**Implementing a Team Communication Solution Using Mattermost and AWS.**

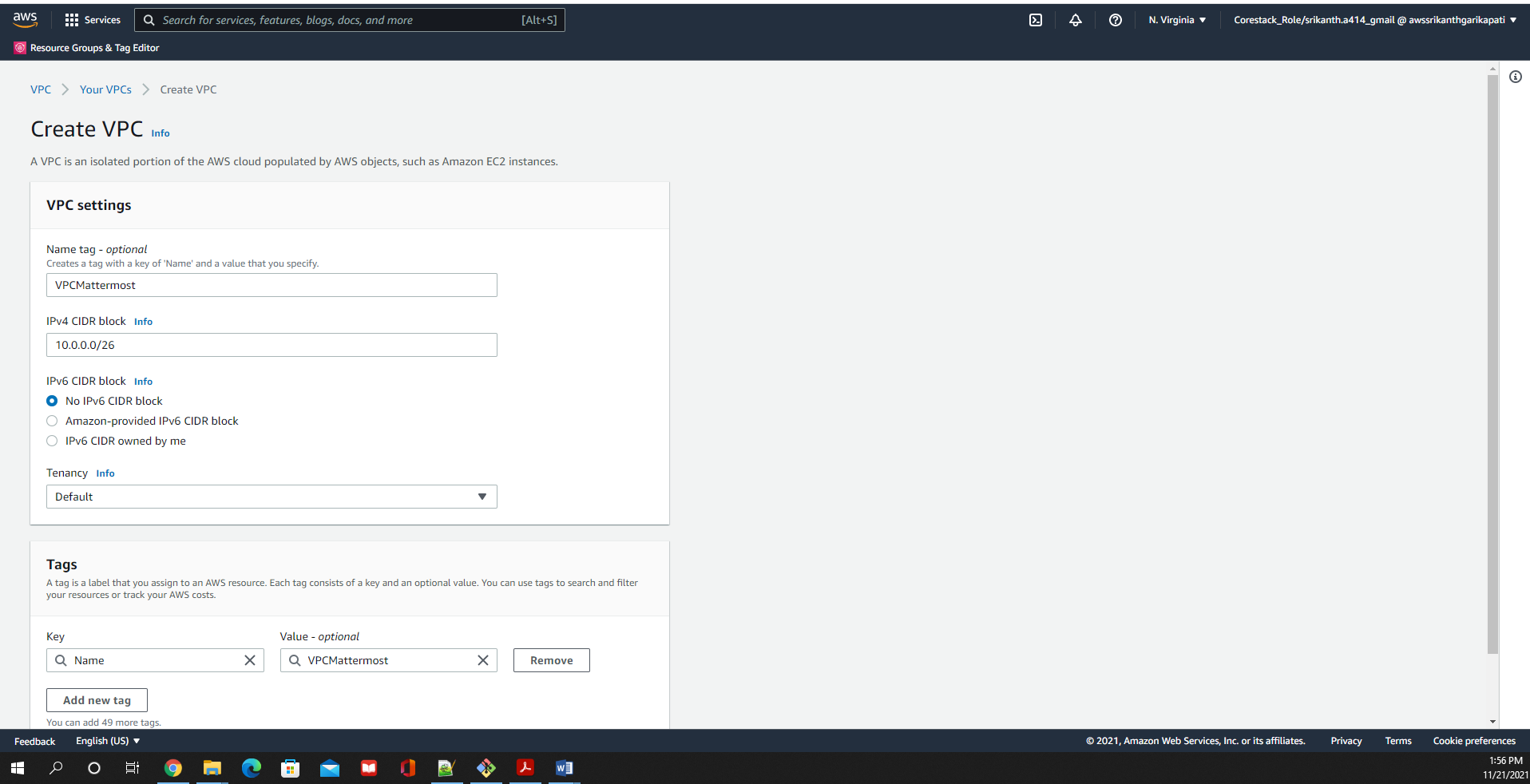
DESCRIPTION

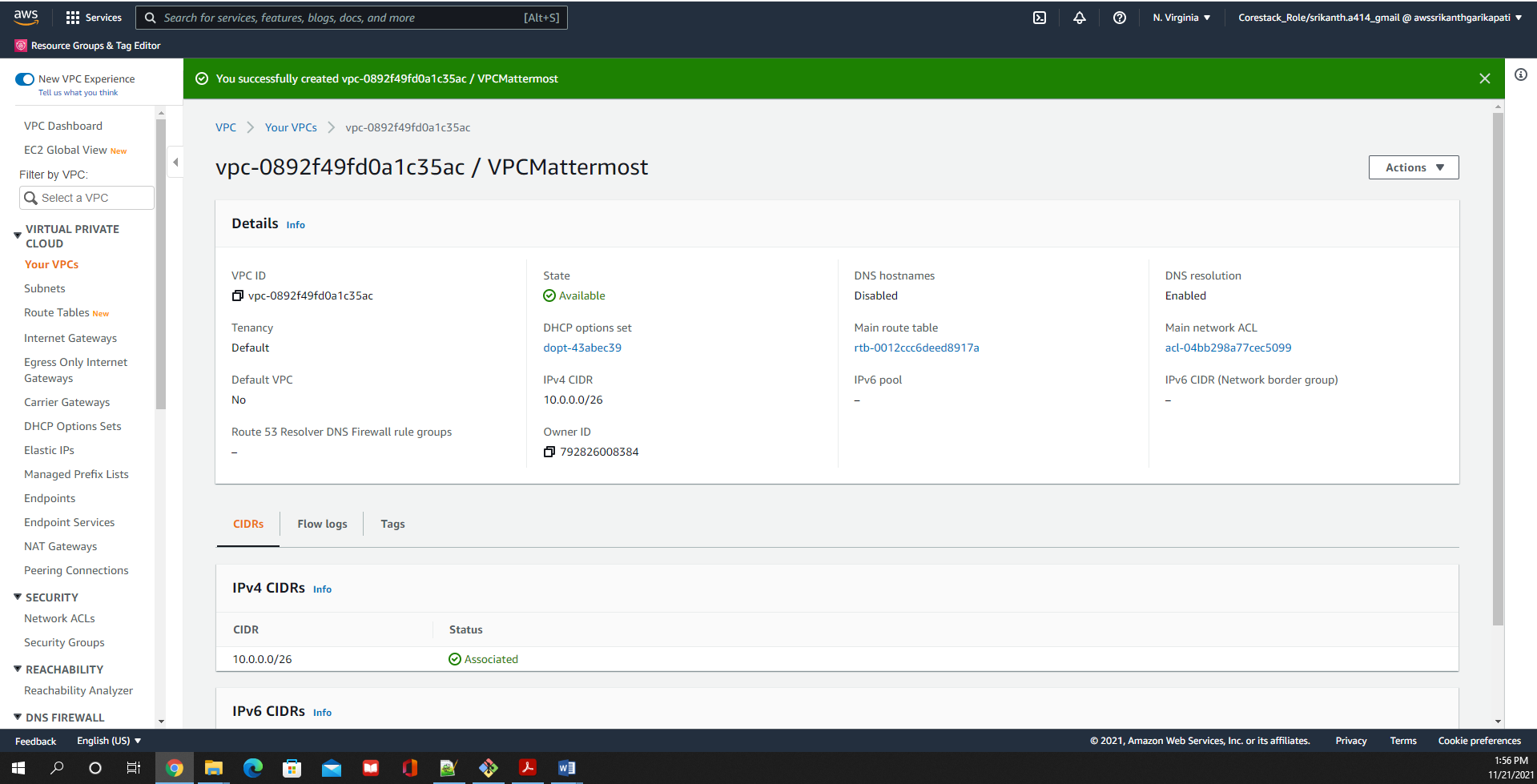
You are a Cloud engineer working on a Team Communication Solution project for a big MNC. The client has compliance policies which do not allow them to use services managed by third parties.  
The client wants a team communication solution that can be managed and hosted on servers that are controlled by them.

**Steps to Perform:**

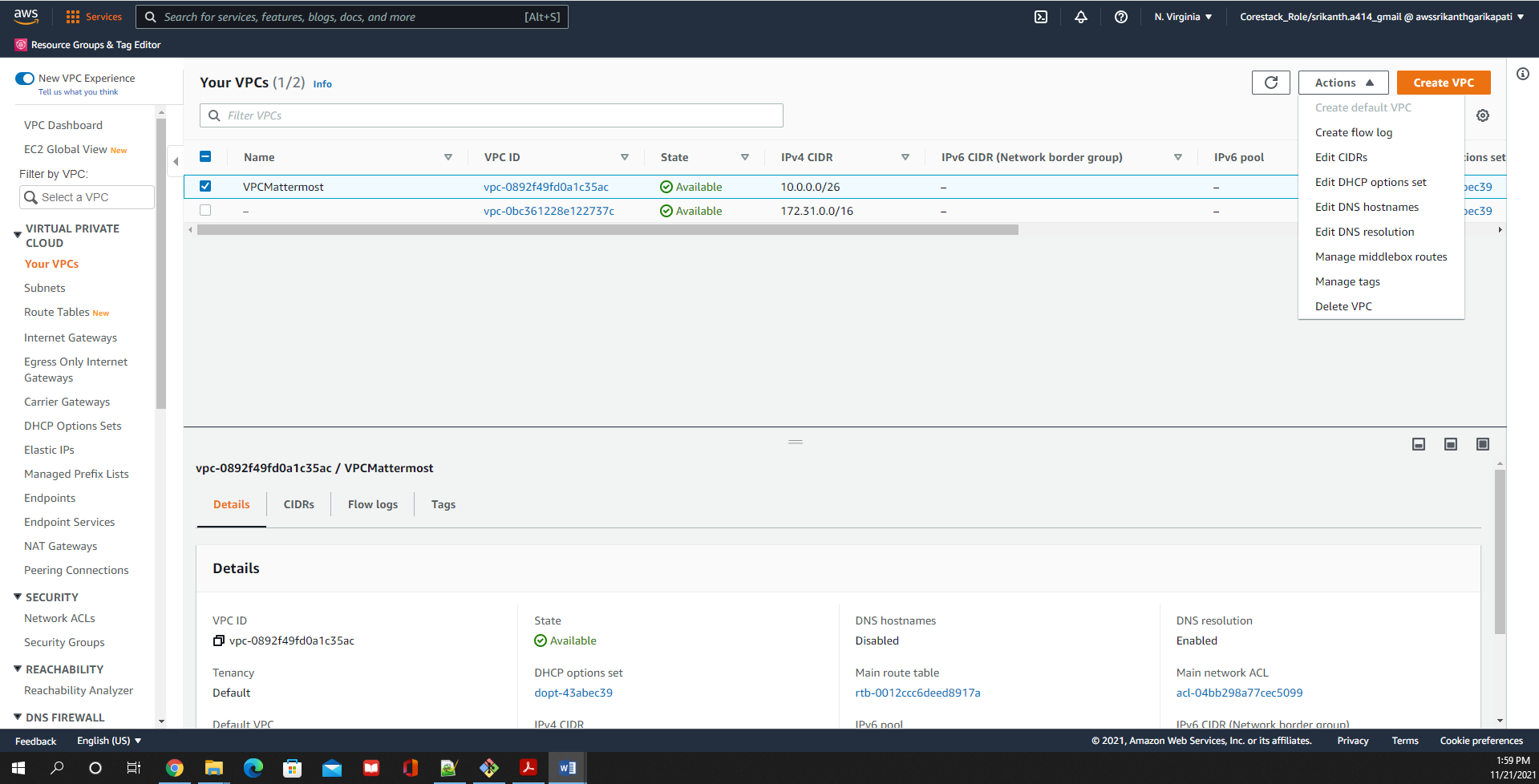
1. Implement two different subnets (one public and the other private) in a custom VPC
2. Install and configure MySQL on an Ubuntu 18.04 instance on the private subnet (Hint: Use a bastion host and a NAT instance)
3. Install and configure Mattermost on an Ubuntu 18.04 instance on the public subnet
4. Configure the security groups to allow the ports
5. Test the installation by accessing the IP of the public instance in a browser

Create VPC

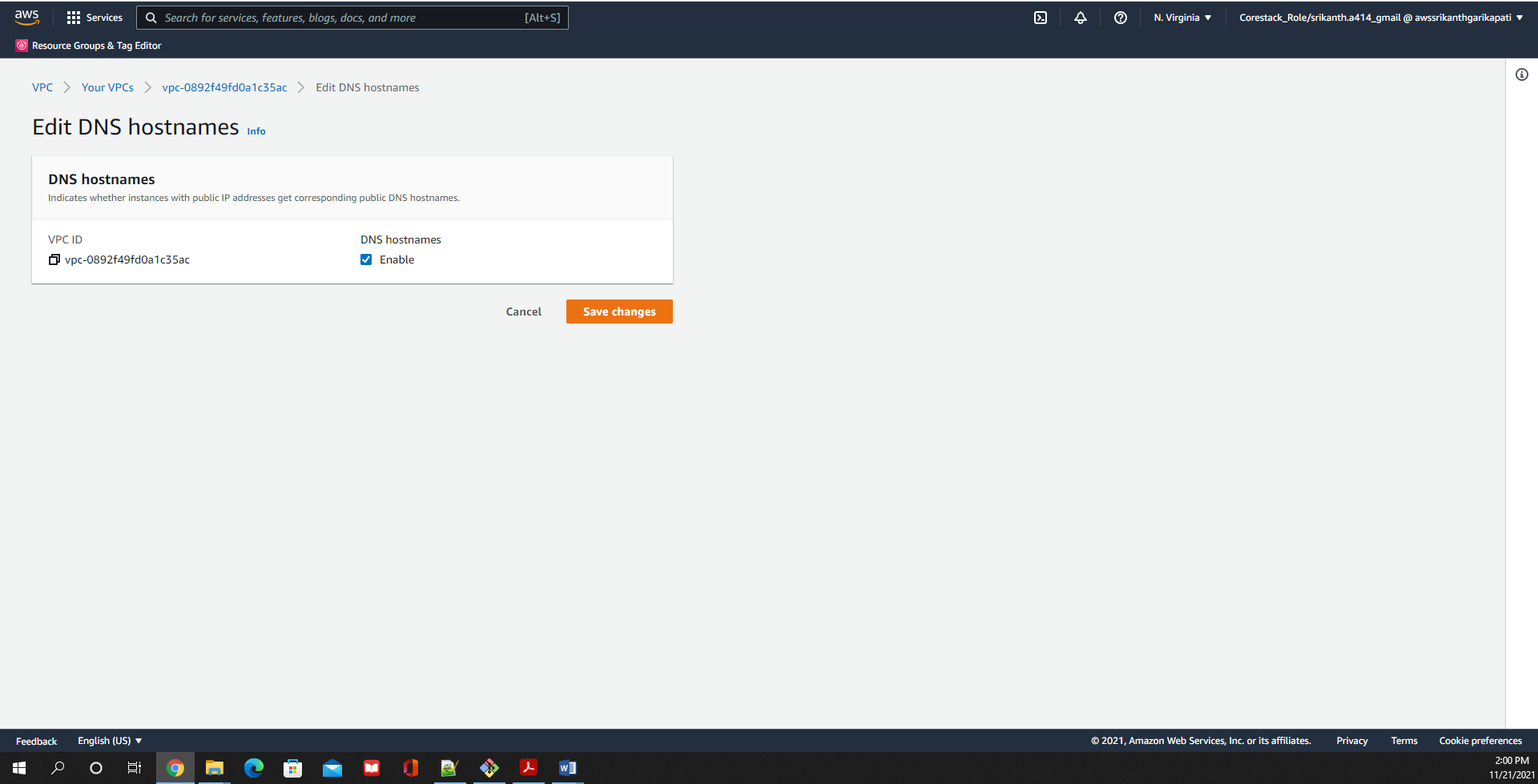




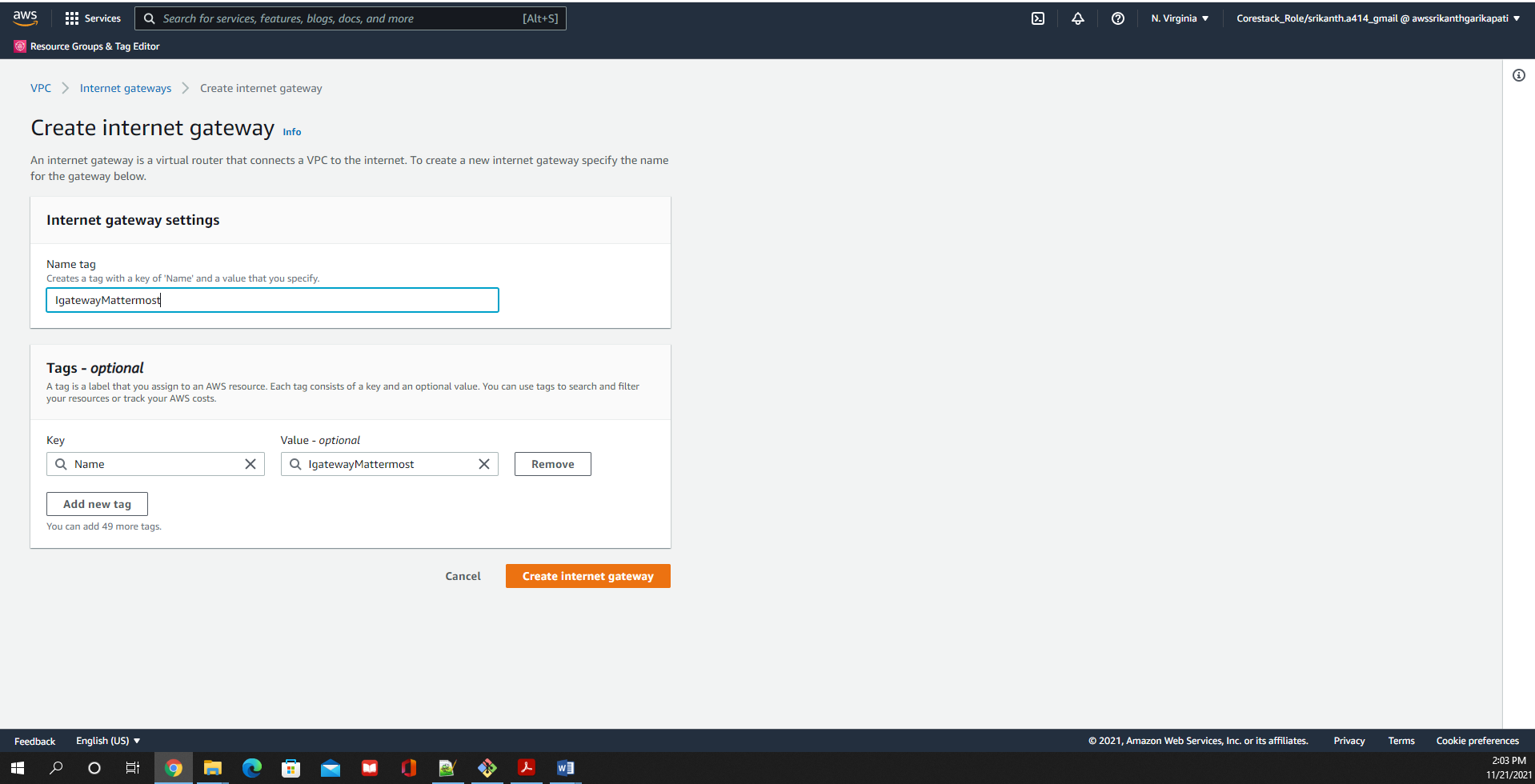
Edit DNSHostNames :



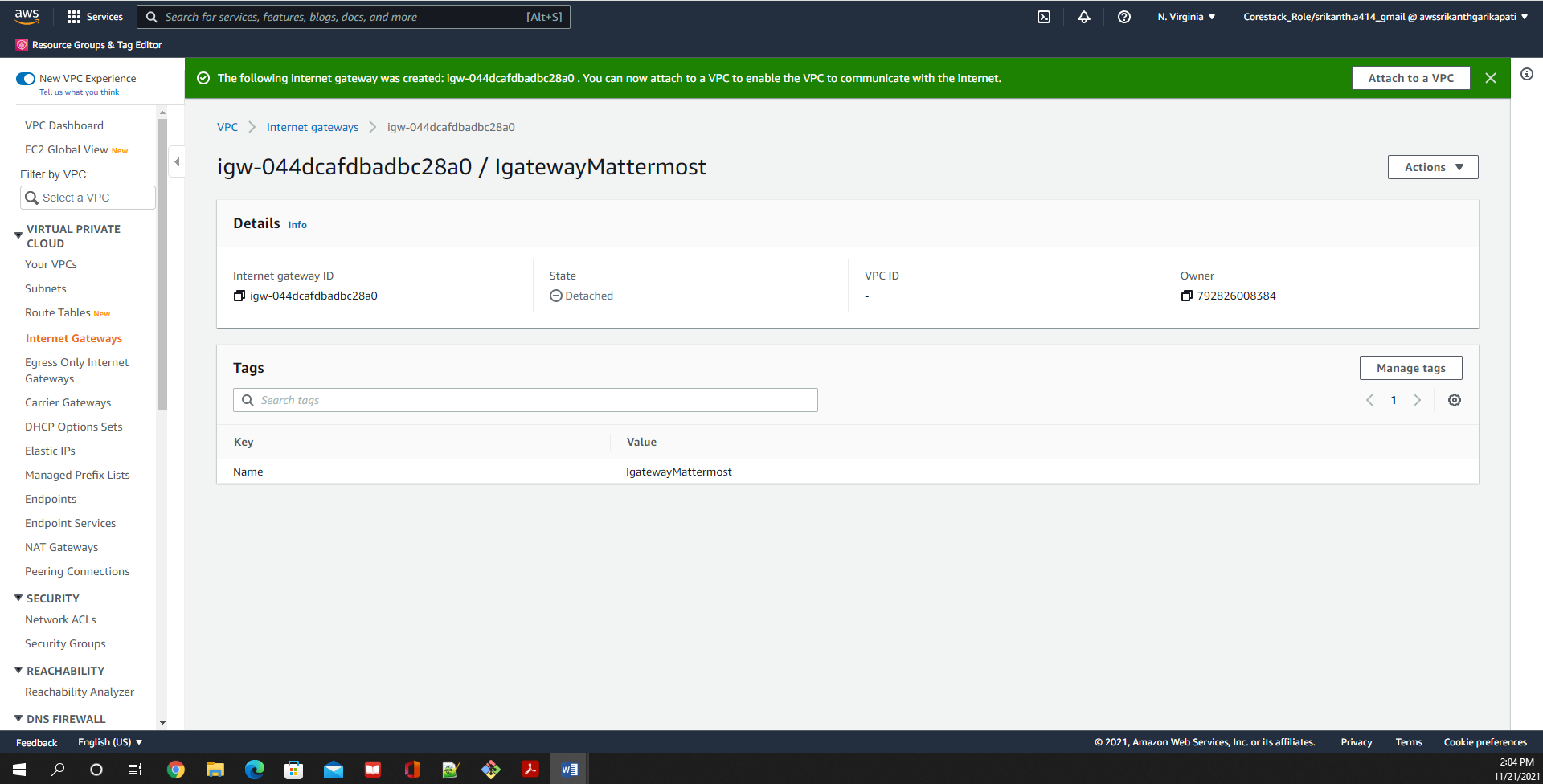
Enable DNSHostNames for the VPN

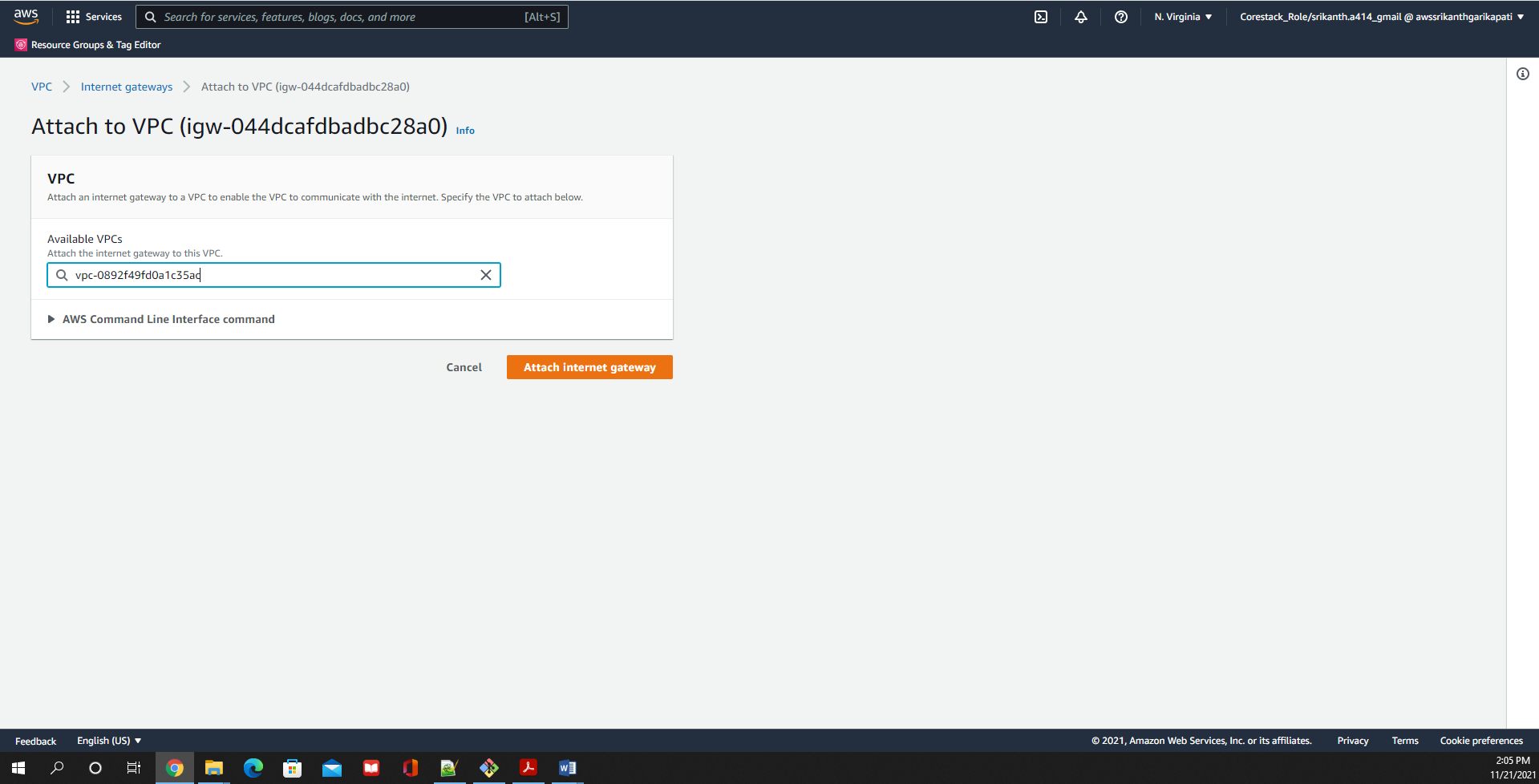


Create Internet Gateway



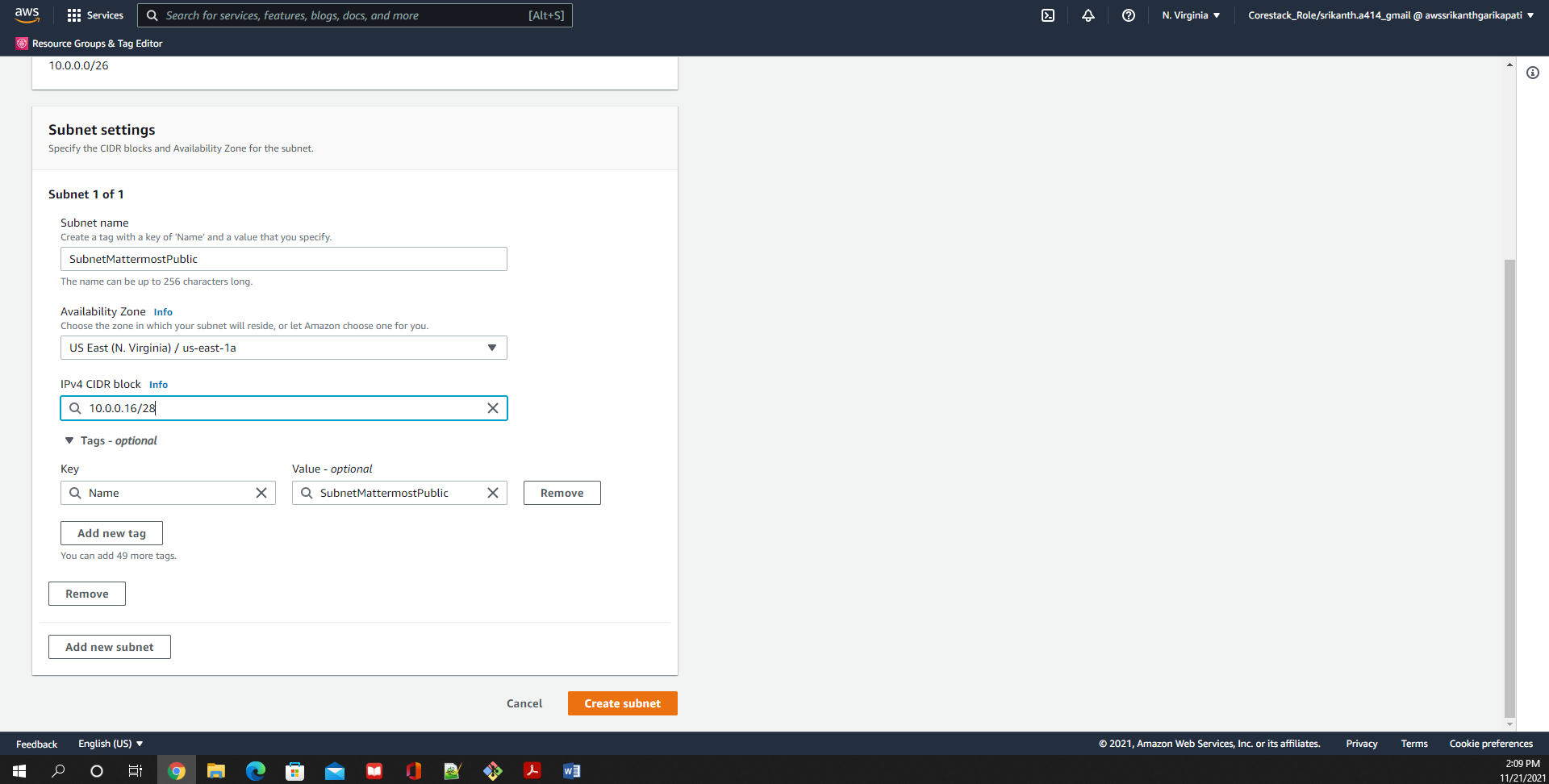
Attach Internet Gateway to VPC

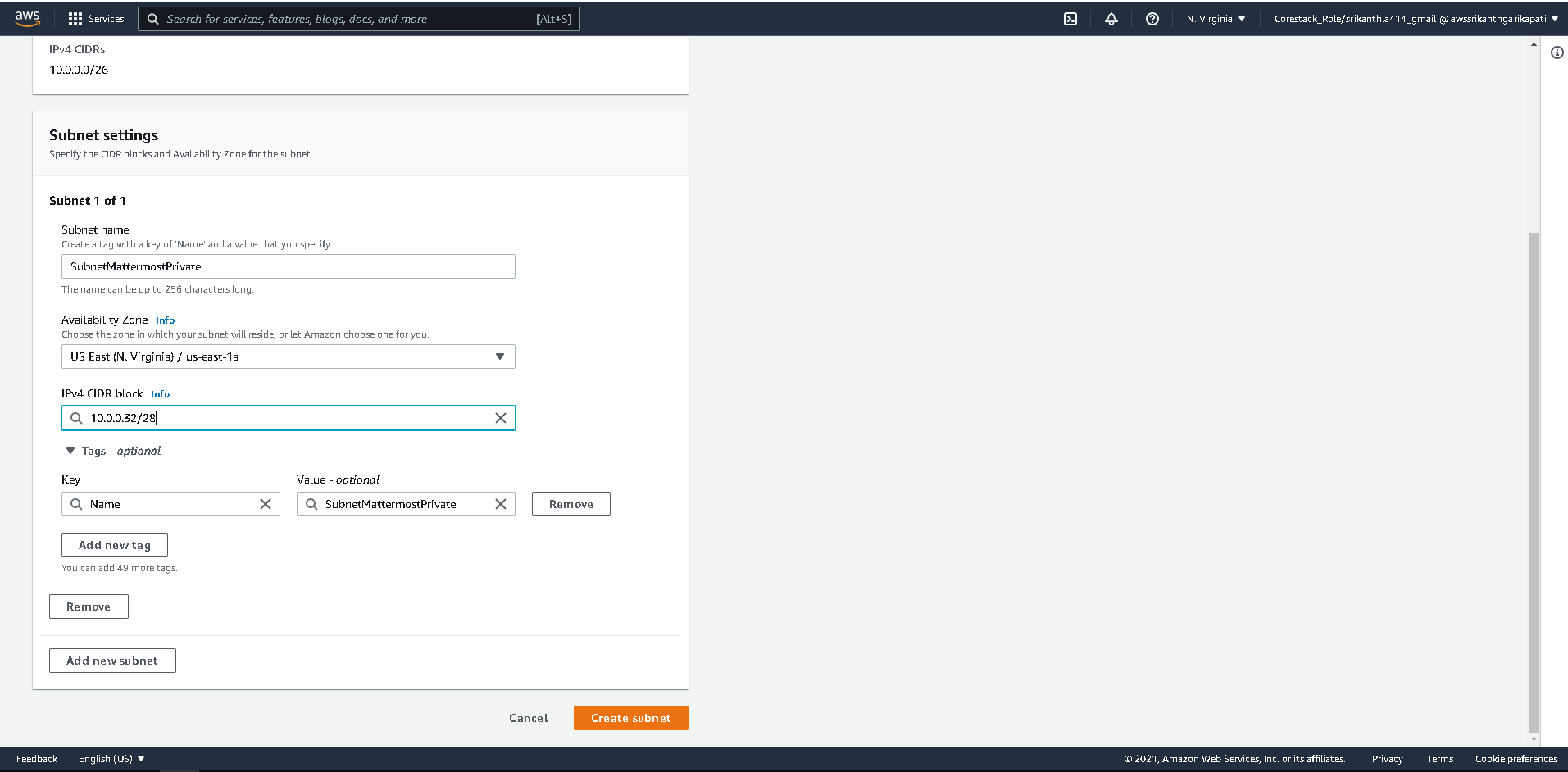


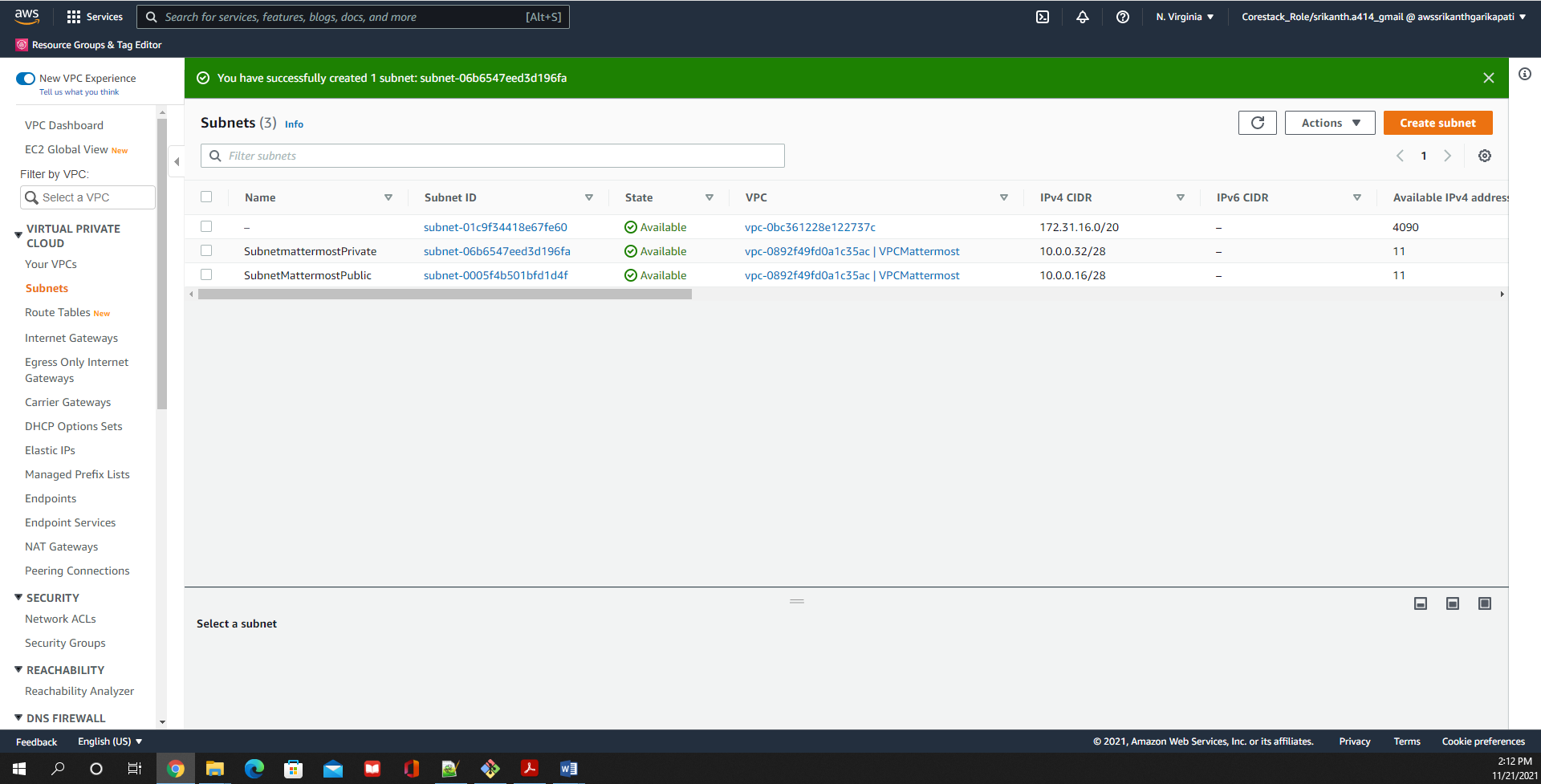




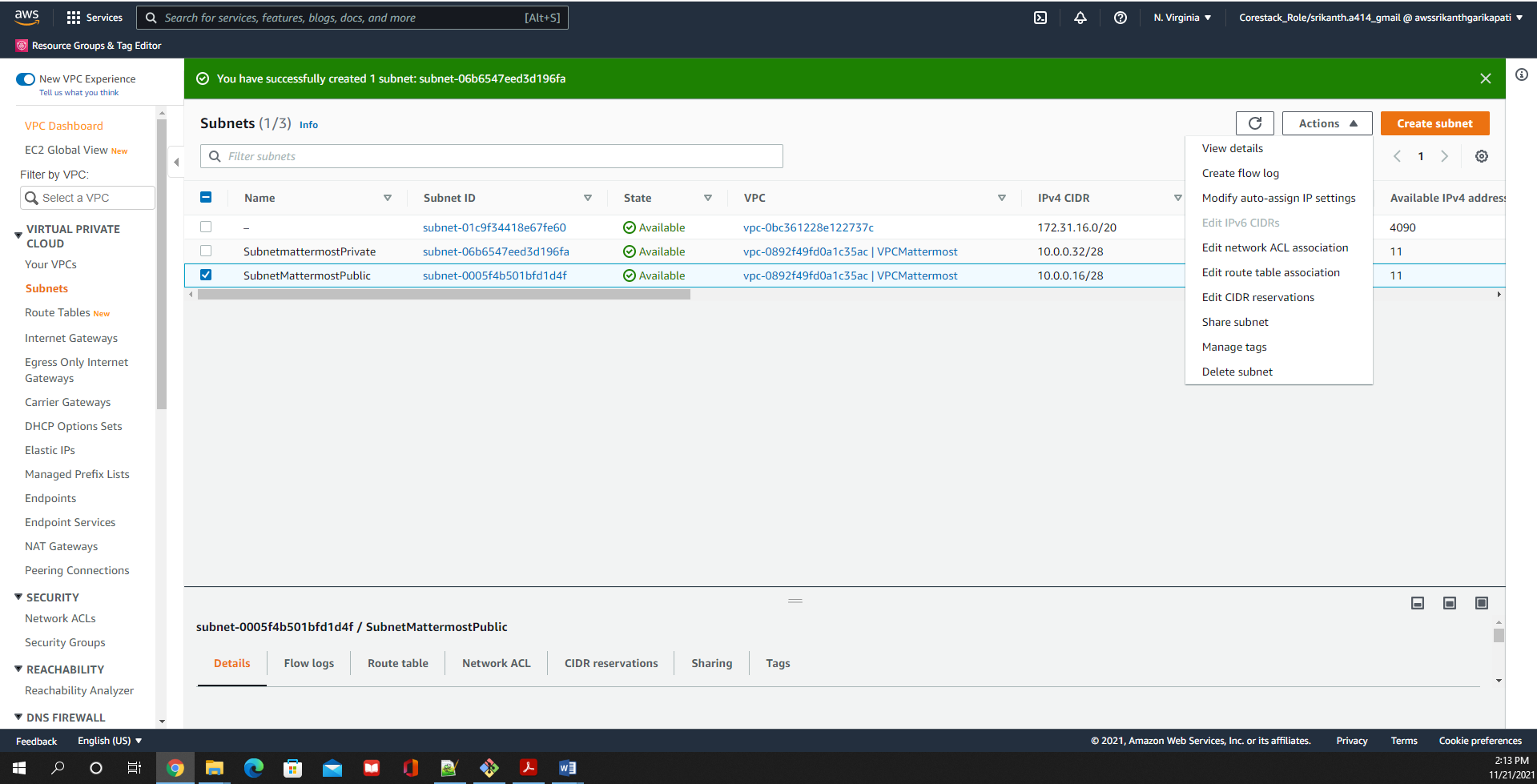
Create 2 subnets ( One Public and One Private )



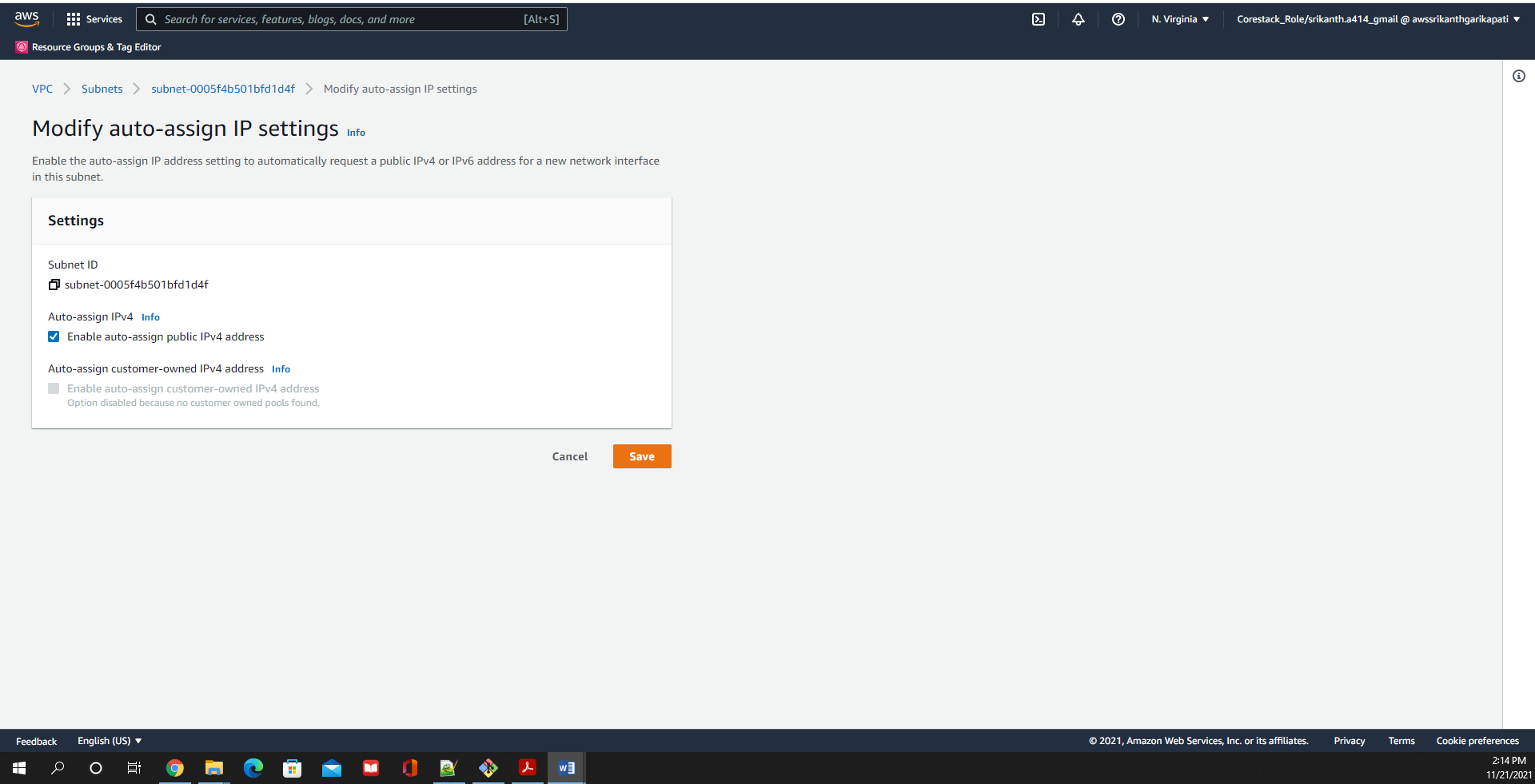




Make Public Subnet Public

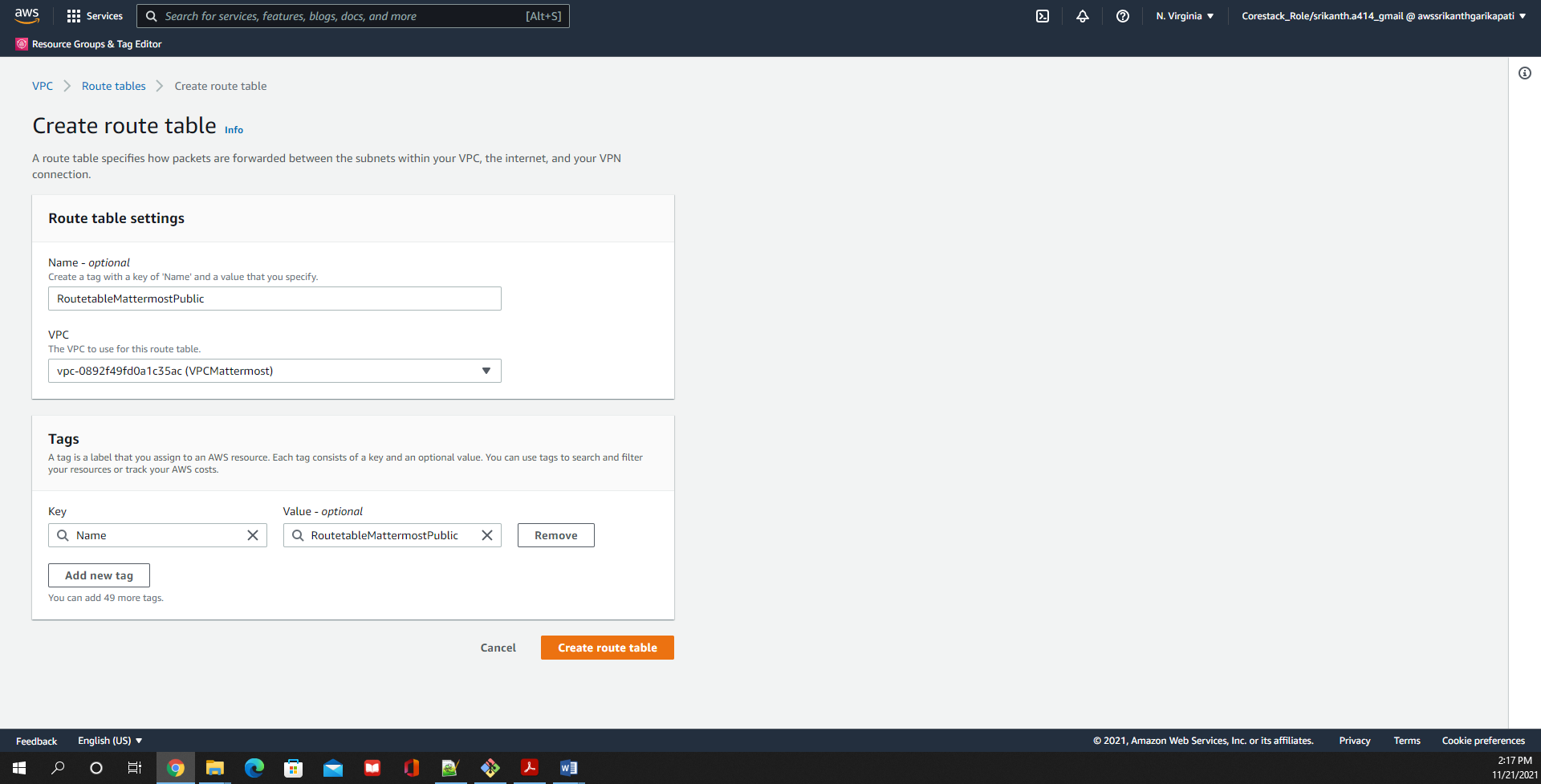


Enable Auto-Assign IPV4

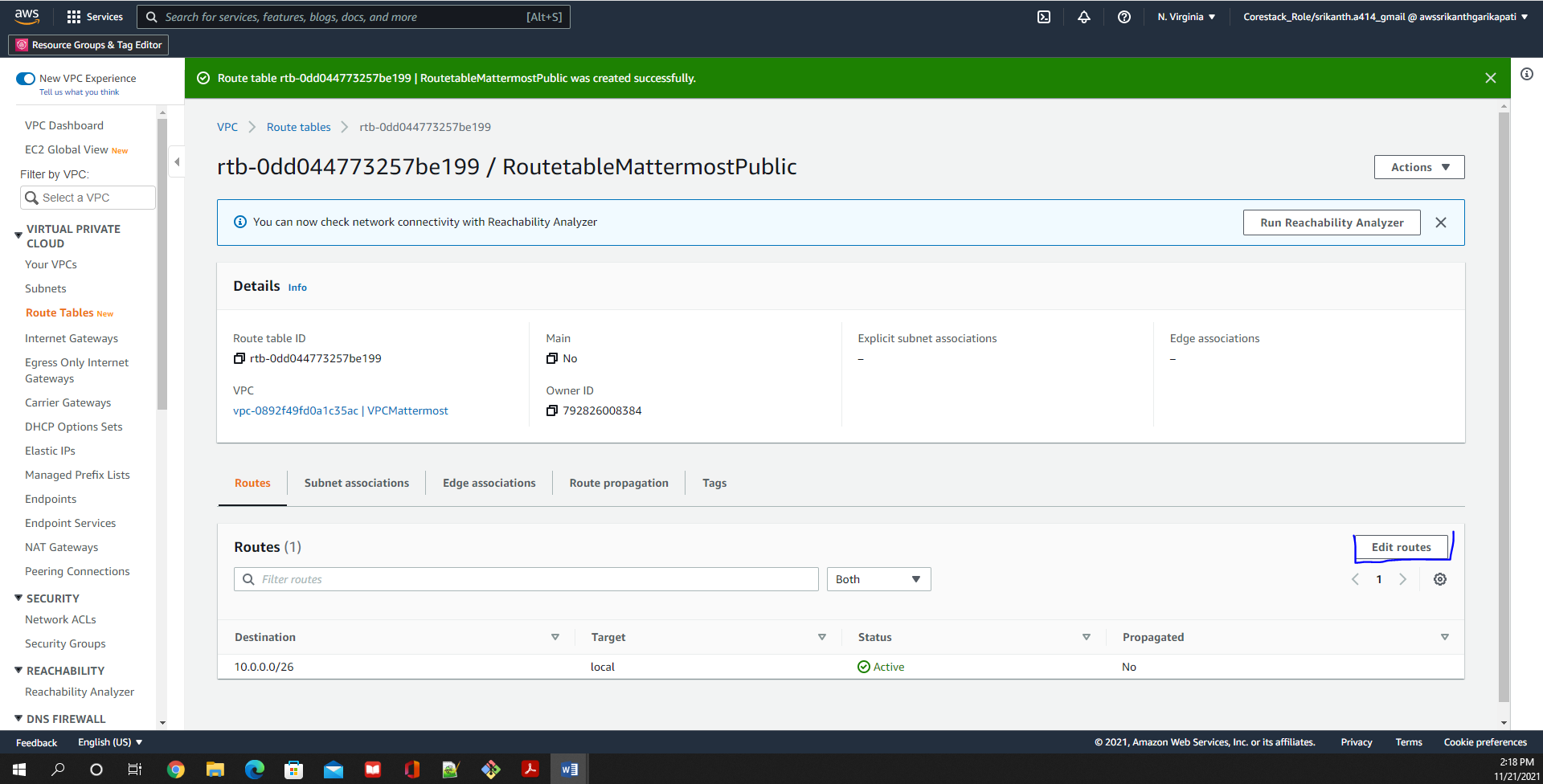


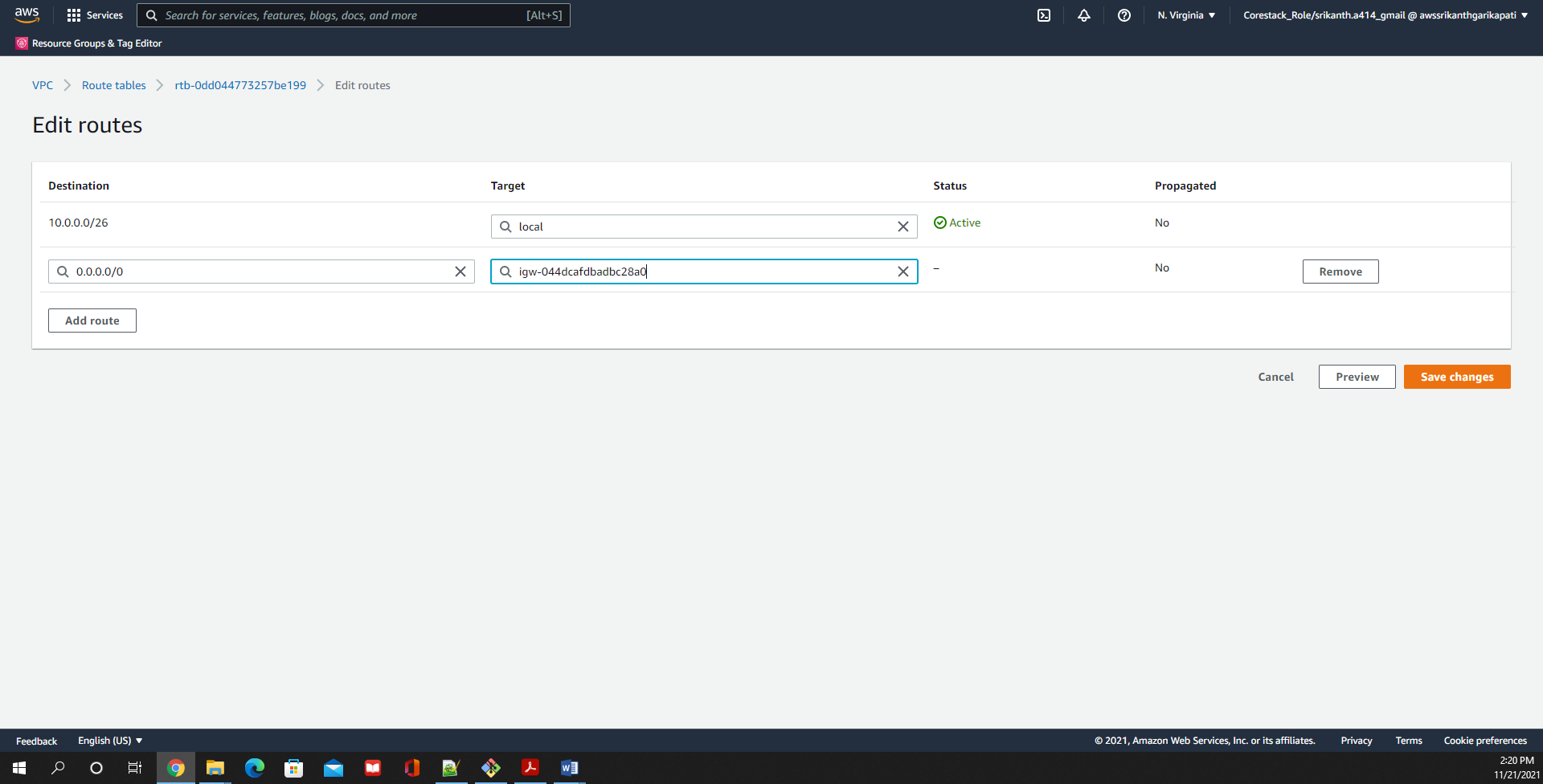
Create Route Tables and attach to Subnets

Create Route Table for Public Subnet and attach to Internet Gateway

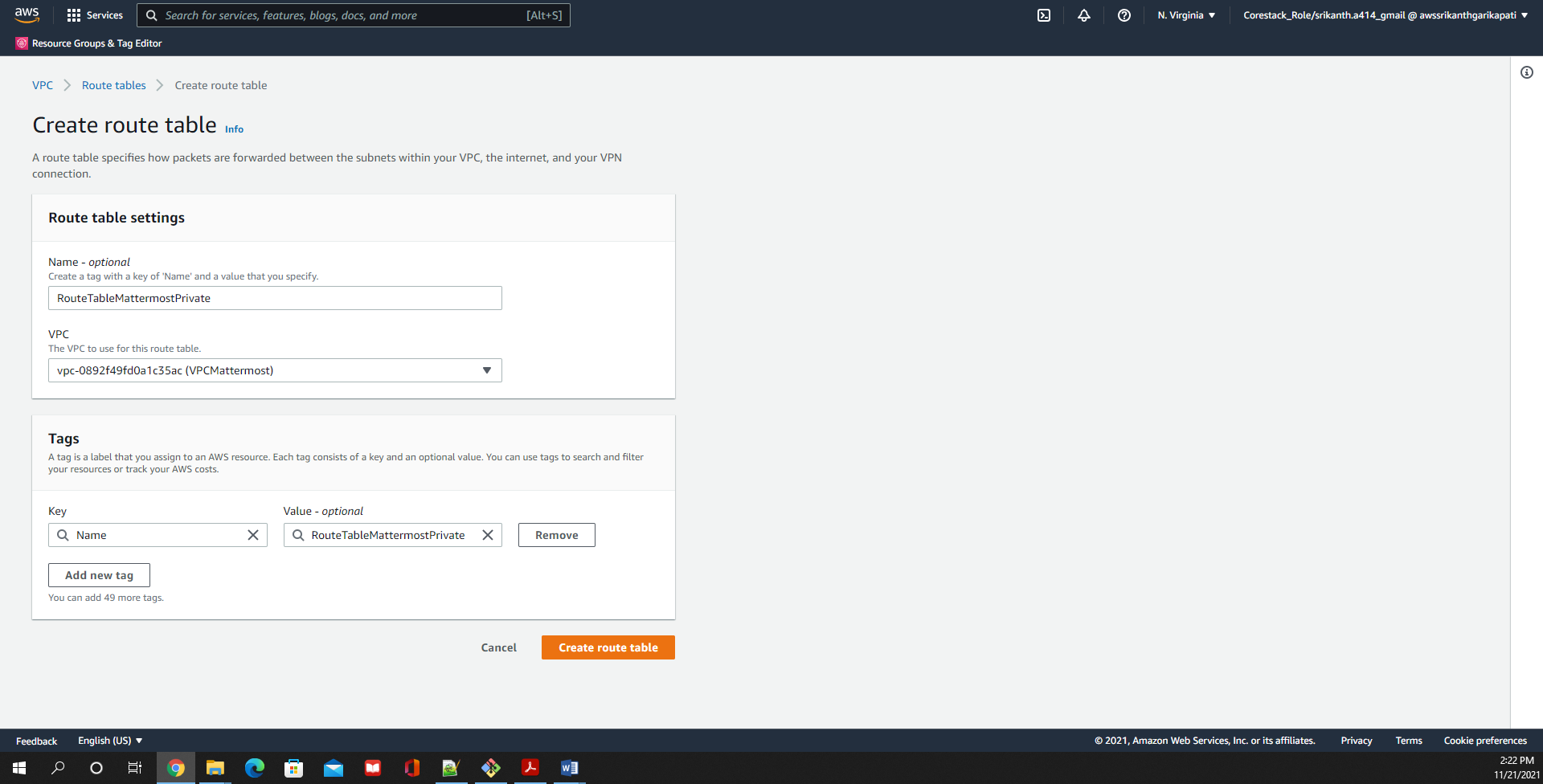


Route Public Route Table to Internet Gateway

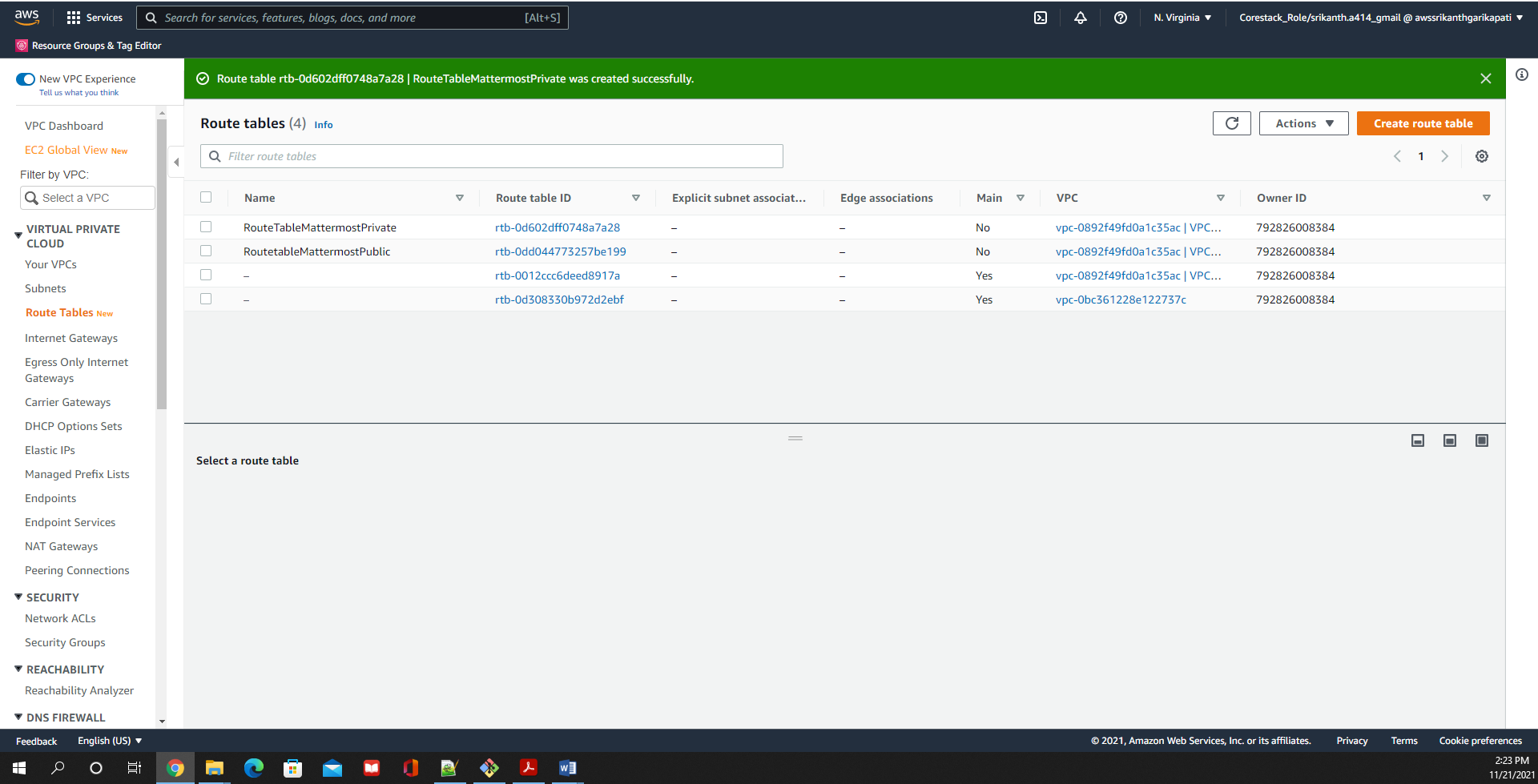




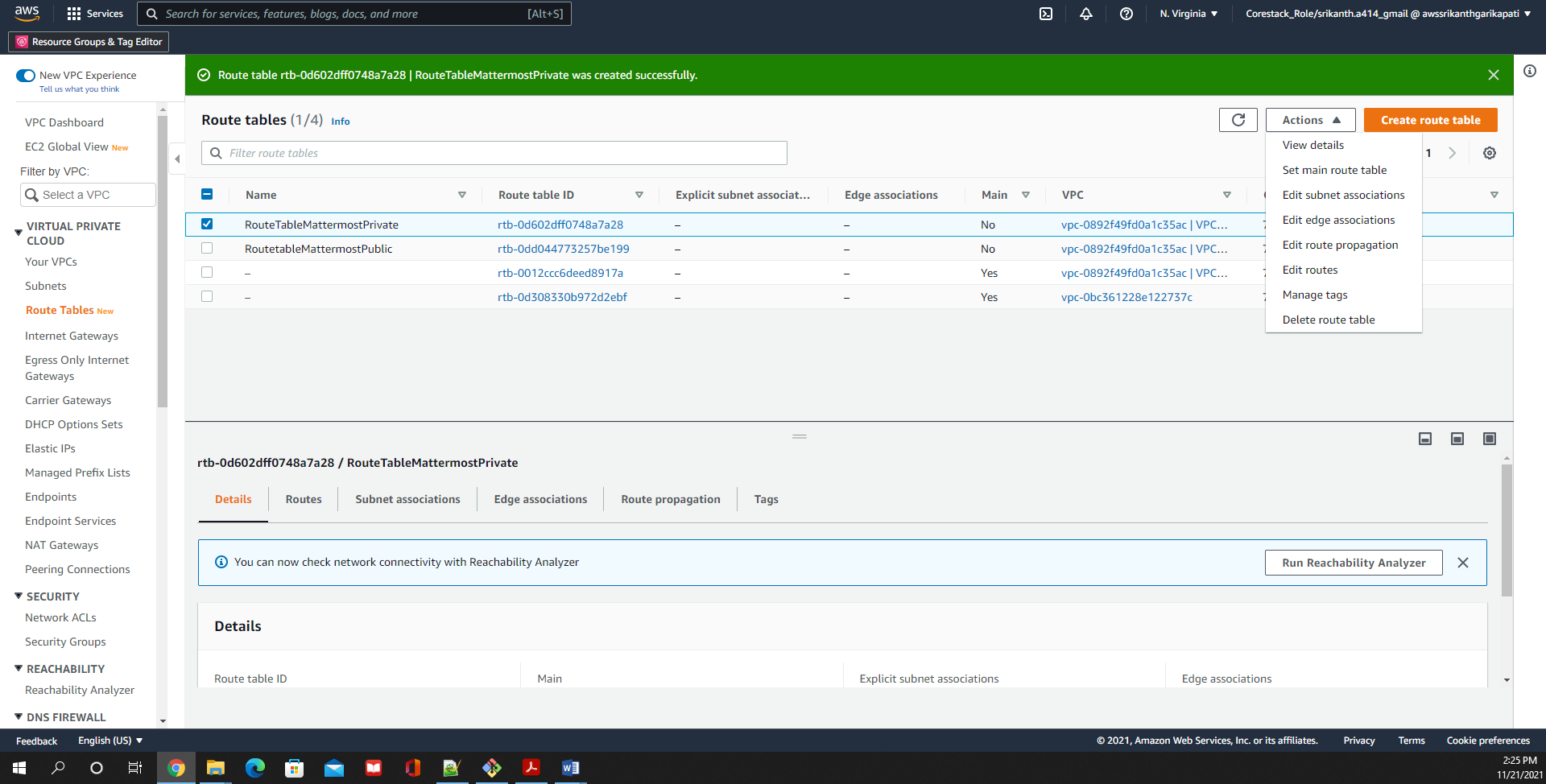
Create Route Table for Private Subnet

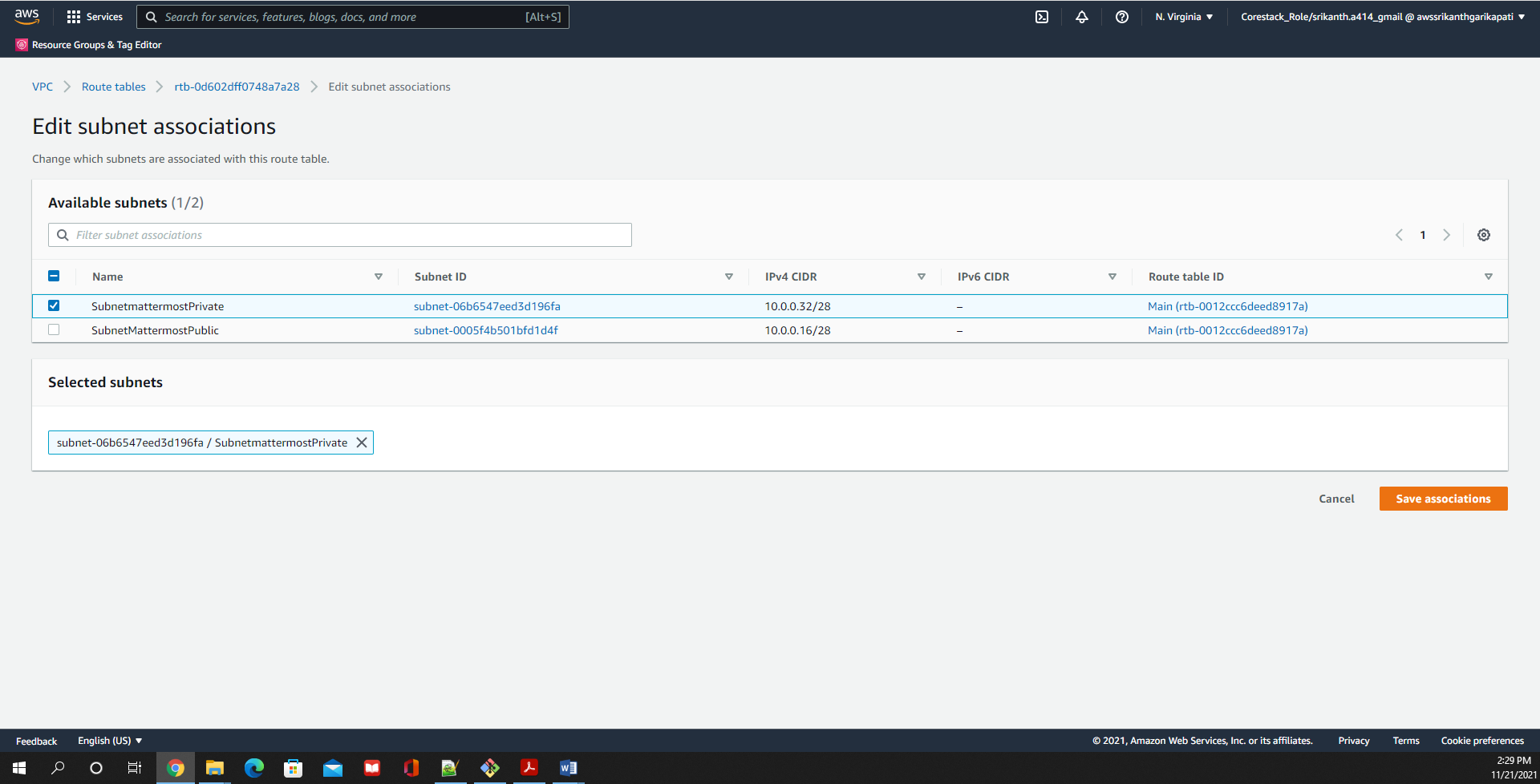


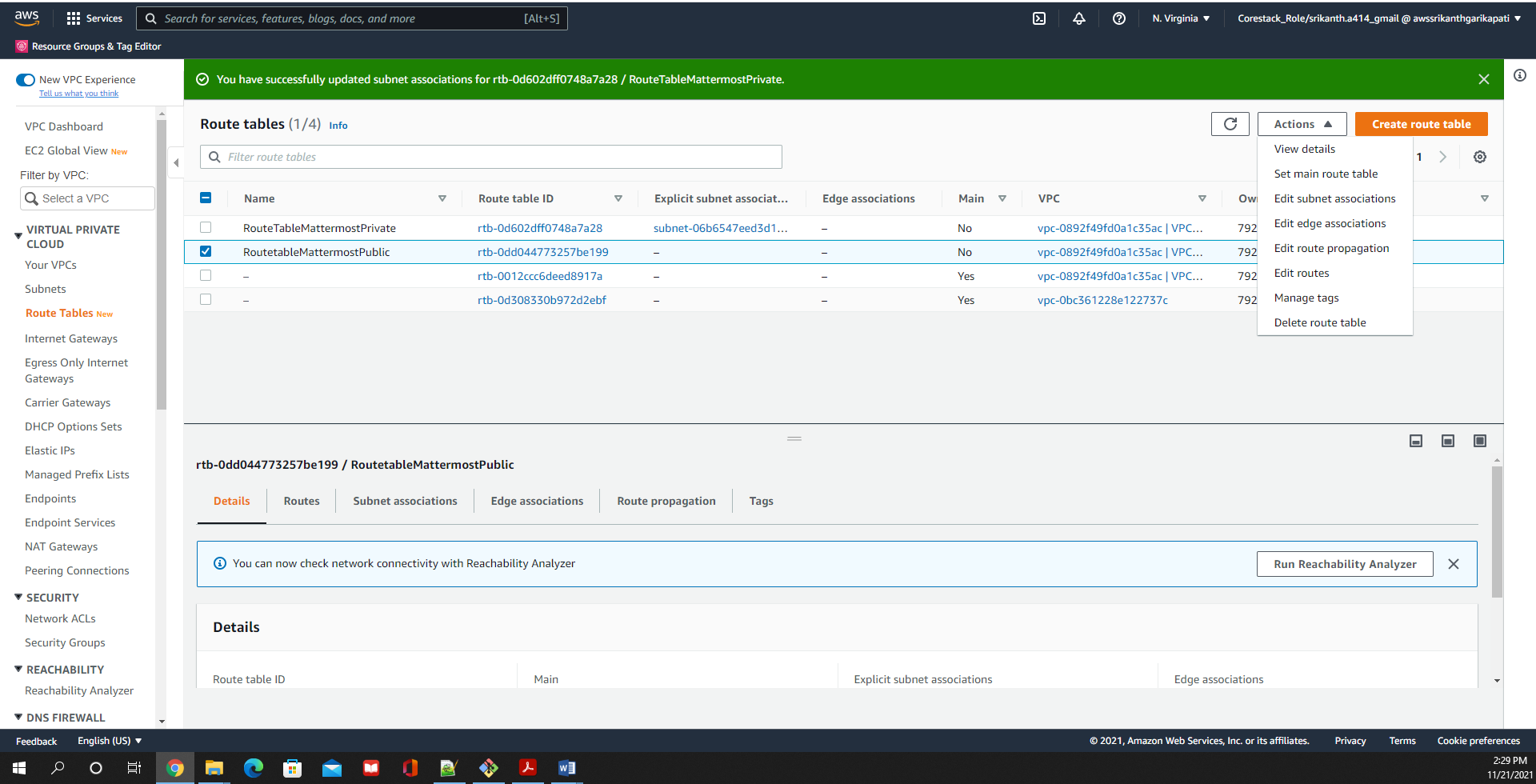
Public & Private Subnets along with Main Subnet are created



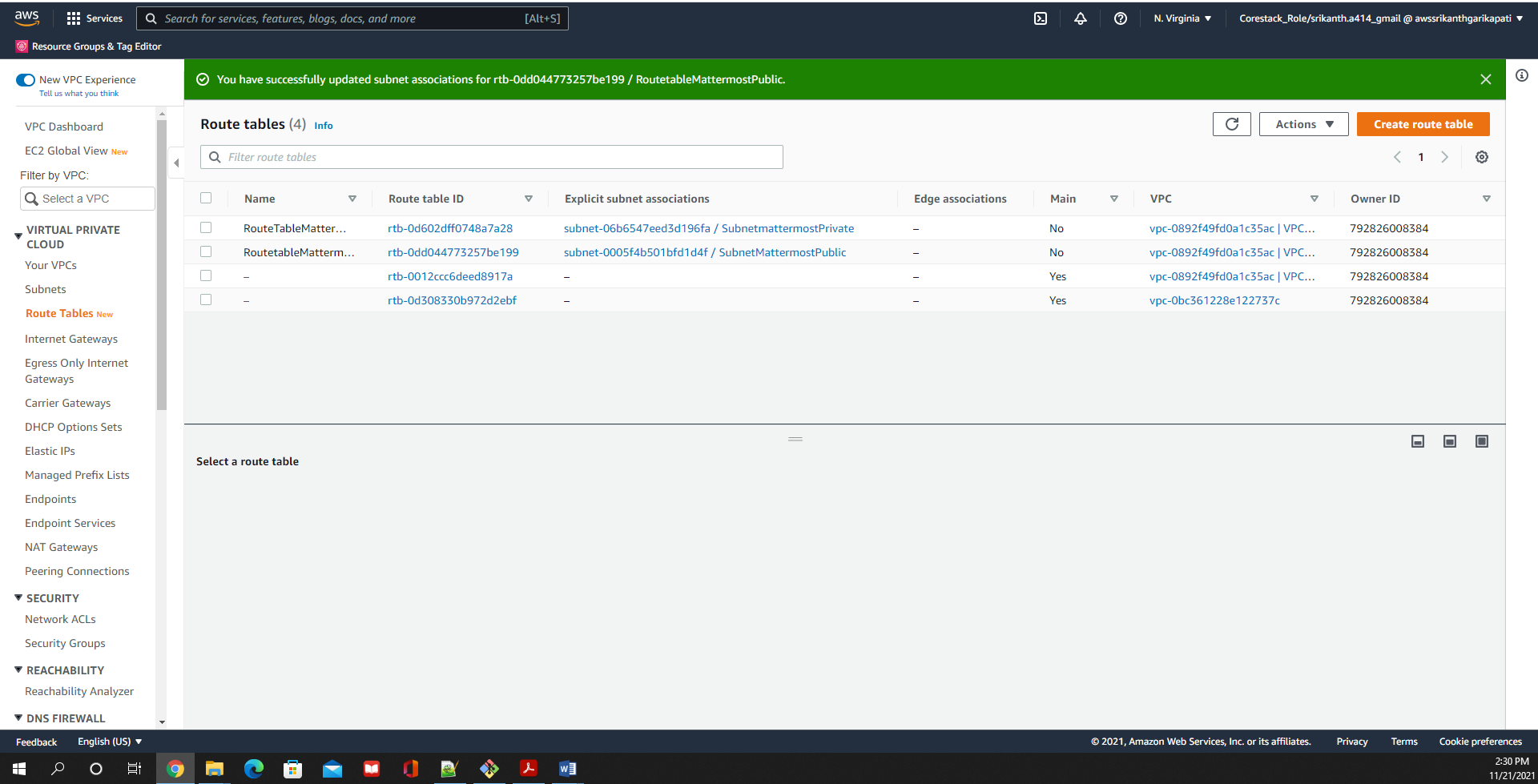
Associate the Route Tables to Public and Private Subnets respectively







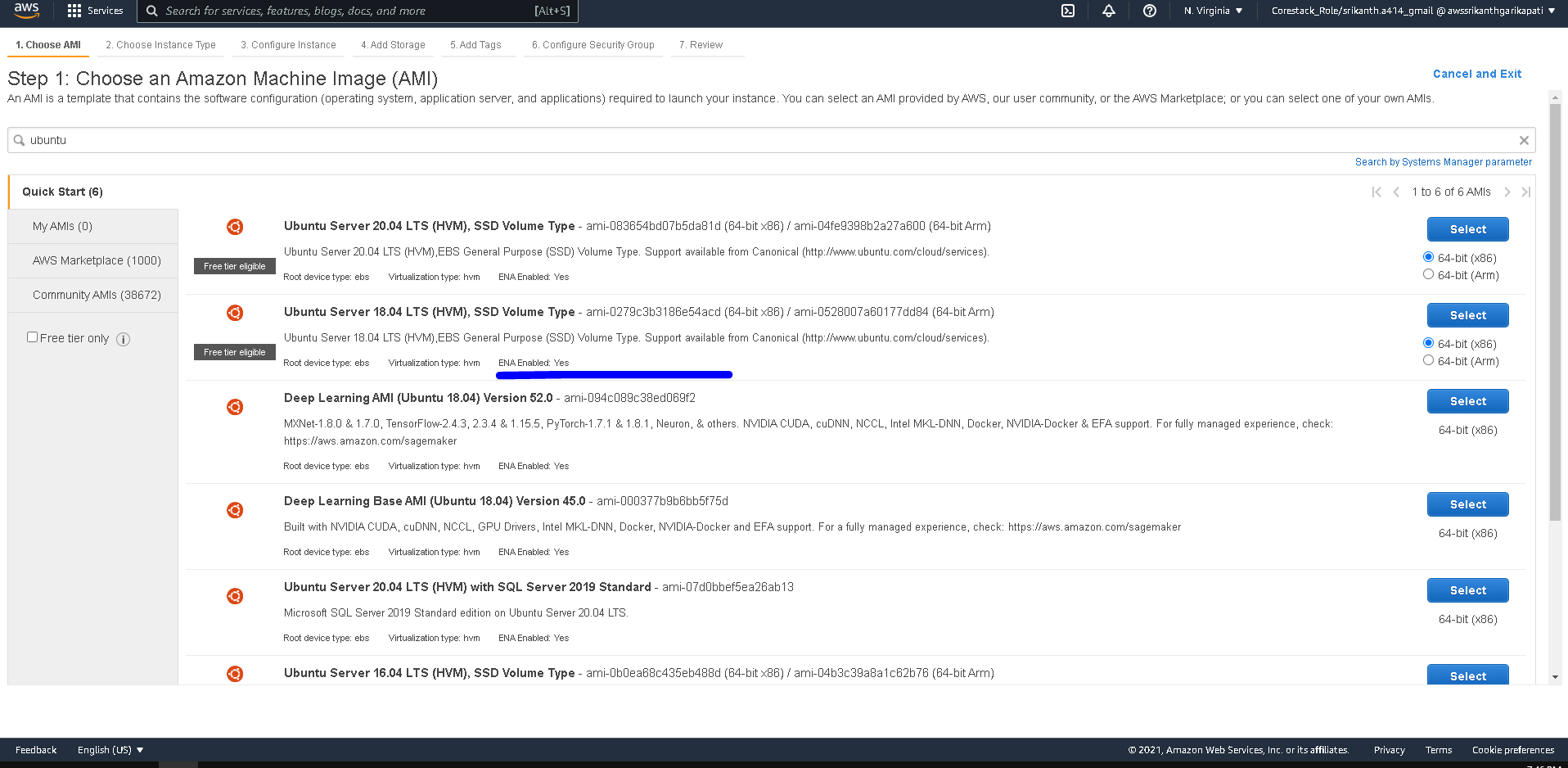
Subnets are associated to route table

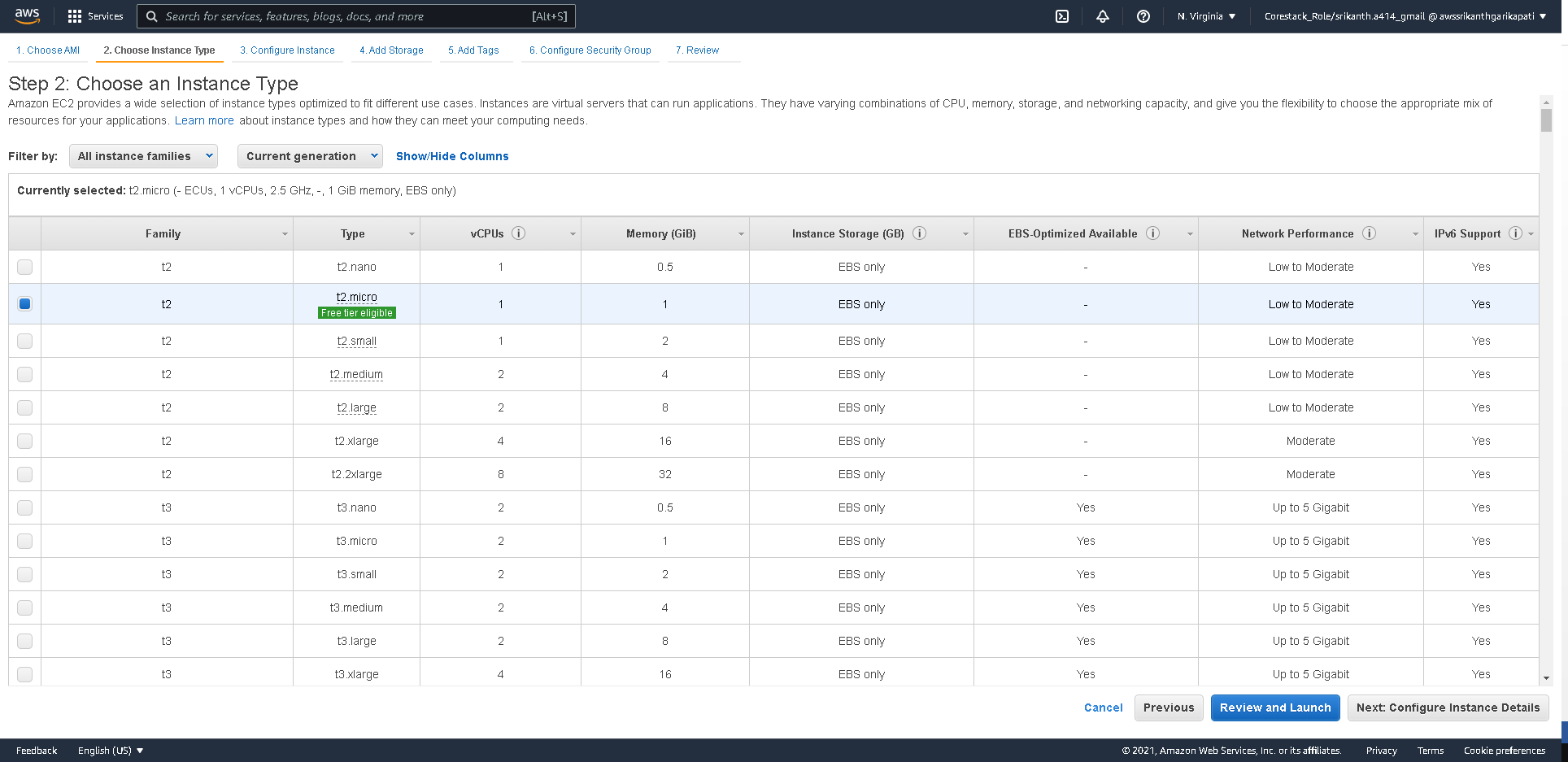


Launch 3 instances

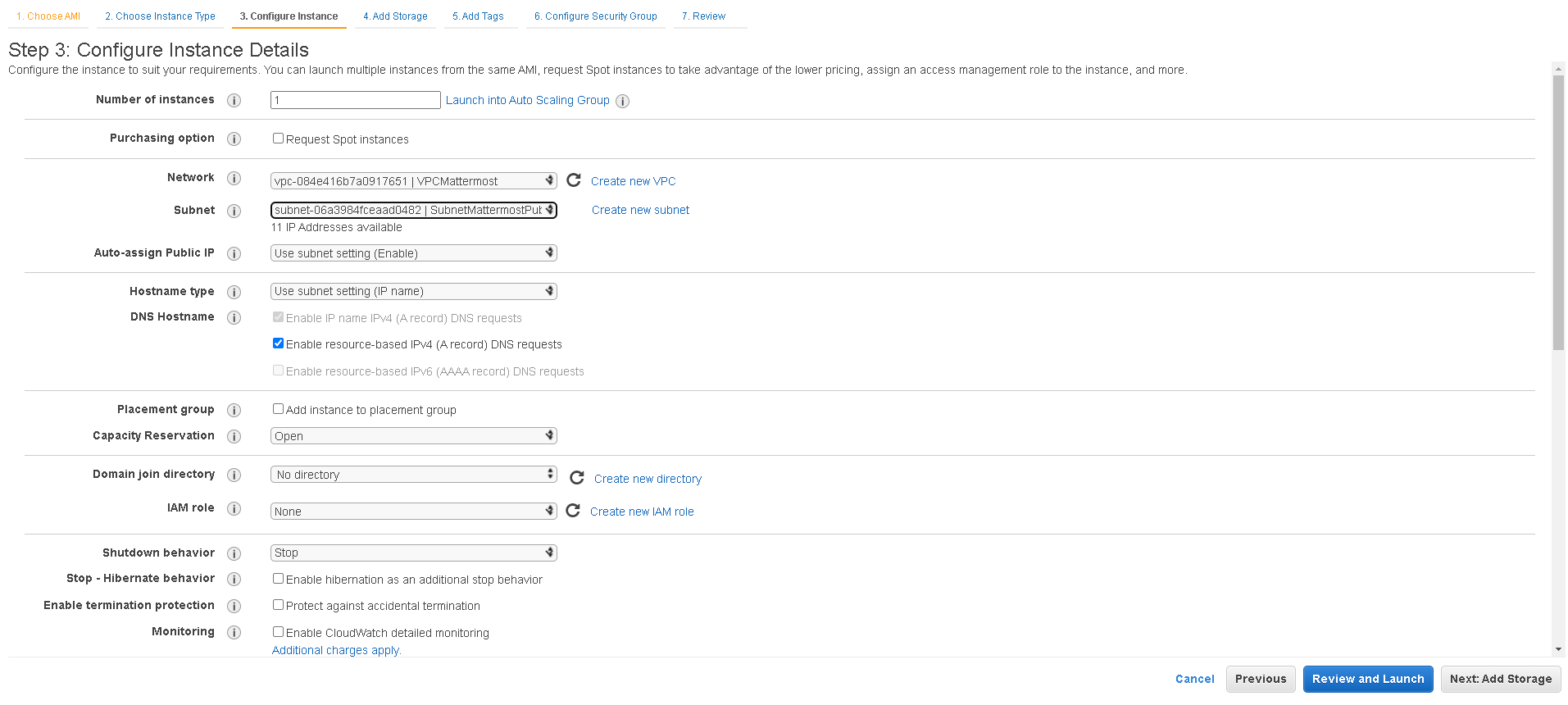
1. Web Instance to host Our Mattermost application
2. DN Instance to host mysql/AuroraDB
3. NAT Instance on Public Subnet

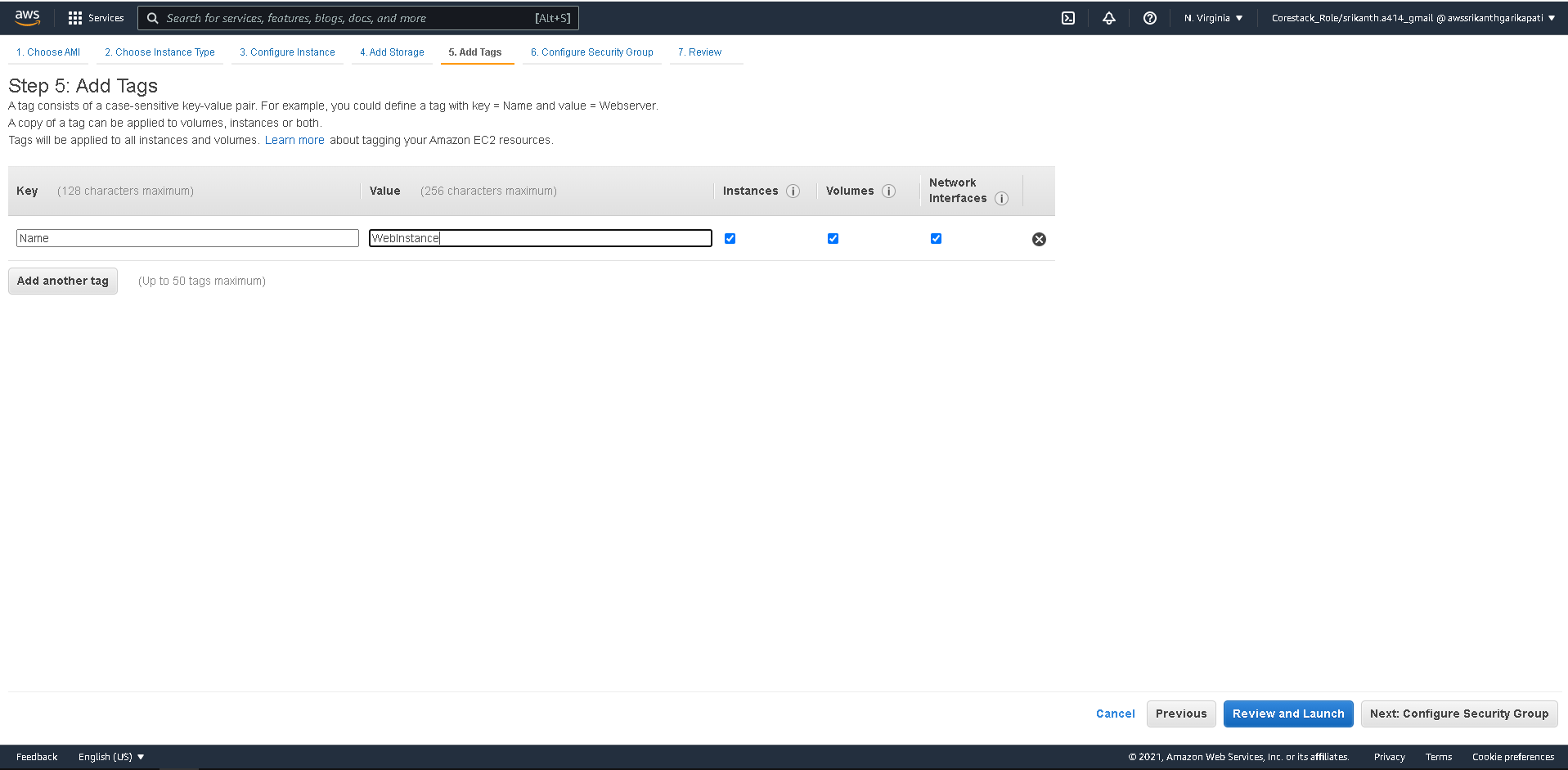
For WebInstance in Public Subnet with UBUNTU 18.04 AMI



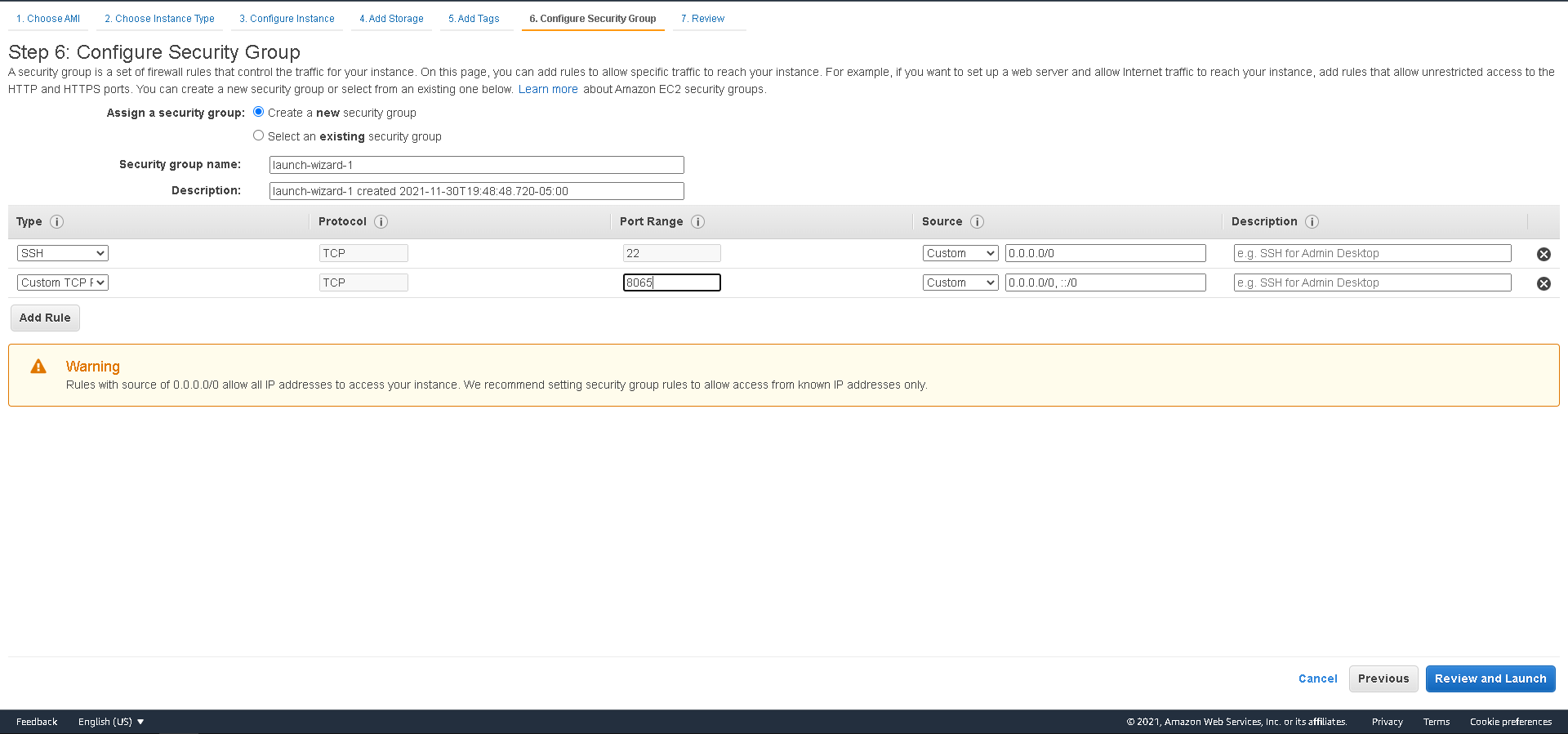


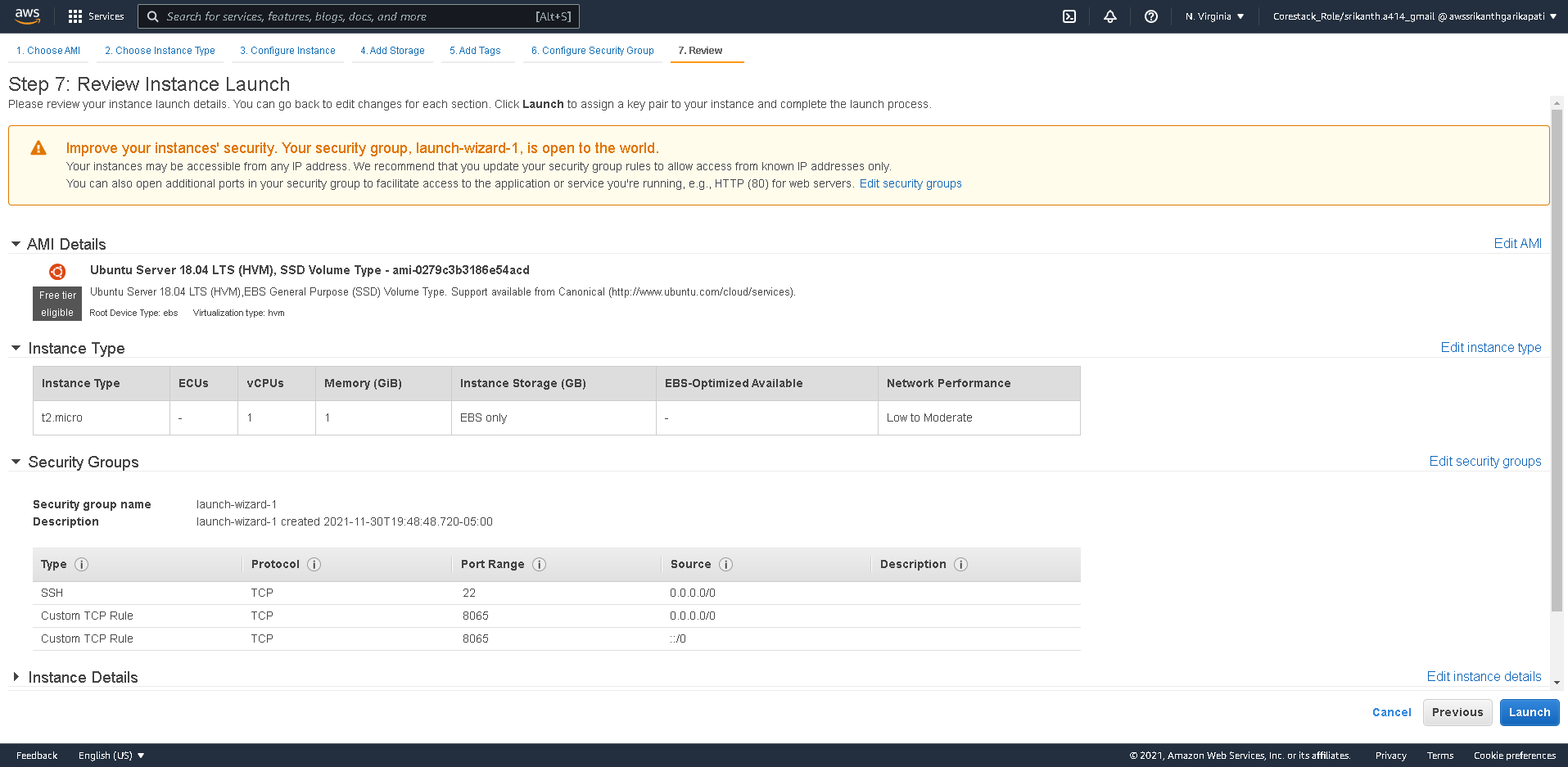
Make sure to launch WebInstance In Public Subnet

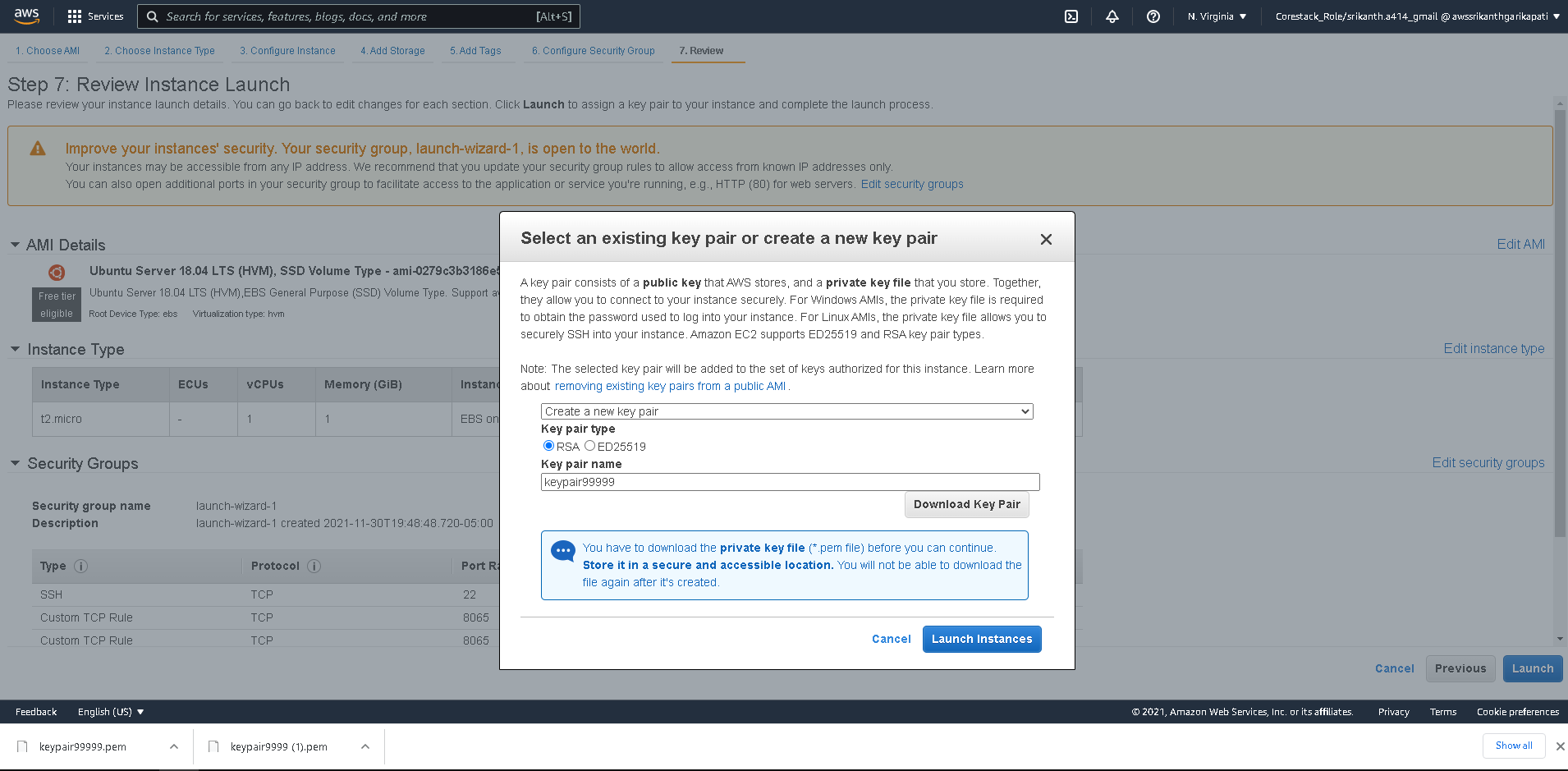




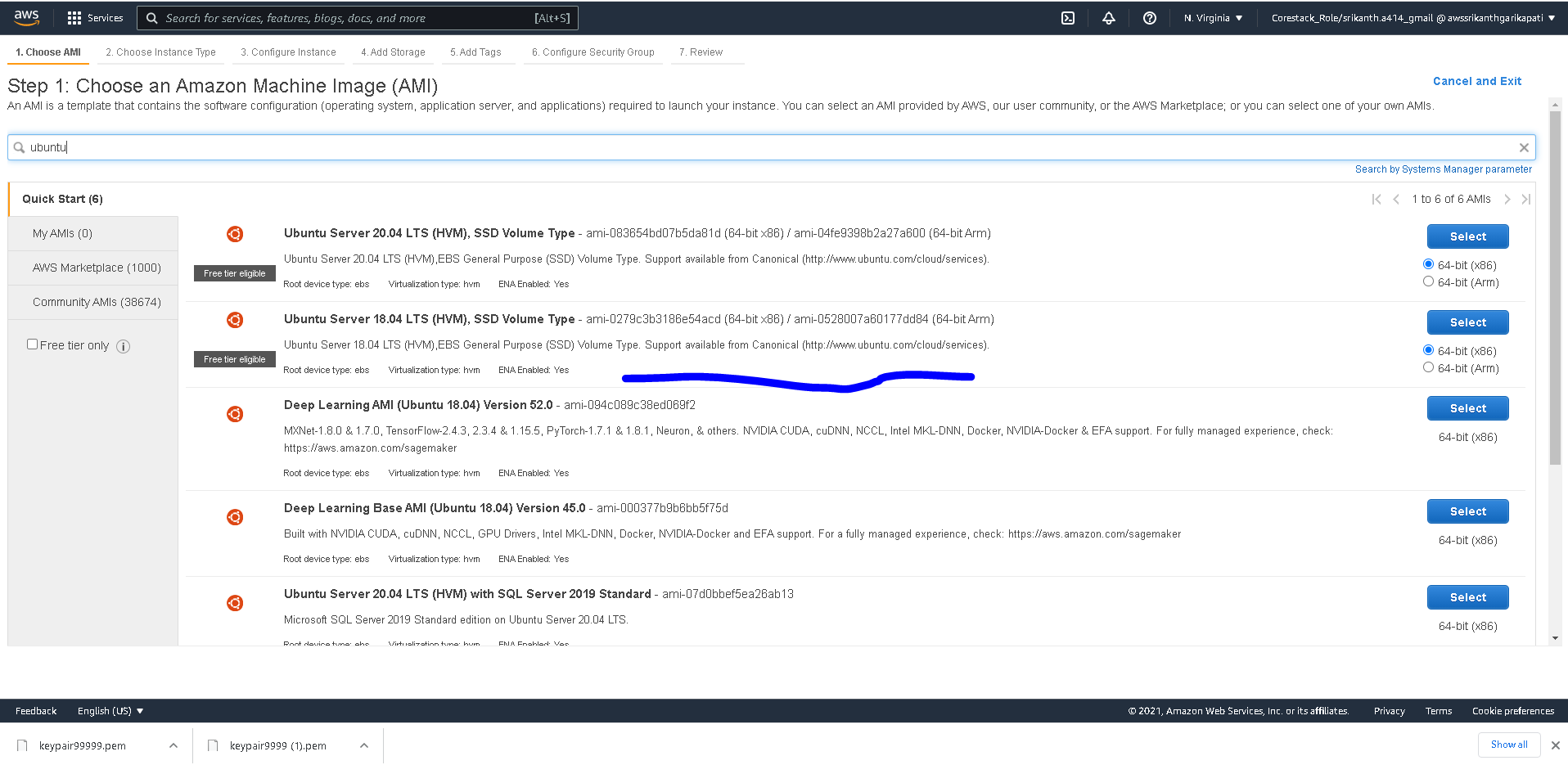
Configure a new Security Group Matter most Application on this Web-instance to listen to Port 8065

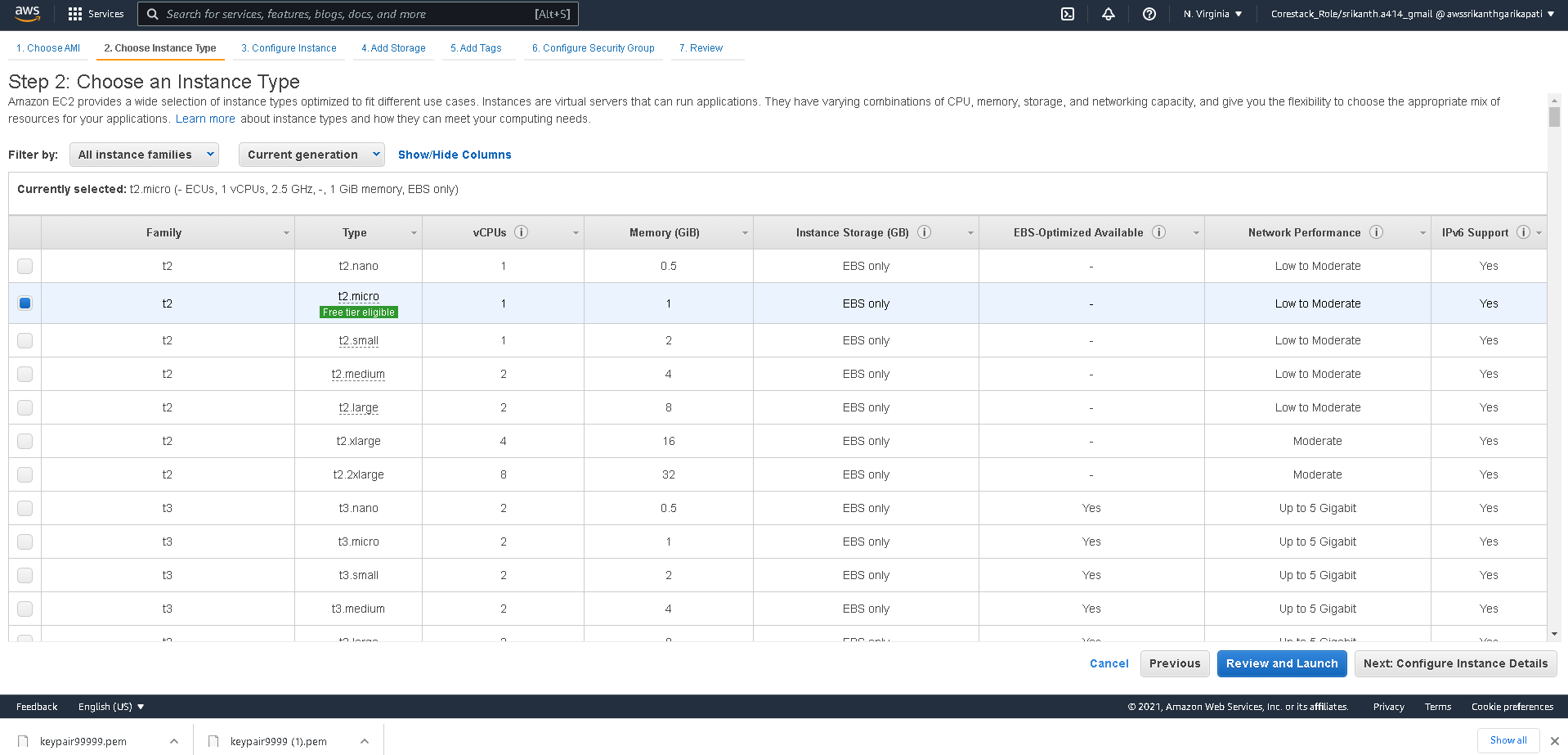






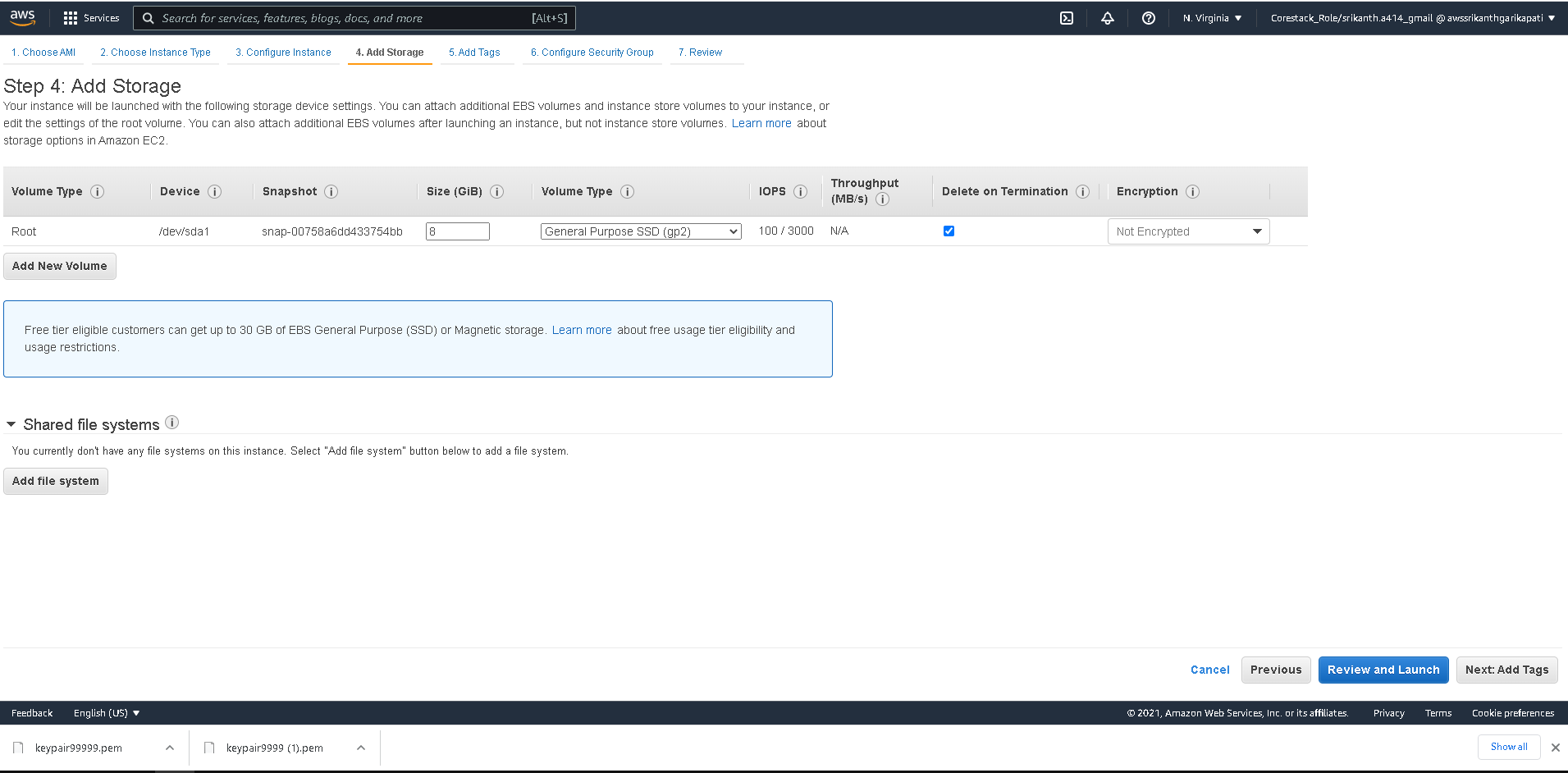
Launch DB-Instance in Private Subnet



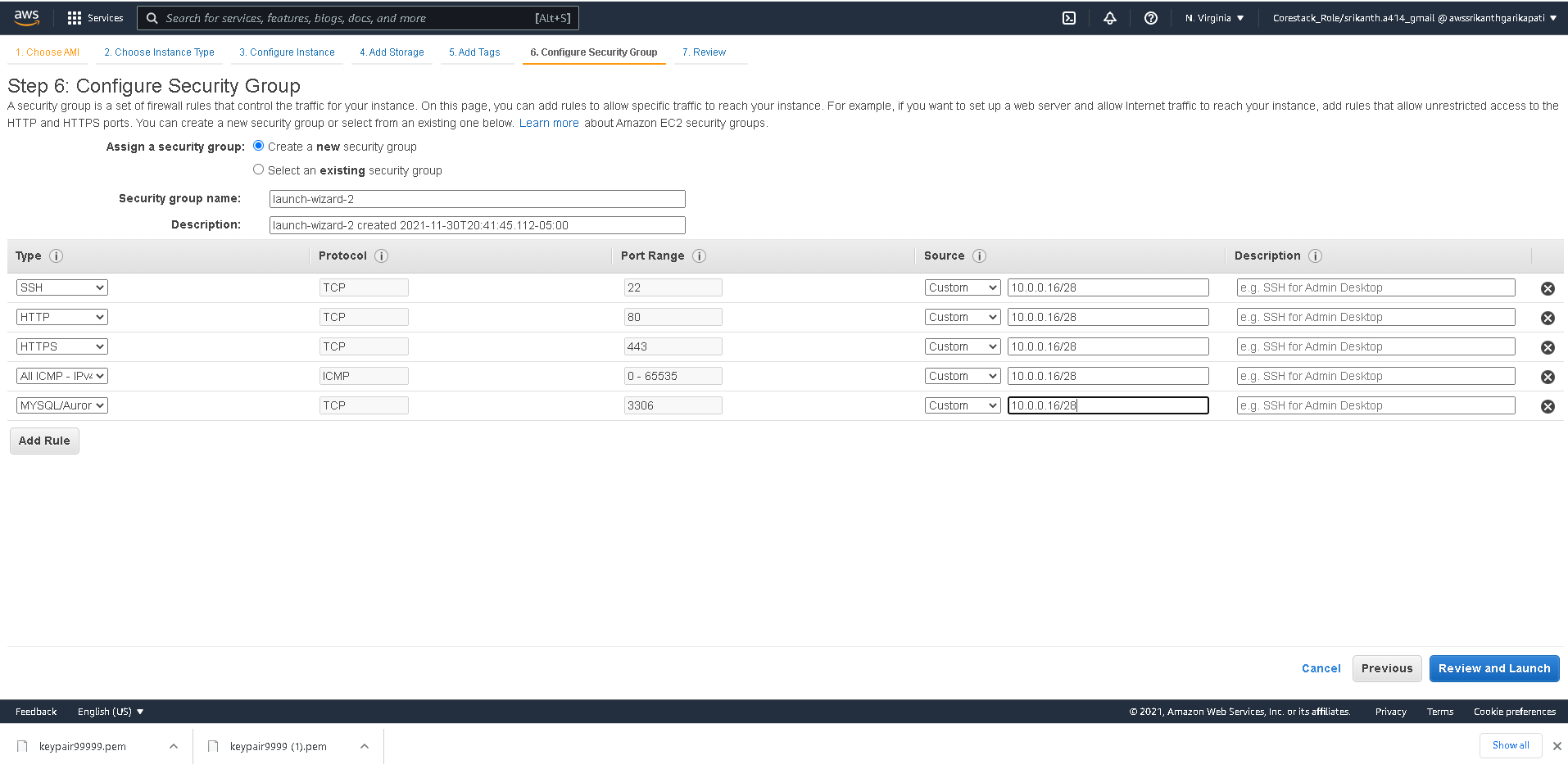


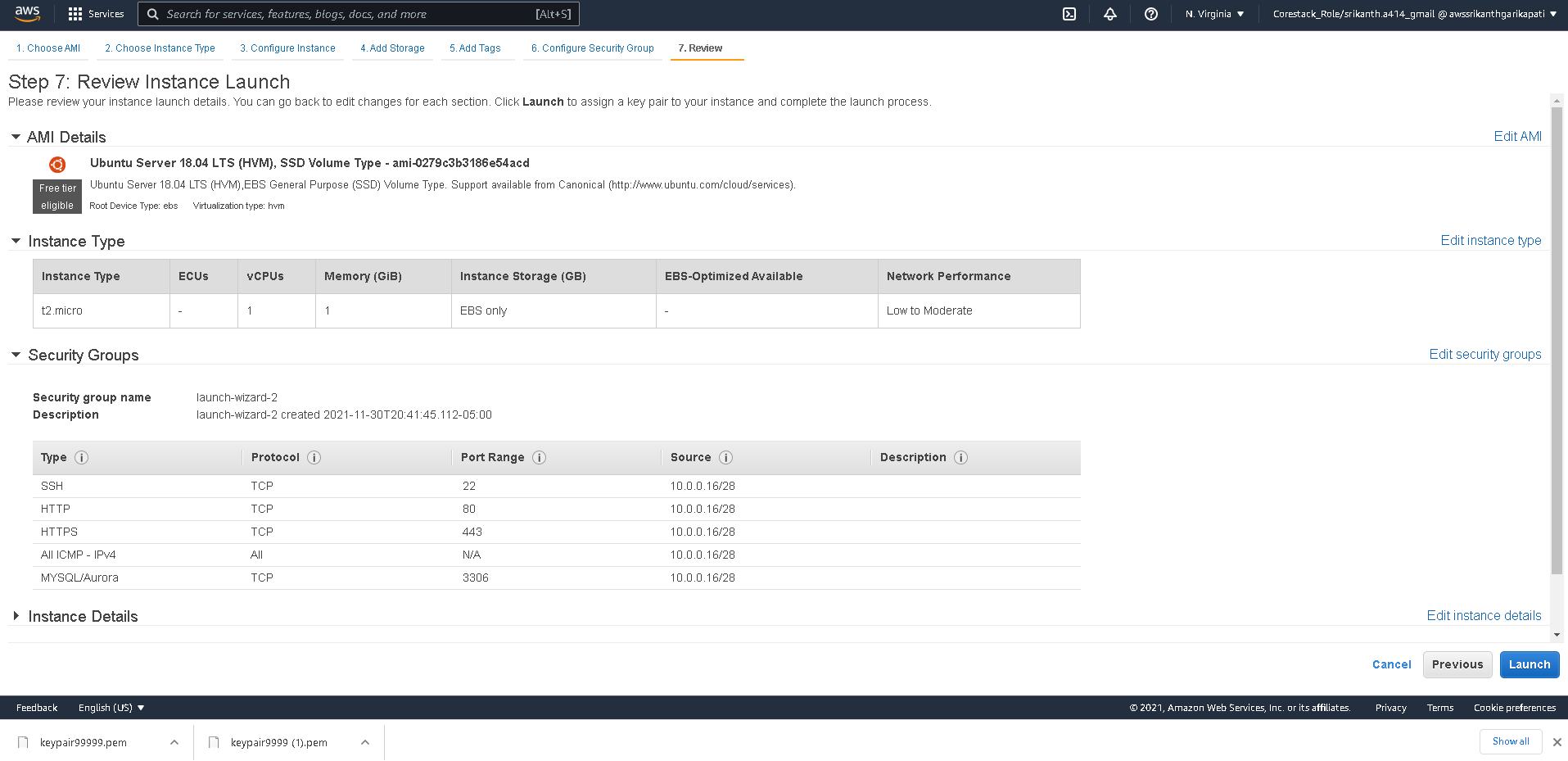
Make sure to select the Private Subnet with Public Ip disabled



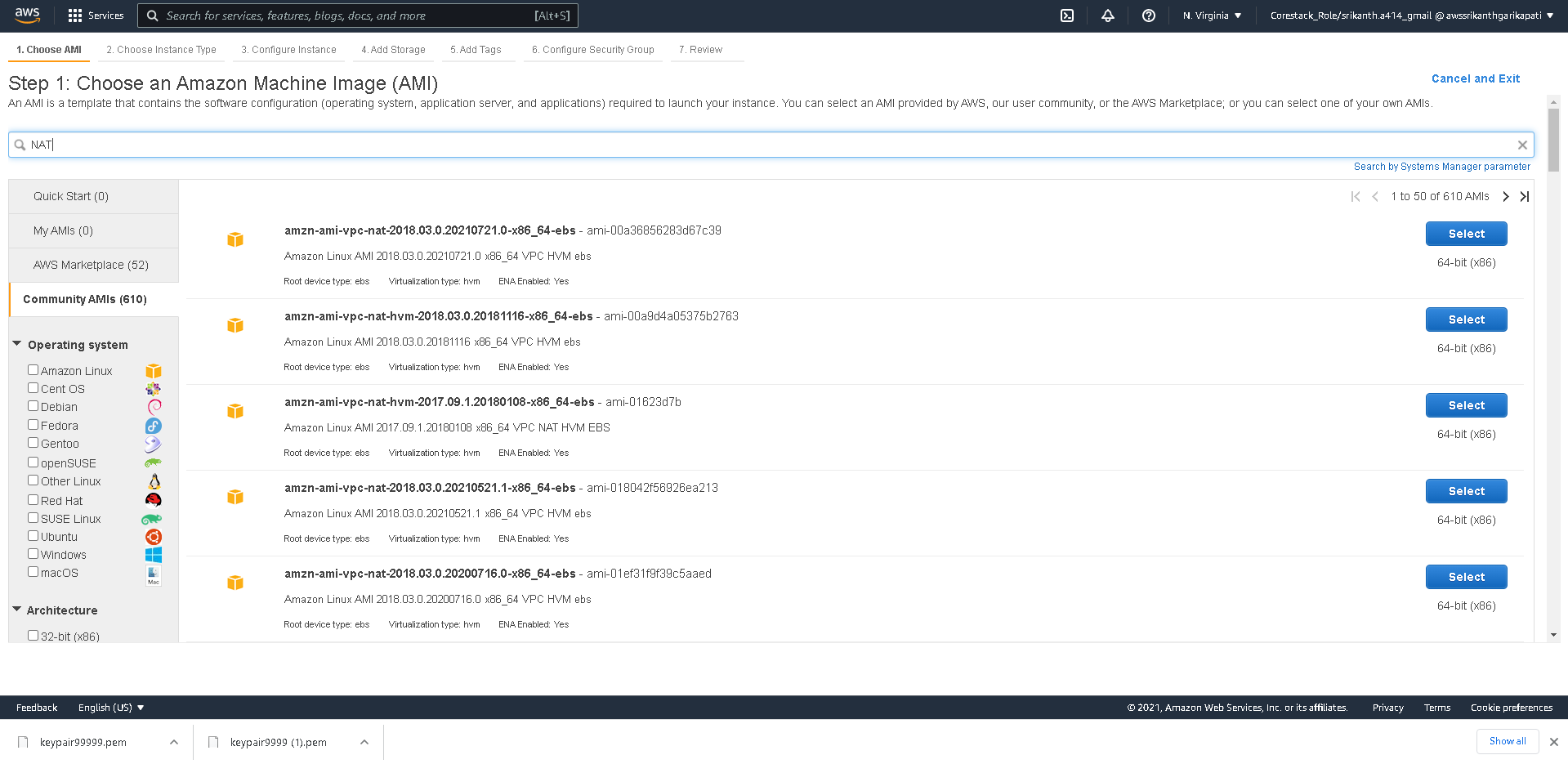


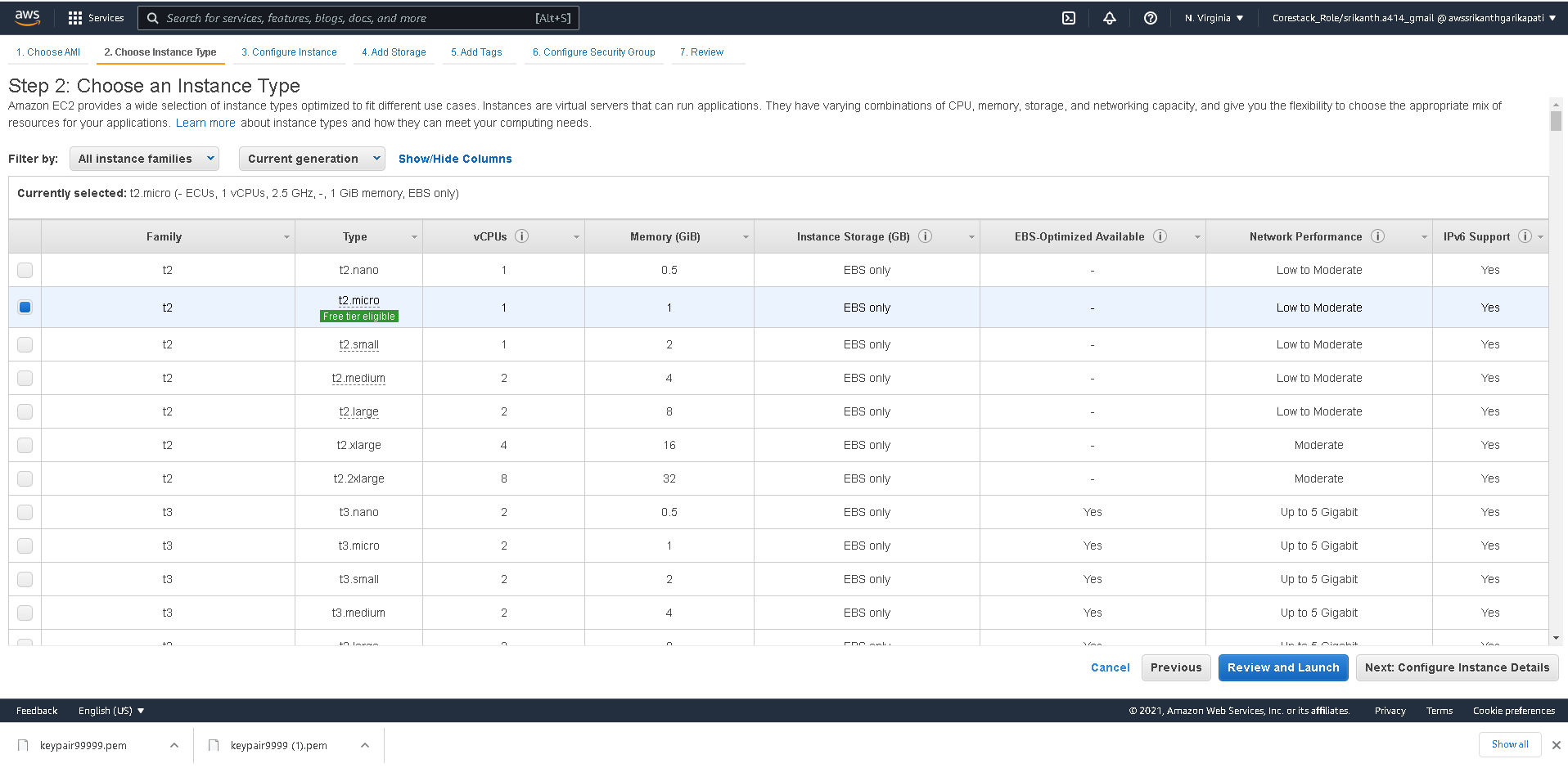
Create a new Security Group for DB instance CIDR range of Public subnet is mentioned as the source



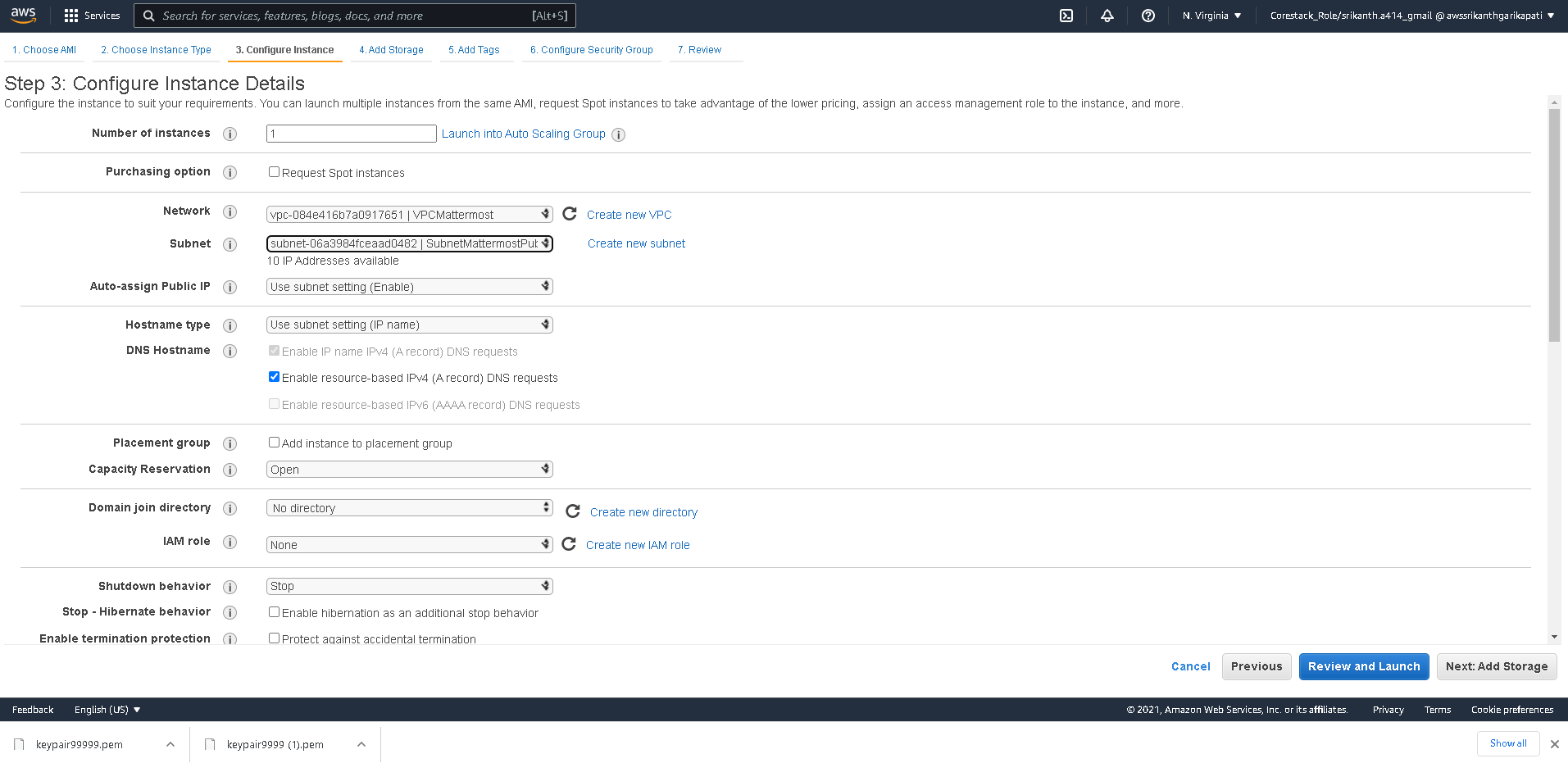


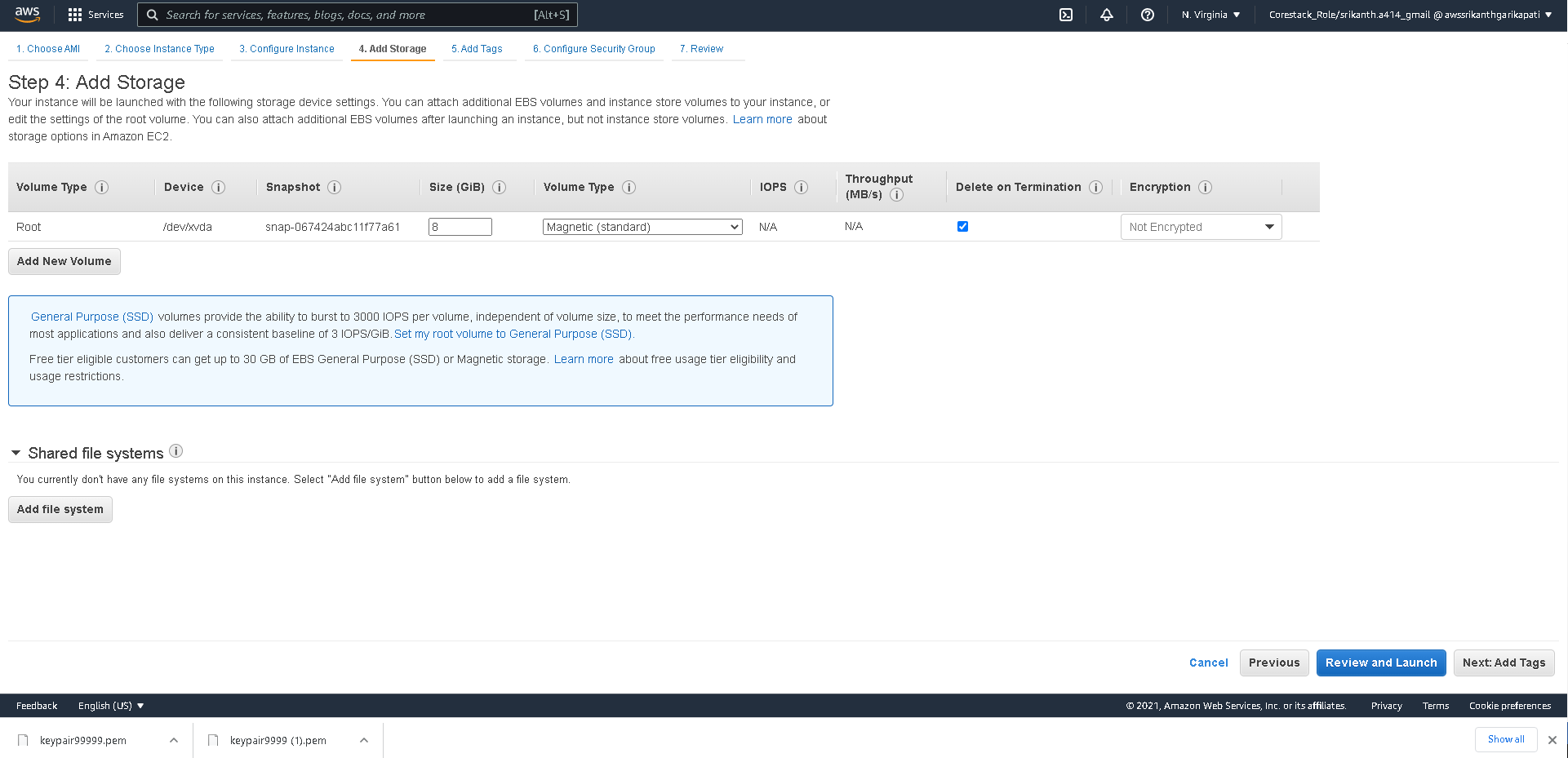
Launch NAT instance by select community AMIs and selecting NAT specific AMI



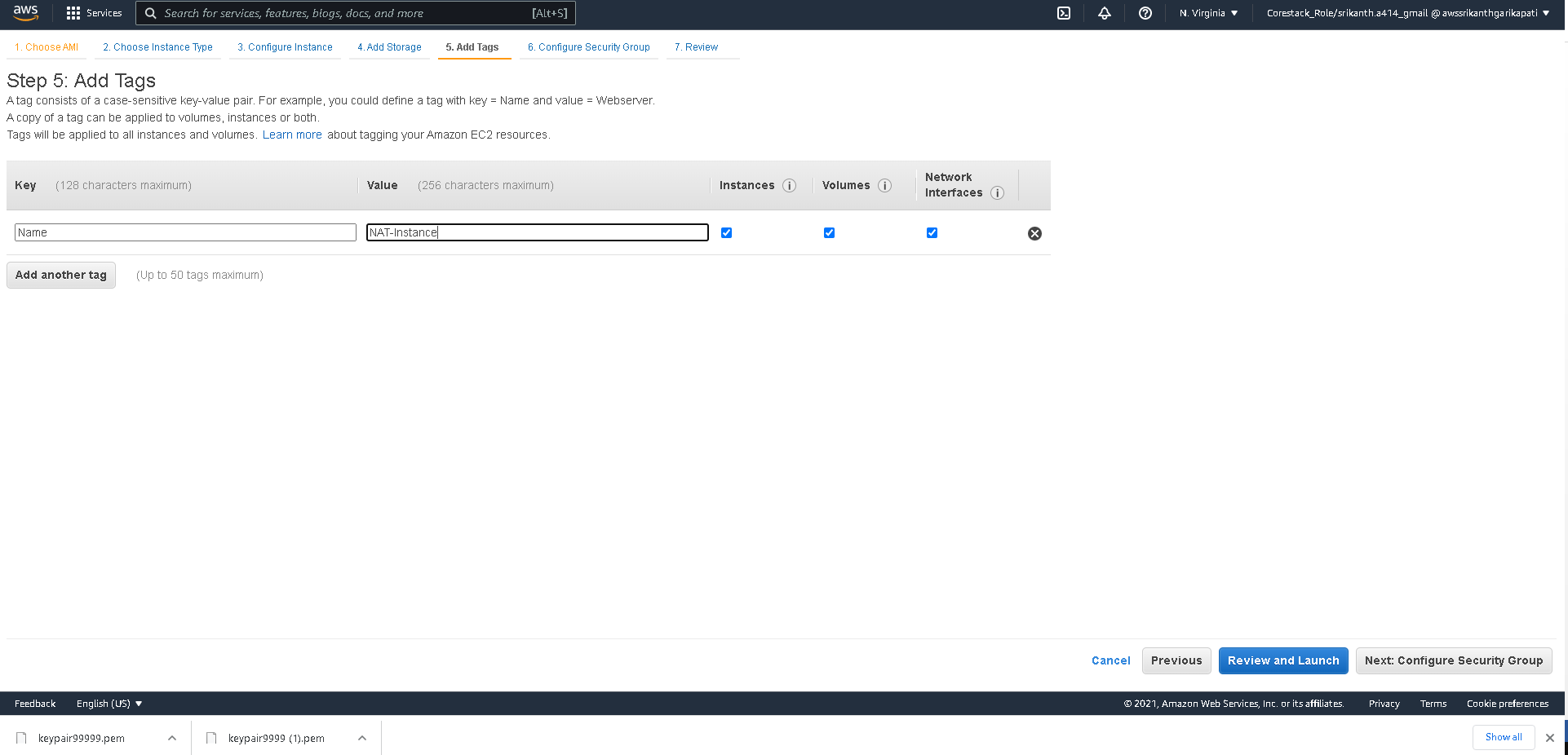


Create NAT instance in the Public Subnet



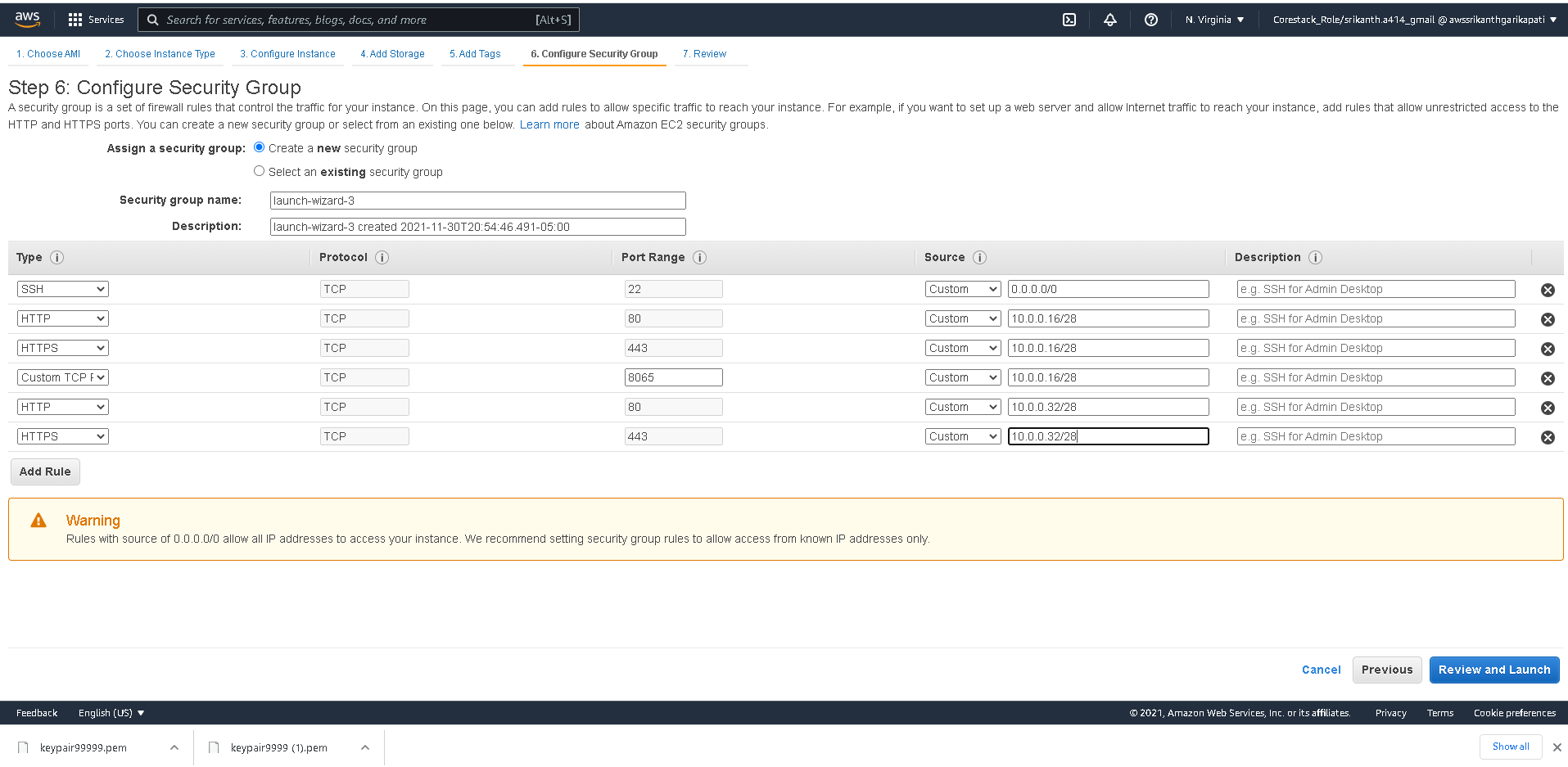


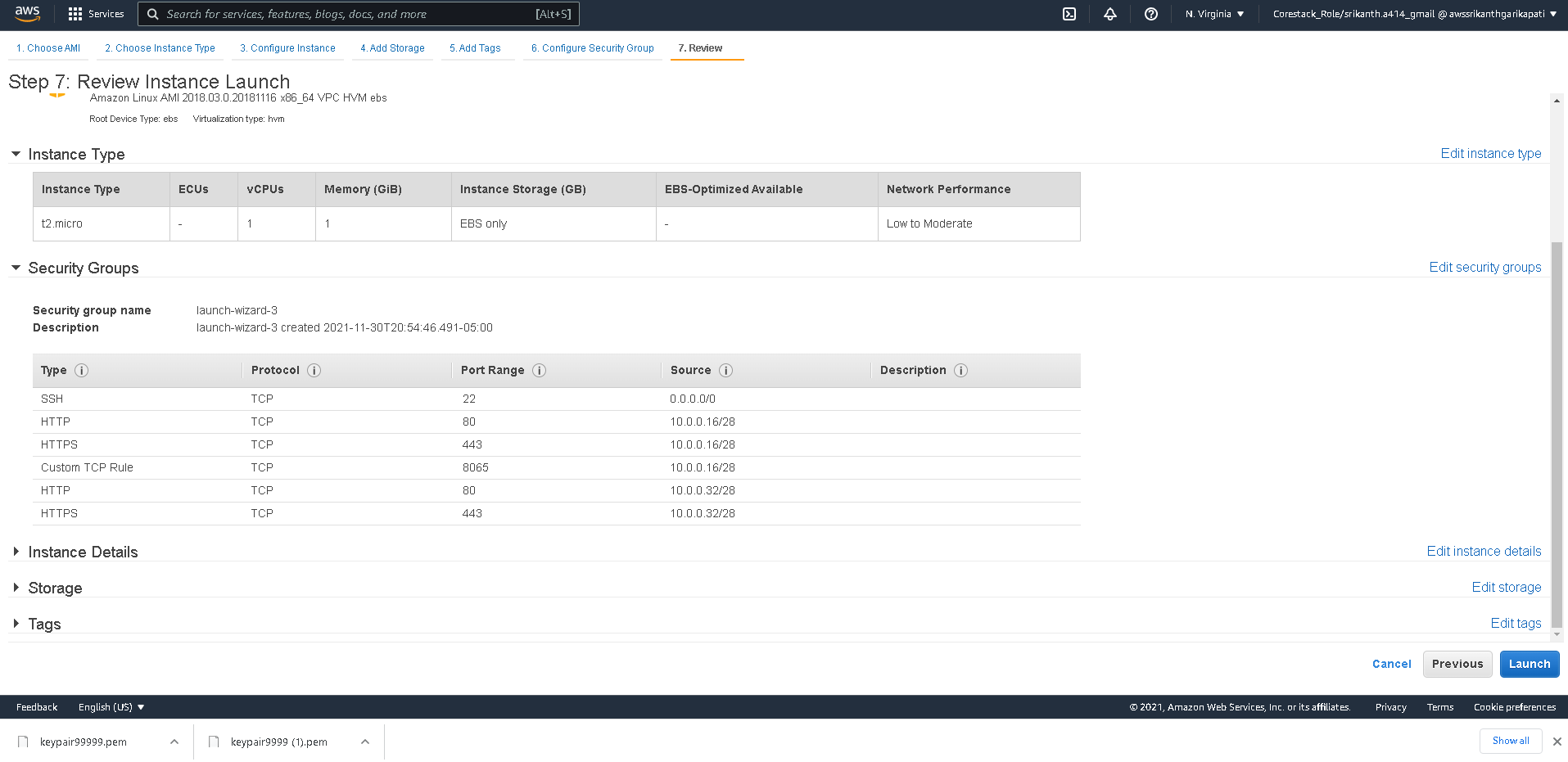
Name your Instance



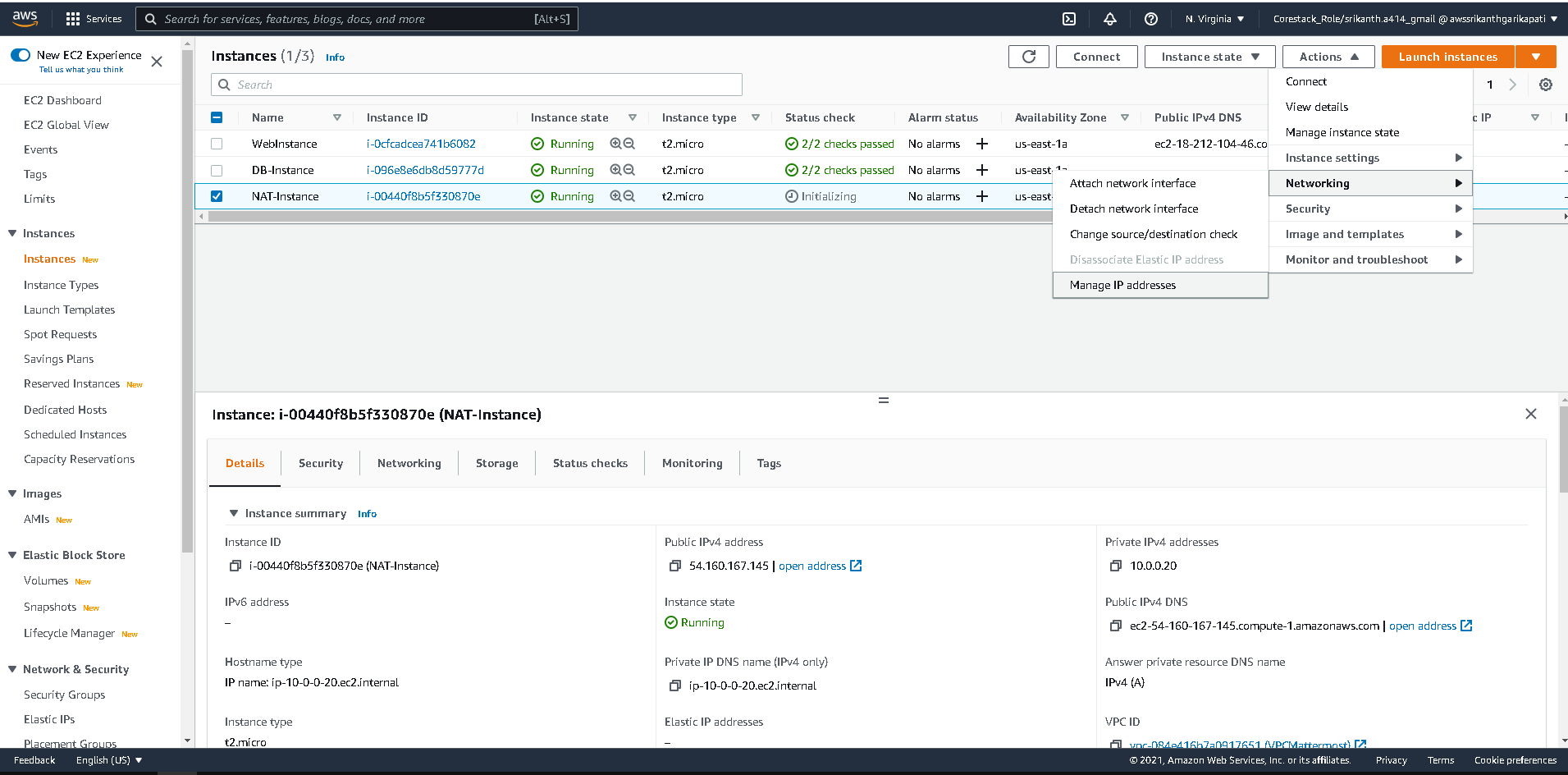
New Security Group for this NAT instance

1. Traffic inflow from private subnet – to interact to the internet
2. Traffic inflow from within the public subnet with port 80,443 is specified which means that the NAT instance is acting as a firewall instance for the traffic inflow for both Public and Private Subnet

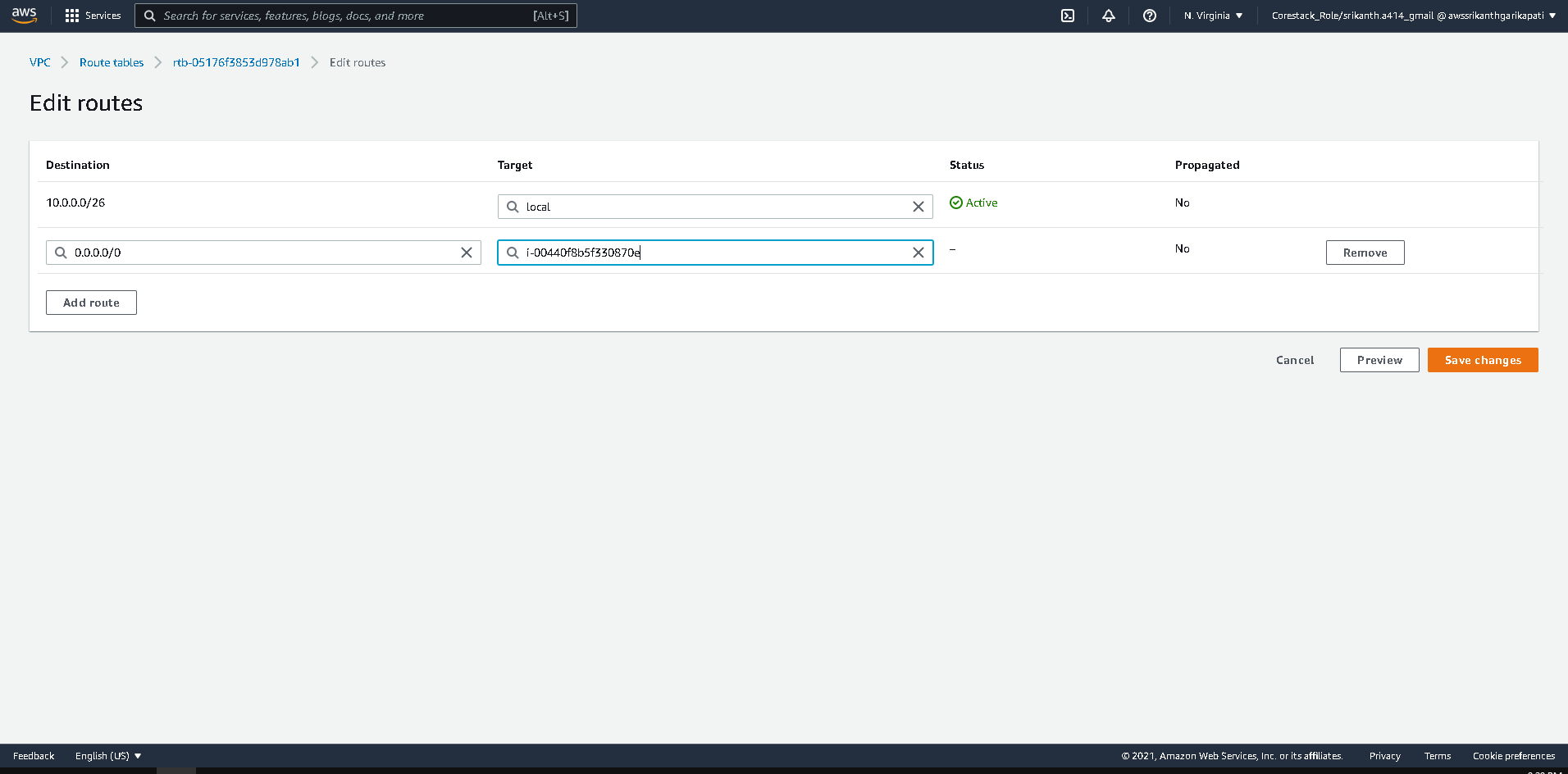




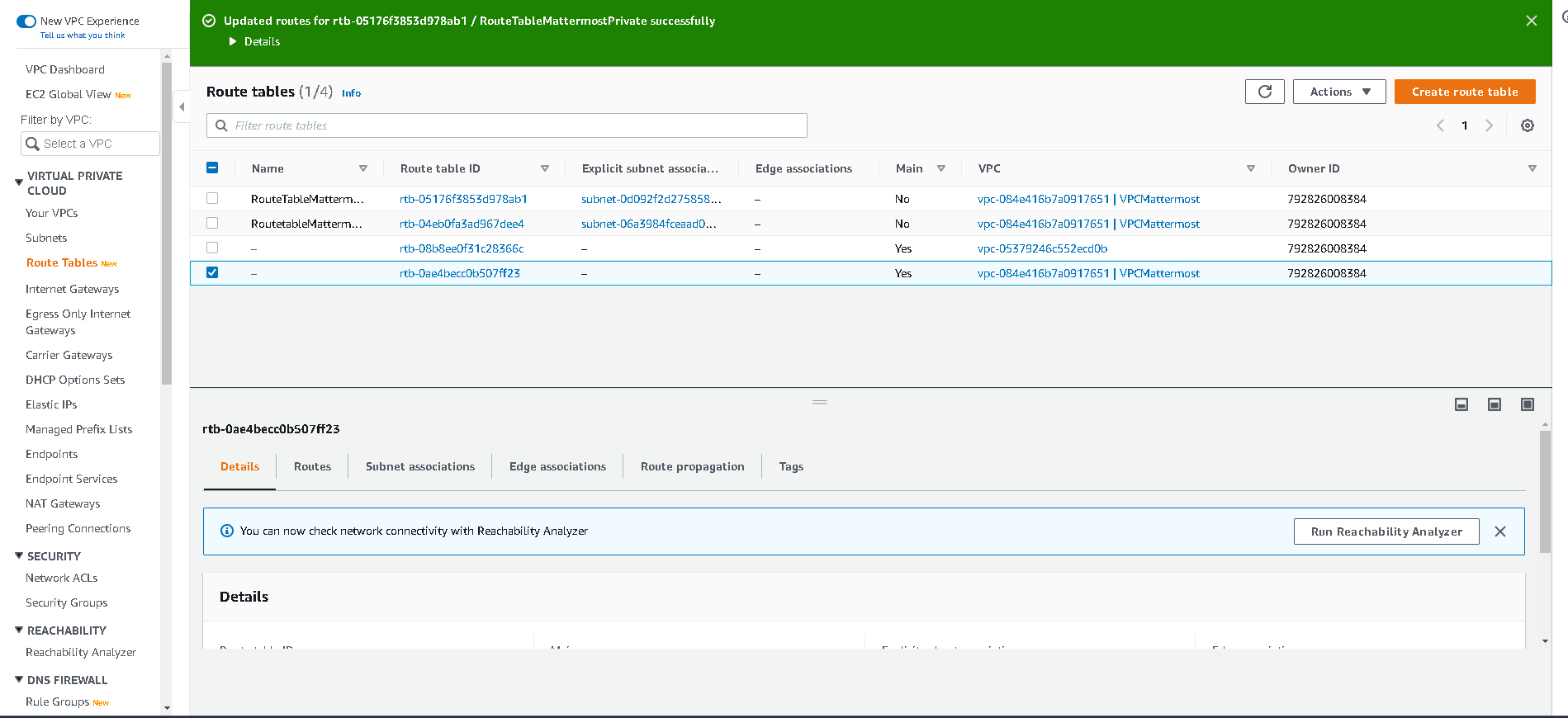
Select NAT instance and Disable source and destination check

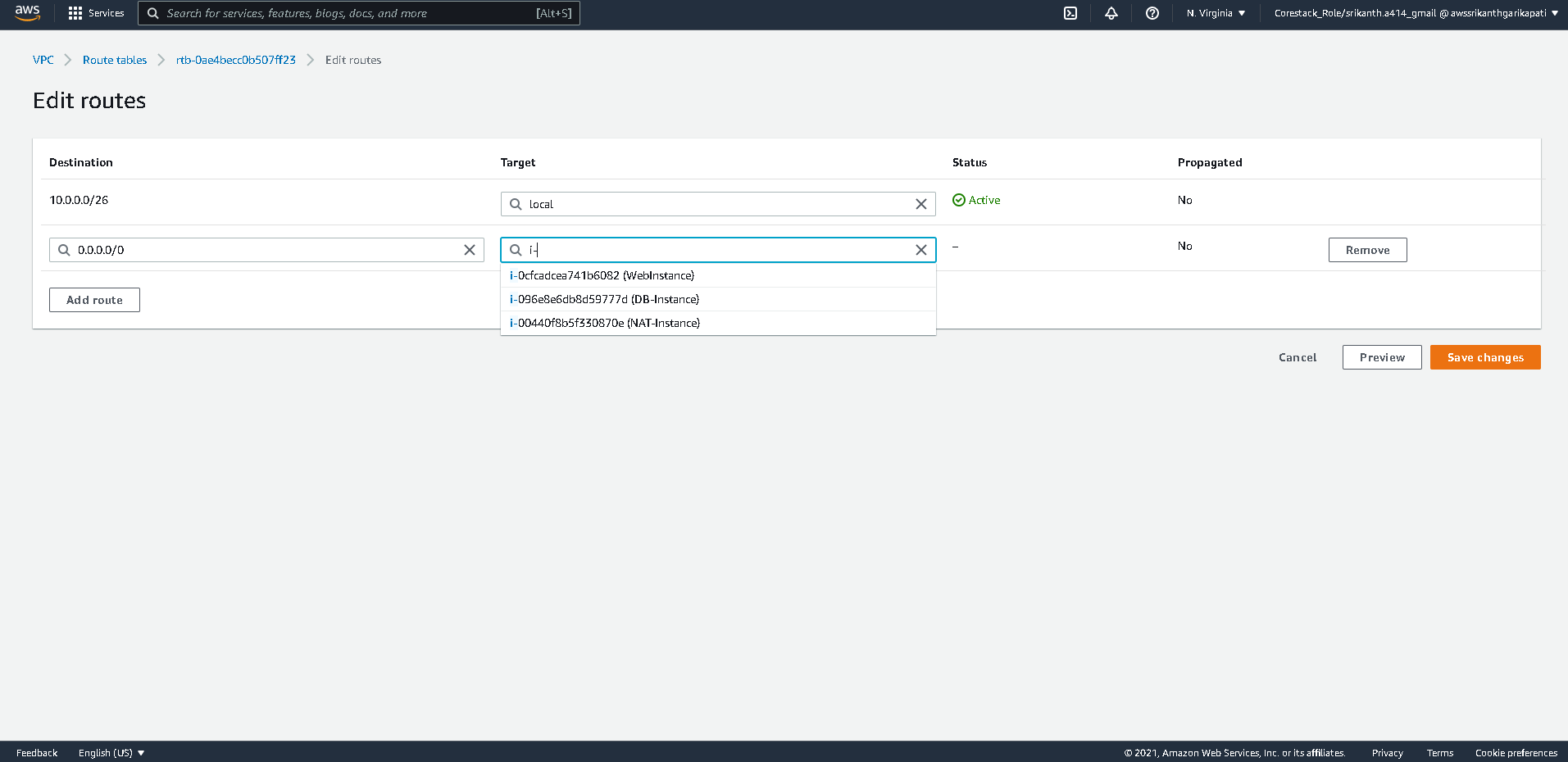


Click on Routes of Private RT and set destination as ‘0.0.0.0/0’ and set the Target as NAT instance.

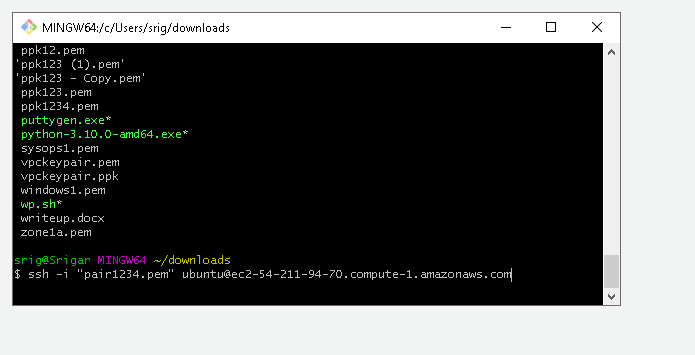


Main RT ( Default RT Launched with VPC ) also gets the route to NAT, same like private RT

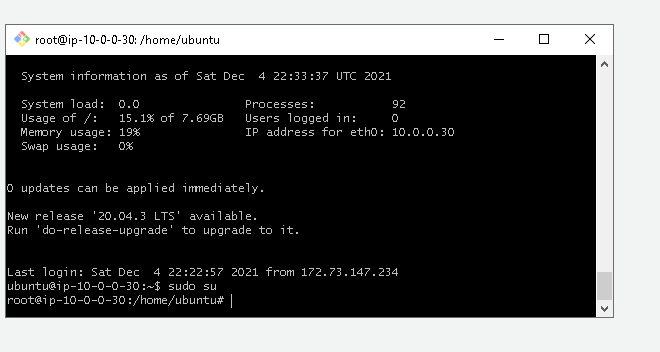




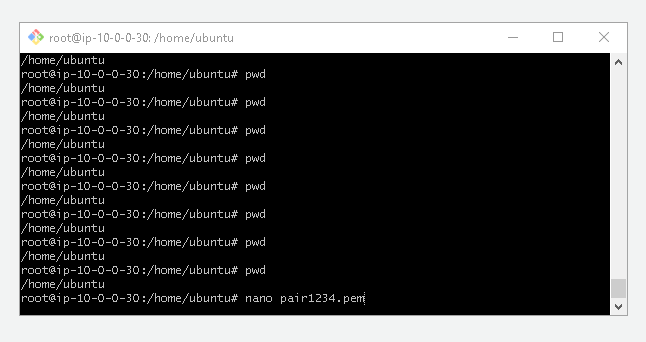
Go to Public Instance and SSH in to instance using Public IP address



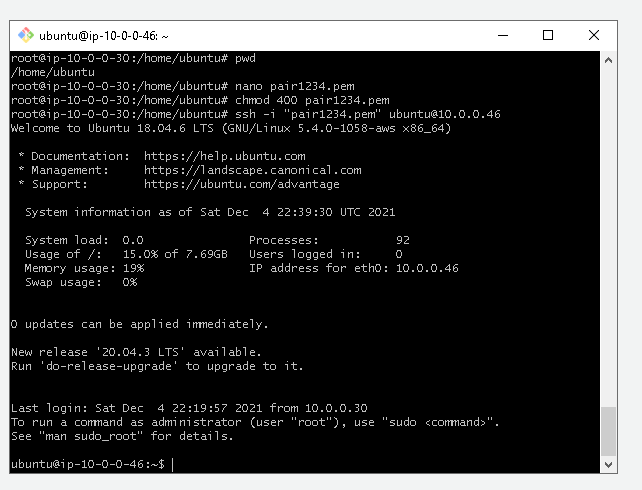
Switch to Super user



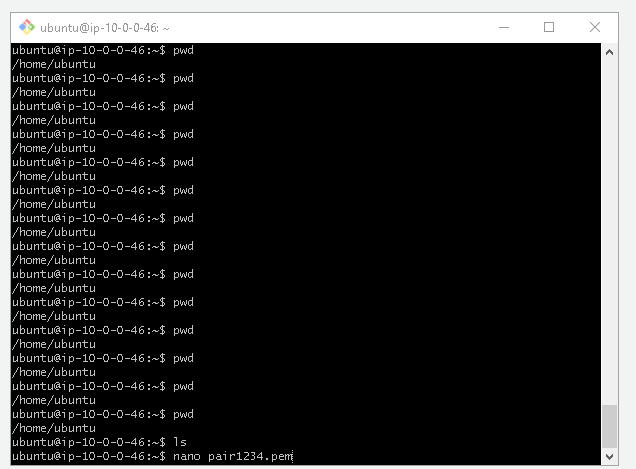
Create SSH key file



Log in to Private instance via Public Instance



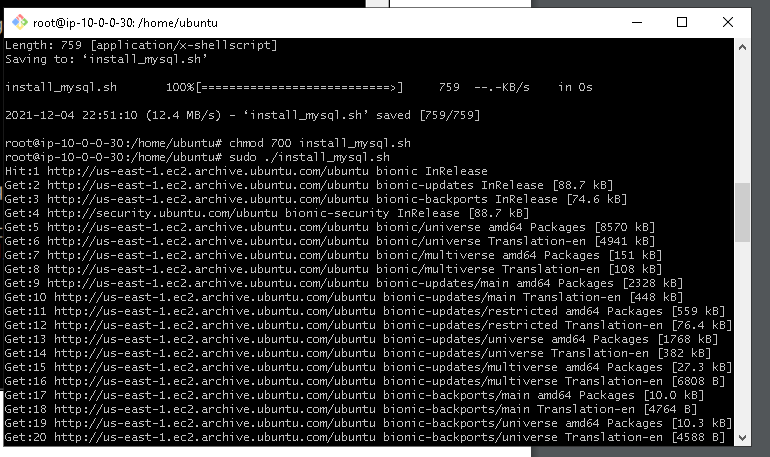
Create SSH key in Private instance as well

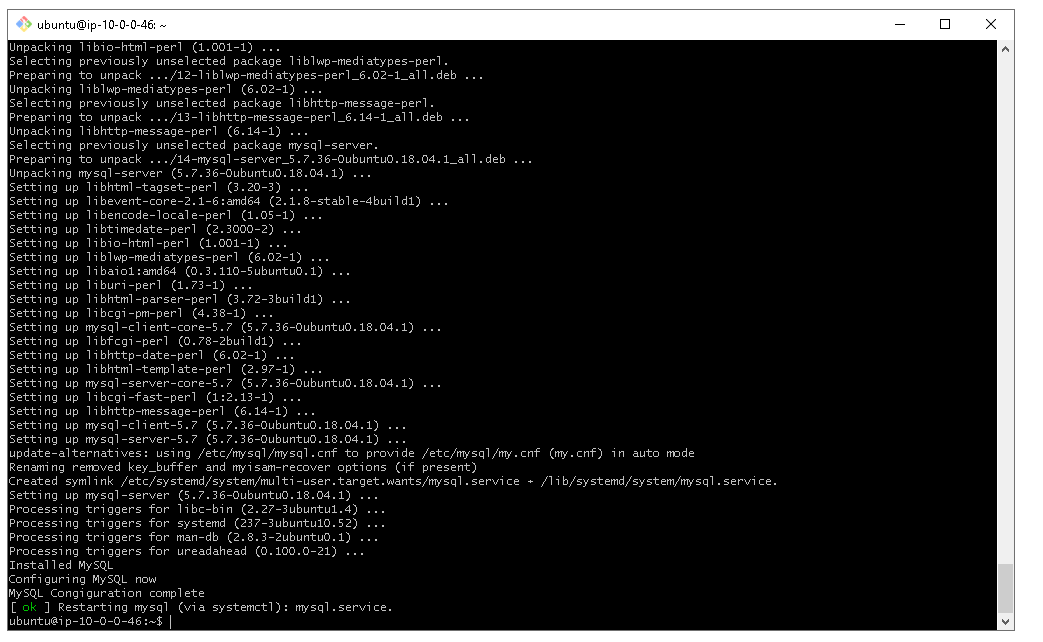


Install DB on Private DB server

Type the following commands step by step

1. Wget <https://storage.googleapis.com/skl-training/aws-codelabs/mattermost/install_mysql.sh>
2. chmod 700 install\_mysql.sh
3. sudo ./install\_mysql.sh





Your DB installation is complete now

Switch Over to your Web-Instance and follow the steps to install Mattermost application

Switch as super user

wget <https://storage.googleapis.com/skl-training/aws-codelabs/mattermost/mattermost_install.sh>

chmod 700 mattermost\_install.sh

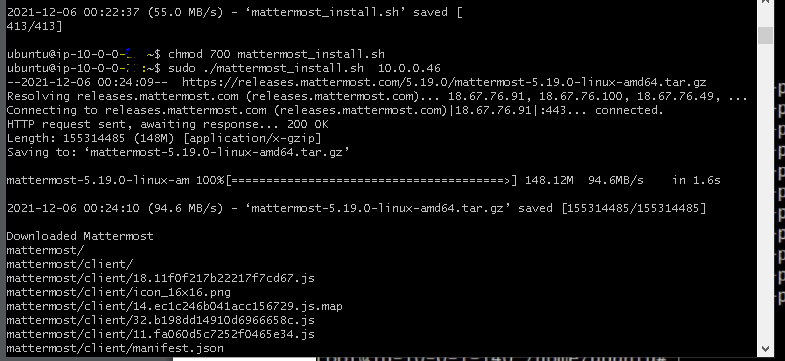
sudo ./mattermost\_install.sh 10.0.0.46

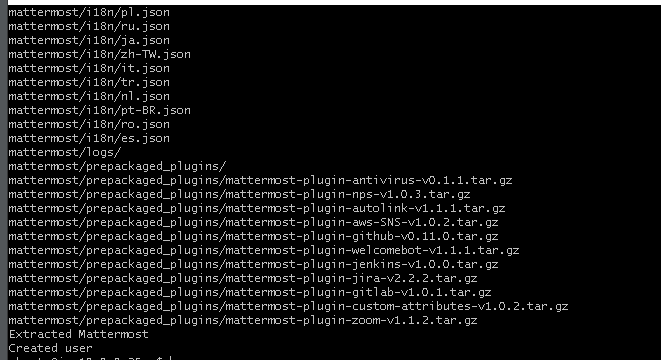
sudo chown -R mattermost:mattermost /opt/mattermost

sudo chmod -R g+w /opt/mattermost

cd /opt/mattermost







sudo -u mattermost ./bin/mattermost



When you receive the message that the server is listening on port: 8065 your setup is complete.

Grab the publicIPv4 of your WebInstance and hit the below url in your browser to view your MatterMost application

<PublicIPV4 of WebInstance>:8065

