI - now go to instance the
same o/p displays
Welcome to Nginx

* Go to AMI
↓
Select (copy AMI)
↓
Now Select the region
↓
the same AMI created will be at
that region.

* Can't create AMI directly → so create instance
Select → Create template → it displays in
AMI - then from here you can launch
AMI with EC2
↓
* Add: 8 gp
15 gp
↓
if not error

Exc-5 Create Amazon EBS Snapshots

Create instance

↓

It will automatically get created in 'Volumes"

↓

Select the volume & click Create Snapshot

↓

Go to Snapshot & click Create volume.

* EBS Snapshot = point in time copy of your data & is used to recover the data
  - provides replica of the data

Ex. If an EC2 instance crashes you can create a new volume from the Snapshot & attach it to a new instance.

- you can copy Snapshots to another AWS region
- If one AWS region goes down, you can restore your application in another region

* Lecture - 4 EFS = Elastic File System

* File-System →

method used by operating Systems & Software to organize & store data on storage devices such as hard-drives, Solid-State drives (SSD) or network storage.

* Elastic File System :→

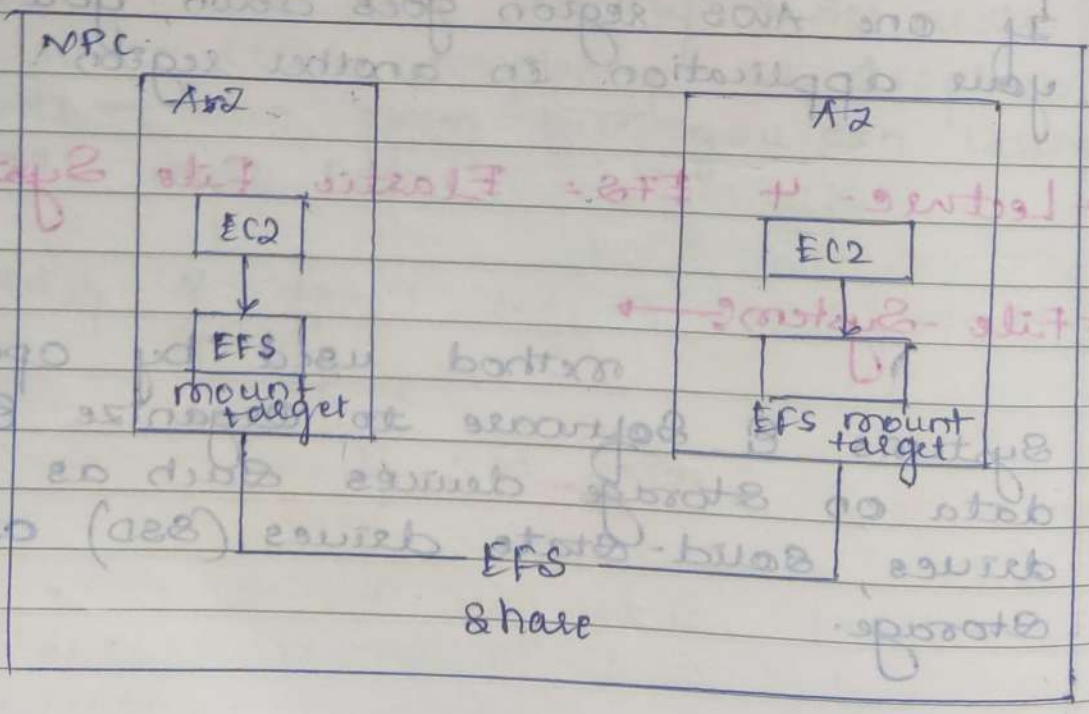
It is a cloud based file storage service that automatically scales as you add or remove files

- EBS which is connected to only one EC2 instance at a time
- EFS can be shared across multiple EC2 instances at once

It is fully managed by AWS, you don't have to worry about setup, scaling, or maintenance

* Mount target :→

network endpoint that allows EC2 instances to connect to an EFS file system within a specific VPC & AZ.



* EFS deployment Options

EFS allows you to choose b/w two deployment options. Regional & one zone

Regional	One zone
• Better Availability	• Cost-effective
• Designed for durability	• Deployed within Single AZ
• High throughput & low latency access	• low latency within Same AZ

* Features of EFS

Scalability
Share file Storage
Fully managed
high availability
Encryption

* Disadvantages of EFS

- Each az needs a Separate mount target which can make Setup more complex
- EFS only Supports NFSv4.1, limits compatibility with Some applications

Exp 6: Create your Amazon EFS file System & mount to EC2 instances.

(Nfs-common)

Create Instance in ubuntu → 2-instances → EFS file System → Connect Each instance Separately → In one terminal write the

Commands as
Sudo Su
apt-get update
apt-get install nfs-common -y
⇓
Same commands in other instance
⇓
Now go to created EFS
⇓
Network
↓
3 & check that is available (Mount targ
↓
Attach (Copy the sudo command of nfs)
↓
paste it on terminal.

Lecture 5 Amazon FSx

It is a fully managed file storage service that provides high-performance file systems compatible with Windows & Lustre

- It offers features like automatic backups, data duplication & integration.

* Features of FSx:-
1. Fast delivery
2. Highly available
3. Secure & Compliant

* Types of Amazon FSx filesystem

1) Amazon FSx for NetApp ONTAP
2) " " Open ZFS
3) " " Windows file Server
4) " " Lustre

MODULE - 3

L1) Elastic load Balancer (ELB)

Automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers & IP addresses, in one or more availability zones.

* Target group :→

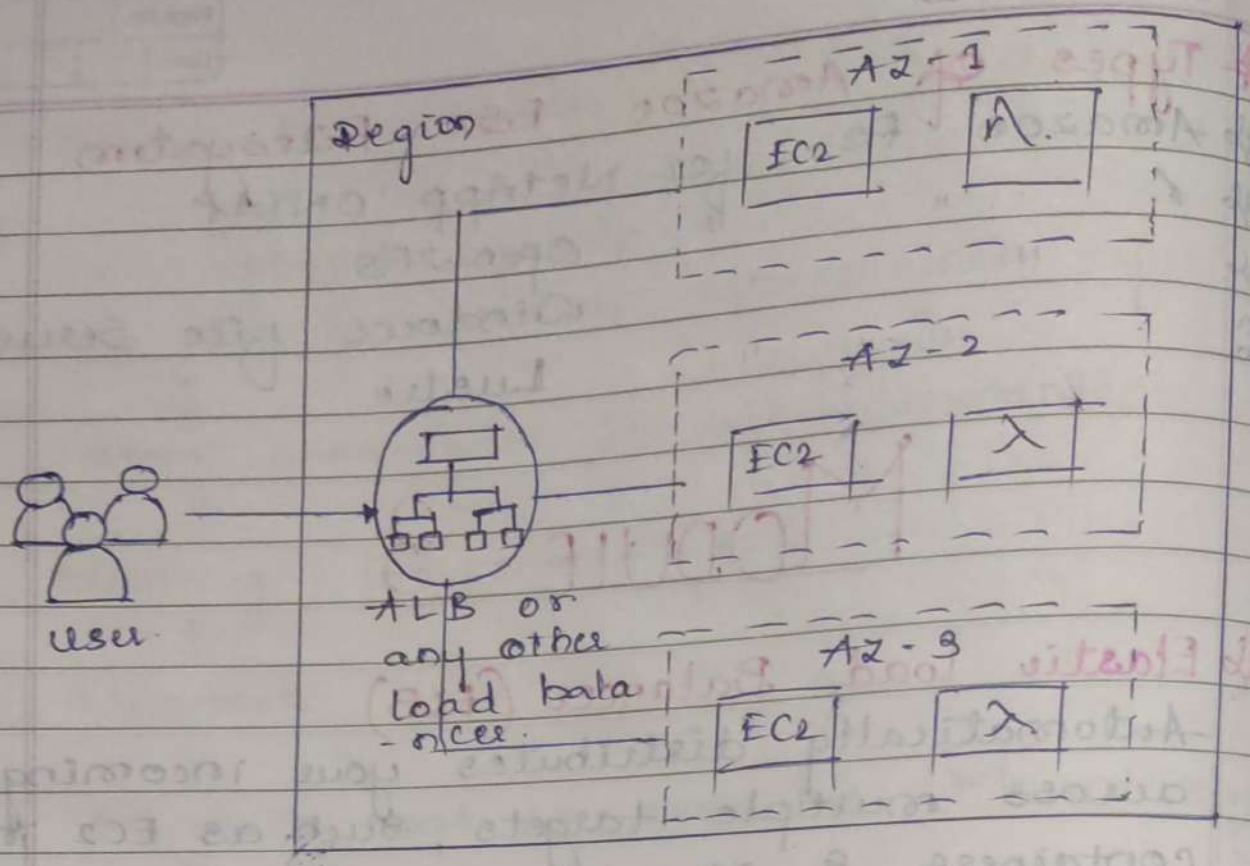
It is a logical grouping of backend servers (EC2, IPs / lambda) that receive traffic from an AWS ELB

* Cross-zone load Balancing :

To distribute traffic evenly across AZ.

* Security in ELB

- Security groups acts as virtual firewalls to control inbound & outbound traffic
- you must configure Security groups for both the load Balancer & EC2 instances.



Types of load Balancer

1. Application load Balancer
2. Network " "
3. Classic " " (1st type)
4. Gateway " "

1. Classic load Balancer (Support IPV4 & IPV6)

- Works at layer 4 & layer 7
- Supports HTTP, HTTPS, TCP & SSL
- Requires manual scaling

Use

- Only if you have legacy applications
- If you don't need advanced routing feature

2. Application load Balancer

- Works at layer 7 (Application layer)
- Supports path-based & host-based routing
- We can route traffic to EC2, IPS, lambda, ECS

Use
- If you need path-based & host-based routing
- If your application uses websockets

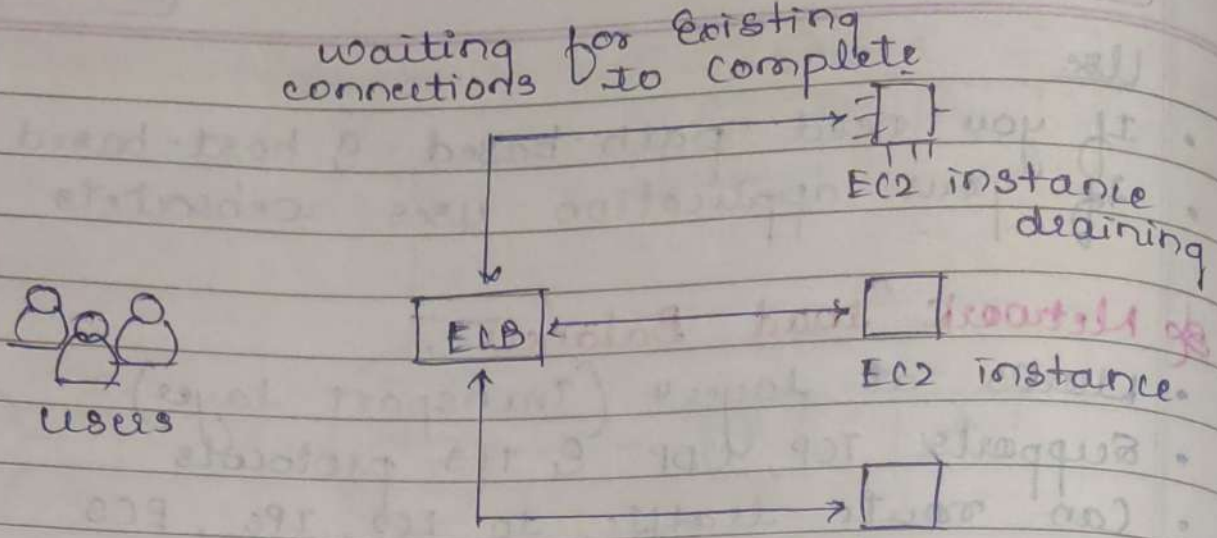
Network load Balancer
- Works at layer 4 (Transport layer)
- Supports TCP, UDP & TLS protocols
- Can route traffic to EC2, IPs, ECS

Uses
- If your app uses UDP or non-HTTP traffic

Gateway load Balancer
- If you need basic global traffic routing
- " regional load balancing
- It distributes network traffic across multiple regions worldwide
- It connects internal networks (like a VPC) to External ones (Internet)

* Sticky Sessions :→
Allow load balancers to bind a users session to a specific target, Ensuring requests are consistently routed to the same instance for a continuous experi-ence

* Connection draining :→
AWS ELB ensures that when an EC2 instance is removed, it finishes processing ongoing requests before shutting down. This prevents dropped connections & ensures a smooth transition during scaling or updates



Exc-1 Create a Target-Group

Go to EC2
↓
Target Group Create → Instance / Lambda / IP address / ALB
↓
Instance
↓
TG name
↓
Create

Exc-2 Launch a Application Load Balancer

EC2
↓
load Balancer → same with NLB
↓
Select ALB
↓
provide name
↓
internet facing
↓
Select target group

Select instance
↓
Create
Go to Instance ⟶ Connect → Suda Su
apt-get install apache2 -y

* Internet facing
- Serves internet-facing traffic
- Has public IP-address
- DNS name is publicly resolvable
- Requires a public Subnet

Internal
- Serves internal traffic
- private IP
- DNS is publicaly resolve.
- Compatible with the IPv4 & Dual-stack IP address type

- Security group = Set of firewall rules that control the traffic to your load Balancer.
- Listner = process that checks for connection requests using the port & protocol you configure

Exc-4 Enable Sticky Session
Created load Balancer
↓
Select → Scroll down
↓
Listeners = Default action
↓
Attributes = Edit
↓
Enable Stickiness

* Stickiness = users requests always go to the same backend server instead of switching b/w different servers

Lecture-3 Amazon Route 53 ↑ It connects user requests to internet applications

It is a DNS (Domain Name System) service that connects domain names (like example.com) to IP address of websites

* Use Cases
- Manage network traffic globally
- Build highly available applications
- Set up private DNS

* Working with hosted zones
A container for records, which include information about you want to route traffic for a domain (such as intellipaat.com) & all of its subdomains (such as www.intellipaat.com), lms.intellipaat.com, & accounting.intellipaat.com)

- you can create both public & private hosted zones in Route 53
- when you create public hosted zone, route 53 automatically generates NS & SOA records

* Routing policy :→ It determines how traffic is directed among multiple resources, such

as by geography, latency, health status or weighted distribution, to optimize response times & resource availability

* Types of routing policy

1. Simple routing
2. Failover " 6. Geo proximity routing
3. Geo-location " 7. IP based
4. Latency "
5. Weighted "

* Types of records

1. CName records
2. SOA "
3. NS "
4. Alias "
5. A "

Exc 5: Register a domain with Route 53

Route 53
↓
Registered domain
↓
pay amount → provide information & create

Exc 6: Create a configured hosted zone

Route 53
↓
hosted zones
↓
Create ~~record~~ hosted zone
↓
fill details - (Select the domain u purchased) - OK

Create record
↓
provide Subdomain name
↓
Route traffic to
ALB
Mumbai
↓
Create

Go to registered domain which created
↓
click on it
↓
Actions - Edit name Servers
↓
Copy the name Servers from hosted zone & paste here
↓
Save

- To check: copy domain name & browse. It works backend.

Lecture-5 Amazon EC2 Auto-Scaling

Auto scaling group (ASG) automatically adds or remove EC2 instance based on traffic or demand.

- Monitors traffic or CPU usage
- Adds more instance when traffic is high
- Remove extra instance when traffic is low
- Replaces unhealthy instances automatically

* Launch template:

It is a reddy made setup for creating EC2 instances. Instead of choosing settings every time, you save them once & reuse them.

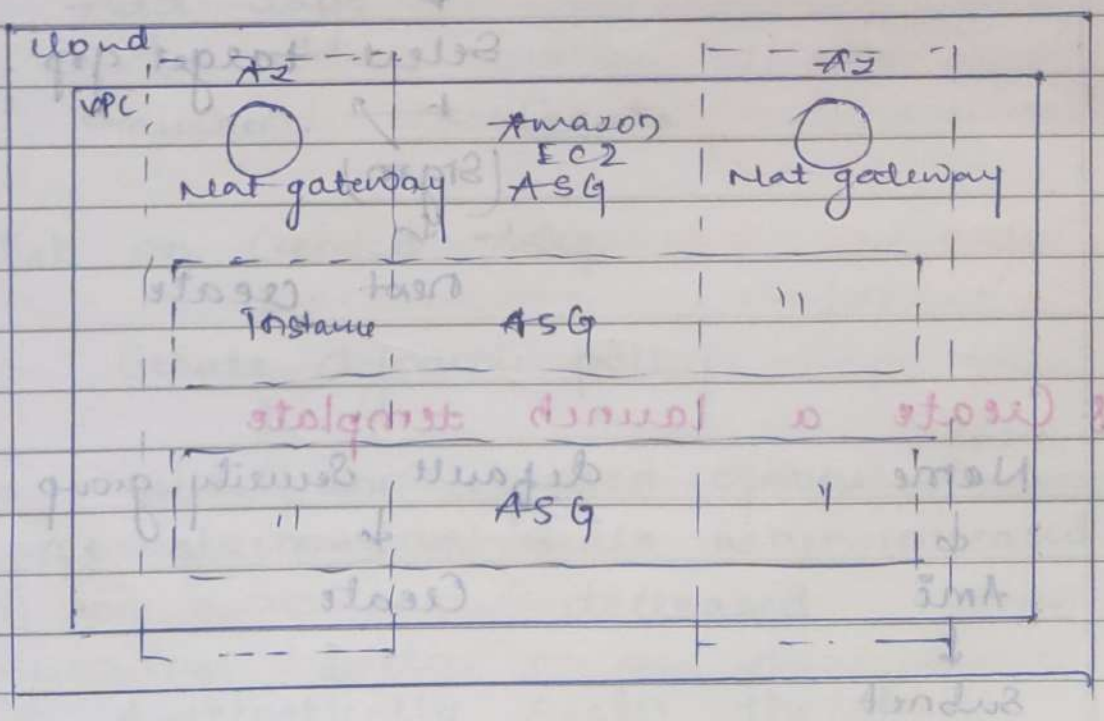
* What does it include?

- AMI
- Instance type (t2.micro)
- Security Setting
- Key pair
- Startup Scripts

* Launch configuration

Older method Same as launch template but once created cannot be modified.

* ASG = collection of EC2 instance managed as a group for automatic Scaling & management.



Exc-7 Enable path-based Routing

EC2 instance
↓
load Balancer
↓
Target Group

load balancer ← If I created demo as target click on it as the traffic present
listners add rule
↓
Name & tag
↓
Sigoinforward - next
↓
Condition = path
↓
/Signin/
↓
Confirm
↓
Select target grp
↓ ↗
(Signin)
↓
next create

Exc 8 Create a launch template

Name
↓
Ami
↓
Subnet

default Security group
↓
Create

- Launch template is used to automate & standardize the creation of EC2 instances. Instead of manually setting up instances every time, a launch template saves the configuration so that EC2 instances can be launched quickly & consistently.

Exc 9: Create a Auto Scaling Group

ASG → Add all details

↓

Choose launch template

↓

Choose instance launch options

↓

Integrate with other Services

↓

Configure Group Size & Scaling

↓

Add notifications

↓

Add tags

↓

Review → Create

↓ Click on Created ASG

↓

Create dynamic policy

↓

Select the required options as alarm → Scale when increased or decreased

↓

Automatically scales the traffic

Module - 4

Lec - 1 Virtual private cloud.

private network in AWS where you can run your EC2 instances, databases, & other resources securely.

* Use
  - Only you control access
  - public & private subnets for better security
  - Can allow or block traffic as needed.
  - Can use multiple availability zones

* VPC Components
  * Subnets, CIDR, internet gateway, nat gateway, Security groups, NACL's.
  - VPC is single region but multi-AZ

* Subnets - Smaller sections of the VPC

Lec - 2 Subnets & Routes

↓

* Smaller sections within a VPC where devices can communicate directly.

- Types of Subnets in a VPC
  1. private
  2. public

- public Subnet = The subnet has a direct route to an internet gateway.
- private " = The subnet does not have a direct route to an internet gateway. Resources

in a private Subnet require a NAT device to access the public internet.

* Route tables :→
Define rules for directing traffic between Subnets & the internet gateway, allowing for efficient network communication within a VPC.

- Route table contains a set of rules called routes, that used to determine where network traffic from your VPC is directed.

* Internet Gateway
Components in AWS that allows resources in a public Subnet to send & receive traffic from the internet

* Works
- Attached to VPC
- Works with route table to allow traffic
- Enables EC2, load Balancers, to access the internet
- public IPs are required for instances to be reachable

* Types of IPs.
1. public IP - Automatically assigned to EC2 instances when launched in a public Subnet. It allows instances to communicate over the internet

2. private IP - Assigned to EC2 instances within a VPC. Used for internal communication within the VPC & cannot be accessed from the internet.

3. Elastic IP = A static IPV4 address designed for dynamic cloud computing. It can be associated with EC2 instances or network interfaces & can be moved b/w instances

↳ valuable for scenarios requiring consistent public IP addresses, such as hosting websites, applications, remote access or setting up secure connections

Lec-9 = Gateway of VPC.

↳ to control traffic & connectivity

→ Types of Gateways

1. Internet Gateway
2. NAT "

1. Internet Gateway = Allows public instances to access the internet inbound & outbound.

2. Nat Gateway = Allows private instances to access the internet Outbound only.

* NAT Instances

Act as intermediaries b/w resources in private Subnets & the internet. They Enable outbound internet connectivity for instances within a VPC while keeping them hidden from inbound traffic.

Lec 4: Security Group & NACL's

↓ (Stateful)

act as Virtual firewalls for EC2 instances, controlling inbound & outbound traffic

- NACL (Network Access Control Lists)
  - It is Stateless, rule-based firewalls that control traffic at the Subnet level.
  - provides Security to Subnet level

Exc-1 Create Custom VPC & its resources

Create VPC
↓
Subnets (private & public)
↓
Create routes
↓
modify private Subnet to private routing
public " public "
↓
Create Internetgateway = Attach to VPC
↓
Route table
routes Edit routes = Target internet gateway

IGW
↓
Routing
↓
Subnet

Lec 2 Working with NAT Gateway.

To configure this we need Elastic IP, public Subnet & VPC

- Nat gateway is always created in public Subnet.

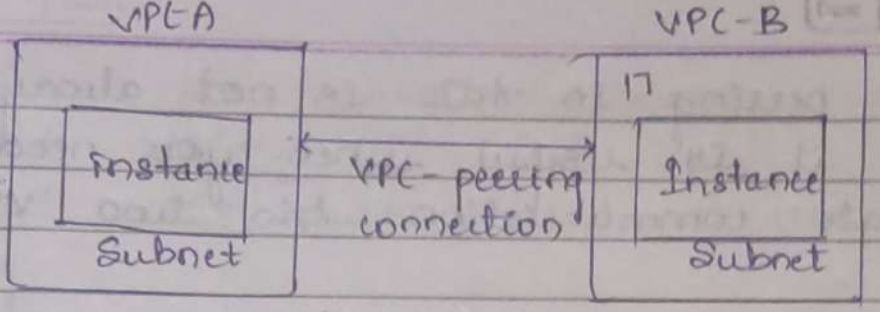
Name
↓
Subnet (The created public Subnet)
↓
public
↓
Allocate Elastic IP

Go to Created private Subnet
↓
Route
↓
Add rule
↓
Select 0.000/0 Created nat
↓
Create

Lec-6 VPC peering

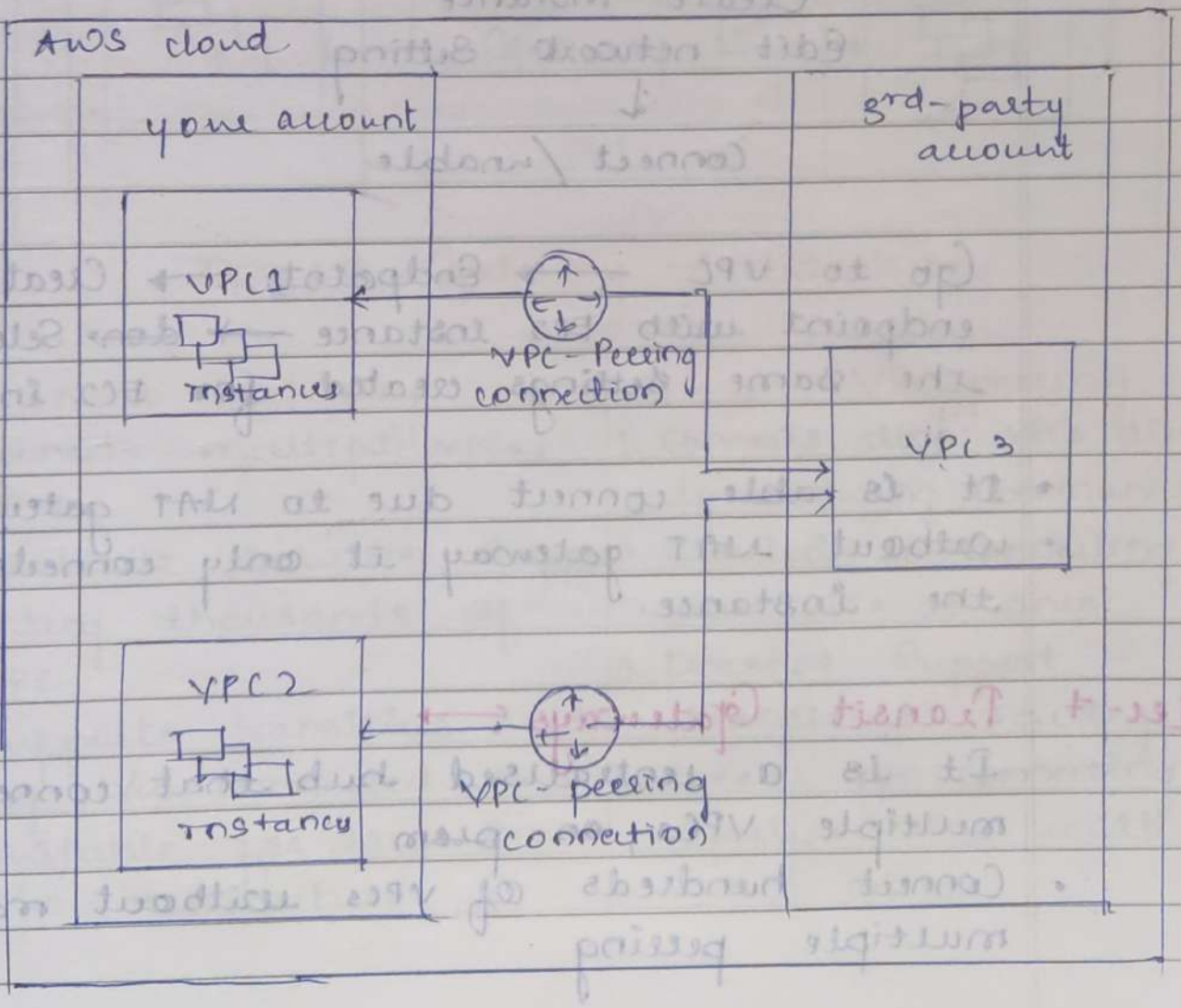
In AWS allows connecting two virtual private clouds (VPCs) within the same region or Separate region.

- It Enables communication b/w resources in the peered VPCs using private IP address as if they where on the same network



* Vpc peering limitations

1. There is a quota on the number of active & pending VPC peering
2. you cannot have more than-1 VPC peering connect -ion b/o two VPCs
3. you cannot create a VPC peering b/o VPCs that have matching or overlapping IPV4
4. VPC peering does not support transitive peering relationships.



- VPC peering in AWS is not always required but it is useful when you need direct, private communication b/w two VPCs

* When to use VPC peering
1. If you have two VPCs that need to communicate without using public internet
2. If you have resources in different regions that need to interact securely.
3. VPC peering avoids the need for NAT gateways or VPNs reducing costs & improving performance.

Exc-8 Configure VPC peering.

Create instance
Edit network setting
↓
Connect / unable

Go to VPC → Endpoint → Create → endpoint with EC2 instance → don't Select all the same settings created for EC2 instance

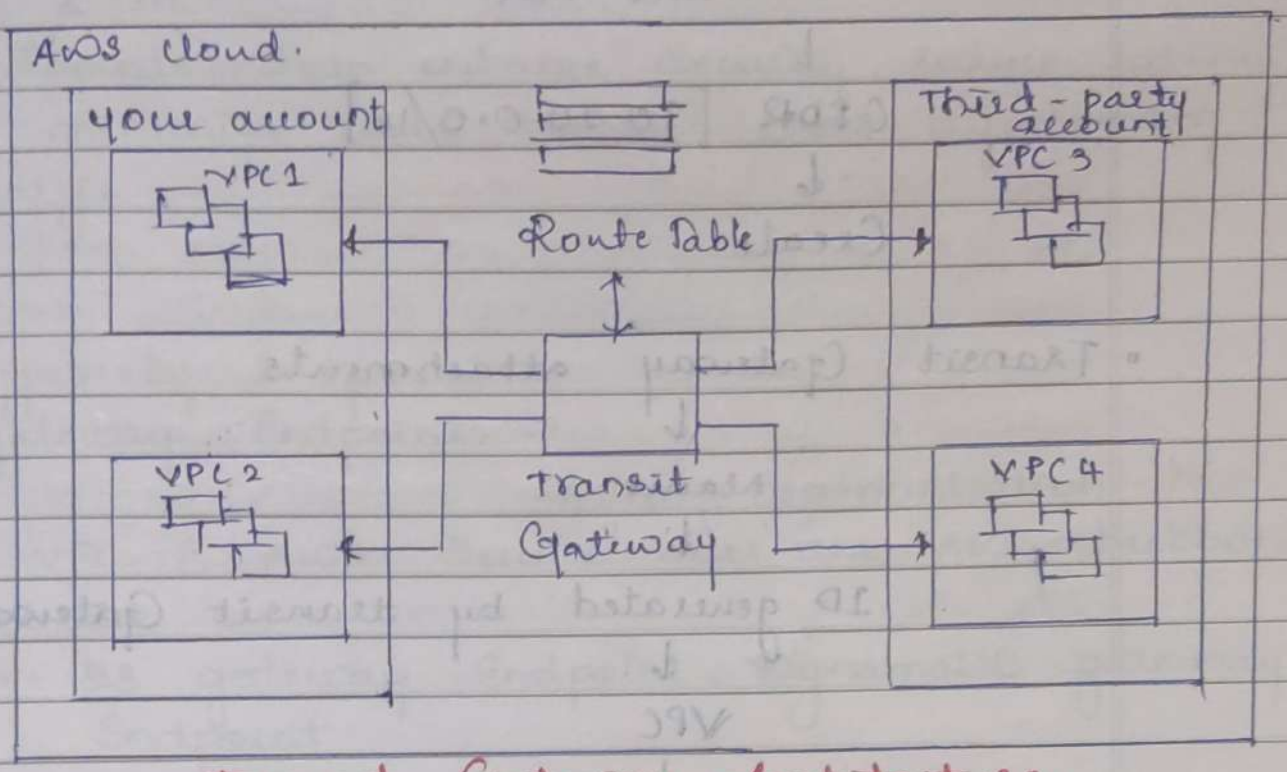
- It is able connect due to NAT gateway
- without NAT gateway it only connects with the instance.

Lec-7 Transit Gateways :→

It is a centralised hub that connects multiple VPCs, on-prem
- Connect hundreds of VPCs without managing multiple peering

* Use Cases for transit Gateways

1. Isolated VPCs
2. Isolated VPCs with shared Services
3. peering
4. Centralized outbound routing
5. Centralized router



Transit Gateway Architecture.

Transit Gateway

1. Connects multiple VPCs, VPNs
2. Scalable Solution Supporting thousands of VPC
3. Supports transitive routing b/w connected n/w
4. Suitable for large scale n/w architecture.

VPC peering.

1. Connects two VPCs directly allowing Communication
2. Limited to connecting two VPCs at once
3. Doesnot Support transitive routing
4. Ideal for connecting Specific VPCs with each other

Exc-4 Transit Gateway peering:

VPC

↓

Transit Gateway

↓

ASN (info) ⟶ number copy paste

↳ click here

↓

CIDR [10.10.0.0/16]

↓

Create

- Transit Gateway attachments

↓

Name

↓

ID generated by transit Gateway peering

↓

VPC

↓

Create

↳ Same process with different VPC

- Check = Transit gateway route = check if 2-VPC is associated

url of the entrypoint for an AWS.

VPC

Lec-8 Endpoints :→

Endpoints in AWS are virtual devices that allow private connectivity to AWS services without requiring traffic to traverse the internet. They serve as entry points for accessing AWS resources such as S3, DynamoDB or other services within a VPC.

- Endpoints help enhance security, reduce latency, & minimize data transfer costs by keeping traffic

* Types of Endpoints

1. Gateway Endpoints -

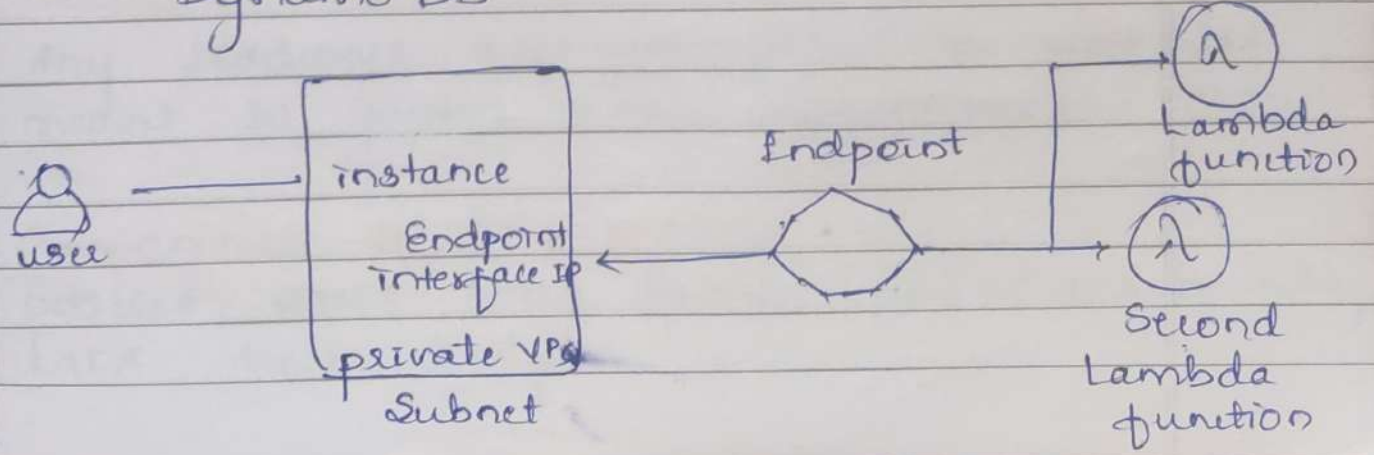
Allow communication b/w a VPC & AWS Service over the AWS backbone network

Ex = S3 gateway Endpoint, DynamoDB gateway Endpoint

2. Interface Endpoints -

Enable private connectivity to AWS Services using Elastic network interface within a VPC.

Ex = Interface VPC endpoints for S3, Dynamo DB



Lec-9 What are the VPC flow logs?

VPC flow logs capture information about the IP traffic flowing into & out of network interfaces in a VPC

- They provide insights into traffic patt help troubleshoot connectivity issues, & enhance security by monitoring & analyz network traffic at the packet level.

- VPC reachability Analyzer
  It is a configuration analysis tool that allows you to perform connectivity testing in your virtual private clouds between a sour resource & a destination resource
- Reachability Analyzer generates hop-by-hop details of the virtual network path between the source & destination when the destinatio is reachable.

MODULE-5 (IAM & Security)

lec-1 AWS Identity & Access management

- It is a web-service that helps you securely control access to AWS resources. With IAM, you can manage permissions that control which AWS resources user can access.
- you use IAM to control who is authenticated (signed in) & authorized (has permissions) to use resource.
- IAM provides the infrastructure necessary to control authentication & authorization for your AWS accounts

* Identities :→

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services & resources in the account. This identity is called the AWS account root user.

* Access management :→

After user setup in IAM, they use their sign-in credentials to authenticate with AWS,

* Security Control :→

Any hardware, software, policy or procedure meant to govern access to resources

* Components of AWS IAM →

policies, users, roles, Groups, permissions boundary MFA, Access analyzer

* IAM groups:
Groups are collections of IAM users. Instead of assigning permissions to individual users, permissions can be assigned to groups, simplifying permissions management.
- Users can belong to multiple groups, inheriting the permissions associated with each group.

Lec-2 IAM roles & policies

* What are IAM policies?
- IAM policies are JSON documents that define permissions & access control rules.
- policies can be attached to users, groups or role to specify what actions are allowed or denied on which AWS resources.
- policies can be custom-defined or pre-defined AWS managed policies

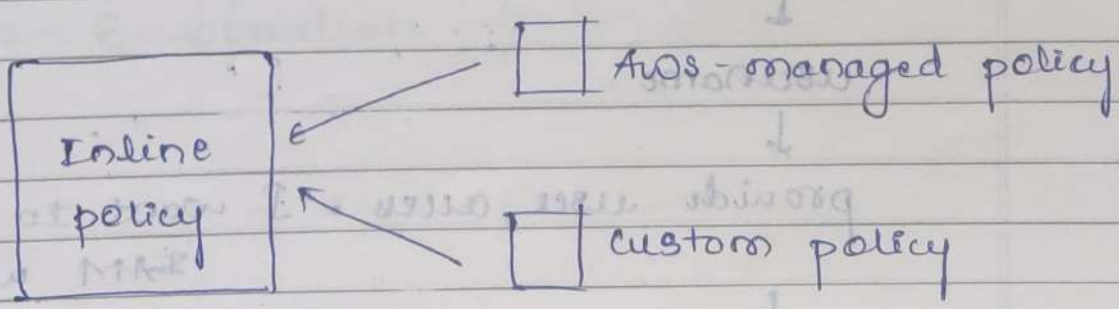
* IAM policies
- JSON-formatted documents
- Attach policy to a user, or role
- Attach policy to select resources eg. Amazon S3 buckets

* Types of IAM policies:
1. Managed policies
2. Inline policies

* Managed IAM policies :→
predefined policies created & managed by AWS. These policies are designed to provide common sets of permissions for various AWS services & resources, making it easier to assign permissions to IAM users, groups or roles without having to write custom policies from scratch

* Managed IAM policies :→
- can be attached to multiple users, groups & roles
- AWS managed policies (created & managed by AWS)
- you can limit who can attach managed policies
- Customer managed policies (created & managed by you)

* Inline IAM policies :→
- policies that are directly embedded within an IAM user, group, or role, rather than being standalone policies attached to these entities.
- These policies are defined within the resources configuration & are considered part of the resource's definition.



* IAM roles :

IAM roles are similar to users, but they are not associated with specific individuals. Instead, roles are assumed by users, applications or AWS services to temporarily gain access to AWS resources.

Roles are often used for cross-account access, temporary access or granting permissions to AWS services.

* When to use IAM roles?

1. Give cross-account access
2. Give access within an account
3. Give access to identities defined outside AWS.

* What are IAM access keys?

Access keys are long term credentials for an IAM user or the AWS account root user. You can use access keys to sign programmatic requests to the AWS CLI or API

Fig-1 Create IAM user with permission

User

↓

username

↓

provide user access - I want to create an IAM user

↓

Custom password
↓
Attach policies directly
↓
S3 full access
↓
Next → Create

- Copy Sign-in details - paste in chrome → username → password → Signin
As I have signed in as IAM user with only S3 full access - I'm unable to access other services EC2, IFS etc only S3 can be accessed.

Ex-2 Create IAM user group

User group
↓
user
↓
EC2 full access

So these selected users will have full access to the EC2

- this is applicable for the users with same access & operation.

Exc-3 Access keys & IAM roles

Create instance

↓

Route [Edit]

↓

Natgateway

↓

instance connect

↓

provide commands

↓

refer AWS command line interface docs in chrome

↓

copy paste the commands

+ Create Access key

. Go to own user

↓

Security Credentials

↓

Create access key

↓

CLI

↓

Next

↓

Create

↓

copy the credentials & paste in terminal

* In case If the nat gateway is not attached.

Roles
↓
Create
↓
Aws service
EC2
↓
next
↓
S3 full access → Next
↓
Name - S3 - intelli
↓
Create

Go to instance
↓
Actions
↓
Security
modify IAM role
↓
write the role name (S3_intellipaat)
↓
update
↓
TO check - Security
↓
aws S3 ls
↓
all details are displayed.

Exc 4 Setup Multi-factor Authentication (MFA)

IAM
↓
Users
↓
Security
↓
Assign MFA
↓
provide name & MFA device (Authenticator app)
↓
Add MFA
↓
QR generated
↓
Scan = provide 2-MFA code
↓
download app & Scan

Lec-3 AWS Key Management Service

+AWS KMS :-

It is a managed service that allows you to create & control encryption keys used to encrypt your data. It integrates with various AWS services & provides a secure & centralized way to manage encryption keys.

* Encryption types :
1. Client Side Encryption
2. Server Side "

1. Client Side Encryption :→

It is a method of Encryption -g data on the client side before it is trans -mitted to a Server or stored in a cloud storage Service. This approach provides an additional layer of Security by Ensuring that data is encrypted before it leaves the client device, thereby minimizing the risk of unauth -orized access to sensitive information

* Works:
Key generation
data Encryption
Key storage
Data transmission
Data Decryption
Secure Key handling

2. Server Side Encryption

It is a method of encrypting data at rest on a Server or storage Service. with SSE, data is Encrypted before being written to disk & decrypted when accessed by authorized users. This helps protect data confidentiality & integrity, Especially when stored in cloud environments or on remote Servers.

AWS KMS

- A Service that Enables you to provision & use encryption keys to protect your data
- Allows you to create, use, & manage encryption keys from within
  - your own applications via AWS SDK
  - supported AWS Services (S3, EBS, RDS, Redshift)
- Available in all commercial regions

* KMS gives you Control

1. Create master key
2. Use a "
3. Create & Export a data key that is encrypted by a master key
4. Enable / disable master key
5. Audit use of master keys in AWS cloudtrail

Exc 5 - Create Symmetric Encryption keys with KMS

KMS
↓
Symmetric
Encrypt & decrypt
↓
Advanced - KMS
↓
Multi range
↓
Next

- Symmetric :
  → Single key used for encrypting & decrypting data or generating & verifying HMAC codes

- Asymmetric :
  → A public & private key pair used for encrypting & decrypting data, signing & verifying messages or deriving shared secrets

Lec-4 AWS Cloudtrail

It is an AWS service that helps you enable operational & risk auditing, governance & compliance of your AWS account.

+ Actions taken by a user, role, or an AWS service are recorded as events in cloudtrail

+ Cloud trail provides 3 ways to record events
  1. Event history
  2. Cloudtrail lake
  3. Trails

* Cloudtrail Events

Cloudtrail Events are records of API activity within an AWS account.

- They provide detailed information such as the identity of the caller, the action performed, the resources affected, & timestamps.
- Cloudtrail events are essential for auditing, security analysis & compliance monitoring in AWS environments

* Types of cloudtrail Events :→
1. Management Events
2. Data Events
3. Insight events

Lec-5 Amazon cloudwatch

cloudwatch enables you to monitor your complete stack (applications, infrastructure, network & services) & use alarms, log & events data to take automated actions & reduce mean time to resolution.

* what does cloudwatch do?

Monitoring, Alarms & Notifications, Logs management, dashboards & Analysis

* Resources monitored by cloudwatch
1. Other resources
2. S3 objects
3. Databases
4. Instances

Exc-6 Configure A cloudwatch Alarm

Instance
↓
Actions
↓
Monitoring & troubleshoot
↓
Manage cloudwatch alarms
↓
Create

Lec-6 AWS Organizations

It is a Service that allows you to centrally manage & govern multiple AWS accounts. It enables you to create & organize accounts into hierarchies, apply policies for Security & Compliance & streamline resource manage-ment & billing across your Organization

+ Service Control policy?

It is a type of policy used in AWS Organizations to manage permissions & control access to AWS Services & resources within an Organization.

- SCPs act as guardrails, allowing Organizations to Enforce Security & compliance policies across all accounts within the organization.

Exc-7 Creating & Managing AWS Organizations.

AWS Organization

↓

Create

↓

Add an AWS account

↓

Existing Email

↓

Send invitation

If you want to add multiple accounts to Organization

MODULE - 6

SIMPLE Storage Service (S3)

- S3 is the backbone of cloud storage

* S3 → It is an object storage service offered by AWS. It provides scalable storage for various types of data, such as files, documents, images & videos.

* Object level storage :→ Also known as object storage, is a data storage architecture that manages data as objects rather than as blocks or files.

+ Components of Every Object :→

1. Data = The actual content of the object, such as files, documents, images, or videos
2. Metadata = Information about the object, including its name, size date created & custom metadata defined by the user.
3. Key = A unique identifier that serves as the object's unique address within the S3 bucket

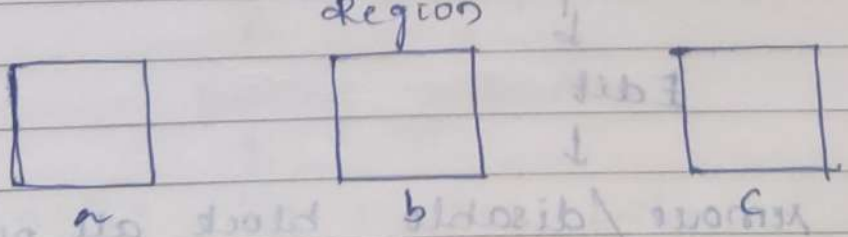
+ Bucket = Container which stores the files, data

* S3 - Cloud based storage service that allows you to store, manage & retrieve large amounts of data like files, images, videos & backups securely & at scale.

- Store data as objects
- Globally unique name
- Region Specific
- Each object within a bucket is stored as a key-value pair

* key is the objects name (which can contain slashes /, mimicking directory structure)
* value is the content of the object (the file/data itself)
  Ex. file is key & the content inside it is value.

* Maximum Object Size = 5TB
* Multipart upload is recommended for objects larger than 5GB (Split the file into smaller parts & upload them separately)



* If you create S3 bucket in one region it replicates in another region

- Create S3 - bucket
  ↓
  upload files
  ↓
  click files
  ↓
  upload

- Public access to website
  click on bucket created
  ↓
  properties
  ↓
  Static Website hosting
  ↓
  Edit
  ↓
  Enable
  It provides end-point website
- By this process the website is in public but not the content inside it
- for this go to permissions
  ↓
  Block public access
  ↓
  Edit
  ↓
  remove / disable block all public access.
  ↓
  Save changes

* S3 - bucket policies :→

- JSON - based access control policies that you attach directly to an S3 bucket to manage permissions for accessing the bucket & its objects.
- They allow you to define who can access the data & what actions they can perform, such as read, write or delete, enabling fine grained control over the security of your data stored in S3

- Write or paste your JSON policy in the bucket policy editor.
- you can use AWS policy generator to create a custom policy, or you can manually write the policy in JSON format
  - Getobject - Used to retrieve or download files from an S3 bucket
  - putObject - Used to upload or add files into an S3 bucket

+ Steps -

permission
↓
Bucket policy
↓
policy generator
↓

Step 1 - S3 bucket policy
Step 2 - Select actions - Getobject
ARN - copy paste from bucket policy
Generate - copy & paste in policy
makes changes in code - ARN "/*"

* S3 Versioning :→

- It allows you to keep multiple version of an object in the same bucket, provid protection against accidental deletions or overwrites
- When versioning is Enabled, S3 stores every version of an object, allowing yo to recover older versions if needed, makin it ideal for data safety & backup.

permissions
↓
Bucket versioning
↓
Enable

- Versioning displays the oldest & Newest files upload.

index.html
└ "
└ "

* If I want to access the old version file then delete the current one or download it & keep.

* S3 replication :→

It allows you to automatically copy objects from one S3 bucket to another which can be

- with in the same region (Same region replica-tion - SRR) or
- in different regions (cross region replication - CRR)

- Its commonly used for compliance, redun-dancy & to improve data access performa-nce by maintaining copies to your users

Creat another bucket for replication
↓
Management
↓
Replication rules
↓
Create - Enter rule Id
Choose a rule Scope
bucket name - browse S3
↓
created one
IAM role - create new one
↓
Save

* S3 - Storage Class

S3 Storage Class	Use Case	Features	Cost
• S3 - Standard	Frequently accessed data	High durability, high availability, low latency	Most expensive, designed for frequent access
• S3 - Intelligent Tiering	Data with unknown or unpredictable access patterns	Moves data automatically b/w frequent & infrequent tiers	Slightly higher than Standard but cost-saving based on usage
• S3 - Standard - IA ↓ infrequent access	Infrequently accessed but quickly retrievable	Lower cost for storage, higher cost for data retrieval	Lower than Standard, ideal for less frequent access
• S3 One Zone IA	non-critical, infrequently accessed data	Stored in a single AZ - lower resilience	Cheaper than Standard-IA good for non-critical
• S3 Glacier	Archival data, rarely accessed	Very low storage cost retrieval time from min to hr	Ideal for data long-term archives with low cost
• S3 Glacier Deep Archive	Deep archival data almost never accessed	Lowest cost, retrieval time up to 12 hrs	Cheapest Storage, ideal for compliance or long-term retention

• S3 outposts	local storage using AWS outpos -ts	provides S3 API locally meets on premises requirements	Dependent on outposts infrasta -ucture Usage

↳ used - If I need to store data for 3 month after 4 month I don't need

* S3 bucket lifecycle :→

you can use lifecycle policies to control the movement of objects b/w different storage classes or delete them entirely, based on specific conditions like age or inactivity

Created bucket
↓
Management
↓
lifecycle rules
↓
Make the changes
Transition Current Versions
↳ Standard IA - 30
One zone IA - 60
Glacier - 90
flexible retrieval
↳ Expires

used when large amount of data need to store on cloud
AWS gives device to store it

* S3 Snow family :→

It is a group of physical devices offered by AWS to help move large amounts of data to the cloud when using the internet isn't practical.

- These devices are used when there's too much data to upload over a regular connection or when dealing with remote areas without good internet

* Snow Family types :→

1. AWS Snowcone =

A small portable device for a few terabytes of data

2. AWS Snowball =

A larger device for moving petabytes of data & can also be used for edge computing

3. AWS Snowmobile =

A massive truck-sized container used for exabyte-scale data transfers, typically used by big companies moving entire data centers.

* These devices help you transfer data quickly, securely, & cost effectively to AWS, especially when internet speed or reliability is an issue.

Amazon S3 Storage Gateway:

- It is a hybrid cloud storage service that connects on premises environments to cloud storage in amazon S3.
- It helps extend your local storage to the cloud by acting as a bridge.

Gateway types :→

1. Amazon S3 file gateway
   Stores & access objects in Amazon S3 from NFS or SMB file data with local caching.

2. Amazon FSx file gateway.
   Access fully managed file shares in amazon FSx for windows file server using SMB.

3. Tape Gateway
   Store virtual tapes in Amazon S3 using iSCSI - VTL, & Store archieved tapes in Amazon S3 Glacier flexible retrieval or Amazon S3 Glacier Deep Archive.

4. Volume Gateway
   Store & access iSCSI block storage volumes in Amazon S3.

* Amazon SES (Simple Email Service) *

- It is a cloud-based email service for sending both transactional & mass emails.
- It lets you send transactional email, marketing message or any other type of high-quality content to your customers.

* SMTP - (Simple Mail transfer protocol)

It is an internet standard communi-cation protocol for electronic mail transmissio

SES

↓

Verified identities

↓

Add mail - Create Identity

↓

Complete the verification

↓

Done

* left menu - SMTP Settings

↓

Create user

↓

username

↓

download credentials

↓

Verified identities

↓

Send a test email

Scenario as custom

↓

mailbox

* SQS (Simple Queue Service)

It is a fully managed message queuing service by AWS, allowing decoupling & scaling of distributed systems.

- It Enables reliable, asynchronous communication b/w microservices, applications & distributed components, enhancing fault tolerance & scalability

* Types of SQS

1. Standard = Send data b/w applications when the throughput is important
2. FIFO (First in first out) = Send data b/w applications when the order of events is important

SQS
↓
Name
Configuration = If required.

* publish messages using SQS Queue.

Create VPC
↓
Endpoint
↓
Create SQS
↓
Connect instance
↓
install aws cli
↓
IAM - user - Access keys → Copy paste on terminal

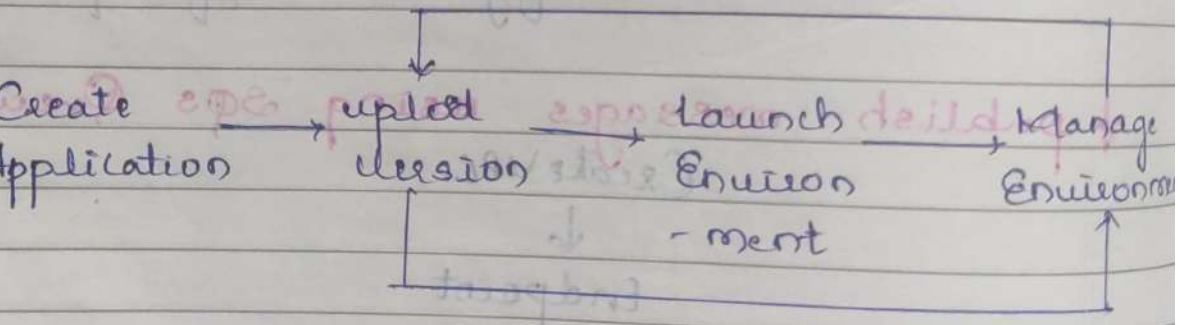
* Elastic Beanstalk on AWS

It is a platform as a service offering by AWS that simplifies deployment & management of web applications.

- It automatically provisions infrastructure
- supports multiple languages & frameworks
- offers features like auto-scaling & monitoring for easy application deployment & scaling.

Upload	Configure	Deploy
• AWS console	Select from multiple platforms	• Auto-Scaling
• AWS CLI	• python	• load Balancing
• CI/CD integration	• Java	• Network
• AWS toolkit for visual studio	• php	• App deployment
	• Docker	

* Workflow for Elastic Beanstalk



servers are managed by AWS]

- function u use the cost will get if not no

* Lambda :→

AWS Lambda is a Serverless computing service that lets you run code in response to events without managing servers.

- you just upload your code & AWS automatically handles the rest, scaling as needed & only charging for the time your code runs.

Lambda
↓
Create function
↓
Author from scratch
Function name = demo.
Runtime = python
↓
Create
↓
Code source
↓
test the code
Create the new test event
↓
now test code

real time run

If you want to make changes in code the deploy it first & then test

* Event-driven Execution :→

- Lambda is an event-driven service, meaning that it runs your code in response to certain triggers or events

- These events can come from many different AWS Services like
  - S3 (file uploads)
  - Dynamo DB (database changes)
  - API gateway (HTTP requests)
  - Cloudwatch (Scheduled events)

* Creating a trigger for lambda.

Add trigger
↓
1st Create S3 bucket
↓
Add this bucket to the trigger of lambda.

log can be seen in lambda

Make changes in S3 bucket [upload]
↓
Now go to created lambda
Monitor
↓
Click Cloudwatch
↓
Changes can be seen

* Lambda function limits

1. Execution time limit - Lambda functions can only run for a maximum of 15 min. If you need longer running tasks, lambda might not be the best choice.
2. Stateless - Lambda functions don't keep state between invocations, so they're best for tasks that don't require long-term memory.

3 Cold start delays = If a lambda function hasn't run in a while, there's sometimes a slight delay - called a 'cold-start' - when it startsup. This can add a little latency, but AWS provides ways to mitigate it for critical functions

* When to use lambda?

1. Image-processing:

Let's say users upload images to your app. you can setup lambda to automatically resize, compress, or even apply filters to each image as its uploaded

2. Data transformation =

If you need to cleanup or process data before storing it in a database, a lambda function can handle that transformation automatically.

3 Real time notifications:

If an event happens like a new user signing up you can use Lambda to trigger an email, SMS, or other notifications instantly.

* Automatic Scaling * Ex- multiple signups & welcome email automatid parallely without delay

- AWS lambda automatically scales the execution of functions in response to the number of incoming requests
- If a thousand requests come in at the same time, AWS lambda will handle them in parallel, making it highly scalable without any configuration