**IP Addressing in AWS**

There are two primary types of IP addresses that are available in AWS:

* Private IP
* Public IP (Dynamic IP)
* Elastic IP (Static IP)

**Private IP**

* Private IP is an IP address that is not reachable from the Internet.

**Public IP**

* Public IP is an IP address that is reachable from the Internet.
* Every instance that is given a public IP address is also given a DNS hostname, something like ec2-52-34-16-123.us-west-2.compute.amazonaws.com
* These are dynamic IP’s.

**Elastic IP**

* Elastic IP is an static IPV4 address and it is associated with your AWS account.
* To use Elastic IP, we first allocate it for our AWS account and then associate it with a instance that we need.
* The IP is associated with your AWS account until you release it.

***Important Note:***

***There is a small hourly charge if the Elastic IP is not associated or if it’s associated with a stopped instance.***

**Virtual Private Cloud (VPC)**

* Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) cloud where you can launch AWS resources in a virtual network that you define.
* Analogous to having your own DC inside AWS (means our own Datacenter on AWS cloud).
* Provides complete control over the virtual networking environment including selection of IP ranges, creation of subnets, and configuration of route tables and gateways.
* A VPC is logically isolated from other VPCs on AWS.
* VPC spans over region.
* A default VPC is created in each region with a subnet in each AZ.
* By default, you can create up to 5 VPCs per region.
* You can define dedicated tenancy for a VPC to ensure instances are launched on dedicated hardware (overrides the configuration specified at launch).

**subnet:**

* It is like smaller network in VPC.
* You must have atleast one subnet to deploy your resources like EC2, RDS etc.

**IP Address Pointer - Analogy**

All the resources inside VPC will have an IP address from the VPC CIDR.

Let’s understand this with an example:

We assign one hundred number to our VPC (0-100)

Every resource in VPC will be assigned one of these numbers.

* EC2 Instance 1 - Number 4
* EC2 Instance 2 - Number 10

**Ex:**

Our VPC has a CIDR of 10.77.0.0/16

Total IP Addresses: 65,536

Every resource in VPC will be assigned IP addresses from the given pool.

* EC2 Instance 1 - 10.77.0.5
* EC2 Instance 2 - 10.77.0.10

In order to create VPC we need to understand the concept of CIDR Block.

CIDR: Classless Inter Domain Routing

There are 2 version of IP address:

* IPv4

Which has **32 bits**

Looks like **X.X.X.X**

Ex: **172.168.0.0**

* IPv6

Which has **128 bits**

Looks like **X.X.X.X.X.X.X.X**

Ex: **2001:0db8:85a3:0000:0000:8a2e:0370:7334**

Let’s consider a IPv4 CIDR.

**X.X.X.X/Y**

Each **X** represents one octet i.e 8bit

**Y** represents network portion

**32-Y** represents host portion

2^(32-Y) is the no. of IP address we will get in the CIDR block

Ex:

10.190.0.0/24

24:- Network portion

32-24=8:- Host portion

Then 2^8 = 256 (Total no. of IP address will get in this CIDR)

Now we understood how to calculate the no. of IP addresses we can get from a CIDR block.

Let’s understand how to divide the IP addresses we got from a CIDR block in order to create multiple subnets in a VPC.

Example1:

Suppose I have a CIDR block 192.168.0.0/24

24 : Network portion

32-24=8 : Host portion

2^8= 256 : Total no. of IP address

Now if I will create one subnet then I can allocate all the IP address to a single subnet

192.168.0.0/24

192.168.0.(0-255)

If I will create two subnets then

192.168.0.0/25 (subnet 1)

192.168.0.(0-127)

192.168.0.128/25 (subnet 2)

192.168.0.(128-255)

Example2:

-----------------

Note:

* While designing the VPC and subnet we must always use private IP address.
* Every subnet in AWS reserves 5 IP address for internal use.

**Subnet Types:**

* **Public Subnet:**
  + - A subnet which is exposed to the internet is called public subnet.
    - All internet facing application must be in public subnet.
    - Ex. Web application
  + **Private Subnet:**

1. A subnet which is not exposed to internet is called private subnet.

Ex. Database

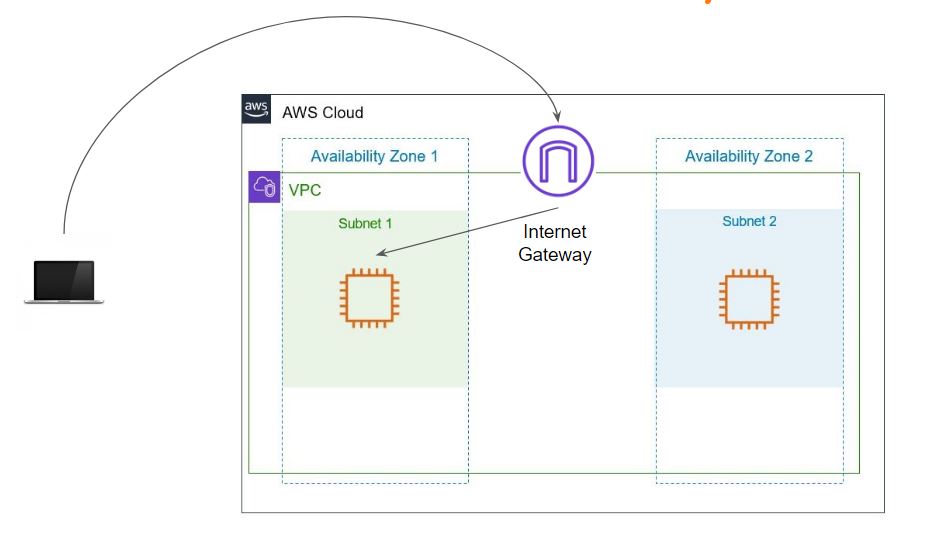
**Note:**

*By default, whatever the subnet we create inside our custom VPC, it is private subnet.*

*In order to make a subnet public we need to attach a Internet Gateway.*

## **Internet Gateways**

* Internet Gateway is a highly available VPC component that allows communication between resources in the VPC and the Internet.



End to End Steps for Public Connectivity:-

* Create an Internet Gateway
* Attach the Internet Gateway to the VPC
* Ensure that subnet’s route table points to the Internet Gateway
* Ensure that the instances have Public / EIP attached.

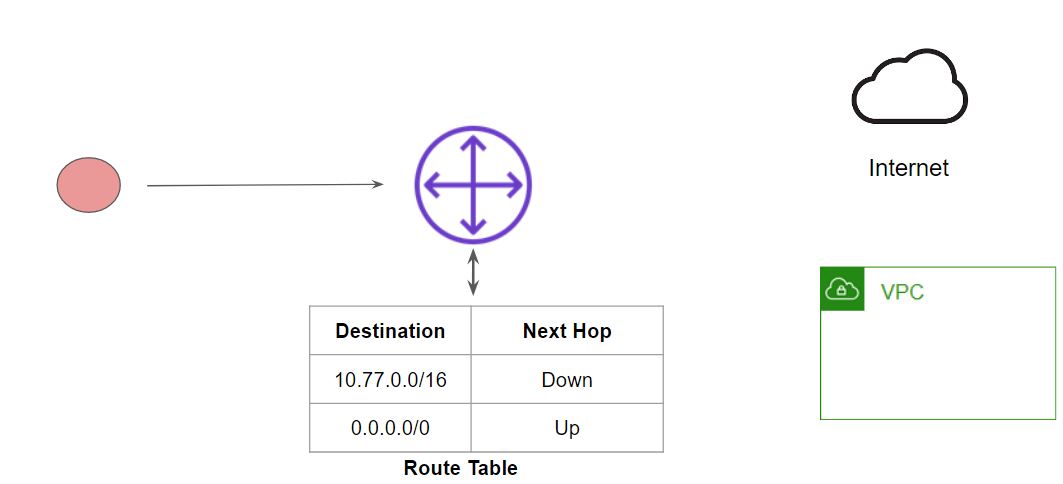
## **Overview of Route Tables**

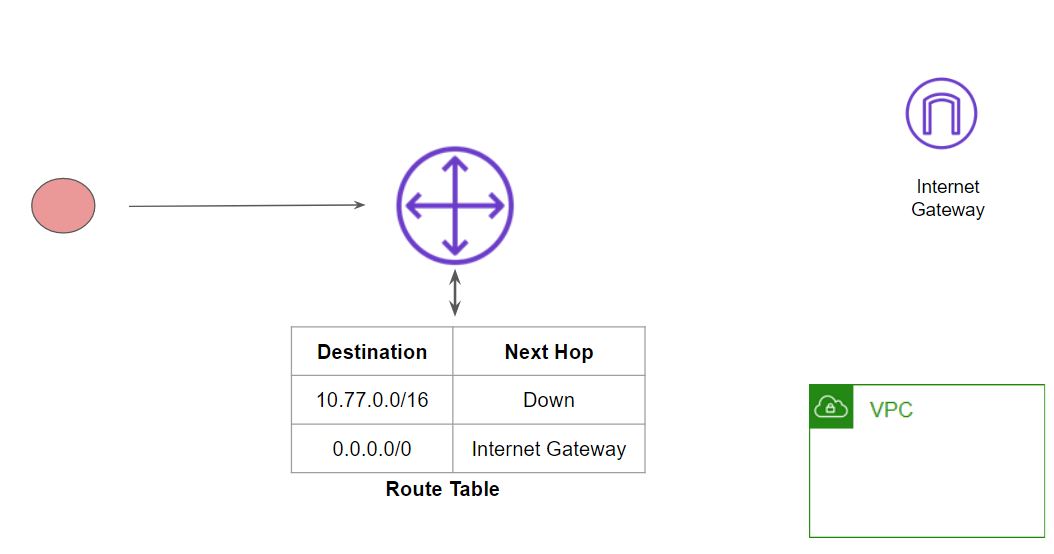
* Suppose you want to travel somewhere direction Billboards are very useful to determine the next turning point to reach the destination.



Overview of Routers

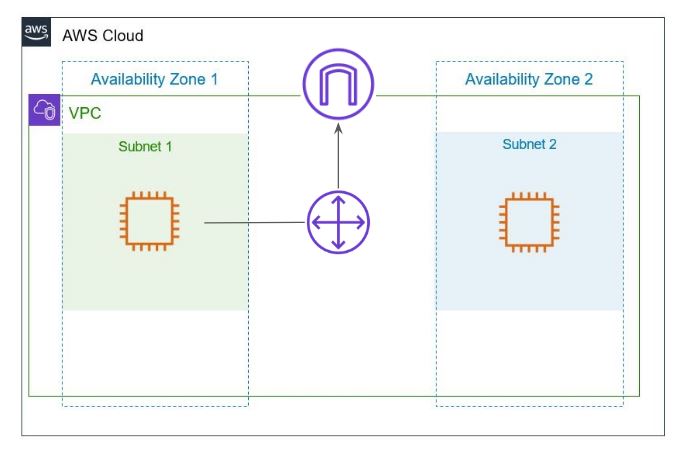
A router is a networking device that forwards data packets between networks.



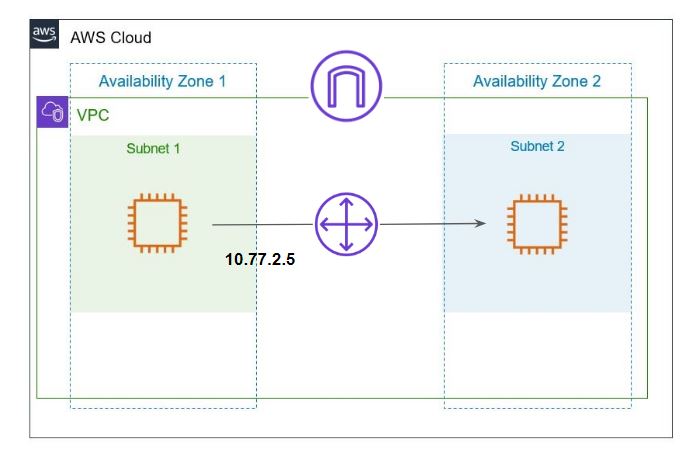


**Route Table in AWS**

* 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.
* By default, whenever a VPC is created, the route table is also created.
* We can create custom route table for our subnets.
* The following diagram depicts High-Level Working - Internet Route



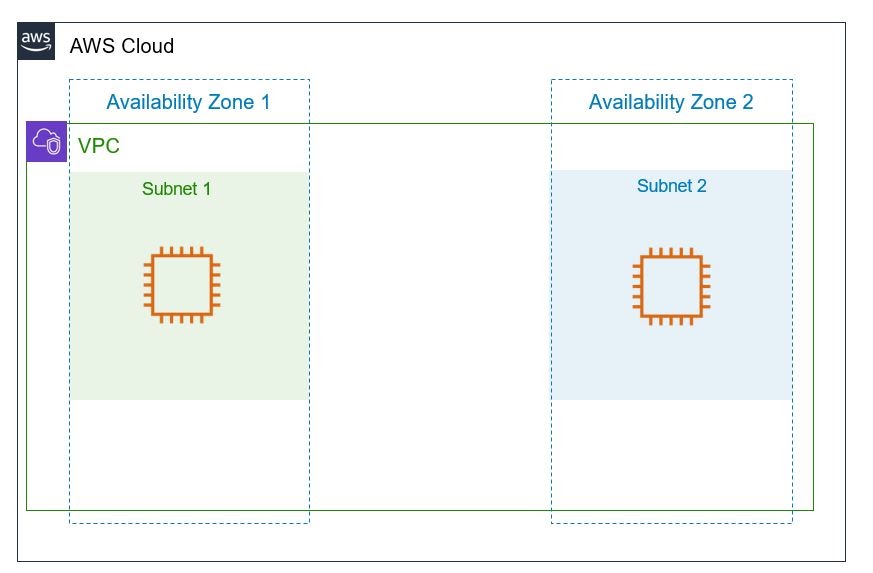
The following diagram depicts High-Level Working - Local Route



## **Public and Private Subnets**

Public Subnets

* Public subnet is a subnet that is associated with an Internet Gateway.
* This subnet is recommended if you want to run a public-facing web application.
* Overall Security Risk:  High



Private Subnets

* Private subnets are the ones that do not have an Internet Gateway attached to it.
* connections from the Internet cannot reach to the EC2 instances within the private subnet.
* Instances in private subnet do not have a Public IP / Elastic IP attached.

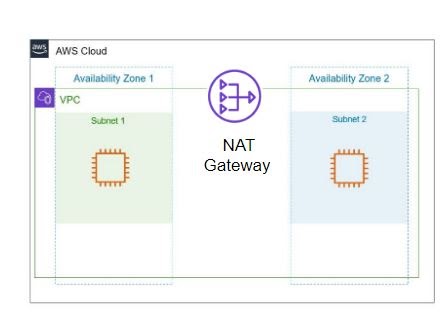
Example Use-Case:

Database Servers

Servers who do not directly interact with Internet resources.

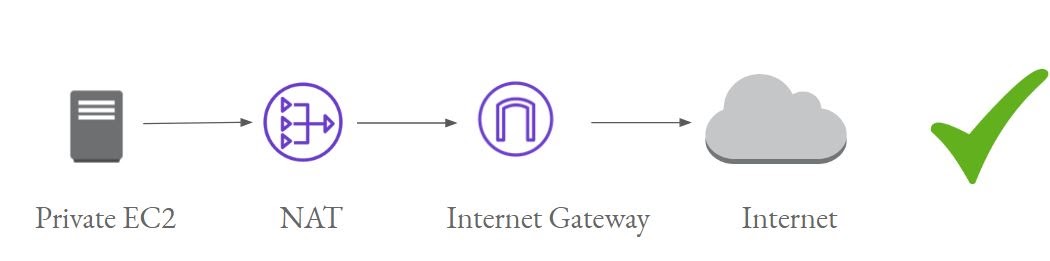
Use-Case of NAT Gateway:

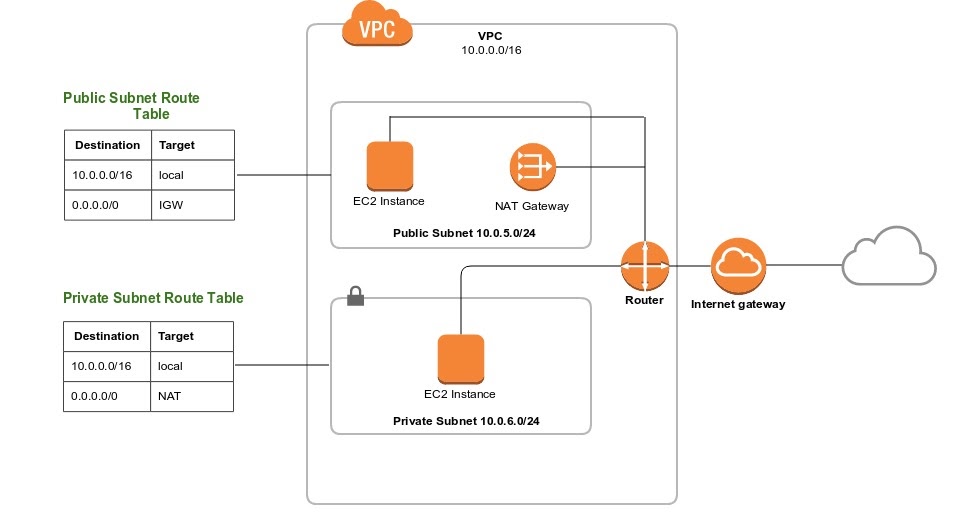
* EC2 Instances from a private subnet should be able to connect to the internet to perform various activities like patch updates, downloading software, and others.
* To achieve this use-case, NAT Gateways are introduced.



Overview of NAT Gateways

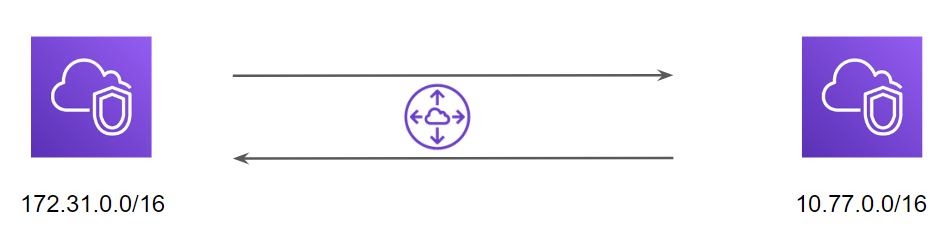
* You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet





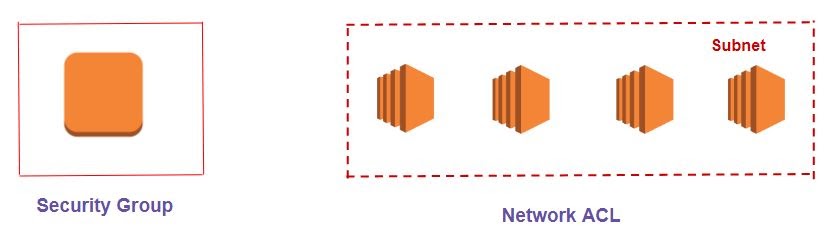
**VPC Peering**

* Connect two VPC, privately using AWS’ network
* Make them behave as if they were in the same network
* Must not have overlapping CIDR (IP address range)
* VPC Peering connection is not transitive (must be established for each VPC that need to communicate with one another)



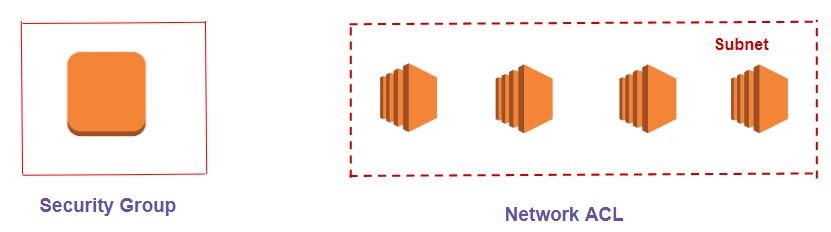
**Network ACL (NACL)**

* Network ACL are stateless in nature.
* They operate at the subnet level instead of instance-level like Security Groups.
* All subnets in VPC must be associated with NACL.
* By default, Network ACL contains full allow in INBOUND and OUTBOUND.



Let’s understand the above with Use-Caes:

* Company XYZ is getting a lot of attacks from a random IP 128.190.12.32. The company has more than 500 servers and the Security team decided to block that IP in the firewall for all the servers.
* How to go ahead and achieve that goal?

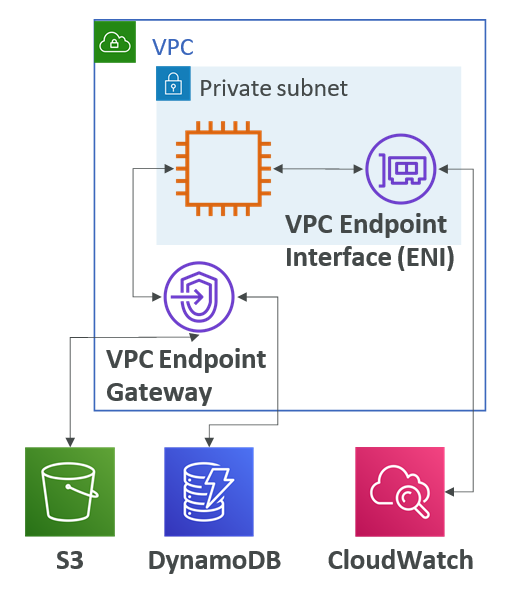


**VPC Flow Logs**

* Capture information about IP traffic going into your interfaces:
* VPC Flow Logs
* Subnet Flow Logs
* Elastic Network Interface Flow Logs
* Helps to monitor & troubleshoot connectivity issues. Example:
  + Subnets to internet
* Subnets to subnets
* Internet to subnets
* Captures network information from AWS managed interfaces too: Elastic Load Balancers, ElastiCache, RDS, Aurora, etc…
* VPC Flow logs data can go to S3 / CloudWatch Logs

**VPC Endpoints**

* Endpoints allow you to connect to AWS Services using a private network instead of the public www network
* This gives you enhanced security and lower latency to access AWS services
* VPC Endpoint Gateway: S3 & DynamoDB
* VPC Endpoint Interface: the rest
* Only used within your VPC



**Site to Site VPN & Direct Connect**

* Site to Site VPN
  + Connect an on-premises VPN to AWS
  + The connection is automatically encrypted
  + Goes over the public internet
* Direct Connect (DX)
  + Establish a physical connection between on- premises and AWS
  + The connection is private, secure and fast
  + Goes over a private network
  + Takes at least a month to establish
* Note: Site-to-site VPN and Direct Connect cannot access VPC endpoints

