Basic Terms and definitions that we should know before we jump into the AWS VPC Learning Curve

Virtual Private Cloud (VPC): A logically isolated section of the AWS cloud where you can define and control a virtual network that is separate from other networks.

Subnet: A range of IP addresses within your VPC. Subnets help organize and secure your resources by grouping them according to their security and operational needs.

CIDR Block (Classless Inter-Domain Routing): A method for specifying IP address ranges. When creating a VPC or subnet, you define a CIDR block to allocate IP addresses.

Internet Gateway (IGW): A gateway that connects your VPC to the internet. It allows instances in your VPC to communicate with the internet and vice versa.

NAT Gateway: A managed service that allows instances in a private subnet to access the internet without exposing them directly. It performs Network Address Translation (NAT).

Route Table: A set of rules, called routes, that are used to determine where network traffic is directed within the VPC. Each subnet must be associated with a route table.

Security Group: A virtual firewall that controls inbound and outbound traffic to your instances. Security groups are stateful, meaning if you allow an incoming request from an IP, the response is automatically allowed.

Network Access Control List (NACL): A set of rules that controls inbound and outbound traffic at the subnet level. NACLs are stateless, meaning responses to allowed inbound traffic are not automatically allowed outbound.

Elastic IP Address: A static, public IP address that you can associate with an instance or a NAT gateway to ensure that it retains a fixed IP address.

VPC Peering: A connection between two VPCs that allows instances in one VPC to communicate with instances in another VPC as if they were in the same network.

VPN Gateway: A virtual private gateway that enables you to create a secure VPN connection between your VPC and your on-premises network.

Transit Gateway: A service that enables you to connect multiple VPCs and on-premises networks through a single gateway, simplifying network management and scalability.

Private Link: A service that allows you to access services hosted on AWS privately and securely without using public IPs or traversing the internet.

DHCP Options Set: A set of DHCP options that you can configure for your VPC to specify network parameters such as domain names and DNS servers.

VPC Endpoint: A network endpoint within your VPC that allows private connections to AWS services and endpoints, avoiding the need for public IP addresses or NAT.

Elastic Network Interface (ENI): A virtual network interface that can be attached to an instance to provide additional networking features like private IP addresses, security groups, and more.

Think of **AWS VPC (Virtual Private Cloud)** as your own private network in the cloud.

Here's how it works in simple terms:

1. **Your Own Cloud Network**: When you create a VPC, it's like building a secure, isolated section of the AWS cloud just for you. It's your private space where your applications and resources (like servers, databases, etc.) can live.
2. **Subnets – Divide the Space**: Within the VPC, you can create **subnets**, which are like smaller sections of your network. You can have a public subnet (accessible from the internet) for things like web servers, and private subnets (hidden from the internet) for things like databases.
3. **Gateways and Internet Access**: If you want resources in your VPC to connect to the internet, you need to set up an **Internet Gateway**. This is like the door between your private network and the public internet. You can control which parts of your VPC are connected to the internet.
4. **Security – Control Who Gets In**: VPCs come with **security groups** (like firewalls) and **network ACLs** (like rules for traffic) to control who can access your resources. You set the rules for which computers or people can enter your VPC and how they interact with your resources.
5. **Communication Between VPCs**: If you have more than one VPC, you can connect them together using **VPC peering** or **Transit Gateway**, so they can talk to each other securely.
6. **Data Flow – Route Tables**: VPCs use **route tables** to control where network traffic goes, like directing internet traffic to the Internet Gateway or sending data between subnets.

If you did not understand anything from the above flow, that's fine – we are going to learn step by step anyway in the coming pages

Topics that we are going to learn, if we learn below 2 topics, we will be done with the VPC

1. Subnets & Types
2. Gateways & Types

1. Subnets & Types

A subnet (short for "subnetwork") is a segment of a VPC's IP address range that you can use to organize and isolate resources within the VPC. Subnets allow you to partition your VPC into smaller, manageable segments, each with its own IP address range.

Purpose of Subnets Inside a VPC

1. **Isolation and Organization:**
   - Subnets allow you to isolate different types of resources (e.g., databases, web servers) within your VPC. This helps in organizing resources based on their function, security requirements, or application needs.
2. **Security:**
   - Subnets can be used to control access to resources through security groups and network access control lists (ACLs). You can place resources that need higher security (e.g., databases) in private subnets and resources that need to be publicly accessible (e.g., web servers) in public subnets.
3. **Routing and Traffic Control:**
   - Subnets enable you to configure routing rules and control traffic flow within your VPC. You can use route tables to direct traffic between subnets or to and from the internet.
4. **Network Design:**
   - Subnets help design the network architecture of your VPC, including how resources are distributed across different availability zones (AZs) for high availability and fault tolerance.

Types of Subnets

1. **Public Subnet:**
   - A subnet whose resources can be accessed from the internet. Typically, public subnets have a route to the internet via an Internet Gateway (IGW).
   - **Use Case:** Hosting web servers, load balancers, and other resources that need to be reachable from the internet.
2. **Private Subnet:**
   - A subnet where resources do not have direct access to the internet. Private subnets often route traffic through a Network Address Translation (NAT) Gateway or NAT Instance to access the internet indirectly.
   - **Use Case:** Hosting databases, application servers, and backend services that do not need direct internet access but require internet access for updates or other purposes.
3. **VPN-Only Subnet:**
   - A subnet designed for resources that need to communicate with an on-premises network via a VPN connection. This subnet does not have internet access.
   - **Use Case:** For secure communication with on-premises systems over a VPN connection.
4. **AWS PrivateLink Subnet:**
   - A subnet configured to use AWS PrivateLink for connecting to AWS services and applications privately.
   - **Use Case:** For private communication with AWS services without traversing the public internet.

Dividing IP Address Space Between Subnets

To divide the IP address space between subnets within a VPC:

1. **Determine the VPC CIDR Block:**
    - The CIDR block defines the total IP address range for your VPC. For example, a VPC with a CIDR block of 10.0.0.0/16 has 65,536 IP addresses.
    - **Example:** 10.0.0.0/16
2. **Plan Your Subnet Sizes:**
    - Decide on the size of each subnet based on the number of IP addresses required for each subnet. Subnet sizes are defined using CIDR notation (e.g., 10.0.1.0/24 provides 256 IP addresses).
    - **Example:** If you need 50 IP addresses, a /26 subnet (64 IP addresses) might be appropriate.
3. **Create Subnets:**
    - Allocate IP address ranges within the VPC CIDR block to create subnets. Ensure that each subnet's IP range does not overlap with others and fits within the VPC's CIDR block.
    - **Example:** For a VPC with 10.0.0.0/16, you might create:
        - 10.0.1.0/24 (Public Subnet)
        - 10.0.2.0/24 (Private Subnet)
4. **Avoid Overlapping Ranges:**
    - Ensure that subnets do not overlap in their IP address ranges. Each subnet must have a unique IP range within the VPC.
5. **Use CIDR Notation:**
    - Define the IP address ranges for each subnet using CIDR notation, which specifies the starting IP address and subnet mask.
    - **Example:** 10.0.1.0/24 specifies a subnet with IP addresses from 10.0.1.0 to 10.0.1.255.

Example of Dividing IP Space

Suppose you have a VPC with a CIDR block of 10.0.0.0/16, and you want to create the following subnets:

1. **Public Subnet**:
    - **CIDR Block**: 10.0.1.0/24
    - **IP Range**: 10.0.1.0 to 10.0.1.255
    - **Purpose**: Hosts resources that need internet access.
2. **Private Subnet**:
    - **CIDR Block**: 10.0.2.0/24
    - **IP Range**: 10.0.2.0 to 10.0.2.255
    - **Purpose**: Hosts resources without direct internet access.

2. Gateways & Types

Internet Gateways (IGW) and **NAT Gateways** are crucial components in AWS VPC networking, but they serve different purposes and are used in different scenarios,

Internet Gateway (IGW)

Internet Gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

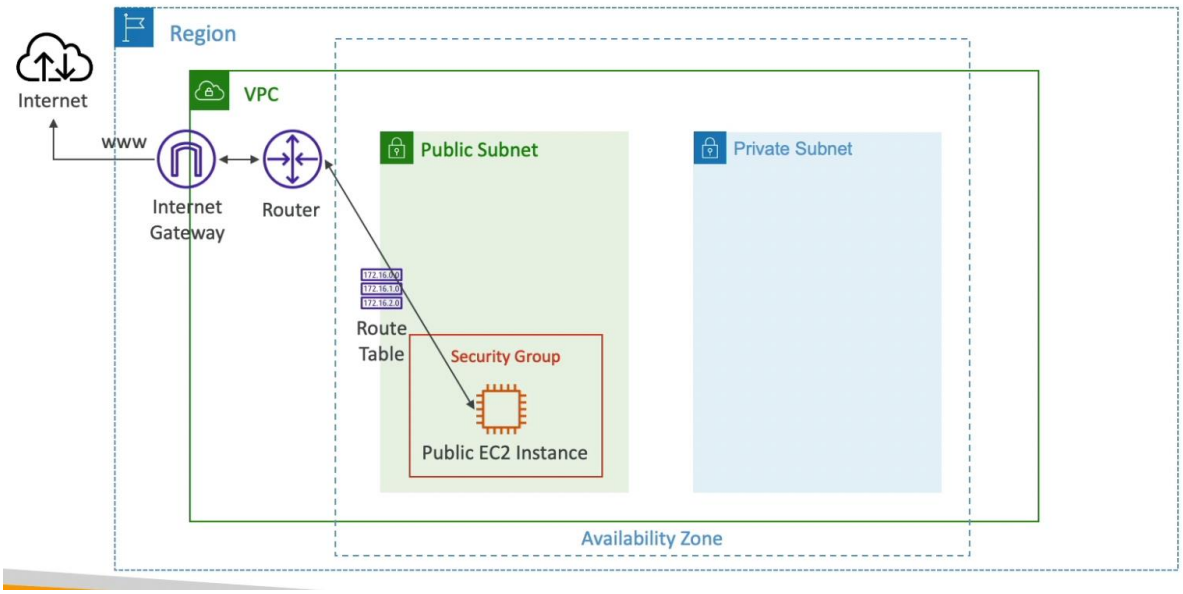
Public Subnets: Instances in a public subnet use an IGW to communicate with the internet.

Why We Use It:

- **Direct Internet Access**: Instances that need to be accessible from the internet (e.g., web servers, load balancers) must be in a public subnet with a route to an IGW.
- **Inbound and Outbound Traffic**: IGWs support both inbound and outbound traffic to/from the internet.

How It Works:

- **Route Table Configuration**: You configure the route table for the public subnet to direct outbound traffic to the IGW.
- **Public IP or Elastic IP**: Instances in a public subnet typically have a public IP or an Elastic IP address to facilitate communication with the internet.



Network Address Translation Gateway (NAT)

NAT Gateway is a managed network address translation (NAT) service that allows instances in a private subnet to access the internet while preventing inbound internet traffic from directly accessing those instances.

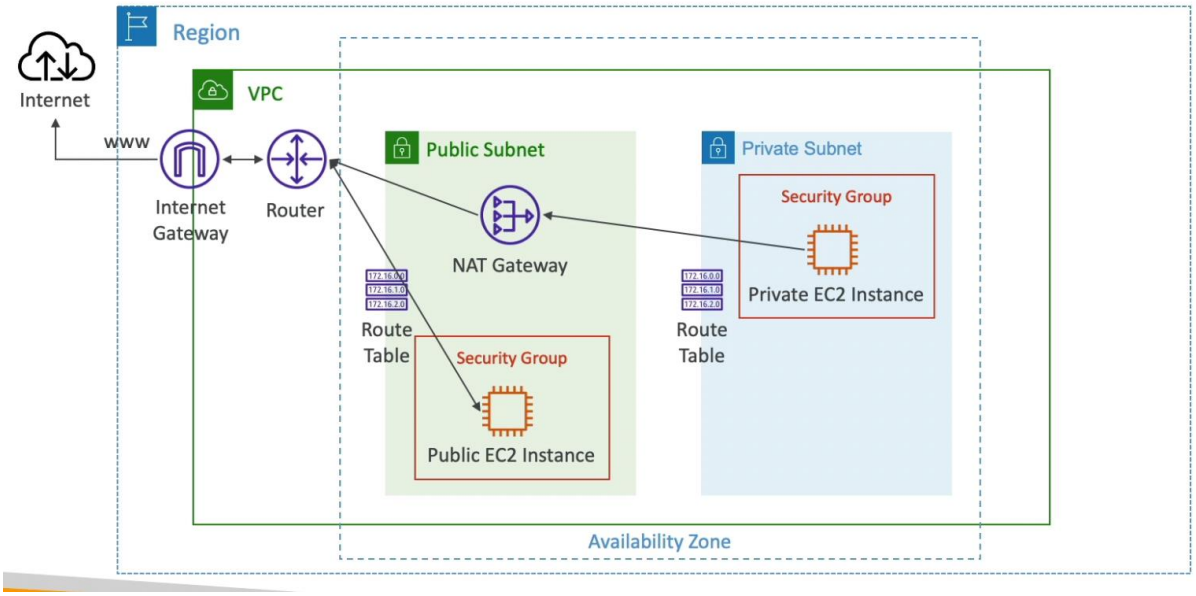
Private Subnets: Instances in a private subnet use a NAT Gateway to access the internet for updates, software downloads, or external API calls.

Why We Use It:

- **Outbound Internet Access for Private Instances**: Instances in a private subnet often need to make outbound connections to the internet for updates or external resources without being exposed to inbound internet traffic.
- **Security**: NAT Gateway prevents direct inbound traffic from the internet to instances in the private subnet, improving security.

How It Works:

- **Route Table Configuration**: You configure the route table for the private subnet to direct outbound traffic to the NAT Gateway.
- **NAT Gateway Location**: NAT Gateways are typically placed in a public subnet to handle outbound traffic from private subnets.



Summary of Differences

Feature	Internet Gateway (IGW)	NAT Gateway
Purpose	Provides internet access to public subnets.	Allows private subnets to access the internet.
Subnets	Used with public subnets.	Used with private subnets.
Inbound Traffic	Supports inbound and outbound traffic.	Only supports outbound traffic; inbound traffic is blocked.
Usage	Public instances need direct internet access.	Private instances need to reach the internet but remain inaccessible from it.

There are few more terms, which are highly important for VPC concept is

- Network Access Control List (NACL)
- VPC Peering
- VPC Endpoints
- VPC Flow logs

Let's start with NACL

Network Access Control List (NACL)

Network Access Control Lists (NACLs) are a crucial component of AWS VPC networking, providing a layer of security at the subnet level. They help manage inbound and outbound traffic to and from your subnets.

What is a NACL?

- NACLs are stateless firewalls that control network traffic at the subnet level in a VPC. They allow or deny traffic based on rules you define.
- **Scope**: NACLs operate at the subnet level, applying rules to all traffic entering or leaving a subnet.
- **Stateless**: NACLs do not maintain any memory of past traffic; each request is evaluated independently.

Why We Need NACLs

1. **Additional Layer of Security**:
   - **Subnet-Level Control**: NACLs provide an additional layer of security beyond Security Groups. While Security Groups are applied at the instance level, NACLs are applied at the subnet level.
   - **Granular Control**: NACLs allow you to define rules for traffic based on IP addresses, protocols, and ports.
2. **Traffic Filtering**:
   - **Inbound and Outbound Rules**: You can specify rules to allow or deny specific types of traffic based on source and destination IP addresses, ports, and protocols.
   - **Deny Rules**: Unlike Security Groups, NACLs can explicitly deny traffic, giving you fine-grained control over what traffic is allowed or blocked.
3. **Default Security**:
   - **Default NACL**: When you create a VPC, AWS automatically creates a default NACL that allows all inbound and outbound traffic. You can modify this default NACL or create custom NACLs to suit your needs.
4. **Logging and Monitoring**:
   - **Audit Traffic**: By configuring NACLs, you can monitor and audit traffic patterns to identify potential security threats or misconfigurations.

How NACLs Work

1. **Rules:**
    - **Rules:** Each NACL consists of a set of rules that you define. Rules are evaluated in order, starting from the lowest rule number.
    - **Allow or Deny:** Each rule specifies whether traffic should be allowed or denied based on criteria such as IP address, port range, and protocol.
2. **Rule Evaluation:**
    - **Stateless Evaluation:** NACLs evaluate each packet individually. If a rule matches the packet, the associated action (allow or deny) is taken. If no rules match, the default action (usually deny) is applied.
3. **Default NACL:**
    - **Default Behaviour:** The default NACL allows all inbound and outbound traffic unless modified. Custom NACLs can be created to enforce stricter rules.
4. **Association:**
    - **Subnet Association:** Each subnet in a VPC can be associated with one NACL. If you do not specify a NACL, the default NACL is used.

NACL vs. Security Groups

- **Security Groups:**
  - Applied at the instance level.
  - Stateful: Return traffic is automatically allowed if an outgoing request is allowed.
  - Typically used to control traffic to and from individual instances.
- **NACLs:**
  - Applied at the subnet level.
  - Stateless: Each request is evaluated independently; return traffic must be explicitly allowed.
  - Useful for controlling traffic across entire subnets.

Security Group	NACL
Firewall for EC2 (applied to ENI)	Firewall for subnets
Supports only Allow rules	Supports both Allow and Deny rules
Stateful (only request will be evaluated against the SG rules)	Stateless (both request and response will be evaluated against the NACL rules)
All rules are evaluated	Only the first matched rule is considered

VPC Peering

VPC Peering is a networking connection between two VPCs (Virtual Private Clouds) that allows them to communicate with each other as if they were within the same network. This communication happens over private IP addresses, without traversing the public internet.

Why We Need VPC Peering

1. **Resource Sharing**:
   - **Inter-VPC Communication**: Allows instances in different VPCs to communicate with each other directly. This is useful when you have resources in separate VPCs that need to interact.
2. **Multi-VPC Architecture**:
   - **Separation of Environments**: Enables a clean separation between different environments (e.g., development, staging, production) while allowing them to securely communicate as needed.
3. **Cost Efficiency**:
   - **Reduced Costs**: Communication over VPC peering is generally cheaper than using public internet or VPN connections.
4. **Security and Compliance**:
   - **Private Communication**: Provides a private and secure communication channel between VPCs without exposing data to the public internet.
5. **Simplified Network Design**:
   - **Centralized Services**: Allows you to centralize services such as logging or monitoring in one VPC while other VPCs access these services via peering.

How VPC Peering Works

1. **Initiate Peering:**
   - One VPC owner initiates a peering connection request to the other VPC owner. This can be within the same AWS account or between different AWS accounts.
2. **Accept Peering Request:**
   - The owner of the recipient VPC must accept the peering connection request.
3. **Update Route Tables:**
   - Both VPCs must update their route tables to route traffic destined for the peered VPC's CIDR block through the peering connection.
4. **Configure Security Groups:**
   - Security groups should be updated to allow traffic from the peered VPC's IP range.

Considerations

1. **CIDR Overlap:**
   - **Non-overlapping CIDR Blocks:** VPC peering does not support overlapping CIDR blocks. Ensure that the CIDR blocks of the VPCs are unique.
2. **Transit Gateway:**
   - **Complex Network Architectures:** For more complex networking needs, such as connecting multiple VPCs or managing cross-region peering, consider using AWS Transit Gateway.
3. **Peering Limits:**
   - **Limits:** AWS imposes limits on the number of VPC peering connections per VPC. Check current limits and consider your network design.
4. **Region:**
   - **Same Region:** VPC peering connections are typically within the same region. Cross-region peering is possible but requires additional setup.

VPC Endpoints

A **VPC Endpoint** is a service provided by AWS that allows you to connect your VPC to AWS services and other resources within the AWS network without traversing the public internet. VPC Endpoints are crucial for enhancing security, improving performance, and reducing costs.

Types of VPC Endpoints

There are two main types of VPC Endpoints:

Interface Endpoints:

- Interface Endpoints provide private connectivity to AWS services using private IP addresses within your VPC. They create Elastic Network Interfaces (ENIs) in your subnets with private IP addresses to connect to the AWS service.
- Use Case: Commonly used for services like Amazon S3, DynamoDB, and AWS Secrets Manager, as well as for accessing AWS services from within your VPC.
- Example Services: Amazon S3, Amazon DynamoDB, AWS Systems Manager.

How It Works:

- You create an Interface Endpoint by specifying the service you want to connect to. AWS provisions ENIs in your subnet for this endpoint.
- DNS: AWS provides DNS names for the service that resolves to the private IP addresses associated with the ENIs.
- Access: Traffic destined for the service is routed through the private IP addresses provided by the Interface Endpoint.

Gateway Endpoints:

- Description: Gateway Endpoints are used specifically for Amazon S3 and DynamoDB. They create a target for a route table entry, allowing traffic to the specified service to be routed through the gateway.
- Use Case: Ideal for accessing Amazon S3 and DynamoDB from within your VPC without needing public internet access.
- Example Services: Amazon S3, Amazon DynamoDB.

How It Works:

- Creation: You create a Gateway Endpoint and specify the service (Amazon S3 or DynamoDB) and the route table(s) to associate with the endpoint.
- Route Tables: A route table entry is created for the specified service, directing traffic for that service through the Gateway Endpoint.
- Access: Traffic to the service is automatically routed through the Gateway Endpoint without leaving the AWS network.

VPC Flow Logs

VPC Flow Logs are a feature of AWS that allow you to capture information about the IP traffic going to and from network interfaces in your Virtual Private Cloud (VPC). They are useful for monitoring, troubleshooting, and analysing network traffic within your VPC.

Why We Need VPC Flow Logs:

1. **Traffic Monitoring:** They provide visibility into network traffic patterns, helping you understand which resources are communicating and how much data is being transferred.
2. **Security Analysis:** By capturing data on all traffic, you can identify potential security threats or suspicious activity. They are useful for auditing and compliance purposes.
3. **Troubleshooting:** Flow logs help diagnose network connectivity issues by providing insights into traffic that is being allowed or denied by security groups or network access control lists (ACLs).
4. **Performance Analysis:** They help in analysing the performance of your network by showing traffic patterns and identifying bottlenecks.

Types of VPC Flow Logs:

There are three types of flow logs you can capture:

1. **Accepted:** Logs only the traffic that is allowed by your security groups and network ACLs.
2. **Rejected:** Logs only the traffic that is denied by your security groups and network ACLs.
3. **All:** Logs all traffic, regardless of whether it is allowed or denied.

VPC Scenario based questions

https://lisireddy.medium.com/aws-vpc-scenario-based-questions-0bc91cf20266

Wish you the best …! Happy Learning ..!

Yours' Love (@lisireddy across all the platforms)