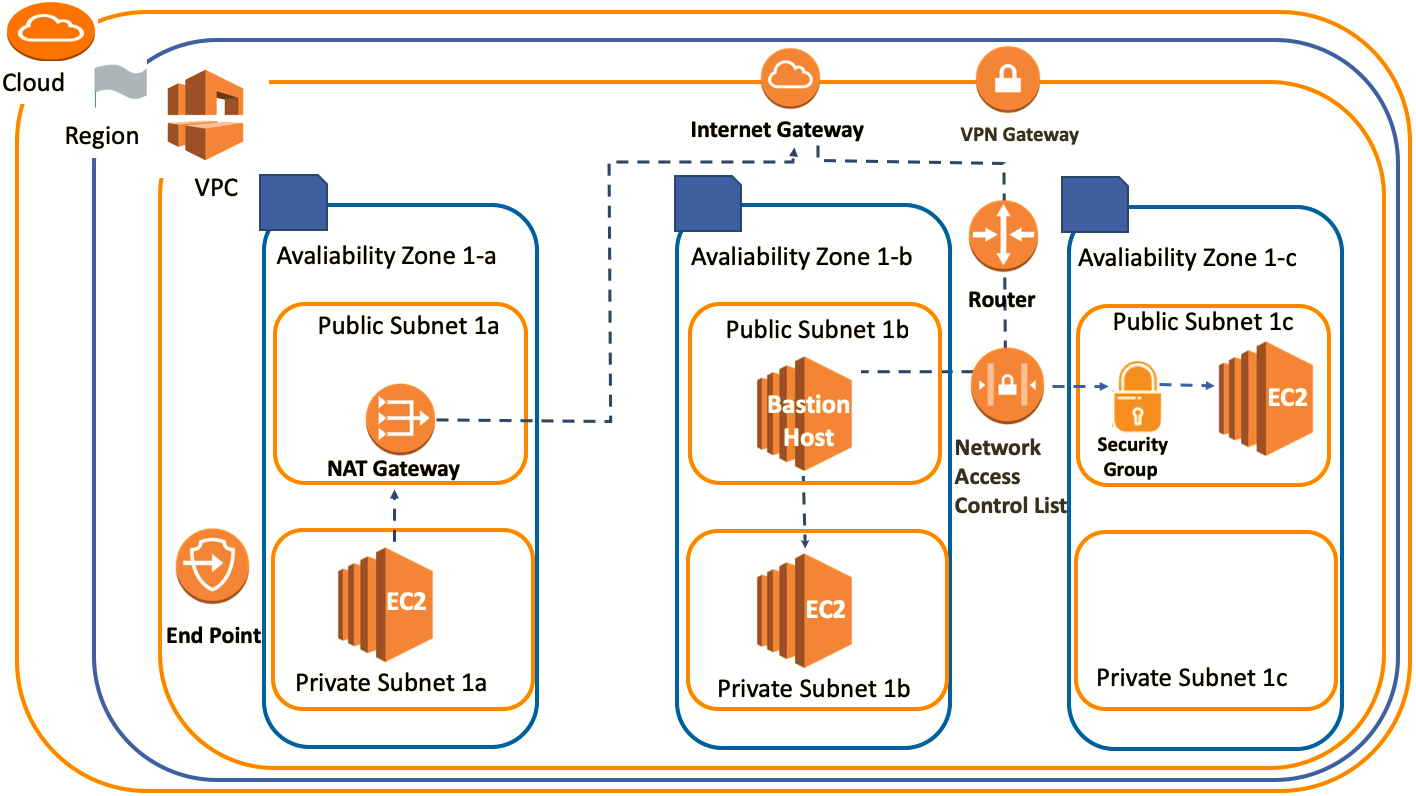
Introduction to VPC

We can provide the communication part of physical networks that we create through devices such as switches, routers, repeaters, firewalls, load balancers connected by network cables.

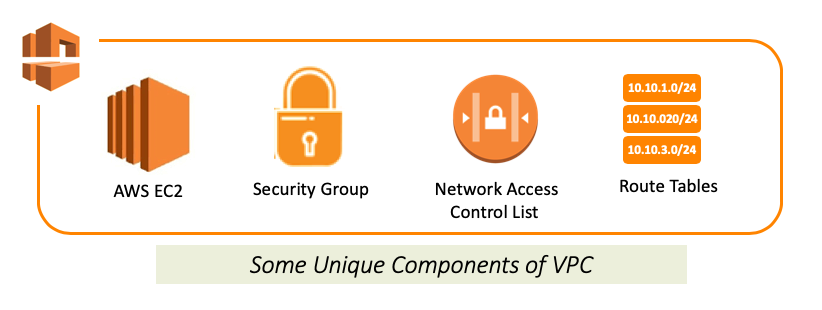
But when it comes to the cloud, there is no longer a physical network that we access and configure. There are virtual structures that simulate this physical network and allow us to build logical structures on it. In the AWS world, this is called VPC.

Briefly, Amazon Virtual Private Cloud (Amazon VPC) is a logically isolated area of the AWS cloud where you can launch AWS resources in a virtual network that you define. So, VPC provides much better security control over your AWS resources.

### **Architecture of VPC**



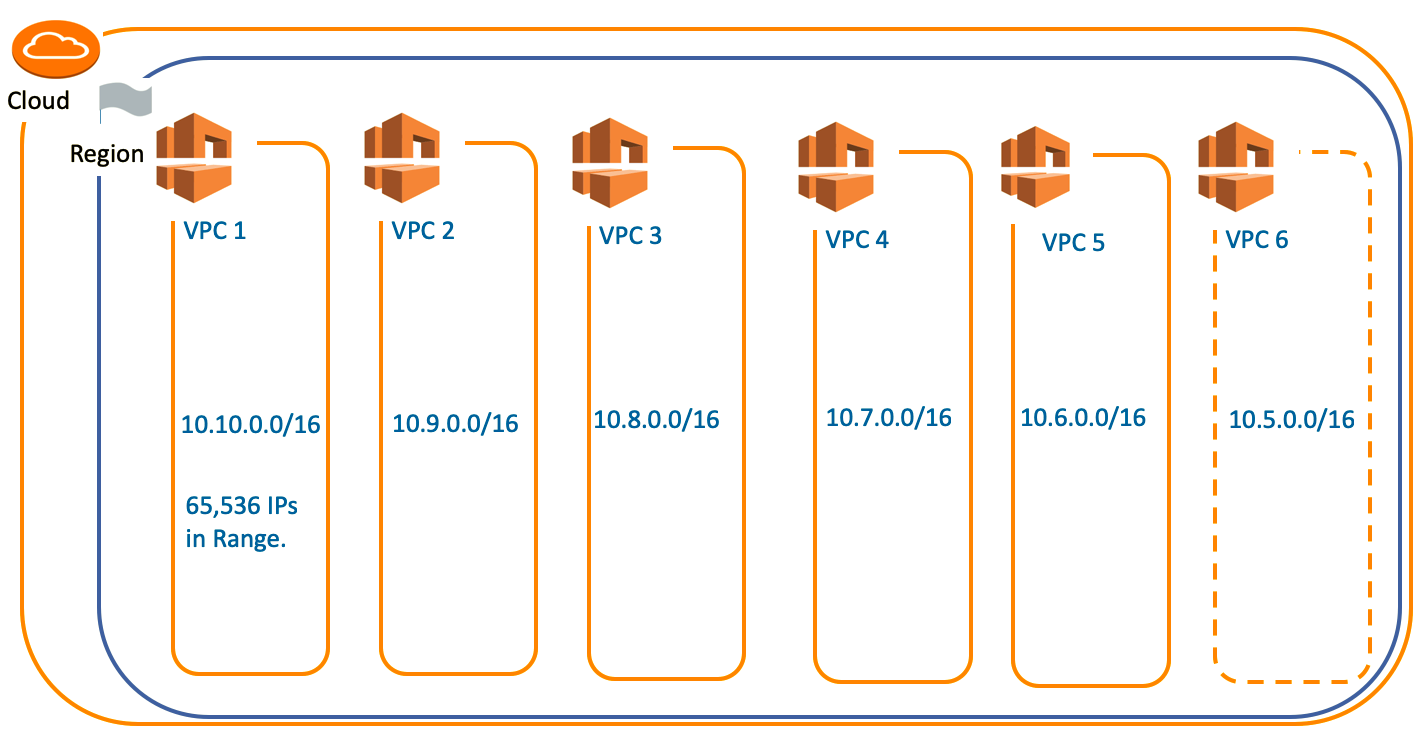
### **VPC Structure**



VPC is a layer of a virtual network where we can create our own private network structure, run our resources and logically isolated. Thus, VPC allows full control over your virtual network environment, including choosing your own IP address range, creating subnets, configuring route tables and gateways.

VPC is a unique area. So when you create or set something associated with VPC, it means these things can only work harmoniously in this environment.

### **VPC Region&AZ**



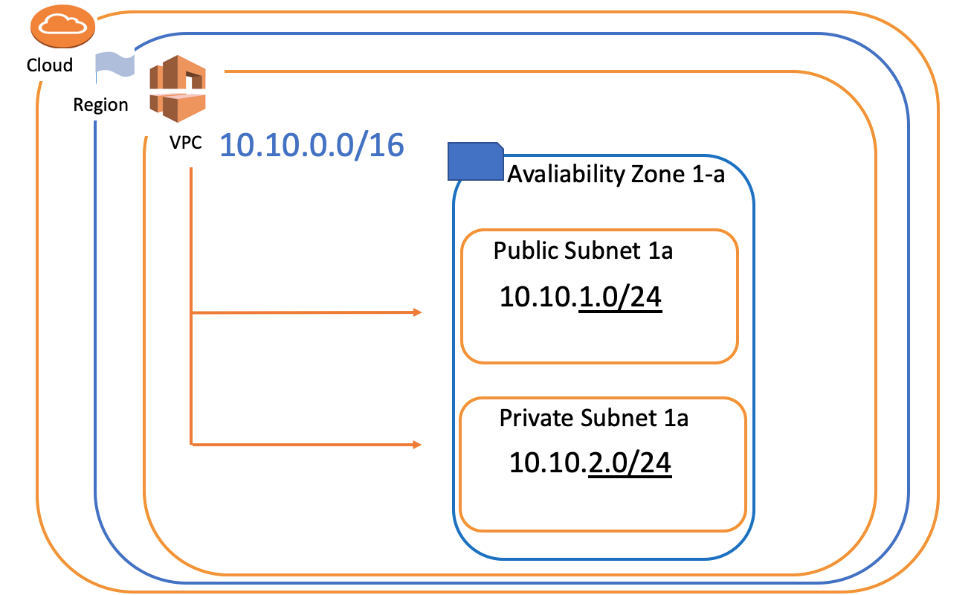
AWS has the concept of a Region, which is a physical location around the world where data centers are clustered. Each AWS Region consists of multiple, isolated, and physically separate AZ's within a geographic area as you remember in the **Getting Started with AWS** section.

Each region; it comes with one default VPC and we can also create 5 more VPCs for each region. This is a soft limit of 5 and if we need more than 5 VPCs, we can request it from AWS.

VPCs are associated with a single region. You cannot span a VPC across regions.

### **VPC Subnets**

A subnet is a range of IP addresses in your VPC.



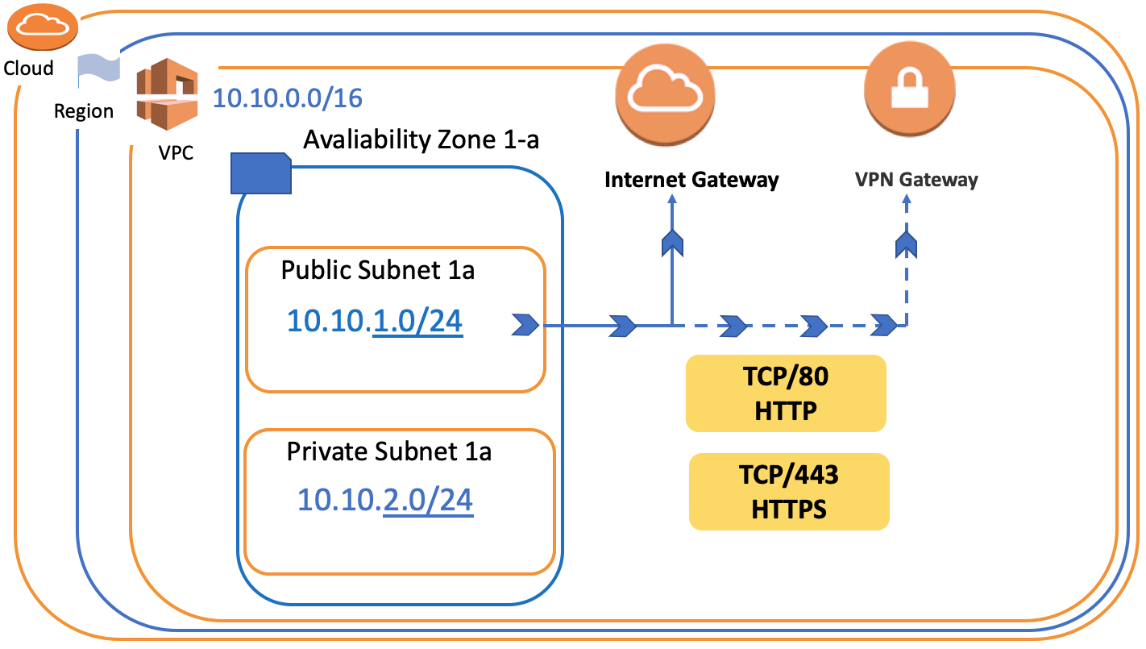
We divide the main IP block that we define in VPC into sub-logical networks called a subnet. In this way, we can easily cluster the components with the same properties.

Subnets can be 2 types: **Public Subnet** and **Private Subnet**.

* When we say Public Subnet, the virtual machines that we place in the subnets can be accessed from the outside of the VPC (Public internet).
* The machines we place in the subnets called Private have the opportunity to communicate only through VPC. In this way, we protect our resources against unauthorized access and ensure network security.

In addition, each subnet can be created only in a specific AZ (Availability Zone). A subnet cannot be associated with more than one AZ.

**Internet Gateway & VPN Gateway**

****

Let's say we created our VPC and subnets. But how do we get them to connect to the internet? The answer is Internet Gateway or VPN Gateway.

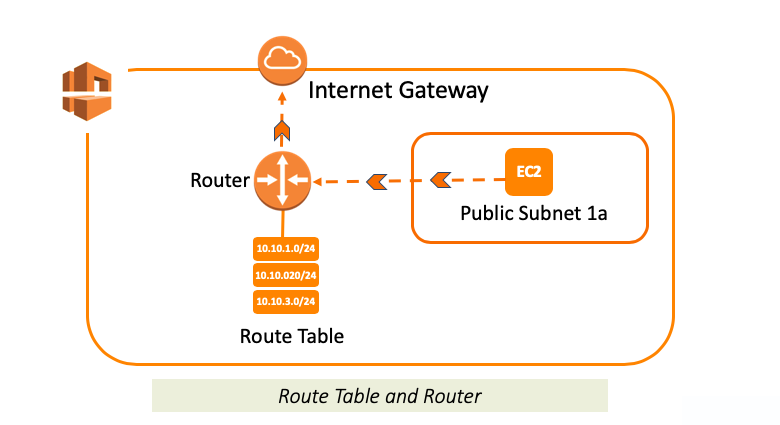
An **Internet Gateway** is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

We can think of internet gateways as basic internet providers with our ADSL or fiber routers that allow us to access the internet in our home.

Besides, this internet connection can be made as a direct connection between its own company network infrastructure and this VPC. This system is also called a **VPN Gateway**.

In short, VPC can connect to the outside world in two basic ways, either VPN Gateway or Internet Gateway.

**Route Table and Router**

****

**Route Table** is a set of rules, that is used to determine where network traffic is directed. It provides all traffic routing under VPC.

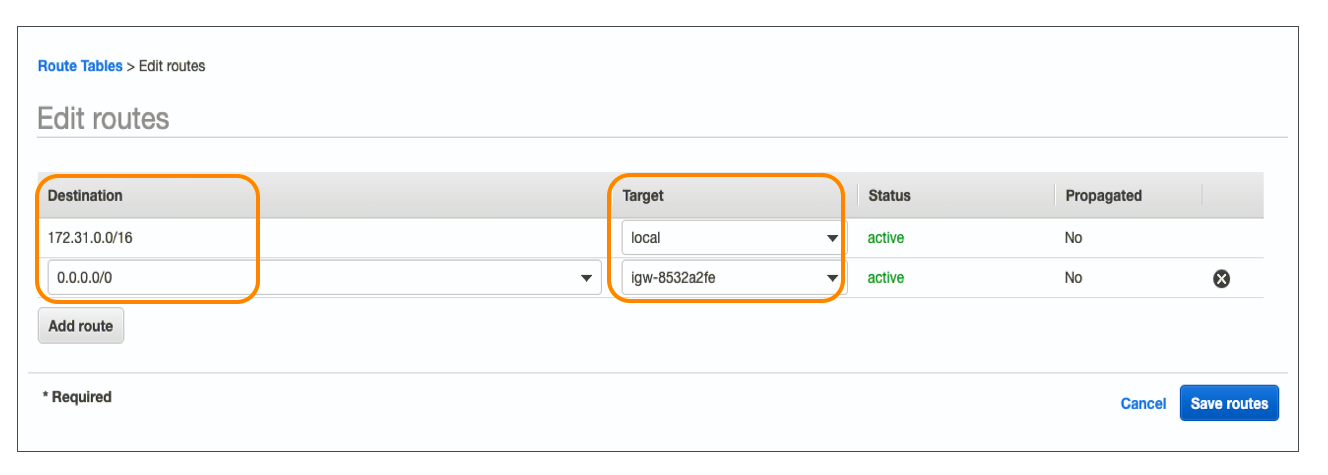
Route Table is roughly configuration files that explain how to go from destination X to destination Y or to the internet and which way to use.

**Routers** are components that manage the Route Tables and they act as “intersections” within the network.

Each AWS VPC has a VPC router.

Briefly, if we consider Route Table as a set of rules, then Router is executive of these rules.

Here, the example of the Route Table seen below.



If you want to reach somewhere in Destination Column you'll be directed to the value in Target Column

For example, You can reach **0.0.0.0/0 (Anywhere in Internet)** by using **igw-8532a2fe (Internet Gateway)** thanks to this route table.

### **Network ACLs**



Network ACL stands for Network Access Control Lists. It is a security component for your VPC that controls the traffic in and out of subnets. In other words, Subnet Network ACLs are **firewall of subnets**

Network ACLs are **subnet-based** security components.

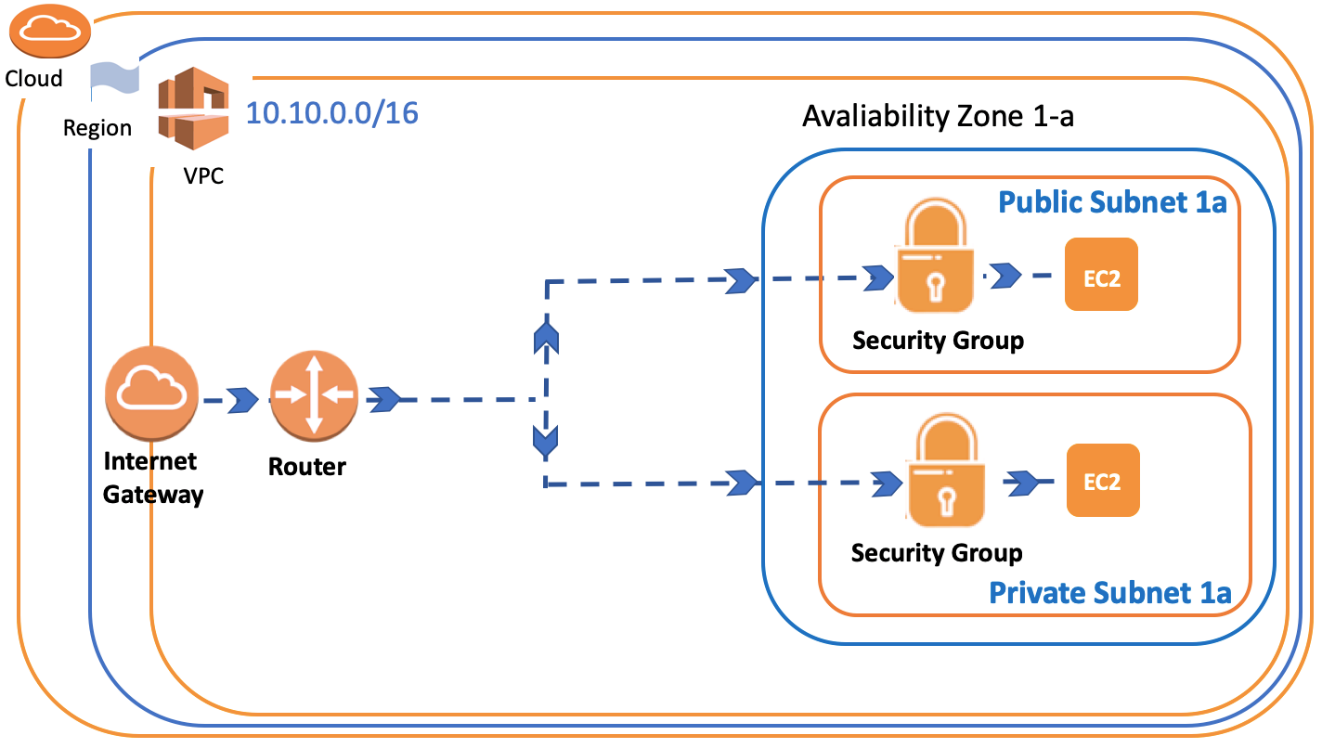
Your VPC automatically comes with a modifiable default Network ACL. **By default**, it **allows** all inbound and outbound IPv4 traffic.

* Unlike default Network ACL, If you want to create a new one, it denies all the inbound and outbound traffic until you add rules

Each subnet in your VPC must be associated with a Network ACL. If you don't explicitly associate a subnet with a Network ACL, the subnet is automatically associated with the default Network ACL.

You can associate multiple subnets with a Network ACLs. However, a subnet can be associated with the single Network ACL.

### **Security Groups**



In addition to the Network ACLs, we can also create Security Groups in VPC for security precautions.

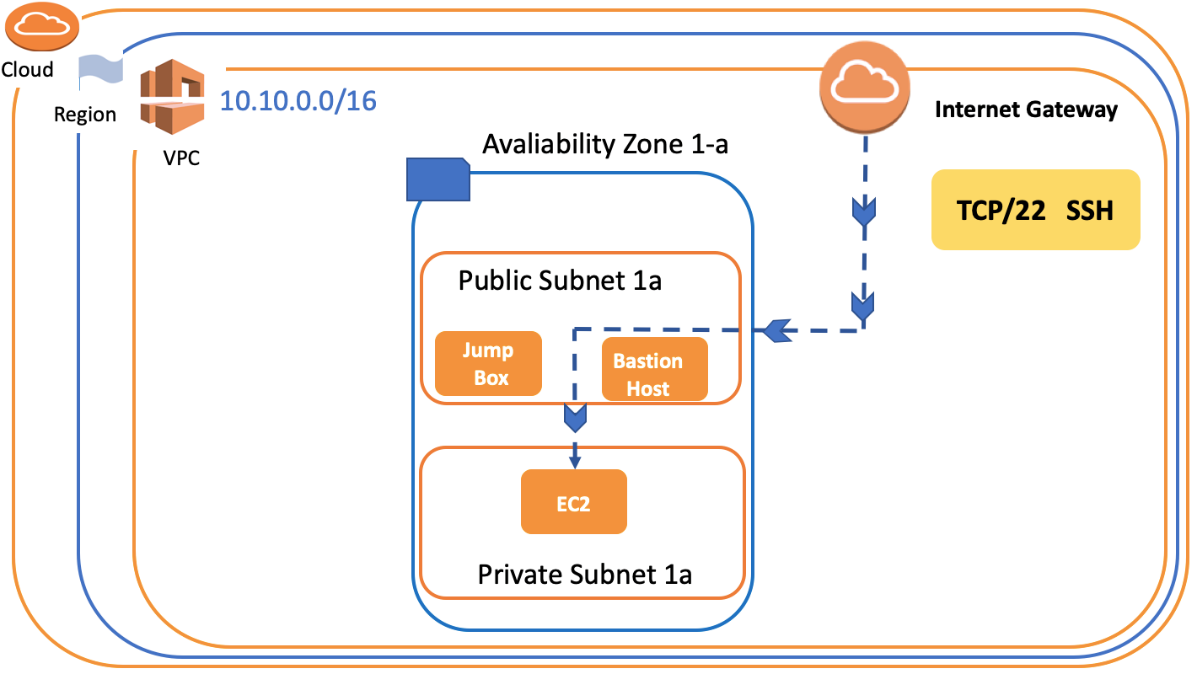
Security Groups, also mentioned in the EC2 section, are used for determining which traffic will access the instance.

In other words, a Security Group is a virtual **Firewall of Instance**.

Security Groups are **instance-based** components, while Network ACLs are the subnet-based. So, you need to consider both Security Groups and Network ACLs about the inbound and outbound traffic for instance in any subnet.

Each instance in a subnet in your VPC can be assigned to a different set of security groups. When you launch an instance in a VPC, you can associate with **five** security groups to one instance.

**Bastion Host/Jump Boxes**



As the name suggests, Private Subnet is closed to Public internet. But, sometimes we need to access the instance located in Private Subnet. So we use a proxy server/instance, Bastion Host, for this job.

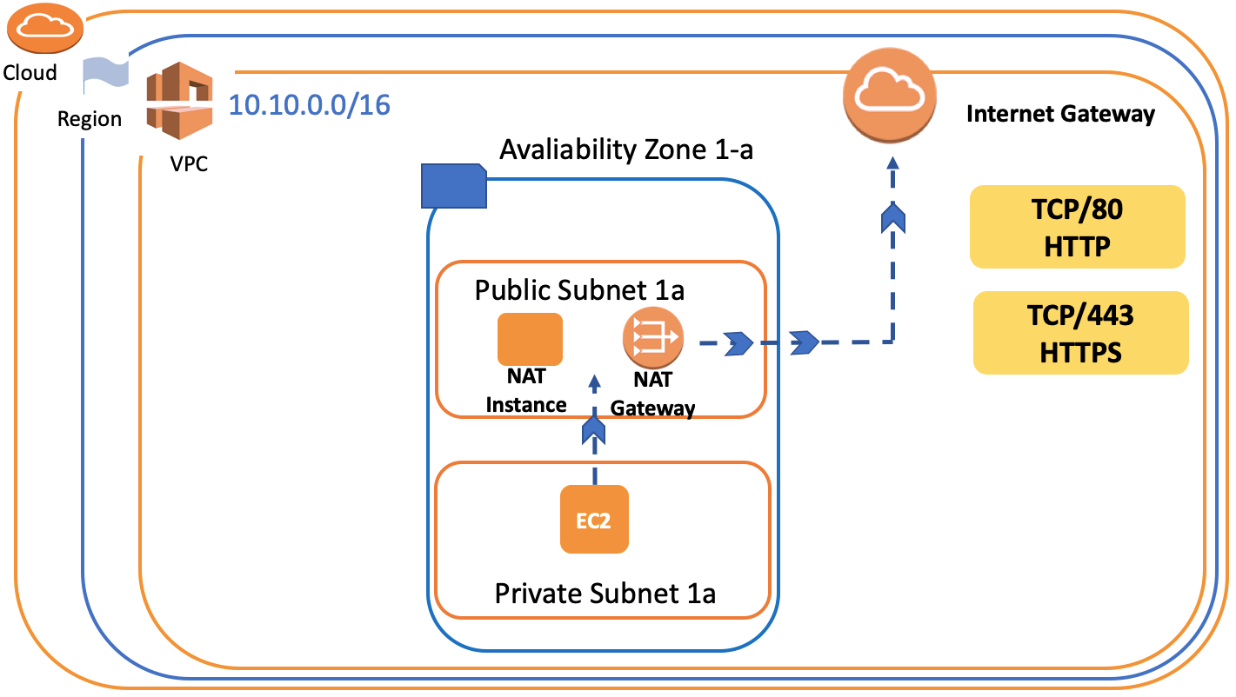
A Bastion Host is a server/instance that is used to ensure access to a Private subnet from the Internet.

Bastion Host is launched in **Public Subnets** and acts as a proxy for the instances in a **Private Subnet**. It provides security by reducing the attacks on your infrastructure.

A Bastion Host is used to administer EC2 instances using SSH or RDP securely. It is also known as **Jump Box**.

Bastion Host/Jump Boxes are used for **Inbound** traffic to the instance in Private Subnet.

### **VPC NAT Gateway & NAT Instance**



**NAT** stands for Network Address Translation in AWS.

You can't directly connect to the internet if your instance is in a Private Subnet. Because of the security precaution, your private subnet blocks the outbound internet connectivity.

We can use **NAT Gateways** or **NAT Instance**, which is created in the Public Subnet as a proxy to tackle this problem.

So if you want to connect from the EC2 instance in the private subnet to the internet, you need to create NAT Gateways or NAT Instance in Public Subnet. Then you'll obtain connectivity to your EC2 Instance via these components.

But it is important that a NAT instance and NAT Gateways allow your private instances to **outbound internet connectivity** while **blocking internet inbound** traffic.

**NOTE**:

All PCs and servers look at the Destination MAC and IP addresses of the incoming packet. If it does not see a MAC and IP of its own, it thinks that the packet arrived on the LAN by mistake and drops it. Since this device will do NAT, it will receive packets with other IP addresses from outside, we remove the **source.destination** check so that it does not drop these packets and route them.

**NAT Instance:**

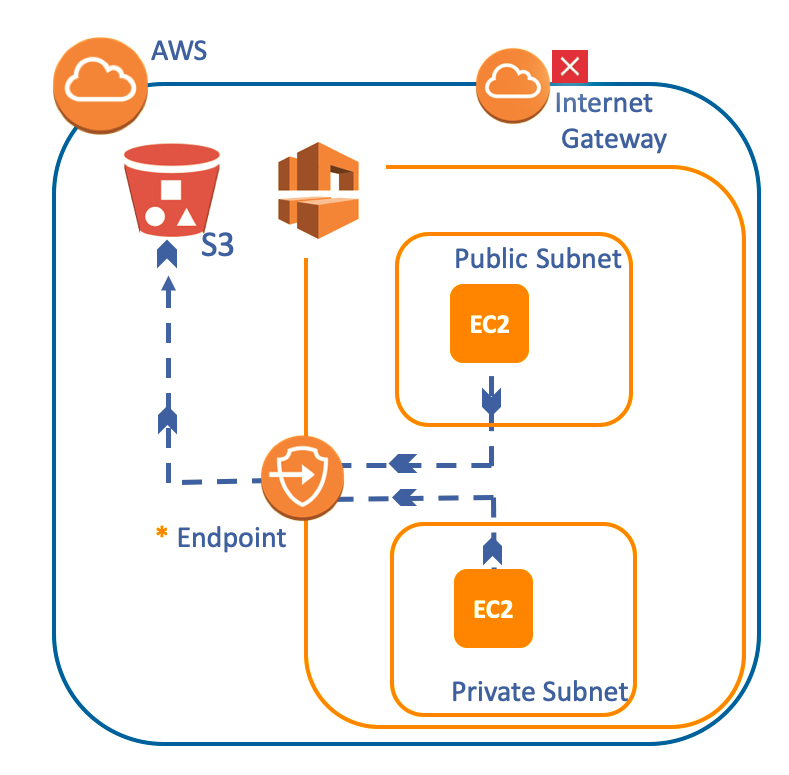
NAT Gateways or NAT Instance are pretty much similar things, and they provide the same functionality. However, a NAT Gateway is an AWS managed NAT service, but NAT Instance is specified and managed by the customer.

In fact, NAT Gateways are widely used in real-time than NAT instances, and NAT Gateways are highly accessible across multiple availability zones.

* You cannot use NAT Gateway as a Bastion host. If you connect with SSH or RDP to an instance in a private subnet, you need to configure a Bastion Host. You cannot use NAT Gateway.

While Bastion Host/Jump Boxes are used for Inbound traffic, NAT Gateway/NAT Instances are used for Outbound traffic.

### **VPC Endpoint**



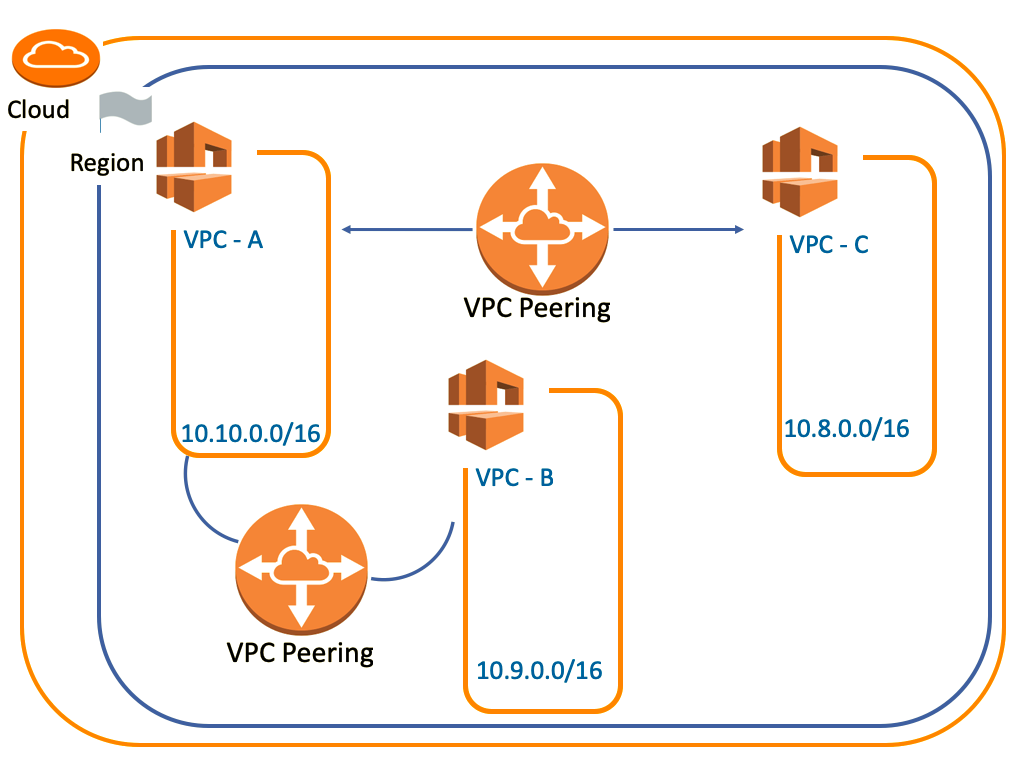
The VPC Endpoint is a component that allows you to privately connect your VPC to supported AWS services such as S3.

In fact, traffic between your VPC and the other service does not leave the Amazon network. So in this virtual environment, AWS offers us such a shortcut.

VPC Endpoint is a horizontally scaled, redundant and highly available VPC component. Thus it allows communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

Thanks to VPC Endpoint, Instances in your VPC do not require public addresses to communicate with the resources in the service. Because, VPC Endpoint services powered by PrivateLink without requiring an internet gateway, VPN Connection, etc.

### **VPC Peering**



A VPC peering is a networking connection between two VPCs.

It enables you to route traffic between them using private IPv4 addresses or IPv6 addresses. Instances in different VPC can communicate with each other as if they are within the same network.

Peering between the two VPC is peculiar to them. Let's evaluate the picture seen above to understand what it means.

Suppose that you have already had VPC Peering between VPC-A and VPC-B. If you create new VPC peering between VPC-A and VPC-C, connectivity between VPC-B and VPC-C does not occur automatically. You need to create another VPC peering between VPC-B and VPC-C for their connectivity.

### **Elastic IP**

An Elastic IP address is a **Static IPv4 Address** designed for dynamic cloud computing. In short, Elastic IP is a permanent IP for your instance.

Public IP addresses are dynamic. If you stop/start your instance you get reassigned a new public IP. But, Elastic IPs get allocated to your account and stay the same.

Elastic IPs are dynamically re-mappable IP addresses so it's up to you to attach them to another instance or not.

But, why do we need Elastic IP?

We use Elastic IPs for various reasons, especially because of its advantages or situations where it is compulsory to use.

For example;

* We may prefer to assign a license to Elastic IP address,
* It may be a legal requirement for some applications to use static IP.
* Some AWS components such as NAT Gateway and Route 53 may need Elastic IP while the process of creating, operating or setting up.

But for now, the part that interests us will be **NAT Gateways** and we'll see this in the following lessons in detail.

* Elastic IPs are totally free as long as they are being used by an instance. However, Amazon will charge you $0.01/hr for each EIP that you reserve and do not use. So don't forget to terminate the Elastic IP or associated component such as NAT Gateway if you'll not use anymore in the short term.

## VPN & Direct Connect

In previous lessons, we made a VPC in AWS accessible from other VPCs with VPC Peering. But what if you want to reach your Data Center outside the AWS through the VPC or vice versa?

AWS VPN lets you establish a secure and private tunnel from your network or device to the AWS Cloud. You can extend your existing on-premises network into a VPC, or connect to other AWS resources from a client. AWS VPN offers two types of private connectivity that feature the high availability and robust security necessary for your data.

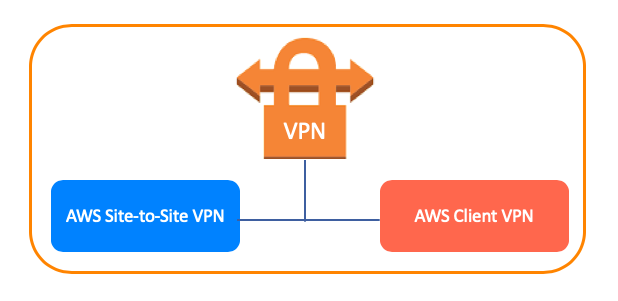
## 

### **VPN Categories**

AWS VPN is comprised of two services: **AWS Site-to-Site VPN** and **AWS Client VPN**.

AWS Site-to-Site VPN enables you to securely connect your **on-premises network** or **branch office** site from your Amazon VPC

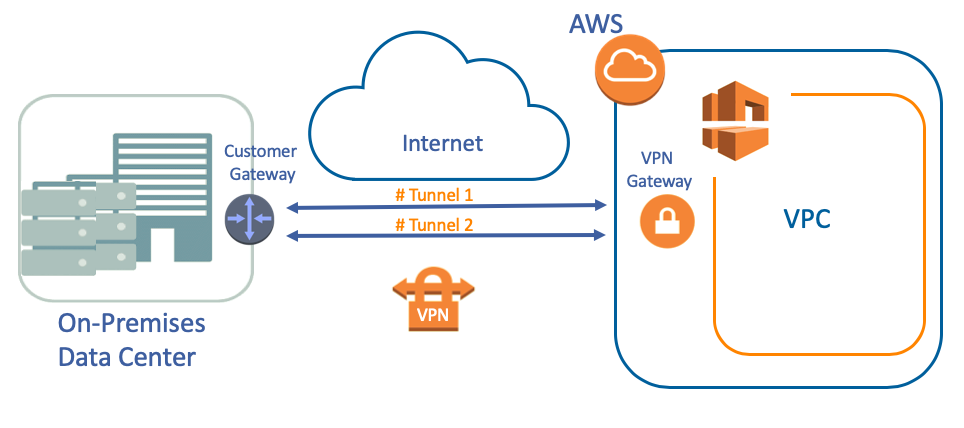
AWS Client VPN enables you to securely connect AWS or on-premises networks from **remote users**. AWS Client VPN includes a free client-side application, which provides access to AWS services from **remote networks**.



### **AWS Site-to-Site VPN**

AWS Site-to-Site VPN is one of the most common ways that customers connect securely to the AWS Cloud from on-premises or data center environments.

Thanks to the virtual network connection protocols we call VPN, we have the chance to establish encrypted communication networks that establish a connection between the company and AWS over the **Internet**.

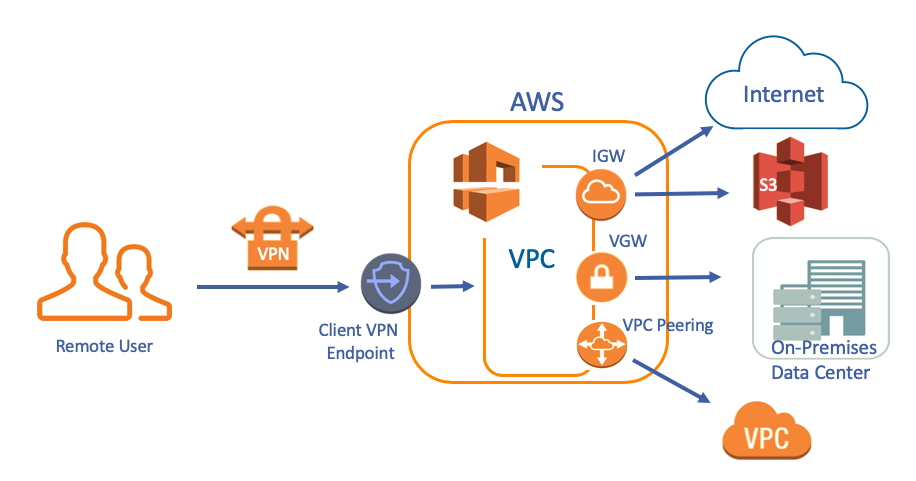


Its virtual edge on the AWS side is called **Virtual Private Gateway** and the end on the firm side is called **Customer Gateway**.

It uses internet protocol security (IPSec) communications to create encrypted VPN tunnels between two locations. Each VPN connection has **2 VPN Tunnels** which you can use for high availability.

You can stream primary traffic through the first tunnel and use the second tunnel for redundancy.

### **AWS Client VPN**



AWS Client VPN is a managed client-based VPN service that enables you to securely access your AWS resources and resources in your on-premises network.

With Client VPN, you can access your resources from any location using an OpenVPN-based VPN client.

Unlike on-premises VPN services, with AWS Client VPN, users can connect from anywhere to your AWS and on-premises networks.

* **For connecting with AWS Client VPN:**
  + First, you need to get a VPN Client Application,
  + Then create a new Client VPN Endpoint,
  + Configure the route tables for Client VPN Endpoint,
  + After that, associate a new Client VPN Endpoint to VPC and subnets
  + Get the Client VPN Endpoint Configuration File from AWS Management Console
  + The last, connect to the VPN.
* After you connecting to VPC, thanks to AWS Client VPN;
  + You can reach AWS resources in this VPC such as **S3 and also Internet** via **Internet Gateway**,
  + You can reach **On-Premises Data Center** via **Virtual Private Gateway**,
  + You can also reach the **other VPC** via **VPC Peering.**

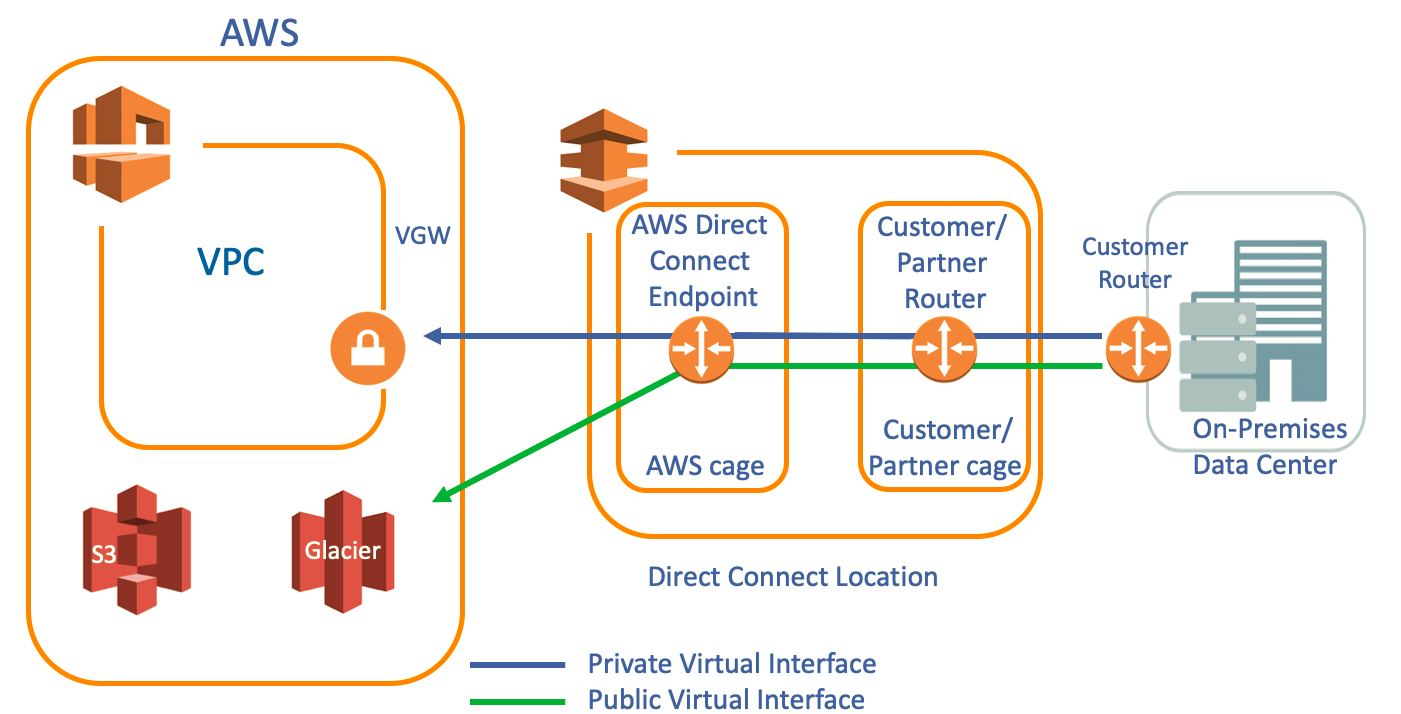
**Direct Connect:**

Although the VPN Site to Site and Client VPN methods are secured by using strong encryption techniques, communication flows over the internet.

In addition, the speed of the connection is determined by the speed of the end on the firm side. It may not meet our needs in fast connection and high bandwidth situations.

So, AWS has developed the Direct Connect service as a solution. Direct Connect is a service that establishes a **dedicated** network from your On-Premises Data Center to AWS.

### **The Architecture of Direct Connect**



The architecture of Direct Connect consists of 3 main components. These are **AWS Region**, **Direct Connect (DX) Location** and **Customer On-Premises Data Center**.

* **AWS Region:**

In AWS Region, we have AWS Public services such as S3, AWS Glacier, DynamoDB, etc. and also we have VPC which could be private.

* **Direct Connect (DX) Location:**
  + Direct Connect (DX) Locations are created by bilateral agreements with global networking partners. Direct Connect Locations are spread across the world. There are more than 100 DX locations available globally across major cities.
  + We have two cages, i.e., **AWS cage** and **Customer/Partner cage** inside Direct Connect Location,
  + AWS cage consists of **Direct Connect routers**, and the Customer/Partner cage consists of **Customer/Partner Routers**.
* **Customer On-Premises Data Center:**

We need to create a **Virtual Interface** to enable access to AWS services via Direct Connect as you see in the picture above.

**A Public Virtual interface** enables access to public services, such as Amazon S3.

**A Private Virtual Interface** enables access to your VPC.

Thanks to this system, AWS establishes a broadband closed-circuit network with its high connection speed service providers and users.