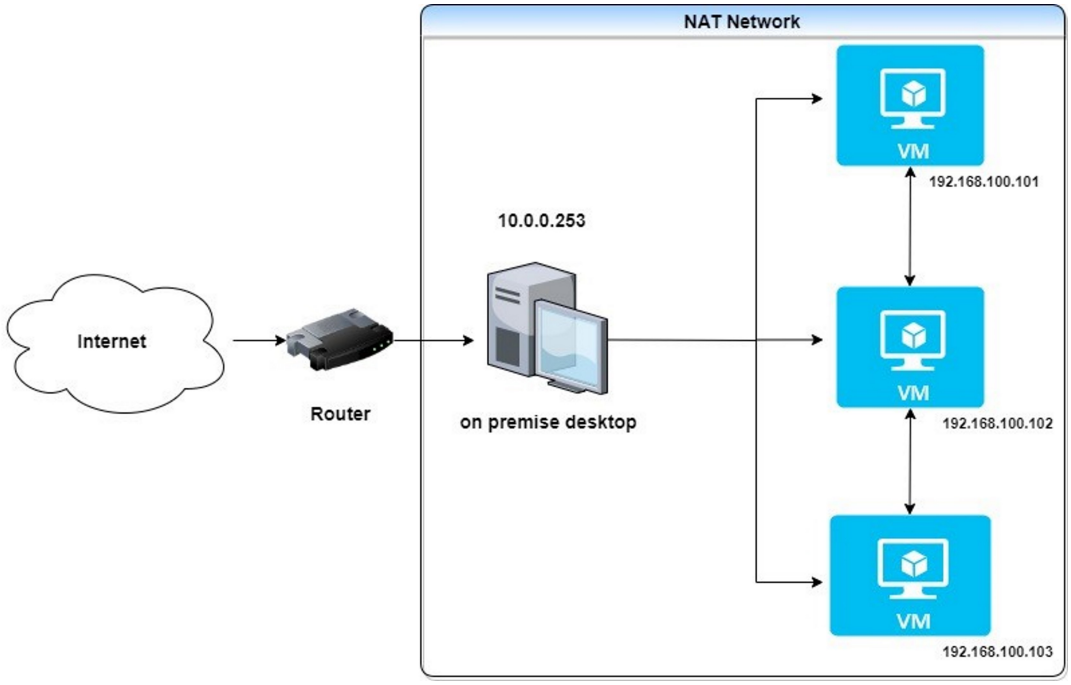
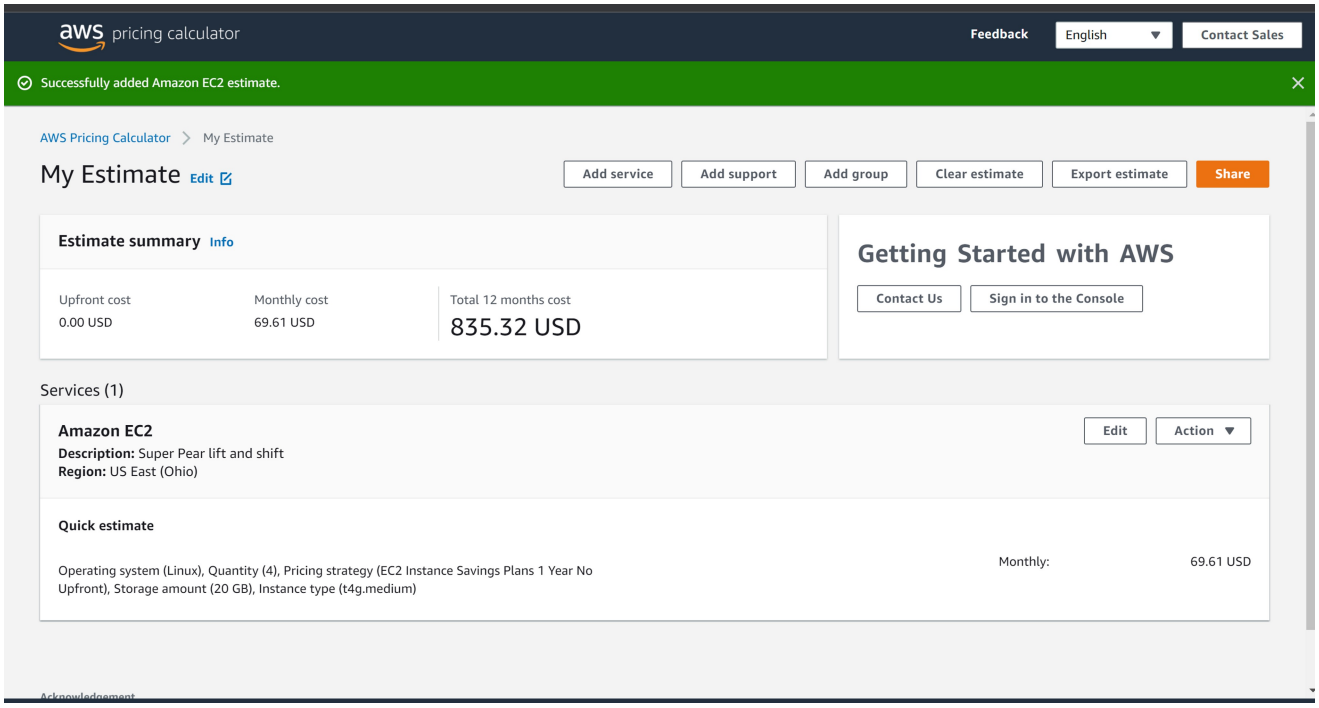
AWS Capstone Project

Monday, November 29, 2021 9:47 AM

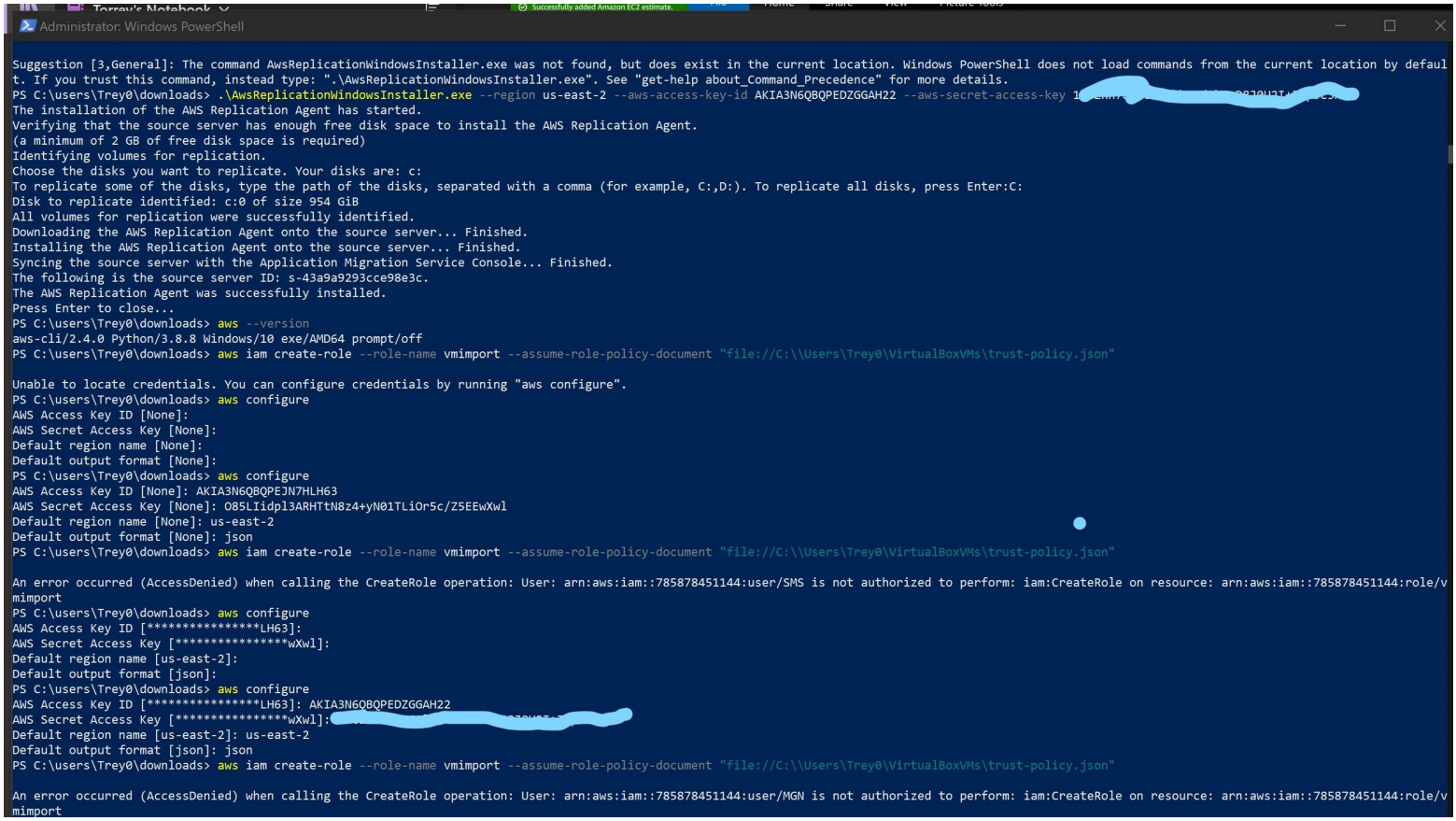
Super Pear is a new tech startup that is looking to convert their current network infrastructure to the cloud using Amazon's AWS. Their current network consists of one router, one physical desktop PC, and 3 laptops running VM software. what would be the most cost effective way to migrate the Super Pear team to AWS? Provide screenshots of the network prior to, during, and after the conversion.



Before we begin discussing the actual process of converting their network to the cloud, it's best to see how the move would financially impact this company.



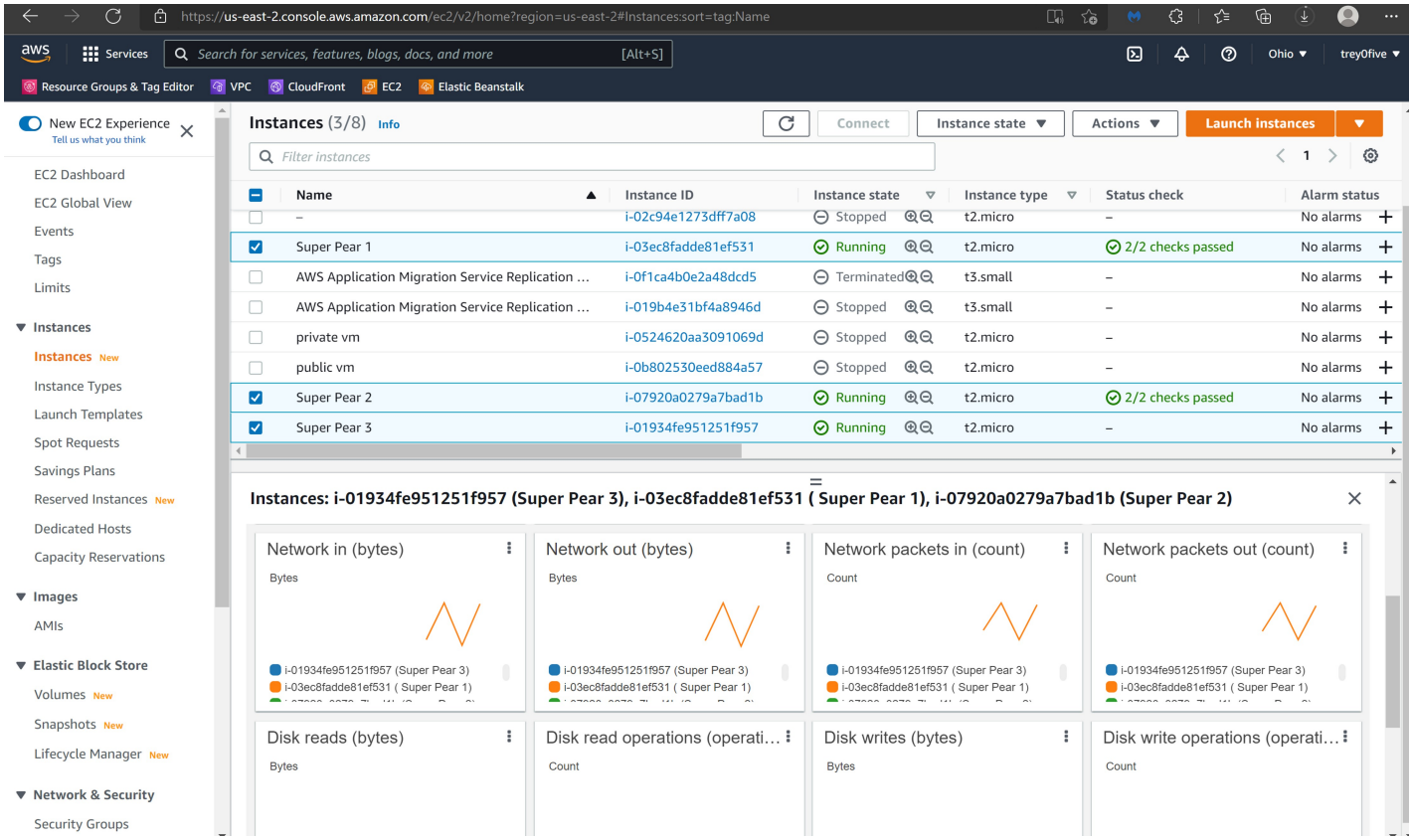
Using the AWS Pricing calculator, we can get an estimate of how much it'll be to virtualize Super Pear's network. Since the company needs constant and consistent performance, it would be more beneficial to sign up using the 1 year savings plan option. This will provide a decent sized discount! Also, since this company is expecting to expand in size in the near future, moving to the cloud will provide the necessary elasticity to expand resources without worrying about Capital expenditure.



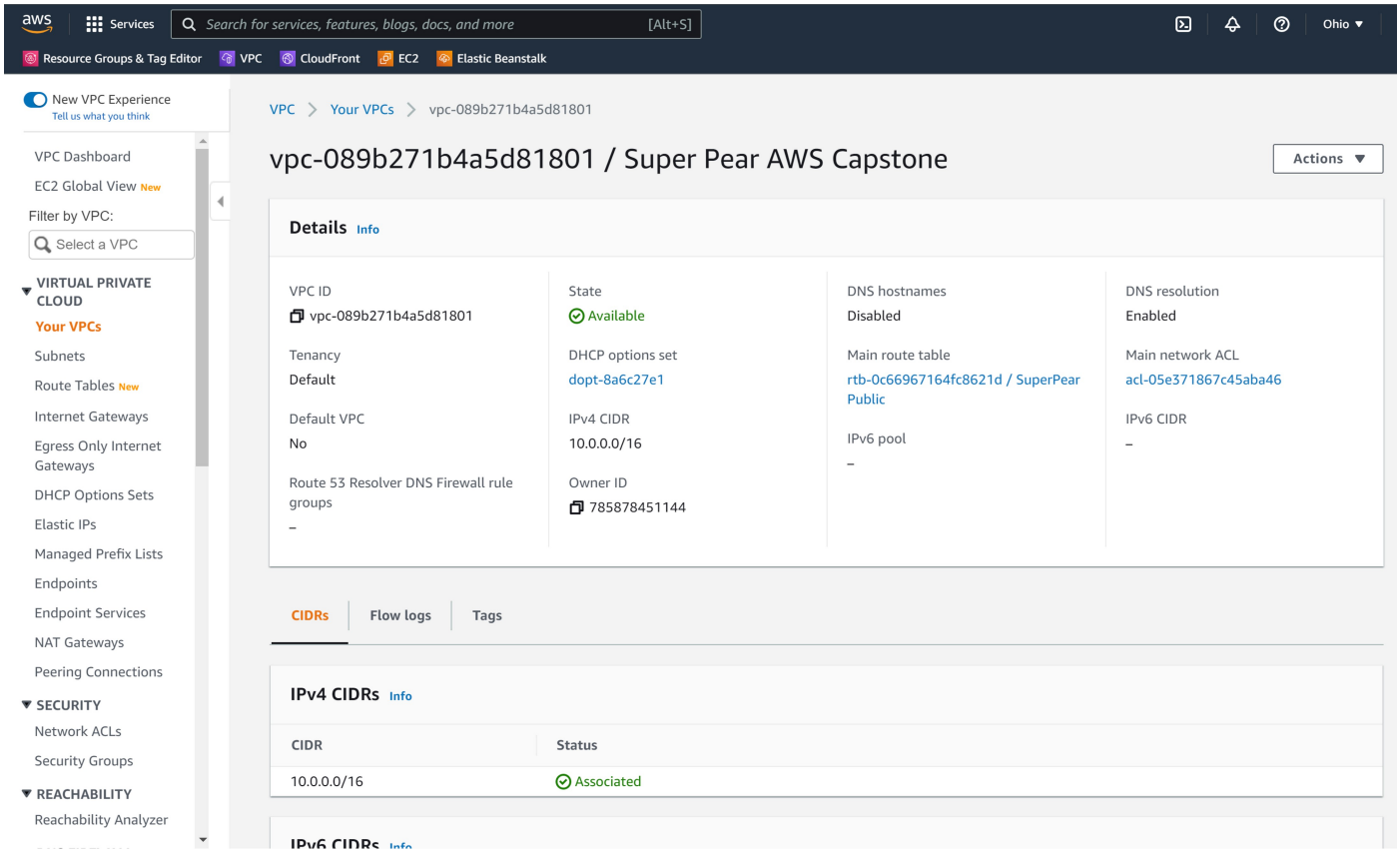
To begin the process, instead of using Amazon migration services which is better for larger operations, it was easier to use Windows PowerShell to run the AWS replication installer to prep the transfer of VMs to AWS. This is the starting process of a lift and shift.

```
                    "Service": "vmie.amazonaws.com"
                },
                "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {
                        "sts:Externalid": "vmimport"
                    }
                }
            }
        ]
    }
}

PS C:\users\Trey0\downloads> aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document "file://C:\Users\Trey0\VirtualBoxVMs\role-policy.json"
PS C:\users\Trey0\downloads> aws ec2 import-image --description "UbuntuVMs" --license-type byol --disk-containers "file://C:\Users\Trey0\VirtualBoxVMs\containers.json"
{
    "Description": "UbuntuVMs",
    "ImportTaskId": "import-ami-022183e411188e4e7",
    "LicenseType": "BYOL",
    "Progress": "1",
    "SnapshotDetails": [
        {
            "Description": "UbuntuVMs",
            "DiskImageSize": 0.0,
            "Format": "OVA",
            "UserBucket": {
                "S3Bucket": "teerobucket9999",
                "S3Key": "UbuntuVMs.ova"
            }
        }
    ],
    "Status": "active",
    "StatusMessage": "pending"
}

PS C:\users\Trey0\downloads>
```

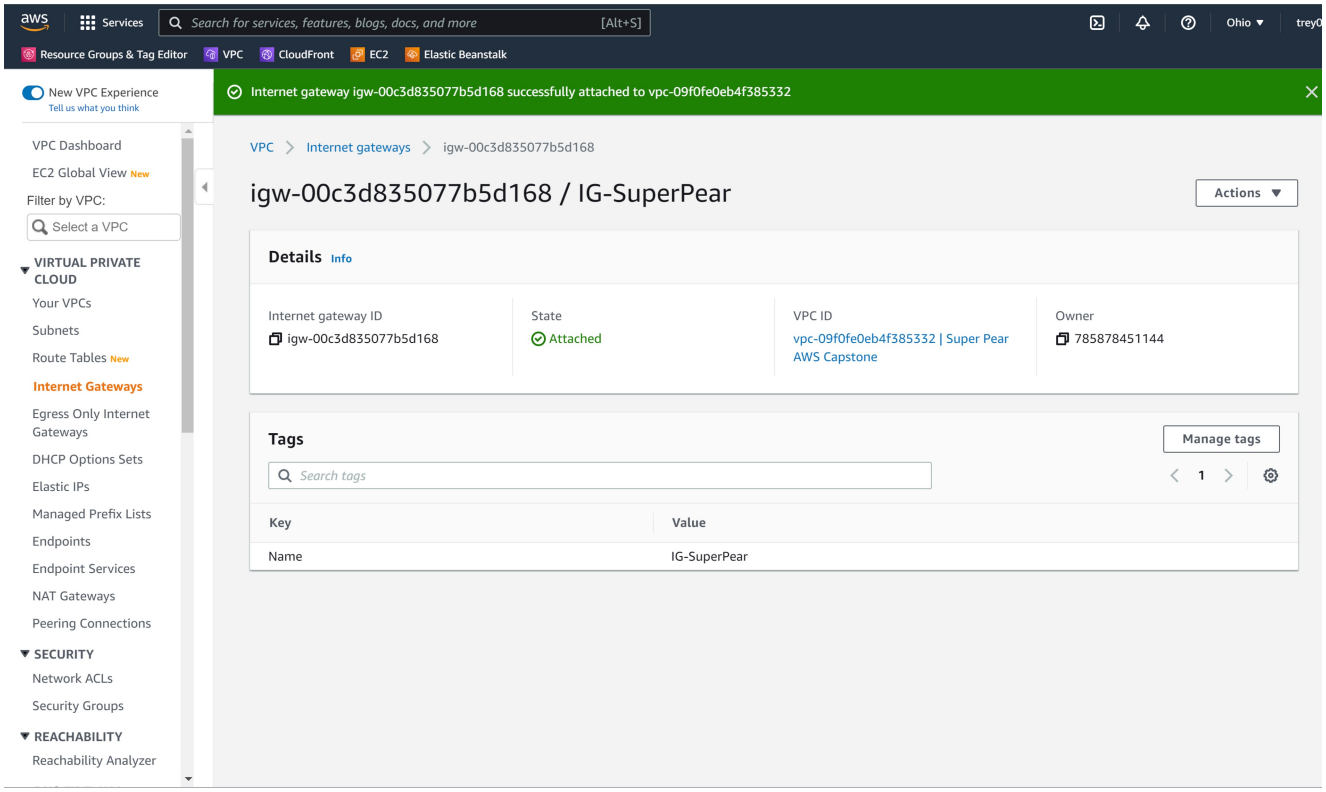
After exporting the Linux Ubuntu 16.04 VMs from virtualbox in .ova format, using PowerShell I was able to begin the import process into AWS and also check the status of the import. This was of course done after uploading the VM .ova files to the designated AWS S3 bucket.



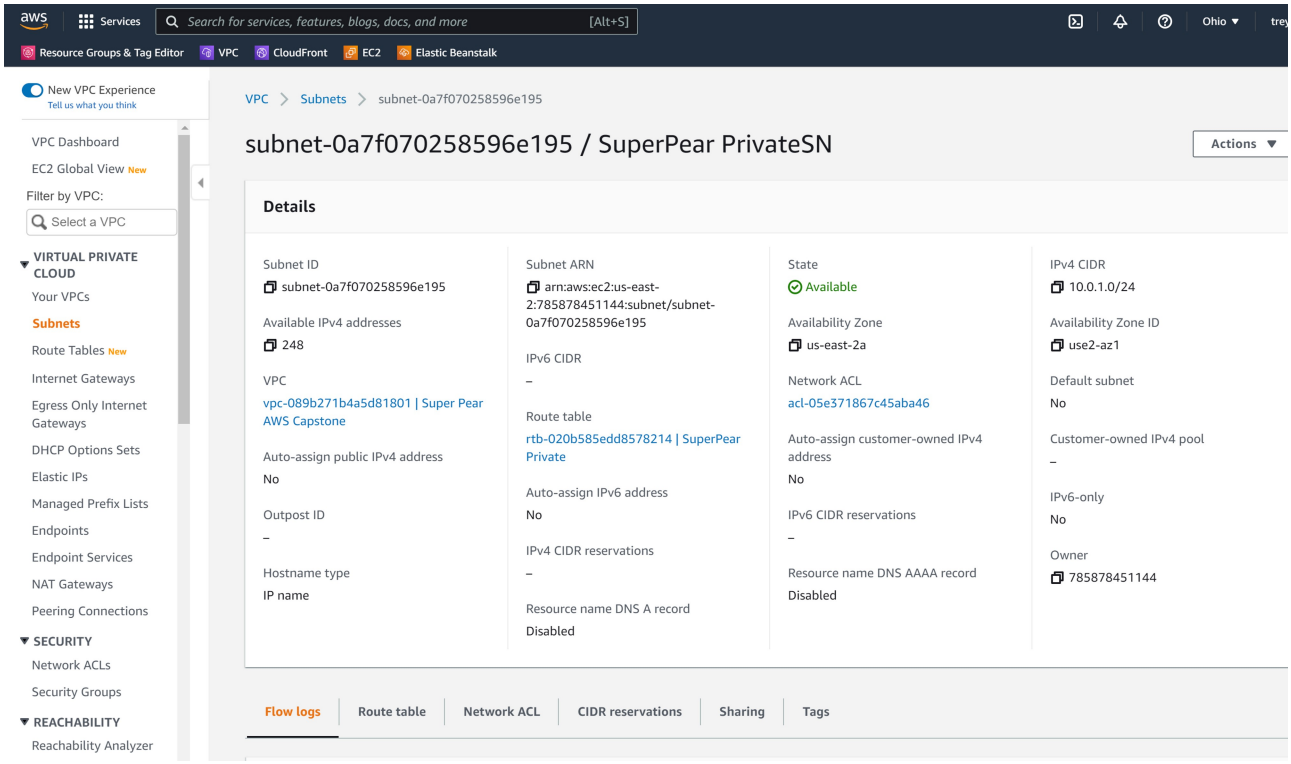
Once the .ova files were imported and formed into AMIs, they were then launched into instances and were able to be seen in the EC2 instance library.

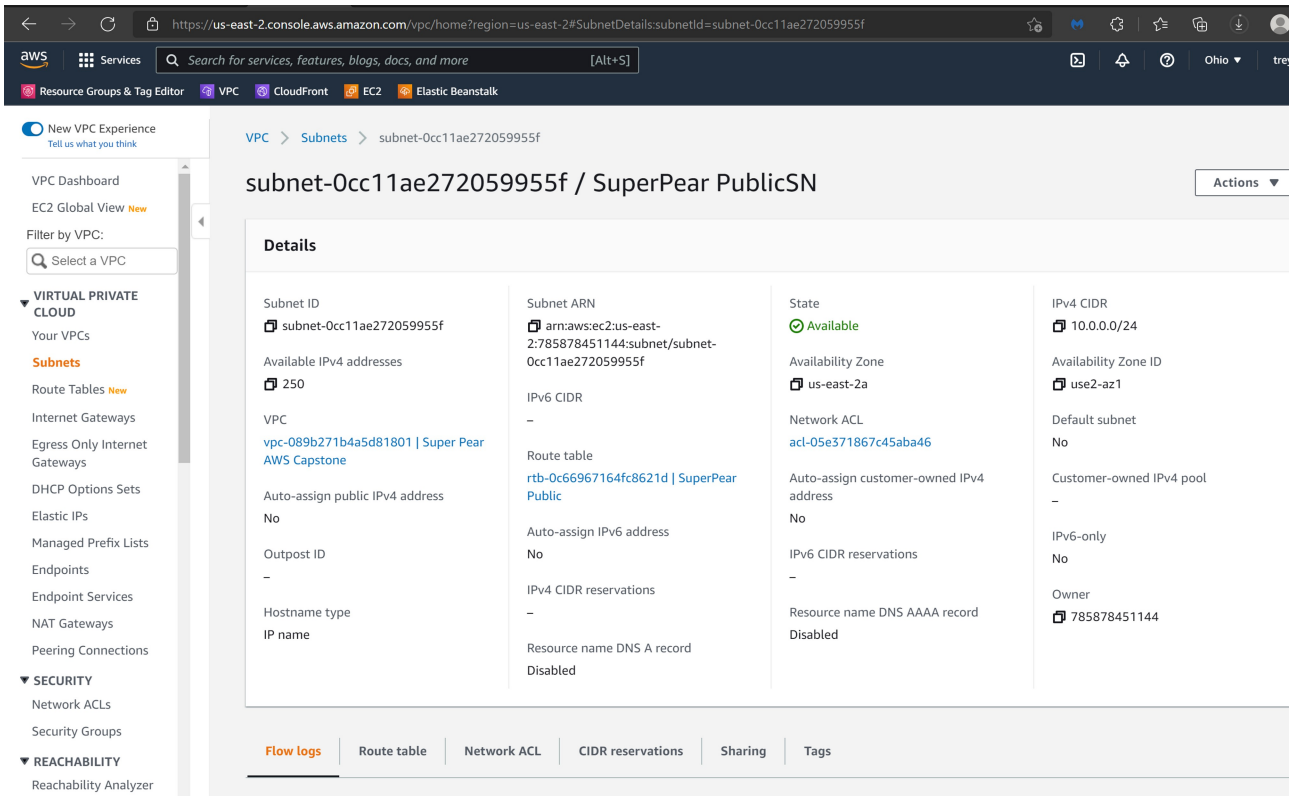


Once we have created the EC2 instances, the Virtual Private Cloud can be established. It was set up with the IPv4 CIDR range of 10.0.0.0/16

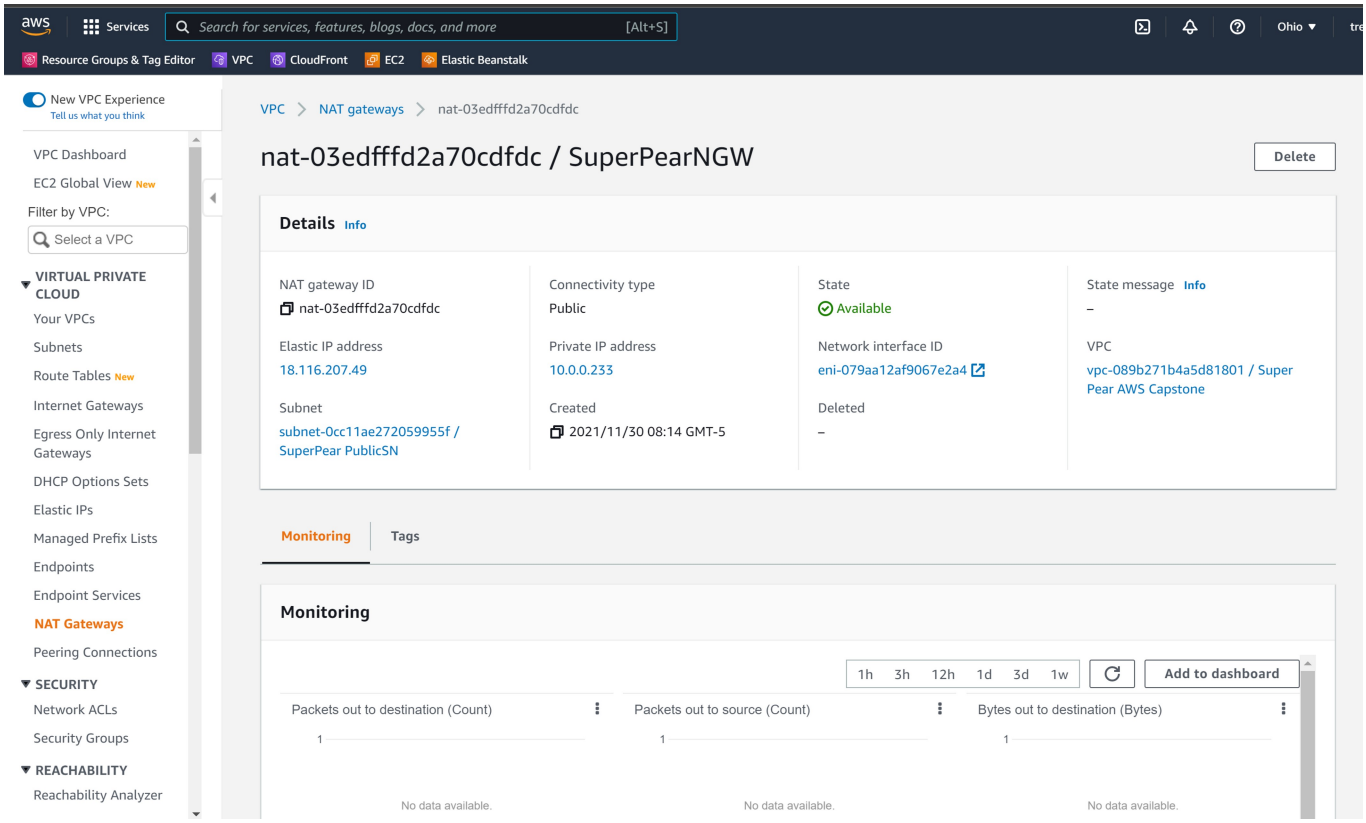


Next, we have to set up an internet gateway for the VPC, which is basically the virtual router that allows internet connectivity to the cloud. Once it is created, it has to be attached to the Super Pear VPC.

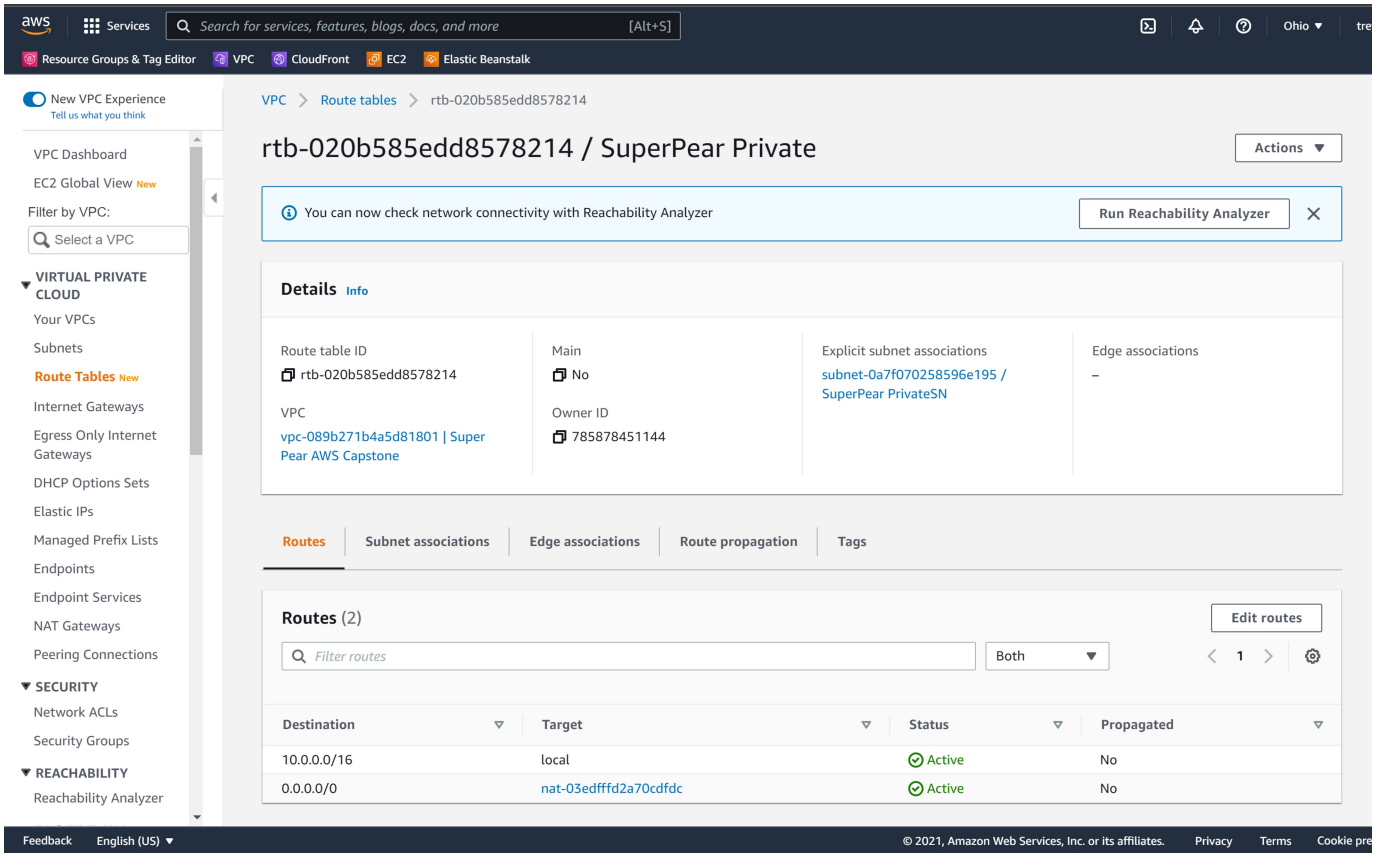




Created a public subnet (10.0.0.0/24) and linked it to the Super Pear public route table which will allow incoming and outgoing web traffic for the master VM. Also created a private subnet (10.0.1.0/24) which will allow the other 3 VMs to only communicate with each other.

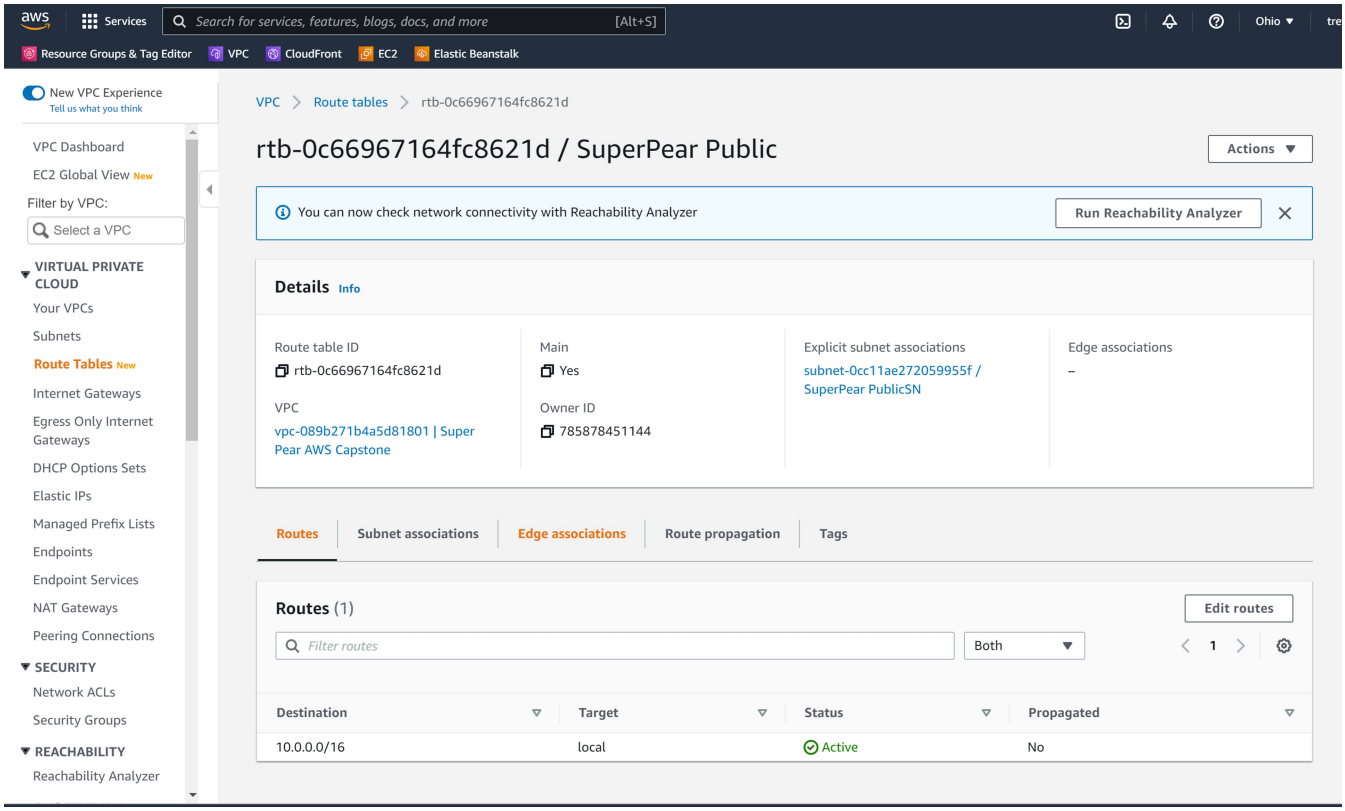


Created and attached a NAT Gateway to the public SuperPear subnet and gave it an Elastic IP address . We will then to adjust the Route table connected to the private subnet so that it can communicate with the NAT Gateway. This will allow the 3 connected VMs to access the web for software updates/patches.

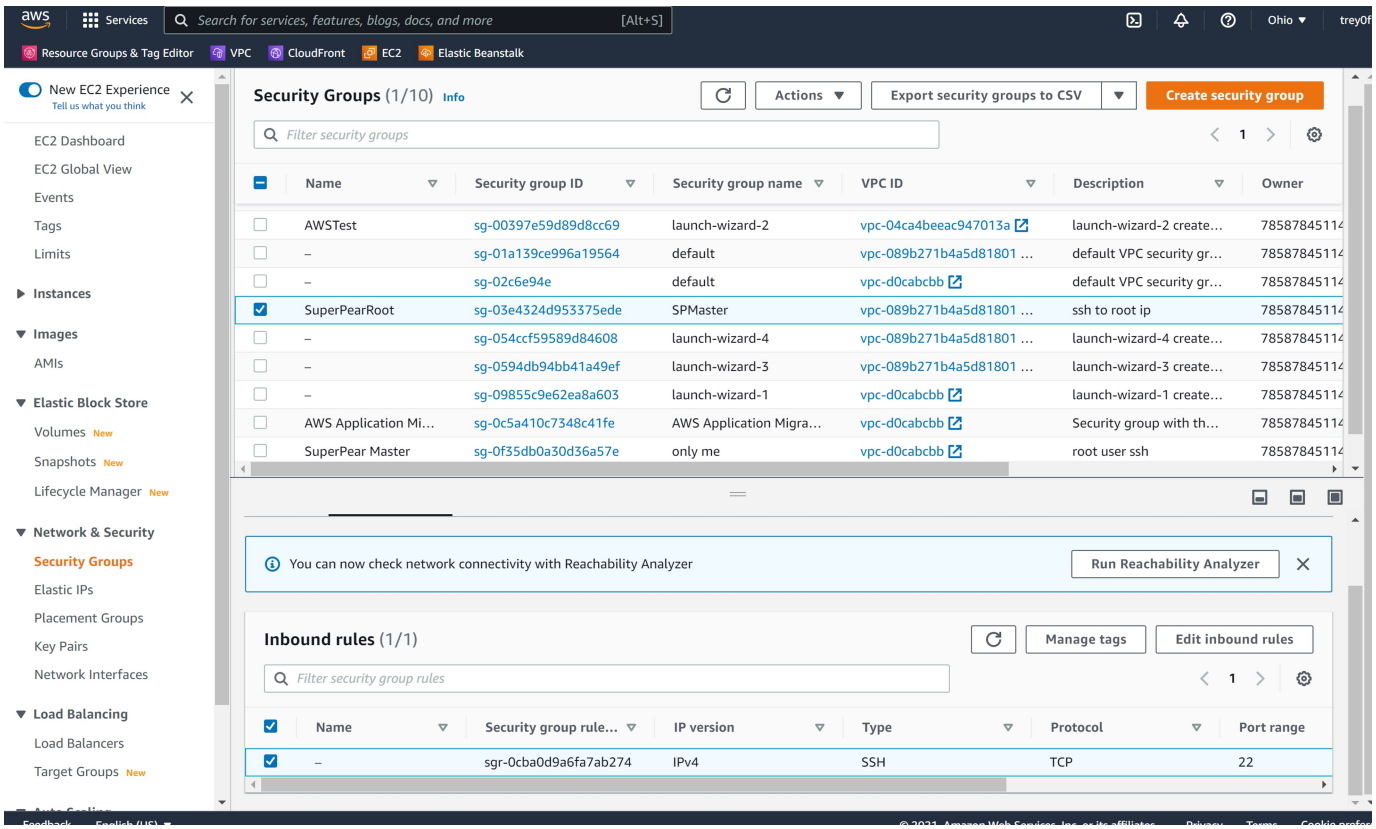


Here is the beforementioned Private route table with an added route that links it to the NAT Gateway

and allows internet access to the VMs in the private subnet.

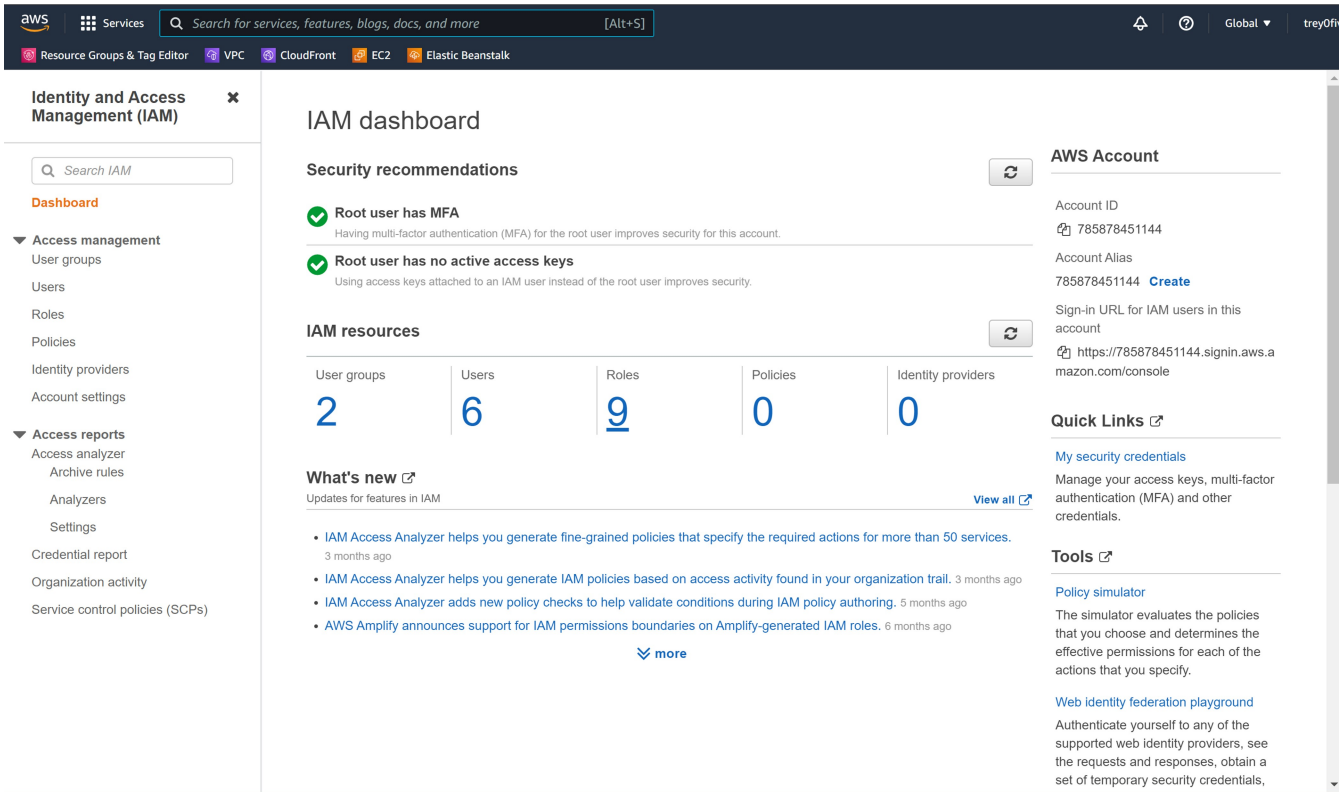


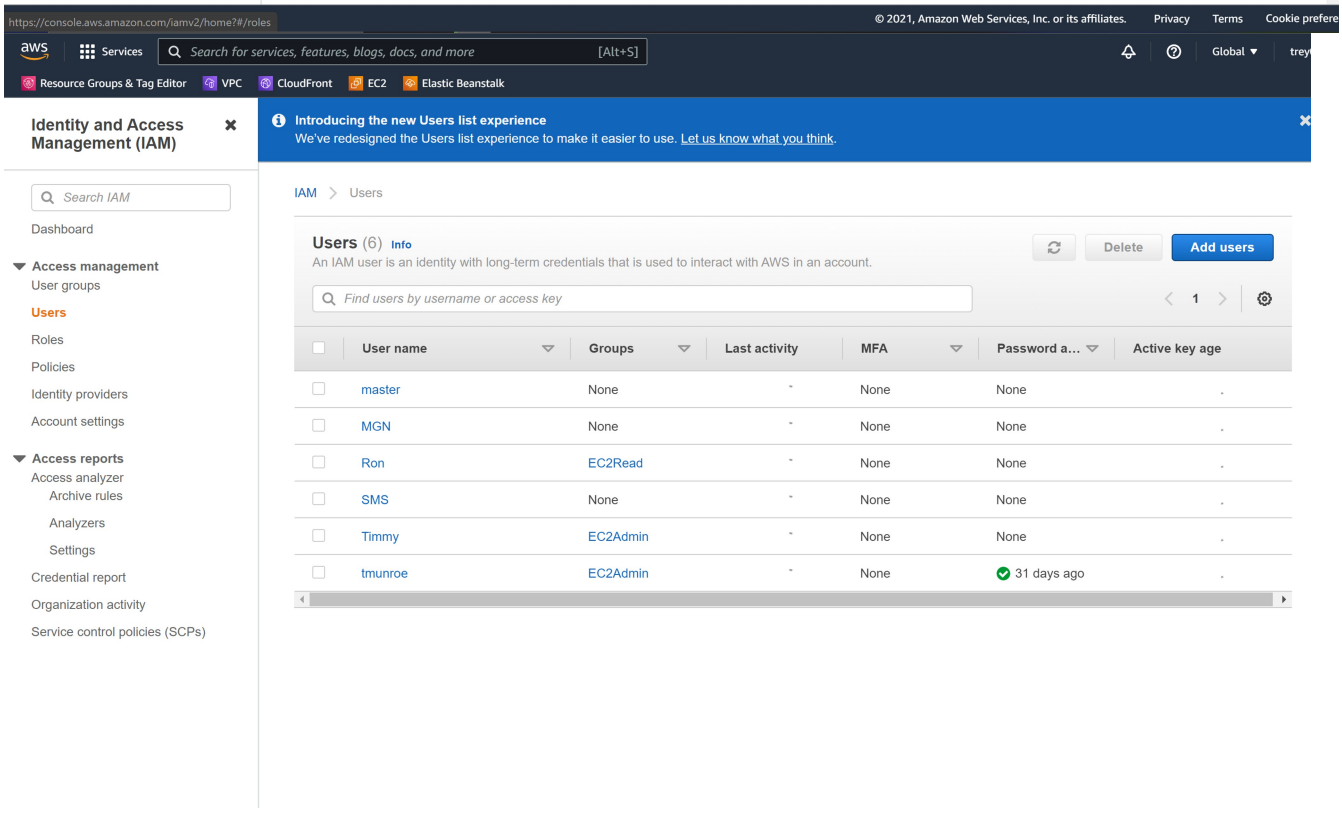
Public route table with an explicit association to the Super Pear public subnet

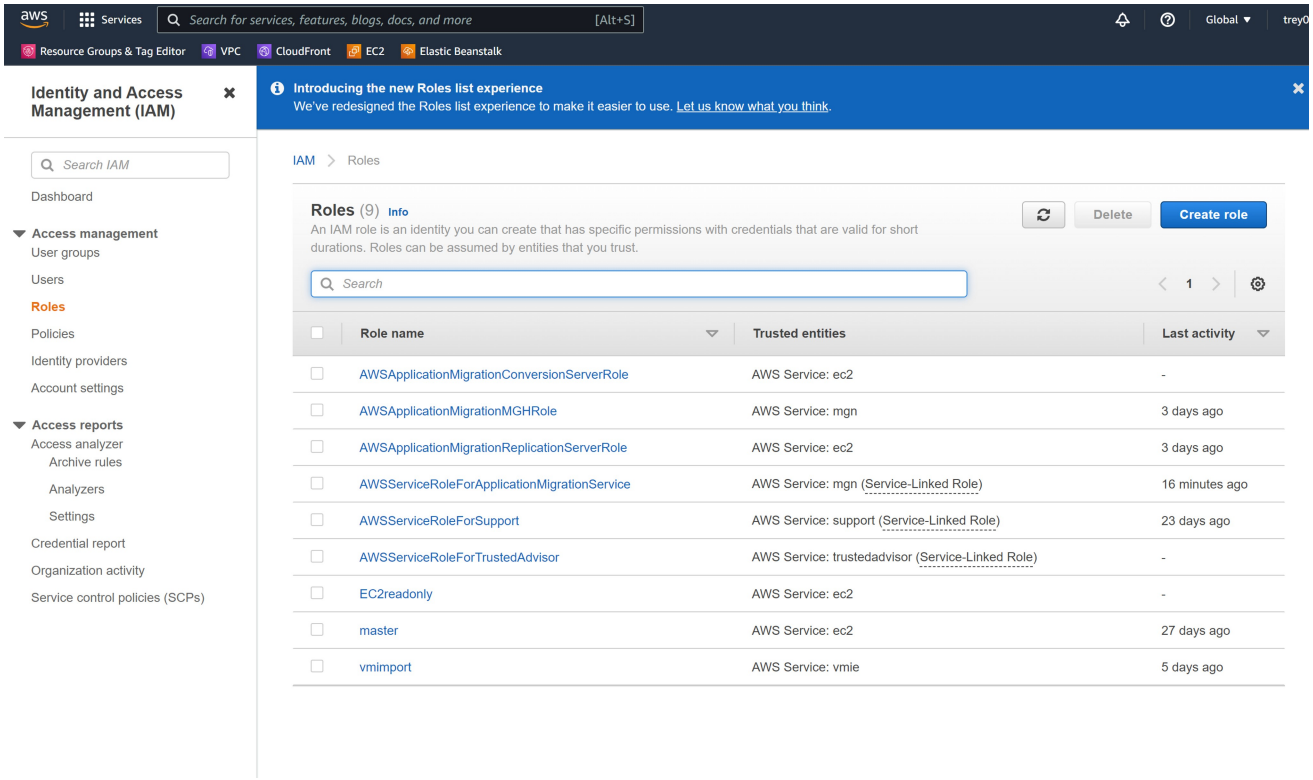


Security group set up for each EC2 instance that allows all outgoing traffic, but only alllows incoming SSH

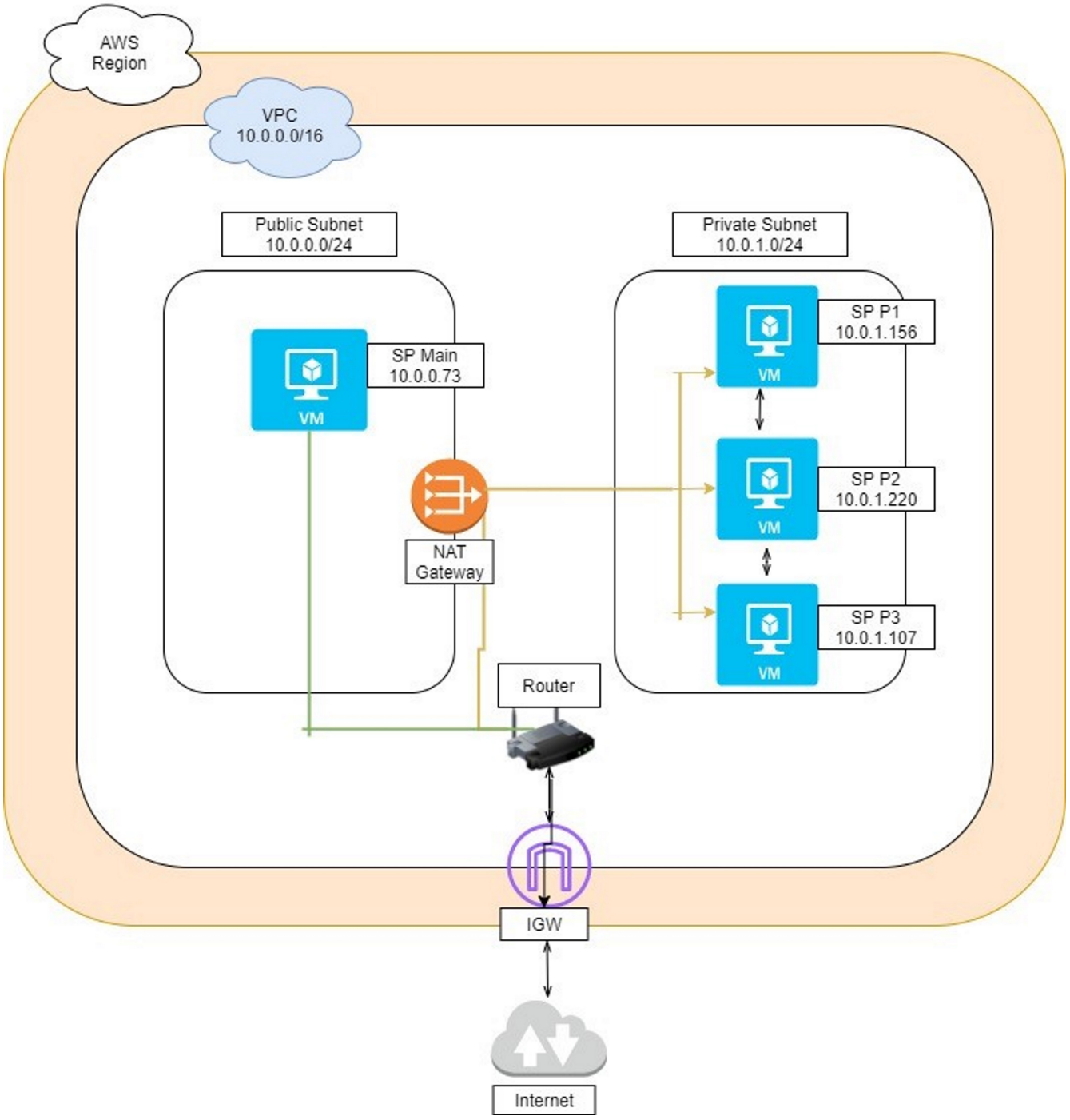
access to only the root user's IP address.







A display of the IAM dashboard. The SuperPear Main VM has root access with MFA set up for extra security. The main VM also has full EC2 admin access. The other 3 VMs on the private subnet only have programmatic access via AWS CLI and secret key and access code. They also only have EC2 read only access.



After completion of the "lift and shift", this is how the network will look and function on AWS!!!