AWS Certified Solutions Architect

Associate C03

Module 4

AWS Virtual Private Cloud

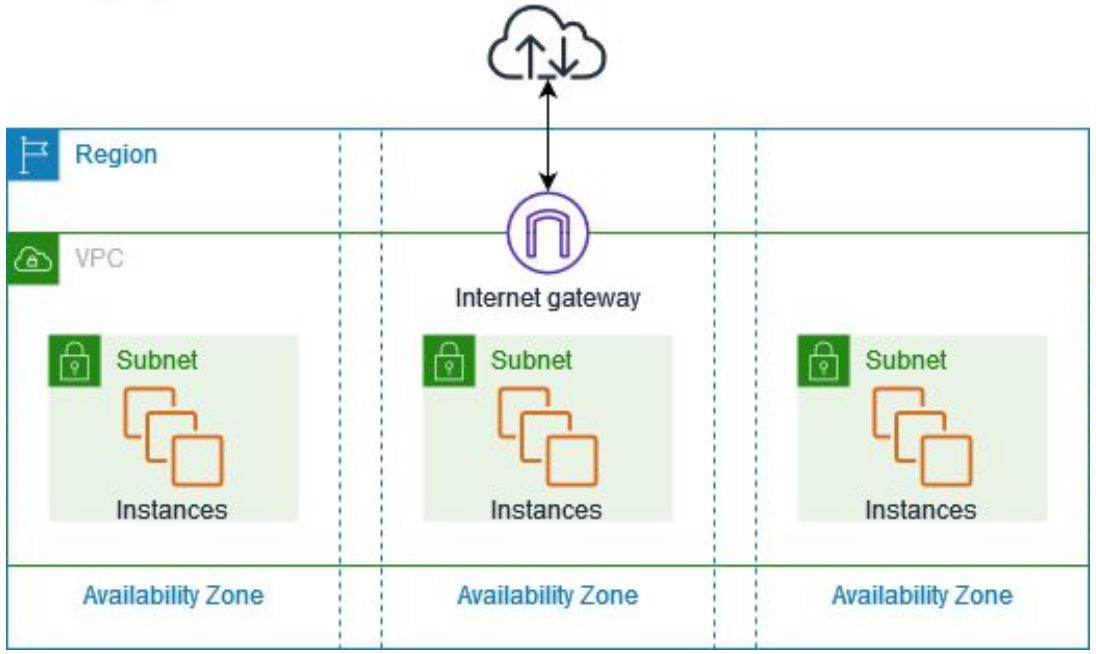
Agenda

1. Virtual Private Cloud
2. Subnets and Routes
3. Types of IPs
4. Gateways in VPC
5. Security Groups and NACLs
6. VPC Peering
7. VPC Endpoints
8. Transit Gateways
9. VPC Flow Logs and Reachability Analyzer

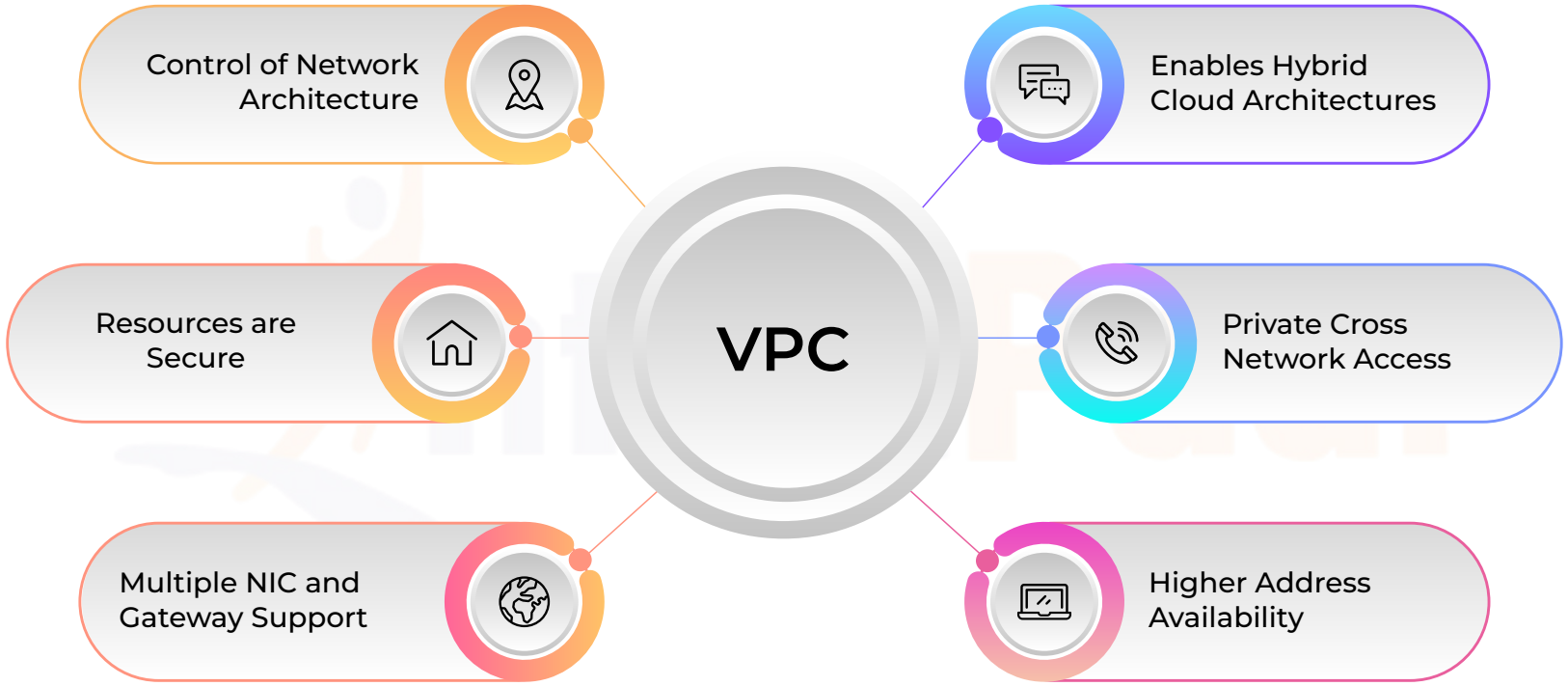
Virtual Private Cloud

What Is a VPC?

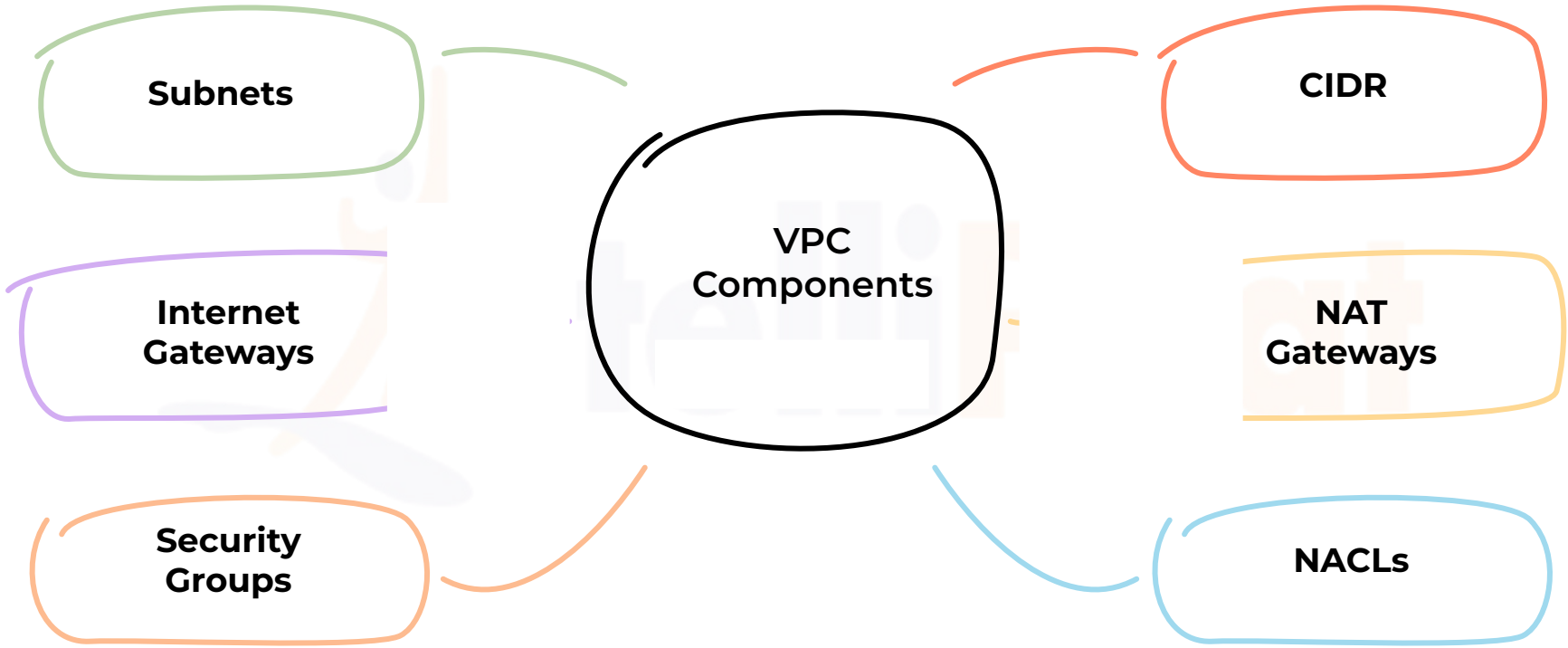
A VPC allows users to create **isolated sections** of the cloud where they can **launch resources** like **virtual machines, databases**, and **storage**. Think of a VPC like your own private space in the cloud. Just like you have a personal room in a shared house, a VPC gives you a dedicated area AWS.



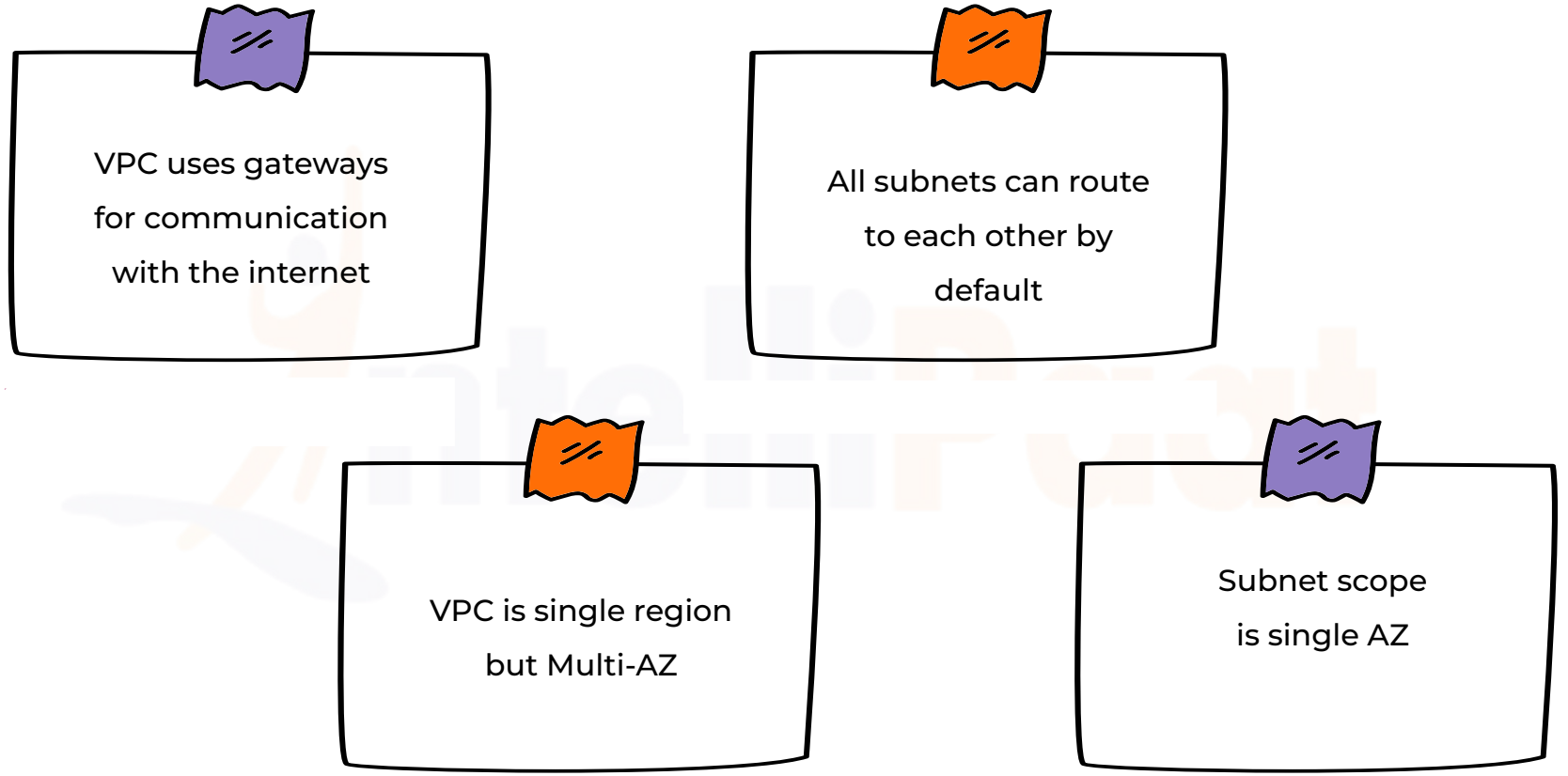
Why Use VPC?


Control of Network Architecture
Enables Hybrid Cloud Architectures
Resources are Secure
VPC
Private Cross Network Access
Multiple NIC and Gateway Support
Higher Address Availability

VPC Components



VPC Topology



Subnets and Routes

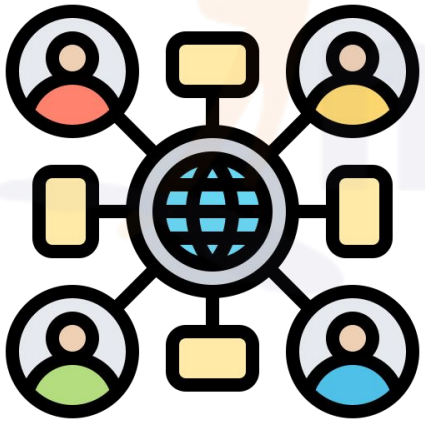
What are Subnets?

Subnets are **smaller sections** within a VPC where devices can communicate directly. Each subnet has its **own address range** and can contain specific types of devices. They help **organize** and **manage** network traffic efficiently.

Types of Subnets in a VPC

In a VPC, there are mainly two types of subnets: **Public** and **Private**.

Public subnets allow resources to communicate **directly** with the **internet**, while **private subnets** are shielded from the internet and **rely on a NAT gateway** or instance for outbound connectivity, enhancing security for sensitive data.

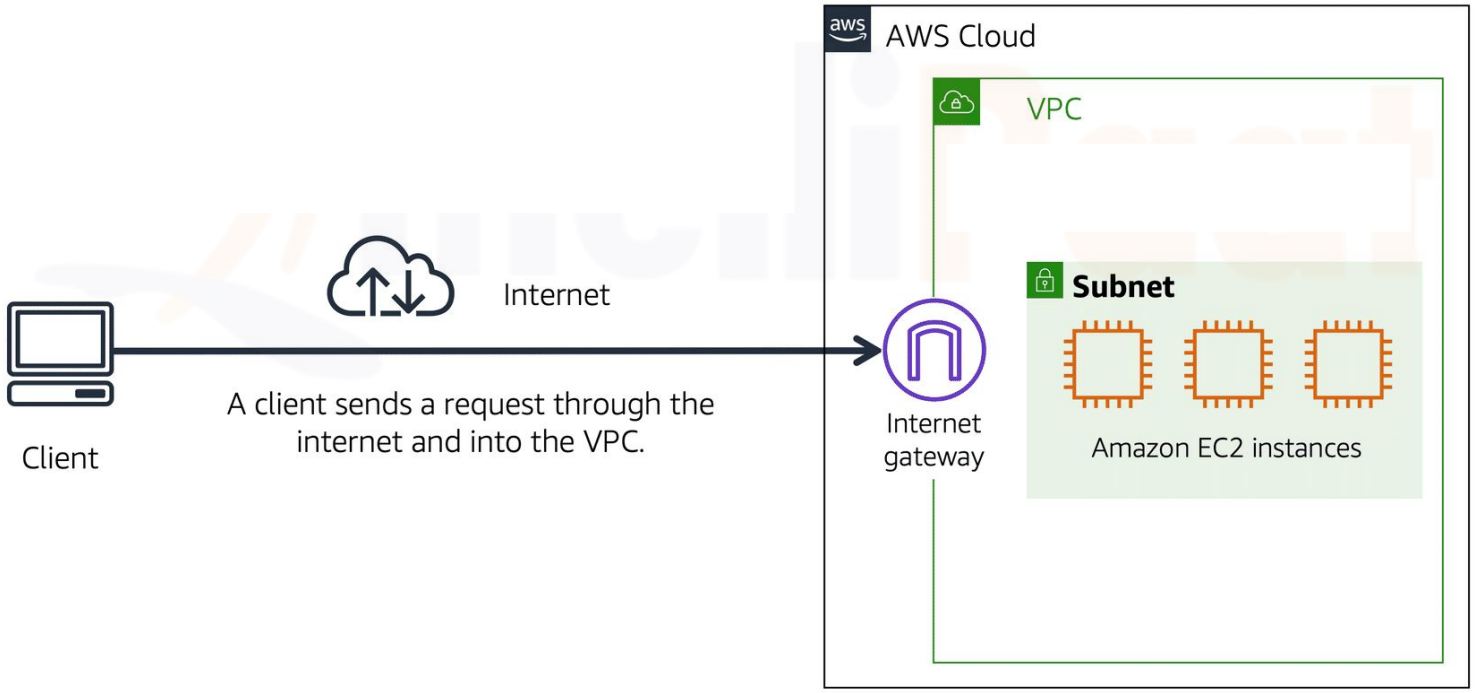


Public Subnets

Private Subnets

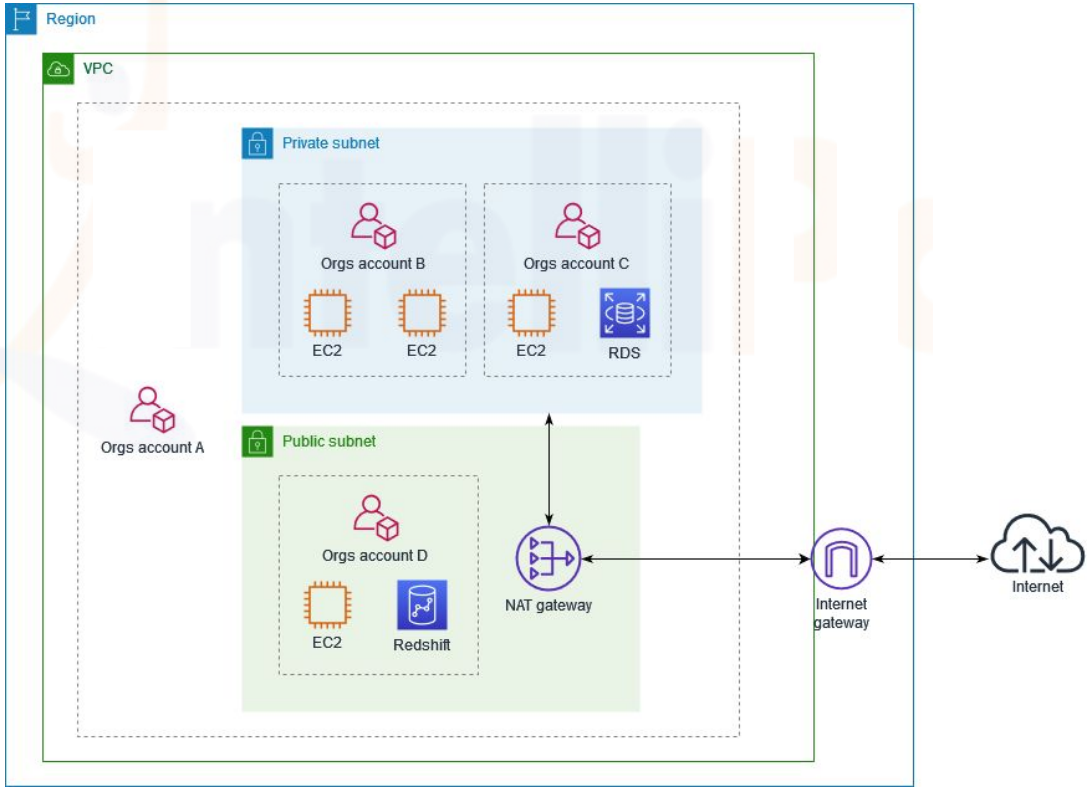
VPC with only Public Subnets

Subnets are **smaller sections** within a VPC where devices can communicate directly. Each subnet has its **own address range** and can contain specific types of devices. They help **organize** and **manage** network traffic efficiently.



VPC with Public and Private subnets

A VPC with **public** and **private subnets** creates a secure environment in the cloud. **Public subnets** host **resources accessible** from the internet, like **web servers**. **Private subnets** house **sensitive data** or backend systems, **shielded** from direct internet access, **ensuring better security** and **control over network traffic.**



Route Tables

A **route table** is a fundamental component of networking that is used in **routing packets** of data from one network location to another.

Route tables in AWS define rules for **directing traffic** between **subnets** and the **internet gateway**, allowing for efficient network communication within a virtual private cloud (VPC).

Routes in Route Tables

In AWS, a Route table contains a **set of rules**, called **routes**, that are used to determine where network traffic from your VPC is directed. Routes can **specify destinations**, such as an **Internet Gateway** for public access or a **Virtual Private Gateway** for VPN connections, among other options.

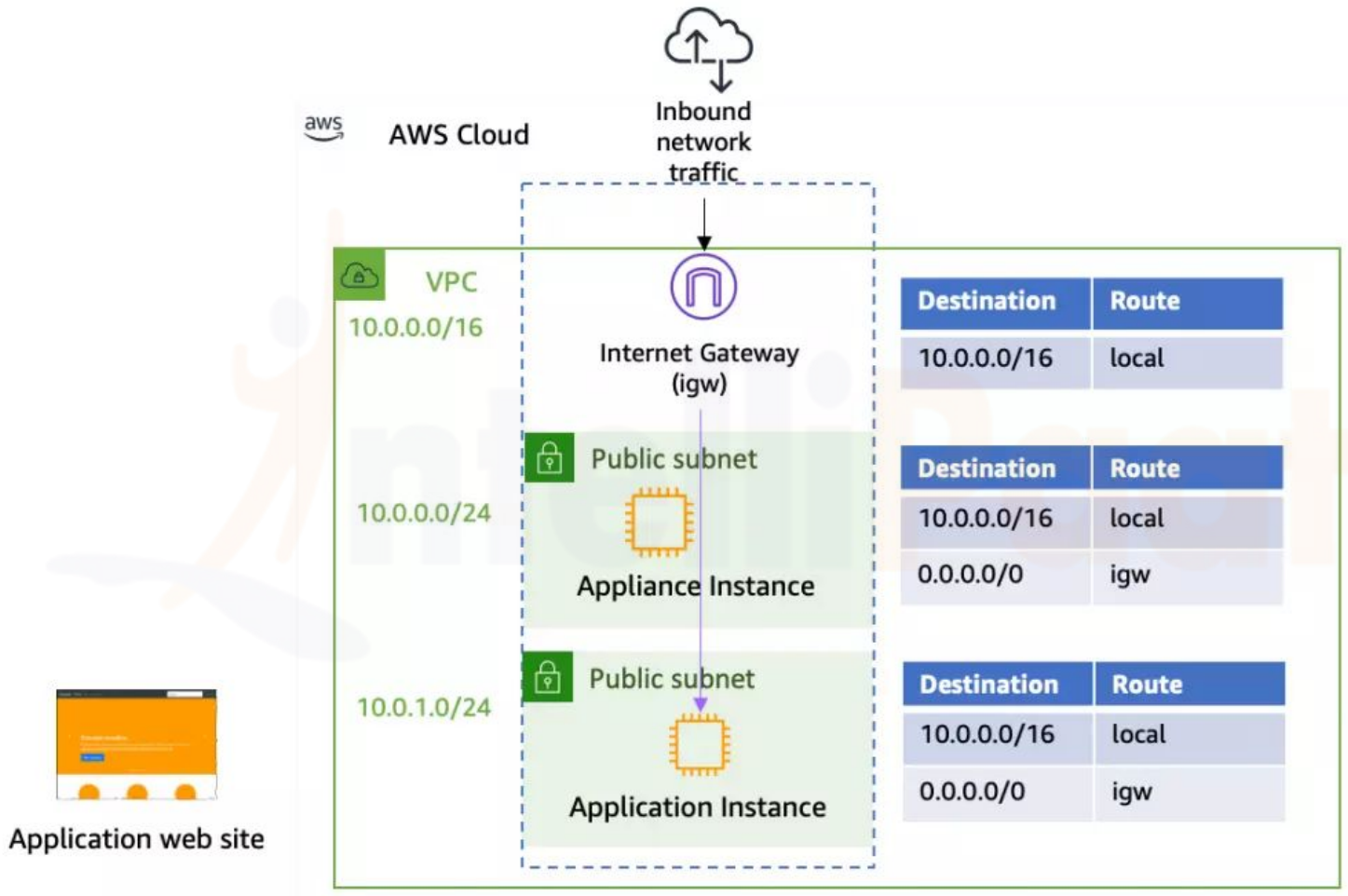
Custom Route Table

Target	Destination
local	10.0.0.0/16
internet-gateway-id	0.0.0.0/0

Main Route Table

Target	Destination
local	10.0.0.0/16
nat-gateway-id	0.0.0.0/0

Diagram of Routes in AWS



Hands-On: Create Custom VPC and Its Resources

- Open the Amazon VPC console.

- Create the **VPC**.

- In the console, **create a subnet**, selecting the VPC, Availability Zone, and IPv4 CIDR block.

- Create a **route table**, naming it and selecting the VPC.

- **Associate the route table** with the subnet.

- Open the Route tables section, select the **route table**, and on the **Subnet associations** tab, choose **Edit Subnet Associations**.

- Select the subnet and **save the associations**.

- Create Internet Gateway to associate with Public Subnets

- Launch instances to verify the configurations

Types of IPs

Different Types of IPs

01

Public IP

Automatically assigned to EC2 instances when launched in a public subnet. It allows instances to communicate over the internet

02

Private IP

Assigned to EC2 instances within a VPC. Used for internal communication within the VPC and cannot be accessed from the internet

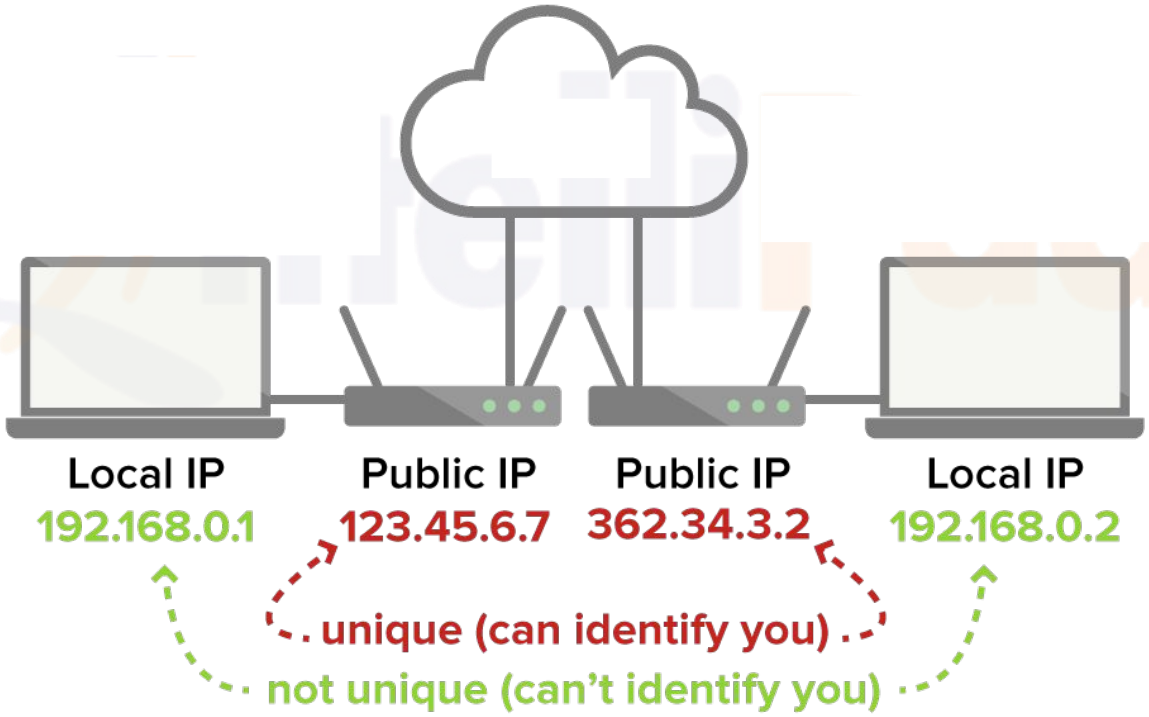
03

Elastic IP

A static IPv4 address designed for dynamic cloud computing. It can be associated with EC2 instances or network interfaces and can be moved between instances.

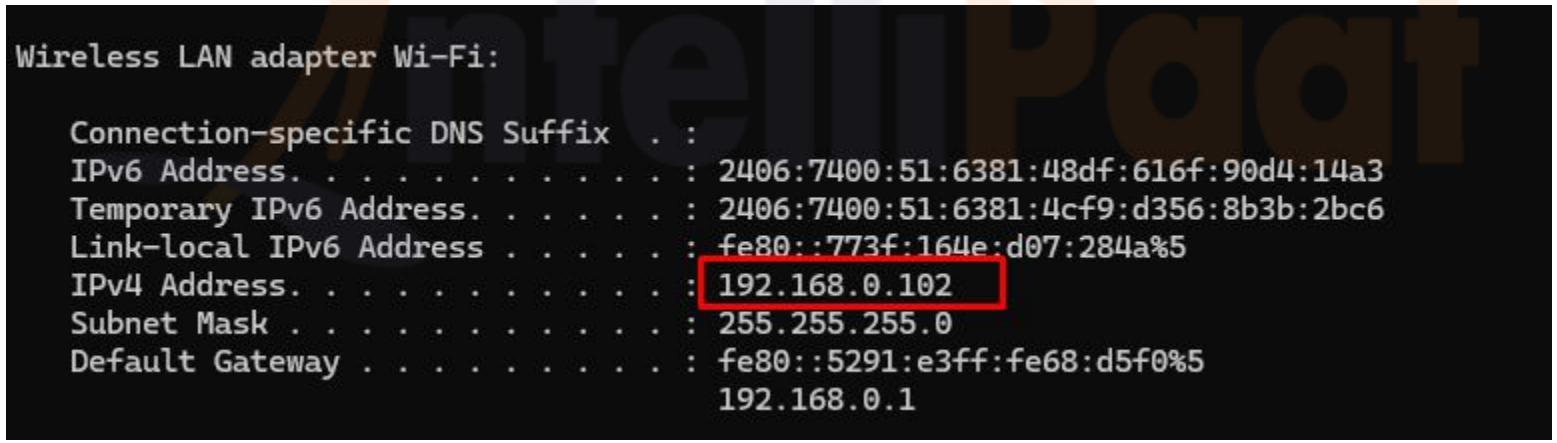
Public IP

Public IPs in AWS are assigned to resources like EC2 instances for internet communication. They facilitate inbound and outbound traffic, enabling **access to** and **from** the **internet**. Public IPs are crucial for hosting web servers, enabling remote access, and establishing communication with external services or clients.



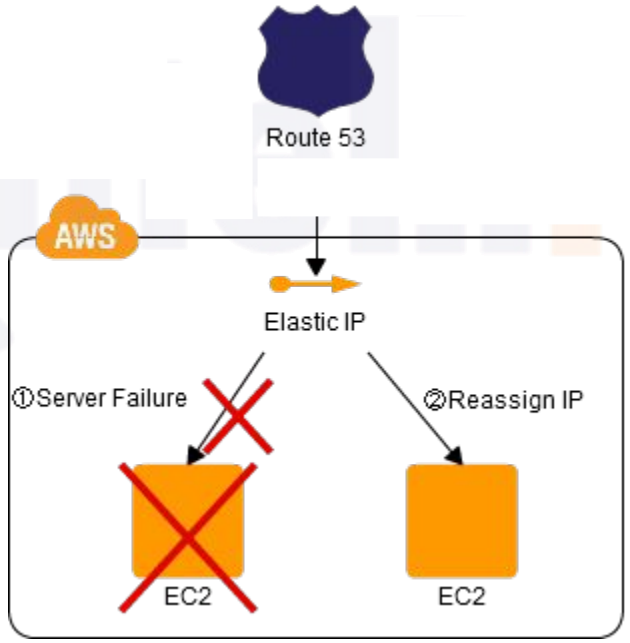
Private IP

Private IPs are assigned to resources within a VPC, facilitating **internal communication**. They ensure secure data transfer between instances, databases, and other services, shielding them from external access. Private IPs are vital for establishing **private networks**, maintaining **data privacy**, and enabling seamless communication between components in AWS infrastructure.



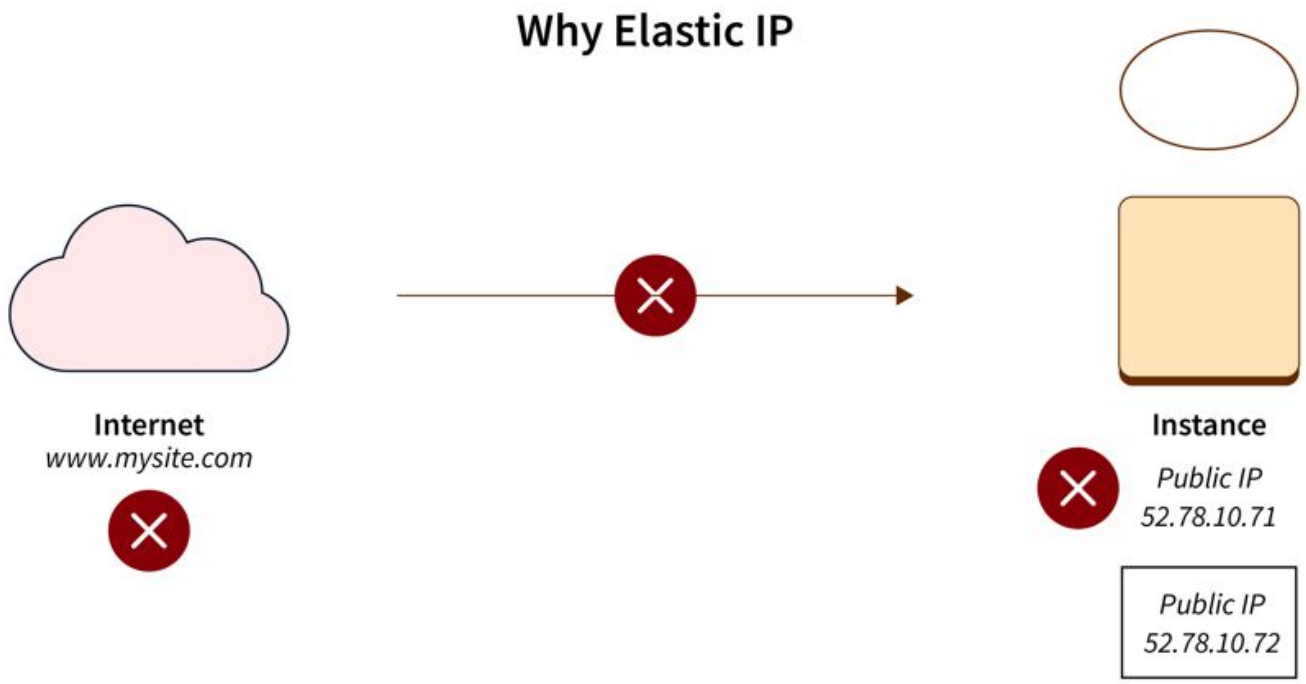
Elastic IP

Elastic IP (EIP) in AWS is a **static IPv4 address** that can be associated with EC2 instances or network interfaces. It's useful for hosting websites, applications, or services that require a fixed public IP address, remote access, or seamless failover scenarios where IP address persistence is necessary during instance replacement or maintenance.



Why Use Elastic IP?

Elastic IP (EIP) in AWS provides a static IPv4 address that can be associated with EC2 instances or network interfaces. It's valuable for scenarios requiring consistent public IP addresses, such as hosting websites or applications, remote access, or setting up secure connections. EIPs facilitate seamless failover and ensure uninterrupted service availability during instance replacements or maintenance, enhancing overall reliability and accessibility.



Rules to Use an Elastic IP

Elastic IP vs. Public IP

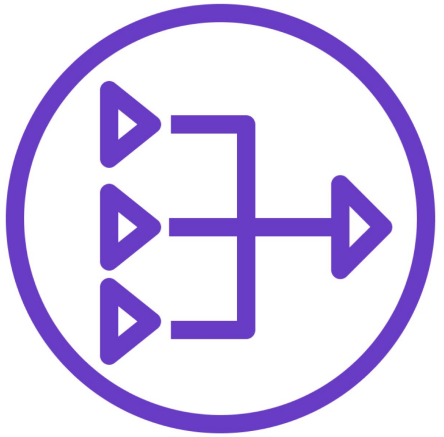
Feature	Elastic IP	Public IP
Type	Static IPv4 address	Dynamic IPv4 address assigned by AWS
Association	Can be associated/dissociated with EC2 instances or network interfaces	Automatically assigned to EC2 instances in public subnets upon launch
Persistence	Remains associated until explicitly disassociated	Changes when instance is stopped/restarted
Availability	Can be moved between instances in the same AWS region	Tied to the lifecycle of the EC2 instance
Use Cases	Ideal for long-term services requiring a fixed IP, failover scenarios, or remote access	Suitable for temporary instances or scenarios where IP changes are acceptable
Cost	Free if the Elastic IP is associated with an instance, charged if not associated or if more than one Elastic IP is allocated	Charges if assigned to any resource

Gateways in VPC

What are Gateways?

Gateways serve as crucial networking components, facilitating communication between **VPC resources** and **external networks**. They serve as the **entry** and **exit** points for network traffic, allowing resources within the VPC to connect to the internet, establish secure VPN connections with on-premises networks, and access AWS services privately without internet exposure.

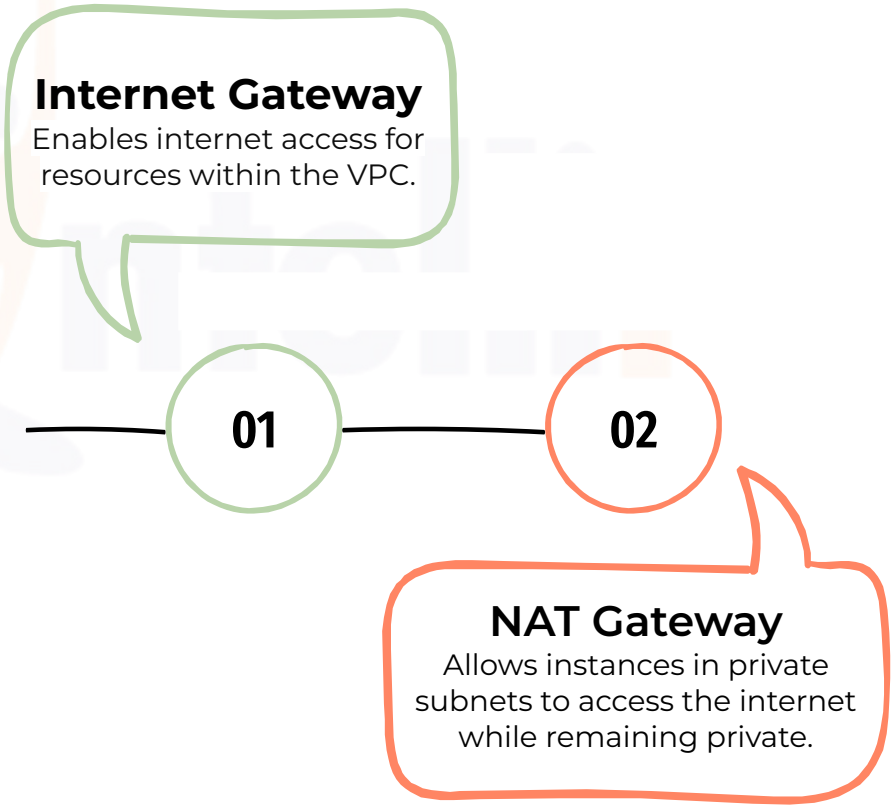
Internet Gateway



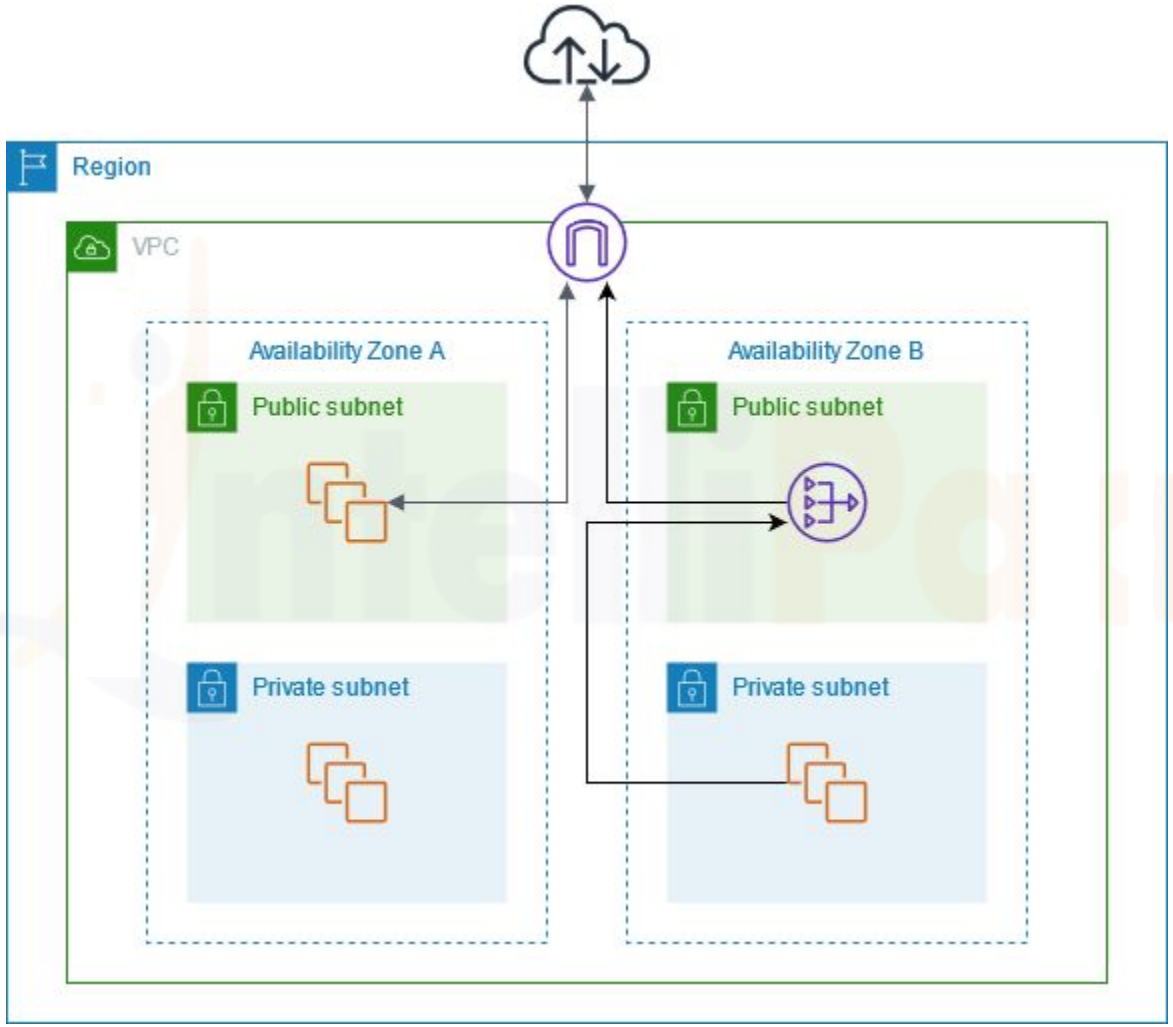
NAT Gateway

Types of Gateways

Gateways include **Internet Gateways** for internet access, **NAT Gateways** for private subnet internet access. Each gateway type plays a vital role in enabling various network functionalities within the VPC environment.

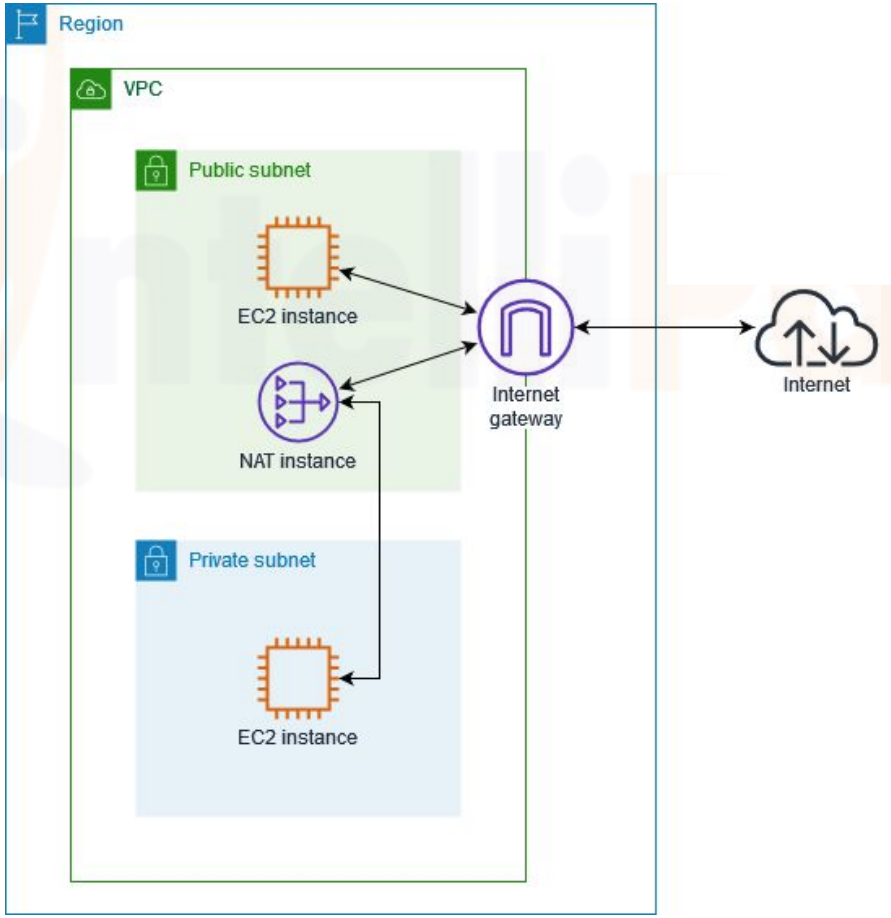


Gateways - Architecture Diagram



NAT Instances

NAT instances in AWS act as intermediaries between resources in **private subnets** and the internet. They enable **outbound internet** connectivity for instances within a VPC while keeping them **hidden from inbound** traffic.



Network Interfaces

Network Interfaces are virtual network cards attached to instances in AWS, enabling them to communicate with other instances, services, or the internet.

For example, you can attach multiple network interfaces to a single EC2 instance for different networking needs, like separating public and private traffic.

Hands-On

Working with NAT Gateway

Hands-On: Working with NAT Gateways

Add a NAT Gateway to an Existing VPC:

- Open the Amazon VPC console.

- Navigate to NAT gateways and choose **Create NAT gateway**.

- Specify a name, select the subnet, and choose the default **Public for Connectivity type**.

- Assign an **Elastic IP** allocation ID and **create the NAT gateway.**

Associate a Gateway with a Route Table:

- In the navigation pane, choose Route tables and select the route table.

- From the Routes tab, choose Edit routes and add a route for **0.0.0.0/0** with the **NAT Gateway as the target.**

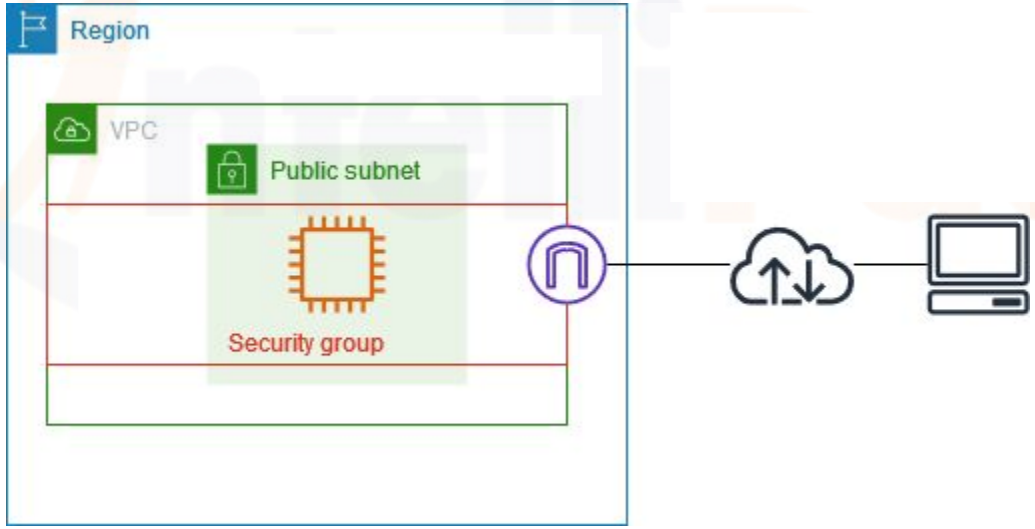
- Connect to an EC2 instance in the private subnet.

- Run a ping test on **www.google.com** to verify NAT Gateway connectivity.

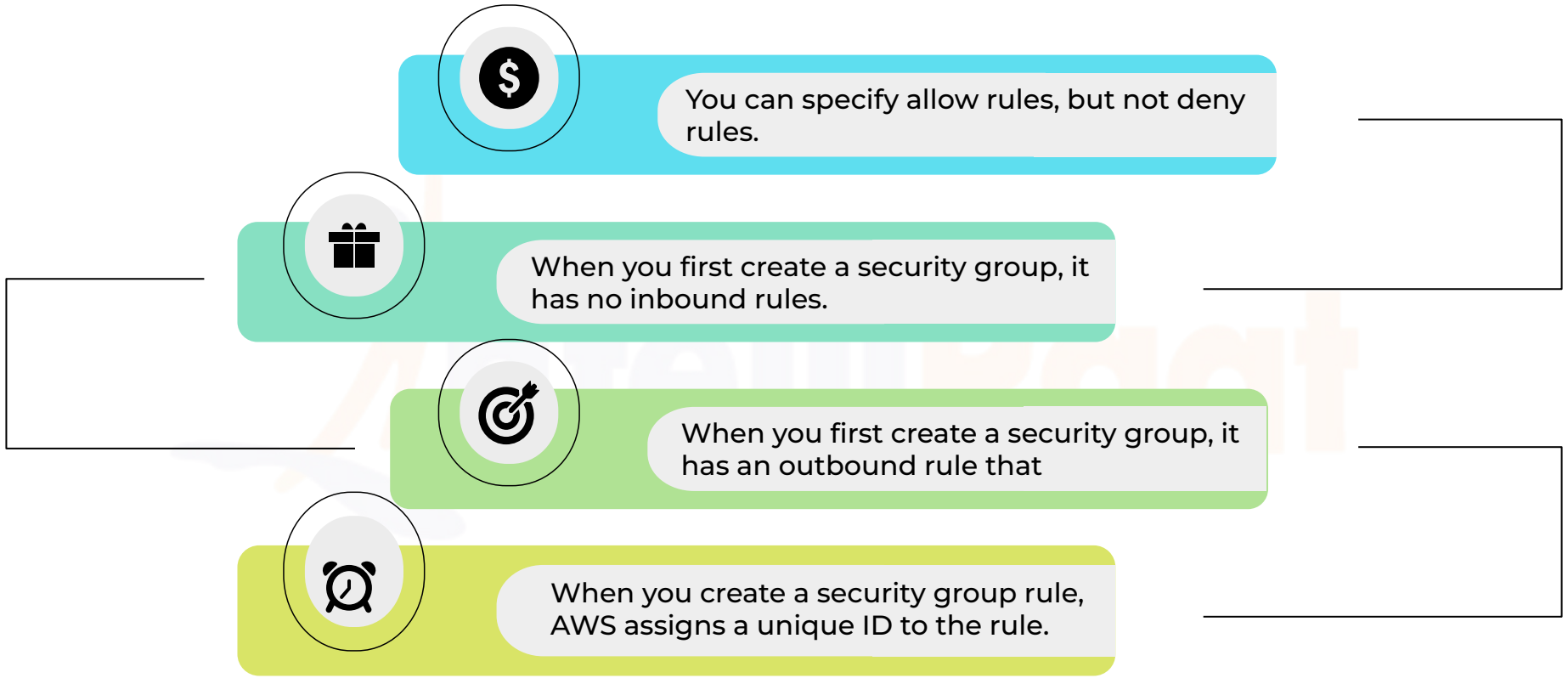
Security Groups and NACLs

Security Groups

Security Groups in AWS act as **virtual firewalls** for **EC2 instances**, controlling inbound and outbound traffic based on rules. Security Groups are **stateful**, meaning they **track the connection state**, and any rules allowing inbound traffic automatically permit the corresponding outbound traffic.

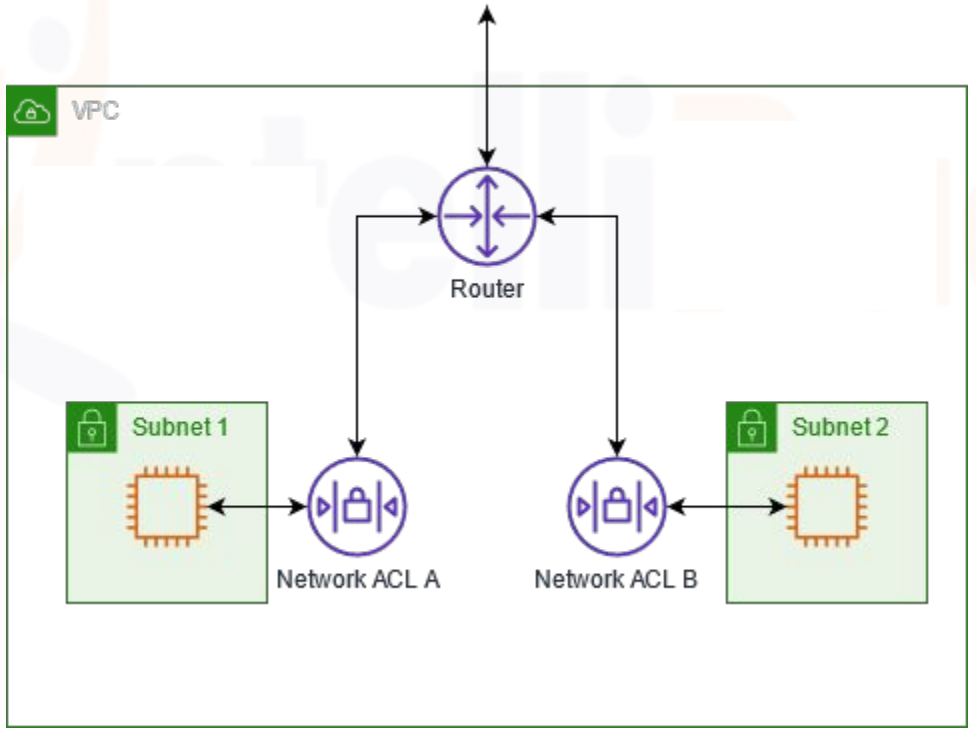


Security Group Rule



Network Access Control Lists

Network Access Control Lists (NACLs) in AWS are **stateless**, rule-based firewalls that control traffic at the **subnet level**. They **allow** or **deny** traffic based on rules defined for **inbound** and **outbound** traffic. NACLs provide an additional layer of security for VPC subnets, complementing security groups.



Network ACL Rules

Rule number	Rules are evaluated starting with the lowest numbered rule.
Type	The type of traffic; for example, SSH. You can also specify all traffic or a custom range.
Protocol	You can specify any protocol that has a standard protocol number.
Port range	The listening port or port range for the traffic.
Source	[Inbound rules only] The source of the traffic
Destination	[Outbound rules only] The destination for the traffic
Allow/Deny	Whether to allow or deny the specified traffic.

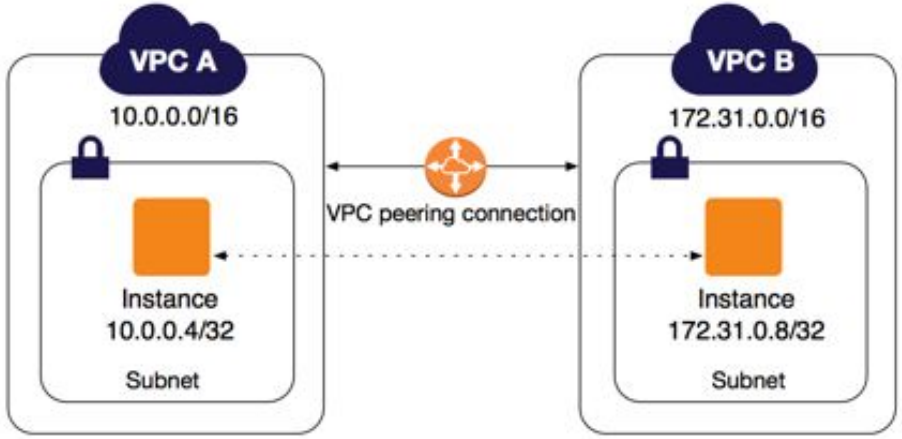
Security Groups vs. Network ACLs

Aspect	Security Group	NACLs
Scope	Instance level (EC2 instances)	Subnet level (within a VPC)
Statefulness	Stateful	Stateless
Traffic Control	Controlled via rules specifying allowed traffic	Govern traffic based on user-defined rules
Flexibility	Flexible, easy to manage	Provide broader, coarse-grained control
Use Cases	Application-level security	Setting baseline security measures

VPC Peering

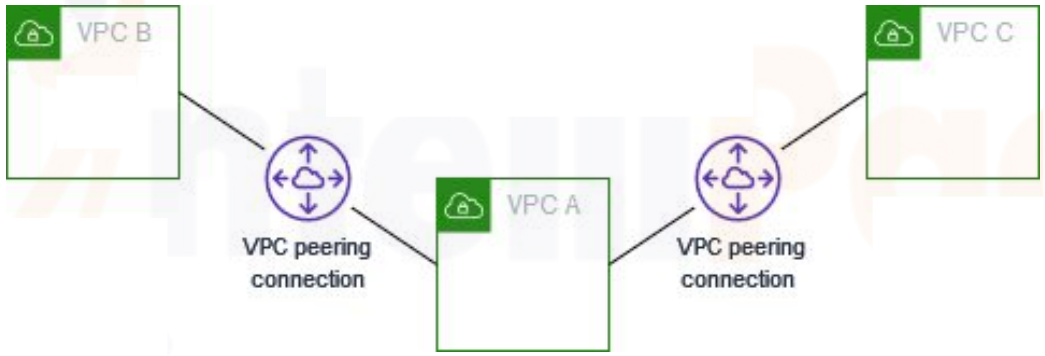
What Is VPC Peering?

VPC Peering in AWS allows **connecting** two **Virtual Private Clouds** (VPCs) within the same region or separate regions. It enables communication between resources in the peered VPCs using **private IP addresses**, as if they were on the same network. VPC peering is **secure**, **scalable**, and **simplifies network management** by eliminating the need for gateways or VPN connections between VPCs.



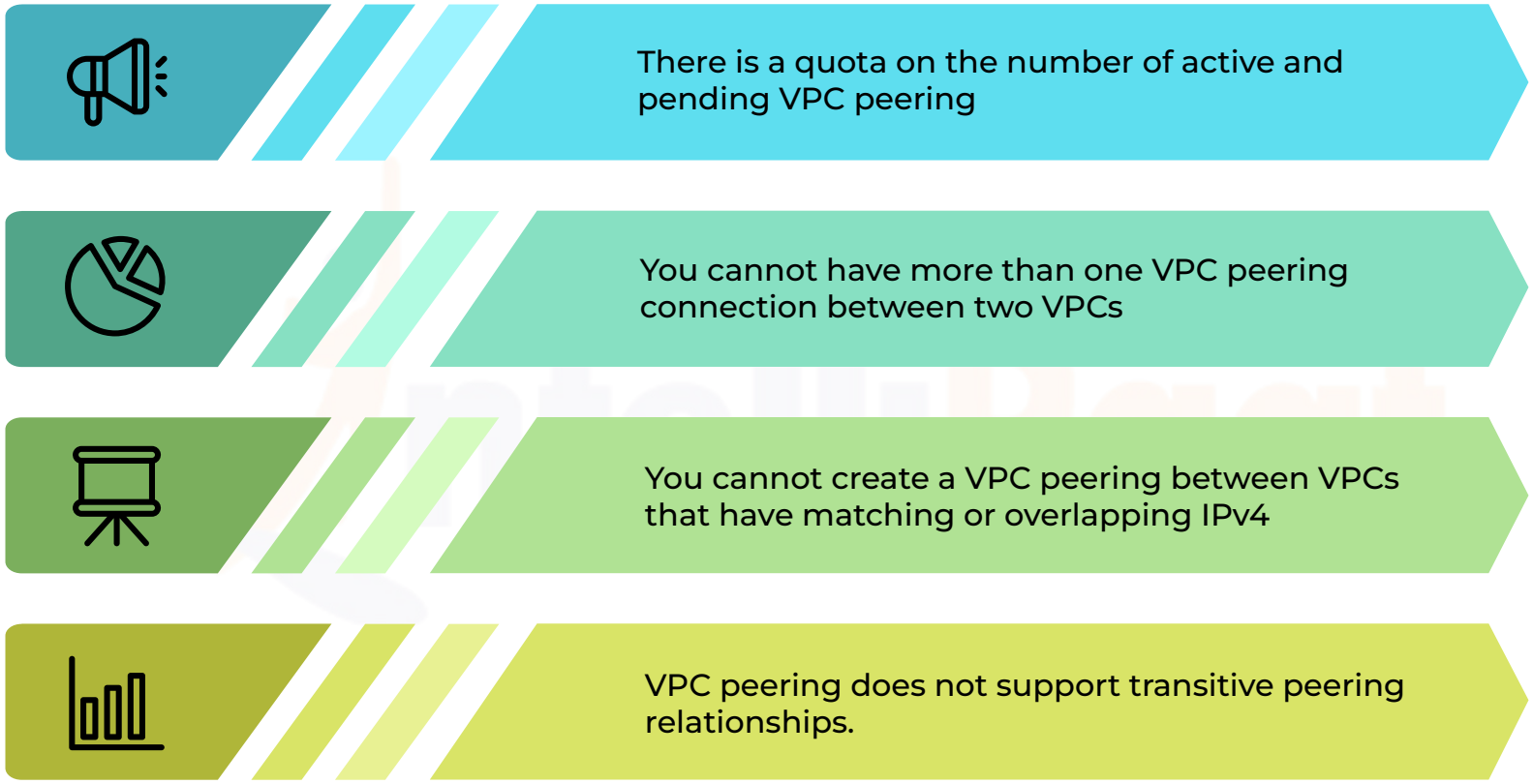
Multiple VPC Peering Connections

The following diagram is an example of **one VPC** peered to two different VPCs. There are two VPC peering connections: **VPC A** is peered with both **VPC B** and **VPC C**. **VPC B** and **VPC C** are not peered, and you cannot use VPC A as a transit point for peering between VPC B and VPC C.

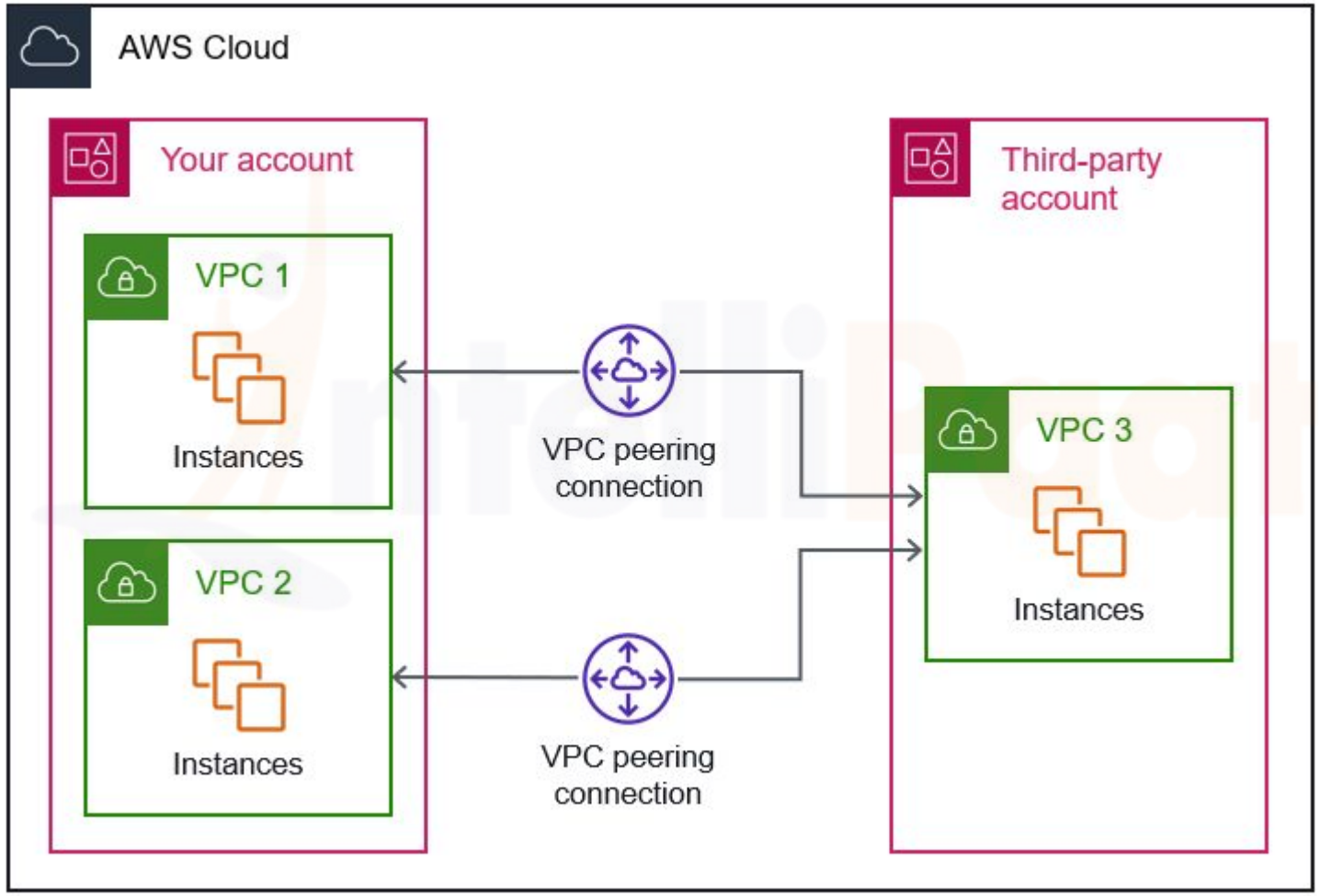


NOTE: If you want to enable routing of traffic between VPC B and VPC C, you must create a unique VPC peering connection between them.

VPC Peering Limitations



VPC Peering Architecture Diagram



Hands-On

Configure VPC Peering

Hands-On: Configure VPC Peering

Configure VPC Peering:

- Open the Amazon VPC console.
- Navigate to Peering connections and choose Create peering connection.
- Enter a name for the peering connection.
- Select the requester VPC in your account and the accepter VPC in another region.
- Choose Create peering connection.
- Select the created peering connection and choose Actions, Accept request.
- Confirm the acceptance.

VPC Endpoints

What are 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**, and **minimize data transfer costs** by keeping traffic within the AWS network.

Types of Endpoints

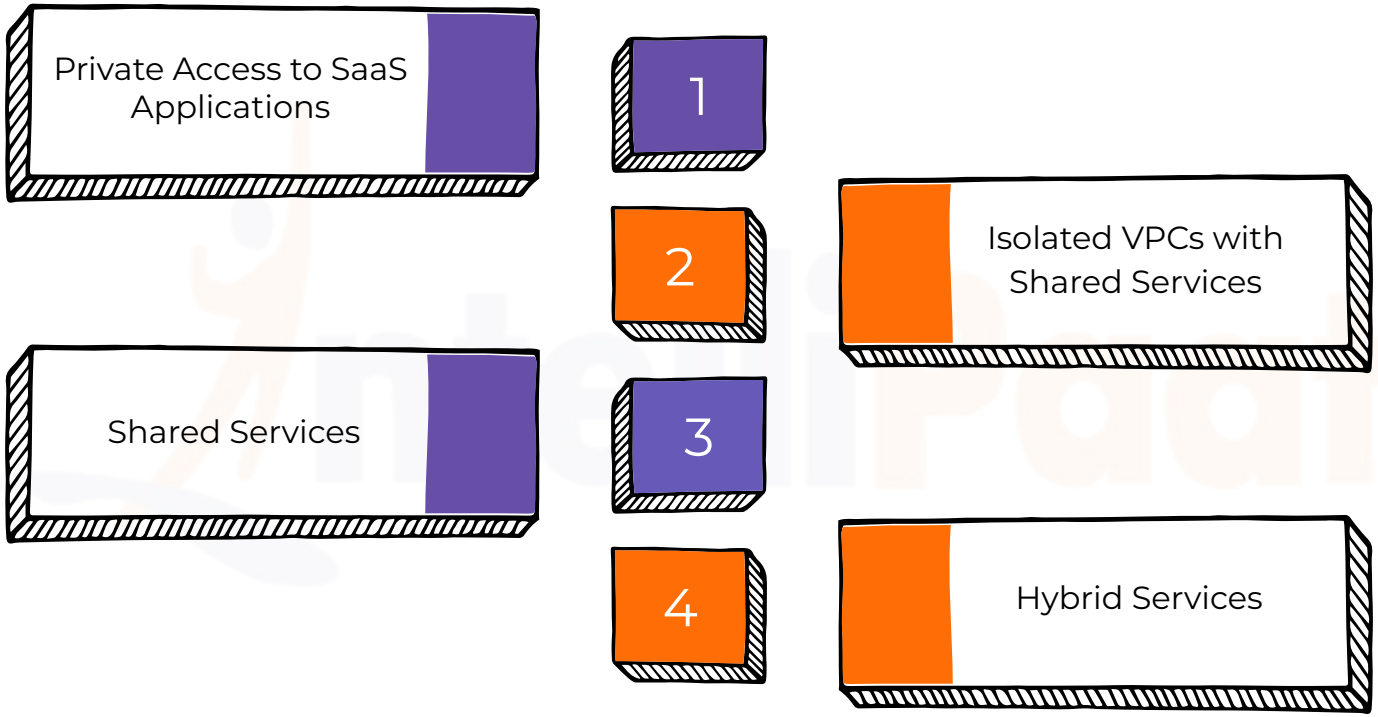
Gateway Endpoints

Allow communication between a VPC and AWS services over the AWS backbone network. Examples include S3 Gateway Endpoint and DynamoDB Gateway Endpoint

Enable private connectivity to AWS services using Elastic Network Interfaces (ENIs) within a VPC. Examples include Interface VPC Endpoints for S3, DynamoDB, and other services

Interface Endpoints

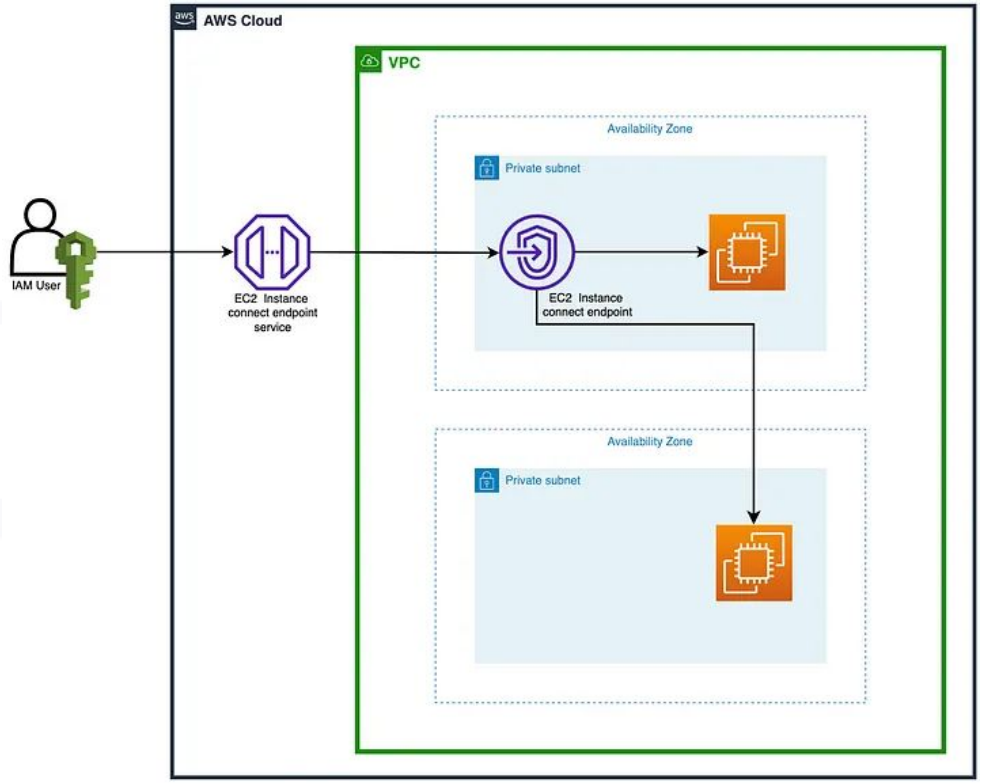
Use Cases for Endpoints



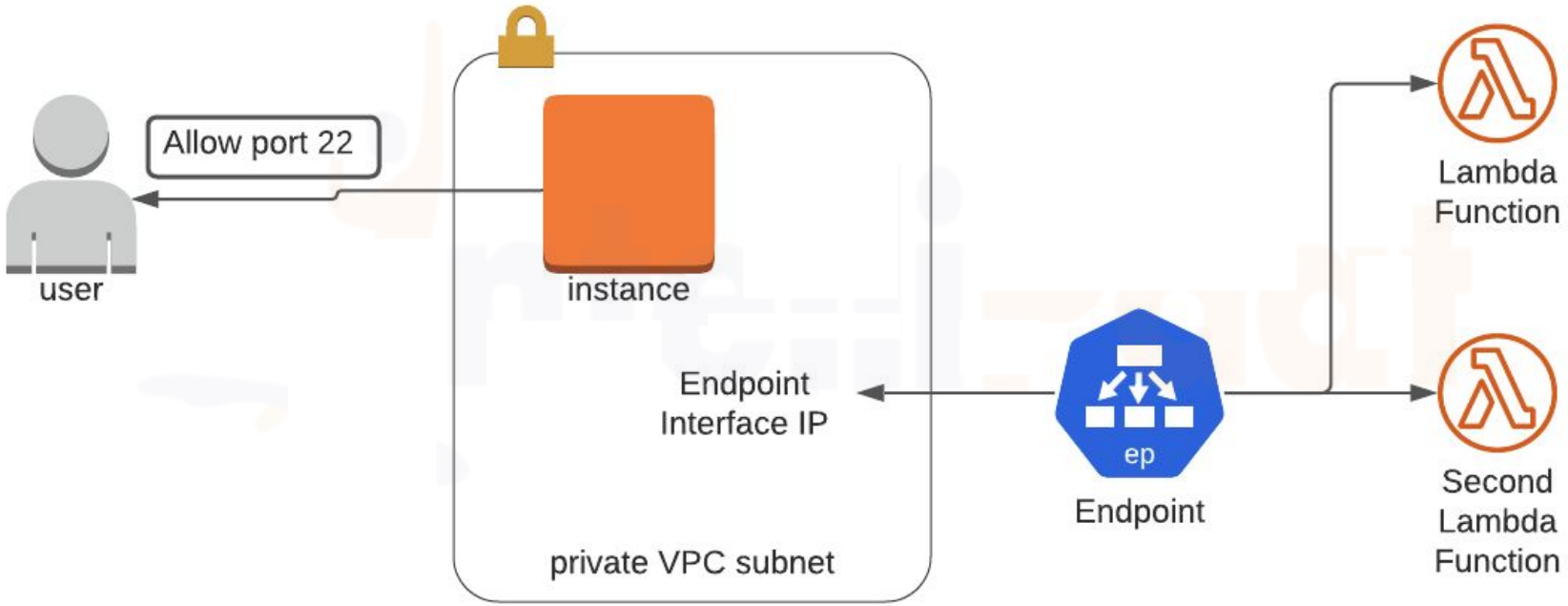
Instance Connect Endpoint

Instance Connect Endpoint is a secure way to access your AWS EC2 instances using a web browser, without needing to manage SSH keys or open inbound SSH ports and no public bastion hosts required.

You simply select the instance in the AWS Management Console and connect instantly. For example, you can quickly troubleshoot server issues.



VPC Endpoints Architecture Diagram



Hands-On

Create VPC Endpoint

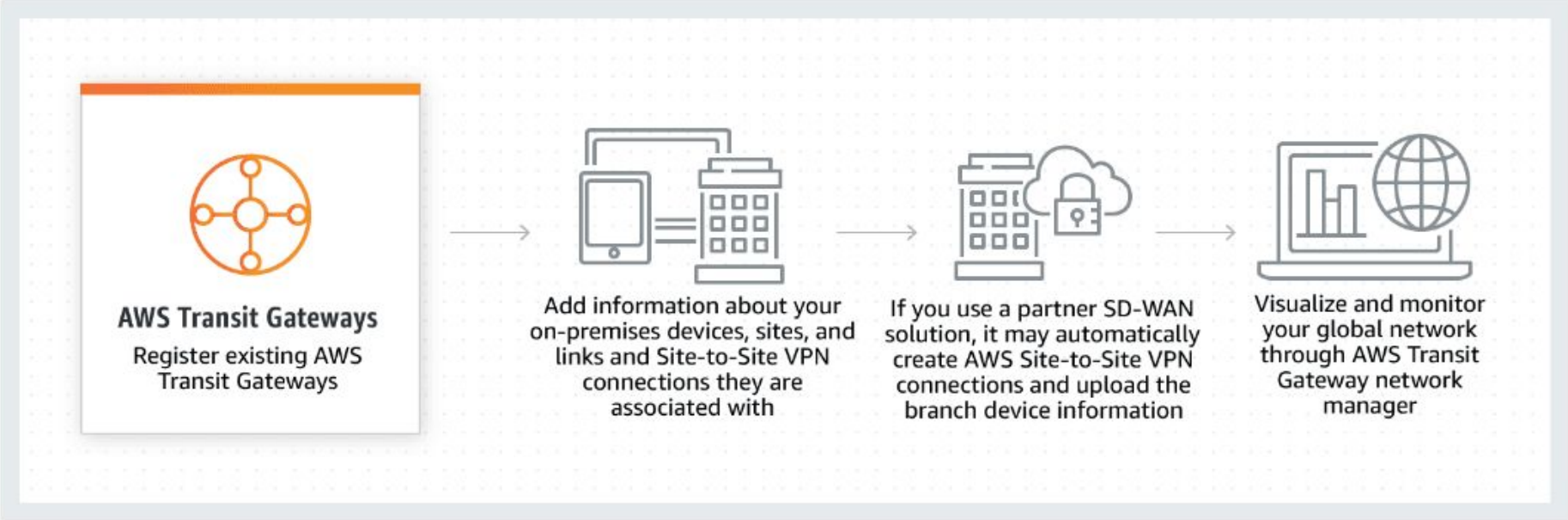
Hands-On: Create a VPC Endpoint

- Open the Amazon VPC console.
- Navigate to **Endpoints** and choose **Create endpoint**.
- Select **EC2 Instance Connect Endpoint** for the Service category.
- Select the default VPC for VPC and one subnet per **Availability Zone** for **Subnets**.
- Choose the default security group of the VPC for Security group.
- **Create the endpoint**.
- Create an EC2 instance in the **Private subnet** of the VPC.
- Click on **Connect** on the AWS Console.
- Choose the **EC2 Instance Connect** tab and select Connect using **EC2 Instance Connect Endpoint**.

Transit Gateways

What are Transit Gateways?

Transit Gateways in AWS are **scalable** and **centrally managed gateways** that simplify network connectivity between multiple VPCs and on-premises networks. Transit Gateways streamline **network management**, **reduce administrative overhead**, and **improve overall** network performance and **security** in complex AWS environments.



Prerequisites for Transit Gateway

A

You cannot have identical routes pointing to two different VPCs.

B

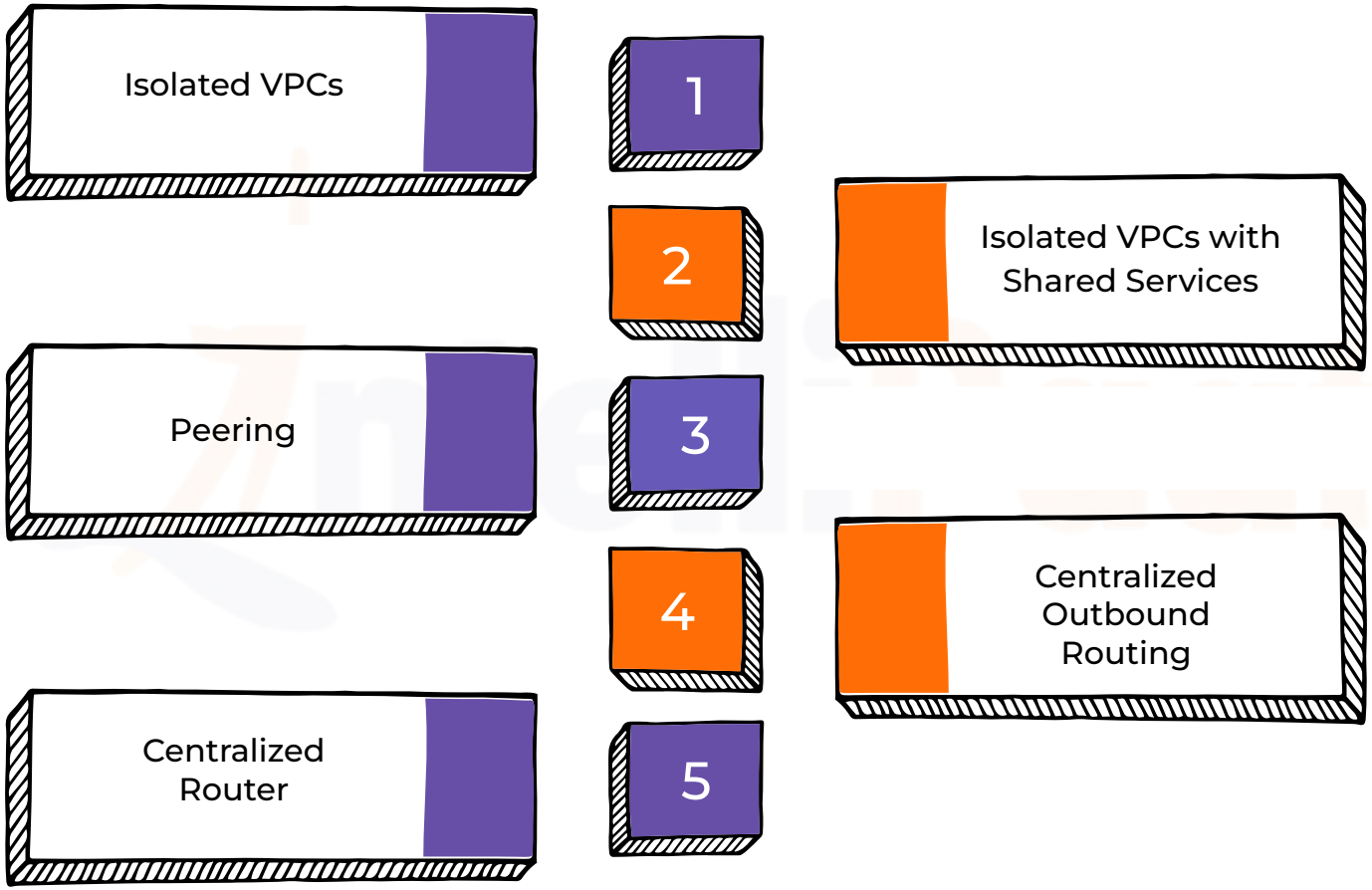
Verify that you have the permissions required to work with transit gateways.

C

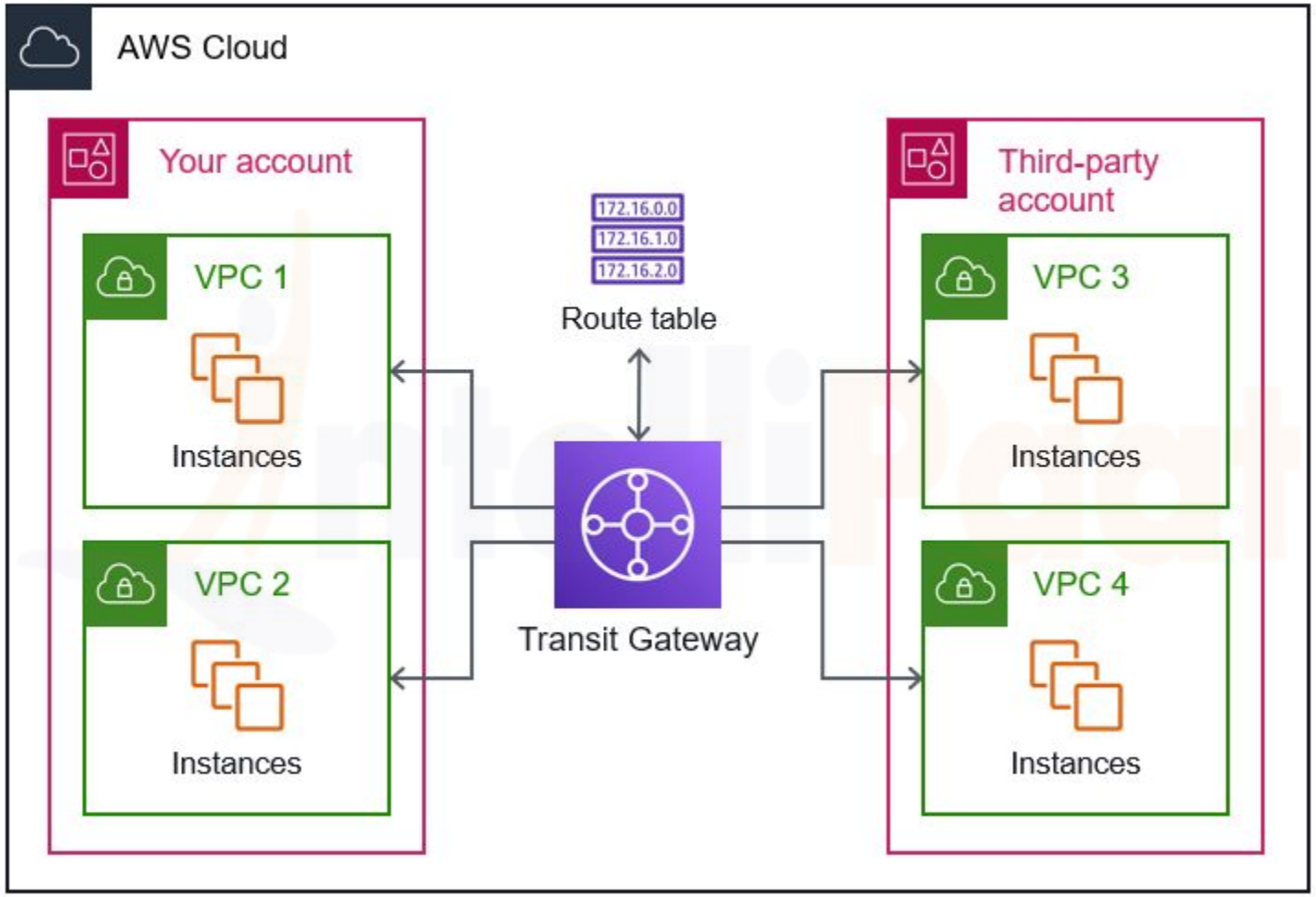
You can't ping between hosts if you haven't added an ICMP rule to each of the host security groups.

Transit Gateway Concepts

Use Cases for Transit Gateways



Transit Gateway Architecture Diagram



Transit Gateway vs. VPC Peering

	Transit Gateway	VPC Peering
Connectivity	Connects multiple VPCs, VPNs, and on-premises networks	Connects two VPCs directly, allowing communication
Scaling	Scalable solution supporting thousands of VPCs	Limited to connecting two VPCs at once
Transitive Routing	Supports transitive routing between connected networks	Does not support transitive routing
Use Cases	Suitable for large-scale network architectures,	Ideal for connecting specific VPCs with each other

Hands-On

ansit Gateway Peering

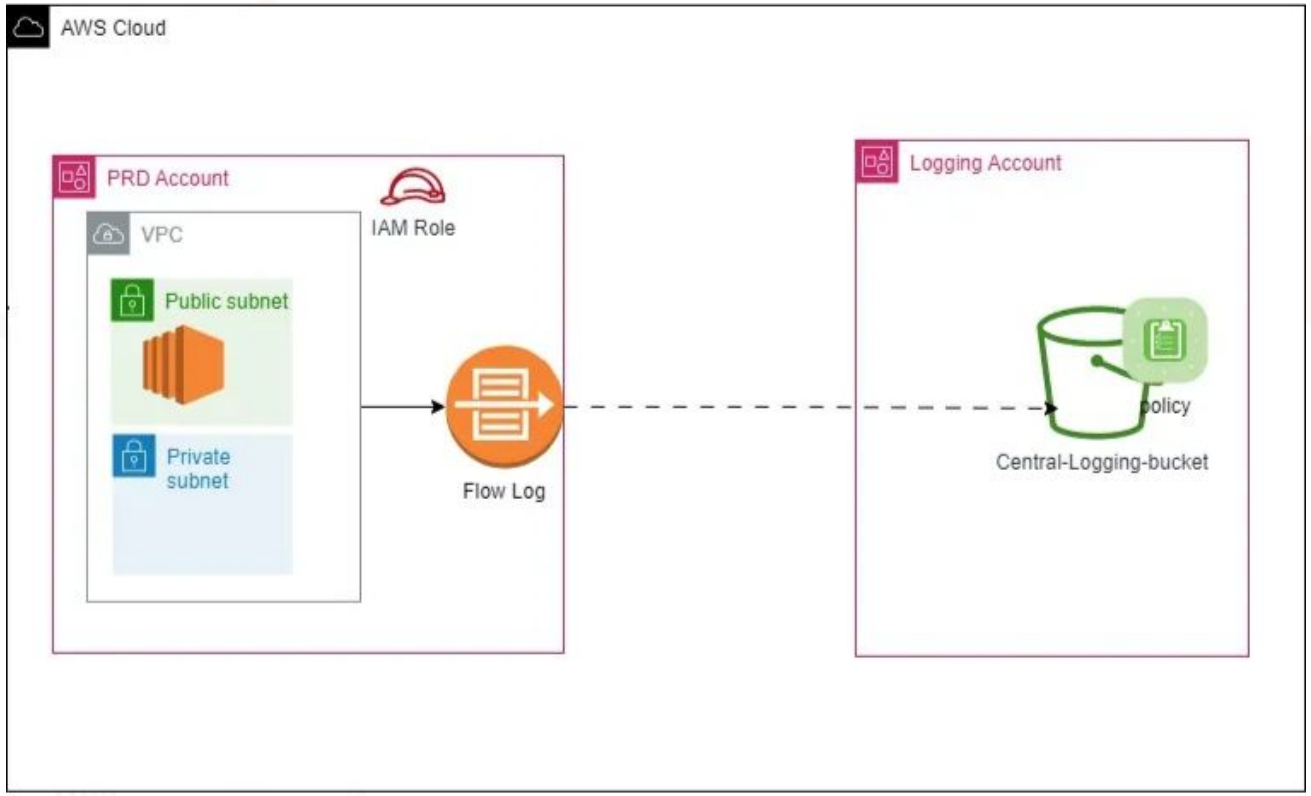
Hands-On: Transit Gateway Peering

- Open the Amazon VPC console.
- Navigate to Transit Gateway Attachments and choose Create transit gateway.
- Optionally enter a name and description for the transit gateway.
- Leave the Amazon side Autonomous System Number (ASN) as default.
- Specify one or more IPv4 or IPv6 CIDR blocks for the transit gateway.
- Create the transit gateway.
- Choose Create transit gateway attachment.
- Select the transit gateway for the attachment and choose VPC for Attachment type.
- Choose the VPC to attach to the transit gateway and create the transit gateway attachment.

VPC Flow Logs and Reachability Analyzer

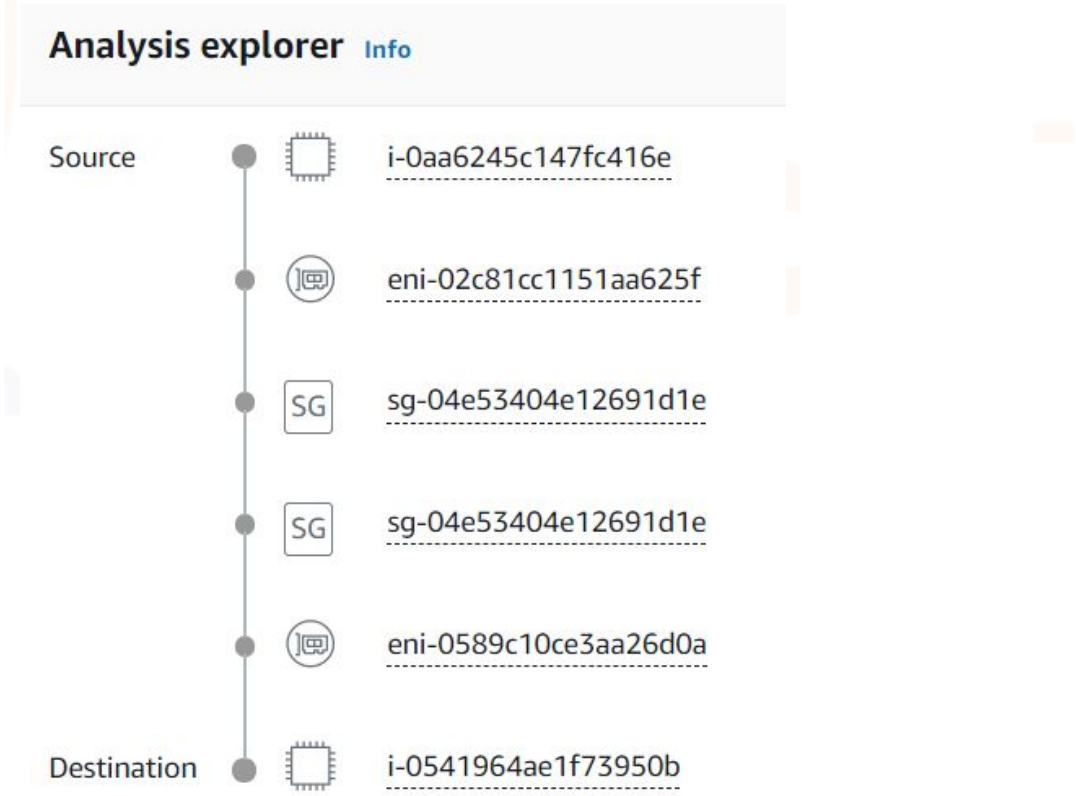
What are VPC Flow Logs?

VPC Flow Logs capture information about the **IP traffic flowing into** and **out of network** interfaces in a VPC. They provide insights into **traffic patterns**, help **troubleshoot connectivity issues**, and **enhance security** by monitoring and analyzing network traffic at the packet level.



VPC Reachability Analyzer

VPC Reachability Analyzer is a **configuration analysis tool** that allows you to perform **connectivity testing** in your virtual private clouds between a source resource and a destination resource (VPCs). Reachability Analyzer generates **hop-by-hop** details of the virtual network path between the **source** and the **destination** when the destination is reachable.



Use Cases of VPC Reachability Analyzer