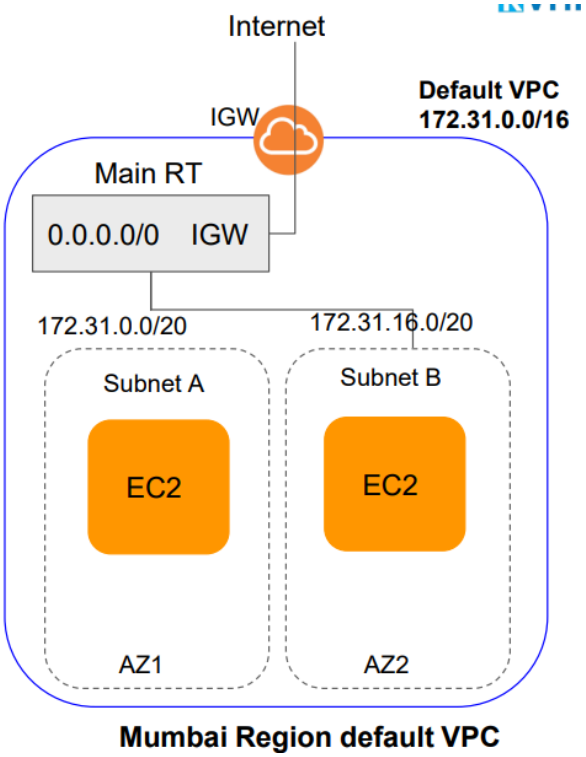
# LAB

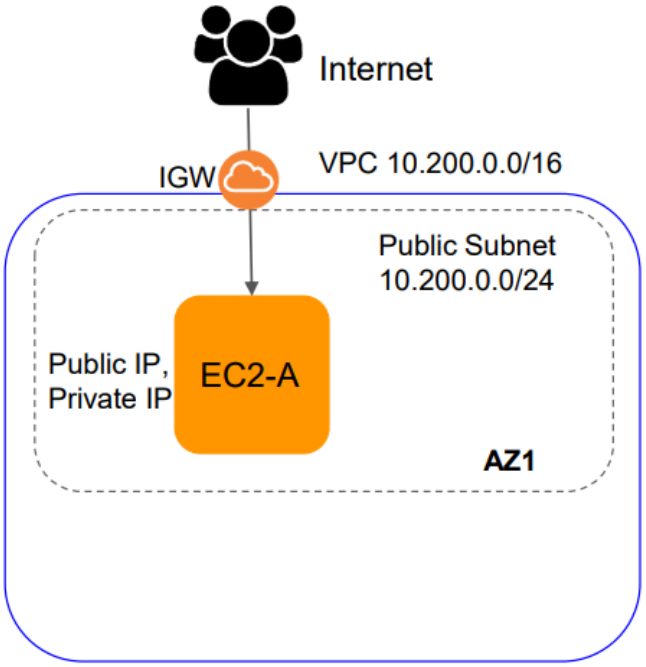
## Default VPC




- AWS Creates Default VPC in each AWS region
- Creates VPC with CIDR - 172.31.0.0/16
- Creates Subnets in every AZ with CIDR /20
- Creates Internet Gateway
- Main route table with route to Internet which make all subnets public
- If deleted, you can recreate default VPC by

  VPC Service -> Your VPCs -> Action -> Create Default VPC

## VPC with Single Public Subnet




Public Subnet Route Table

| Destination | Target |
|---|---|
| 10.200.0.0/16 | local |
| 0.0.0.0/0 | igw-xxx |

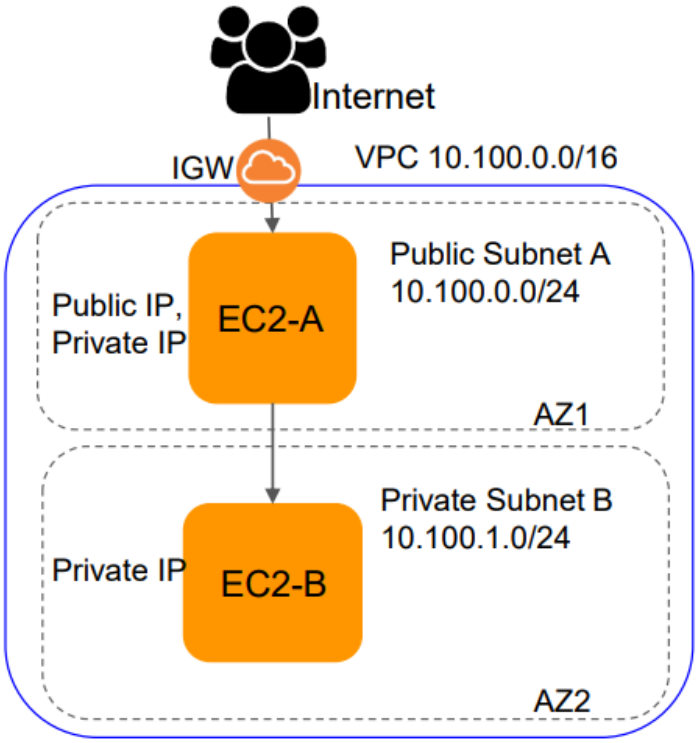
Note: For EC2 instance to be reachable from internet, it must be in Public Subnet and must have Public IP

**Steps:**

1. Create VPC a. Go to VPC service -> Your VPCs -> Create VPC (Name: MyVPC, CIDR: 10.100.0.0/16) -> Create
2. Create Internet Gateway
   Internet Gateways -> Create internet gateway
3. Attach Internet Gateway to VPC
   Select Internet gateway -> Actions -> Attach to VPC -> Select your VPC
4. Create Subnet
   A. Subnets -> Create subnet (Name: MyVPC-Public, VPC: MyVPC, AZ: Select first az - ap-south1a, CIDR: 10.100.0.0/24)
   B. Select Subnet -> Action -> Modify Auto Assign Public IP -> Enable -> Save
5. Create Route table
   A. Route Tables -> Create Route Table (Name: MyVPC-Public, VPC: MyVPC)
   B. Select Route table -> Routes -> Edit -> Add another route (Destination: 0.0.0.0/0, Target: Internet gateway -> igw-xxx) -> Save
6. Associate Route table with Subnet to make it Public subnet
   A. Select Route table -> Subnet Associations -> Edit -> Check the MyVPC-Public subnet -> Save

7. Launch EC2 instance in newly created Public Subnet
   A. Go to EC2 Service -> Instances
   B. Launch EC2 Instance -> Select Amazon Linux 2 -> Select t2.micro
   C. Configure Instance Details:
      i. Network: MyVPC
      ii. Subnet: MyVPC-Public (rest all defaults)
   D. Add storage (all defaults)
   E. Add Tags
      i. Key=Name, Value=EC2-A
   F. Configure Security Group
      i. Add rule for SSH port 22 for source as MyIP
   G. Review and Launch
   H. 8. Connect to EC2 instance (Public IP) from your laptop using Putty or terminal (ec2-user)

## VPC with Public and Private Subnet




Private Subnet Route Table

| Destination | Target |
|---|---|
| 10.100.0.0/16 | local |

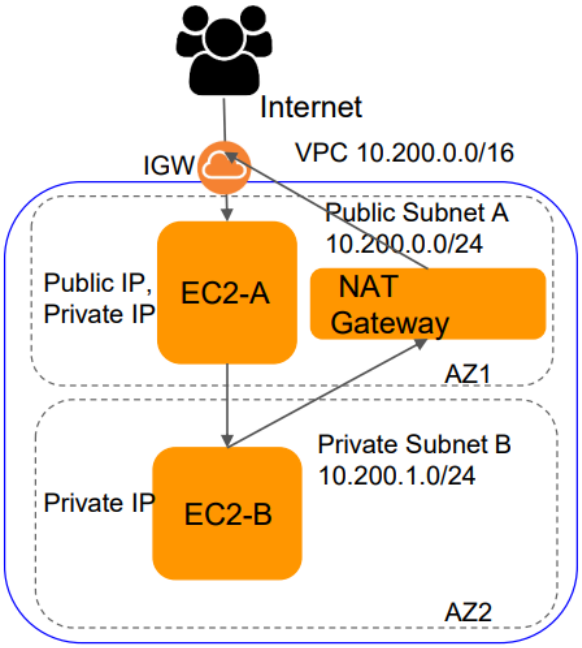
Continuing with previous setup

1. Create a Private subnet

A. Create subnet (Name: MyVPC-Private, VPC: MyVPC, AZ: Select different az (ap-south-1b), CIDR: 10.100.1.0/24)

2. Create Private route table a. Route Tables -> Create Route Table (Name: MyVPC-Private, VPC: MyVPC)
3. Associate Route table with Subnet to make it Private subnet a. Select Route table -> Subnet Associations -> Edit -> Check the MyVPC-Private subnet -> Save
4. Launch another EC2 instance in same VPC but in newly created Private subnet.
   A. Tag this instance with Name=EC2-B
   B. New security group
      - Add rule SSH for CIDR of Public Subnet source CIDR
      - Add rule All-ICMP IPv4 for Public Subnet source CIDR
5. Note down EC2-B private IP address
6. Try to ping EC2-B Private IP from EC2-A instance -> Should work
7. Try to connect to EC2-B instance from EC2-A (Permissions denied..Why?)
   A. $ssh ec2-user@10.100.1.x (Replace this ip with your EC2-B IP address)
8. Get your ssh .pem file on EC2-A instance
   A. Open local .pem file with nodepad and copy the content (CTRLA -> CTRL+C)
   B. On EC2 A terminal -> vi key.pem -> enter -> press i -> paste using right click -> esc -> :wq -> enter
   C. chmod 600 key.pem
   D. ssh -i key.pem ec2-user@10.100.1.x -> should be able to connect
9. Try to ping google.com from EC2-B instance
   A. ping google.com (You should not be able to ping. Why?)

## VPC with Public, Private Subnet and NAT

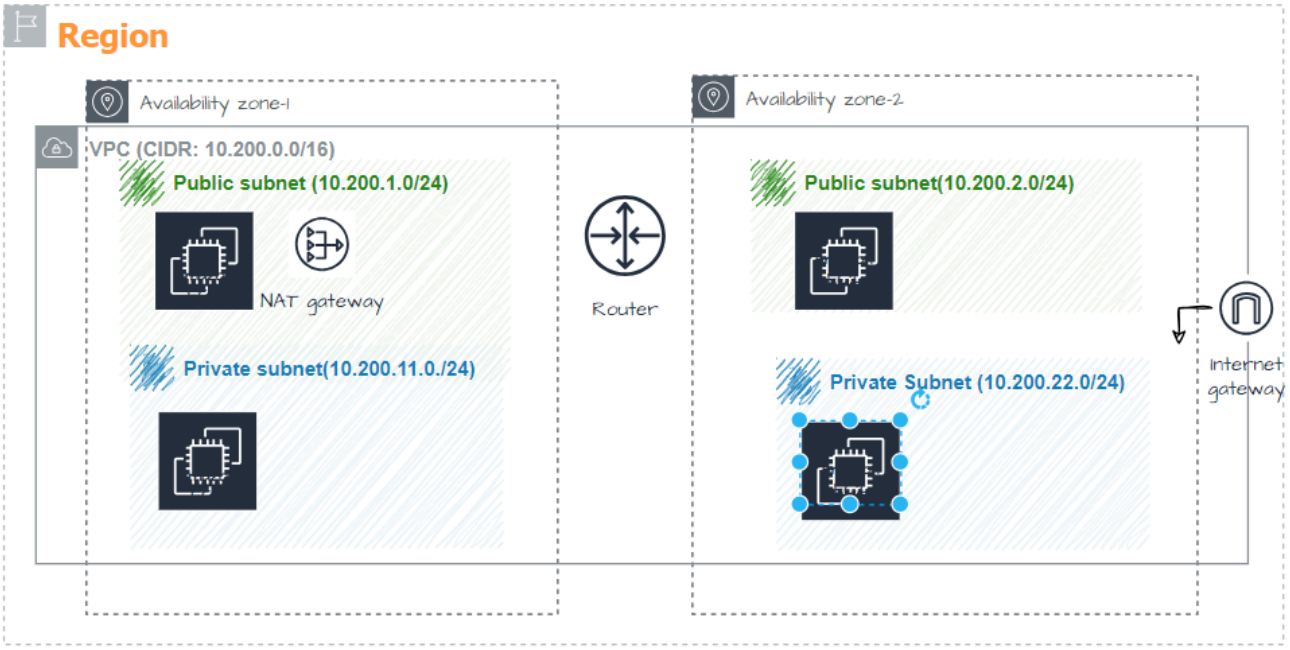



Private Subnet Route Table

| Destination | Target |
|---|---|
| 10.200.0.0/16 | local |
| 0.0.0.0/0 | NAT-Gateway-ID |

Continuing with previous setup

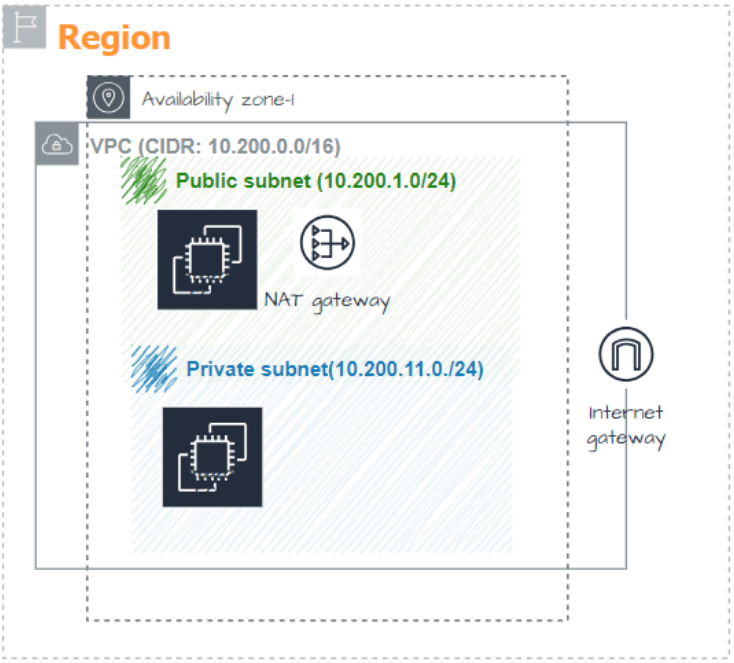
1. Create a NAT gateway in your VPC
   - VPC -> NAT Gateways -> Create NAT Gateway
     - Subnet: MyVPC-Public (Must select Public Subnet)
     - EIP: Create New EIP
     - Create NAT Gateway
     - It takes 5-10 minutes for NAT gateway to be Active
2. Add a route in Private subnet for internet traffic and route through NAT Gateway
   - Route Tables -> Select MyVPC-Private route table - Routes -> Edit -> Add another route
     - Destination: 0.0.0.0/0
     - Target: nat-gateway
     - Save
3. Now again try to ping google.com from EC2-B
   - ping google.com




| Public Route Table | |
|---|---|
| Destination | Route |
| 10.0.0.0/16 | Local |
| 0.0.0.0/0 | IG |

| Private Route Table | |
|---|---|
| Destination | Route |
| 10.0.0.0/16 | Local |
| 0.0.0.0/0 | NAT+GW |

# Task 1: Create VPC Network Single Availability Zone




| Public Route Table | |
|---|---|
| Destination | Route |
| 10.2000.0.0/16 | Local |
| 0.0.0.0/0 | IG |

| Private Route Table | |
|---|---|
| Destination | Route |
| 10.200.0.0/16 | Local |
| 0.0.0.0/0 | NAT+GW |

## Step 1: Create VPC

Go to the VPC Dashboard —----> Your VPC —--> Create VPC

## VPC settings

Resources to create Info

Create only the VPC resource or the VPC and other networking resources.

- (•) VPC only
- ( ) VPC and more

Name tag - *optional*

Creates a tag with a key of 'Name' and a value that you specify.

HCL-VPC

IPv4 CIDR block Info

- (•) IPv4 CIDR manual input
- ( ) IPAM-allocated IPv4 CIDR block

IPv4 CIDR

10.200.0.0/16

IPv6 CIDR block Info

- (•) No IPv6 CIDR block
- ( ) IPAM-allocated IPv6 CIDR block
- ( ) Amazon-provided IPv6 CIDR block
- ( ) IPv6 CIDR owned by me

Tenancy Info

Default

## Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

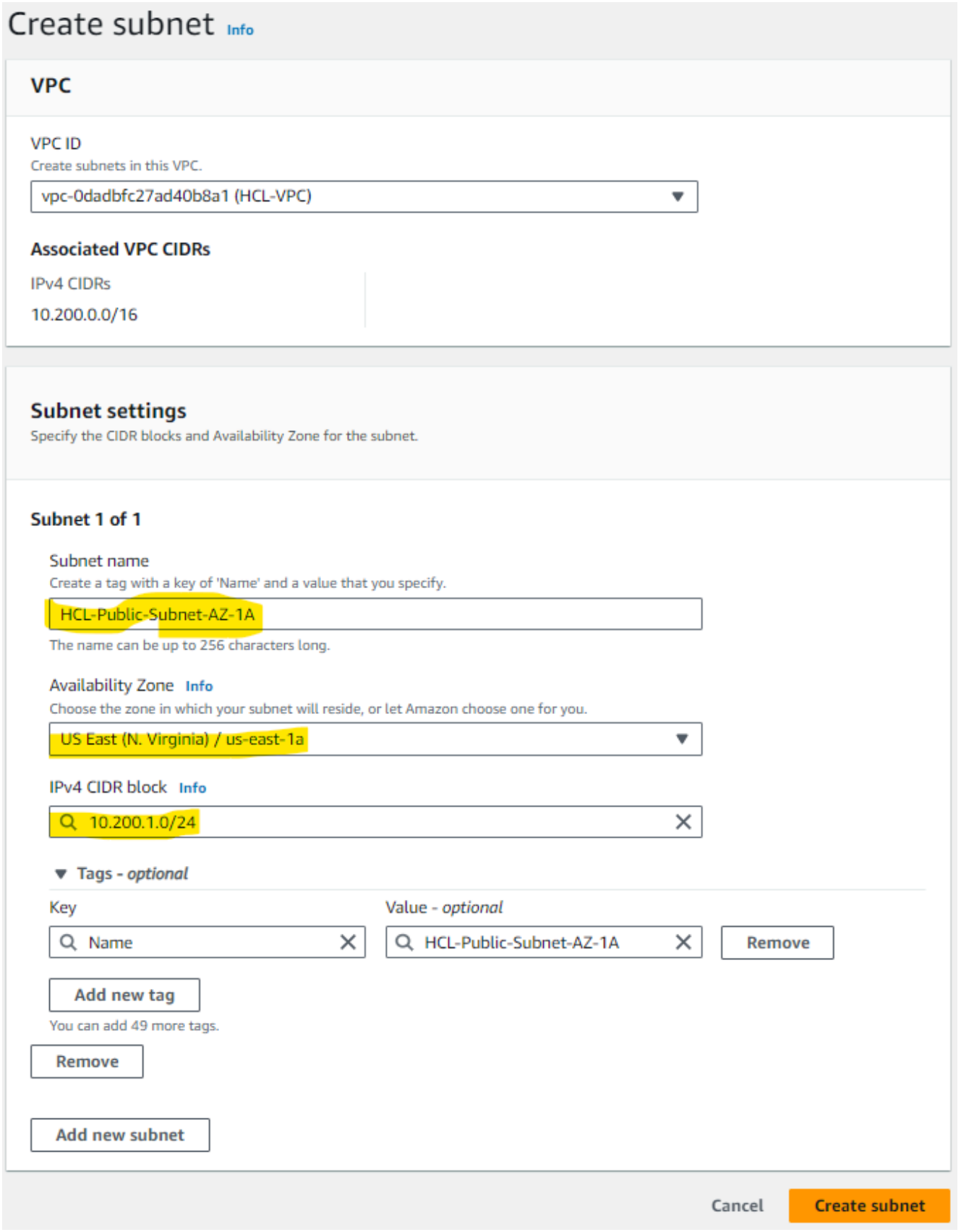
| Key | Value - *optional* | |
|---|---|---|
| Name | HCL-VPC | Remove tag |

Add tag

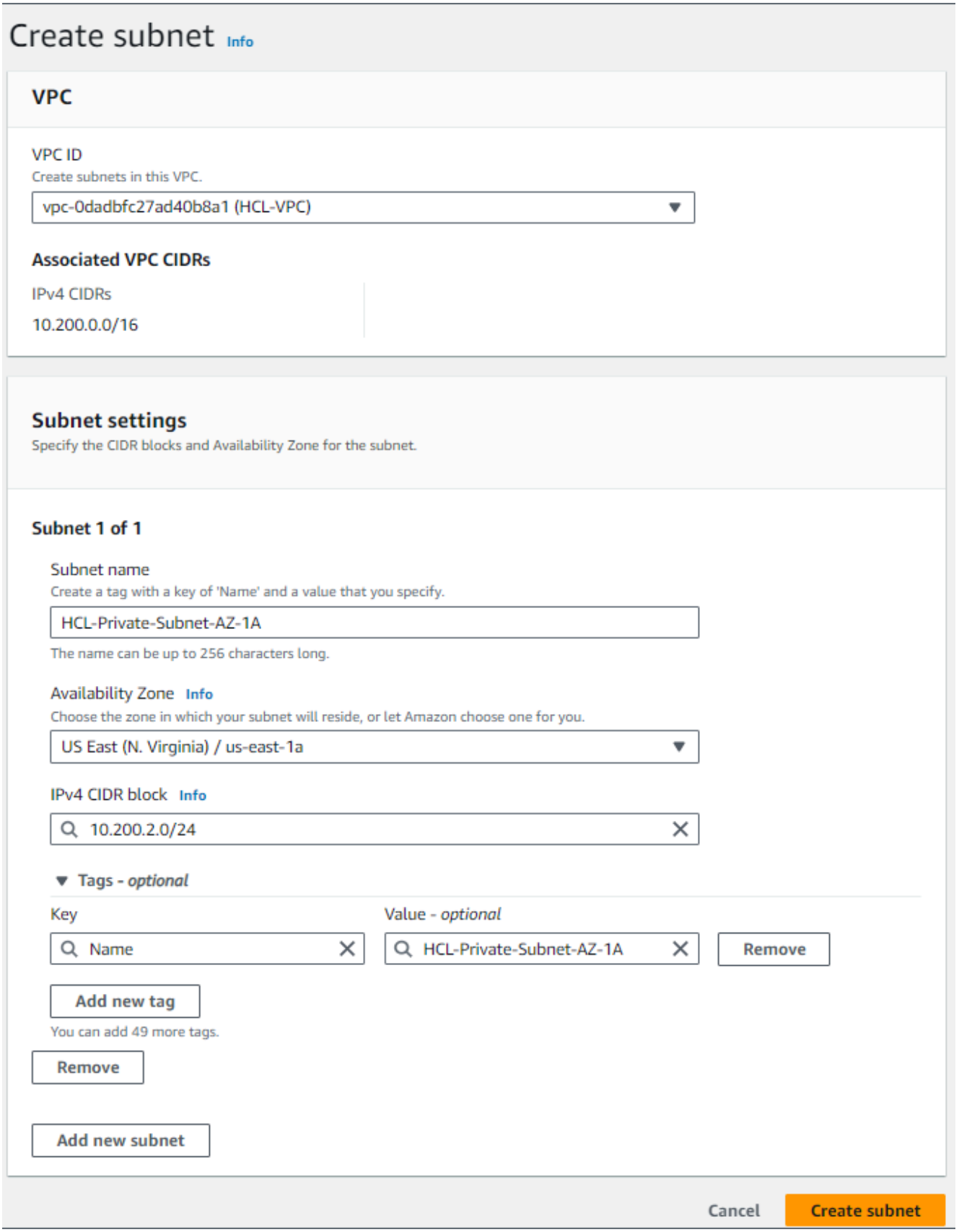
You can add 49 more tags

Cancel Create VPC

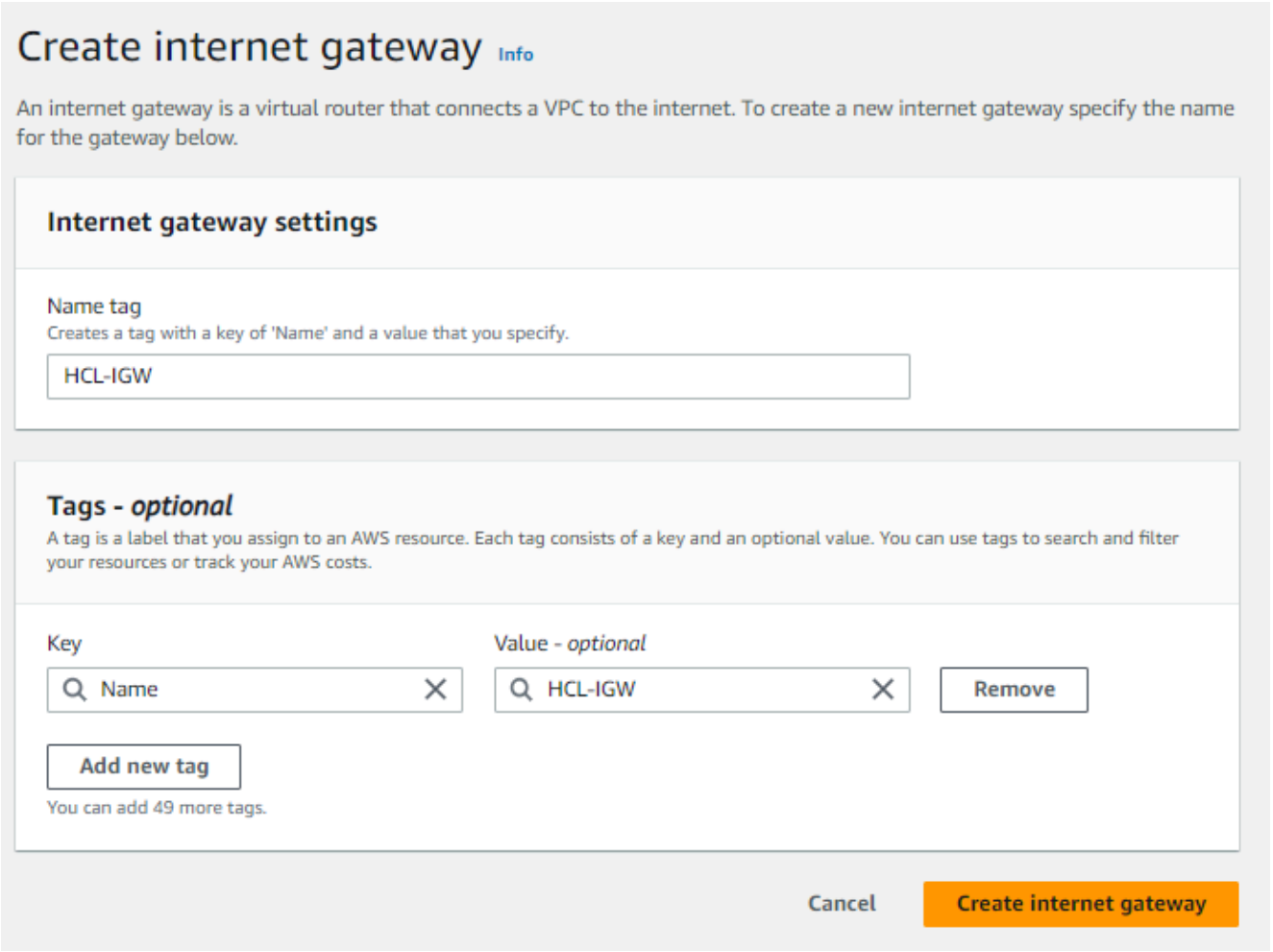
# Step 2: Create Public Subnets



# Step 3: Create Private Subnets



## Step 4: Create Internet Gateway




**Then attached to VCP**

## Step 5: Create Public Route Table

# Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

## Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

HCL-Public-Route-Table-AZ1

VPC

The VPC to use for this route table.

vpc-0dadbfc27ad40b8a1 (HCL-VPC)

## Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

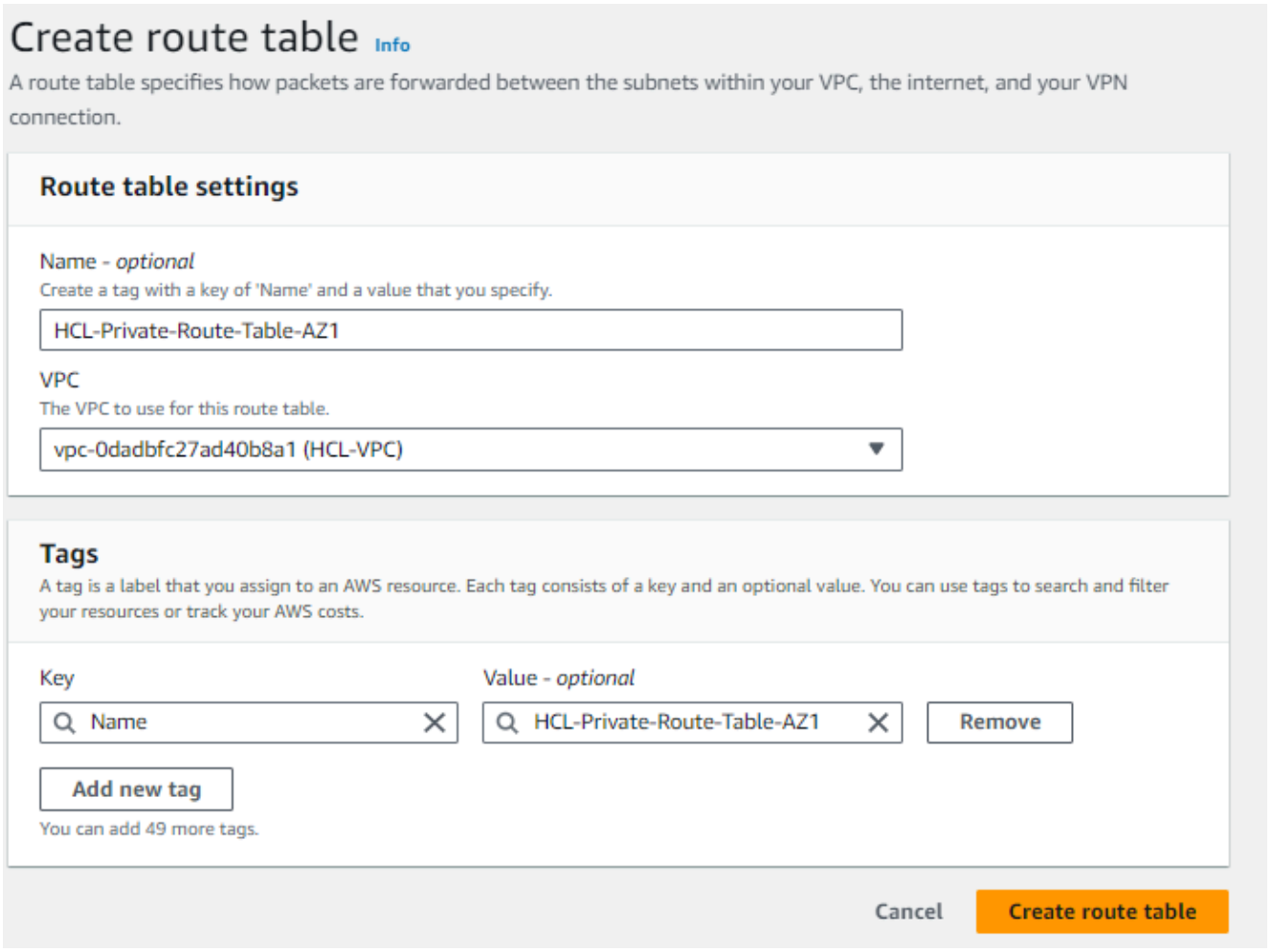
| Key | Value - *optional* | |
|---|---|---|
| Name | HCL-Public-Route-Table-AZ1 | Remove |

Add new tag

You can add 49 more tags.

Cancel | Create route table

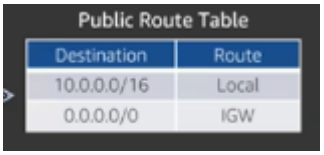
## Step 6: Create Private Route Table

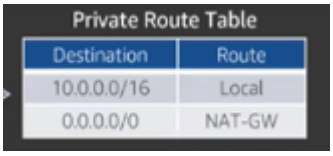



## Step 7: Add IGW Route with public Route table

| Destination | Target |
|---|---|
| 0.0.0.0/0 | igw-0d2962943f4d6748e |
| 10.200.0.0/16 | local |

Next Create Private Route table

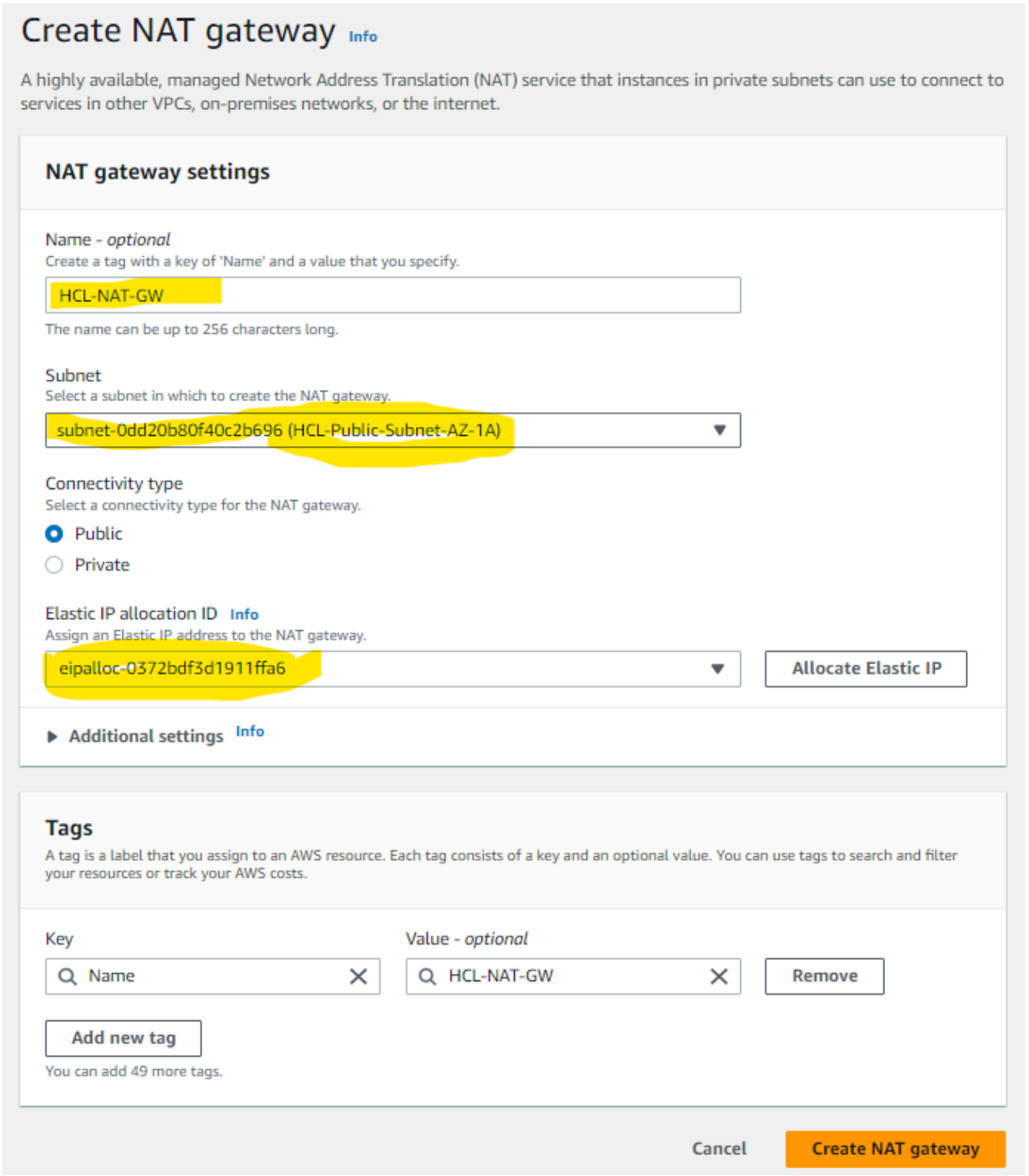




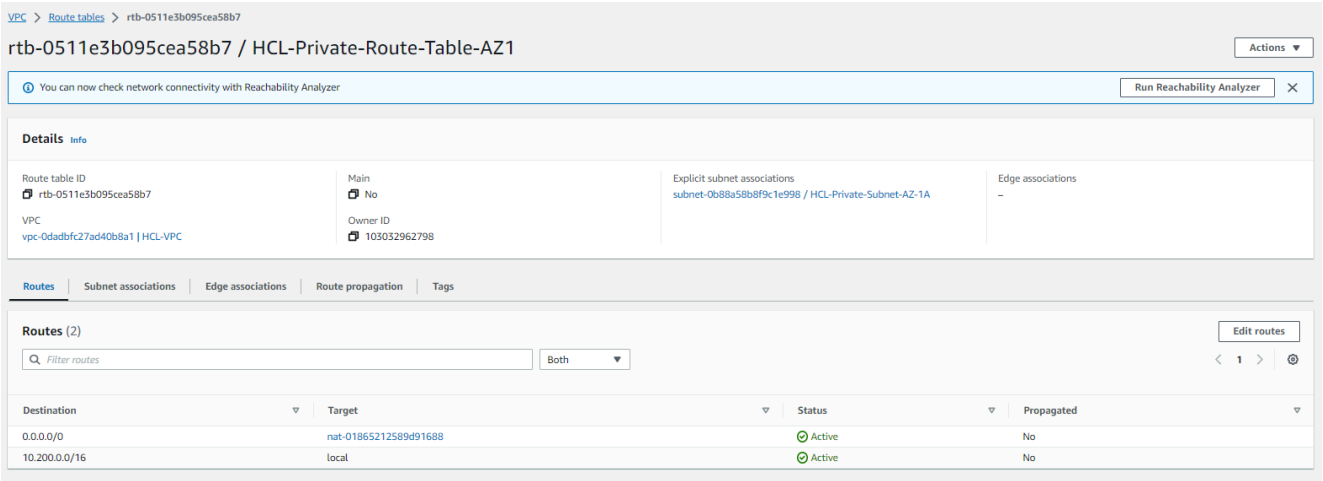

## Step 8: Subnet Association

- Private route table must be associate with Private Subnet
- Public route table must be associate with Public Subnet

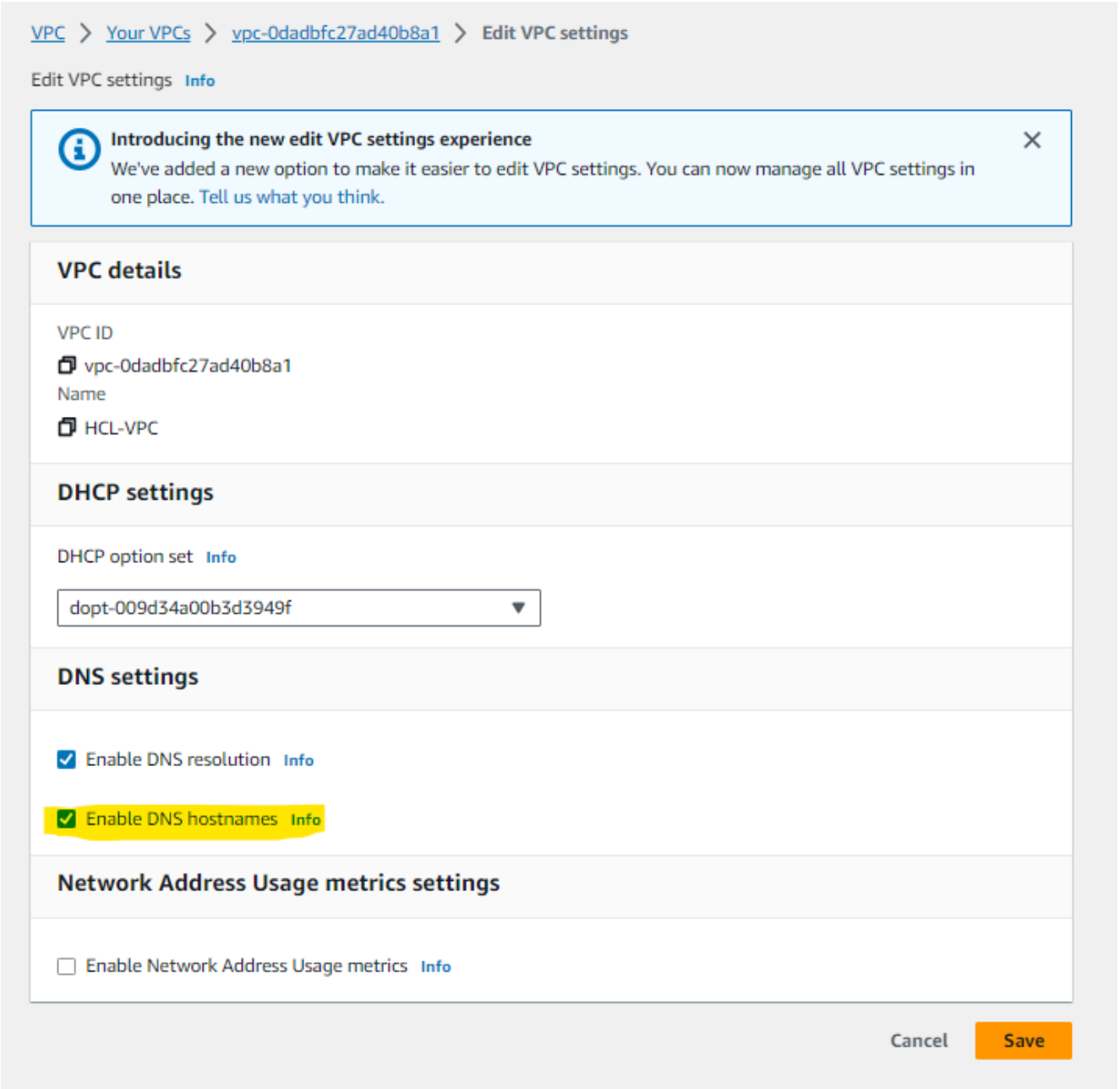
## Step 9: Create NAT Gateway




Step 10: Add NAT-GW Route with Private Route table


## Step 11: **Enable DNS host name in VPC**

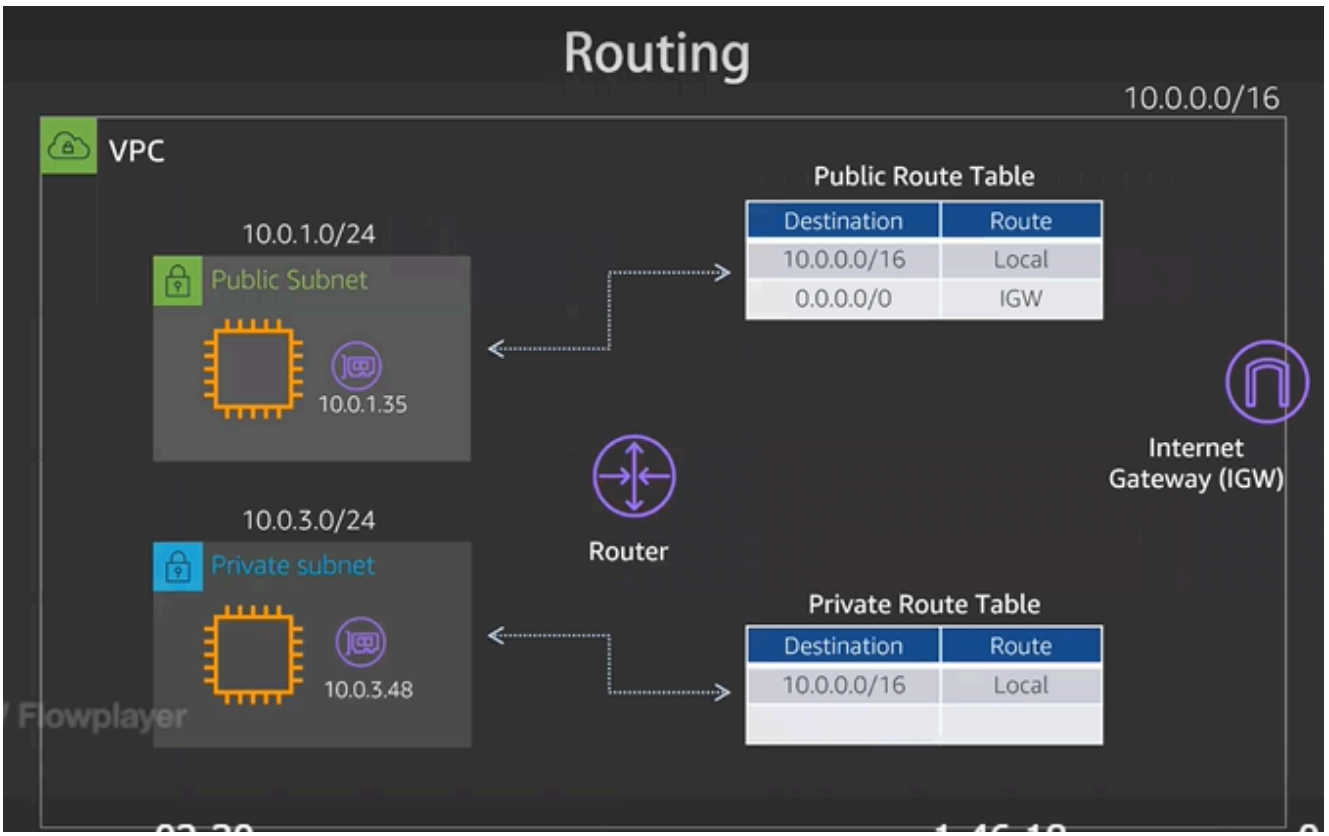


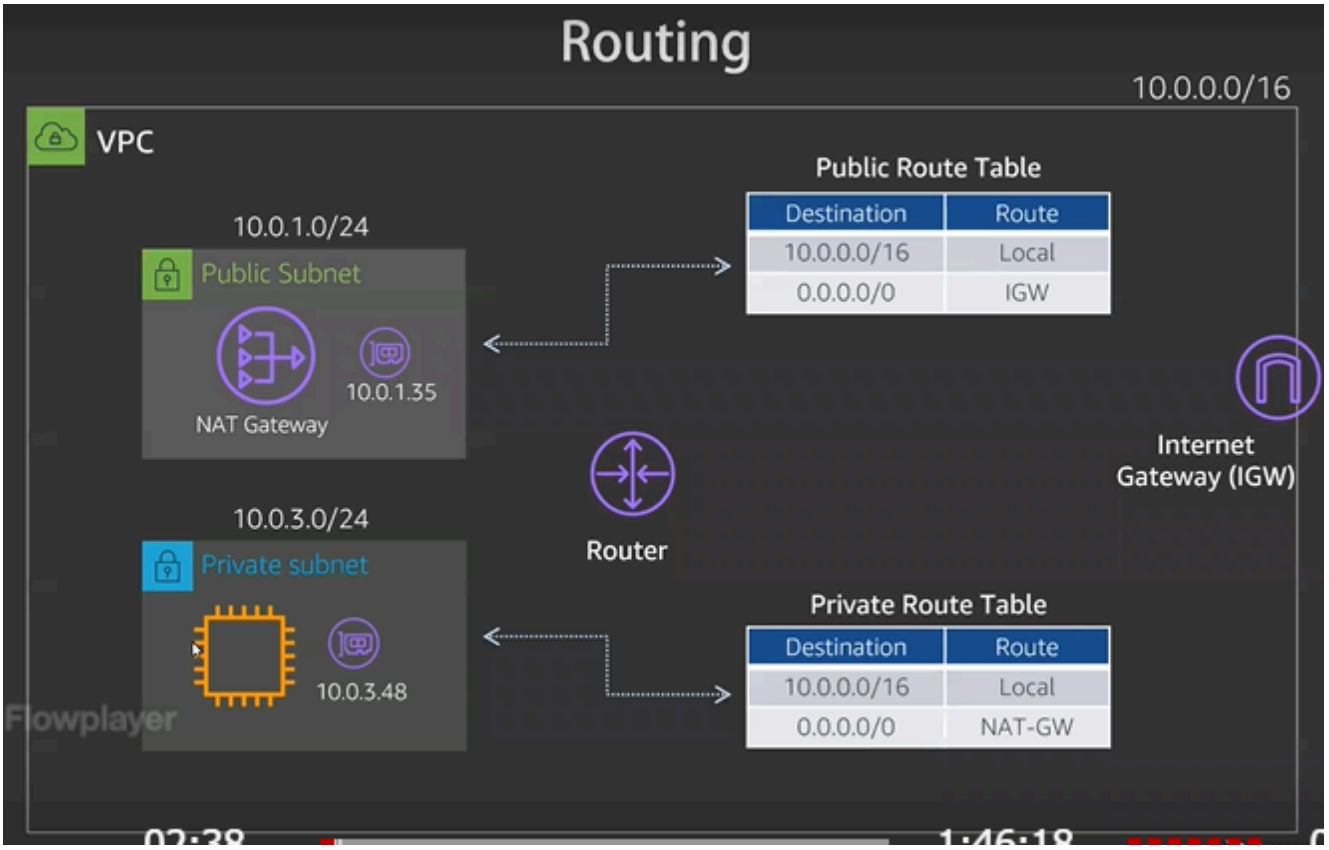

Step 12: Create Security group

# Task 2: Create VPC Network Two Availability Zone

**We are going to create a VPC with two Availability zones in the same region. So here : One VPC and 2 Public Subnet and 2 Private Subnet in two Availability Zone**


10.0.0.0/16
VPC
Availability Zone 1
Public subnet
10.0.1.0/24
Private subnet
10.0.2.0/24
Availability Zone 2
Public subnet
10.0.21.0/24
Private subnet
10.0.22.0/24

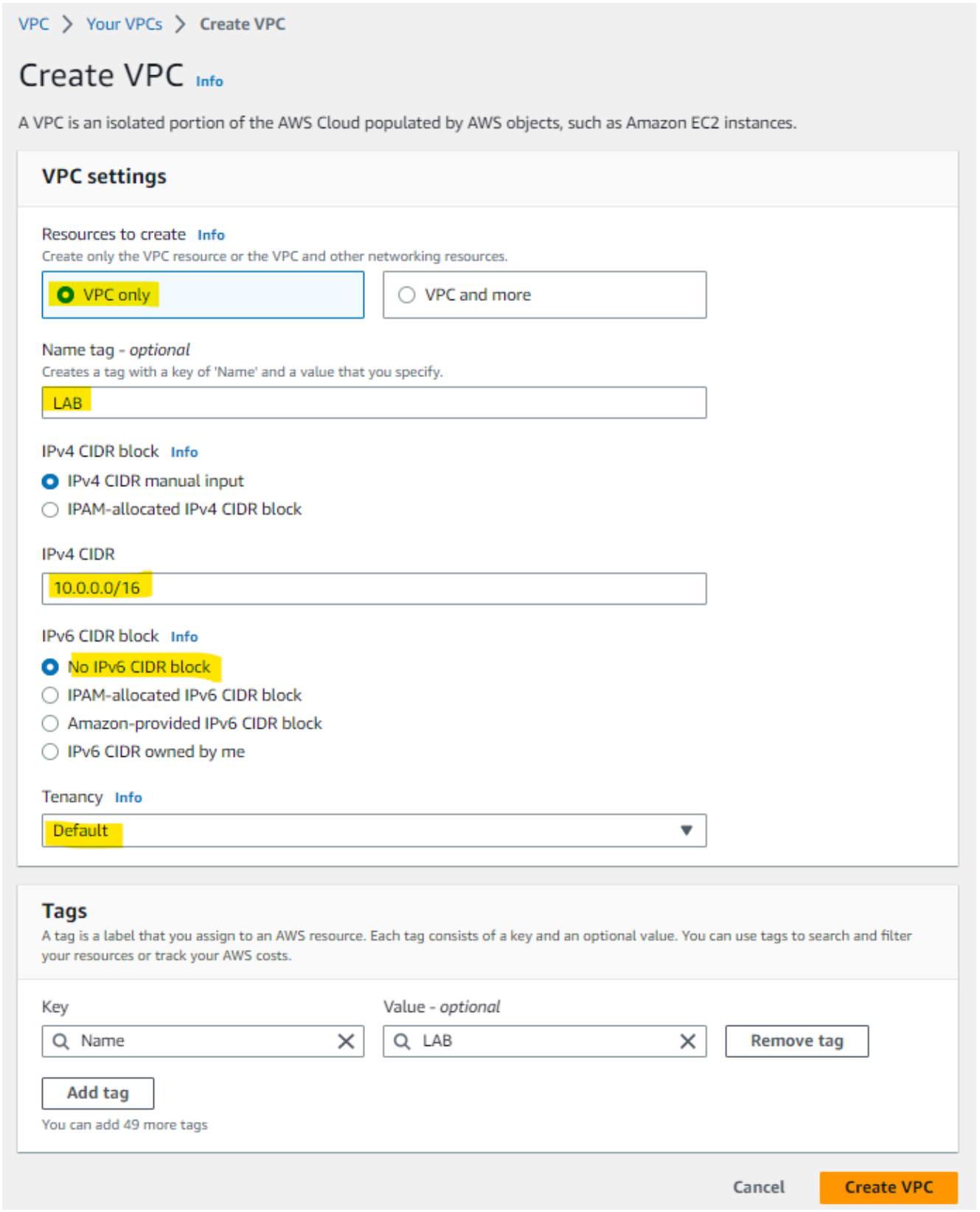

Routing
10.0.0.0/16
VPC
Public Route Table
Destination
Route
10.0.0.0/16
Local
0.0.0.0/0
IGW
10.0.1.0/24
Public Subnet
10.0.1.35
Internet Gateway (IGW)
Router
10.0.3.0/24
Private subnet
10.0.3.48
Private Route Table
Destination
Route
10.0.0.0/16
Local




## Step 1 : Create VPC

**Important:**

- There is always a default VPC in all regions inside my AWS account. So never delete or update this default VCP




**Important: After VPC Creation , you will get a default route table and security group for newly created VPC**

## Step 2: Create Subnets

1-public and 1 private subnet for single AZ as per billow

# Create subnet Info

## VPC

VPC ID
Create subnets in this VPC.

vpc-05ac025921c474329 (Indian-VPN)

### Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Indian-Publicc-Subnet

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info

10.0.1.0/24

▼ Tags - *optional*

| Key | Value - *optional* | |
|---|---|---|
| Name | Indian-Publicc-Subnet | Remove |

Add new tag

You can add 49 more tags.

Remove

Add new subnet

Cancel Create subnet

# Create subnet Info

## VPC

VPC ID
Create subnets in this VPC.

vpc-05ac025921c474329 (Indian-VPN)

### Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

## Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

### Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Indian-Private-Subnet

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info

10.0.2.0/24

▼ Tags - *optional*

| Key | Value - *optional* | |
|---|---|---|
| Name | Indian-Private-Subnet | Remove |

Add new tag

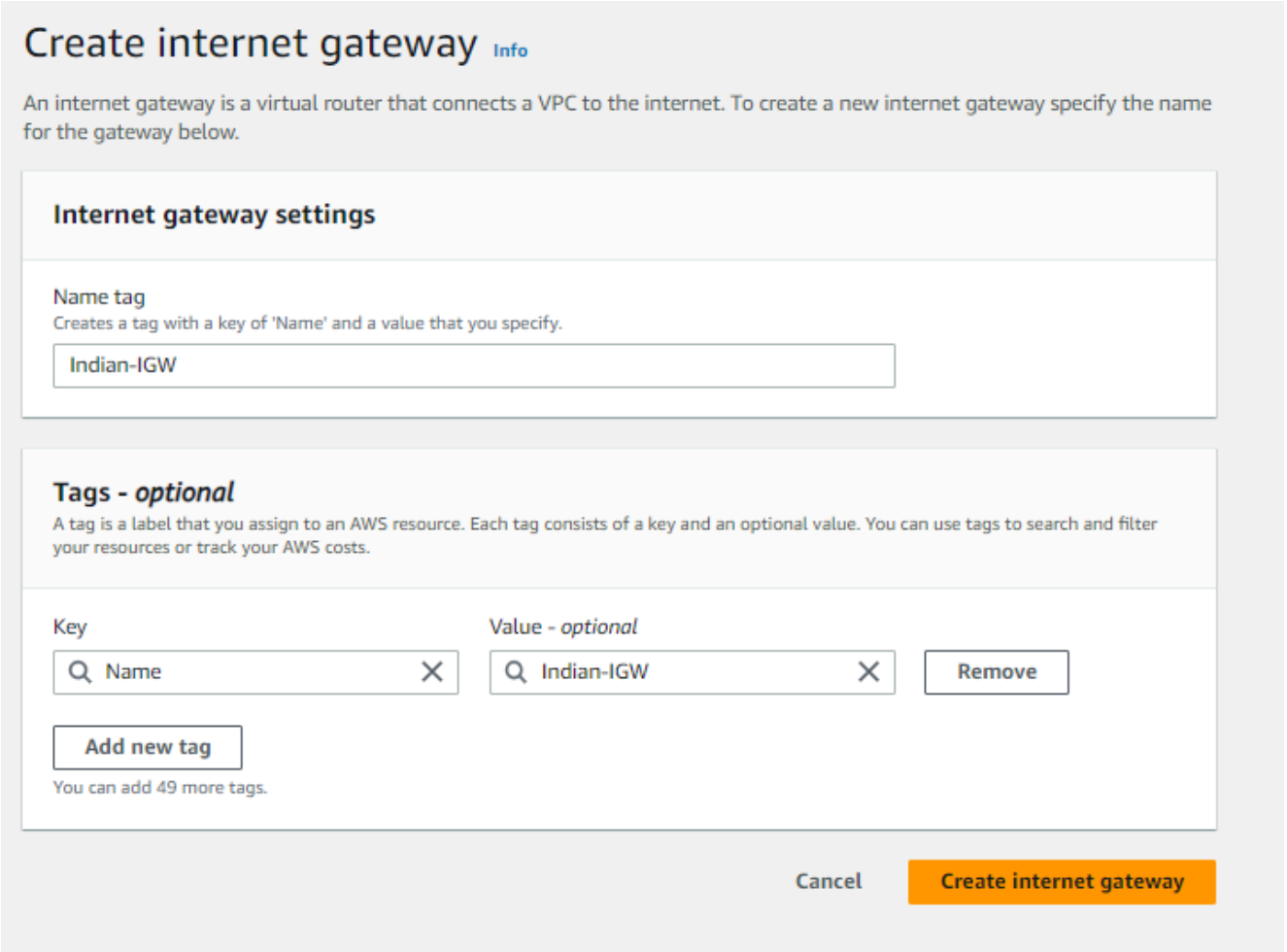
You can add 49 more tags.

Remove

Add new subnet

Cancel Create subnet

## Step 3: Create internet gateway




**Important : Then attached the IGW with your VPC**

## Step 4: Create Route Table

Create Public and private route table

# Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

## Route table settings

Name - *optional*

Create a tag with a key of 'Name' and a value that you specify.

Indian-Public-RT

VPC

The VPC to use for this route table.

vpc-05ac025921c474329 (Indian-VPN)

## Tags

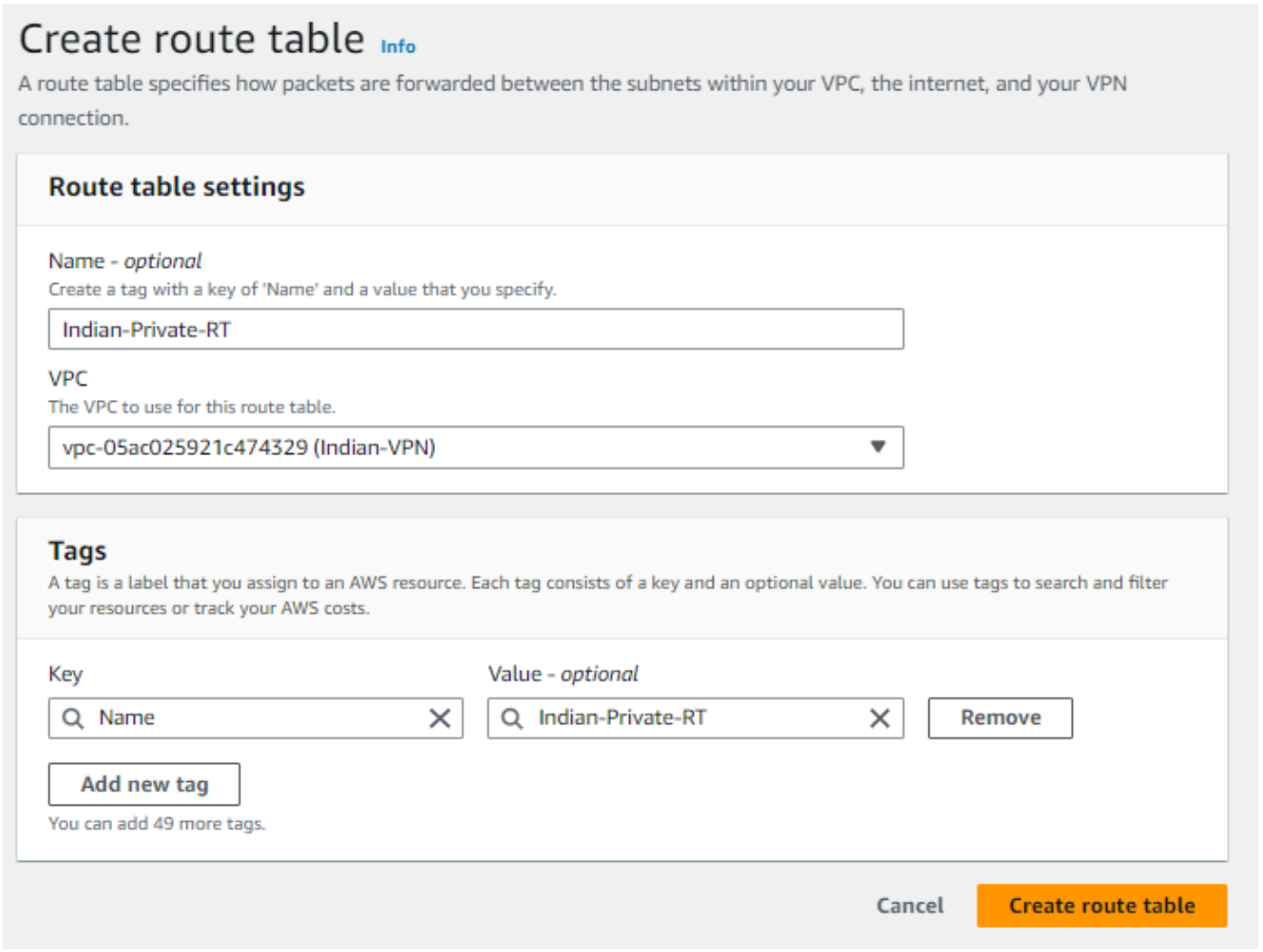
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

| Key | Value - *optional* | |
|---|---|---|
| Name | Indian-Public-RT | Remove |

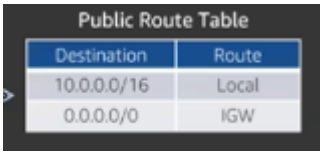
Add new tag

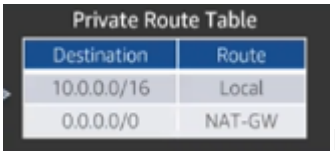
You can add 49 more tags.

Cancel Create route table




Add below entry in Public AND PRIVATE ROUTE TABLE **Then add routes IGW with Public Route table AND subnet associate**



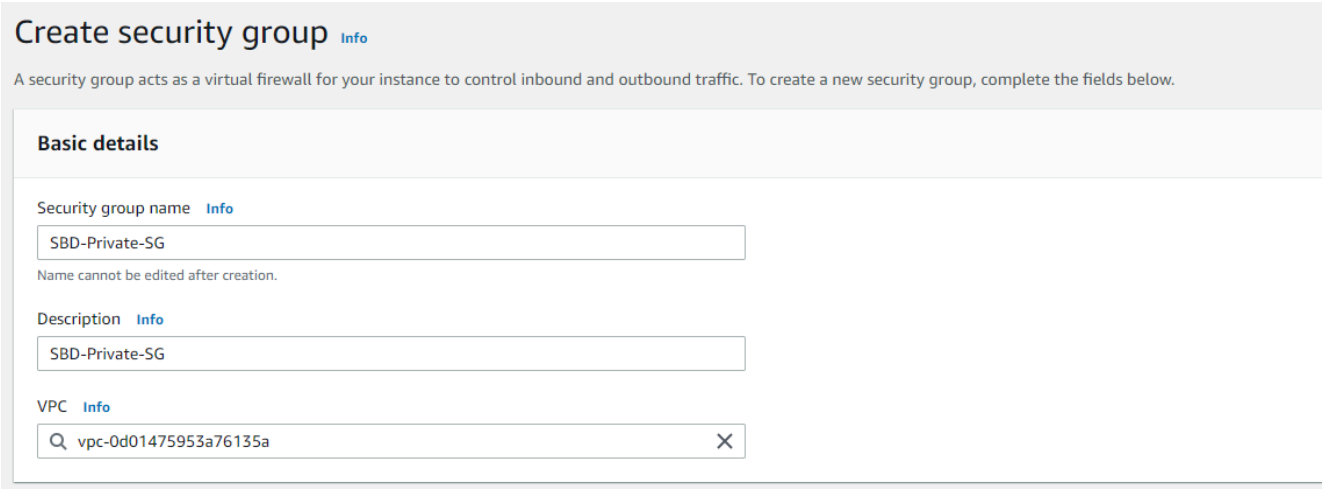



PENDING

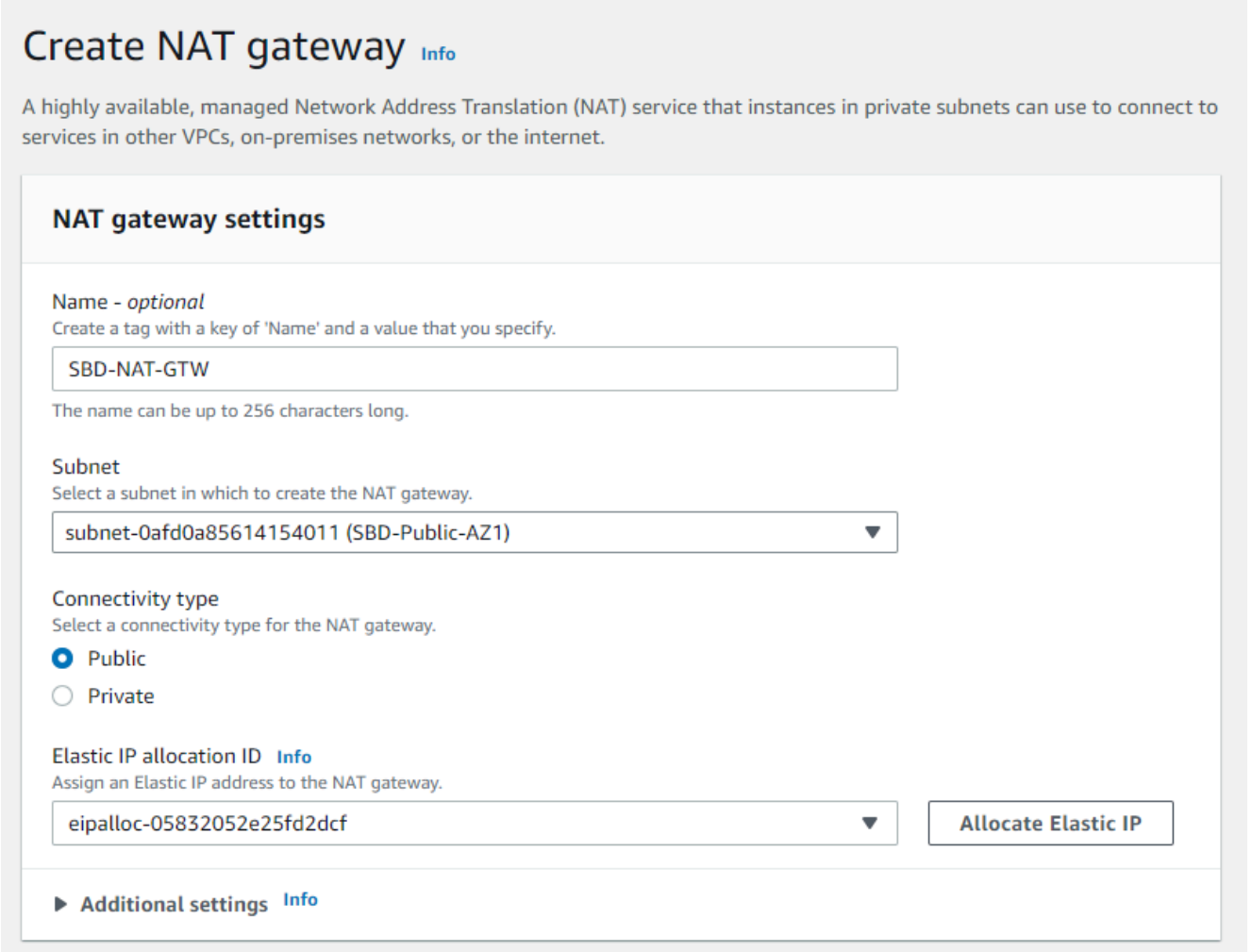
**Enable DNS host name in VPC**

## Step 5: Create Security group for Private and Public both

**Rules will be defined later**




## Step 6: Create NAt Gateway



Step 7: Finally Go to the route table of private subnet and add NET gateway

# Best practice for creating VPC on AWS Cloud.

Creating a Virtual Private Cloud (VPC) in AWS is a fundamental step in setting up your cloud infrastructure. A well-designed VPC is crucial for security, scalability, and resource isolation.

Here are best practices for creating a VPC on AWS:

**1. Plan Your VPC Layout:**

- Define your IP address range (CIDR block) carefully. Consider your current and future needs, but avoid making it too large, as AWS charges for unused IP addresses.
- Separate your VPC into different subnets (public, private, and possibly DMZ) based on the resources' accessibility and security requirements.

**2. Use Multiple Availability Zones (AZs):**

- Distribute your resources across multiple availability zones to ensure high availability and fault tolerance.

**3. Design for Security:**

- Use Network Access Control Lists (NACLs) and Security Groups to control inbound and outbound traffic.
- Isolate public and private resources into separate subnets to minimize exposure.
- Use a bastion host or VPN for secure remote access to private instances.

**4. Use Private Subnets for Databases:**

- Place databases and sensitive services in private subnets with no direct internet access. Access them through a secure application layer.

**5. Implement Proper Routing:**

- Set up route tables to control traffic between subnets and the internet.
- Use a Network Address Translation (NAT) gateway or NAT instance in public subnets to allow private instances to access the internet for updates, etc.

**6. Tag Resources:**

- Tag your resources with meaningful labels to help with cost allocation and resource management.

**7. Plan for Future Growth:**

- Leave room for expansion in your IP address space and subnets.
- Consider future requirements for new services or regions.

**8. Use VPC Peering or Transit Gateway:**

- If you have multiple VPCs, use VPC peering or AWS Transit Gateway to connect them securely.

**9. Monitor and Log:**

- Implement CloudWatch Logs and CloudTrail for monitoring and auditing.
- Use VPC Flow Logs for network traffic analysis.

**11. Documentation:**

- Document your VPC design, including CIDR blocks, subnet layouts, security group rules, and routing tables.

Remember that the specific requirements of your organization and applications will influence your VPC design. It's essential to have a clear understanding of your use case and security requirements when creating your VPC. Additionally, staying up-to-date with AWS best practices and security updates is crucial for maintaining a secure and efficient VPC.