# AWS VPC AND CLI

Amazon VPC is the networking layer for Amazon EC2 where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address ranges, creation of subnets, and configuration of route tables and network gateways.

A **VIRTUAL PRIVATE CLOUD (VPC)** is a virtual network dedicated to your AWS account.
Amazon VPC supports five (5) IP address ranges, one (1) primary and four (4) secondary for IPv4. Each of these ranges can be between /28 (in CIDR notation) and /16 in size.

A **SUBNET**: is a range of IP addresses in your VPC. By default, we can create 200 subnets per VPC.
Amazon reserves the first four (4) IP addresses and the last one (1) IP address of every subnet for IP networking purposes.

A **ROUTE TABLE**: contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed.
The custom route table has a route to the internet (0.0.0.0/0) through the internet gateway.

An **INTERNET GATEWAY**: is a VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic. To enable access to or from the internet for instances in a VPC subnet, you must do the following:

- Attach an internet gateway to your VPC.
- Add a route to your subnet's route table that directs internet-bound traffic to the internet gateway. If a subnet is associated with a route table that has a route to an internet gateway, it's known as a **public subnet**.
- If a subnet is associated with a route table that does not have a route to an internet gateway, it's known as a **private subnet**.
- Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
- Ensure that your network access control lists and security group rules allow the relevant traffic to flow to and from your instance.

**NETWORK ADDRESS TRANSLATION (NAT) GATEWAY**: to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
NAT gateway is created in a multiple Availability Zone to create an Availability Zone-independent architecture.
Process: The main route table sends internet traffic from the instances in the private subnet to the NAT gateway. The NAT gateway sends the traffic to the internet gateway using the NAT gateway's Elastic IP address as the source IP address.
To avoid data processing charges for NAT gateways when accessing Amazon S3 and DynamoDB that are in the same Region, set up a gateway endpoint and route the traffic through the gateway endpoint instead of the NAT gateway. There are no charges for using a gateway endpoint.

The main route table is associated with the private subnet and sends the traffic from the instances in the private subnet to the NAT instance in the public subnet. The NAT instance sends the traffic to the Internet gateway for the VPC. The traffic is attributed to the Elastic IP address of the NAT instance. The NAT instance specifies a high port number for the response; if a response comes back, the NAT instance sends it to an instance in the private subnet based on the port number for the response.

A **VPC PEERING CONNECTION**: is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account or different region VPC. You cannot have more than one VPC peering connection between the same two VPCs at the same time.
Terms: VPC (Requester) VPC (Accepter) Account ID.

A **VPC ENDPOINT**: enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by Private-Link without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

A **NETWORK ACCESS CONTROL LIST (ACL):** is an optional layer of security that's acts as a firewall for controlling traffic in and out of one or more subnets.
Security groups in a VPC specify which traffic is allowed to or from an Amazon EC2 instance. Network ACLs operate at the subnet level and evaluate traffic entering and exiting a subnet. Network ACLs can be used to set both Allow and Deny rules.
NACL rule starts from 100 - 32766 means ALLOW. If * means Deny. Deny is higher priority.

**SECURITY GROUPS VS NETWORK ACLS**
**Security groups control inbound and outbound traffic for your instances, and network ACLs control inbound and outbound traffic for your subnets. In most cases, security groups can meet your needs; however, you can also use network ACLs if you want an additional layer of security for your VPC.**

**CIDR -Classless Inter-Domain Routing**
Subnet -> route Table (public main rt) -> Routes -> Subnet Association
10.0.0.0/24, the following five IP addresses are reserved:

- 10.0.0.0: Network address.
- 10.0.0.1: Reserved by AWS for the VPC router.
- 10.0.0.2: Reserved by AWS. The IP address of the DNS server
- 10.0.0.3: Reserved by AWS for future use.
- 10.0.0.255: Network broadcast address.

172.22.130.0/28
32-28 = 4bits So at most 2^4 = 16 IP address. Of which 5 AWS Reserved and 11 we can use.
http://www.vlsm-calc.net/ipclasses.php

Nat Gateway- network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances. *NAT gateway, you must specify the public subnet in which the NAT gateway should reside to create an Availability Zone-independent architecture, create a NAT gateway in each Availability Zone and configure your routing to ensure that resources use the NAT gateway in the same Availability Zone.* **PEERING CONNECTION** to connect between different regions and accounts

## HOW TO CREATE A NETWORK IN AWS

1.Create a VPC
2.Create subnets
3.Create Route Table
4.Create Route: Main route-table for private subnet and Custom route-table for Public subnet
5.Create Internet gateways: Attach this to custom Route table
6.Generate Elastic IP
7.Create NAT gateways: Attach Nat Gateway to Main Route
8.Create Network ACLs

Testing the Internet Connection
https://docs.aws.amazon.com/vpc/latest/userguide/vpc-nat-gateway.html#nat-gateway-testing-example

1. **CREATE-VPC**
$ ***aws ec2 create-vpc --cidr-block***
Example: aws ec2 create-vpc --cidr-block 10.0.0.0/16
"VpcId": "vpc-01397012145c7f846"

2. **CREATE-SUBNET**
$ ***aws ec2 create-subnet --cidr-block --vpc-id***
Example: aws ec2 create-subnet --vpc-id vpc-a01106c2 --cidr-block 10.0.1.0/24
O/P: 1. Public SubnetId": "subnet-0ea9bd84889cd2b95
2.Private SubnetId": "subnet-0fa21e5f1983202f0

3. **CREATE-ROUTE-TABLE**
$ ***aws ec2 create-route-table --vpc-id***
Example: aws ec2 create-route-table --vpc-id vpc-a01106c2
O/P: Public: "RouteTableId": "rtb-03363f99cef54023e"
Private: "RouteTableId": "rtb-0a71494f6e192b30e"

4. **CREATE-ROUTE**
$ ***aws ec2 create-route --route-table-id --destination-cidr-block --gateway-id***
Example: aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-block 0.0.0.0/0 --gateway-id igw-c0a643a9

5. **CREATE-INTERNET-GATEWAY**
$ ***aws ec2 create-internet-gateway***
O/P: InternetGatewayId": "igw-0f7c11231b5ed30f0
$ *aws ec2 attach-internet-gateway --vpc-id "vpc-01397012145c7f846" --internet-gateway-id "igw-0f7c11231b5ed30f0" --region ap-south-1*

6. **ALLOCATE-ADDRESS**
$ ***aws ec2 allocate-address***
O/P: "PublicIp": "13.127.255.14", "AllocationId": "eipalloc-09784301836bb917d", "Domain": "vpc"

7. **CREATE-NAT-GATEWAY**
$ ***aws ec2 create-nat-gateway --allocation-id --subnet-id***

Example: aws ec2 create-nat-gateway --subnet-id subnet-1a2b3c4d --allocation-id eipalloc-37fc1a52
O/P: "NatGatewayId": "nat-0bb7075a9d003fbb0",
"SubnetId": "subnet-0ea9bd84889cd2b95",
"VpcId": "vpc-01397012145c7f846"

8. **CREATE-NETWORK-ACL**

$ ***aws ec2 create-network-acl --vpc-id***

Example: aws ec2 create-network-acl --vpc-id vpc-a01106c2

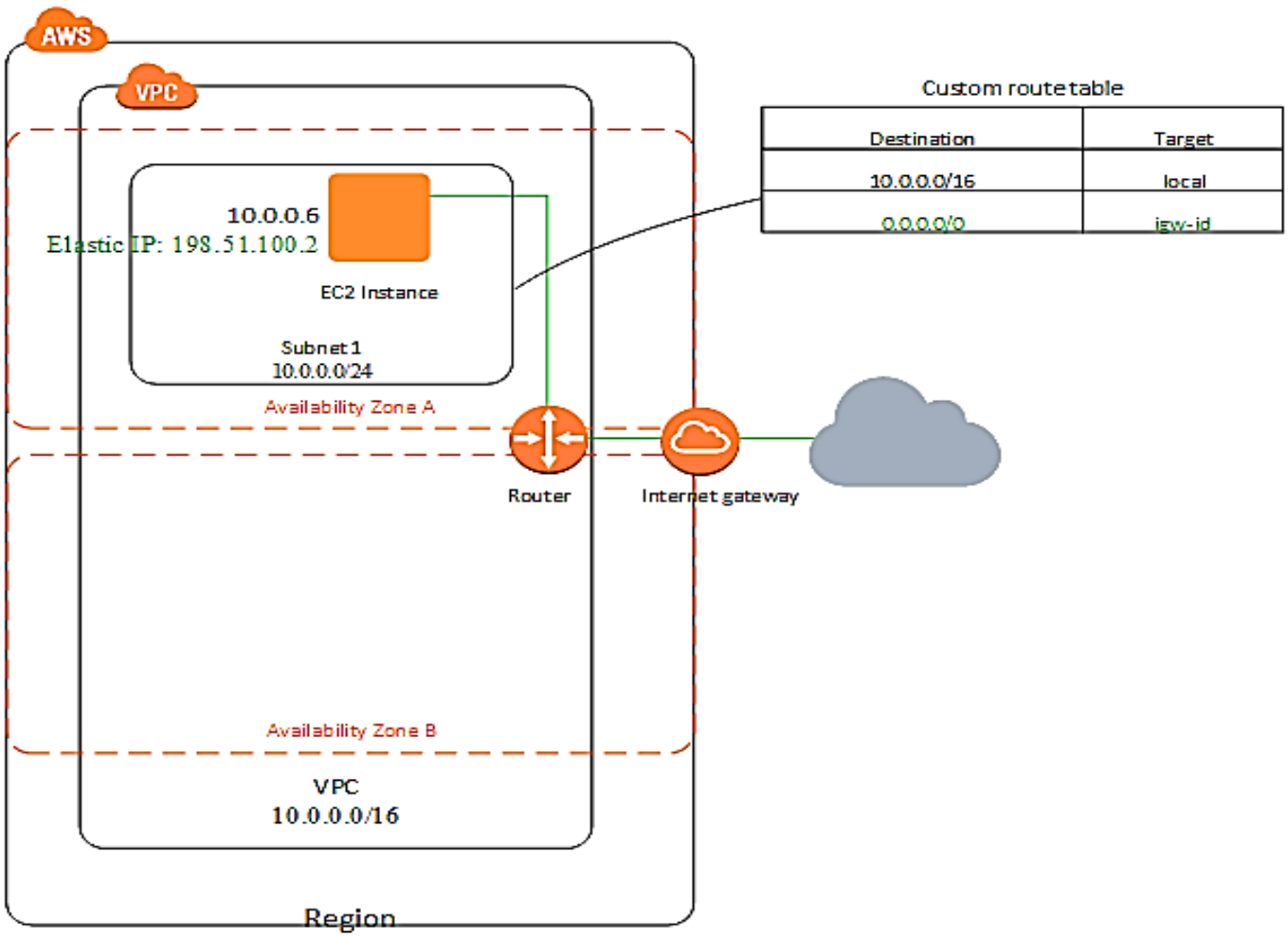
O/P: "NetworkAclId": "acl-0e7d7f2f50db1f608",
"VpcId": "vpc-01397012145c7f846"

$ *aws ec2 create-network-acl-entry --ingress --cidr-block 10.10.2.0/24 --network-acl-id acl-0e7 d7f2f50db1f608 --port-range From=80,To=80 --protocol all --rule-action allow --rule-number 200*

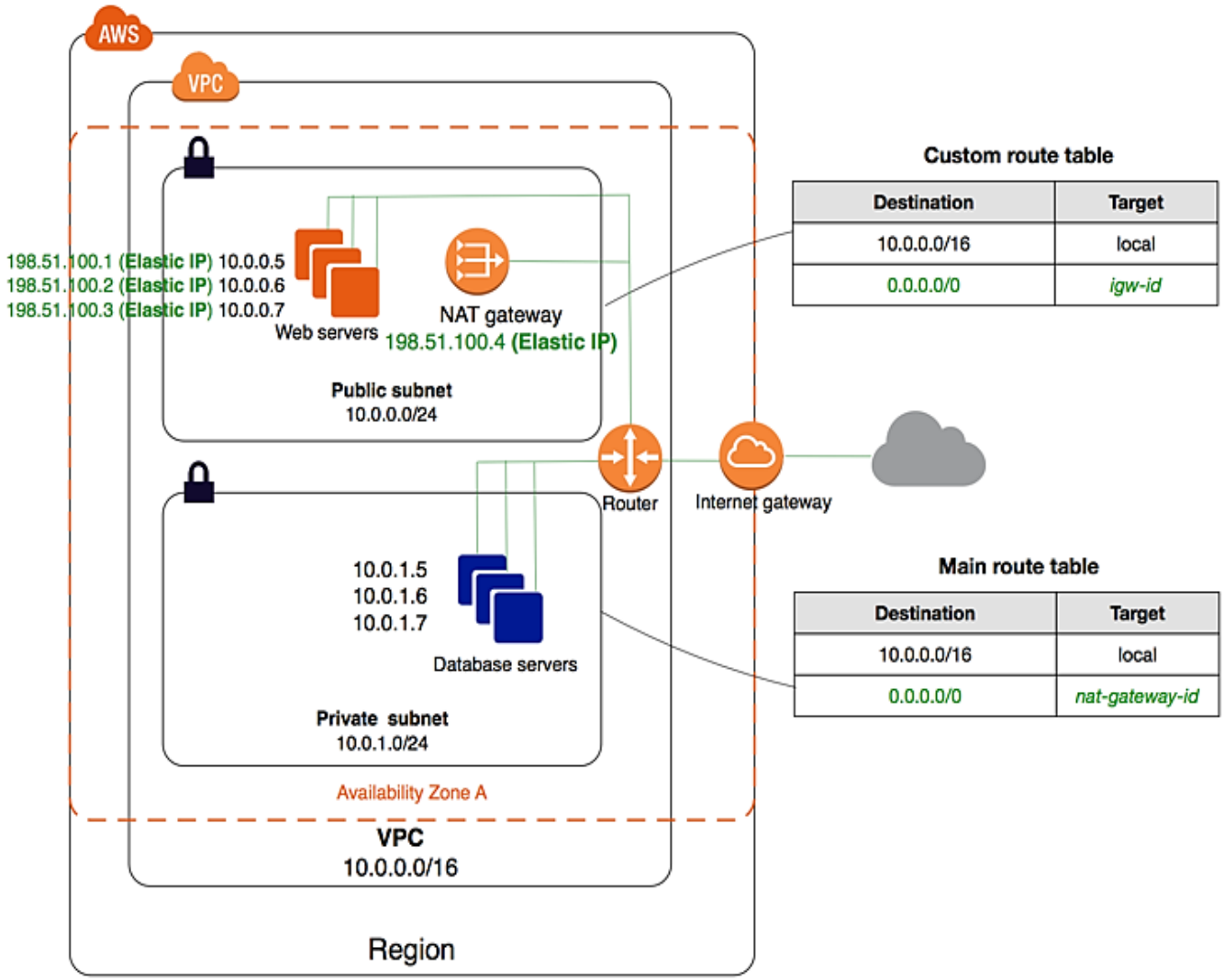
$ *aws ec2 create-network-acl-entry --ingress --cidr-block 192.168.0.0/16 --network-acl-id acl-0e05c00ed549b2870 --port-range From=22,To=22 --protocol all --rule-action allow --rule-number 210*

$ *aws ec2 create-network-acl-entry --egress --cidr-block 192.168.0.0/16 --network-acl-id acl-0e05c00ed549b2870 --port-range From=22,To=22 --protocol all --rule-action allow --rule-number 100*

**VPC with a single public subnet:**

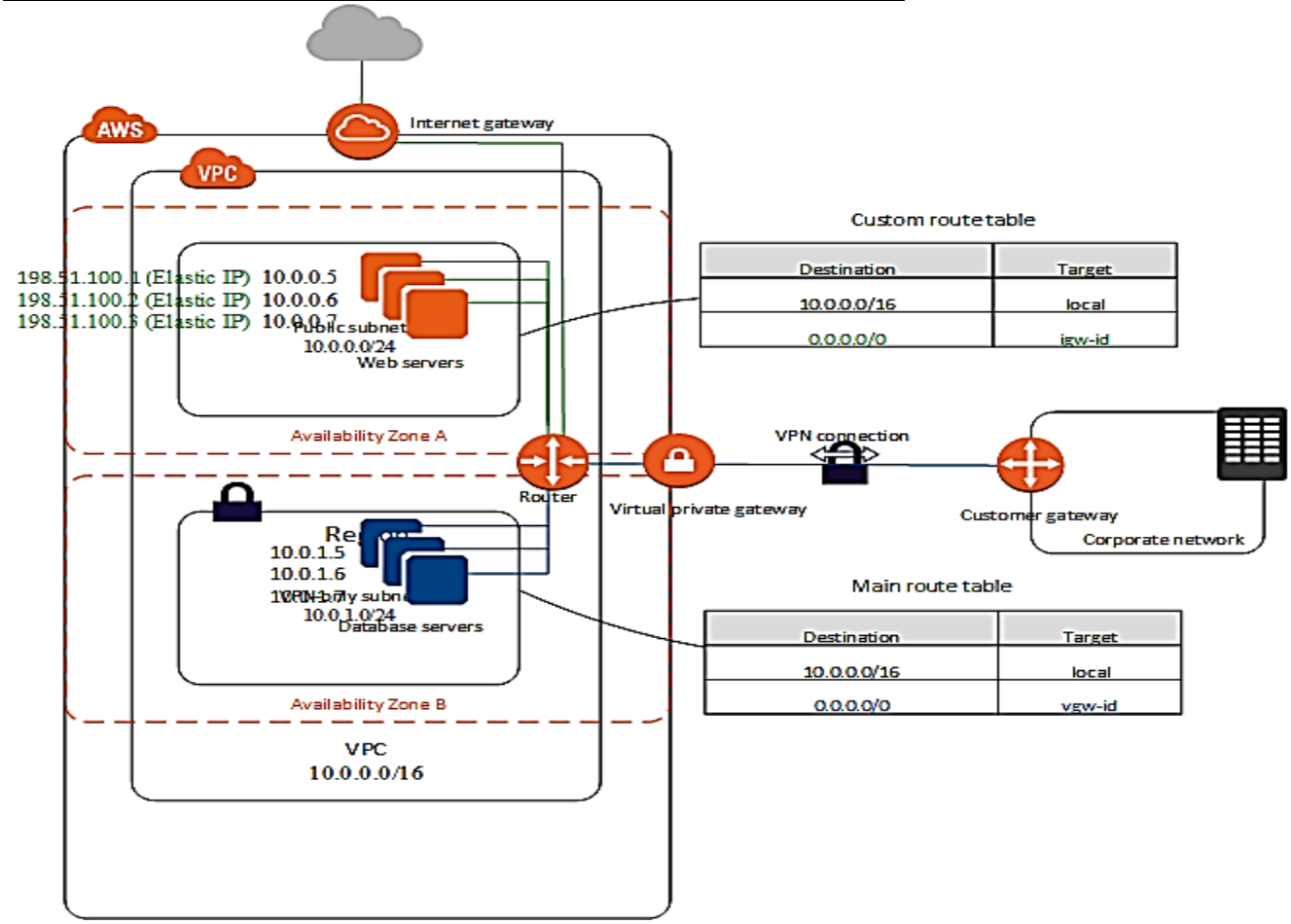


**VPC with public and private subnets (NAT):**

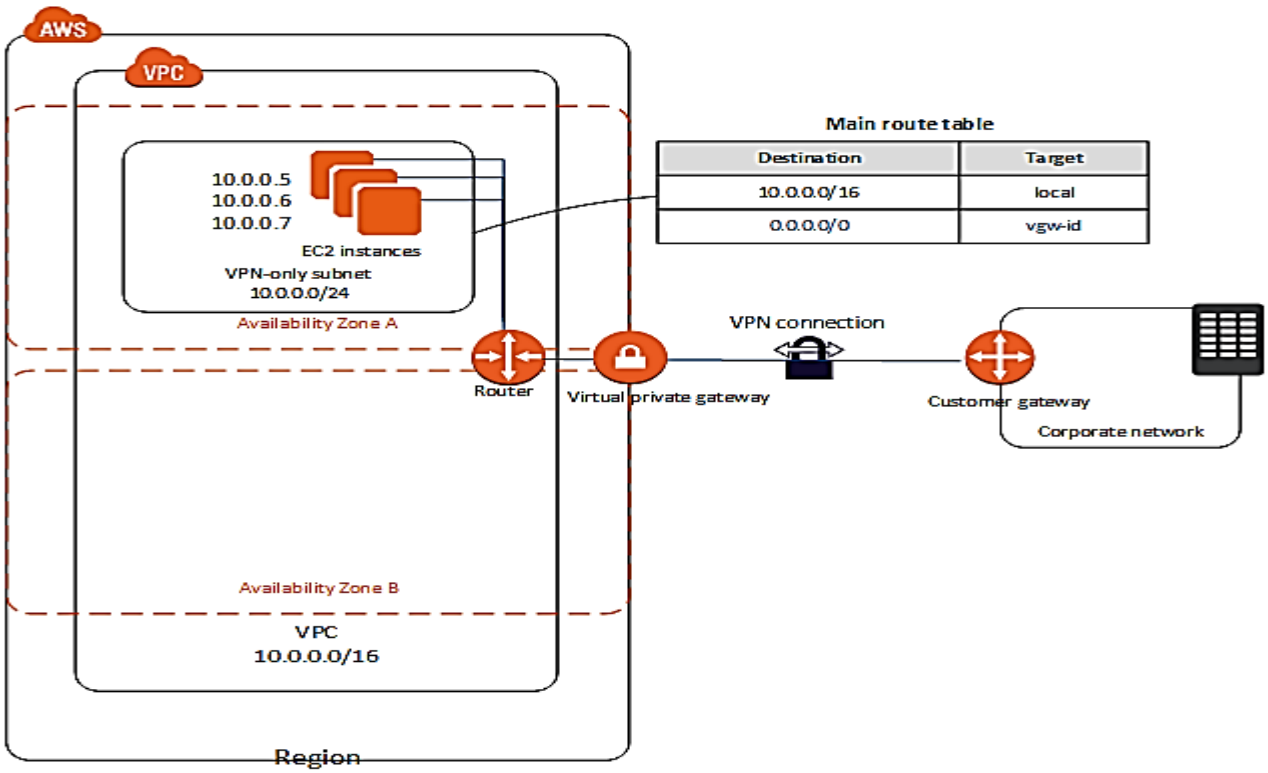


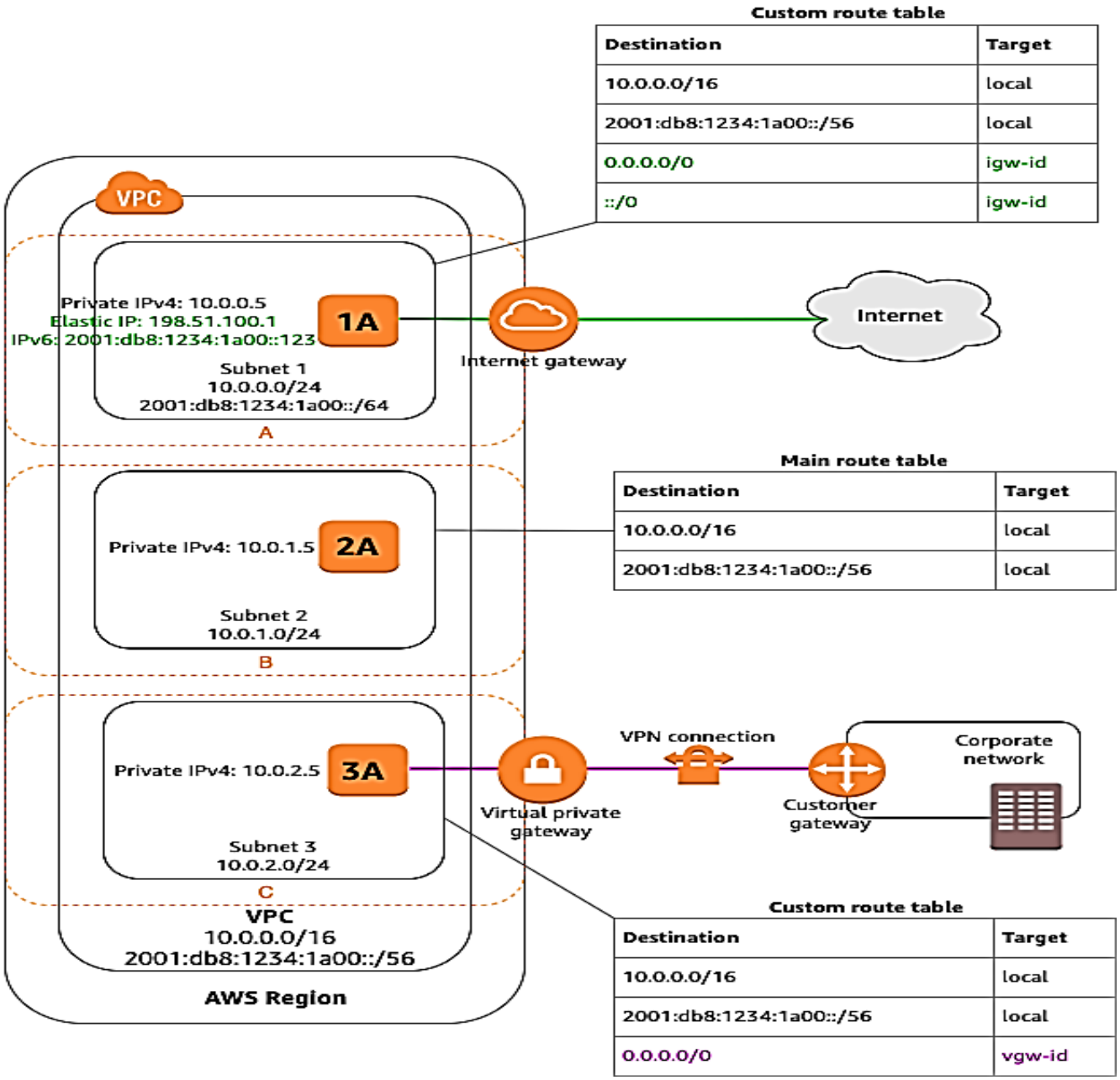

- A public subnet is a subnet that's associated with a route table that has a route to an Internet gateway-Internet gateway. This connects the VPC to the Internet and to other AWS services.
- Internet gateway. This connects the VPC to the Internet and to other AWS services.
- Public subnet with public IPv4 addresses that enable them to be reached from the Internet.
- Private subnet are back-end servers that don't need to accept incoming traffic from the Internet and therefore do not have public IP addresses; however, they can send requests to the Internet using the NAT gateway. NAT gateway with its own Elastic IPv4 address.
- Instances in the private subnet can send requests to the Internet through the NAT gateway
- A custom route table associated with the public subnet to communicate with other instances in the VPC and communicate with internet over IPv4.
- The main route table associated with the private subnet to communicate with other instances in VPC and communicate with the Internet through the NAT gateway over IPv4

## VPC with public and private subnets and AWS Site-to-Site VPN access:




## VPC with a private subnet only and AWS Site-to-Site VPN access:






- If a subnet's traffic is routed to an internet gateway, the subnet is known as a public subnet. In this diagram, subnet 1 is a public subnet. If you want your instance in a public subnet to communicate with the internet over IPv4, it must have a public IPv4 address or an Elastic IP address (IPv4).
- If a subnet doesn't have a route to the internet gateway, the subnet is known as a private subnet. In this diagram, subnet 2 is a private subnet.
- If a subnet doesn't have a route to the internet gateway, but has its traffic routed to a virtual private gateway for a Site-to-Site VPN connection, the subnet is known as a VPN-only subnet. In this diagram, subnet 3 is a VPN-only subnet.

You can connect your Amazon VPC to remote networks and users using the following VPN connectivity options.

| VPN connectivity option | Description |
| --- | --- |
| **AWS Site-to-Site VPN** | You can create an **IPsec VPN connection between your VPC and your remote network**. On the AWS side of the Site-to-Site VPN connection, a virtual private gateway or transit gateway provides two VPN endpoints (tunnels) for automatic failover. ***You configure your* customer gateway device *on the remote side of the Site-to-Site VPN connection***. For more information, see the AWS Site-to-Site VPN User Guide. |
| **AWS Client VPN** | **AWS Client VPN** is a managed client-based VPN service that enables you to securely access your AWS resources or your on-premises network. ***With AWS Client VPN, you configure an endpoint to which your users can connect to establish a secure TLS VPN session***. This enables clients to access resources in AWS or an on-premises from any location using an OpenVPN-based VPN client. For more information, see the AWS Client VPN Administrator Guide. |
| **AWS VPN Cloud Hub** | If you have more than one remote network (for example, multiple branch offices), ***you can create multiple AWS Site-to-Site VPN connections via your virtual private gateway to enable communication between these networks***. For more information, see Providing secure communication between sites using VPN Cloud-Hub in the *AWS Site-to-Site VPN User Guide*. |
| **Third party software VPN appliance** | You can create ***a VPN connection to your remote network by using an Amazon EC2 instance in your VPC that's running a third-party software VPN appliance***. AWS does not provide or maintain third party software VPN appliances; however, you can choose from a range of products provided by partners and open source communities. ***Find third party software VPN appliances on the AWS Marketplace*** |