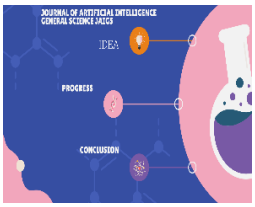
Vol.1,Issue1,January 2024
Journal of Artificial Intelligence General Science JAIGS

https://ojs.boulibrary.com/index.php/JAIGS



Design and Create VPC in AWS

Mr. Kondala Rao Patibandla

Senior Software Engineer, Aircraft Technical & Operations, Southwest Airlines, Irving, TX 75038

ARTICLEINFO
Article History:
Received:
05.01.2024
Accepted:
10.01.2024
Online: 22.01.2024

Keyword: Design VPC, VPC CIDR, Virtual Private Network, AWS, Subnets, NATs, ACLs, Security Groups

ABSTRACT

This article describes the design and creation of Amazon Virtual Private Network (VPC) using the VPC Designer tool and Cloud Formation templates. It also provides details of VPC Components such as Subnets, Route tables, Security Groups, Internet Gateway, NAT Gateway, VPC endpoints, Network Interfaces, Network Access Control Lists (ACLs), and VPC Peering. Amazon Virtual Private Cloud (VPC) enables you to create your own dedicated, logically isolated virtual private network in your AWS account. This virtual network closely resembles a traditional network that you operate in your own data center (on-premises). It provides the ability to define and have full control over the virtual network environment, including security, connectivity, and resource deployment. VPC spans multiple availability zones in an AWS Region.

I. Introduction

An Amazon VPC is primarily focused on the following capabilities: Isolating your own AWS resources from other AWS Accounts. Protecting your AWS resources deployed in VPC from network Threats. Routing network traffic to and from your resources deployed in VPC.

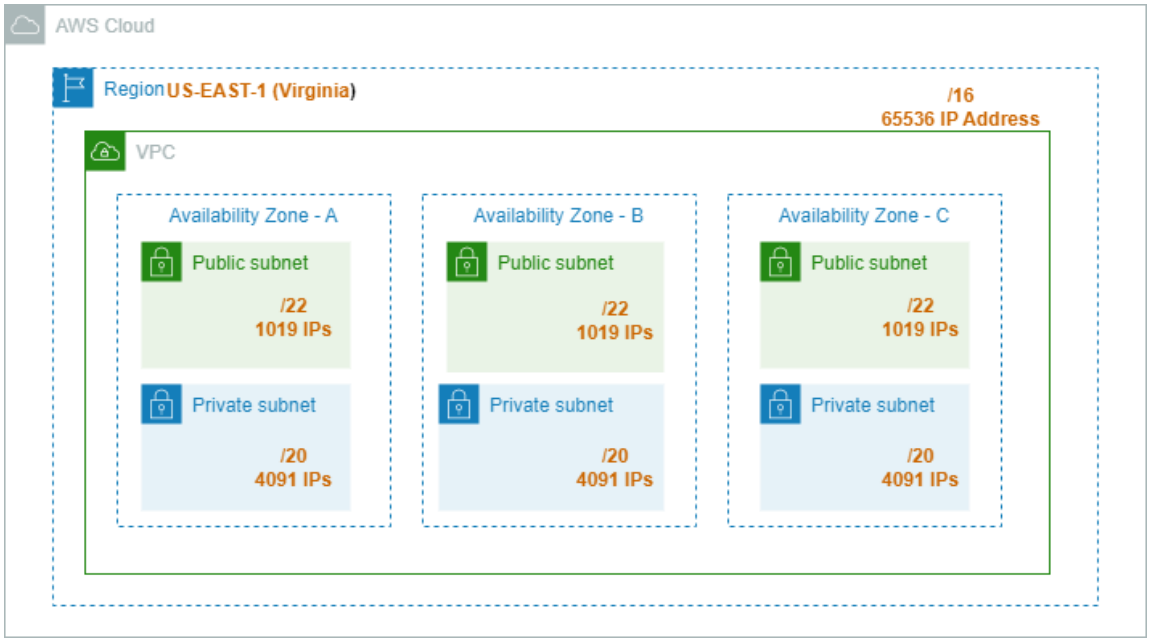


Illustration of a Virtual Private Cloud with Subnets.

II. VPC Components

The fundamental VPC components are:

- VPC CIDR Block
- Subnets
- Security Groups
- Network Access Control Lists (ACLs)
- Gateways (Internet, NAT)
- Route Table

VPC CIDR Block

VPCs and Subnets are associated with an IP address range that is a part of a Classless Inter-Domain Routing (CIDR) block, which will be used to allocate private IP addresses for the resources deployed in VPC. You can create a VPC by assigning either an IPv4 or IPv6 address.

Subnets

A subnet is associated with a CIDR block that is a subset of the '/16 CIDR block' of its VPC. A subnet must reside in a single availability zone. You can deploy AWS resources in a VPC after adding subnets to it.

Security Groups

A security group is considered the first-level defense and acts as a firewall for AWS resources deployed in a subnet to control incoming and outgoing traffic. Inbound rules control the incoming traffic to the instances, whereas outbound rules control the outbound traffic from the instances within the VPC.

Network Access Control Lists (ACLs)

An NACL is considered the second-level defense and acts as a firewall to the subnet that controls incoming and outgoing traffic. We can create rules to allow or deny network traffic to specific protocols through specific ports and specific IP address ranges. Network ACLs are stateless and have separate inbound and outbound rules.

Gateways

A Gateway connects a VPC to another network. The AWS resources residing in a VPC require connectivity outside of the AWS network, i.e., to the internet. A VPC must be attached to the Internet Gateway to access the Internet. NAT Gateway is used to enable the resources residing in a private subnet to connect to the Internet or other AWS services.

Route Table

A Route Table contains a set of rules, called routes, that determine where network traffic from a subnet or gateway needs to be directed. A Route Table is always associated with a subnet.

VPC Peering

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. We can create a VPC peering connection between our own VPCs or a VPC in another AWS account. Instances in either VPC can communicate with each other as if they are within the same network. The VPCs can be in different Regions i.e., also known as an inter-Region VPC peering connection.

III. Prerequisites

We have provided a sample solution along with this article. To deploy this solution, we must have the following:

- An AWS account.
- AWS CLI with administrator permissions.

IV. Design a VPC Using VPC Design Tool

Amazon VPC supports both IPv4 and IPv6 addresses. Here, we will use IPv4 to design a VPC. When we create a VPC, we must specify an IPv4 address for the VPC. The acceptable address block will be between a '/16 netmask' (65,536 IP address) and a '/28 netmask' (16 IP address). AWS recommends that we specify the CIDR block from the private address ranges specified in RFC 1918.

RFC 1918 RANGE	EXAMPLE CIDR BLOCK
10.0.0.0 - 10.255.255.255 (10/8 prefix)	10.0.0.0/16
172.16.0.0 - 172.31.255.255 (172.16/12 prefix)	172.31.0.0/16
192.168.0.0 - 192.168.255.255 (192.168/16 prefix)	192.168.0.0/20

RFC 1918 Standard

Let us discuss more IPv4 addresses.

An IPv4 address is 32 bits. An IP Address is shown as four decimal numbers representing 4 bytes (each byte is 8 bits, i.e. 8 * 4 = 32 bits).

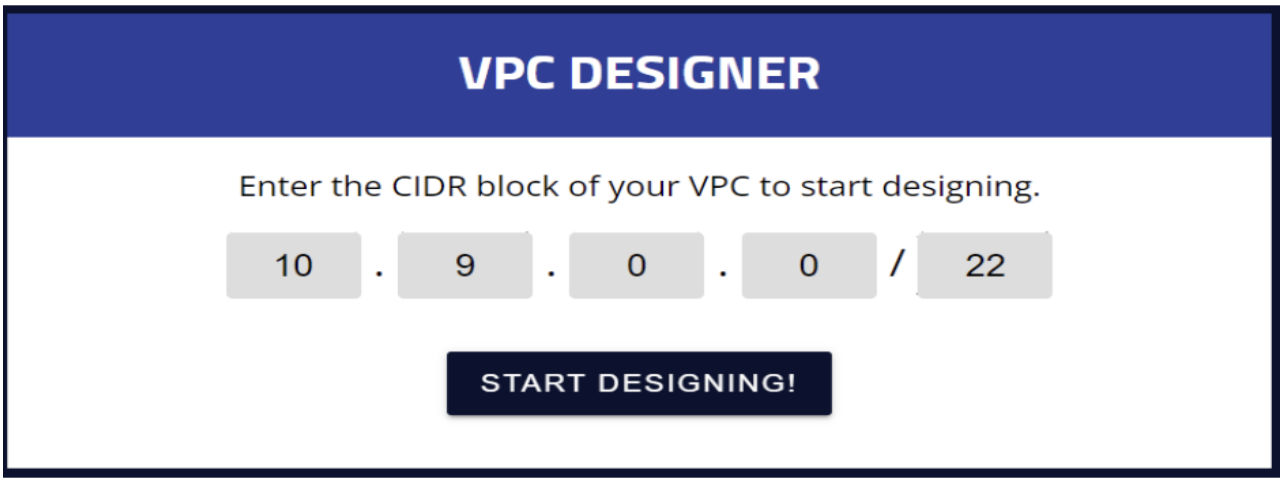
Example of an IP CIDR Range:

CIDR RANGE	EXPLANATION
10. 9 . 0 . 0 /16 **8 + 8**	CIDR / 16, first two digits will not change in this IP address range. **10 . 9 . (0 - 255) . (0 - 255) = 256 * 256 = 65536** private IP addresses are available for usage.
10. 9 . 0 . 0 /18 **8 + 8 +2**	CIDR / 18, first two digits will not change in this IP address, and the third digit allows a range between 0 through 63. **10 . 9 . (0 - 63) . 0 = 64 * 256 = 16384** private IP addresses are available for usage.
10. 9 . 0 . 0 /22 **8 +8 +6**	CIDR / 18, first two digits will not change in this IP address, and the third digit allows a range between 0 through 3. **10 . 9 . (0 - 3) . 0 = 4 * 256 = 1024** private IP addresses are available for usage.
10. 9 . 0 . 0/24 **8 + 8 + 8**	CIDR / 24, first three digits will not change in this IP address. **10 . 9 . 0 . 0 - 255 = 256** private IP addresses are available for usage.
10. 9 . 0. 0 /26 8+8+8+2	CIDR / 18, first three digits will not change in this IP address, and fourth digit will allow a range between 0 through 63. **10 . 9 . 0 . (0 - 63) = 64** private IP addresses are available for usage.
10. 9 . 0 . 0/32 **8 + 8 + 8+ 8**	CIDR / 32, all the four digits will not change in this IP address. **10 . 9 . 0 . 0 = 1** private IP address is available for usage.

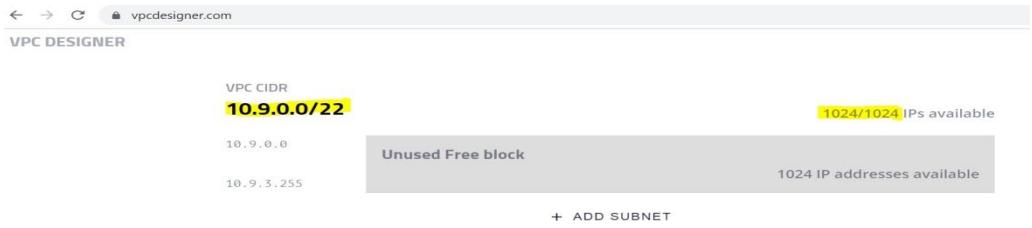
Let us now design a VPC and its subnets with an IP CIDR range. Please ensure the VPC CIDR range we use does not overlap with existing IP address ranges currently in use in your organization.

I am using a VPC designer tool to design subnets in a VPC.

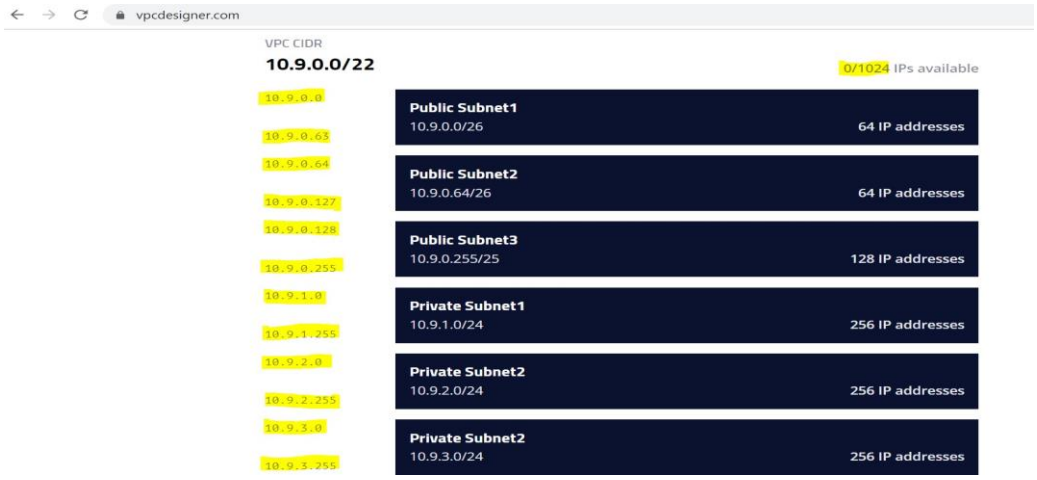
VPC CIDR Range: **10.9.0.0/22**



There are 1024 IP addresses available to allocate to our subnets.



Let us design two subnets in a VPC, one public and another private. This example takes three availability zones into consideration. We allocate 256 IP addresses for each private subnet using a 'CIDR /24' range. We then allocate 64 IP addresses for the first two public subnets using a 'CIDR /26' range. We then assign not 64, but 128 IP addresses to the last public subnet using a 'CIDR /25' range. This is possible as we would be left with 64 unused IP addresses, which we can assign to our third public subnet.
Refer to the screenshot below to understand how the allocation works.



V. VPC Creation

Let us now create a VPC using the information provided above from the VPC designer tool and design the CIDR range for our public and private subnets using the CloudFormation template.

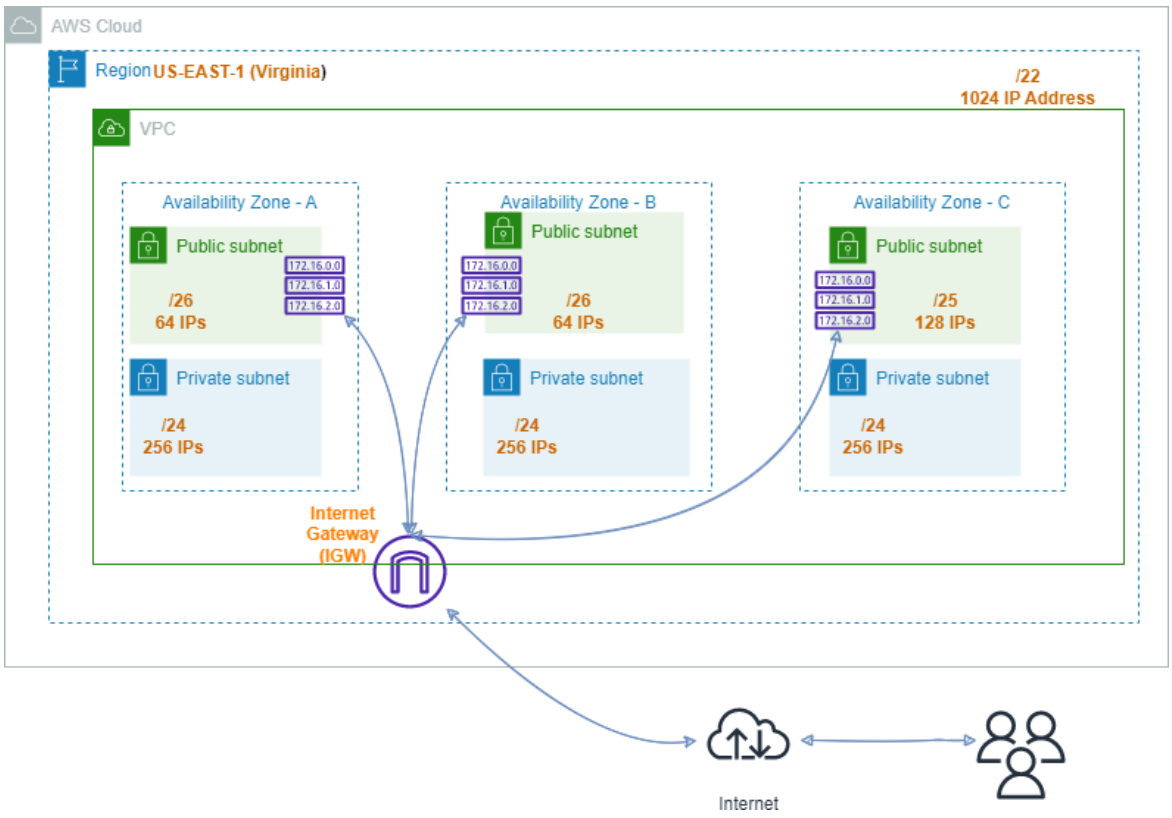
The above code snippet creates a VPC with public and private subnets, NACLs, Internet Gateway, NAT Gateway, and Route Tables.

All CIDR ranges of subnets in a VPC should be a subset of a VPC CIDR range. In this example, the VPC creates a '/22 CIDR block' (with 1024 IPs) with three availability zones, and each availability zone has a set of public and private subnets. All our public subnets will be created using the '/26 CIDR' block, and all our private subnets will be created using the '/24 CIDR' block.

This template will create a VPC with a public and private subnet, an internet gateway, and appropriate route table configurations to enable internet access for instances within the public subnet. For more detailed examples and customization options, you can visit the AWS CloudFormation VPC documentation.

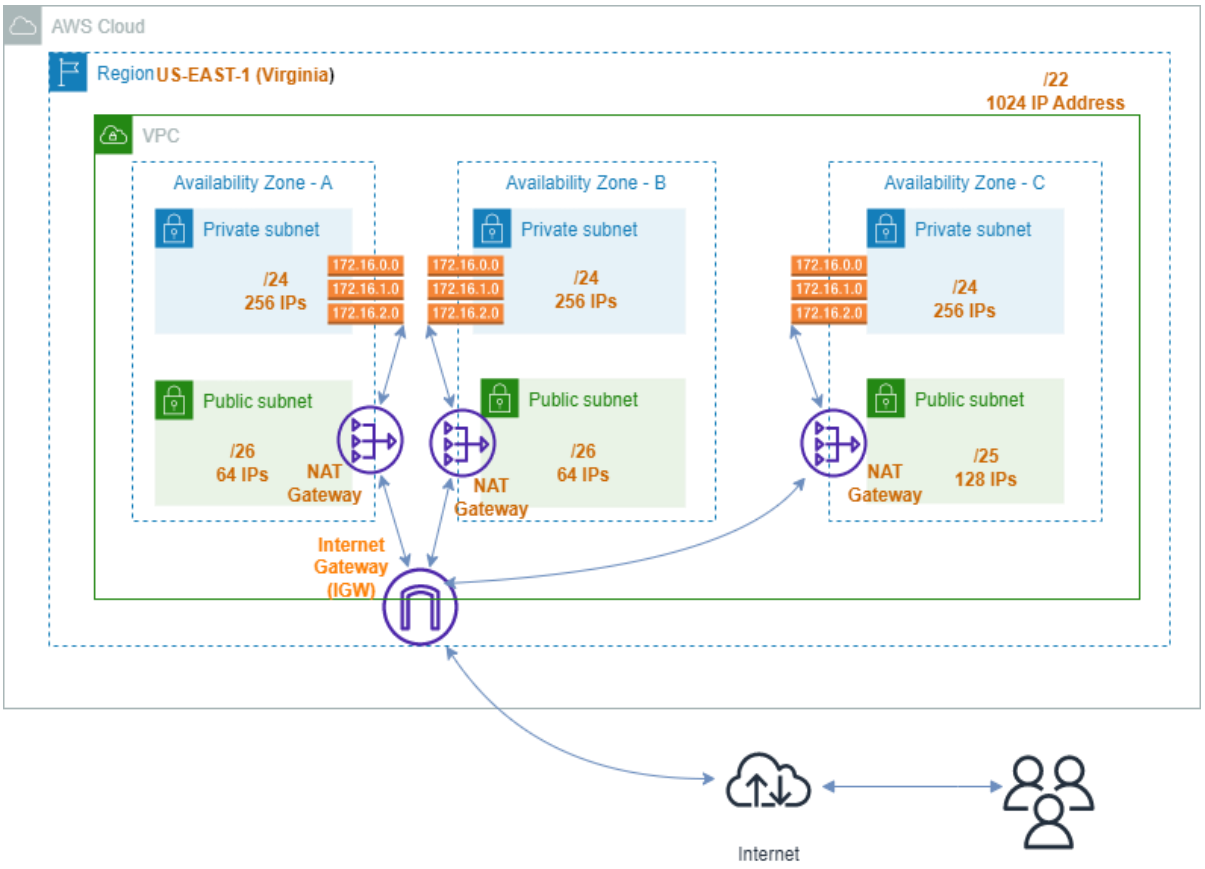
VI. Public Subnets

Each Public subnet has its own route table attached to route the internet traffic through the Internet gateway. The resources deployed in EC2 Instances in public subnets can have their own public IPs or Elastic IPs (EIPs) that have a NAT attached to their Elastic Network Interface (ENI).



VII. Private Subnets

AWS resources deployed in the private subnets are assigned only private IPs, and these resources cannot directly access the internet. Infrastructure in a private subnet gets access to resources or users on the Internet through a NAT Gateway. An AWS NAT gateway is a fully managed and highly available service (in a given availability zone). To provide high availability across the availability zones (AZs), it is recommended to have a minimum of two NAT gateways in different AZs. In the event that one AZ should become unavailable, the design choice of having two NAT gateways in two different AZs allows us to switch to an available NAT gateway.



Applications hosted on instances within a private subnet have diverse access requirements. Some applications may need to connect to the Internet to access third-party services, download updates, or communicate with external users. On the other hand, some services may require secure connections to databases, applications, and users that are hosted on-premises, ensuring seamless integration and secure data transfer within the organization's infrastructure. To meet these varied access requirements, AWS offers robust networking solutions, namely the Virtual Private Gateway and the Transit Gateway.

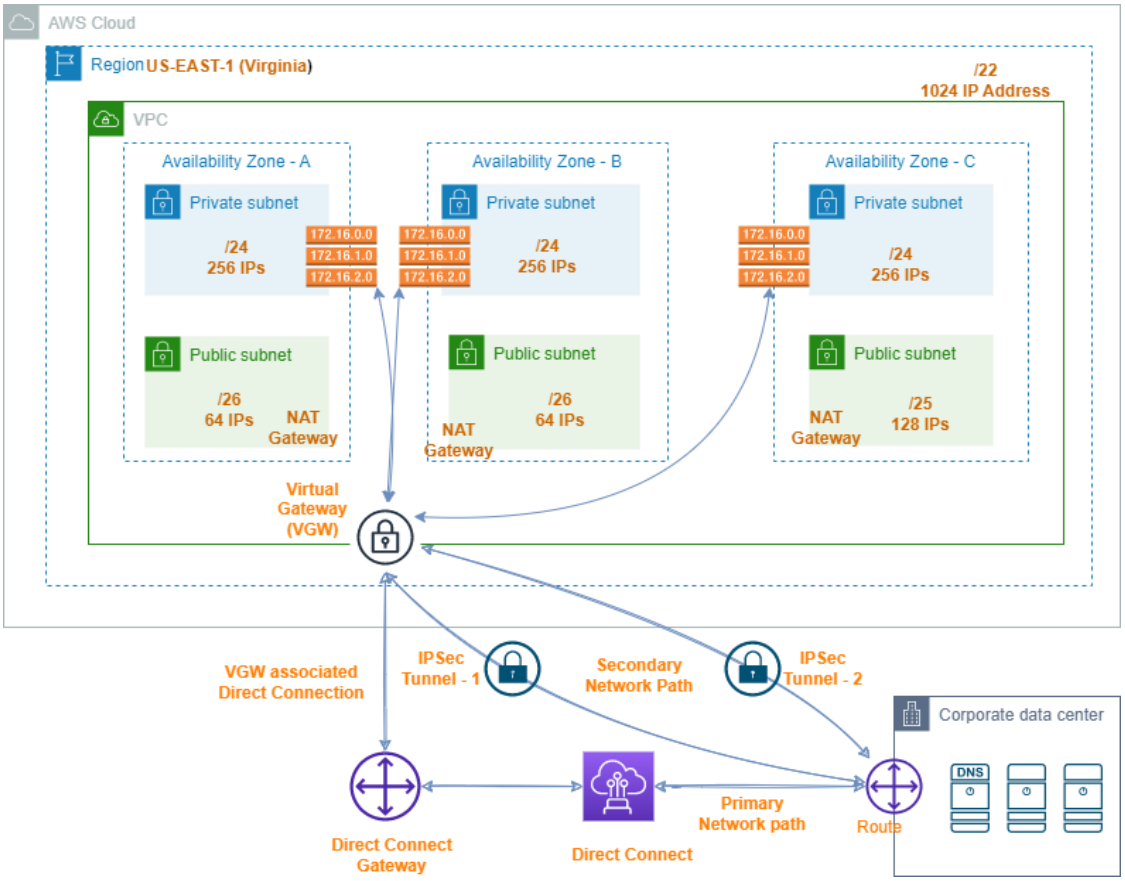
The **Virtual Private Gateway** is designed for simpler connectivity needs, where a single Virtual Private Cloud (VPC) requires secure access to on-premises resources. This service establishes a dedicated connection between the VPC and the on-premises network through a Virtual Private Network (VPN) connection or a Direct Connect link. The Virtual Private Gateway is suitable for organizations with straightforward networking requirements, where only one VPC needs to communicate with the on-premises environment. It ensures secure, encrypted data transfer and reliable connectivity for applications that need to interact with on-premises systems.

In contrast, the **Transit Gateway** is built to handle more complex networking scenarios, involving multiple VPCs and on-premises networks. The Transit Gateway acts as a central hub, simplifying the interconnectivity of tens or even hundreds of VPCs. It allows these VPCs to communicate with each other and aggregate their connectivity to resources residing on-premises. This solution is ideal for large-scale, enterprise-level deployments where multiple VPCs need to interact with each other and with on-premises systems efficiently. The Transit Gateway supports scalable, high-performance, and resilient networking, making it a preferred choice for complex network architectures.

By leveraging the Transit Gateway, organizations can achieve streamlined network management, reducing the complexity and operational overhead associated with managing individual VPN connections for each VPC. This centralized approach enhances network visibility and control, making it easier to implement security policies, monitor traffic, and troubleshoot issues. Additionally, the Transit Gateway supports advanced features like multicast, allowing efficient distribution of data to multiple destinations within the network.

In summary, while the Virtual Private Gateway is suitable for simpler, single-VPC connectivity scenarios, the Transit Gateway excels in managing complex, large-scale network environments with multiple VPCs and on-premises integration needs.

Organizations must carefully evaluate their networking requirements to choose the appropriate solution that aligns with their infrastructure goals and ensures efficient, secure, and reliable connectivity for their applications and services.



AWS provides two options for establishing private connectivity between our VPC and on-premises network: AWS Direct Connect and AWS Site-to-Site VPN. AWS Site-to-Site VPN configuration leverages IPSec, with each connection providing two redundant IPSec tunnels. AWS supports both static routing and dynamic routing (through the use of BGP).

Note: BGP is recommended, as it allows dynamic route advertisement, high availability through failure detection, and failover between tunnels, in addition to decreased management complexity.

VIII. Conclusion

In this article, we learned how to design a VPC and subnets and defined their components. We ensured our VPC has internet and private network connectivity to the resources deployed within our VPC by defining an Internet Gateway, NAT Gateway, Virtual Gateway (VGW), and Direct Connect. We learned to describe components such as Route Table and NACLs. I hope this article helps you to understand, design, and define a virtual private cloud in simple and easy steps.