**AWS NOTES:**

**IAM:**

AWS IAM (Identity and Access Management) is a service provided by Amazon Web Services (AWS) that helps you manage access to your AWS resources. It's like a security system for your AWS account.

IAM allows you to create and manage users, groups, and roles. Users represent individual people or entities who need access to your AWS resources. Groups are collections of users with similar access requirements, making it easier to manage permissions. Roles are used to grant temporary access to external entities or services.

With IAM, you can control and define permissions through policies. Policies are written in JSON format and specify what actions are allowed or denied on specific AWS resources. These policies can be attached to IAM entities (users, groups, or roles) to grant or restrict access to AWS services and resources.

IAM follows the principle of least privilege, meaning users and entities are given only the necessary permissions required for their tasks, minimizing potential security risks. IAM also provides features like multi-factor authentication (MFA) for added security and an audit trail to track user activity and changes to permissions.

By using AWS IAM, you can effectively manage and secure access to your AWS resources, ensuring that only authorized individuals have appropriate permissions and actions are logged for accountability and compliance purposes.

Overall, IAM is an essential component of AWS security, providing granular control over access to your AWS account and resources, reducing the risk of unauthorized access and helping maintain a secure environment.

**Components of IAM**

**Users:** IAM users represent individual people or entities (such as applications or services) that interact with your AWS resources. Each user has a unique name and security credentials (password or access keys) used for authentication and access control.

**Groups:** IAM groups are collections of users with similar access requirements. Instead of managing permissions for each user individually, you can assign permissions to groups, making it easier to manage access control. Users can be added or removed from groups as needed.

**Roles:** IAM roles are used to grant temporary access to AWS resources. Roles are typically used by applications or services that need to access AWS resources on behalf of users or other services. Roles have associated policies that define the permissions and actions allowed for the role.

**Policies:** IAM policies are JSON documents that define permissions. Policies specify the actions that can be performed on AWS resources and the resources to which the actions apply. Policies can be attached to users, groups, or roles to control access. IAM provides both AWS managed policies (predefined policies maintained by AWS) and customer managed policies (policies created and managed by you).

**INTRODUCTION TO EC2:**

**What is EC2, and why is it important?**

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.

- Access reliable, scalable infrastructure on demand. Scale capacity within minutes with SLA commitment of 99.99% availability.

- Provide secure compute for your applications. Security is built into the foundation of Amazon EC2 with the AWS Nitro System.

- Optimize performance and cost with flexible options like AWS Graviton-based instances, Amazon EC2 Spot instances, and AWS Savings Plans.

**EC2 usecases**

Deliver secure, reliable, high-performance, and cost-effective compute infrastructure to meet demanding business needs.

Access the on-demand infrastructure and capacity you need to run HPC applications faster and cost-effectively.

Access environments in minutes, dynamically scale capacity as needed, and benefit from AWS’s pay-as-you-go pricing.

Deliver the broadest choice of compute, networking (up to 400 Gbps), and storage services purpose-built to optimize price performance for ML projects

**EC2 Instance Types**

Recommended to follow [this](https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html) page for very detailed and updated information.

**General purpose**

General Purpose instances are designed to deliver a balance of compute, memory, and network resources. They are suitable for a wide range of applications, including web servers,

small databases, development and test environments, and more.

**Compute optimized**

Compute Optimized instances provide a higher ratio of compute power to memory. They excel in workloads that require high-performance processing such as batch processing,

scientific modeling, gaming servers, and high-performance web servers.

**Memory optimized**

Memory Optimized instances are designed to handle memory-intensive workloads. They are suitable for applications that require large amounts of memory, such as in-memory databases,

real-time big data analytics, and high-performance computing.

**Storage optimized**

Storage Optimized instances are optimized for applications that require high, sequential read and write access to large datasets.

They are ideal for tasks like data warehousing, log processing, and distributed file systems.

**Accelerated computing**

Accelerated Computing Instances typically come with one or more types of accelerators, such as Graphics Processing Units (GPUs),

Field Programmable Gate Arrays (FPGAs), or custom Application Specific Integrated Circuits (ASICs).

These accelerators offload computationally intensive tasks from the main CPU, enabling faster and more efficient processing for specific workloads.

[](https://user-images.githubusercontent.com/43399466/247236527-fc8e083c-dba5-41a6-94b9-14ebef0255c1.png)

Instance families

C – Compute

D – Dense storage

F – FPGA

G – GPU

Hpc – High performance computing

I – I/O

Inf – AWS Inferentia

M – Most scenarios

P – GPU

R – Random access memory

T – Turbo

Trn – AWS Tranium

U – Ultra-high memory

VT – Video transcoding

X – Extra-large memory

Additional capabilities

a – AMD processors

g – AWS Graviton processors

i – Intel processors

d – Instance store volumes

n – Network and EBS optimized

e – Extra storage or memory

z – High performance

**EC2 Instance Basics:**

Understanding the concept of virtual servers and instances. Key components of an EC2 instance: AMI (Amazon Machine Image), instance types, and instance states. Differentiating between On-Demand, Reserved, and Spot instances.

**Launching an EC2 Instance:**

* Step-by-step guide on launching an EC2 instance using the AWS Management Console.
* Configuring instance details, such as instance type, network settings, and storage options.
* Understanding security groups and key pairs for securing instances.

**Managing EC2 Instances:**

* Starting, stopping, and terminating instances.
* Monitoring instance performance and utilization.
* Basic troubleshooting and accessing instances using SSH (Secure Shell).

**VPC:**

Imagine you want to set up a private, secure, and isolated area in the cloud where you can run your applications and store your data. This is where a VPC comes into play.

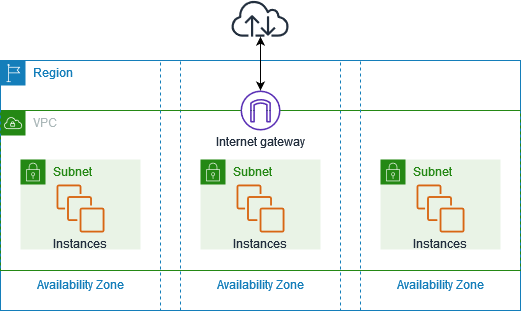
A VPC is a virtual network that you create in the cloud. It allows you to have your own private section of the internet, just like having your own network within a larger network. Within this VPC, you can create and manage various resources, such as servers, databases, and storage.

Think of it as having your own little "internet" within the bigger internet. This virtual network is completely isolated from other users' networks, so your data and applications are secure and protected.

Just like a physical network, a VPC has its own set of rules and configurations. You can define the IP address range for your VPC and create smaller subnetworks within it called subnets. These subnets help you organize your resources and control how they communicate with each other.

To connect your VPC to the internet or other networks, you can set up gateways or routers. These act as entry and exit points for traffic going in and out of your VPC. You can control the flow of traffic and set up security measures to protect your resources from unauthorized access.

With a VPC, you have control over your network environment. You can define access rules, set up firewalls, and configure security groups to regulate who can access your resources and how they can communicate.

[](https://user-images.githubusercontent.com/43399466/247909766-12cc10b6-724c-42c9-b07b-d8a7ce124e24.png)

By default, when you create an AWS account, AWS will create a default VPC for you but this default VPC is just to get started with AWS. You should create VPCs for applications or projects.

## VPC components

The following features help you configure a VPC to provide the connectivity that your applications need:

**Virtual private clouds (VPC)**

A VPC is a virtual network that closely resembles a traditional network that you'd operate in your own data center. After you create a VPC, you can add subnets.

**Subnets**

A subnet is a range of IP addresses in your VPC. A subnet must reside in a single Availability Zone. After you add subnets, you can deploy AWS resources in your VPC.

**IP addressing**

You can assign IP addresses, both IPv4 and IPv6, to your VPCs and subnets. You can also bring your public IPv4 and IPv6 GUA addresses to AWS and allocate them to resources in your VPC, such as EC2 instances, NAT gateways, and Network Load Balancers.

Network Access Control List (NACL)

A Network Access Control List is a stateless firewall that controls inbound and outbound traffic at the subnet level. It operates at the IP address level and can allow or deny traffic based on rules that you define. NACLs provide an additional layer of network security for your VPC.

**Security Group**

A security group acts as a virtual firewall for instances (EC2 instances or other resources) within a VPC. It controls inbound and outbound traffic at the instance level. Security groups allow you to define rules that permit or restrict traffic based on protocols, ports, and IP addresses.

**Routing**

Use route tables to determine where network traffic from your subnet or gateway is directed.

**Gateways and endpoints**

A gateway connects your VPC to another network. For example, use an internet gateway to connect your VPC to the internet. Use a VPC endpoint to connect to AWS services privately, without the use of an internet gateway or NAT device.

Peering connections

Use a VPC peering connection to route traffic between the resources in two VPCs.

**Traffic Mirroring**

Copy network traffic from network interfaces and send it to security and monitoring appliances for deep packet inspection.

**Transit gateways**

Use a transit gateway, which acts as a central hub, to route traffic between your VPCs, VPN connections, and AWS Direct Connect connections.

**VPC Flow Logs**

A flow log captures information about the IP traffic going to and from network interfaces in your VPC.

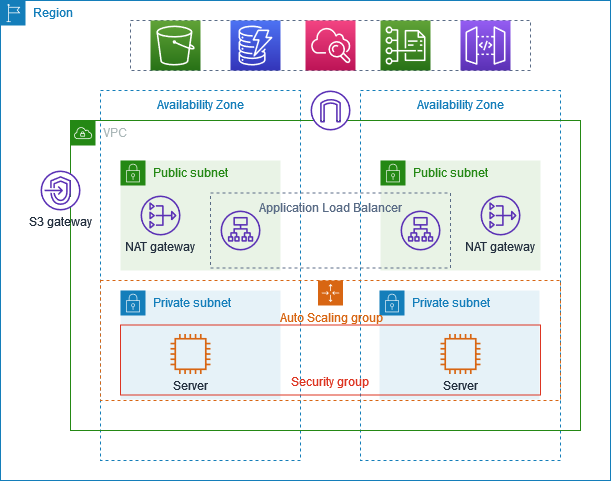
**VPN connections**

Connect your VPCs to your on-premises networks using AWS Virtual Private Network (AWS VPN).

**Resources**

VPC with servers in private subnets and NAT

<https://docs.aws.amazon.com/vpc/latest/userguide/vpc-example-private-subnets-nat.html>

[](https://user-images.githubusercontent.com/43399466/248278052-89d8316e-7b70-4821-a6bf-67d1dcc4d2fb.png)

AWS SECURITY GROUPS & NACL:

AWS (Amazon Web Services) provides multiple layers of security to protect resources and data within its cloud infrastructure. Two important components for network security in AWS are Security Groups and Network Access Control Lists (NACLs). Let's explore how each of them works:

**Security Groups:**

Security Groups act as virtual firewalls for Amazon EC2 instances (virtual servers) at the instance level. They control inbound and outbound traffic by allowing or denying specific protocols, ports, and IP addresses.

Each EC2 instance can be associated with one or more security groups, and each security group consists of inbound and outbound rules.

Inbound rules determine the traffic that is allowed to reach the EC2 instance, whereas outbound rules control the traffic leaving the instance.

Security Groups can be configured using IP addresses, CIDR blocks, security group IDs, or DNS names to specify the source or destination of the traffic.

They operate at the instance level and evaluate the rules before allowing traffic to reach the instance.

Security Groups are stateful, meaning that if an inbound rule allows traffic, the corresponding outbound traffic is automatically allowed, and vice versa.

Changes made to security group rules take effect immediately.

**Network Access Control Lists (NACLs):**

NACLs are an additional layer of security that operates at the subnet level. They act as stateless traffic filters for inbound and outbound traffic at the subnet boundary.

Unlike Security Groups, NACLs are associated with subnets, and each subnet can have only one NACL. However, multiple subnets can share the same NACL.

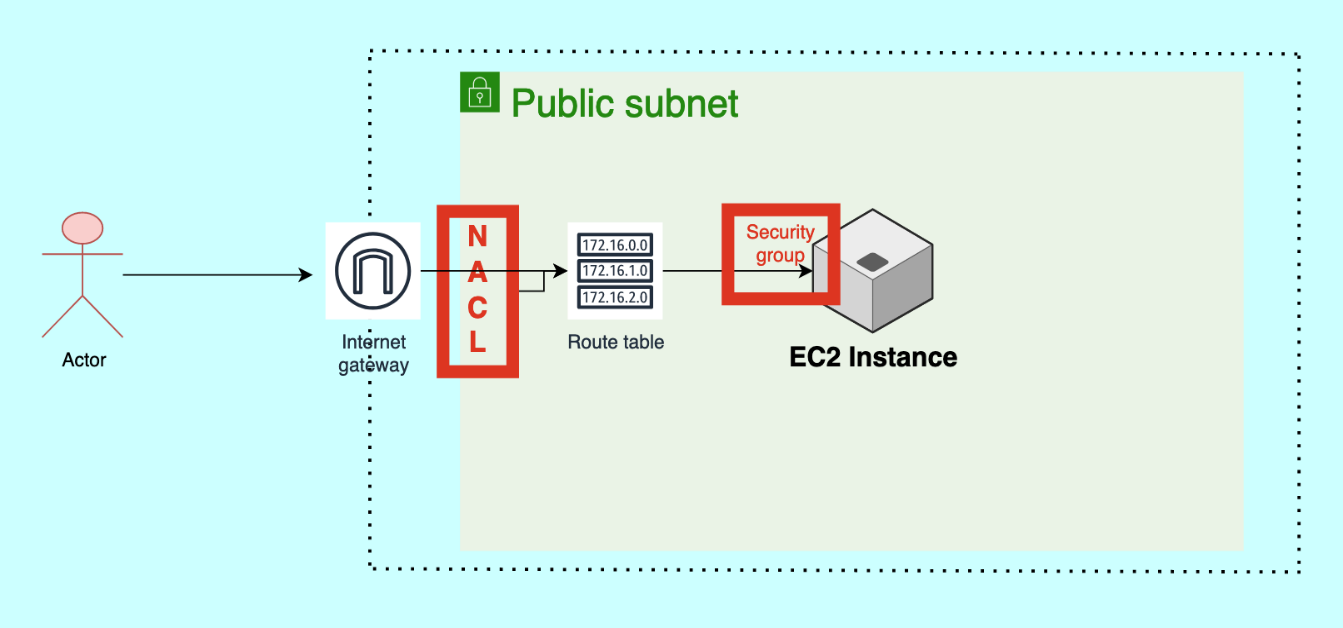
NACLs consist of a numbered list of rules (numbered in ascending order) that are evaluated in order from lowest to highest.

Each rule in the NACL includes a rule number, protocol, rule action (allow or deny), source or destination IP address range, port range, and ICMP (Internet Control Message Protocol) type.

NACL rules can be configured to allow or deny specific types of traffic based on the defined criteria.

They are stateless, which means that if an inbound rule allows traffic, the corresponding outbound traffic must be explicitly allowed using a separate outbound rule.

Changes made to NACL rules may take some time to propagate to all the resources using the associated subnet.

[](https://user-images.githubusercontent.com/43399466/249581348-30bbc9e8-6502-438b-8adf-ece8b81edce9.png)