Where you can launch & manage your resources like EC2, DBs

Tuesday
13 may 2025

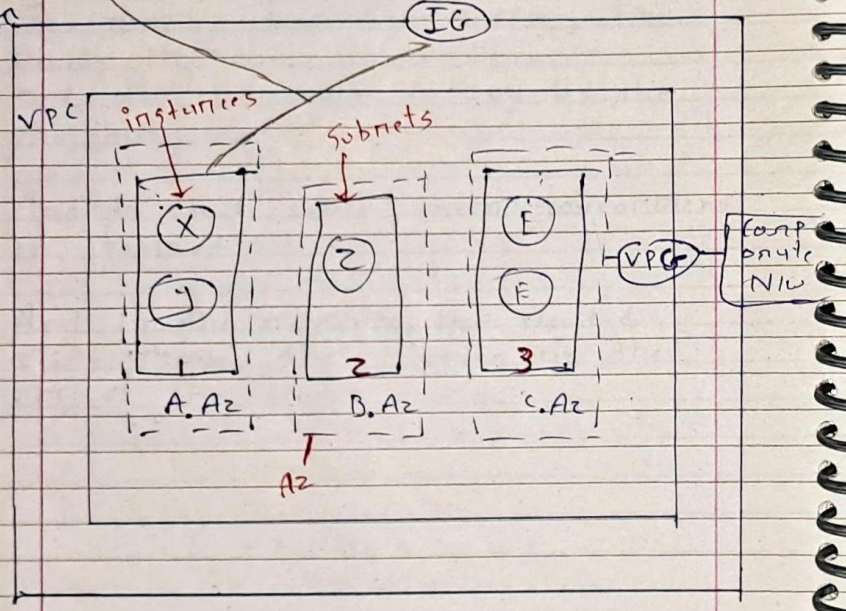
→ VPC (Virtual Private Cloud)

- It is simply a Virtual network dedicated to your AWS account. So you can launch EC2 instances.

Subnets

can be accessed from internet

1. Public
2. Private (cannot be)
3. VPN-only



- VPC scans all available Azs (multiple)
- You can create one or more subnet within one Az.
- IG: through IG we can communicate over internet

VPG: through VPG we can communicate with our corporate network.

If one subnet's traffic routed to the internet gateway then we can say that subnet is a public subnet.

Subnet 2's traffic is not routed to IG so we can say that is a private subnet.

With third subnet there is a no route between SB & IG, but it has a route with VPG, so we can say thy is VPN-only subnet

→ Why there is a need of VPC?

- to ensure security, control & Isolation
- Security: You can control who can access what using subnets, firewalls, & route tables.
- Isolation: Your resources are separated from others, like your own private space in AWS.
- Custom Networking: You can define your own IP ranges, subnets, & gateways.
- Internet Control: You decide which resources are public or private.

Subnet represents a range of IP addresseses within a virtual private cloud. Each subnet must be contained within a single Az & cannot span across multiple Az.

→ IG (Internet Gateway)
- It is used to connect your VPC to the internet.
- allows resources in a public subnet (like a web server) to send / receive data over the internet.
- without IG, your VPC is completly private.

→ VPG (Virtual Private Gateway)
- used to connect your VPC to your on-premise network (like your office or data center) using a VPN connection.
- secure communication between AWS and your physical network. (doorway)
- used in hybrid cloud setups.

→ VPN (Virtual Private Network)
- it is a secure, encrypted connection between 2 networks over the internet.
- in AWS, it is used to connect your on-premise network to your AWS VPC securely.

Why VPN?
- to safely transfer data between your AWS cloud & physical network.
- works through VPG.
- ensures privacy & protection even when using public subnet.

IMP Property of Subnet

→ Route Table
- it is a set of rules in your VPC that tells AWS where to send network traffic.
- Each route in the table specifies a destination CIDR (like 0.0.0.0/0 for all IPs) and a target (IG / NAT / VPG).
- Every subnet uses one route table - traffic from instances in that subnet follows its table's rules.
- By editing rules, you control whether traffic goes to the internet, to a VPN, or stays within your VPC.
- In short, Route Tables direct your packets along the right roads inside and outside your VPC.

→ CIDR (Classless Inter-Domain Routing)
- it is a way to represent IP address ranges more efficiently.

Format: IP address / Prefix-length
Example: 192.168.0.0/24
starting IP — means first 24 bits are fixed for the network, and the rest can vary for host.

VPCs in peering connection cannot have overlapping CIDR blocks.

PAGE NO. DATE

→ IP Address? (Internet Protocol)
- unique number given to each device on a network so they can communicate with each other
- it is like or home address

Why IP is needed?
- it is like a phone number or home address for devices on network.

Example: Imagine sending a letter, you need
- your address (sender)
- receiver's address

The same way, when a computer sends data (like opening website) it needs:
- Your IP address (so the server known where to send data back)
- Server's IP address (so your request knows where to go)

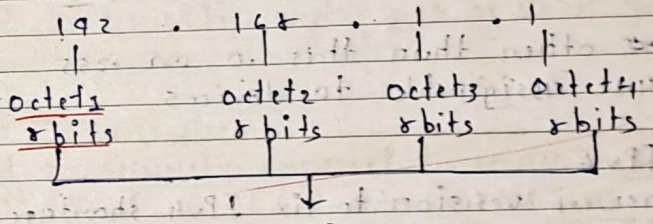
Without IPs:
- Devices would not know where to send or receive data.

192.168.1.0 → network address (not assigned to any device) — represents entire network
192.168.1.255 → broadcast address (used to send data to all devices in the subnet)
routers

Two versions of IP:
1. IPv4
2. IPv6

→ IPv4
- Most commonly used
- 4 numbers separated by dots
  - 192.168.1.1
- Each number ranges from 0 - 255
- Each number is called an octet (8 bits)
- Each octet ranges from 0 - 255

usable host IPs: 192.168.1.1 to 254



Role of Segments:
- Depending on the subnet (network size), the octets are split into:
  - Network part (used to identify network)
  - Host Part (used to identify device inside the network)

For example: in a /24 CIDR like 192.168.1.0
- first 3 octets (192.168.1) → Network part
- last octet (1) → Host Device

Public IPv4 addresses enable instances to communicate with internet services and allow inbound traffic from the public internet.

PAGE NO. DATE

→ if .0 in the last octet → Network Address
- represents entire network
- not assigned to any device
- used by routers or systems to identify subnets

→ if .255 in last octet → Broadcast address
- used to send a message to all devices in same subnet.
- also not assigned to any single device

→ so total IPs : 256 (0 to 255)
Usable for devices → 192.168.1.1 to 254

other than this .0 or .255 are assignable to devices

→ IPv6
- newer version to fix IPv4 shortage
-

Format: 8 groups of hexadecimal nums :->
2001 : 0db8 : 8543 : 0000 : 0000 : 8420 : 0307 : 7334

→ Security Groups
- these are virtual firewall in AWS that control inbound & outbound traffic for your resources like EC2 instances
- Inbound Rules: Define what traffic is allowed to come in (e.g. allow HTTP or SSH)

Private IPv4 addresses enable instances to communicate securely within a VPC or network, but cannot directly access or be accessed from the public internet without a NAT or other gateway.

PAGE NO. DATE

- Outbound Rules: Define what traffic is allowed to go out. (e.g. allow internet access.)

By default:
- All inbound traffic is blocked
- All outbound traffic is allowed

- In short, SG protect your resources by controlling who can connect & how.

→ HTTP or HTTPS (secure) → with encryption using SSL/TLS
- used to transfer web pages between your browser and a website.

→ SSH (Secure shell)
- used to securely connect to remote servers or computers.
- encrypts all communication
- lets you log in, run commands, and transfer files safely over internet

what it does, who uses it
- used by developers, sysadmins, and cloud users to manage servers

→ SGs are linked to single VPC. You can assign a SG to one or more EC2 instances, but that each must be in the same VPC as the SG.

A Gateway is a network component that acts as a bridge between different networks, such as connecting your VPC to the IG, or to on-premises networks (VPG)

By default VPCs are isolated

networking connection between 2 VPCs, that you can use to route traffic between them by using Private IP addresses

PAGE NO. DATE

VPC Peering Connection

- it is a network link between 2 VPCs
- that allows them to ~~connect~~ communicate with each other privately, as if they are on the same network.

Why it's used: Let's say you have
- one VPC for web servers &
- another VPC for your databases

You can peer them so they can talk to each other securely; without going over internet

- one-to-one communication (not transitive)
- Private IP communication only

VPC A ⟷ VPC B ✓

VPC A ⟷ VPC B ⟷ VPC C ✗

(A ⟷ B, B ⟷ C, but A ✗ C)

Needs direct peering

→ Create separate peering A ⟷ C ✓

not transitive means, communication does not automatically pass through another VPC

- Each VPC must directly peered to talk.

After peering connection, route tables must be configured.

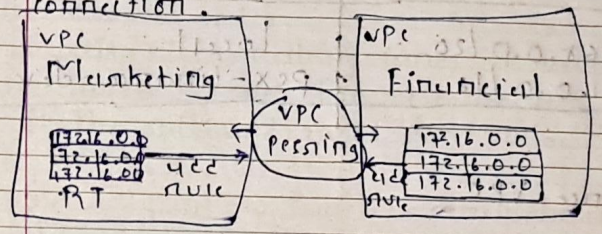
New rules must be created for each VPC that points to the VPC peering connection

RT must be configured for VPC peering to work effectively in both VPCs

PAGE NO. DATE

Example:

To access finance VPC, the marketing VPC route table points to the VPC peering connection.



After the peering connection has been established and the RTs are configured, the ries of 2 VPCs can communicate with each other.

Examples of Route Tables

2. Marketing VPC (CIDR: 10.10.0.0/16)

Destination	Target
10.10.0.0/16	local
172.31.0.0/16	pcx-finance mktg

VPC Peering Connection to Finance VPC

allows Marketing to talk to finance only.

A NAT allows instances with private IP addresses to access the internet, while still preventing inbound traffic from the internet from reaching those instances directly

PAGE NO. DATE / /

2. Developer VPC (CIDR: 192.168.0.0/20)

Destination	Target
192.168.0.0/20	local
172.31.0.0/16	pcx-finance-dev

3. Finance VPC

Destination	Target
172.31.0.0/16	local
10.0.0.0/16	pcx-finance-mktg
192.168.0.0/20	pcx-finance-dev

You can connect to an Amazon EC2 instance in 4 ways:
- EC2 instance connect
- Session Manager
- SSH Client
- EC2 serial console

→ With session manager, you can manage your Amazon EC2 instances, edge devices, and on-premises servers & VMs

→ You can either use an interactive one click browser based shell or the AWS CLI.

SGs do not automatically accept data from peered VPCs. You must update SGs to allow a peered VPC as an incoming source.

SGs are stateful: if your instance sends a request to somewhere, the reply is allowed to come back in - even if there is no rule allowing that in the inbound rules.

Also if something comes in because of an allowed inbound rule, the response can go back out - even if the outbound rules don't allow it.

SG rules only allow traffic - they don't block it:
- You can't create rules that deny access.
- Instead, you use this rules to allow specific traffic based on things like protocol (e.g. TCP, UDP) and port numbers. (22 for SSH, 80 for HTTP, 443 for HTTPS, 3306 for MySQL)

→ 8th First NoSQL Database

DynamoDB is a Key-Value (stores data as key-value pairs) & document database (stores data in JSON-like documents) that delivers single digit milliseconds performance at any scale

→ DynamoDB Table Terms

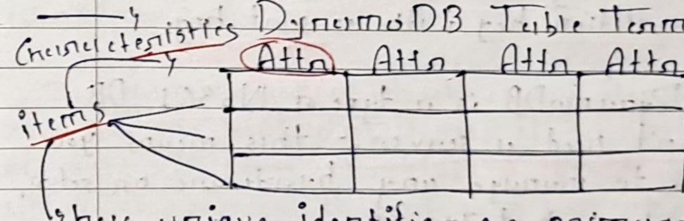


Table is a collection of data (items)

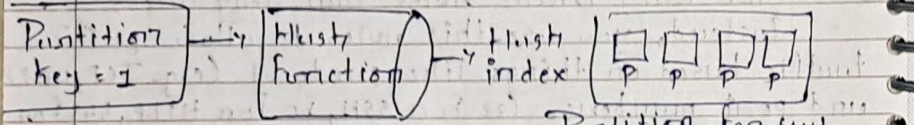
↳ has unique identifier or primary key

DynamoDB supports 2 kind of Primary keys
- Partition key &
- Sort key

The Partition key is Mandatory. DynamoDB uses the Partition key's value to an internal hash function.

The output from the hash function determines the Partition, which is physical storage internal to DynamoDB in which the item will be stored.

The Partition key enables partitions for scalability purposes.



The sort key is optional & provides additional flexibility for querying data.

14th May, 2025
Wednesday

DynamoDB supports Two Primary Keys
- a primary key that only uses a partition key
- a composite key that combines
  a partition key with a sort key

- Amazon DynamoDB is a type of NoSQL DB, that doesn't need a server. - this means you dont have to manage any hardware or setup.
- Only Pay for what you use
- Auto scale up or down
- Amazon manages updates, patch, or maintenance

- DynamoDB saves data in partitions, which are like storage blocks. These are stored on fast SSDs and are automatically copied across different locations (AZs) in same AWS region to keep data safe & always available.

- DynamoDB table has many items, and each item is a set of attributes (like columns) that is uniquely identified. (like row in DB)

- When you add an item to a DynamoDB table, it uses the partition key value and runs it through a hash function. The hash result decides which partition the item will be saved in.

- Query operation in DynamoDB lets you find items using the Primary key. You must give the partition key name and its value. it will then return all items with that key
  if the table also has a sort key, you can use it with filters (like greater than, equals) to narrow down the results.
  When you Run Query in DynamoDB, it looks for an exact match of the partition key. If you also give a sort key, it helps to filter the results even more.

- Scan operation checks every item in table or index and returns matching items. if Scan goes over 1MB of data, it stops and gives a lastEvaluatedKey, which you can use to continue scanning later.

9th File Systems in the Cloud

- Amazon EFS is a shared cloud storage service that lets multiple EC2 instances access the same files at the same time.
- It works like a regular folder on each server, but the data is stored in the cloud. EFS automatically grows or shrinks as you add or remove files, and it is easy to set up & use across different AZs.
- To make web management easier, Amazon EFS is used so that all servers can share the same files instead of storing them separately.

Once EFS is setup, EC2 instances in the same VPC can access it at the same time.

Each AZ gets a mount target (IP endpoint), which servers in that AZ use to connect EFS as a local folder.

You can also add more EC2 instances anytime by connecting them to the mount target in their AZ.

Network File System protocol

- Amazon EFS requires an inbound NFS rule. By selecting a SG as the incoming source, any EC2 instances linked to the SG you select will have NFS client access to the file system.
- Amazon FSx is a fully managed service that lets you use popular third party FSs in AWS cloud.

It removes the need to manage or setup your own file servers & storage.

After creating EFS, you create mount targets on each subnet.

The mount target enables communication from EC2 instances on the subnet.

The Amazon EFS client software (amazon-efs-utils) is a free, open source tool used to connect & access EFS from Linux system.

10th Auto Healing & Scaling apps

Thursday 15th May 2025

An EC2 Auto Scaling group is created to automatically add or remove EC2 instances based on the workload of an application.

First, an Amazon Machine Image (AMI) (template) of the app server is created. Then, a launch template is made to define how new instances should be started (like type, AMI, key pair).

The auto scaling group uses this template to scale EC2 instances up or down. For example, if CPU usage goes up, a CloudWatch alarm adds a new instance. If usage drops, it removes extra servers.

You can also set scheduled actions to start or stop instances at specific times, like during peak hours or after the workload is done.

- An AMI is a template that contains the software configuration (like OS, app servers, & apps) required to launch a new EC2 instance in AWS.
You can capture the contents of an instance & its volume into an AMI.
Steps: Select instance → Actions → Image & templates → Create Image

- By default, when you create an AMI, EC2 shuts down the instance, takes snapshots, creates and registers the AMI, and then reboots the instance. (If you don't want to reboot - then "No reboot" option).
Also, AMIs are region-specific. To use an AMI in another region, you need to copy it to that region.

- A Launch Template in EC2 is a feature that lets you save instance settings (like AMI ID, instance type, and network settings) so you don't have to enter them every time you launch an instance.
You can have up to 5,000 templates per Region and 10,000 versions per template.
Steps: EC2 → Launch Templates → Create Launch Template

- A Key Pair is a security credential to log in to EC2 instances - SSH for Linux & RDP for windows.
It includes public key (stored by EC2) & a Private key (downloaded & kept securely by user). (It must be kept safe & should not be uploaded to cloud storage.

- Use of Launch Template: helps to set up Auto scaling and self-healing. If a server fails, the template provides the details needed to automatically launch a new instance with the same configuration.

→ AWS Auto-Scaling Overview:
- AWS Auto-Scaling helps you automatically add or remove resources (like EC2 instances) based on demand. It balances performance and cost.
- You can set desired instance limits (min, max, and target count).
It uses EC2 and Load Balancer health checks to decide if an instance is unhealthy.
With target tracking policies, you set a goal (e.g. 70% CPU usage), and Auto-Scaling adjusts the instance count to meet that target.
CloudWatch alarms are automatically set up to trigger scaling actions.
You can get notifications through Amazon SNS when instances are launched or removed.
You can also track the scaling history and see which condition triggered the scale.

Steps: EC2 → Auto Scaling Groups → Create Auto Scaling G.
1. Choose launch template or configuration.
2. Choose instance launch options
3. Integrate with other services
4. Configure group size & scaling
5. Add notifications
6. Add tags
7. Review

→ Scheduled Scaling & Target Tracking

Scheduled Scaling lets you plan when to scale your EC2 instances based on expected traffic patterns. For example, if your app gets more traffic every monday morning, you can schedule scaling in advance to handle it.

Target Tracking Policies use CloudWatch alarms to monitor a metric (like CPU usage).

Two alarms are created:

1. One to scale out (add instances), when usage is high.
2. One to scale in (remove instances), when usage drops.

CloudWatch monitors each instance's CPU usage and when set threshold is hit, it notifies the Auto scaling group to take action. This ensures your app has enough power when needed and saves cost when demand is low.

11th Highly Available Web Applications

→ Making a app highly available with AWS

1. Domain Management with Route 53

When a user types your website name, Amazon Route 53 translates that name into an IP address - basically telling the browser where to go. This is just like using someone's name to find their house address.

2. Faster content with CloudFront

Once the IP is found, the user connects to a CloudFront application distribution, which is a global CDN. It stores (caches) copies of your site's static content - like Images & Videos - closer to the users, so the site loads much faster.

This static content originally stored in Amazon S3.

3. Load Balancing with ELB

To make sure no single server is overloaded, ELB spreads incoming traffic across multiple EC2 instances. These instances are set up in different AZs so even if one zone fails, your app still runs.

4. Scaling Automatically with EC2 Auto Scaling

As the number of users grow or drops, Amazon EC2 Auto Scaling adds or removes servers automatically. For example, if CPU usage crosses 80% for some time, a new EC2 instance is launched to handle the load. This keeps your app responsive without wasting money on unused resources.

These scaling decisions are based on metrics tracked by Amazon CloudWatch.

5. Database with Amazon RDS

Your app likely needs a database to store user data or content. That's where Amazon RDS comes in. It provides a managed, reliable database server. If the main database crashes, AWS can failover to a backup instance, so your app keeps working.

→ How a Load Balancer works with Auto Scaling in AWS

Think of Load Balancer as the reception desk for your web application. Every visitor (user req.) first goes to this desk. The load balancer then decides which web server (EC2 Instance) will handle the request.

When using it with Auto Scaling. It auto keeps track of which servers are active:
- New EC2 instances are auto. registered.
- When instances shut down, they are deregistered.

What is an ALB?
- type of LB
- operates at Layer 7 - Application Layer of OSI which means it understands things like HTTP req., URLs, cookies etc.
- It distributes traffic across:
  - EC2 instances
  - Containers (Docker)
  - IP addresses.

So, instead of overloading one server, it spreads the work evenly across many.

Target Groups & AZs

ALB sends traffic to a group of EC2 instances called target groups.

- To improve availability:
  - AWS lets you enable multiple AZs.
  - ALB routes traffic only to instances in healthy AZs

Security Groups for Load Balancer
- SG - firewall rules that control traffic in and out.
- To allow public traffic (e.g. from internet) use 0.0.0.0/0
- To increase security instead of allowing all traffic, you can restrict it by using specific security groups. like:
  - Only allow traffic to ECIs from LB.
  - This avoids exposing EC2s directly to the internet.
- These SGs are linked to your VPC & can only be assigned to resources within that VPC.
- You can even assign multiple SGs to an instance.

Subnets & Auto Scaling Groups:
- Your Auto Scaling Groups lives in a specific subnets.
You can:
- Add or remove subnets to control where EC2s are launched.
- For security, run your app in private subnet and let the ALB (which is public) in handle traffic.

→ Auto Scaling Behavior
↱ The Auto Scaling Group:
- Keeps a set number of EC2s running based on demand.
- If you manually increase desired capacity it launches new EC2s.
- If any instance fails or is terminated, it launches a new one automatically (to meet the minimum capacity).

↱ For High Availability:
- Spread your ASG across multiple AZs.
- If one AZ becomes unhealthy, AWS can try launching in a healthy one.

↱ AWS uses AZ distribution strategies:
- Balanced Best Effort: Tries healthy zones first.
- Balanced only: Tries only the selected zones, even if unhealthy.

→ Testing & Connection Draining
You can test scaling by changing the desired capacity and observing how EC2 instances are launched or terminated.

If one instance is being removed, Connection Draining ensures that ongoing requests are completed before it's fully removed. This prevents sudden failures for users.

12th Core Security Concepts | Sunday 18th May 2025

→ Managing Permissions with IAM in AWS
Imagine a company has several support engineers who need read-only access to certain AWS services - like EC2 & RDS.
Instead of setting permissions for each engineer one by one, AWS lets you use a user group to manage them all at once.

User Group: like a bucket of permissions - any user added to this group automatically gets those permissions.

IAM is:
- Manages secure access to AWS services.
- free - you only pay for the services users access.
- helps to control who can do what in your AWS account.

IAM User: Represent a person or app; his credentials (username/password or access keys).

IAM Group: A container for users; permissions are assigned to the group & inherited by all users in it.

Policy: A document (in JSON) that defines what actions are allowed / denied on AWS res.