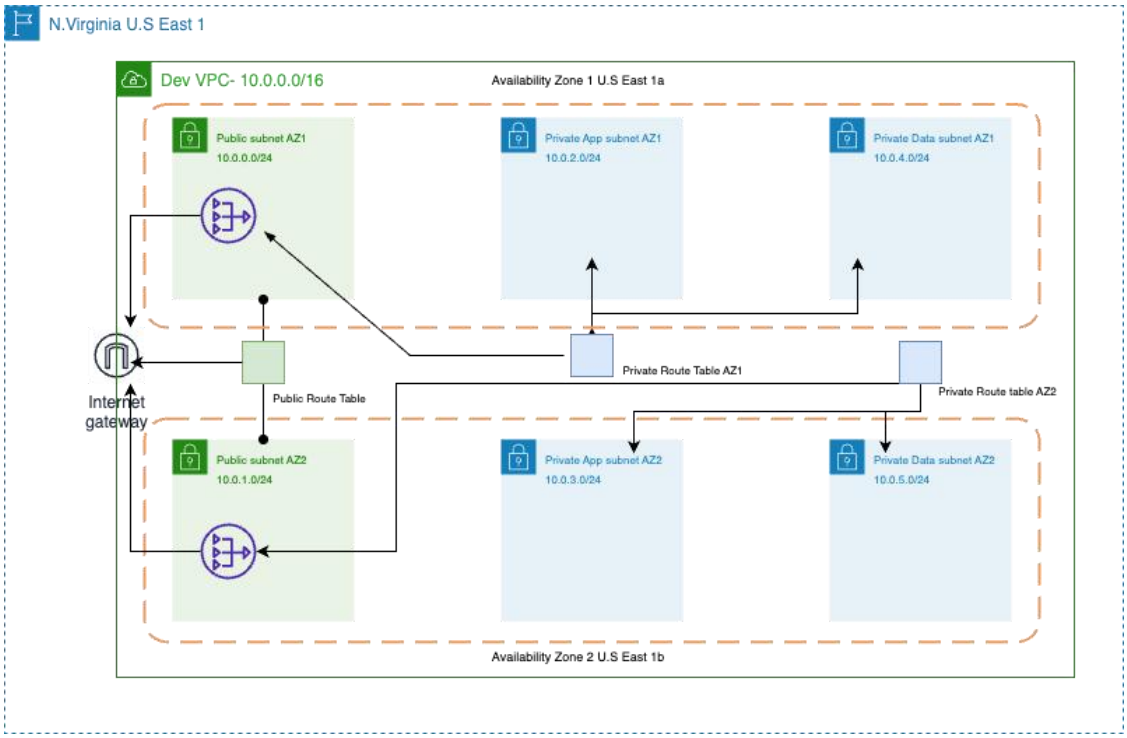
How To Build a Three Tier AWS Network VPC

The objective of building a three-tier AWS network VPC is to create a scalable environment that can meet the specific requirements of the application or system being developed, while ensuring the highest levels of security, availability, and performance. The three tiers typically consist of the following:

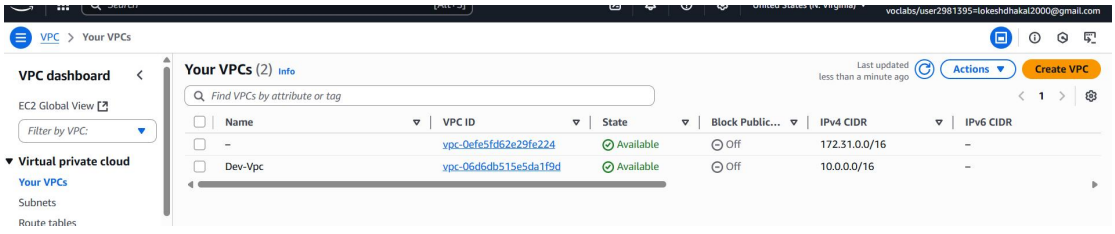
1. Presentation tier or web tier: This tier handles user interface functions and serves as the entry point for the application. It can include web servers, load balancers, and content delivery networks (CDNs).
2. Application tier or middle tier: This tier handles the application logic and can include application servers and databases.
3. Data tier or back-end tier: This tier manages data storage and processing and can include databases and data warehouses.



From the above architecture, the infrastructure is divided into three tiers. The first tier includes the public subnet which holds resources such as the Nat Gateway and Bastion Host. The second tier includes the private subnet which holds our web servers i.e our EC2 instances. Lastly, the third tier, which includes another private subnet which holds our databases.

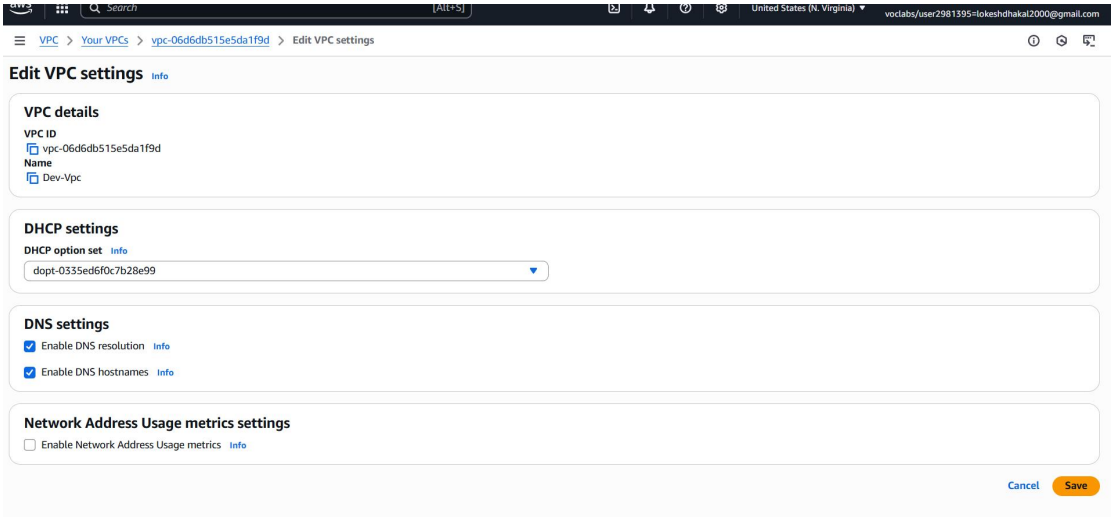
For the purpose of high availability and fault tolerance, we will duplicate the subnets across another availability zone. We will also create an internet gateway to allow the resources in the VPC to have access to the internet. To create the VPC, we'll move to the management console.

1. Select the region where the VPC will be created. Here we'll be using the N. Virginia (U.S East 1) region.
2. On the search box, type "VPC" and select VPC under the "services" drop down.
3. In your VPC dashboard, select VPC then click on "create VPC".
4. Give your VPC a name. Here i'll be using "Dev VPC" as my VPC name.
5. Under the IPv4 CIDR Block, leave the default setting "IPv4 CIDR manual input", then enter the CIDR block (10.0.0.0/16) referencing the architecture above.
6. Leave the other settings as they are (default), then scroll down and click on "create VPC".



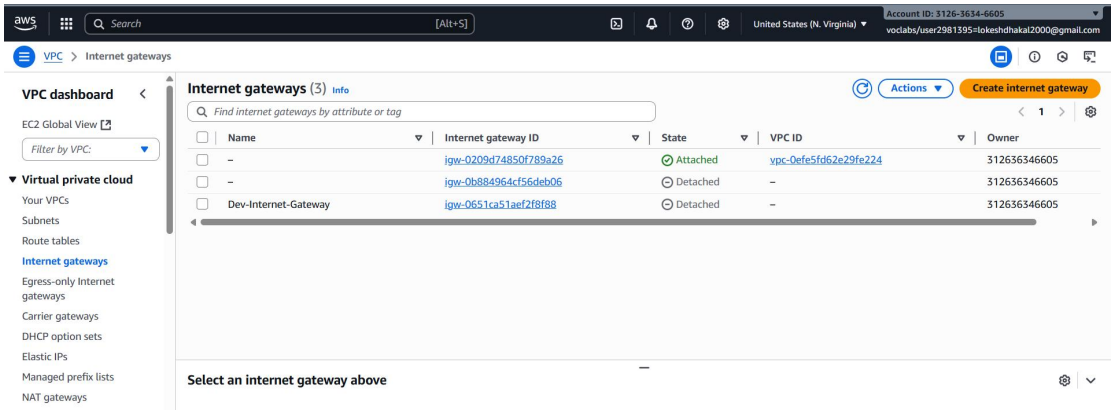
The next step is to enable DNS Hostname in the VPC we just created. To do that;

1. In the VPC we just created, select “Actions” on the top right corner
2. Select “Edit VPC settings”
3. Check the box which enables DNS Hostnames, then click on “Save Changes”



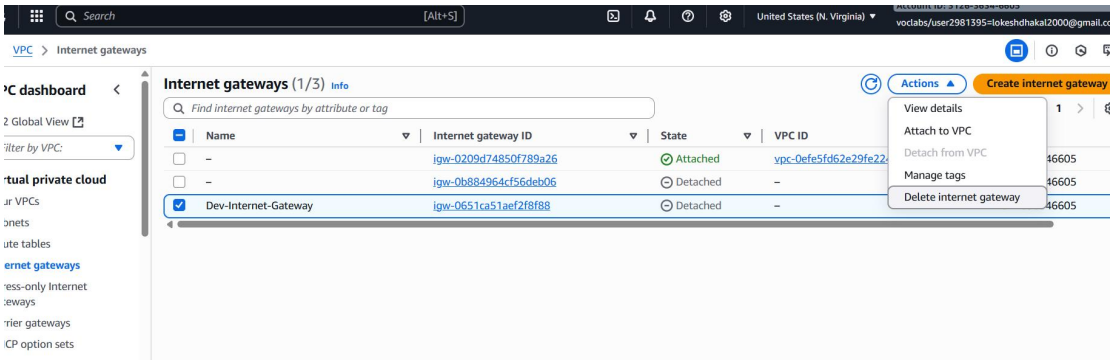
The next step is to create an Internet Gateway for our VPC and to do this;

1. On the left side where the you have the VPC dashboard drop down menu, select “Internet Gateway”
2. Then click on “Create Internet Gateway”
3. Give your Internet Gateway a name. In this case, the name is “Dev Internet Gateway”
4. Click on “create Internet Gateway”



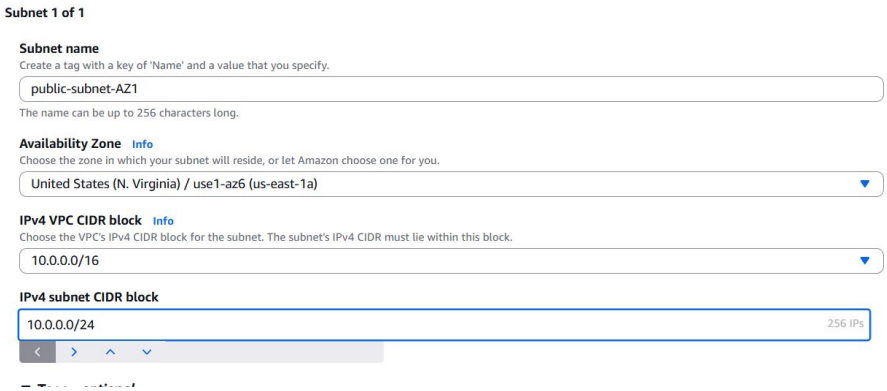
The next step is to attach the Internet Gateway to the Dev VPC to allow the VPC to communicate with the internet. To do this, we could either click on the pop up on the Internet Gateway interface which appears after creating the Internet Gateway or we could click on “Actions”, then select “attach VPC” from the drop down menu.

Under “Available VPCs”, click on the search box and select the VPC you wish to attach the Internet Gateway to. In this case, select “Dev VPC”, then click on “Attach Internet Gateway”.



The Next step is to create our public subnets in the first and second availability zones according to our reference architecture. To do this;

1. Select “Subnets” on the left side where you have the VPC dashboard drop down menu.
2. On the top right corner, click on “Create Subnet”
3. Select the VPC where you want the subnet to create the subnet in and in this case, it’d be the Dev VPC
4. Give the subnet a name; Public Subnet AZ1
5. Under availability zones, select U.S East 1a referencing our architecture above
6. Under IPv4 CIDR Block, enter and select 10.0.0.0/24 according to our architecture.
7. Click on “Create Subnet”

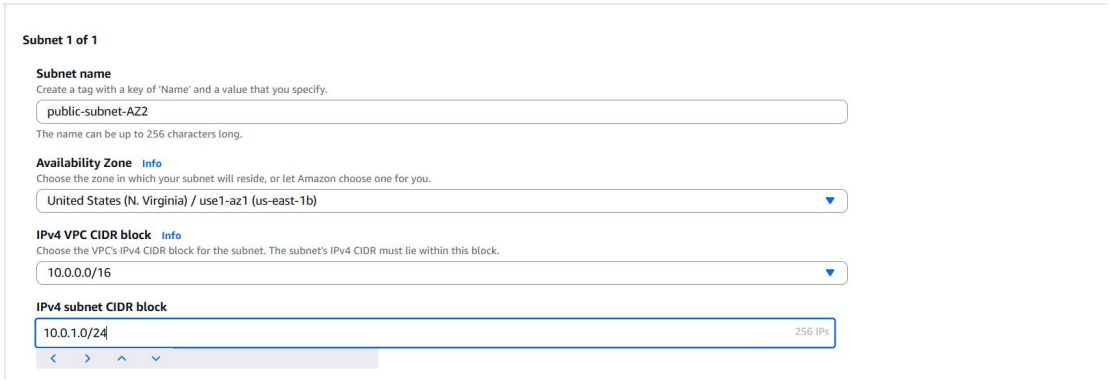


Knowing that we need two subnets but in different availability zones, we'll repeat the above steps but with some changes and they are;

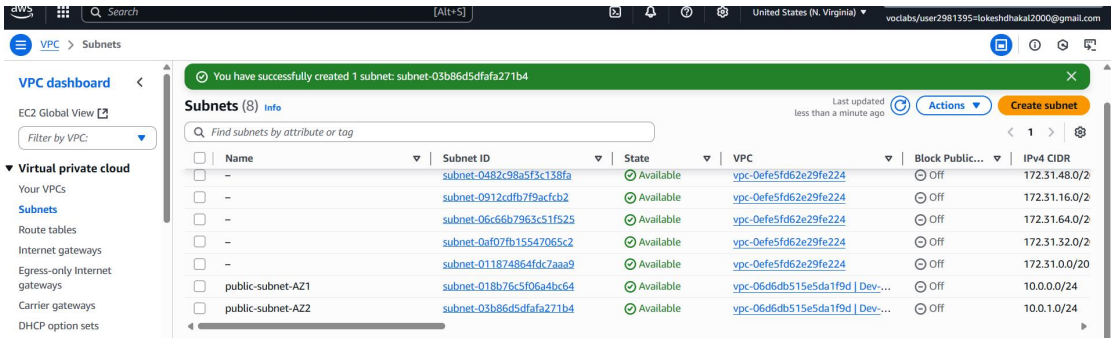
Subnet name- Public Subnet AZ2

Availability Zone- U.S East 1b

CIDR Block- 10.0.1.0/24

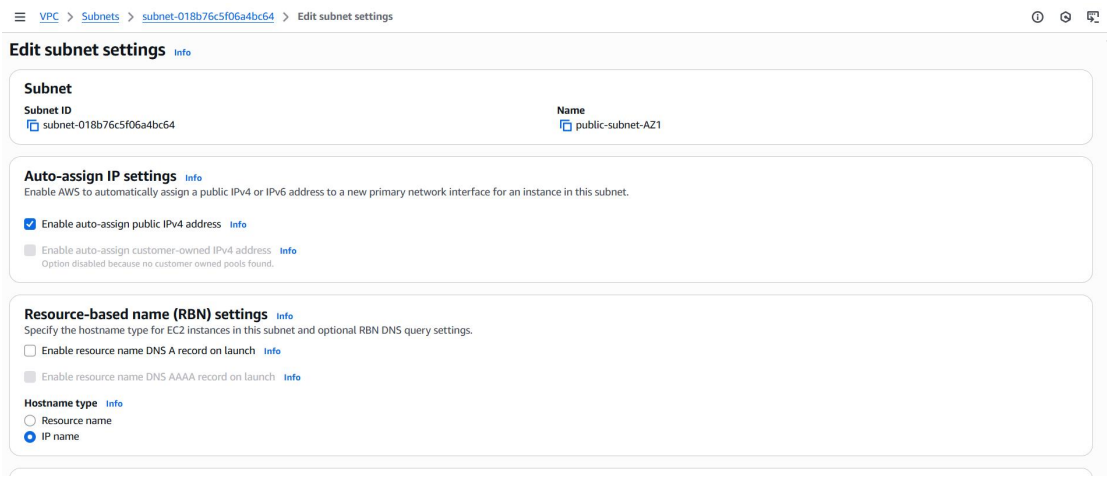


To see both subnets, on the left side, click on "filter by VPC" then select "Dev VPC". It'll show the subnets in Dev VPC i.e Public Subnet AZ1 and Public Subnet AZ2.

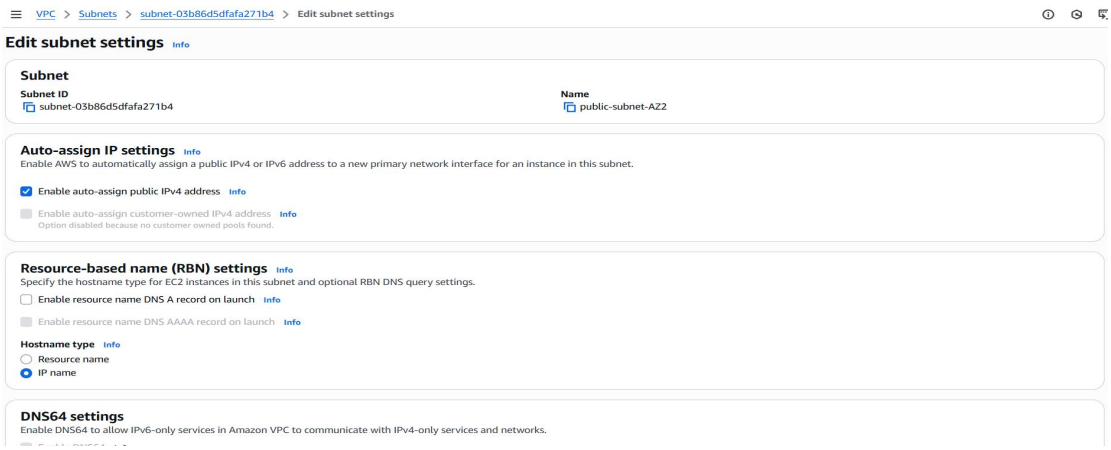


The next step is to enable “Auto Assign” for the two public subnets. This is to ensure that EC2 instances launched in these public subnets will be assigned a public IPv4 address. To enable this feature;

1. Select the first subnet (Public Subnet AZ1)
2. Click on “Actions” and click on “Edit Subnet settings”
3. Enable auto assign public IPv4 address by checking the box
4. Scroll down and click on “save changes”

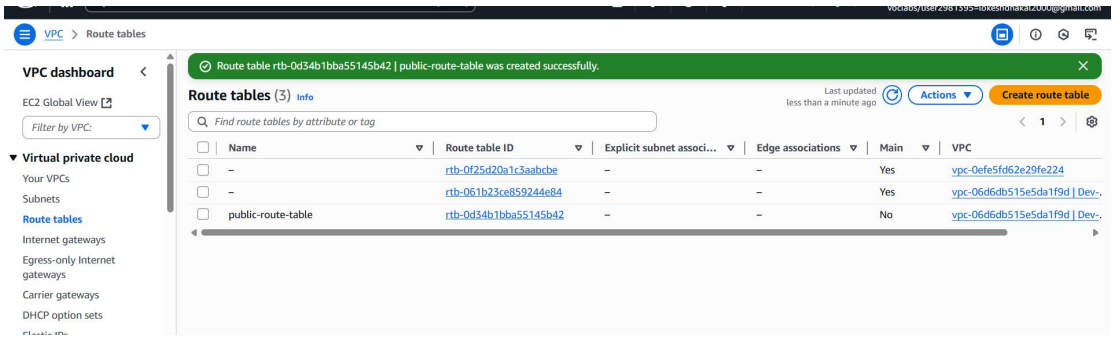


Repeat this steps for Public Subnet AZ2.



The next step is to create a Route Table and this would would be called “Public Route Table” according to our reference architecture. To create a Route Table;

1. Select Route Tables from the left side. Here, you’d see an already existing route table and this was automatically created when the Dev VPC was created. This route table is called the Main Route Table and it is private by default.
2. Click on “create route table”
3. Give the route table a name. In this case, it would be “Public Route Table”
4. Select the VPC you want to create the route table in i.e the Dev VPC.
5. Click “create Route Table”



After creating the route table, the next step is to add a public route to the route table. This would allow the route table to route traffic to the internet. To add a route to the public route table;

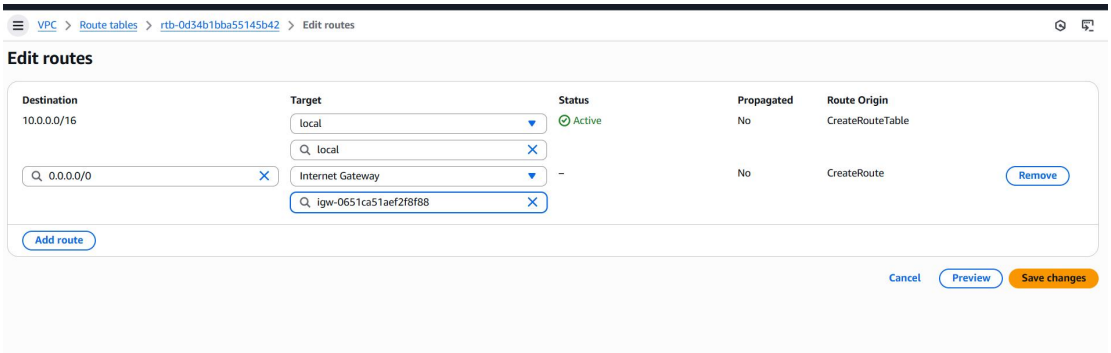
1. On the Public Route Table interface, select the “route” tab, then click on “edit routes”

2. Click on “add route”

3. Under destination, type and select 0.0.0.0/0

4. Under target, select the Internet Gateway i.e Dev Internet Gateway

5. Click on “save changes”

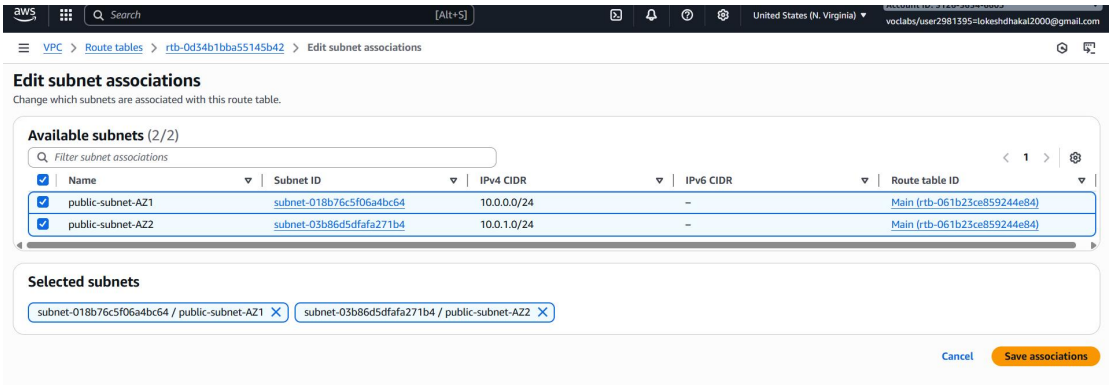


The next step is to associate the two public subnets with this route table. To do this;

1. On the Route Table interface, click on “subnet associations” then click on “edit subnet associations”

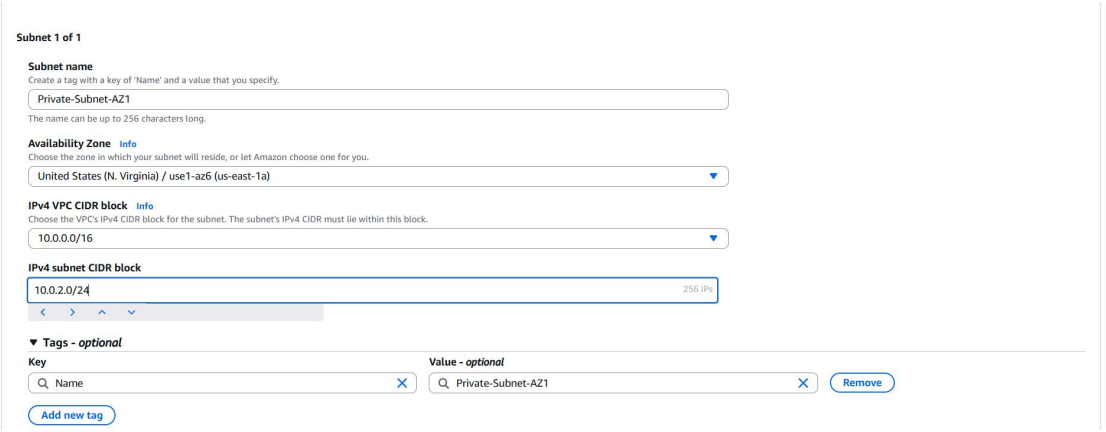
2. On this page, you’ll see the two subnets we created (Public Subnet AZ1 and Public Subnet AZ2)

3. Select both subnets by checking the boxes then click on “save associations”



The next step is to create our four private subnets. To do this;

1. Select “subnets” on the left side
2. Click on “create subnets”
3. Select the VPC we want the subnet to be in, in the case; Dev VPC
4. Give the subnet a name. According to our reference architecture, it’d be Private App Subnet AZ1
5. Select the availability zone we want the subnet to be in and that is U.S East 1a according to our reference architecture.
6. Under IPv4 CIDR Block, type and select 10.0.2.0/24 according to our reference architecture.
7. Scroll down and Click on “create subnet”



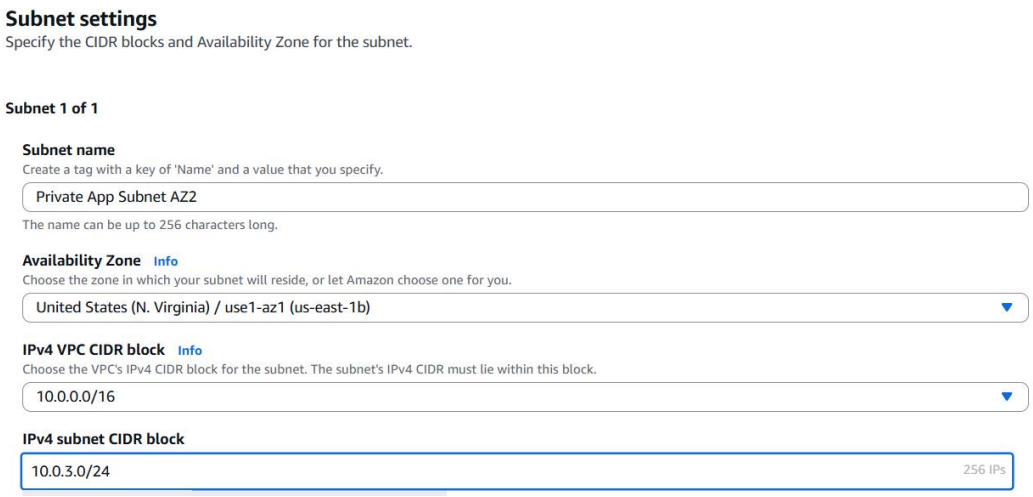
Now we would repeat the steps for other private subnets bearing in mind that their names, availability zone and CIDR Block would be according to the reference architecture.

For the second private subnet, we'll have;

Name: Private App Subnet AZ2

CIDR Block: 10.0.3.0/24

Availability zone: U.S East 1b

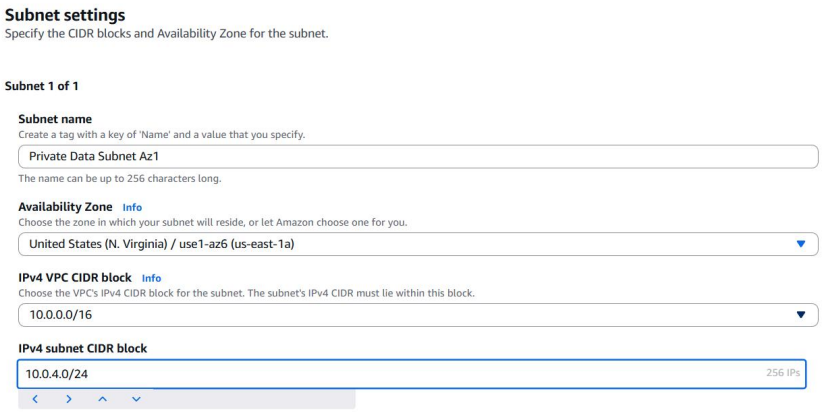


For the third private subnet, we'll have;

Name: Private Data Subnet AZ1

CIDR Block: 10.0.4.0/24

Availability zone: U.S East 1a

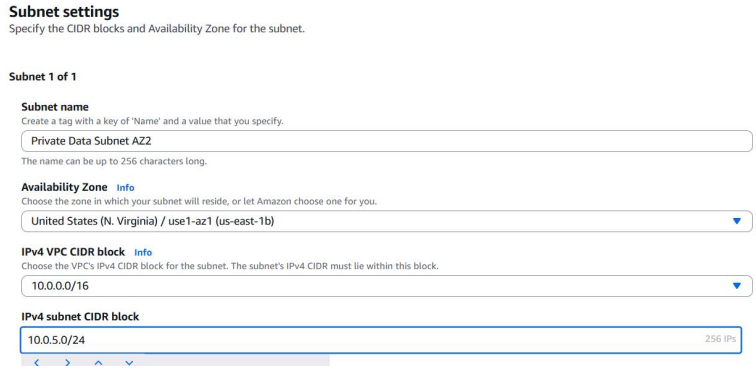


For the fourth private subnet, we'll have

Name: Private Data Subnet AZ2

CIDR Block: 10.0.5.0/24

Availability zone: U.S East 1b

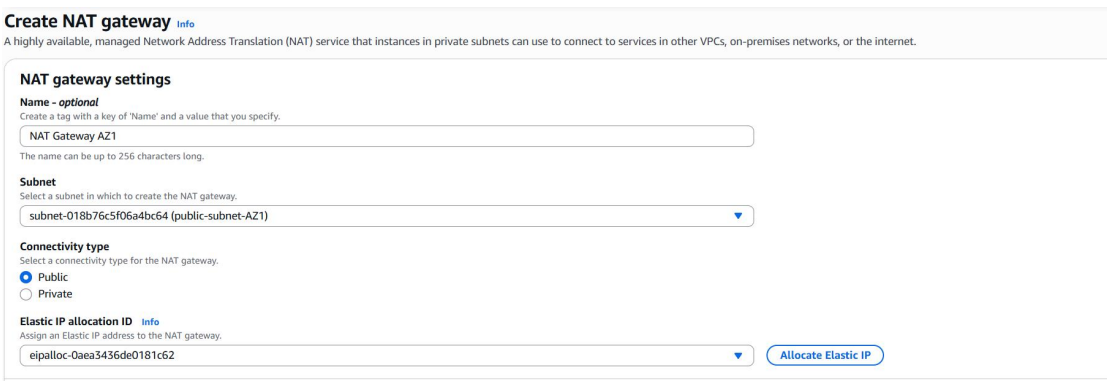


One important thing to know is the difference between a public subnet and a private subnet. When you create a route table and add a route to the route table to route traffic to the internet, any subnet you associate with that route table becomes public. So basically, a public subnet is associated with a route table that is able to route traffic to the internet. On the other hand, when you have a route table that does not have a route to the internet, any subnet associated to that route table is private.

The next step is to create a NAT Gateway which allows the instances in the private subnets to access the internet. According to our reference architecture, we'll create a NAT Gateway in the public subnet AZ1, then we'll create a private route table i.e Private Route Table AZ1. Afterwards, we'll add a route to the route table to route traffic to the internet through the NAT Gateway, then we'll associate the private subnets in AZ1 to the private route table. We'll repeat the process in the second availability zone so as to ensure that the private subnets in AZ2 can access the internet.

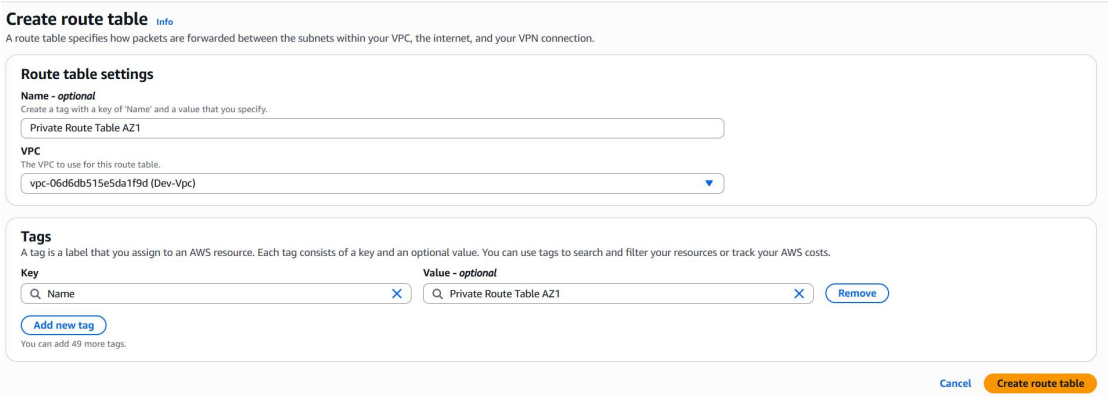
To create a NAT Gateway, make sure you're in the region where you created your VPC, in this case N.Virginia for Dev VPC.

1. In the search box of the management console, type "VPC", then select VPC under services
2. In the VPC dashboard, on the left side, select NAT Gateways
3. Click on "create NAT GAteway"
4. According to our reference architecture, we'll create our first NAT Gateway in Public Subnet AZ1
5. Give the NAT Gateway a name i.e NAT GAteway AZ1
6. Select the subnet we want to create the NAT Gateway in i.e Public Subnet AZ1
7. Under Elastic IP Allocation ID, click on "allocate elastic IP" which will allocate an Elastic IP for you.
8. Scroll down and click on "create NAT Gateway"



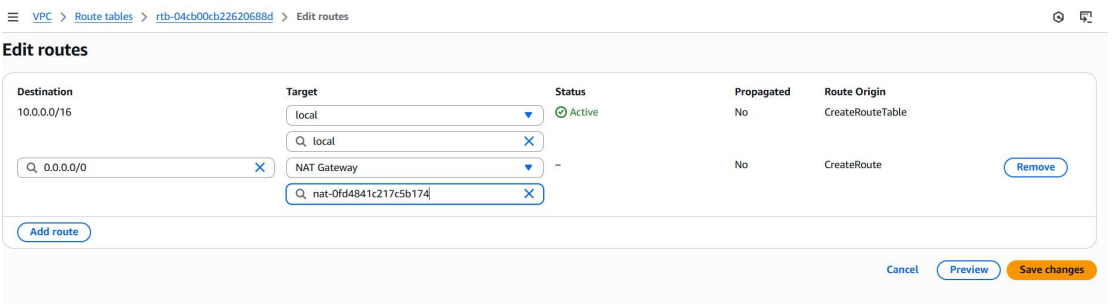
The next step is to create a route table and we'll call it "Private Route Table AZ1". To do this;

1. On the left side, click on "Route Tables"
2. Give the route table a name i.e "Private Route Table AZ1"
3. Select the VPC you want the route table to be created in. In this case; Dev VPC
4. Then Click on "create Route Table"



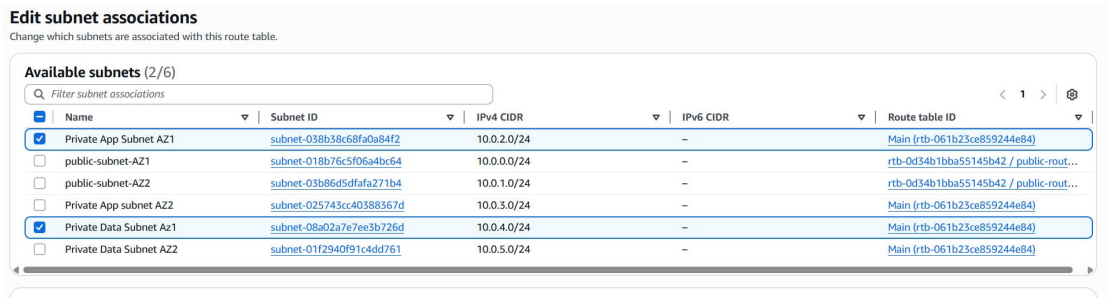
Next, we'll add a route to the Private Route Table AZ1. To do this;

1. On the Private Route Table AZ1 interface, under routes, click on "edit routes"
2. Click on "add route"
3. Under destination, type and select 0.0.0.0/0 and under target, it'd be NAT Gateway AZ1
4. Then click on "save changes"



The next step is to associate this route table to the private app subnet AZ1 and private data subnet AZ1. To do this;

1. Still on the Private Route Table AZ1 interface, Click on “subnet associations”, then click on “edit Subnet associations”
2. Under the available subnets, select only the Private App Subnet AZ1 and Private Data Subnet AZ1
3. Then click on “save Associations”



The next step is to create another NAT Gateway in the Public Subnet AZ2, then repeat the process but for private subnets that are in the second availability zone referencing our architecture. To do this;

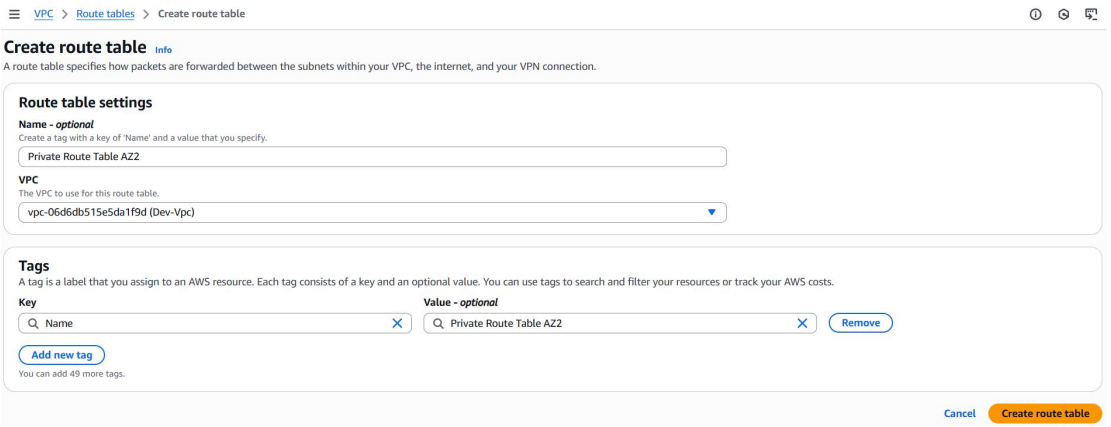
1. Click on NAT Gateways on the left side
2. Click on “create NAT Gateway”
3. Give the NAT GAteway a name. In this case, it’ll be NAT Gateway AZ2

4. Select the Subnet for this NAT Gateway which is Public Subnet AZ2

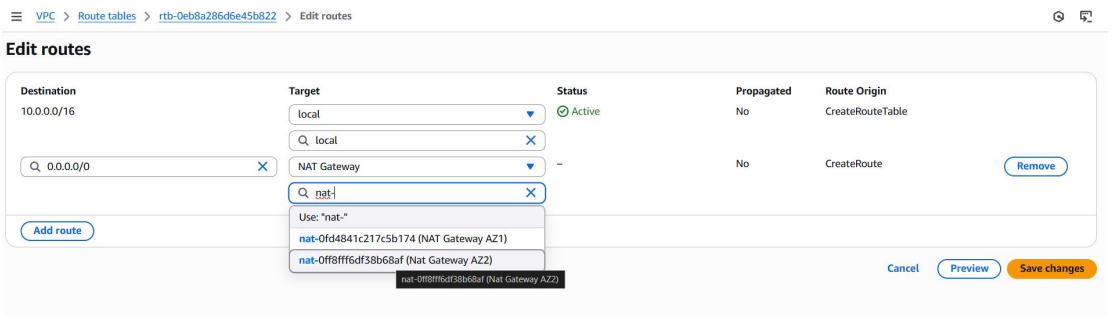
5. Under Elastic IP Allocation ID, click “Allocate Elastic IP”

6. Scroll down and click on “create NAT Gateway”

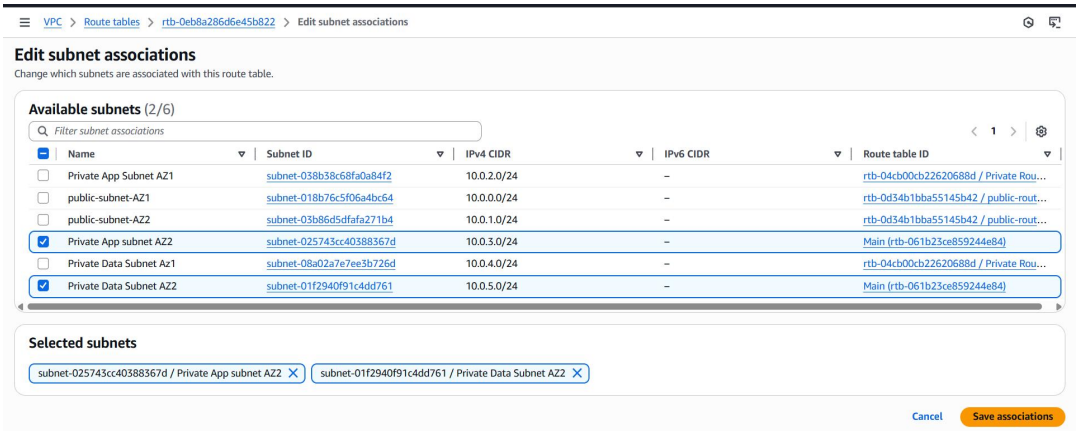
The next step is to create another route table (Private Route Table AZ2) following the previous step for creating the first private route table.



The next step is to add a route to the Private Route Table AZ2. We’ll also repeat the steps for this process but in this case, we’ll select the NAT Gateway AZ2.



The next and final step is to associate this route table to the Private App Subnet AZ2 and Private Data Subnet AZ2. We'll also repeat the steps for this process but in this case, we'll only select the aforementioned Subnets.



Finally, there it is, our Three Tier AWS Network VPC.