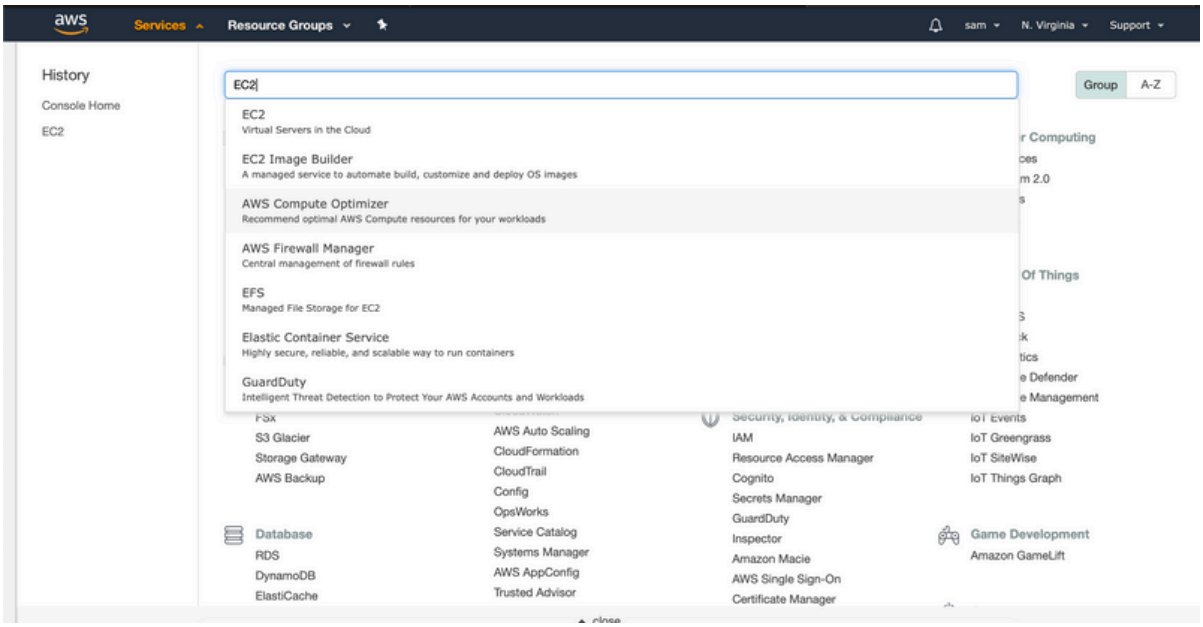
# Hands-On Study Notes

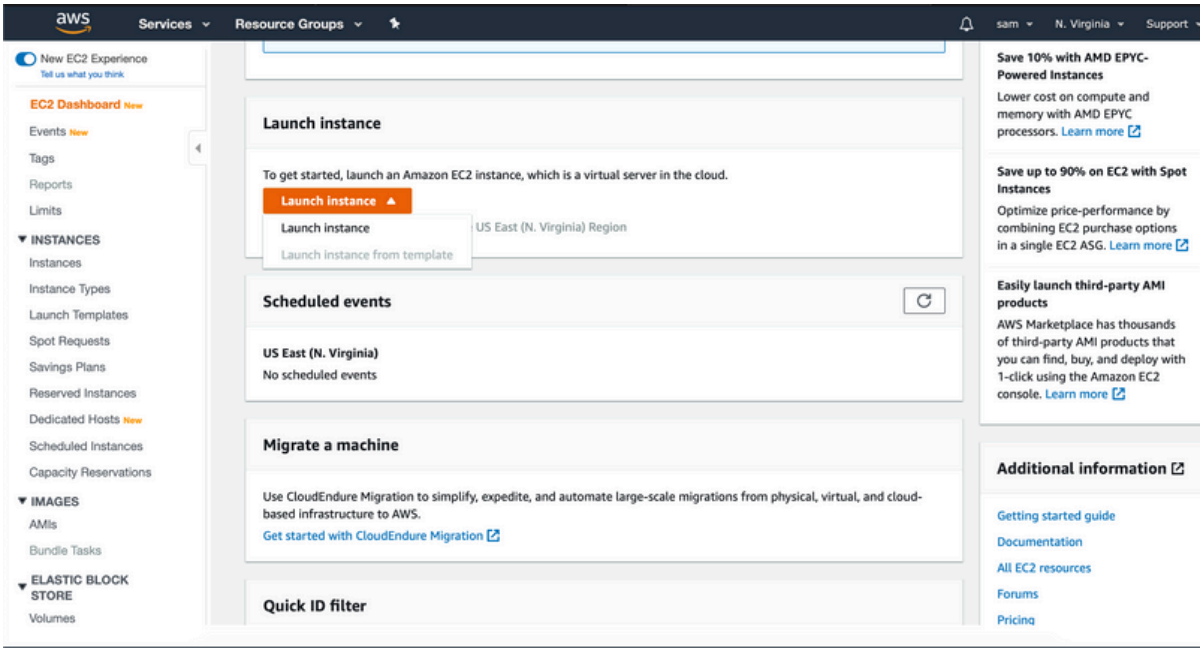
**Andrew Brown**

## How to launch a server using EC2

- 1.1 [ ] Go to services and type EC2 and we will make our way over to the EC2 console



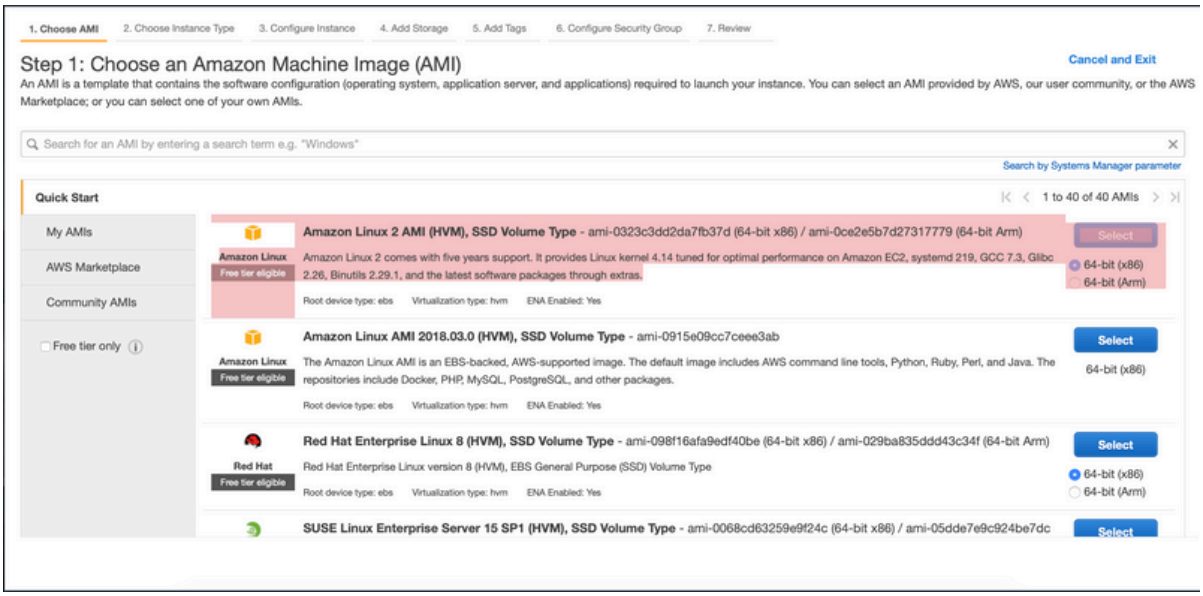

- 1.2 [ ] Scroll down and click on **Launch Instance**




## Choose AMI

We will be presented with a bunch of options to configure our server and we will choose what OS we want to use

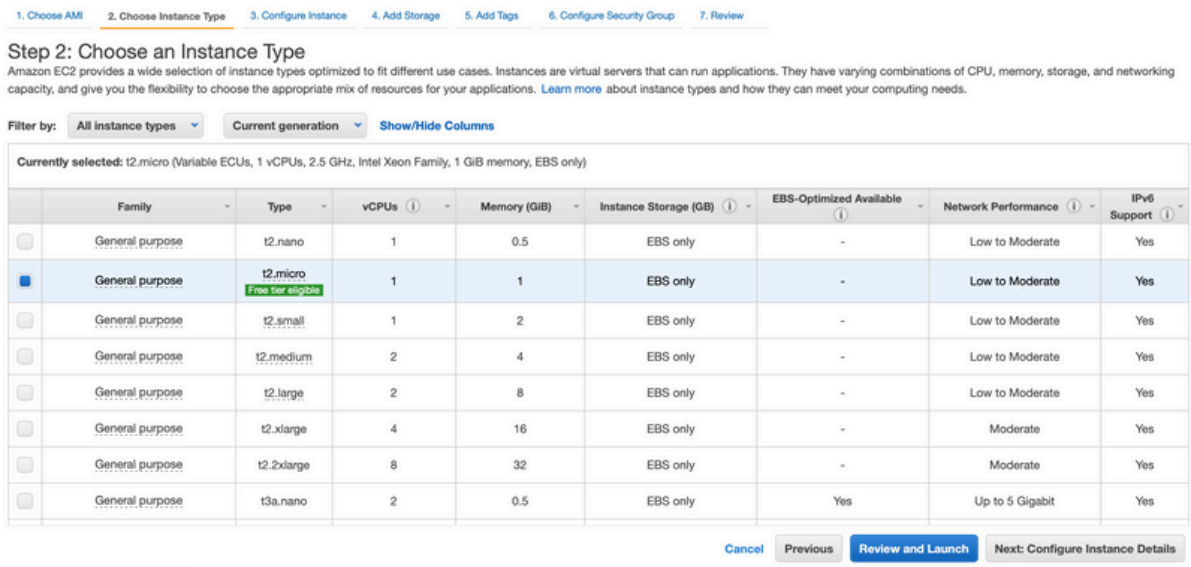
- 1.3 [ ] We will click on **Amazon Linux** because it's part of the Free tier



## Choose an Instance Type

We will choose the size of our server and these are called instance types

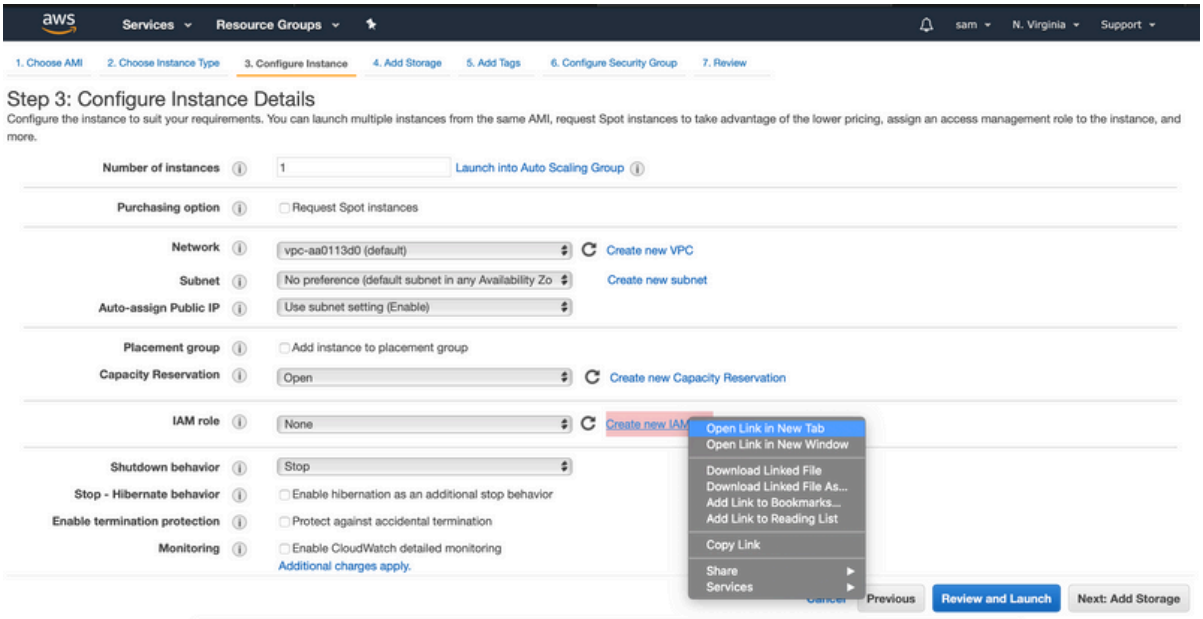
- 1.4 [ ] Click on **t2.micro** because it is part of Free tier and click **Next: Configure Instance Details**




## Configure Instance Details

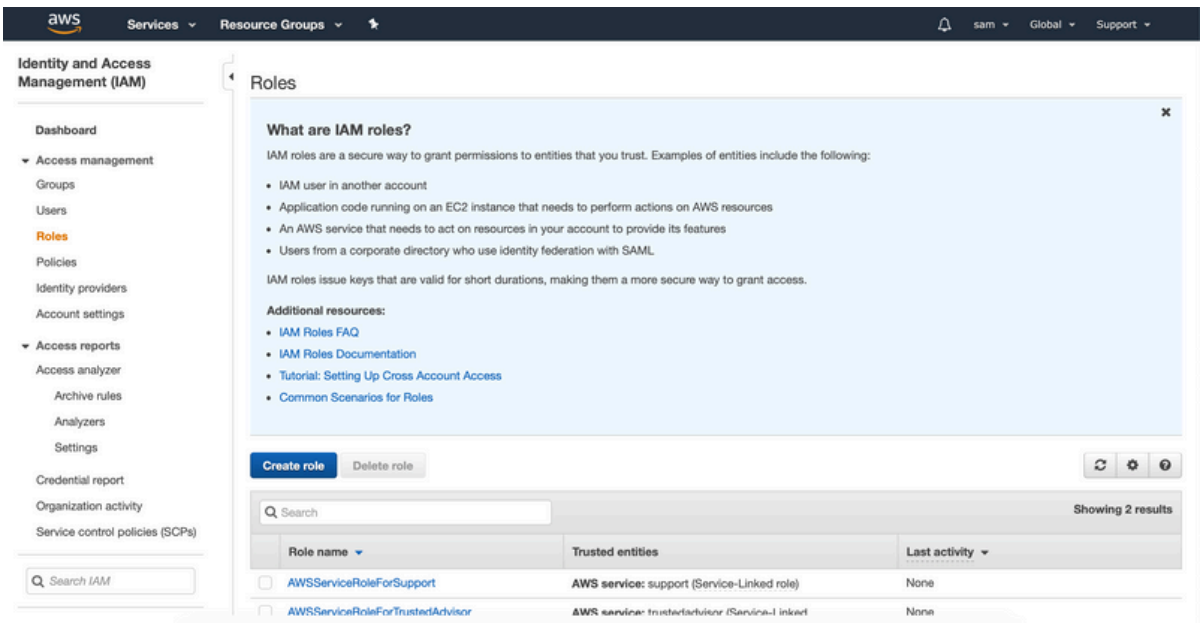
Now going to Instance details we can choose how many instances we want to start

- 1.5 [ ] Right click on **IAM role** and make a new tab

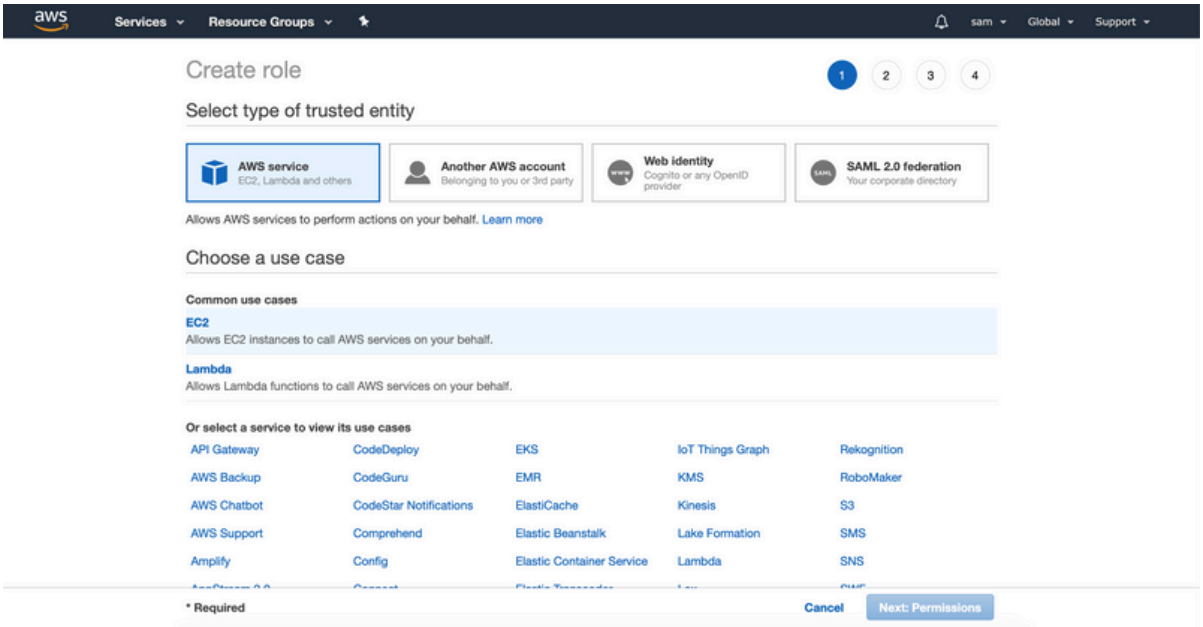



## Create a role

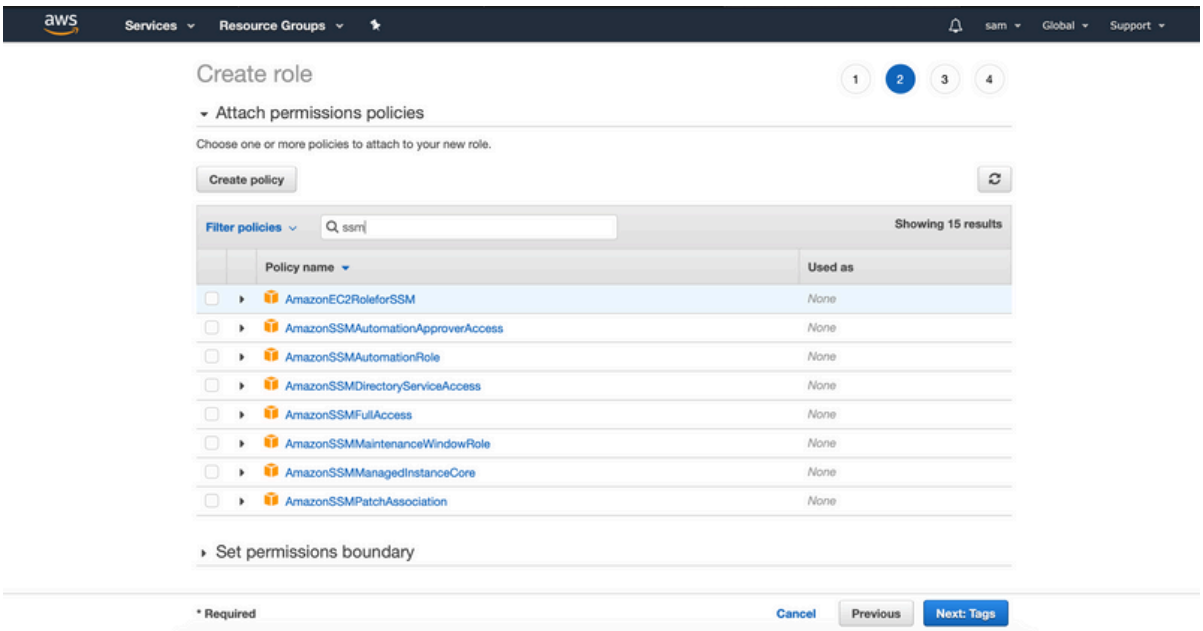
- 1.6 [ ] Go to IAM management console go down and click on **create a new role**

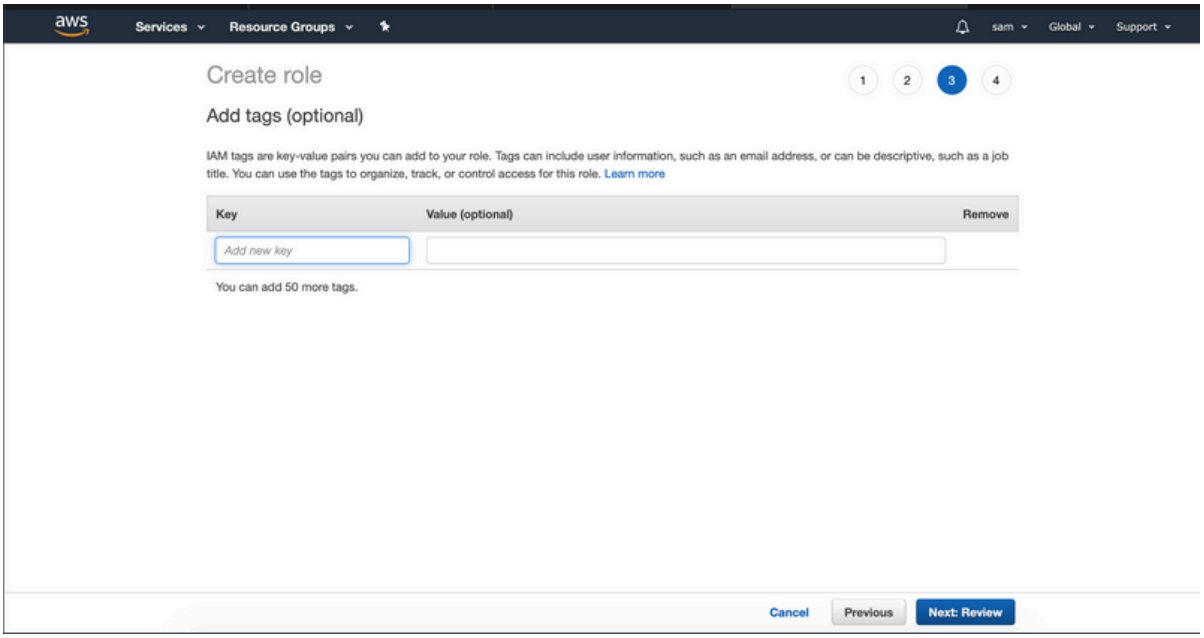



- 1.7 [ ] Click on **EC2** and click **Next: Permissions**

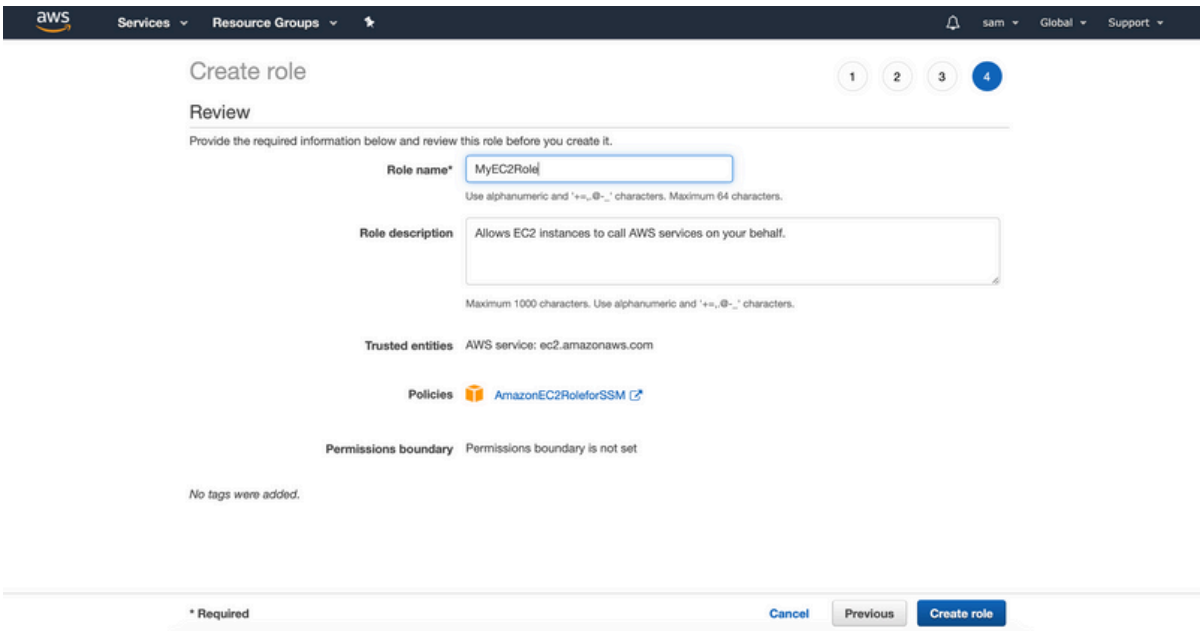


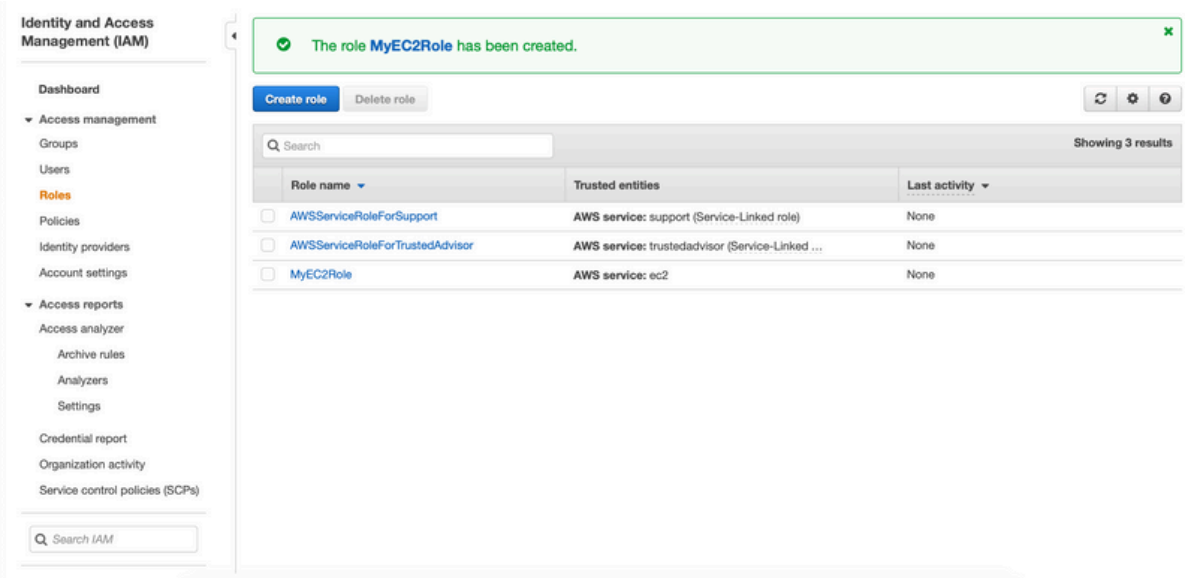

- 1.8 [ ] Type in ssm (simple systems manager which will be to log into that machine) and check mark **AmazonEc2RoleforSSM** and click **Next: Tags** and click **Next review**





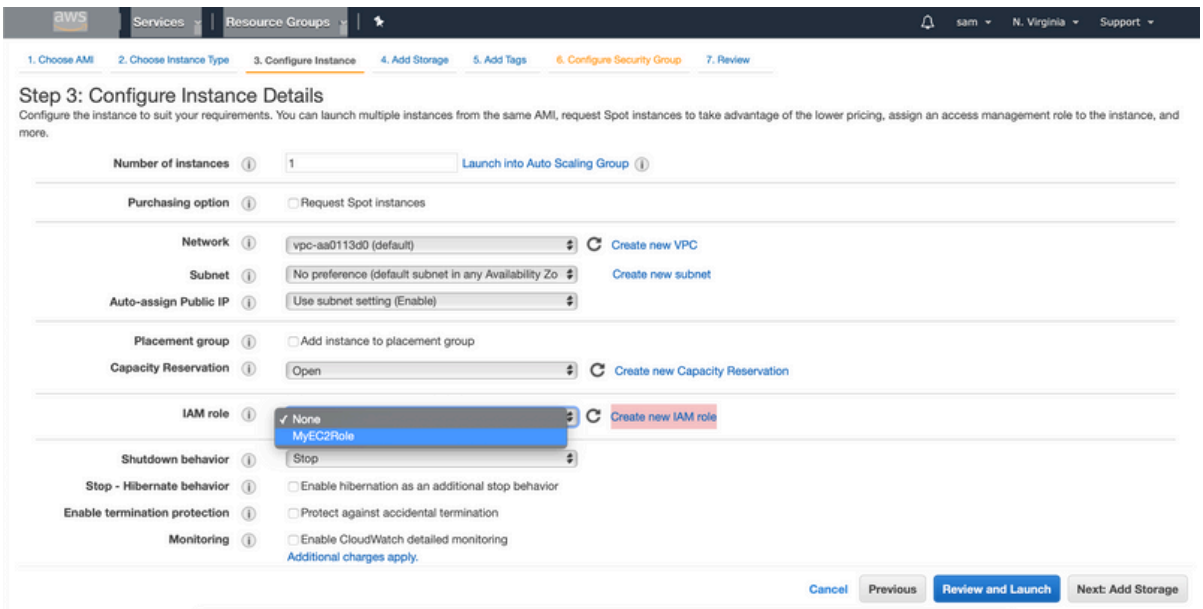

- 1.9 [ ] Type in **MyEc2Role** in Role name and click **Create role**. Now that role has been created, we will just go ahead and close that tab





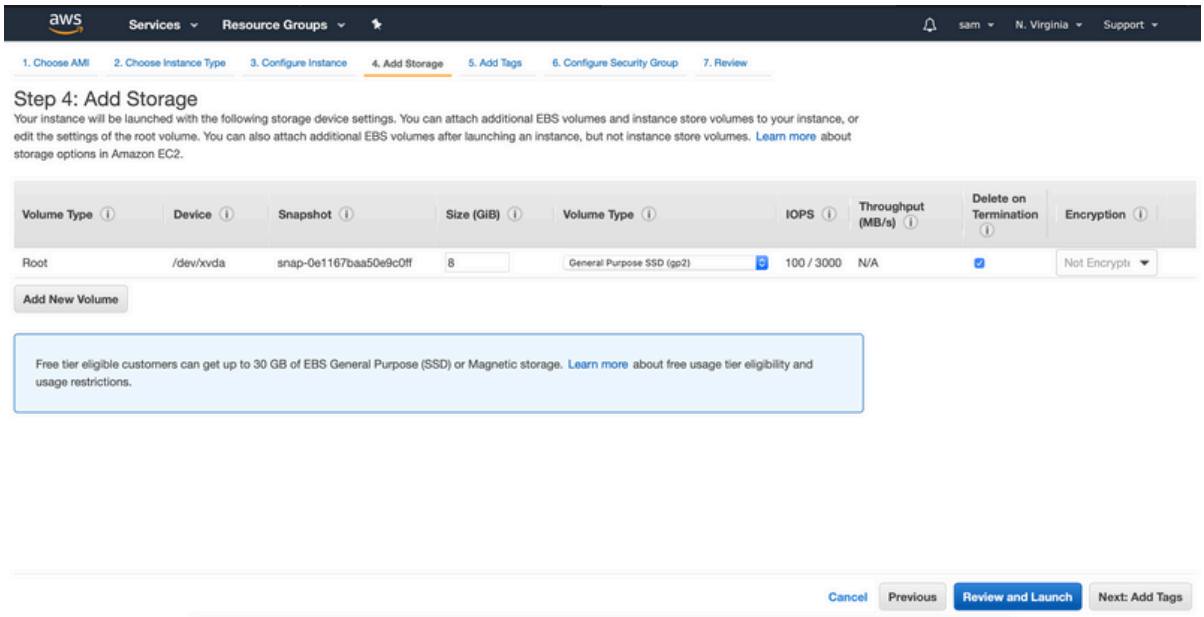

## Configure Instance

- 1.10 [ ] Go to IAM role and and click **refresh and choose MyEC2Role** and now we will leave everything else blank and click **Next: Add Storage**



## Add Storage

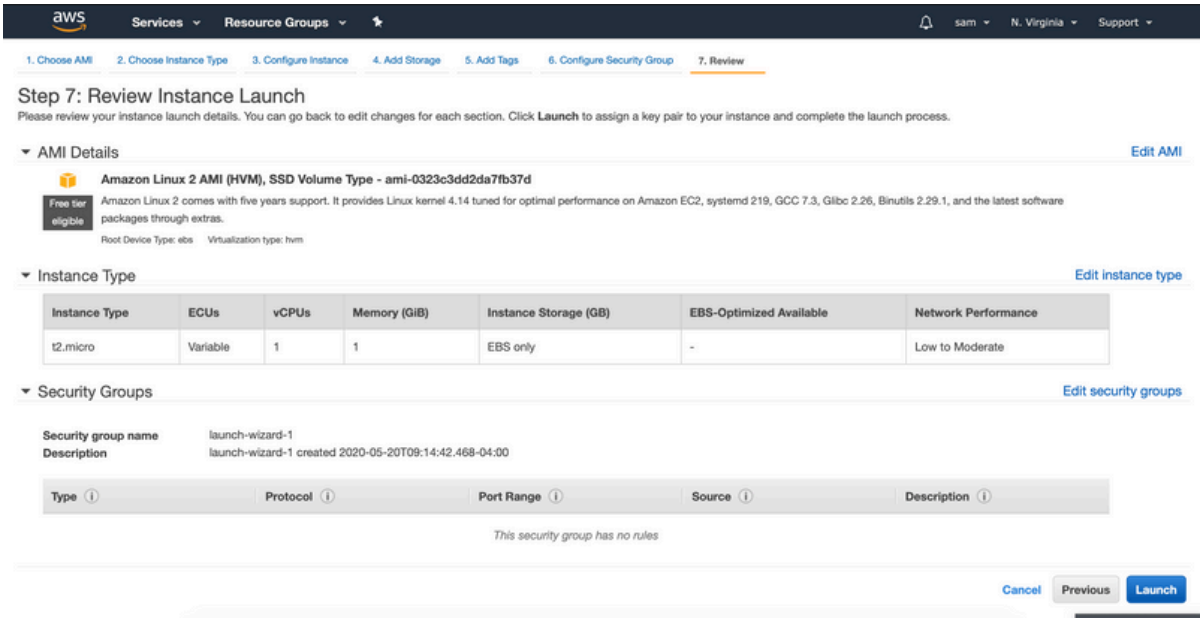
- 1.11 [ ] Now you can choose your storage we will leave it at 8GB and we will stick with **General Purpose SSD** for Volume type and click **Review and Launch**

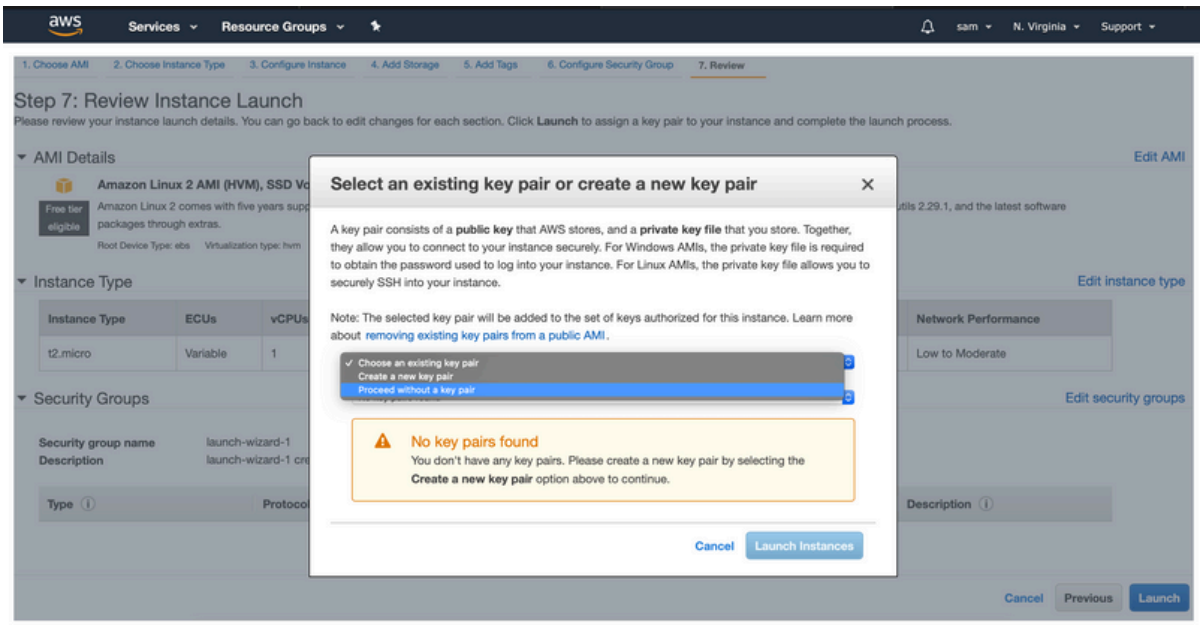


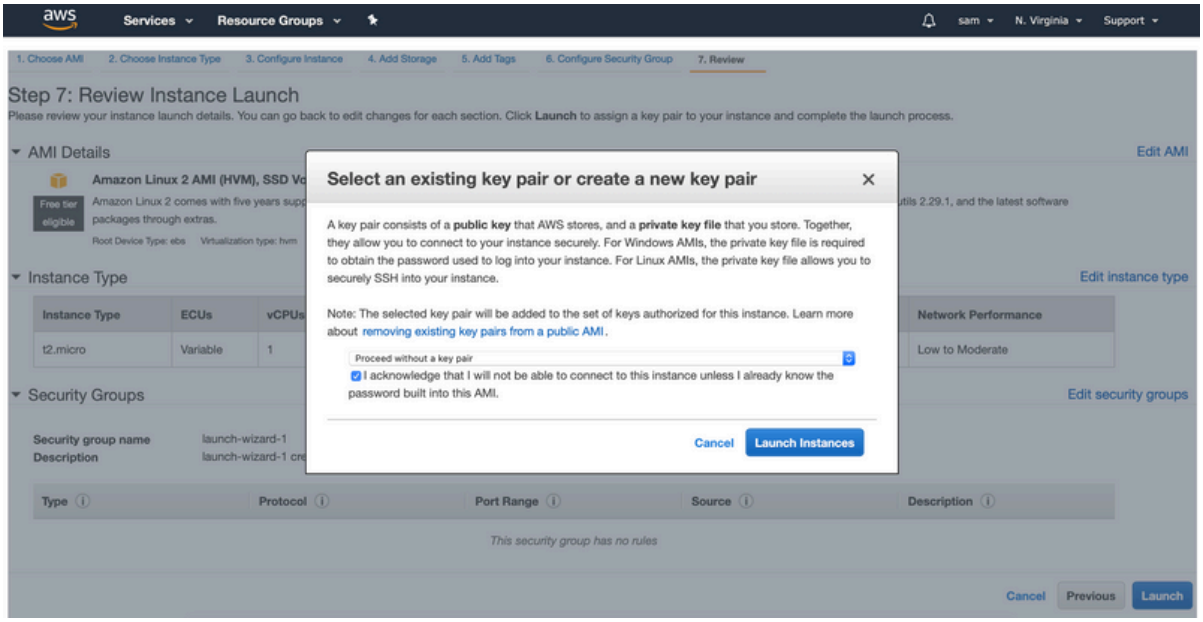

## Review

- 1.12 [ ] Click **Launch** It will ask you to create a key pair and click the **drop down box and select Proceed without a key pair**

```
        Check mark `I acknowledge that I will not be able to connect to this
instance unless I already know the password built into
        this AMI` and Click `Launch`
```

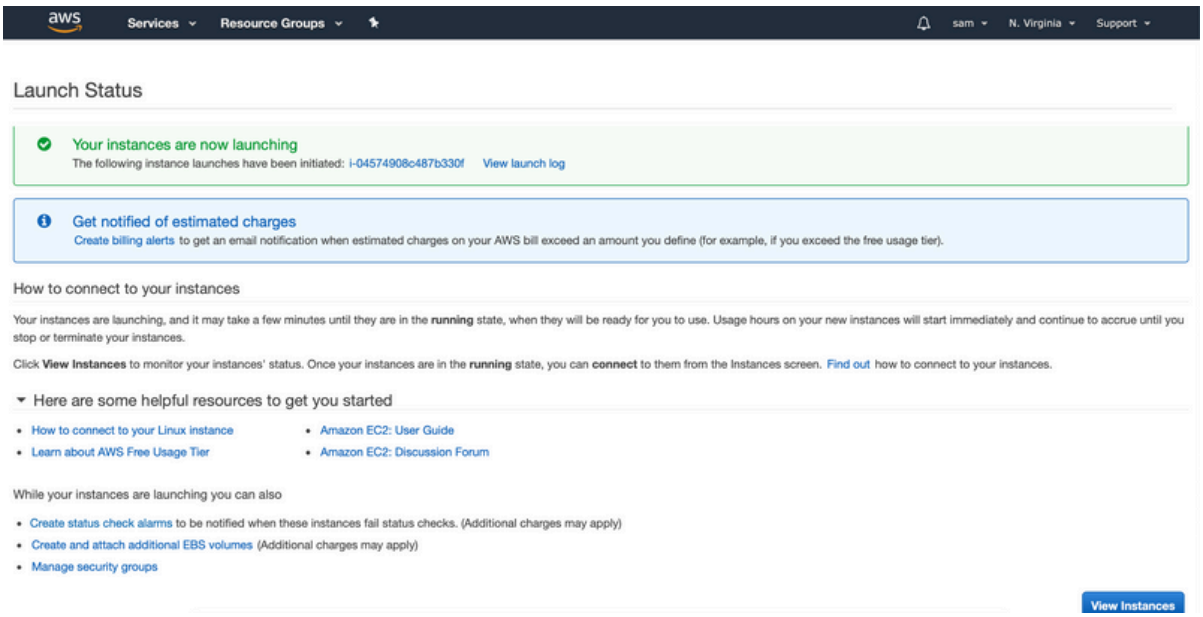

aws
Services
Resource Groups
sam
N. Virginia
Support
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
AMI Details
Edit AMI
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0323c3dd2da7fb37d
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs
Virtualization type: hvm
Instance Type
Edit instance type
Instance Type
ECUs
vCPUs
Memory (GiB)
Instance Storage (GB)
EBS-Optimized Available
Network Performance
t2.micro
Variable
1
1
EBS only
-
Low to Moderate
Security Groups
Edit security groups
Security group name
launch-wizard-1
Description
launch-wizard-1 created 2020-05-20T09:14:42.468-04:00
Type
Protocol
Port Range
Source
Description
This security group has no rules
Cancel
Previous
Launch


aws
Services
Resource Groups
sam
N. Virginia
Support
Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click Launch to assign a key pair to your instance and complete the launch process.
AMI Details
Edit AMI
Instance Type
Edit instance type
Security Groups
Edit security groups
Select an existing key pair or create a new key pair
A key pair consists of a public key that AWS stores, and a private key file that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.
Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.
Choose an existing key pair
Create a new key pair
Proceed without a key pair
No key pairs found
You don't have any key pairs. Please create a new key pair by selecting the Create a new key pair option above to continue.
Cancel
Launch Instances
Cancel
Previous
Launch

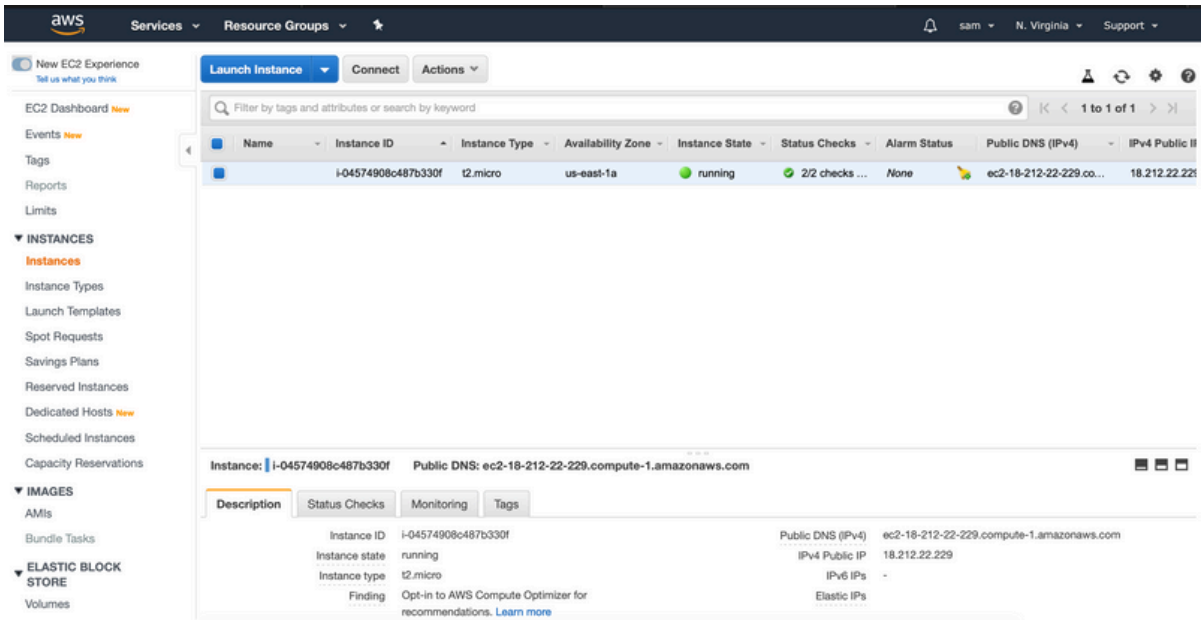



## Launch Status

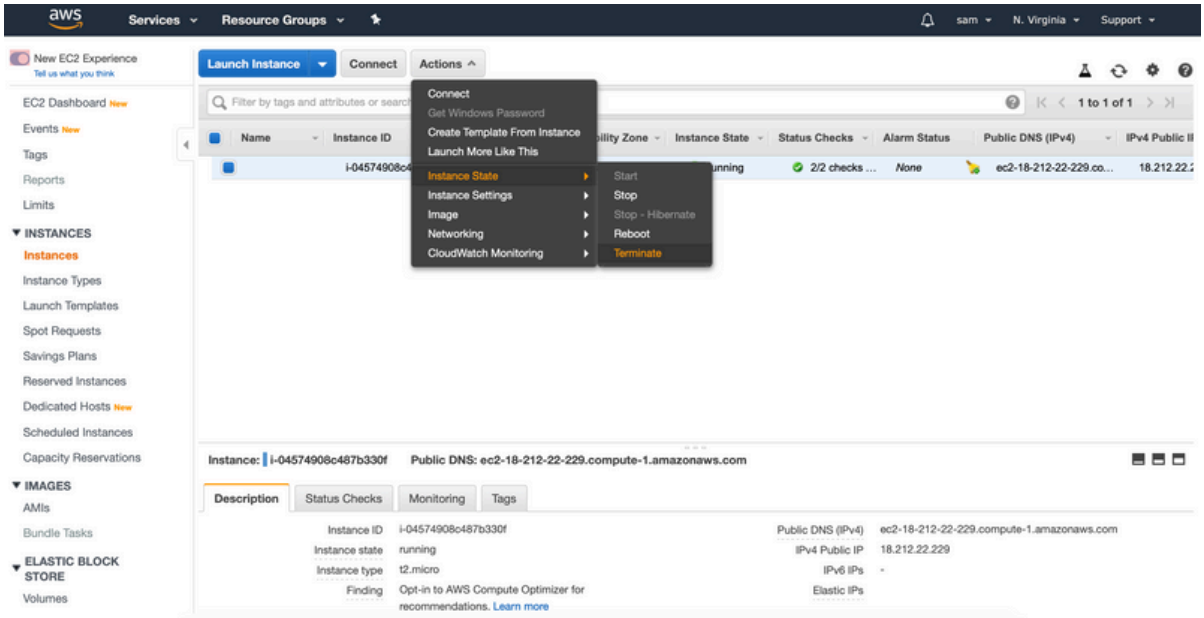
- 1.13 [ ] Click **View Instances** to be able to view it and now this instance is launching and you will see it in pending state (It will turn from yellow to green and we will wait for it to initialize)



- 1.14 [ ] After a 3 - 4 min wait the server is now running and it will have 2 checks which means the server is in good shape




> *Were not going to shut the server down just yet because we chose the free tier but if you were paying for it you would want to shut it down when your done with it. To shut the server down you would click on Actions, Instance State and Terminate (you can also click stop and that would just stop the server and not destroy it)*

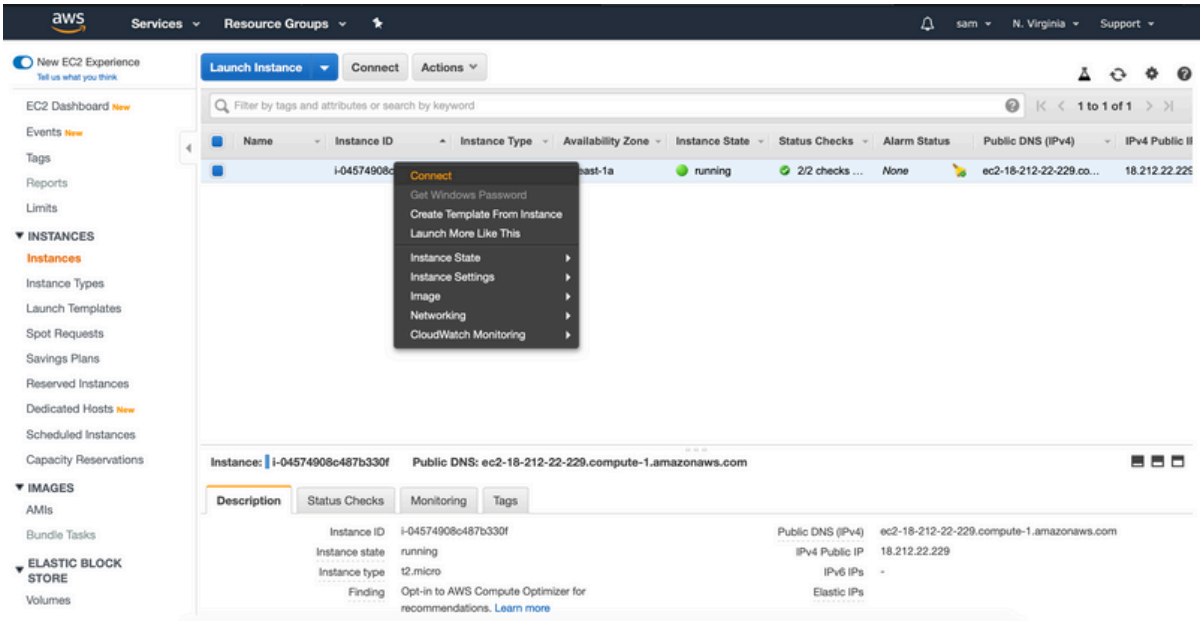


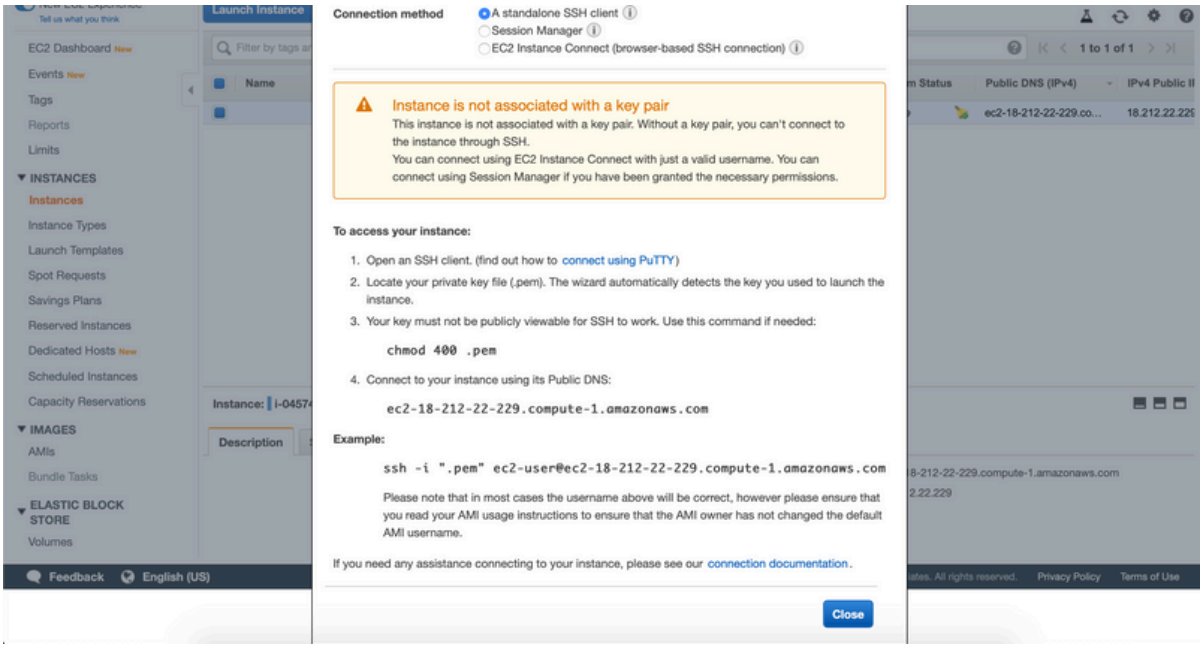

In the next video we will learn how to get access to this instance

## Sessions Manager

There are a couple of ways we can get into this instance. One way is using ssh so if we had created that key pair we could have used it to get into that server or we can use ssm (simple systems manager) and AWS's recommended way.

- 1.1 [ ] Before we go over to ssm right-click on **Instance ID** and click on **Connect** and click **Close**

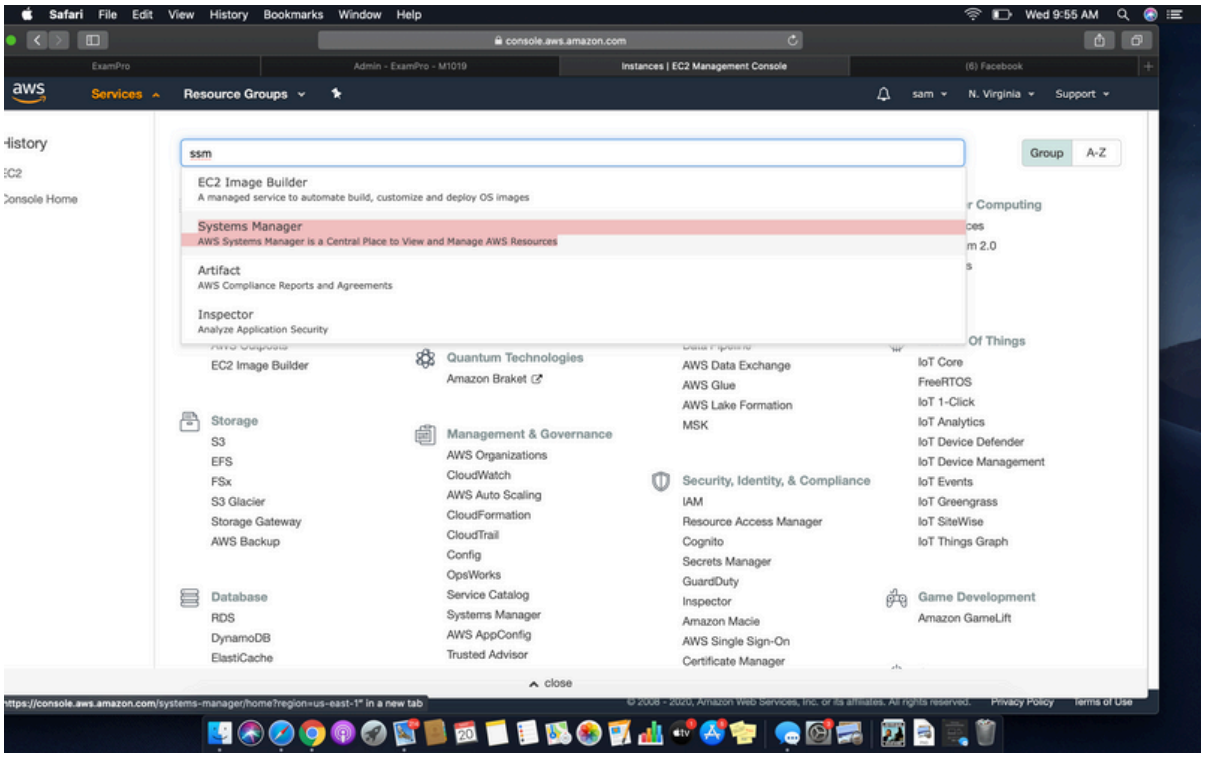


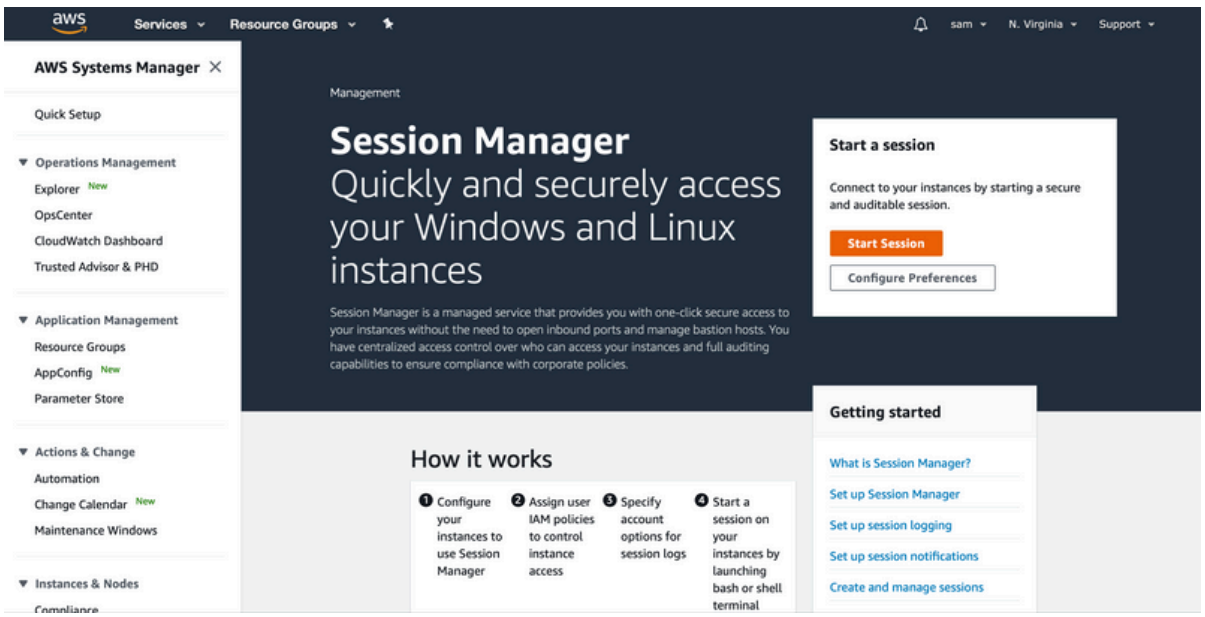




## Start a session

- 1.2 [ ] Go to **Services** at the top and type in **ssm** and click on **Systems Manager** on the left-hand side, once on that page scroll down on the left-hand side and click on **Session Manager** and click on the **Start Session** orange

button


Safari File Edit View History Bookmarks Window Help
Wed 9:55 AM
console.aws.amazon.com
Instances | EC2 Management Console
aws
Services
Resource Groups
sam
N. Virginia
Support
ssm
EC2 Image Builder
A managed service to automate build, customize and deploy OS images
Systems Manager
AWS Systems Manager is a Central Place to View and Manage AWS Resources
Artifact
AWS Compliance Reports and Agreements
Inspector
Analyze Application Security
Group
A-Z
EC2 Image Builder
Quantum Technologies
Amazon Braket
AWS Data Exchange
AWS Glue
AWS Lake Formation
MSK
IoT Core
FreeRTOS
IoT 1-Click
IoT Analytics
IoT Device Defender
IoT Device Management
IoT Events
IoT Greengrass
IoT SiteWise
IoT Things Graph
Storage
S3
EFS
FSx
S3 Glacier
Storage Gateway
AWS Backup
Management & Governance
AWS Organizations
CloudWatch
AWS Auto Scaling
CloudFormation
CloudTrail
Config
OpsWorks
Service Catalog
Systems Manager
AWS AppConfig
Trusted Advisor
Security, Identity, & Compliance
IAM
Resource Access Manager
Cognito
Secrets Manager
GuardDuty
Inspector
Amazon Macie
AWS Single Sign-On
Certificate Manager
Database
RDS
DynamoDB
ElastiCache
Game Development
Amazon GameLift
close


aws
Services
Resource Groups
sam
N. Virginia
Support
AWS Systems Manager
Quick Setup
Operations Management
Explorer New
OpsCenter
CloudWatch Dashboard
Trusted Advisor & PHD
Application Management
Resource Groups
AppConfig New
Parameter Store
Actions & Change
Automation
Change Calendar New
Maintenance Windows
Instances & Nodes
Management
Session Manager
Quickly and securely access your Windows and Linux instances
Session Manager is a managed service that provides you with one-click secure access to your instances without the need to open inbound ports and manage bastion hosts. You have centralized access control over who can access your instances and full auditing capabilities to ensure compliance with corporate policies.
Start a session
Connect to your instances by starting a secure and auditable session.
Start Session
Configure Preferences
How it works
1 Configure your instances to use Session Manager
2 Assign user IAM policies to control instance access
3 Specify account options for session logs
4 Start a session on your instances by launching bash or shell terminal
Getting started
What is Session Manager?
Set up Session Manager
Set up session logging
Set up session notifications
Create and manage sessions

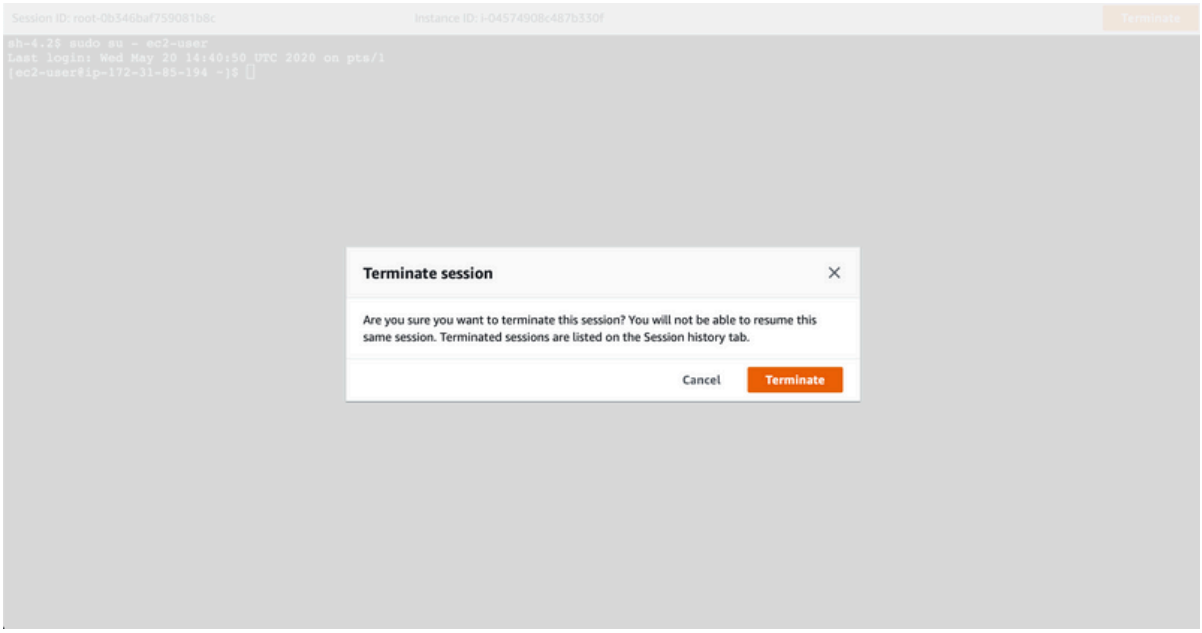
## We have our Instance

- 1.3 [ ] Select the **Instance** and click **Start Session**

- 1.4 [ ] It logs into the root user not the EC2 user, so we have to type in **sudo su - ec2-user**

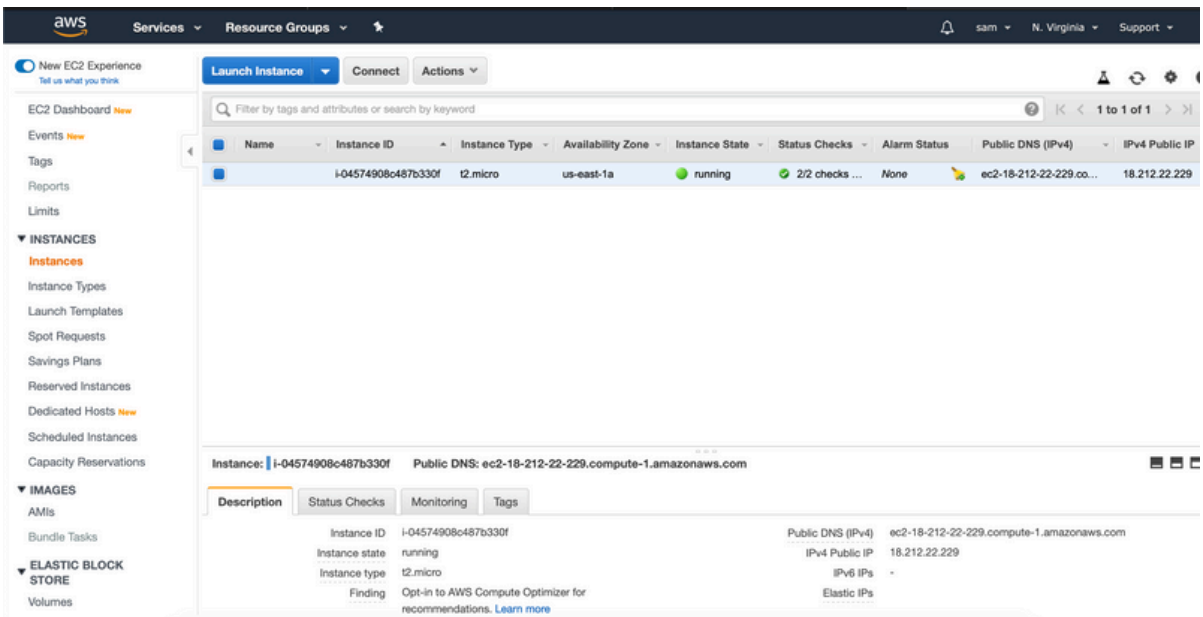

## Terminate: We're not going to do much with it today so you will go ahead and terminate it

- 1.5 [ ] Click **Terminate**



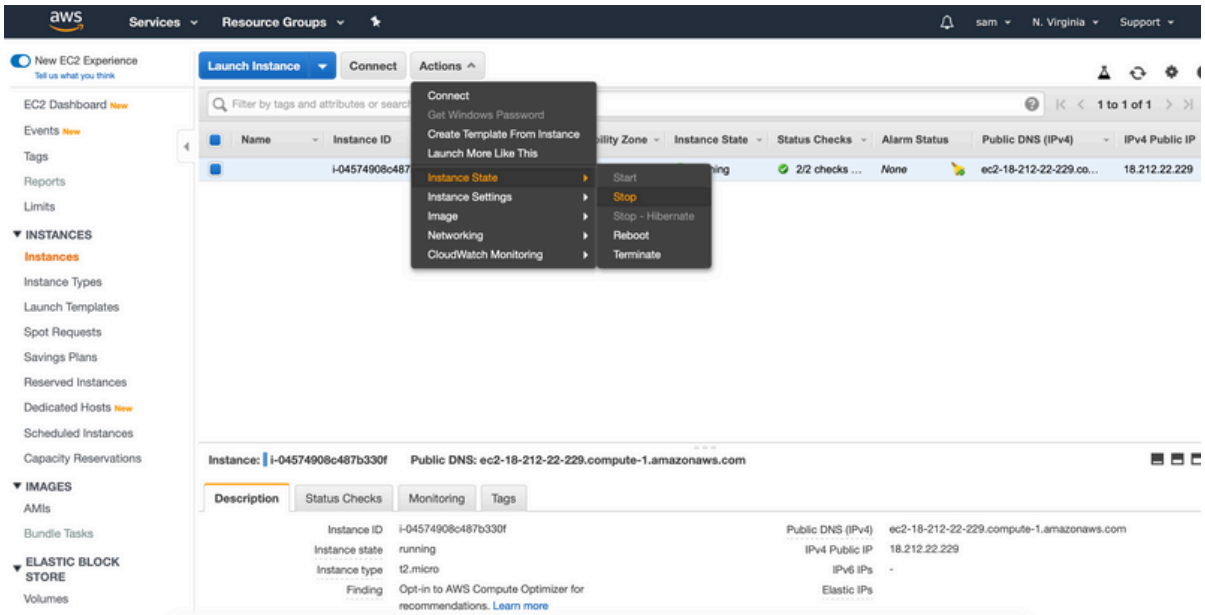

## Go back to EC2

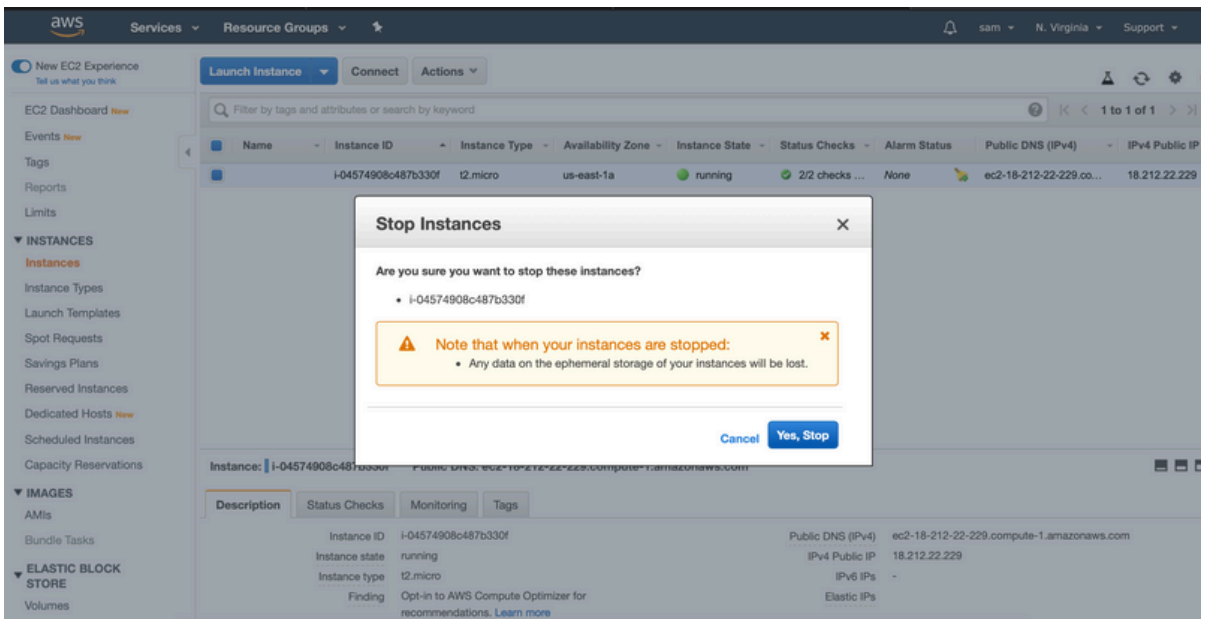
- 1.6 [ ] Click on **Services** at the top and type in **EC2** and go to the left-hand side and click on **Instances**



## Stop Instance

- 1.7 [ ] Click on **Actions Instance State** and **Stop** and select **Yes, Stop**



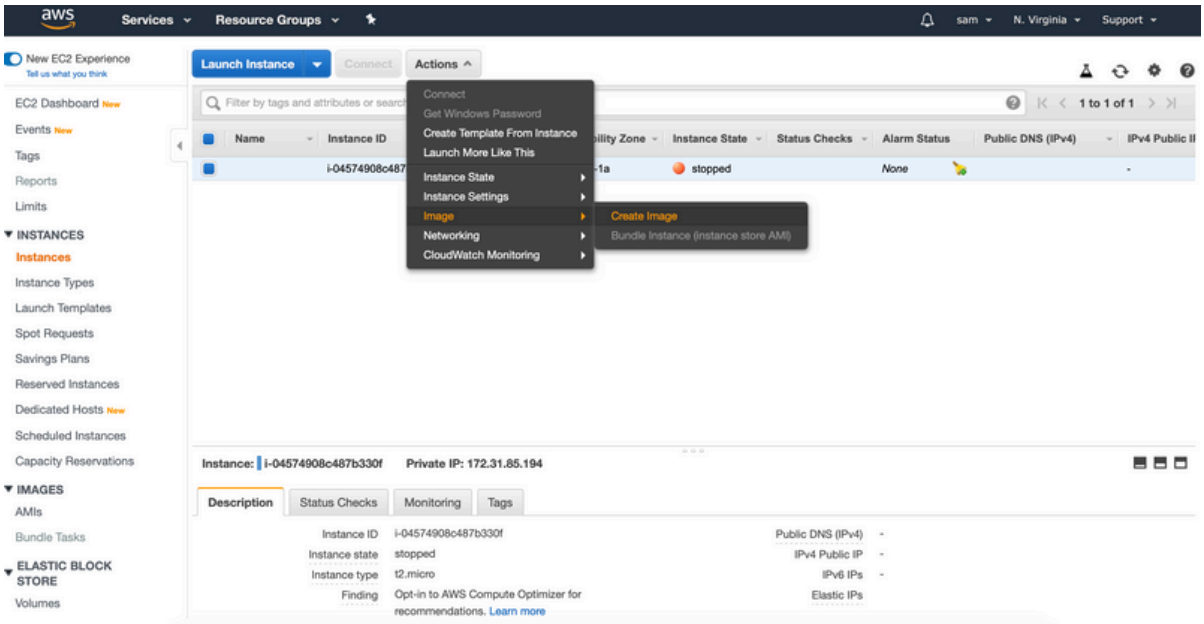

Next, we will create an AMI

# Creating an AMI:

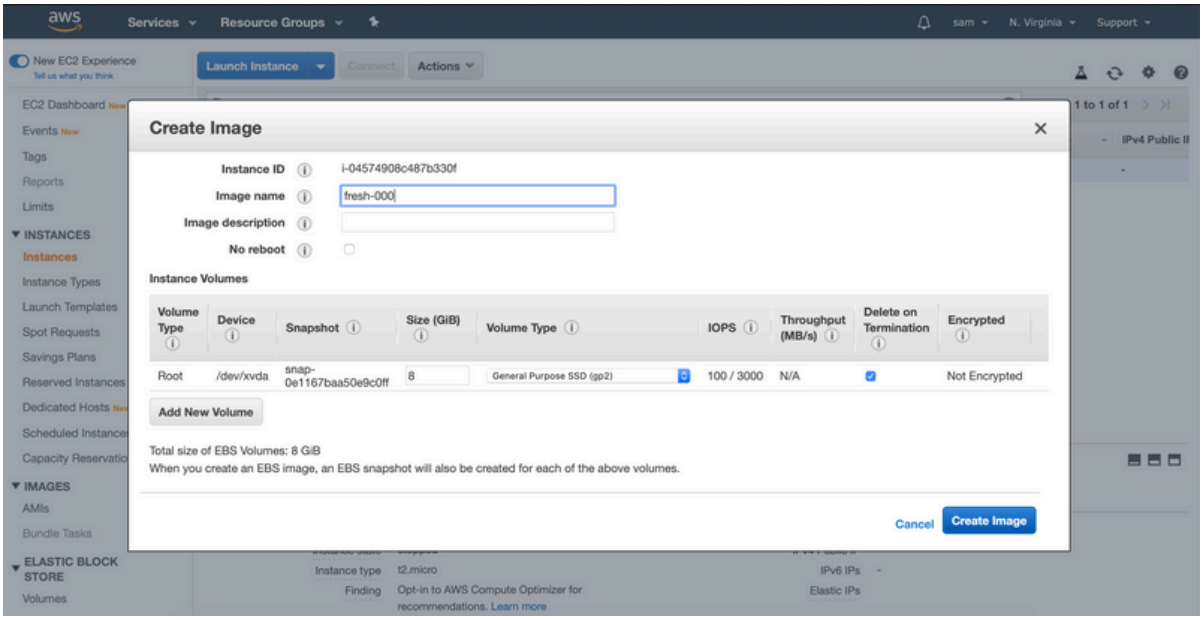
We are going to learn how to create an AMI (which is a snapshot or saving a copy of your entire server)

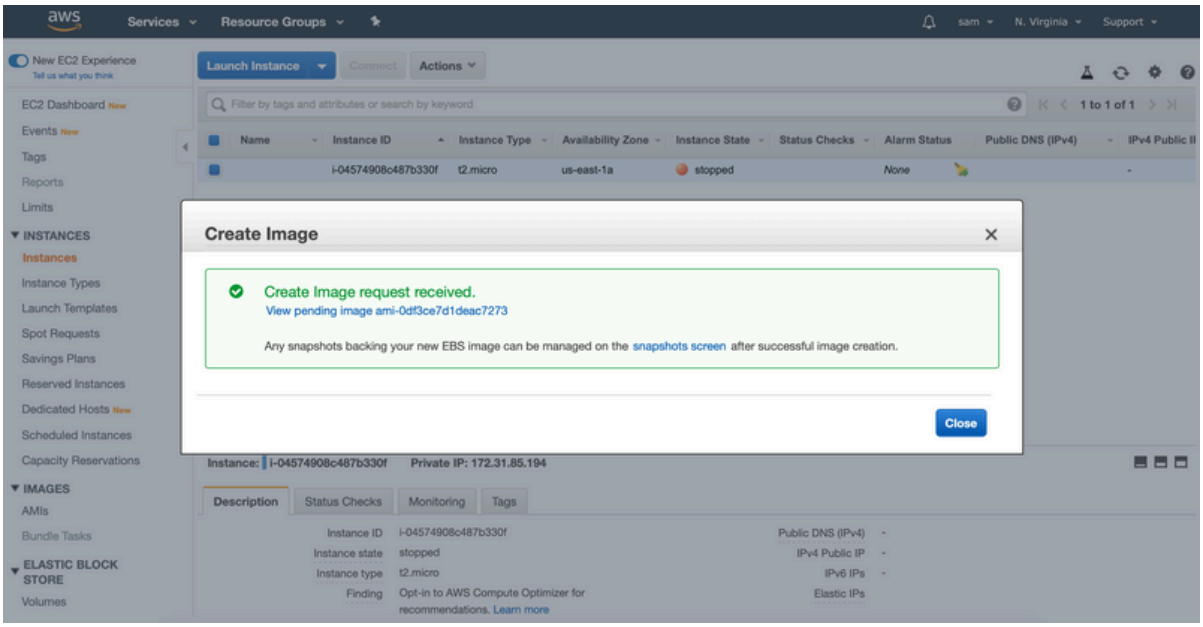
## Create Image

- 1.1 [ ] Go to the top of the page and select **Actions** then **Image** and **Create Image**



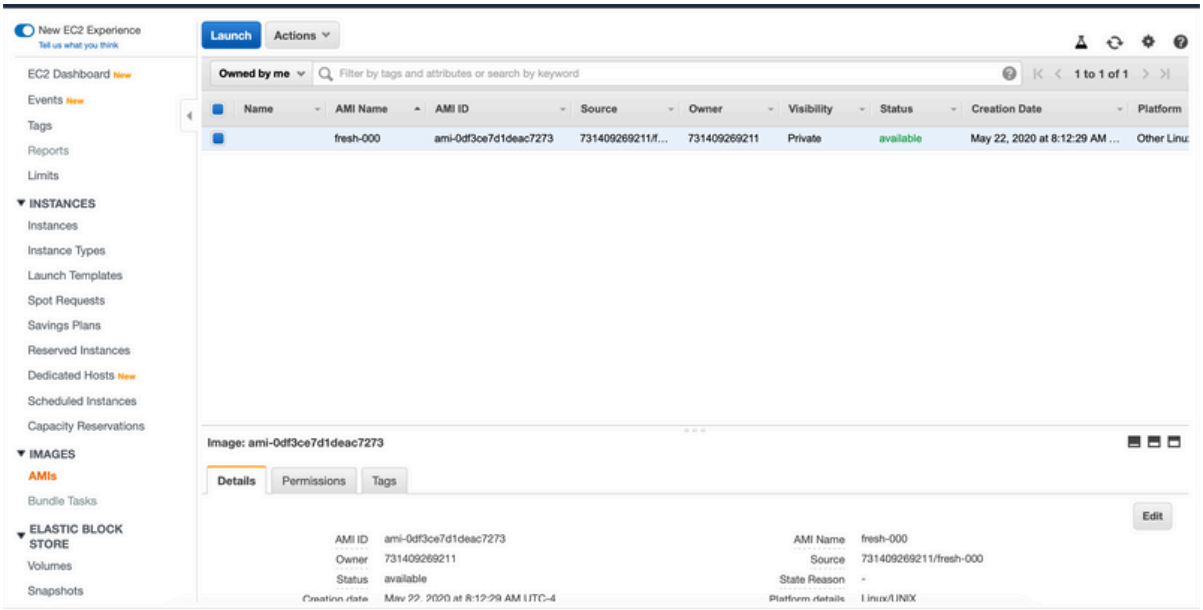

- 1.2 [ ] Fill out **Image Name** and click **Create Image** and click **View Pending image ami-0ae4eee56681b6324**



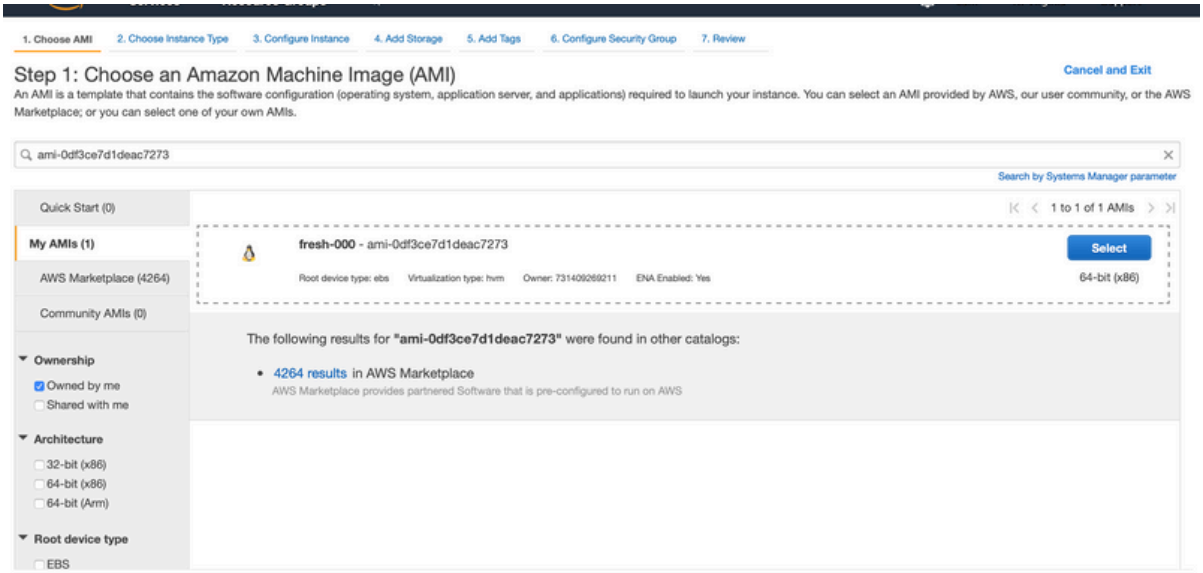

## Launch Server

- 1.3 [ ] Click **Launch**



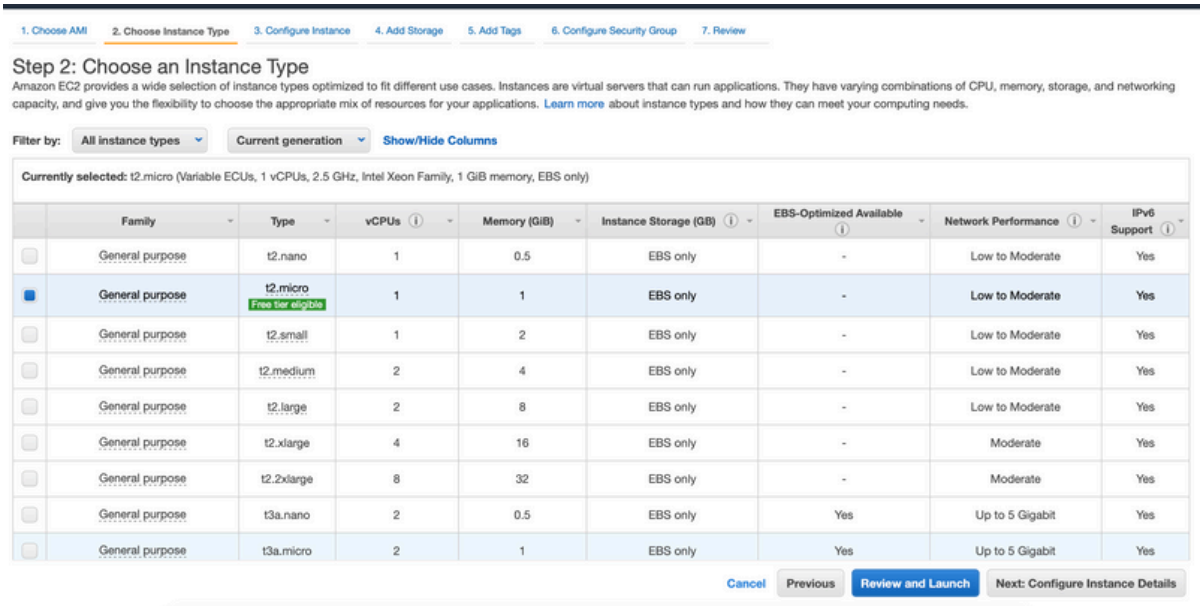

## Choose AMI

- 1.4 [ ] Click on **Choose AMI** at the top and you can see it chose fresh-000 - ami-0df3ce7d1deac7273



### Choose Instance Type: Is a way for us to upgrade our server, make other changes to it or just so we have another copy of it so we can launch multiple servers

- 1.5 [ ] Click back on **Choose Instance Type** at the top of the screen and click **Cancel** at the bottom




- 1.6 [ ] To get back to AMI just click on the left hand side

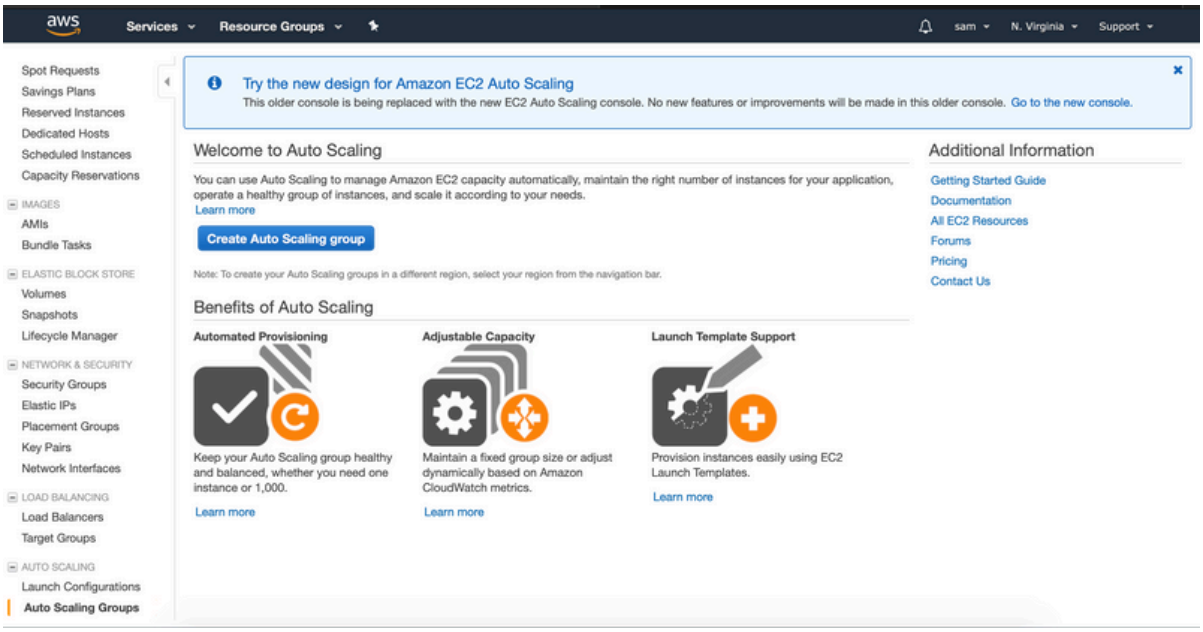
Now we will move onto Auto Scaling groups

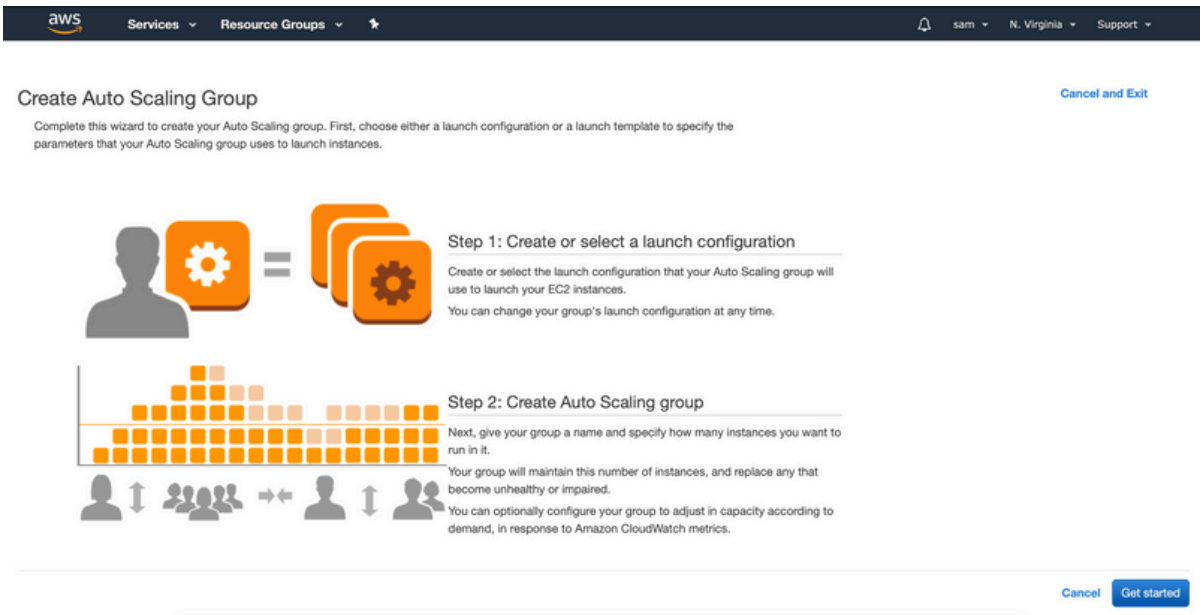
## Autoscaling Groups

What an Autoscaling group does is it allows you to insure that multiple instances or severs are running. So if you always want to always guarantee that one server is always running then auto scaling group would have a rule to check to say that at least one is running and if not to launch a new server. Also, auto scaling groups are used to meet the demand of whatever traffic you have.

- 1.1 [ ] Scroll down on the left hand side and click on **Auto Scaling Groups** and

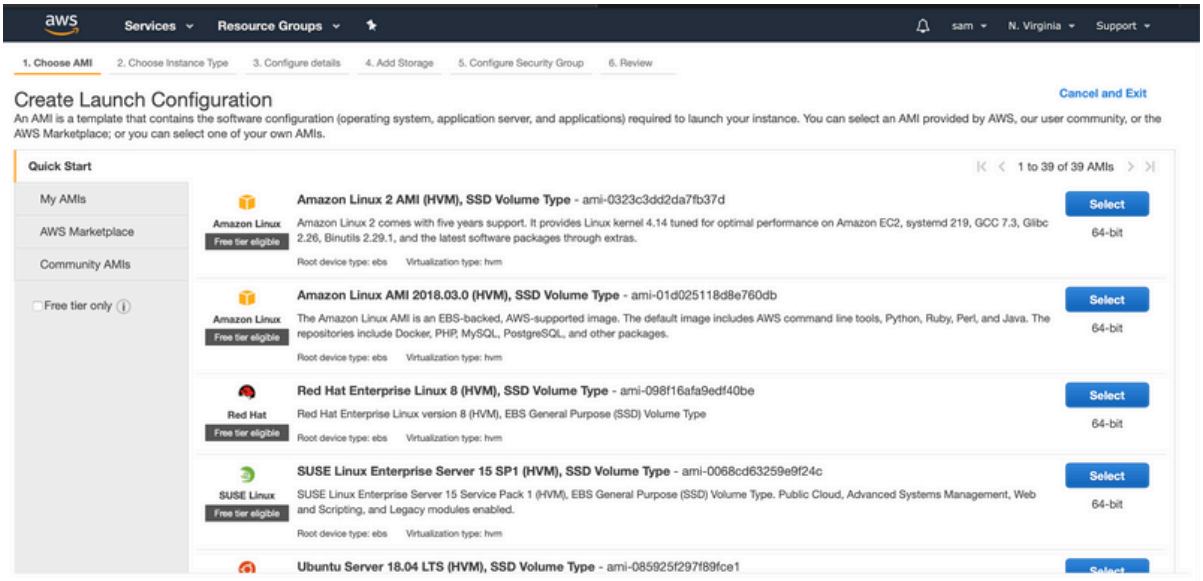
click on **Create Auto Scaling group** and then click on **Get started**

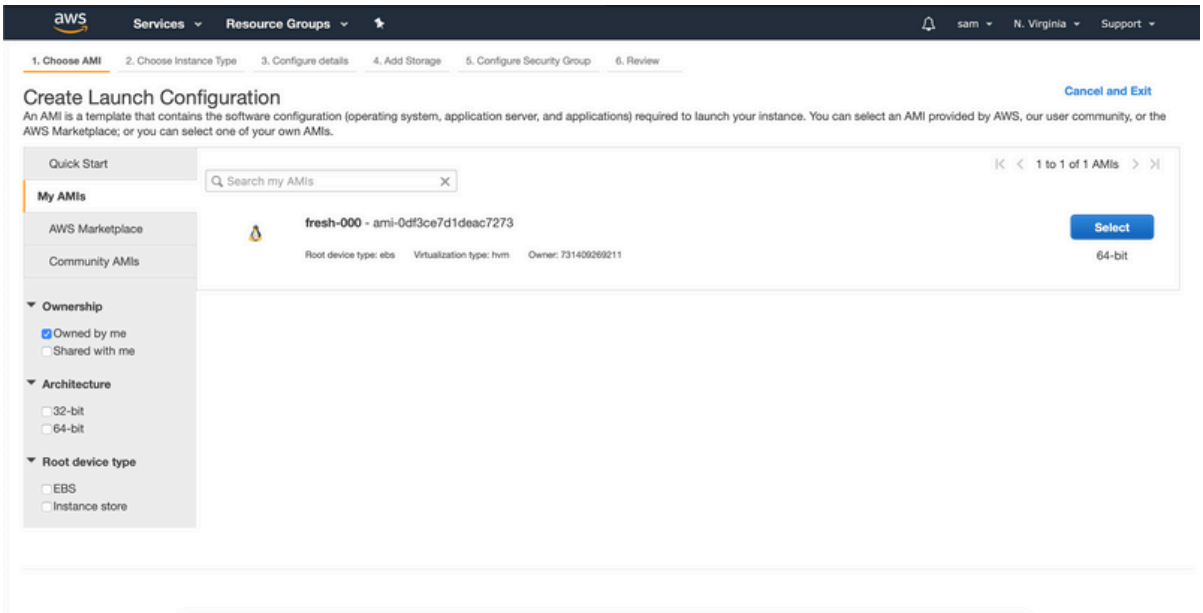



## Choose AMI

- 1.2 [ ] Click on **My AMIs** and choose the fresh-000 and click **select**



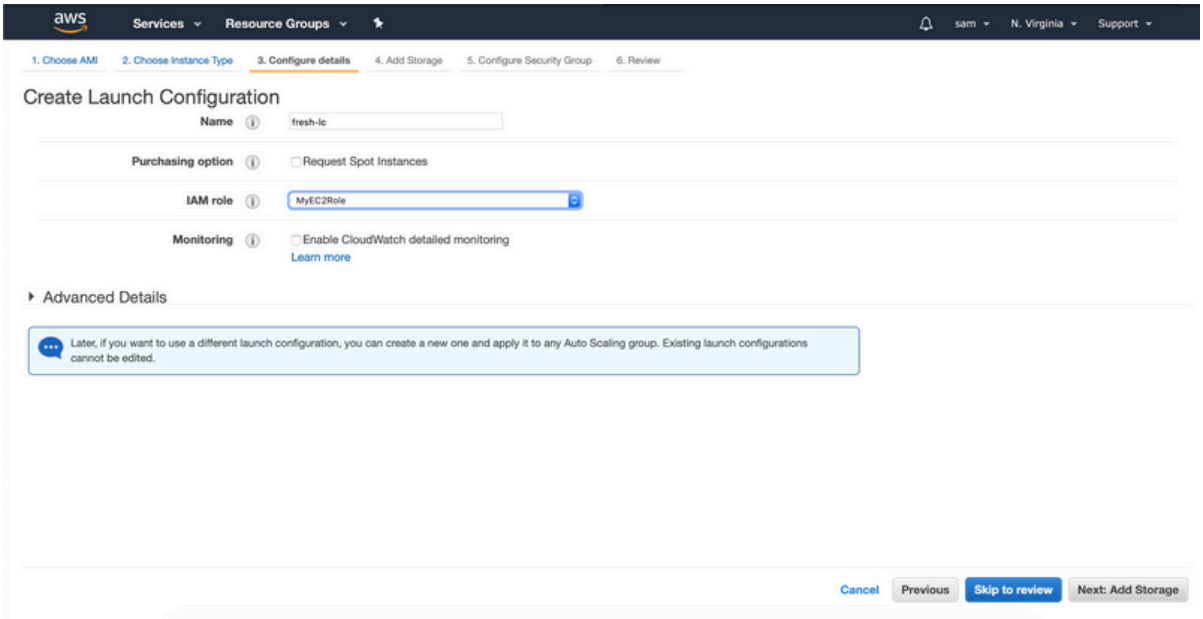




## Choose Instance Type

- 1.3 [ ] We will stick with t2.micro and click **Next: Configure details**

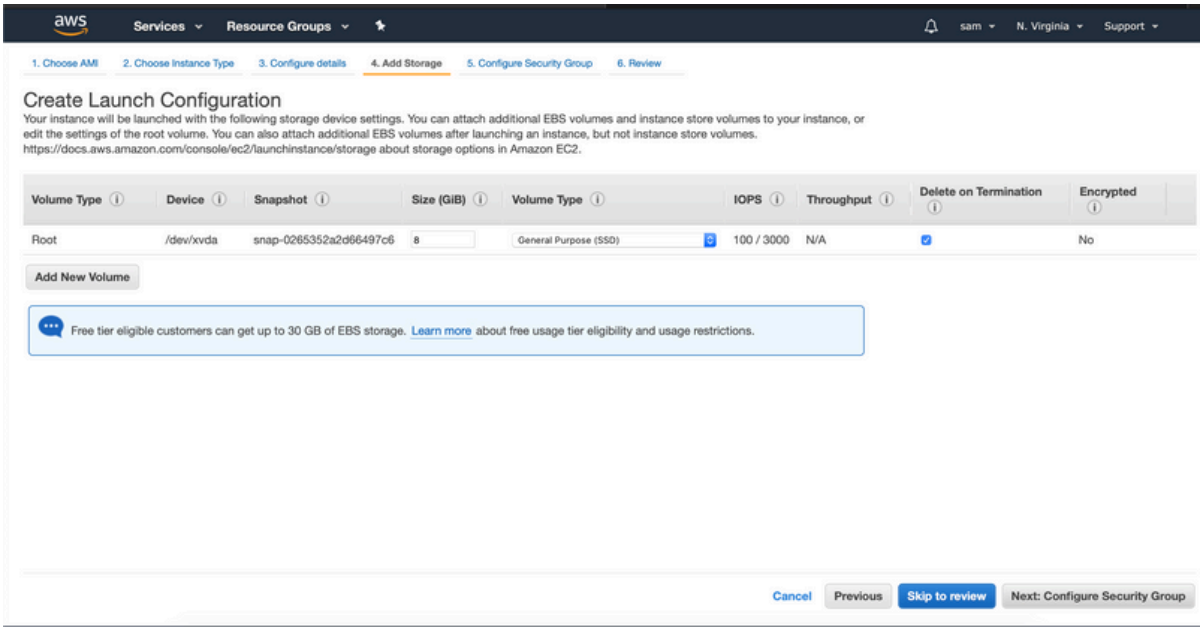
## Configure details

- 1.4 [ ] We will name the Launch Configuration fresh-lc, click on the drop down box for IAM role and select **MyEC2Role** and click **Next: Add Storage**



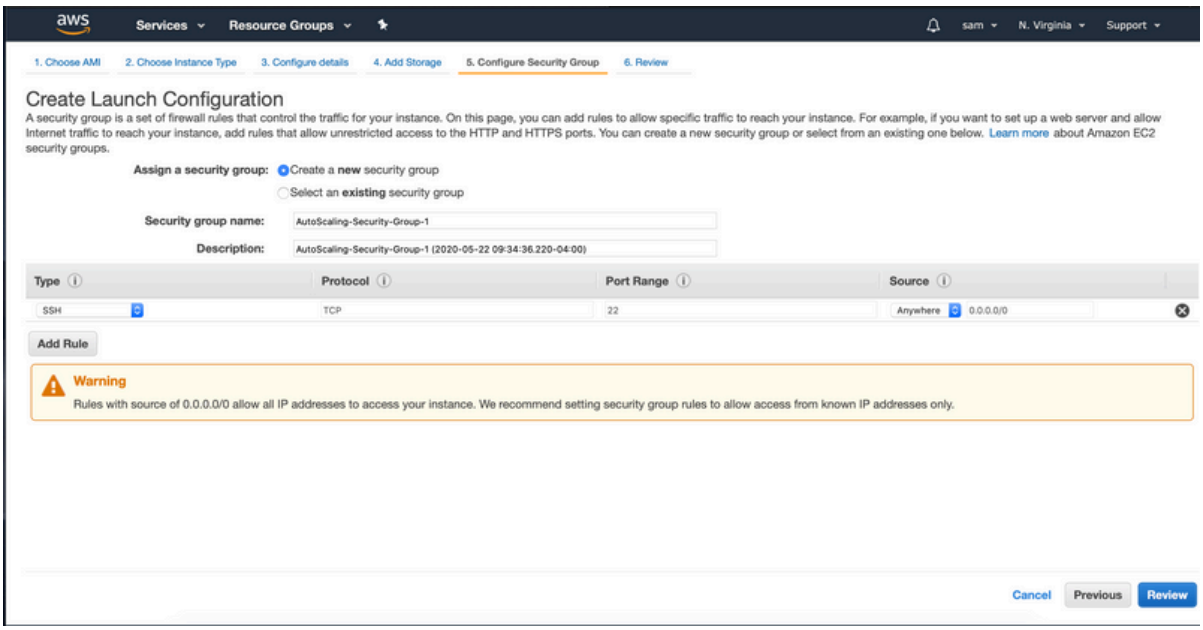

## Add storage

- 1.5 [ ] The defaults look good so click **Next: Configure Security Group**



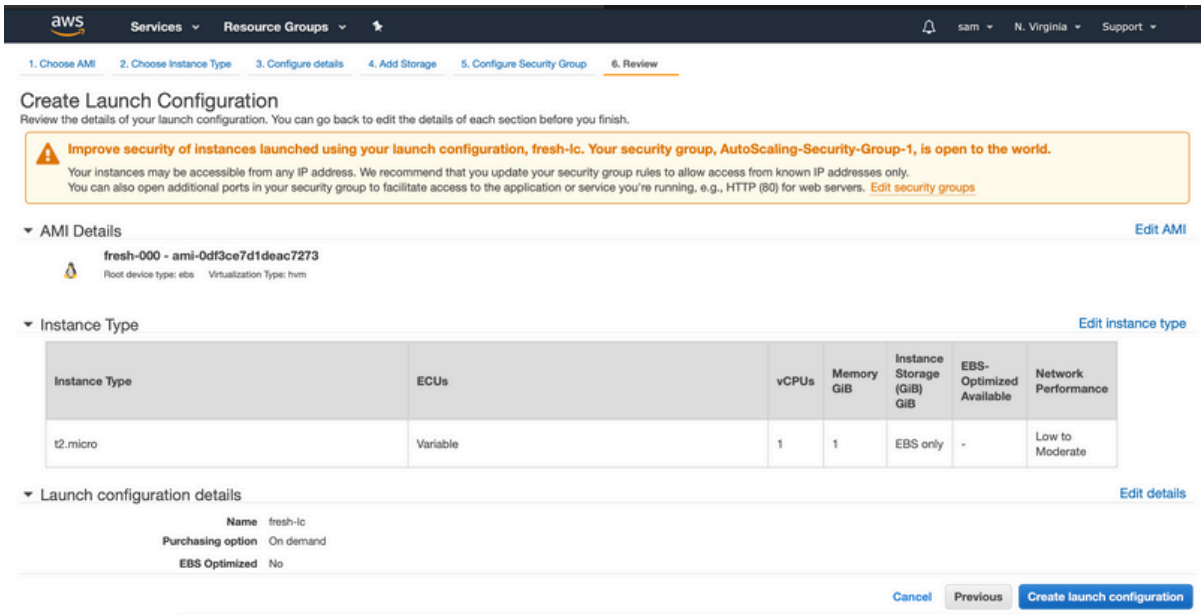

## Configure Security Group

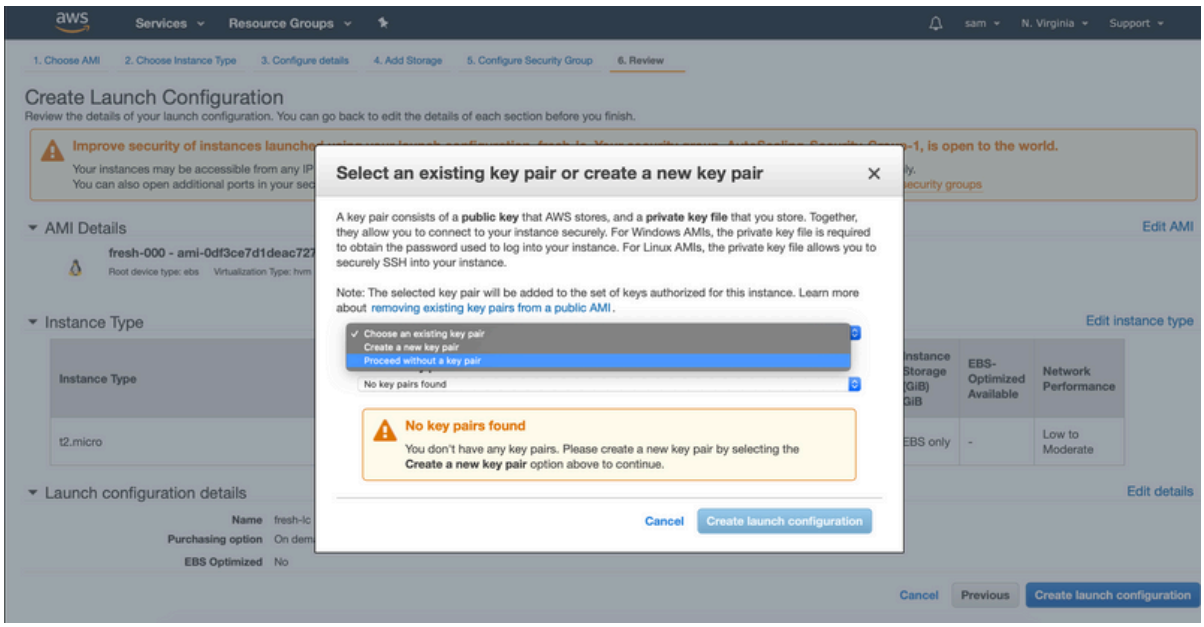
- 1.6 [ ] The security groups look good so click **Review**

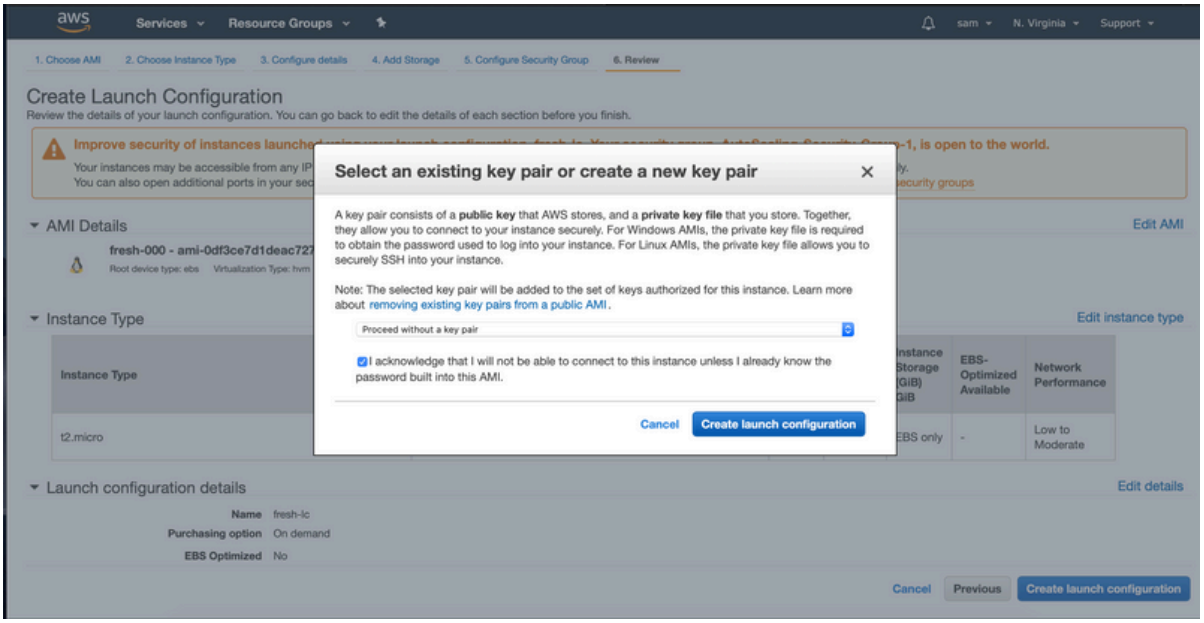


## Review

- 1.7 [ ] Click **Create launch configuration** and in the drop down box click on **Proceed without a key pair** and check mark **I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI** and click **Create launch configuration**

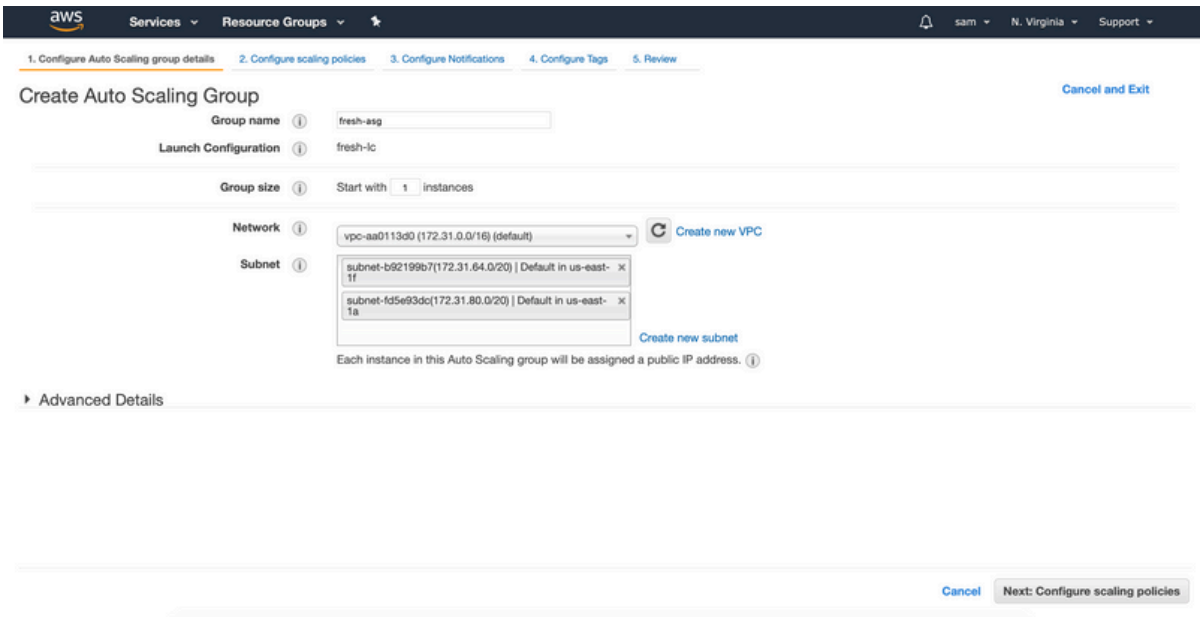


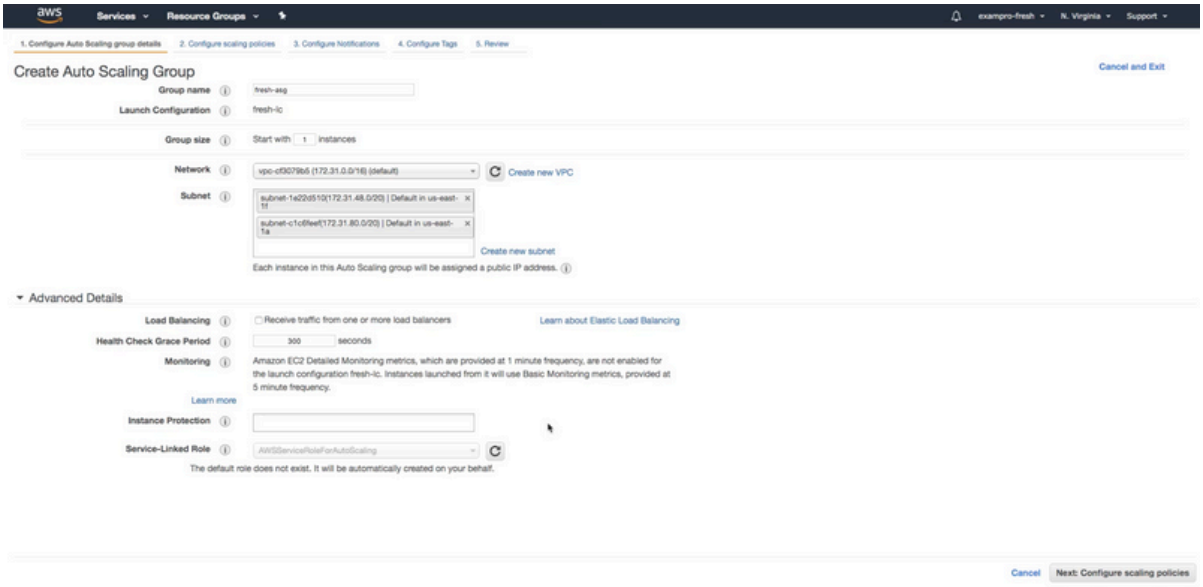






## Configure Auto Scaling group details

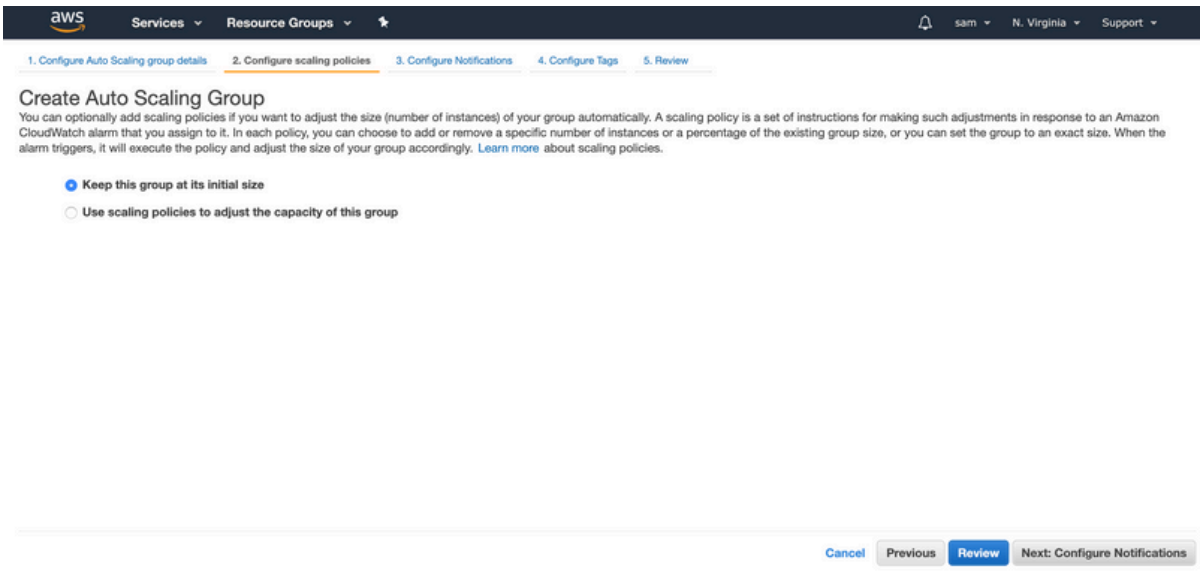
- 1.8 [ ] Name Auto Scaling Group **fresh-asg** and group size leave at 1, select the Network as **vac-cf3079b5 (default)** and select the drop down box for subnet and choose two subnets. Click on **Advanced Details** and it all looks good so click **Next: Configure scaling policies**





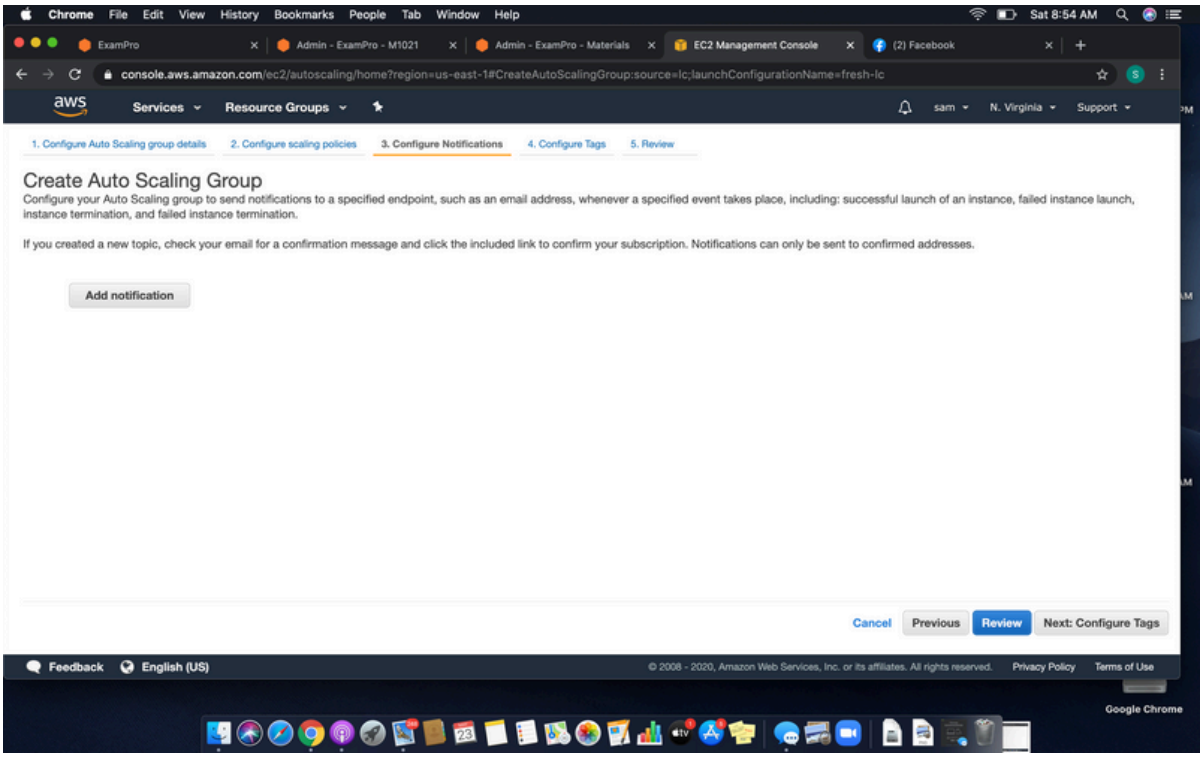

## Configure scaling policies

- 1.9 [ ] Select **Keep this group at its initial size** and click **Next:Configure Notifications**



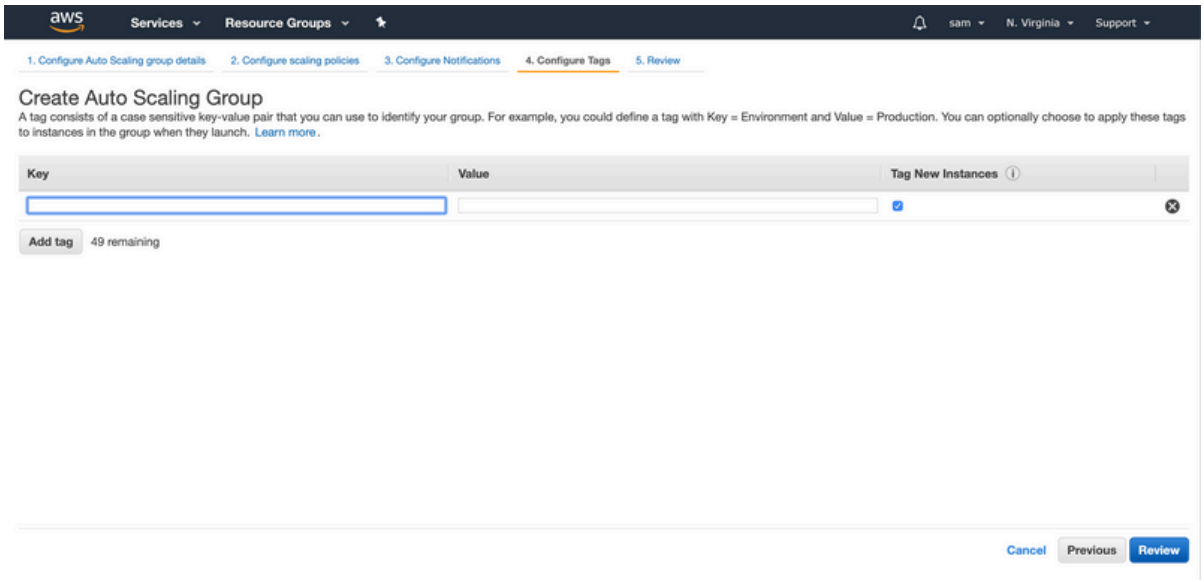
## Configure Notifications

- 1.10 [ ] Select **Next: Configure Tags**



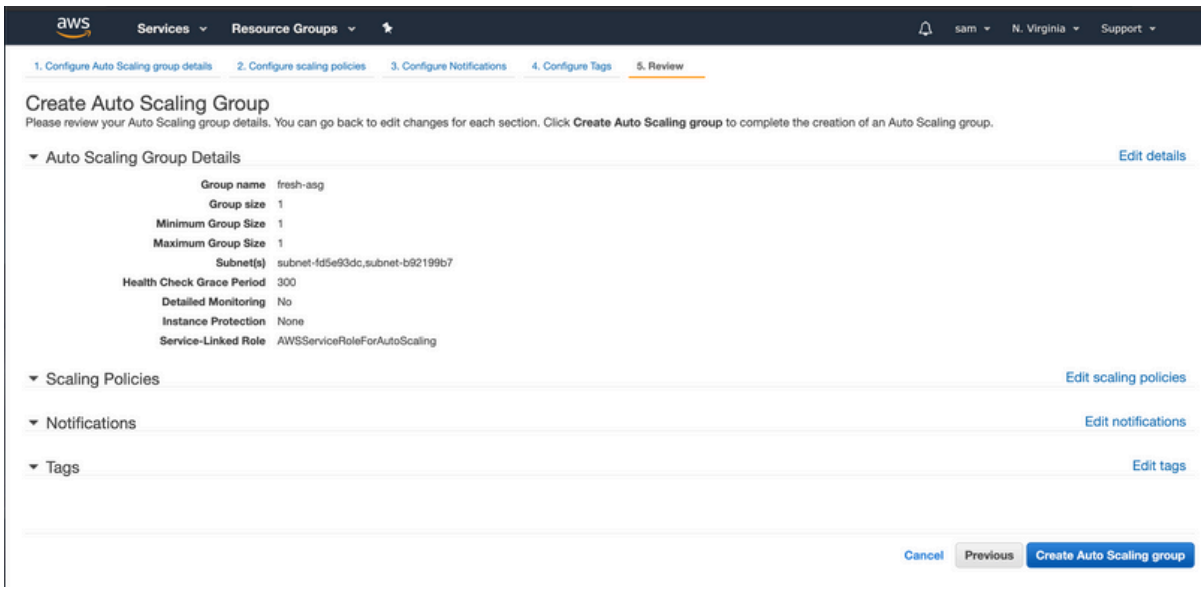

## Configure Tags

- 1.11 [ ] Select **Review**

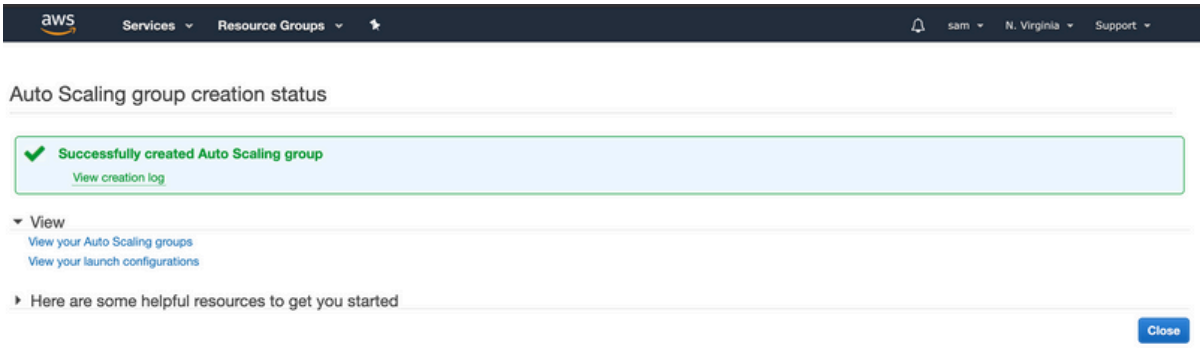



## Review

- 1.12 [ ] Select **Create Auto Scaling group**

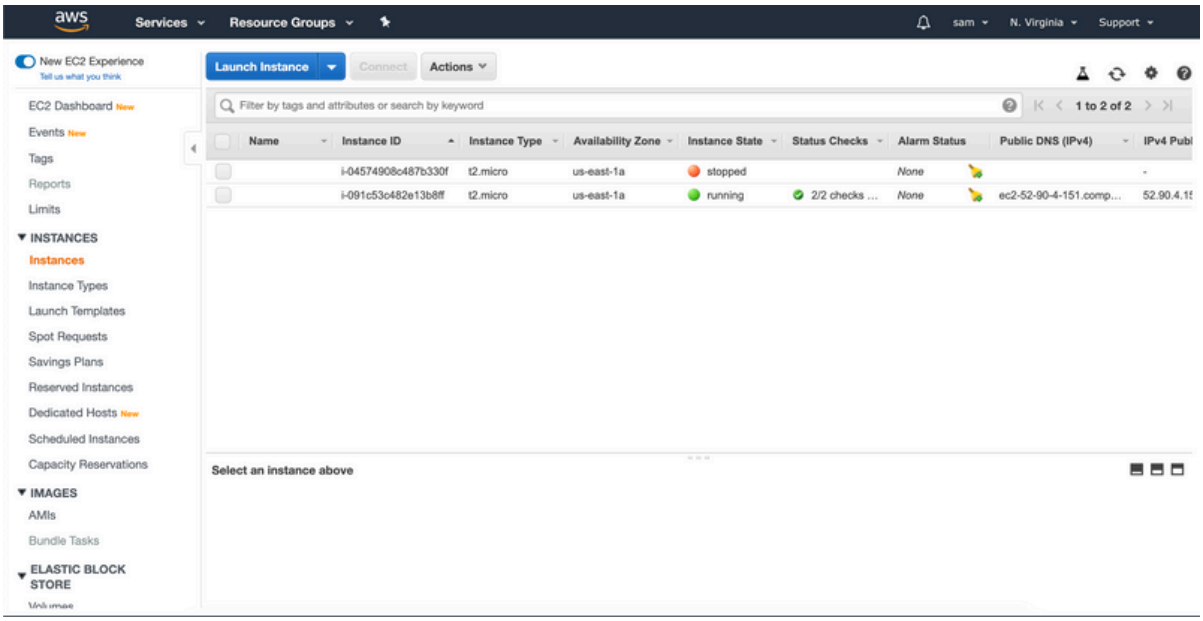



- 1.13 [ ] Auto Scaling group has been created so click **Close**




## Instances

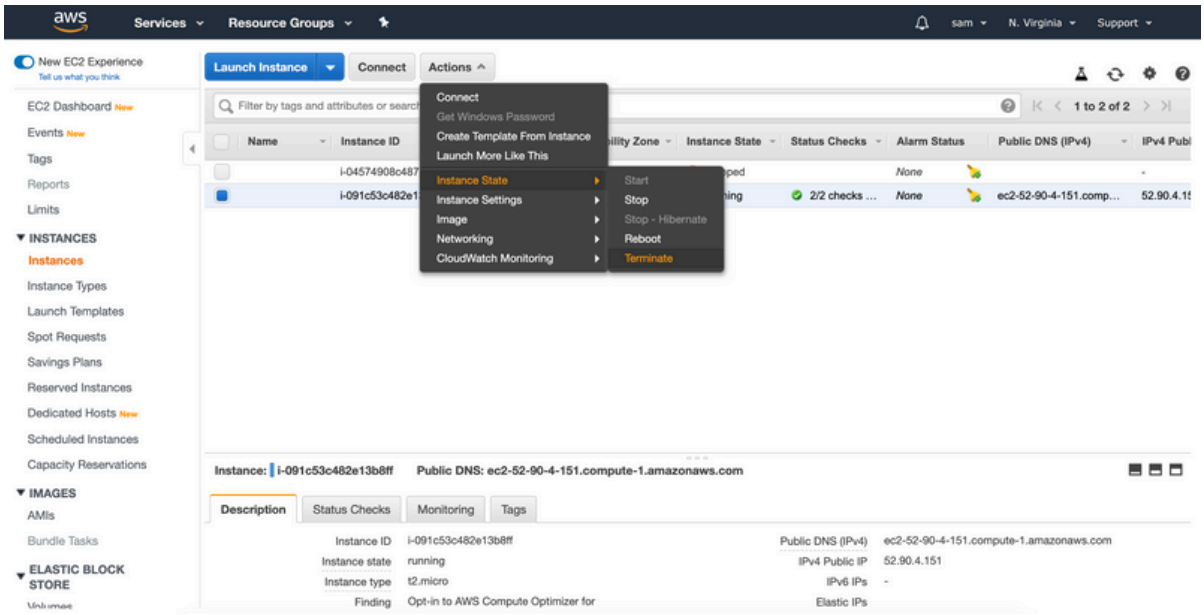
- 1.14 [ ] Right Click on **Instances** on the left hand side and you will start to see instances, you can refresh your instance tab and see more instances spit up

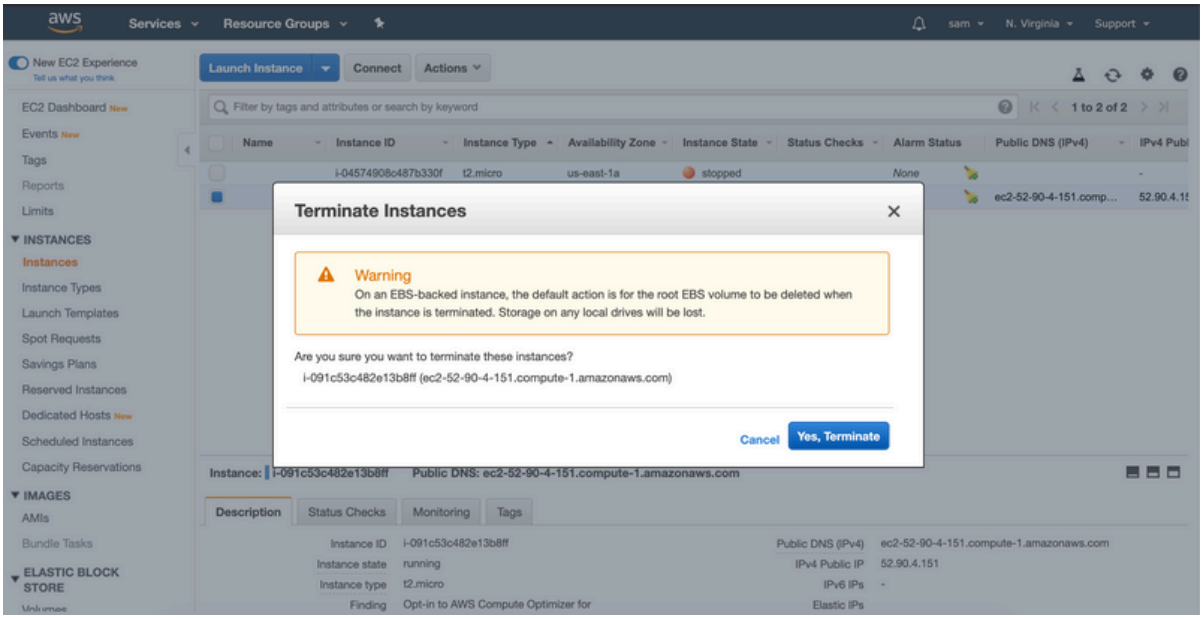


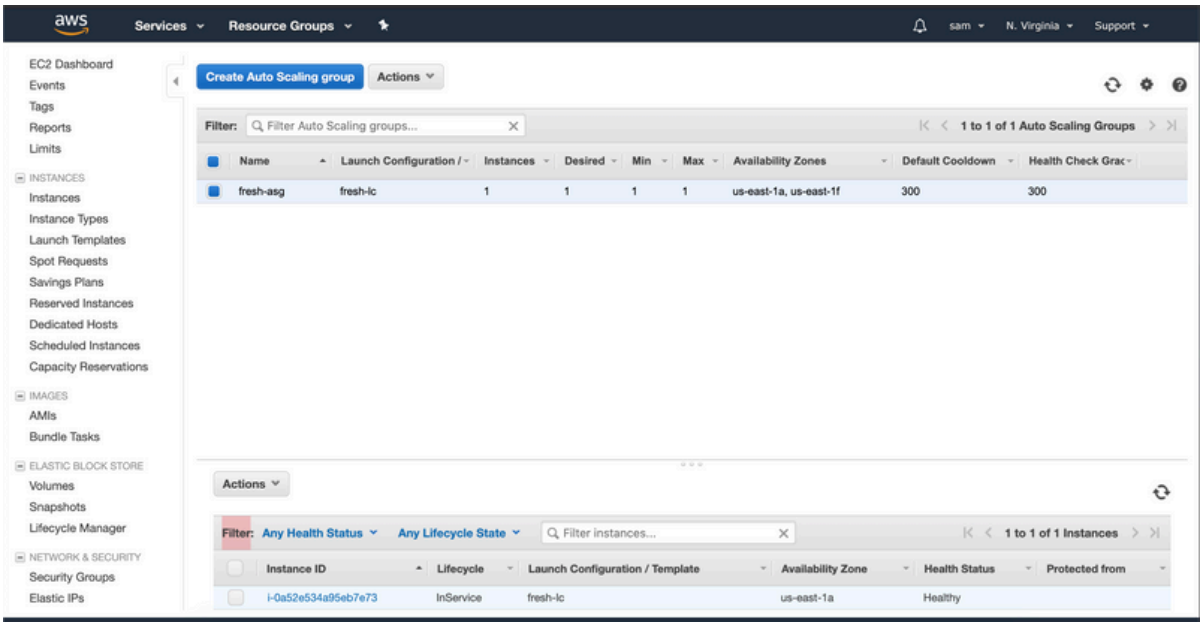

Auto Scaling can ensure that there are always a minimum of servers running but

if we were to terminate the instance it's going to detect that this one is no longer healthy. After a while it will determine that it's unhealthy and the health status will change from healthy to unhealthy. The way this Auto Scaling group is going to respond is to launch a new instance. You can keep hitting the refresh button until you see another instance replace this unhealthy one. You may have to wait a few minutes to see the new replacement instance.

- 1.15 [ ] Check mark the **running server** select **Actions** select **Instance State** and **Terminate** and then select **Yes Terminate**

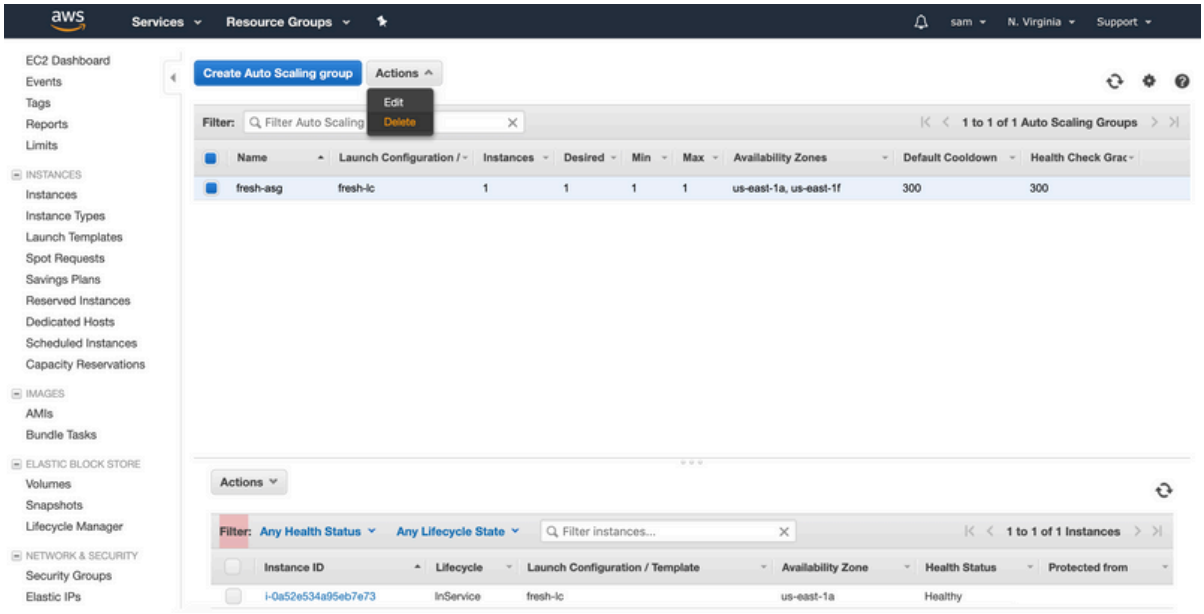


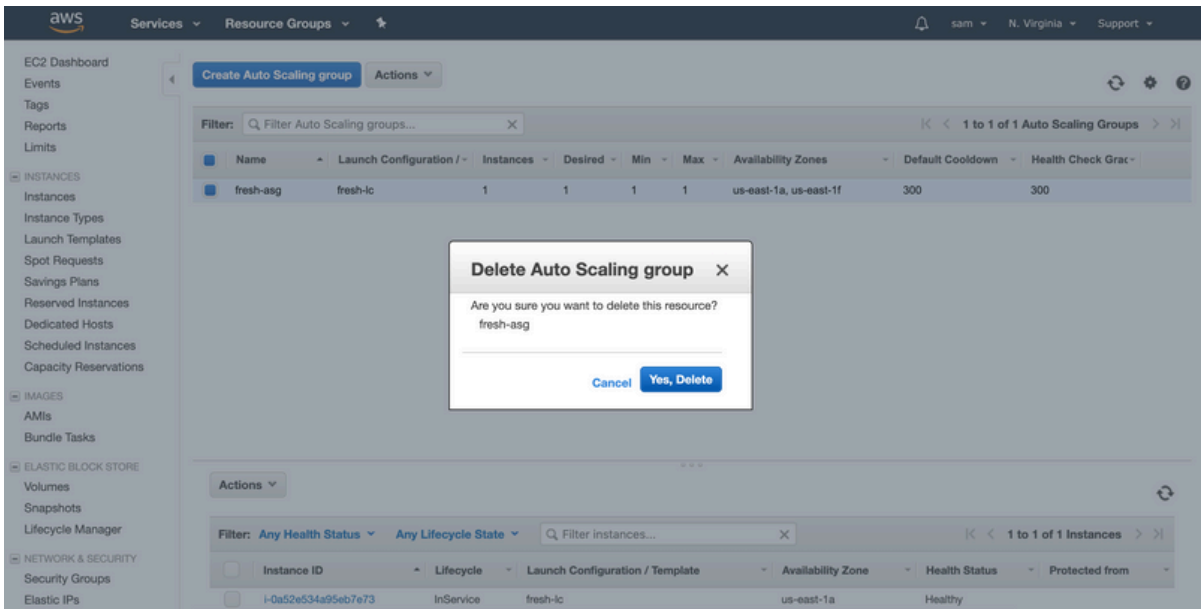


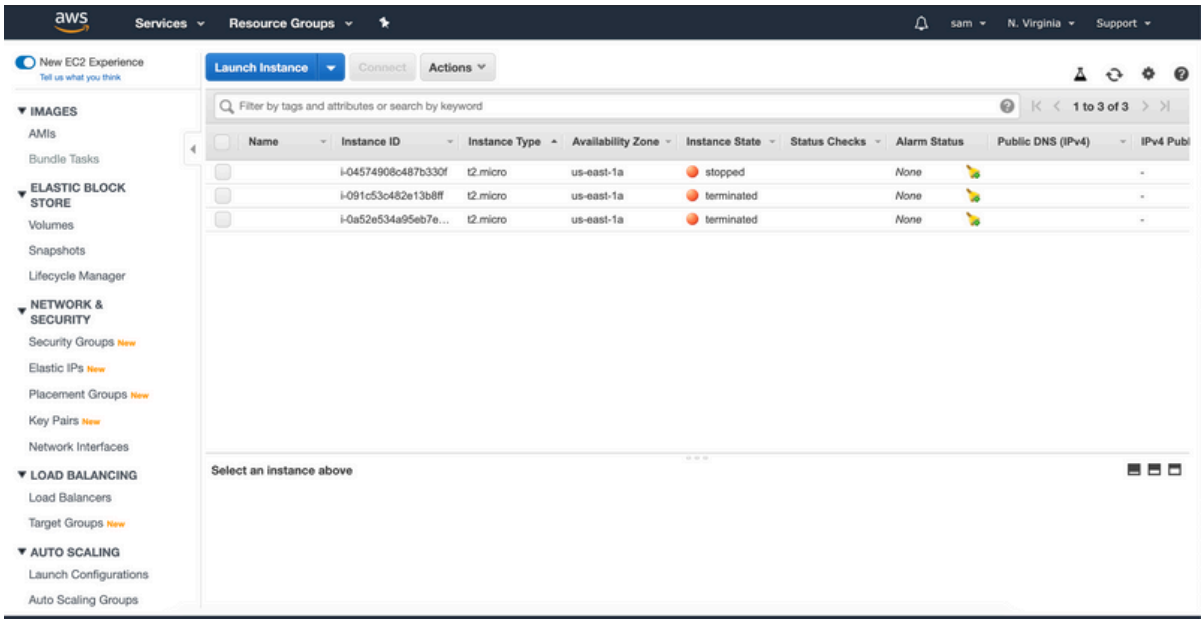

##Delete Auto Scaling Group

- 1.16 [ ] Select **Actions** and **Delete** and click **Yes, Delete**




- 1.17 [ ] You can click on**Instance** on your left hand side to make sure it was deleted, you may need to refresh the page



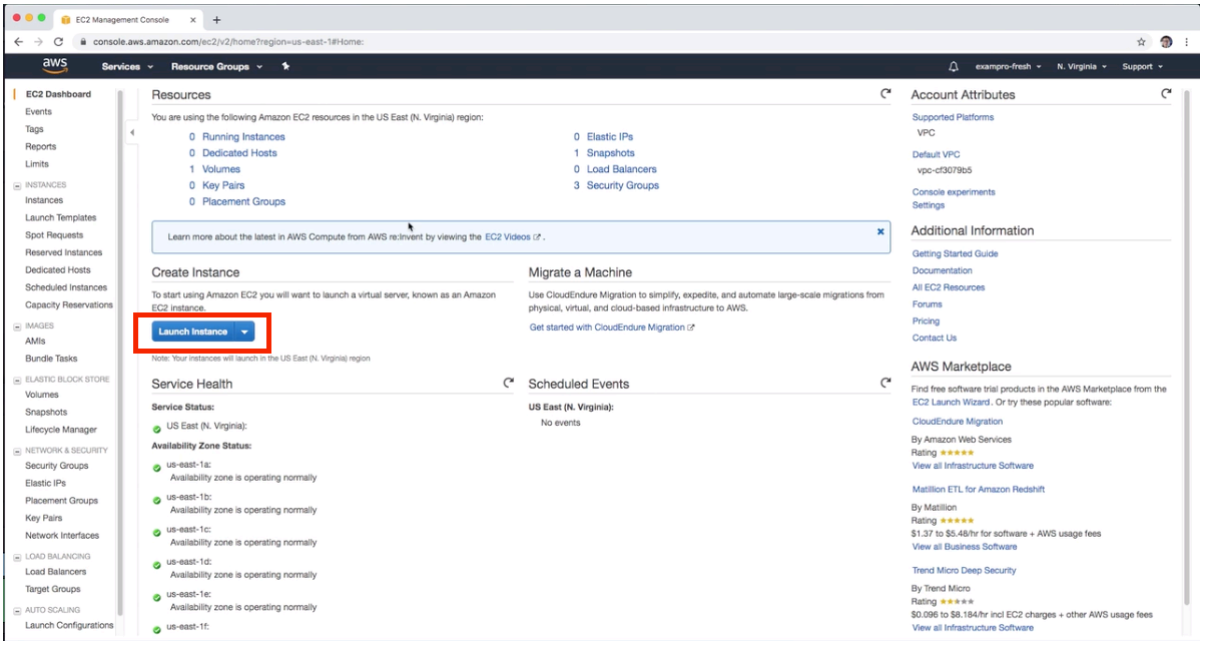

Now we will move onto Elastic Load Balancer

# Elastic Load Balancer

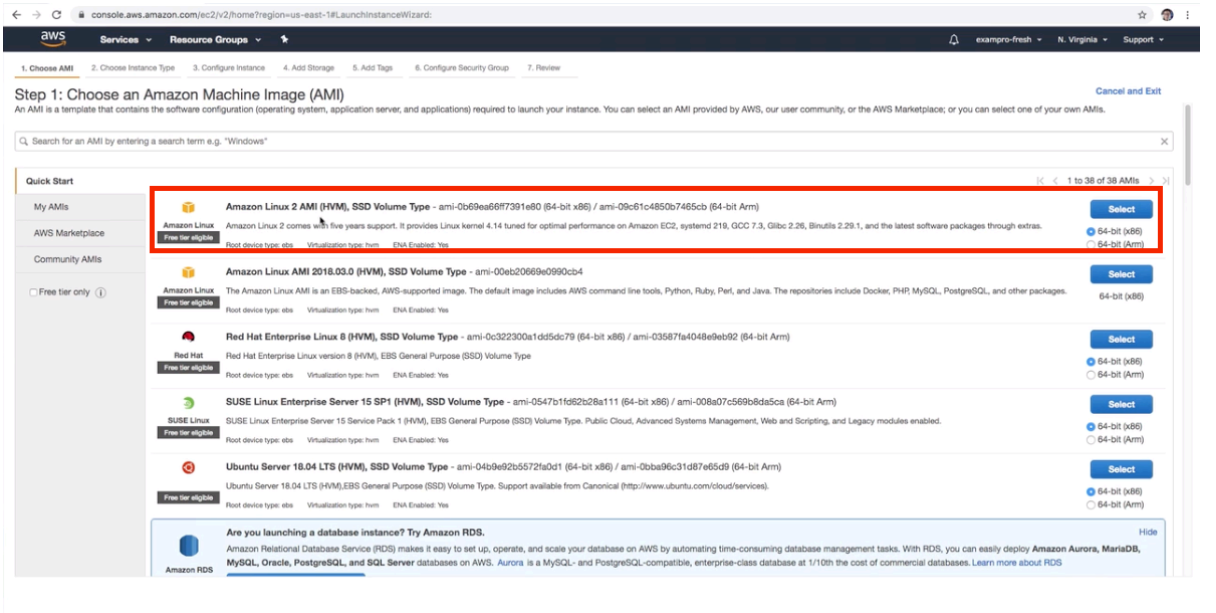
We are going to learn how to set up Elastic Load Balancer in front of your EC2 instances.

## Launching an instance

- 1.1 [ ] In your EC2 dashboard under **Create Instance** click "Launch Instance"



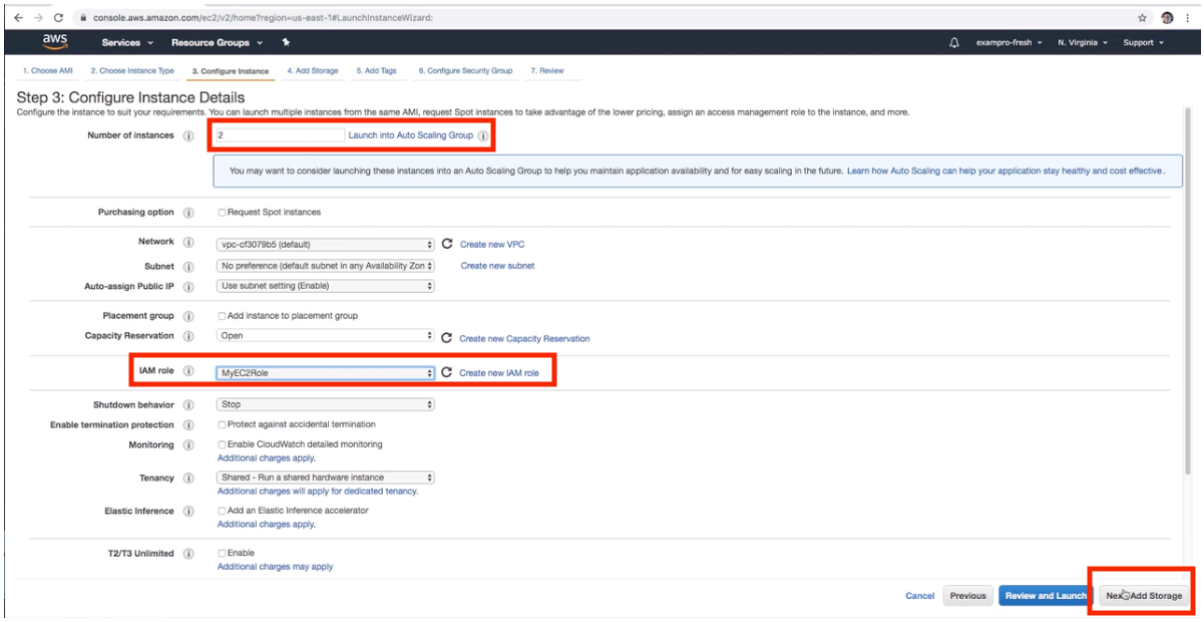

- 1.2 [ ] Select "Amazon Linux 2"




- 1.3 [ ] On the Step 2 page keep the t2.micro option and hit the "Next: Configure Instance Details" button
- 1.4 [ ] In Configure Instance Details form add "2" to the number of instances

- 1.5 [ ] Add an IAM role "myEC2Role"
- 1.6 [ ] Click the "Next: Add Storage" button

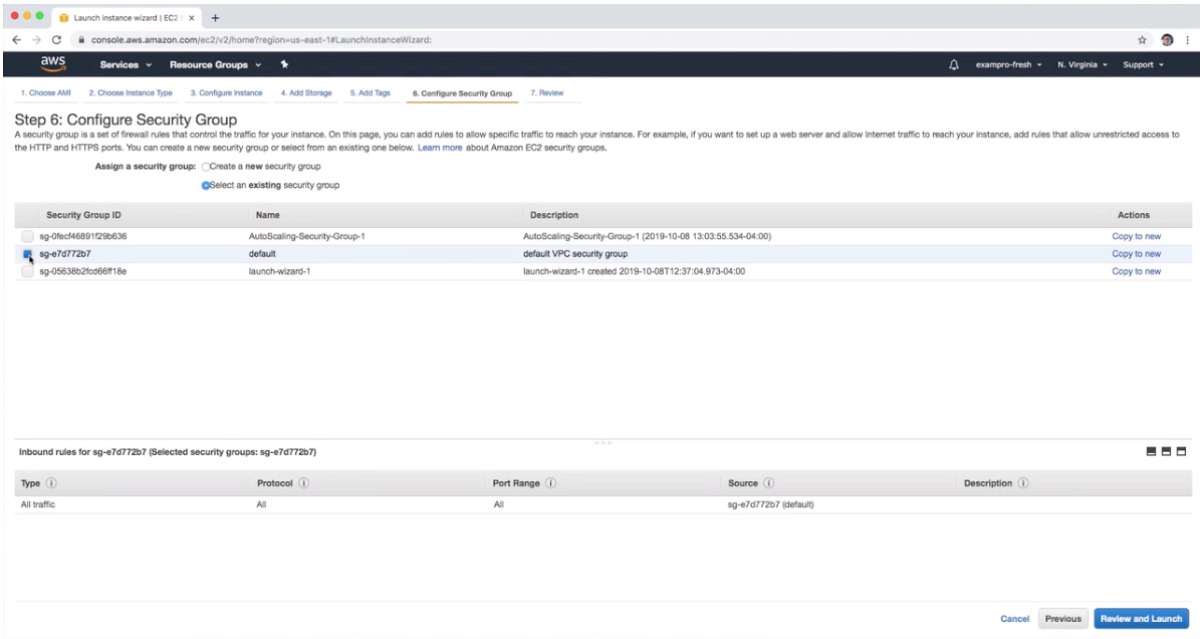
You will not need to change any thing in Storage



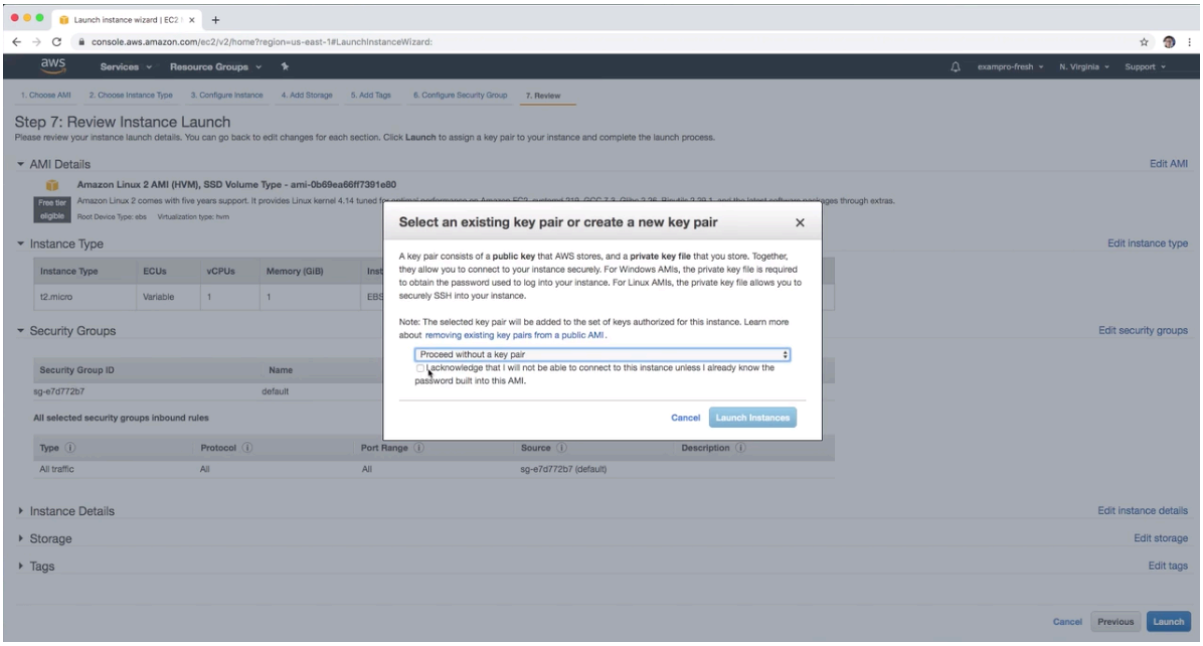

- 1.7 [ ] Click the "Next: Add Tags" button

Again you will not need to change anything here.

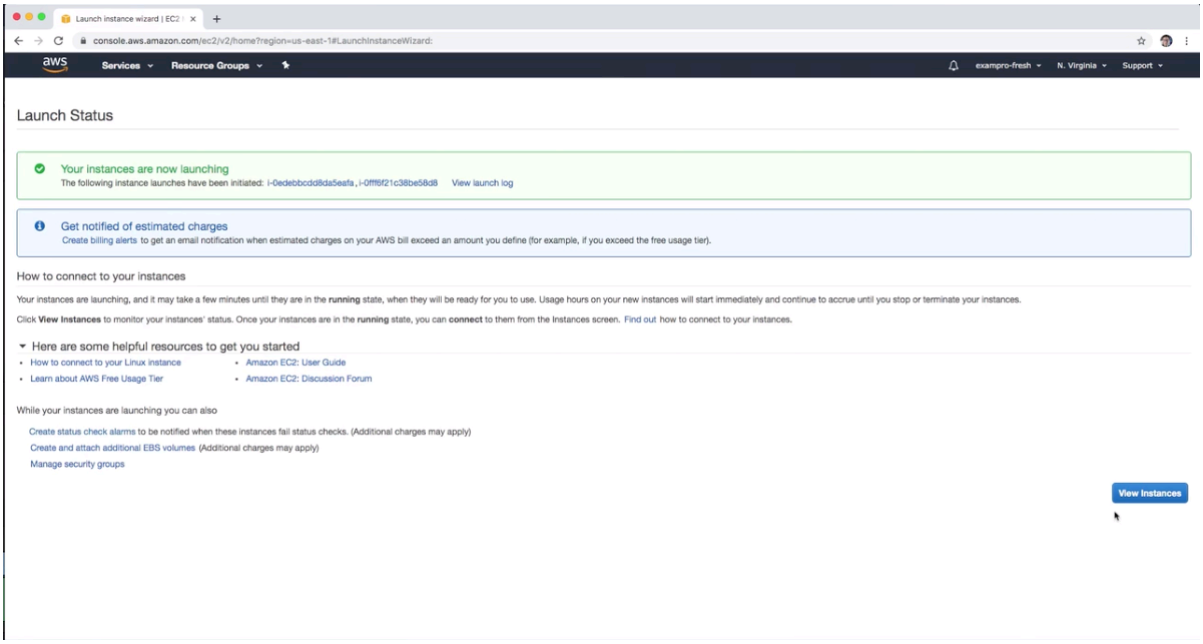
- 1.8 [ ] Select an existing security group
- 1.9 [ ] Click the "Review and Launch" button




- 1.10 [ ] In the dropdown choose "Proceed without a key pair"

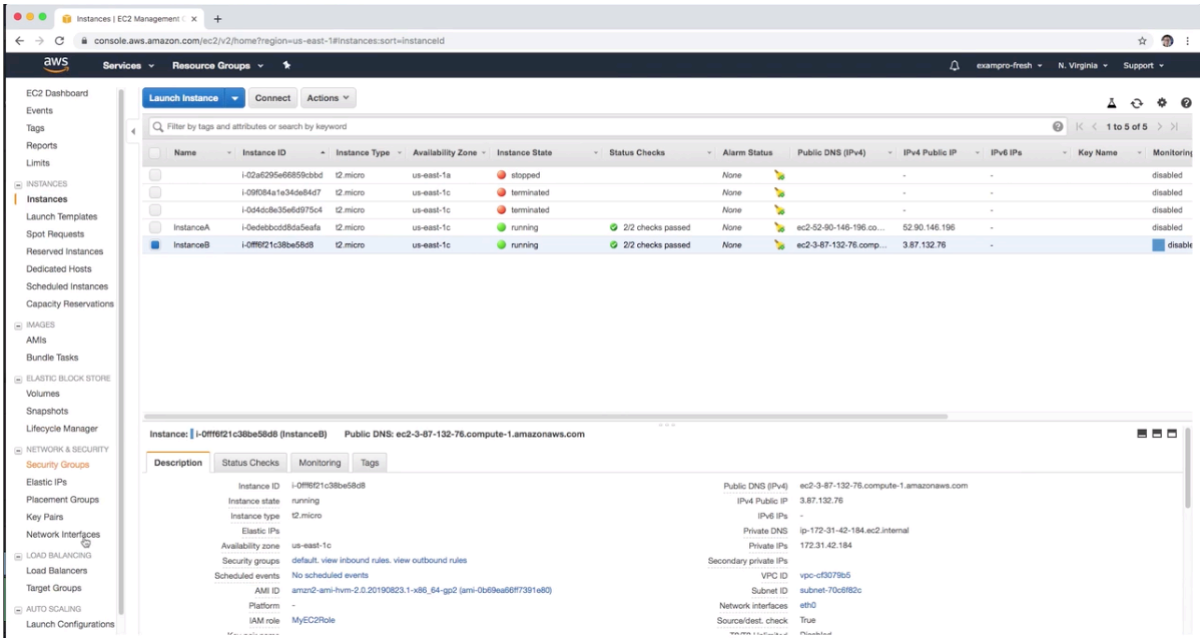



- 1.11 [ ] Click the "View Instance" button



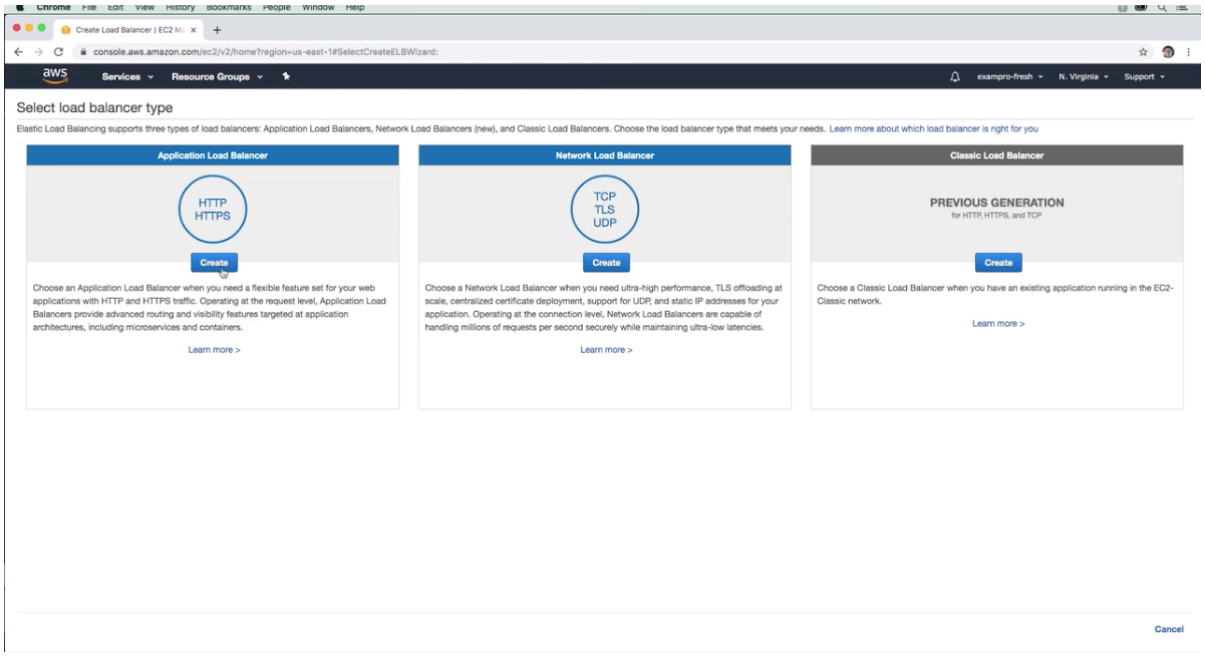

## Creating the ELB

- 2.1 [ ] Once the instances are running give them a new name.

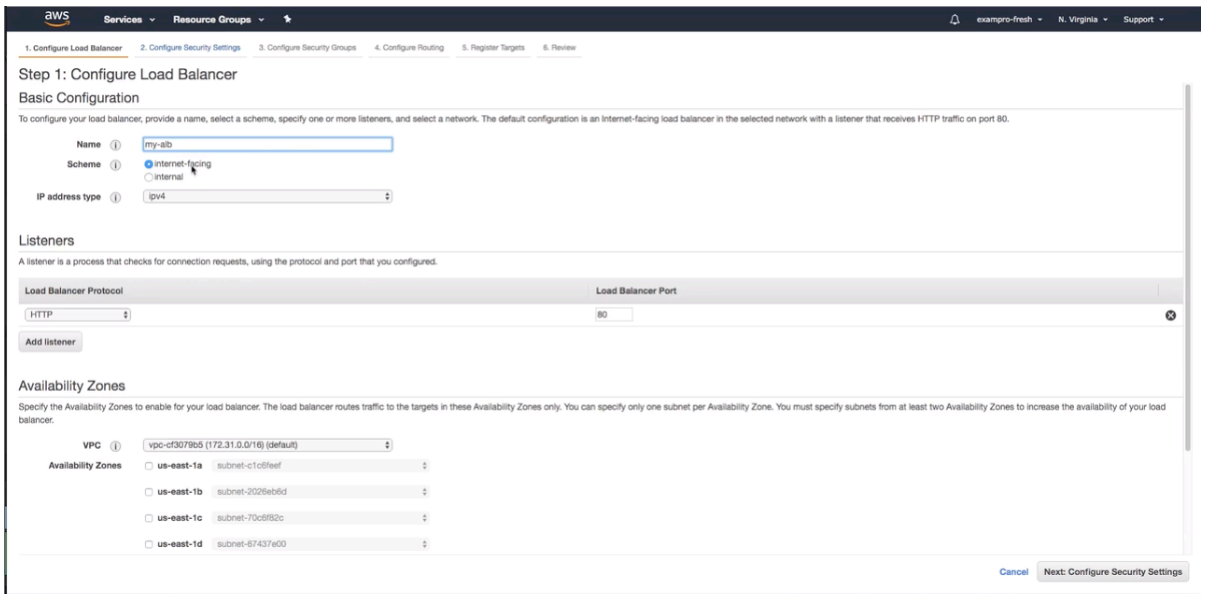


- 2.2 [ ]Under the EC2 Console click on the link to "Load Balancers"

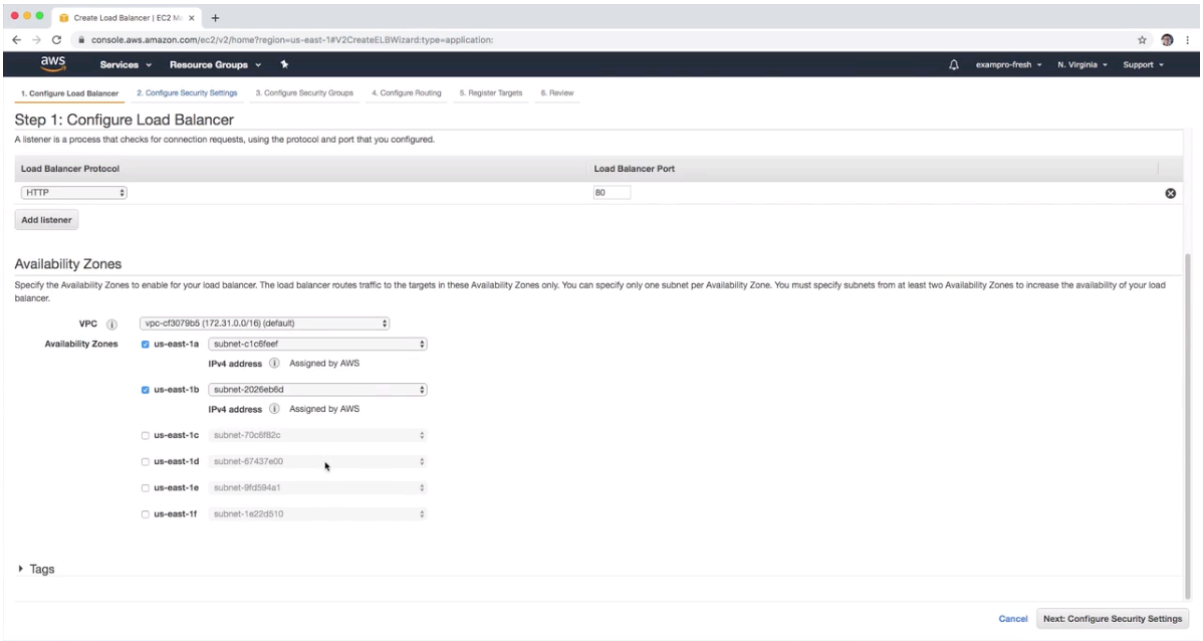
- 2.3 [ ] Click the "Create" button for Application Load Balancer



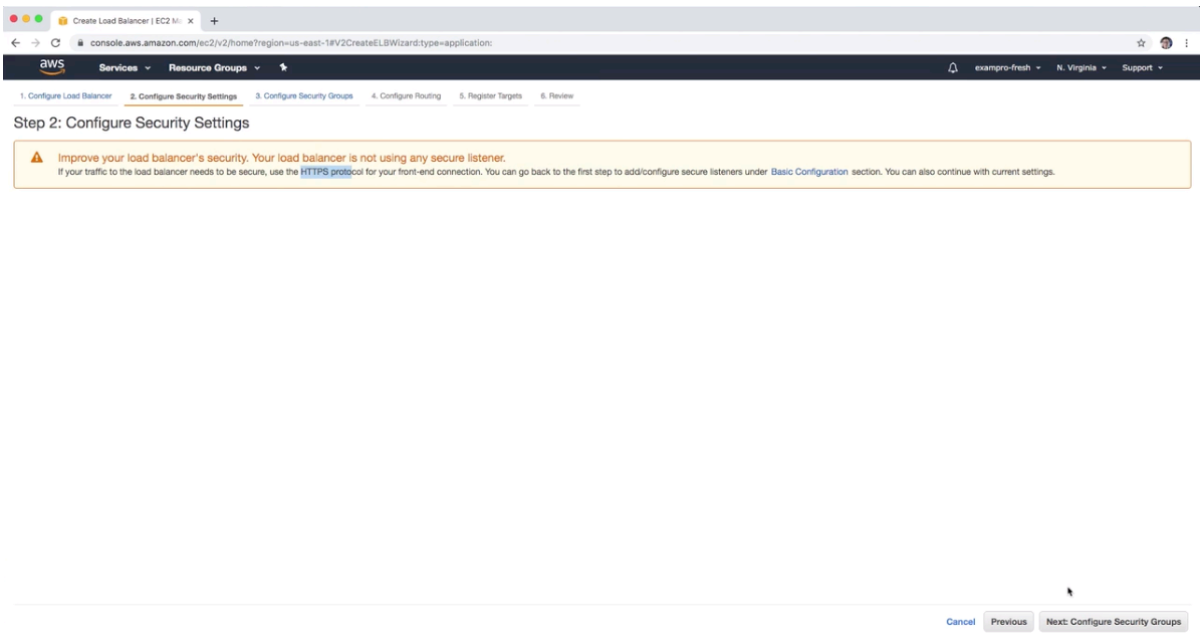

- 2.4 [ ] Give the load balancer a name and keep it internet-facing




- 2.5 [ ] Select at least two Availability Zones

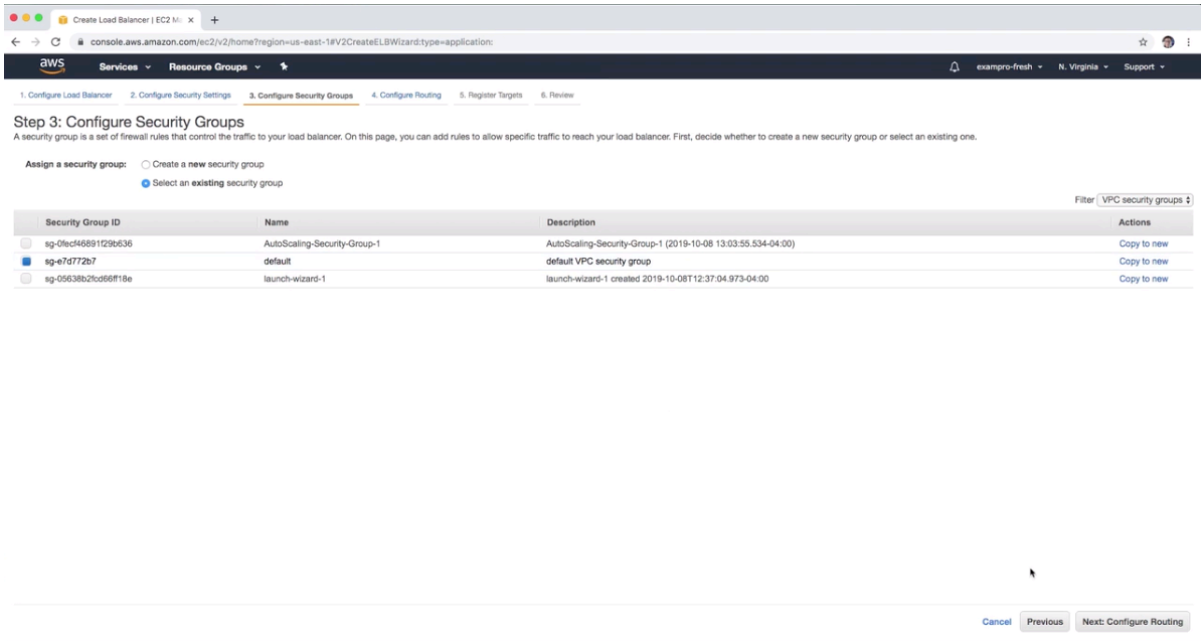



- 2.6 [ ] Click the "Next: Configure Security Settings" button
- 2.7 [ ] You do not need to configure any security settings. Click the "Next: Configure Security Groups" button

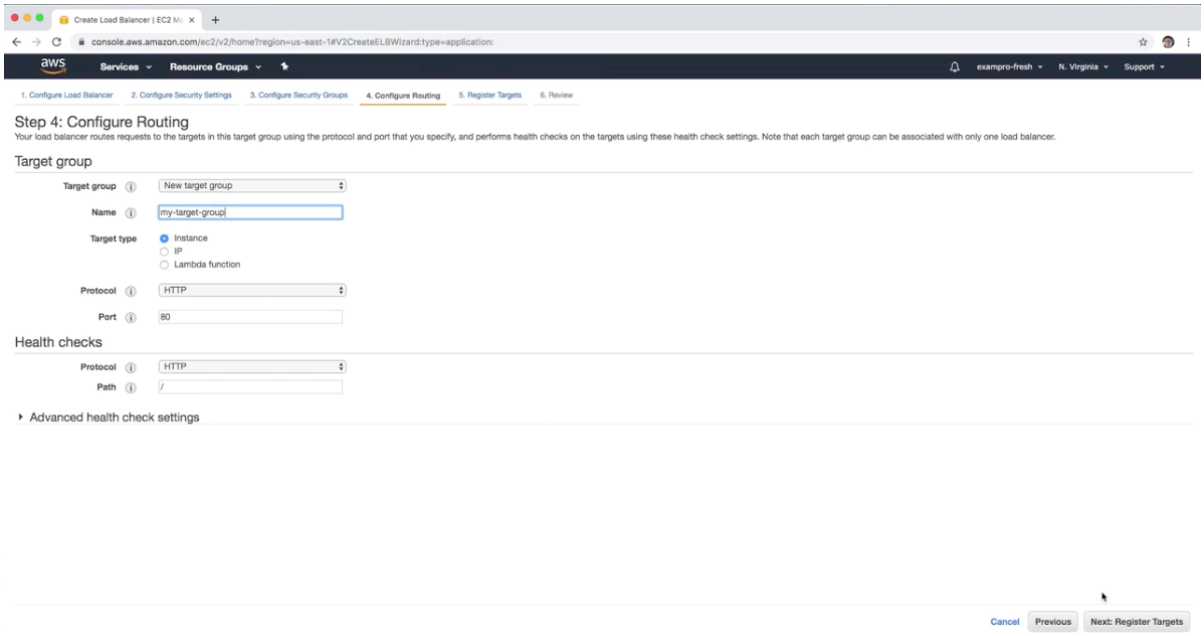



- 2.8 [ ] You can leave the default Security Group and click the "Next: Configure

Routing" button

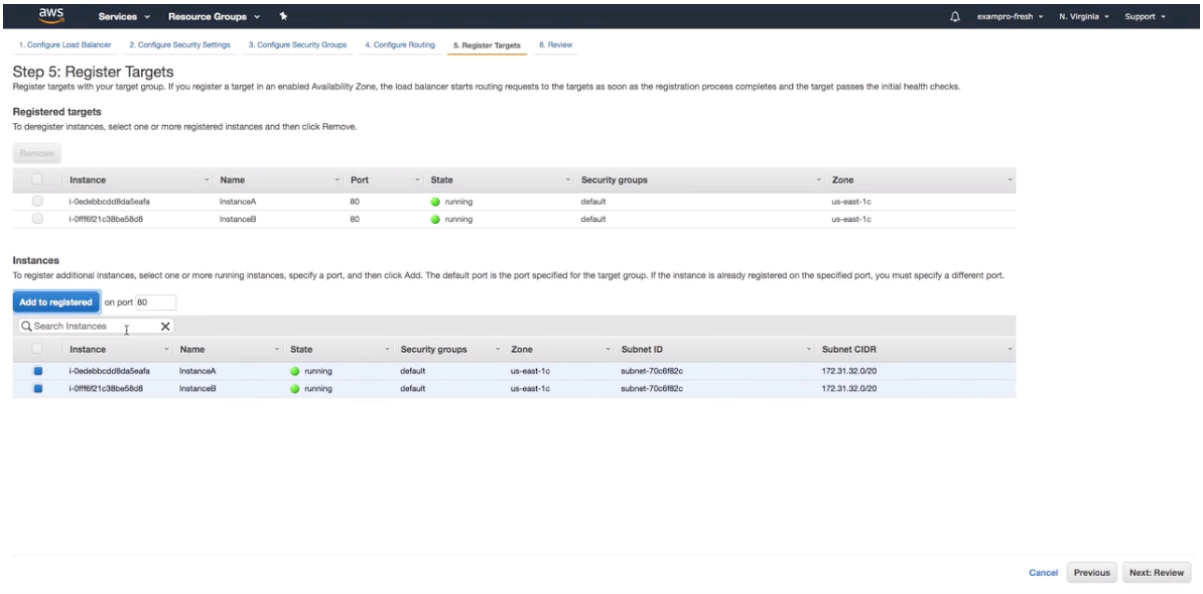



- 2.9 [ ] Create a new target group
- 2.10 [ ] Click the "Next: Register Targets" button

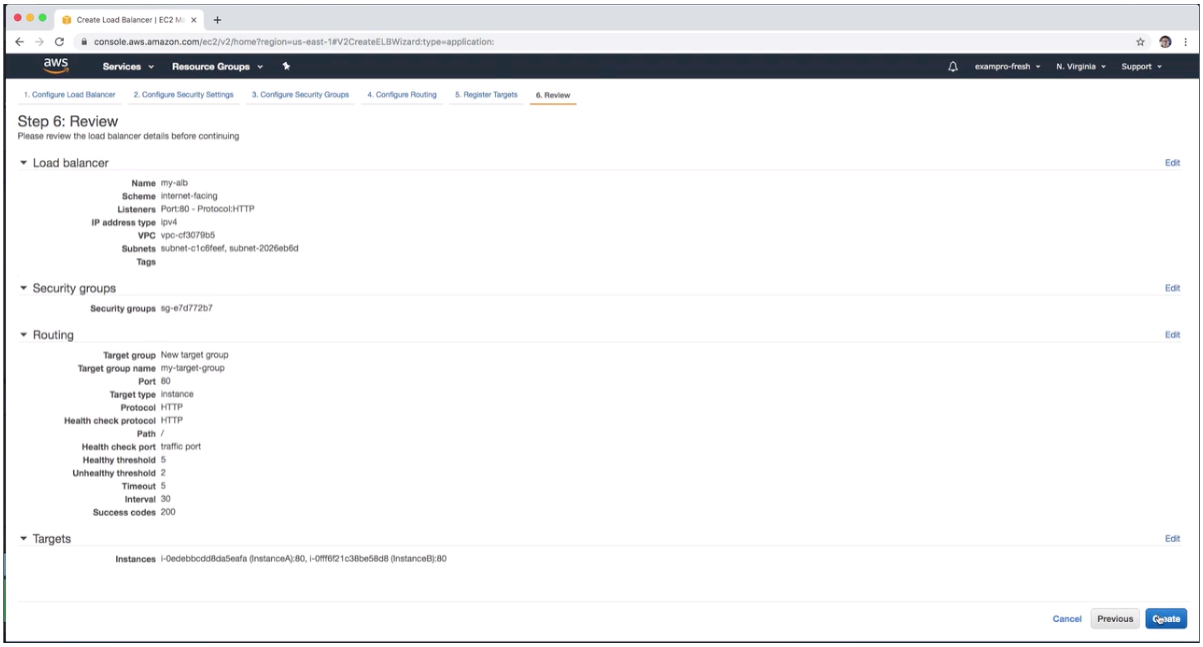


# Registering Targets

- 3.1 [ ] Select both instances and click "Add to registered"
- 3.2 [ ] Click the "Next: Review" button




- 3.3 [ ] Click the "Create" button



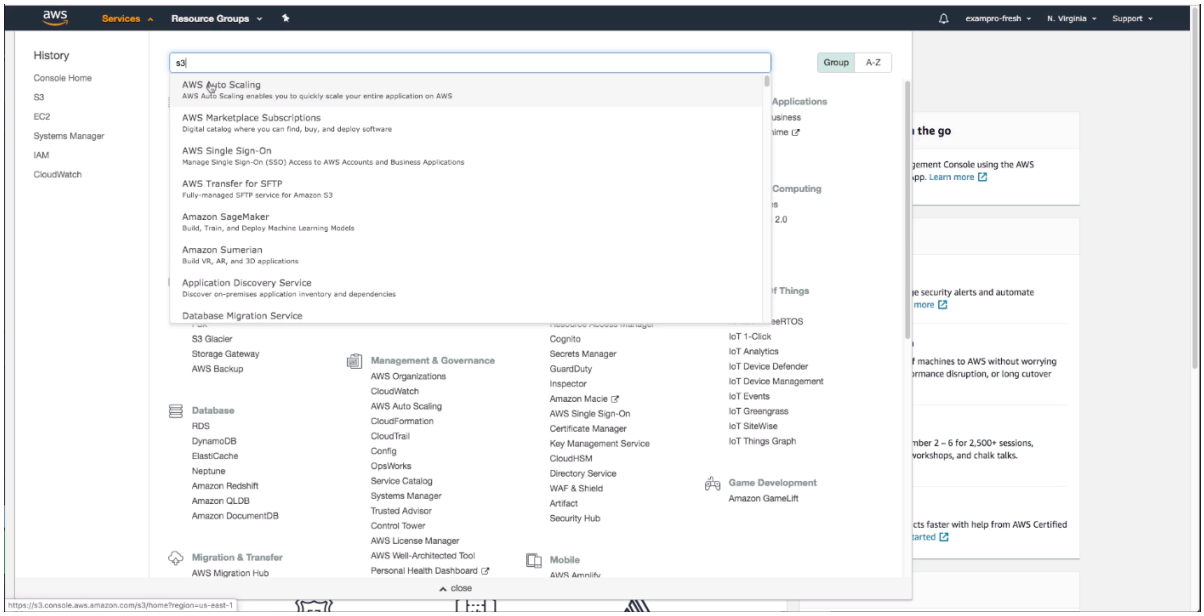
You should now have a Load Balancer created. When you are done you can Disable the ELB to avoid incurring any costs.

# Uploading a File to S3:

We are going to learn how to use S3 for file storage.

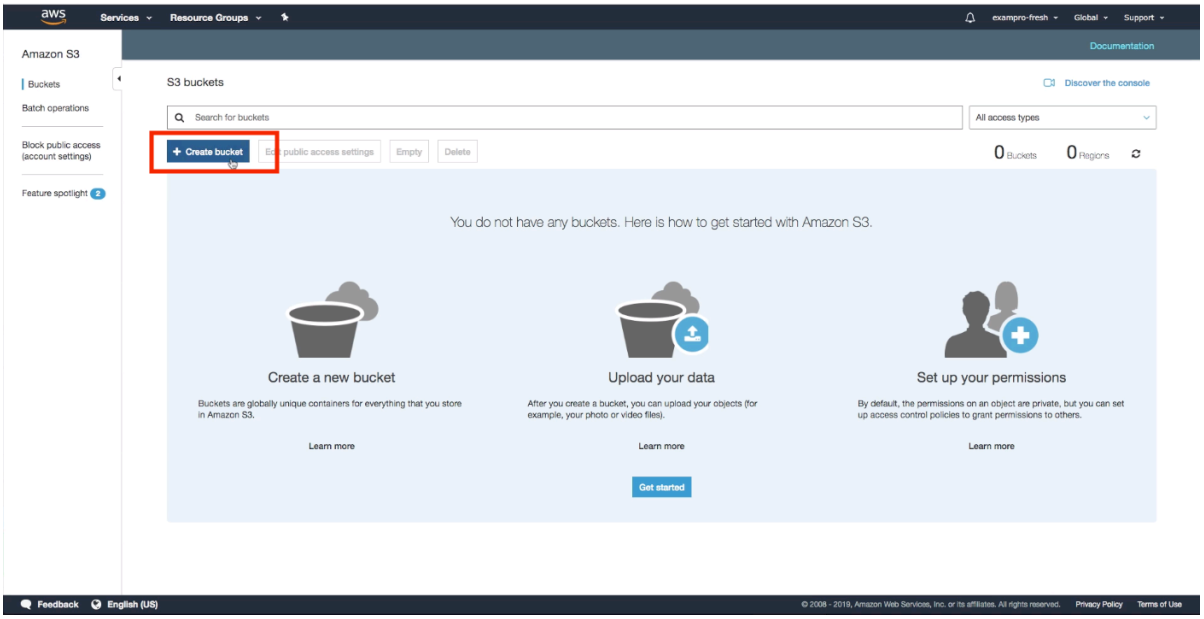
## Navigating to S3 Services

- 1.1 [ ] Inside the AWS Management Console use the search bar and type in S3.

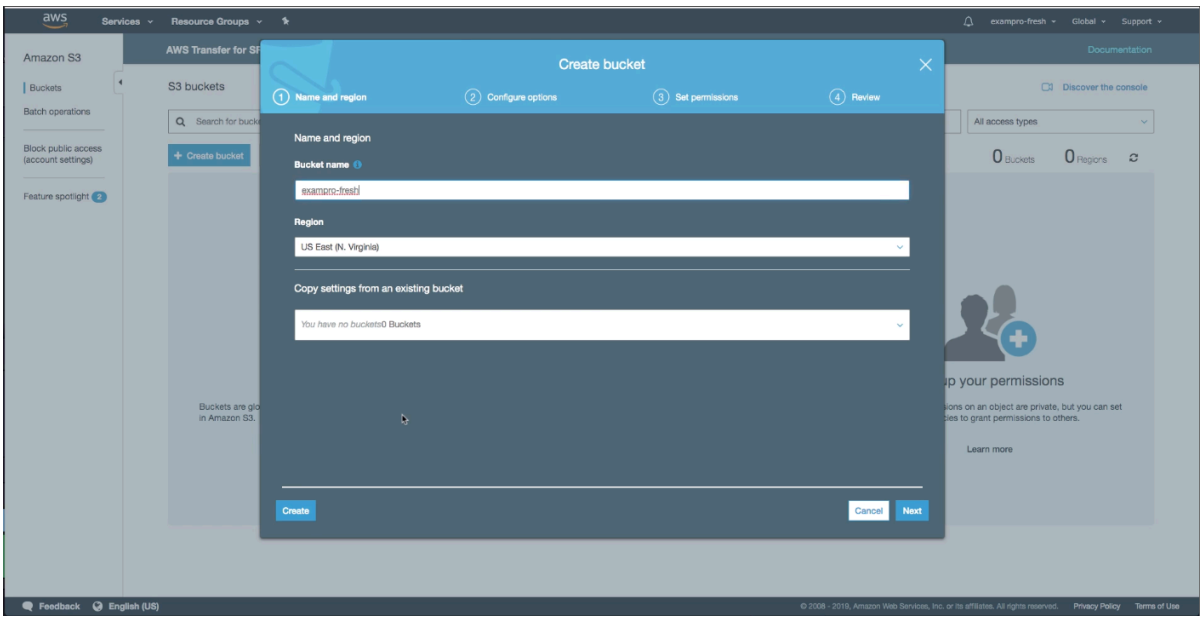



## Creating a Bucket

- 2.1 [ ] Click the "Create bucket " button

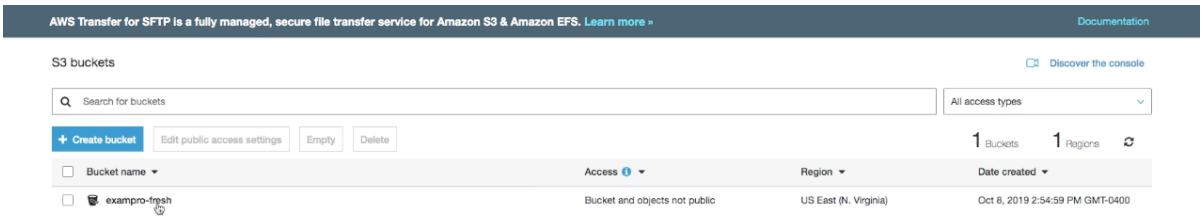



- 2.2 [ ] Create a unique Bucket name
- 2.3 [ ] Pick a region
- 2.4 [ ] Click the "Create" button on the bottom of the model

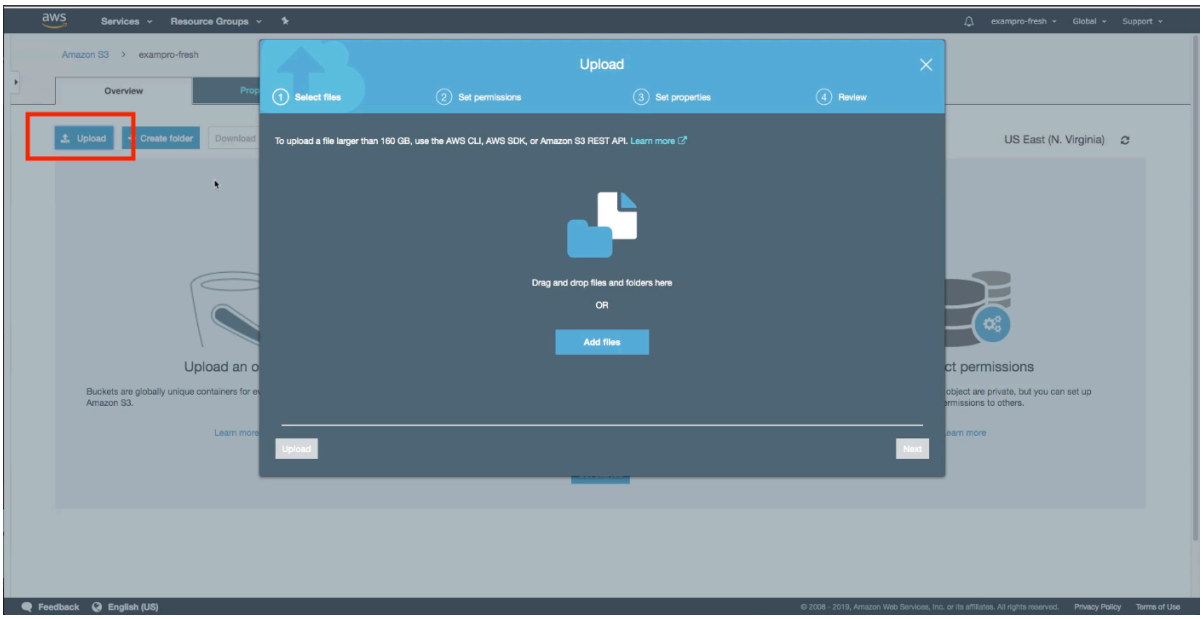


# Uploading a File to the Bucket

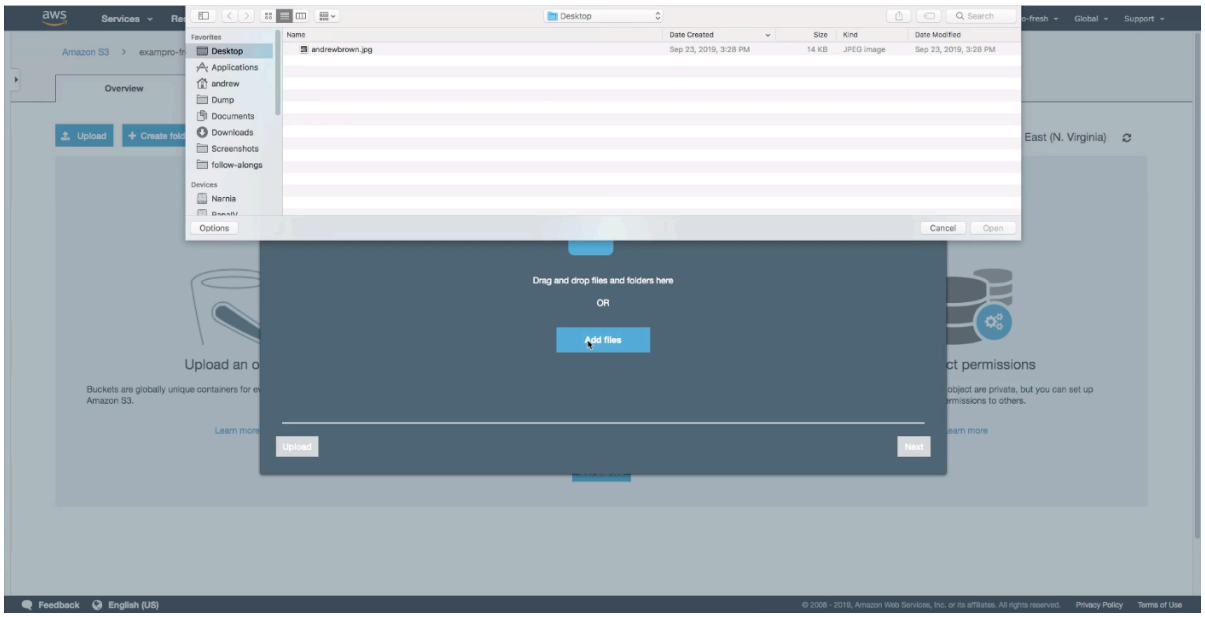
- 3.1 [ ] Click on the newly created bucket



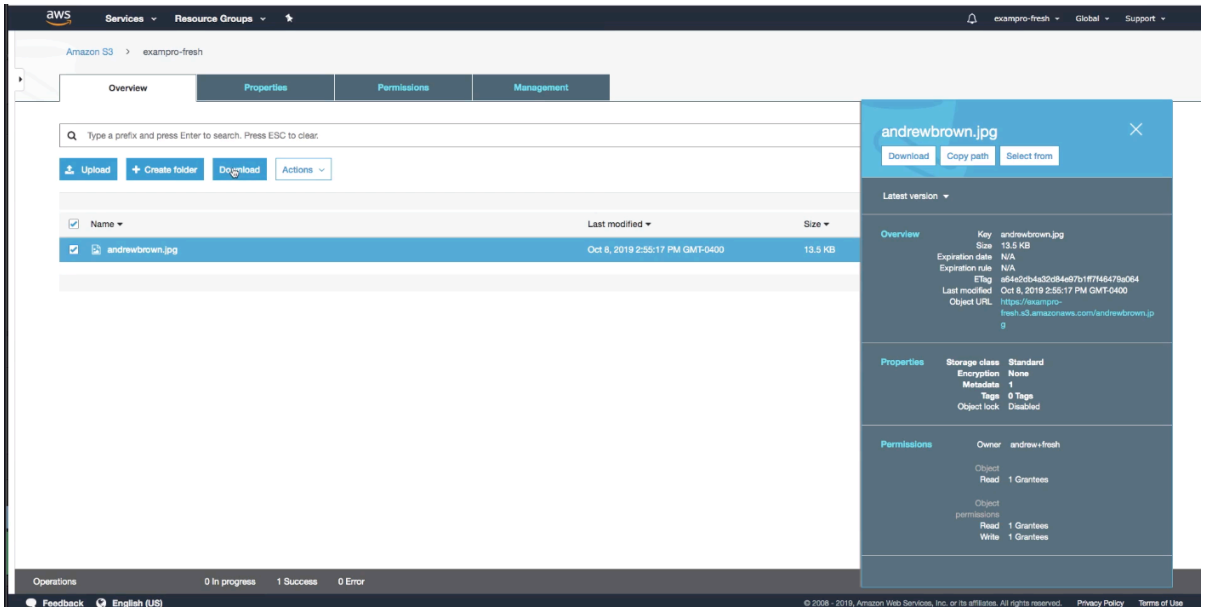

- 3.2 [ ] Click the Upload button



- 3.3 [ ] Drop in a file from your computer like an image
- 3.4 [ ] Click the Upload button in the bottom right corner of the upload model




- 3.5 [ ] You should now see the file you uploaded in your S3 bucket

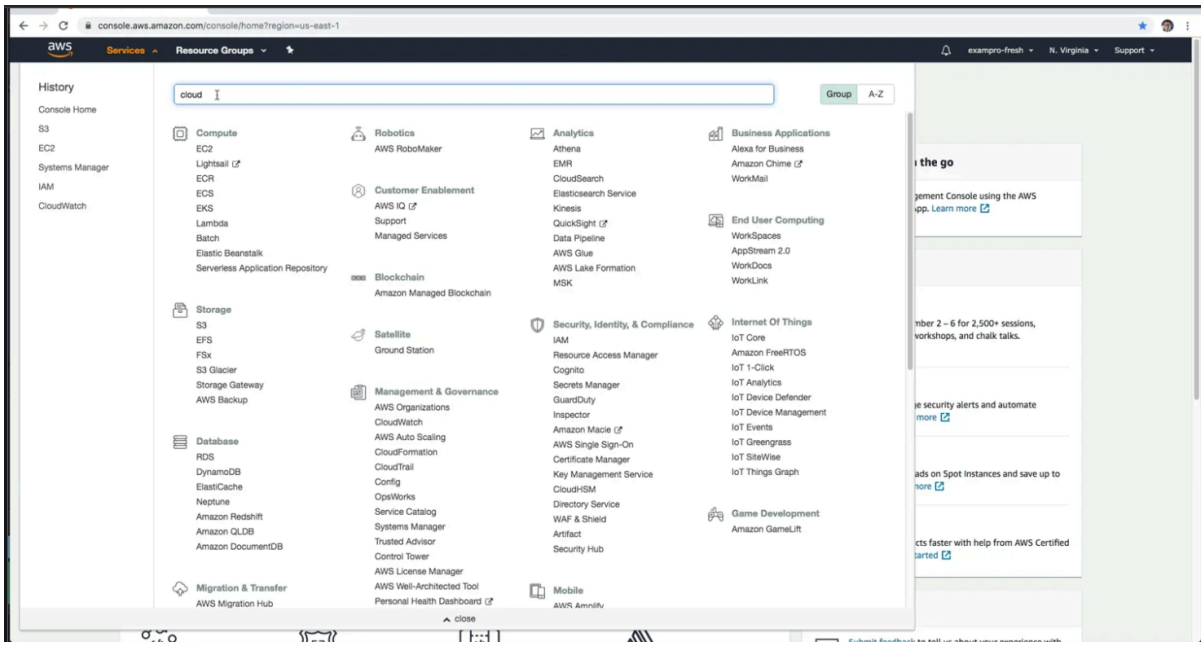


# Creating a CloudFront Distribution

We are going to walk through how to create a CloudFront Distribution

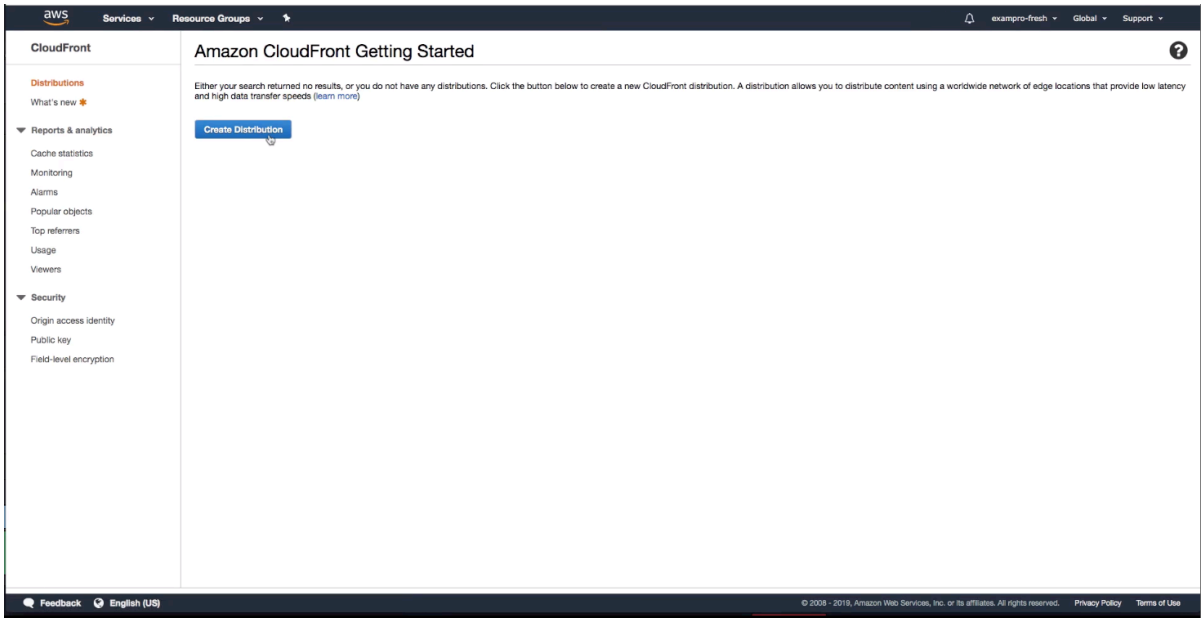
## Navigating to CloudFront

- 1.1 [ ] Inside the AWS Management Console use the search bar and type in CloudFront

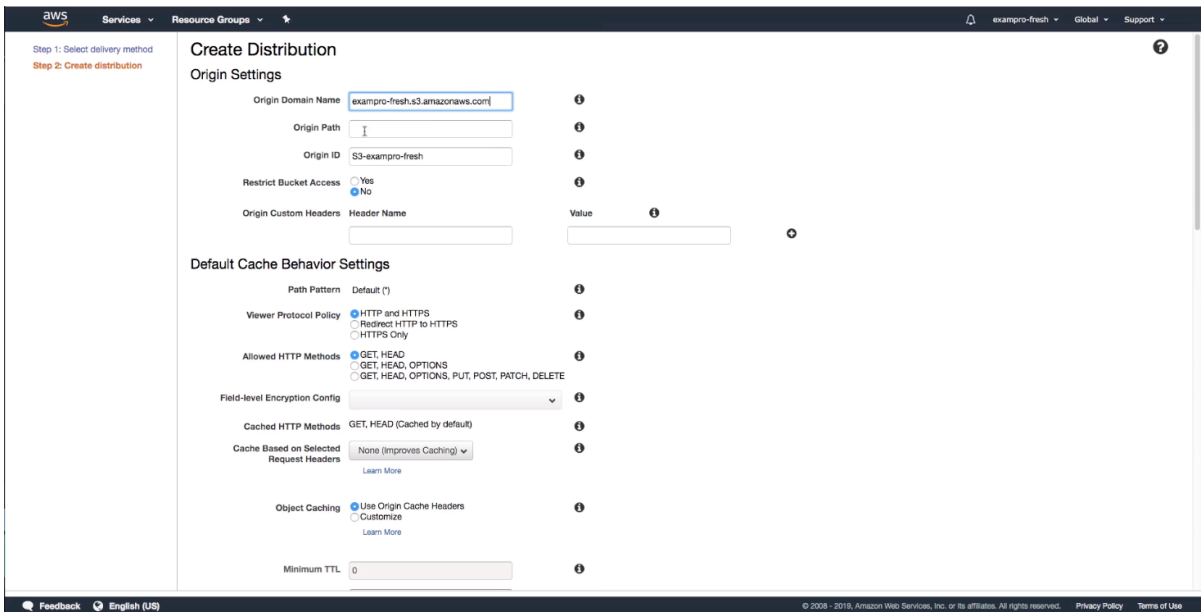



## Creating a New Distribution

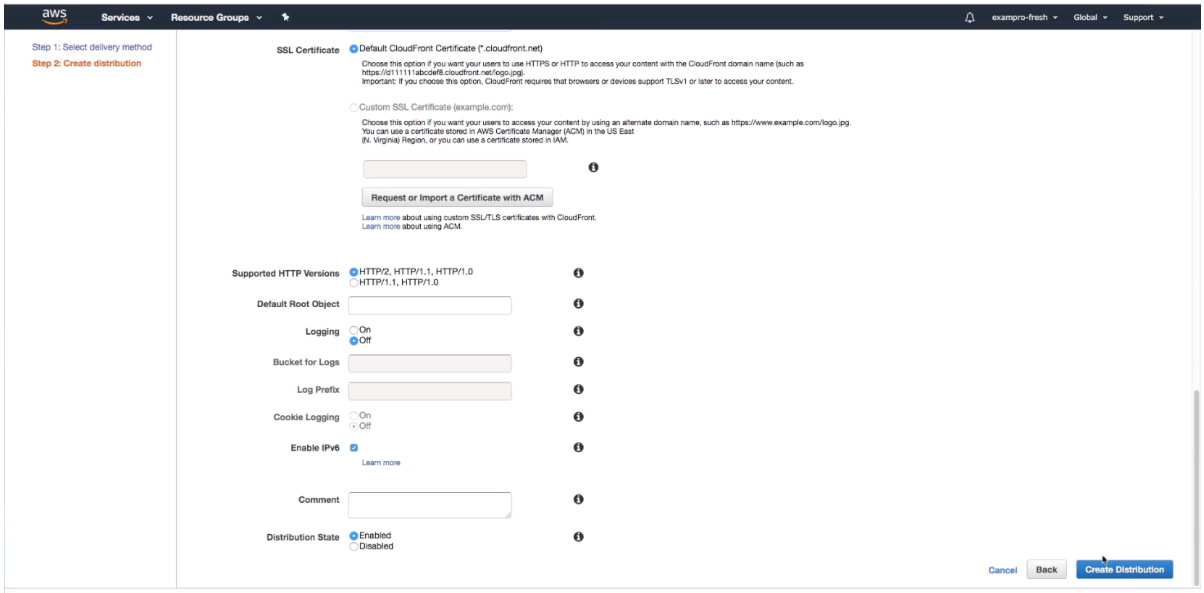
- 2.1 [ ] On the Amazon CloudFront Getting Started page click the "Create Distribution" button.



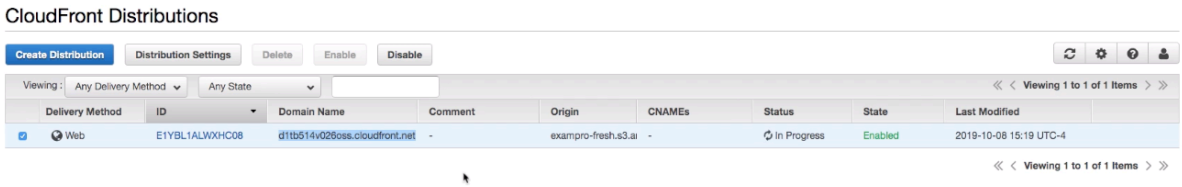

- 2.2 [ ] On the Create Distribution Form click the Origin Domain Name to get a drop down and choose the S3 bucket you just created.




- 2.3 [ ] At the bottom of the page click the "Create Distribution" button.

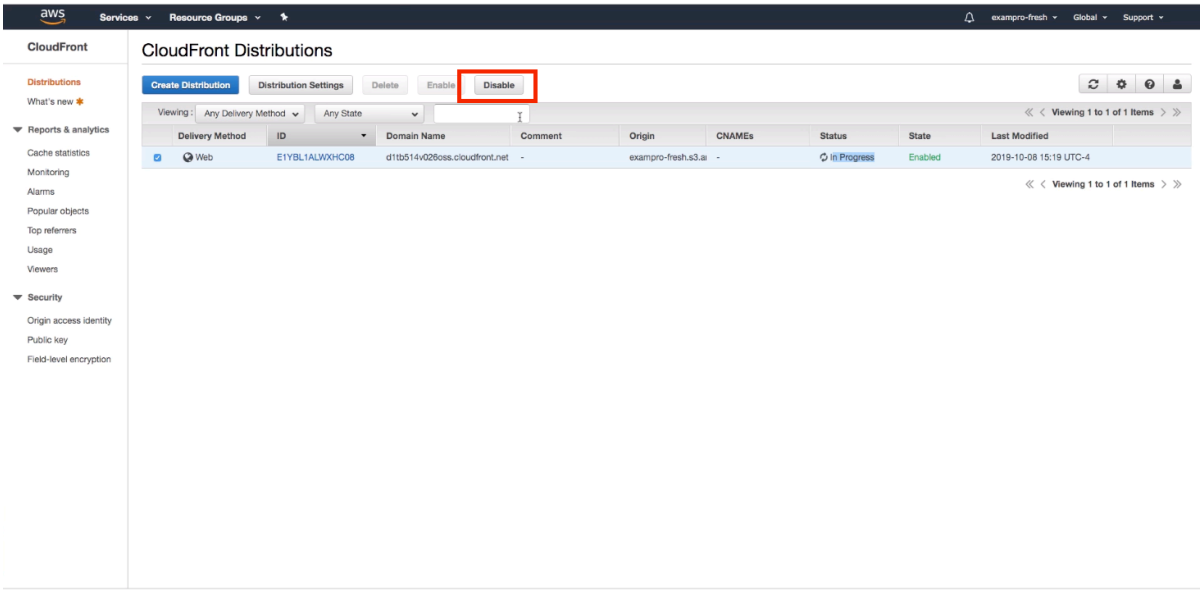



- 2.4 [ ] Confirm that the distribution is in progress and you have a unique domain name




## Disabling the Distribution

- 3.1 [ ] Click the Disable button to disable your CloudFront Distribution

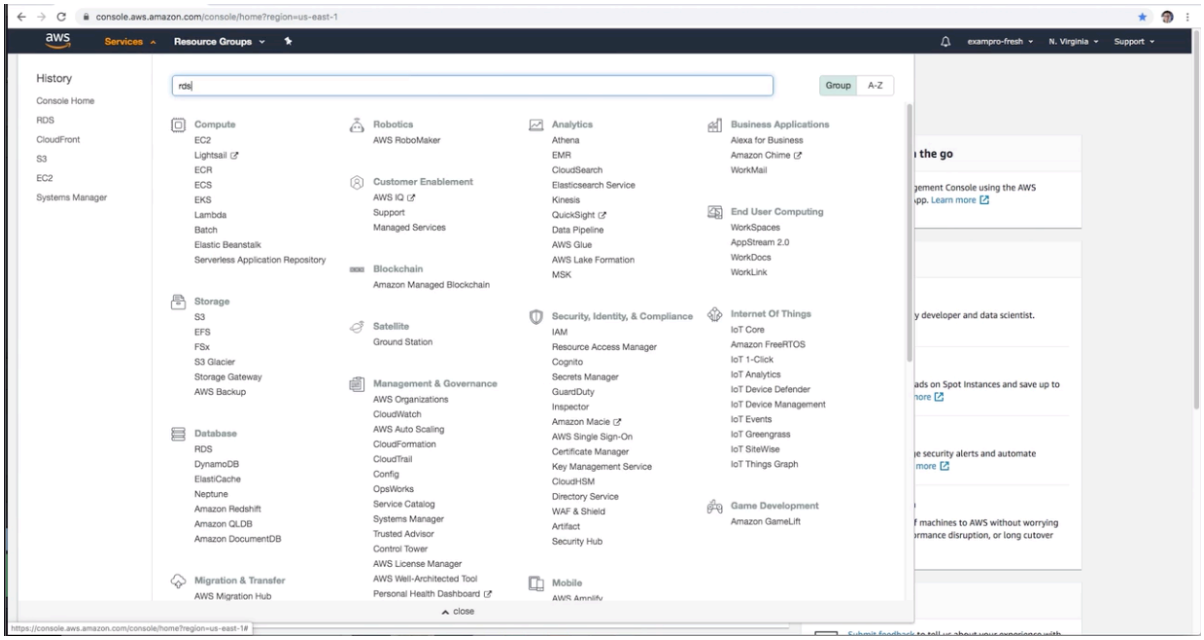



# Creating a Relational Database

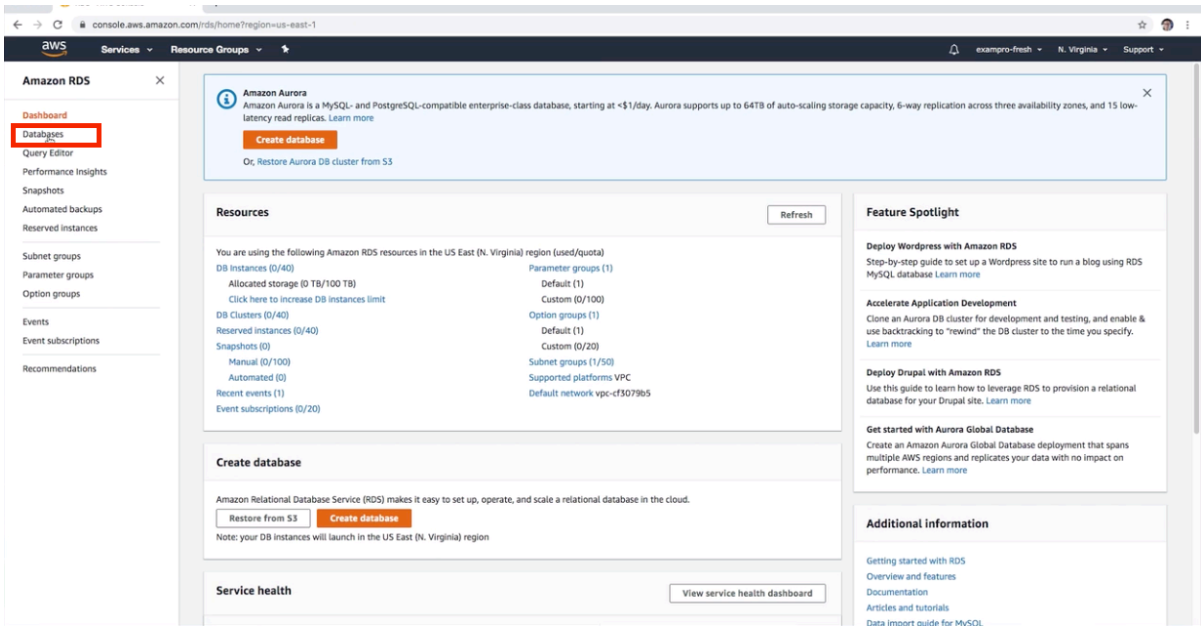
We are going to walk through how to create a Relational Database

## Creating a RDS

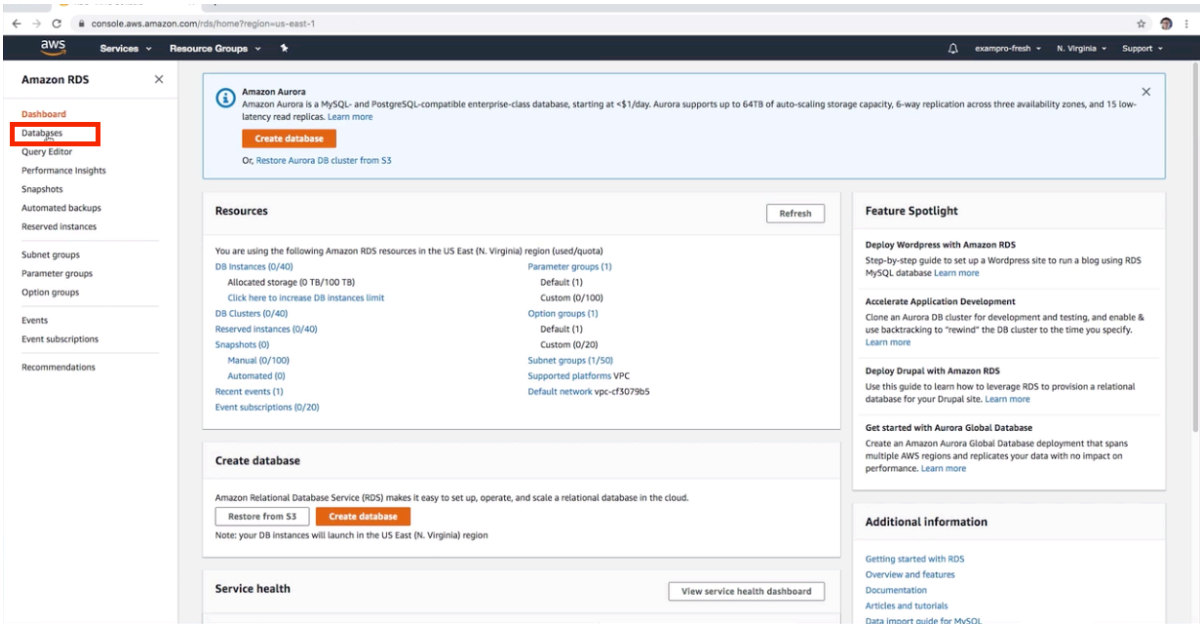
- 1.1 [ ] Inside the AWS Management Console use the search bar and type in RDS



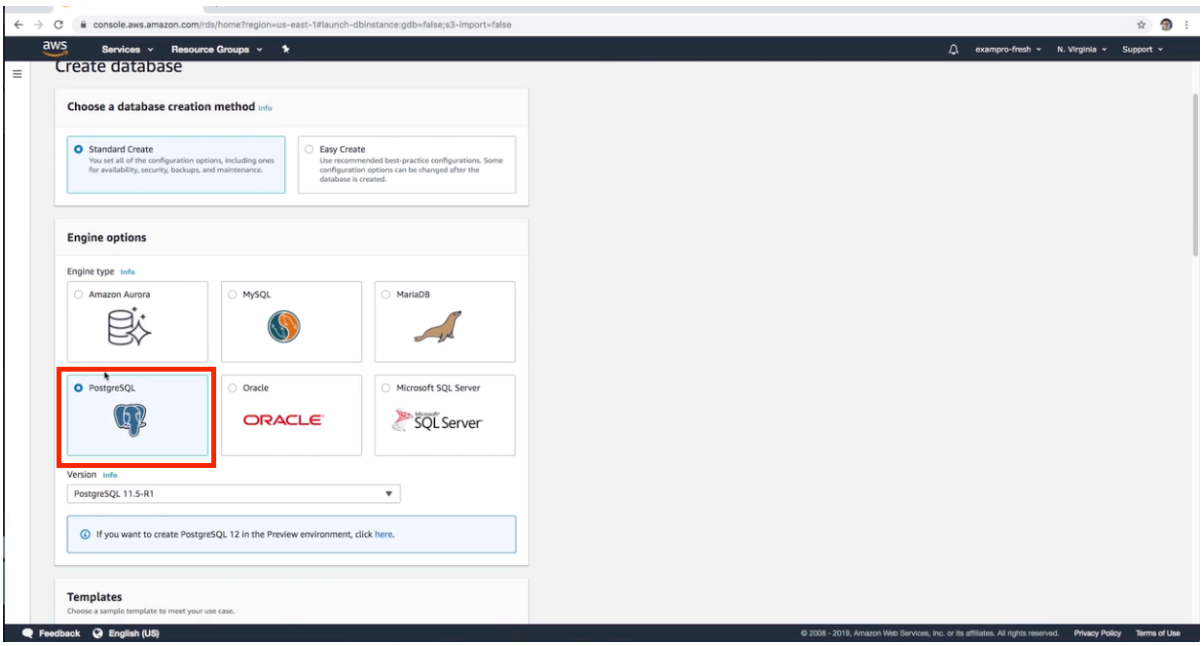

- 1.2 [ ] Click the "Databases" link



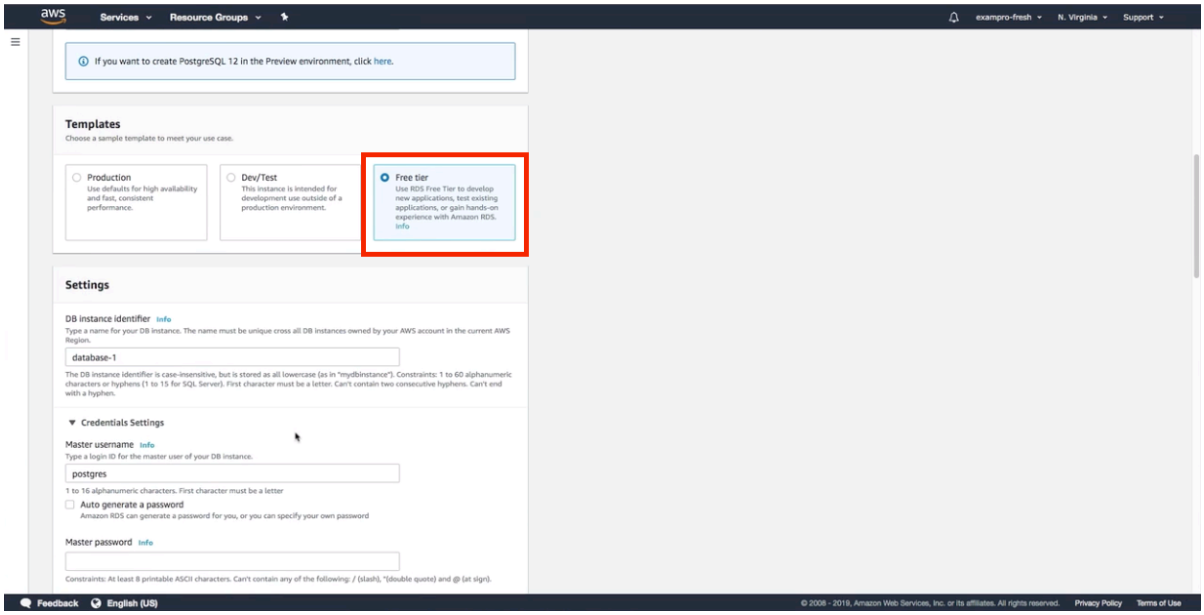

- 1.3 [ ] Click the "Create database" button



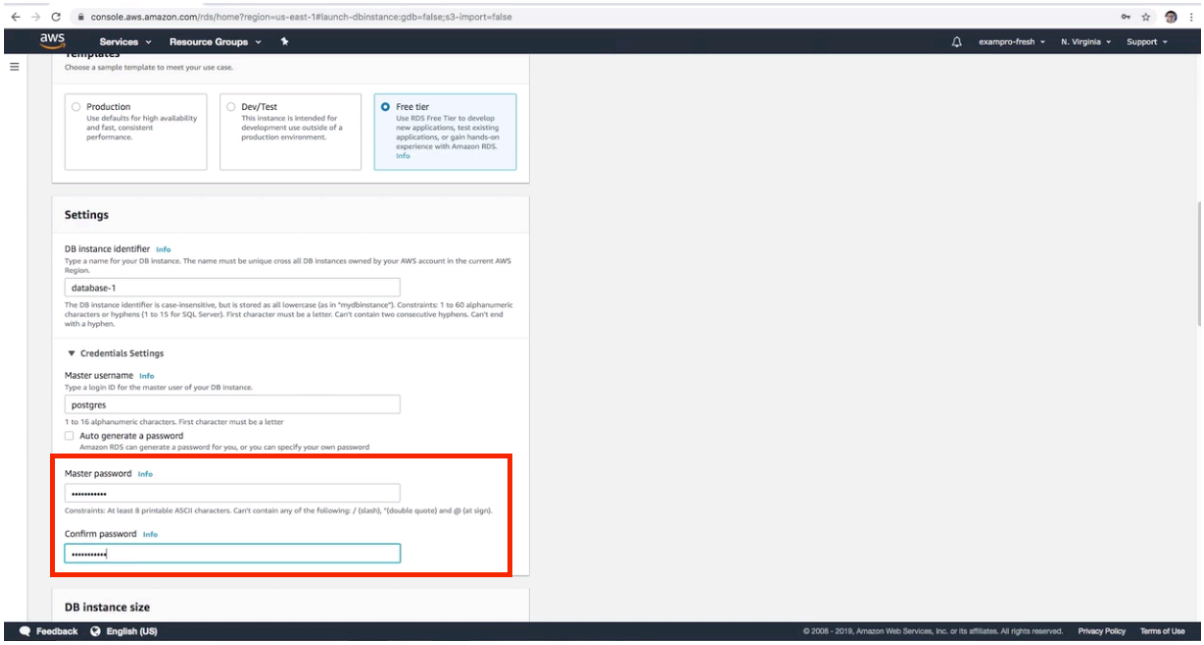

- [ ] 1.4 [ ] Choose the PostgreSQL engine option



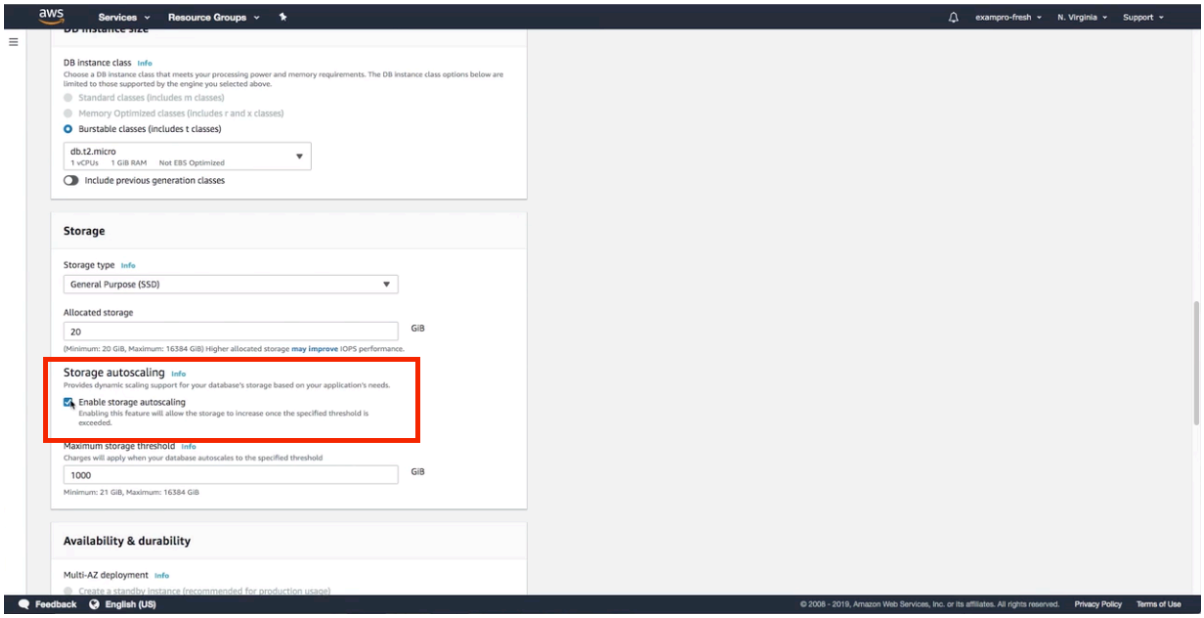

- [ ] 1.5 [ ] Choose the Free Tier option under Templates



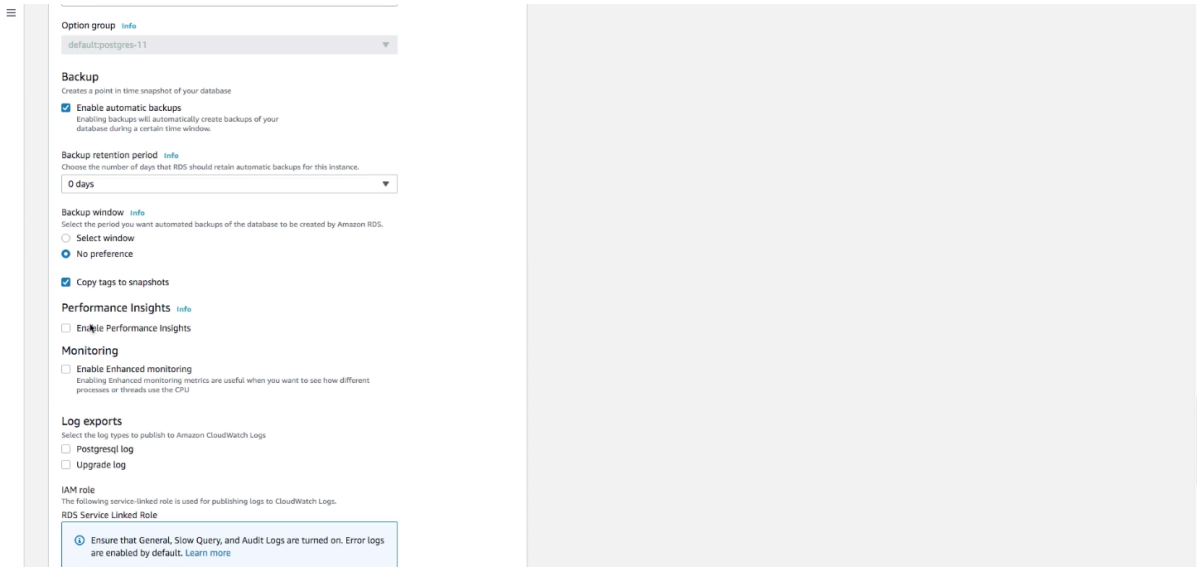

- 1.6 [ ] Set a password



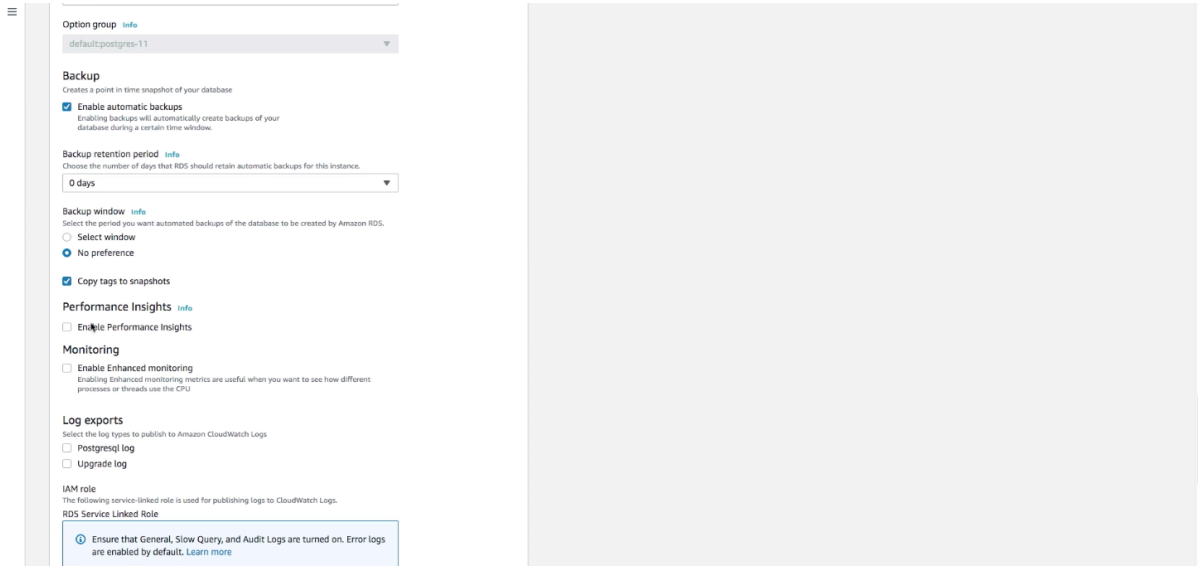

- 1.7 [ ] Leave the DB instance size on db.t2.mirco
- 1.8 [ ] Disable the storage autoscaling




- 1.9 [ ] Set up additional configuration with an initial database name
- 1.10 [ ] Turn backups off
- 1.11 [ ] Change backup retention period to 0 days




- 1.12 [ ] Disable performance insights




- 1.13 [ ] Click the "Create database" button

When you are finished make sure to delete the database.