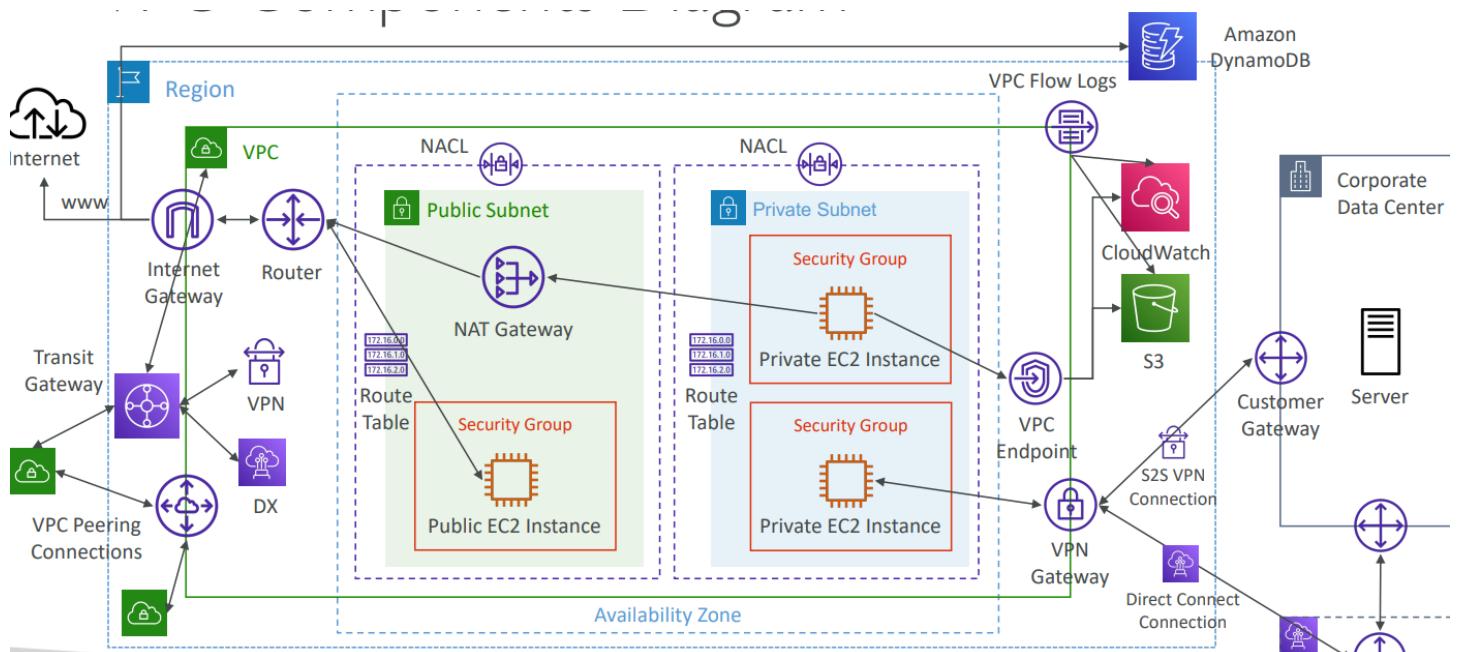


# Creating an AWS VPC from Scratch

## (EnterPrise Grade )

~Ritvik Kant



This is the final VPC which I will construct step by step

# **VPC**

Stands for Virtual Private Cloud

In this guide, I'll create a custom VPC in AWS from scratch, including subnets, route tables, Internet connectivity, and essential components. This provides better control and security than using the default VPC.

This VPC **WILL BE** an Enterprise grade VPC with options and features which ensure

1. High Availability and scalability
2. Resilience
3. Security
4. High performance and traffic spiking workloads

This is Ritvik and here is my take of a fault proof and an all rounder cost optimized and **ENTERPRISE GRADE VPC**

Let's make it from scratch

**Step1. Creating a VPC and establishing subnets in a region**

VPC as stated is a virtual private cloud in short like an independant society with residents not dependant on the outside for any resources

Imagine Subnets like the houses in the Neighbourhood Some are public like movie halls and parks

Rest are private like residential Homes



Screenshot of the AWS VPC Subnets page. The left sidebar shows the 'Virtual private cloud' section with 'Subnets' selected. The main area displays a table titled 'Subnets (6) info' with the following data:

Subnet ID	State	VPC	Block Public...	IPv4 CIDR	IPv6 CIDR
subnet-087e486d2b3fbce5d	Available	vpc-03d1f5a9b8e9c526e   dem...	Off	10.0.1.0/24	-
subnet-0f9e0fa1c50402b5d	Available	vpc-02c0cb56c281c19c9   defa...	Off	172.31.0.0/20	-
subnet-096d3b467ce2bcbf	Available	vpc-02c0cb56c281c19c9   defa...	Off	172.31.16.0/20	-
subnet-09d268e3052159647	Available	vpc-03d1f5a9b8e9c526e   dem...	Off	10.0.0.0/24	-
subnet-07a84f294ce1eea07	Available	vpc-02c0cb56c281c19c9   defa...	Off	172.31.32.0/20	-
subnet-0c548e7b3005d8fe6	Available	vpc-03d1f5a9b8e9c526e   dem...	Off	10.0.2.0/24	-

Screenshot of the AWS VPC Details page for 'demo-vpc'. The left sidebar shows the 'VPC' section with 'Show details'. The main area displays the following information:

- Subnets (3)**: Subnets within this VPC
- ap-south-1a**:
  - A newBabySubnet
- ap-south-1b**:
  - B babysubnet2
- ap-south-1c**:
  - C newbabysubnet3

Step-2

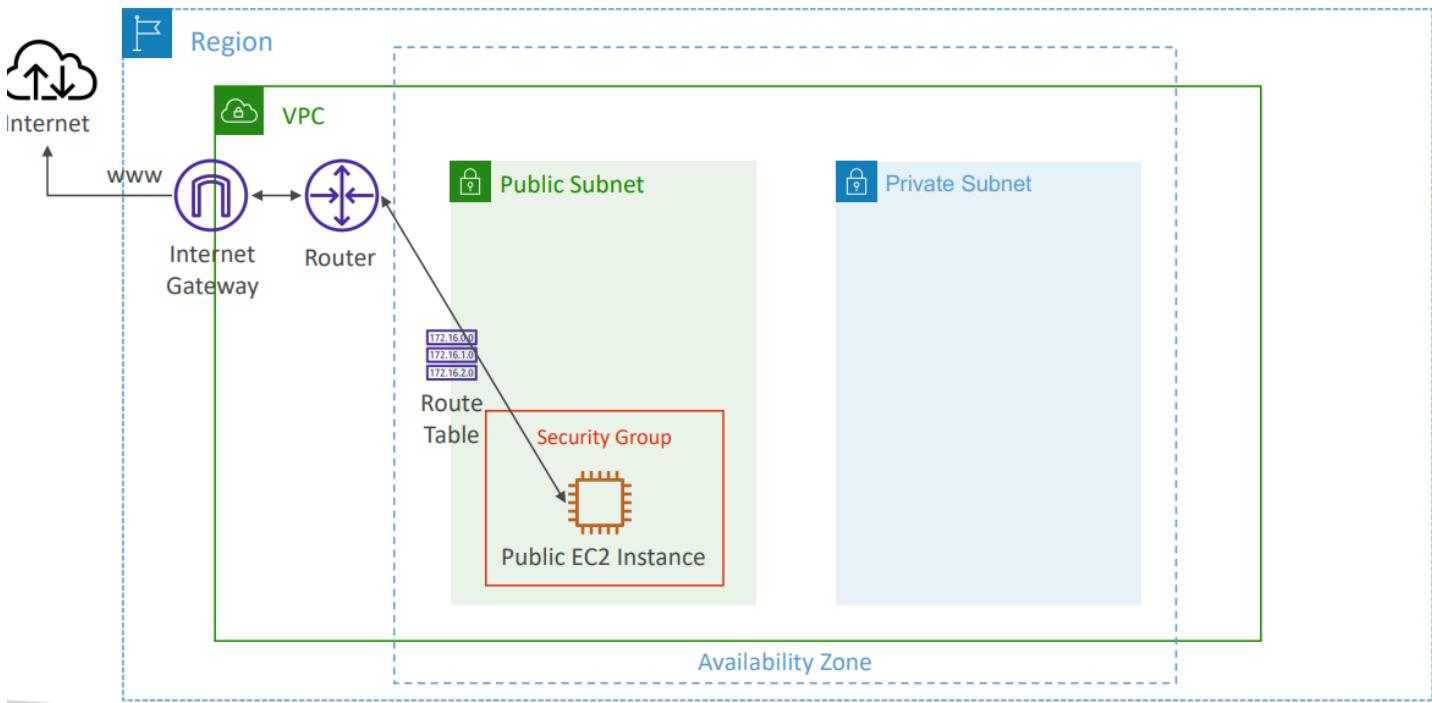
# Internet GateWay

Now that the neighbourhoods created ,we require a connection for import exports or in cloud terms Internet Access to the instances containing the app

I created an internet Gateway now the diagram is like this

Note:

- It scales horizontally and is highly available and redundant
- Must be created separately from a VPC
- One VPC can only be attached to one IGW and vice versa
- Internet Gateways on their own do not allow Internet access...
- Route tables must also be edited!



Below shows the attached Internet Gateway  
and the route table records being attached to the Internet Gateway

PUBLIC SUBNET	PRIVATE SUBNET
Its personal Route table is created pointing the Internet Gateway	Its personal Route table is created either pointing to a Bastion Host or private networks only

The screenshot shows the AWS VPC Internet Gateways page. The top navigation bar includes the AWS logo, search bar, and navigation links for VPC, Internet gateways, and the specific gateway ID 'igw-Off212bc85a35b62e'. The main content area displays the 'igw-Off212bc85a35b62e / demoInternetGateway' details. It shows the Internet gateway ID, state (Attached), VPC ID, and owner information. Below this, the 'Tags' section lists a single tag 'Name: demoInternetGateway'. The bottom section, titled 'Routes', lists three routes: one to the target 'igw-Off212bc85a35b62e' (Status: Active, Propagated: No), one to 'local' (Status: Active, Propagated: No), and one to the target 'pcx-00ca29223482e24c7' (Status: Active, Propagated: No). Navigation links for Subnet associations, Edge associations, Route propagation, and Tags are also present.

Now Ritvik You just talked About how public subnets are visible but Private subnets are not  
So what if I want to access my instances /App on the private instance for the public but I don t want to show Them **#Principle of OOPs-->Abstraction**

# Bastion Hosts (Step-2.5)

These are the types of instances which are hosted on public subnets have access and are visible to the public and can operate resources to the private subnet instances simply but doing SSH into those instances

Here I have

1. Created instances named bastion host in public subnet and one in private subnet
2. Modified the security group inbound rules of private instance to allow traffic on port 22(SSH Port) to point **FROM THE SECURITY GROUP OF BASTION HOST**
3. Now I ssh into the Bastion host
4. From here I ssh into the private instance – make a .pem file ,import it in the terminal itself and echo a “Hello” in Private instance
5. As soon as I curl cmd(For calling) it in the public instance it returns the desired output

The screenshot shows the AWS EC2 Instances page. A specific instance, 'i-07750df17e97ae6ae', is selected. The 'Security' tab is active, displaying the following details:

- Required:** arn:aws:ec2:ap-south-1:254159011250:instance/i-0 7750df17e97ae6ae
- Operator:** -
- Details:** Owner ID (redacted)
- Launch time:** Thu Jul 24 2025 14:33:32 GMT+0530 (India Standard Time)
- Security groups:** sg-0e1592c1232eeef243 (PrivatrSG)
- Inbound rules:** Port range: 22, Protocol: TCP, Source: sg-02ec7f9850eb46715, Security groups: PrivatrSG, Description: allow ssh from bastion host.
- Outbound rules:** (empty)

The screenshot shows an AWS CloudShell terminal window. The user is connected via SSH to a private instance (IP: 10.0.0.54). The session output is as follows:

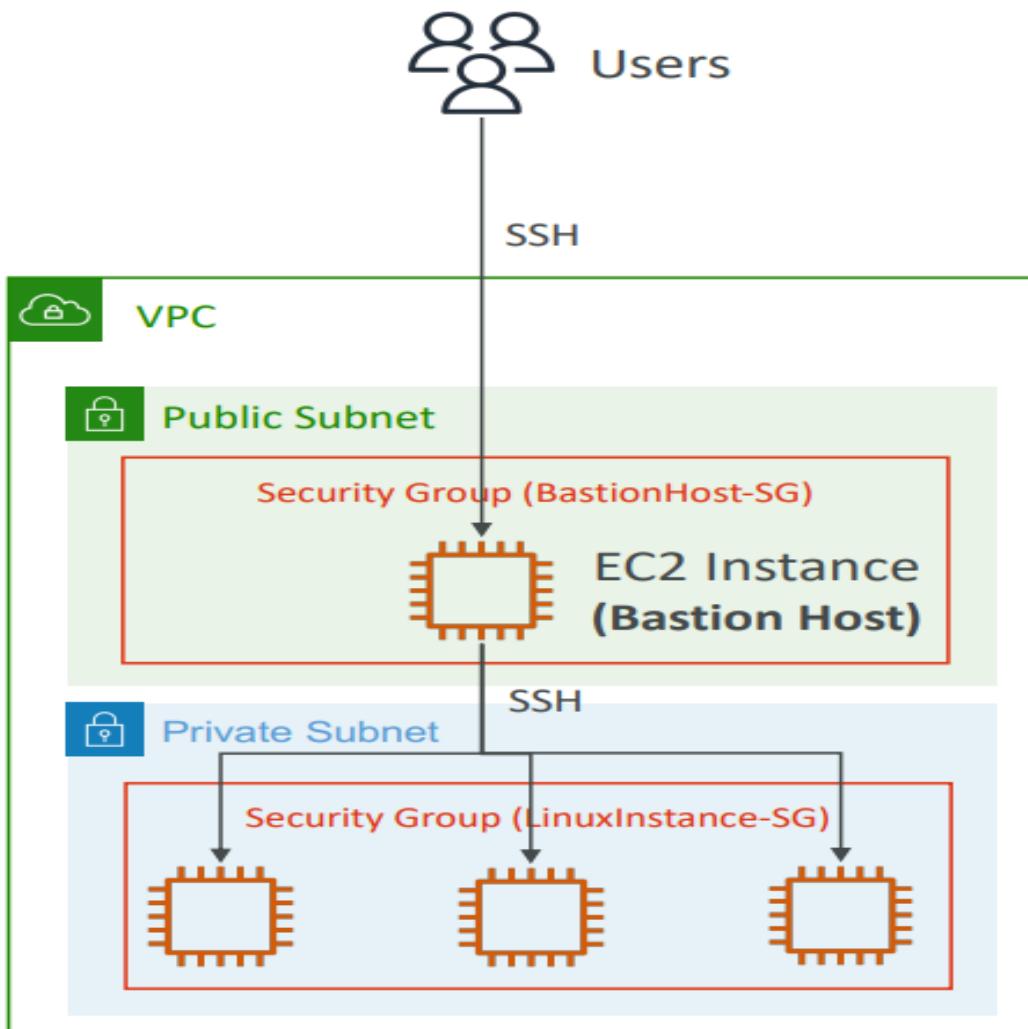
```
ec2-user@ip-10-0-1-50:~$ ssh ec2-user@10.0.0.54 -i demokey.pem
The authenticity of host '10.0.0.54 (10.0.0.54)' can't be established.
ED25519 key fingerprint is SHA256:CB0Tp6m8iwdsGSMED2gAMPC2BcO/RYvypAWqFJGsk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.0.54' (ED25519) to the list of known hosts.
# 
# Amazon Linux 2023
# https://aws.amazon.com/linux/amazon-linux-2023
/m/
ec2-user@ip-10-0-0-54:~]$
```

At the bottom of the terminal, the instance identifier is shown: **i-0bdb8658ca93ef1ed (BastionInstance)**. Below that, it says **PublicIPs: 13.232.248.149 PrivateIPs: 10.0.1.50**.

## ExpectedOutput

```
[ec2-user@ip-172-31-3-7 ~]$ curl 10.0.1.50  
hello  
[ec2-user@ip-172-31-3-7 ~]$ █
```

NOW THE Structure LOOK LIKE THIS



# Step 3

Increasing level [#####----]

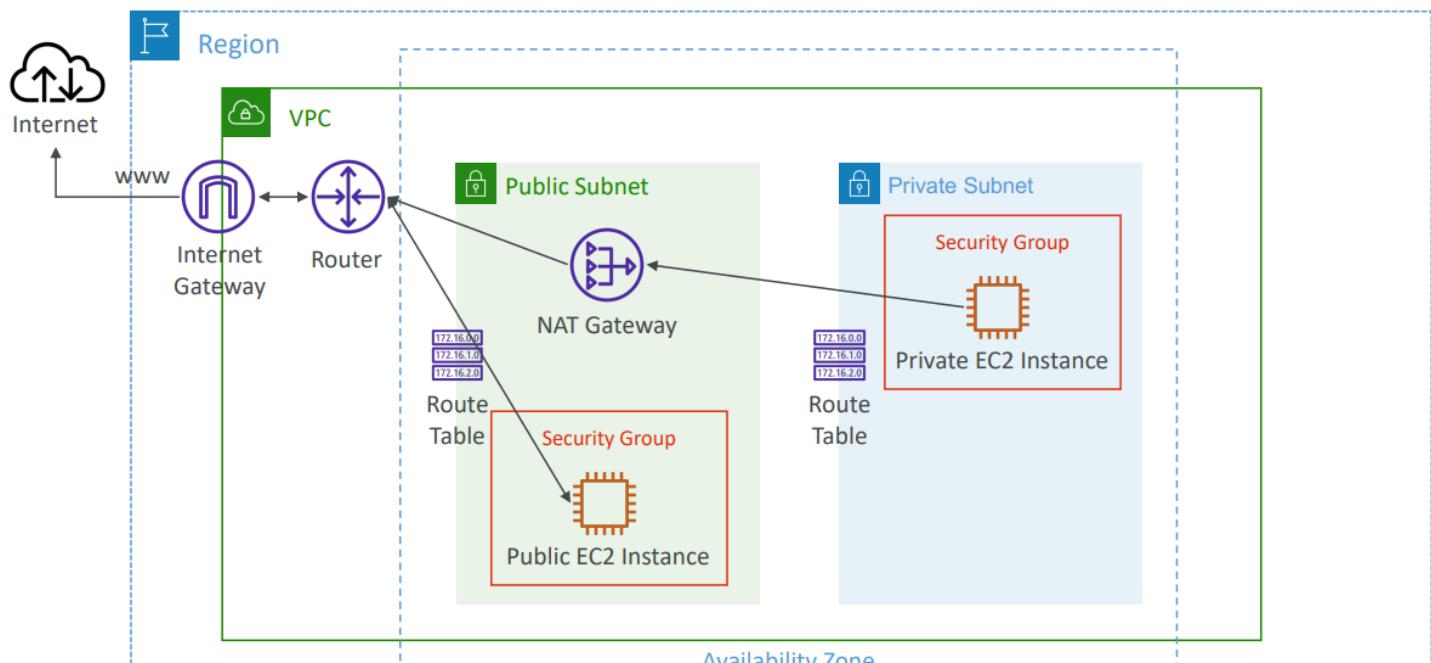
This method is great but not reliable for bigger workloads so

## NAT GATEWAY

This thing ensures a stable internet connection both in and out be given to private subnets with a window for security checks ofc!

AWS-managed NAT, higher bandwidth, high availability, no administration

- NATGW is created in a specific Availability Zone, uses an Elastic IP
- Can't be used by EC2 instance in the same subnet (only from other subnets)
- Requires an IGW (Private Subnet => NATGW => IGW)
- 5 Gbps of bandwidth with automatic scaling up to 100 Gbps



Now Comes the Security Part to secure the infrastructure for

- 1.Malicious Traffic
2. Misguided Traffic

## Step-4

### NACL(s) /Security Groups

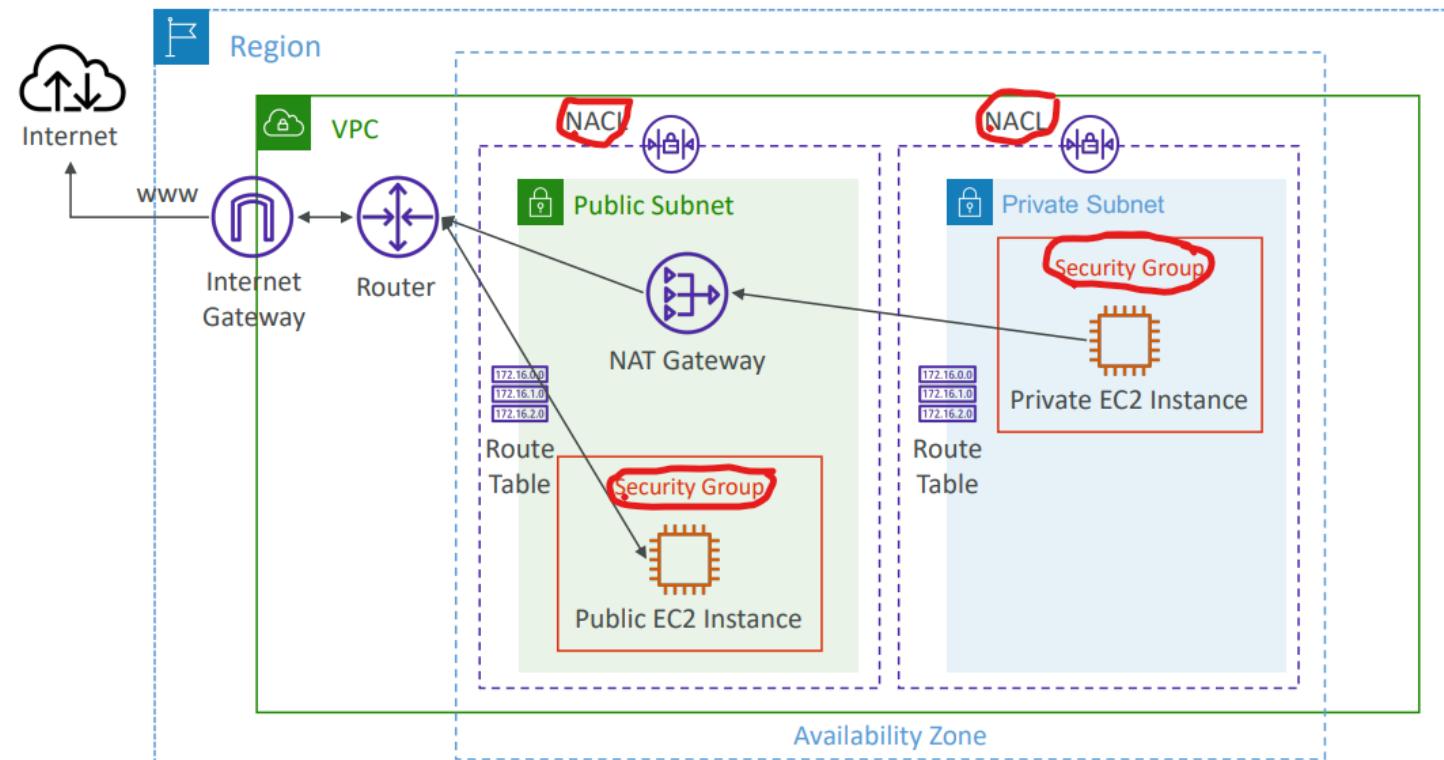
NACL(s) are Subnet Level completely stateless(Check both in and out traffic) and are based on Rules priority no.s

Security groups are instance level to further prevent the misguided traffic to reach its correct path and may not activate a wrong instance these are stateful(logical people if in then out and vice versa types) and have no priority no. System just declarations

There are many places where I used NACLs to block certain CIDRs

And off those Which enter inside to guide through security groups

So here is our progress



## **NOTE:**

**EPHIMERAL PORTS**- these are on spot usable ports used instead of defined ports for communication for added security

Now If the app is Enterprise grade and the infrastructure is also Enterprise grade  
It can not rely on only 1 VPC

We have to have multiple VPC but that is not the point that is simple the point is  
**WE NEED SYNC** and updated Information between all the VPCs for the companies  
Also an Advantage of this will be to share info across VPC ,,,next Feature

## **Step 4**

### **VPC Peering**

**It simply means establishing a peering connection or simply a connection between VPCs **CROSS ACCOUNT & CROSS REGION****

**Here I set up**

- 1. TWO VPCs in different region**
- 2. Accept the request of peering**
- 3. MOST IMP. ALWAYS update it in the route tables which I will see in common once both VPCs are connected as shown**

## Accept VPC peering connection request X

### Info

Are you sure you want to accept this VPC peering connection request? (pcx-00ca29223482e24c7 / my-pc-1)

#### Requester VPC

vpc-03d1f5a9b8e9c526e / demo-vpc

#### Acceptor CIDRs

-

#### Requester owner ID

 254159011250(This account)

#### Acceptor VPC

vpc-02c0cb56c281c19c9 / defaultVPC

#### Requester Region

Mumbai (ap-south-1)

#### Acceptor owner ID

 254159011250(This account)

#### Requester CIDRs

 10.0.0.0/16

#### Acceptor Region

Mumbai (ap-south-1)

[Cancel](#)

[Accept request](#)

≡ [VPC](#) > [Peering connections](#) > Create peering connection ( )

Select a local VPC to peer with

VPC ID (Requester)  
vpc-02c0cb56c281c19c9 (defaultVPC)

VPC CIDRs for vpc-02c0cb56c281c19c9 (defaultVPC)

CIDR	Status	Status reason
172.31.0.0/16	 Associated	-

Select another VPC to peer with

Account  
 My account  
 Another account

Region  
 This Region (ap-south-1)  
 Another Region

VPC ID (Acceptor)  
vpc-02c0cb56c281c19c9 (defaultVPC)

VPC CIDRs for vpc-02c0cb56c281c19c9 (defaultVPC)

CIDR	Status	Status reason
172.31.0.0/16	 Associated	-

Tags

## Changing the route Table

**Edit routes**

**Route 1**  
**Destination** 10.0.0.0/16  
**Status** Active

**Target** local

**Propagated** No

---

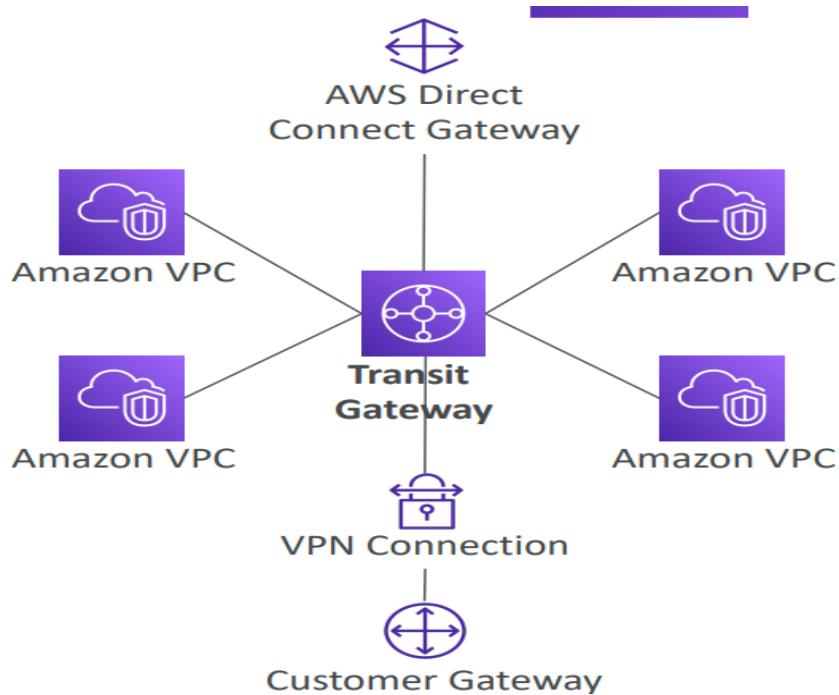
**Route 2**  
**Destination** 172.31.0.0/16  
**Status** -

**Target** Peering Connection  
   
 Use: "pcx-00ca29223482e24c7"  
**pcx-00ca29223482e24c7 (my-pc-1)"**

## ADDITIONAL STEP FOR ENTERPRISE GRADE

Setup a Transit Gateway for incase of connecting multiple VPCs together using peerings

**Reason:** Network Topology becomes **complicated** and difficult to maintain



## Some Cost Optimizing approaches....

Okay So Now the thing is I want my Private Instance to access AWS services or my private DATA Center

Why should i incur such costs of setting up a NAT gateway then accessing those services through public

That's expensive and not even secure

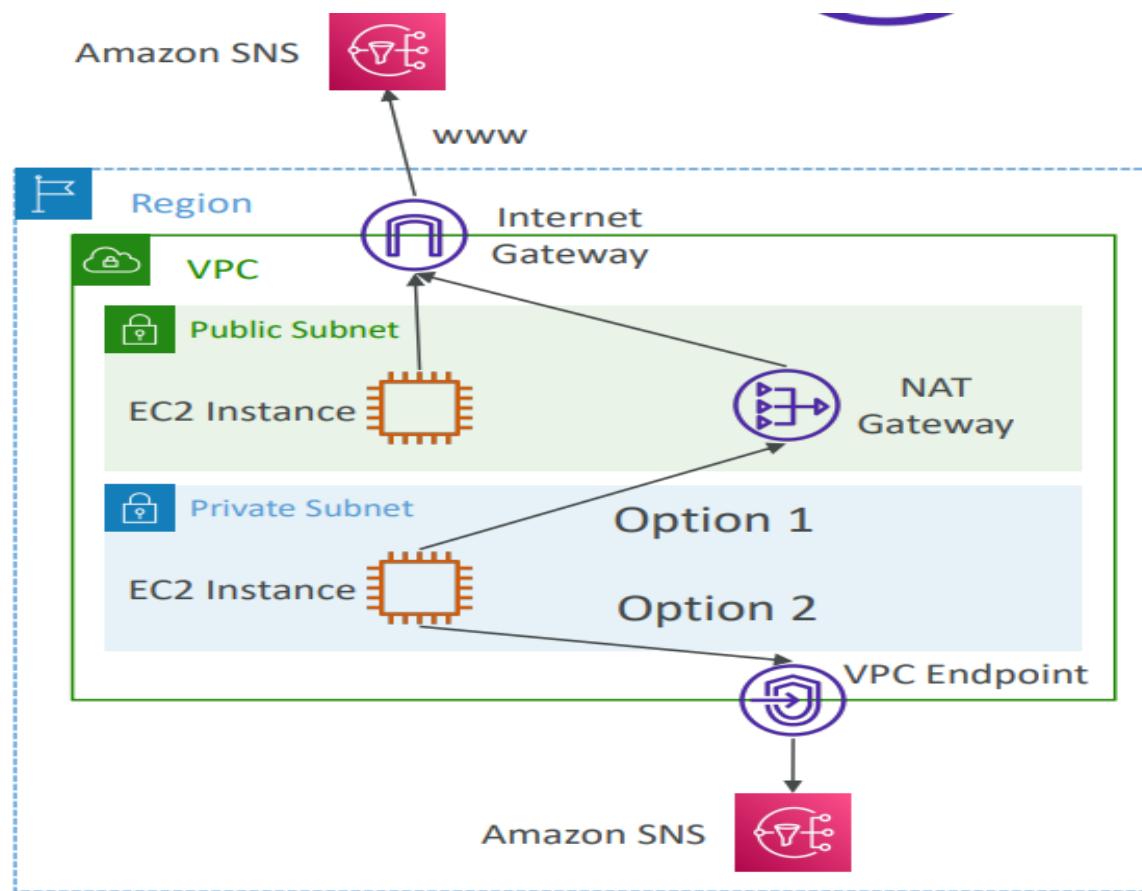
Hence

### Step 5

## VPC Endpoints

These are like metro connecting dots 😊

VPC Endpoints (powered by AWS PrivateLink) allows you to connect to AWS services using a private network instead of using the public Internet



## **Now this is What I meant**

These are also of two types one is a gateway endpoint supporting less services(FOR Small companies)

2. is Interface endpoints(Basically Endpoint premium subscription for Enterprises)

Finally.... So long but my VPC is all set !!!!

**A final step remains.....**

**Imagine a city without police or fire Department**

**A school without rules and regulations**

**A college Without 75% criteria(Definitely  
mine 😊)**

A VPC should also require to have a logging system to catch all the flow logs like how the traffic is flowing in and out and how is it behaving

# STEP-Final

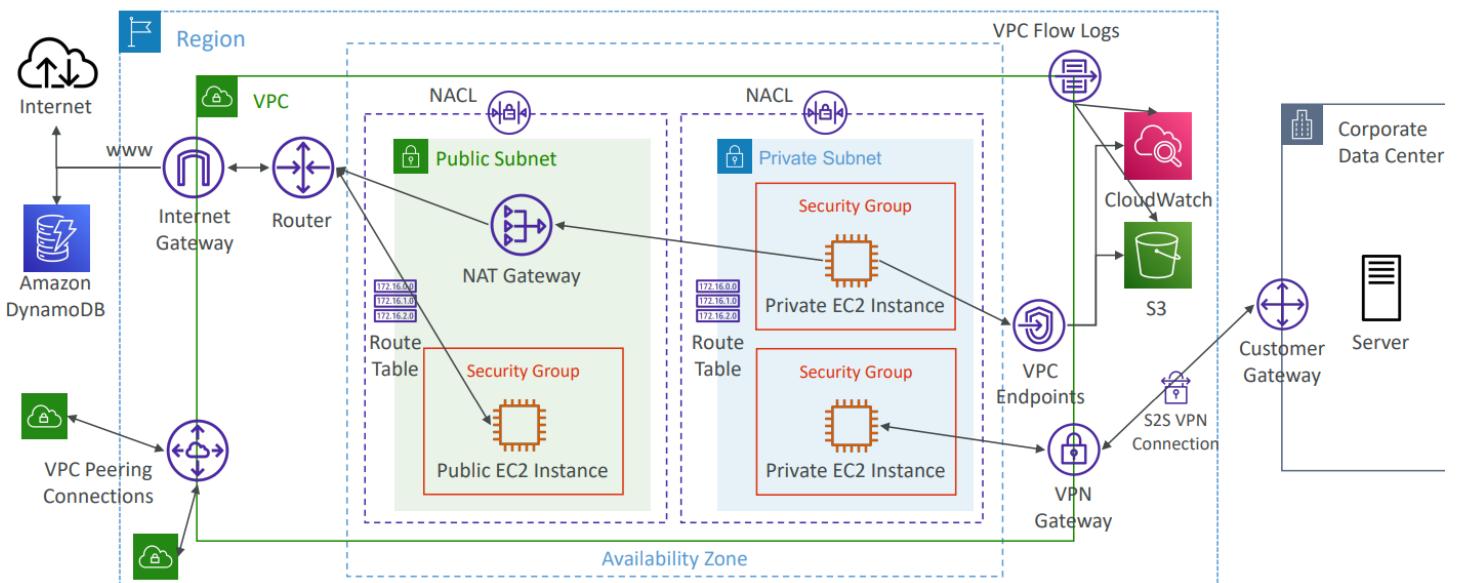
## VPC FlowLogs

Flow logs data can go to S3, CloudWatch Logs, and Kinesis Data Firehose

Captures network information from AWS managed interfaces too: ELB, RDS, ElastiCache, Redshift, WorkSpaces, NATGW, Transit Gateway...

Some important things while viewing the log data from **log groups**

- srcaddr & dstaddr – help identify problematic IP
- srcport & dstport – help identify problematic ports
- Action – success or failure of the request due to Security Group / NACL
- Can be used for analytics on usage patterns, or malicious behavior
- Query VPC flow logs using Athena on S3 or CloudWatch Logs Insights



Some of the Parts like the direct connection to Corporate data centers will be covered later in the documentation

I am still writing it.. Figuring out how to document

IF YOU HAVE MADE THIS FAR 

IF you liked my way of documenting things or would like to suggest changes feel free to message me and like the post

Hope you have understood my take of best practices in creating a vpc for an enterprise grade or a startup grade web app

**SEE YOU SOON**