VPC Network Setup & Teardown Process

In this homework assignment, I will specify the planning, setup and teardown of a virtual private cloud network within the AWS cloud platform.

In the planning section, I explain what VPC network and subnets I chose and why. For the setup and teardown, I utilize a series of screenshots, giving a visual representation of the entire process within the AWS console.

Planning a VPC Network

Below are the details of why and how I planned out the VPC network.

VPC Network
10.136.0.0/16

I chose the 136 network for several reasons. I wanted to have fun with creating the network on my own, so picked the number 36, which represents the Wu Tang 36 chambers legendary album. On this album is the world famous song C.R.E.A.M: Cash Rules Everything Around Me, Get the money, dollar, dollar bill y'all!

The very essence of this song illustrates one of major benefits of learning cloud computing which is the near limitless high income potential in blue(pacific) ocean market of opportunities.

The number 1 represents there being only 1 Wu Tang group.

I changed the subnet digits within the CIDR block to even numbers, because I like even numbers.

Yet the single, double and triple digit pattern scheme that signifies which subnet and what is contained in each
Subnet remains the same as taught in the last class.

Subnets
Public: Load Balancer
Ports: 443, 8080, 22
AZa: 10.136.2.0/24
AZb: 10.136.4.0/24
AZc:10.136.6.0/24

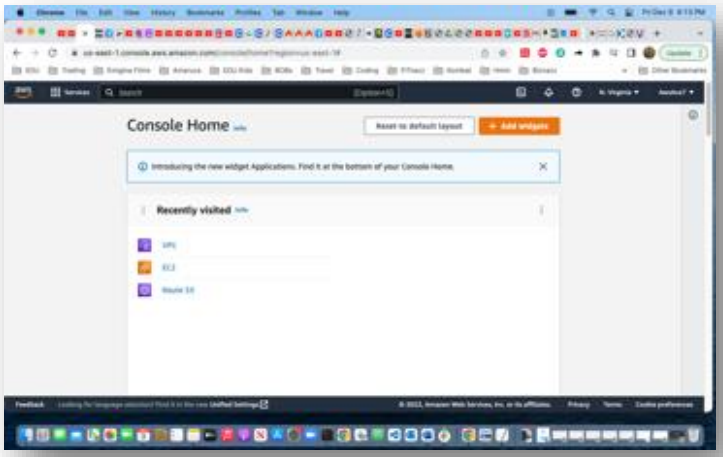
Private1: Front EC2:
Ports: 443, 8080, 22
AZa: 10.136.22.0/24
AZb: 10.136.24.0/24
AZc: 10.136.26.0/24

Private2: Database:
Security Group Port: 1521
AZa: 10.136.212.0/24
AZb: 10.136.214.0/24
AZc: 10.136.216.0/24

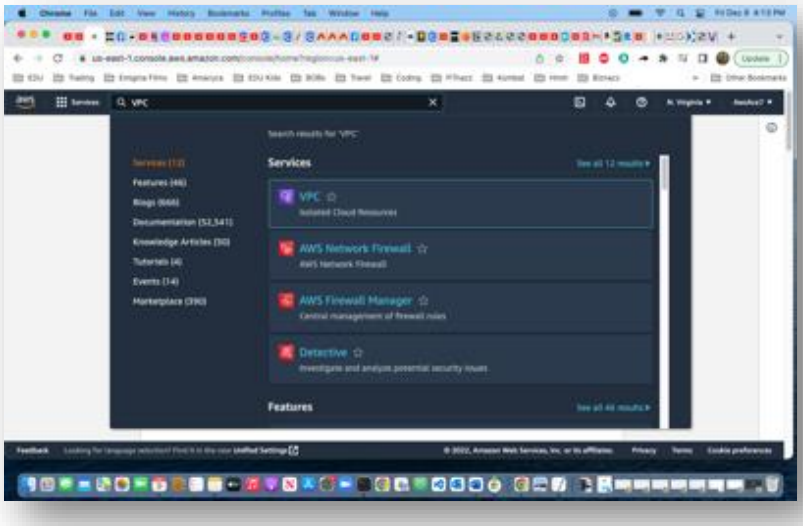
VPC NETWORK SETUP

Here in the series of screenshots, details the virtual private cloud network setup in AWS.

This is where we begin, at the AWS console home page, within our root user AWS account.

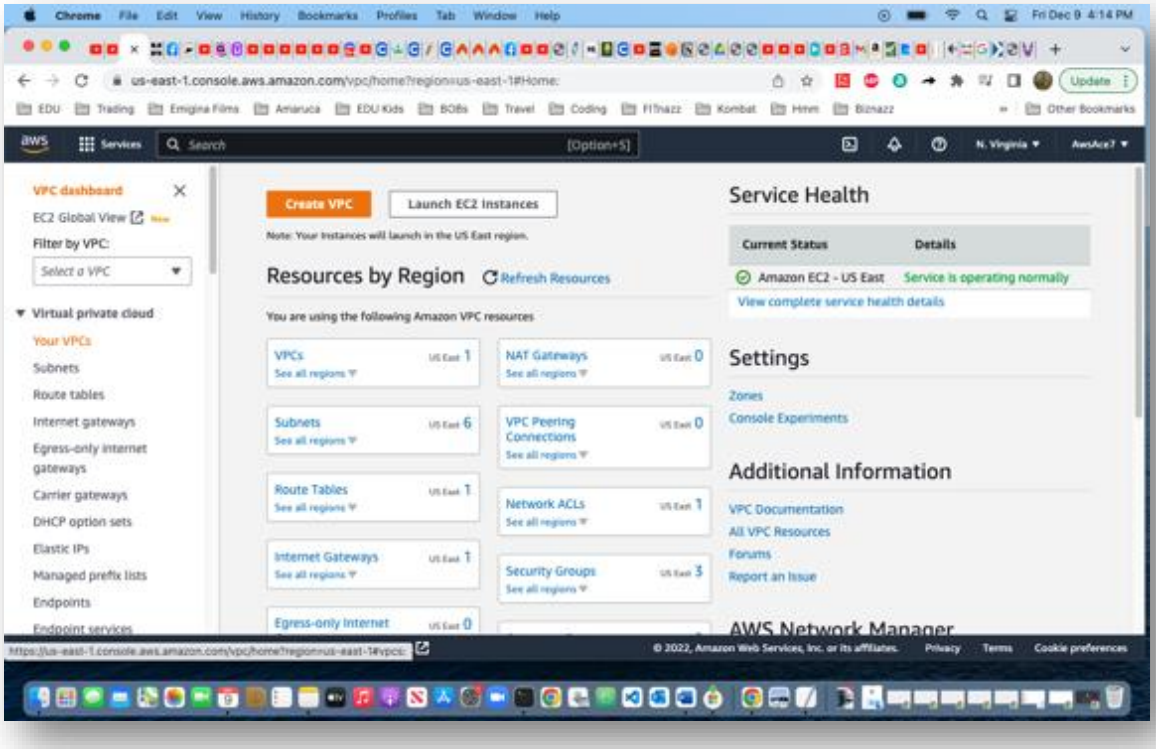


Here we type “VPC” in the search box at the type of our screen, to bring up the VPC option we see below. We select the VPC option.



We have arrived at the VPC Dashboard where we will begin creating a VPC network and its various components.

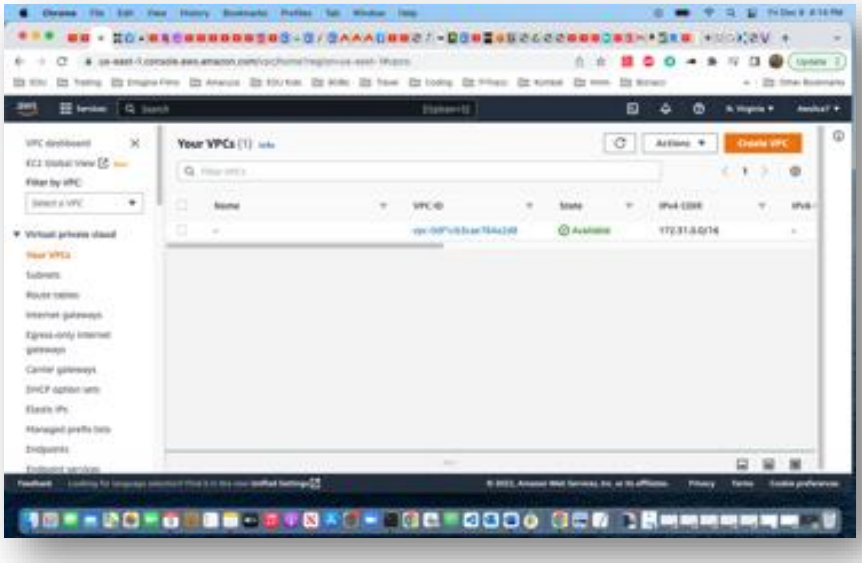
On the left side of the screen, we can select the option “Your VPC’s”, to see all the VPC’s we have available.



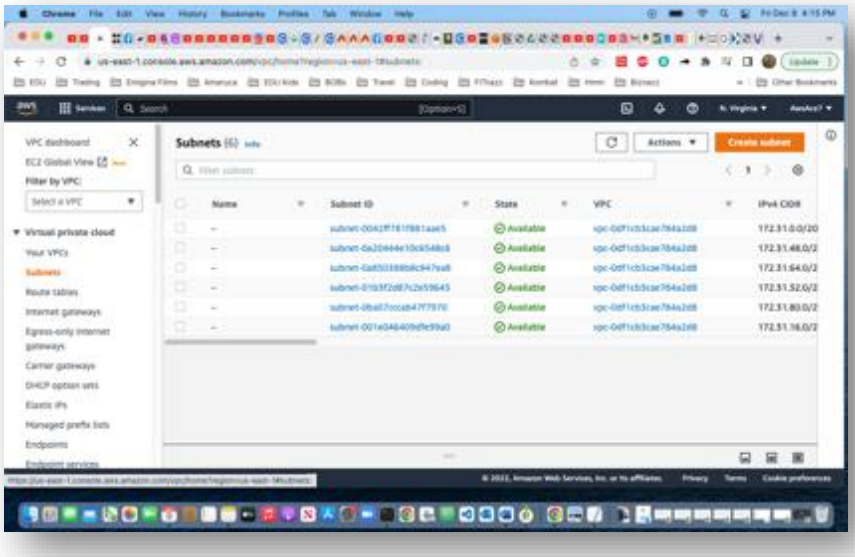
As you can see in in the pictures below, we already have a default VPC. Each AWS account comes with a standard default VPC, subnets, net gateways, routers etc.

Note: ***NEVER ever alternate or delete your default VPC and its devices.***

From here we can select the “Create VPC” button on the right of the screen to begin setup.

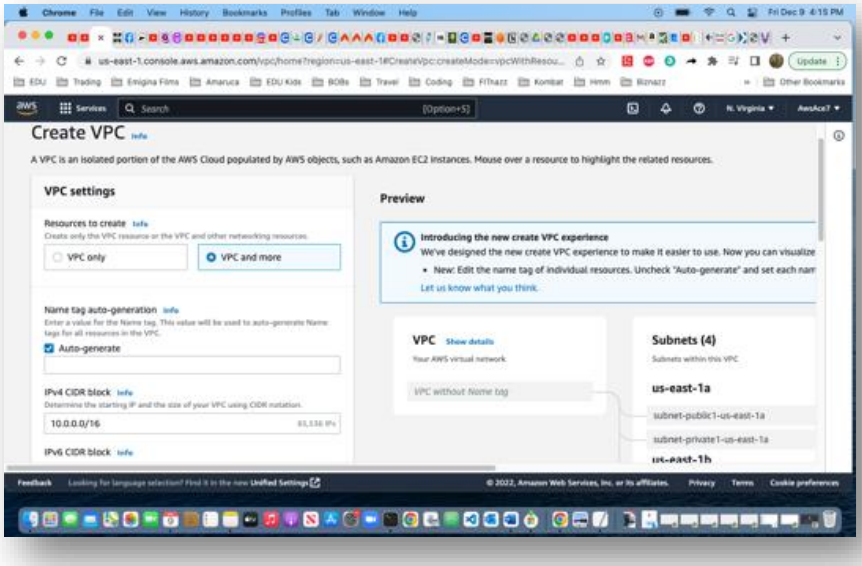


This picture showcases our default six subnets, spread over multiple availability zones.



Now we can begin setting up the VPC as well as observe the diagram of the network on the right to check what components we may need and what we have already possess.

As you can see in the picture below we already have selected “VPC and more” so we can set up the various devices within the network simultaneously with creating the VPC itself.



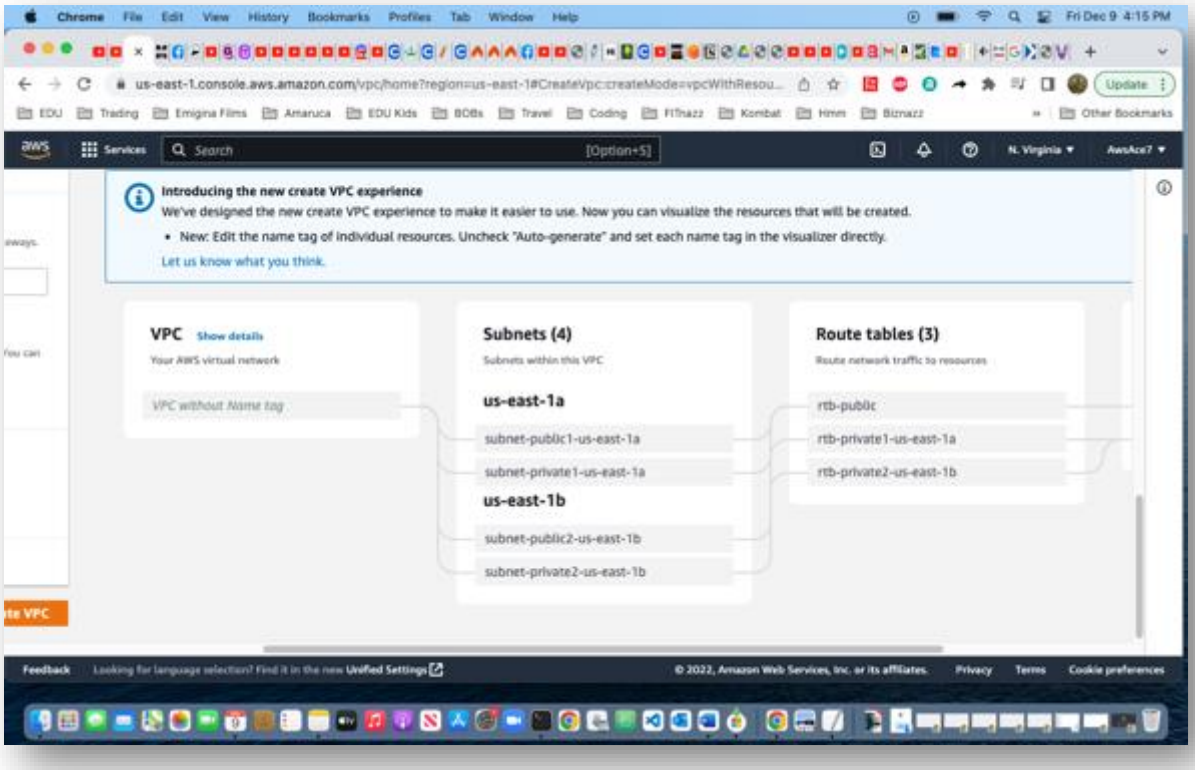
Here is more of the VPC network diagram. It is broken in several sections.

The first section on the left side of the diagram, is the VPC itself with its name.

The middle section shows the subnets divided into 2 availability zones, located in the eastern region of the United States.

The next section are the route tables, which are used to connect the subnets to public/private gateways for internet access.

The last section off pictue, are the gateways; one internet gateway(for public subnets) the connects to the internet and 2 nat gateways(for private subnets) that connect to the internet gateway for internet access.



Now we are back on the left side of the “Create VPC” page. In the Name tag auto generation section, we will name our VPC project.

Type any name for the VPC network.

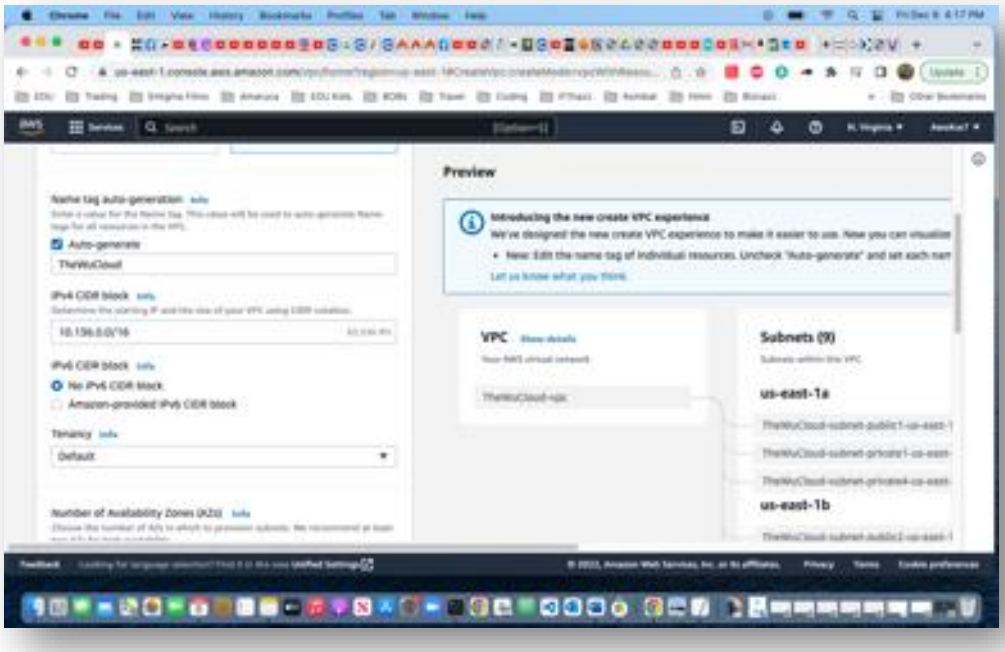
***Note:* it is advised to name your VPC network in relation to the purpose of its design**.

Next, we will assign its CIDR block range in the IPv4 CIDR block field below.

***Note:* The CIDR block range is the Classless Inter-Domain Routing block. This is the range IP addresses available to assign to our VPC as well as an IP address for each item within the VPC.**

Lastly, we go to the IPv6 CIDR block section and make sure we check off the “No IPv6 CIDR block”, option. We will be only using the IPv4 CIDR

block of IP addresses for this VPC.



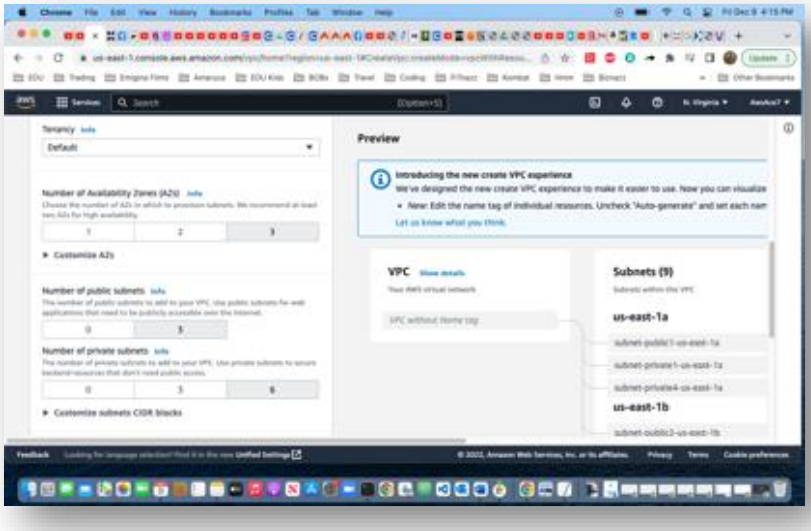
Here we go to the **Number of Availability zones** section and select “3”.

This means we are accessing 3 availiability zones in the U.S. Eastern region, to place all our future subnets and other devices for our VPC.

Note: ***Subnets are subnetworks within the VPC network. Subnets are used to house and organize similar devices. For example one subnet may contain all of the databases of the VPC while another subnet contains a software application.***

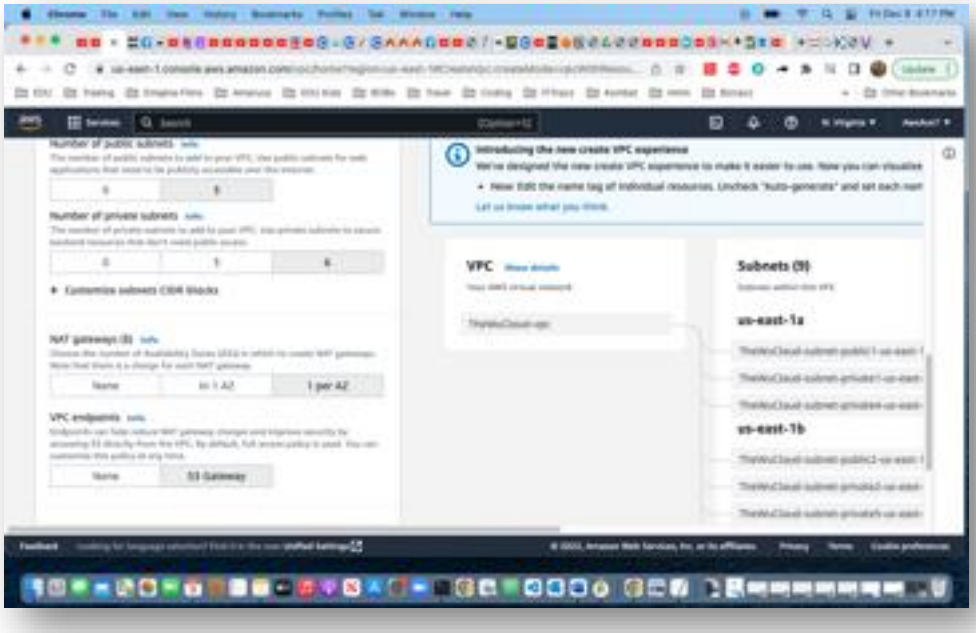
Next under the **Number of Public subnets**, we select 3 subnets. We’re creating 3 public subnets for the VPC.

Lastly, for **Number of Private subnets**, we select 6 subnets.

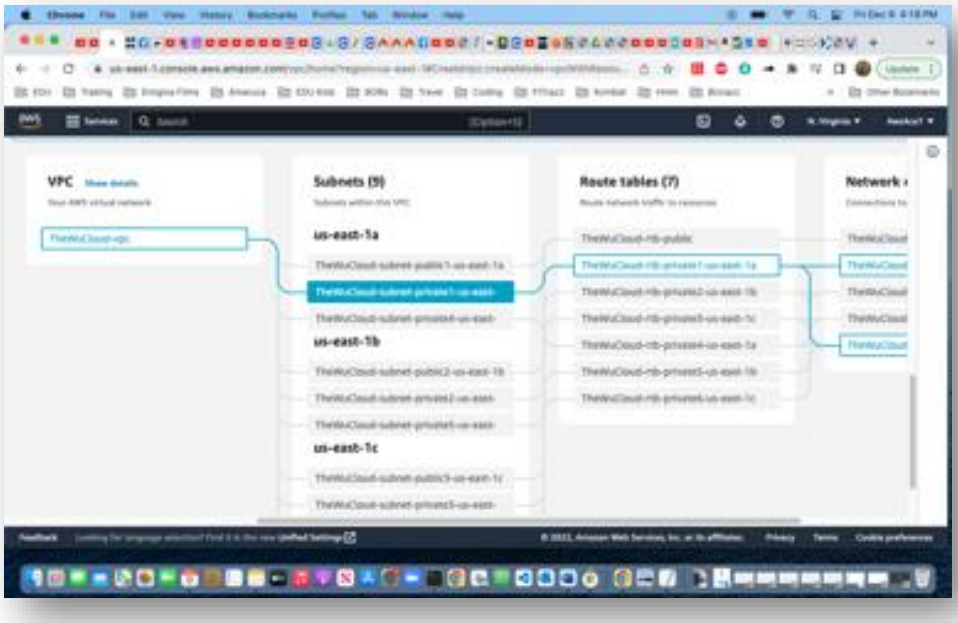


Now, assign 1 NAT gateway to each availability zone, shown below in the Nat Gateways section. Then we add a VPC endpoint S3 Gateway to the VPC.

Note: ***A VPC endpoint allows you to privately connect your VPC to other AWS services. In this case, we're connecting our VPC to the S3 Gateway which allows for more seamless data transfer between the devices in the VPC.***



Here is a highlighted example of how data is transferred/routed in and out the VPC.

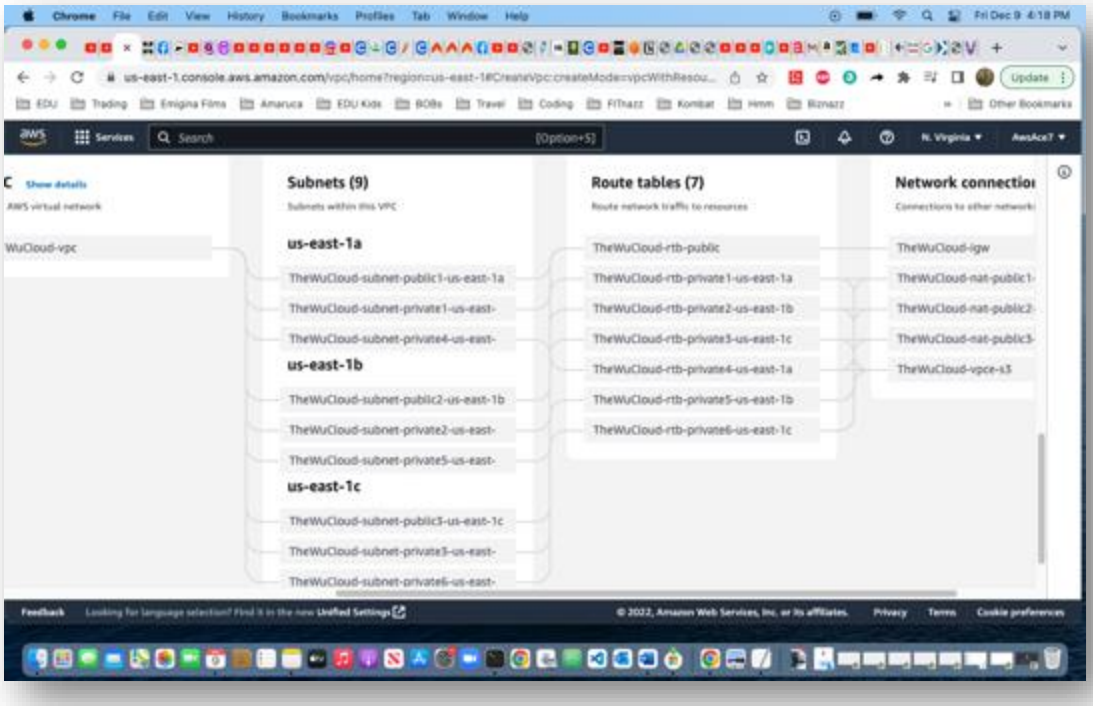


Here is the new VPC we are creating. As you can see below we have 9 subnets in 3 availability zones. One public subnet in each zone and 2 private subnets in each zone.

We have 1 routing table for each public subnet and a routing table for each private subnet. These routing tables connect to the gatewats. The public subnets use 1 routing table to communicate with the internet gateway.

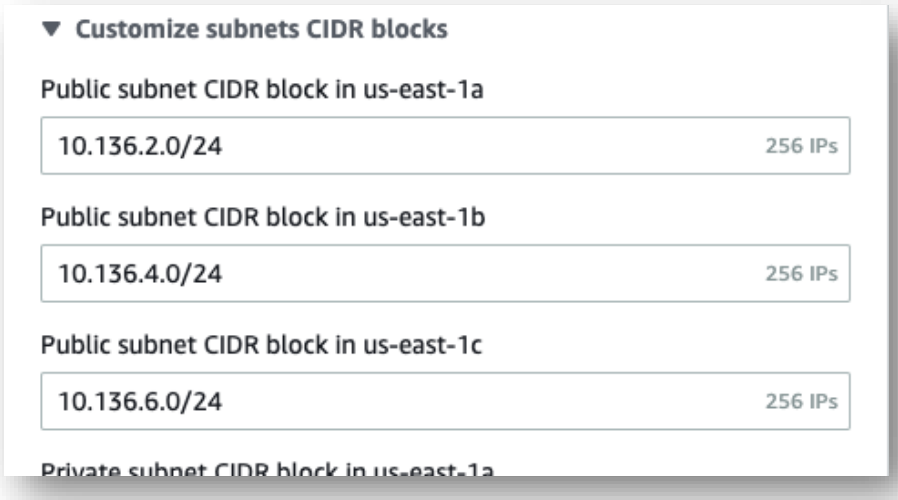
Each private subnet with its our routing table connects to availability zone assigned NAT gateway, which then connects to the internet gateway. The S3 Gateway allows for all th data to move more efficiently.

Note: ***Nat Gateways are Network Access Translation Gateways. They convernt private IP addresses of the private subnets into Public IP addresses, that are then sent to the internet gatetway for internet access.***

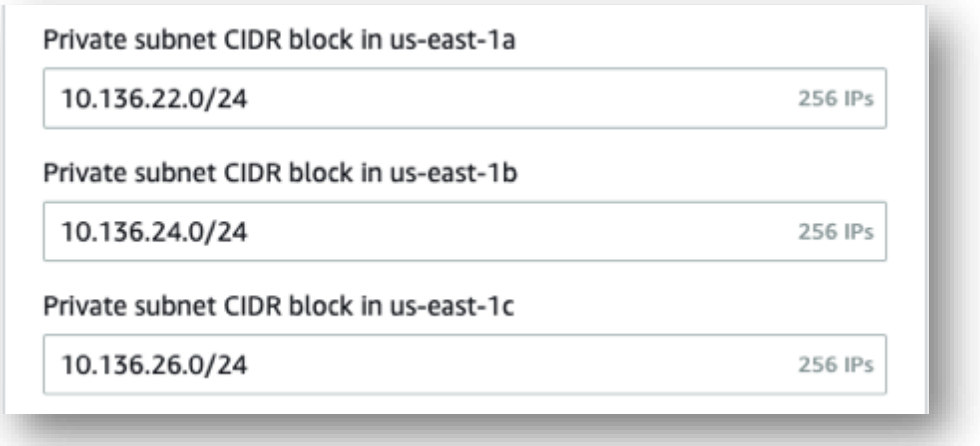


In the **Customize subnets CIDR blocks** section, we will begin to inputthe IP address ranges for the public and private subnetworks.

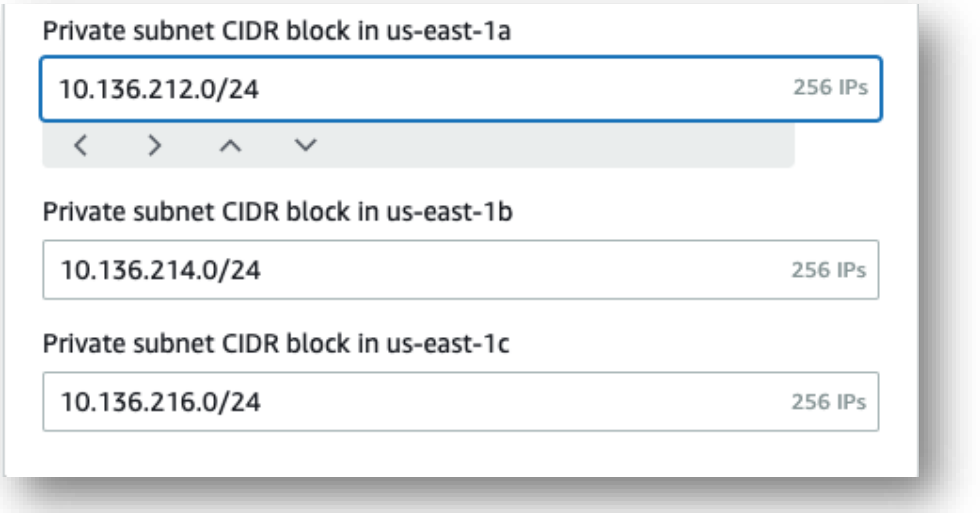
For the Public Subnets they will be used to house load balancers, which help divert traffic so no devices in the VPC are overwhelmed with in and out transfers of data.



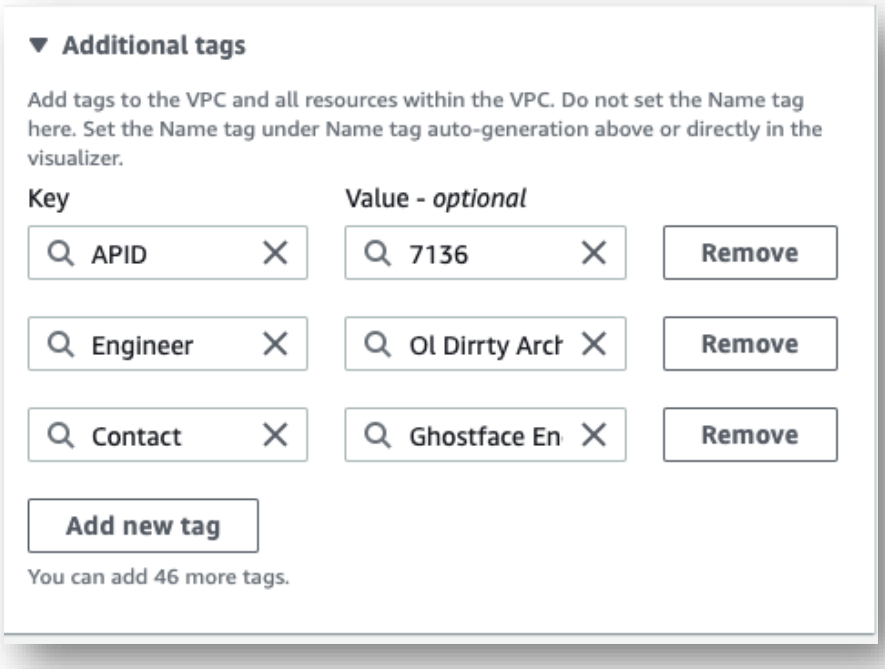
For the 1st Private subnet, it will house virtual machine functioning as web servers.



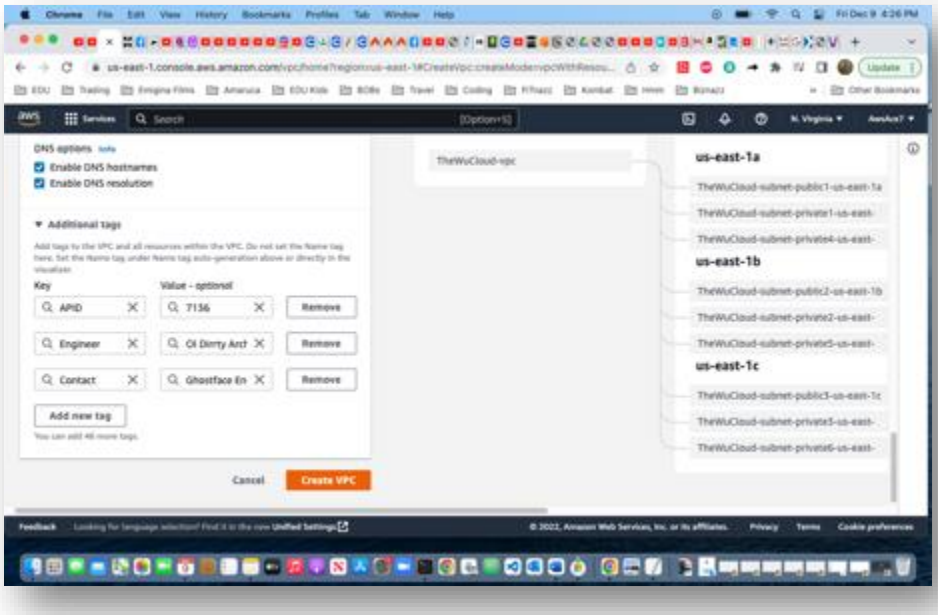
For the 2nd Private subnet, it will house databases with a security port, because of the sensitive information being transferred here.



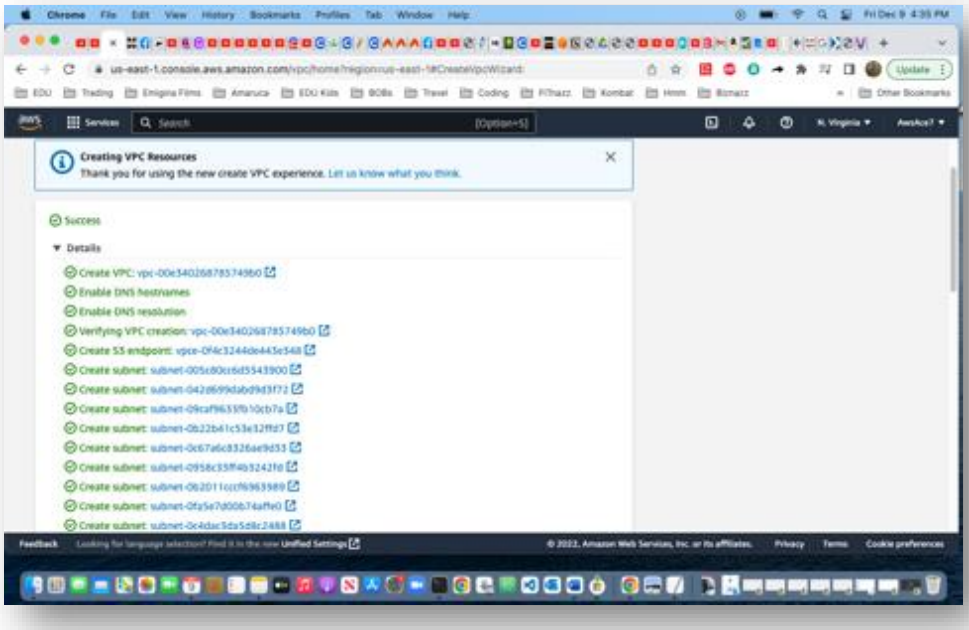
In the **Additional Tags**, section we add tags to help us to easily find and catalog our VPC project for retrieval later on.

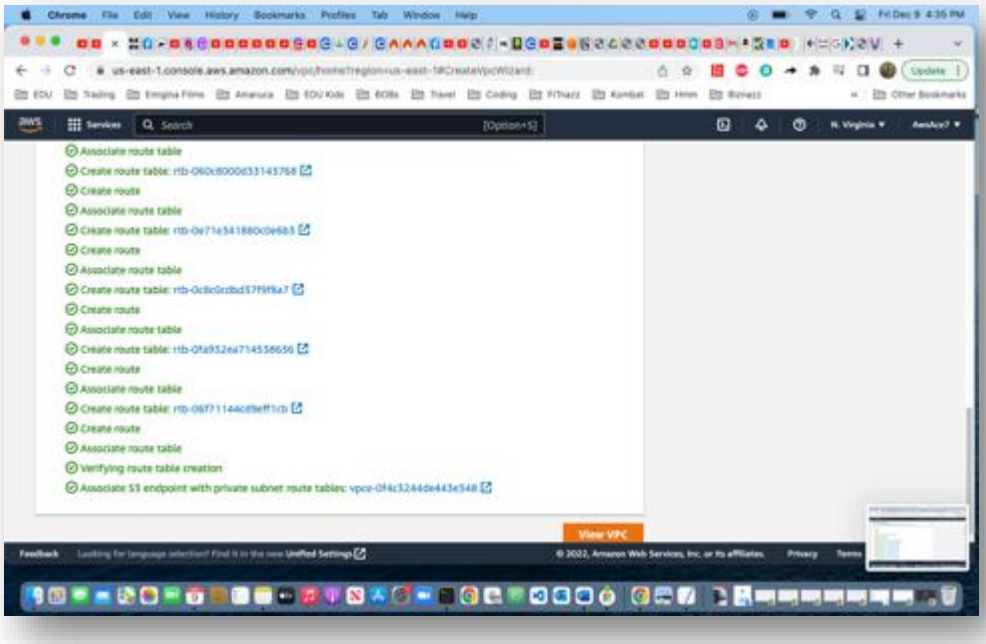


To finish setting up our VPC network we click the “Create VPC” button for the construction to commence.

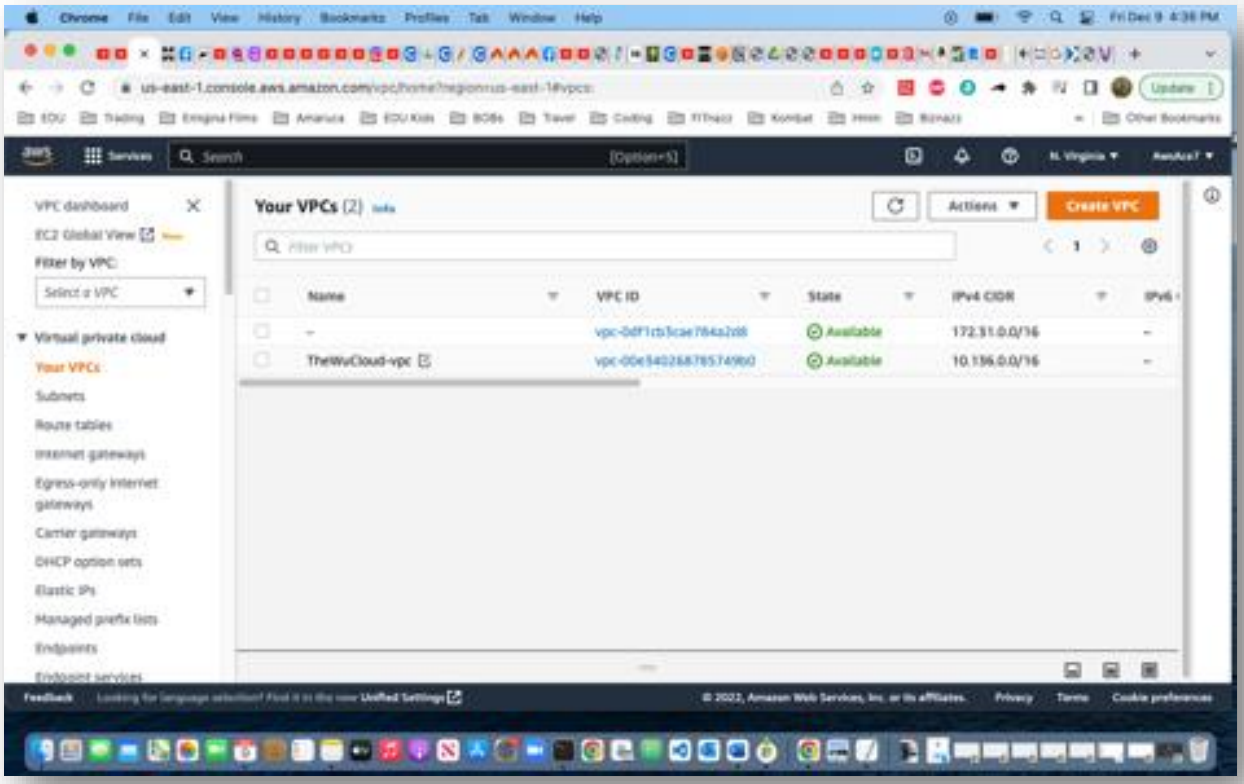


The next 2 images display a successful build out of the VPC network.

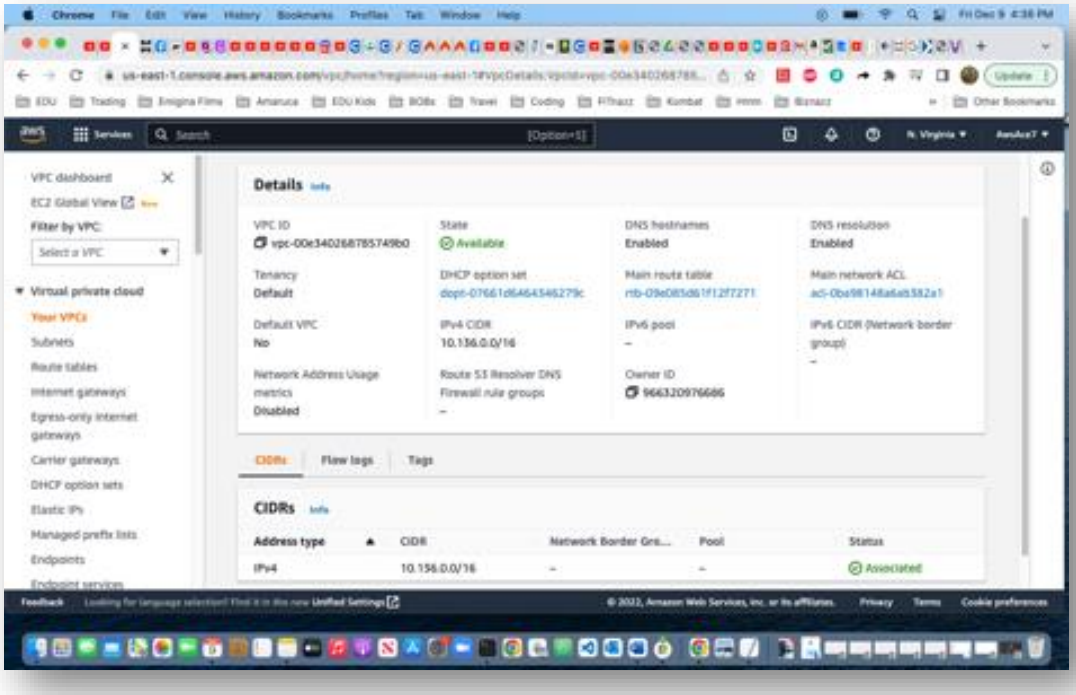


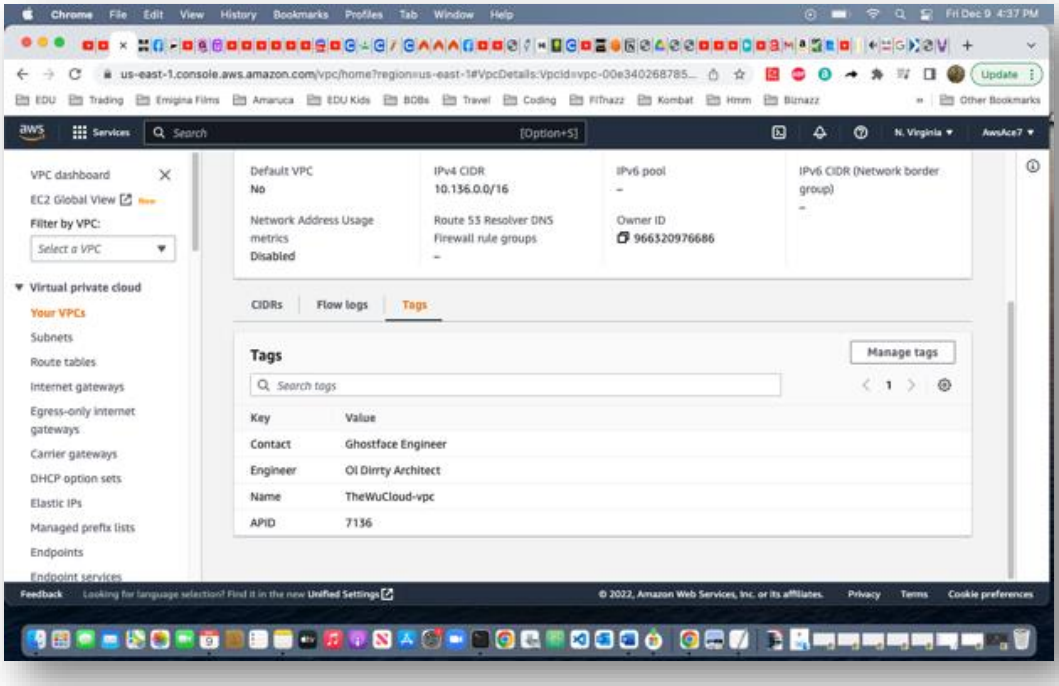


Once you click “View VPC” it takes you “Your VPC’s” page to see all the VPC’s available. You can see you finished VPC network along with your default VPC.



The next 2 images show you the details of our new VPC network.

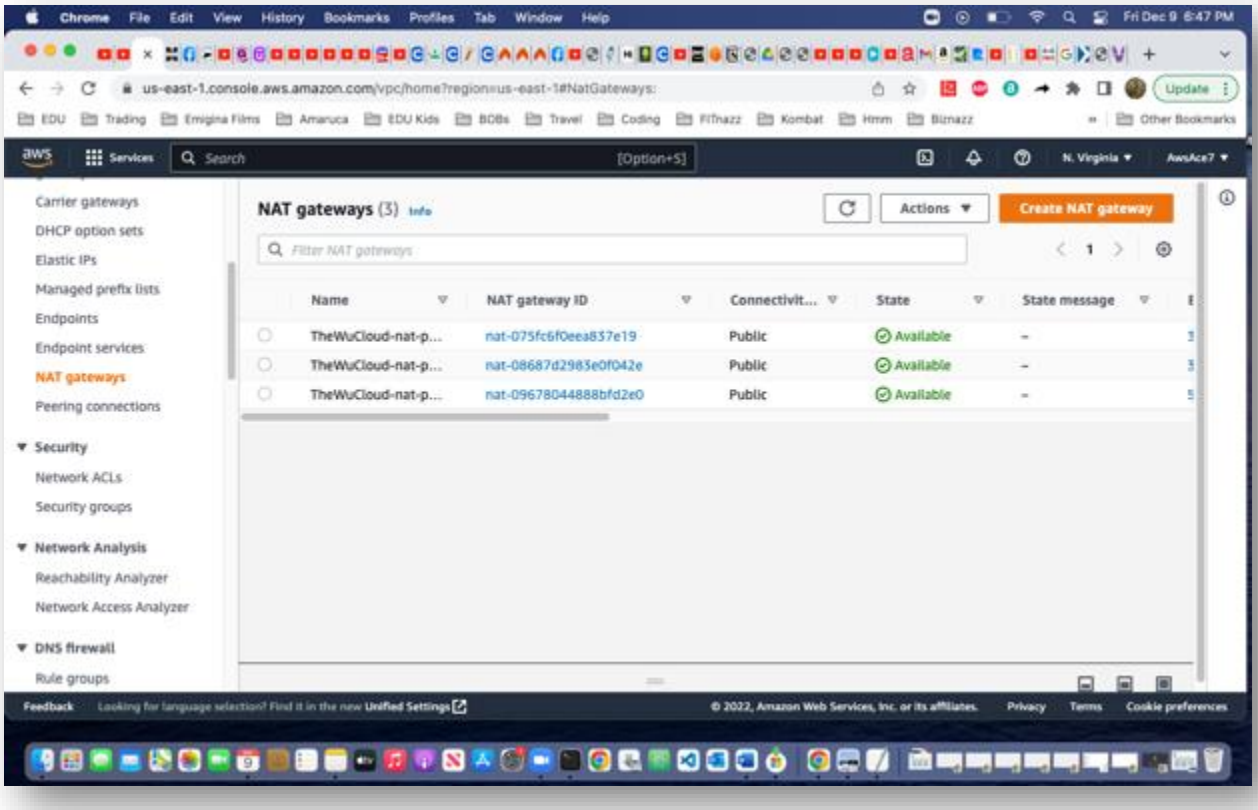




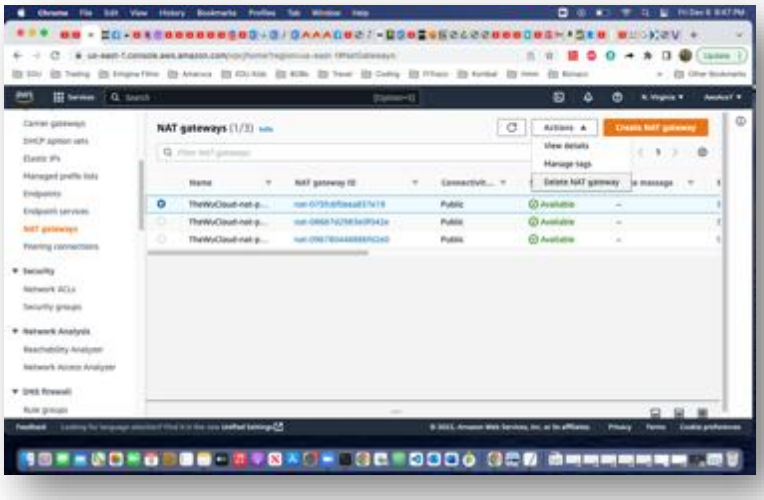
Tearing Down our VPC Network

To successfully tear down a VPC network, the sequence is as follows: we will terminate the NAT gateways, terminate the VPC network itself and finally release the Elastic IP addresses that were associated with the NAT gateways.

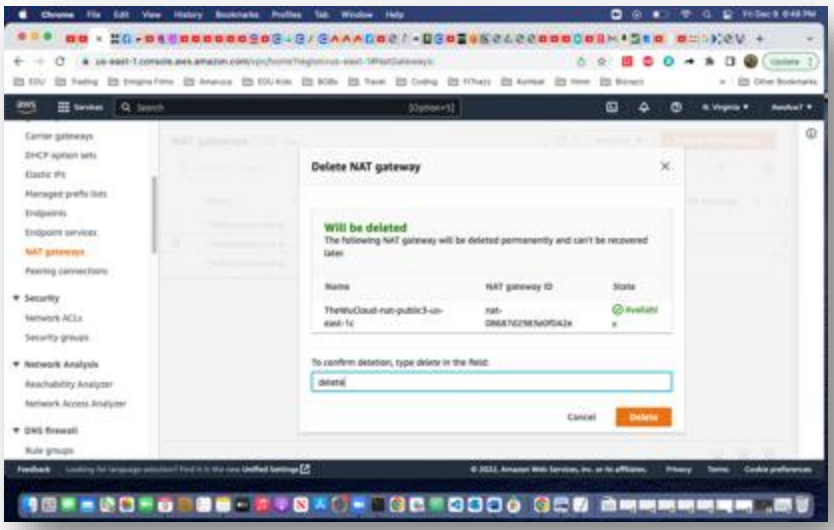
On the left side of the screen we scroll down the device menu to select NAT gateways. As you can see in the pictures below, we will select and delete each NAT gateway from the VPC network.



After selecting a NAT gateway, we go to the Actions button on the right of the page, scroll to the “delete NAT gateway”, option.

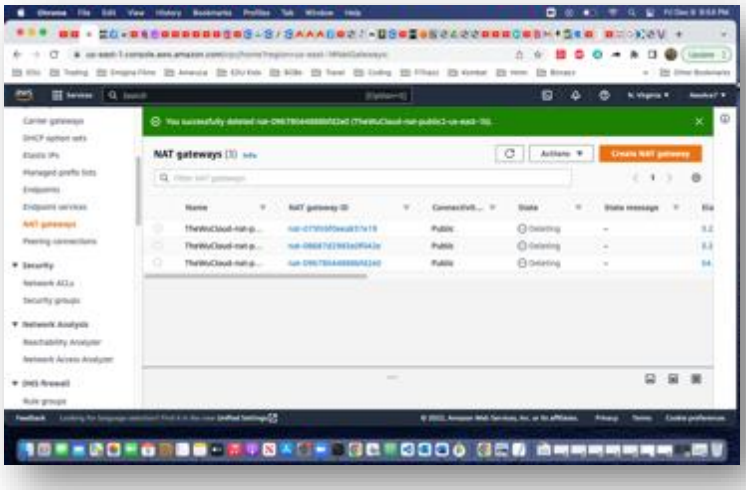


Once the Delete NAT gateway box pops up, we will have to type “Delete” in the confirmation box to authorize the deletion of the gateway. Then click the “Delete”, button.

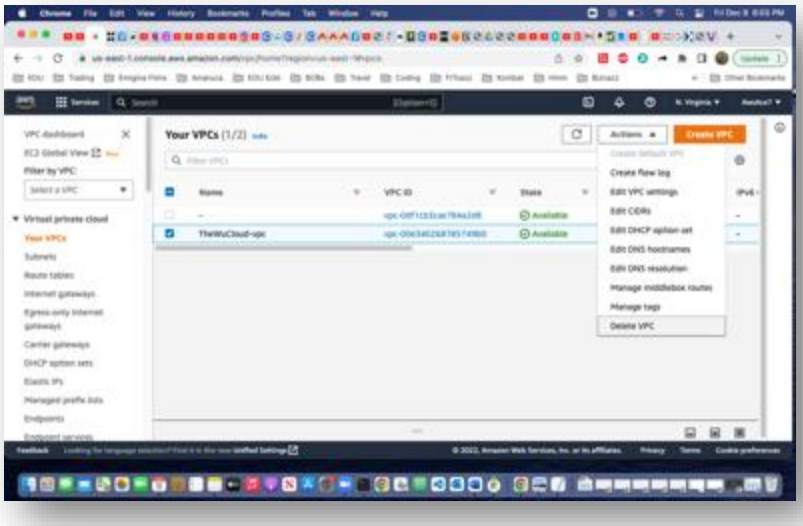


As can be seen below the NAT Gateways are being successfully deleted.

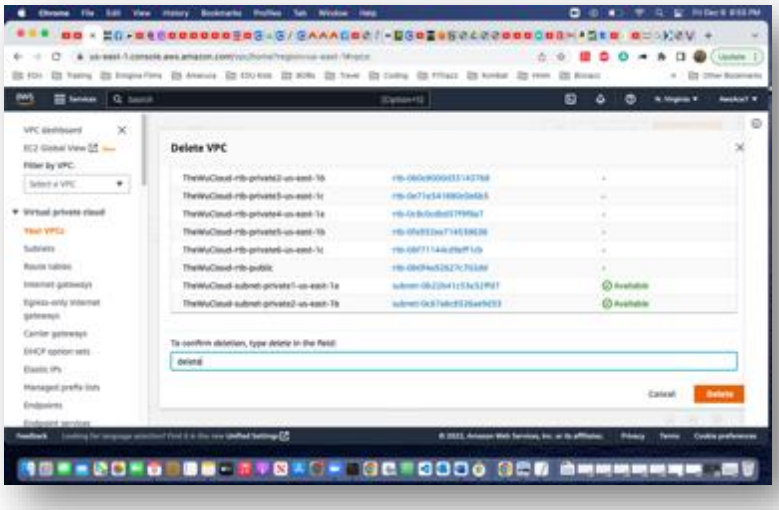
***Note:* It may take up to several minutes for the gateways to be fully terminated from the system.**



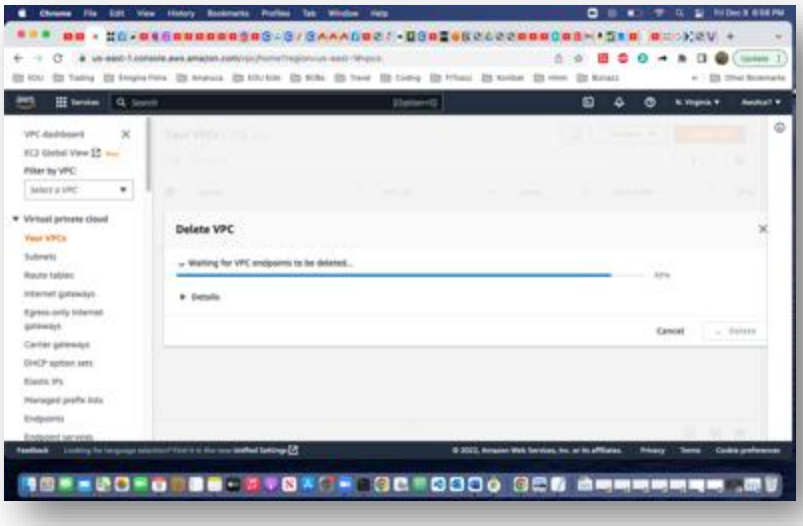
Next we delete the VPC network itself, following a similar step process as the NAT gateway terminations. We select our new VPC network (NOT the default, anything BUT the default VPC) and go to the **Actions** button and select “Delete VPC”.



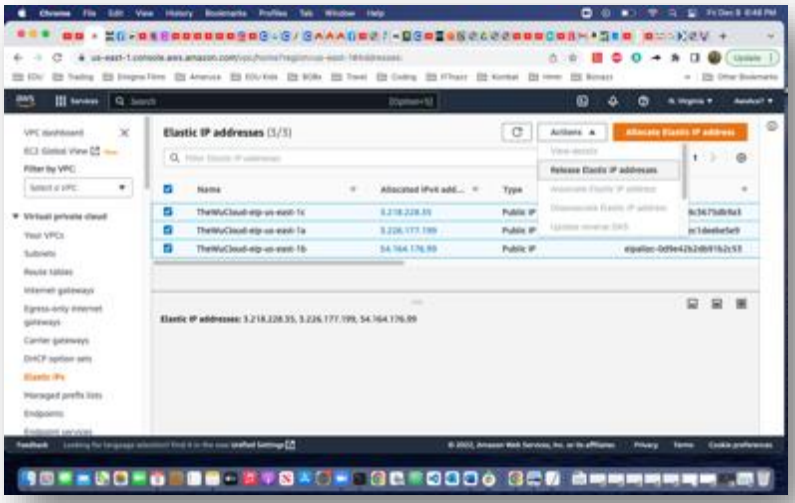
Here we again type “delete”” in the confirmation to authorize the termination of the entire VPC network.



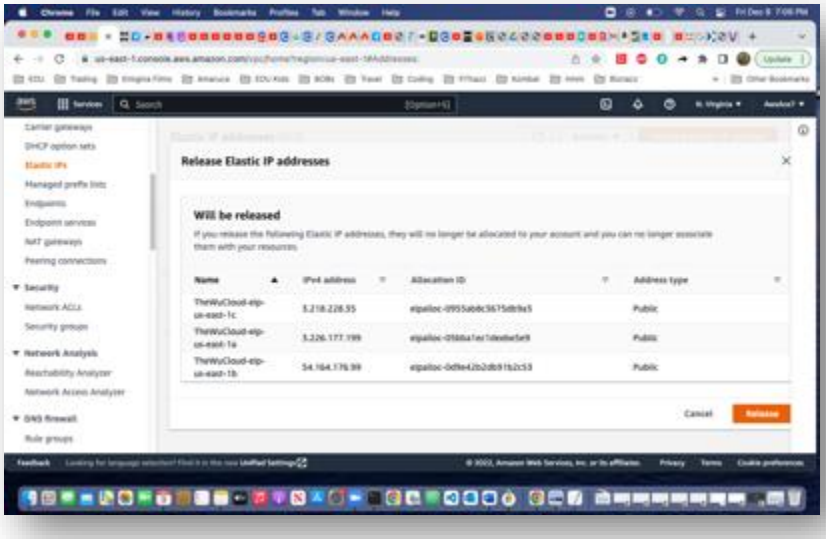
As can be seen below, the VPC is being terminated piece by piece.



Lastly we go to the Elastic IPs button from left side device menu. We select all the EIPs, then go to the **Actions** button to find the “release Elastic IP addresses” option.



After the **Release Elastic IP addresses** dialog box appears, we select the “Release” button to finish the tear down.



As shown below in the green bar going across the page, the EIPs have been released and the VPC tear down process is complete.

You can go to the VPC dashboard to see the several devices as well as the

VPC itself, deleted from the AWS region