Amity School of Engineering and Technology (ASET)

Cloud Computing Practitioner [CSE314]

Practical Lab File

Name Dhruv Rastogi
Class 5CSE 10Y
Roll No A2372019006
Batch 2019 - 2023
Faculty Ms. Dolly Sharma

CONTENTS

Practical 1

AIM
Introduction to Amazon EC2

THEORY
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

By the end of this lab, you will be able to:
- Launch a web server with termination protection enabled
- Monitor Your EC2 instance
- Modify the security group that your web server is using to allow HTTP access
- Resize your Amazon EC2 instance to scale
- Explore EC2 limits
- Test termination protection
- Terminate your EC2 instance

PROCEDURE

Task 1: Launch Your Amazon EC2 Instance

In this task, you will launch an Amazon EC2 instance with *termination protection*. Termination protection prevents you from accidentally terminating an EC2 instance. You will deploy your instance with a User Data script that will allow you to deploy a simple web server.

5. In the **AWS Management Console** on the **Services** menu, click **EC2**.
6. Choose **Launch Instance**, then select **Launch Instance**

Step 1: Choose an Amazon Machine Image (AMI)

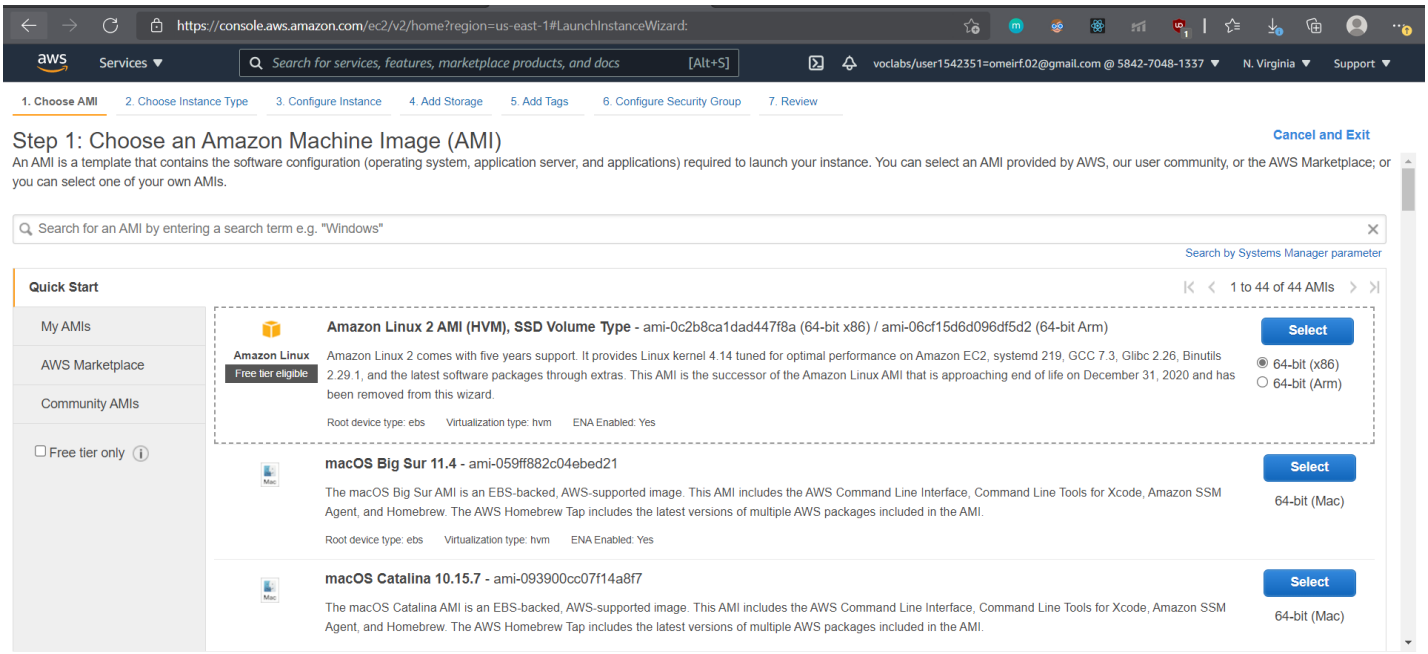
An **Amazon Machine Image (AMI)** provides the information required to launch an instance, which is a virtual server in the cloud. An AMI includes:

- A template for the root volume for the instance (for example, an operating system or an application server with applications)

- Launch permissions that control which AWS accounts can use the AMI to launch instances
- A block device mapping that specifies the volumes to attach to the instance when it is launched

The **Quick Start** list contains the most commonly-used AMIs. You can also create your own AMI or select an AMI from the AWS Marketplace, an online store where you can sell or buy software that runs on AWS.

7. Click **Select** next to **Amazon Linux 2 AMI** (at the top of the list).

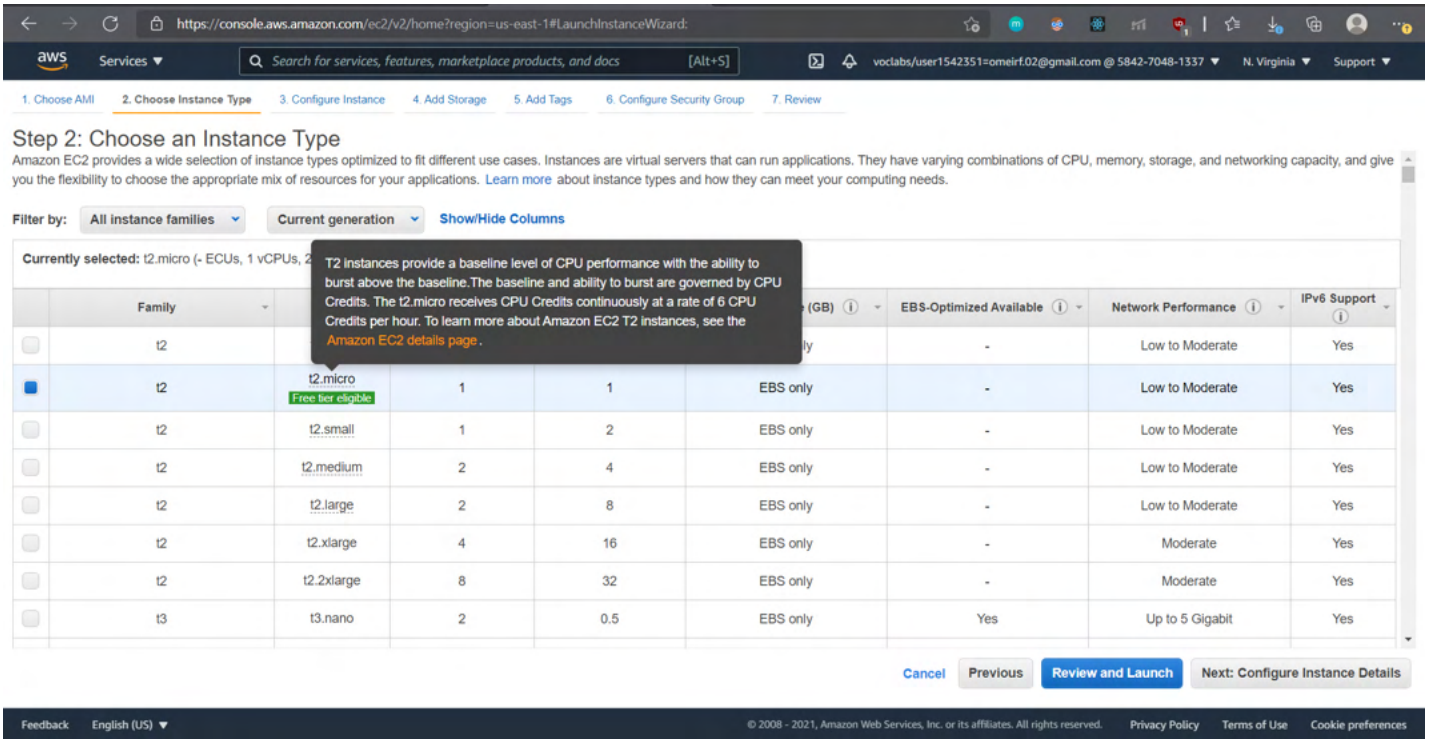


Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of *instance types* optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more *instance sizes*, allowing you to scale your resources to the requirements of your target workload.

You will use a **t2.micro** instance which should be selected by default. This instance type has 1 virtual CPU and 1 GiB of memory. **NOTE**: You may be restricted from using other instance types in this lab.

8. Click **Next: Configure Instance Details**



Step 3: Configure Instance Details

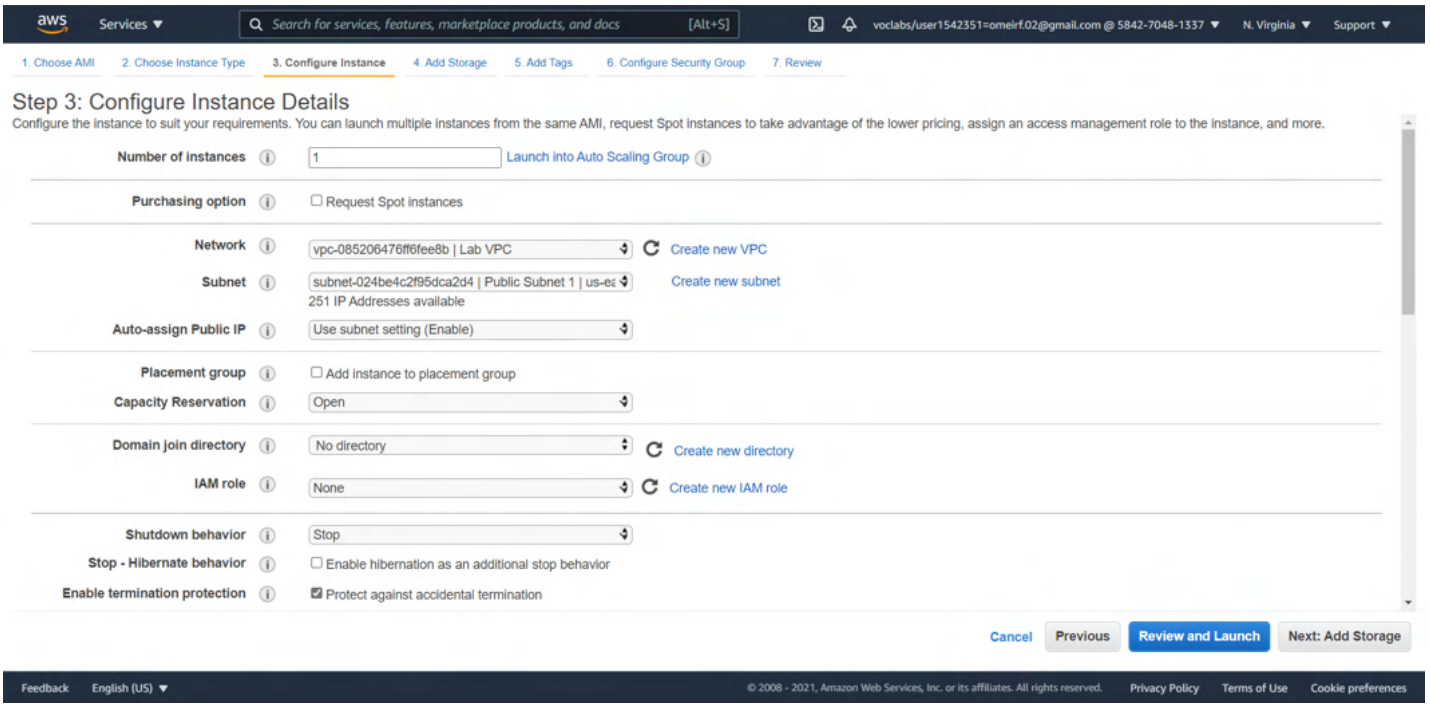
This page is used to configure the instance to suit your requirements. This includes networking and monitoring settings.

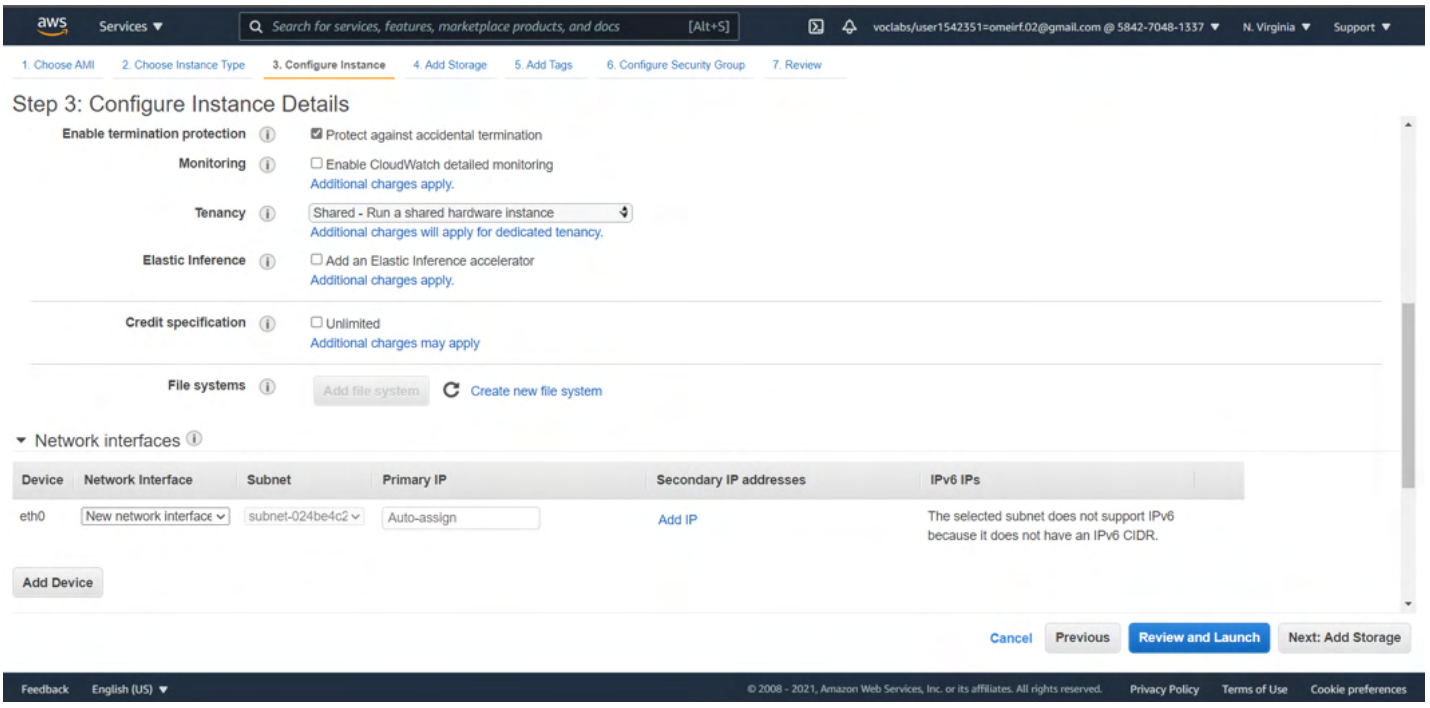
The **Network** indicates which Virtual Private Cloud (VPC) you wish to launch the instance into. You can have multiple networks, such as different ones for development, testing and production.

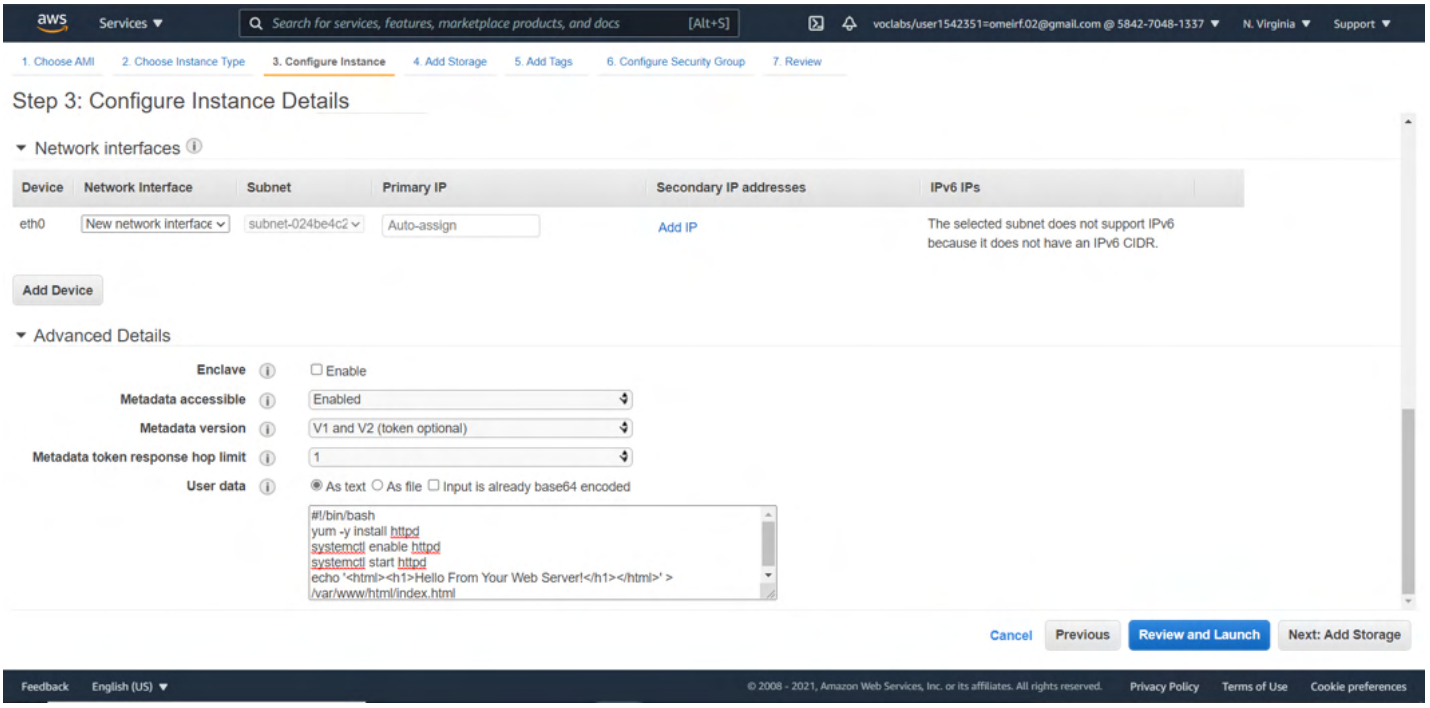
9. For **Network**, select **Lab VPC**.
   The Lab VPC was created using an AWS CloudFormation template during the setup process of your lab. This VPC includes two public subnets in two different Availability Zones.
10. For **Enable termination protection**, select **Protect against accidental termination**.
    When an Amazon EC2 instance is no longer required, it can be *terminated*, which means that the instance is stopped and its resources are released. A terminated instance cannot be started again. If you want to prevent the instance from being accidentally terminated, you can enable *termination protection* for the instance, which prevents it from being terminated.
11. Scroll down, then expand **Advanced Details**.
    A field for **User data** will appear.
    When you launch an instance, you can pass *user data* to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.
    Your instance is running Amazon Linux, so you will provide a *shell script* that will run when the instance starts.
12. Copy the following commands and paste them into the **User data** field:
13.
```
#!/bin/bash
yum -y install httpd
systemctl enable httpd
systemctl start httpd
```

```
echo '<html><h1>Hello From Your Web Server!</h1></html>' >
/var/www/html/index.html
```

14. The script will:
    - Install an Apache web server (httpd)
    - Configure the web server to automatically start on boot
    - Activate the Web server
    - Create a simple web page
13. Click **Next: Add Storage**





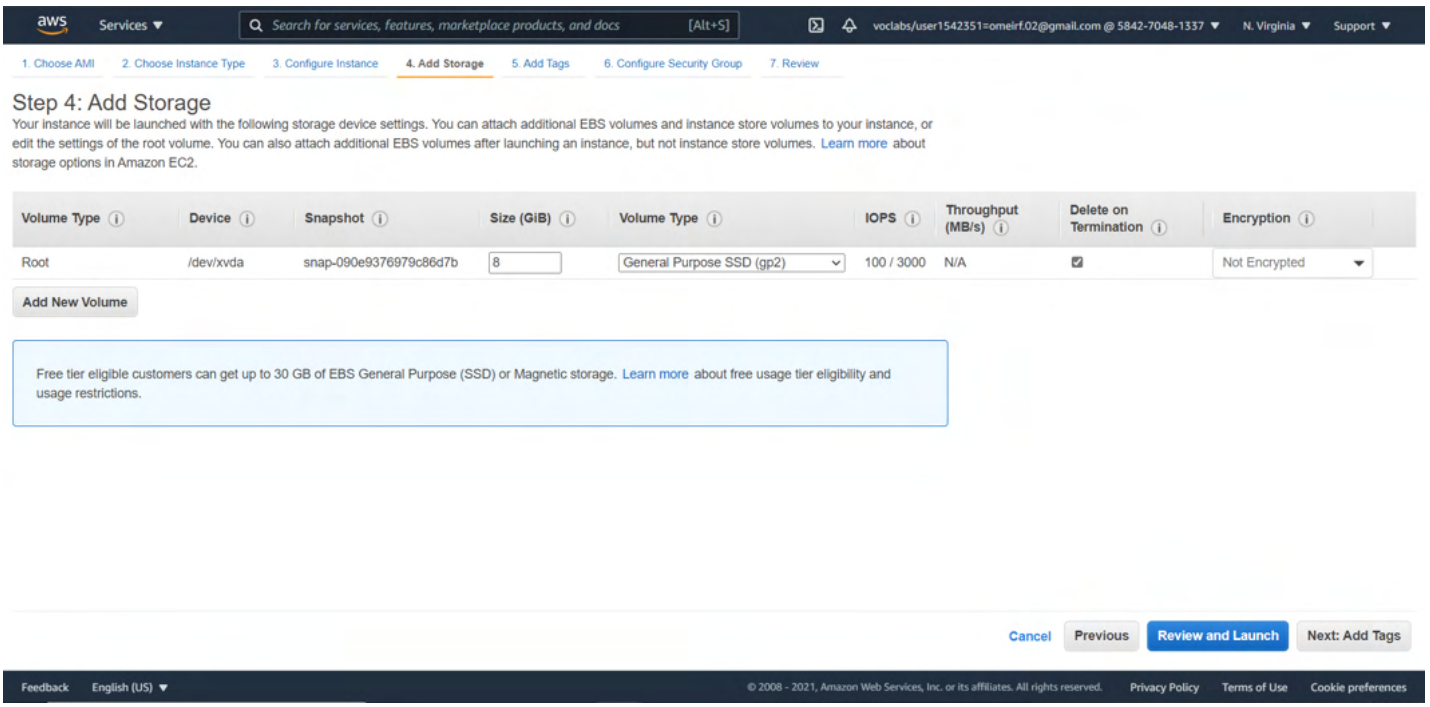


Step 4: Add Storage

Amazon EC2 stores data on a network-attached virtual disk called *Elastic Block Store*.

You will launch the Amazon EC2 instance using a default 8 GiB disk volume. This will be your root volume (also known as a 'boot' volume).

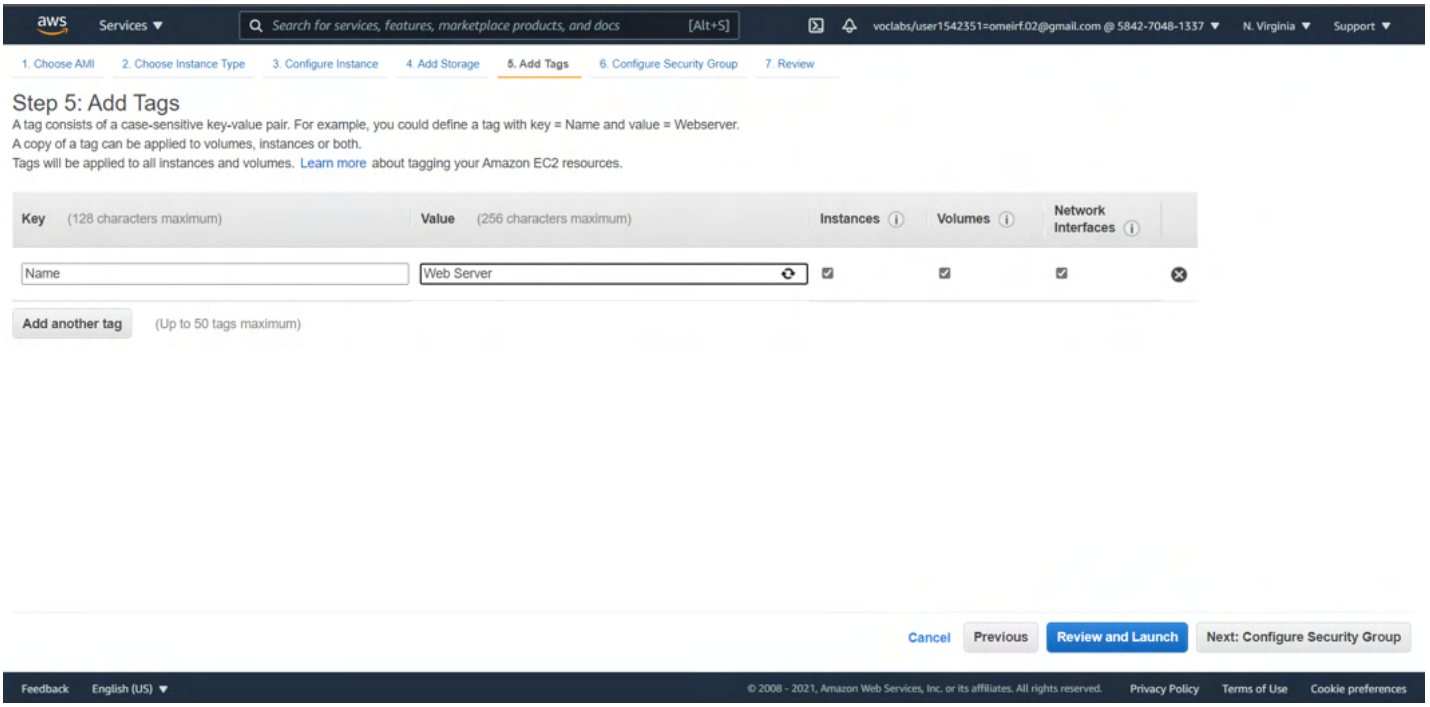
14. Click **Next: Add Tags**



Step 5: Add Tags

Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you have assigned to it. Each tag consists of a Key and a Value, both of which you define.

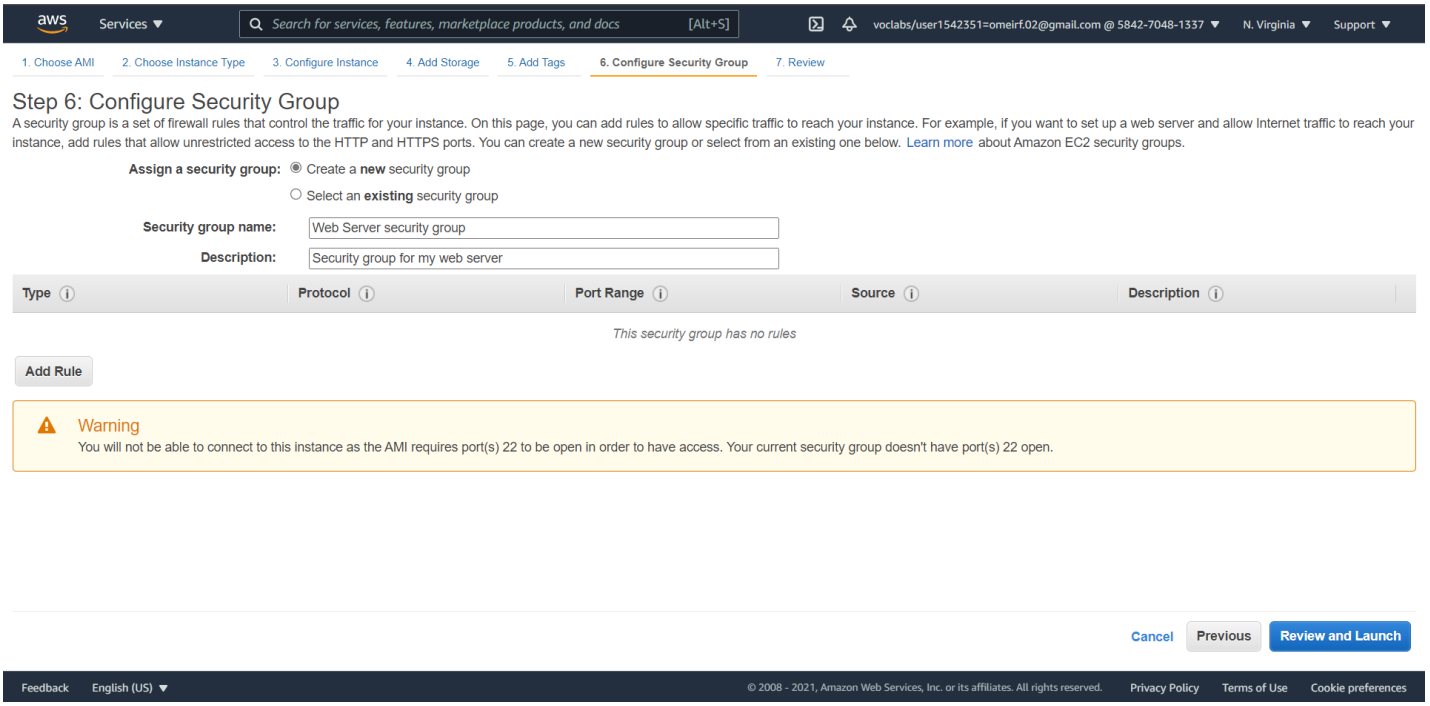
15. Click **Add Tag** then configure:
    - **Key:** Name
    - **Value:** Web Server
16. Click **Next: Configure Security Group**



Step 6: Configure Security Group

A *security group* acts as a virtual firewall that controls the traffic for one or more instances. When you launch an instance, you associate one or more security groups with the instance. You add *rules* to each security group that allow traffic to or from its associated instances. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances that are associated with the security group.

17. On **Step 6: Configure Security Group**, configure:
    - **Security group name:** Web Server security group
    - **Description:** Security group for my web server
18. In this lab, you will not log into your instance using SSH. Removing SSH access will improve the security of the instance.
19. Delete the existing SSH rule.
20. Click **Review and Launch**



Step 7: Review Instance Launch

The Review page displays the configuration for the instance you are about to launch.

20. Click **Launch**

A **Select an existing key pair or create a new key pair** window will appear.

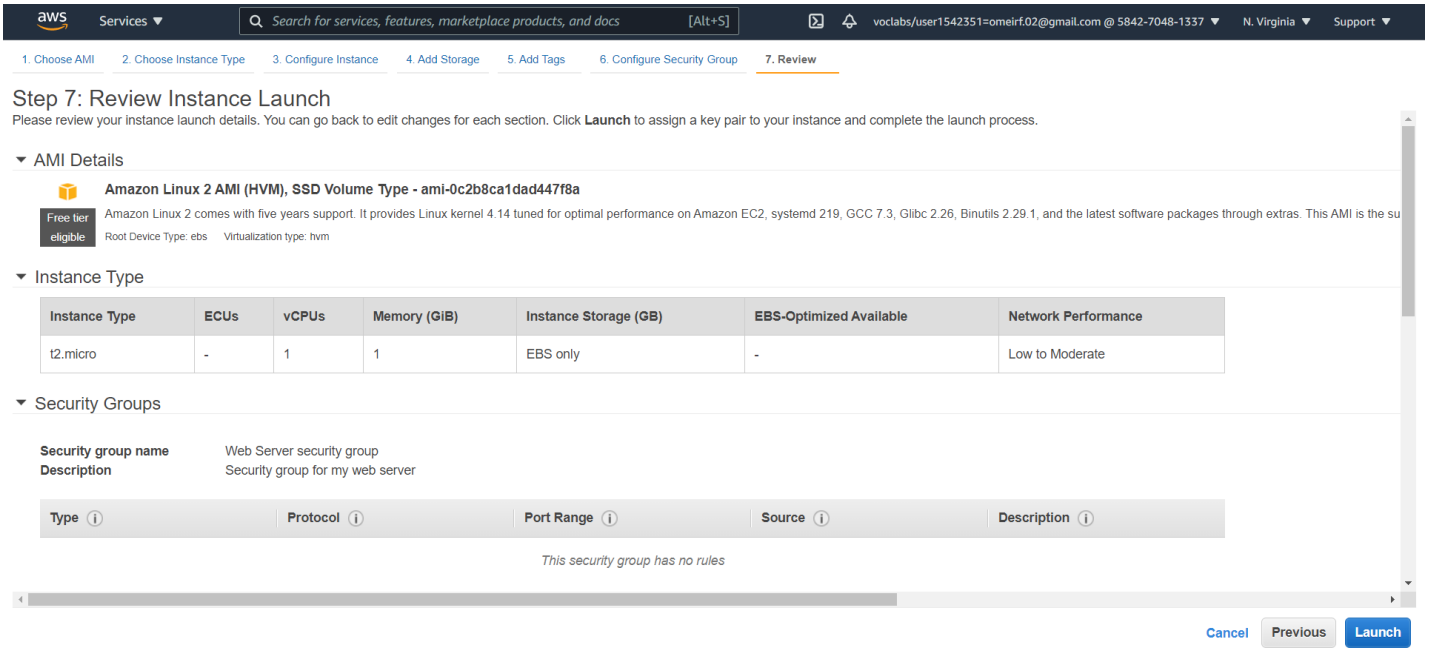
Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

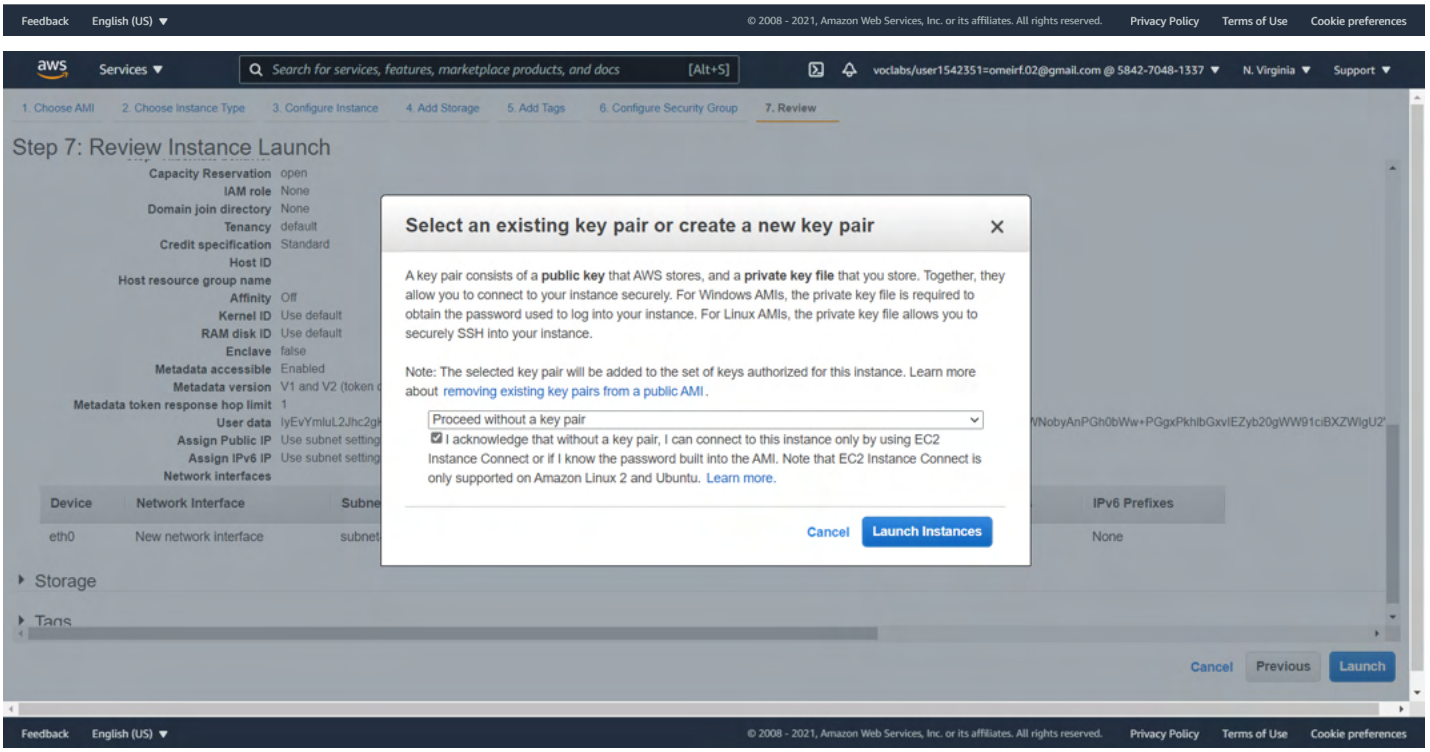
In this lab you will not log into your instance, so you do not require a key pair.

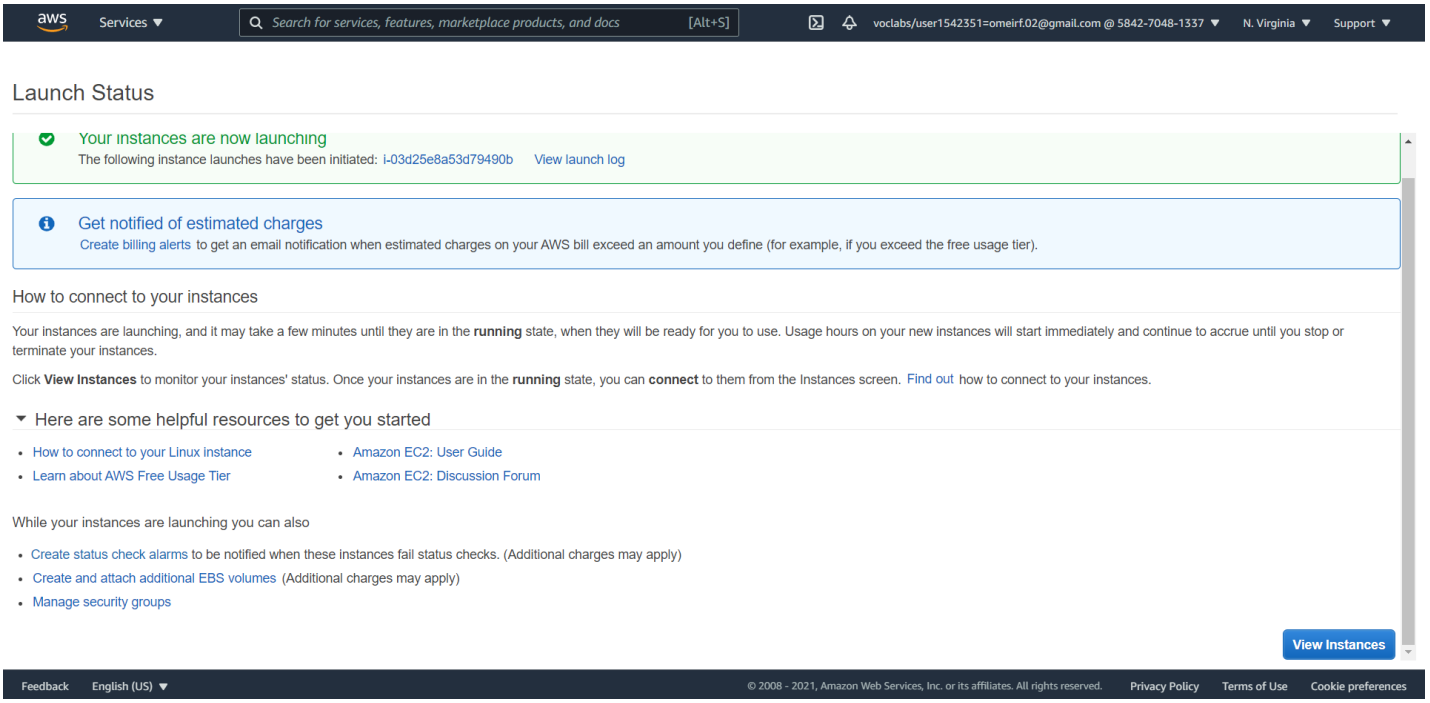
21. Click the **Choose an existing key pair** drop-down and select *Proceed without a key pair*.
22. Select **I acknowledge that ...**.
23. Click **Launch Instances**
    Your instance will now be launched.
24. Click **View Instances**
    The instance will appear in a *pending* state, which means it is being launched. It will then change to *running*, which indicates that the instance has started booting. There will be a short time before you can access the instance.
    The instance receives a *public DNS name* that you can use to contact the instance from the Internet.
    Your **Web Server** should be selected. The **Description** tab displays detailed information about your instance.
    To view more information in the Description tab, drag the window divider upwards.
    Review the information displayed in the **Description** tab. It includes information about the instance type, security settings and network settings.

25. Wait for your instance to display the following:
- **Instance State:** running
- **Status Checks:** 2/2 checks passed

Congratulations! You have successfully launched your first Amazon EC2 instance.







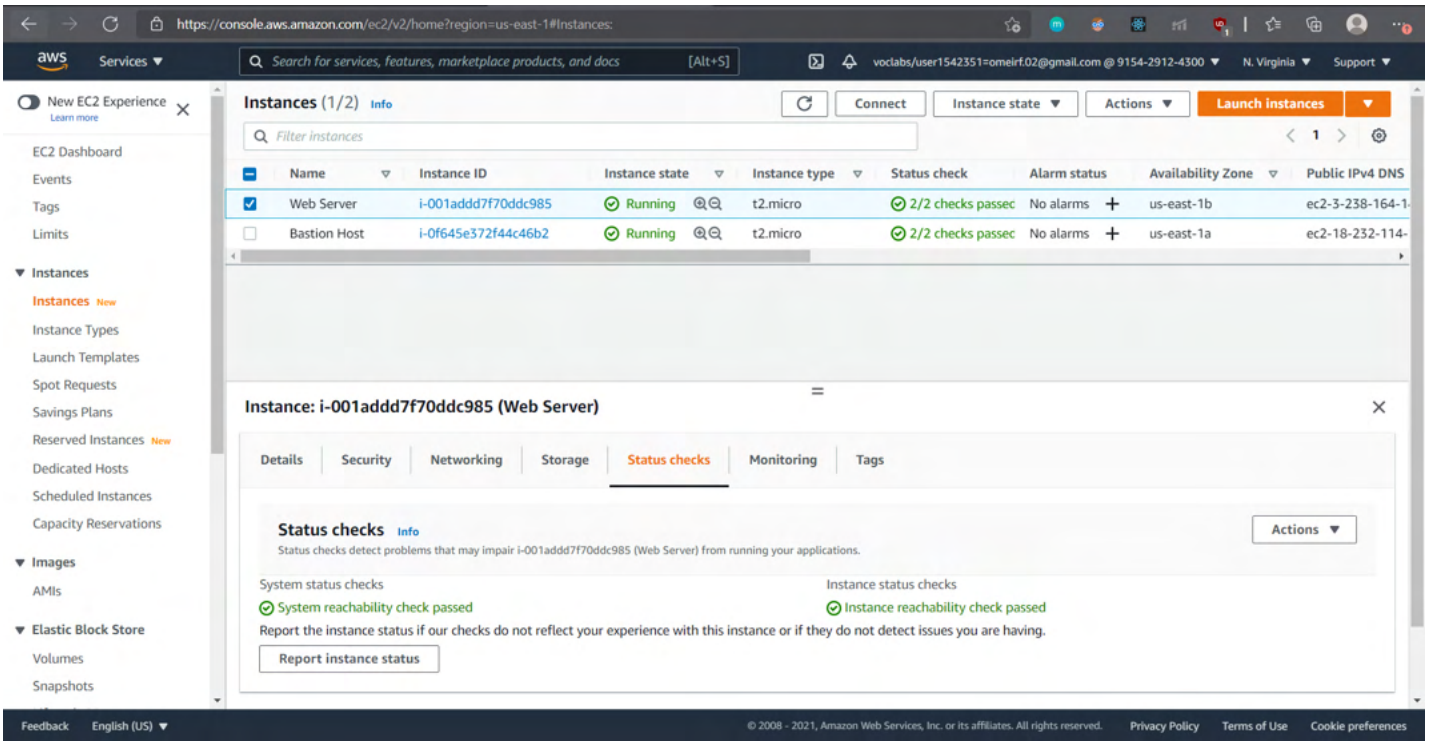
Task 2: Monitor Your Instance

Monitoring is an important part of maintaining the reliability, availability, and performance of your Amazon Elastic Compute Cloud (Amazon EC2) instances and your AWS solutions.

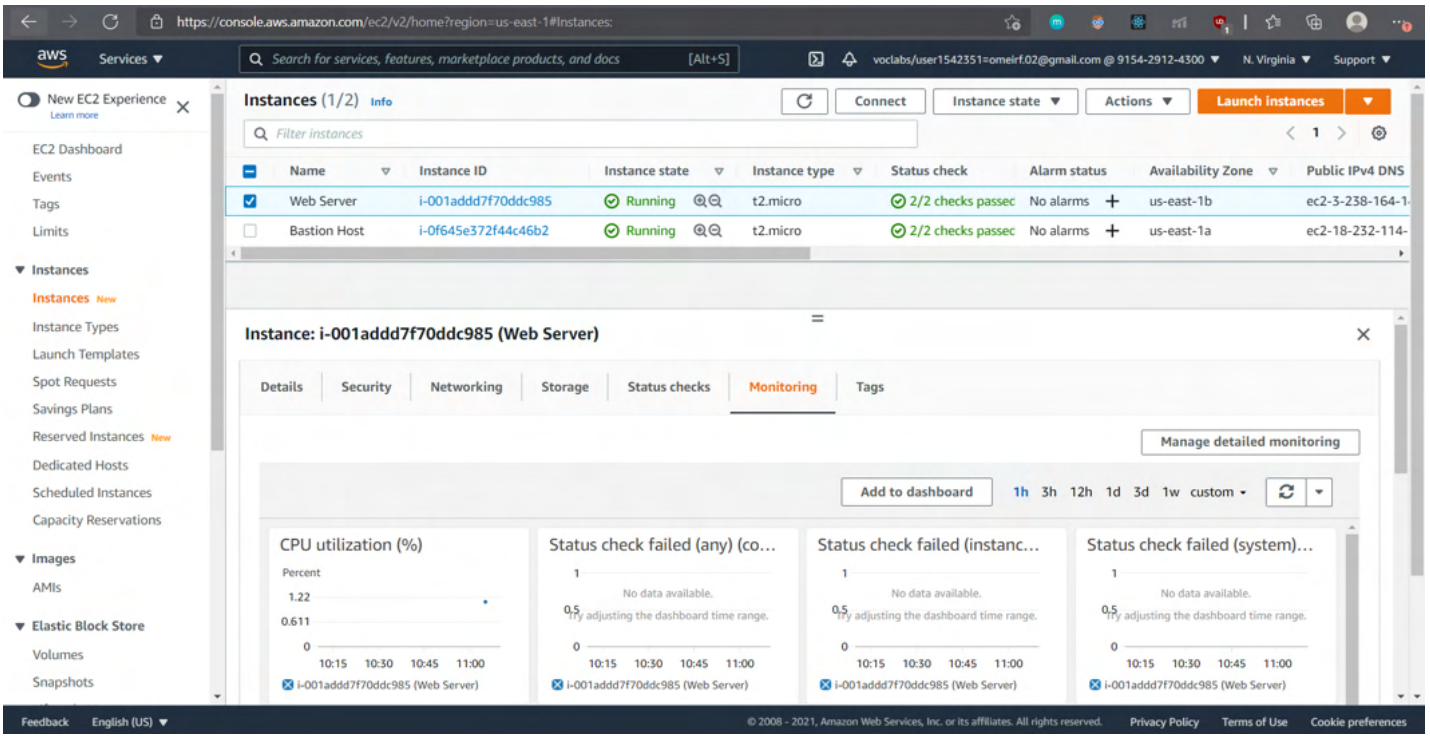
26. Click the **Status Checks** tab.

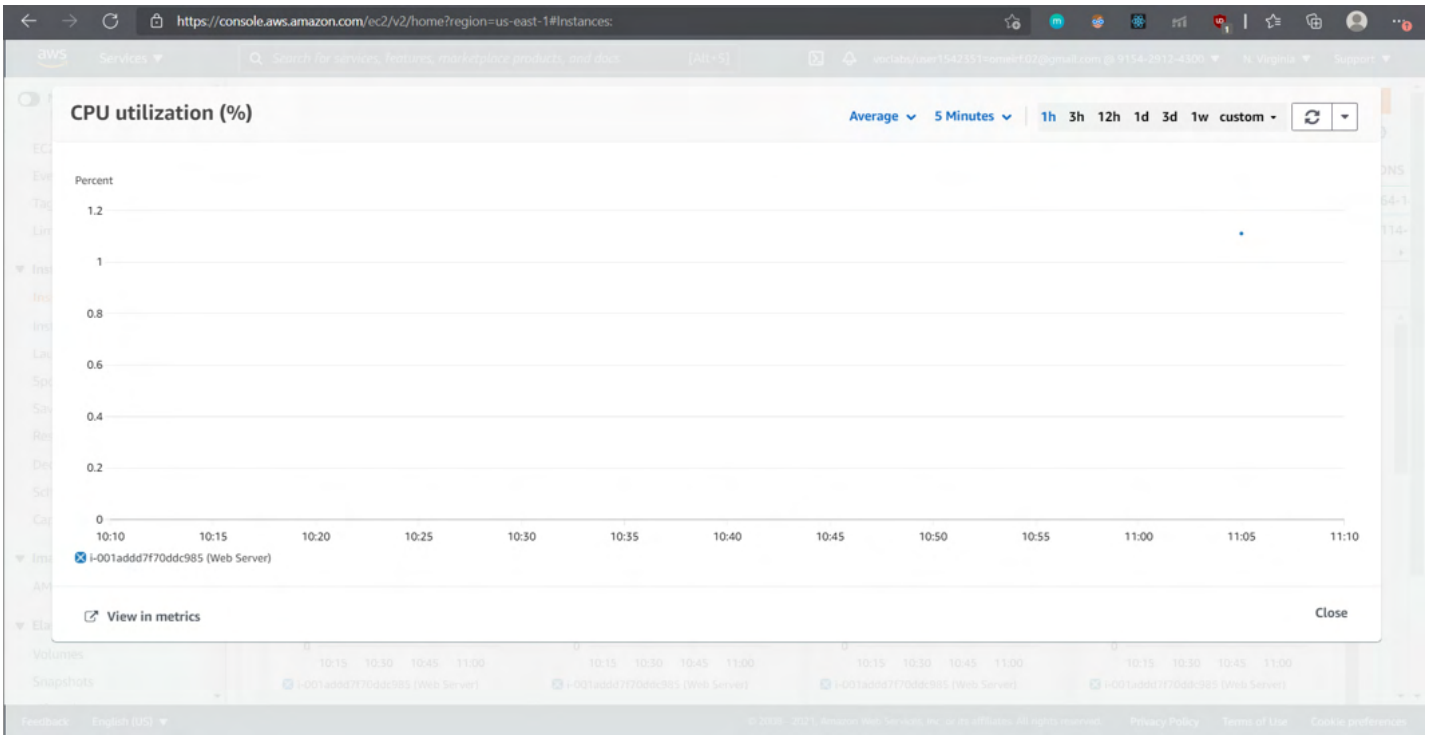
    With instance status monitoring, you can quickly determine whether Amazon EC2 has detected any problems that might prevent your instances from running applications. Amazon EC2 performs automated checks on every running EC2 instance to identify hardware and software issues.

    Notice that both the **System reachability** and **Instance reachability** checks have passed.



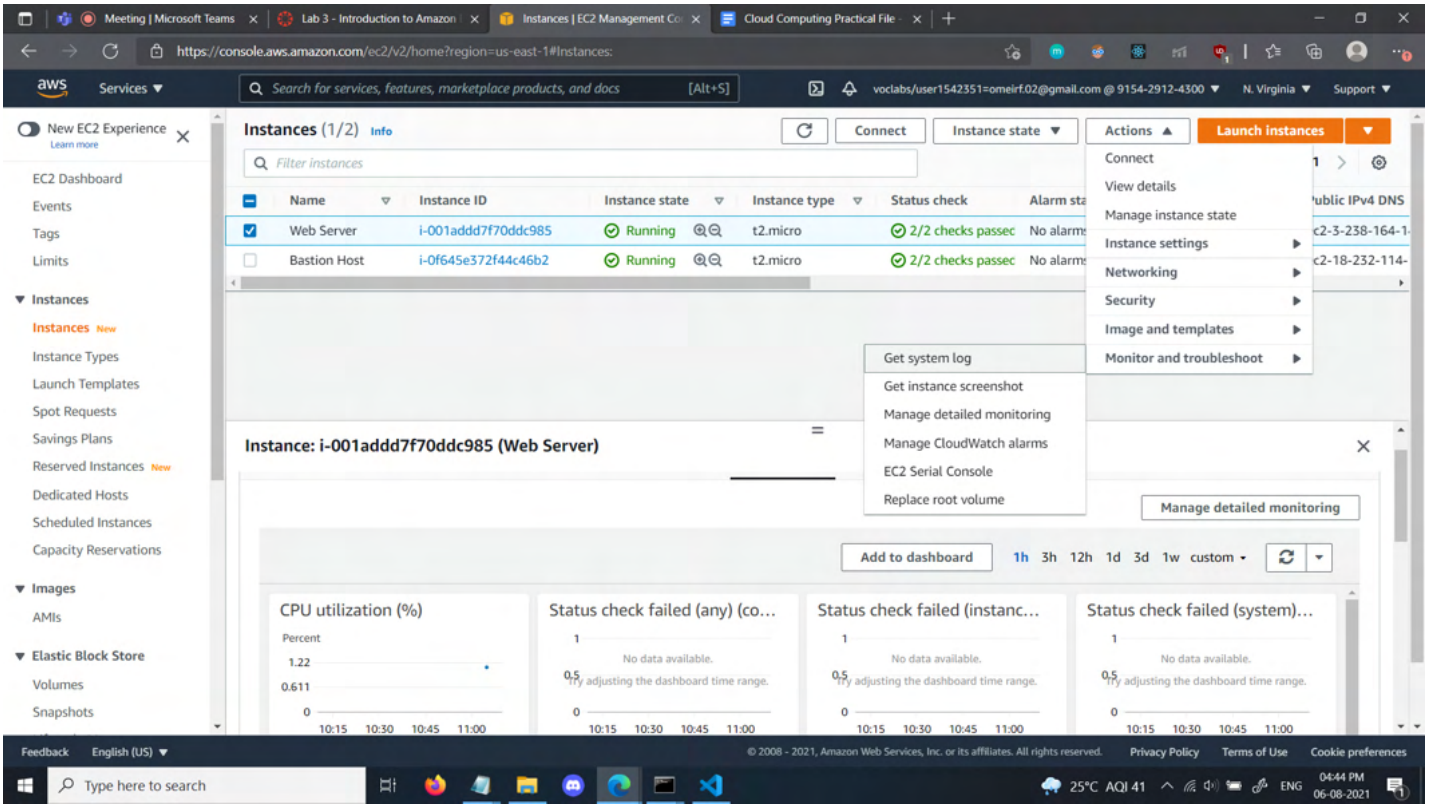
27. Click the **Monitoring** tab.
    This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched.
    You can click on a graph to see an expanded view.
    Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default. You can enable detailed (one-minute) monitoring.



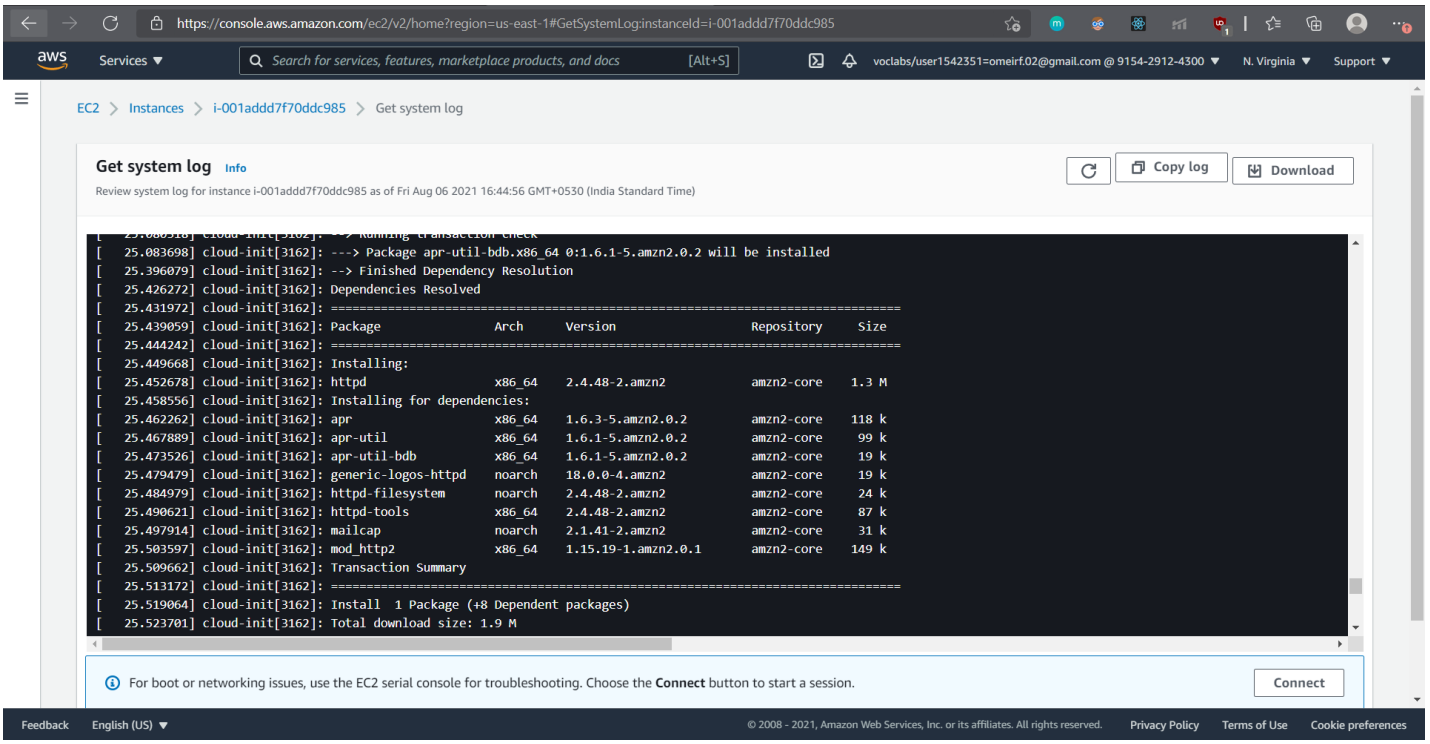


28. In the **Actions** menu, select **Monitor and troubleshoot Get System Log**.
    The System Log displays the console output of the instance, which is a valuable tool for problem diagnosis. It is especially useful for troubleshooting kernel problems and service configuration issues that could cause an instance to terminate or become unreachable before

its SSH daemon can be started. If you do not see a system log, wait a few minutes and then try again.

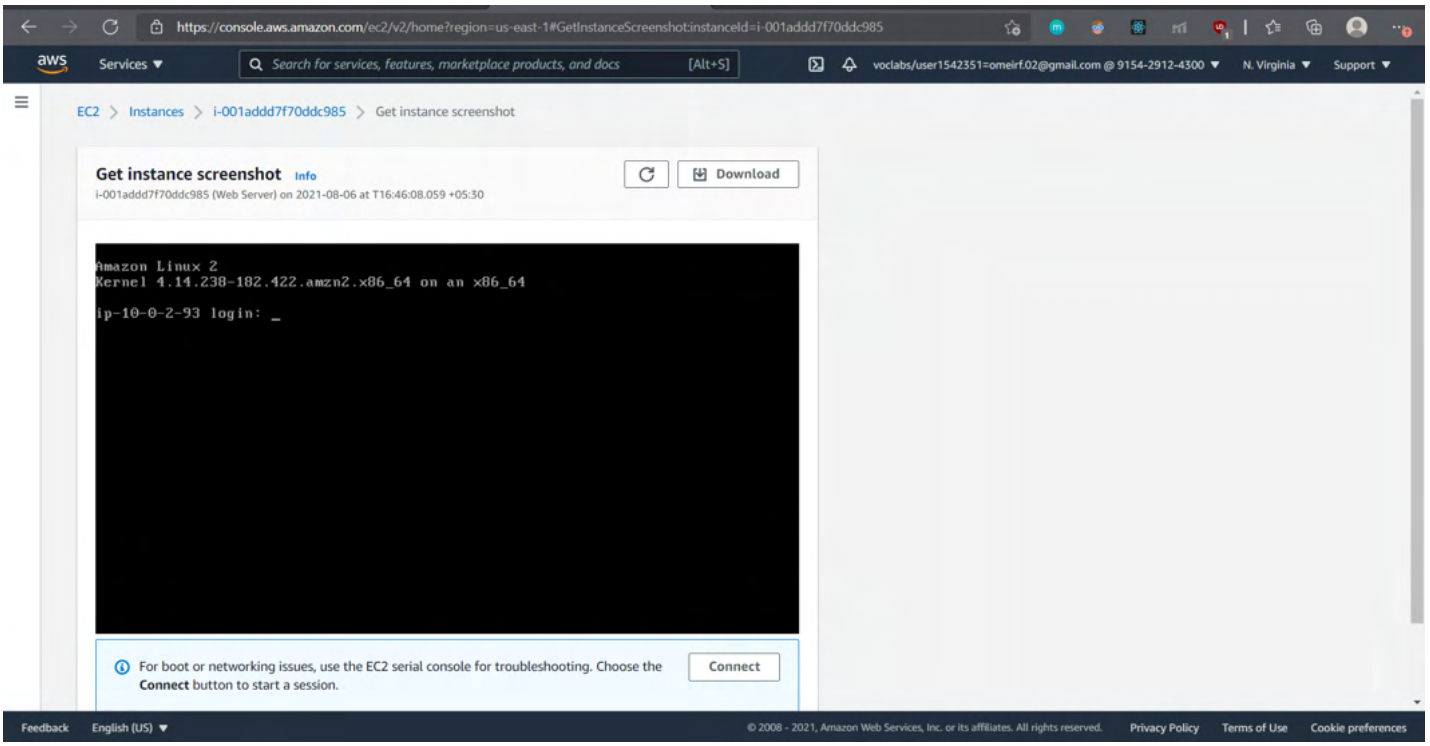


29. Scroll through the output and note that the HTTP package was installed from the **user data** that you added when you created the instance.



30. Choose **Cancel**.

31. In the **Actions** menu, select **Monitor and troubleshoot Get Instance Screenshot**.
    This shows you what your Amazon EC2 instance console would look like if a screen were attached to it.



If you are unable to reach your instance via SSH or RDP, you can capture a screenshot of your instance and view it as an image. This provides visibility as to the status of the instance, and allows for quicker troubleshooting.

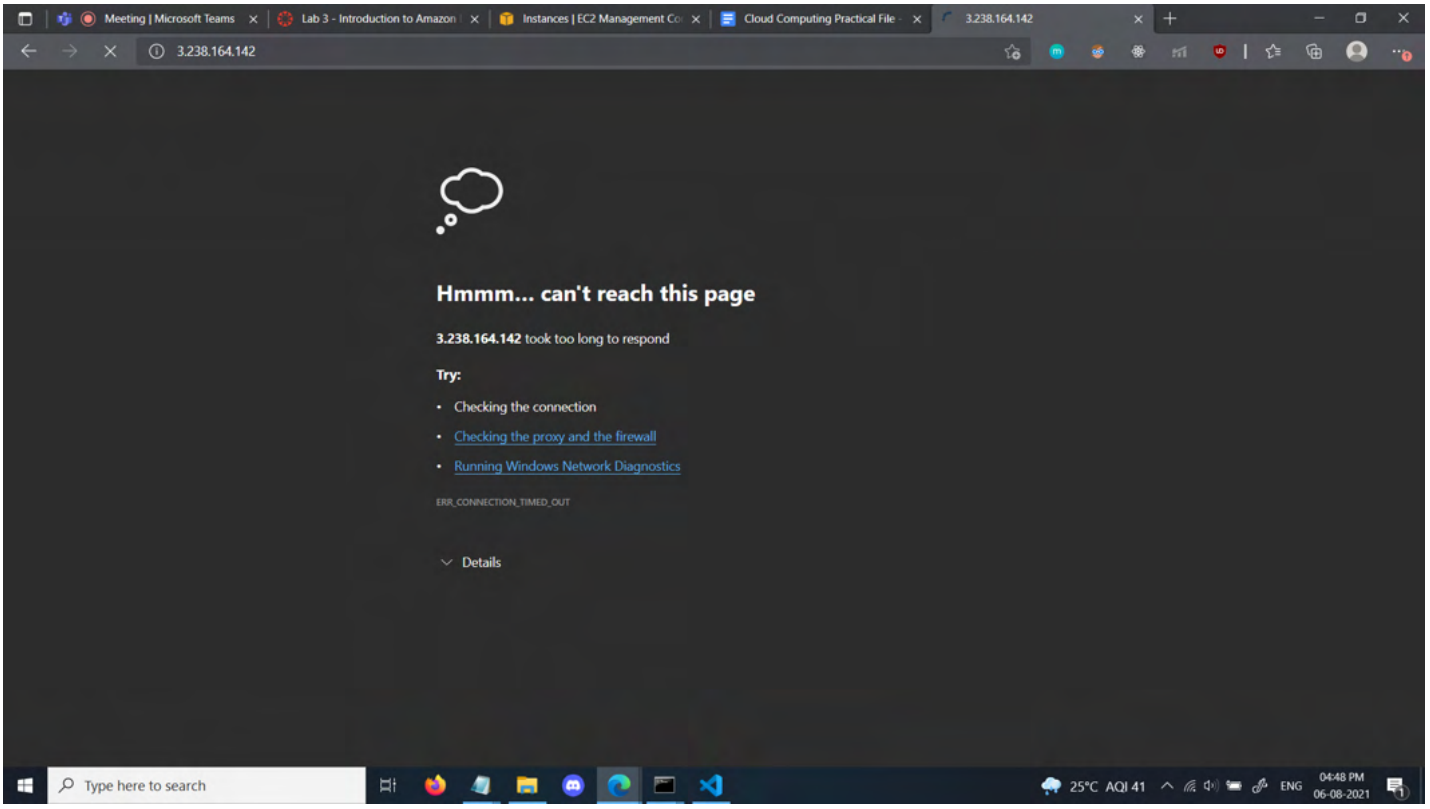
32. Choose **Cancel**.
    Congratulations! You have explored several ways to monitor your instance.

Task 3: Update Your Security Group and Access the Web Server

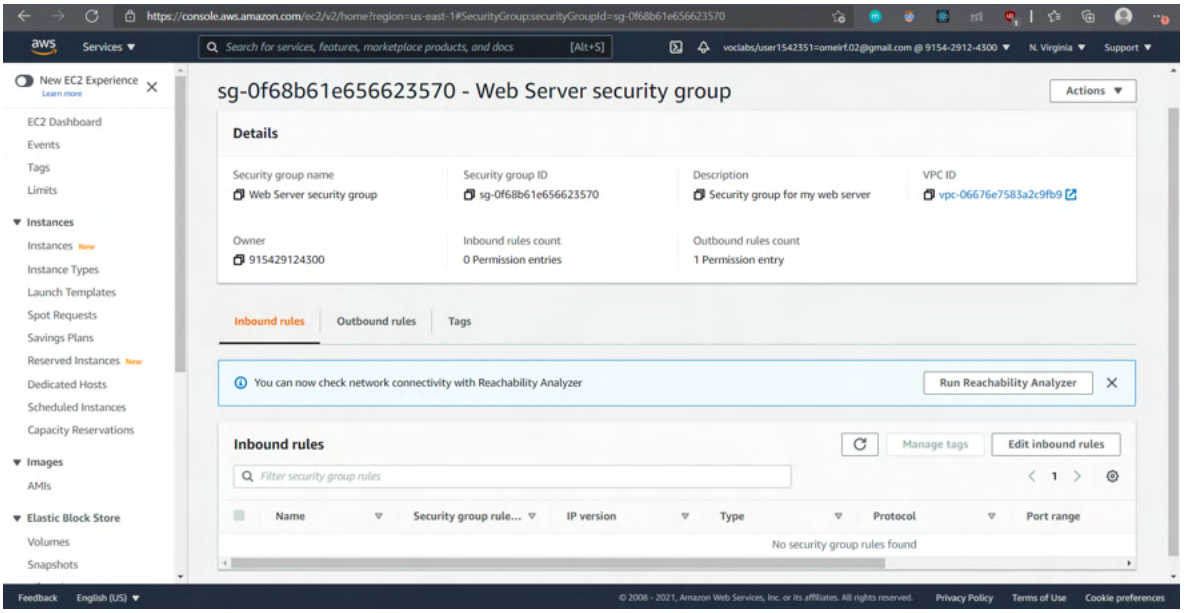
When you launched the EC2 instance, you provided a script that installed a web server and created a simple web page. In this task, you will access content from the web server.

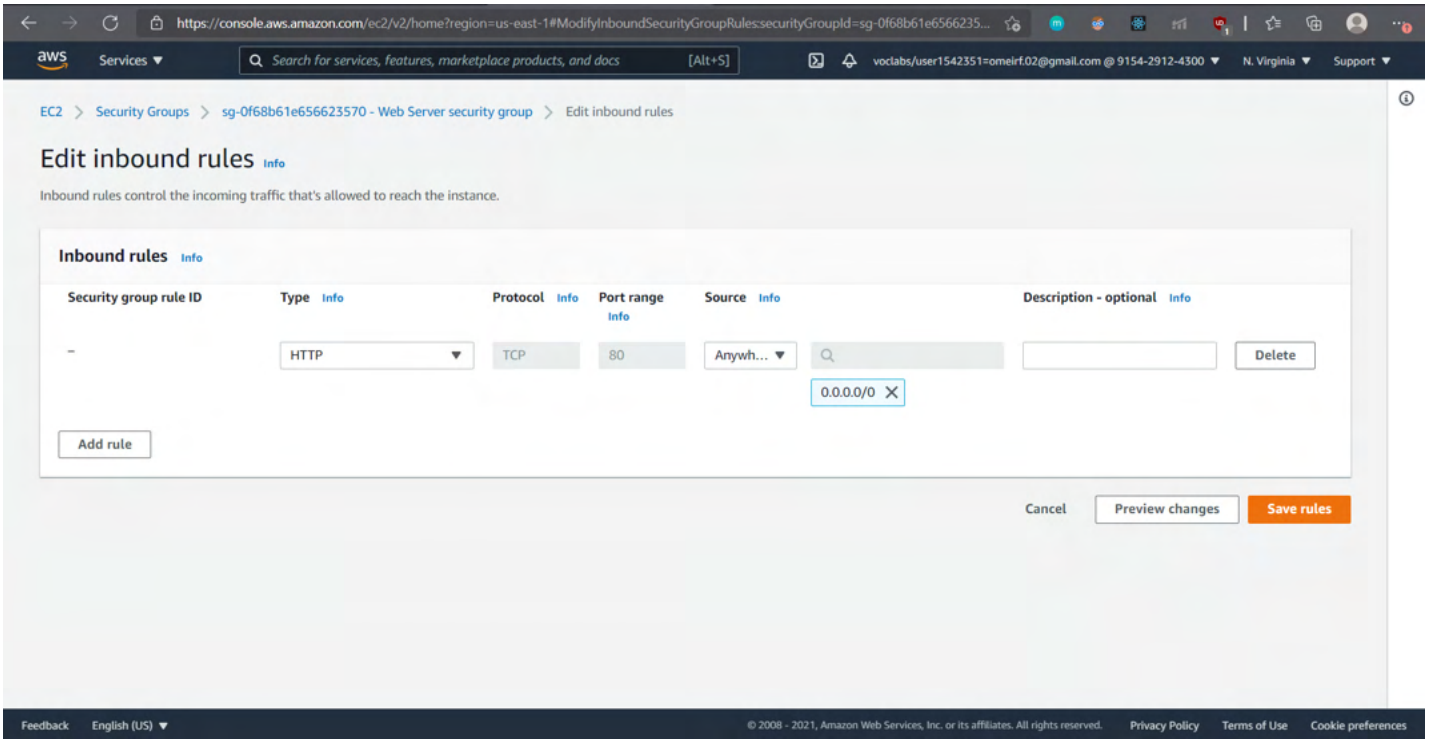
33. Click the **Details** tab.
34. Copy the **IPv4 Public IP** of your instance to your clipboard.
35. Open a new tab in your web browser, paste the IP address you just copied, then press **Enter**.
    Question: Are you able to access your web server? Why not?
    You are **not** currently able to access your web server because the *security group* is not permitting inbound traffic on port 80, which is used for HTTP web requests. This is a demonstration of using a security group as a firewall to restrict the network traffic that is

allowed in and out of an instance.
To correct this, you will now update the security group to permit web traffic on port 80.

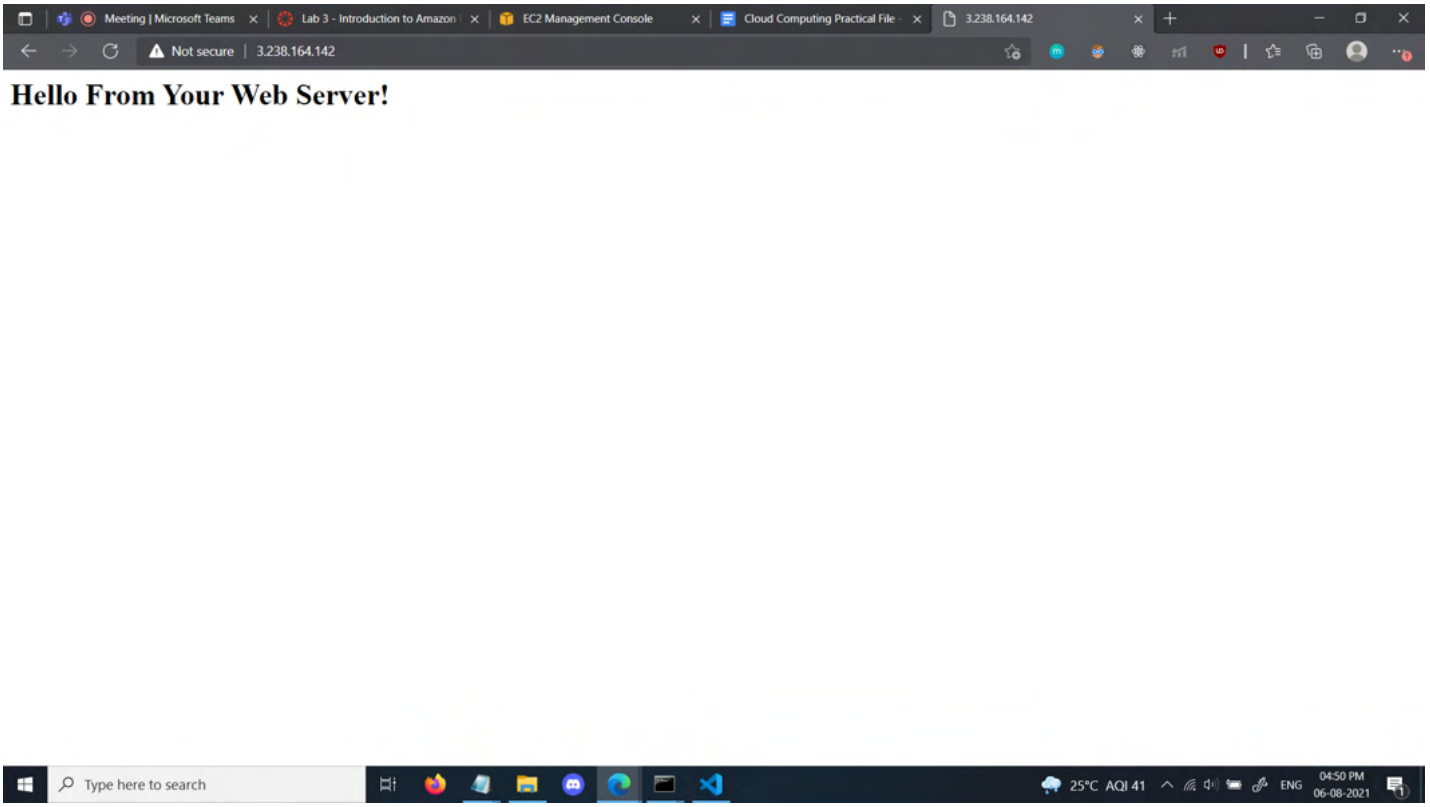


36. Keep the browser tab open, but return to the **EC2 Management Console** tab.
37. In the left navigation pane, click **Security Groups**.
38. Select **Web Server security group**.
39. Click the **Inbound** tab.
    The security group currently has no rules.
40. Click **Edit inboud rules** then configure:
    - **Type:** *HTTP*
    - **Source:** *Anywhere*
    - Click **Save rules**





41. Return to the web server tab that you previously opened and refresh the page.
You should see the message *Hello From Your Web Server!*
Congratulations! You have successfully modified your security group to permit HTTP traffic into your Amazon EC2 Instance.



Task 4: Resize Your Instance: Instance Type and EBS Volume

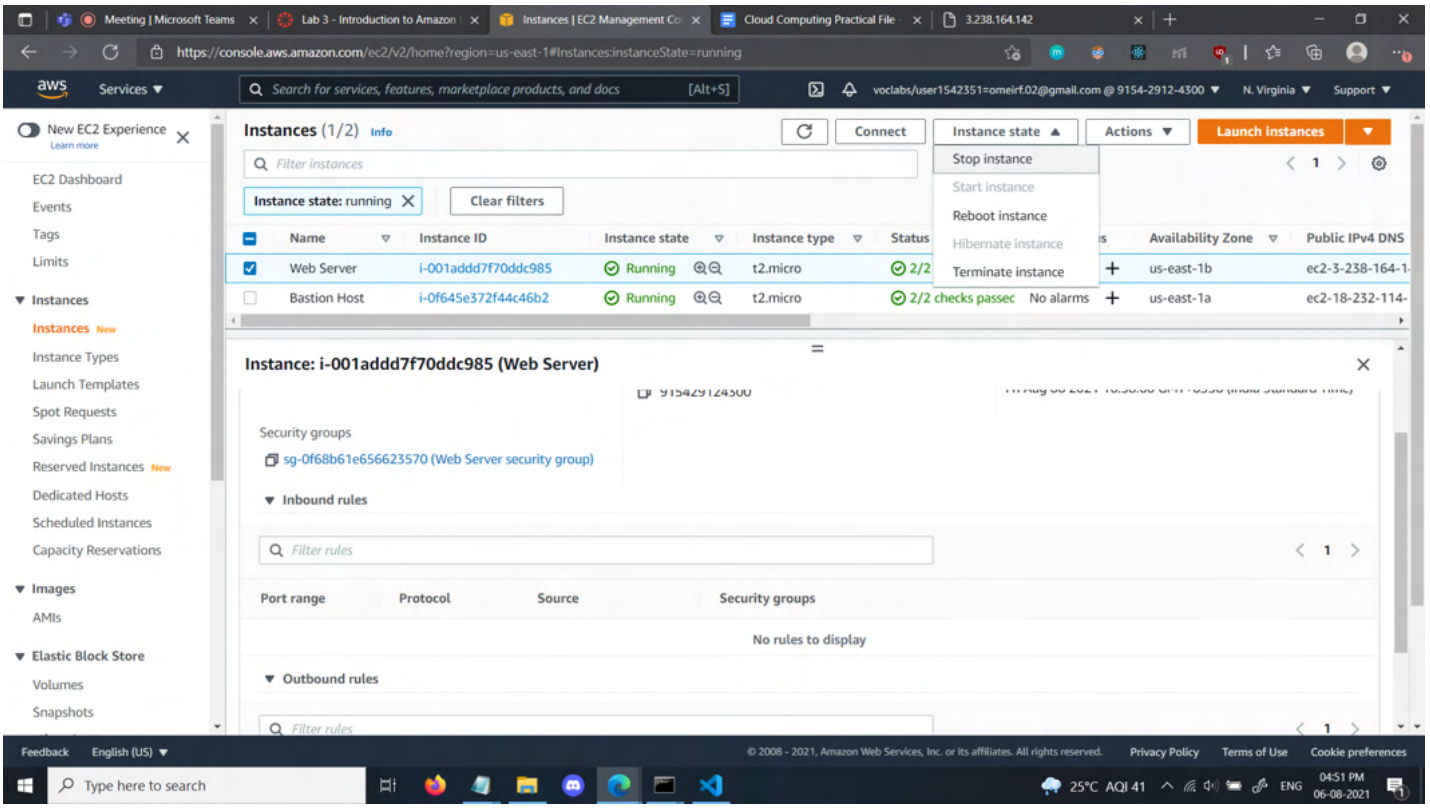
As your needs change, you might find that your instance is over-utilized (too small) or under-utilized (too large). If so, you can change the *instance type*. For example, if a *t2.micro* instance is too small for its workload, you can change it to an *m5.medium* instance. Similarly, you can change the size of a disk.

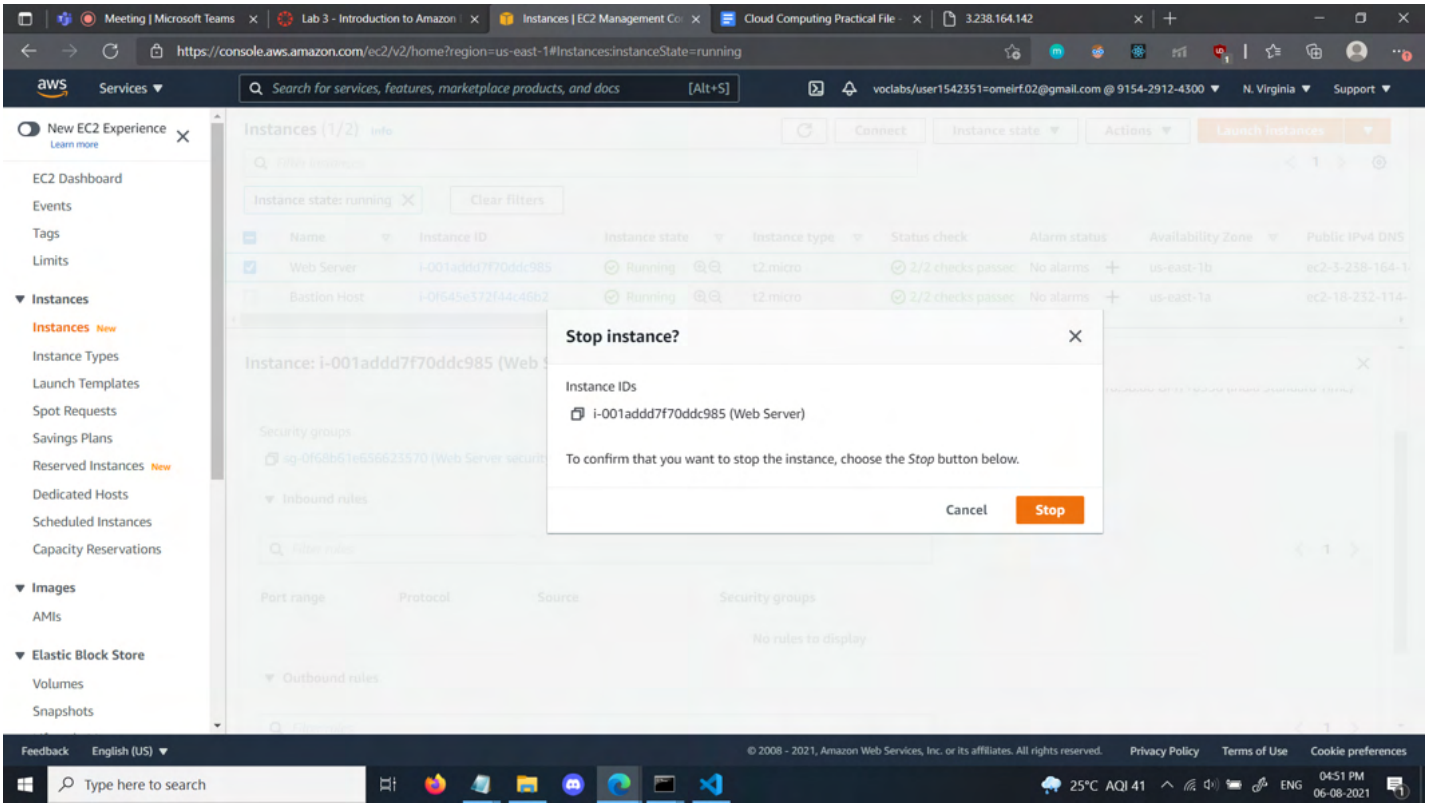
Stop Your Instance

Before you can resize an instance, you must *stop* it.

When you stop an instance, it is shut down. There is no charge for a stopped EC2 instance, but the storage charge for attached Amazon EBS volumes remains.

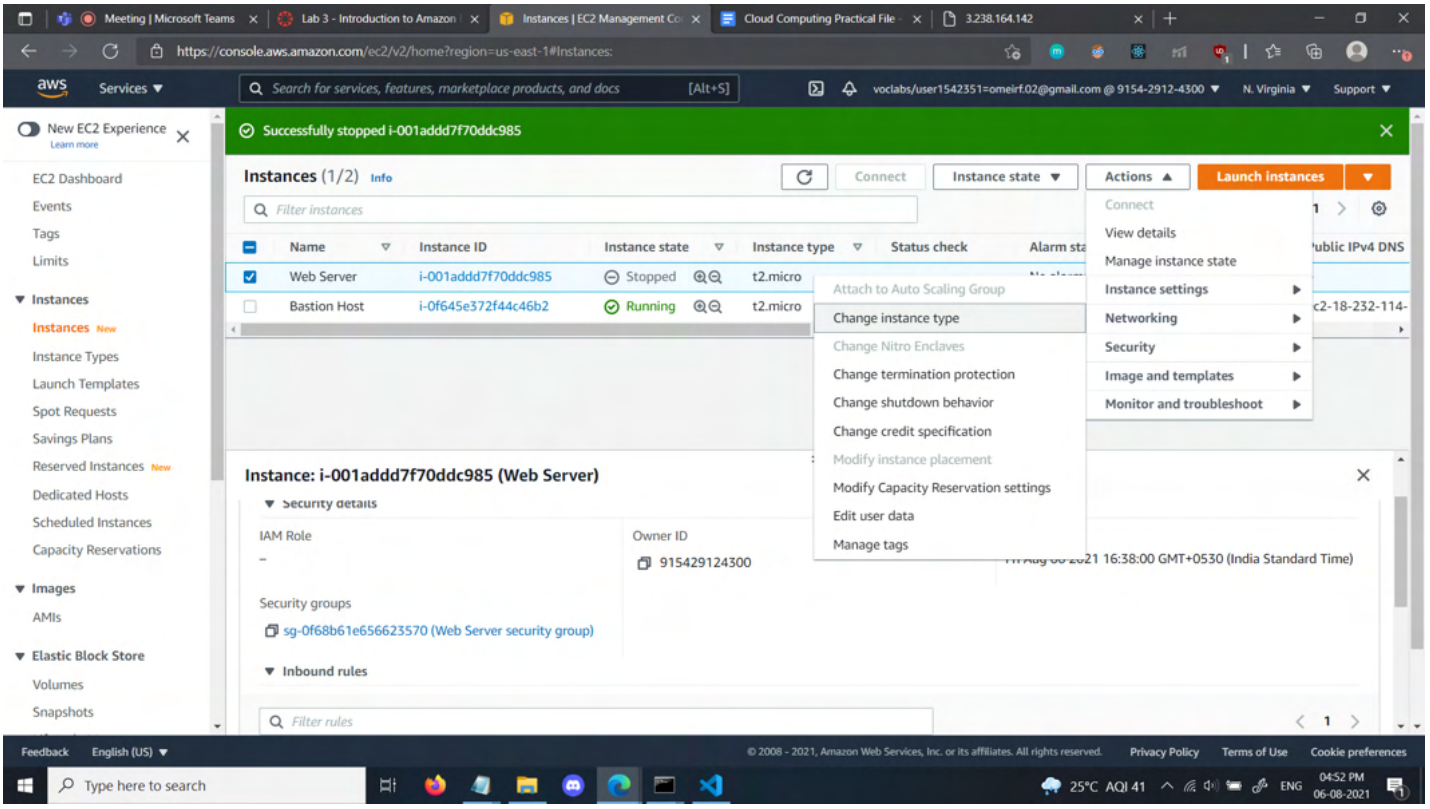
42. On the **EC2 Management Console**, in the left navigation pane, click **Instances**.
    Web Server should already be selected.
43. In the **Instance State** menu, select **Stop instance**.
44. Choose **Stop**
    Your instance will perform a normal shutdown and then will stop running.
45. Wait for the **Instance State** to display: stopped

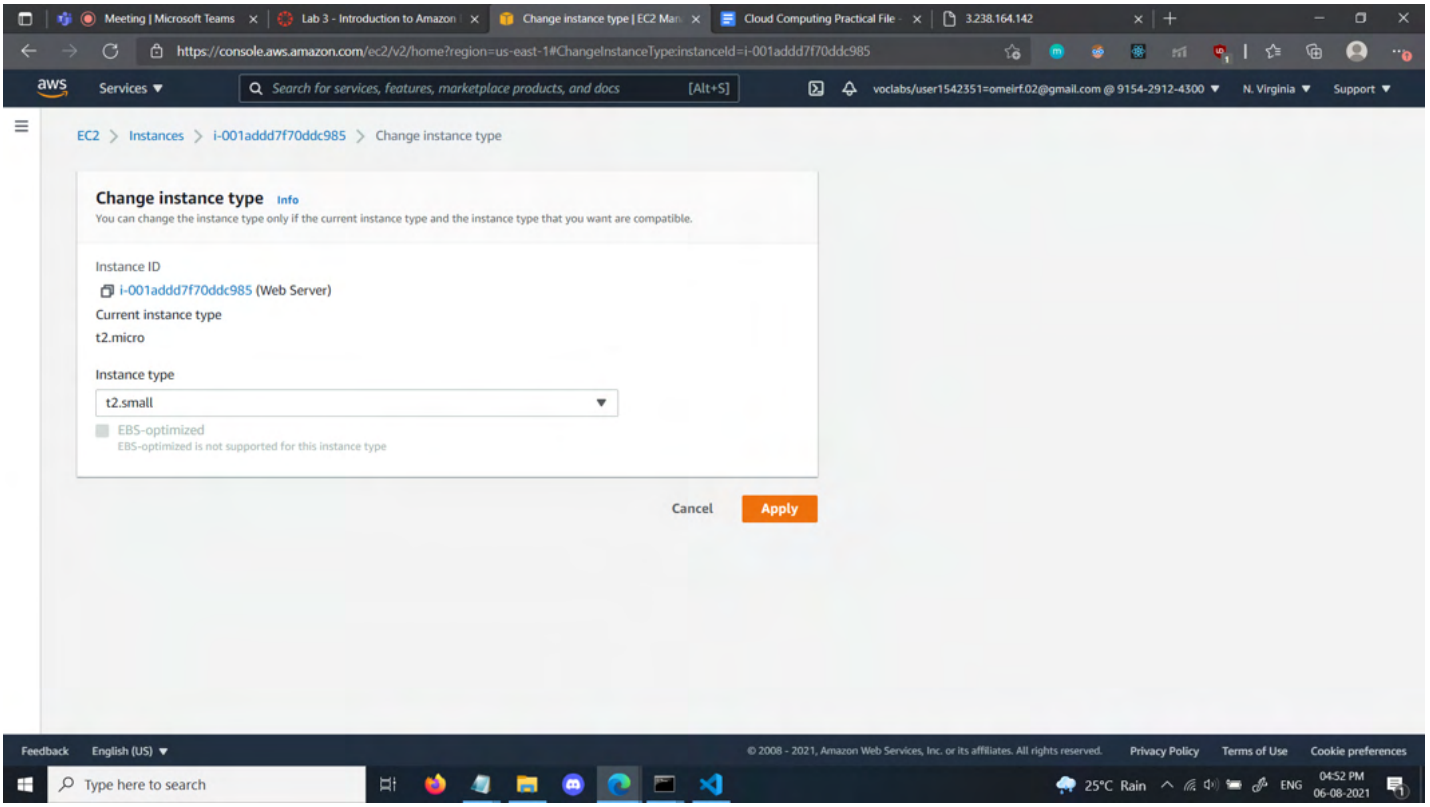




Change The Instance Type

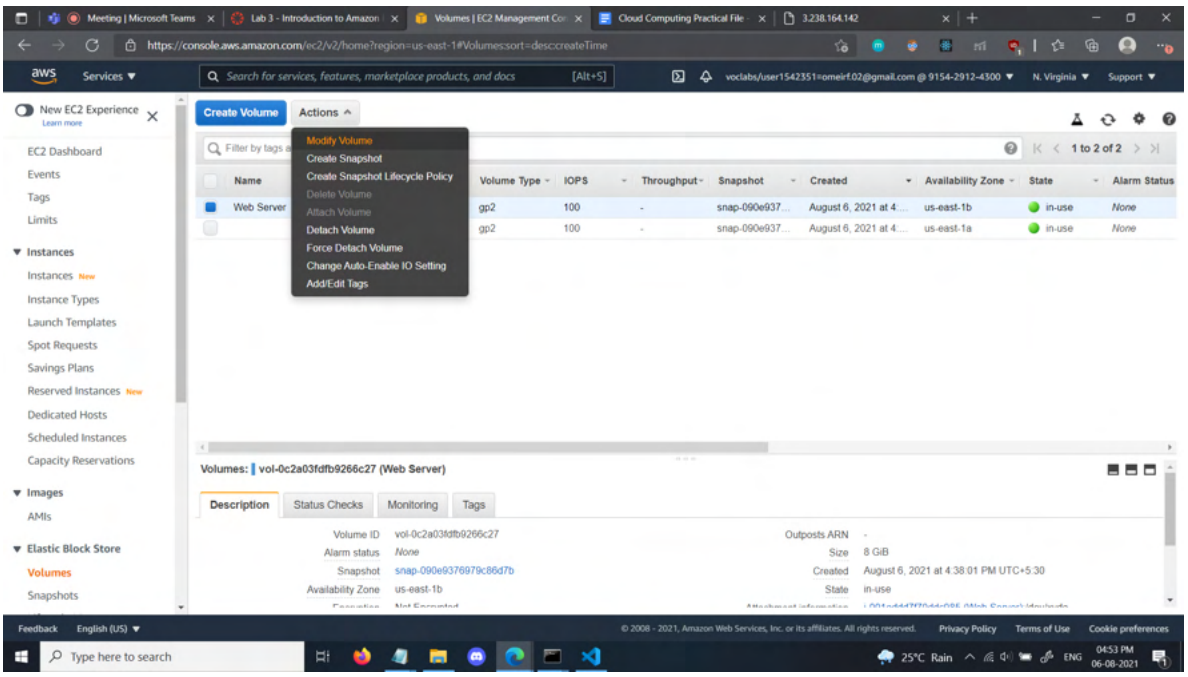
46. In the **Actions** menu, select **Instance Settings** **Change Instance Type**, then configure:
    - **Instance Type:** *t2.small*
    - Choose **Apply**
47. When the instance is started again it will be a *t2.small*, which has twice as much memory as a *t2.micro* instance. **NOTE**: You may be restricted from using other instance types in this lab.

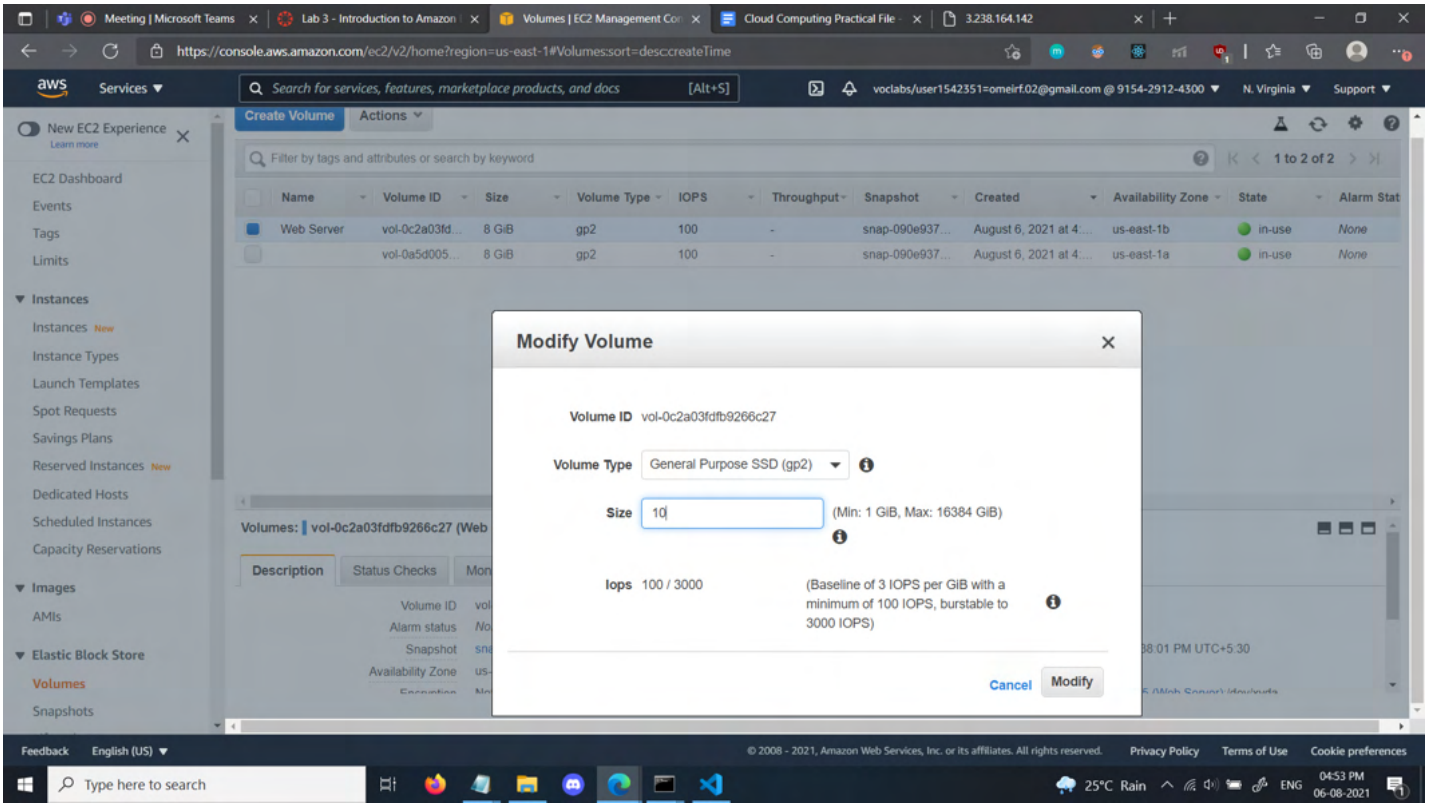


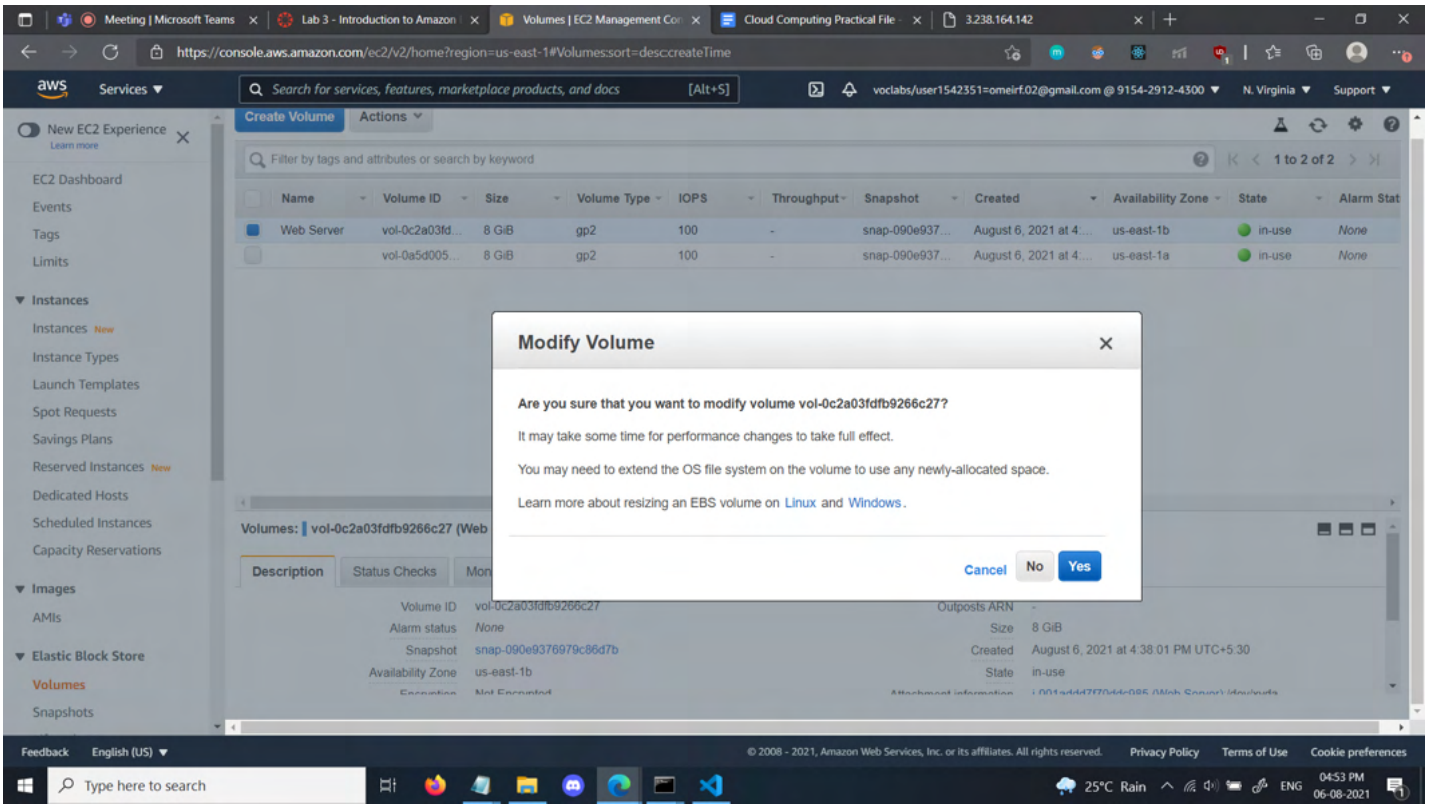


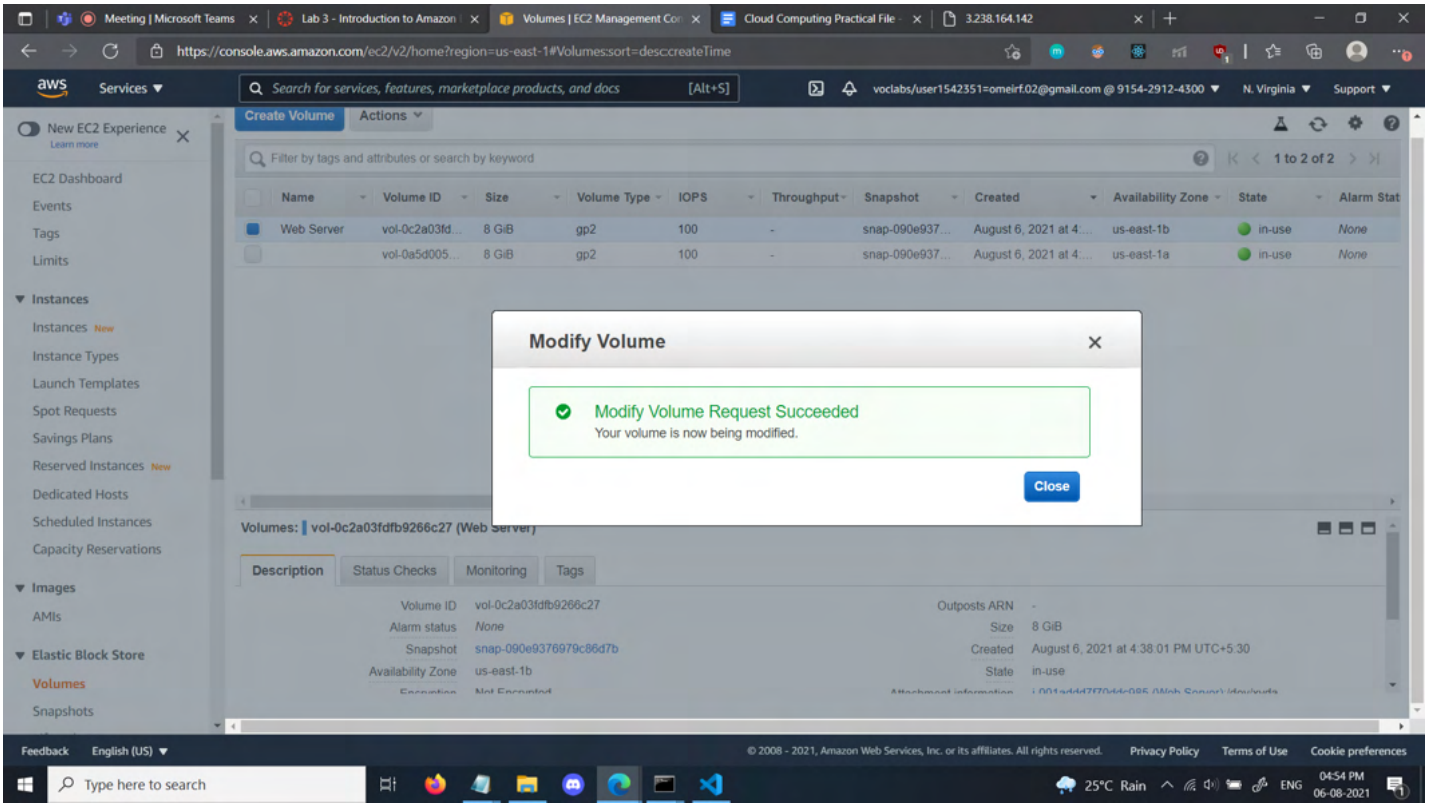
Resize the EBS Volume

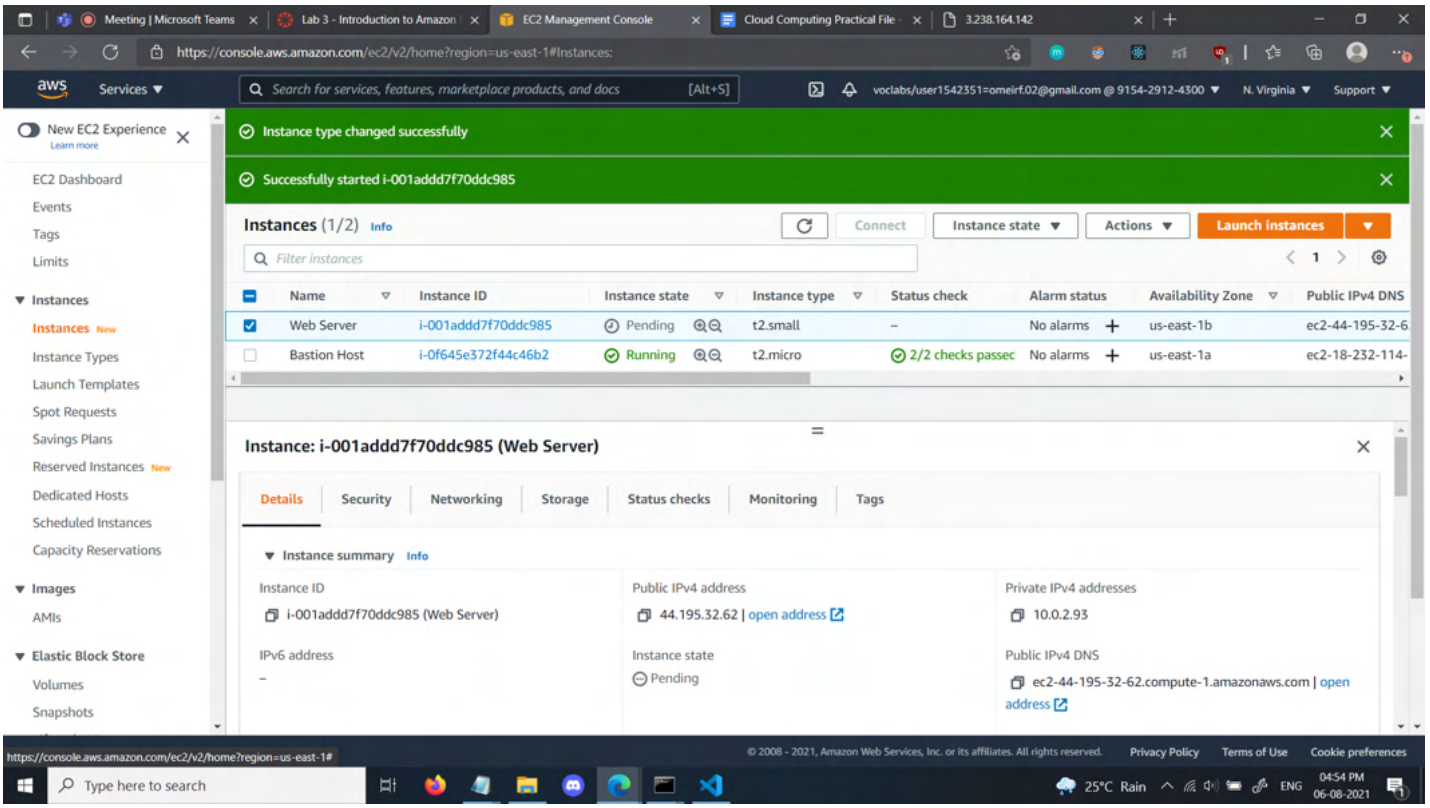
47. In the left navigation menu, click **Volumes**.
48. In the **Actions** menu, select **Modify Volume**.
    The disk volume currently has a size of 8 GiB. You will now increase the size of this disk.
49. Change the size to: 10 **NOTE**: You may be restricted from creating large Amazon EBS volumes in this lab.
50. Choose **Modify**
51. Choose **Yes** to confirm and increase the size of the volume.
52. Choose **Close**




Modify Volume
Volume ID vol-0c2a03fdfb9266c27
Volume Type General Purpose SSD (gp2)
Size 10
(Min: 1 GiB, Max: 16384 GiB)
Iops 100 / 3000
(Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
Cancel
Modify


Modify Volume
Are you sure that you want to modify volume vol-0c2a03fdfb9266c27?
It may take some time for performance changes to take full effect.
You may need to extend the OS file system on the volume to use any newly-allocated space.
Learn more about resizing an EBS volume on Linux and Windows.
Cancel
No
Yes





Start the Resized Instance

You will now start the instance again, which will now have more memory and more disk space.

53. In left navigation pane, click **Instances**.
54. In the **Instance State** menu, select **Start instance**.
55. Choose **Start**

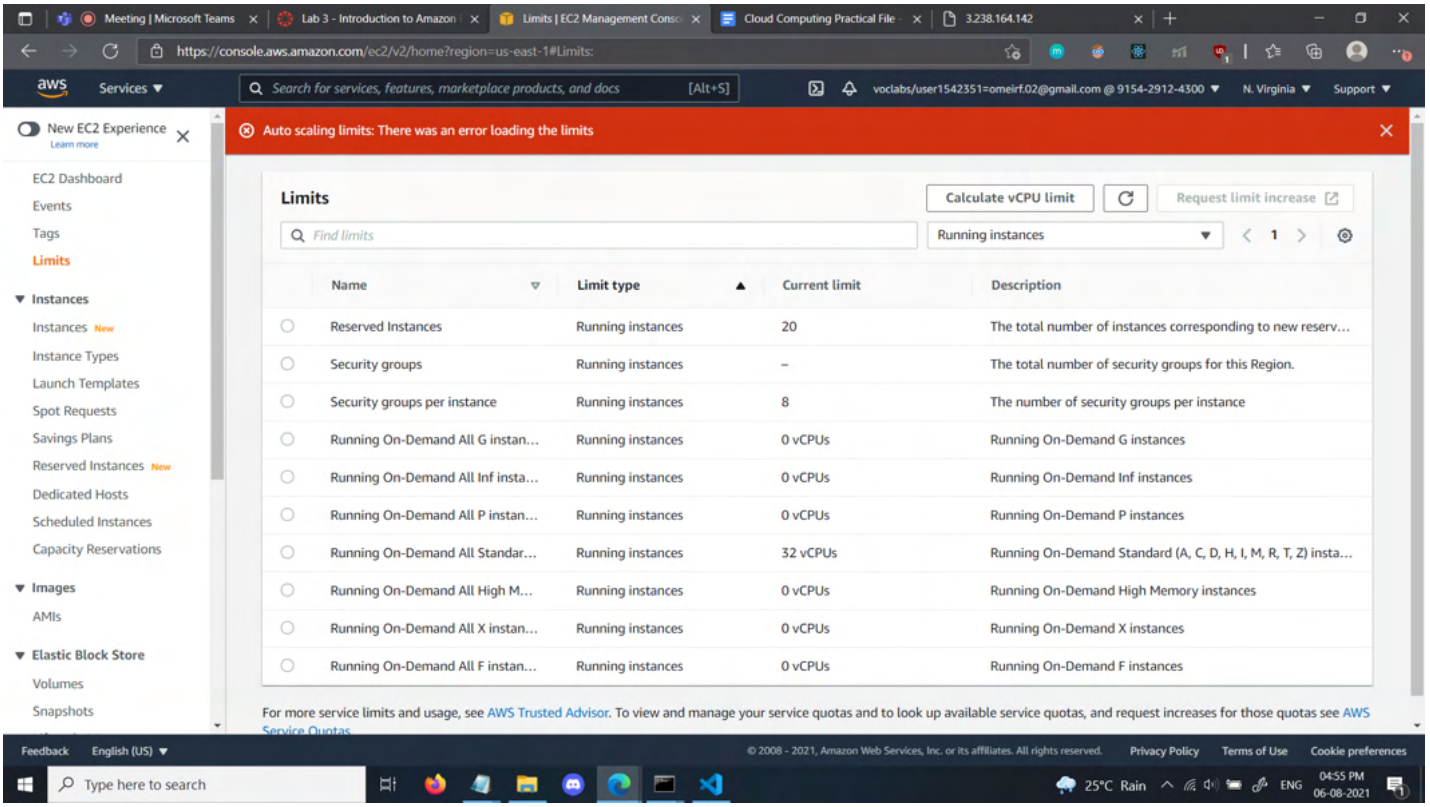
Congratulations! You have successfully resized your Amazon EC2 Instance. In this task

you changed your instance type from *t2.micro* to *t2.small*. You also modified your root disk volume from 8 GiB to 10 GiB.

Task 5: Explore EC2 Limits

Amazon EC2 provides different resources that you can use. These resources include images, instances, volumes, and snapshots. When you create an AWS account, there are default limits on these resources on a per-region basis.

56. In the left navigation pane, click **Limits**.



57. From the drop down list, choose **Running instances**.
    Note that there is a limit on the number of instances that you can launch in this region. When launching an instance, the request must not cause your usage to exceed the current instance limit in that region.
    You can request an increase for many of these limits.

Task 6: Test Termination Protection

You can delete your instance when you no longer need it. This is referred to as *terminating* your instance. You cannot connect to or restart an instance after it has been terminated.

In this task, you will learn how to use *termination protection*.

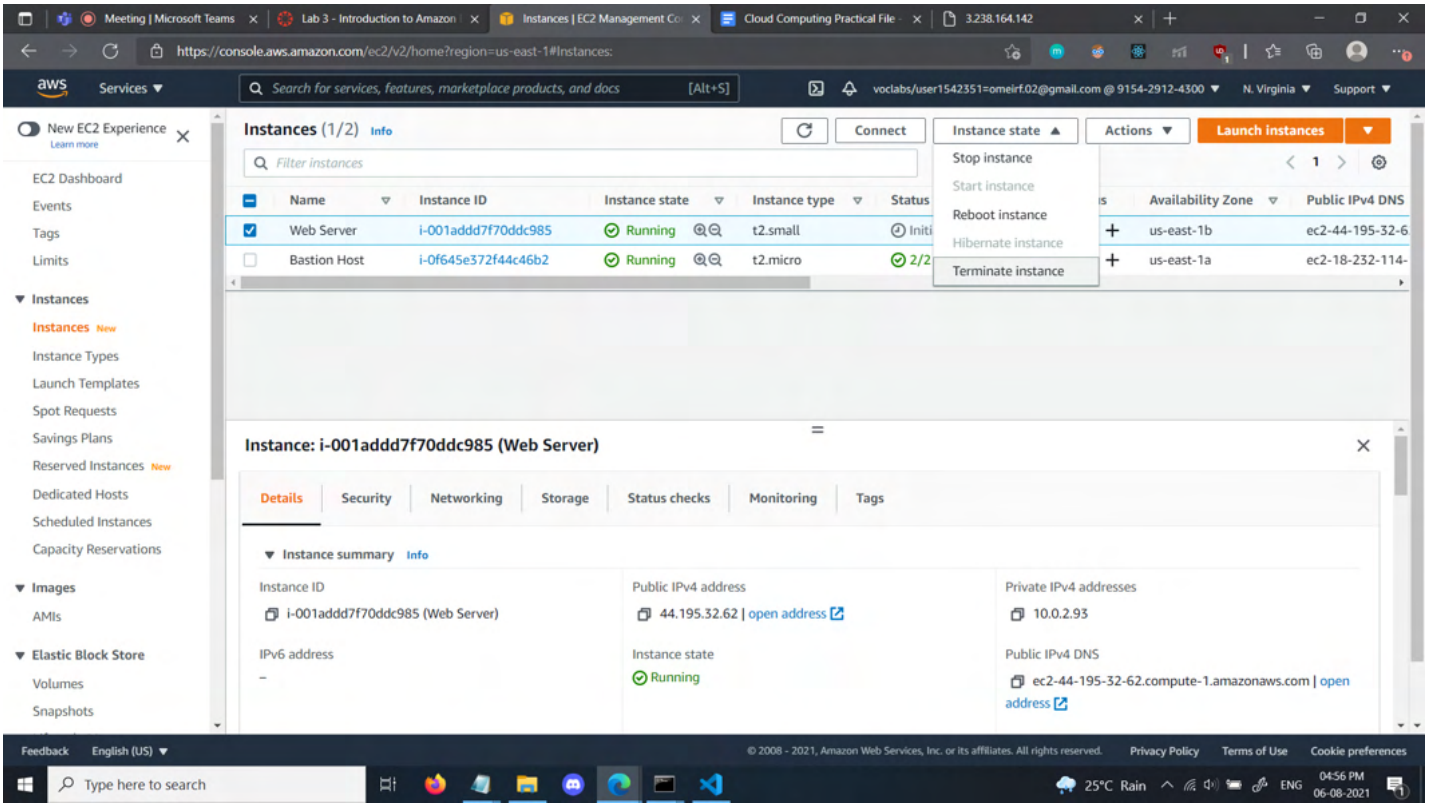
58. In left navigation pane, click **Instances**.
59. In the **Instance State** menu, select **Terminate instance**.
60. Then choose **Terminate**

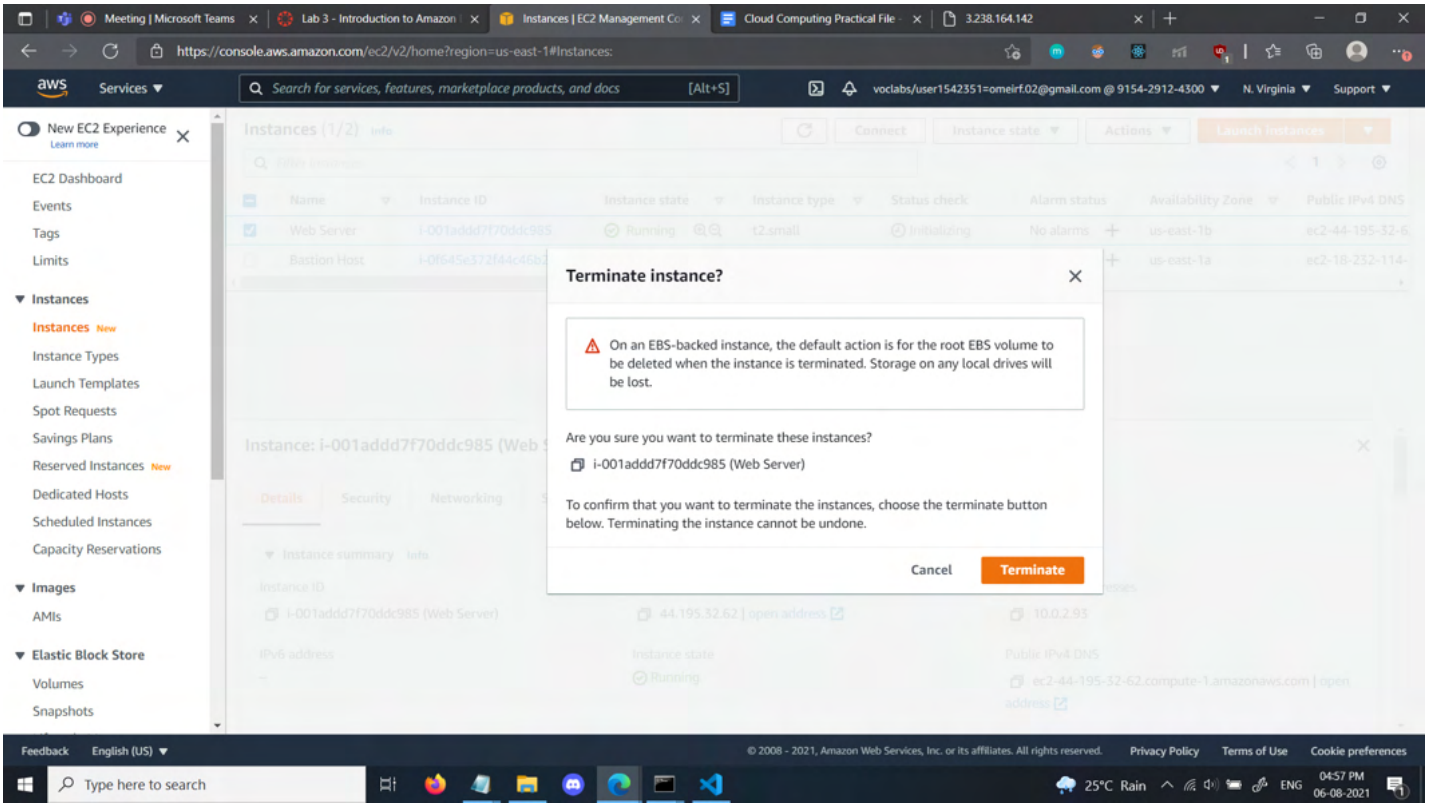
    Note that there is a message that says: *Failed to terminate the instance i-1234567xxx. The instance 'i-1234567xxx' may not be terminated. Modify its 'disableApiTermination' instance attribute and try again.*

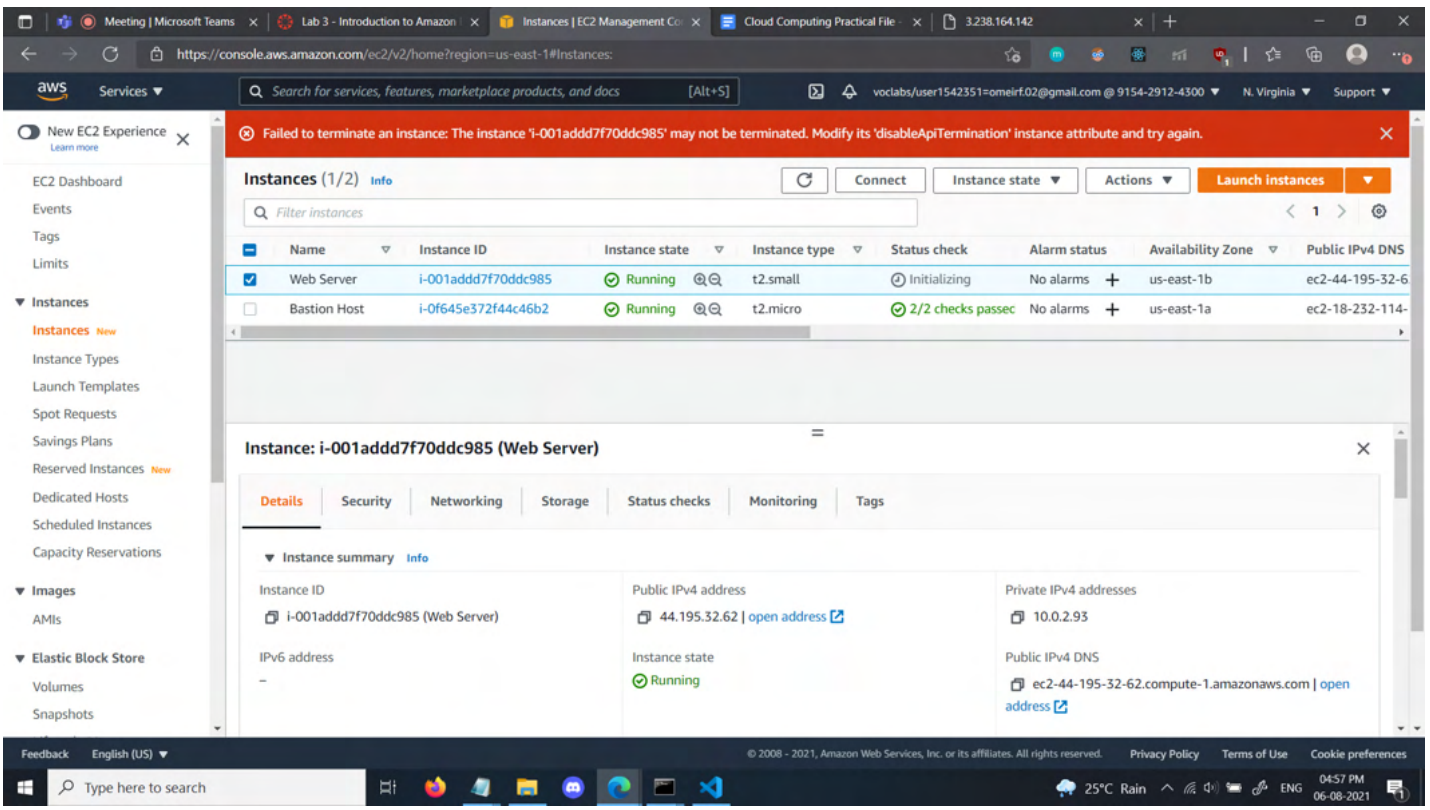
    This is a safeguard to prevent the accidental termination of an instance. If you really want to terminate the instance, you will need to disable the termination protection.
61. In the **Actions** menu, select **Instance Settings Change Termination Protection**.
62. Remove the check next to **Enable**.
63. Choose **Save**

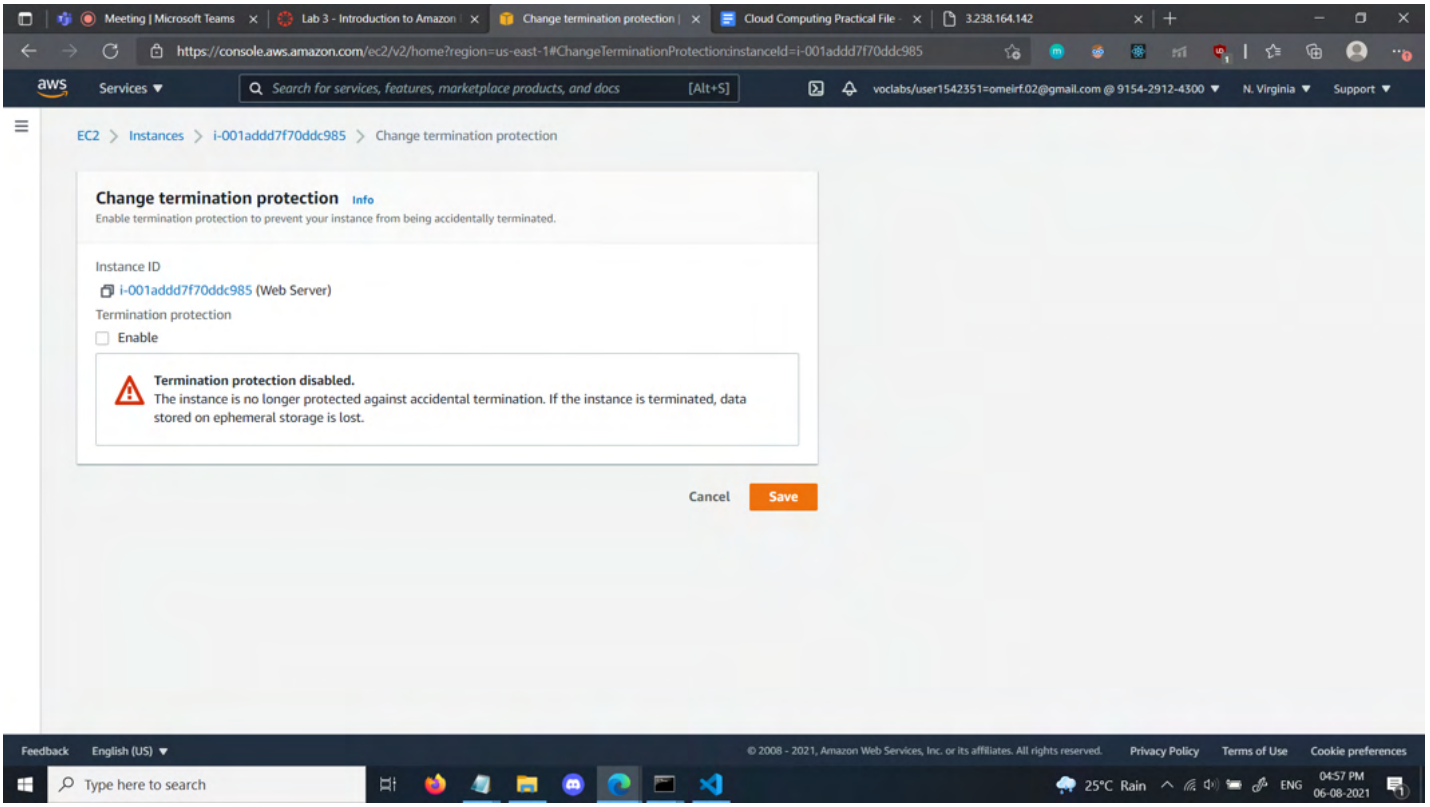
    You can now terminate the instance.
64. In the **Instance State** menu, select **Terminate**.
65. Choose **Terminate**

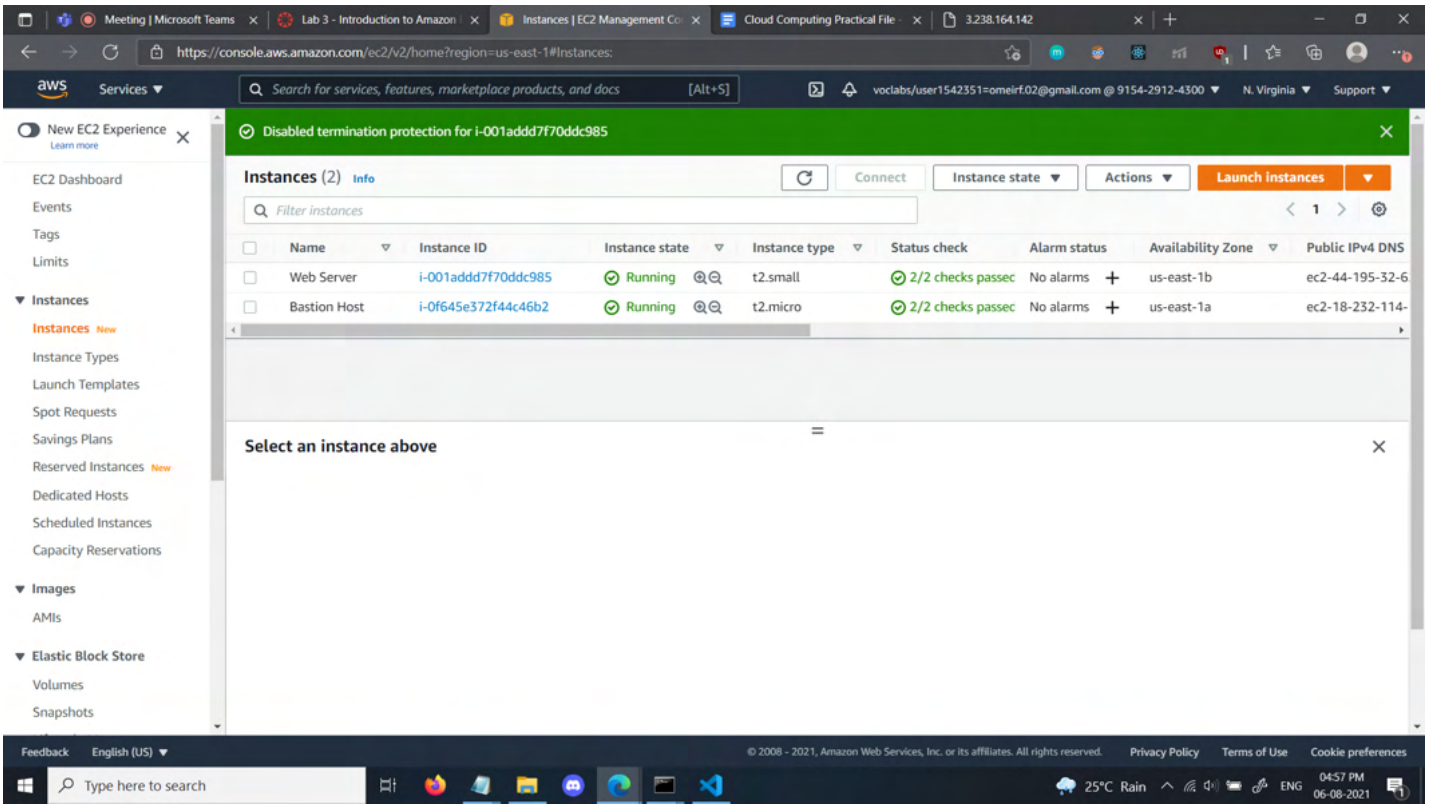
    Congratulations! You have successfully tested termination protection and terminated your instance.

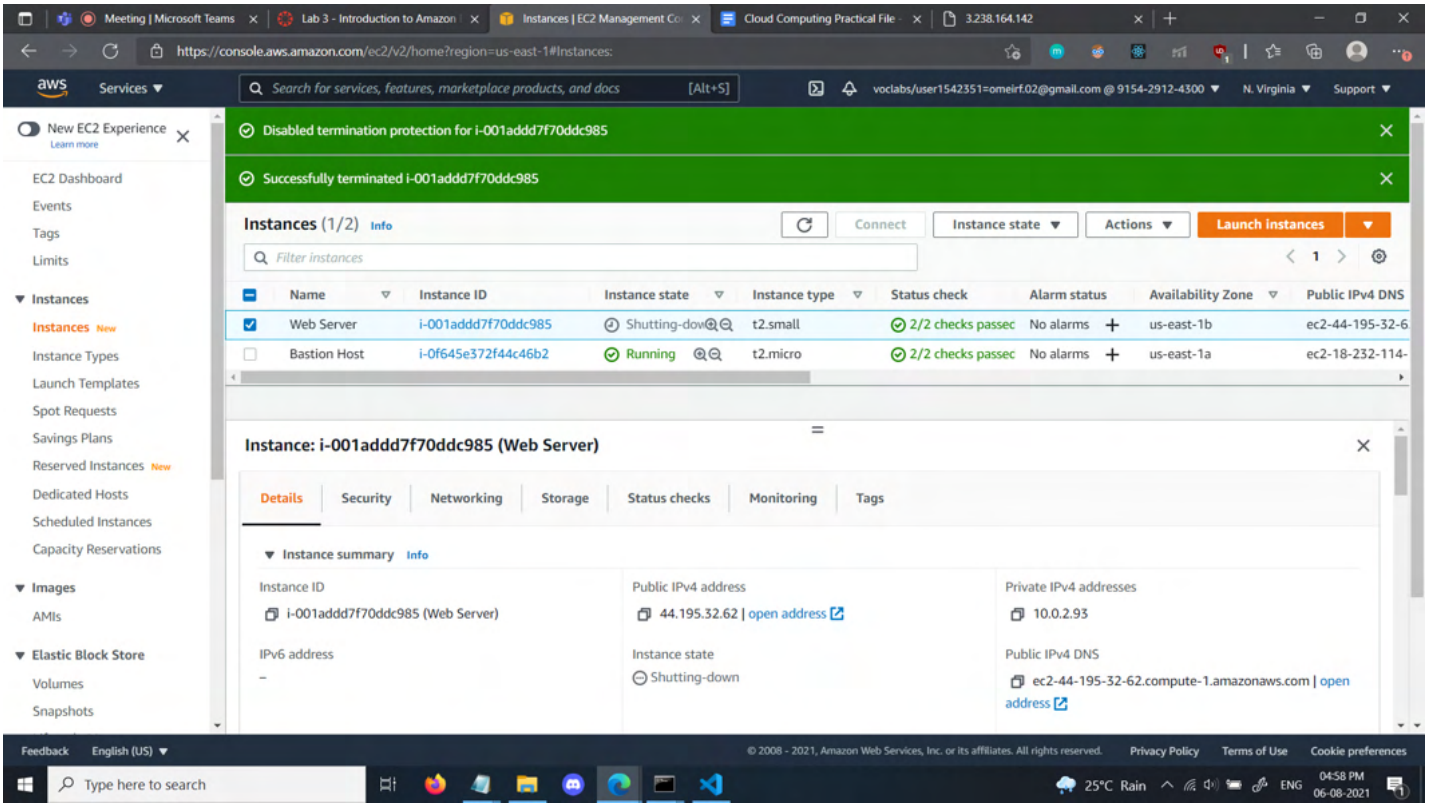


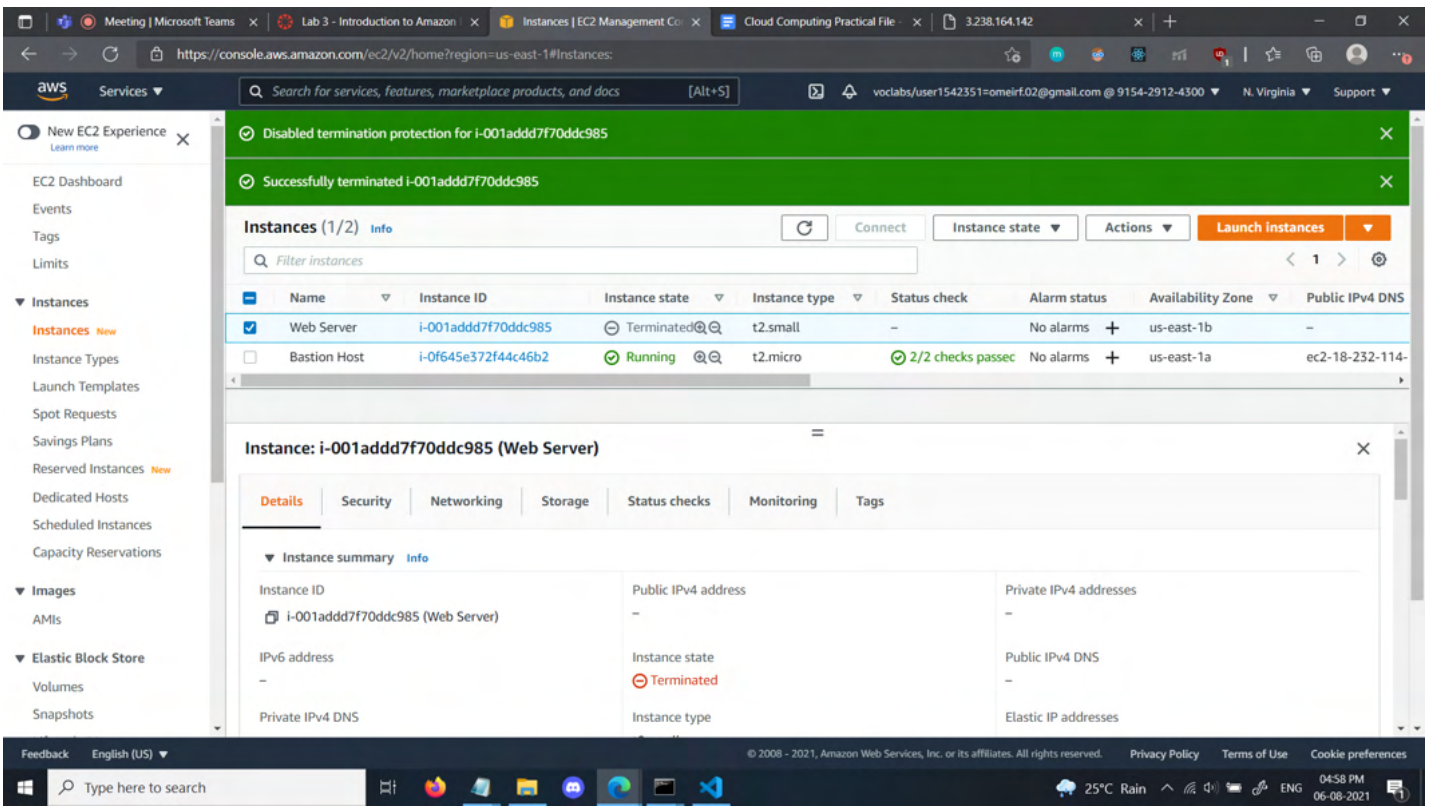

Terminate instance?
On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated. Storage on any local drives will be lost.
Are you sure you want to terminate these instances?
i-001addd7f70ddc985 (Web Server)
To confirm that you want to terminate the instances, choose the terminate button below. Terminating the instance cannot be undone.
Cancel
Terminate


Failed to terminate an instance: The instance 'i-001addd7f70ddc985' may not be terminated. Modify its 'disableApiTermination' instance attribute and try again.
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Web Server
i-001addd7f70ddc985
Running
t2.small
Initializing
No alarms
us-east-1b
ec2-44-195-32-6
Bastion Host
i-0f645e372f44c46b2
Running
t2.micro
2/2 checks passec
No alarms
us-east-1a
ec2-18-232-114-
Instance: i-001addd7f70ddc985 (Web Server)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-001addd7f70ddc985 (Web Server)
Public IPv4 address
44.195.32.62 | open address
Private IPv4 addresses
10.0.2.93
IPv6 address
Instance state
Running
Public IPv4 DNS
ec2-44-195-32-62.compute-1.amazonaws.com | open address


Meeting | Microsoft Teams
Lab 3 - Introduction to Amazon
Change termination protection
Cloud Computing Practical File -
3.238.164.142
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#ChangeTerminationProtection:instanceId=i-001addd7f70ddc985
Services
Search for services, features, marketplace products, and docs [Alt+S]
voclabs/user1542351=omeirf.02@gmail.com @ 9154-2912-4300
N. Virginia
Support
EC2 > Instances > i-001addd7f70ddc985 > Change termination protection
Change termination protection Info
Enable termination protection to prevent your instance from being accidentally terminated.
Instance ID
i-001addd7f70ddc985 (Web Server)
Termination protection
Enable
Termination protection disabled.
The instance is no longer protected against accidental termination. If the instance is terminated, data stored on ephemeral storage is lost.
Cancel
Save
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
Type here to search
25°C Rain
ENG
04:57 PM
06-08-2021


Meeting | Microsoft Teams
Lab 3 - Introduction to Amazon
Instances | EC2 Management Co
Cloud Computing Practical File -
3.238.164.142
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
Services
Search for services, features, marketplace products, and docs [Alt+S]
voclabs/user1542351=omeirf.02@gmail.com @ 9154-2912-4300
N. Virginia
Support
New EC2 Experience
Learn more
EC2 Dashboard
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Disabled termination protection for i-001addd7f70ddc985
Instances (2) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Web Server
i-001addd7f70ddc985
Running
t2.small
2/2 checks passec
No alarms
us-east-1b
ec2-44-195-32-6
Bastion Host
i-0f645e372f44c46b2
Running
t2.micro
2/2 checks passec
No alarms
us-east-1a
ec2-18-232-114-
Select an instance above
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
Type here to search
25°C Rain
ENG
04:57 PM
06-08-2021


Meeting | Microsoft Teams
Lab 3 - Introduction to Amazon
Instances | EC2 Management Co
Cloud Computing Practical File -
3.238.164.142
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
Services
Search for services, features, marketplace products, and docs [Alt+S]
voclabs/user1542351=omeirf.02@gmail.com @ 9154-2912-4300
N. Virginia
Support
New EC2 Experience
Learn more
EC2 Dashboard
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Disabled termination protection for i-001addd7f70ddc985
Successfully terminated i-001addd7f70ddc985
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Web Server
i-001addd7f70ddc985
Shutting-dow
t2.small
2/2 checks passec
No alarms
us-east-1b
ec2-44-195-32-6
Bastion Host
i-0f645e372f44c46b2
Running
t2.micro
2/2 checks passec
No alarms
us-east-1a
ec2-18-232-114-
Instance: i-001addd7f70ddc985 (Web Server)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-001addd7f70ddc985 (Web Server)
Public IPv4 address
44.195.32.62 | open address
Private IPv4 addresses
10.0.2.93
IPv6 address
Instance state
Shutting-down
Public IPv4 DNS
ec2-44-195-32-62.compute-1.amazonaws.com | open address
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
Type here to search
25°C Rain
ENG
04:58 PM
06-08-2021


Meeting | Microsoft Teams
Lab 3 - Introduction to Amazon
Instances | EC2 Management Co
Cloud Computing Practical File -
3.238.164.142
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
Services
Search for services, features, marketplace products, and docs [Alt+S]
voclabs/user1542351=omeirf.02@gmail.com @ 9154-2912-4300
N. Virginia
Support
New EC2 Experience
Learn more
EC2 Dashboard
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Disabled termination protection for i-001addd7f70ddc985
Successfully terminated i-001addd7f70ddc985
Instances (1/2) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Web Server
i-001addd7f70ddc985
Terminated
t2.small
No alarms
us-east-1b
Bastion Host
i-0f645e372f44c46b2
Running
t2.micro
2/2 checks passec
No alarms
us-east-1a
ec2-18-232-114-
Instance: i-001addd7f70ddc985 (Web Server)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-001addd7f70ddc985 (Web Server)
Public IPv4 address
Private IPv4 addresses
IPv6 address
Instance state
Terminated
Public IPv4 DNS
Private IPv4 DNS
Instance type
Elastic IP addresses
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences
Type here to search
25°C Rain
ENG
04:58 PM
06-08-2021

Practical 2

AIM
Build your VPC and Launch a Web Server

THEORY

OBJECTIVE
- Create a VPC.
- Create subnets.
- Configure a security group.
- Launch an EC2 instance into a VPC.

PROCEDURE

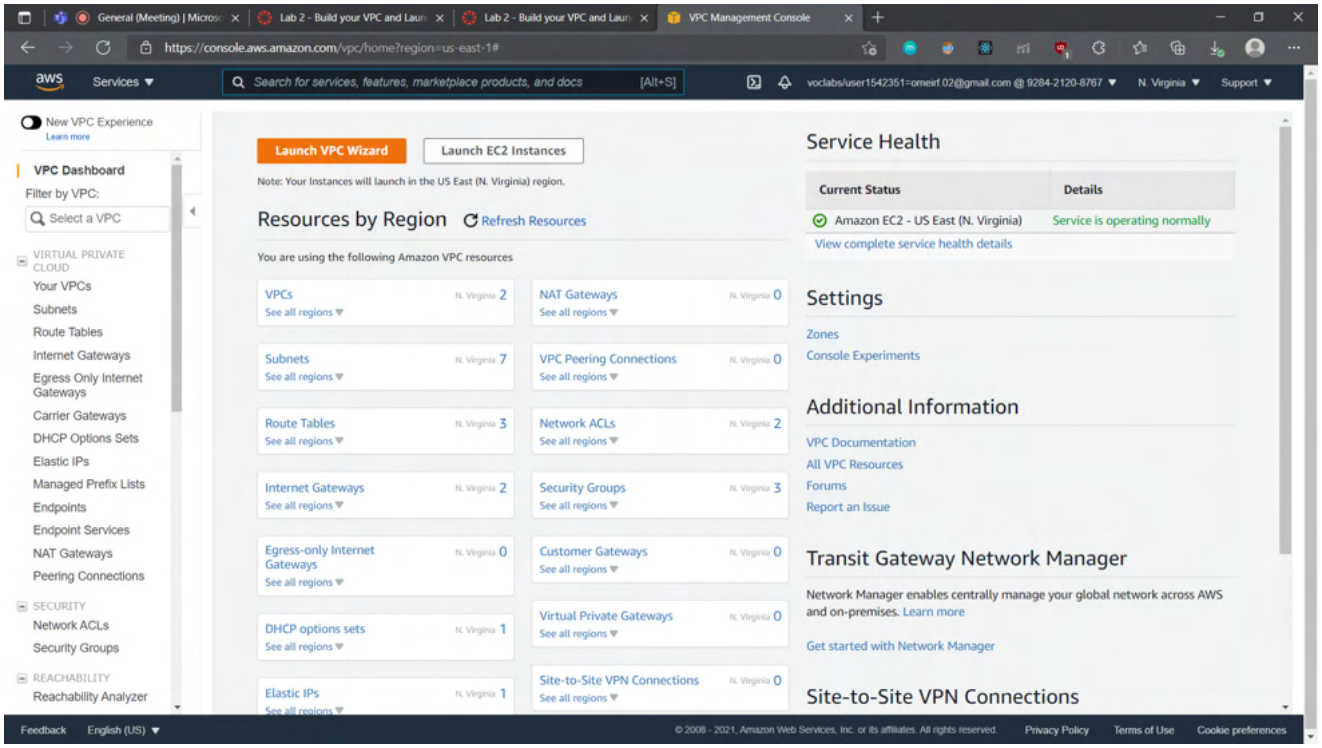
Task 1: Create Your VPC
In this task, you will use the VPC Wizard to create a VPC, an Internet Gateway and two subnets in a single Availability Zone. An Internet gateway (IGW) is a VPC component that allows communication between instances in your VPC and the Internet.

After creating a VPC, you can add subnets. Each subnet resides entirely within one Availability Zone and cannot span zones. If a subnet's traffic is routed to an Internet Gateway, the subnet is known as a public subnet. If a subnet does not have a route to the Internet gateway, the subnet is known as a private subnet.

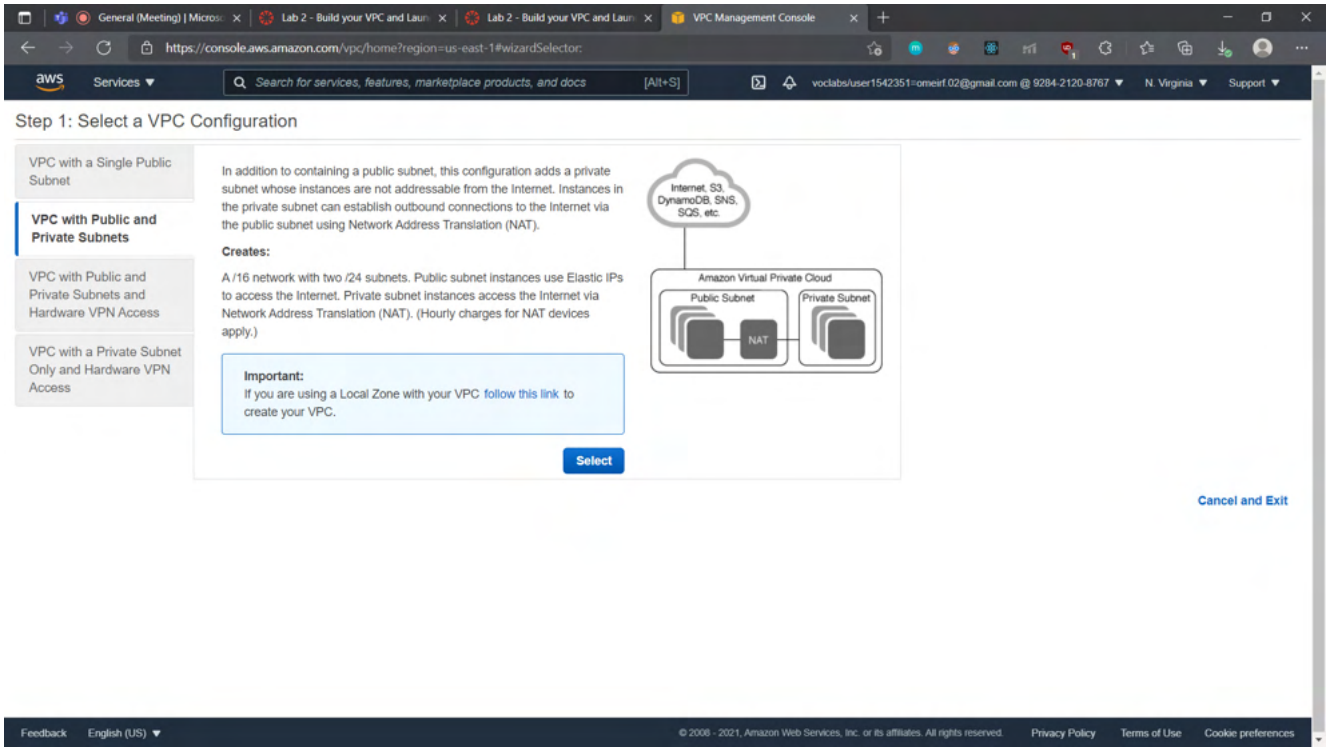
The wizard will also create a NAT Gateway, which is used to provide internet connectivity to EC2 instances in the private subnets.

1. In the **AWS Management Console**, on the **Services** menu, click **VPC**.

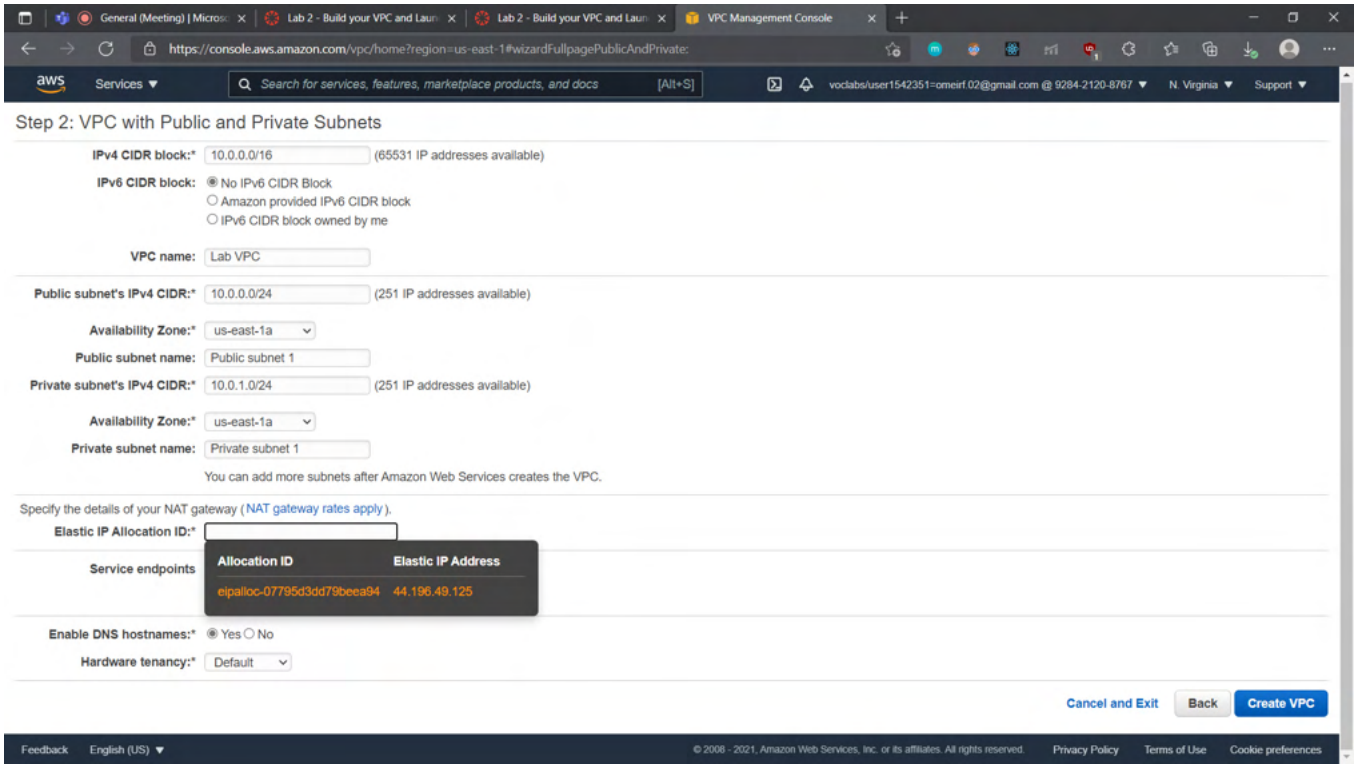
2. Click **Launch VPC Wizard**



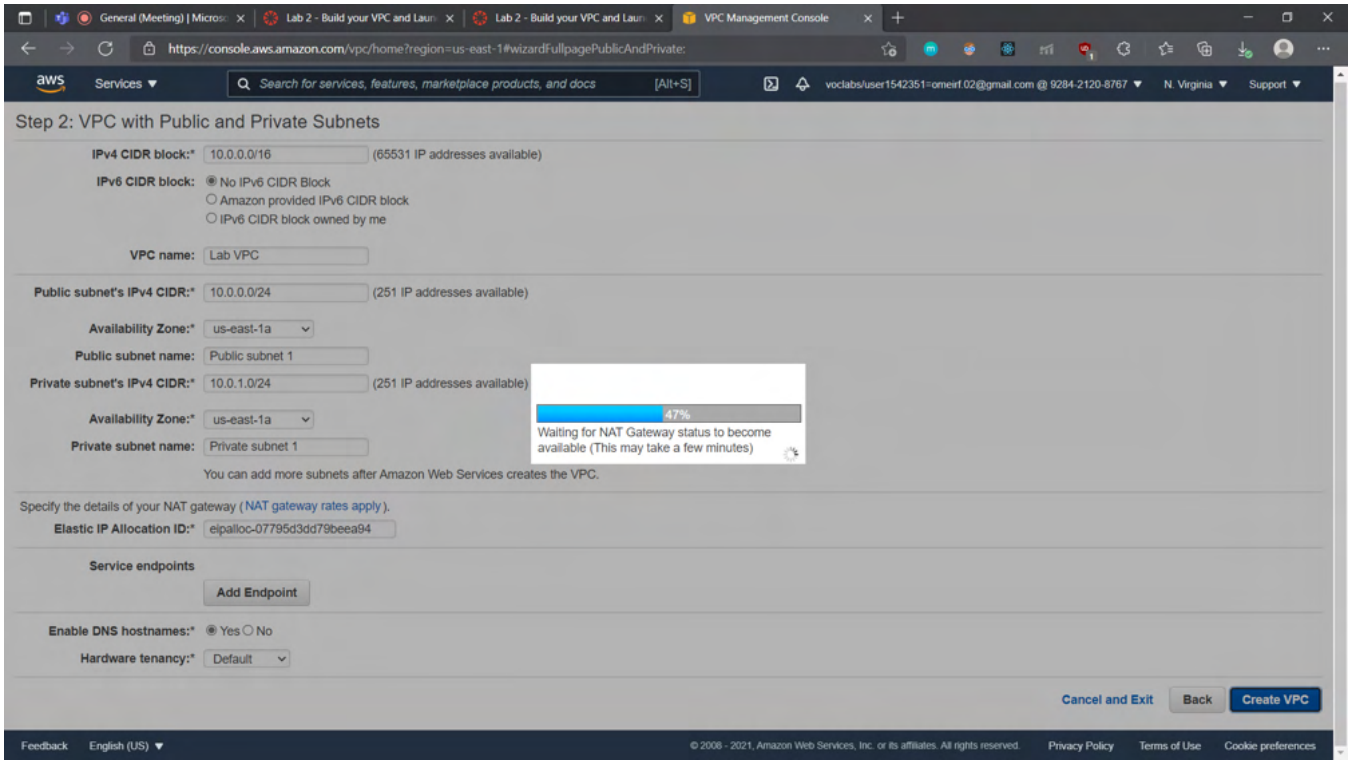
3. In the left navigation pane, click **VPC with Public and Private Subnets** (the second option).



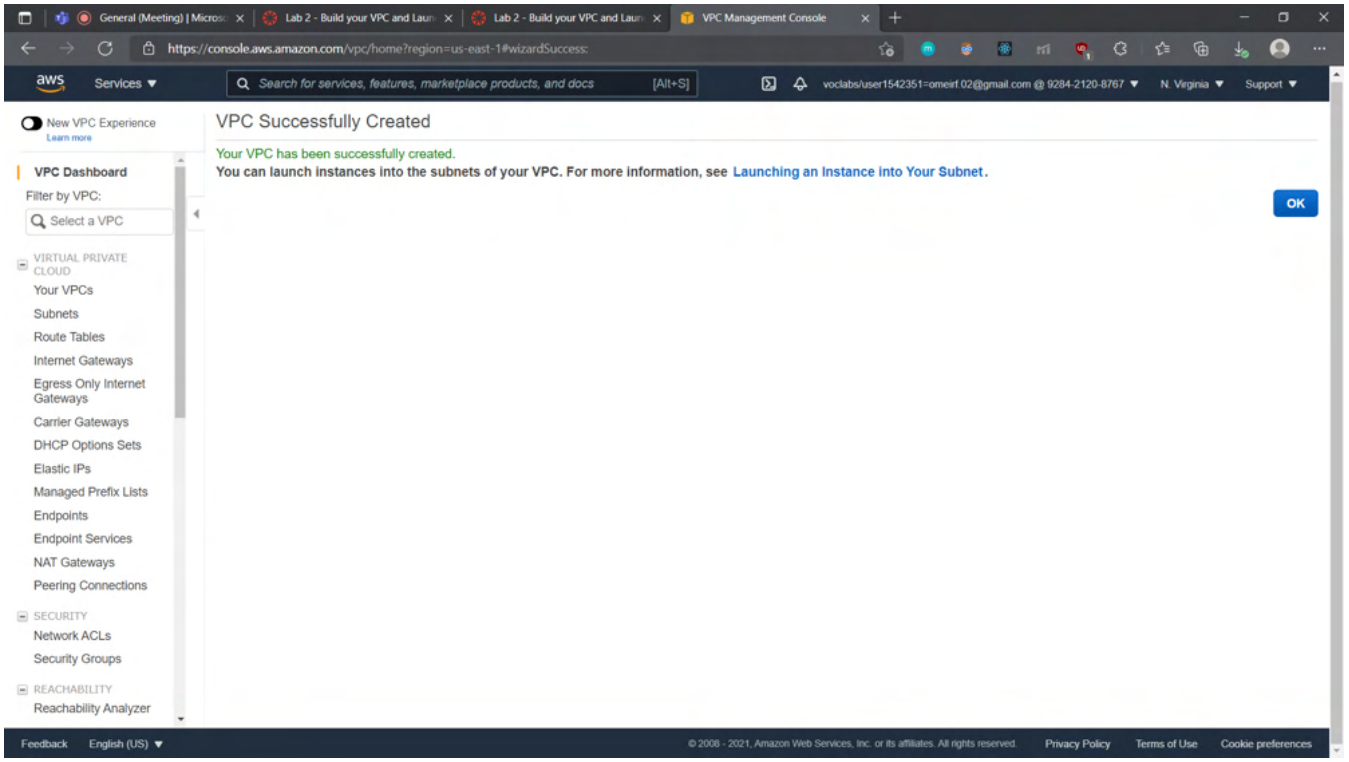
4. Click **Select** then configure:
   - **VPC name:** Lab VPC
   - **Availability Zone:** Select the *first* Availability Zone
   - **Public subnet name:** Public Subnet 1
   - **Availability Zone:** Select the *first* Availability Zone (the same as used above)
   - **Private subnet name:** Private Subnet 1
   - **Elastic IP Allocation ID:** Click in the box and select the displayed IP address



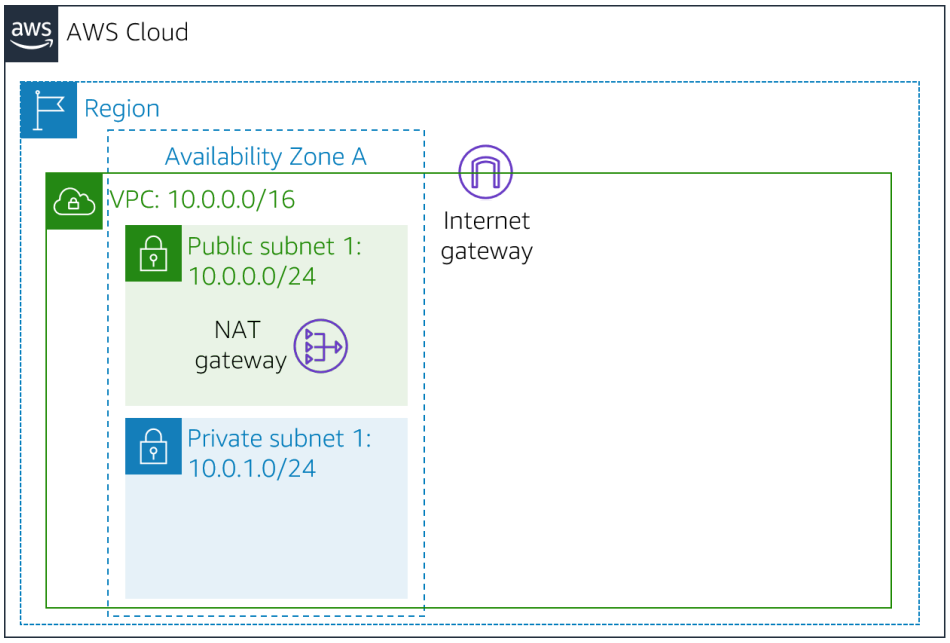
5. Click **Create VPC.**
The wizard will create your VPC.



6. Once it is complete, click **OK**



The wizard has provisioned a VPC with a public subnet and a private subnet in the same Availability Zone, together with route tables for each subnet:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

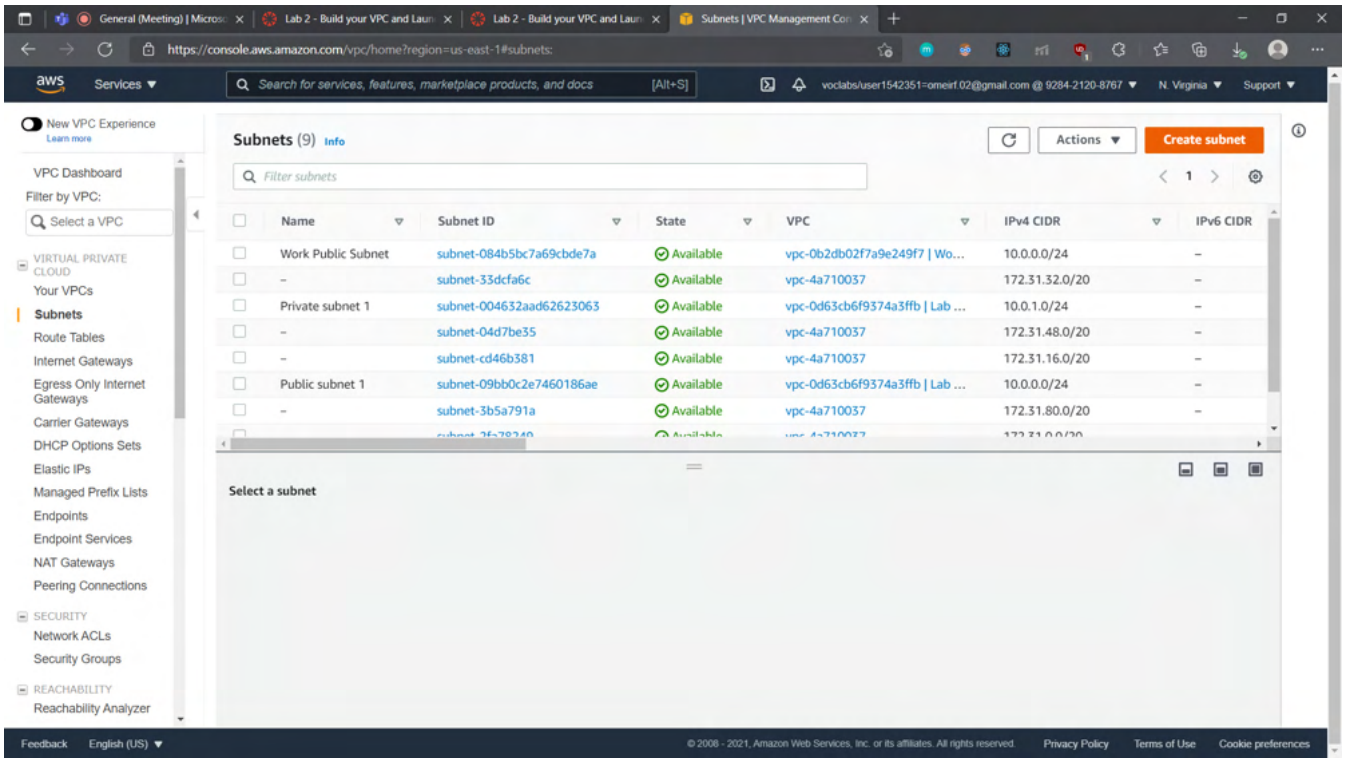
The Public Subnet has a CIDR of **10.0.0.0/24**, which means that it contains all IP addresses starting with **10.0.0.x**.

The Private Subnet has a CIDR of **10.0.1.0/24**, which means that it contains all IP addresses starting with **10.0.1.x**.

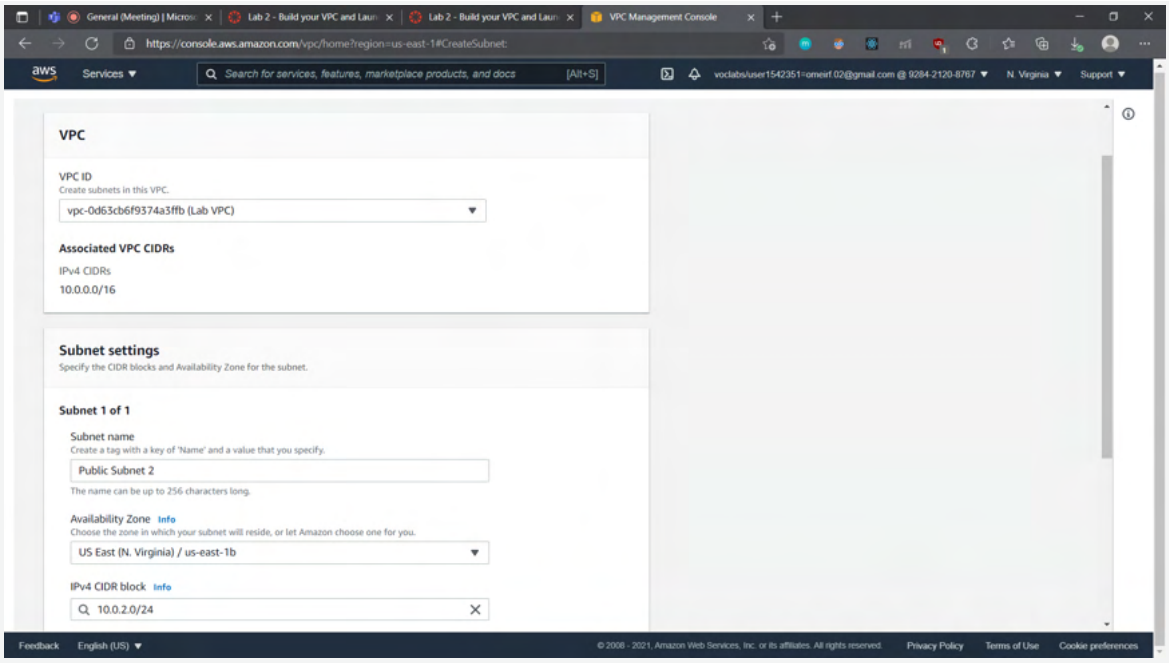
Task 2: Create Additional Subnets

In this task, you will create two additional subnets in a second Availability Zone. This is useful for creating resources in multiple Availability Zones to provide *High Availability*.

7. In the left navigation pane, click **Subnets**.
   First, you will create a second Public Subnet.

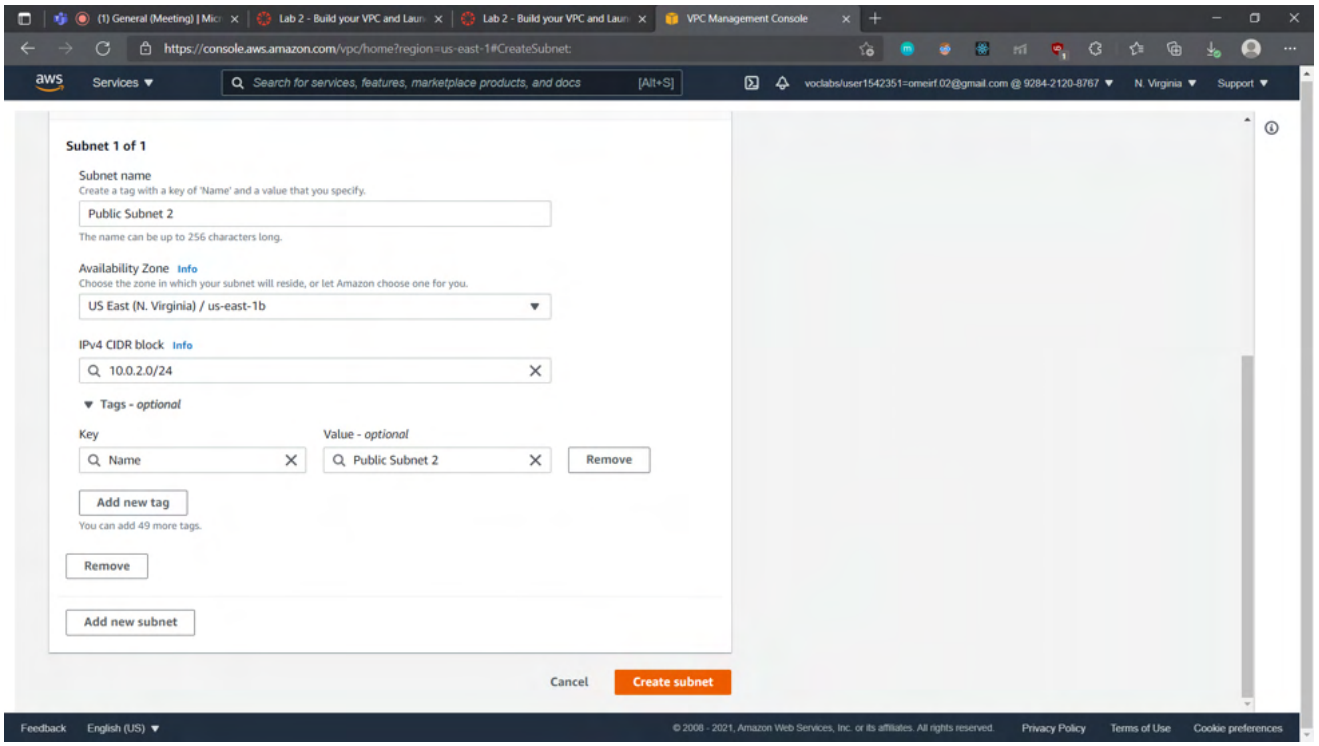


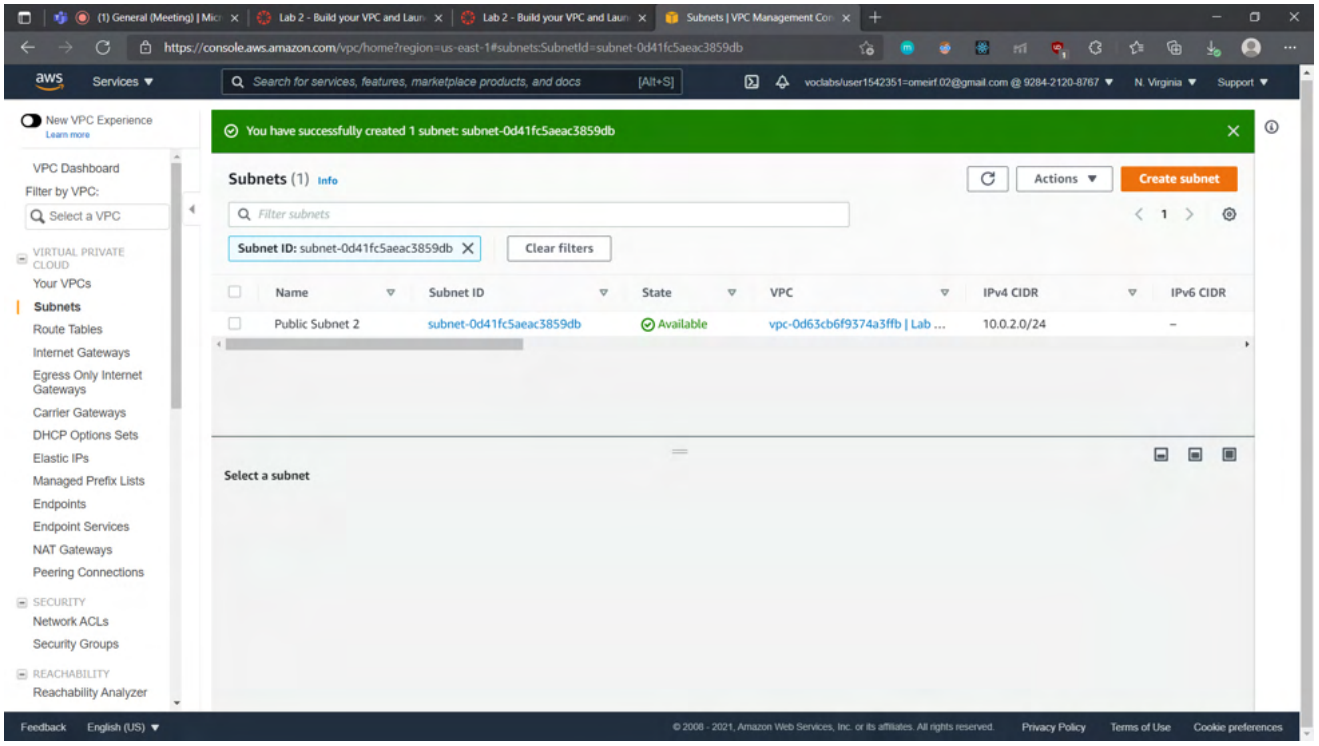
8. Click **Create subnet** then configure:
   - **VPC ID:** Lab VPC
   - **Subnet name:** Public Subnet 2
   - **Availability Zone:** Select the *second* Availability Zone
   - **IPv4 CIDR block:** 10.0.2.0/24



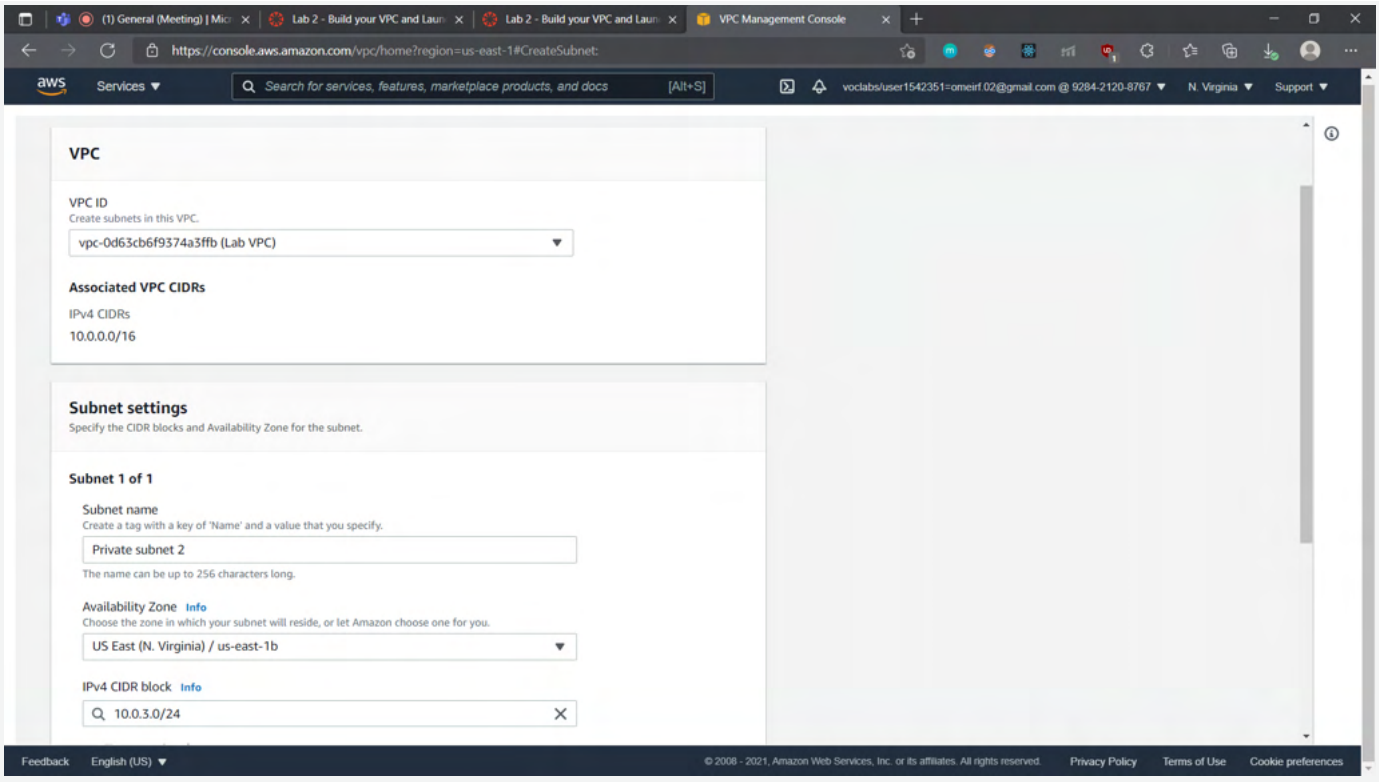
9. The subnet will have all IP addresses starting with **10.0.2.x**.
10. Click **Create subnet**

    You will now create a second Private Subnet.

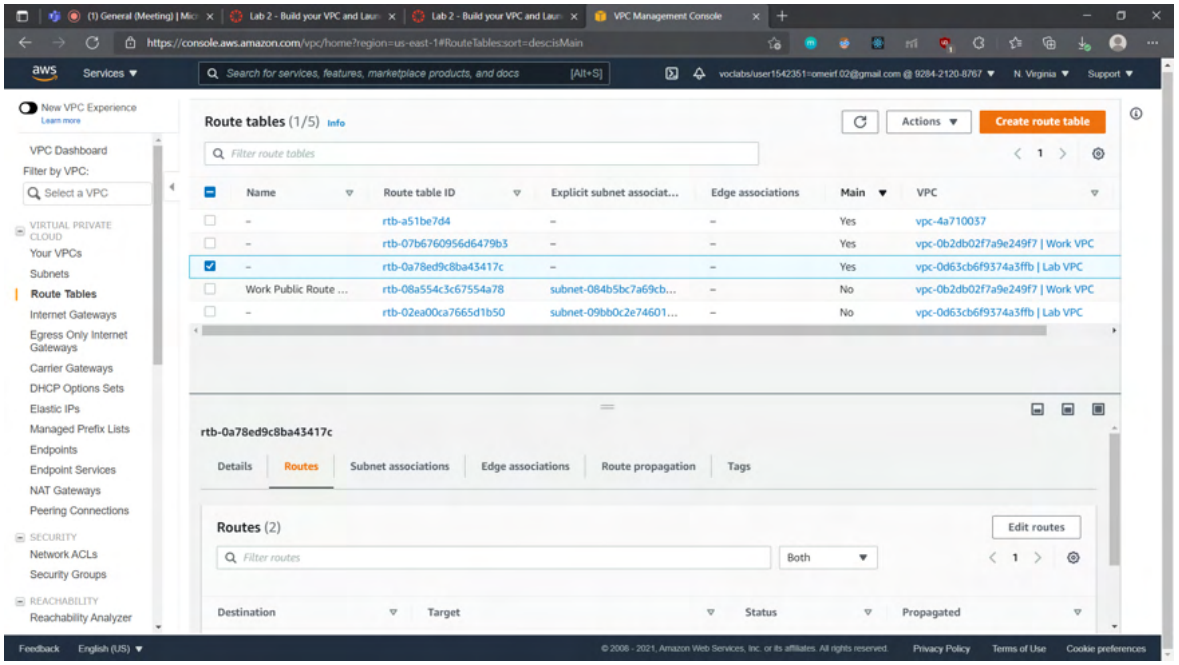




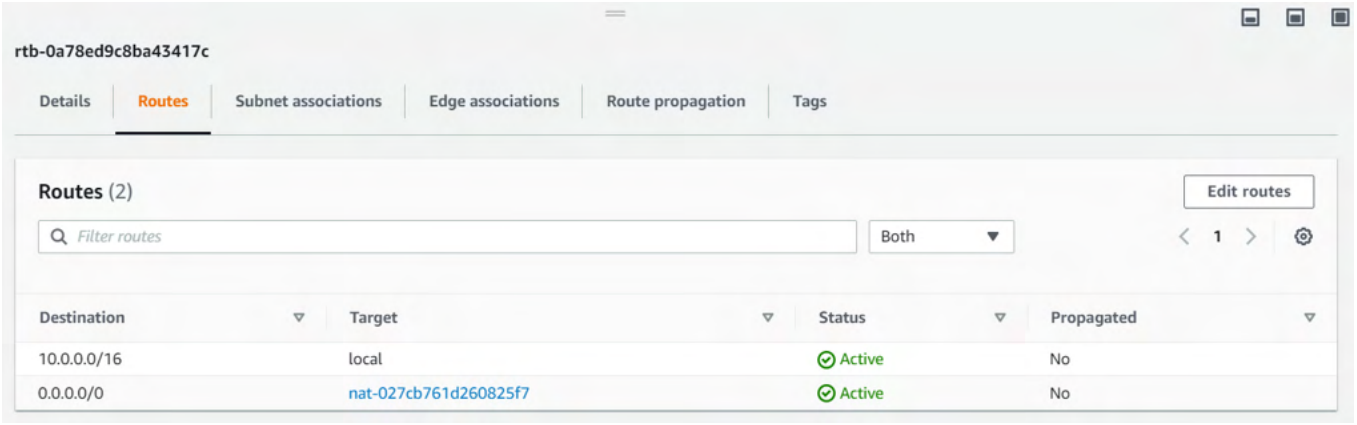
11. Click **Create subnet** then configure:
    - **VPC ID:** Lab VPC
    - **Subnet name:** Private Subnet 2
    - **Availability Zone:** Select the *second* Availability Zone
    - **CIDR block:** 10.0.3.0/24



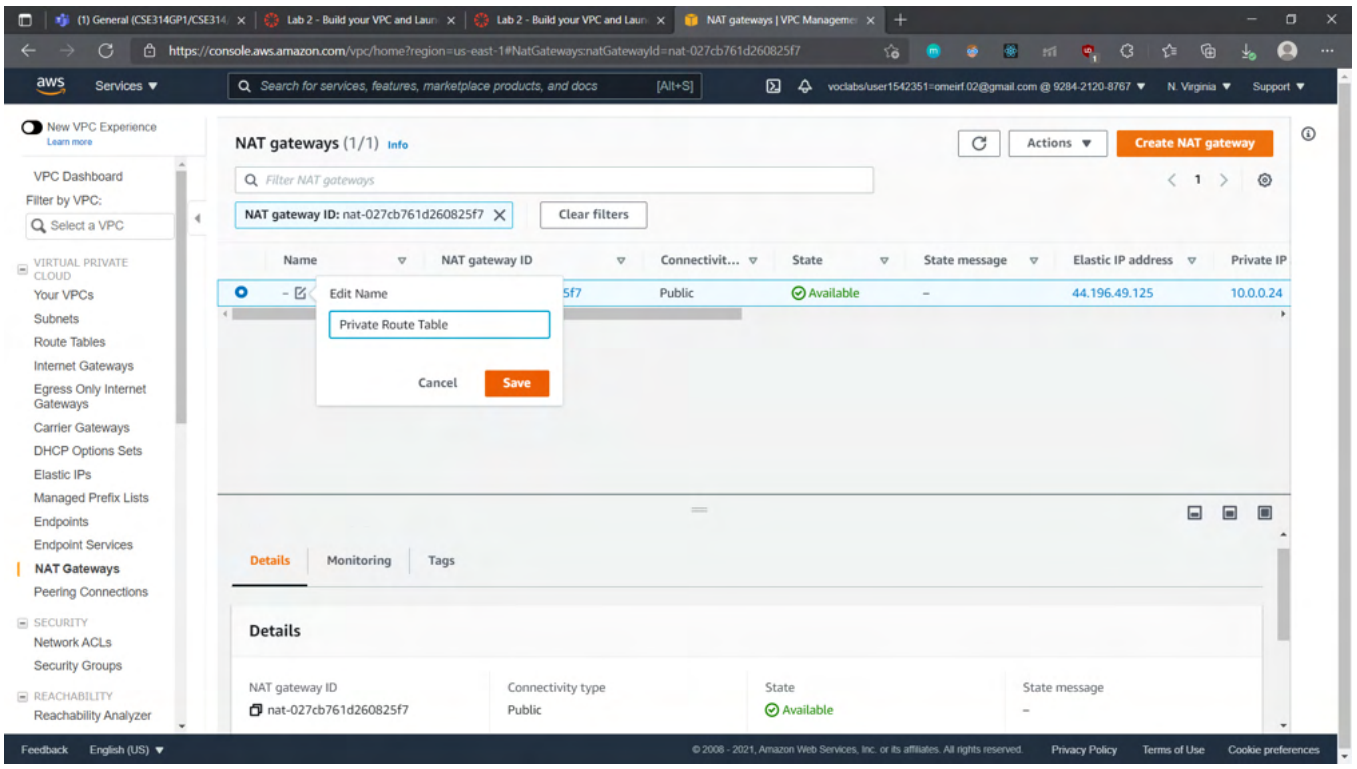
12. The subnet will have all IP addresses starting with **10.0.3.x**.
13. Click **Create subnet**
    You will now configure the Private Subnets to route internet-bound traffic to the NAT Gateway so that resources in the Private Subnet are able to connect to the Internet, while still keeping the resources private. This is done by configuring a *Route Table*.
    A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed. Each subnet in a VPC must be associated with a route table; the route table controls routing for the subnet.
14. In the left navigation pane, click **Route Tables**.
15. Select the route table with **Main = Yes** and **VPC = Lab VPC**. (Expand the *VPC ID* column if necessary to view the VPC name.)



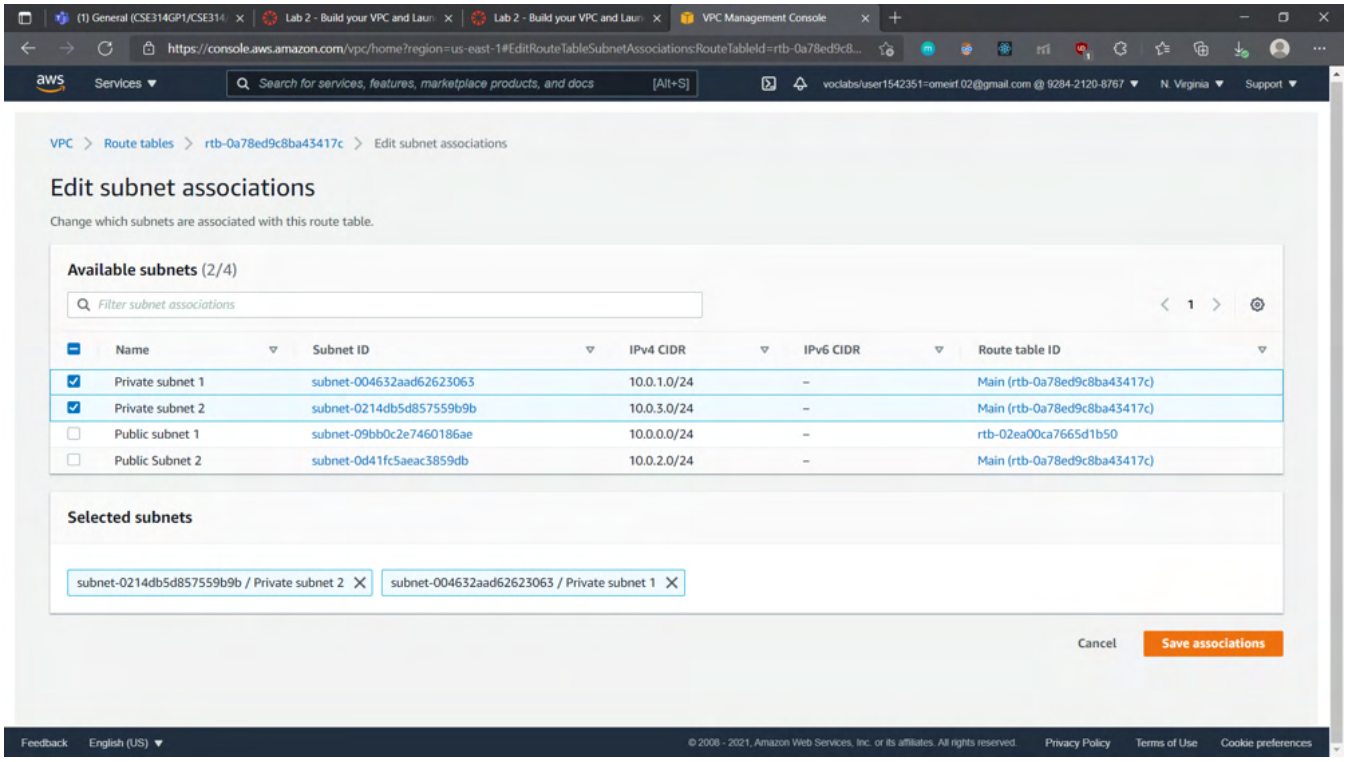
16. In the lower pane, click the **Routes** tab.
    Note that **Destination 0.0.0.0/0** is set to **Target nat-xxxxxxxx**. This means that traffic destined for the internet (0.0.0.0/0) will be sent to the NAT Gateway. The NAT Gateway will then forward the traffic to the internet.
    This route table is therefore being used to route traffic from Private Subnets. You will now add a name to the Route Table to make this easier to recognize in future.



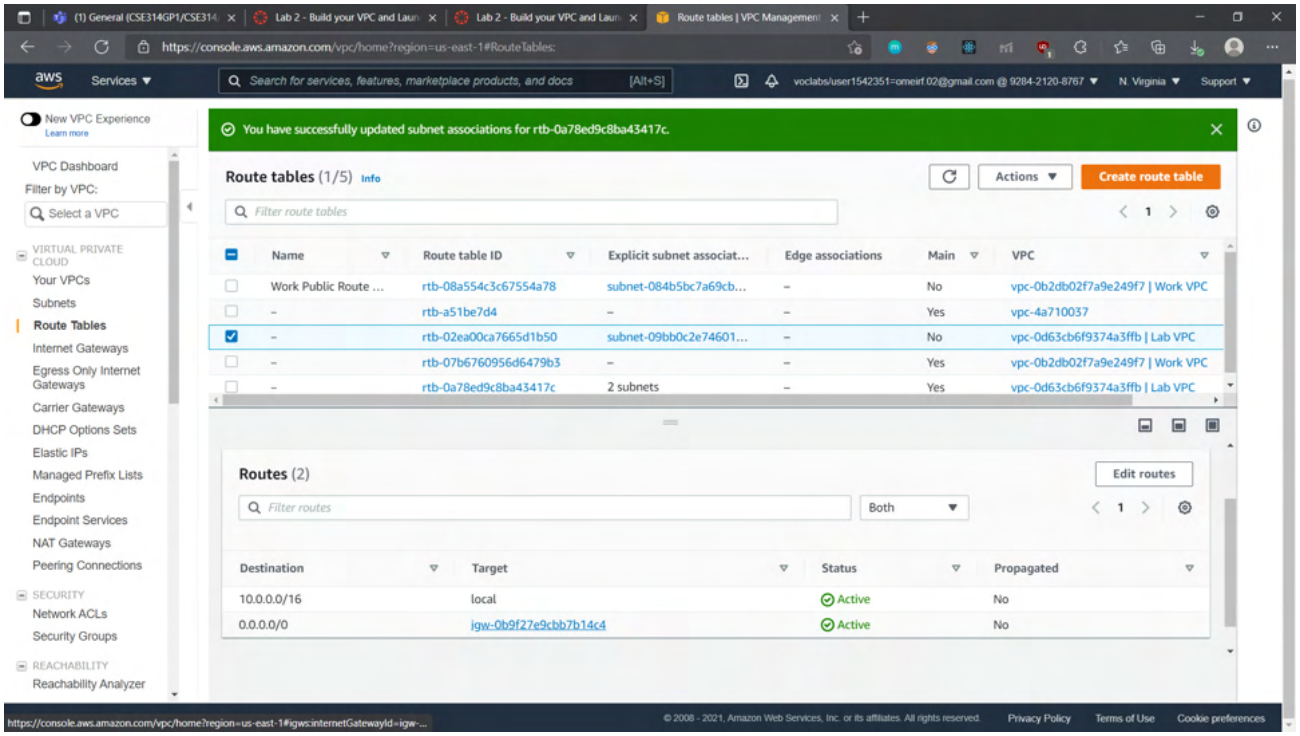
17. In the **Name** column for this route table, click the pencil then type Private Route Table and click **Save**



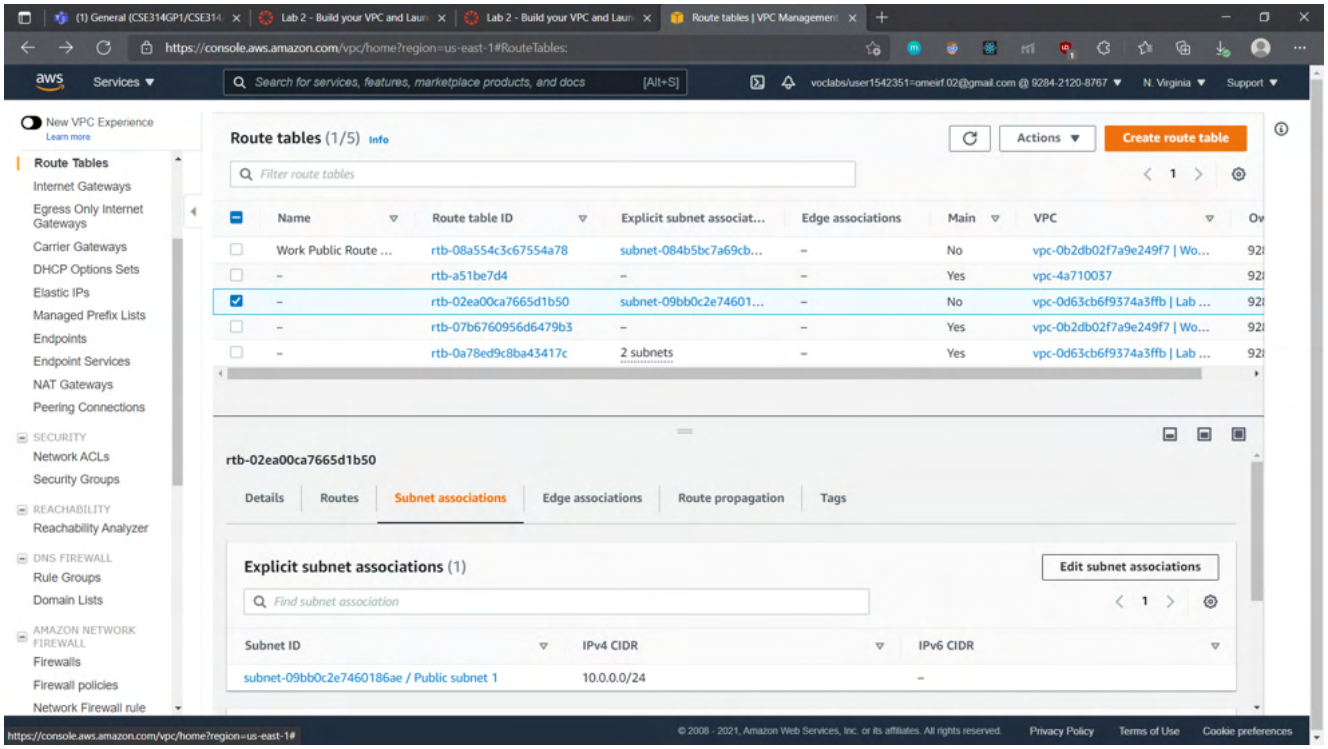
18. In the lower pane, click the **Subnet Associations** tab.
    You will now associate this route table to the Private Subnets.
19. Click **Edit subnet associations**
20. Select both **Private Subnet 1** and **Private Subnet 2**.
    You can expand the *Subnet ID* column to view the Subnet names.



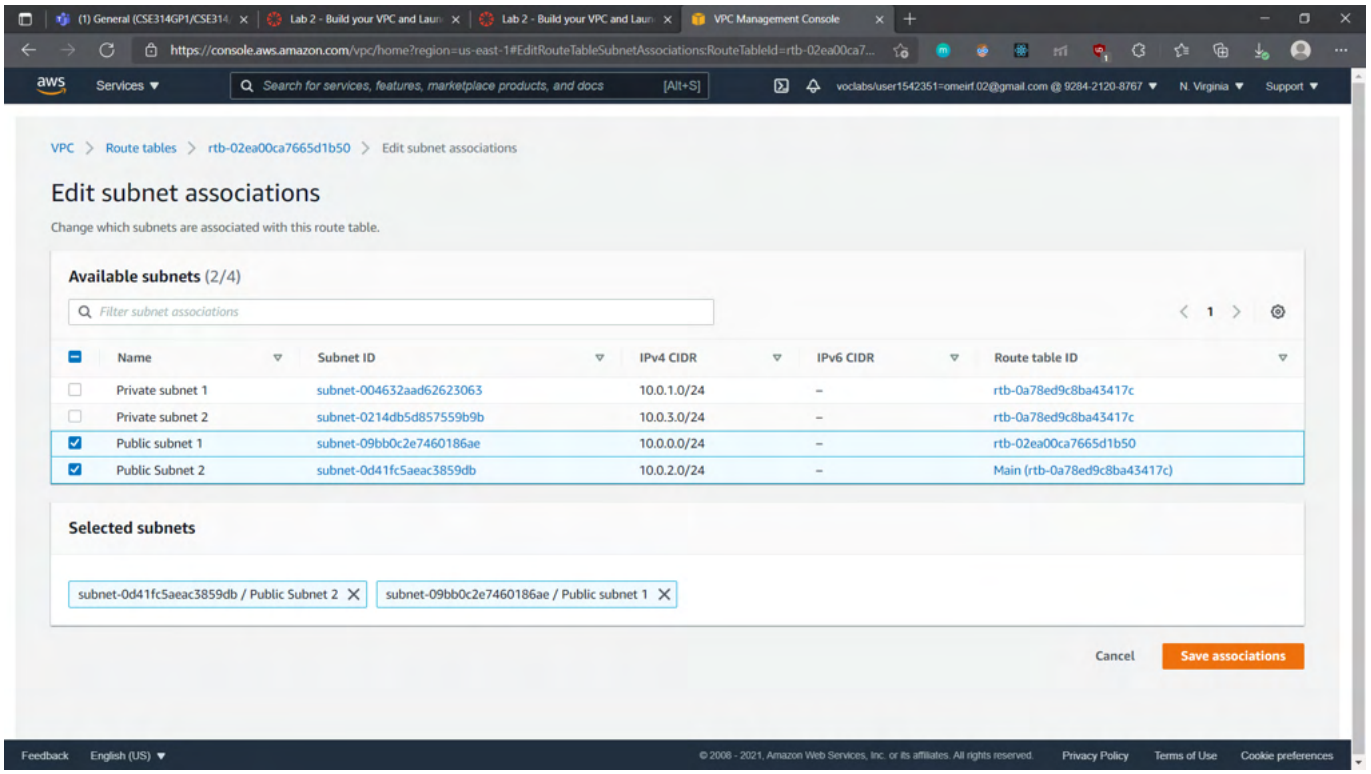
21. Click **Save associations**
    You will now configure the Route Table that is used by the Public Subnets.
22. Select the route table with **Main = No** and **VPC = Lab VPC** (and deselect any other subnets).
23. In the **Name** column for this route table, click the pencil then type Public Route Table, and click **Save**
24. In the lower pane, click the **Routes** tab.
    Note that **Destination 0.0.0.0/0** is set to **Target igw-xxxxxxxx**, which is the Internet Gateway. This means that internet-bound traffic will be sent straight to the internet via the Internet Gateway.
    You will now associate this route table to the Public Subnets.



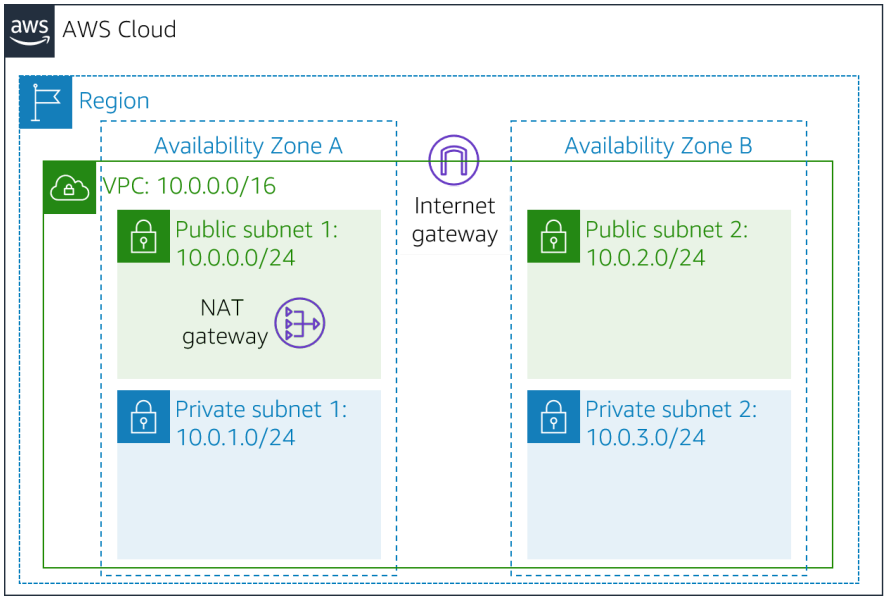
25. Click the **Subnet Associations** tab.
26. Click **Edit subnet associations**



27. Select both **Public Subnet 1** and **Public Subnet 2**.
28. Click **Save associations**



Your VPC now has public and private subnets configured in two Availability Zones:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

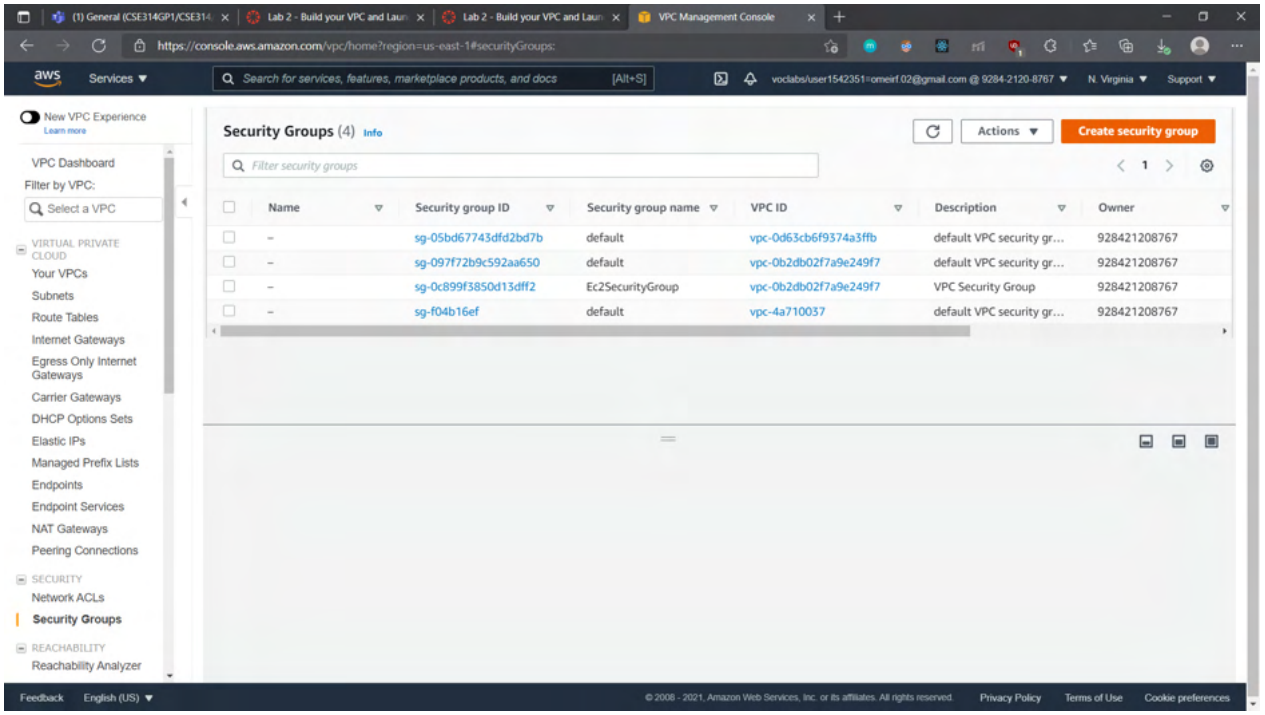
Private Route Table

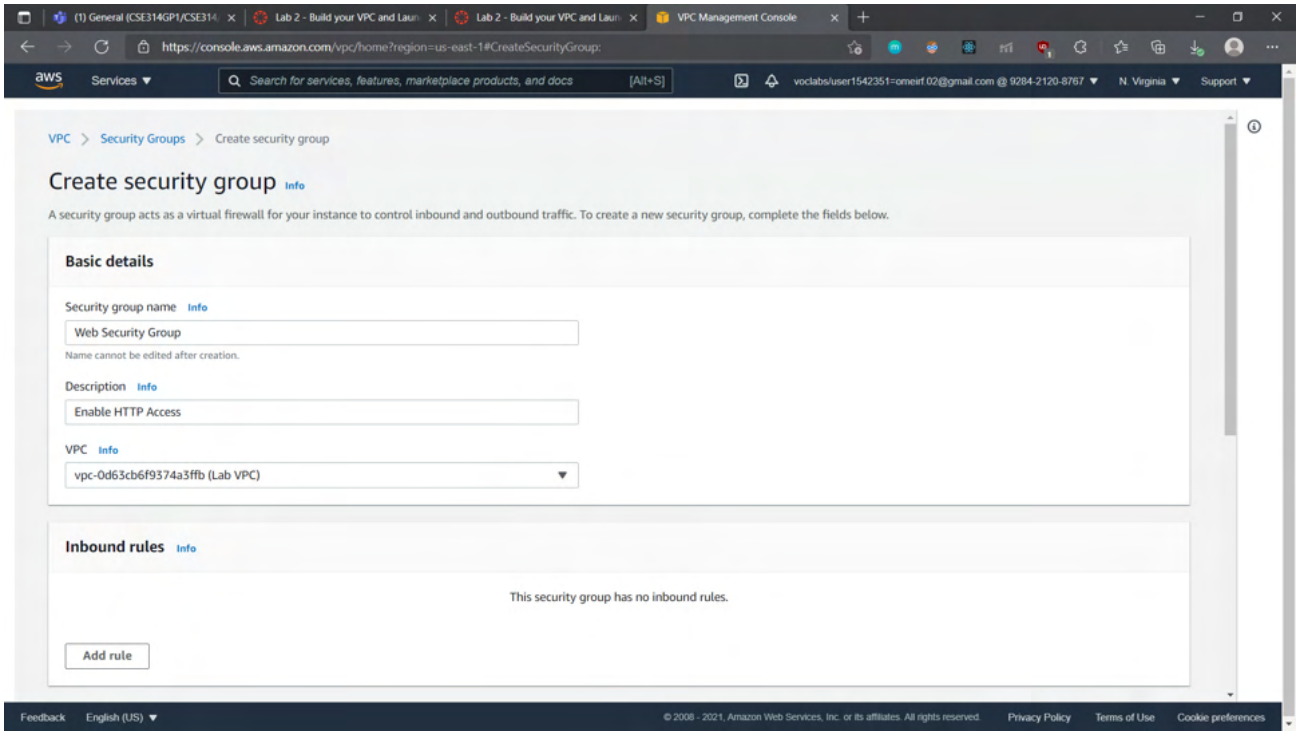
Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Task 3: Create a VPC Security Group

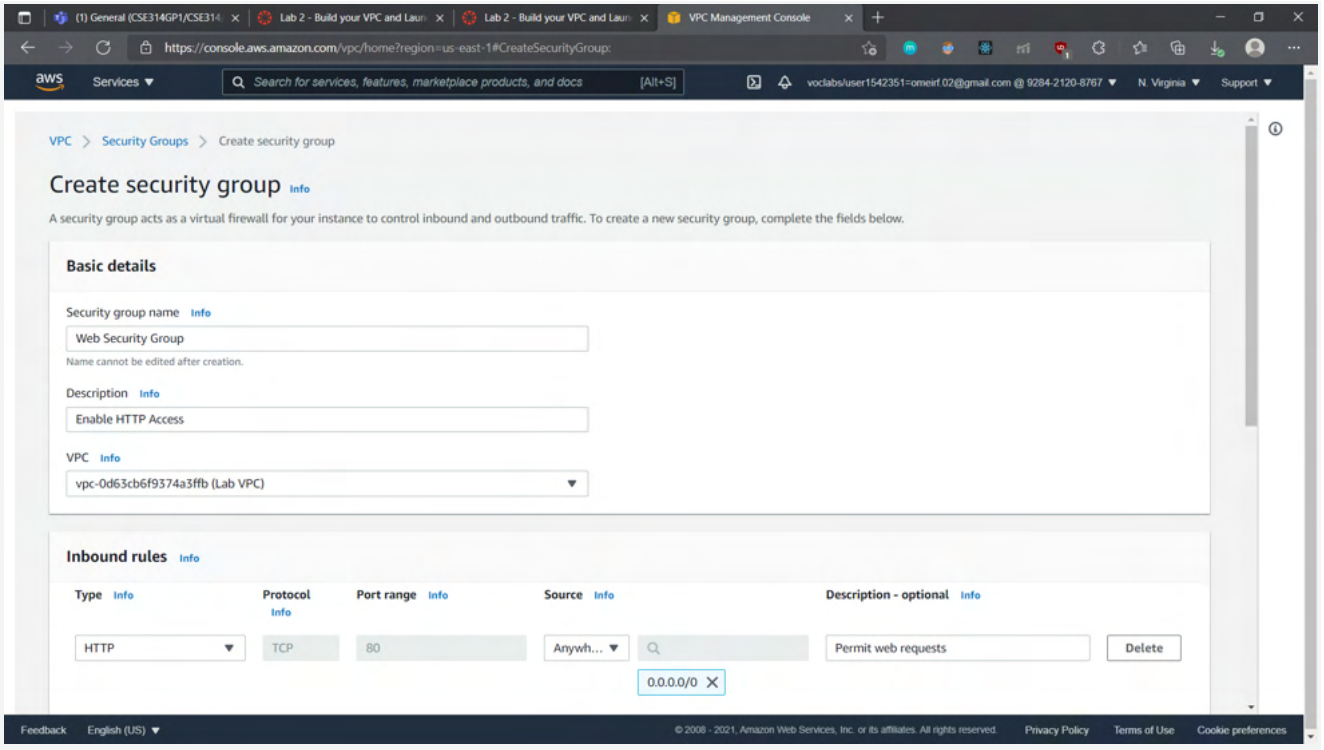
In this task, you will create a VPC security group, which acts as a virtual firewall. When you launch an instance, you associate one or more security groups with the instance. You can add rules to each security group that allow traffic to or from its associated instances.

29. In the left navigation pane, click **Security Groups**.
30. Click **Create security group** and then configure:
    a. **Security group name:** Web Security Group
    b. **Description:** Enable HTTP access
    c. **VPC:** *Lab VPC*

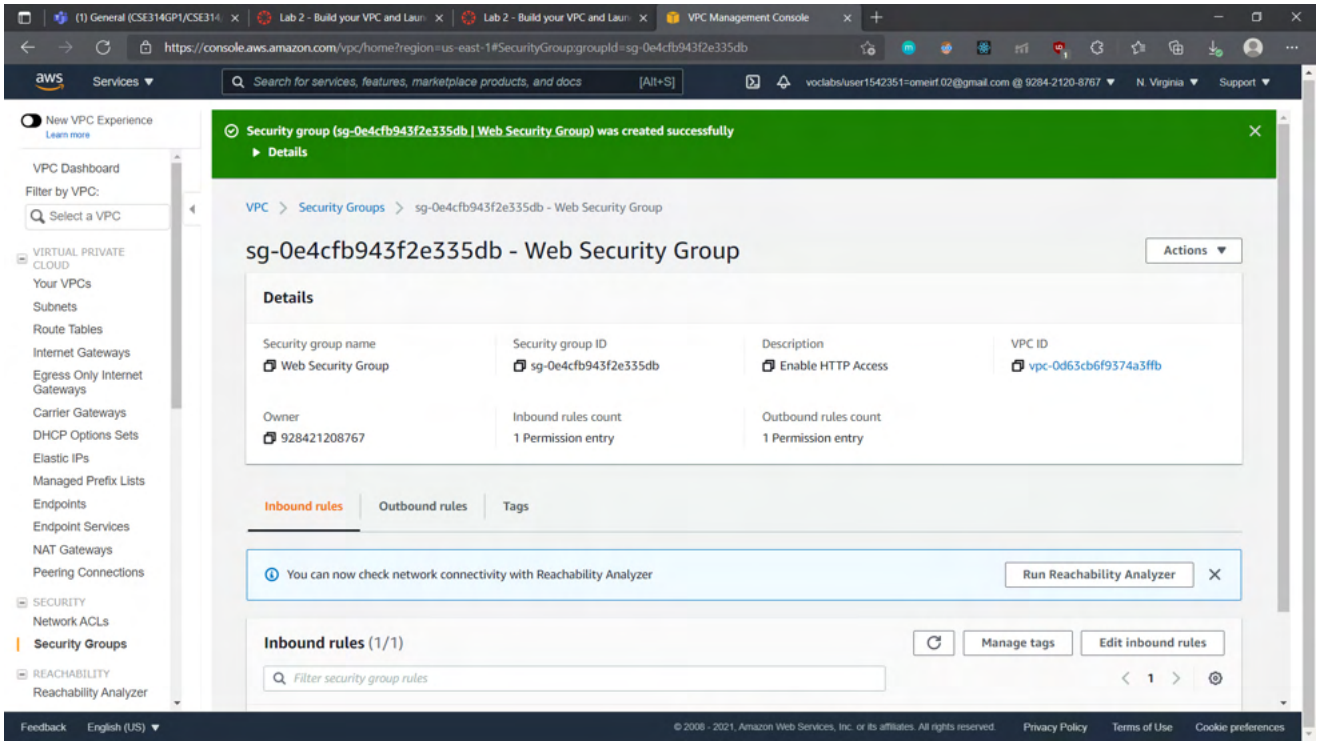




31. In the **Inbound rules** pane, choose **Add rule**
32. Configure the following settings:
    a. **Type:** *HTTP*
    b. **Source:** *Anywhere*
    c. **Description:** Permit web requests



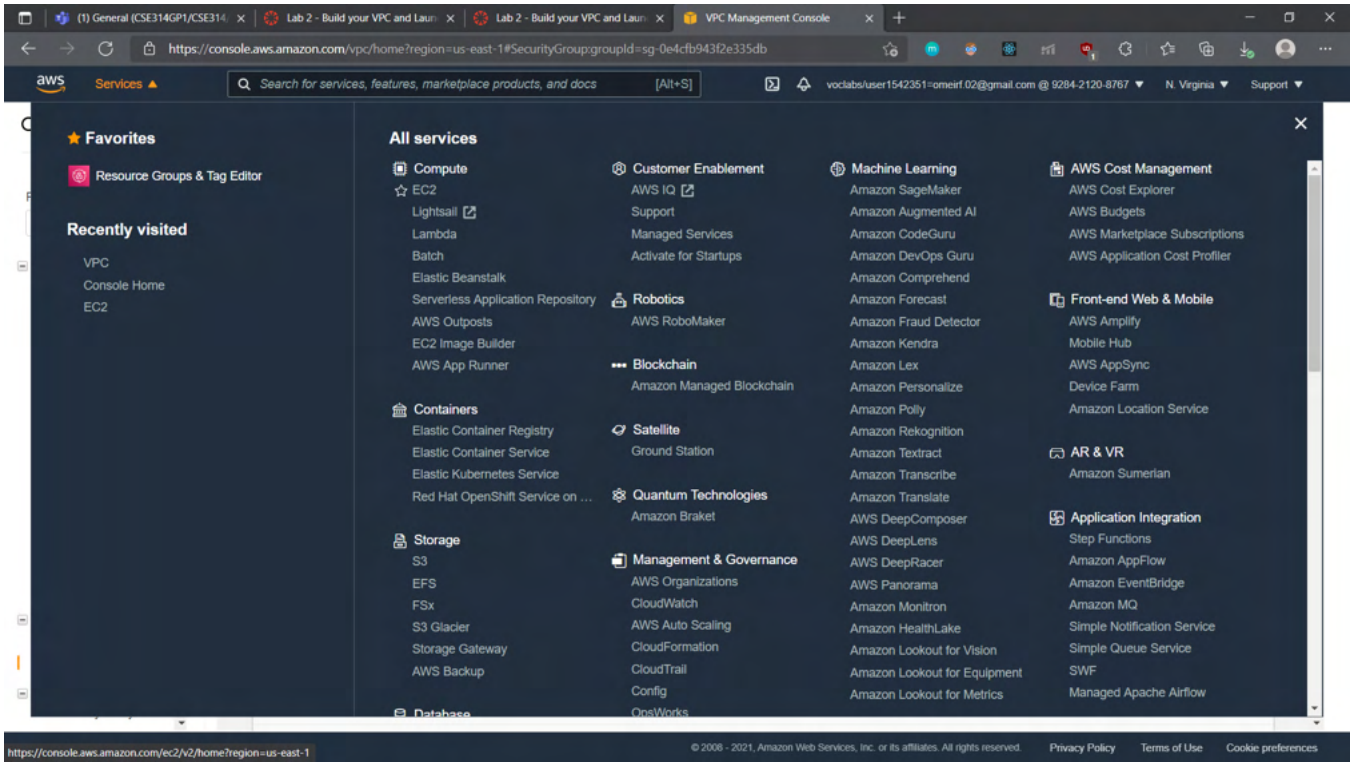
33. Scroll to the bottom of the page and choose **Create security group**
    You will use this security group in the next task when launching an Amazon EC2 instance.



Task 4: Launch a Web Server Instance

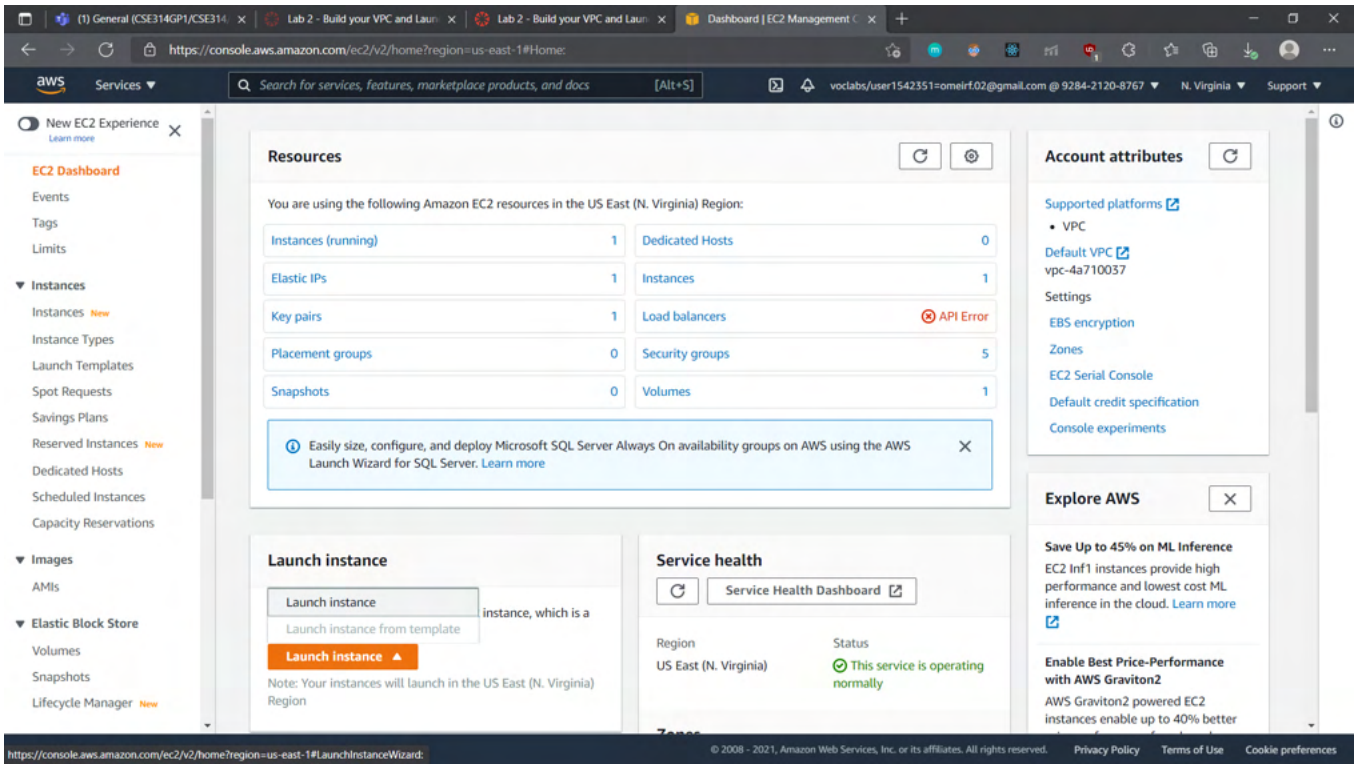
In this task, you will launch an Amazon EC2 instance into the new VPC. You will configure the instance to act as a web server.

34. On the **Services** menu, click **EC2**.



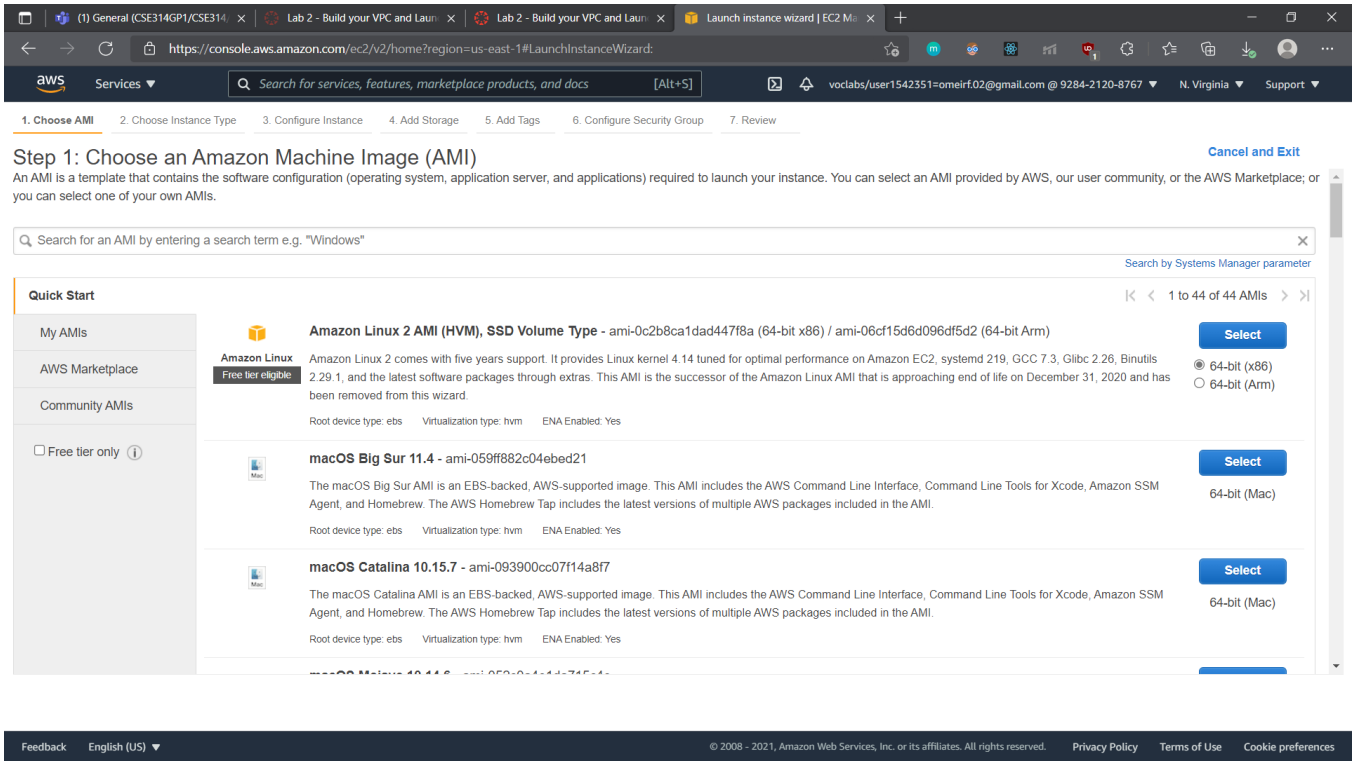
35. Click **Launch Instance**, and then choose **Launch Instance**

    First, you will select an *Amazon Machine Image (AMI)*, which contains the desired Operating System.



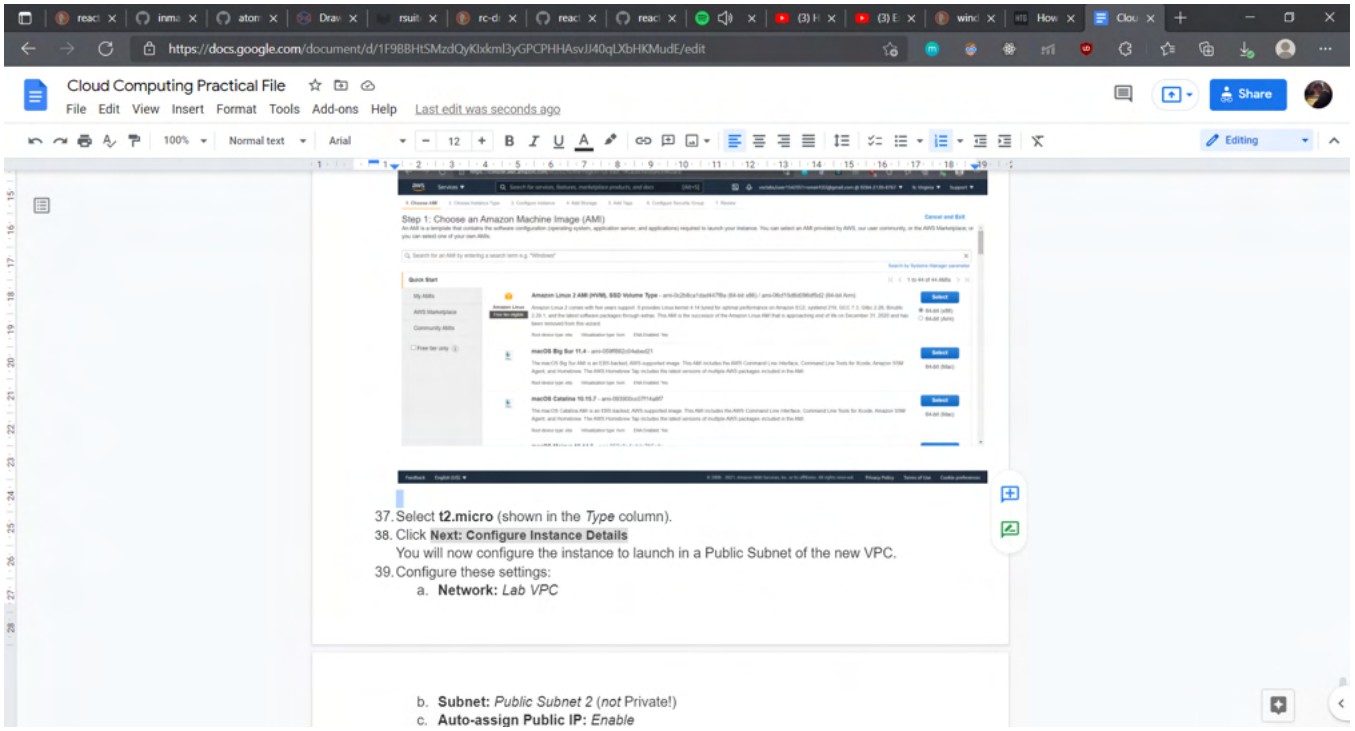
36. In the row for **Amazon Linux 2** (at the top), click **Select**

    The *Instance Type* defines the hardware resources assigned to the instance.

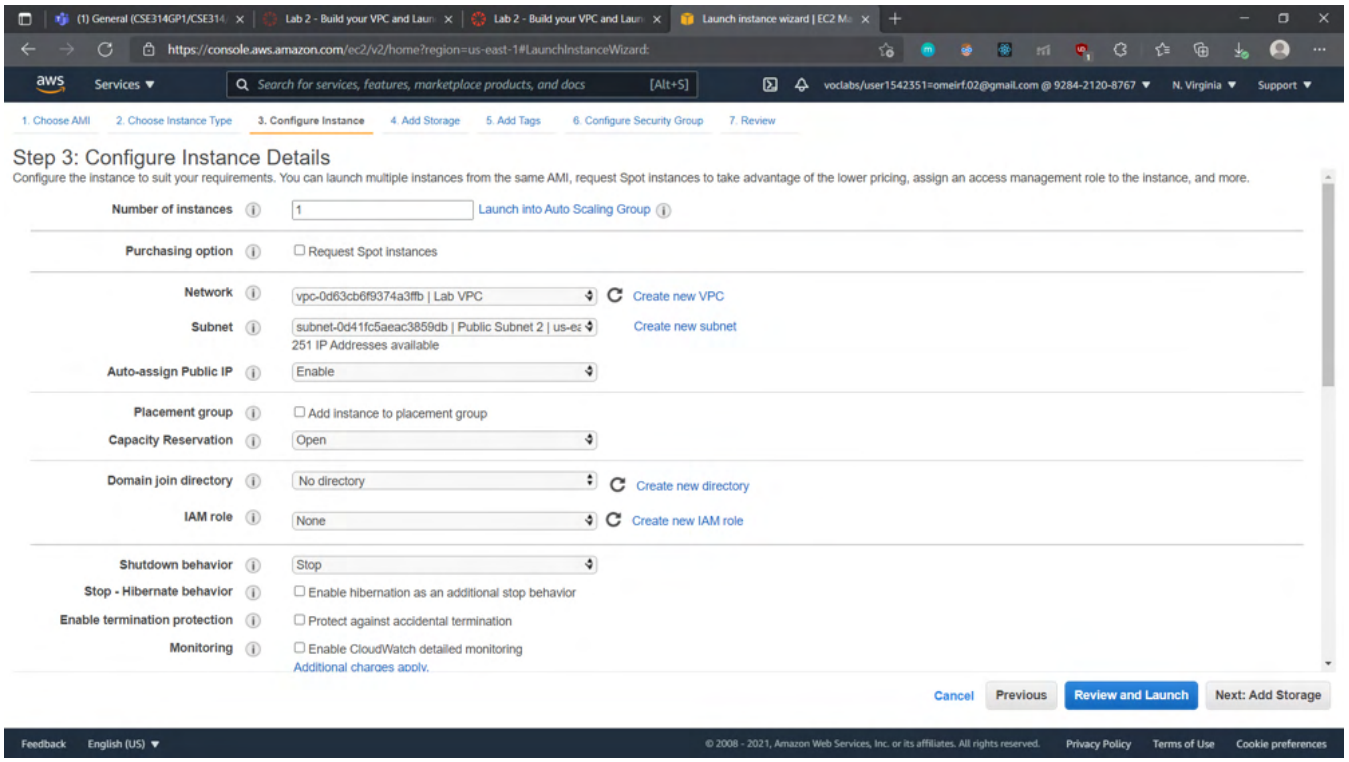


37. Select **t2.micro** (shown in the *Type* column).
38. Click **Next: Configure Instance Details**

    You will now configure the instance to launch in a Public Subnet of the new VPC.



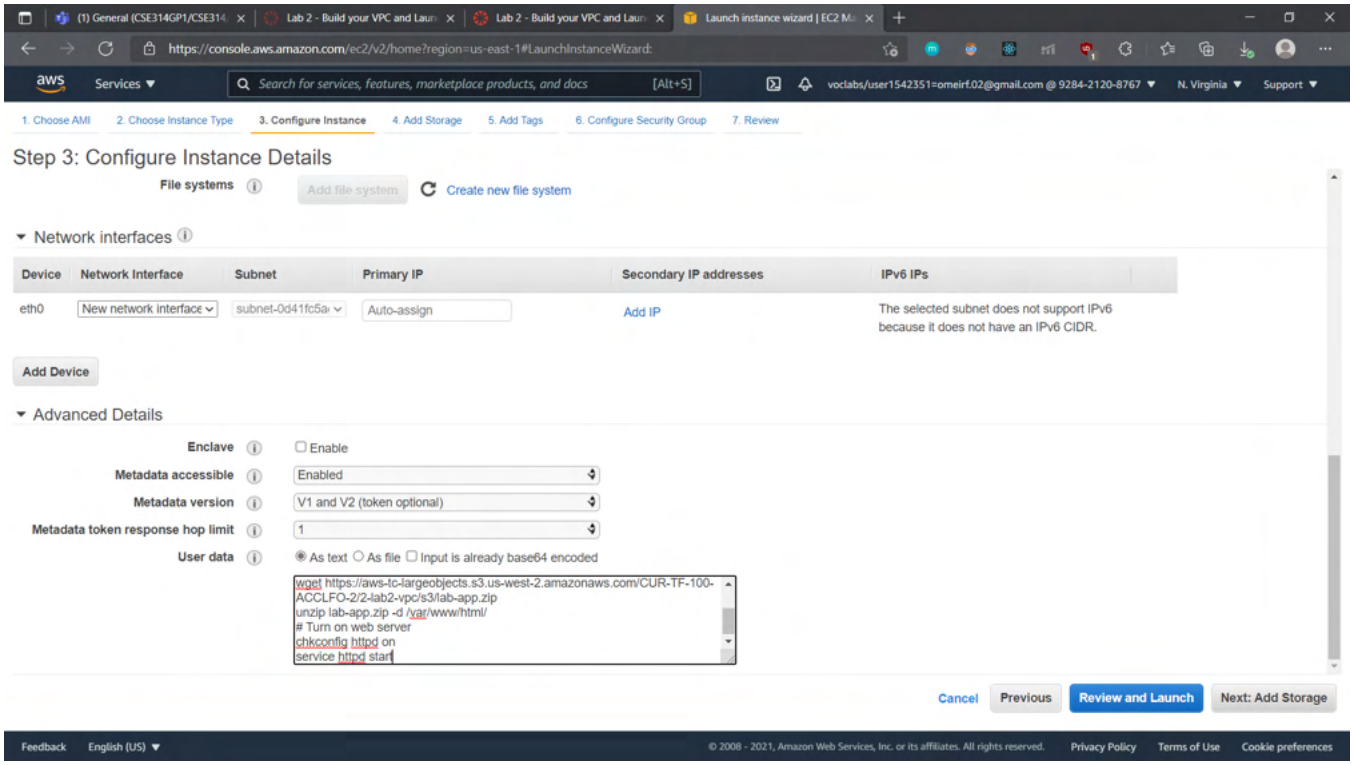
39. Configure these settings:
    a. **Network:** *Lab VPC*
    b. **Subnet:** *Public Subnet 2* (*not* Private!)
    c. **Auto-assign Public IP:** *Enable*



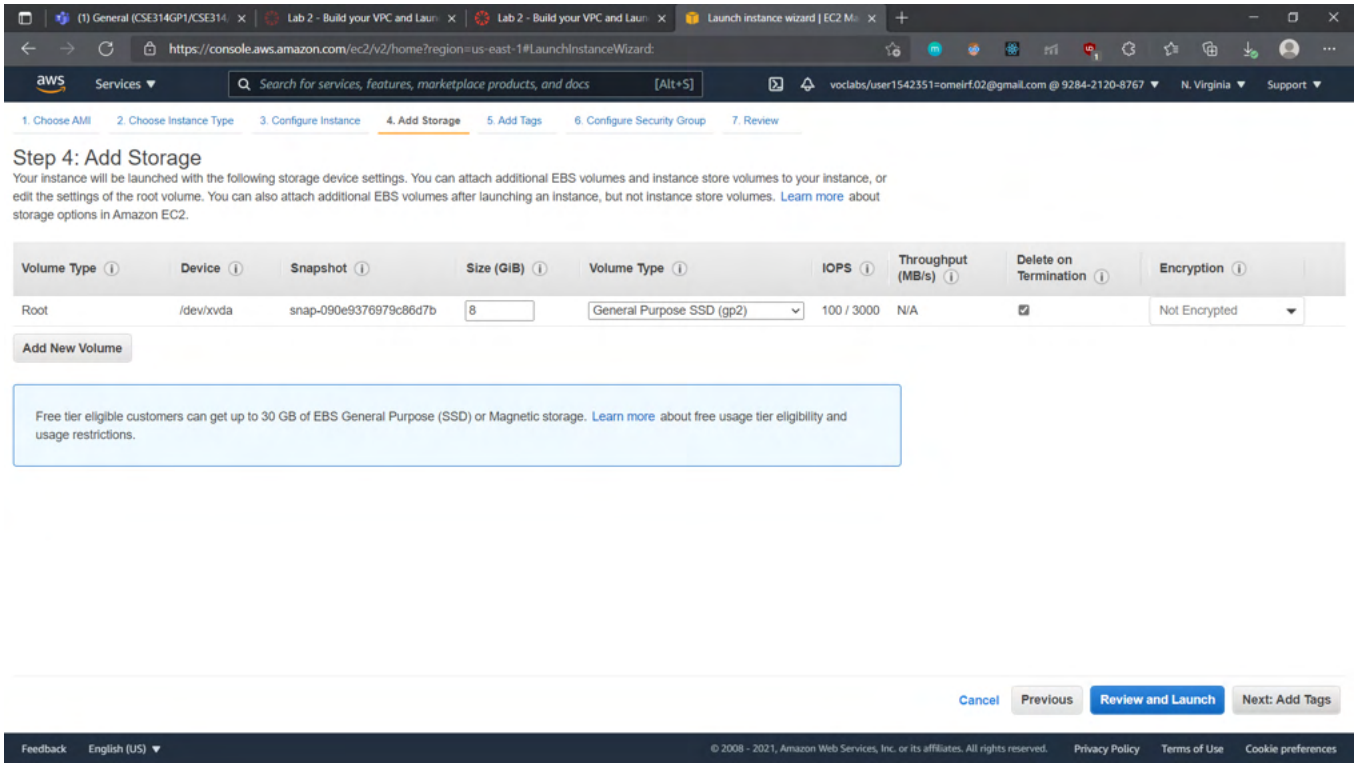
40. Expand the **Advanced Details** section (at the bottom of the page).
41. Copy and paste this code into the **User data** box:

```
#!/bin/bash
# Install Apache Web Server and PHP
yum install -y httpd mysql php
# Download Lab files
wget
https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-100-AC
CLFO-2/2-lab2-vpc/s3/lab-app.zip
unzip lab-app.zip -d /var/www/html/
# Turn on web server
chkconfig httpd on
service httpd start
```

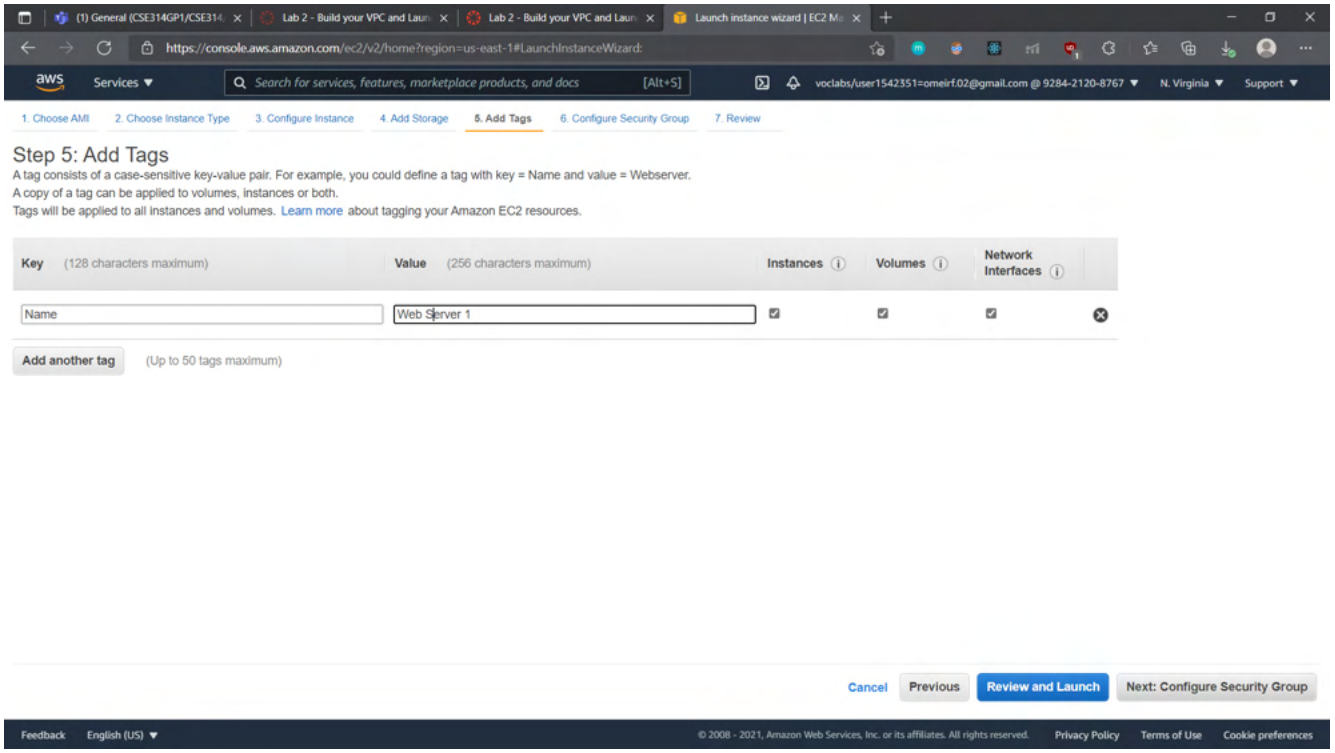
This script will be run automatically when the instance launches for the first time. The script loads and configures a PHP web application.



42. Click **Next: Add Storage**
    You will use the default settings for storage.

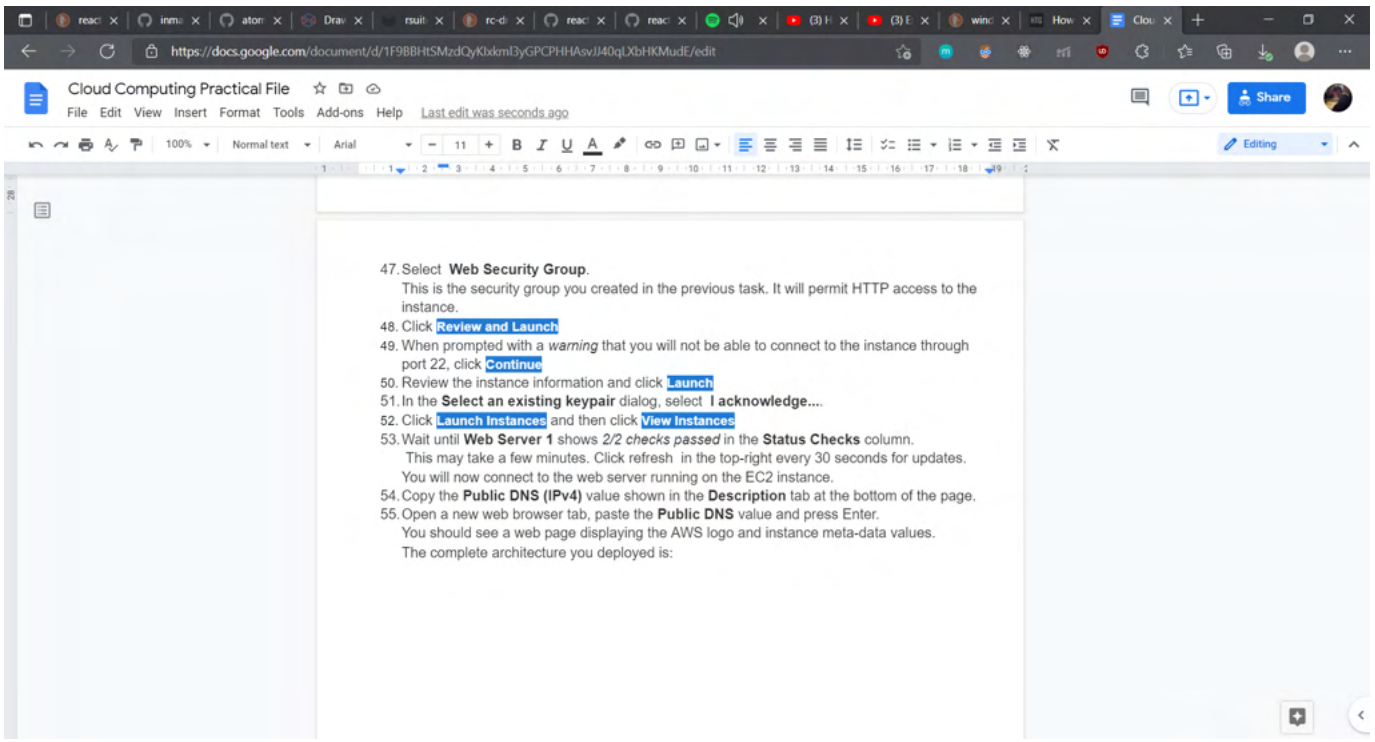


43. Click **Next: Add Tags**
    Tags can be used to identify resources. You will use a tag to assign a Name to the instance.
44. Click **Add Tag** then configure:
    a. **Key:** Name
    b. **Value:** Web Server 1

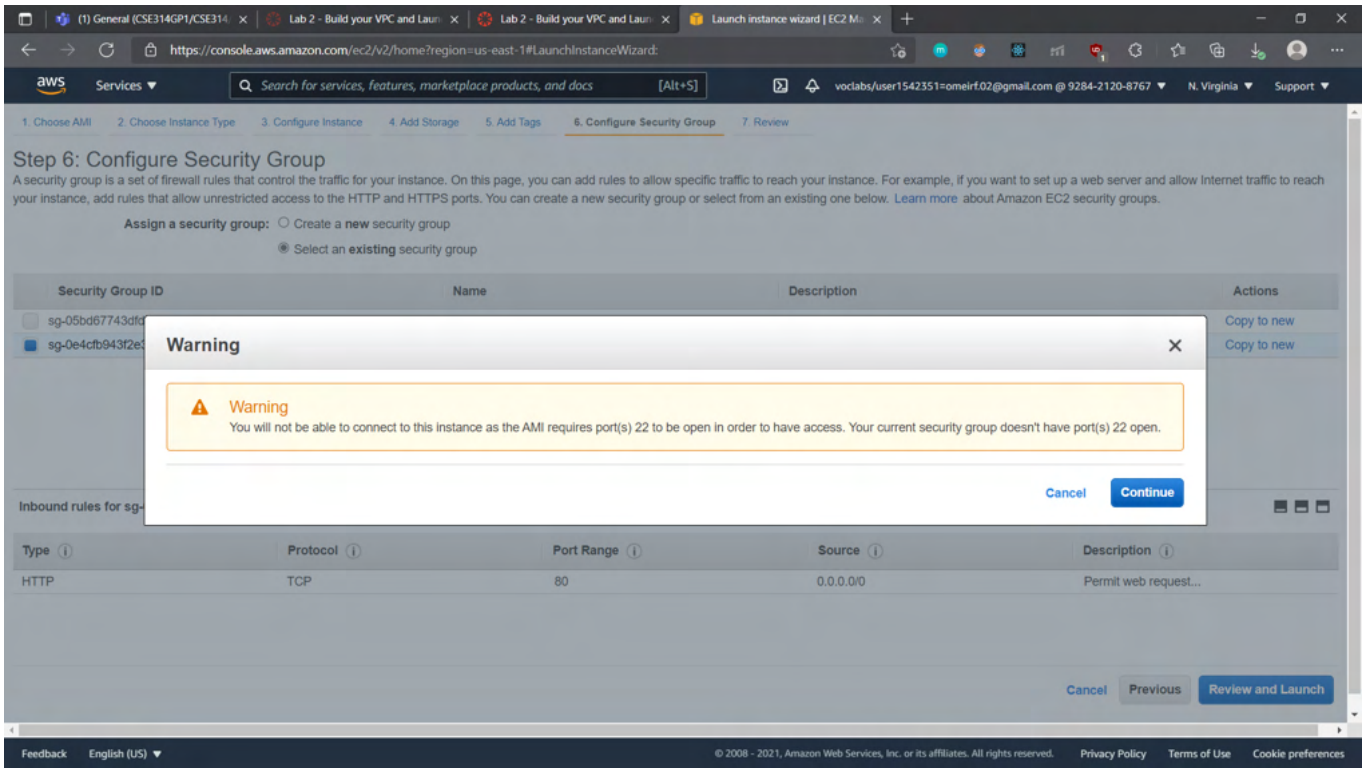


45. Click **Next: Configure Security Group**
    You will configure the instance to use the *Web Security Group* that you created earlier.
46. Select **Select an existing security group**

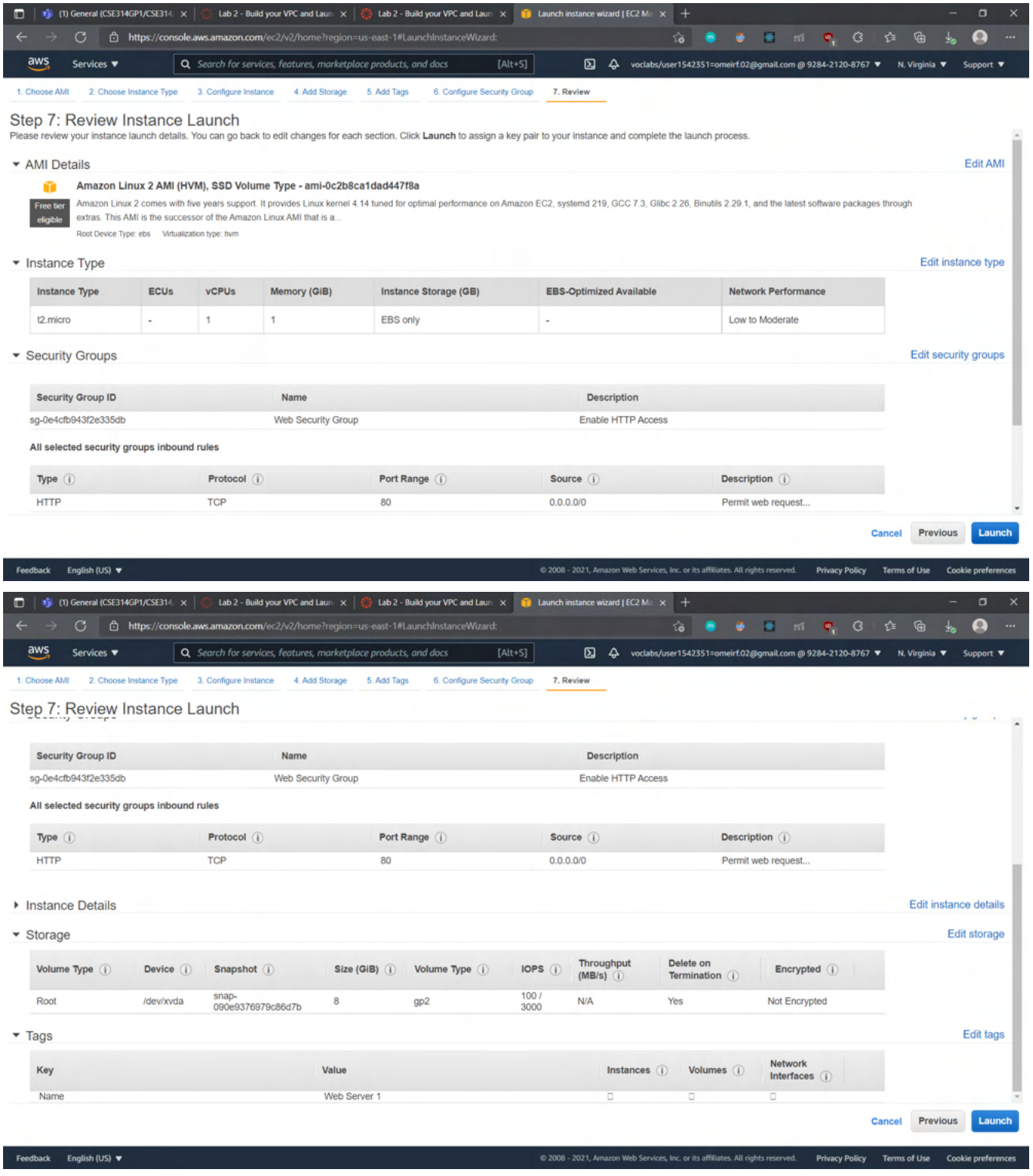
47. Select **Web Security Group**.
    This is the security group you created in the previous task. It will permit HTTP access to the instance.
48. Click **Review and Launch**



49. When prompted with a *warning* that you will not be able to connect to the instance through port 22, click **Continue**

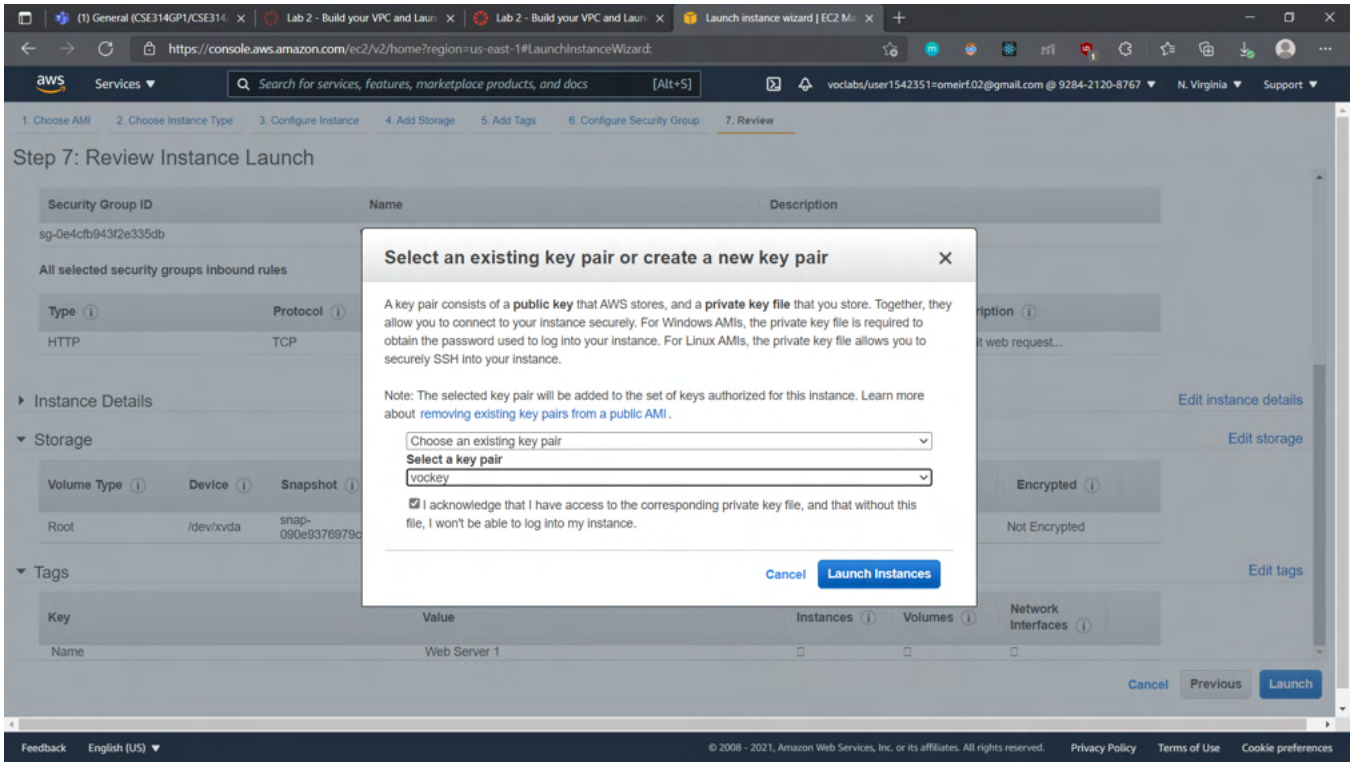


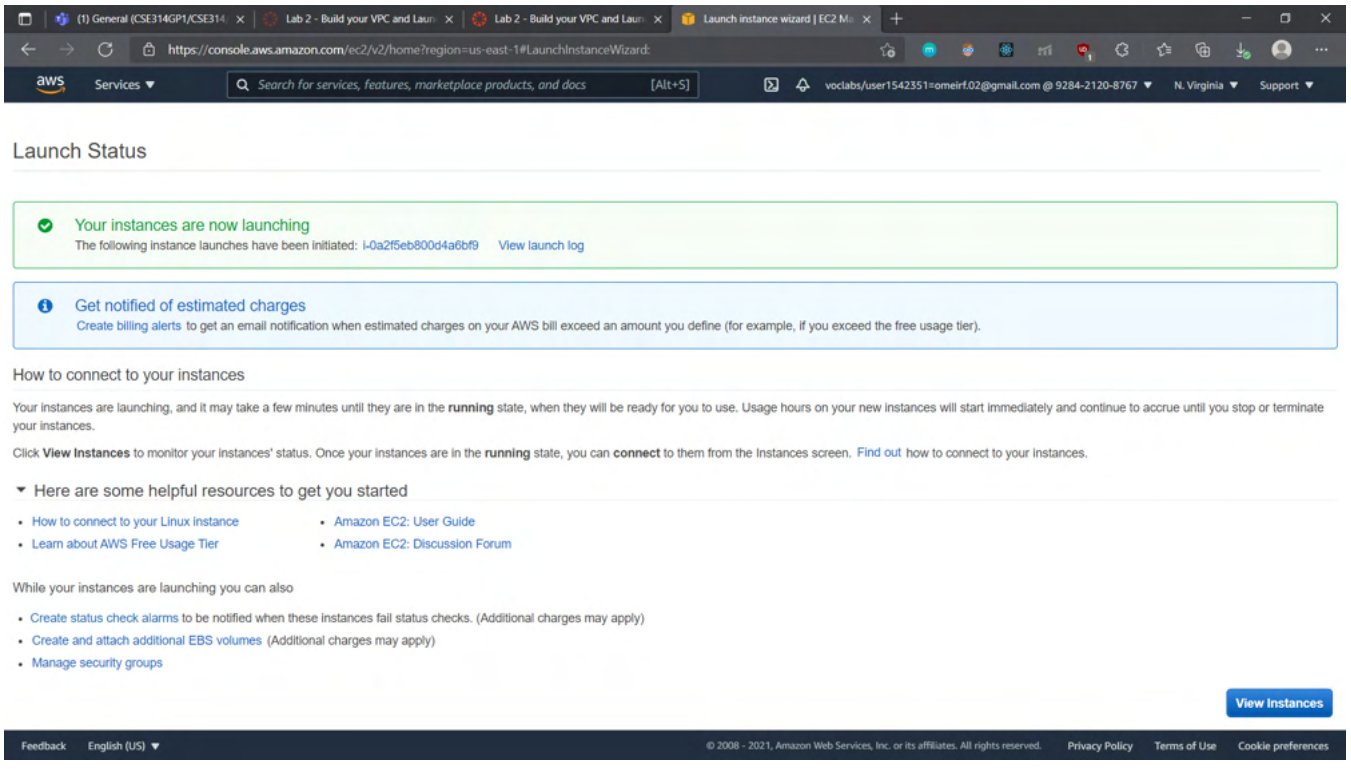
50. Review the instance information and click **Launch**



51. In the **Select an existing keypair** dialog, select **I acknowledge...**.

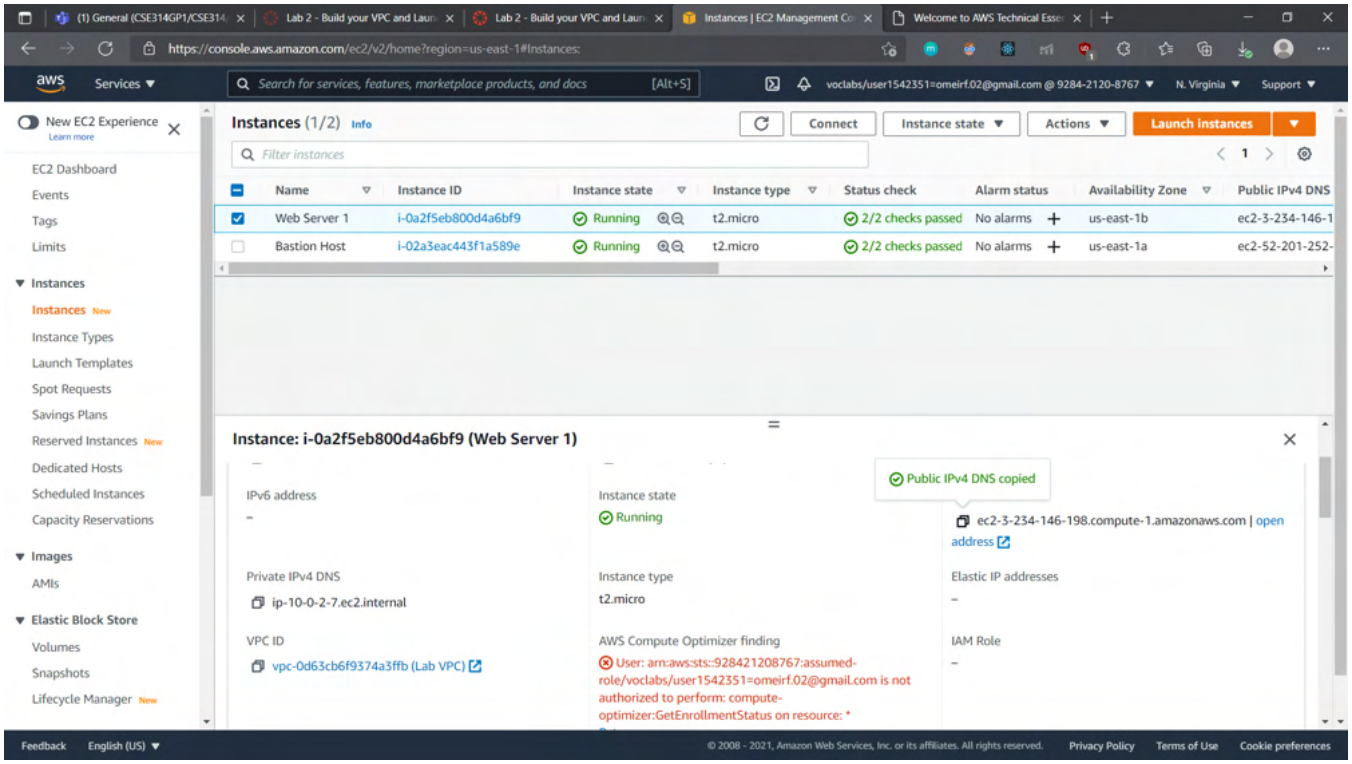
52. Click **Launch Instances** and then click **View Instances**



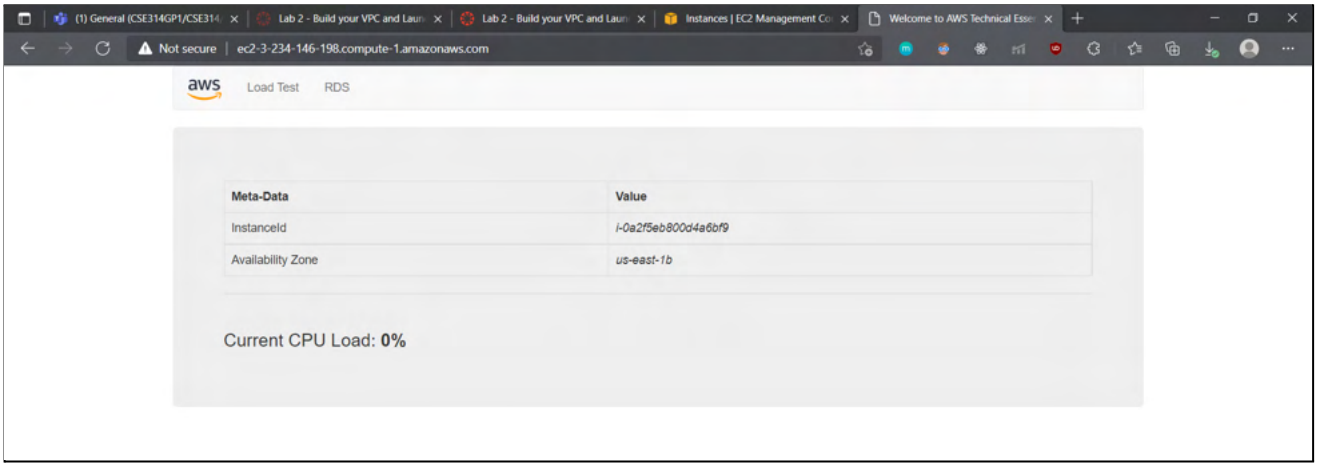


53. Wait until **Web Server 1** shows *2/2 checks passed* in the **Status Checks** column.
This may take a few minutes. Click refresh in the top-right every 30 seconds for updates.
You will now connect to the web server running on the EC2 instance.

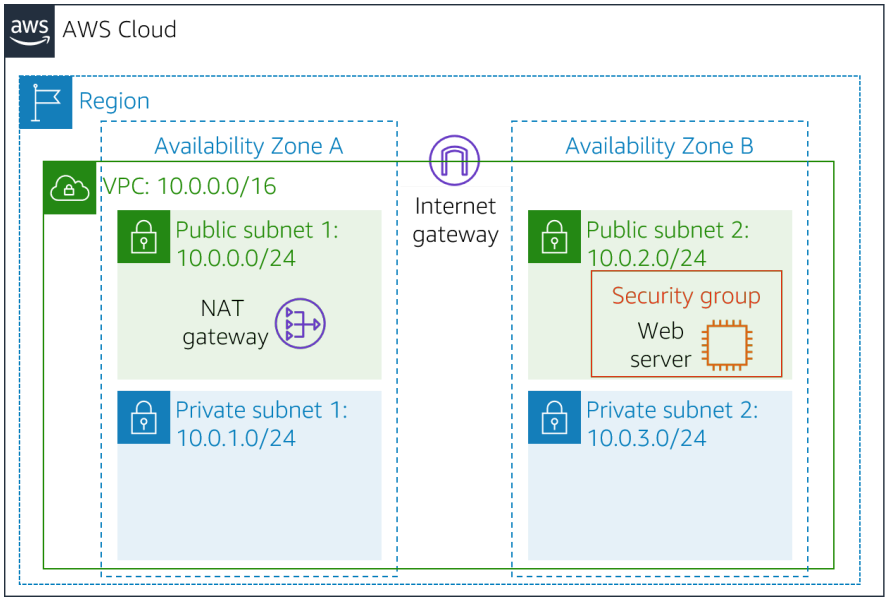
54. Copy the **Public DNS (IPv4)** value shown in the **Description** tab at the bottom of the page.



55. Open a new web browser tab, paste the **Public DNS** value and press Enter.
You should see a web page displaying the AWS logo and instance meta-data values.



The complete architecture you deployed is:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	NAT gateway

Practical 3

AIM

Working with EBS (Elastic Block Storage)

THEORY

This lab focuses on Amazon Elastic Block Store (Amazon EBS), a key underlying storage mechanism for Amazon EC2 instances. In this lab, you will learn how to create an Amazon EBS volume, attach it to an instance, apply a file system to the volume, and then take a snapshot backup.

What is Amazon Elastic Block Store?

Amazon Elastic Block Store (Amazon EBS) offers persistent storage for Amazon EC2 instances. Amazon EBS volumes are network-attached and persist independently from the life of an instance. Amazon EBS volumes are highly available, highly reliable volumes that can be leveraged as an Amazon EC2 instances boot partition or attached to a running Amazon EC2 instance as a standard block device.

When used as a boot partition, Amazon EC2 instances can be stopped and subsequently restarted, enabling you to pay only for the storage resources used while maintaining your instance's state. Amazon EBS volumes offer greatly improved durability over local Amazon EC2 instance stores because Amazon EBS volumes are automatically replicated on the backend (in a single Availability Zone).

For those wanting even more durability, Amazon EBS provides the ability to create point-in-time consistent snapshots of your volumes that are then stored in Amazon Simple Storage Service (Amazon S3) and automatically replicated across multiple Availability Zones. These snapshots can be used as the starting point for new Amazon EBS volumes and can protect your data for long-term durability. You can also easily share these snapshots with co-workers and other AWS developers.

This lab guide explains basic concepts of Amazon EBS in a step-by-step fashion. However, it can only give a brief overview of Amazon EBS concepts. For further information, see the Amazon EBS documentation.

Amazon EBS Volume Features

Amazon EBS volumes deliver the following features:

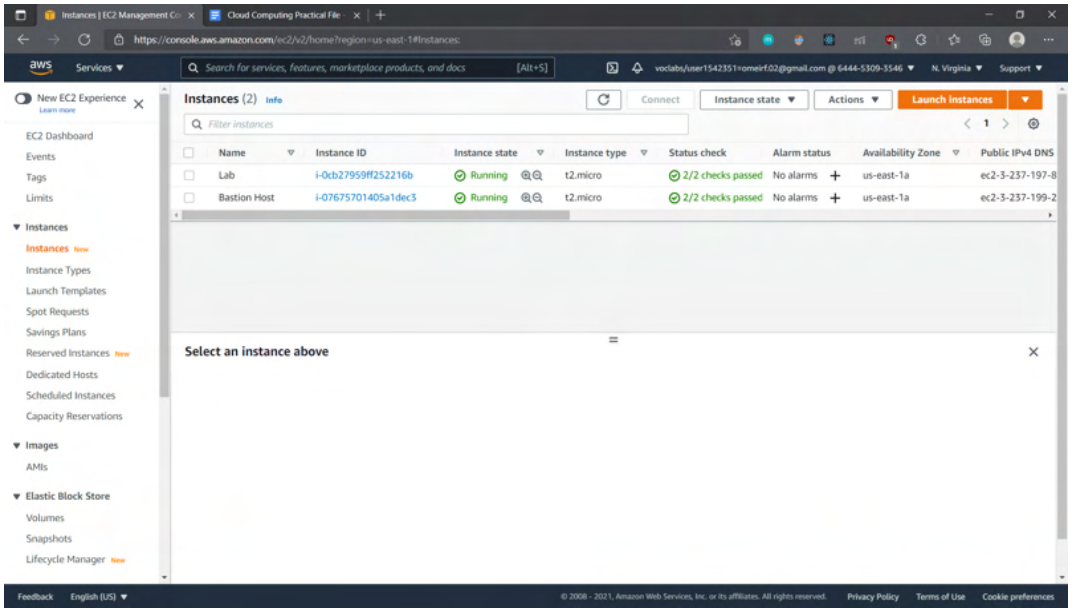
- **Persistent storage:** Volume lifetime is independent of any particular Amazon EC2 instance.

- **General purpose:** Amazon EBS volumes are raw, unformatted block devices that can be used from any operating system.
- **High performance:** Amazon EBS volumes are equal to or better than local Amazon EC2 drives.
- **High reliability:** Amazon EBS volumes have built-in redundancy within an Availability Zone.
- **Designed for resiliency:** The AFR (Annual Failure Rate) of Amazon EBS is between 0.1% and 1%.
- **Variable size:** Volume sizes range from 1 GB to 16 TB.
- **Easy to use:** Amazon EBS volumes can be easily created, attached, backed up, restored, and deleted.

PROCEDURE

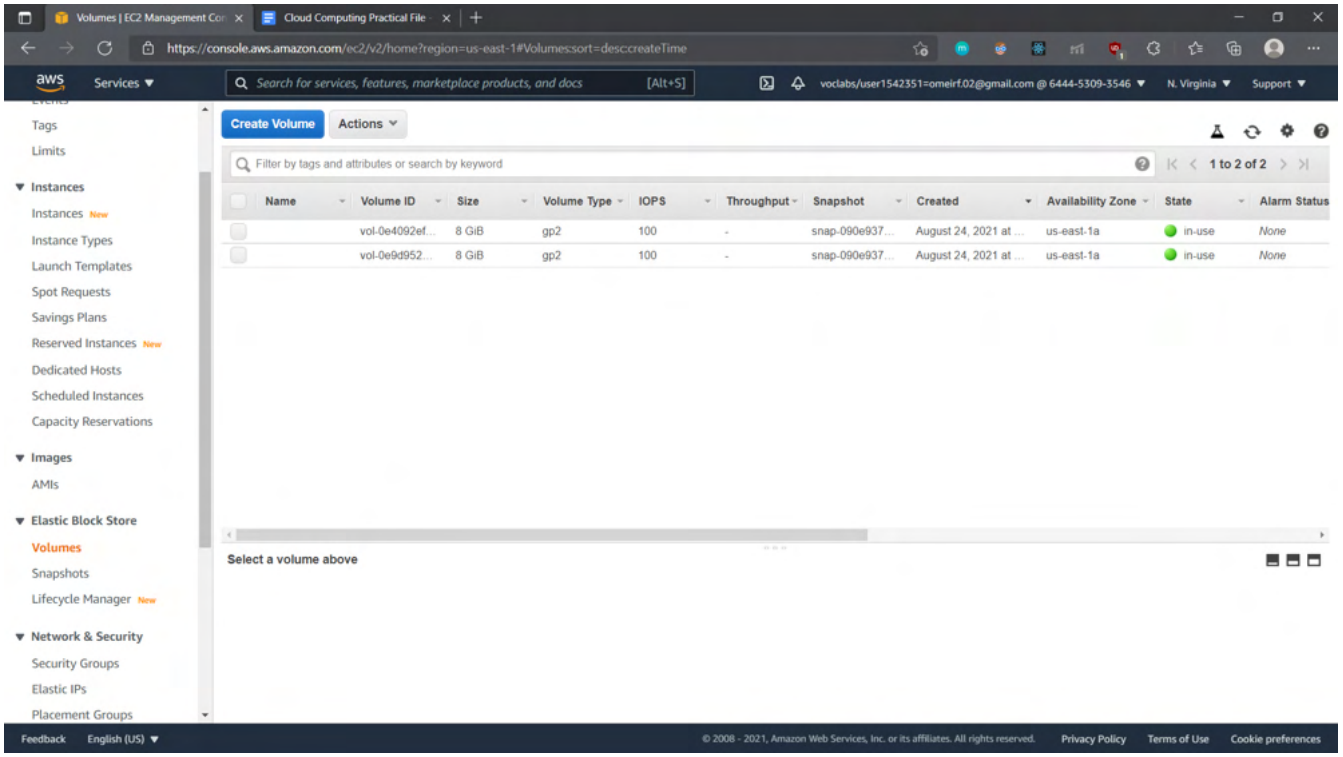
Task 1: Create a New EBS Volume

1. In the AWS Management Console, on the Services menu, click EC2.
2. In the left navigation pane, click Instances.
   An Amazon EC2 instance named Lab has already been launched for your lab.
3. Note the Availability Zone of the instance. It will look similar to us-west-2a.

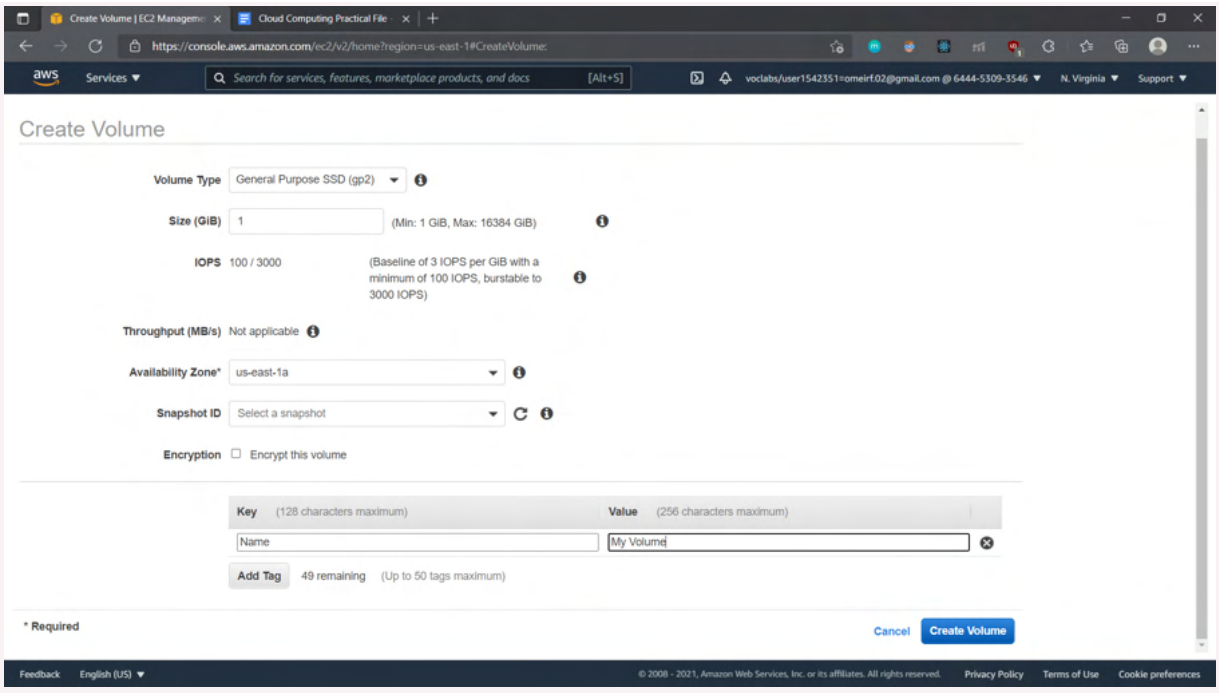


4. In the left navigation pane, click Volumes.
   You will see an existing volume that is being used by the Amazon EC2 instance.
   This volume has a size of 8 GiB, which makes it easy to distinguish from the volume you will

create next, which will be 1 GiB in size.

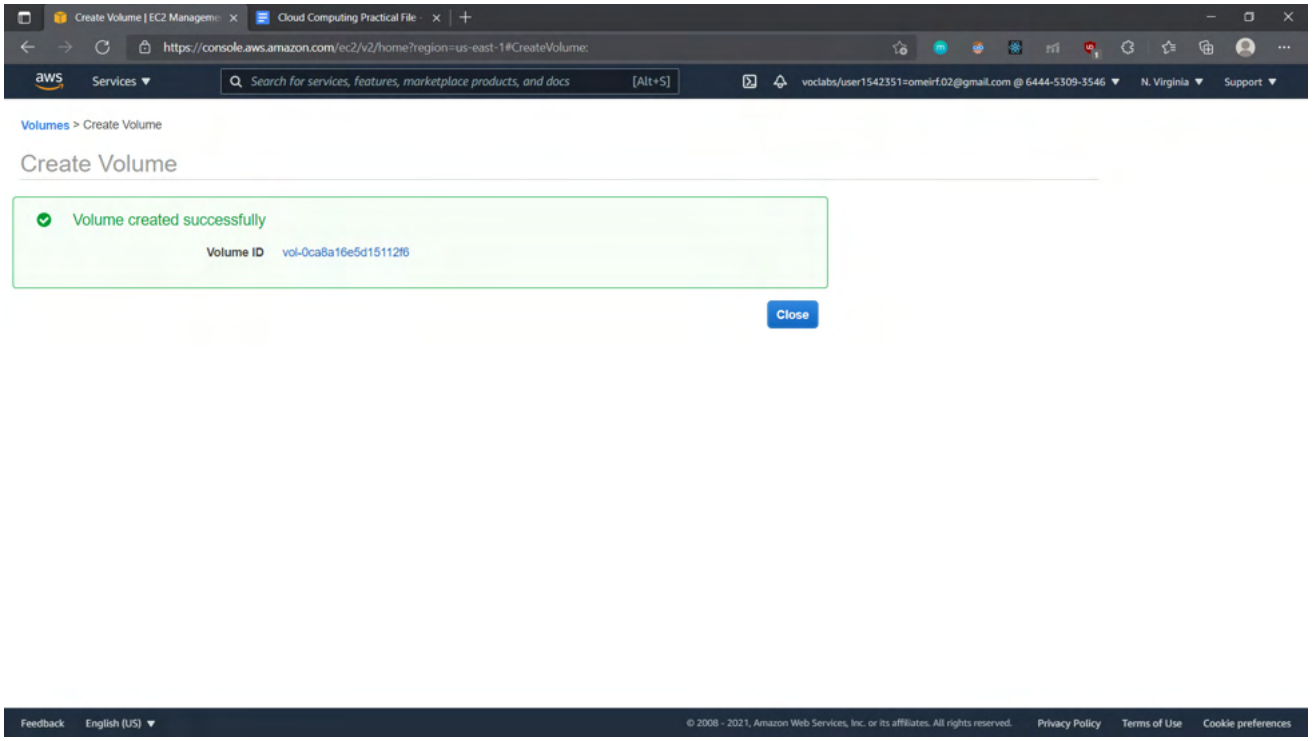


5. Click **Create Volume** then configure:
    - **Volume Type:** *General Purpose SSD (gp2)*
    - **Size (GiB):** `1`. **NOTE**: You may be restricted from creating large volumes.
    - **Availability Zone:** Select the same availability zone as your EC2 instance.
    - Click **Add Tag**
    - In the Tag Editor, enter:
        - **Key:** `Name`
        - **Value:** `My Volume`



6. Click **Create Volume** then click **Close**
   Your new volume will appear in the list, and will move from the *creating* state to the *available*

state. You may need to click **refresh** to see your new volume.

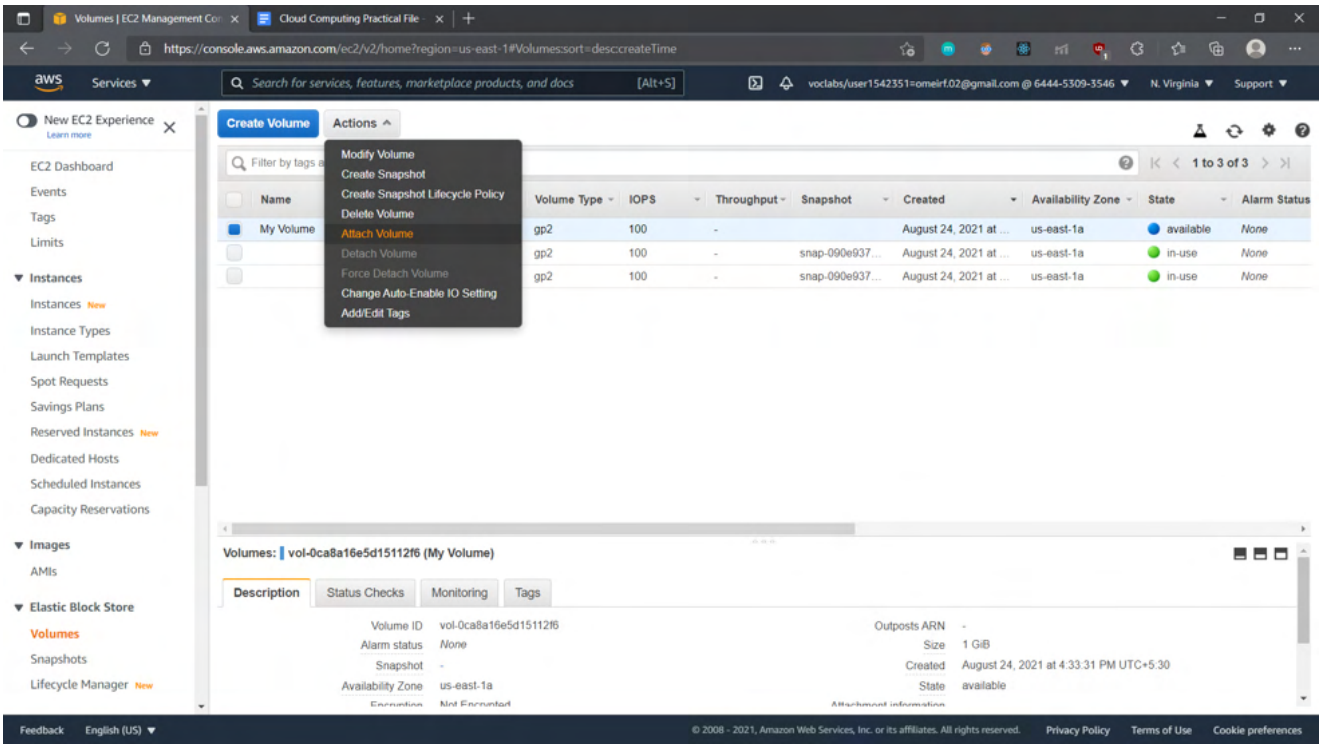


Task 2: Attach the Volume to an Instance

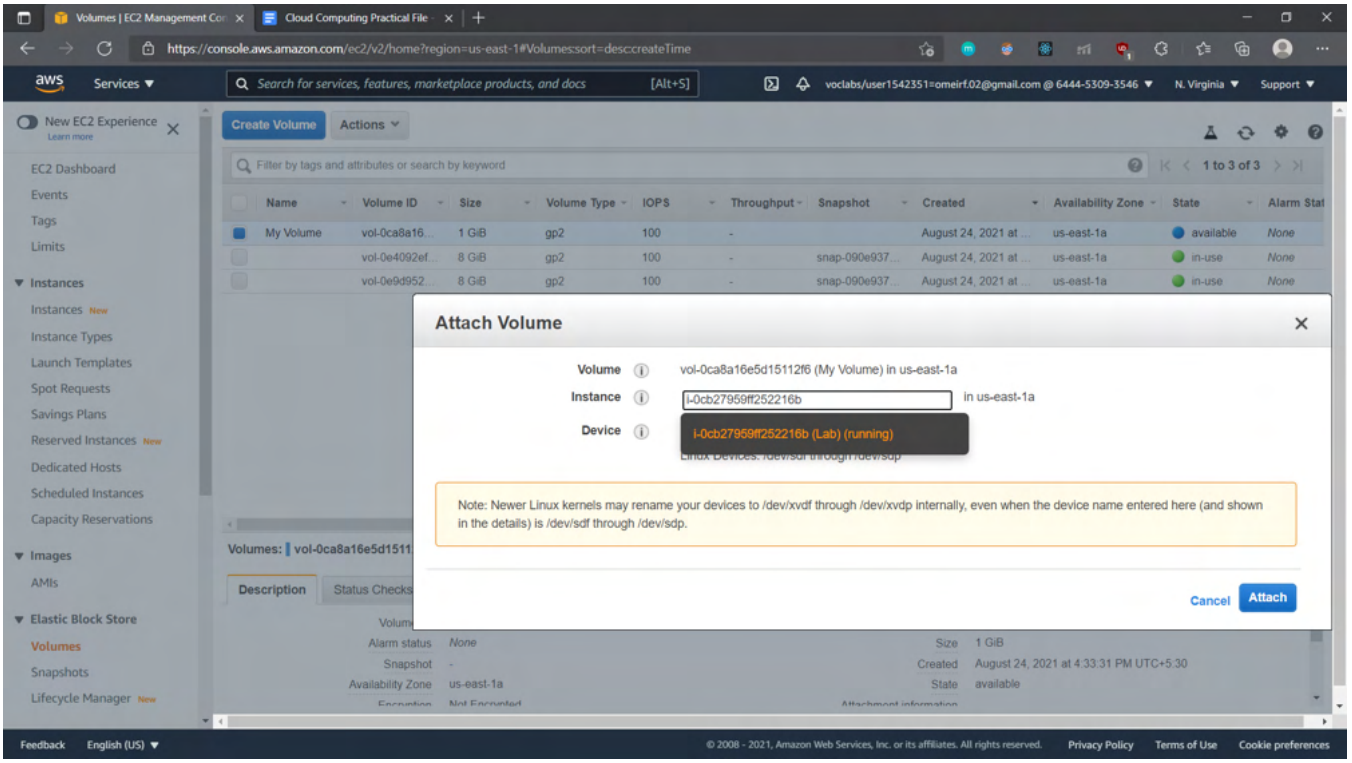
You can now attach your new volume to the Amazon EC2 instance.

7. Select **My Volume**.

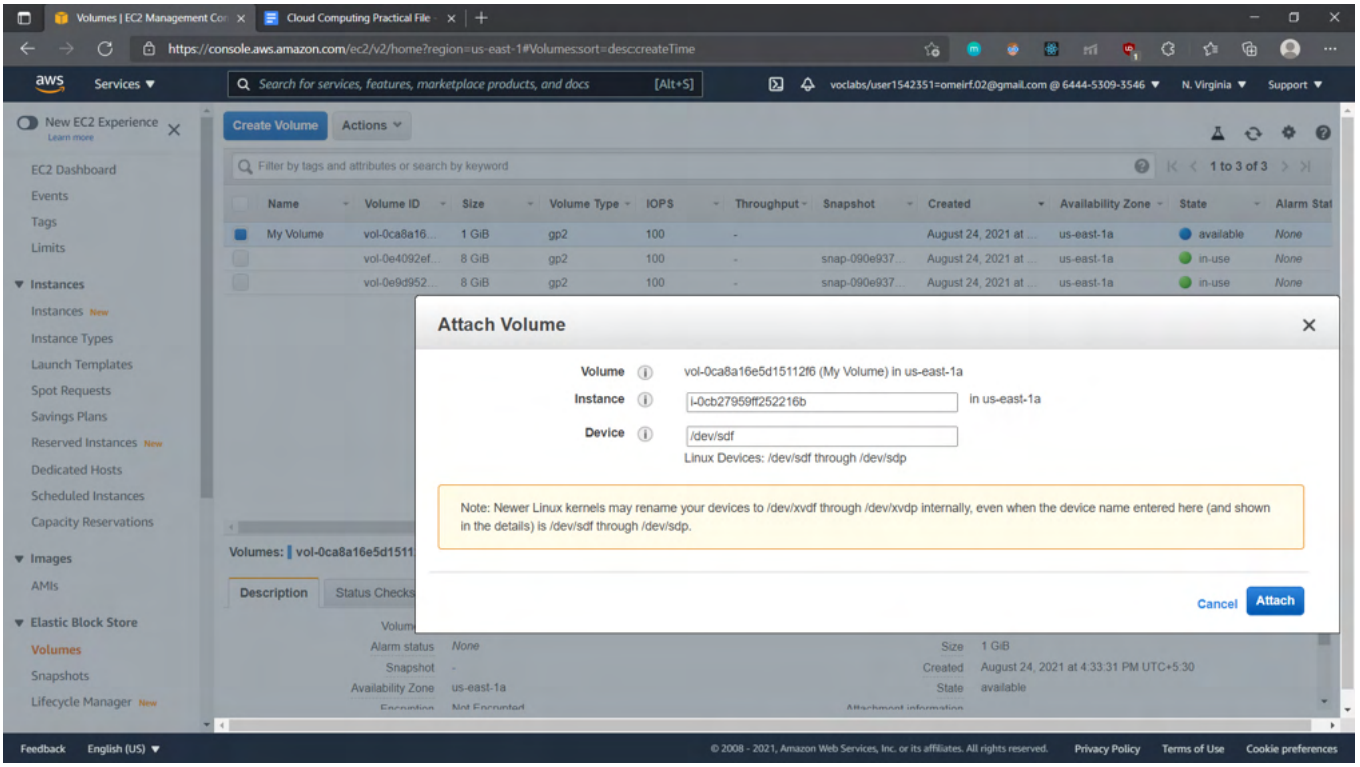
8. In the **Actions** menu, click **Attach Volume**.



9. Click in the **Instance** field, then select the instance that appears (Lab).



Note that the **Device** field is set to */dev/sdf*. You will use this device identifier in a later task.



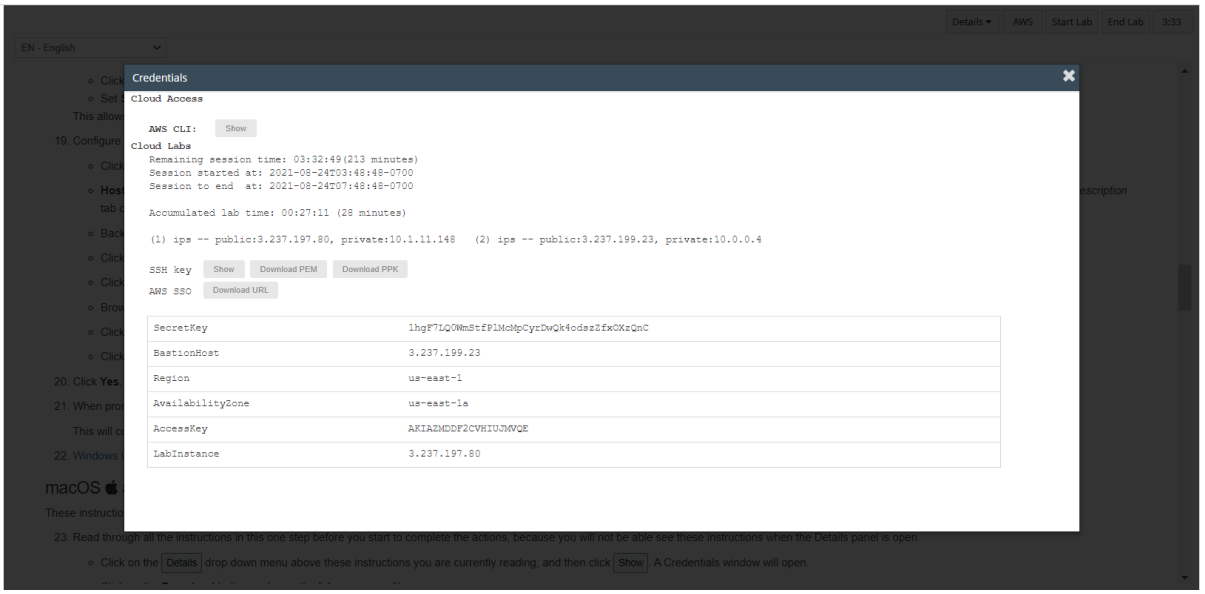
10. Click **Attach** The volume state is now *in-use*.

Task 3: Connect to Your Amazon EC2 Instance

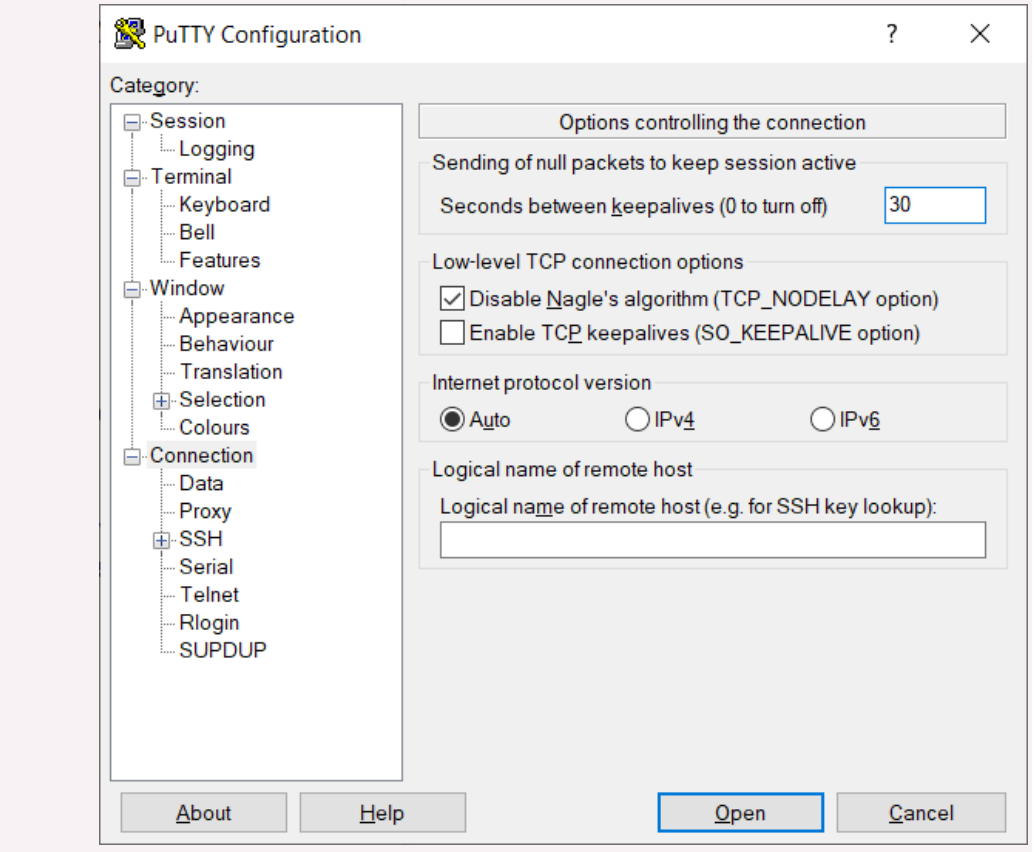
Windows Users: Using SSH to Connect

These instructions are for Windows users only.

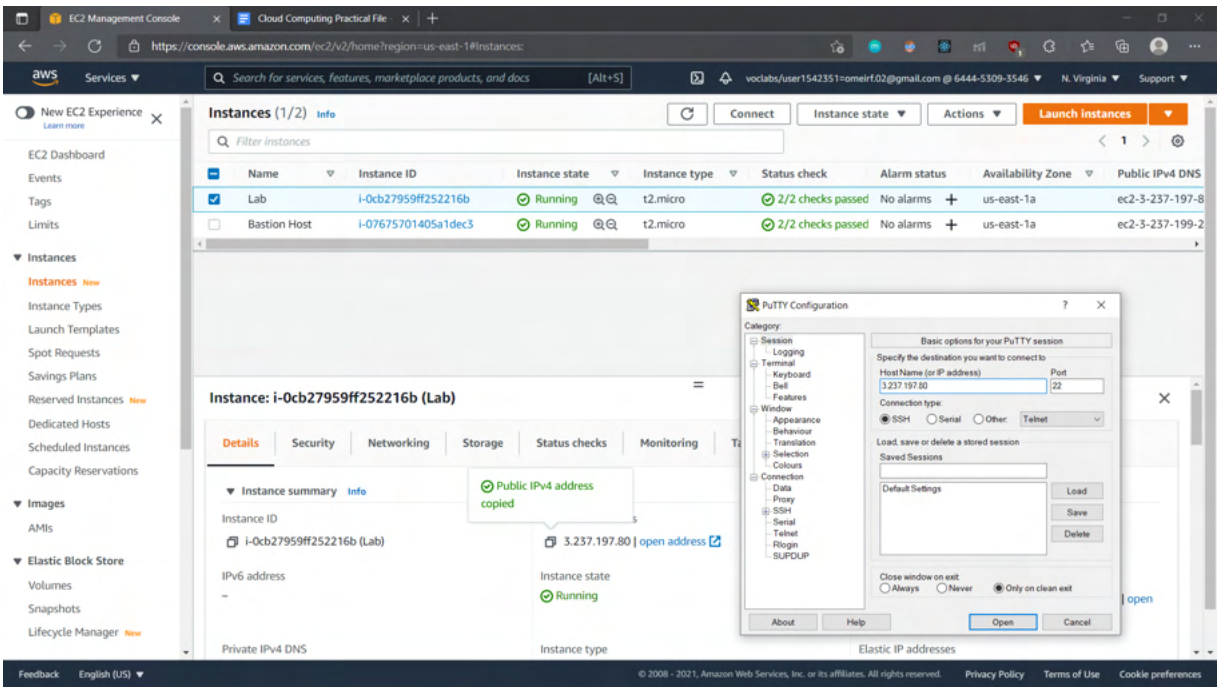
1. Read through the three bullet points in this step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.
    a. Click on the Details drop down menu above these instructions you are currently reading, and then click Show. A Credentials window will open.



    b. Click on the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser will save it to the Downloads directory.
    c. Then exit the Details panel by clicking on the **X**.
2. Download needed software.
    a. You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, download it here.
3. Open **putty.exe**
4. Configure PuTTY to not timeout:
    a. Click **Connection**
    b. Set **Seconds between keepalives** to `30`



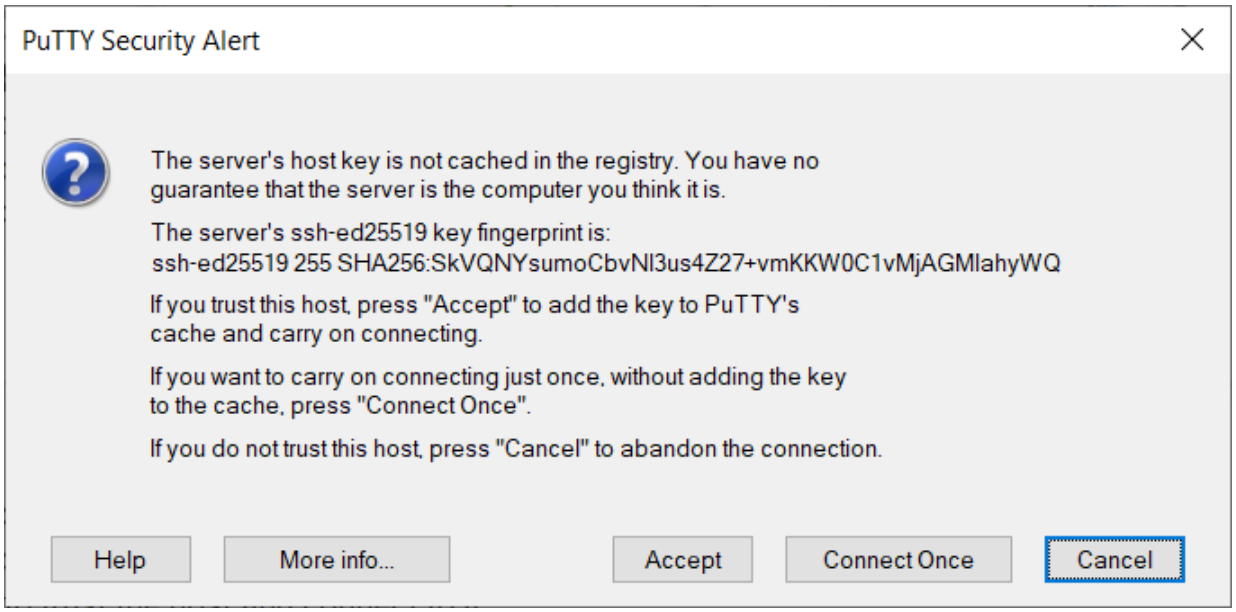
5. This allows you to keep the PuTTY session open for a longer period of time.
6. Configure your PuTTY session:
    a. Click **Session**
    b. **Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the instance. To find it, return to the EC2 Console and click on **Instances**. Check the box next to the instance and in the *Description* tab copy the **IPv4 Public IP** value.



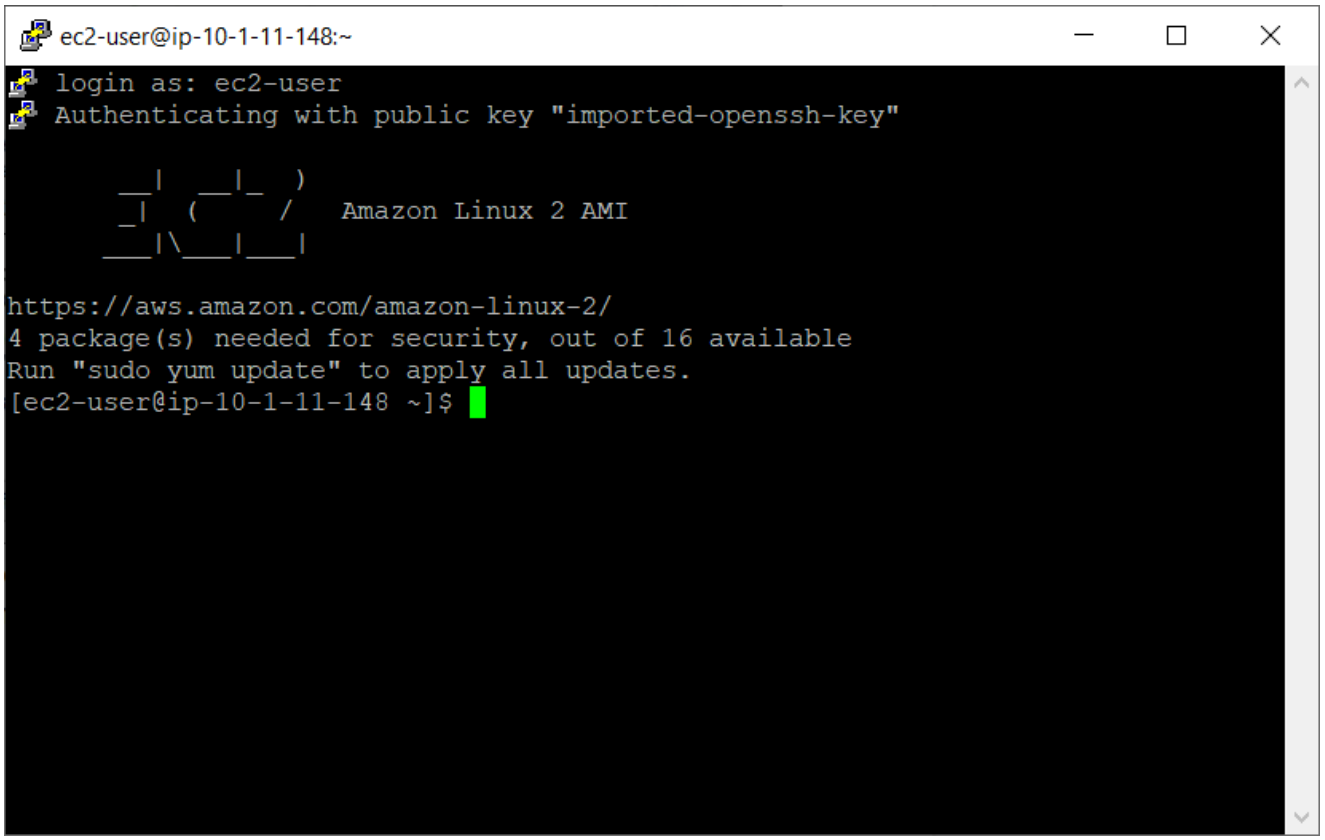
    c. Back in PuTTy, in the **Connection** list, expand **SSH**
    d. Click **Auth** (don't expand it)
    e. Click **Browse**
    f. Browse to and select the labsuser.ppk file that you downloaded

g. Click **Open** to select it
h. Click **Open**

7. Click **Yes**, to trust the host and connect to it.



8. When prompted **login as**, enter: `ec2-user`
This will connect you to the EC2 instance.

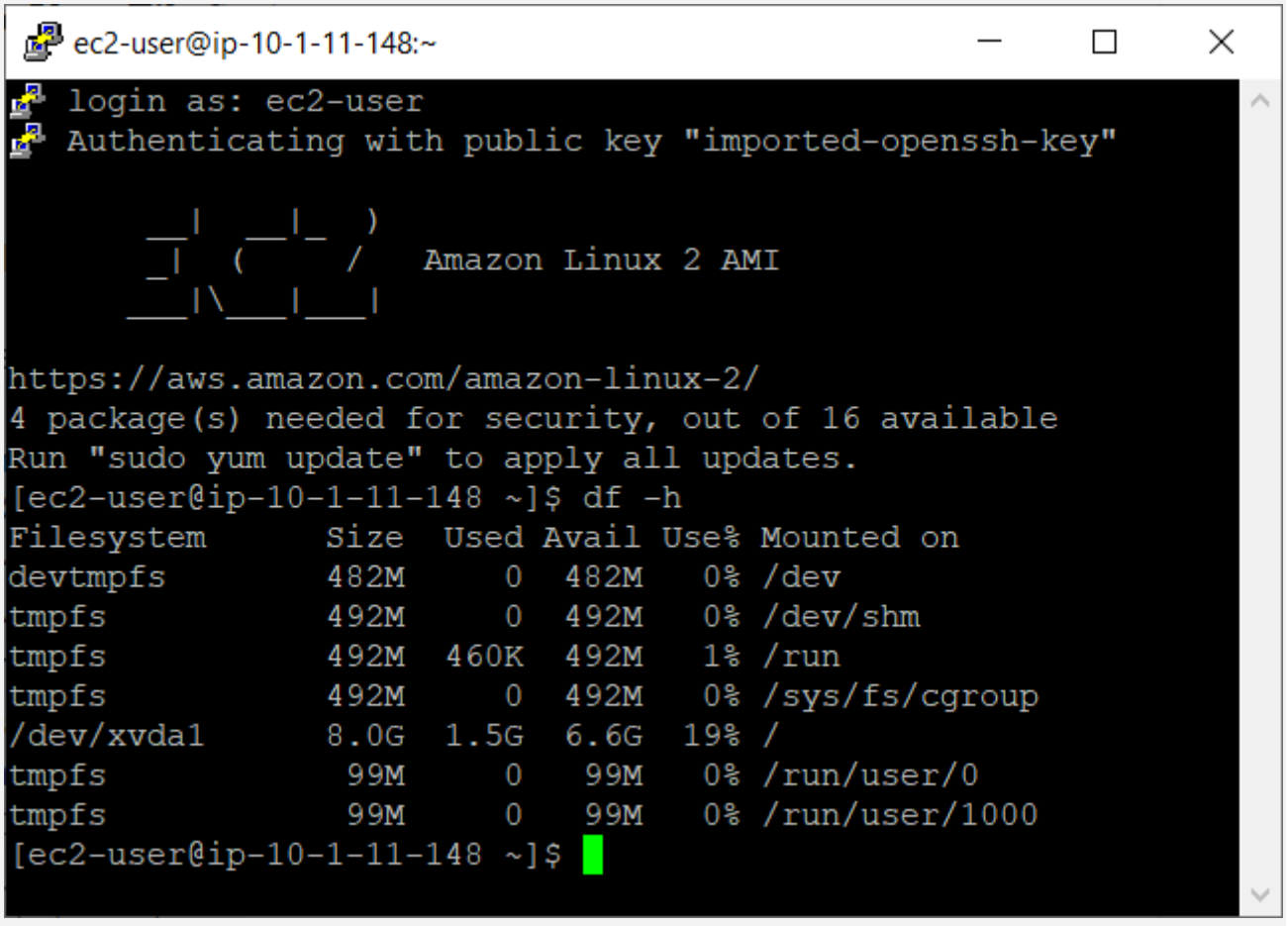


Task 4: Create and Configure Your File System

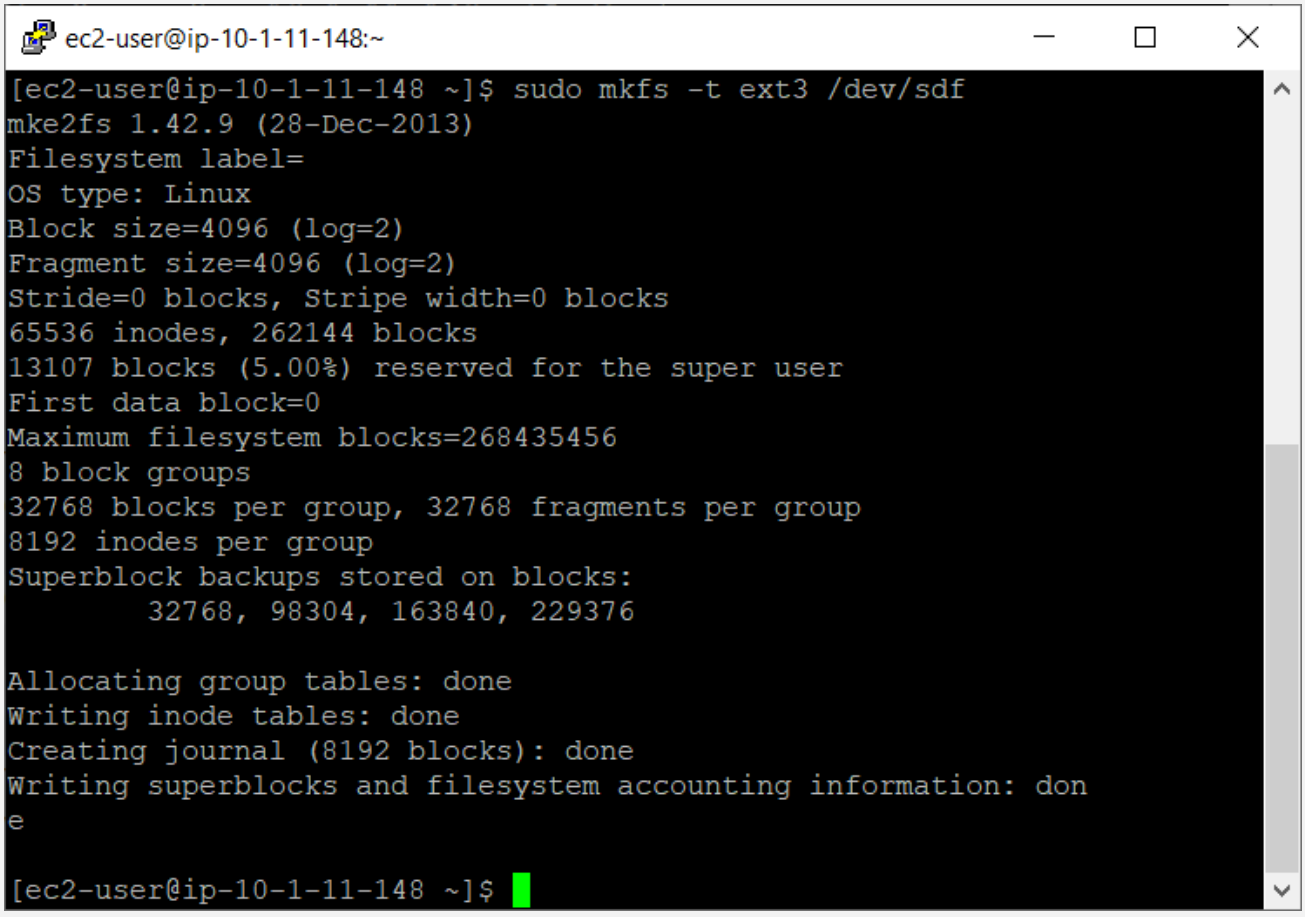
In this task, you will add the new volume to a Linux instance as an ext3 file system under the /mnt/data-store mount point.

If you are using PuTTY, you can paste text by right-clicking in the PuTTY window.

1. View the storage available on your instance:
   `df -h`
   You should see output similar to:
   ```
   Filesystem      Size  Used Avail Use% Mounted on
   devtmpfs        488M   60K  488M   1% /dev
   tmpfs           497M     0  497M   0% /dev/shm
   /dev/xvda1      7.8G  982M  6.7G  13% /
   ```



   This is showing the original 8GB disk volume. Your new volume is not yet shown.
2. Create an ext3 file system on the new volume:
   `sudo mkfs -t ext3 /dev/sdf`



3. Create a directory for mounting the new storage volume:
   `sudo mkdir /mnt/data-store`
4. Mount the new volume:
   `sudo mount /dev/sdf /mnt/data-store`
   To configure the Linux instance to mount this volume whenever the instance is started, you will need to add a line to */etc/fstab*.
   `echo "/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab`
5. View the configuration file to see the setting on the last line:
   `cat /etc/fstab`
6. View the available storage again:
   `df -h`
   The output will now contain an additional line - */dev/xvdf*:
   ```
   Filesystem      Size  Used Avail Use% Mounted on
   devtmpfs        488M   60K  488M   1% /dev
   tmpfs           497M     0  497M   0% /dev/shm
   /dev/xvda1      7.8G  982M  6.7G  13% /
   /dev/xvdf       976M  1.3M  924M   1% /mnt/data-store
   ```
7. On your mounted volume, create a file and add some text to it.
   `sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"`
8. Verify that the text has been written to your volume.
   `cat /mnt/data-store/file.txt`

```
ec2-user@ip-10-1-11-148:~

[ec2-user@ip-10-1-11-148 ~]$ sudo mkdir /mnt/data-store
[ec2-user@ip-10-1-11-148 ~]$ sudo mount /dev/sdf /mnt/data-store
[ec2-user@ip-10-1-11-148 ~]$ echo "/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2" | sudo tee -a /etc/fstab
/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2
[ec2-user@ip-10-1-11-148 ~]$ cat /etc/fstab
#
UUID=04b92f2f-4366-4687-868b-7c403cc59901     /          xfs    defaults,noatime  1   1
/dev/sdf   /mnt/data-store ext3 defaults,noatime 1 2
[ec2-user@ip-10-1-11-148 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        482M     0  482M   0% /dev
tmpfs           492M     0  492M   0% /dev/shm
tmpfs           492M  460K  492M   1% /run
tmpfs           492M     0  492M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.5G  6.6G  19% /
tmpfs            99M     0   99M   0% /run/user/0
tmpfs            99M     0   99M   0% /run/user/1000
/dev/xvdf       976M  1.3M  924M   1% /mnt/data-store
[ec2-user@ip-10-1-11-148 ~]$ sudo sh -c "echo some text has been written > /mnt/data-store/file.txt"
[ec2-user@ip-10-1-11-148 ~]$ cat /mnt/data-store/file.txt
some text has been written
[ec2-user@ip-10-1-11-148 ~]$
```

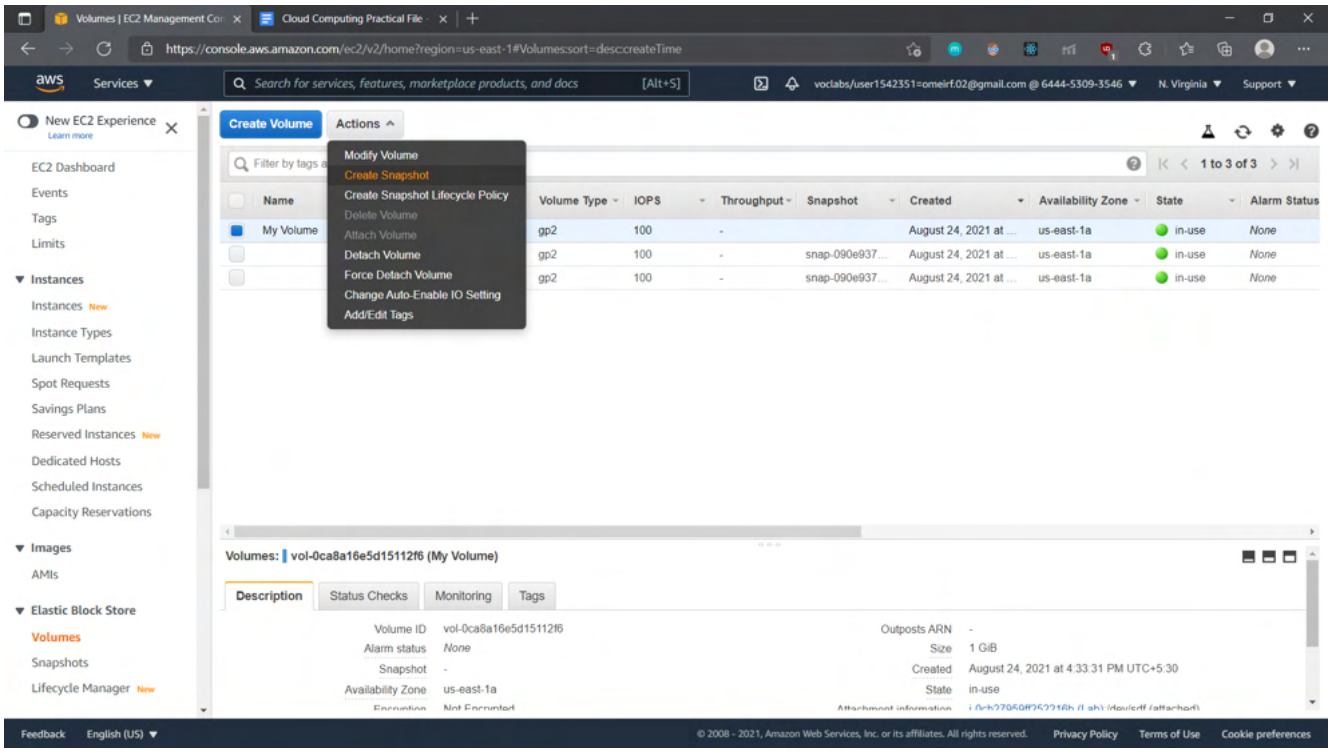
Task 5: Create an Amazon EBS Snapshot

In this task, you will create a snapshot of your EBS volume.

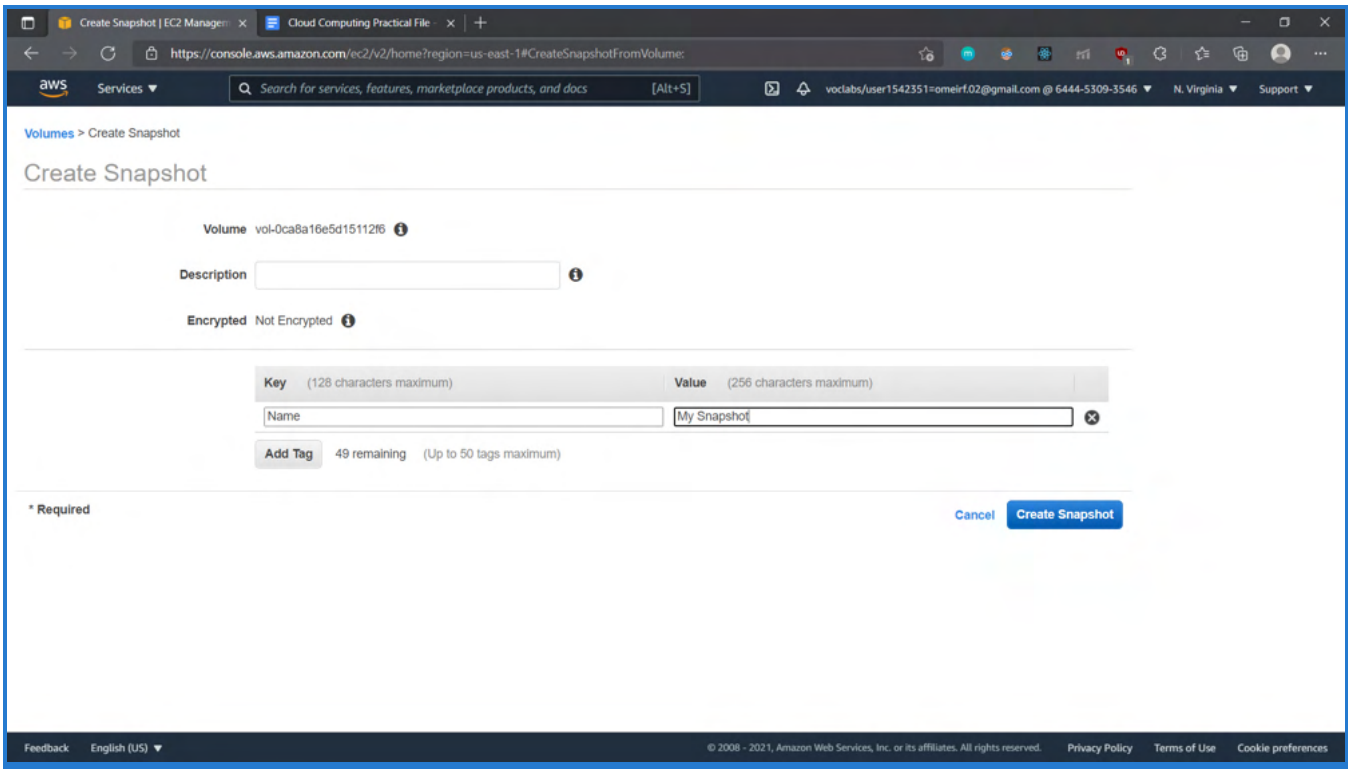
You can create any number of point-in-time, consistent snapshots from Amazon EBS volumes at any time. Amazon EBS snapshots are stored in Amazon S3 with high durability. New Amazon EBS volumes can be created out of snapshots for cloning or restoring backups. Amazon EBS snapshots can also be easily shared among AWS users or copied over AWS regions.

1. In the **AWS Management Console**, click on **Volumes** and select **My Volume**.

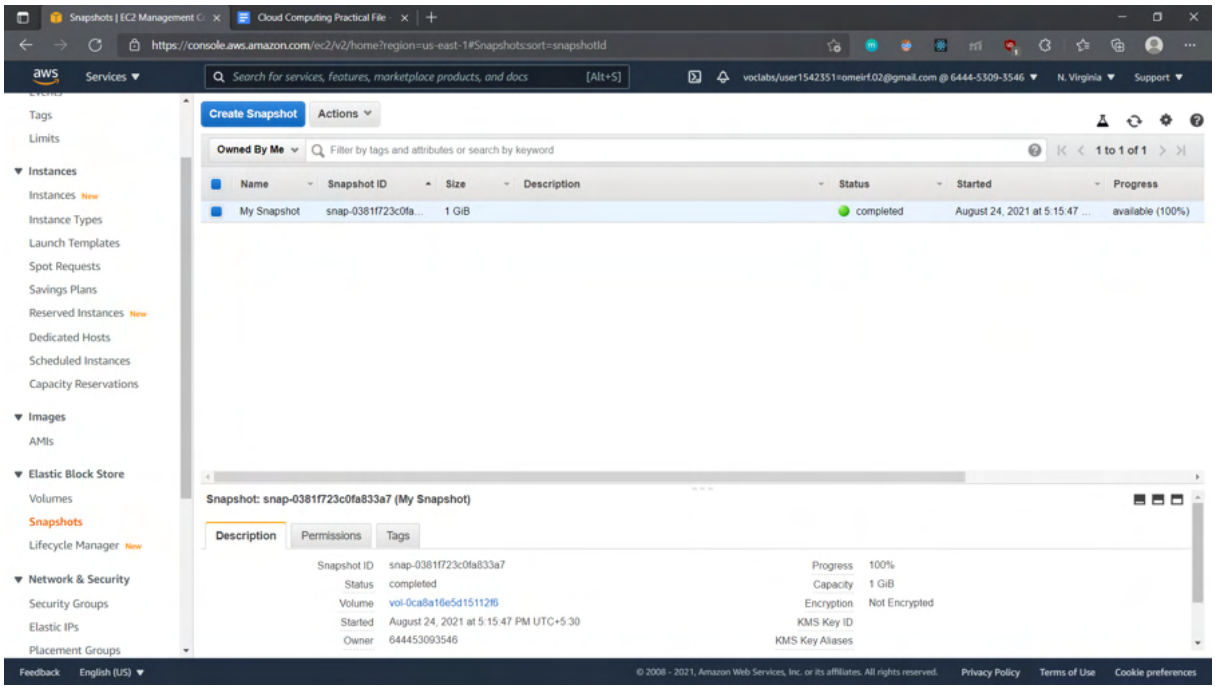
2. In the **Actions** menu, click **Create Snapshot**.



3. Click **Add Tag** then configure:
    a. **Key:** `Name`
    b. **Value:** `My Snapshot`
    c. Click **Create Snapshot** then click **Close**



4. Your snapshot will be listed in the **Snapshots** console.
5. In the left navigation pane, click **Snapshots**.
   Your snapshot is displayed. It will start with a state of *pending*, which means that the snapshot is being created. It will then change to a state of *completed*. Only used storage blocks are copied to snapshots, so empty blocks do not take any snapshot storage space.



6. In your remote SSH session, delete the file that you created on your volume.
   `sudo rm /mnt/data-store/file.txt`

7. Verify that the file has been deleted.
   `ls /mnt/data-store/`
   Your file has been deleted.

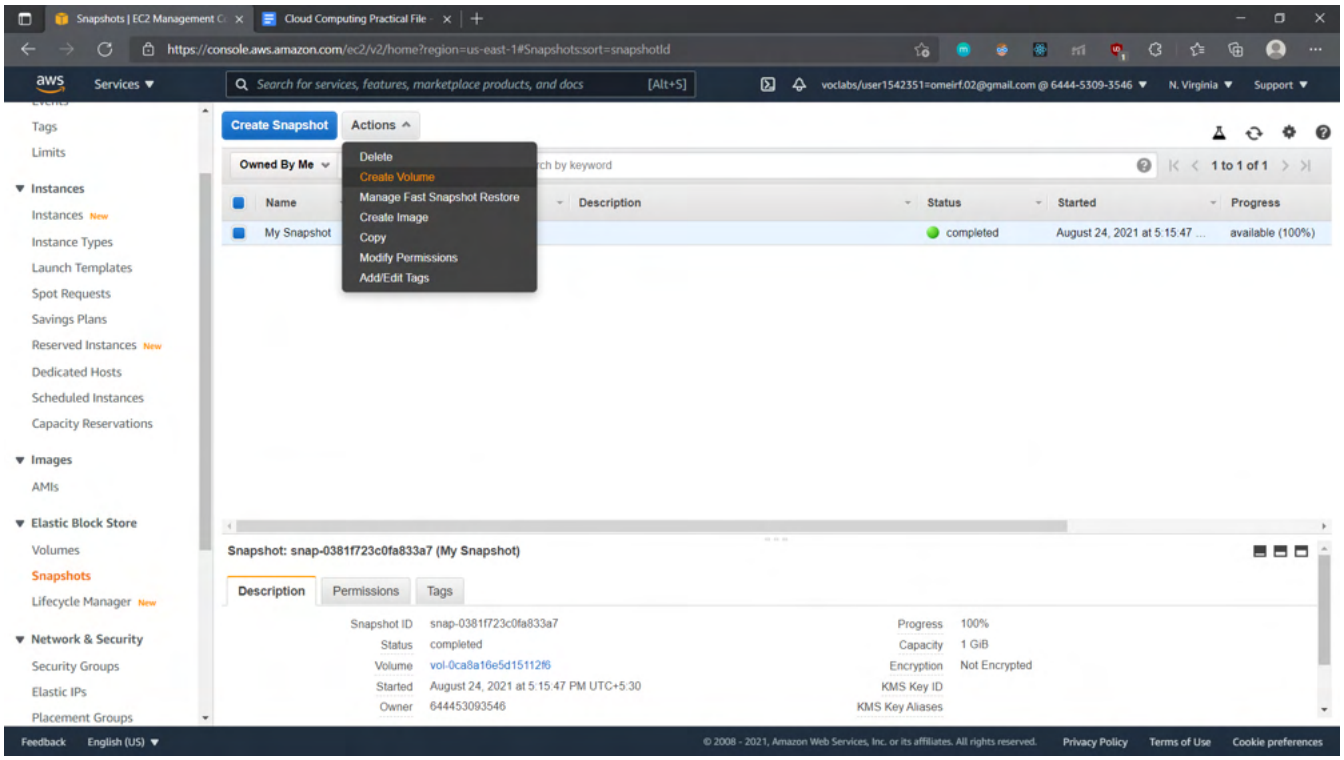
```
[ec2-user@ip-10-1-11-148 ~]$ sudo rm /mnt/data-store/file.txt
[ec2-user@ip-10-1-11-148 ~]$ ls /mnt/data-store/
lost+found
[ec2-user@ip-10-1-11-148 ~]$
```

Task 6: Restore the Amazon EBS Snapshot

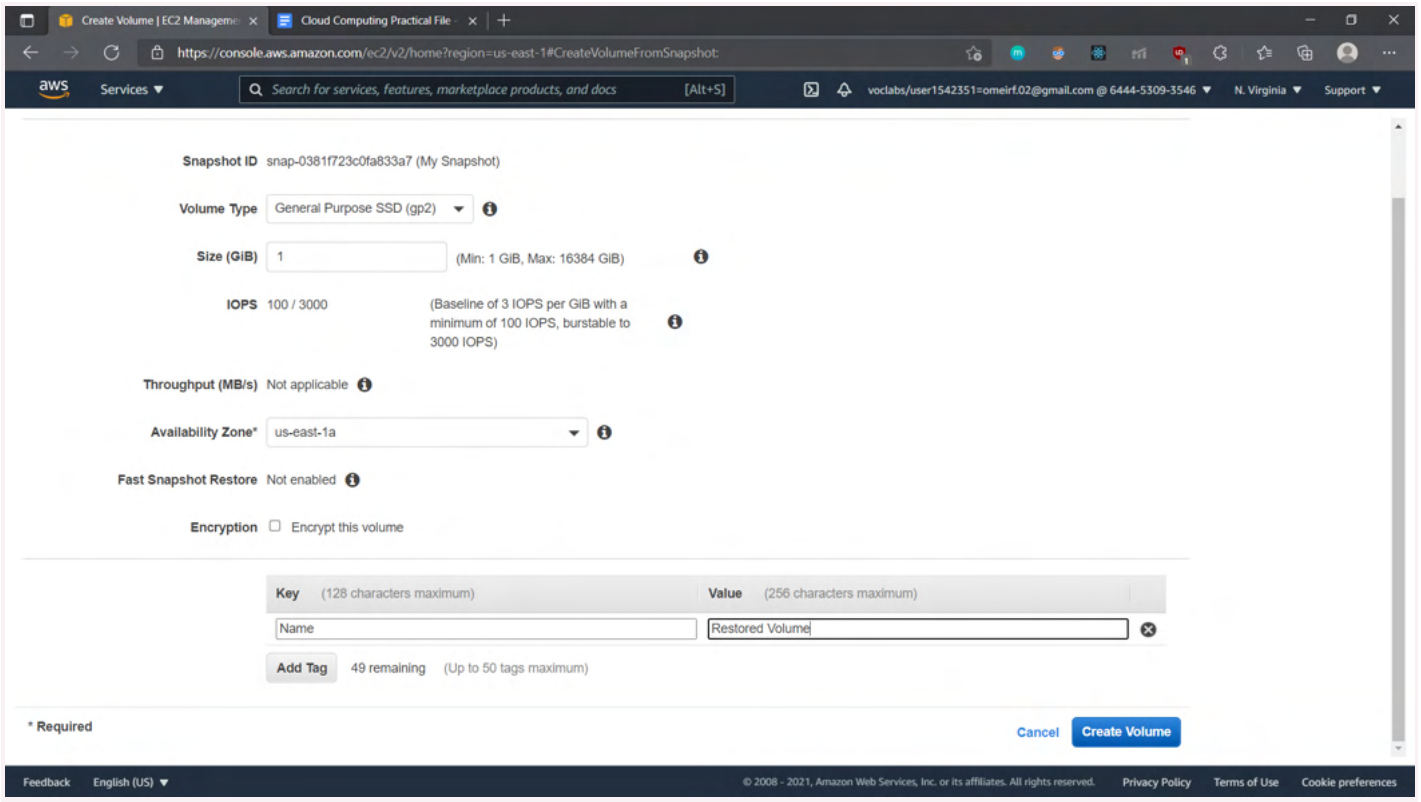
If you ever wish to retrieve data stored in a snapshot, you can **Restore** the snapshot to a new EBS volume.

Create a Volume Using Your Snapshot

1. In the **AWS Management Console**, select **My Snapshot**.
2. In the **Actions** menu, click **Create Volume**.



3. For **Availability Zone** Select the same availability zone that you used earlier.
4. Click **Add Tag** then configure:
   a. **Key:** `Name`
   b. **Value:** `Restored Volume`

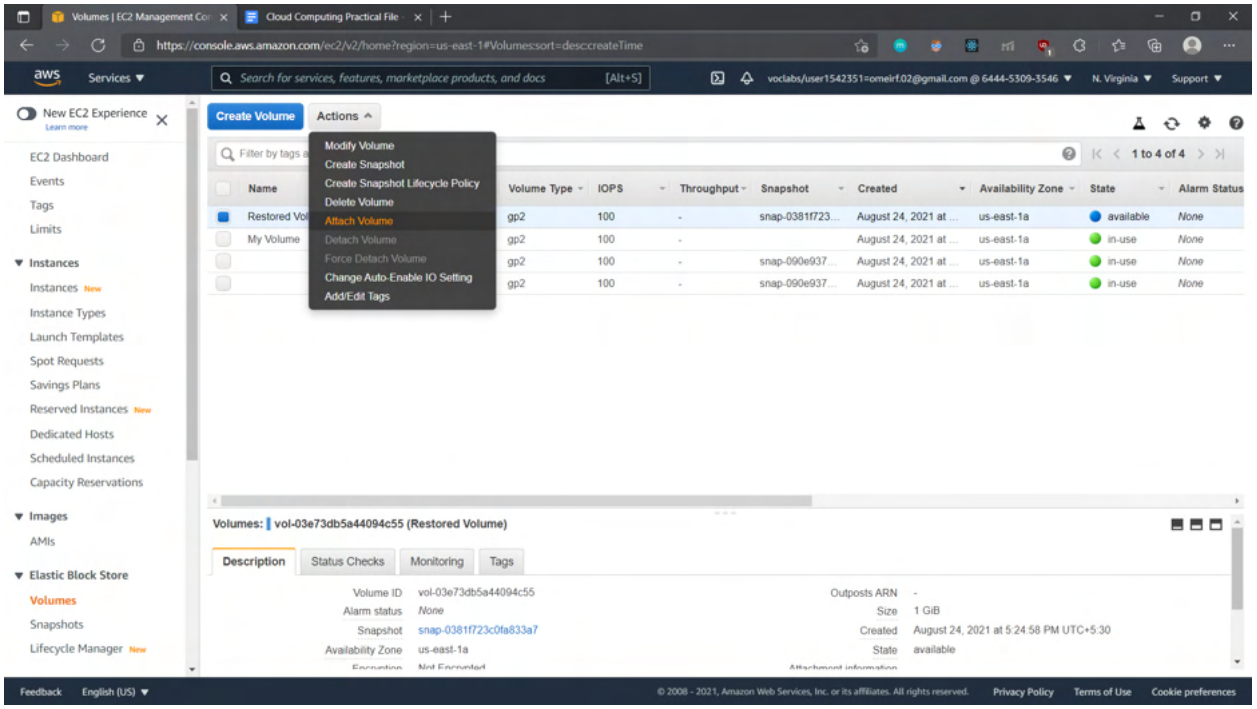


5. Click **Create Volume**
6. Click **Close**

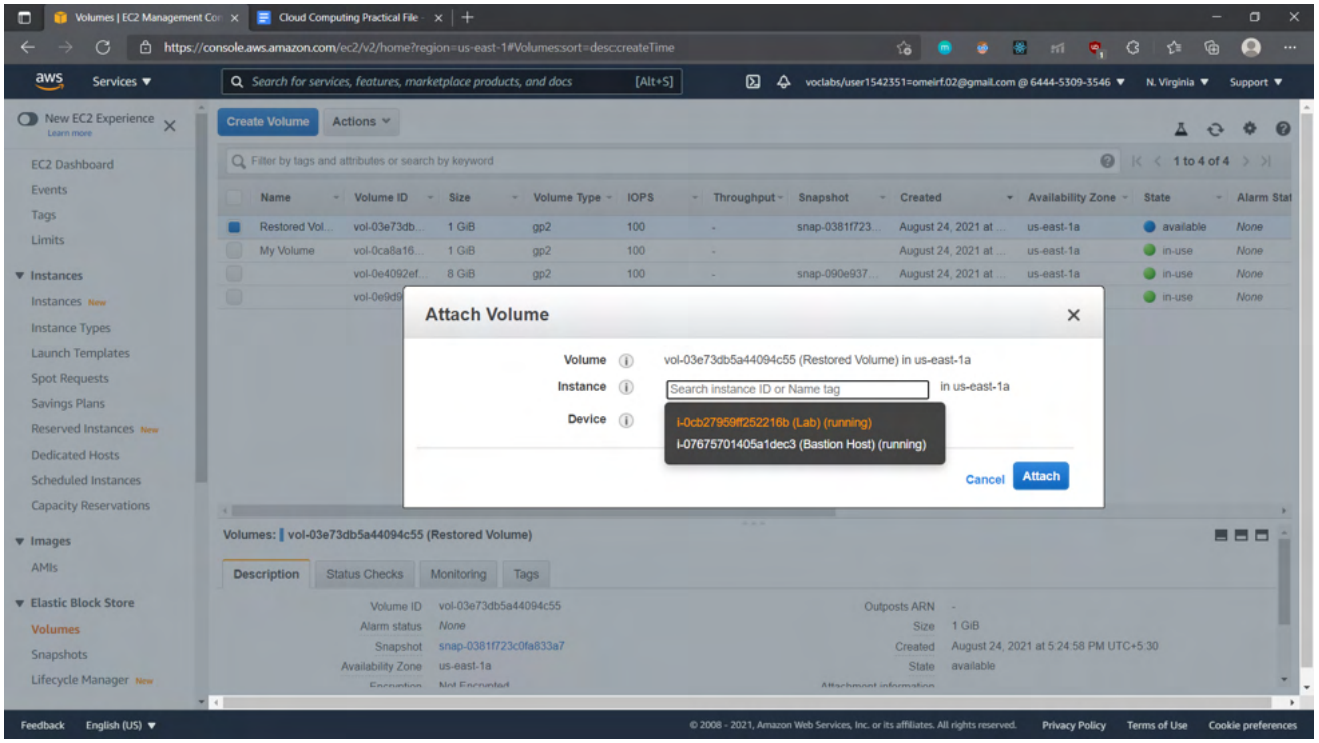
   When restoring a snapshot to a new volume, you can also modify the configuration, such as changing the volume type, size or Availability Zone.

Attach the Restored Volume to Your EC2 Instance

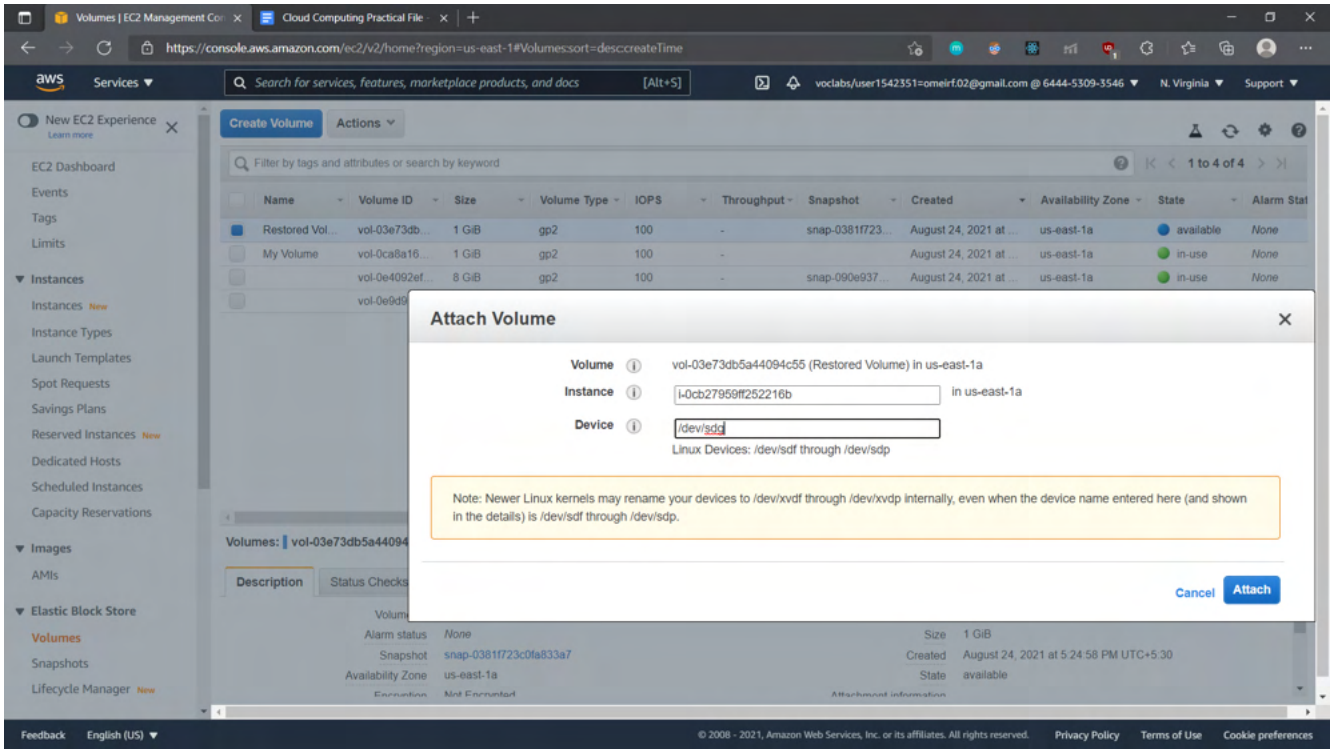
1. In the left navigation pane, click **Volumes**.
2. Select **Restored Volume**.
3. In the **Actions** menu, click **Attach Volume**.



4. Click in the **Instance** field, then select the instance that appears (Lab).



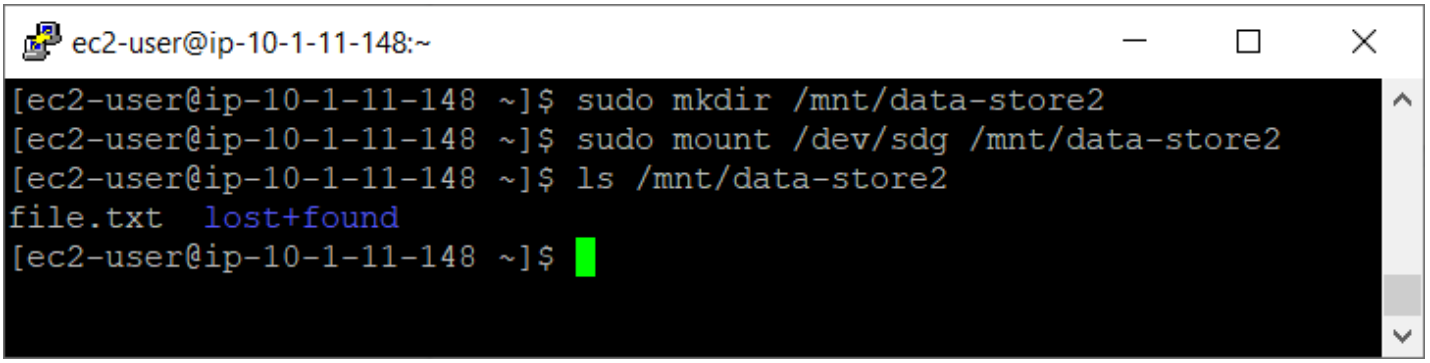
Note that the **Device** field is set to */dev/sdg*. You will use this device identifier in a later task.



5. Click **Attach**
   The volume state is now *in-use*.

Mount the Restored Volume

53. Create a directory for mounting the new storage volume:
    `sudo mkdir /mnt/data-store2`
54. Mount the new volume:
    `sudo mount /dev/sdg /mnt/data-store2`
55. Verify that the volume you mounted has the file that you created earlier.
    `ls /mnt/data-store2/`
    You should see file.txt.



CONCLUSION

Congratulations! You now have successfully:

- Created an Amazon EBS volume
- Attached the volume to an EC2 instance
- Created a file system on the volume
- Added a file to volume
- Created a snapshot of your volume
- Created a new volume from the snapshot
- Attached and mounted the new volume to your EC2 instance
- Verified that the file you created earlier was on the newly created volume

Practical 4

AIM

Build Your DB Server and Interact With Your DB Using an App

THEORY

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

Amazon Relational Database Service (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

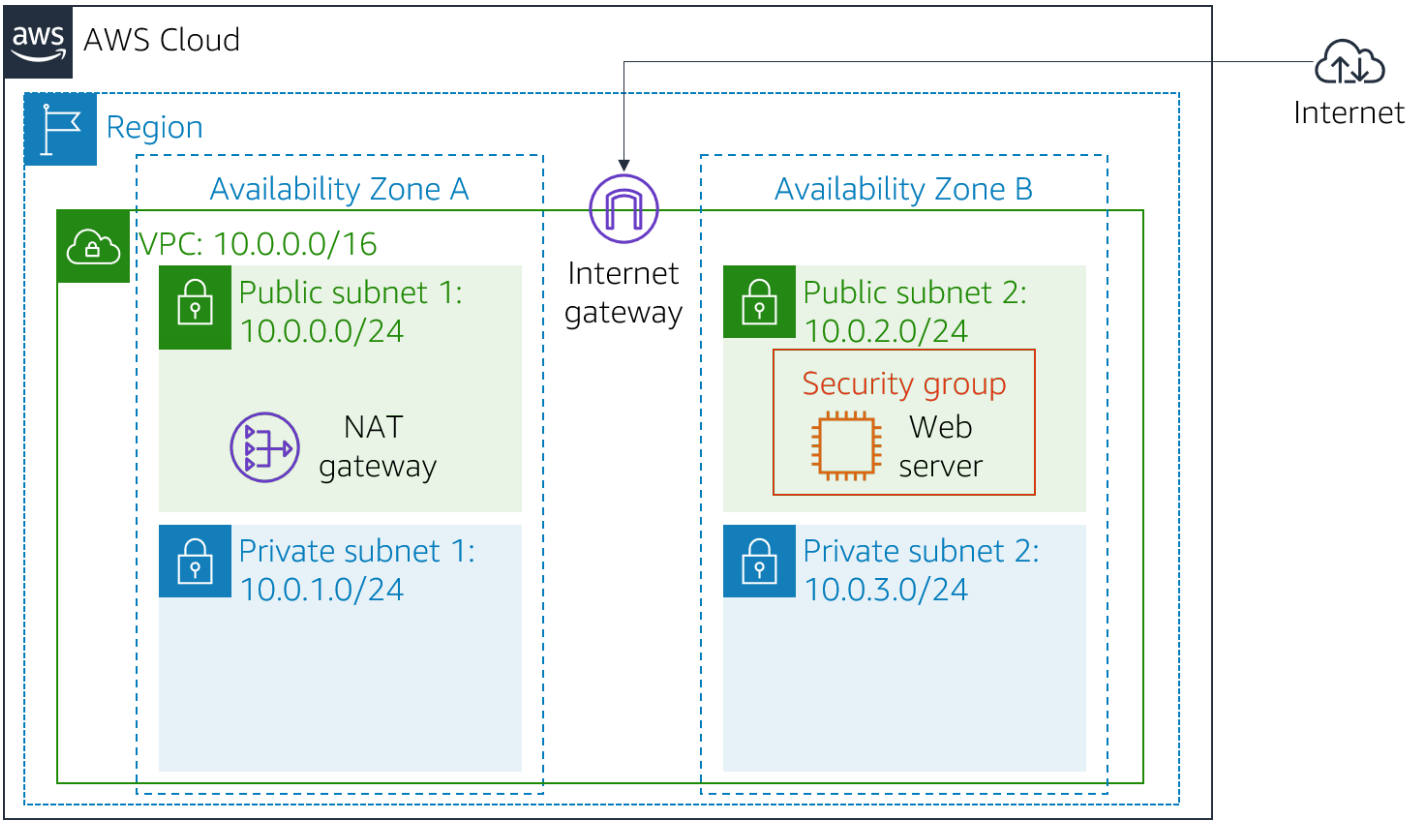
OBJECTIVES

After completing this lab, you can:

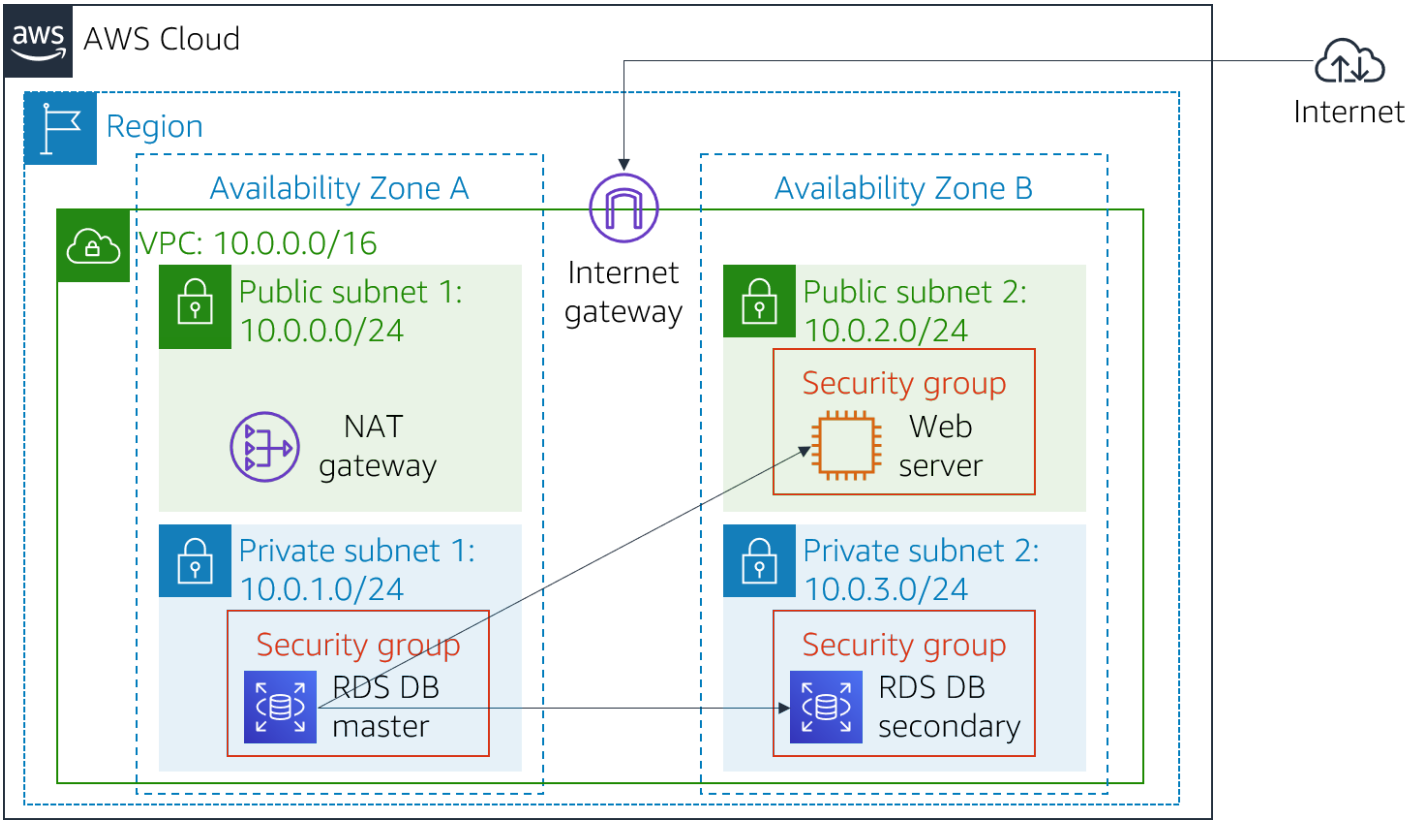
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

PROCEDURE

Infrastructure before this lab:



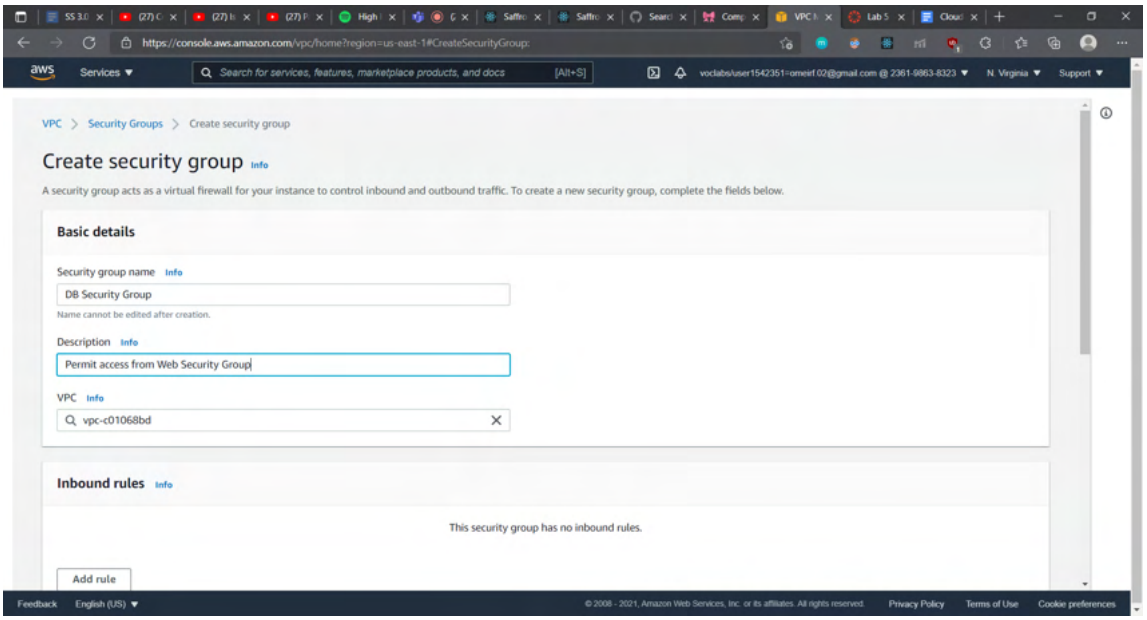
Infrastructure after this lab:



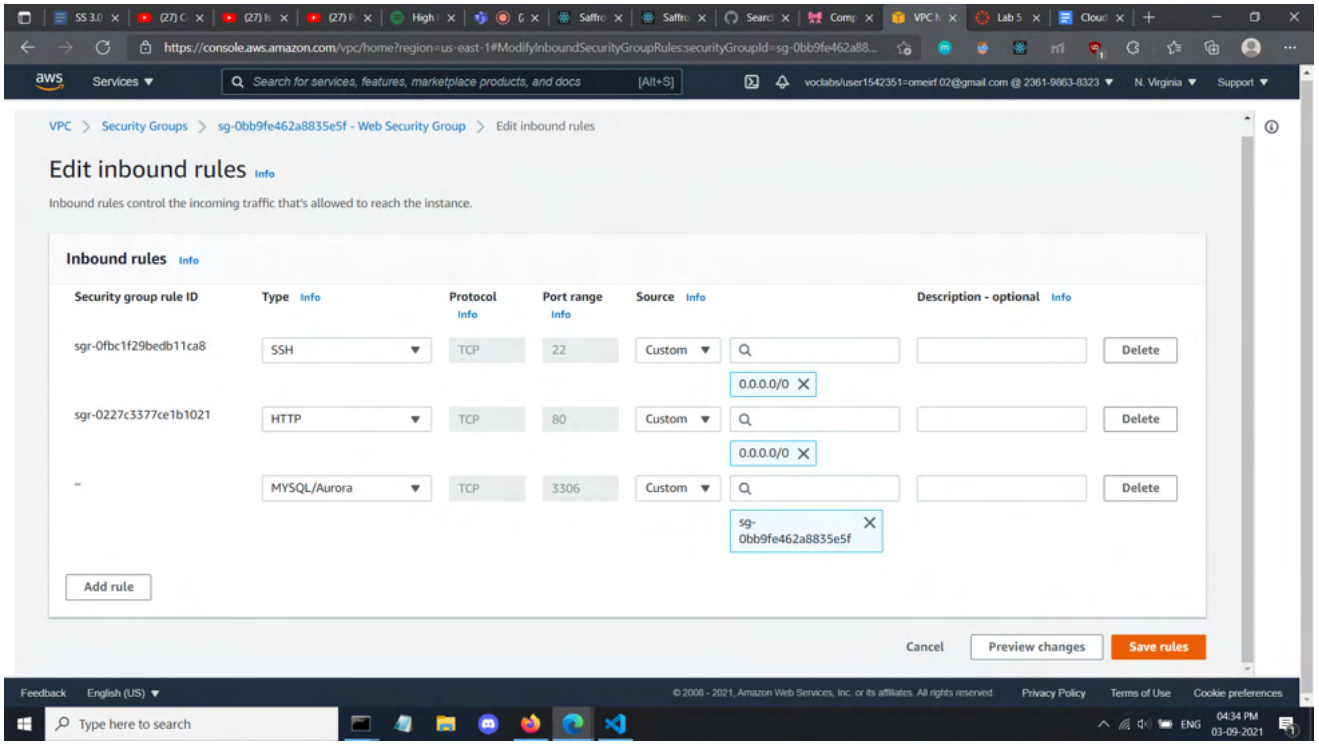
Task 1: Create a Security Group for the RDS DB Instance

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

5. In the **AWS Management Console**, on the **Services** menu, click **VPC**.
6. In the left navigation pane, click **Security Groups**.
7. Click **Create security group** and then configure:
    - **Security group name:** DB Security Group
    - **Description:** Permit access from Web Security Group
    - **VPC:** *Lab VPC*



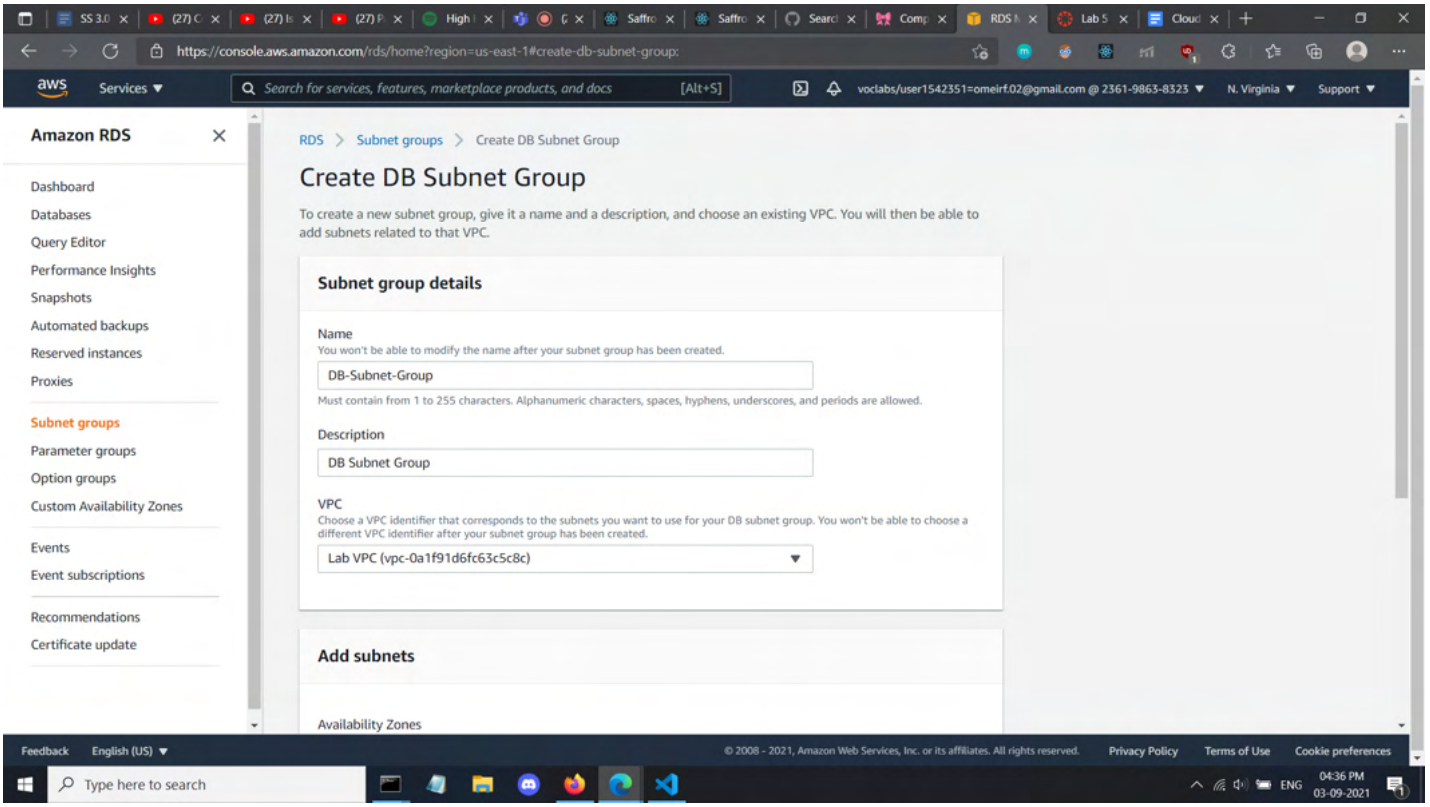
8. You will now add a rule to the security group to permit inbound database requests.
9. In the **Inbound rules** pane, choose **Add rule**
   The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.
10. Configure the following settings:
    - **Type:** *MySQL/Aurora (3306)*
    - **CIDR, IP, Security Group or Prefix List:** Type sg and then select *Web Security Group*.
11. This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.
12. Choose **Create security group**
    You will use this security group when launching the Amazon RDS database.



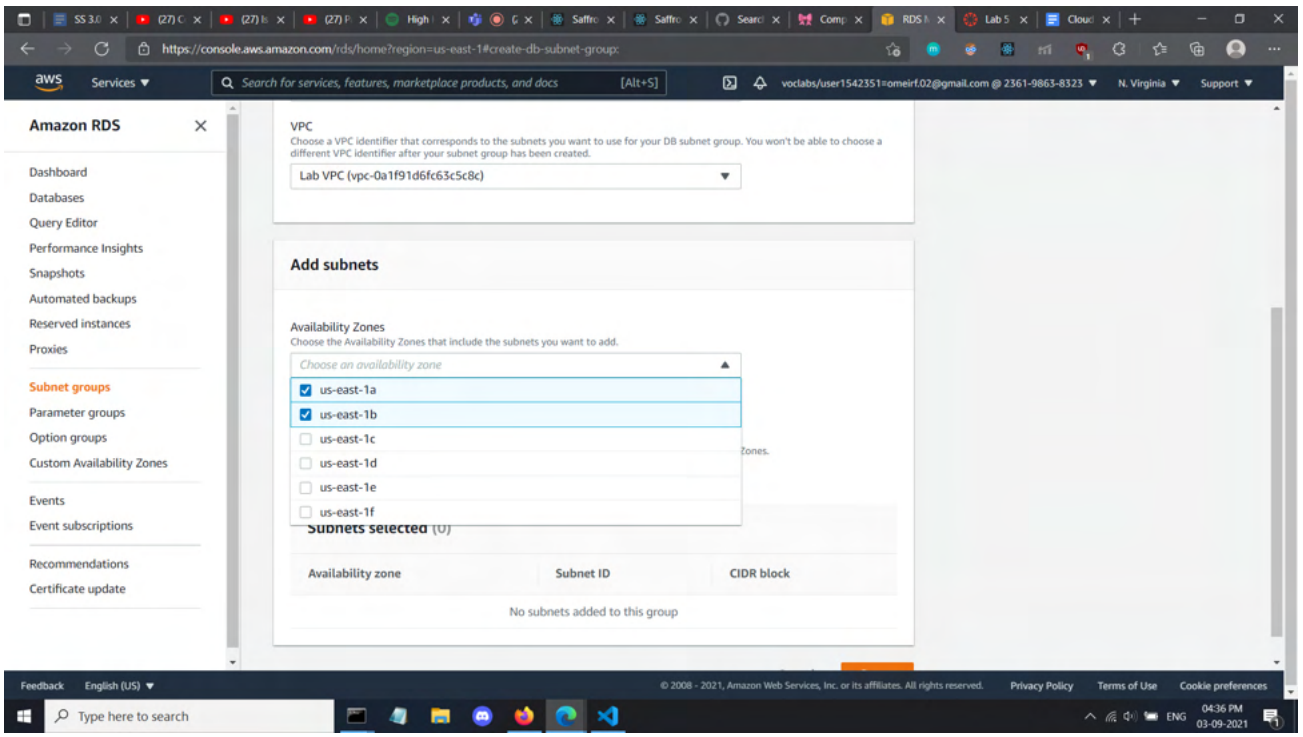
Task 2: Create a DB Subnet Group

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

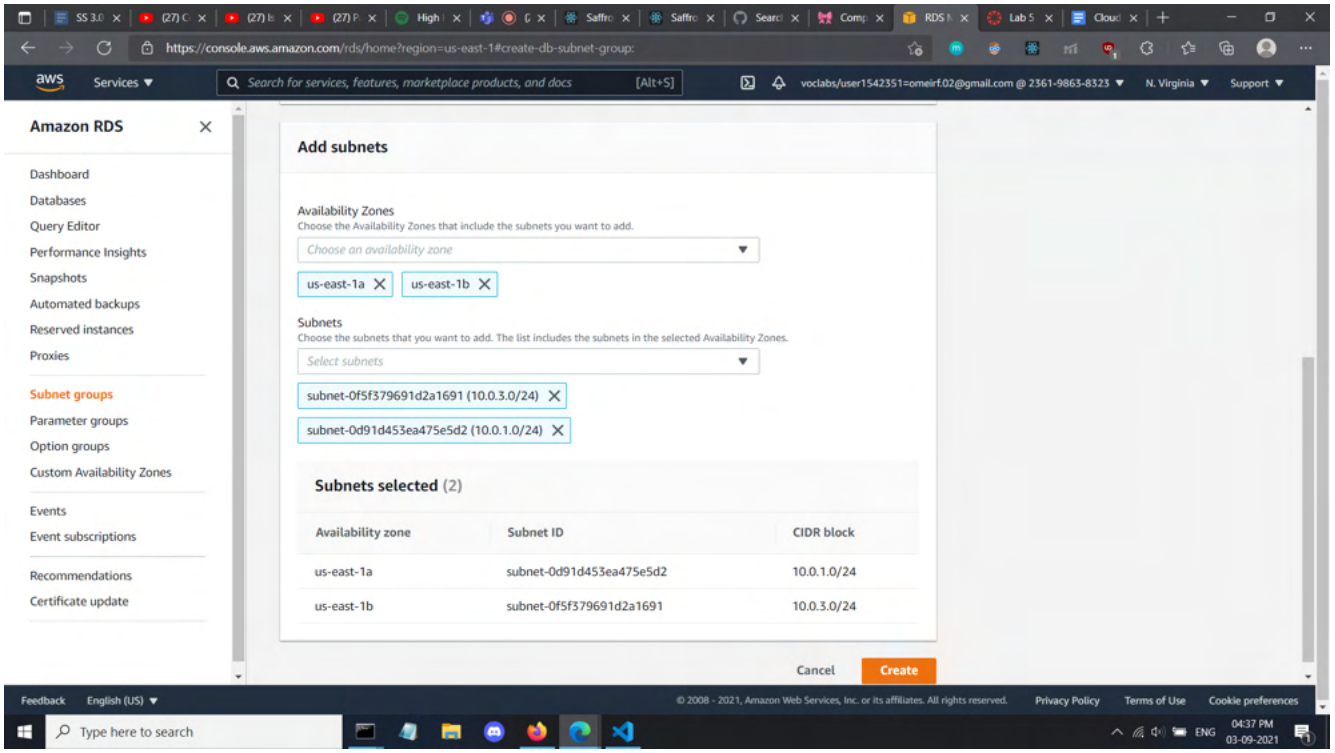
11. On the **Services** menu, click **RDS**.
12. In the left navigation pane, click **Subnet groups**.
    If the navigation pane is not visible, click the menu icon in the top-left corner.
13. Click **Create DB Subnet Group** then configure:
    - **Name:** DB-Subnet-Group
    - **Description:** DB Subnet Group
    - **VPC:** *Lab VPC*



14. Scroll down to the **Add Subnets** section.
15. Expand the list of values under **Availability Zones** and select the first two zones: **us-east-1a** and **us-east-1b**.



16. Expand the list of values under **Subnets** and select the subnets associated with the CIDR ranges **10.0.1.0/24** and **10.0.3.0/24**.
    These subnets should now be shown in the **Subnets selected** table.



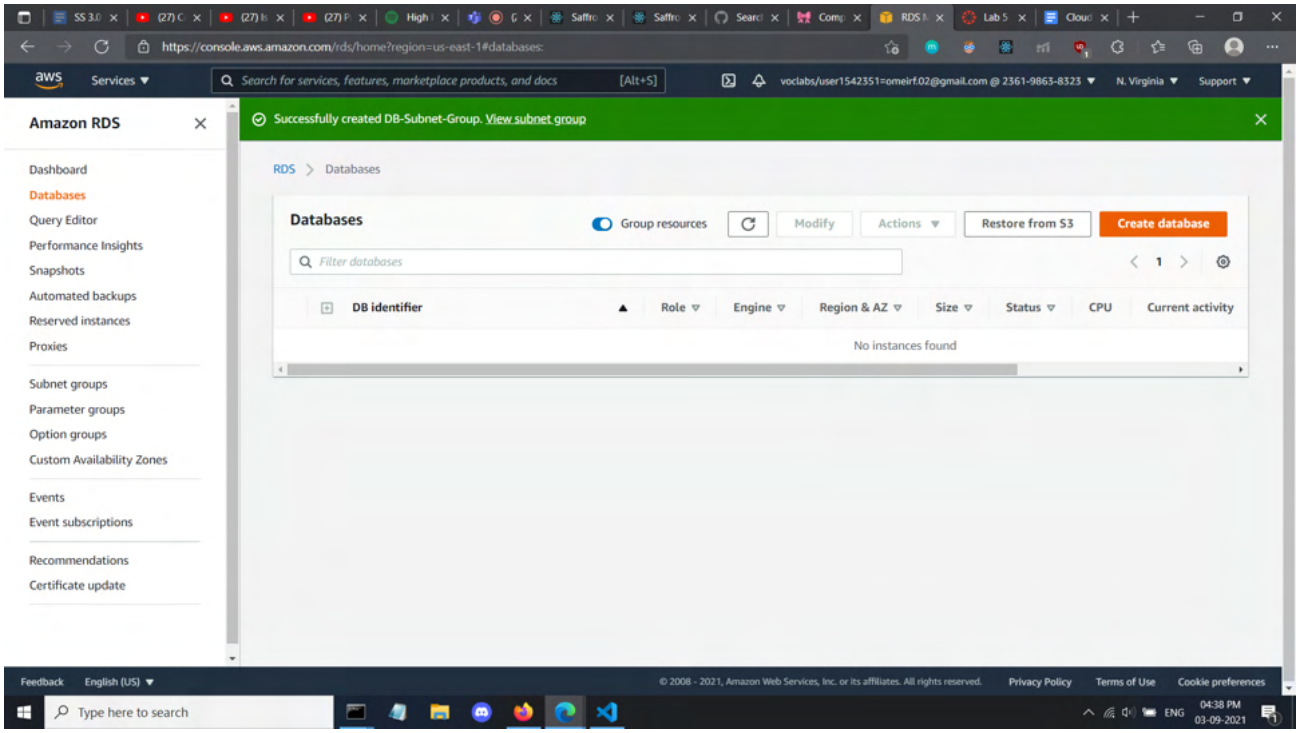
17. Click **Create**
    You will use this DB subnet group when creating the database in the next task.

Task 3: Create an Amazon RDS DB Instance

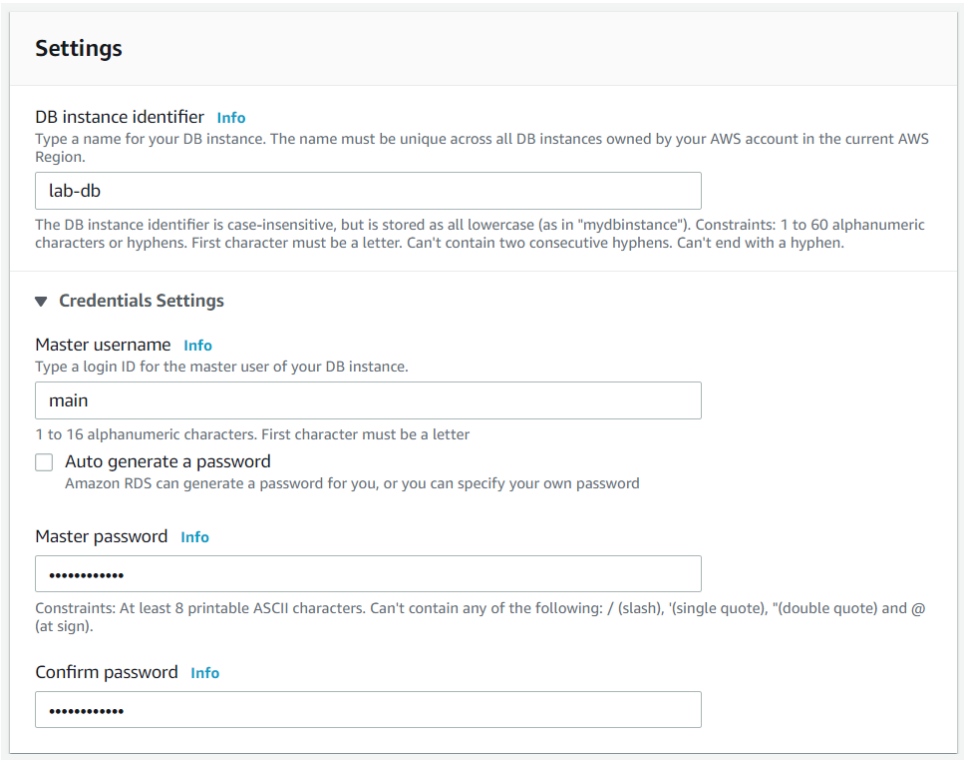
In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS ***Multi-AZ*** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ).

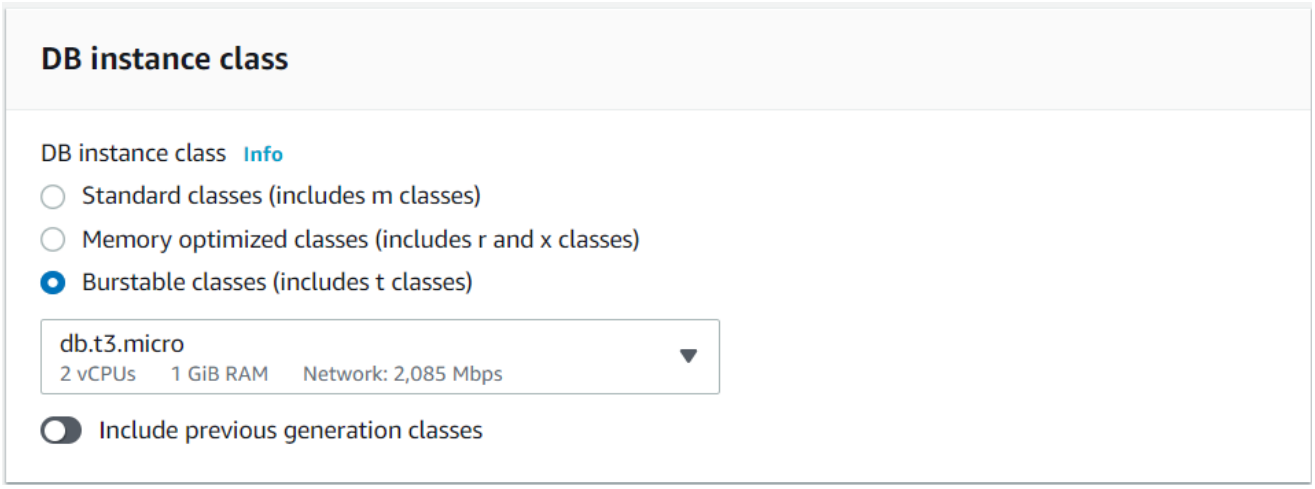
18. In the left navigation pane, click **Databases**.
19. Click **Create database**
    If you see **Switch to the new database creation flow** at the top of the screen, please click.



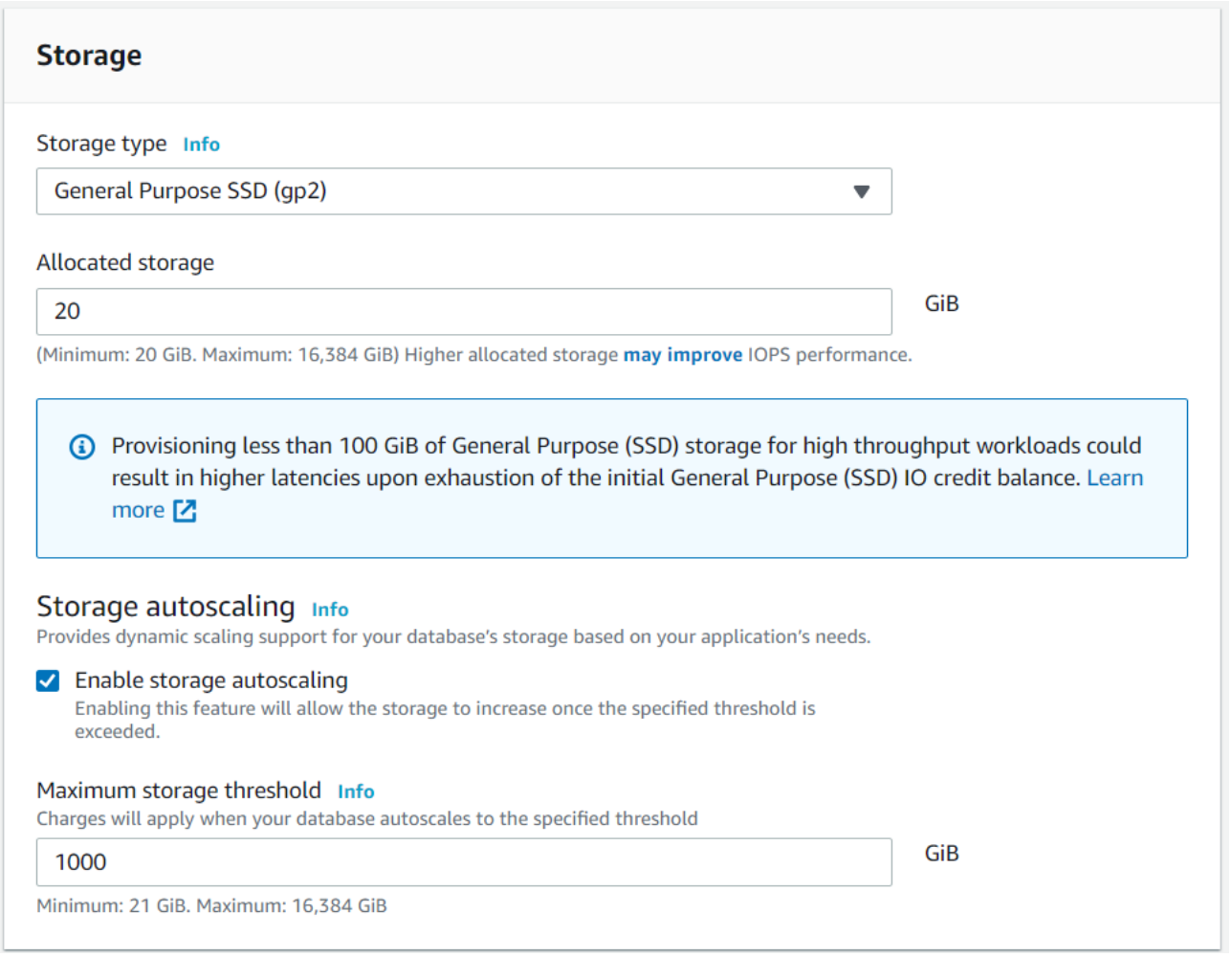
20. Select **MySQL**.
21. Under **Settings**, configure:
    - **DB instance identifier:** lab-db
    - **Master username:** main
    - **Master password:** lab-password
    - **Confirm password:** lab-password



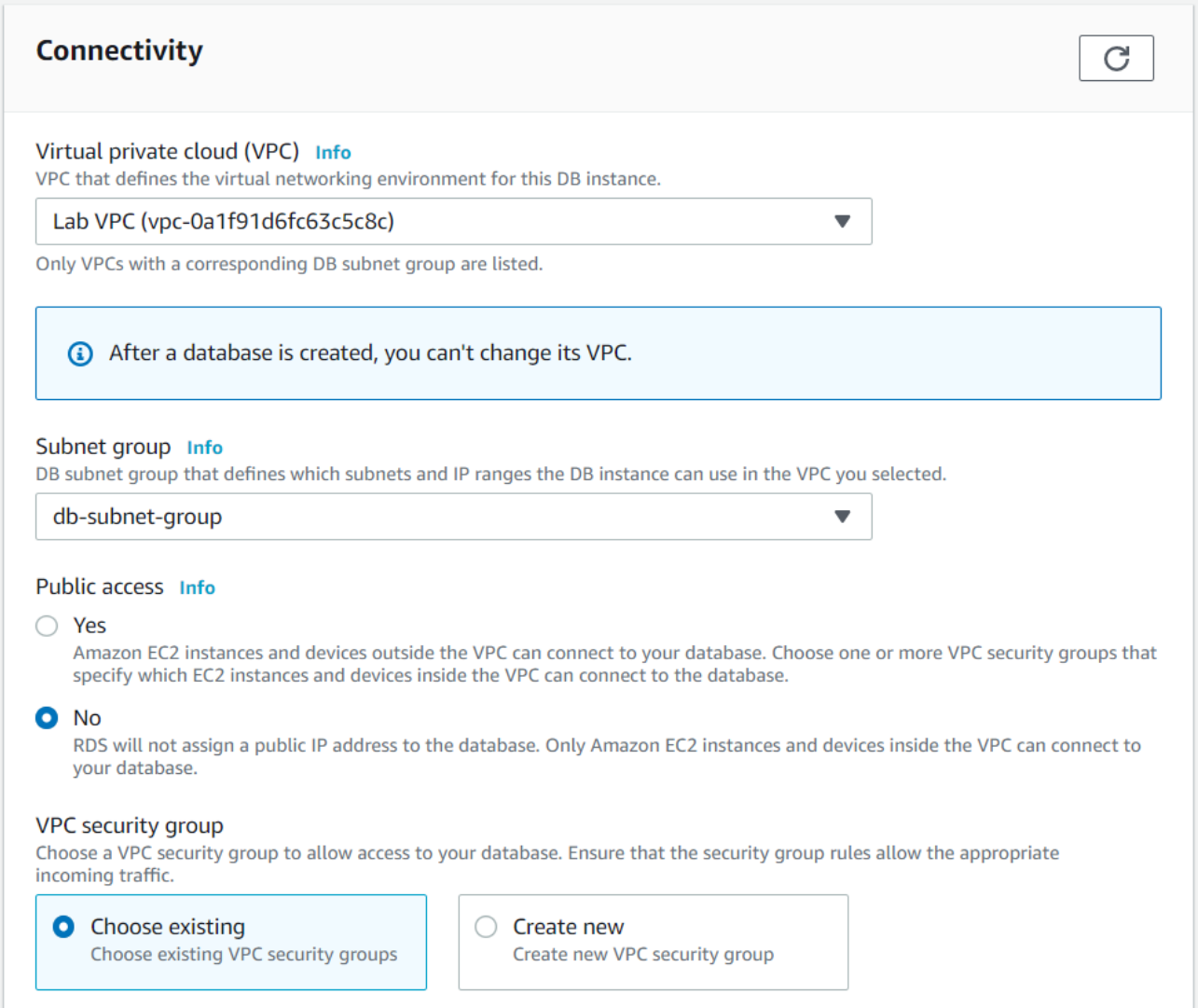
22. Under **DB instance size**, configure:
    - Select **Burstable classes (includes t classes)**.
    - Select *db.t3.micro*



23. Under **Storage**, configure:
    - **Storage type:** *General Purpose (SSD)*
    - **Allocated storage:** *20*



24. Under **Connectivity**, configure:
    - **Virtual Private Cloud (VPC):** *Lab VPC*



25. Under **Existing VPC security groups**, from the dropdown list:
    - Choose *DB Security Group*.
    - Deselect *default*.

26. Expand **Additional configuration**, then configure:
    - **Initial database name:** lab
    - Uncheck **Enable automatic backups**.
    - Uncheck **Enable Enhanced monitoring**.

Additional configuration

Database options, encryption enabled, backup disabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection enabled

Database options

Initial database name Info

lab

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group Info

default.mysql8.0

Option group Info

default:mysql-8-0

Backup

Enable automated backups

Creates a point-in-time snapshot of your database

Encryption

Enable encryption

Choose to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the AWS Key Management Service console. Info

AWS KMS Key Info

(default) aws/rds

Account

236198638323

KMS key ID

alias/aws/rds

Monitoring

Enable Enhanced monitoring

Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU

Log exports

Select the log types to publish to Amazon CloudWatch Logs

- Audit log
- Error log
- General log
- Slow query log

IAM role

The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS service-linked role

Ensure that general, slow query, and audit logs are turned on. Error logs are enabled by default. Learn more

Maintenance

Auto minor version upgrade Info

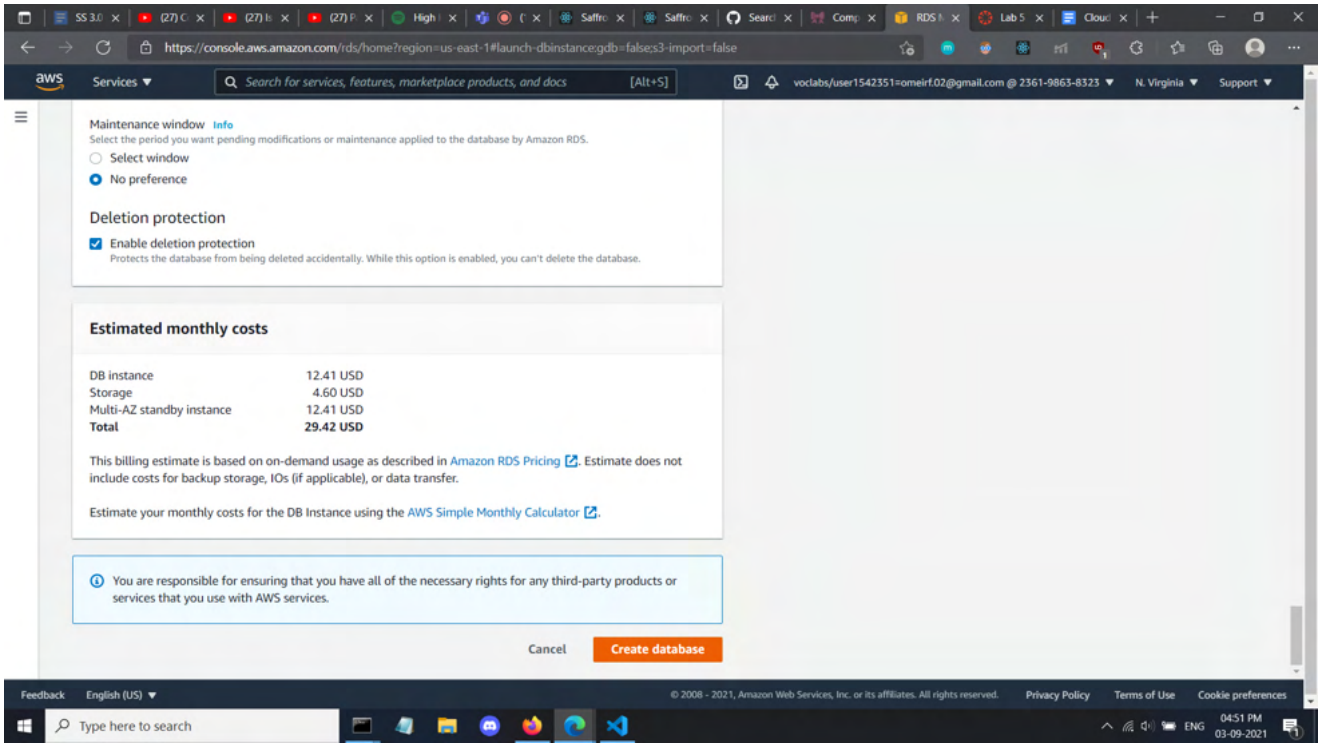
Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the

27. This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.
28. Click **Create database**

    Your database will now be launched.

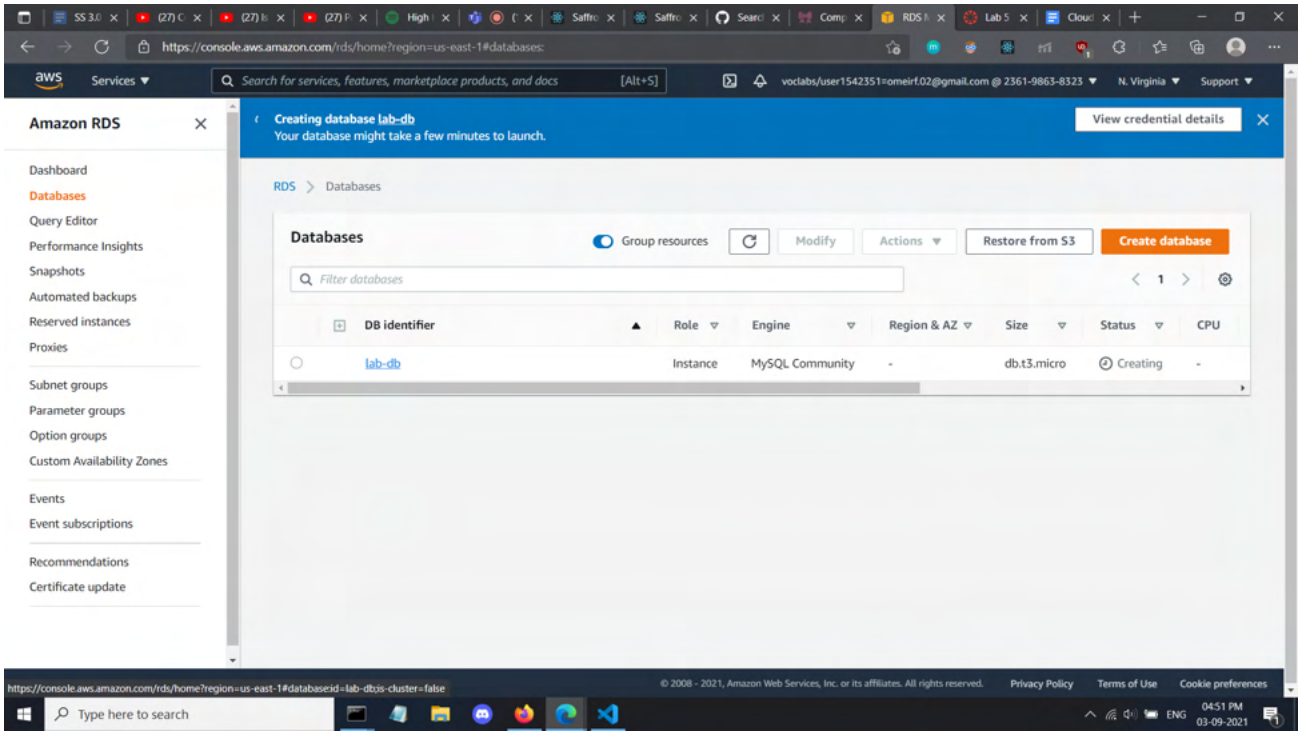
    If you receive an error that mentions "not authorized to perform: iam:CreateRole", make sure you unchecked *Enable Enhanced monitoring* in the previous step.



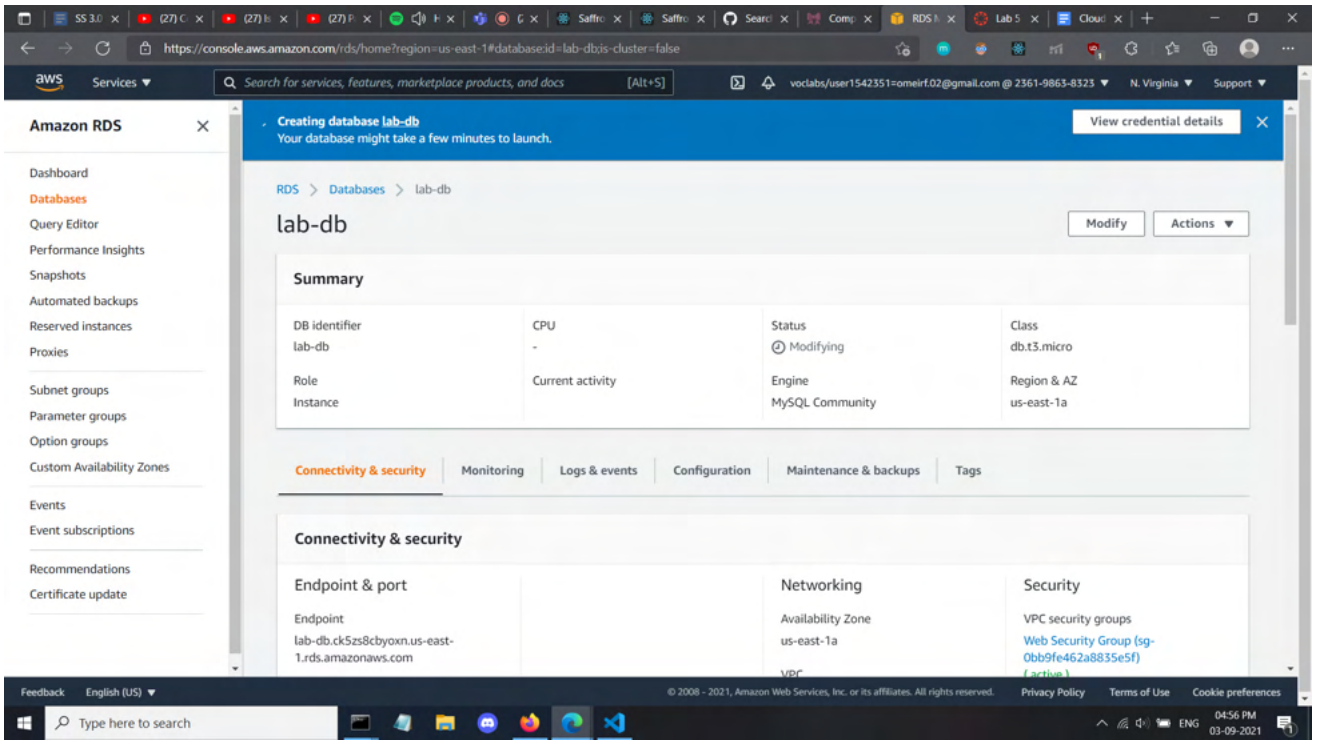
29. Click **lab-db** (click the link itself).

    You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

    While you are waiting, you might want to review the Amazon RDS FAQs or grab a cup of coffee.



30. Wait until **Info** changes to **Modifying** or **Available**.

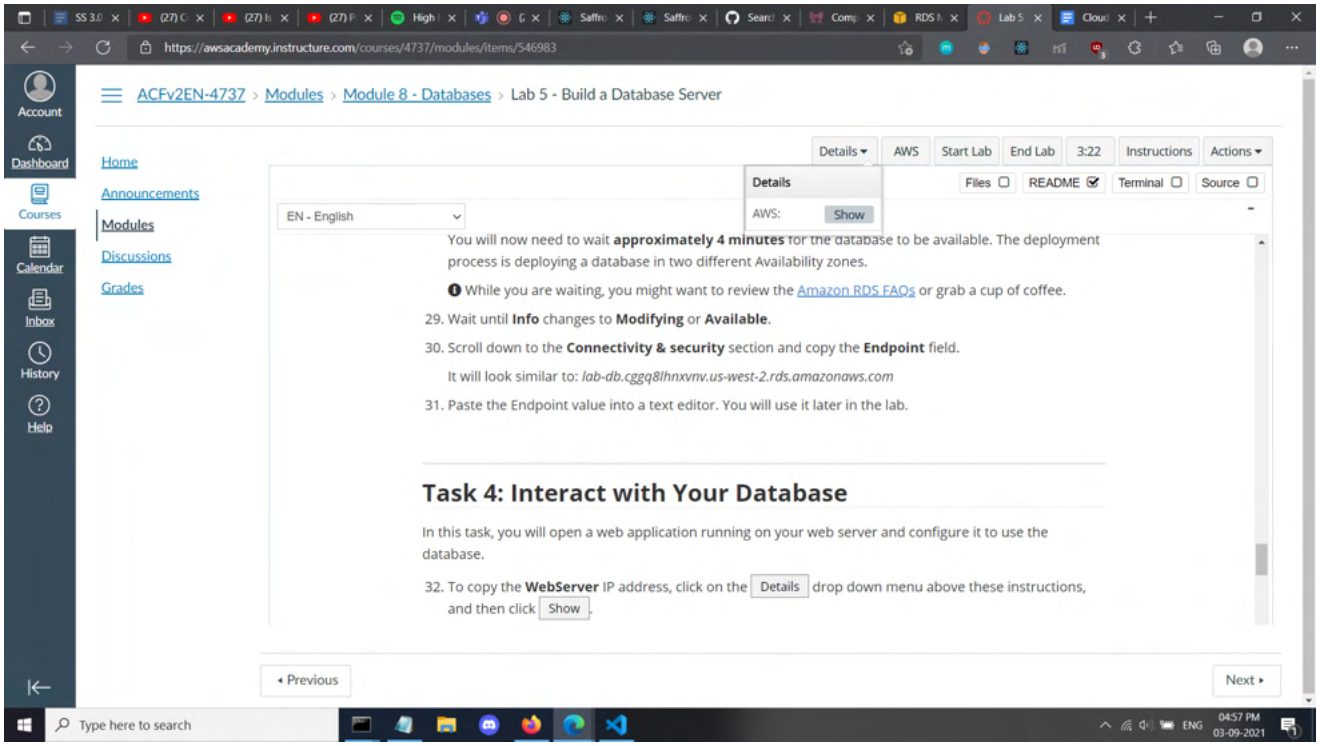


31. Scroll down to the **Connectivity & security** section and copy the **Endpoint** field.
    It will look similar to: *lab-db.cggq8lhnxvnv.us-west-2.rds.amazonaws.com*
32. Paste the Endpoint value into a text editor. You will use it later in the lab.

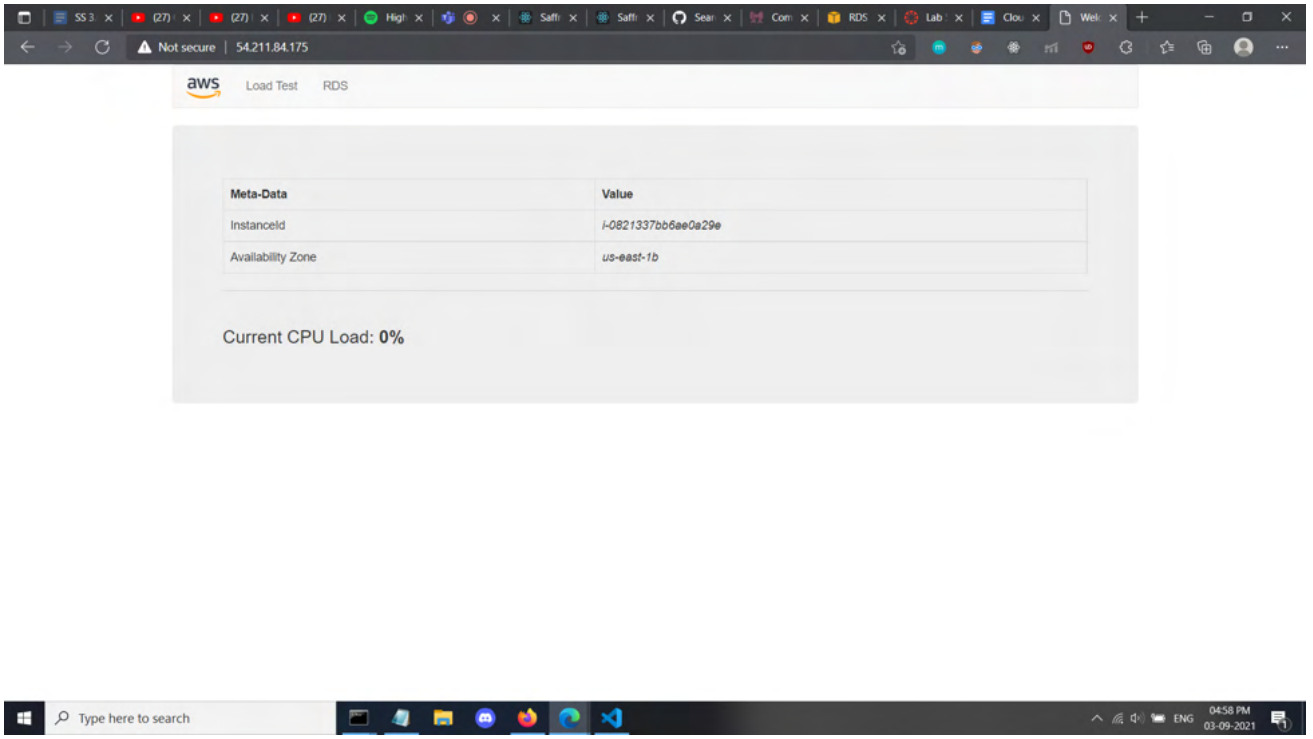
Task 4: Interact with Your Database

In this task, you will open a web application running on your web server and configure it to use the database.

32. To copy the **WebServer** IP address, click on the Details drop down menu above these instructions, and then click Show.

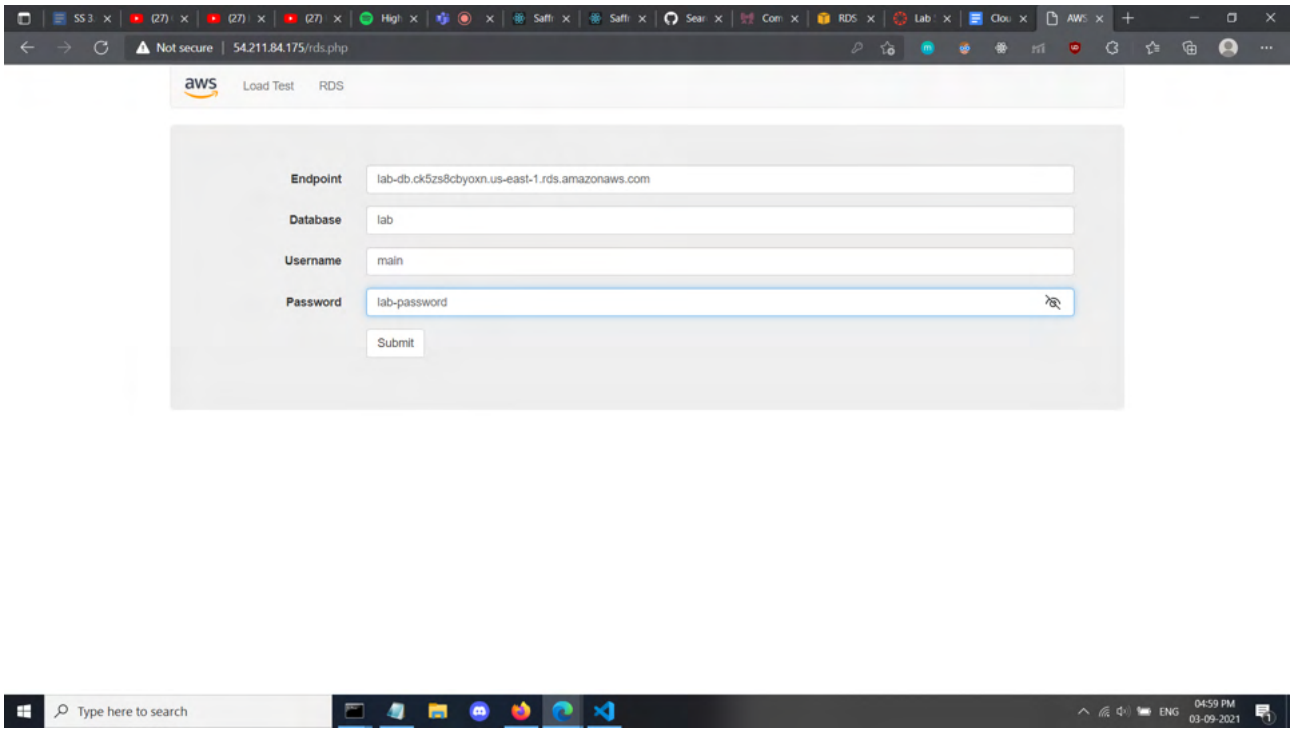


33. Open a new web browser tab, paste the *WebServer* IP address and press Enter.
The web application will be displayed, showing information about the EC2 instance.

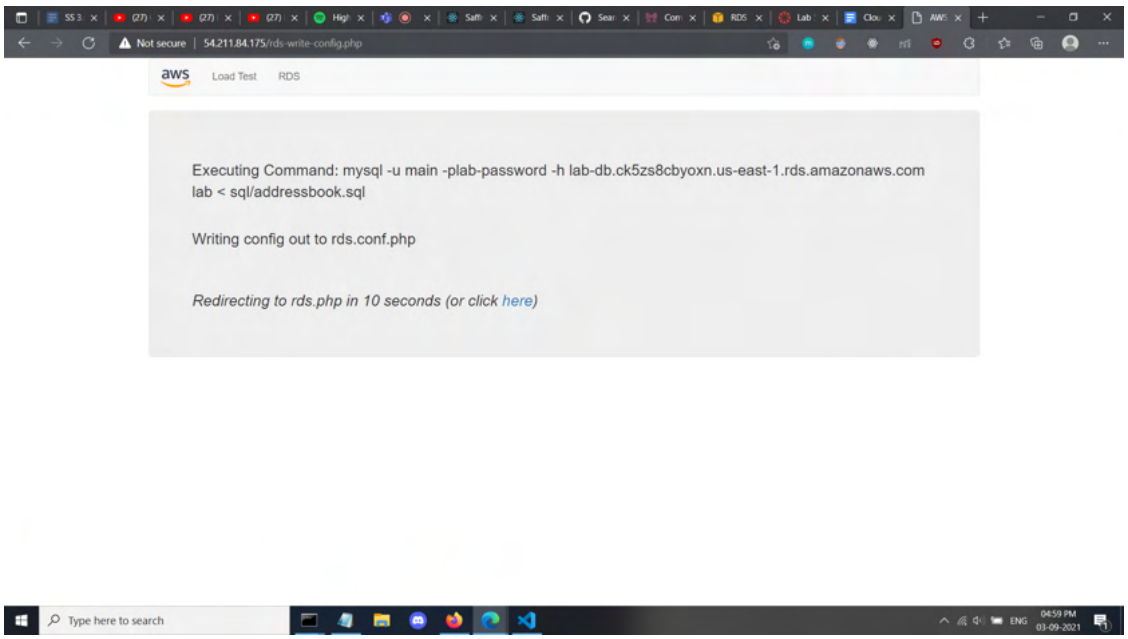


34. Click the **RDS** link at the top of the page.
You will now configure the application to connect to your database.

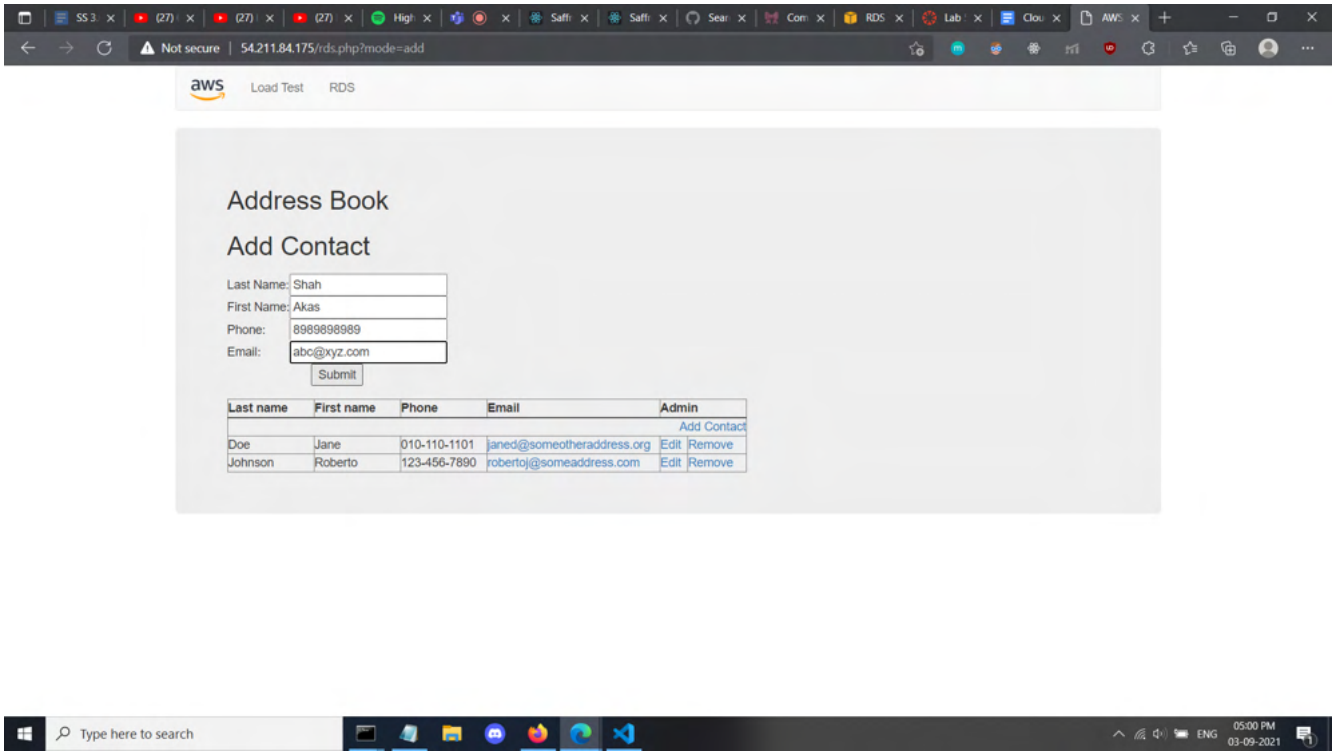
35. Configure the following settings:
    - **Endpoint:** Paste the Endpoint you copied to a text editor earlier
    - **Database:** lab
    - **Username:** main
    - **Password:** lab-password
    - Click **Submit**



36. A message will appear explaining that the application is running a command to copy information to the database. After a few seconds the application will display an **Address Book**.
    The Address Book application is using the RDS database to store information.



37. Test the web application by adding, editing and removing contacts.
    The data is being persisted to the database and is automatically replicating to the second Availability Zone.



Practical 5

AIM

Scale and Load Balance Your Architecture

THEORY

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon EC2 capacity out or in automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

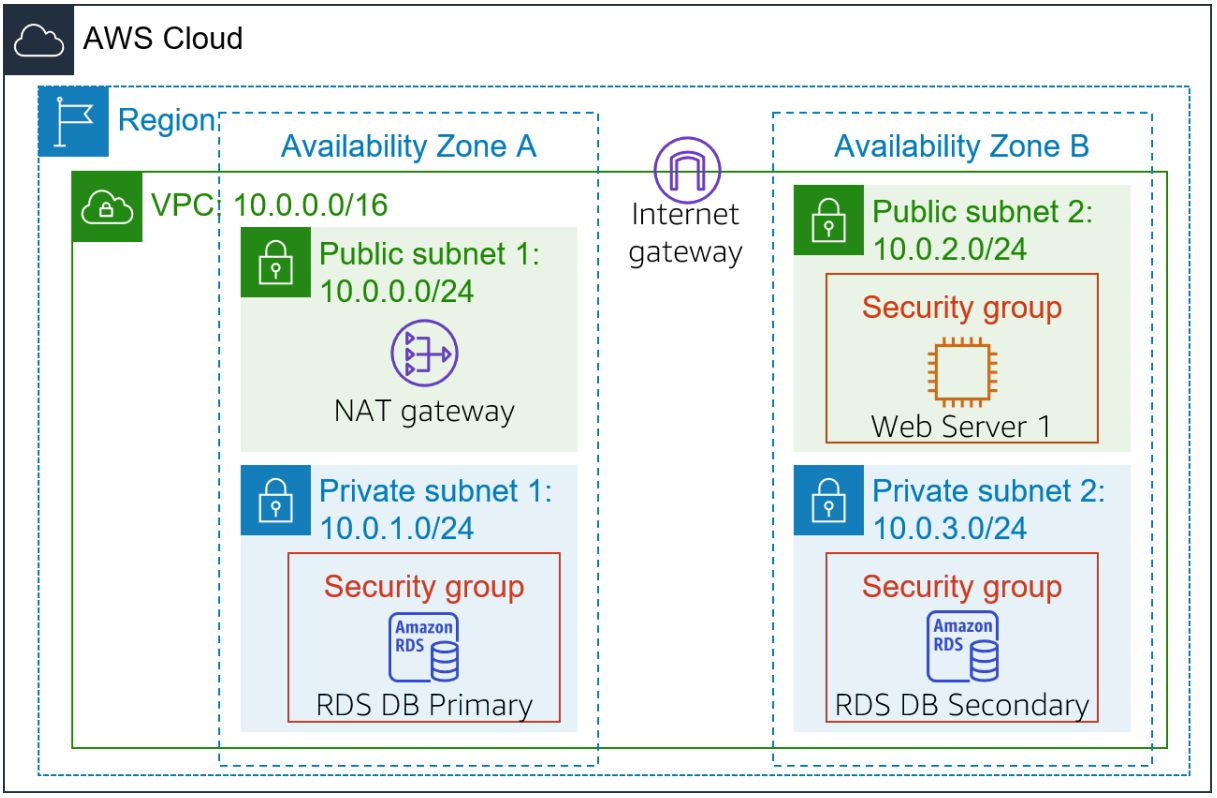
OBJECTIVES

After completing this lab, you can:

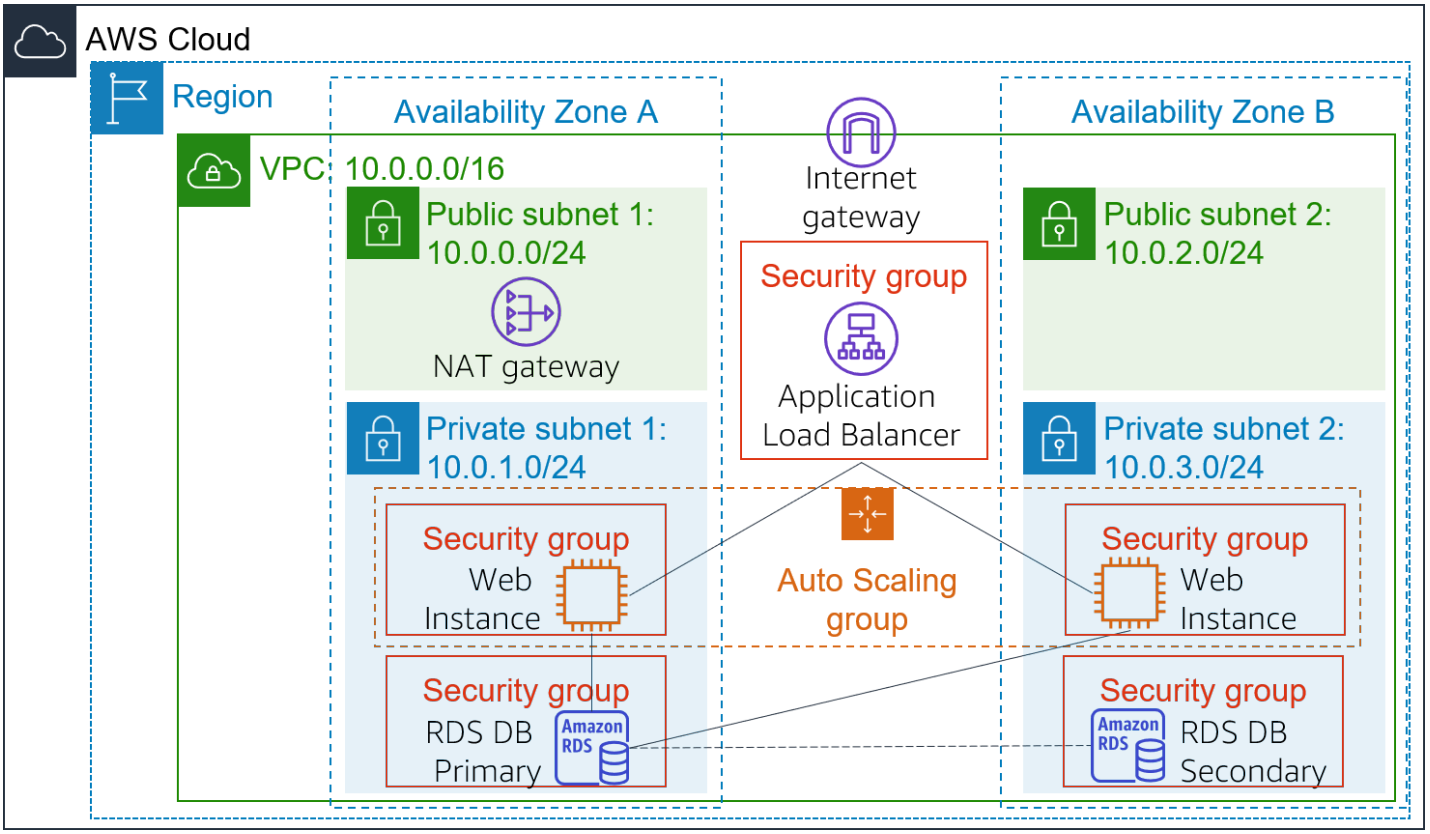
- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

PROCEDURE

You start with the following infrastructure:



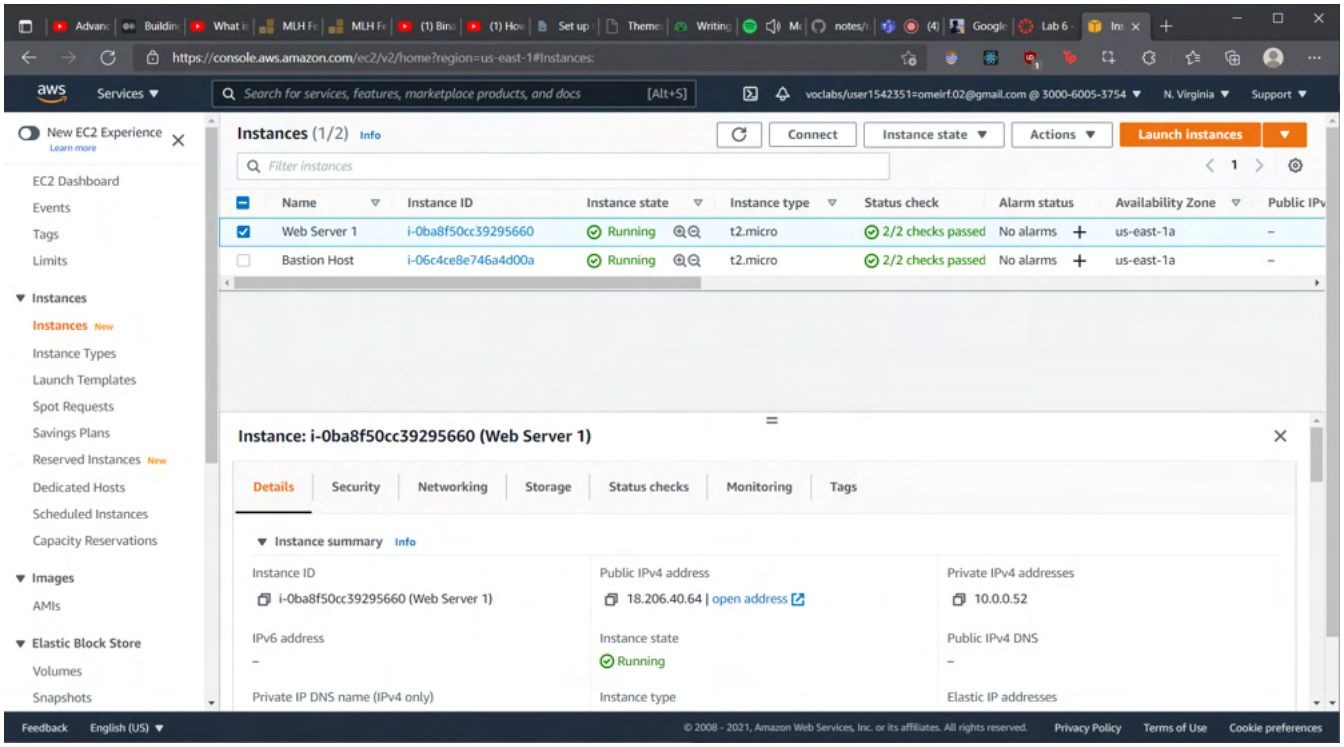
The final state of the infrastructure is:



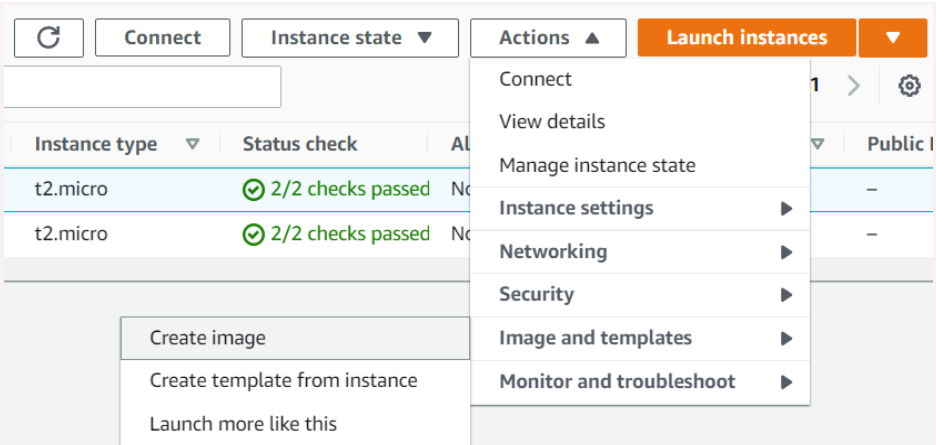
Task 1: Create an AMI for Auto Scaling

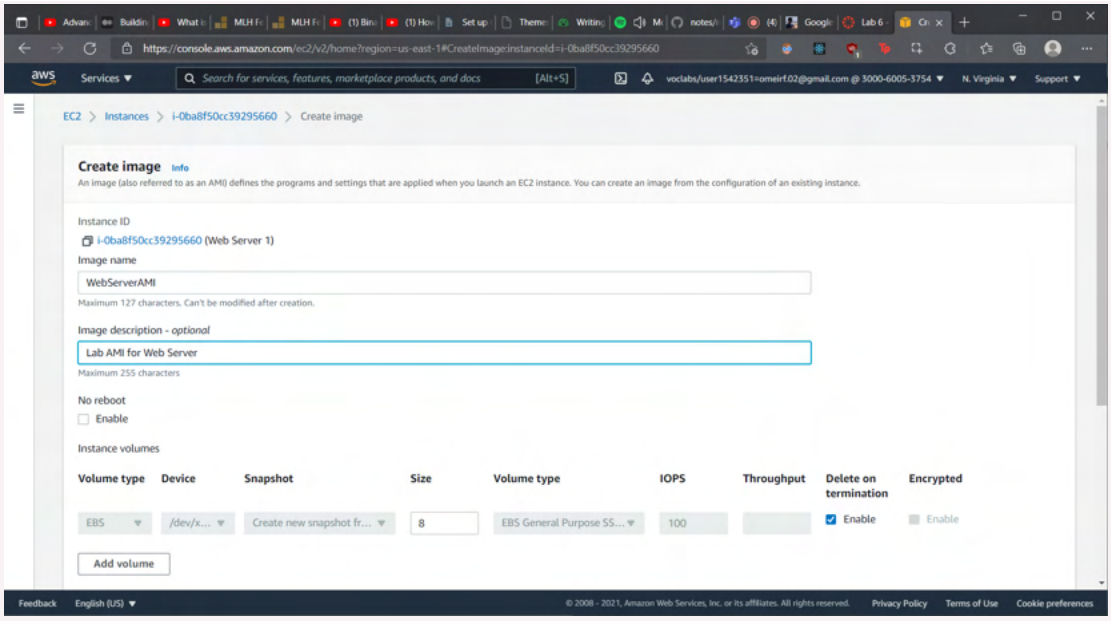
In this task, you will create an AMI from the existing *Web Server 1*. This will save the contents of the boot disk so that new instances can be launched with identical content.

5. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
6. In the left navigation pane, click **Instances**.
   First, you will confirm that the instance is running.
7. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. Click refresh to update.
   You will now create an AMI based upon this instance.
8. Select **Web Server 1**.

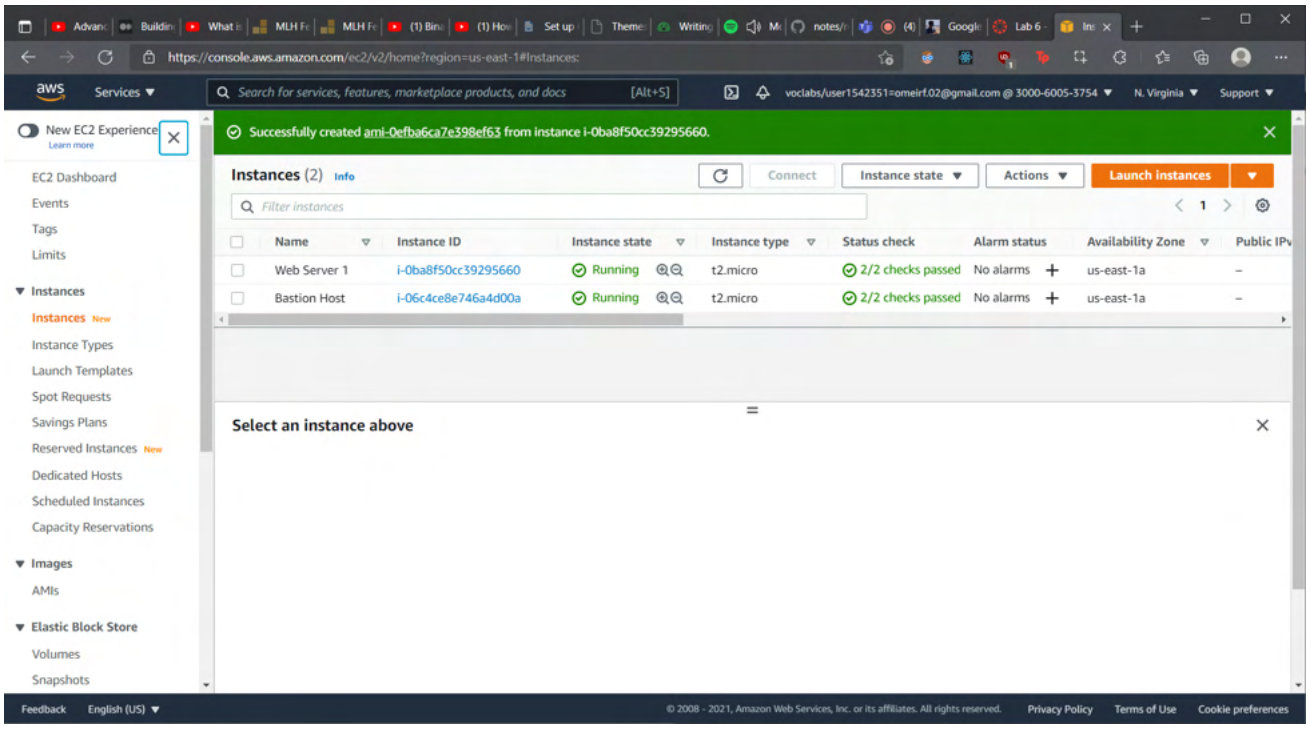


9. In the **Actions** menu, click **Image and templates** > **Create image**, then configure:
   - **Image name:** `WebServerAMI`
   - **Image description:** `Lab AMI for Web Server`





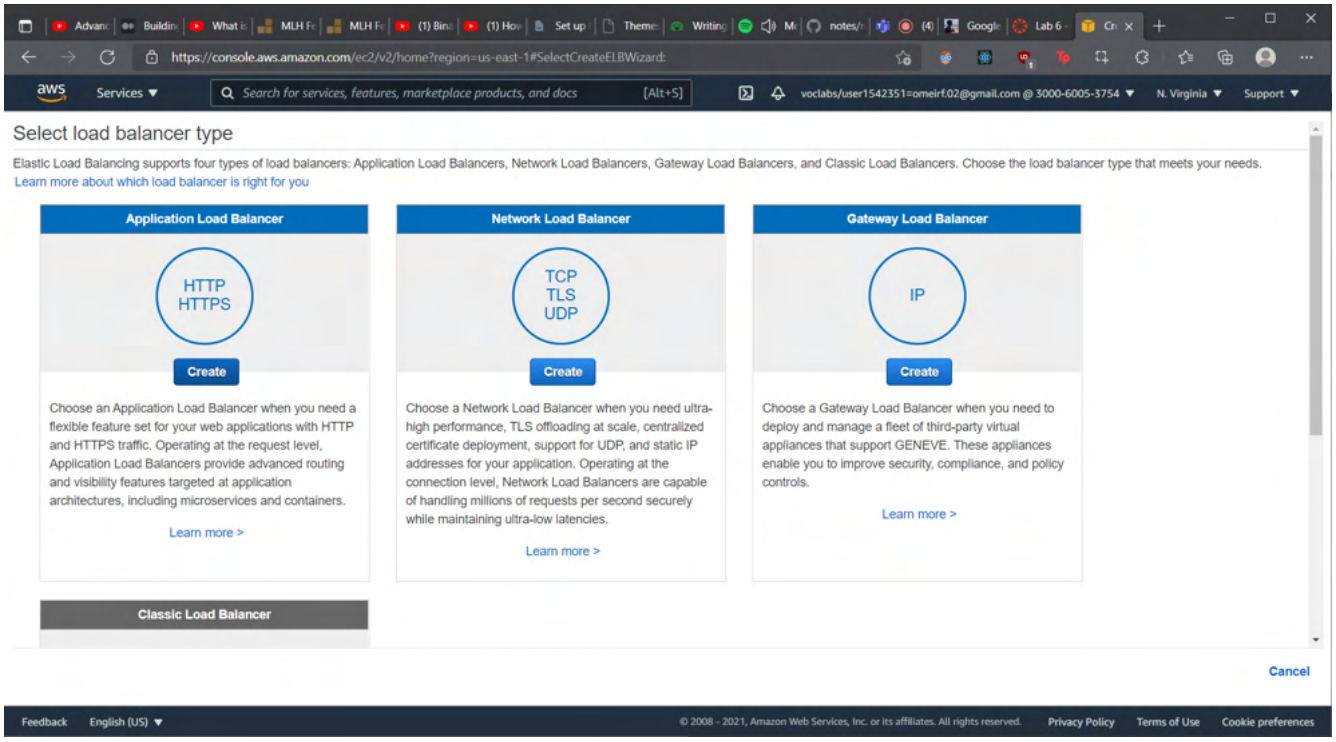
10. Click **Create image**
    A confirmation banner displays the **AMI ID** for your new AMI.
    You will use this AMI when launching the Auto Scaling group later in the lab.
    ID: ami-0efba6ca7e398ef63



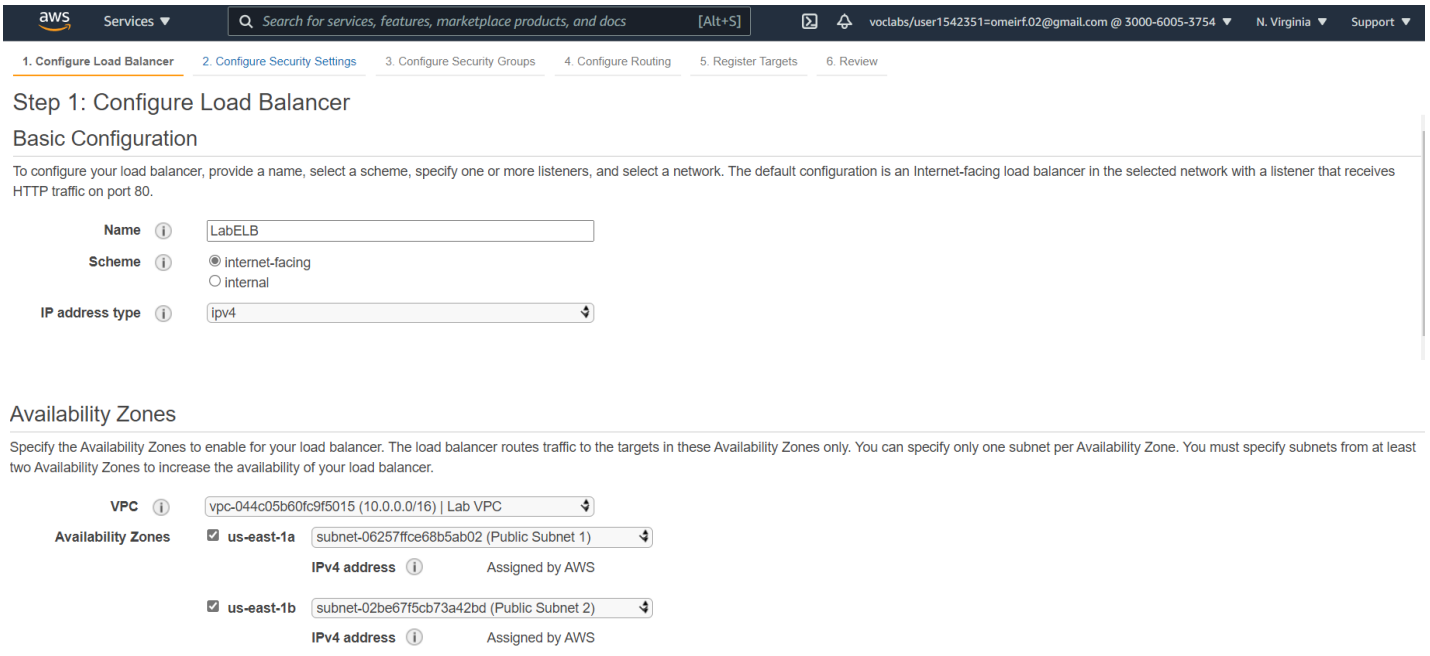
Task 2: Create a Load Balancer

In this task, you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

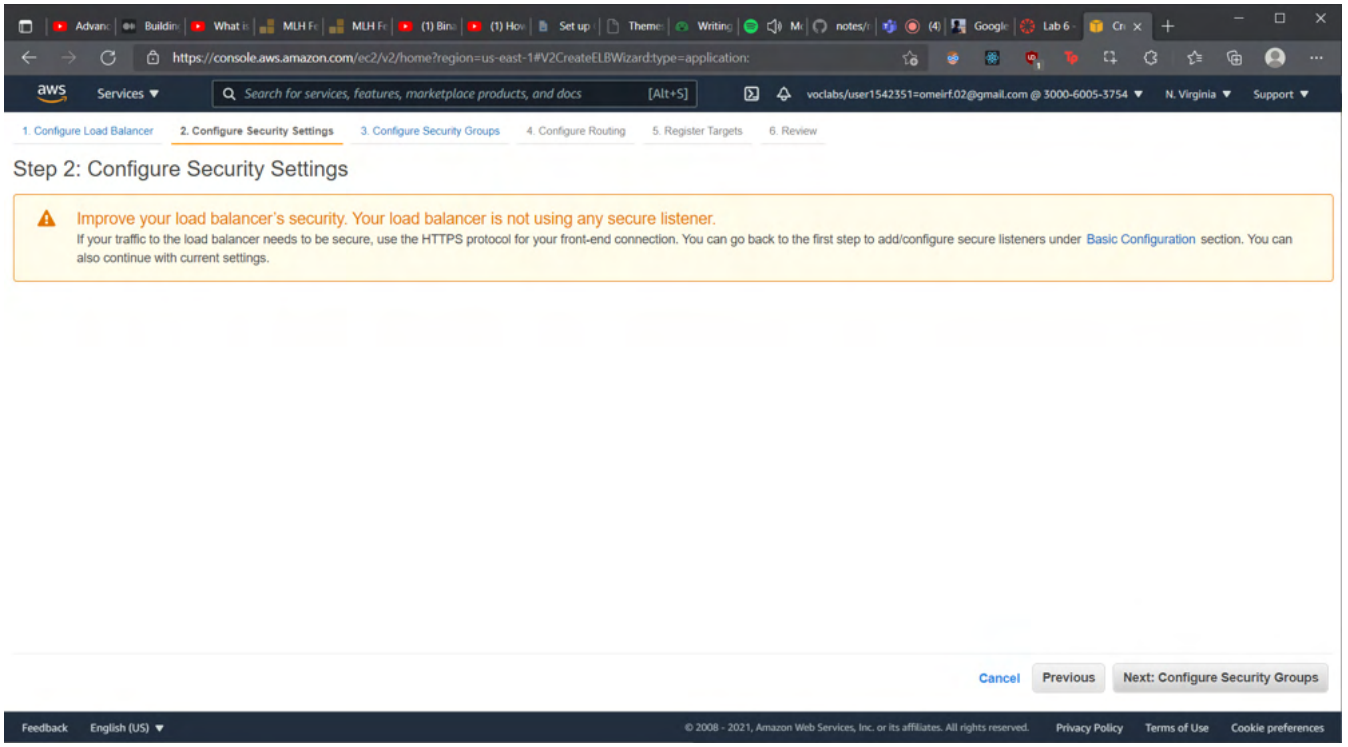
11. In the left navigation pane, click **Load Balancers**.
12. Click **Create Load Balancer**
    Several different types of load balancer are displayed. You will be using an *Application Load Balancer* that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: Comparison of Load Balancers



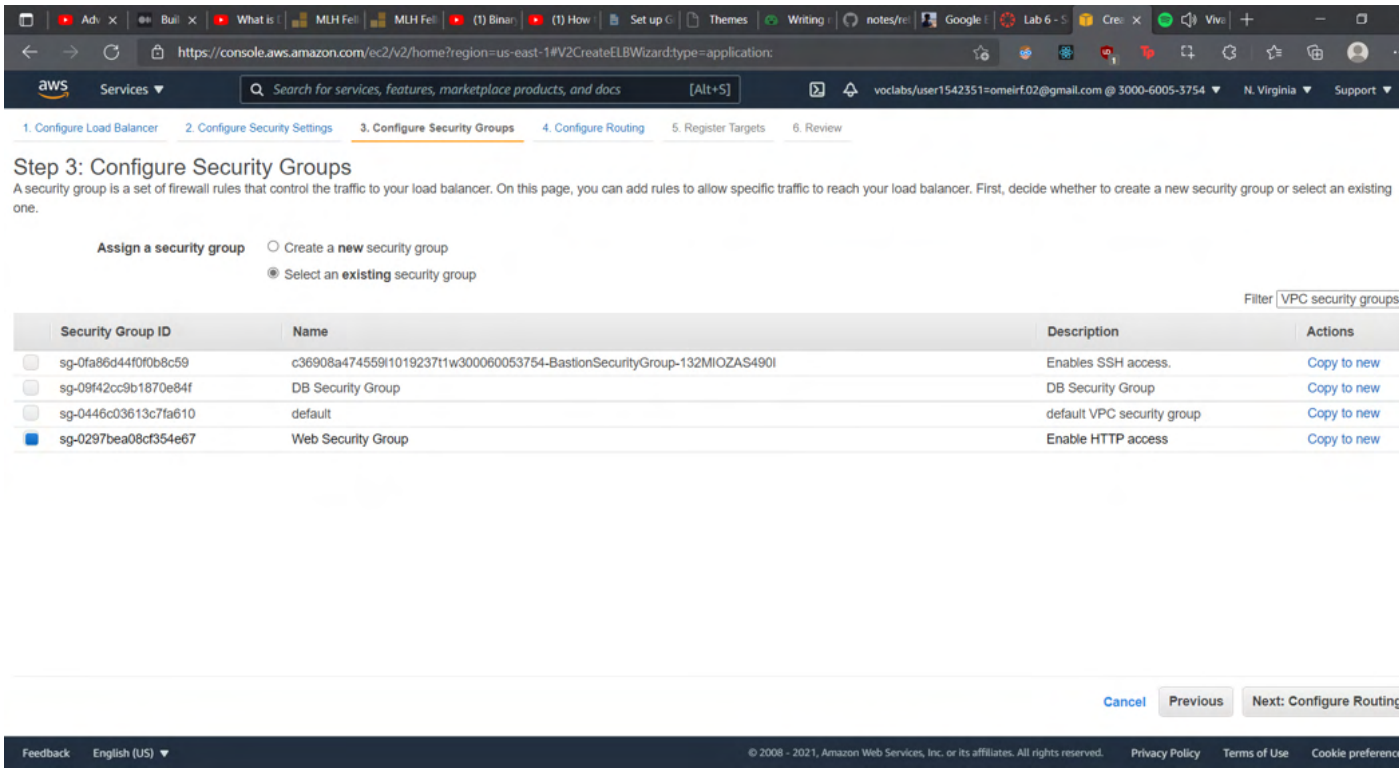
13. Under **Application Load Balancer** click **Create** and configure:
    - **Name:** `LabELB`
    - **VPC:** *Lab VPC* (In the **Availability Zones** section)
    - **Availability Zones:** Select both to see the available subnets.
    - Select **Public Subnet 1** and **Public Subnet 2**



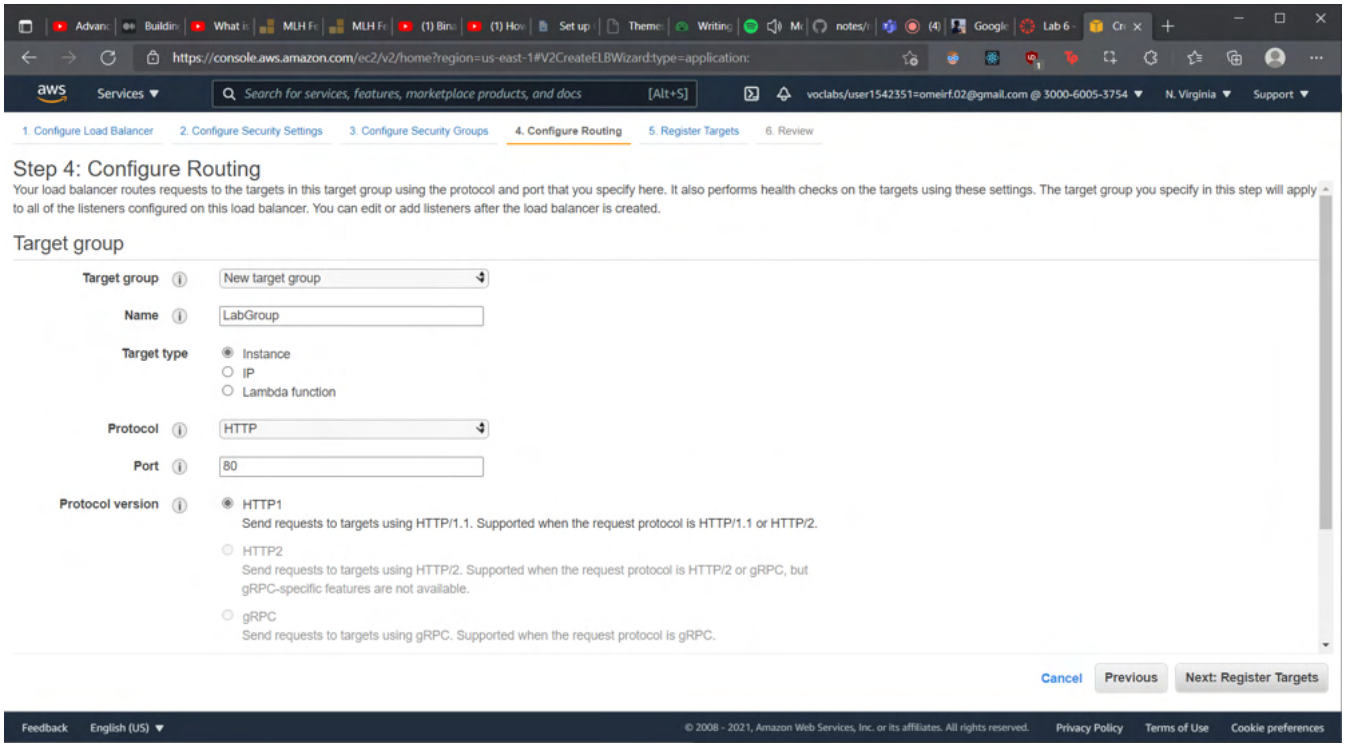
14. This configures the load balancer to operate across multiple Availability Zones.
15. Click **Next: Configure Security Settings**
    You can ignore the *"Improve your load balancer's security."* warning.



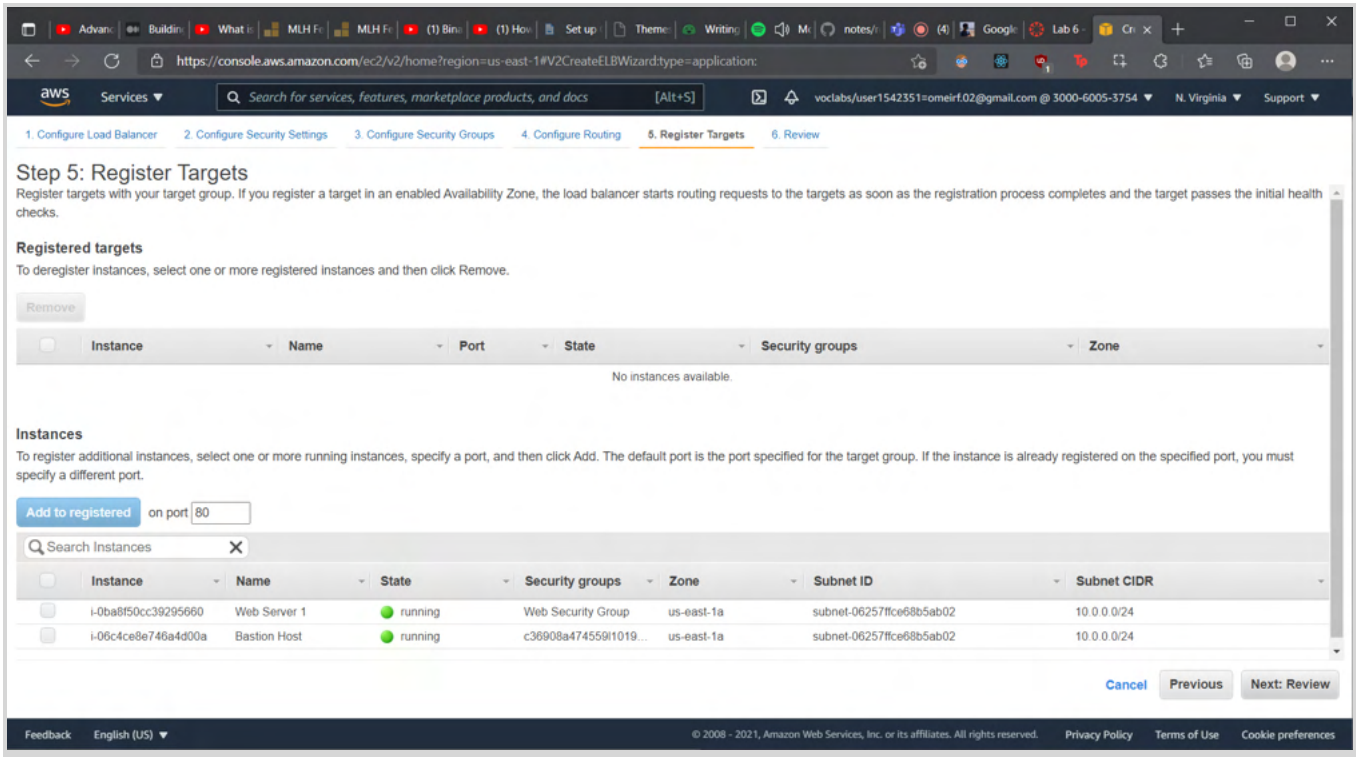
16. Click **Next: Configure Security Groups**
    A *Web Security Group* has already been created for you, which permits HTTP access.
17. Select **Web Security Group** and deselect **default**.



18. Click **Next: Configure Routing**
    Routing configures where to send requests that are sent to the load balancer. You will create a *Target Group* that will be used by Auto Scaling.
19. For **Name**, enter: `LabGroup`
20. Click **Next: Register Targets**
    Auto Scaling will automatically register instances as targets later in the lab.

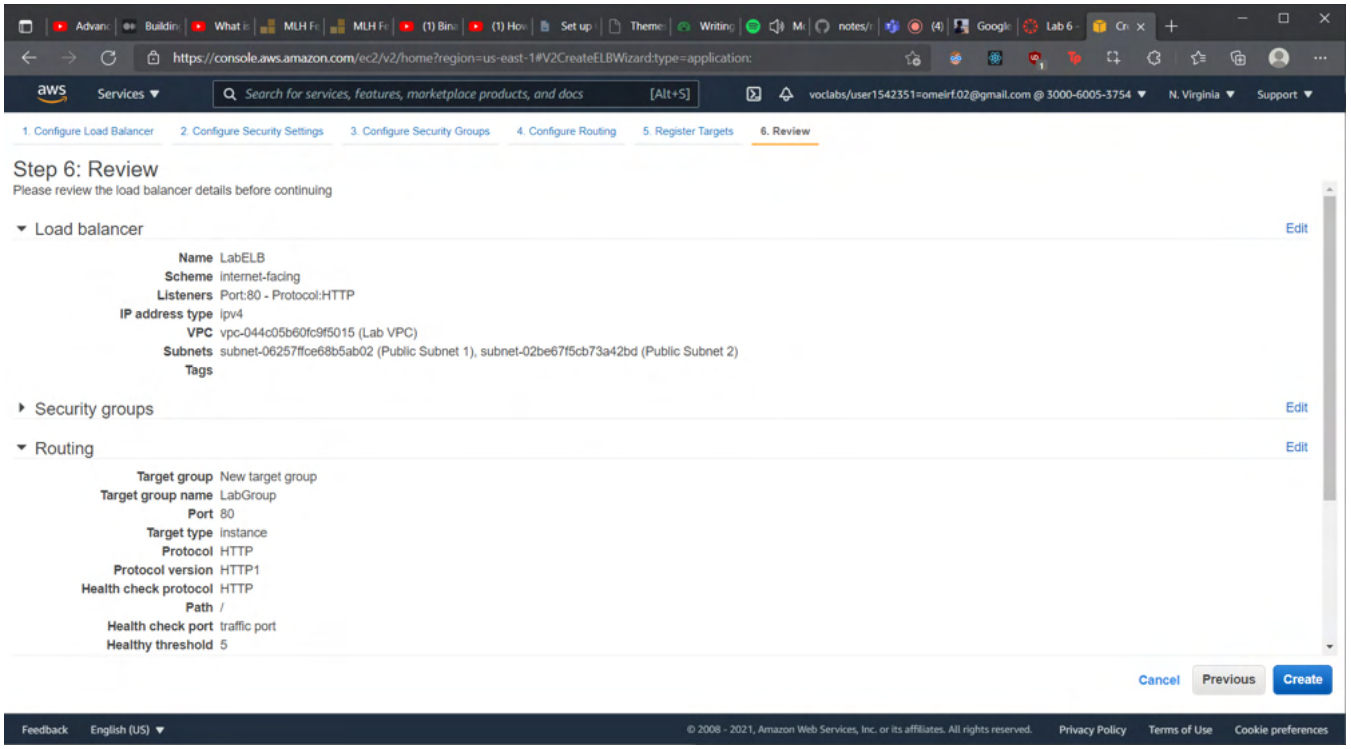


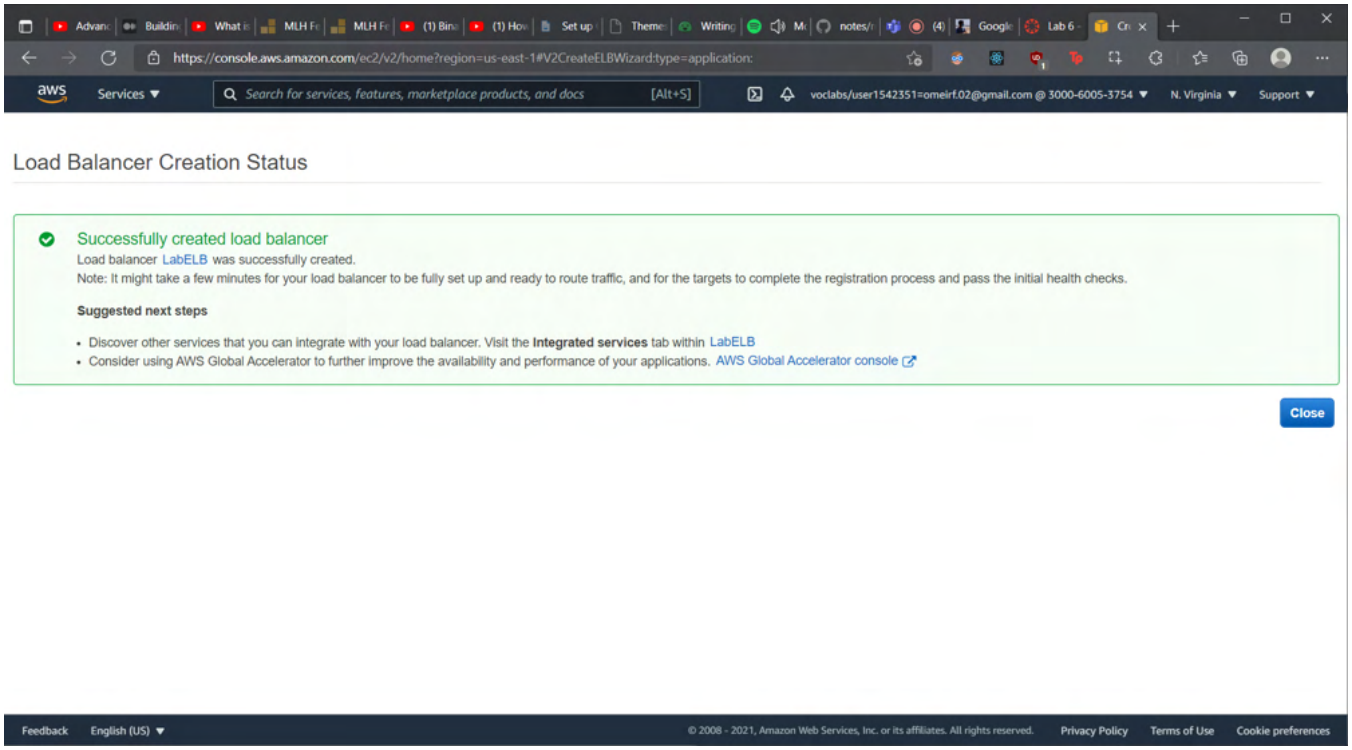
21. Click **Next: Review**



22. Click **Create** then click **Close**

    The load balancer will show a state of *provisioning*. There is no need to wait until it is ready. Please continue with the next task.

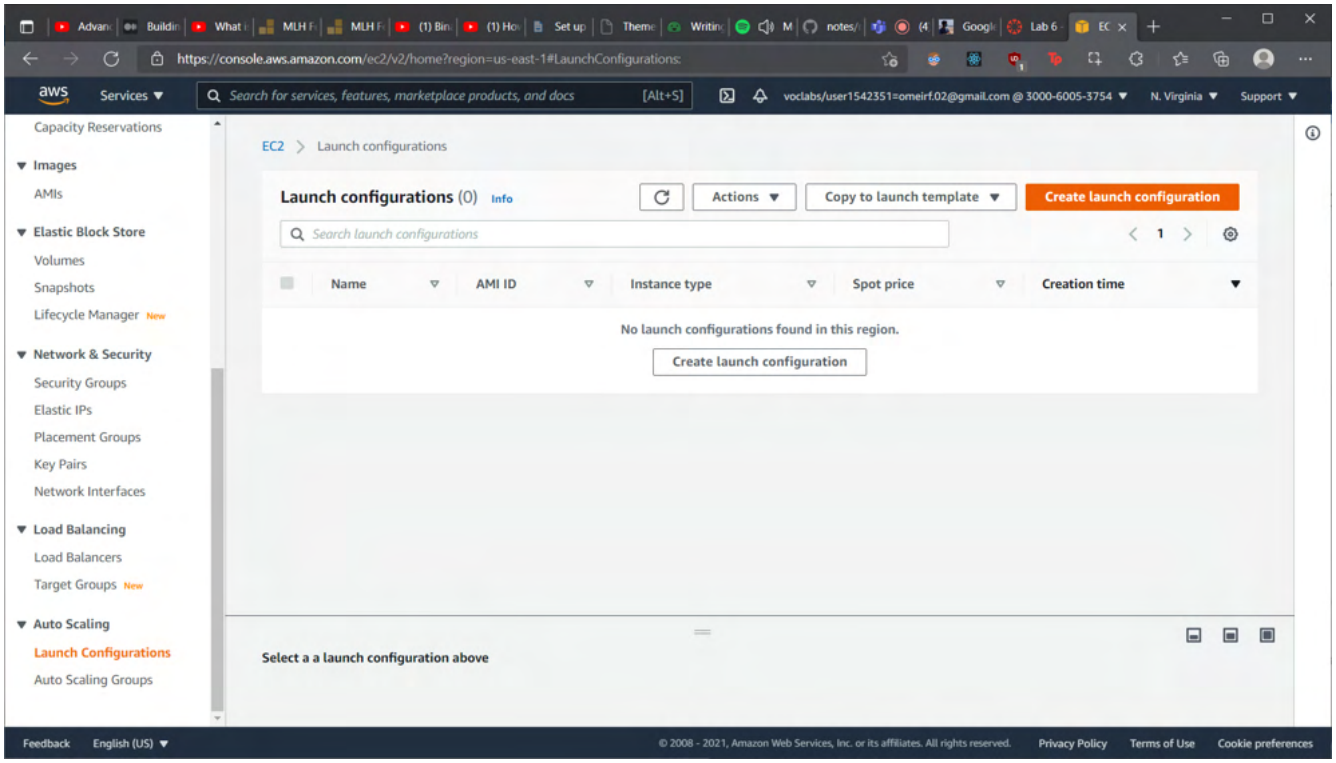




Task 3: Create a Launch Configuration and an Auto Scaling Group

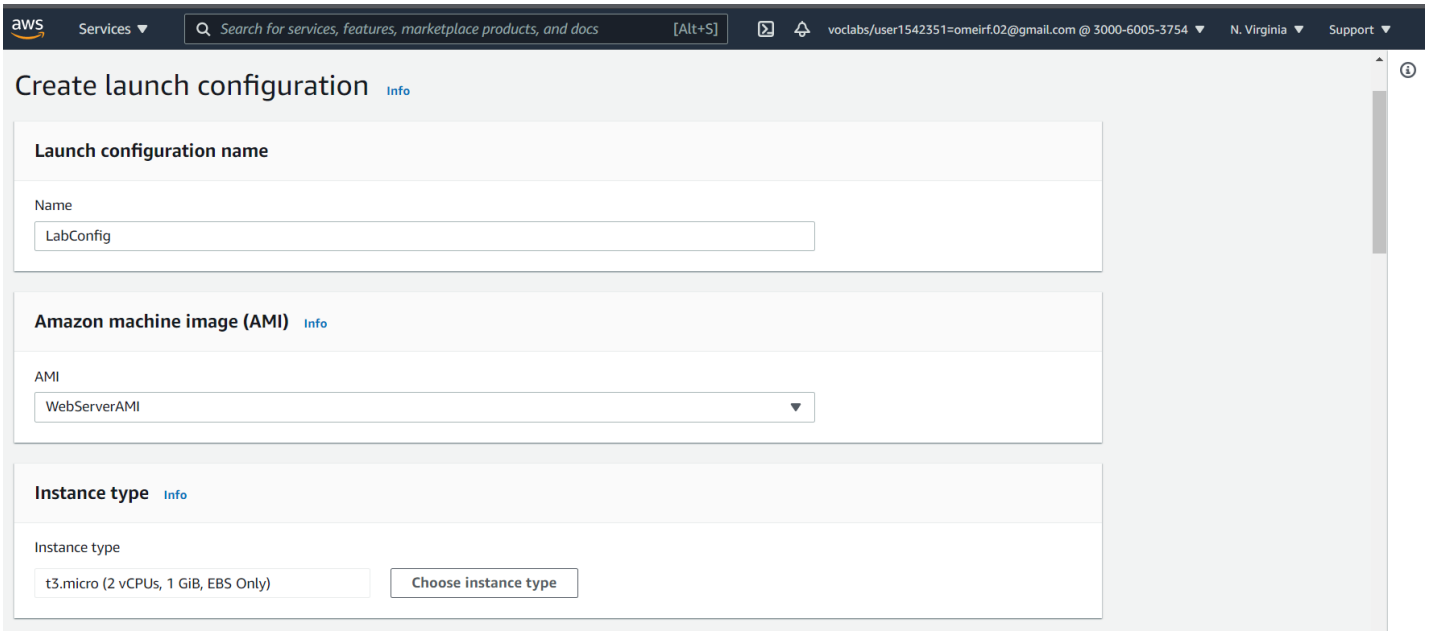
In this task, you will create a *launch configuration* for your Auto Scaling group. A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the AMI, the instance type, a key pair, security group and disks.

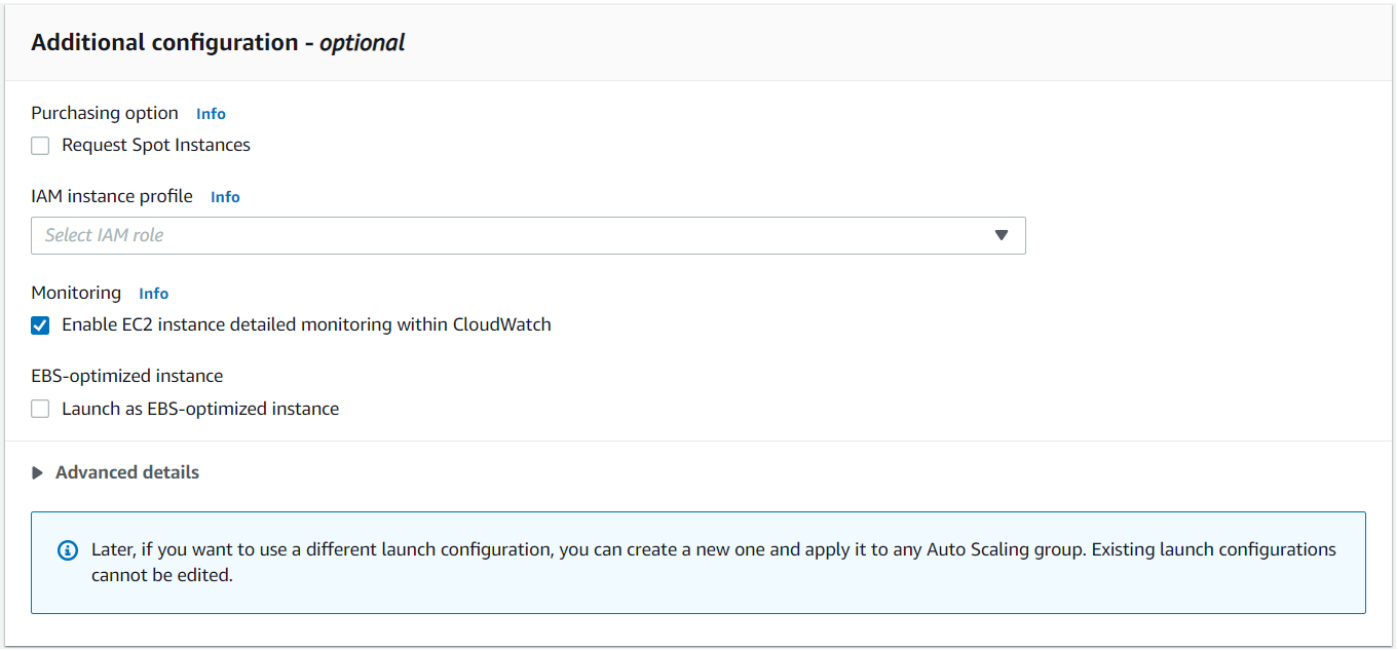
22. In the left navigation pane, click **Launch Configurations**.
23. Click **Create launch configuration**



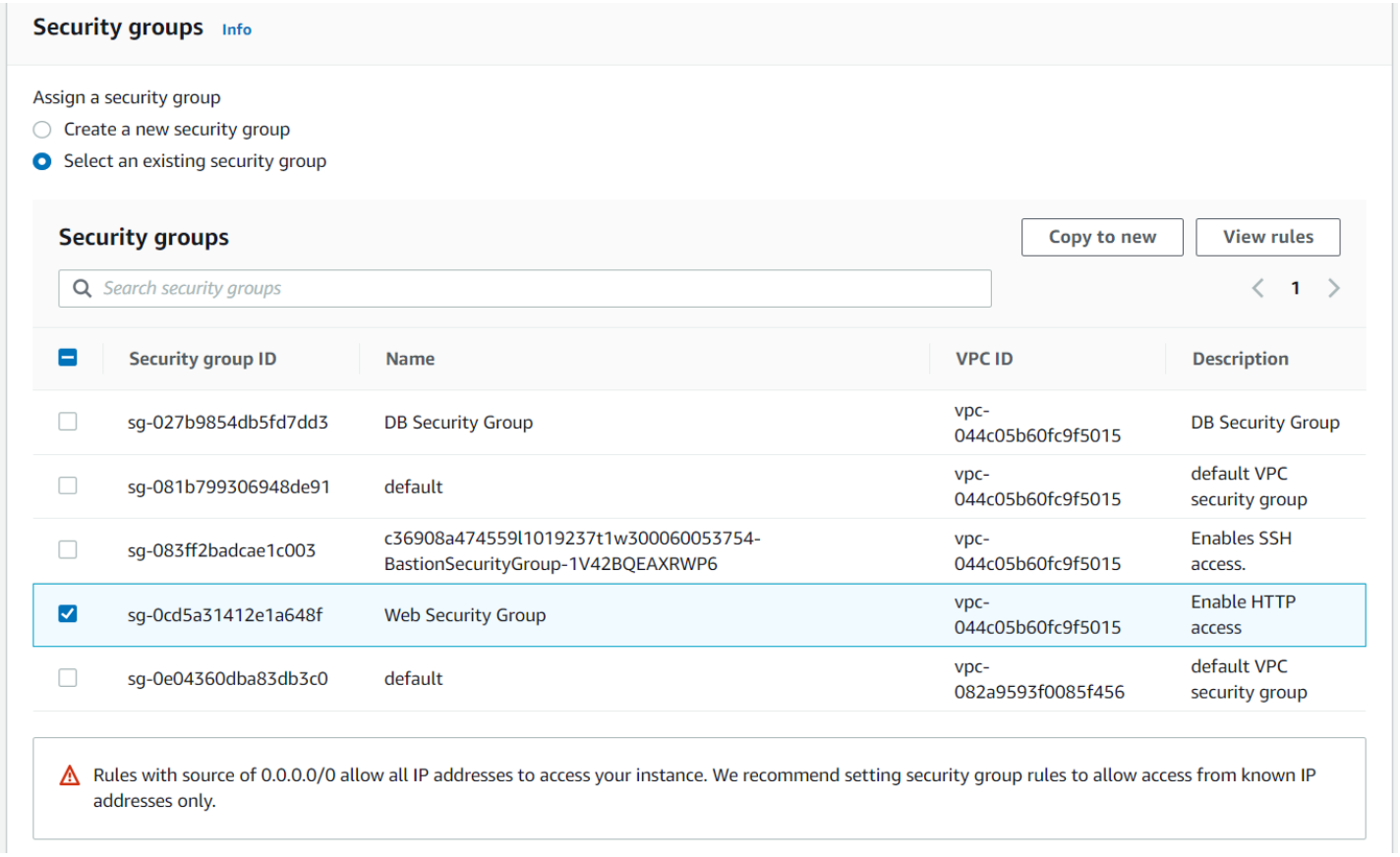
24. Configure these settings:
    - **Launch configuration name:** `LabConfig`
    - **Amazon Machine Image (AMI)** Choose *Web Server AMI*
    - **Instance type:**
        - Choose **Choose instance type**
        - Select *t3.micro*
        - Choose **Choose**
    - **Note:** If you have launched the lab in the us-east-1 Region, select the **t2.micro** instance type. To find the Region, look in the upper right-hand corner of the Amazon EC2 console.

      Note: If you receive the error message "Something went wrong. Please refresh and try again.", you may ignore it and continue with the exercise.
    - **Additional configuration**
        - **Monitoring:** Select *Enable EC2 instance detailed monitoring within CloudWatch*
    - This allows Auto Scaling to react quickly to changing utilization.

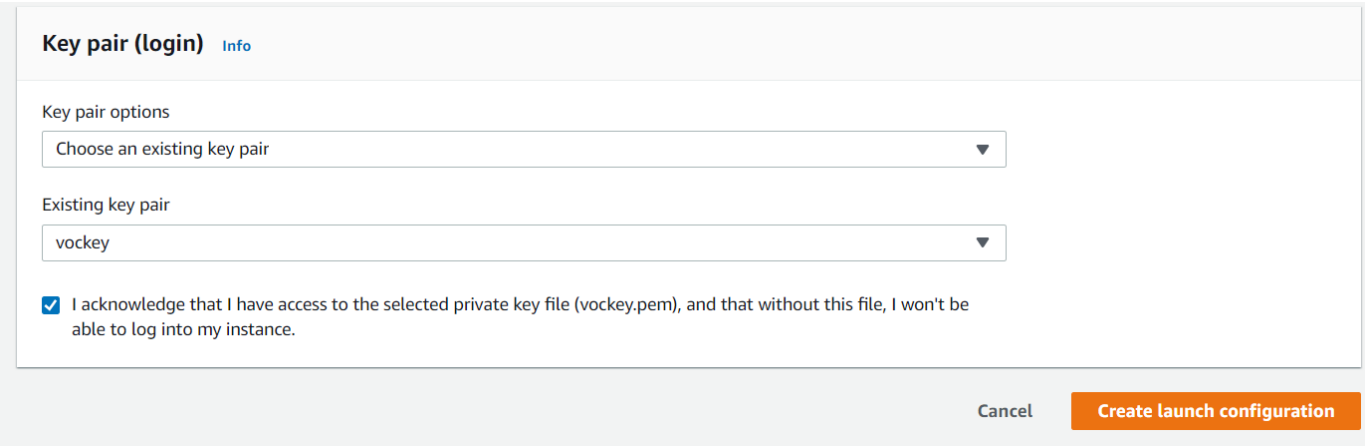




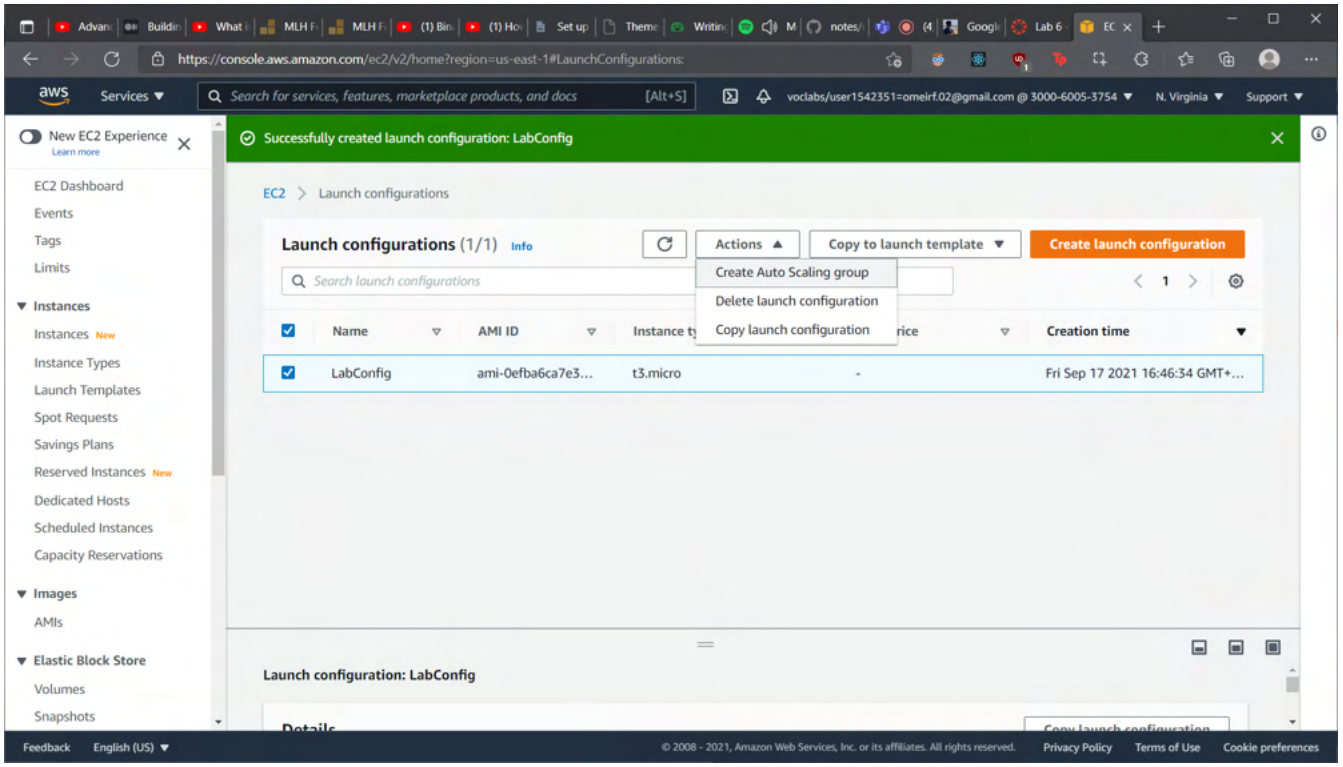
25. Under **Security groups**, you will configure the launch configuration to use the *Web Security Group* that has already been created for you.
    - Choose **Select an existing security group**
    - Select **Web Security Group**



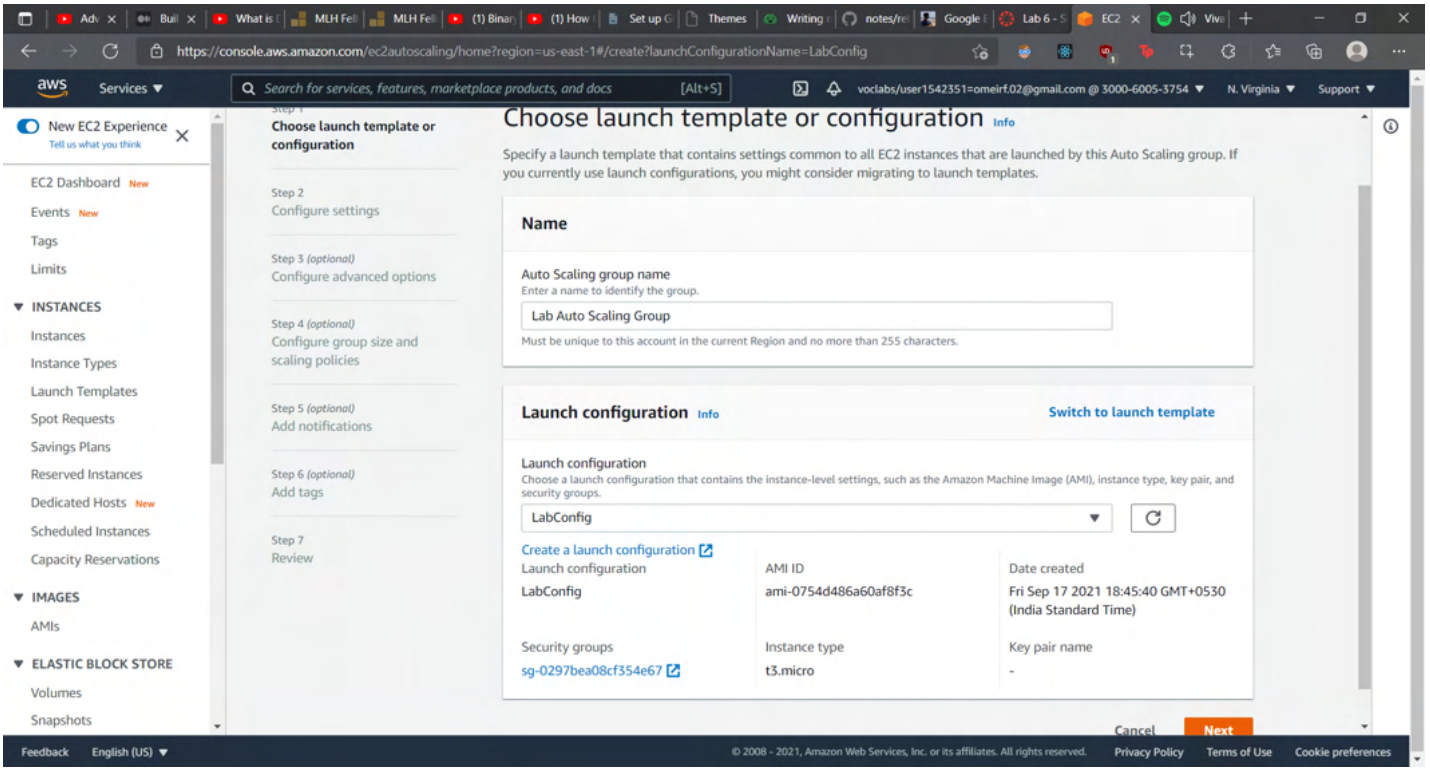
26. Under **Key pair** configure:
    - **Key pair options:** *Choose an existing key pair*
    - **Existing key pair:** vockey
    - Select **I acknowledge...**
    - Click **Create launch configuration**



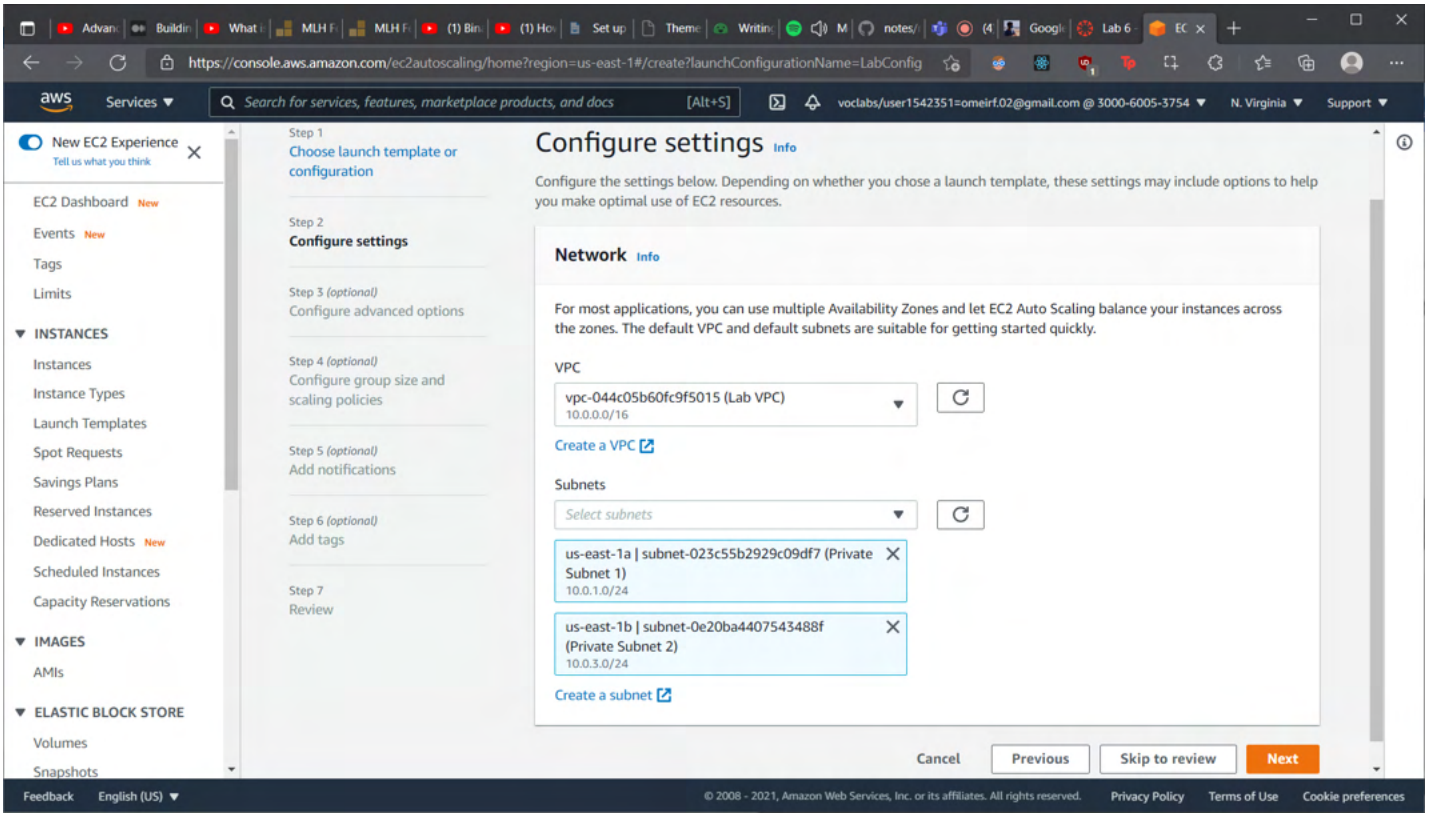
27. You will now create an Auto Scaling group that uses this Launch Configuration.
28. Select the checkbox for the *LabConfig* Launch Configuration.
29. From the **Actions** menu, choose *Create Auto Scaling group*



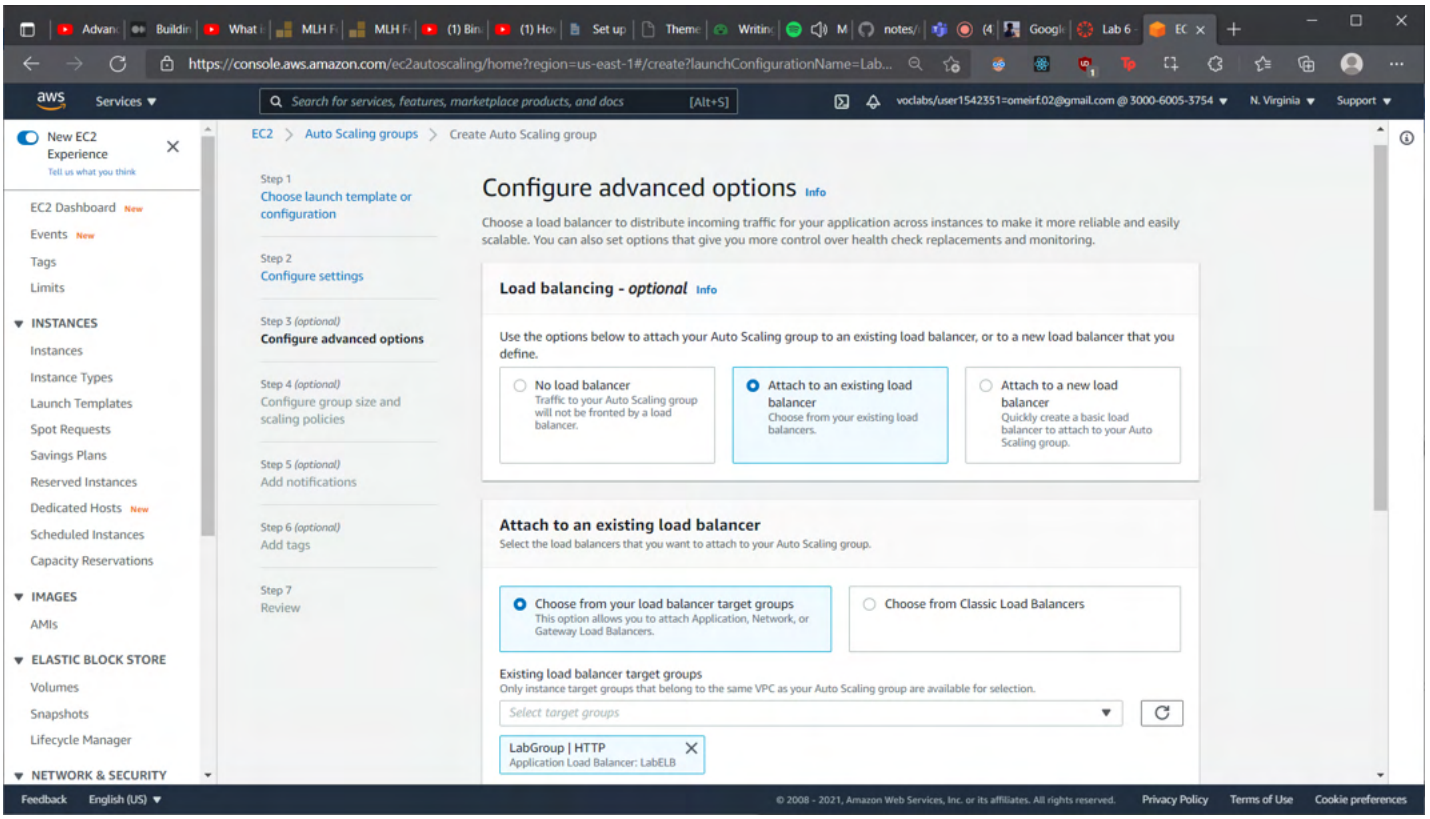
30. Enter Auto Scaling group name:
    - **Name:** `Lab Auto Scaling Group`
31. Choose **Next**



32. On the **Network** page configure
    - **Network:** *Lab VPC*
      You can ignore the message regarding "No public IP address"
    - **Subnet:** Select *Private Subnet 1 (10.0.1.0/24)* **and** *Private Subnet 2 (10.0.3.0/24)*



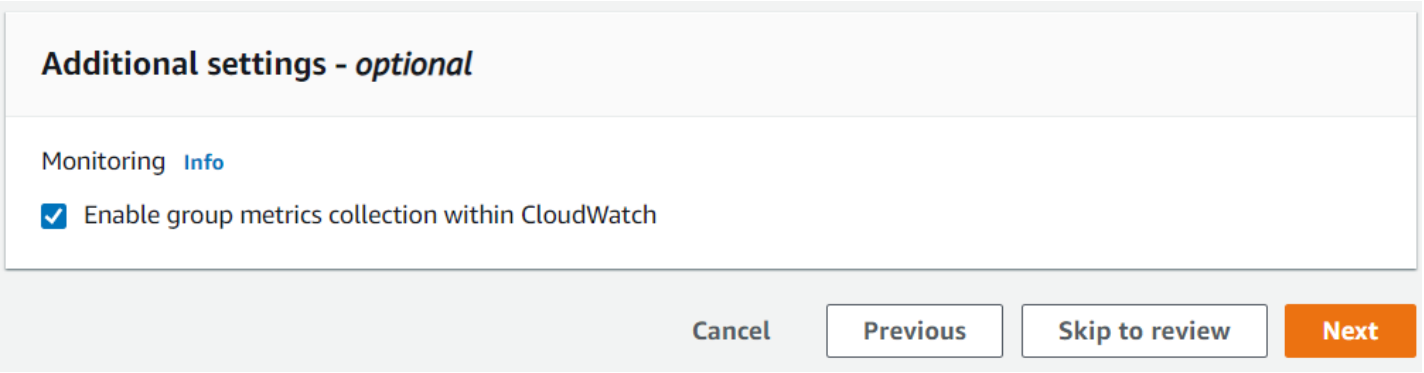
33. This will launch EC2 instances in private subnets across both Availability Zones.
34. Choose **Next**
35. In the **Load balancing - *optional*** pane, choose **Attach to an existing load balancer**
36. In the **Attach to an existing load balancer** pane, use the dropdown list to select *LabGroup*.



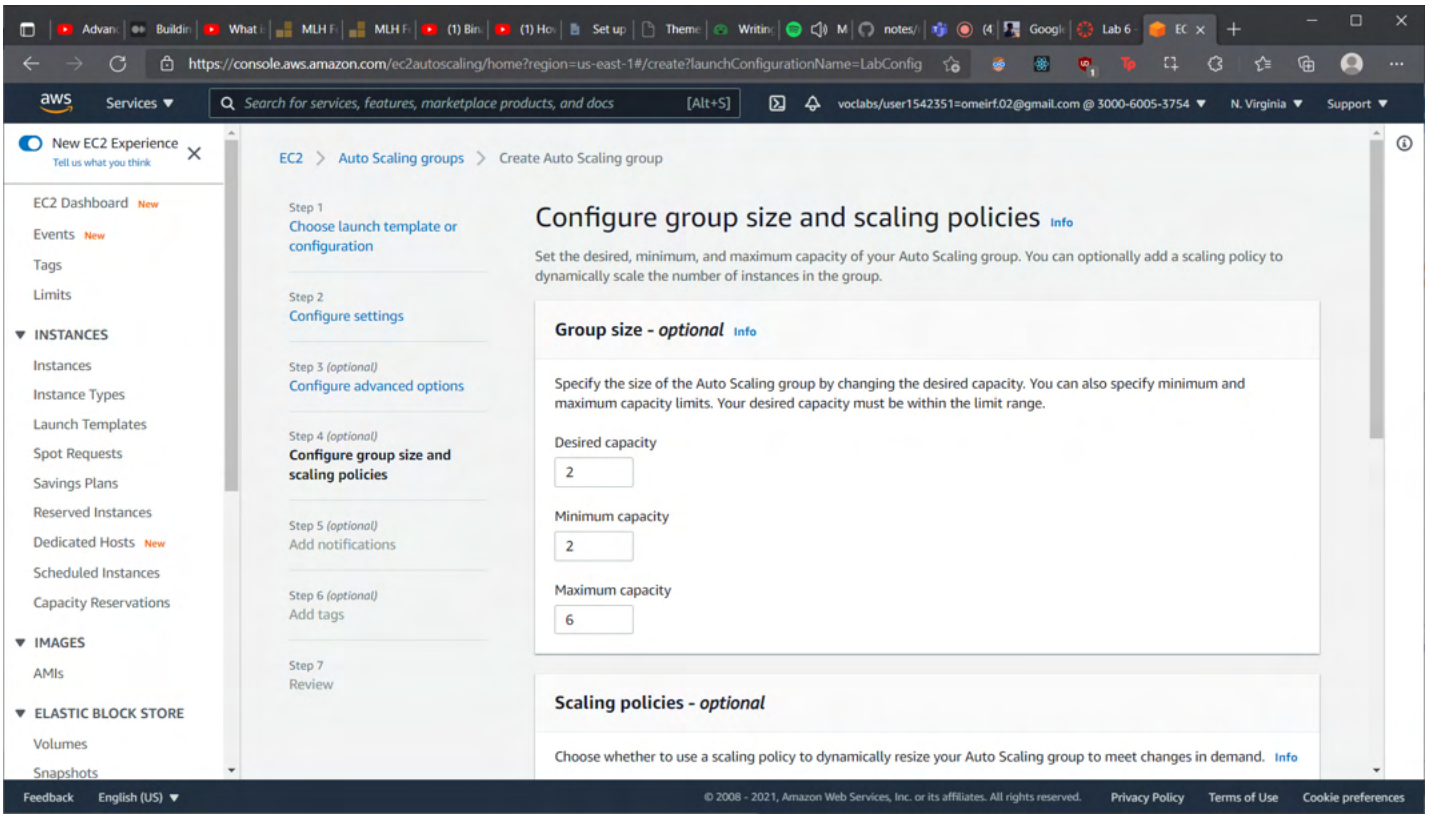
37. In the **Additional settings - *optional*** pane, select **Enable group metrics collection within CloudWatch**

This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.

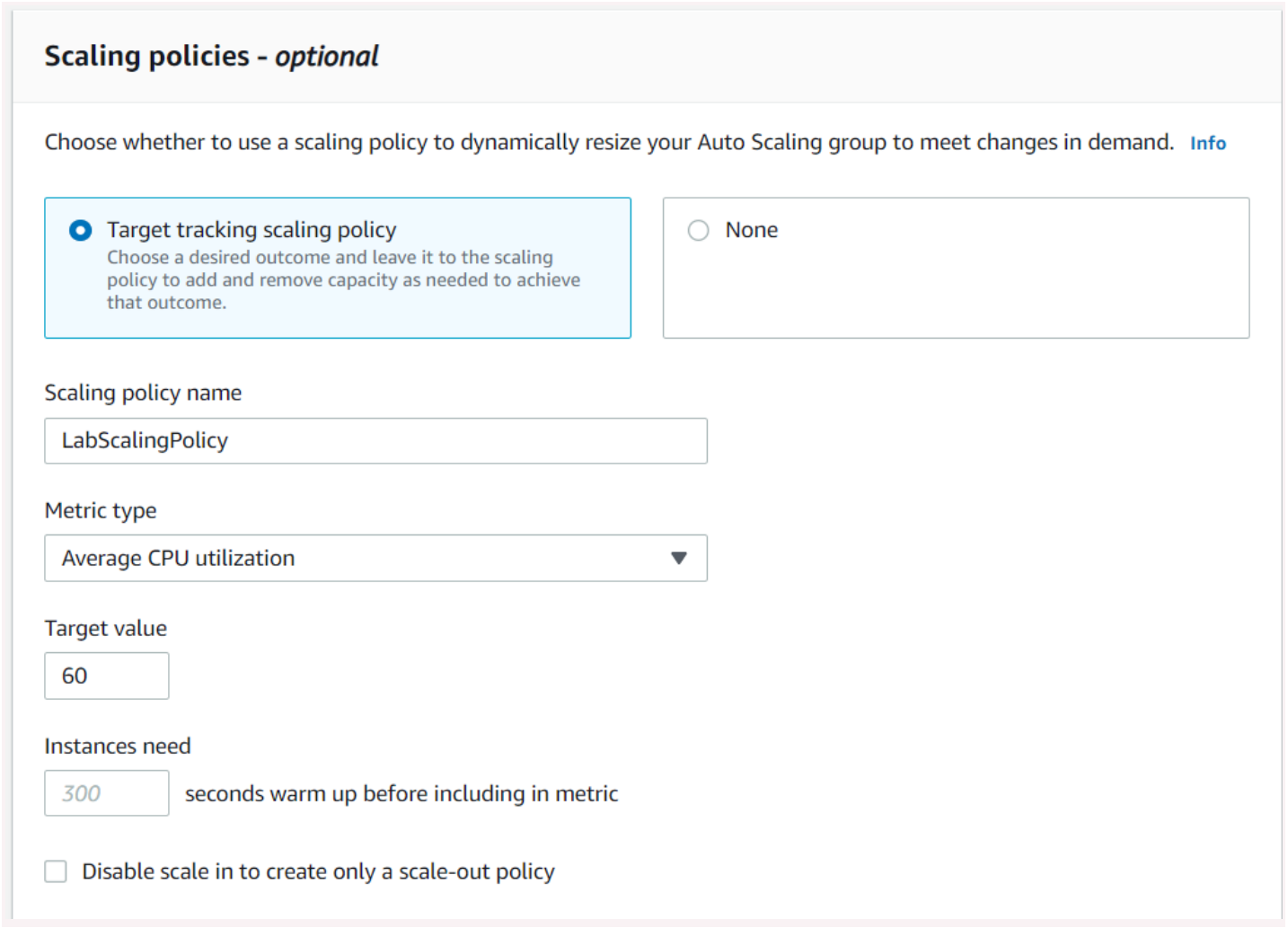
38. Choose **Next**



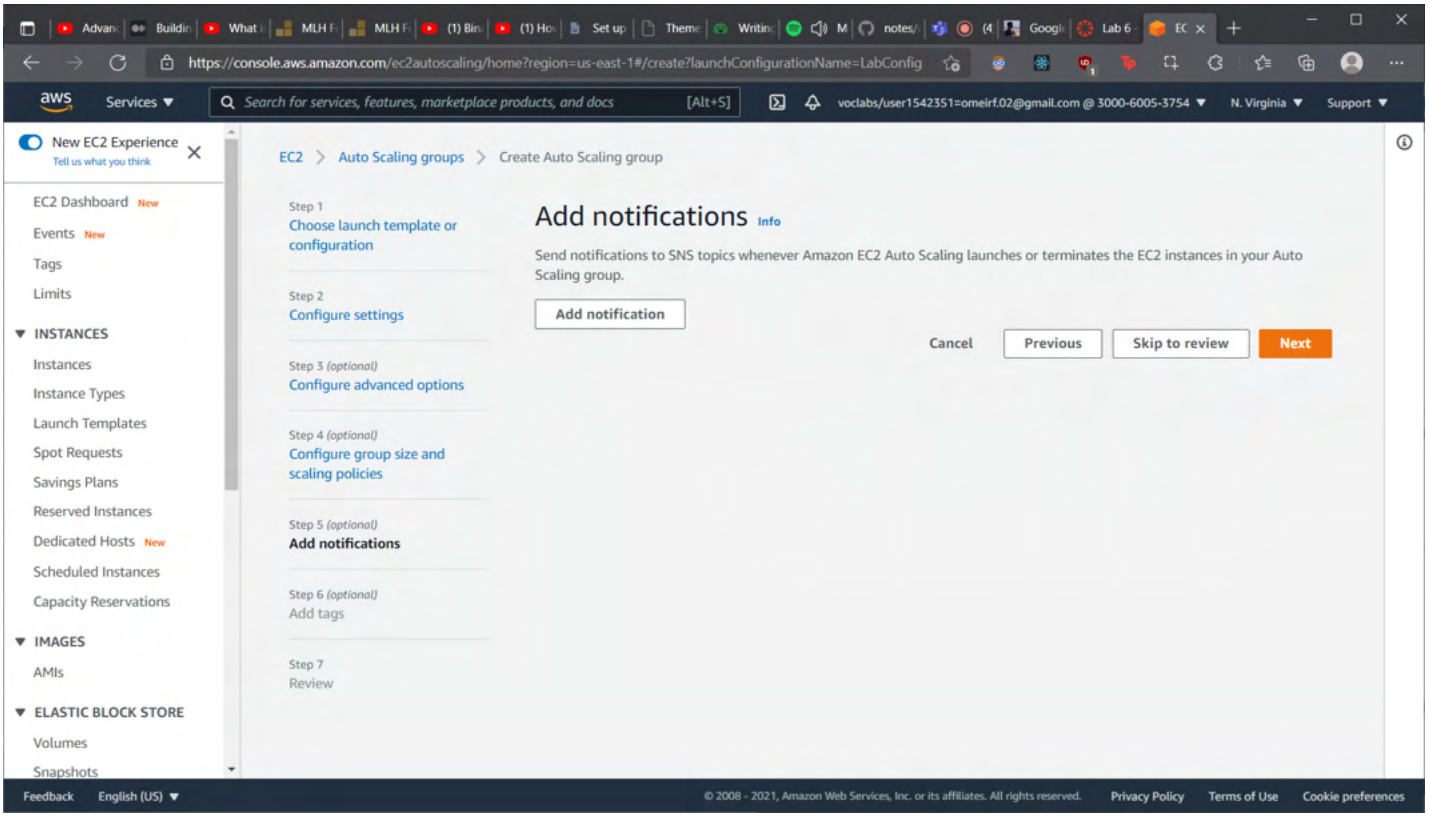
39. Under **Group size**, configure:
    - **Desired capacity:** 2
    - **Minimum capacity:** 2
    - **Maximum capacity:** 6
40. This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.



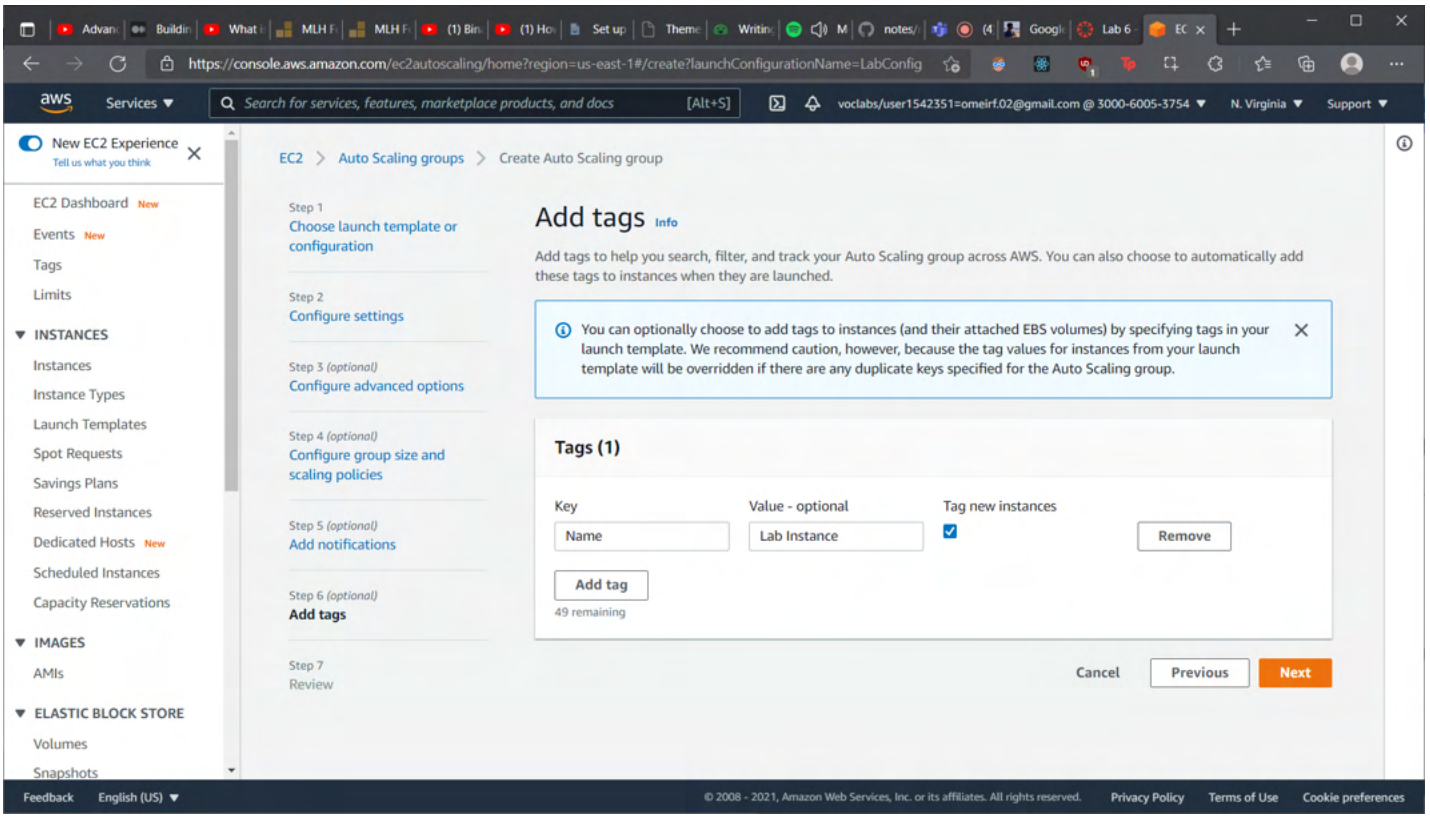
41. Under **Scaling policies**, choose *Target tracking scaling policy* and configure:
    - **Lab policy name:** `LabScalingPolicy`
    - **Metric type:** *Average CPU Utilization*
    - **Target value:** `60`



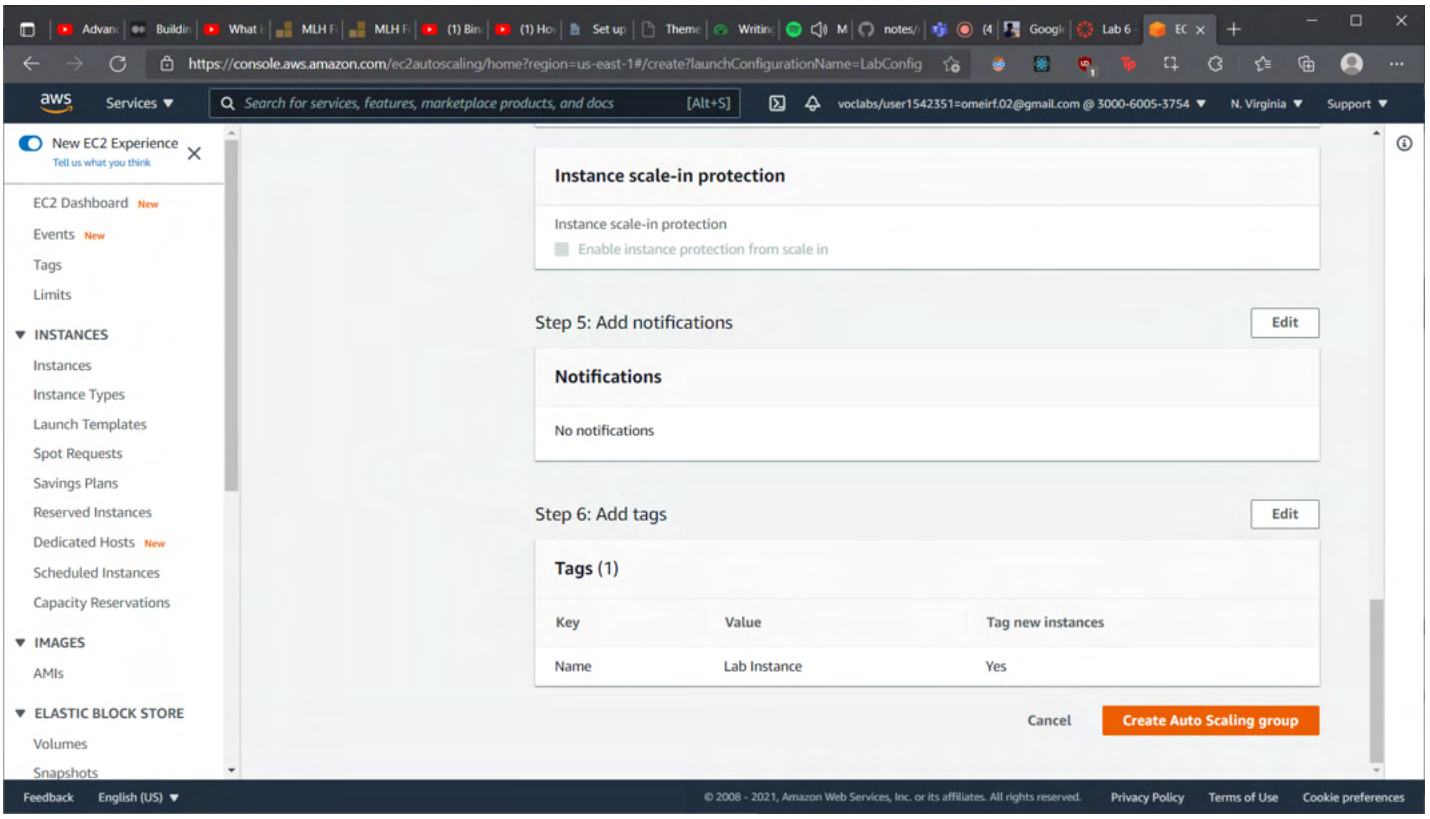
42. This tells Auto Scaling to maintain an *average* CPU utilization *across all instances* at 60%. Auto Scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.
43. Choose **Next**
    Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.

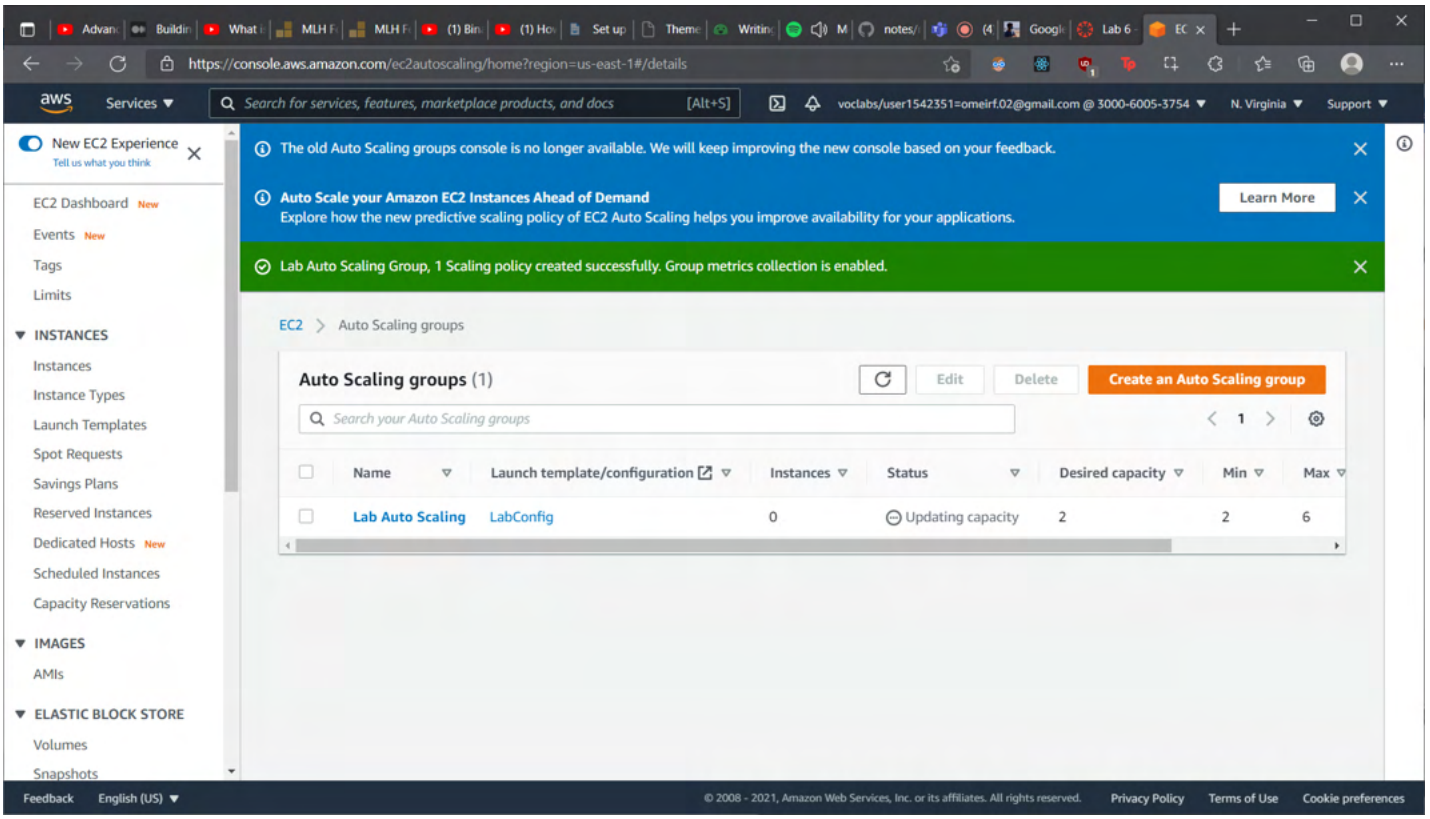


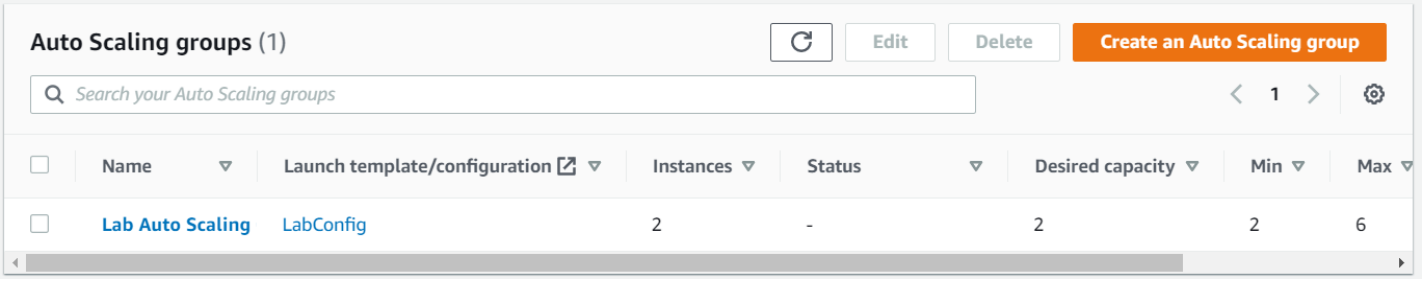
44. Choose **Next**
    Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.
45. Choose **Add tag** and Configure the following:
    - **Key:** `Name`
    - **Value:** `Lab Instance`
46. Click **Next**



47. Review the details of your Auto Scaling group, then click **Create Auto Scaling group**. If you encounter an error **Failed to create Auto Scaling group**, then click **Retry Failed Tasks**.
Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.



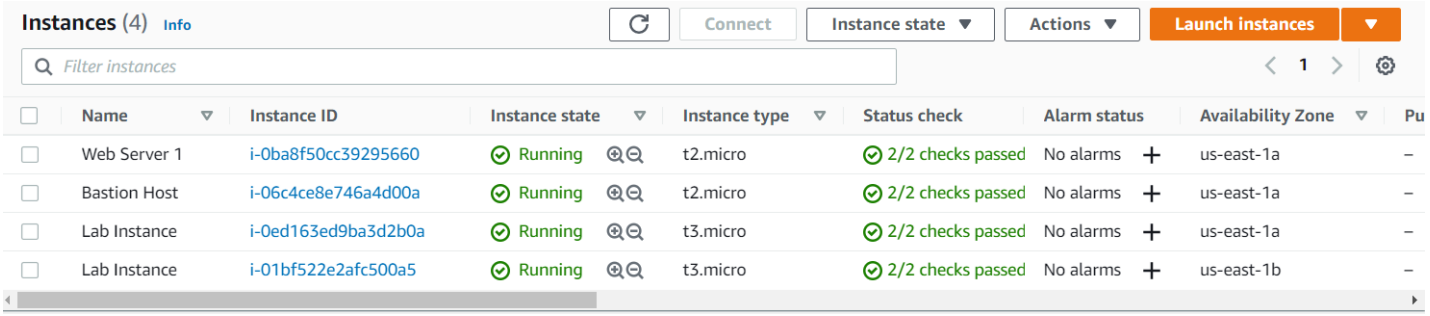




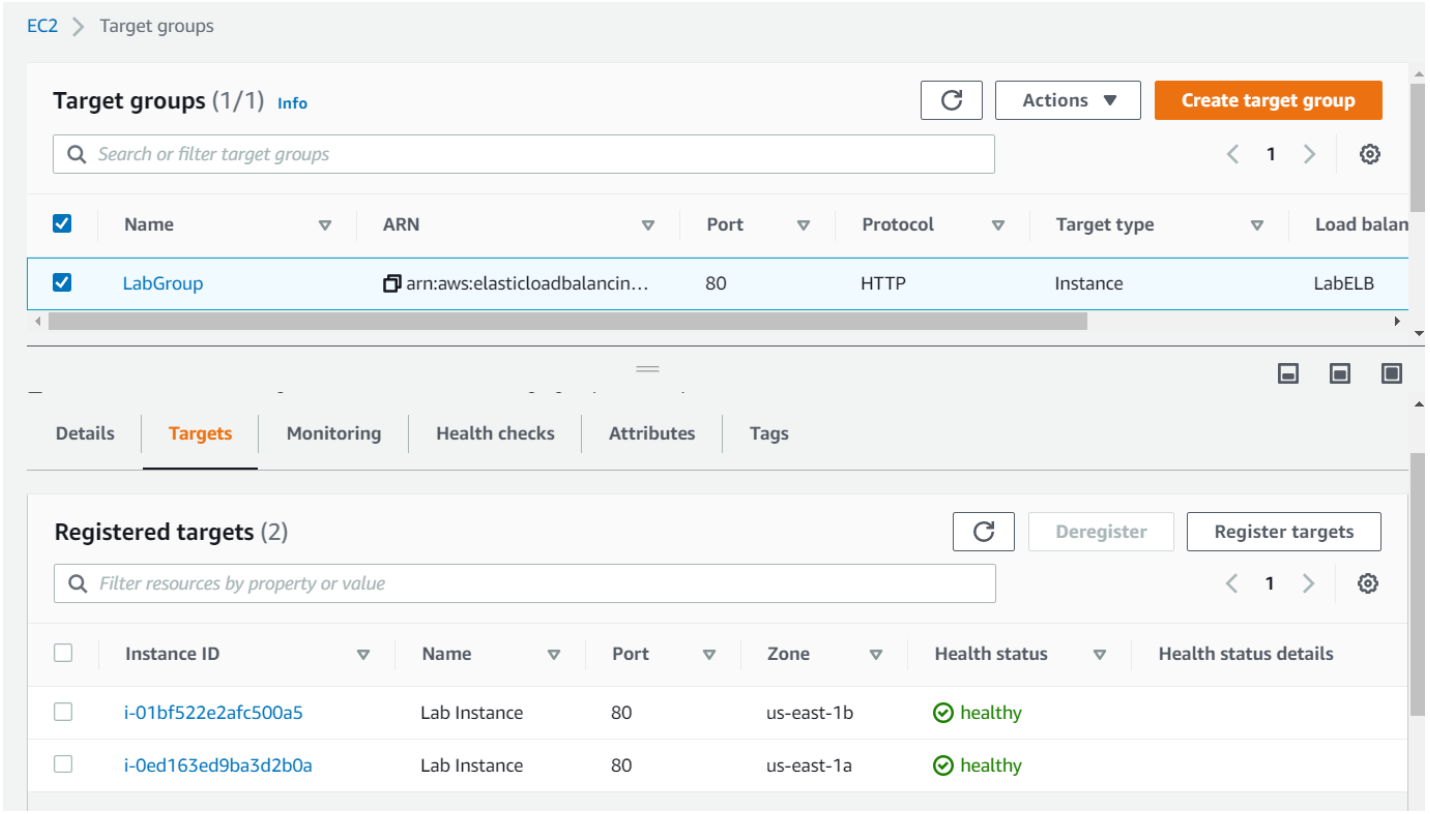
Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

44. In the left navigation pane, click **Instances**.
    You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.
    If the instances or names are not displayed, wait 30 seconds and click refresh in the top-right.
    First, you will confirm that the new instances have passed their Health Check.



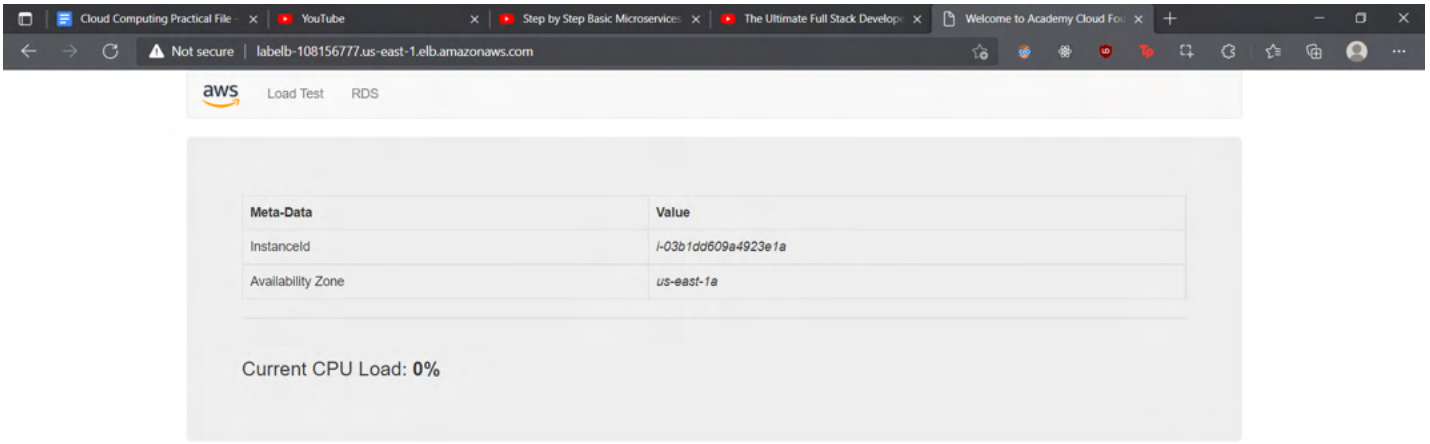
45. In the left navigation pane, click **Target Groups** (in the *Load Balancing* section).
46. Choose *LabGroup*
47. Click the **Targets** tab.
    Two **Lab Instance** targets should be listed for this target group.
48. Wait until the **Status** of both instances transitions to *healthy*. Click Refresh in the upper-right to check for updates.
    Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.
    You can now access the Auto Scaling group via the Load Balancer.



49. In the left navigation pane, click **Load Balancers**.
50. In the lower pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)". It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*

    DNS Name: LabELB-108156777.us-east-1.elb.amazonaws.com

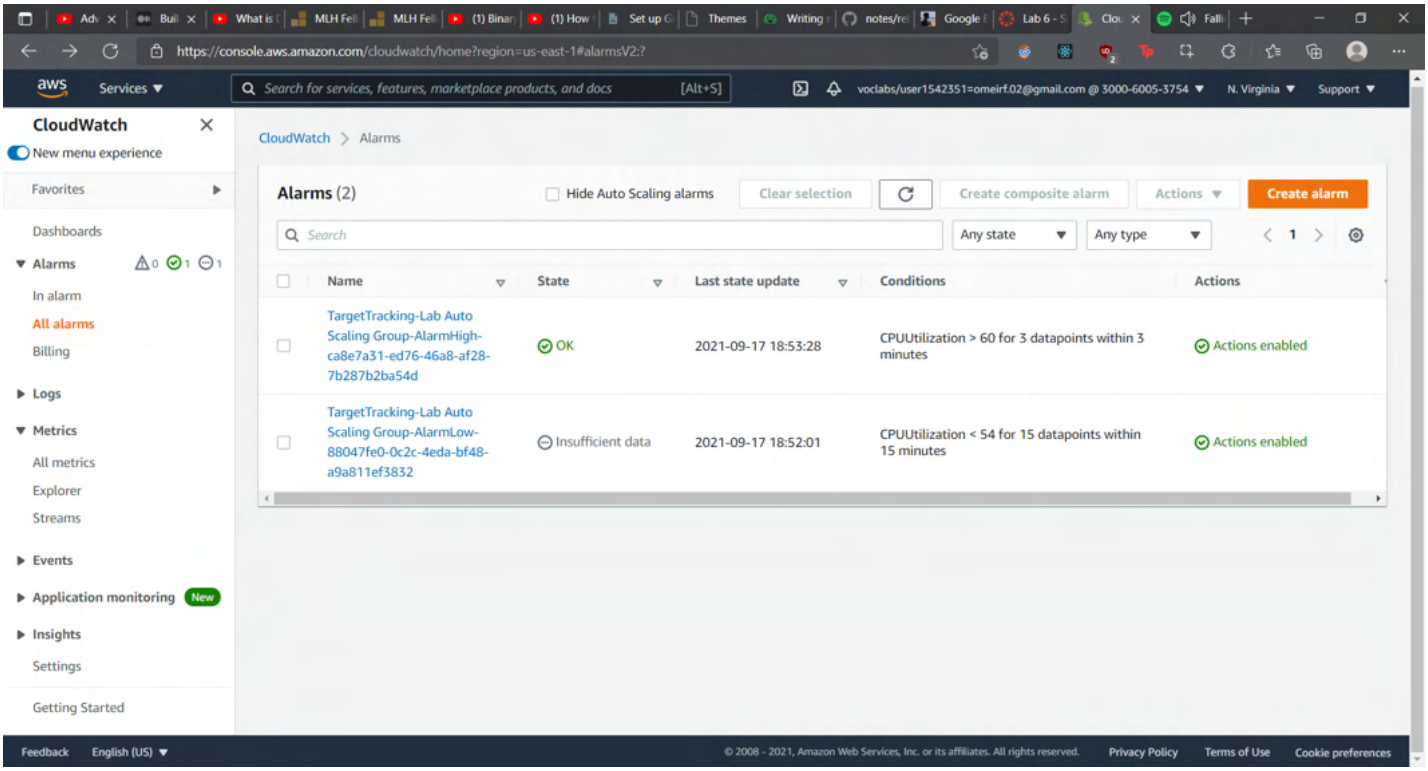
51. Open a new web browser tab, paste the DNS Name you just copied, and press Enter. The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.



Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

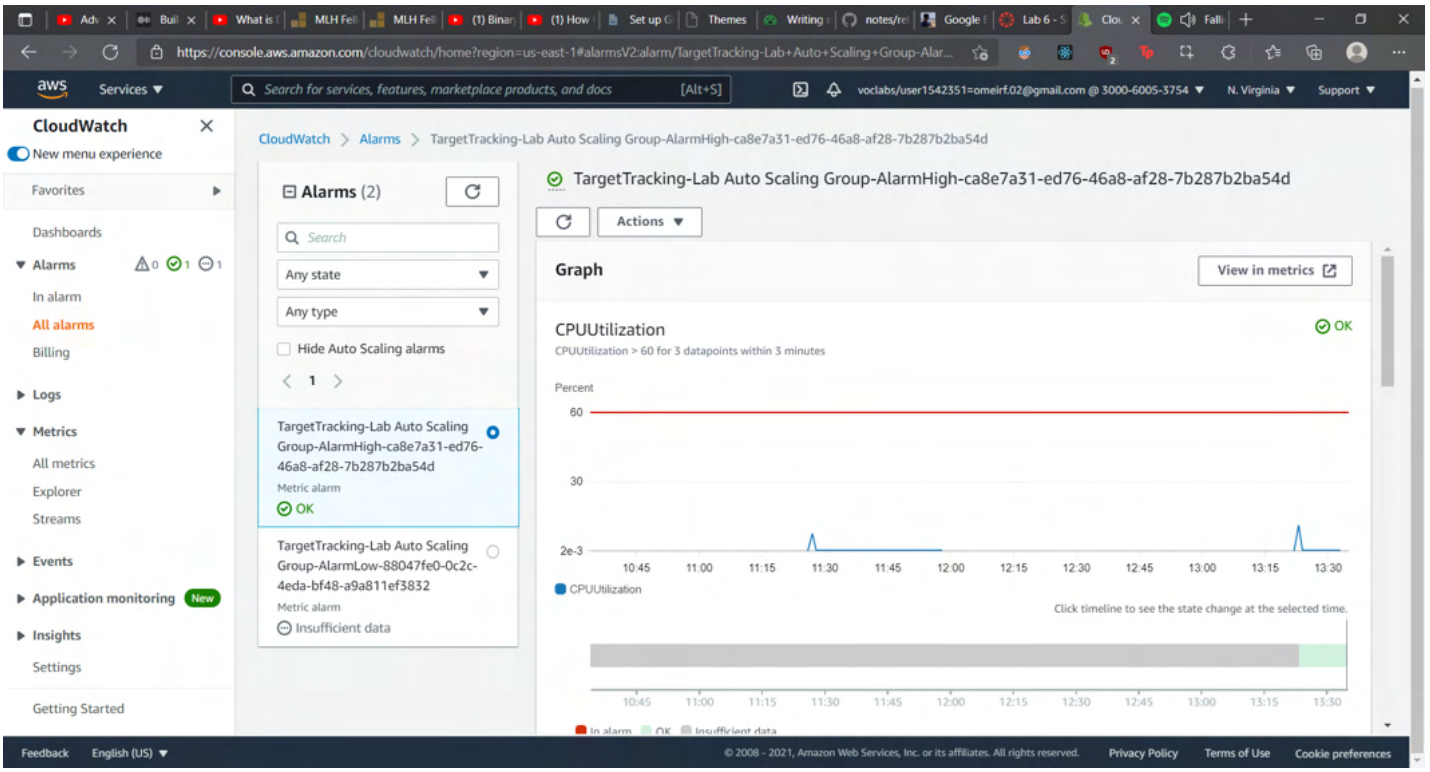
52. Return to the AWS management console, but do not close the application tab — you will return to it soon.
53. On the **Services** menu, click **CloudWatch**.
54. In the left navigation pane, click **Alarms** (*not* **ALARM**).
    Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.
    Note: Please follow these steps only if you do not see the alarms in 60 seconds.
    - On the **Services** menu, click **EC2**.
    - In the left navigation pane, choose **Auto Scaling Groups**.
    - Select **Lab Auto Scaling Group**.
    - In the bottom half of the page, choose the **Automatic Scaling** tab.
    - Select **LabScalingPolicy**.
    - Click **Actions** and **Edit**.
    - Change the **Target Value** to `50`.
    - Click **Update**
    - On the **Services** menu, click **CloudWatch**.
    - In the left navigation pane, click **Alarms** (*not* **ALARM**) and verify you see two alarms.



55. Click the **OK** alarm, which has *AlarmHigh* in its name.
    If no alarm is showing **OK**, wait a minute then click refresh in the top-right until the alarm status changes.

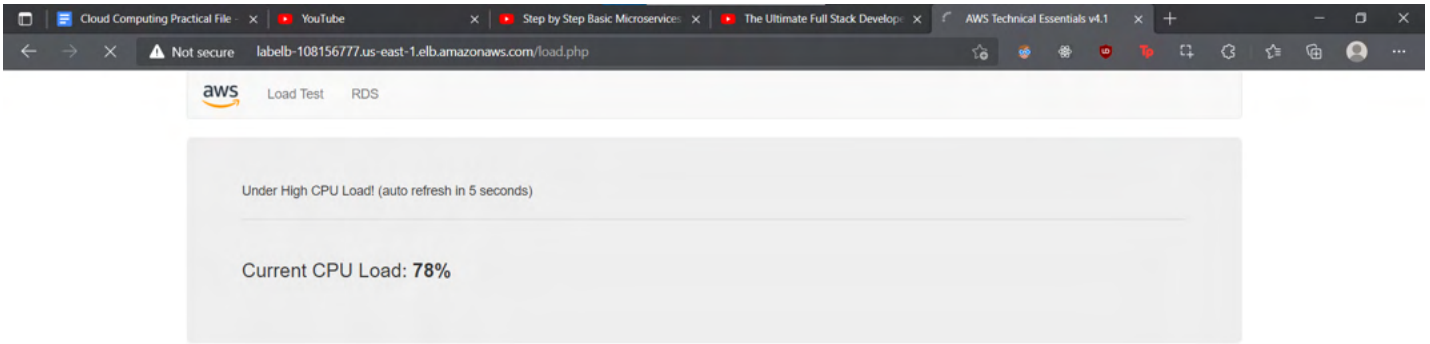
The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.

You will now tell the application to perform calculations that should raise the CPU level.



56. Return to the browser tab with the web application.
57. Click **Load Test** beside the AWS logo.

    This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.

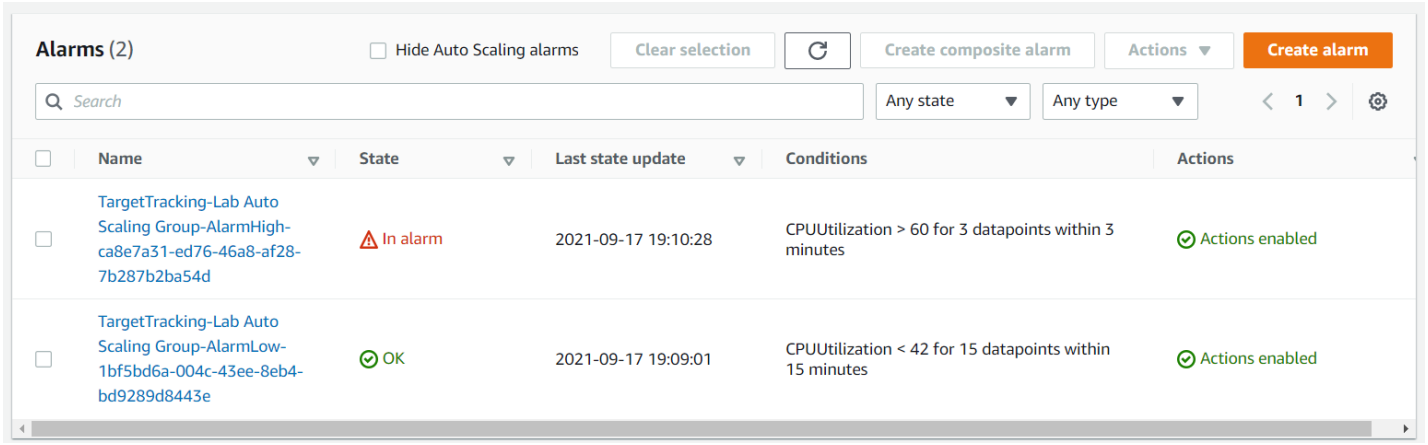


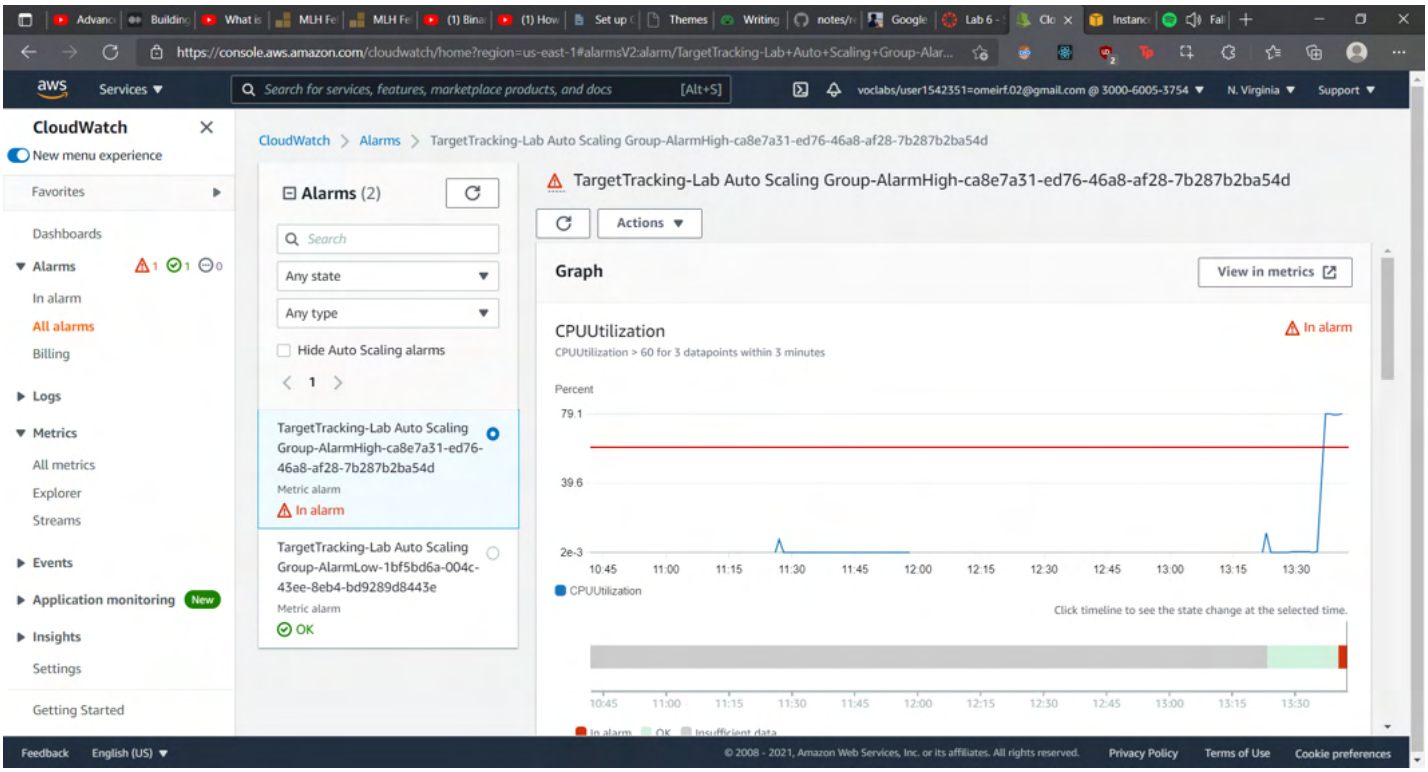
58. Return to the browser tab with the **CloudWatch** console.

    In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *ALARM*.

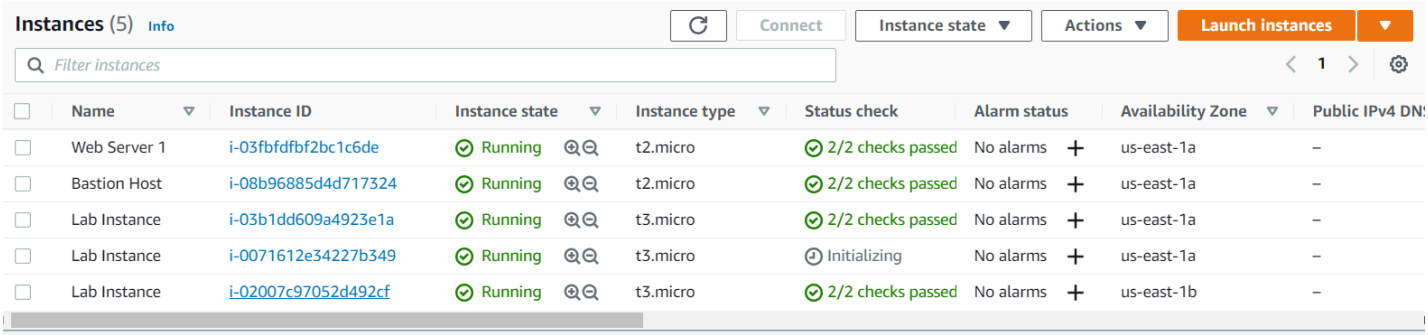
    You can click Refresh in the top-right every 60 seconds to update the display.

    You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.





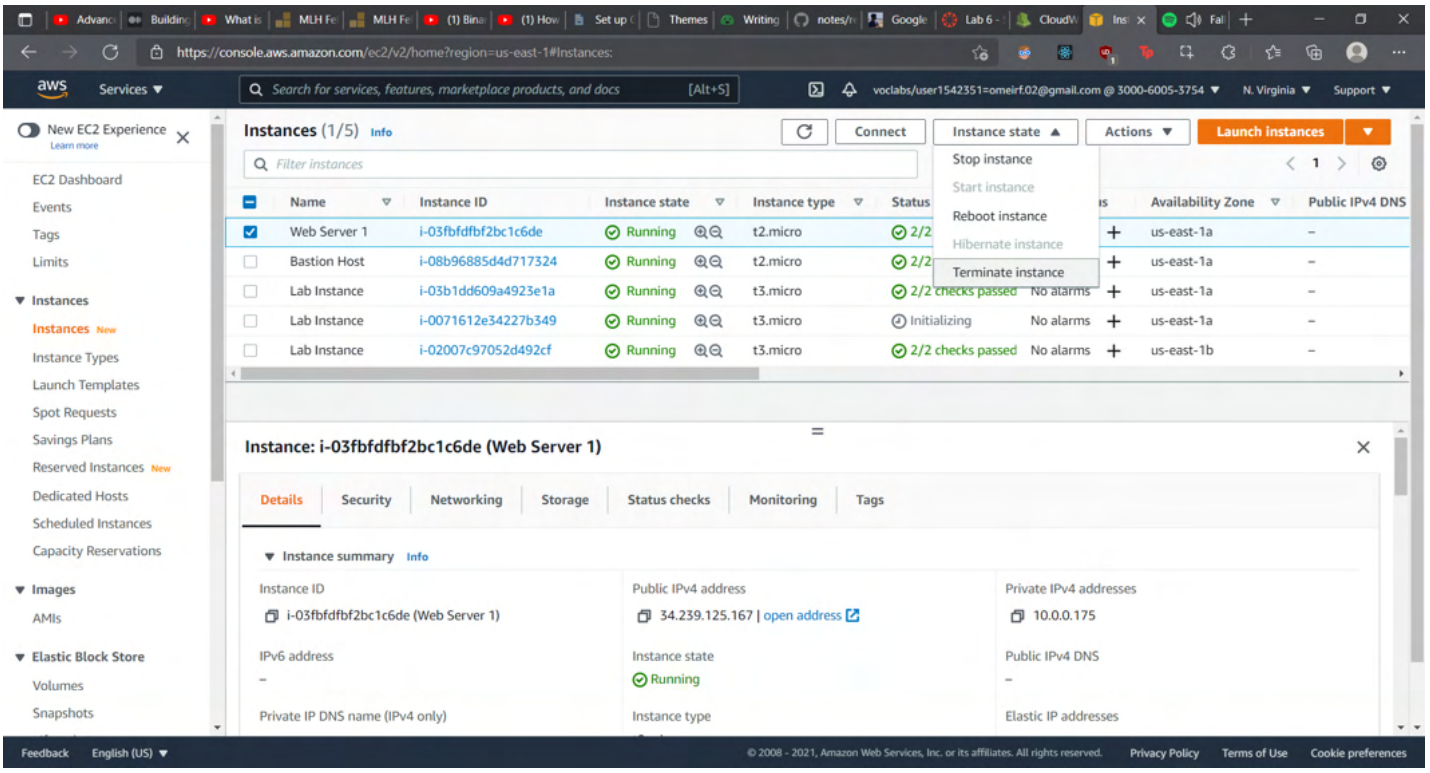
59. Wait until the **AlarmHigh** alarm enters the *ALARM* state.
    You can now view the additional instance(s) that were launched.
60. On the **Services** menu, click **EC2**.
61. In the left navigation pane, click **Instances**.
    More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the Alarm.



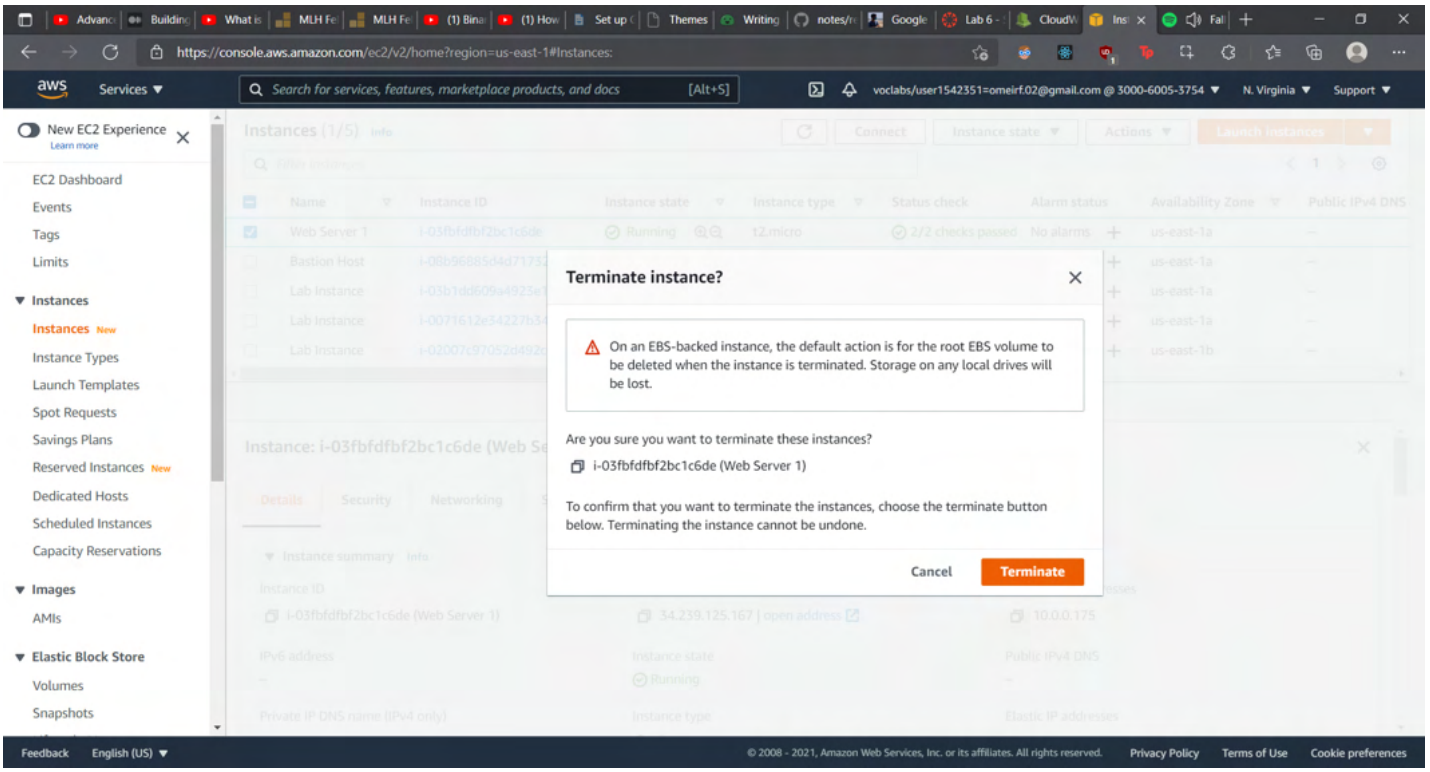
Task 6: Terminate Web Server 1

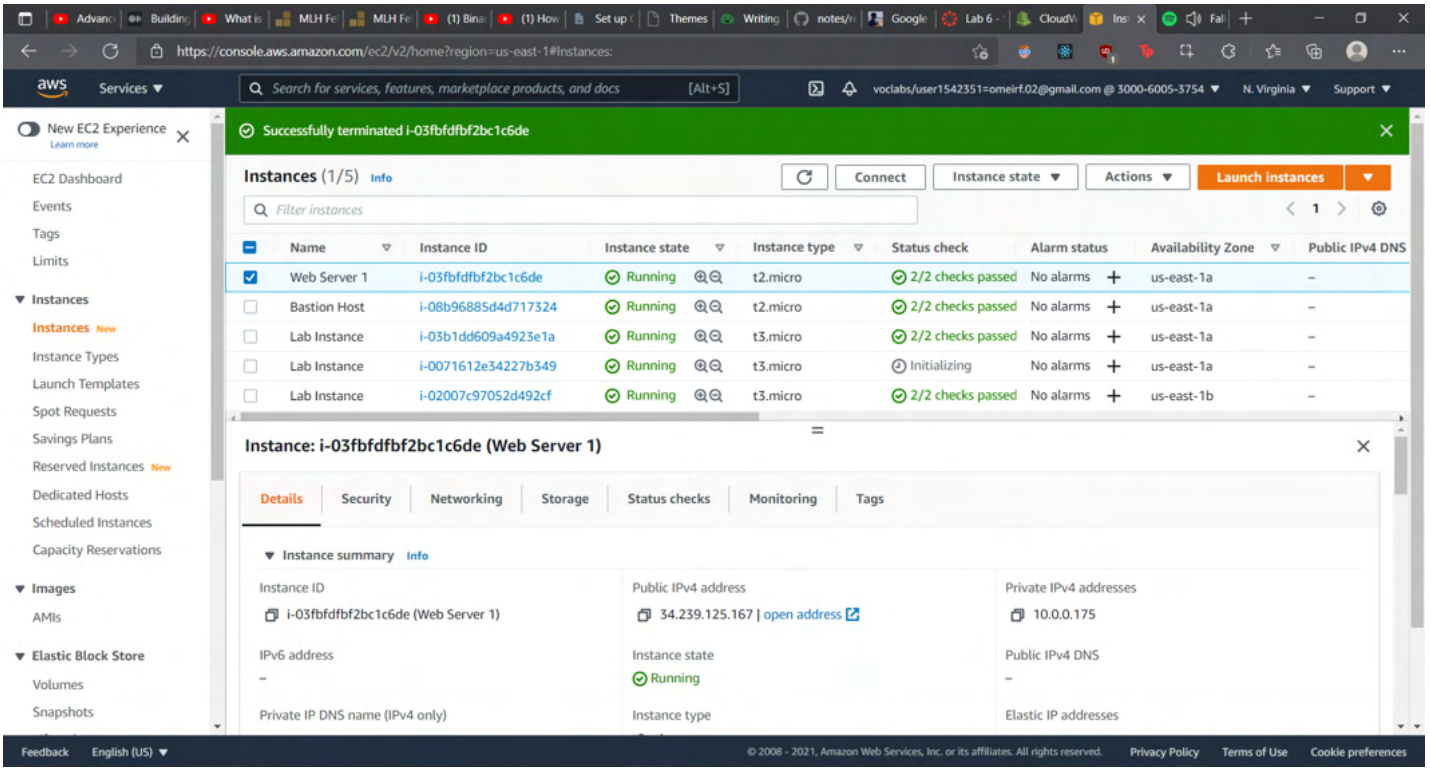
In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

62. Select **Web Server 1** (and ensure it is the only instance selected).
63. In the **Instance state** menu, click **Instance State** > **Terminate Instance**.



64. Choose **Terminate**



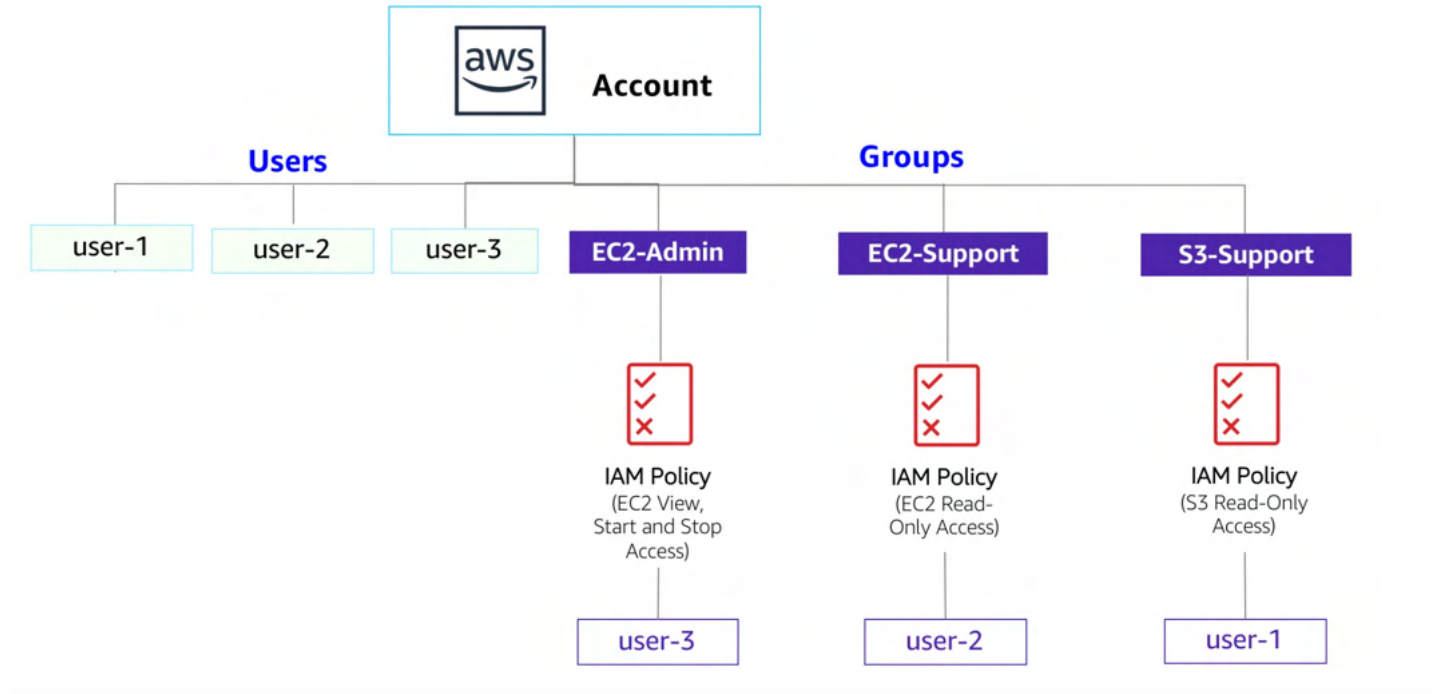

https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:
Services
Search for services, features, marketplace products, and docs [Alt+S]
voclabs/user1542351=omeirf.02@gmail.com @ 3000-6005-3754
N. Virginia
Support
New EC2 Experience
Learn more
EC2 Dashboard
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Successfully terminated i-03fbfdfbf2bc1c6de
Instances (1/5) Info
Connect
Instance state
Actions
Launch instances
Filter instances
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Web Server 1 i-03fbfdfbf2bc1c6de Running t2.micro 2/2 checks passed No alarms us-east-1a
Bastion Host i-08b96885d4d717324 Running t2.micro 2/2 checks passed No alarms us-east-1a
Lab Instance i-03b1dd609a4923e1a Running t3.micro 2/2 checks passed No alarms us-east-1a
Lab Instance i-0071612e34227b349 Running t3.micro Initializing No alarms us-east-1a
Lab Instance i-02007c97052d492cf Running t3.micro 2/2 checks passed No alarms us-east-1b
Instance: i-03fbfdfbf2bc1c6de (Web Server 1)
Details
Security
Networking
Storage
Status checks
Monitoring
Tags
Instance summary Info
Instance ID
i-03fbfdfbf2bc1c6de (Web Server 1)
Public IPv4 address
34.239.125.167 | open address
Private IPv4 addresses
10.0.0.175
IPv6 address
Instance state
Running
Public IPv4 DNS
Private IP DNS name (IPv4 only)
Instance type
Elastic IP addresses
Feedback
English (US)
© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
Cookie preferences

Practical 6

AIM

AWS Identity and Access Management (IAM) is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS. With IAM, you can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.

THEORY



This lab will demonstrate:

- *Exploring pre-created IAM Users and Groups*
- *Inspecting IAM policies as applied to the pre-created groups*
- *Following a real-world scenario, adding users to groups with specific capabilities enabled*
- *Locating and using the IAM sign-in URL*
- *Experimenting with the effects of policies on service access*

Other AWS Services

During this lab, you may receive error messages when performing actions beyond the steps in this lab guide. These messages will not impact your ability to complete the lab.

AWS Identity and Access Management

AWS IAM can be used to:

- *Manage IAM Users and their access: You can create Users and assign them individual security credentials (access keys, passwords, and multi-factor authentication devices). You can manage permissions to control which operations a User can perform.*
- *Manage IAM Roles and their permissions: An IAM Role is similar to a User, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a Role is intended to be assumable by anyone who needs it.*

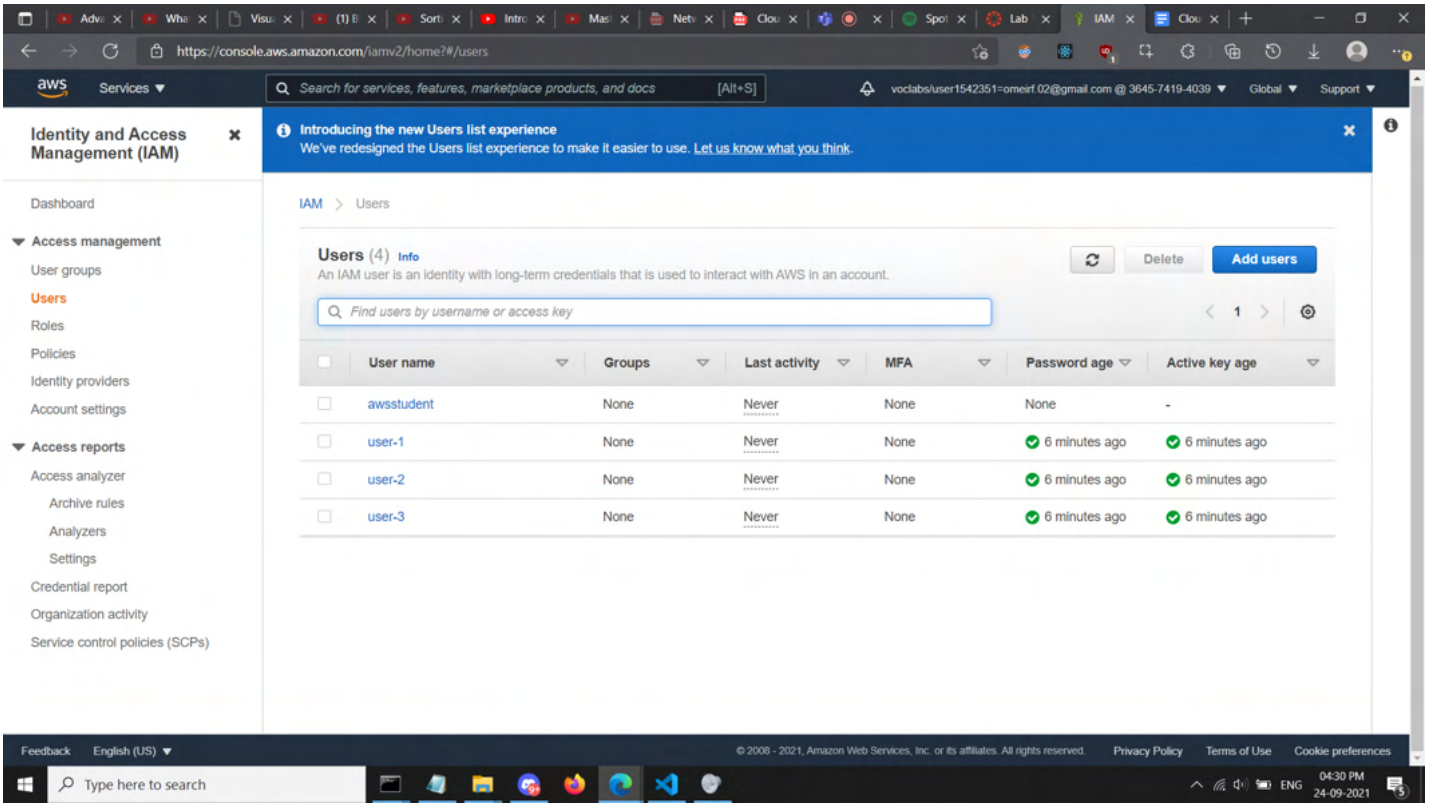
- *Manage federated users and their permissions: You can enable identity federation to allow existing users in your enterprise to access the AWS Management Console, to call AWS APIs and to access resources, without the need to create an IAM User for each identity.*

PROCEDURE

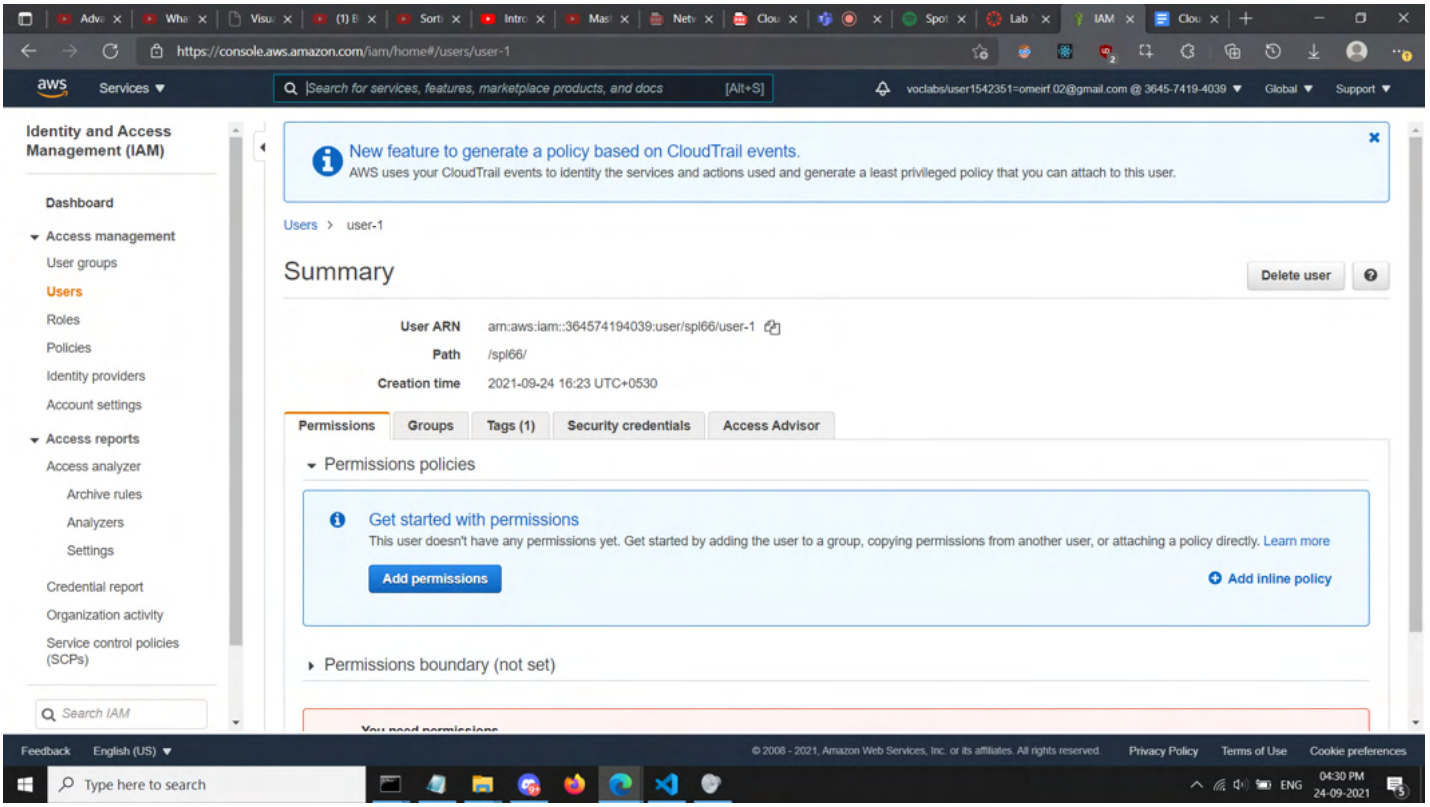
Task 1: Explore the Users and Groups

In this task, you will explore the Users and Groups that have already been created for you in IAM.

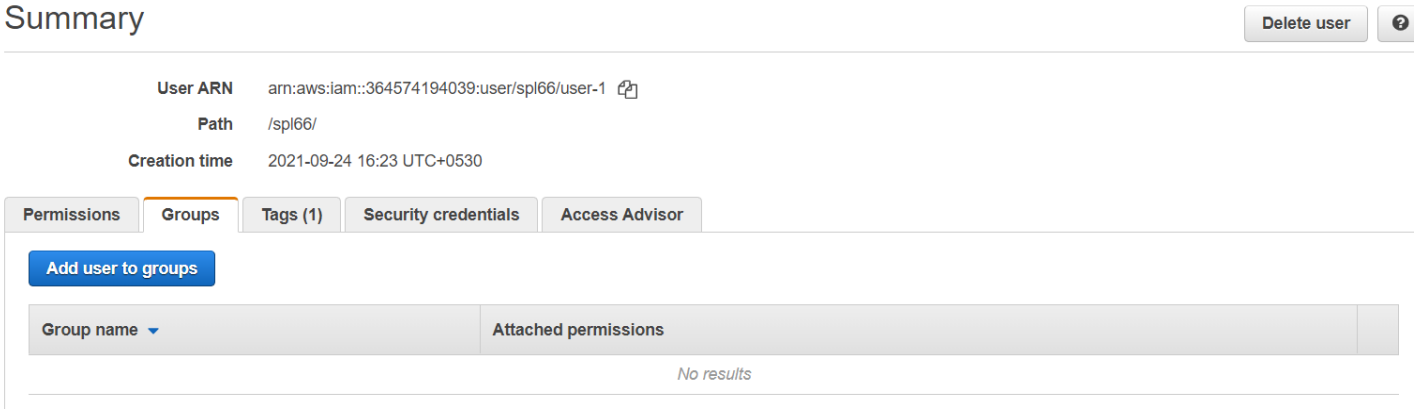
5. In the **AWS Management Console**, on the **Services** menu, click **IAM**.
6. In the navigation pane on the left, click **Users**.
   The following IAM Users have been created for you:
   - user-1
   - user-2
   - user-3



7. Click **user-1**.
   This will bring to a summary page for user-1. The **Permissions** tab will be displayed.
8. Notice that user-1 does not have any permissions.



9. Click the **Groups** tab.
   user-1 also is not a member of any groups.



10. Click the **Security credentials** tab.
    user-1 is assigned a **Console password**

Summary

Delete user

User ARN	arn:aws:iam::364574194039:user/spl66/user-1
Path	/spl66/
Creation time	2021-09-24 16:23 UTC+0530

Permissions | Groups | Tags (1) | Security credentials | Access Advisor

Sign-in credentials

Summary	• Console sign-in link: https://364574194039.signin.aws.amazon.com/console
Console password	Enabled (never signed in) | Manage
Assigned MFA device	Not assigned | Manage
Signing certificates	None

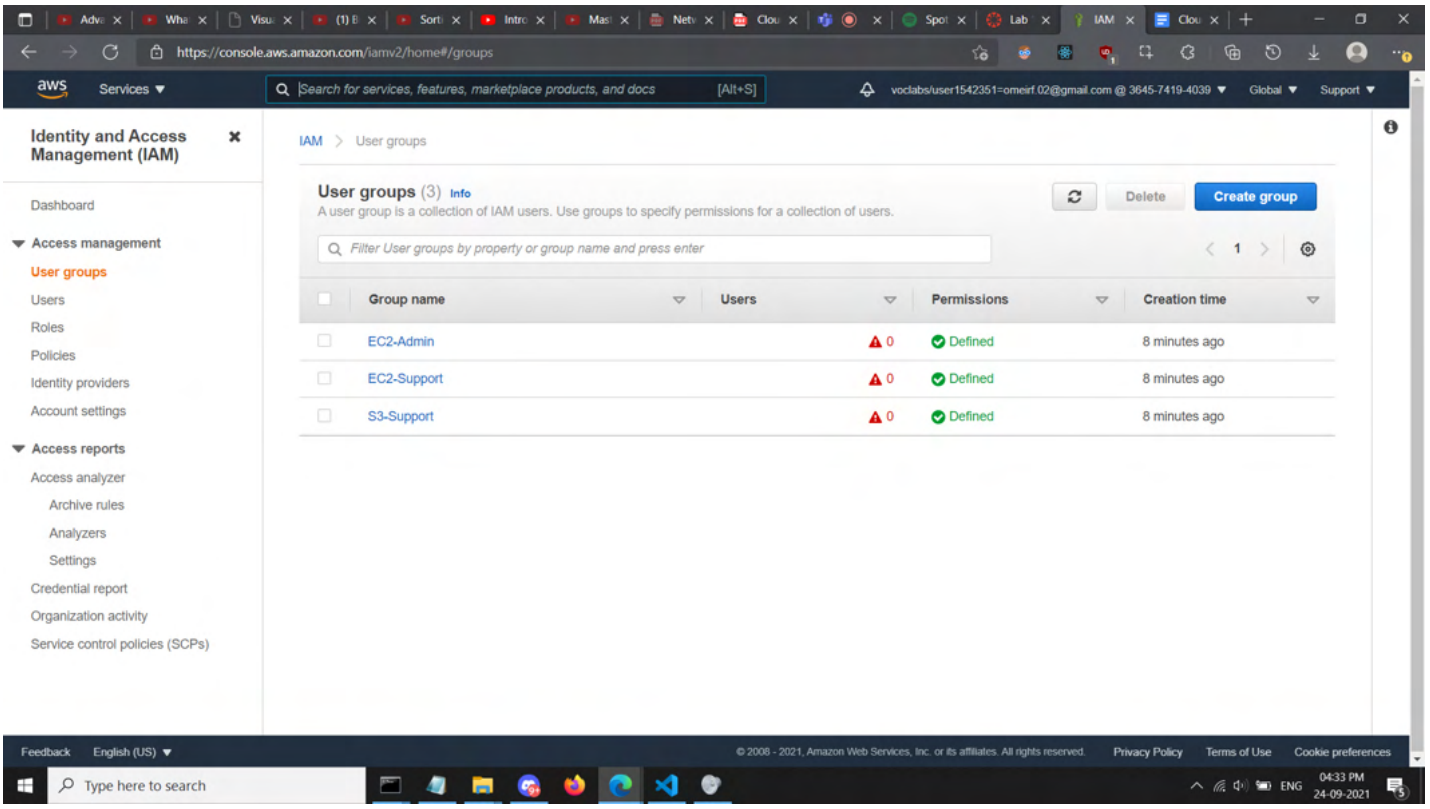
Access keys

Use access keys to make programmatic calls to AWS from the AWS CLI, Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time.

For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation.
If you lose or forget your secret key, you cannot retrieve it. Instead, create a new access key and make the old key inactive. Learn more

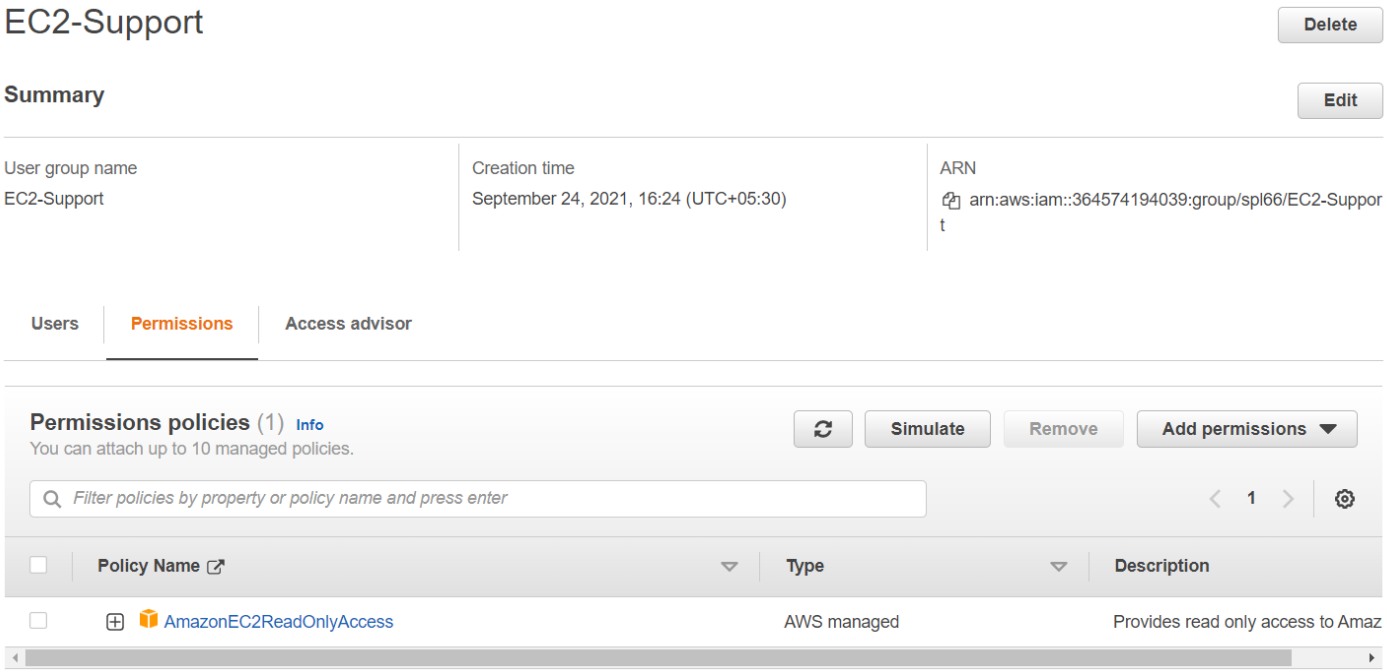
Create access key

11. In the navigation pane on the left, click **Groups**.
    The following groups have already been created for you:
    - EC2-Admin
    - EC2-Support
    - S3-Support

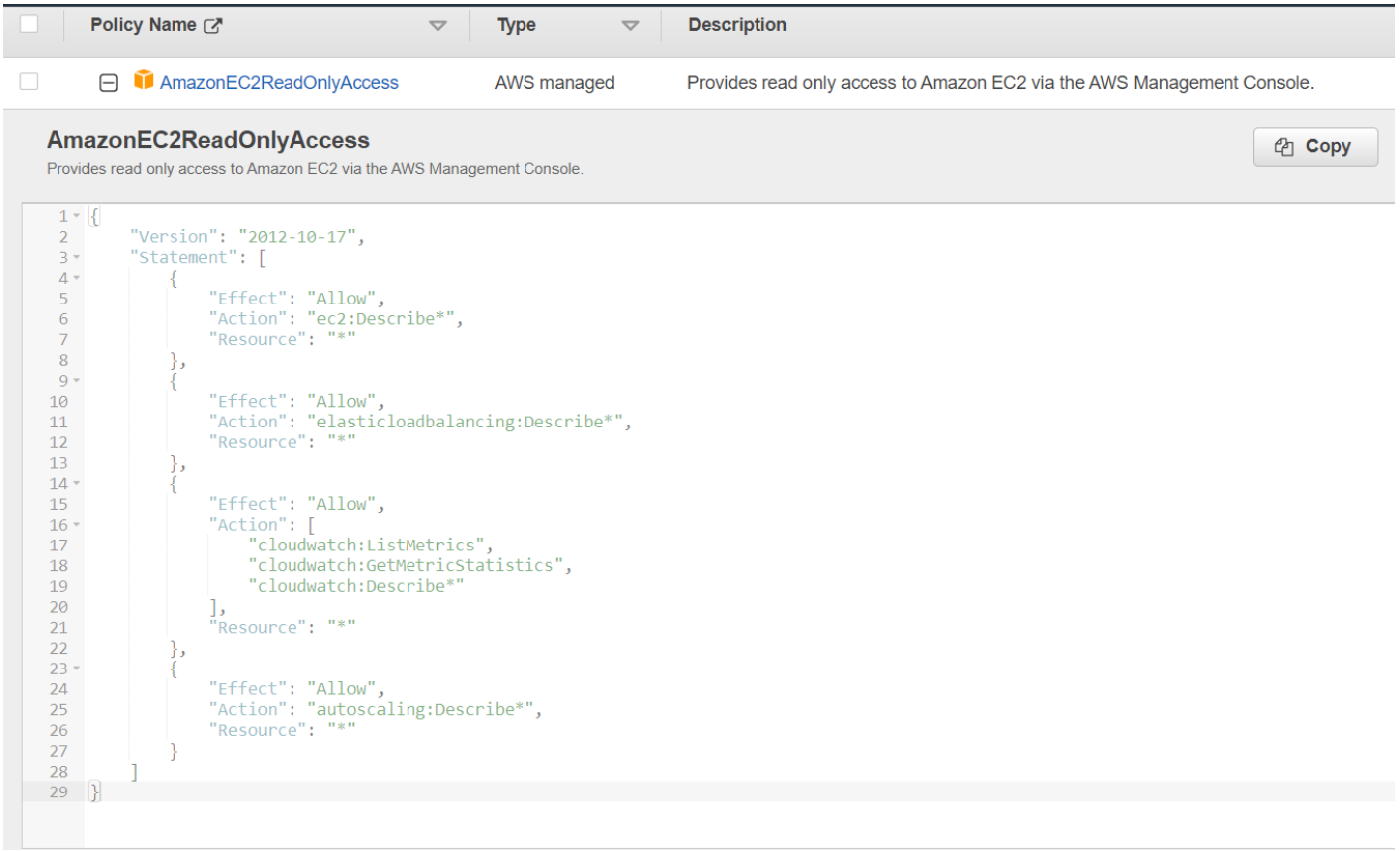


12. Click the **EC2-Support** group.
    This will bring you to the summary page for the **EC2-Support** group.
13. Click the **Permissions** tab.
    This group has a Managed Policy associated with it, called **AmazonEC2ReadOnlyAccess**.

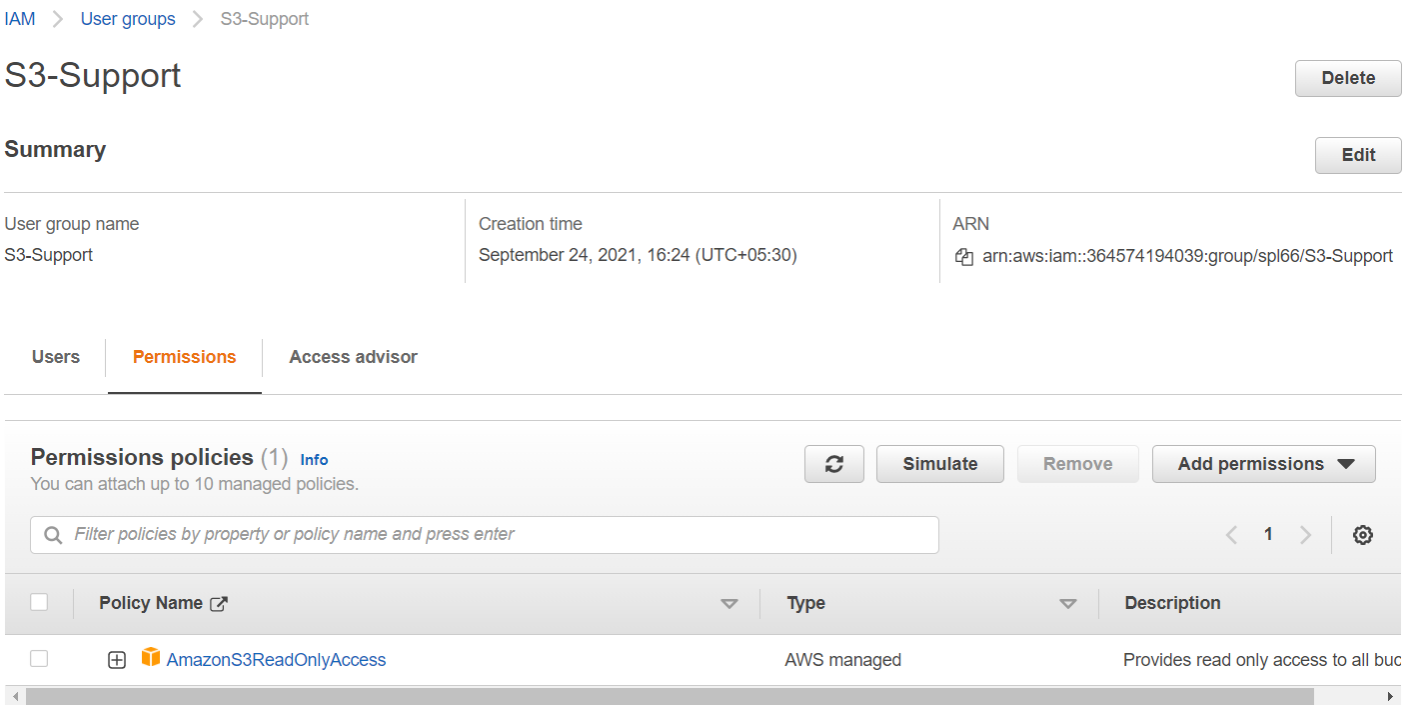
Managed Policies are pre-built policies (built either by AWS or by your administrators) that can be attached to IAM Users and Groups. When the policy is updated, the changes to the policy are immediately apply against all Users and Groups that are attached to the policy.



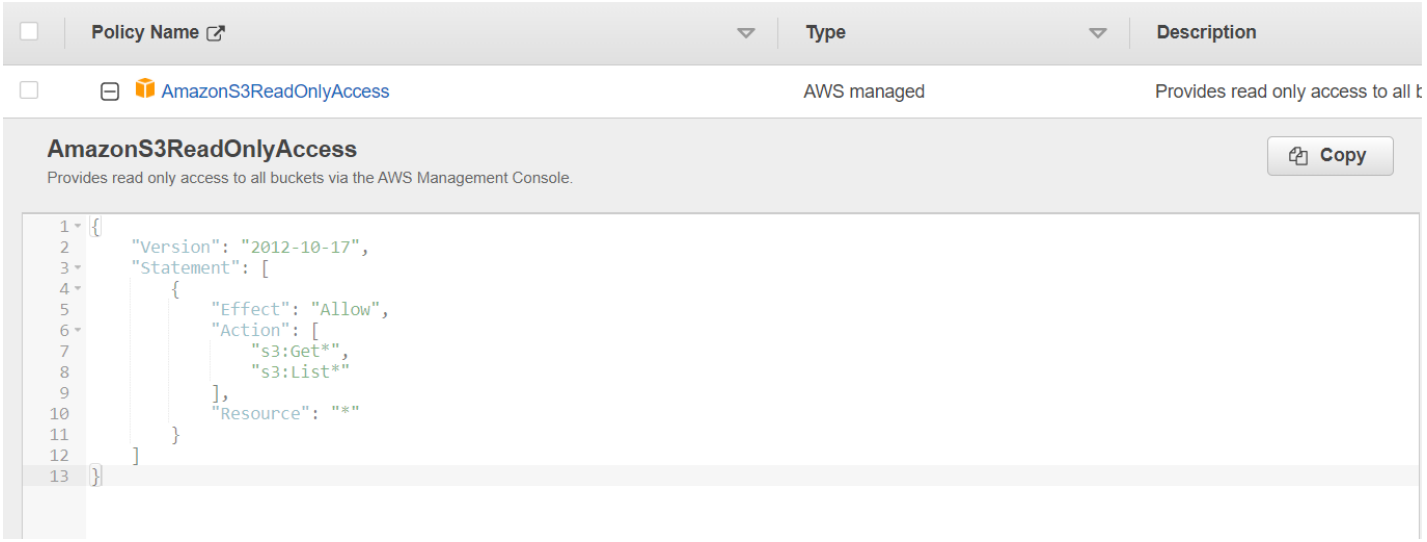
14. Under **Actions**, click the **Show Policy** link.
    A policy defines what actions are allowed or denied for specific AWS resources. This policy is granting permission to List and Describe information about EC2, Elastic Load Balancing, CloudWatch and Auto Scaling. This ability to view resources, but not modify them, is ideal for assigning to a Support role.
    The basic structure of the statements in an IAM Policy is:
    - **Effect** says whether to *Allow* or *Deny* the permissions.
    - **Action** specifies the API calls that can be made against an AWS Service (eg *cloudwatch:ListMetrics*).
    - **Resource** defines the scope of entities covered by the policy rule (eg a specific Amazon S3 bucket or Amazon EC2 instance, or * which means *any resource*).



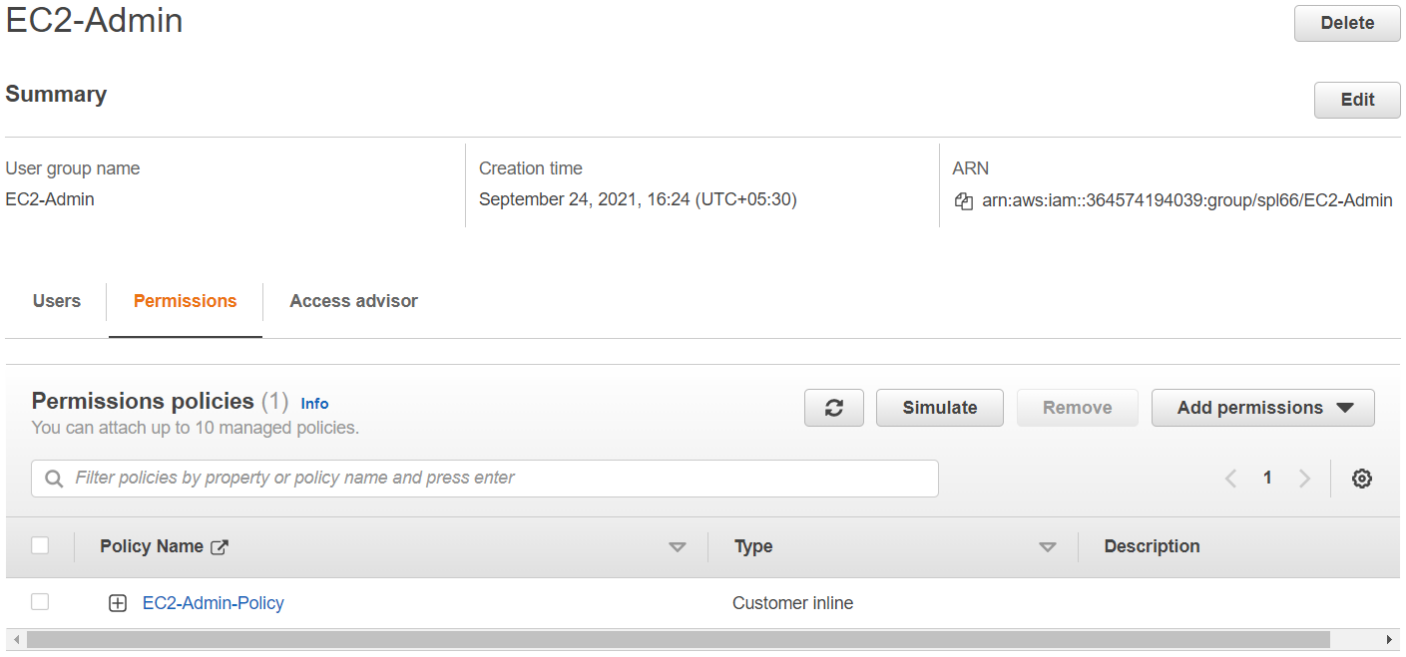
15. Close the **Show Policy** window.
16. In the navigation pane on the left, click **Groups**.
17. Click the **S3-Support** group.
    The S3-Support group has the **AmazonS3ReadOnlyAccess** policy attached.



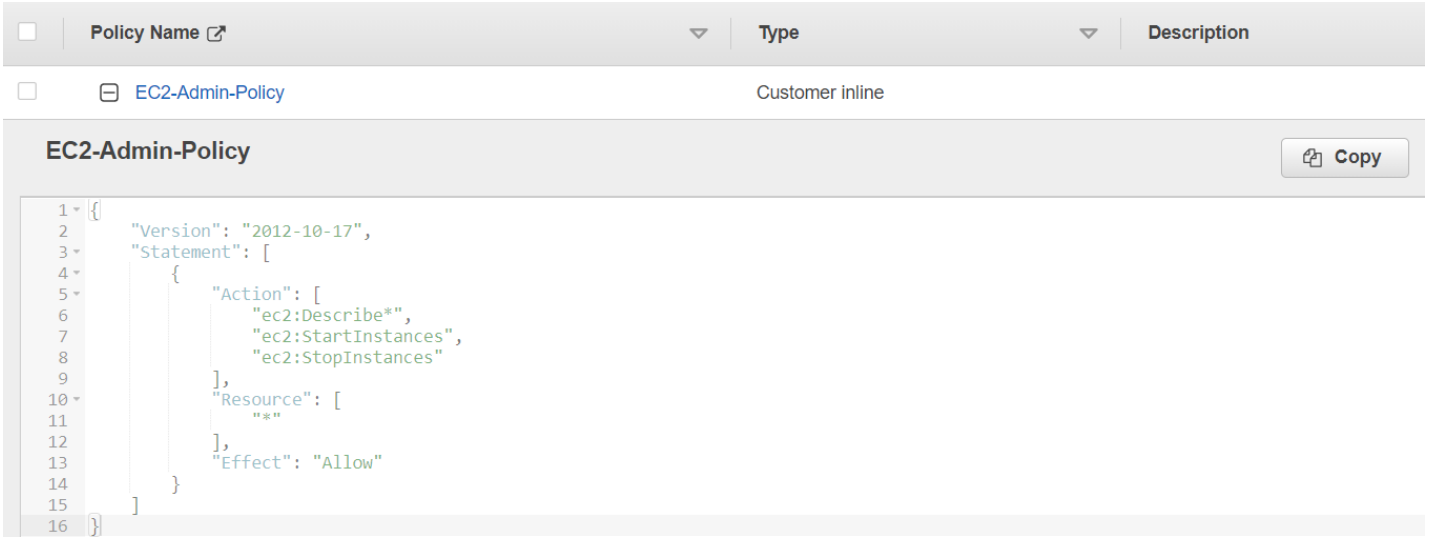
18. Below the **Actions** menu, click the **Show Policy** link.
    This policy has permissions to Get and List resources in Amazon S3.



19. Close the **Show Policy** window.
20. In the navigation pane on the left, click **Groups**.
21. Click the **EC2-Admin** group.
    This Group is slightly different from the other two. Instead of a *Managed Policy*, it has an **Inline Policy**, which is a policy assigned to just one User or Group. Inline Policies are typically used to apply permissions for one-off situations.



22. Under **Actions**, click **Show Policy** to view the policy.
    This policy grants permission to view (Describe) information about Amazon EC2 and also the ability to Start and Stop instances.



23. At the bottom of the screen, click **Cancel** to close the policy.

Business Scenario

For the remainder of this lab, you will work with these Users and Groups to enable permissions supporting the following business scenario:

Your company is growing its use of Amazon Web Services, and is using many Amazon EC2 instances and a great deal of Amazon S3 storage. You wish to give access to new staff depending upon their job function:

User	In Group	Permissions
user-1	S3-Support	Read-Only access to Amazon S3
user-2	EC2-Support	Read-Only access to Amazon EC2
user-3	EC2-Admin	View, Start and Stop Amazon EC2 instances

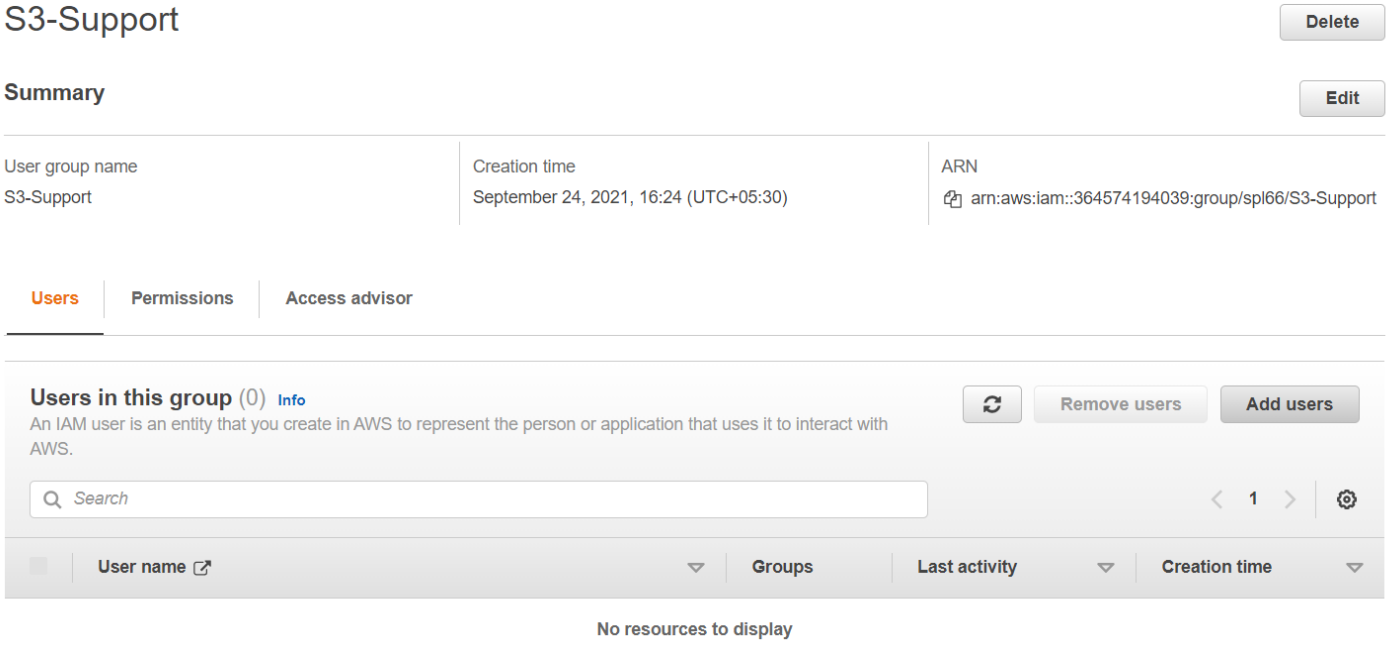
Task 2: Add Users to Groups

You have recently hired **user-1** into a role where they will provide support for Amazon S3. You will add them to the **S3-Support** group so that they inherit the necessary permissions via the attached *AmazonS3ReadOnlyAccess* policy.

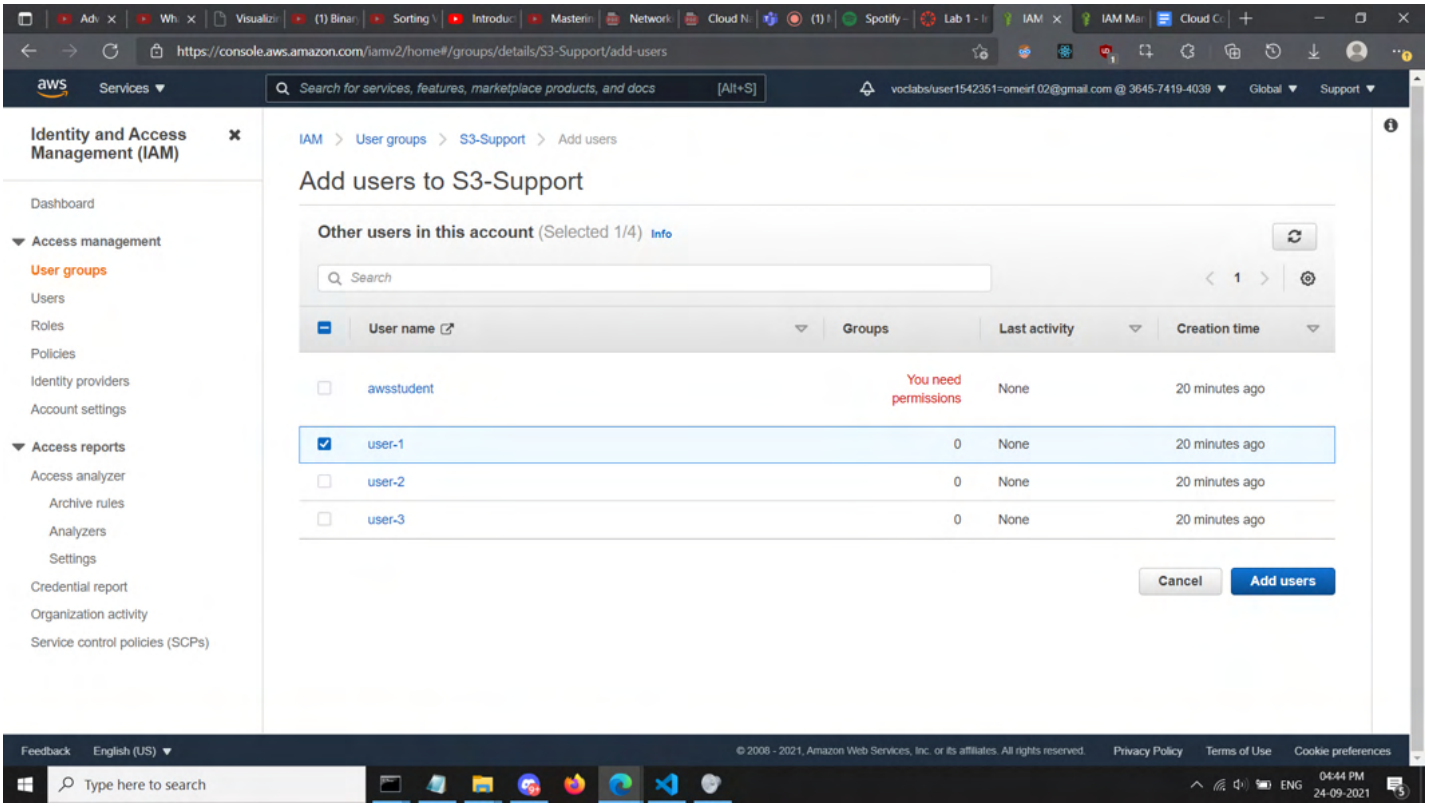
You can ignore any "not authorized" errors that appear during this task. They are caused by your lab account having limited permissions and will not impact your ability to complete the lab.

Add user-1 to the S3-Support Group

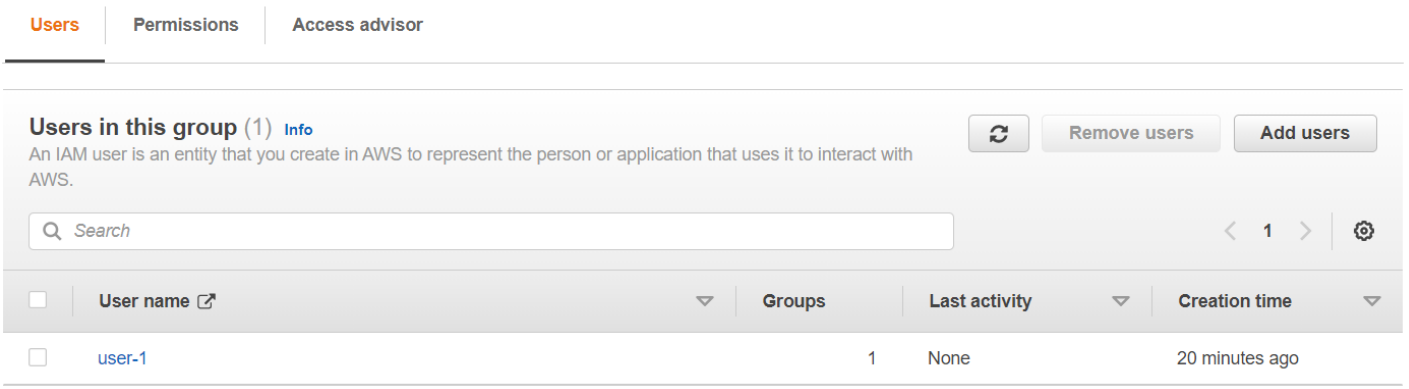
24. In the left navigation pane, click **Groups**.
25. Click the **S3-Support** group.
26. Click the **Users** tab.



27. In the **Users** tab, click **Add Users to Group**.
28. In the **Add Users to Group** window, configure the following:
    - Select **user-1**.
    - At the bottom of the screen, click **Add Users**.



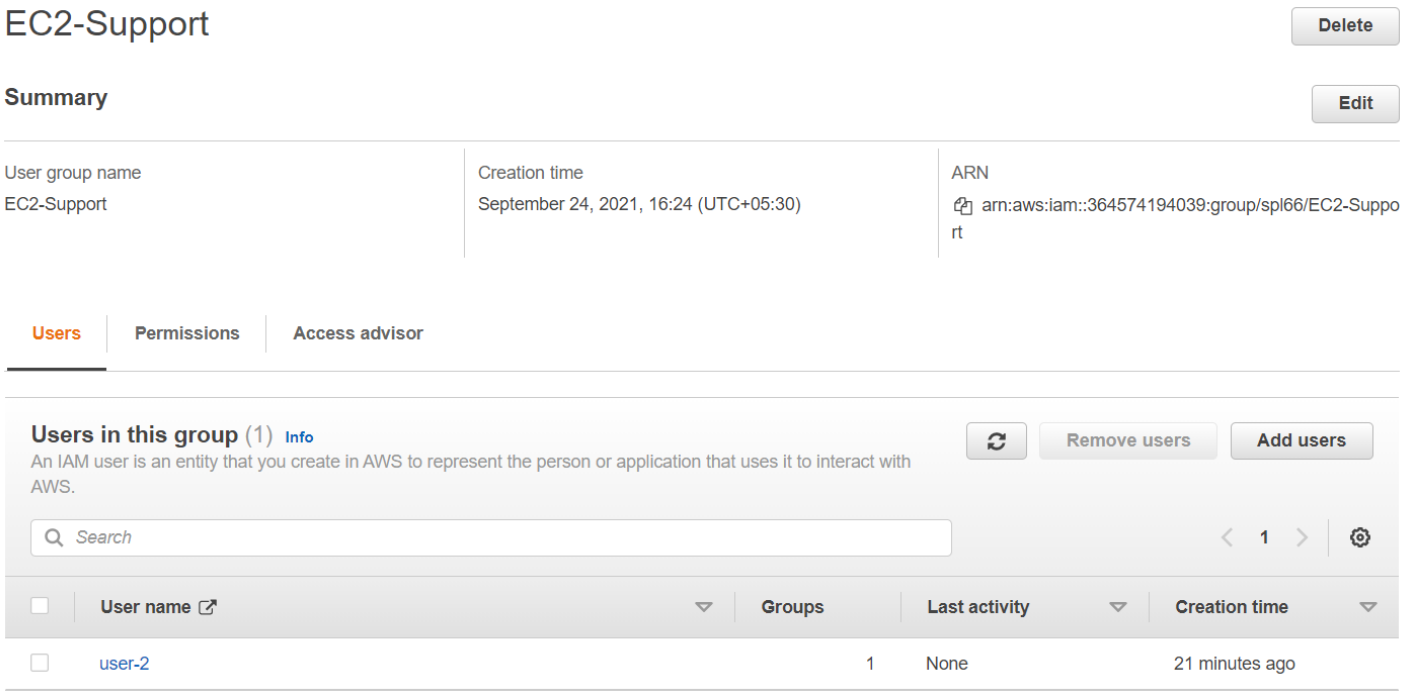
29. In the **Users** tab you will see that user-1 has been added to the group.



Add user-2 to the EC2-Support Group

You have hired **user-2** into a role where they will provide support for Amazon EC2.

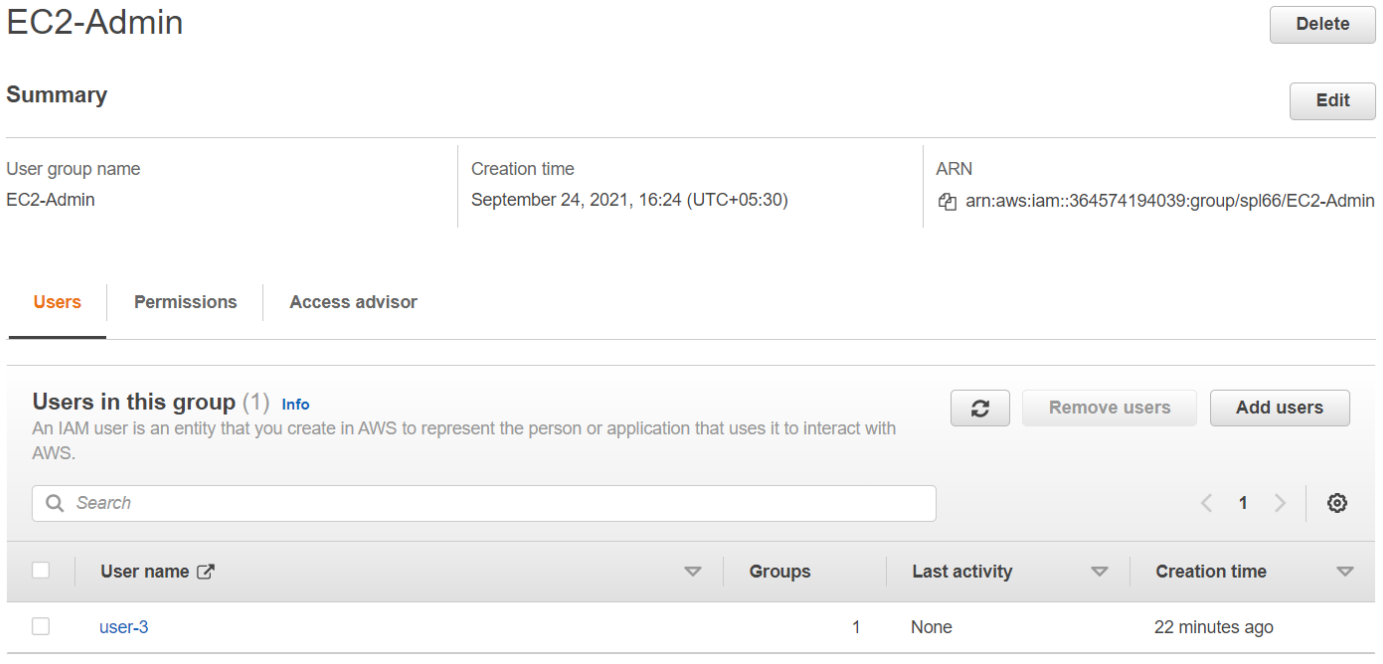
29. Using similar steps to the ones above, add **user-2** to the **EC2-Support** group.
    user-2 should now be part of the **EC2-Support** group.



Add user-3 to the EC2-Admin Group

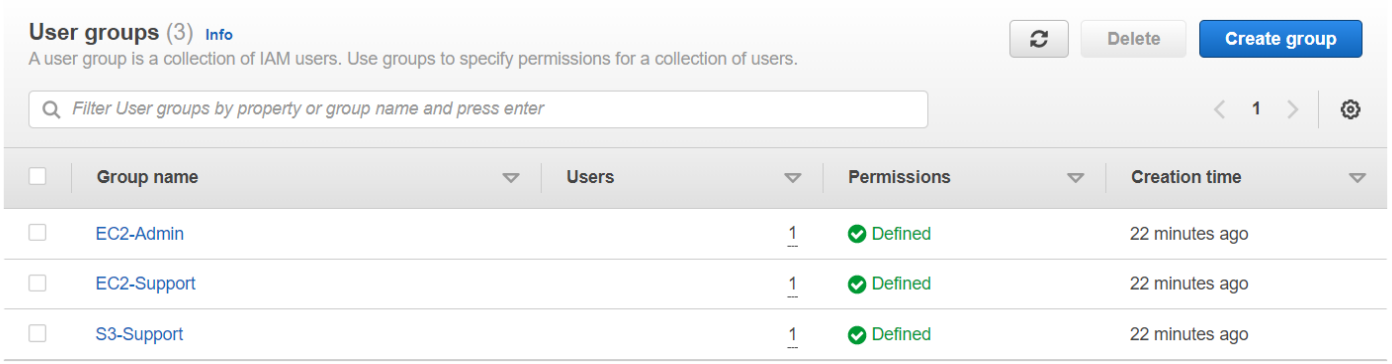
You have hired **user-3** as your Amazon EC2 administrator, who will manage your EC2 instances.

30. Using similar steps to the ones above, add **user-3** to the **EC2-Admin** group.
    user-3 should now be part of the **EC2-Admin** group.



31. In the navigation pane on the left, click **Groups**.
    Each Group should have a **1** in the Users column for the number of Users in each Group.

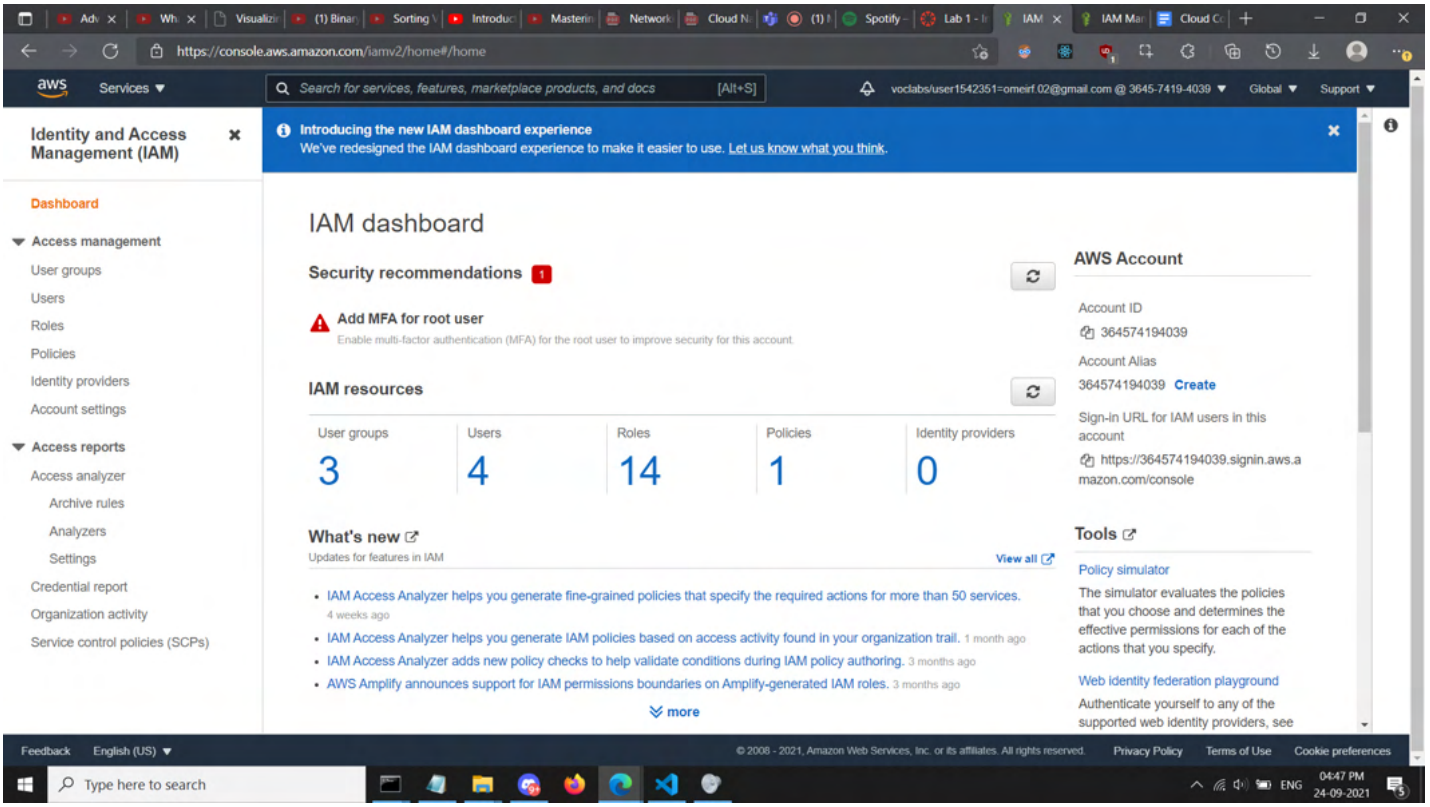
If you do not have a **1** beside each group, revisit the above instructions to ensure that each user is assigned to a Group, as shown in the table in the Business Scenario section.



Task 3: Sign-In and Test Users

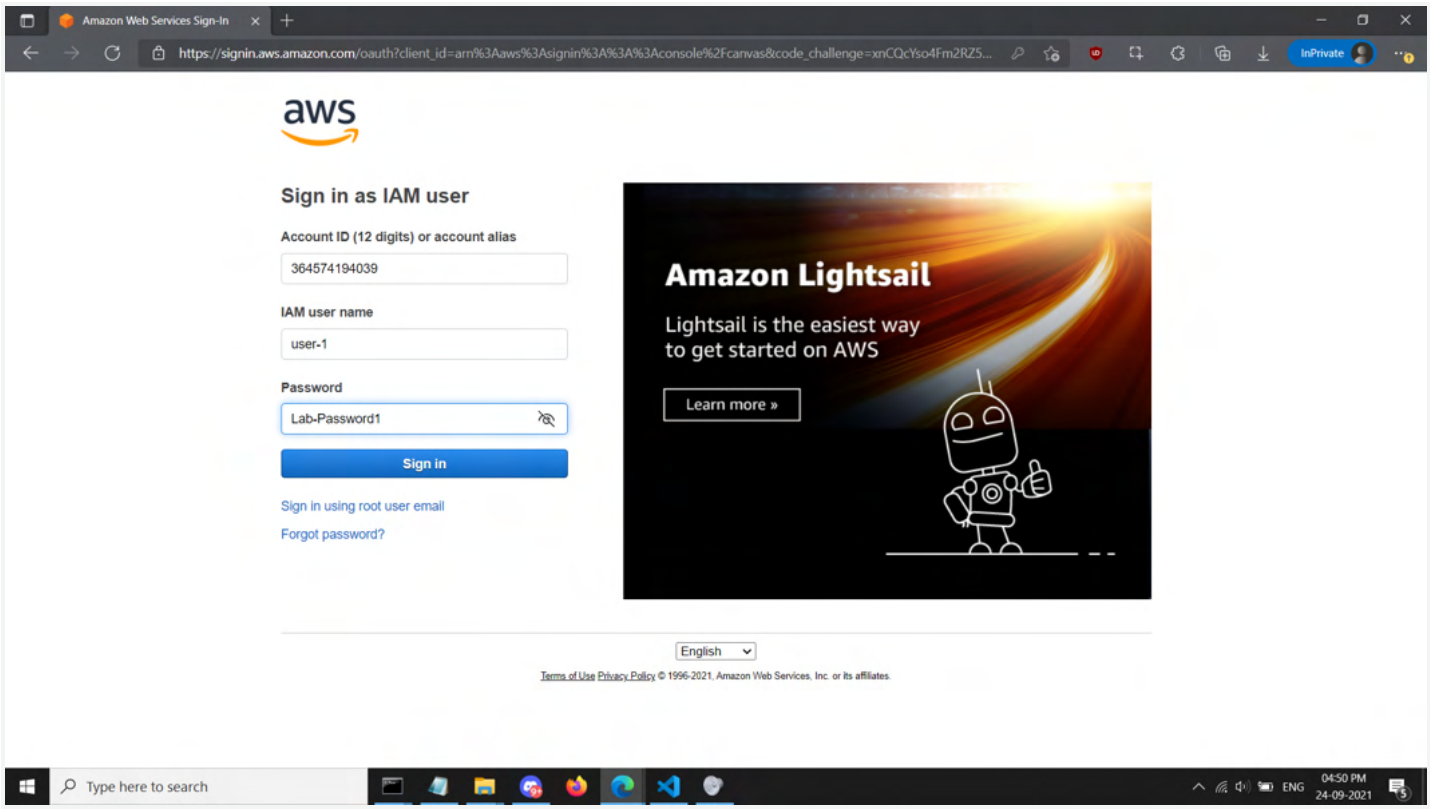
In this task, you will test the permissions of each IAM User.

32. In the navigation pane on the left, click **Dashboard**.
    An **IAM users sign-in link** is displayed It will look similar to:
    https://123456789012.signin.aws.amazon.com/console
    This link can be used to sign-in to the AWS Account you are currently using.

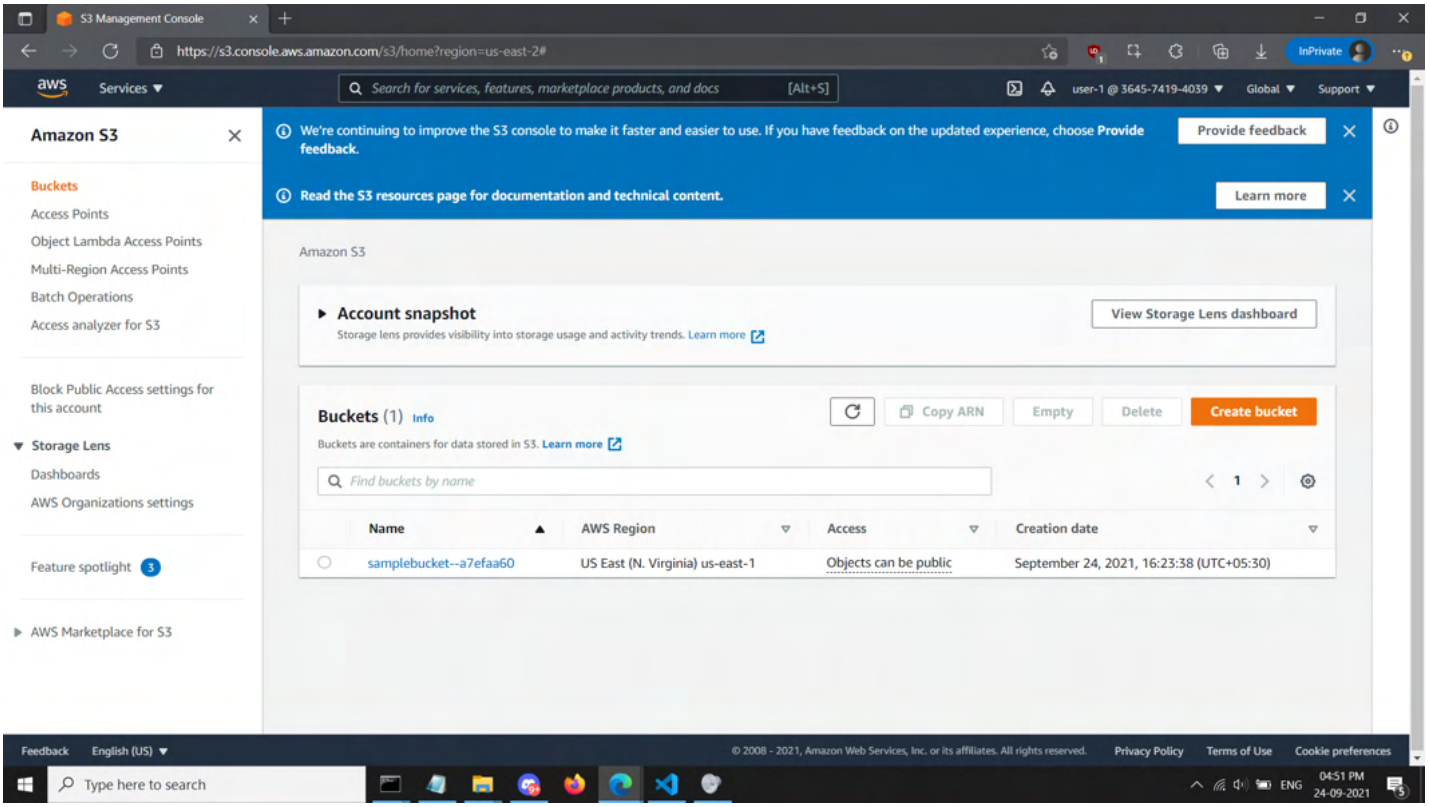


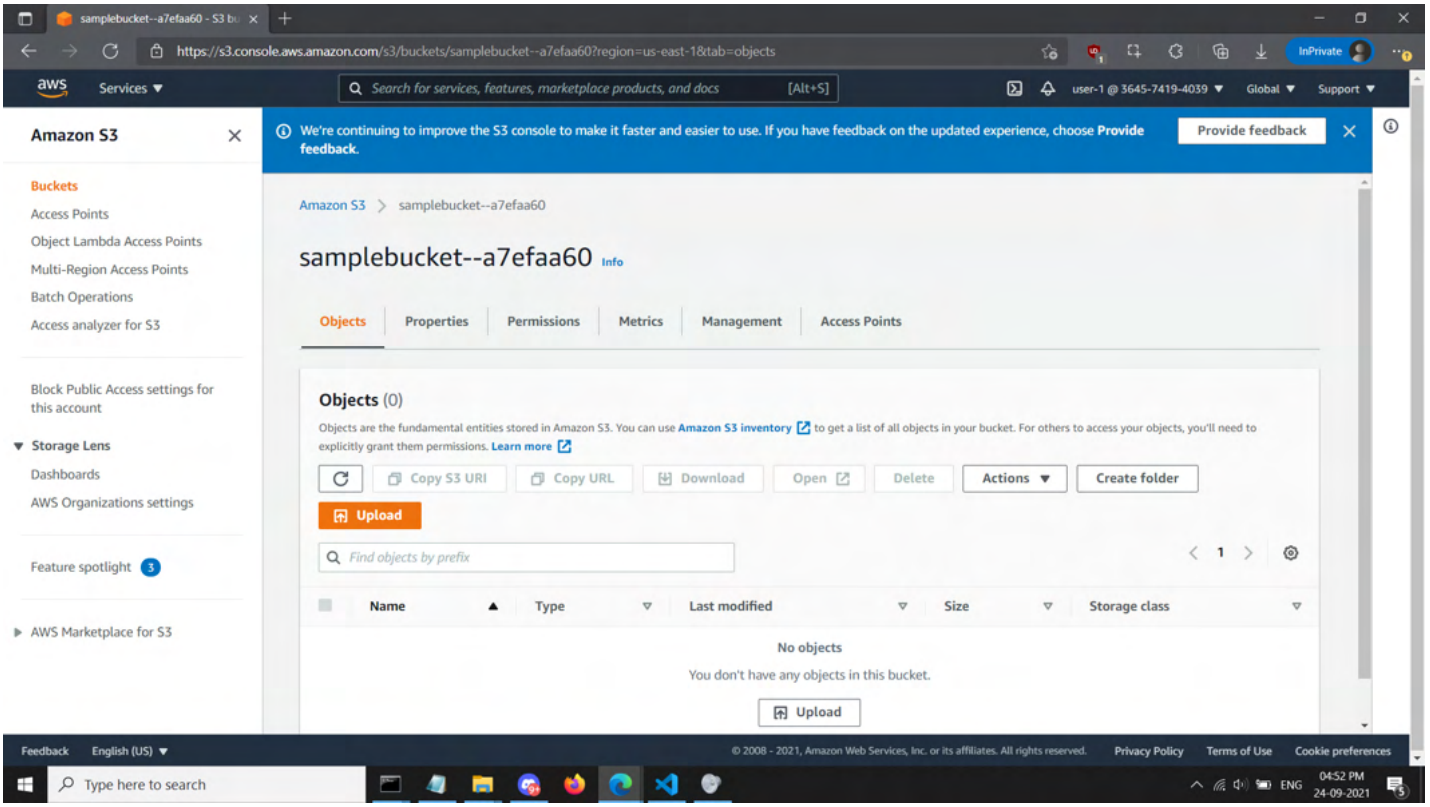
33. Copy the **IAM users sign-in link** to a text editor.

34. Open a private window.
    Mozilla Firefox
    - Click the menu bars at the top-right of the screen
    - Select **New Private Window**
35. **Google Chrome**
    - Click the ellipsis at the top-right of the screen
    - Click **New incognito window**
36. **Microsoft Edge**
    - Click the ellipsis at the top-right of the screen
    - Click **New InPrivate window**
37. **Microsoft Internet Explorer**
    - Click the **Tools** menu option
    - Click **InPrivate Browsing**
38. Paste the **IAM users sign-in** link into your private window and press **Enter**.
    You will now sign-in as **user-1**, who has been hired as your Amazon S3 storage support staff.
39. Sign-in with:
    - **IAM user name:** user-1
    - **Password:** Lab-Password1



40. In the **Services** menu, click **S3**.
41. Click the name of one of your buckets and browse the contents.
    Since your user is part of the **S3-Support** Group in IAM, they have permission to view a list of Amazon S3 buckets and their contents.
    Now, test whether they have access to Amazon EC2.

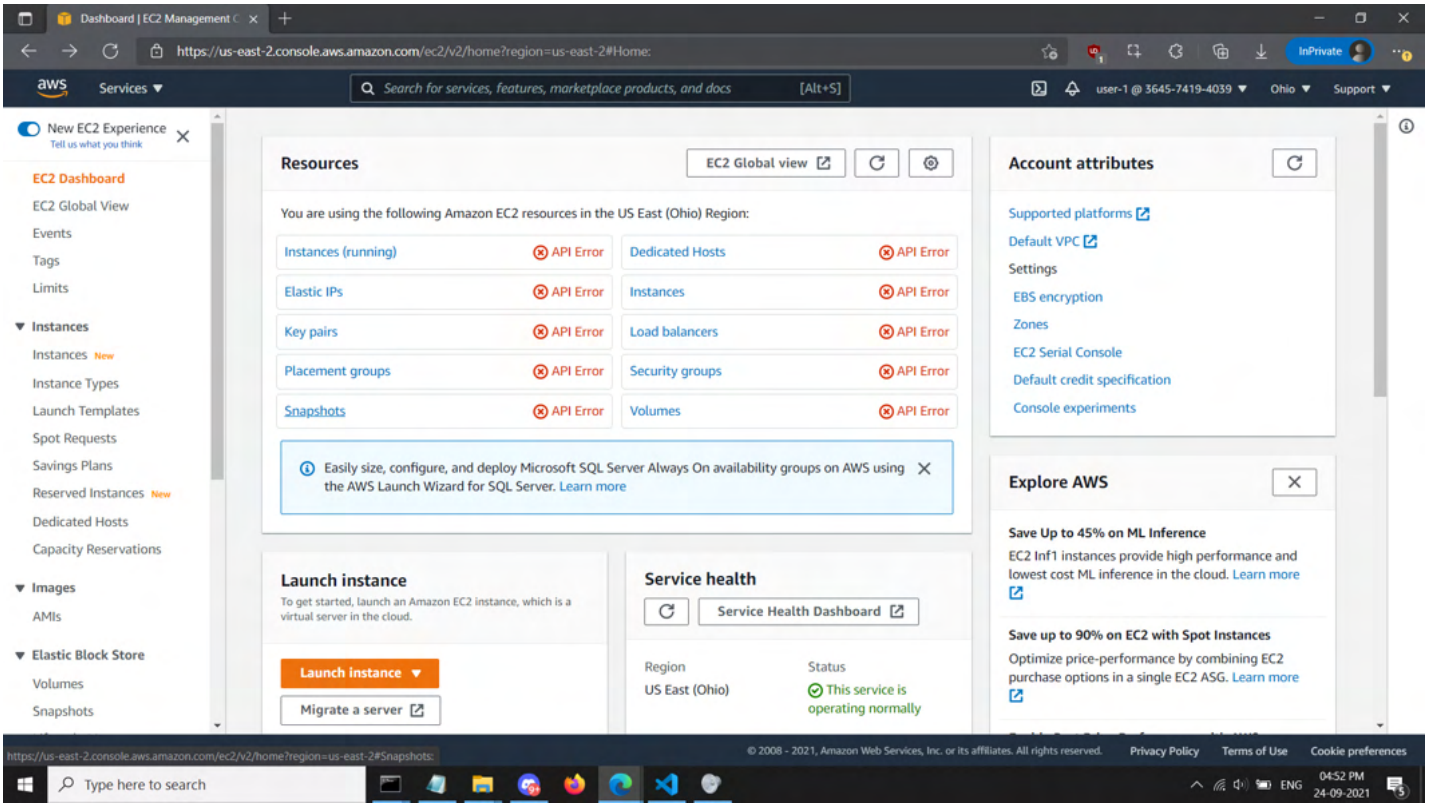


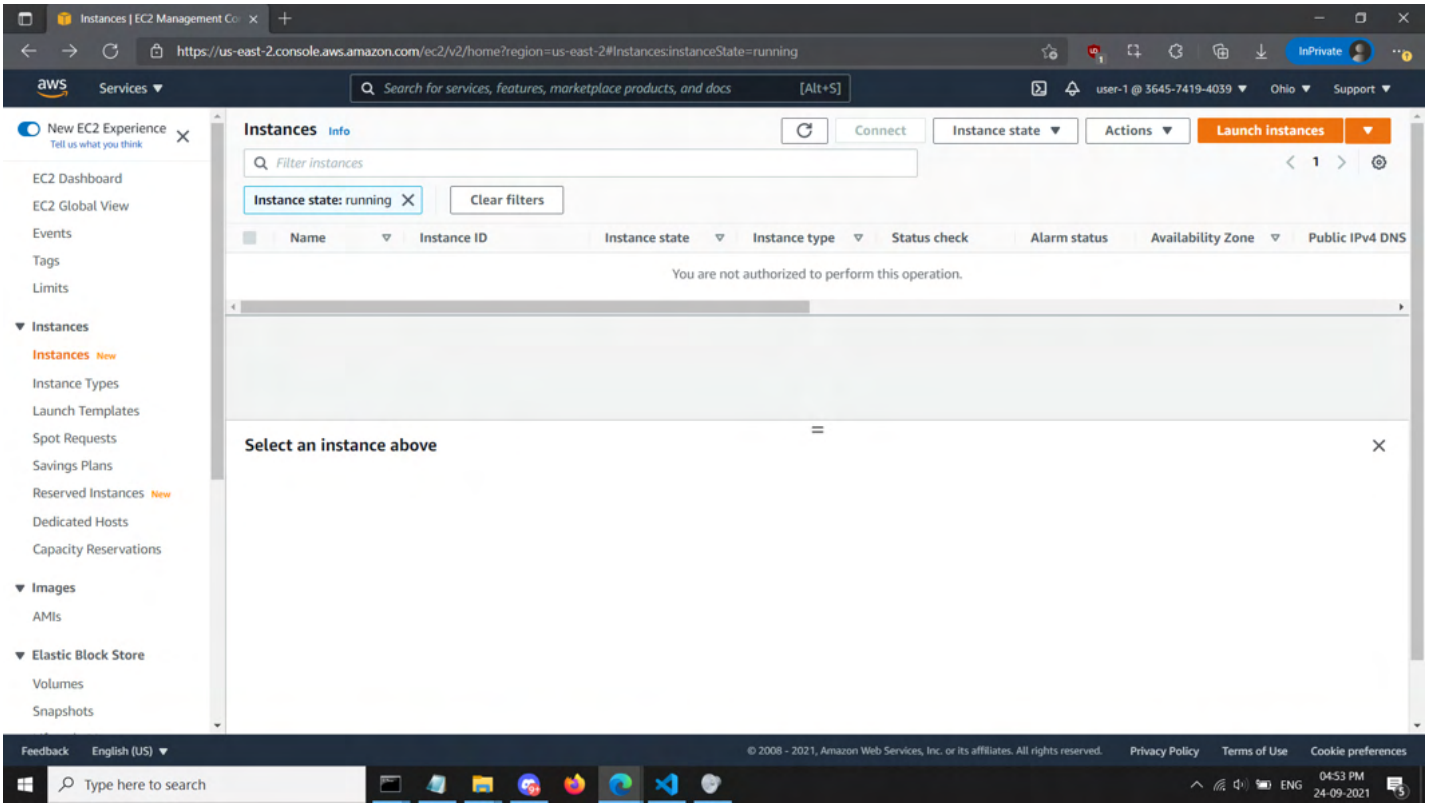


42. In the **Services** menu, click **EC2**.
43. In the left navigation pane, click **Instances**.

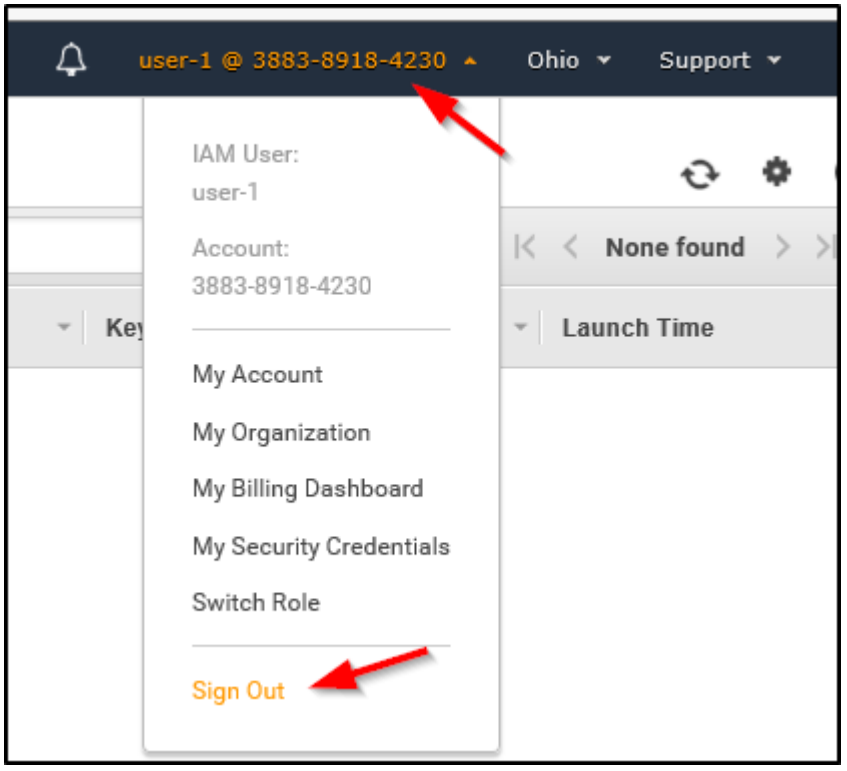
    You cannot see any instances! Instead, it says *You do not have any instances in this region*. This is because your user has not been assigned any permissions to use Amazon EC2.

    You will now sign-in as **user-2**, who has been hired as your Amazon EC2 support person.

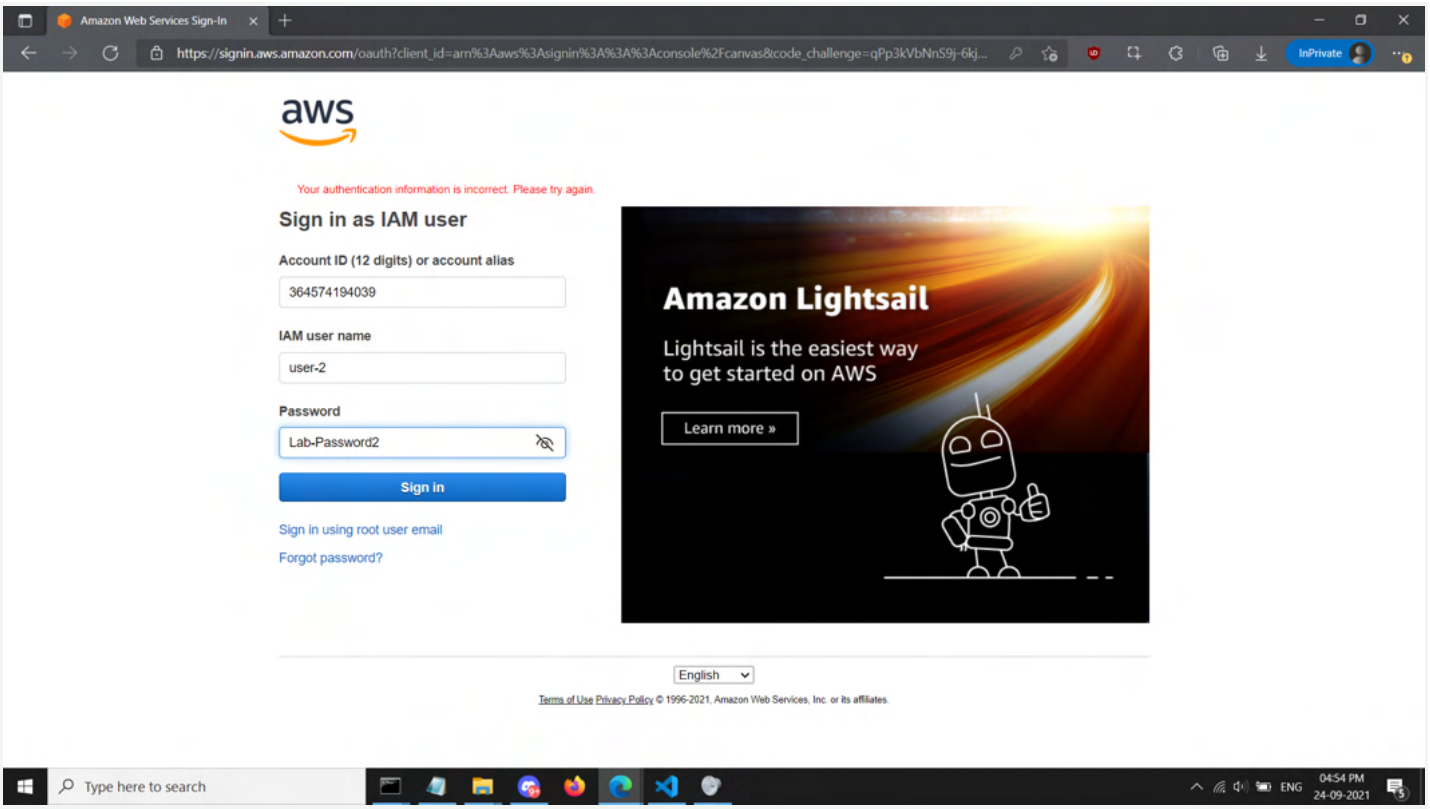




44. Sign user-1 out of the **AWS Management Console** by configuring the following:
    - At the top of the screen, click **user-1**
    - Click **Sign Out**

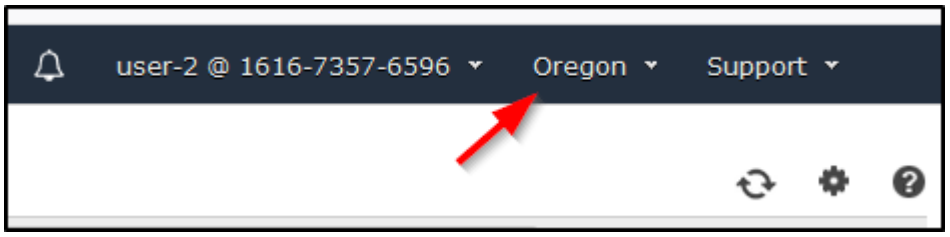
45. 

46. Paste the **IAM users sign-in** link into your private window and press **Enter**.
    This links should be in your text editor.
47. Sign-in with:
    - **IAM user name:** user-2
    - **Password:** Lab-Password2

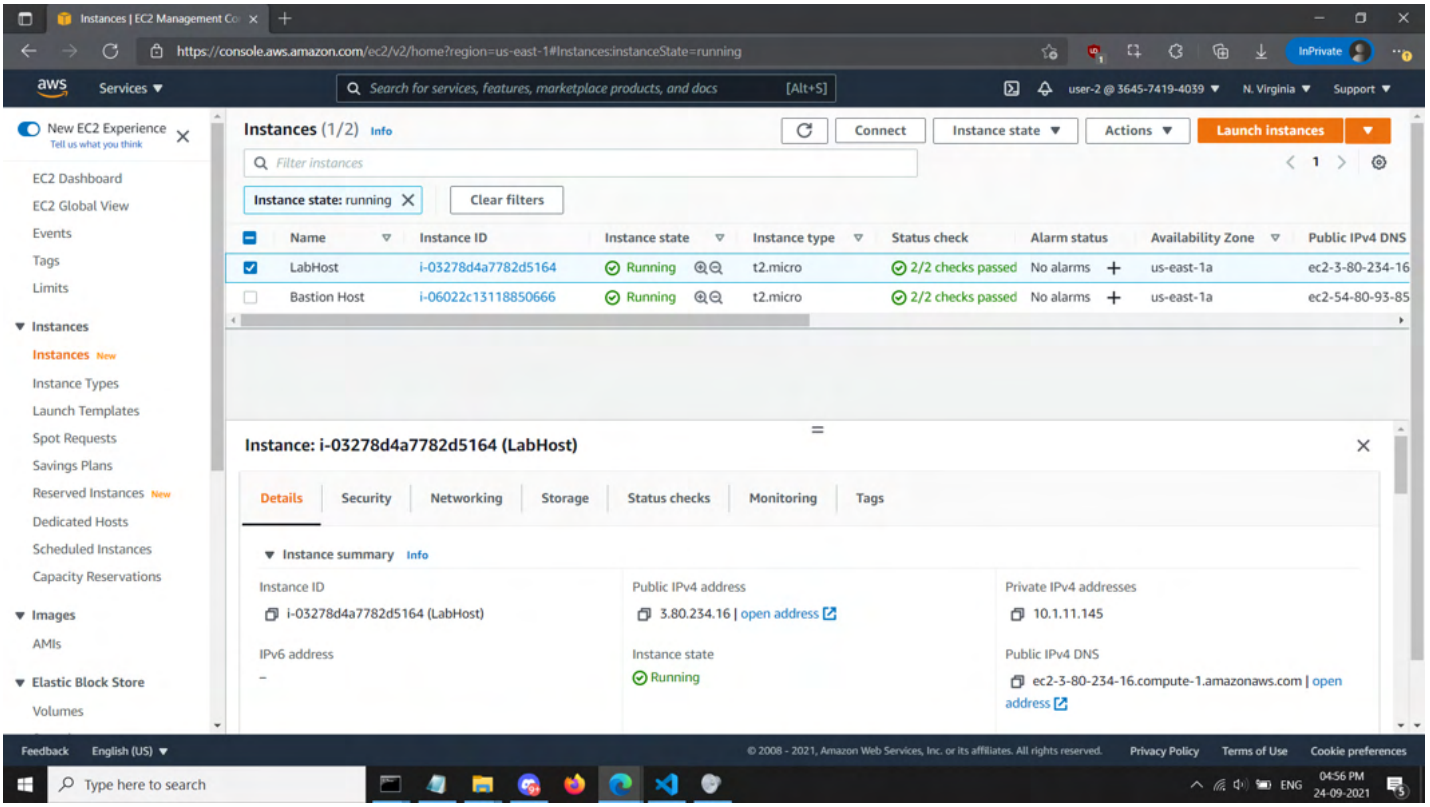


48. In the **Services** menu, click **EC2**.
49. In the navigation pane on the left, click **Instances**.
    You are now able to see an Amazon EC2 instance because you have Read Only

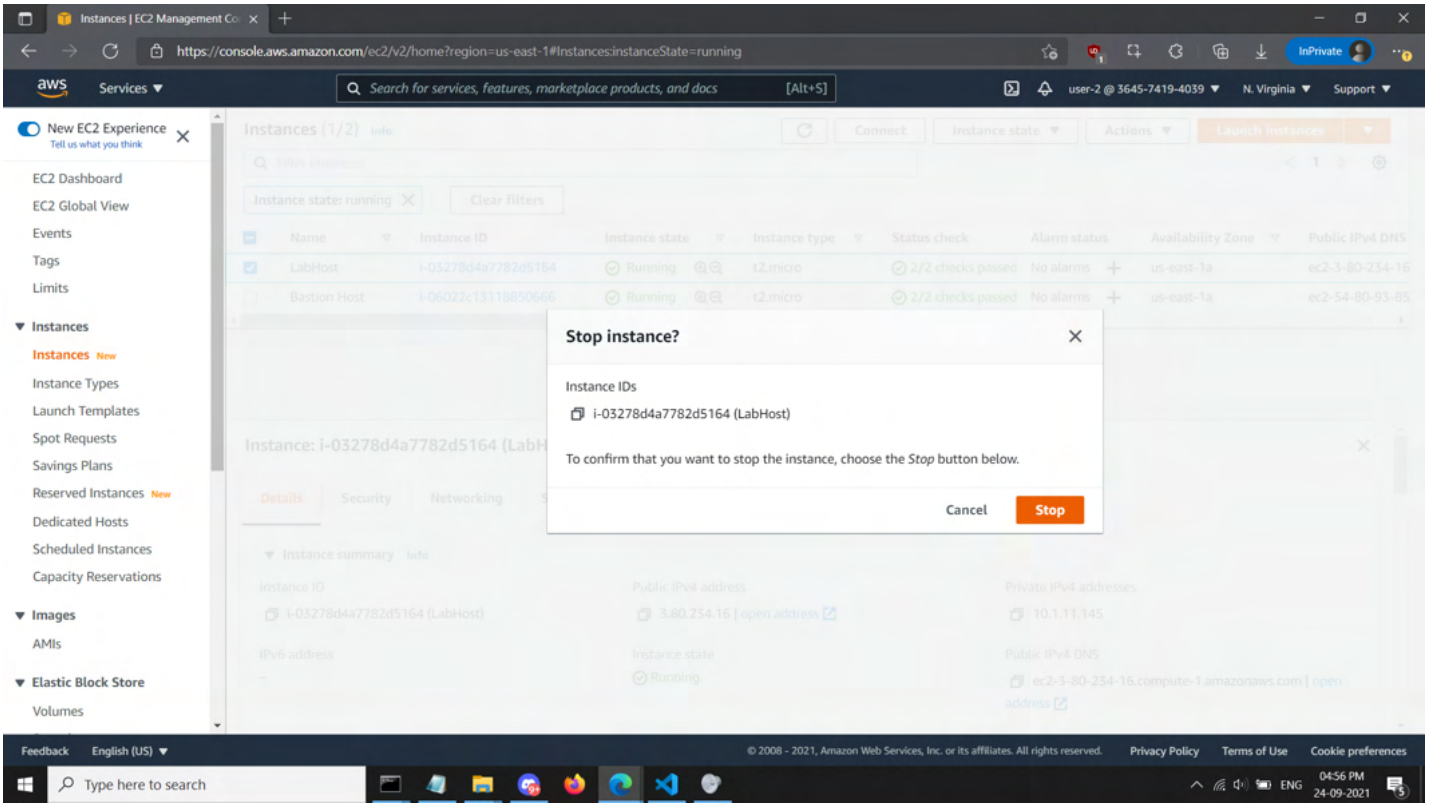
permissions. However, you will not be able to make any changes to Amazon EC2 resources. If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (e.g., **N. Virginia**).



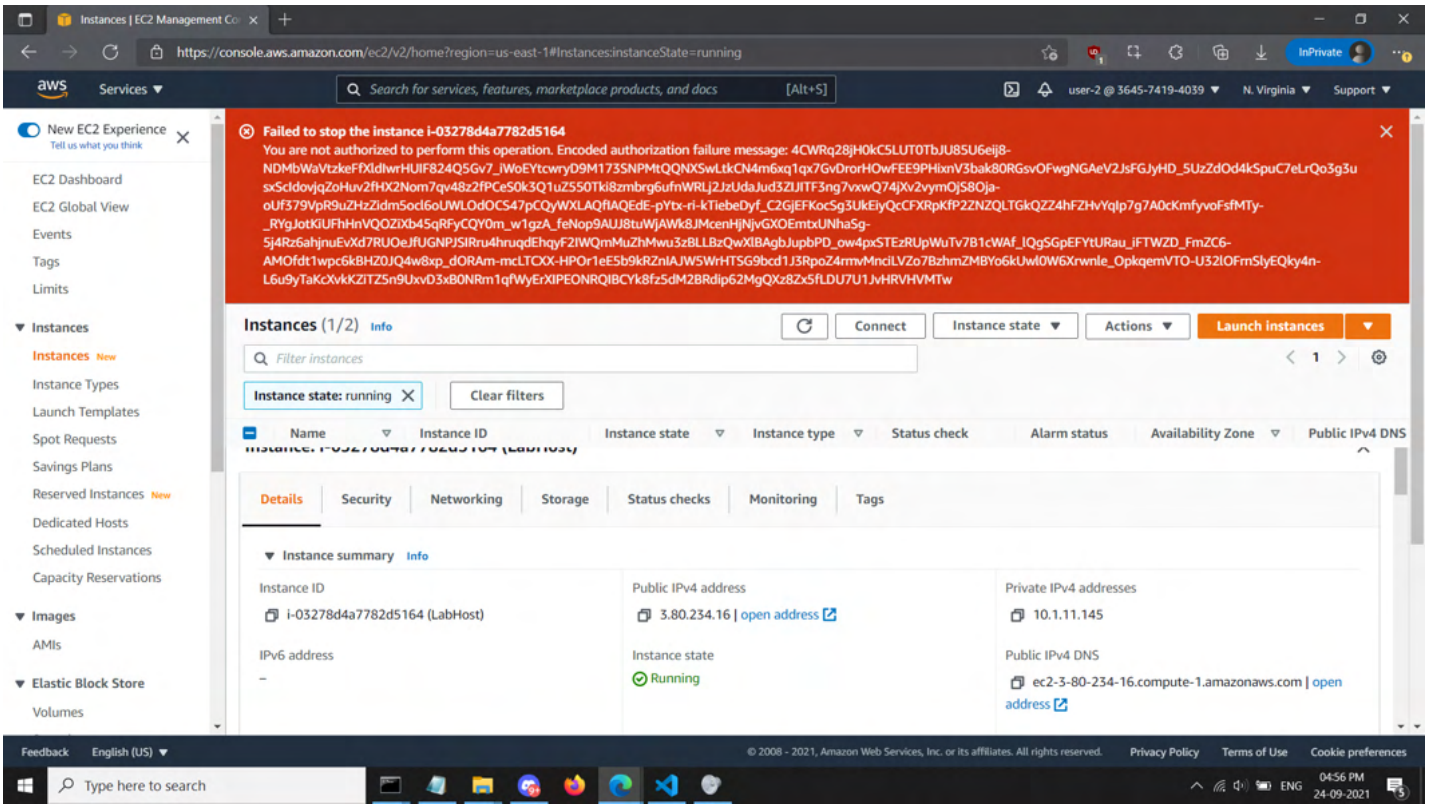
Your EC2 instance should be selected . If it is not selected, select the instance named *LabHost* .



50. In the **Instance state** menu, choose **Stop instance**.

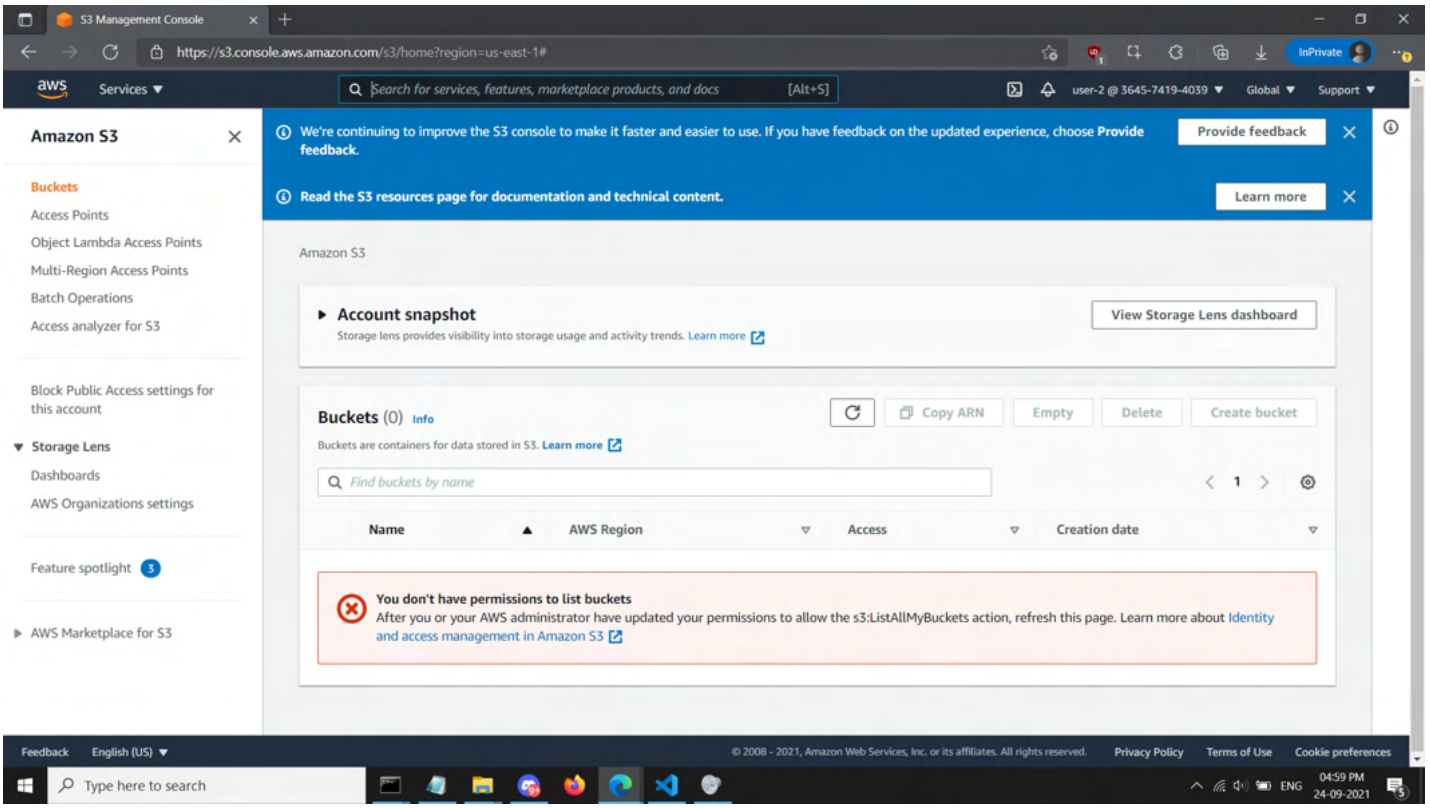


51. In the **Stop Instance** window, click **Stop**.
    You will receive an error stating *You are not authorized to perform this operation*. This demonstrates that the policy only allows you to view information, without making changes.



52. At the **Stop instances** window, click **Cancel**.
    Next, check if user-2 can access Amazon S3.

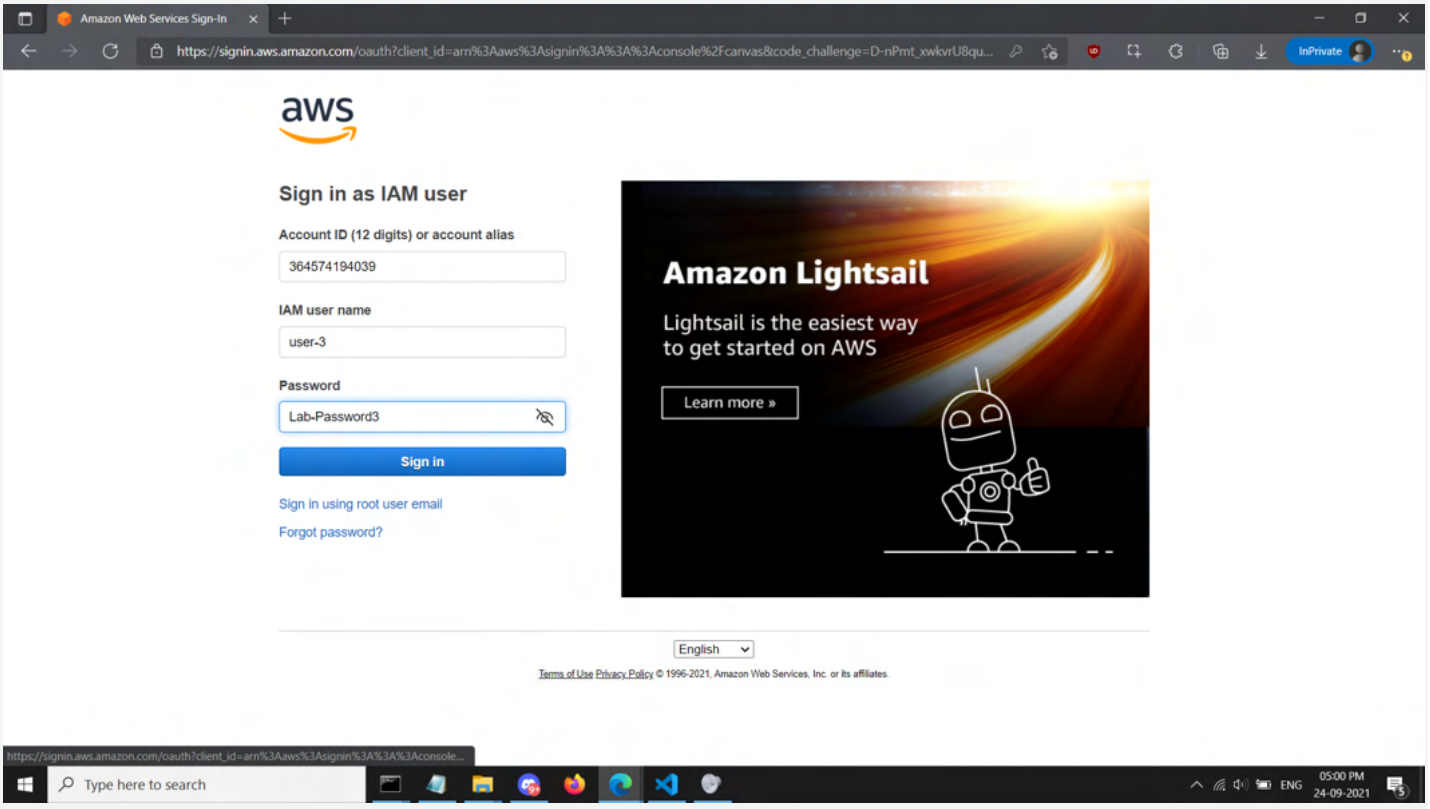
53. In the **Services**, click **S3**.
    You will receive an **Error Access Denied** because user-2 does not have permission to use Amazon S3.



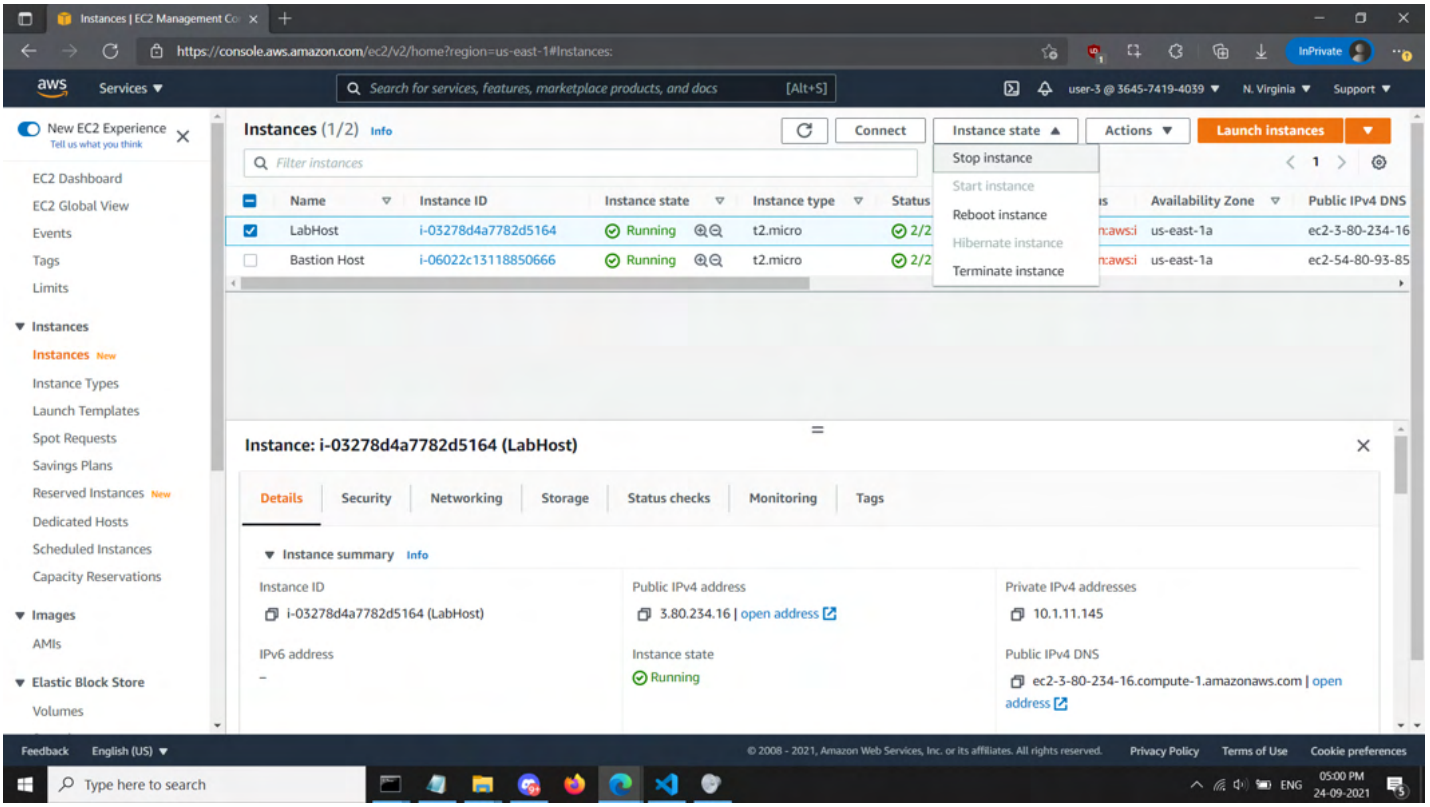
You will now sign-in as **user-3**, who has been hired as your Amazon EC2 administrator.

54. Sign user-2 out of the **AWS Management Console** by configuring the following:
    - At the top of the screen, click **user-2**
    - Click **Sign Out**

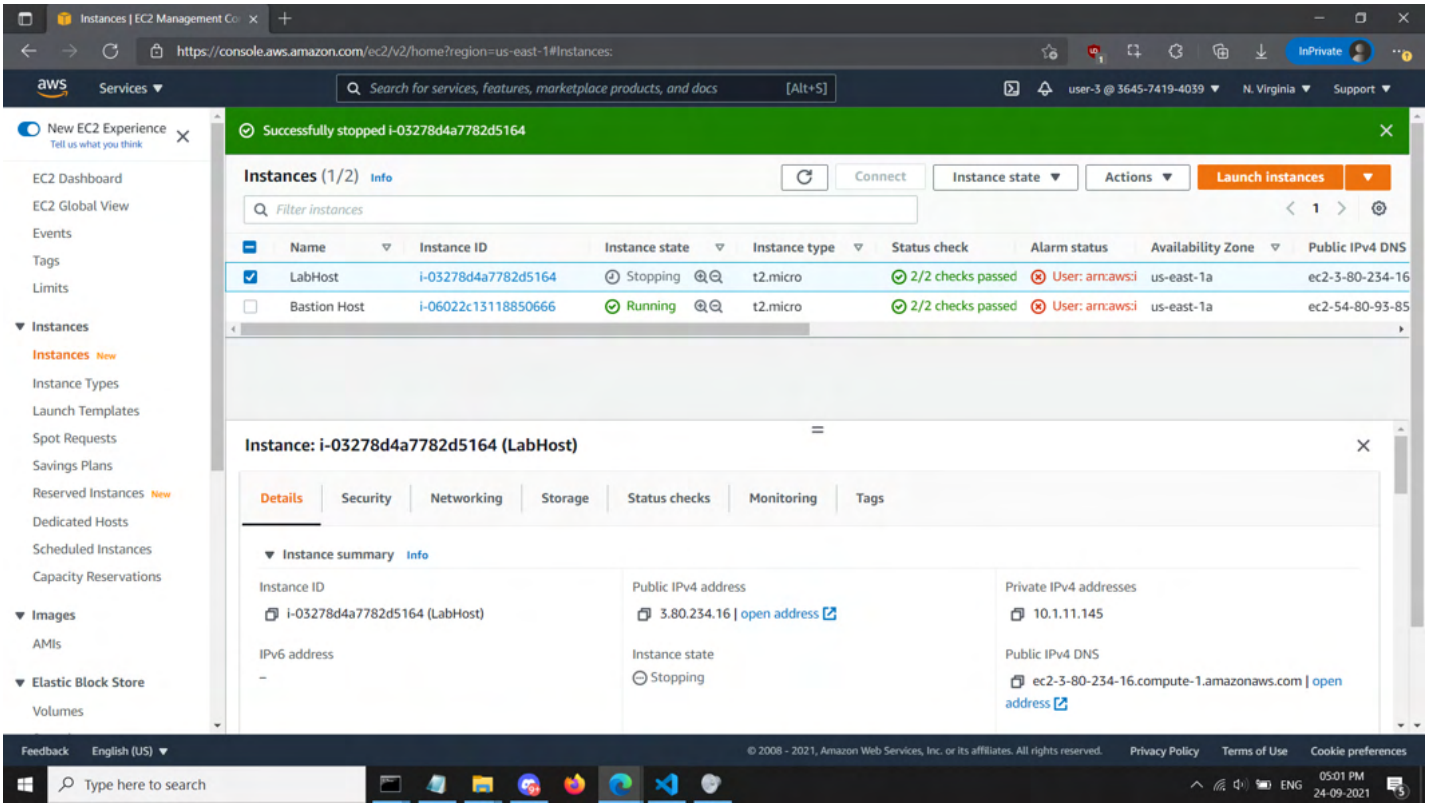
55. Paste the **IAM users sign-in** link into your private window and press **Enter**.
56. Paste the sign-in link into your web browser address bar again. If it is not in your clipboard, retrieve it from the text editor where you stored it earlier.
57. Sign-in with:
    - **IAM user name:** user-3
    - **Password:** Lab-Password3



58. In the **Services** menu, click **EC2**.
59. In the navigation pane on the left, click **Instances**.
    As an EC2 Administrator, you should now have permissions to Stop the Amazon EC2 instance.
    Your EC2 instance should be selected . If it is not, please select the instance named *LabHost* .
    If you cannot see an Amazon EC2 instance, then your Region may be incorrect. In the top-right of the screen, pull-down the Region menu and select the region that you noted at the start of the lab (eg **Oregon**).
60. In the **Instance state** menu, choose **Stop instance**.



61. In the **Stop instance** window, click **Stop**.
    The instance will enter the *stopping* state and will shutdown.



62. Close your private window.

CONCLUSION

Congratulations! You now have successfully:

- Explored pre-created IAM users and groups
- Inspected IAM policies as applied to the pre-created groups
- Followed a real-world scenario, adding users to groups with specific capabilities enabled
- Located and used the IAM sign-in URL
- Experimented with the effects of policies on service access

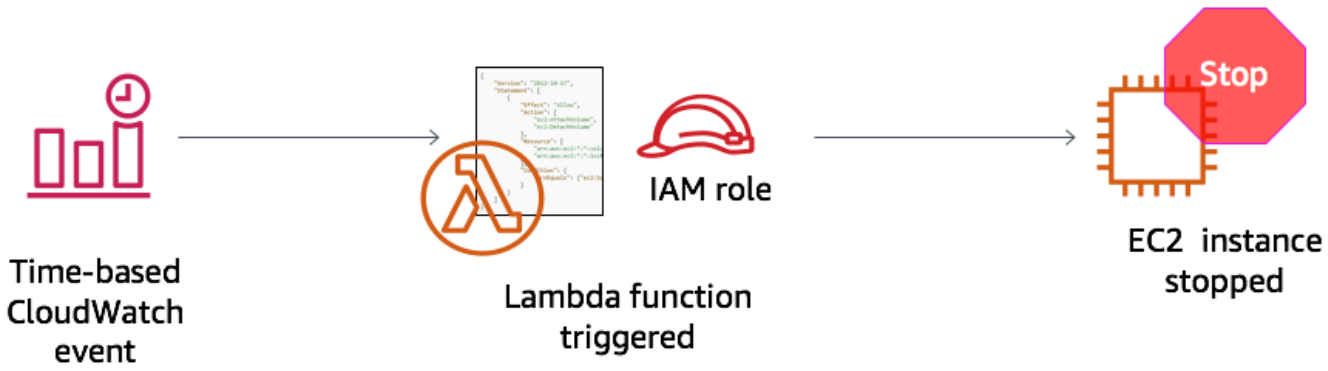
Activity 1

AIM

AWS Lambda

THEORY

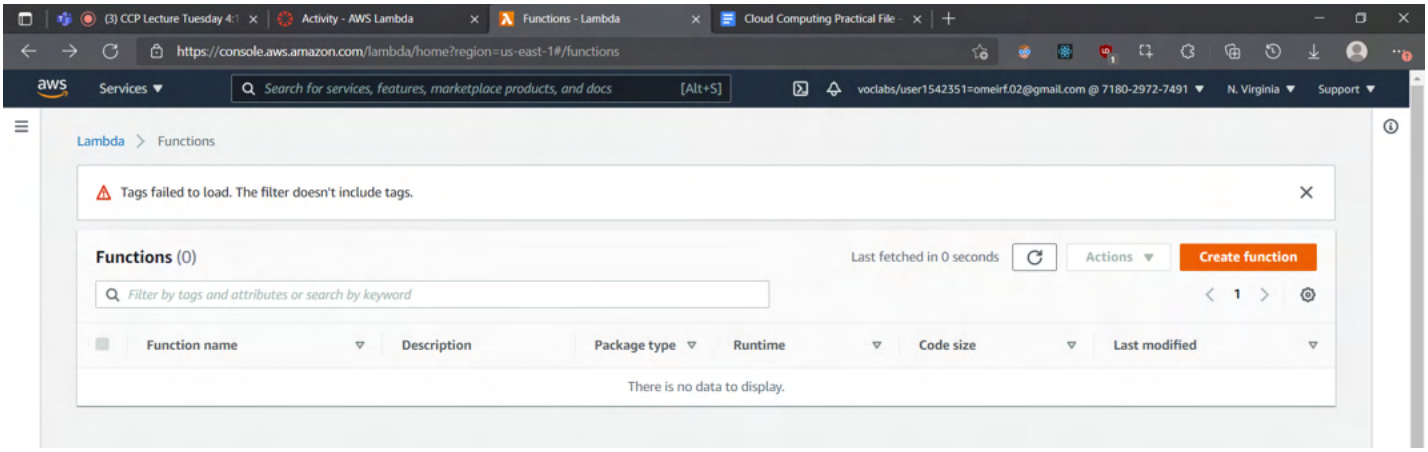
In this hands-on activity, you will create an AWS Lambda function. You will also create an Amazon CloudWatch event to trigger the function every minute. The function uses an AWS Identity and Access Management (IAM) role. This IAM role allows the function to stop an Amazon Elastic Compute Cloud (Amazon EC2) instance that is running in the Amazon Web Services (AWS) account.



PROCEDURE

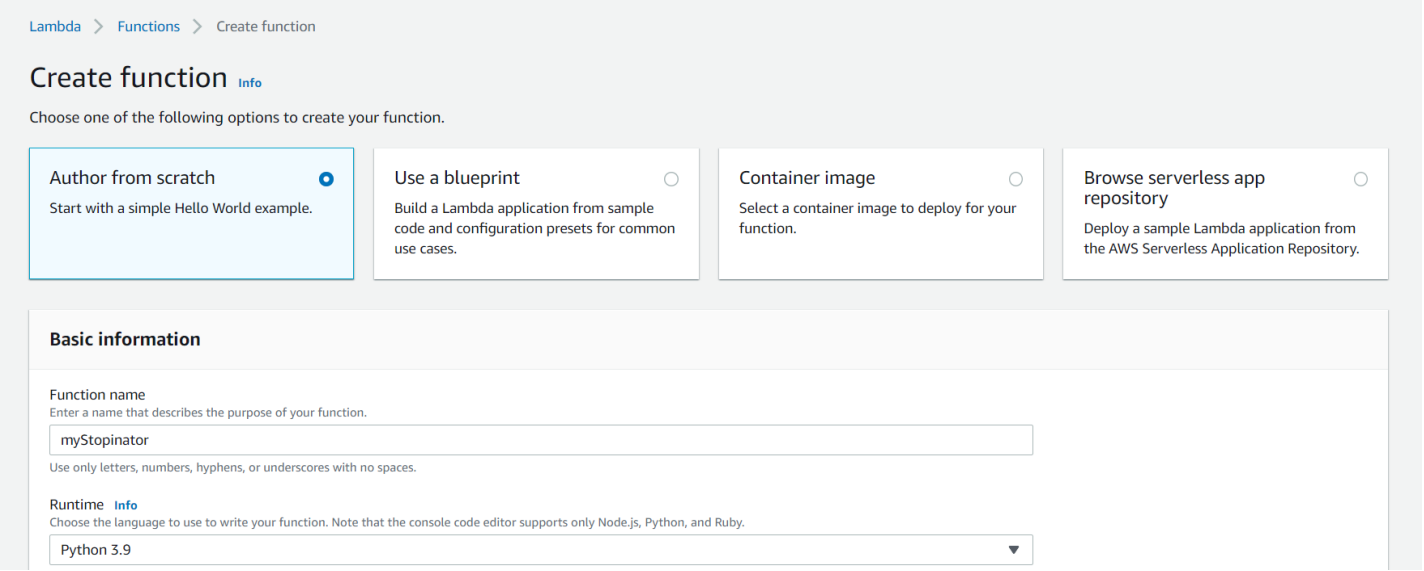
Task 1: Create a Lambda function

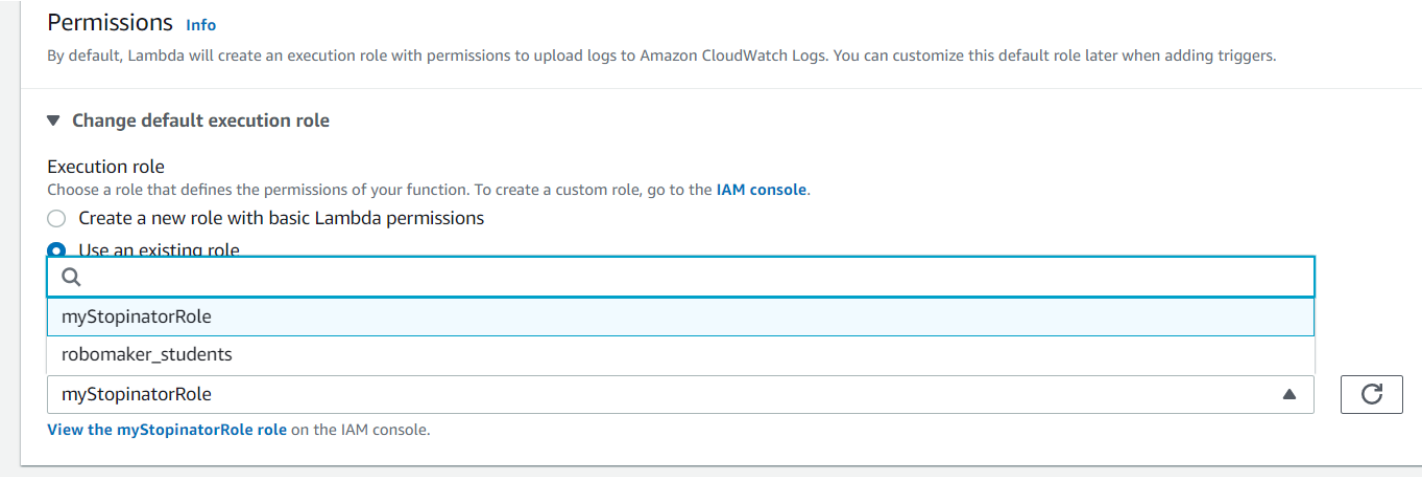
5. In the AWS Management Console, from the Services menu, choose Lambda.
   Note: If you see a warning message that says *tags failed to load*, you can ignore it.



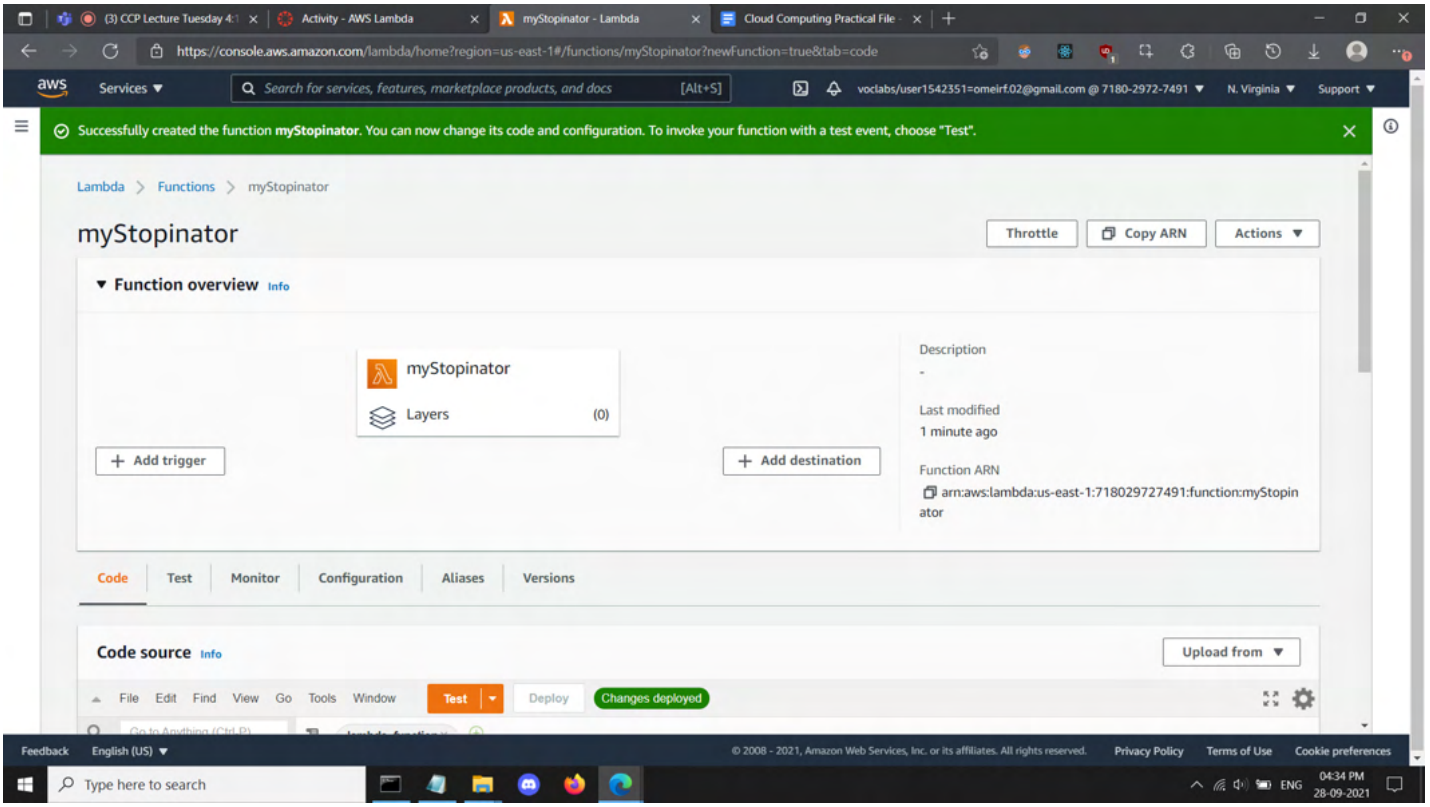
6. Click Create function.
7. In the Create function screen, configure these settings:

- Choose Author from scratch
- Function name: myStopinator
- Runtime: Python 3.8
- Click Choose or create an execution role
- Execution role: Use an existing role
- Existing role: From the dropdown list, choose myStopinatorRole





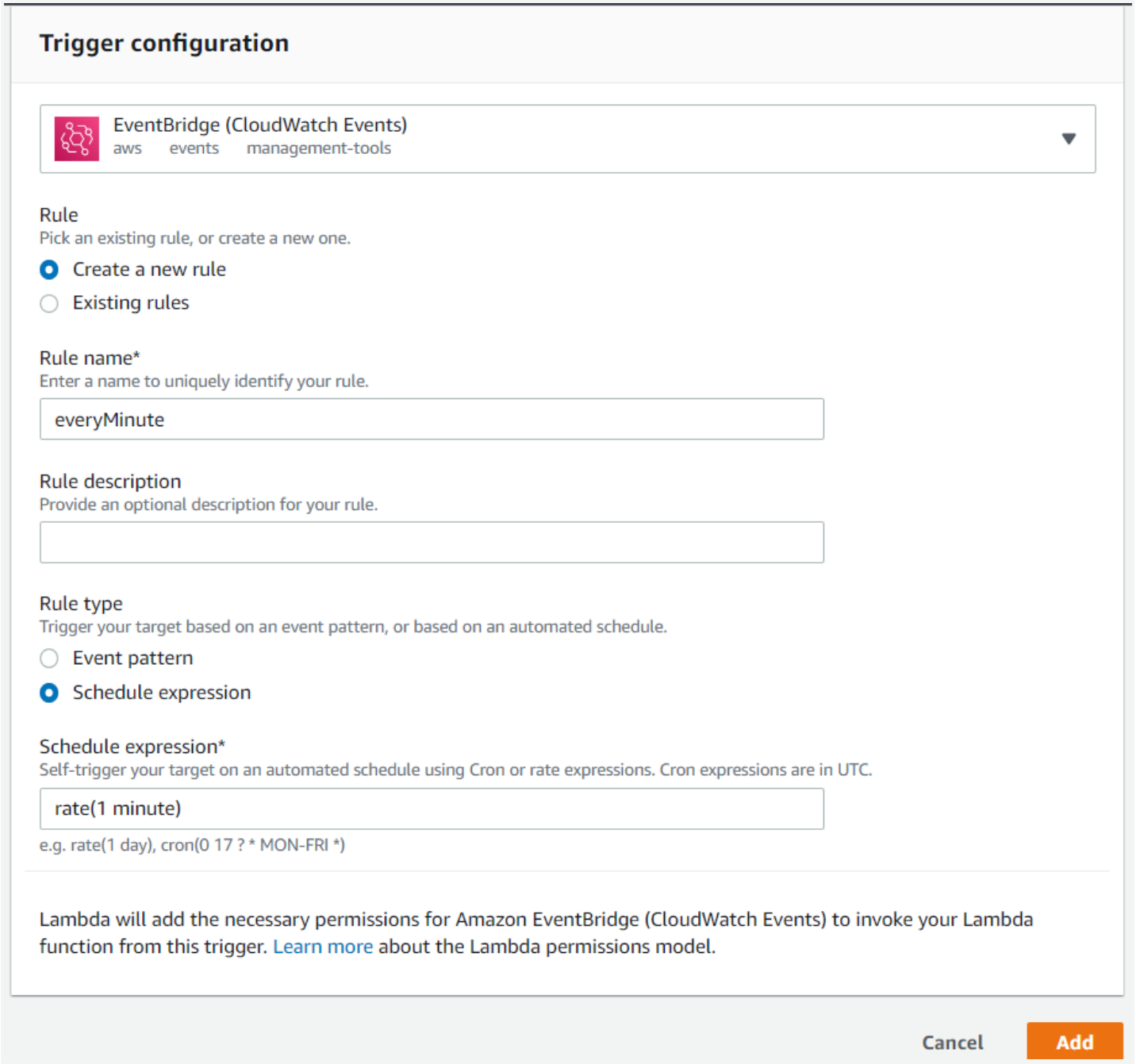
8. Click Create function.



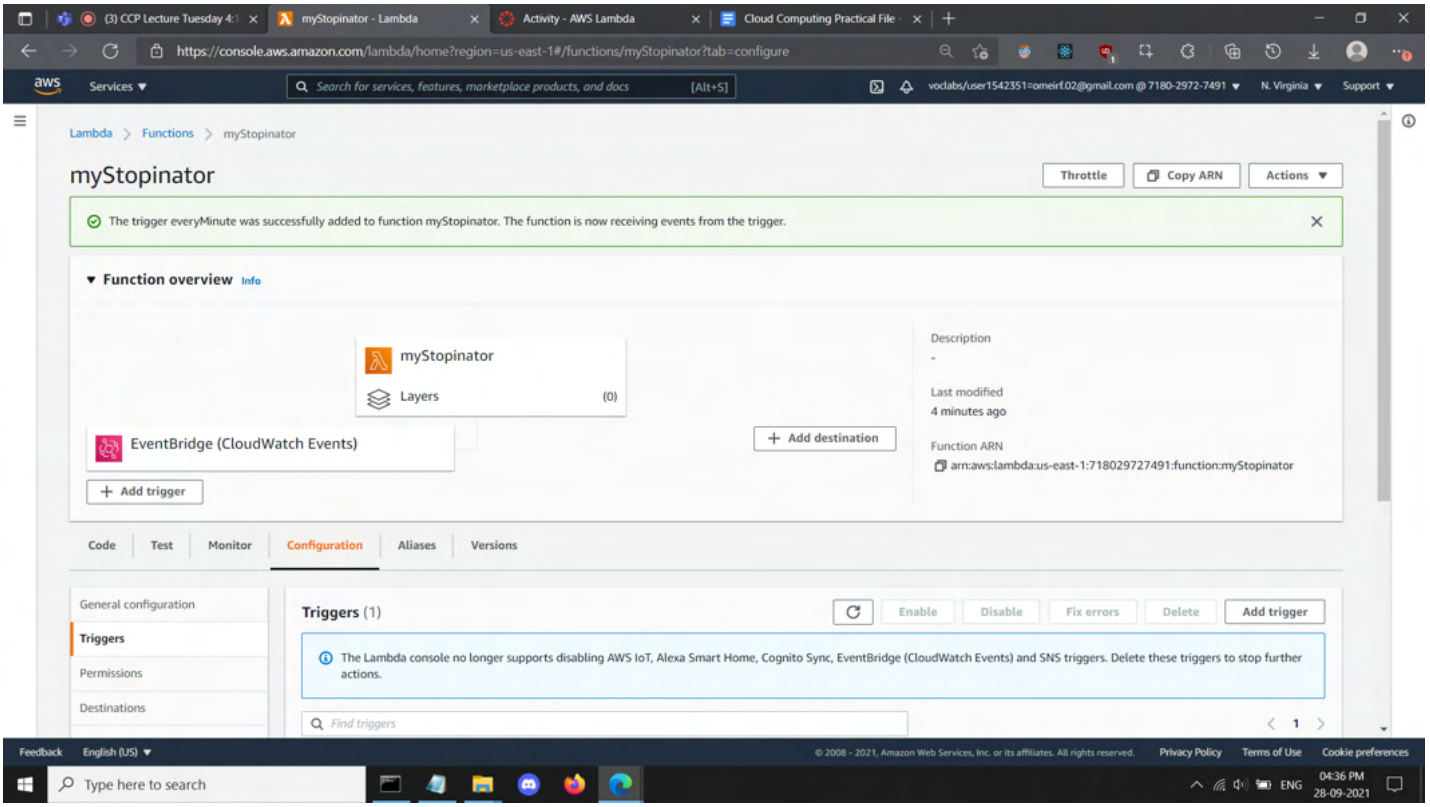
Task 2: Configure the trigger

In this task, you will configure a scheduled event to trigger the Lambda function by setting a CloudWatch event as the event source (or trigger). The Lambda function can be configured to operate much like a cron job on a Linux server, or a scheduled task on a Microsoft Windows server. However, you do not need to have a server running to host it.

9. Click **+ Add trigger**.
10. Click the **Select a trigger** dropdown menu, and choose **EventBridge (CloudWatch Events)**.
11. For the rule, choose **Create a new rule** and configure these settings:
    - Rule name: `everyMinute`
    - Rule type: **Schedule expression**
    - Schedule expression: `rate(1 minute)`



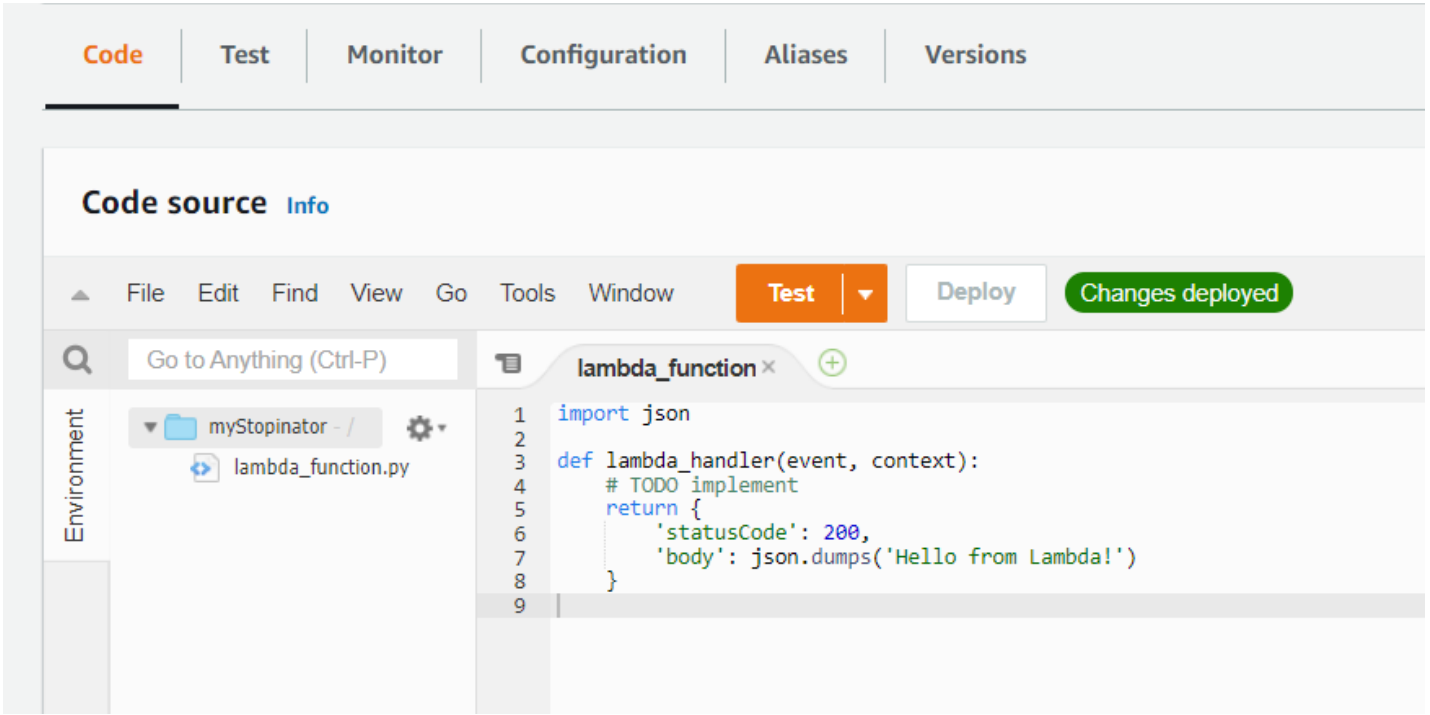
12. **Note**: A more realistic, schedule-based stopinator Lambda function would probably be triggered by using a cron expression instead of a rate expression. However, for the purposes of this activity, using a rate expression ensures that the Lambda function will be triggered soon enough that you can see the results.
13. Click Add.



Task 3: Configure the Lambda function

In this task, you will paste a few lines of code to update two values in the function code. You do not need to write code to complete this task.

13. Below the **Function overview** pane, choose **Code**, and then choose *lambda_function.py* to display and edit the Lambda function code.
14. In the **Code source** pane, delete the existing code. Copy the following code, and paste it in the box:

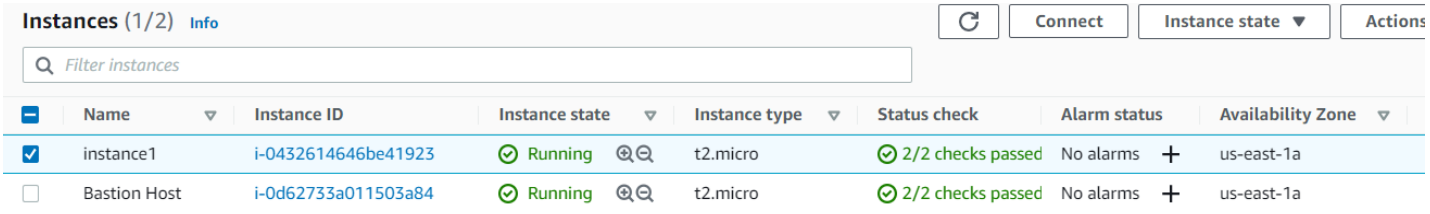


```
import boto3
region = '<REPLACE_WITH_REGION>'
instances = ['<REPLACE_WITH_INSTANCE_ID>']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

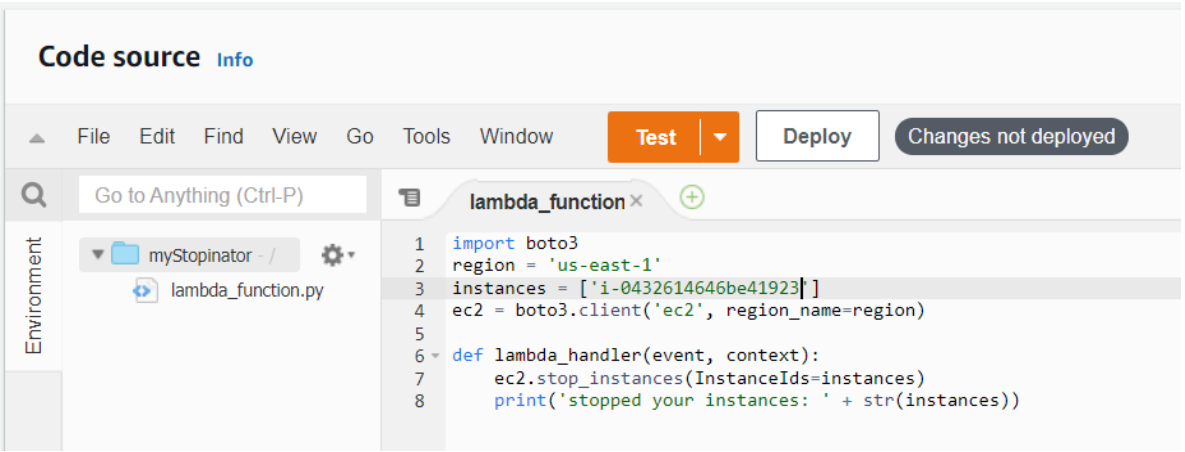
Note: After pasting the code into the **Code source** box, review line 5. If a period (.) was added, delete it.

15. Replace the `<REPLACE_WITH_REGION>` placeholder with the actual Region that you are using. To do this:
    Click on the region on the top right corner and use the region code. For example, the region code for US East (N. Virginia) is *us-east-1*.
    Important: Keep the single quotation marks (' ') around the Region in your code. For example, for the N. Virginia, it would be `'us-east-1'`
16. **Challenge section**: Verify that an EC2 instance named *instance1* is running in your account, and copy the *instance1* **instance ID**.

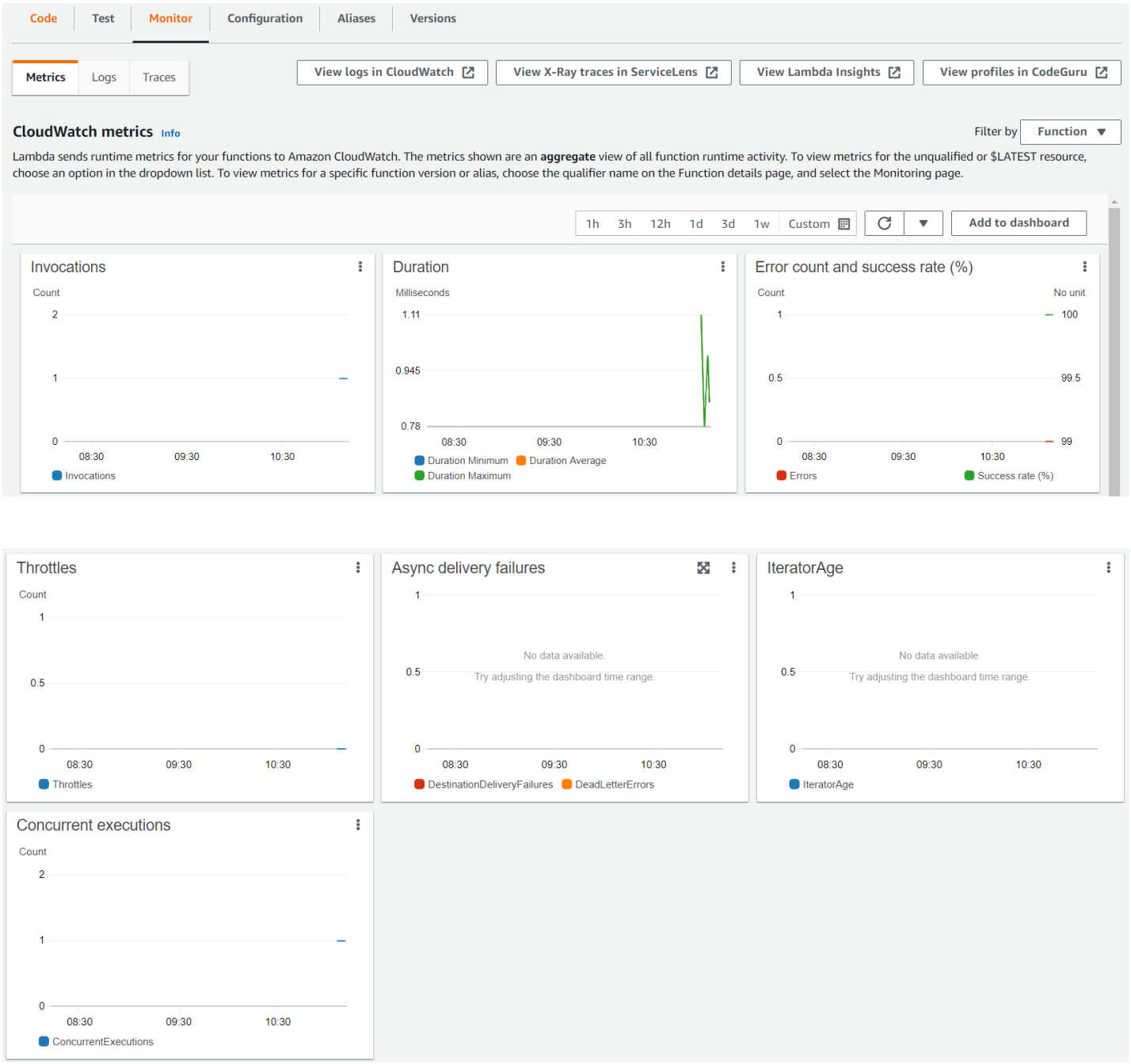


You are encouraged to figure out how to do this task without specific step-by-step guidance. However, **if you need detailed guidance, click here**.

17. Return to the **AWS Lambda console** browser tab, and replace `<REPLACE_WITH_INSTANCE_ID>` with the actual instance ID that you just copied.
    Important: Keep the single quotation marks (' ') around the instance ID in your code.
    Your code should now look similar to the following example. However, you might have a different value for the Region, and you will have a different value for the instance ID:

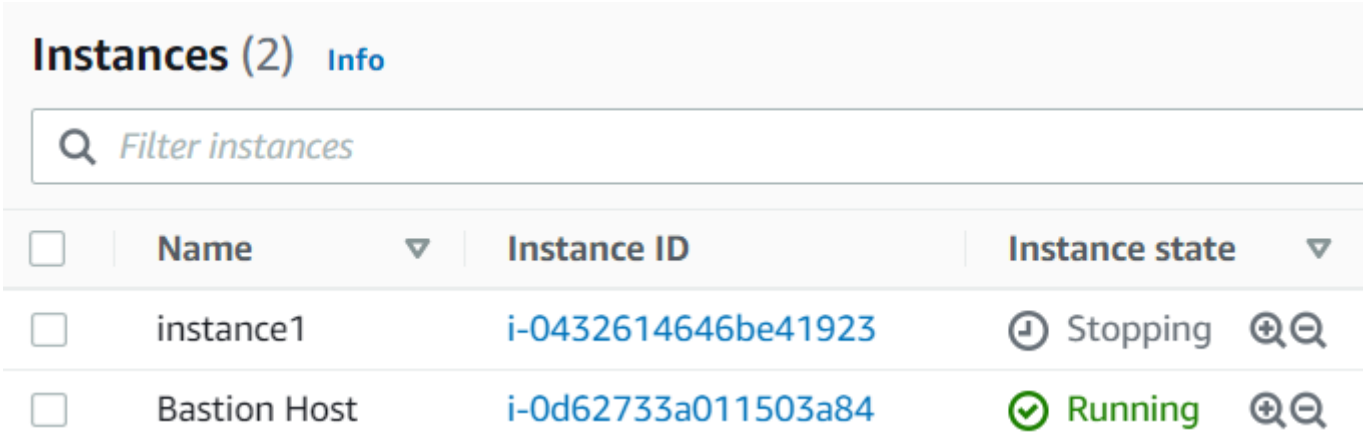


18. In the top-right corner of the **Code source** box, choose **Deploy**.
    Your Lambda function is now fully configured. It should attempt to stop your instance every minute.
19. Click **Monitor** (the tab near the top of the page).
    Note that one of the charts shows you how many times your function has been invoked. There is also a chart that shows the error count and the success rate as a percentage.

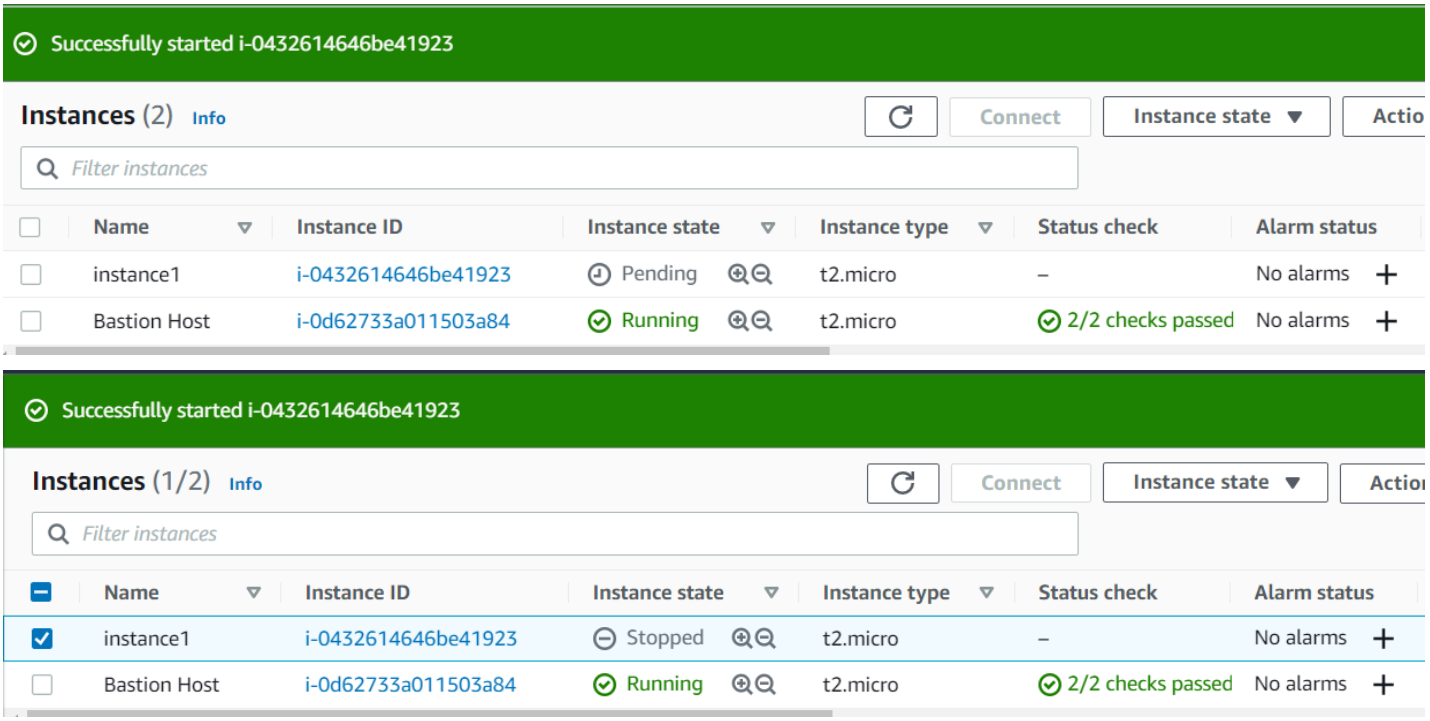


Task 4: Verify that the Lambda function worked

20. Return to the **Amazon EC2 console** browser tab and see if your instance was stopped.
    Tip: You can click the refresh icon or refresh the browser page to see the change in state more quickly.



21. Try starting the instance again. What do you think will happen?
The instance will be stopped again within 1 minute.



Activity 2

AIM

AWS Elastic Beanstalk

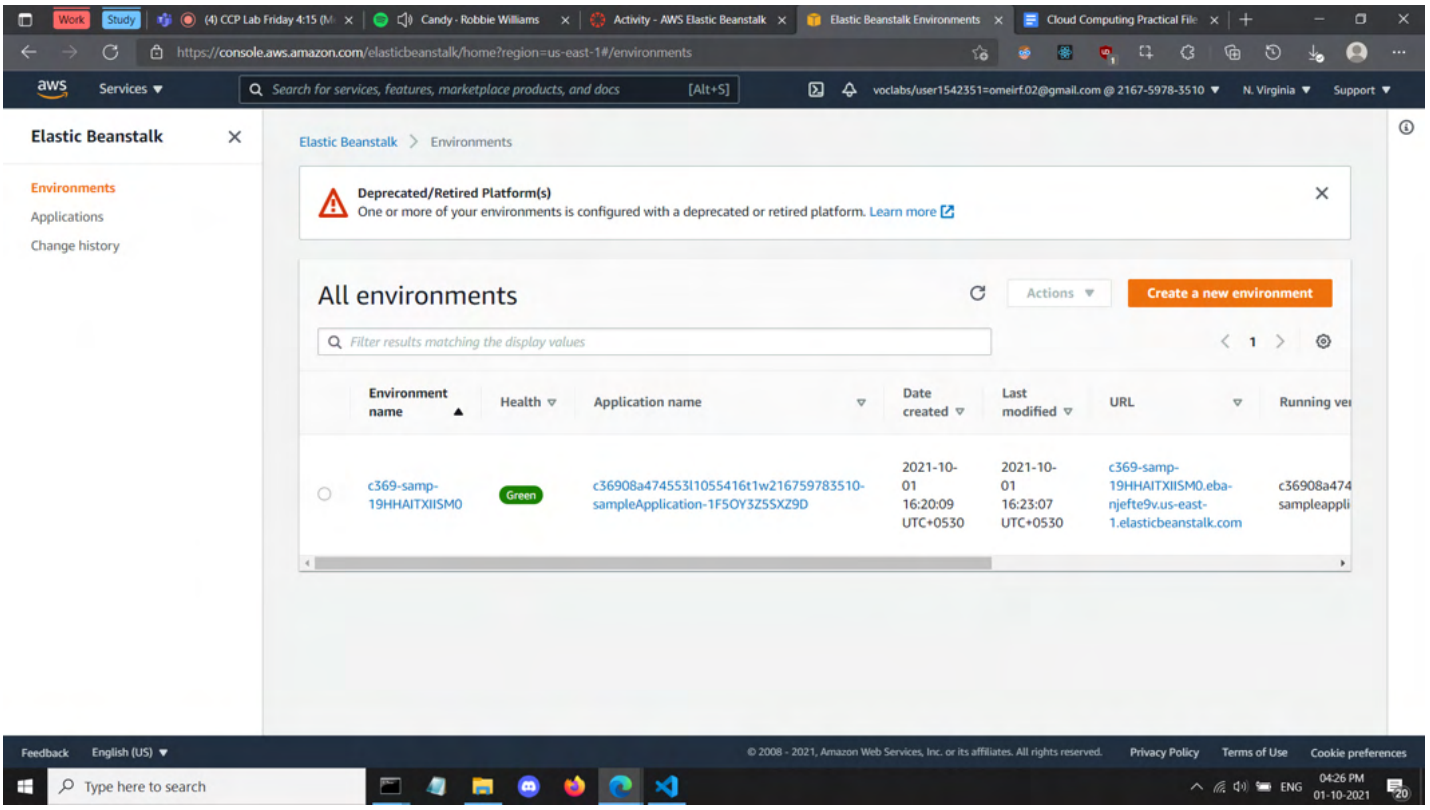
THEORY

This activity provides you with an Amazon Web Services (AWS) account where an AWS Elastic Beanstalk environment has been pre-created for you. You will deploy code to it and observe the AWS resources that make up the Elastic Beanstalk environment.

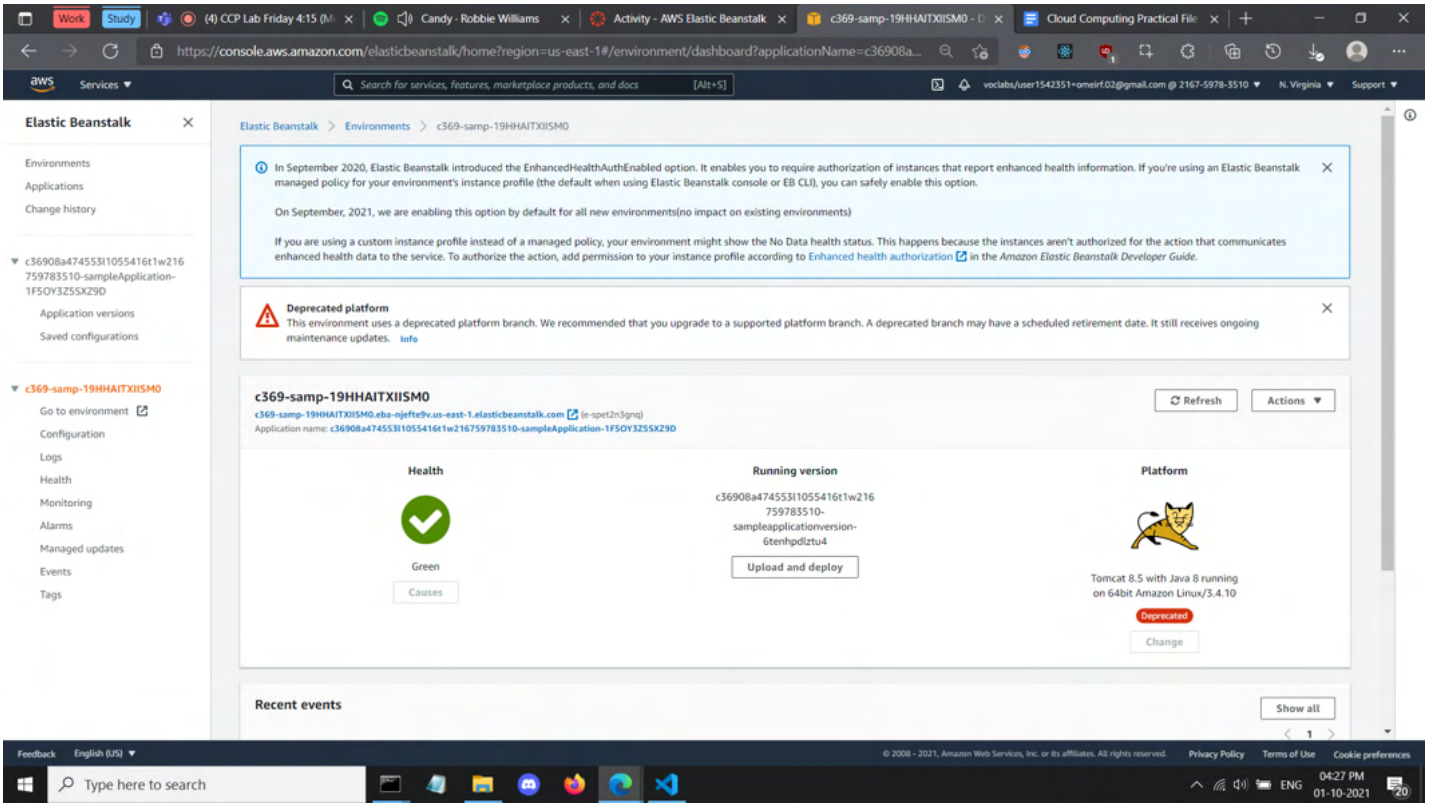
PROCEDURE

Task 1: Access the Elastic Beanstalk environment

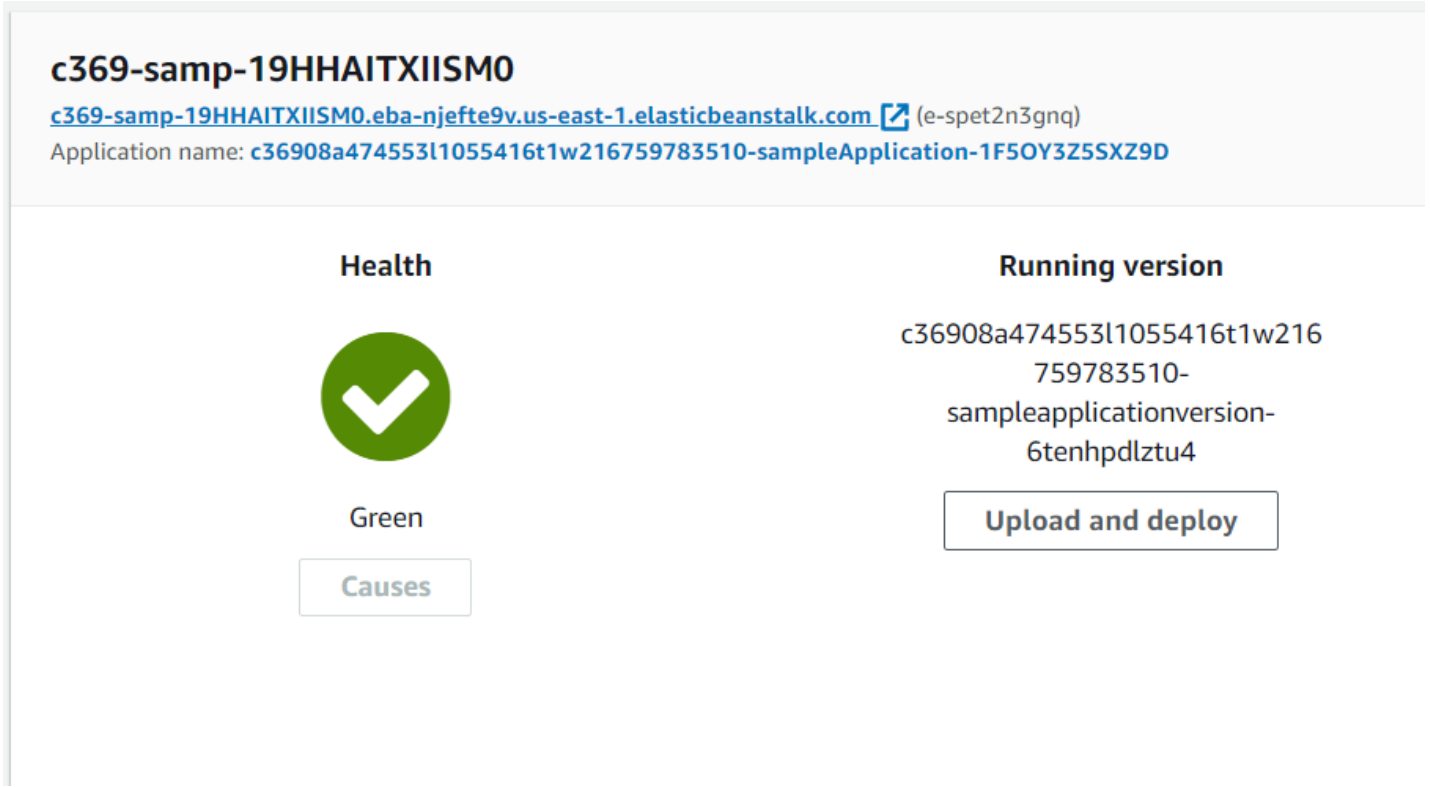
5. In the **AWS Management Console**, from the **Services** menu, choose **Elastic Beanstalk**.
   A page titled **All environments** should open, and it should show a table that lists the details for an existing Elastic Beanstalk application.
   Note: If the status in the **Health** column is not Green, it has not finished starting yet. Wait a few moments, and it should change to Green.

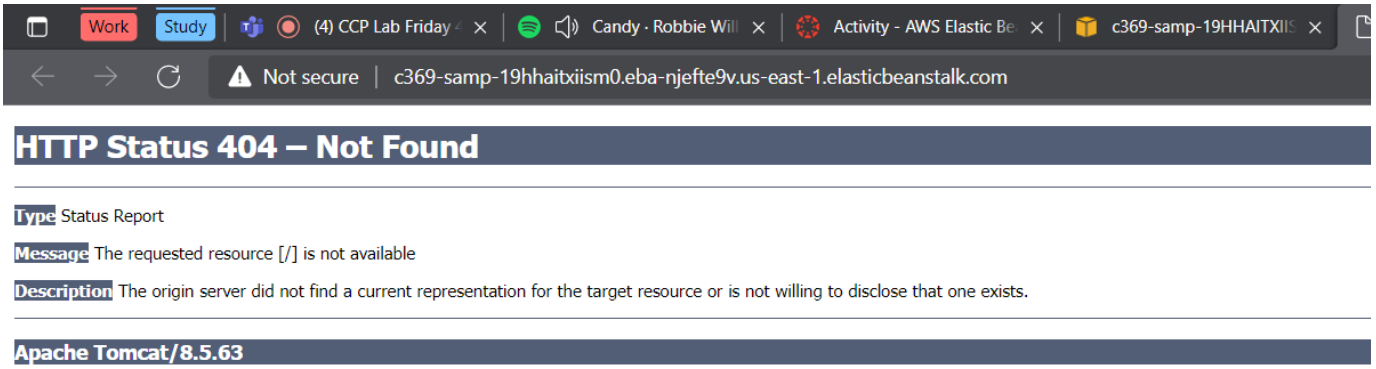


6. Under the **Environment name** column, click on the name of the environment.
   The **Dashboard** page for your Elastic Beanstalk environment opens.



7. Notice that the page shows that the health of your application is Green (good).
   The Elastic Beanstalk environment is ready to host an application. However, it does not yet have running code.
8. Near the top of the page, click the URL (the URL ends in *elasticbeanstalk.com*).
   When you click the URL, a new browser tab opens. However, you should see that it displays an *"HTTP Status 404 - Not Found"* message. *This behavior is expected* because this application server doesn't have an application running on it yet. Return to the Elastic Beanstalk console.

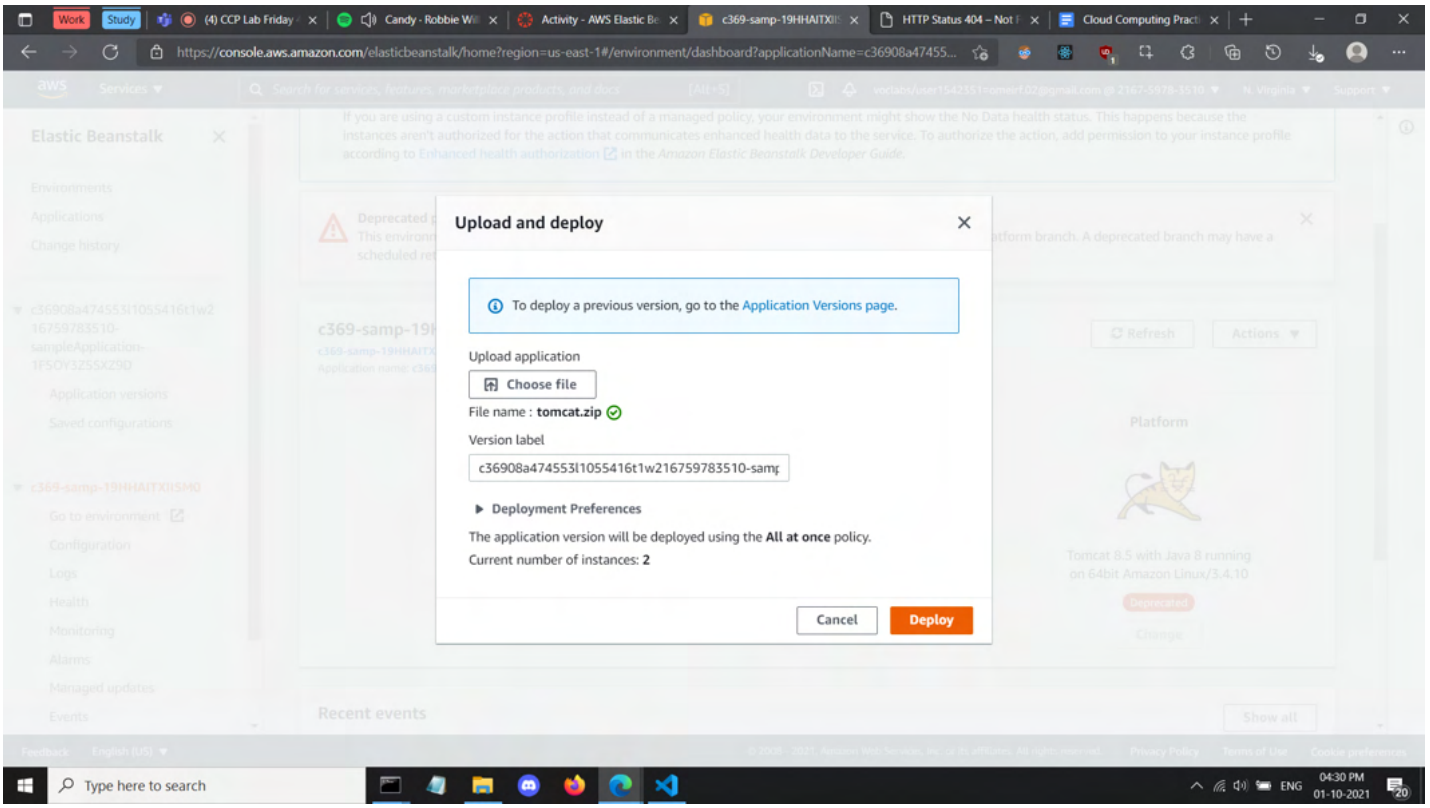




In the next step, you will deploy code in your Elastic Beanstalk environment.

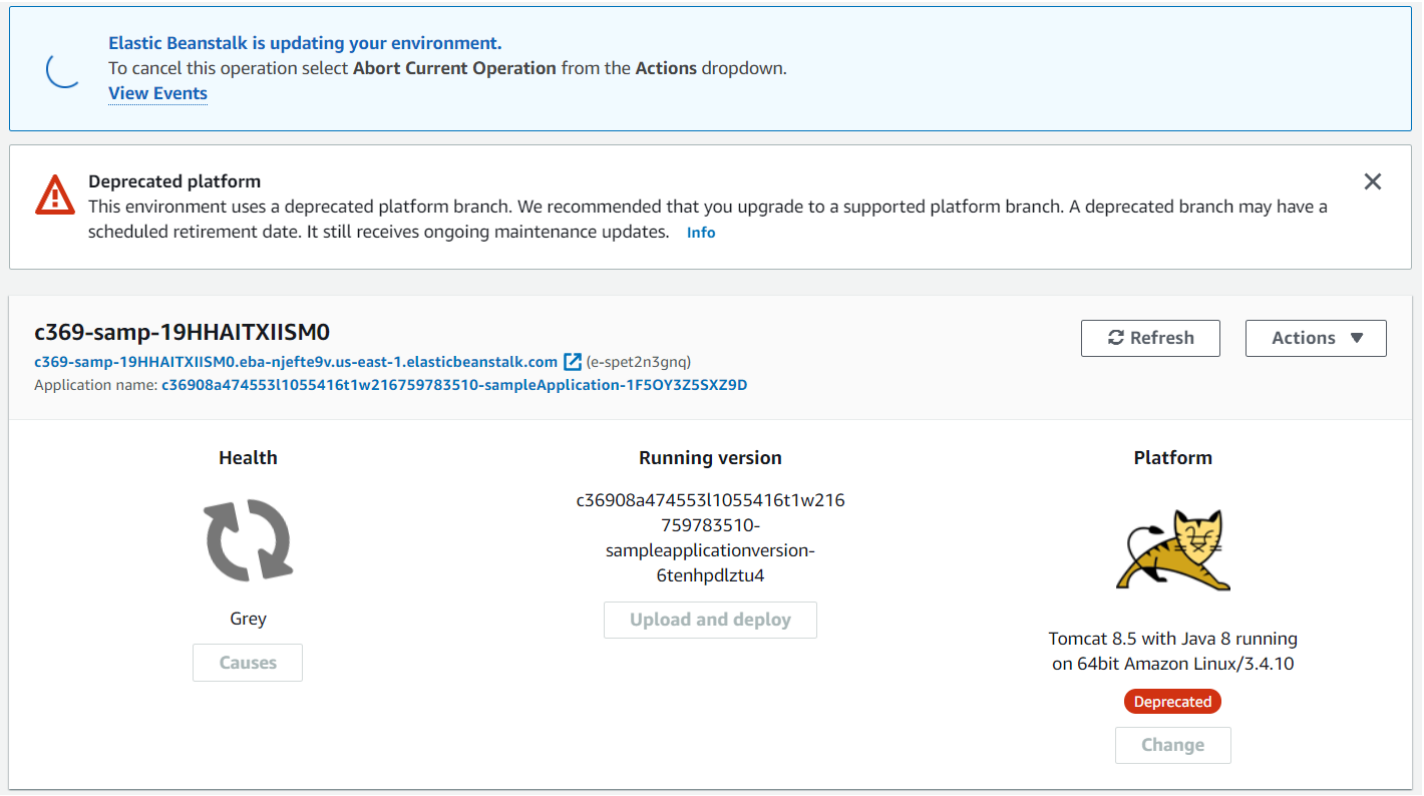
Task 2: Deploy a sample application to Elastic Beanstalk

9. To download a sample application, click this link: https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/tomcat.zip
10. Back in the Elastic Beanstalk Dashboard, click **Upload and Deploy**.
11. Click **Browse or Choose File**, then navigate to and open the **tomcat.zip** file that you just downloaded.



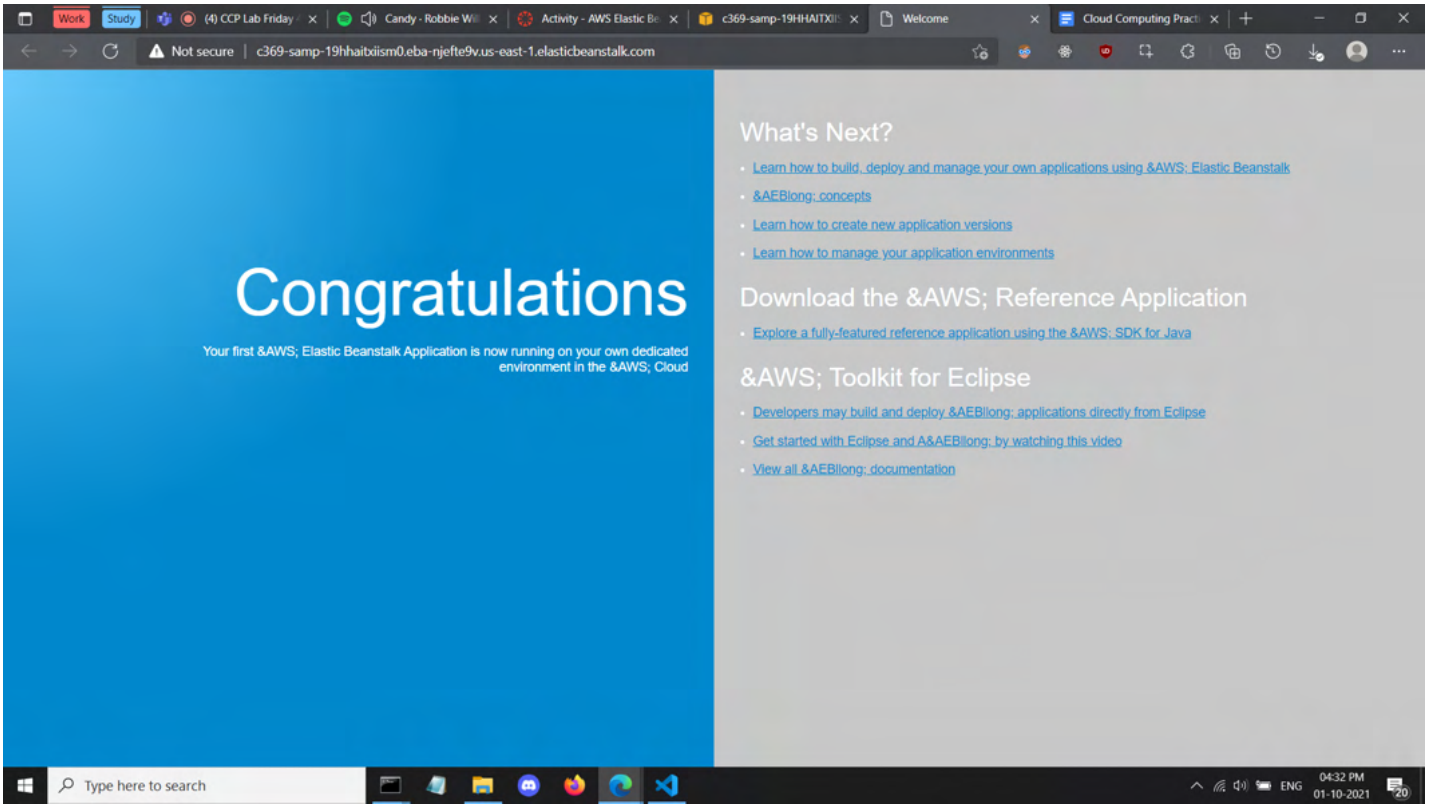
12. Click **Deploy**.
    It will take a minute or two for Elastic Beanstalk to update your environment and deploy the application.

Note: If you see a warning in the Elastic Beanstalk dashboard page that an instance profile is required to integrate with the AWS X-Ray service, you can ignore the warning.

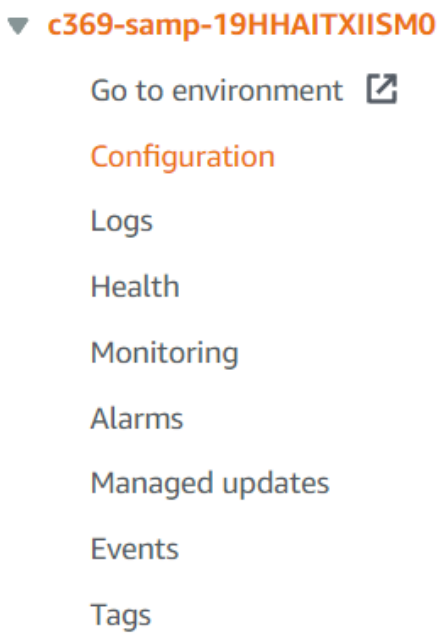


13. After the deployment is complete, click the URL value near the top of the screen (or, if you still have the browser tab that displayed the 404 status, refresh that page).
The web application that you deployed displays.

Congratulations, you have successfully deployed an application on Elastic Beanstalk!



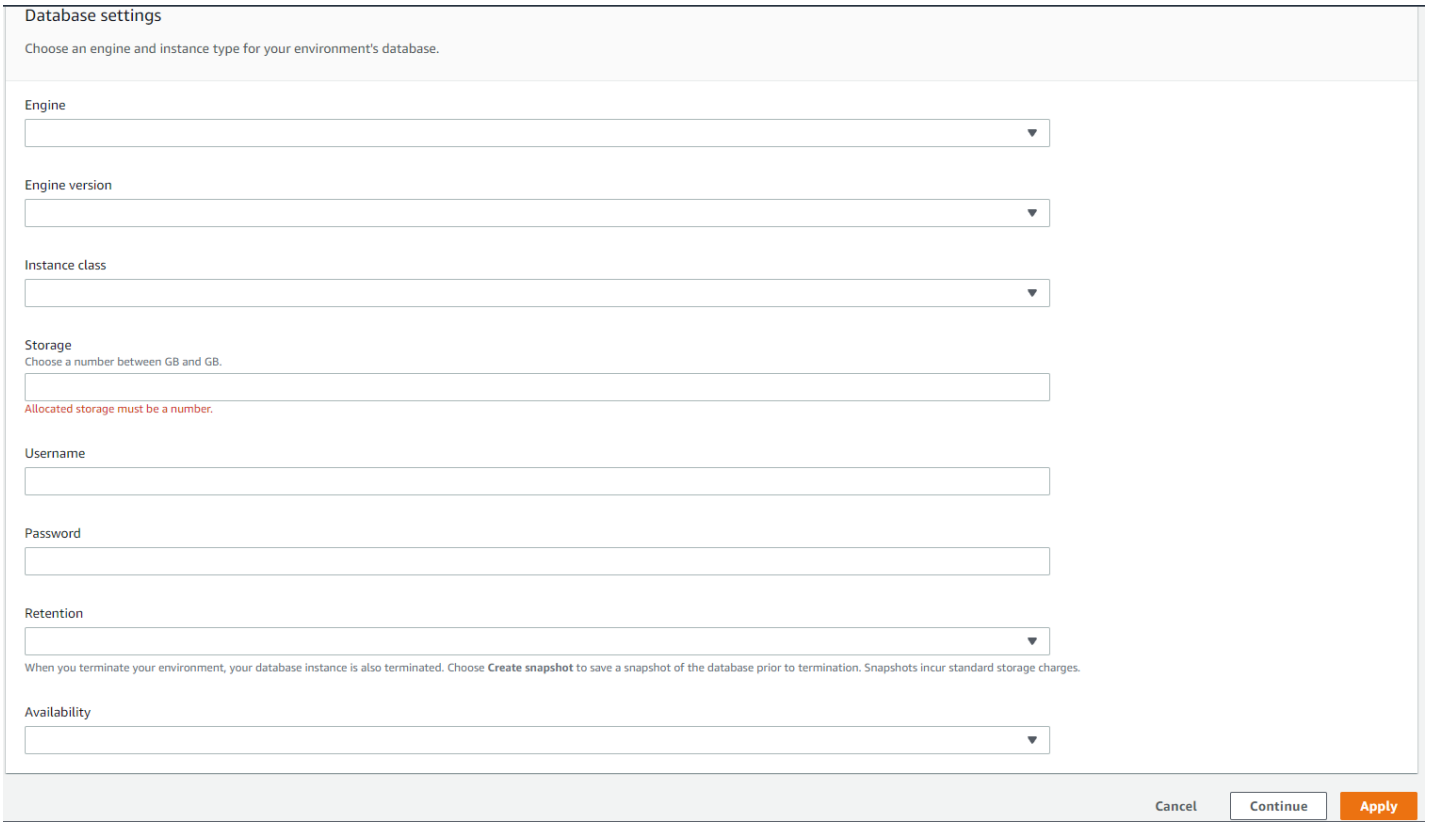
14. Back in the Elastic Beanstalk console, click **Configuration** in the left pane.



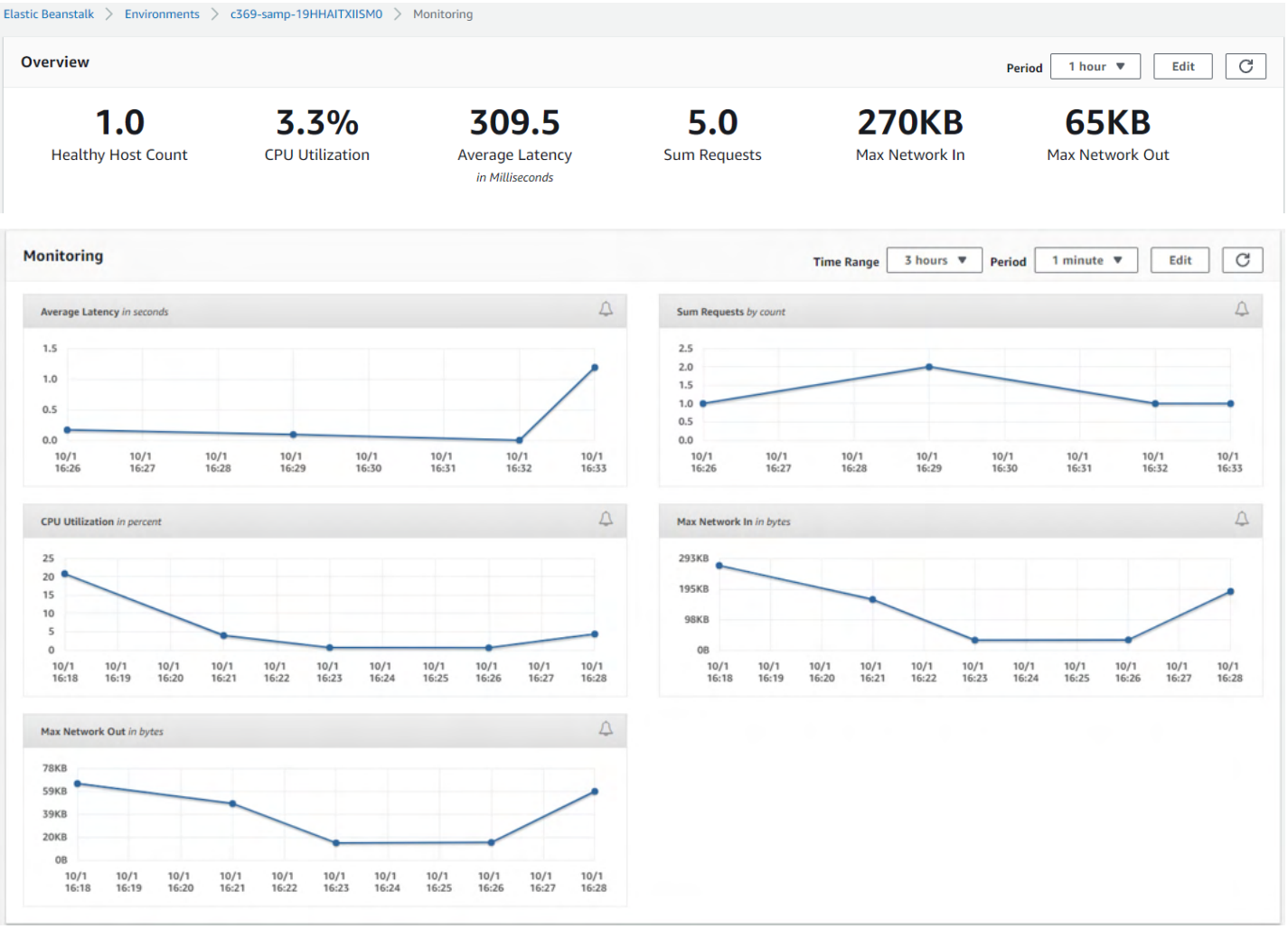
Notice the details here.
For example, in the **Instances** row, it indicates the Monitoring interval, EC2 Security groups, and Root volume type details of the Amazon Elastic Compute Cloud (Amazon EC2) instances that are hosting your web application.

Category	Options	Actions
Software	Environment properties: JDBC_CONNECTION_STRING Gzip compression: enabled Initial JVM heap size (Xms): 256m JVM options: -- Log streaming: disabled Max JVM heap size (Xmx): 256m Proxy server: apache Rotate logs: disabled X-Ray daemon: disabled XX:MaxPermSize: 64m	Edit
Instances	EC2 security groups: awseb-e-spet2n3gnq-stack-AWSEBSecurityGroup-1O4IIVQTJGD7C IOPS: container default Monitoring interval: 5 minute Root volume type: container default Size: container default Throughput: container default	Edit
Capacity	AMI ID: ami-0ed7054c38dad8473 Availability Zones: Any Breach duration: 5 Capacity rebalancing: disabled Environment type: load balancing, auto scaling Instance types: t2.micro,t2.small Lower threshold: 2000000 Max: 6 Metric: NetworkOut Min: 2 Period: 5 Placement: Scale down increment: -1 Scale up increment: 1 Scaling cooldown: 360 seconds Statistic: Average Unit: Bytes Upper threshold: 6000000	Edit

15. Scroll to the bottom of the page to the **Database** row.
    The **Database** row does not have any details because the environment does not include a database.
16. In the **Database** row, click **Edit**.
    Note that you could easily add a database to this environment if you wanted to: you only need to set a few basic configurations and click **Apply**. (However, for the purposes of this activity, you do not need to add a database.)

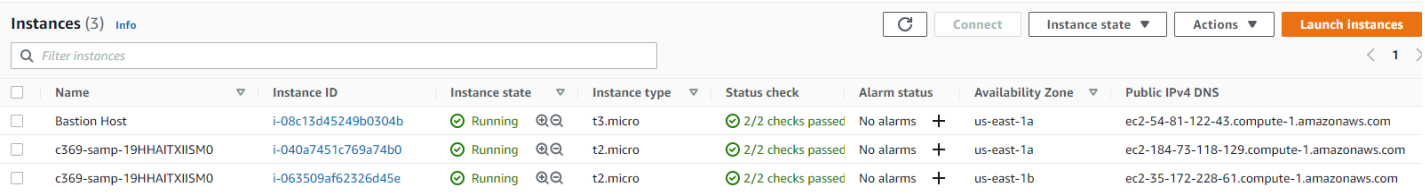


17. In the left panel, click **Monitoring**.
    Browse through the charts to see the kinds of information that are available to you.

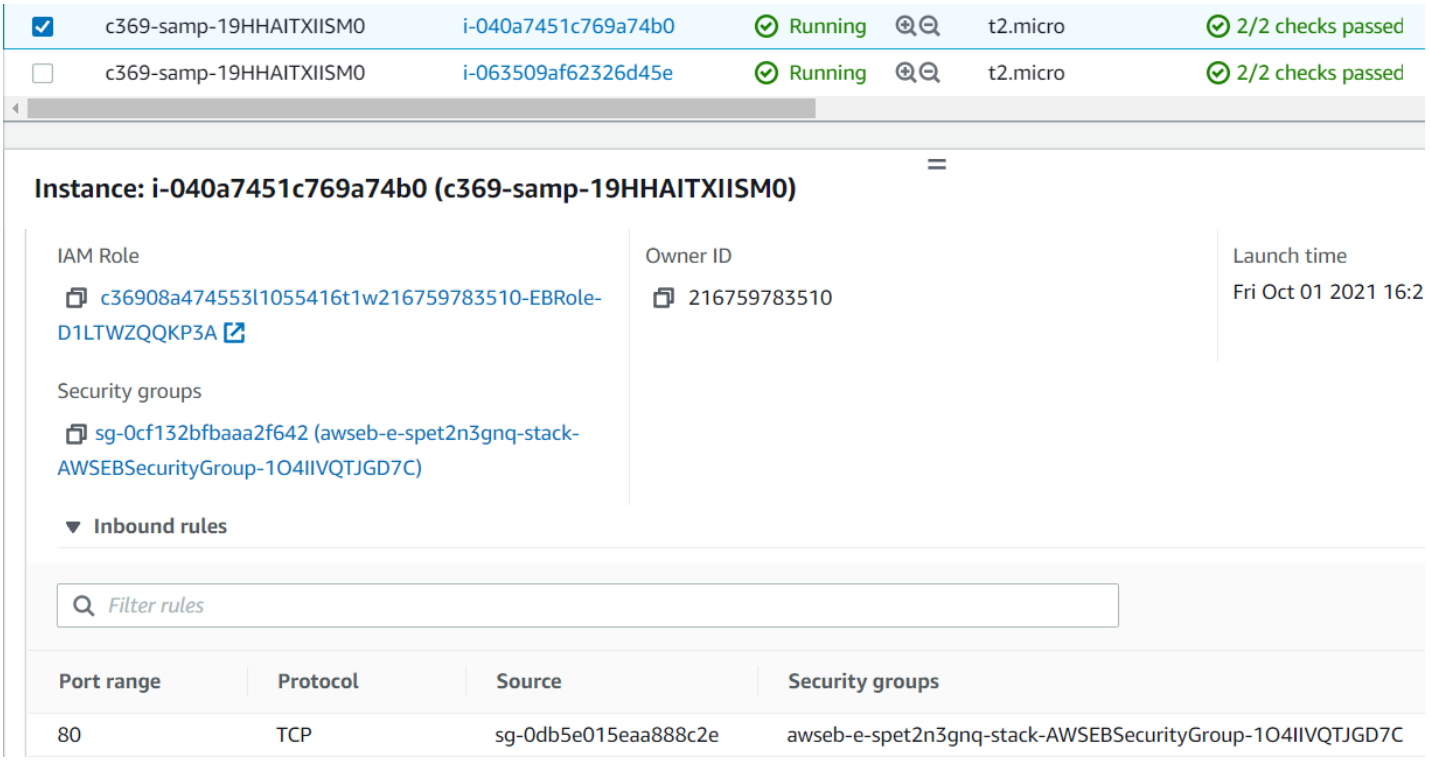


Task 3: Explore the AWS resources that support your application

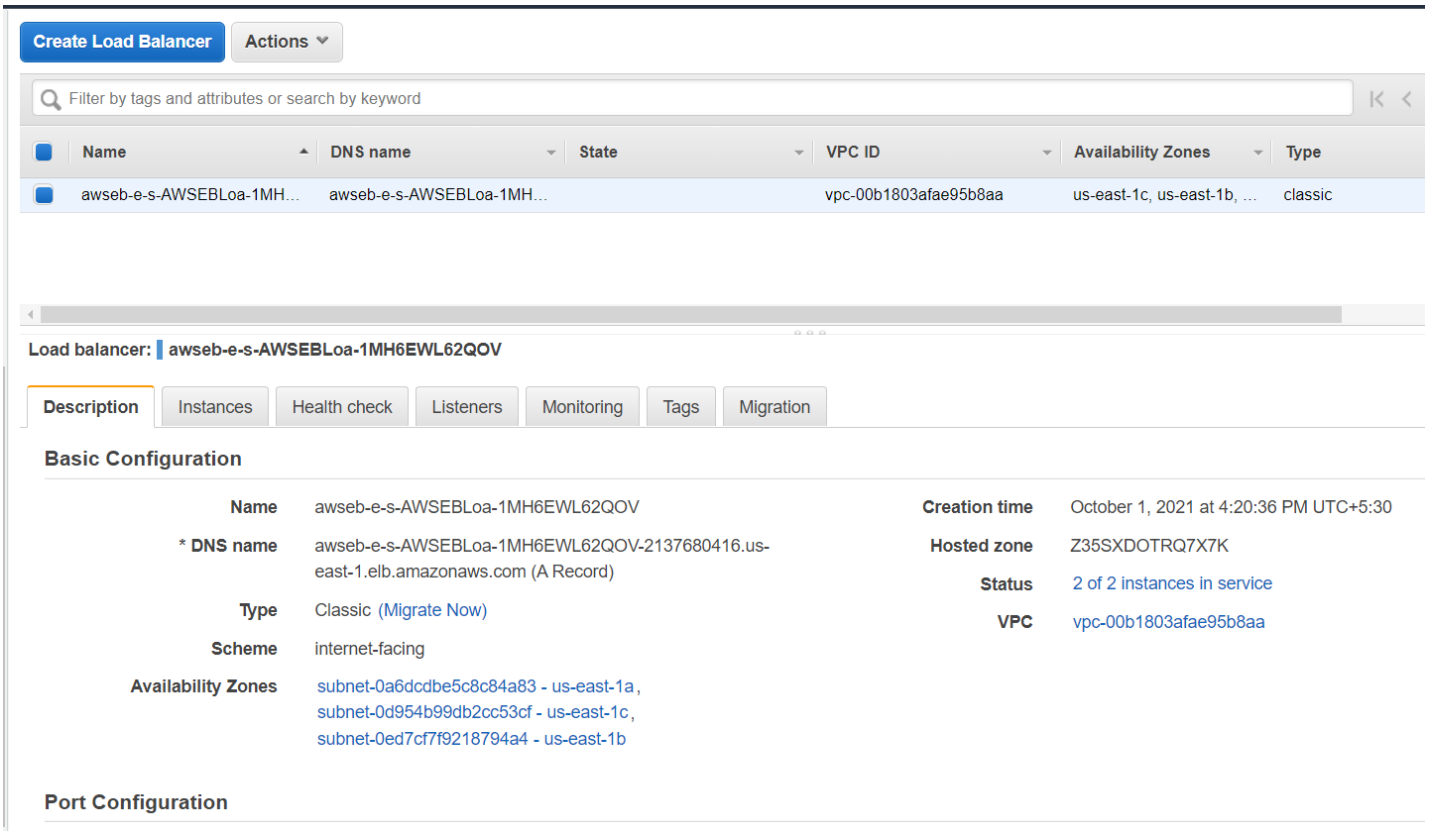
18. From the **Services** menu, choose **EC2**
19. Click **Instances**.
    Note that two instances are running (they both contain *samp* in their names). Both instances support your web application.



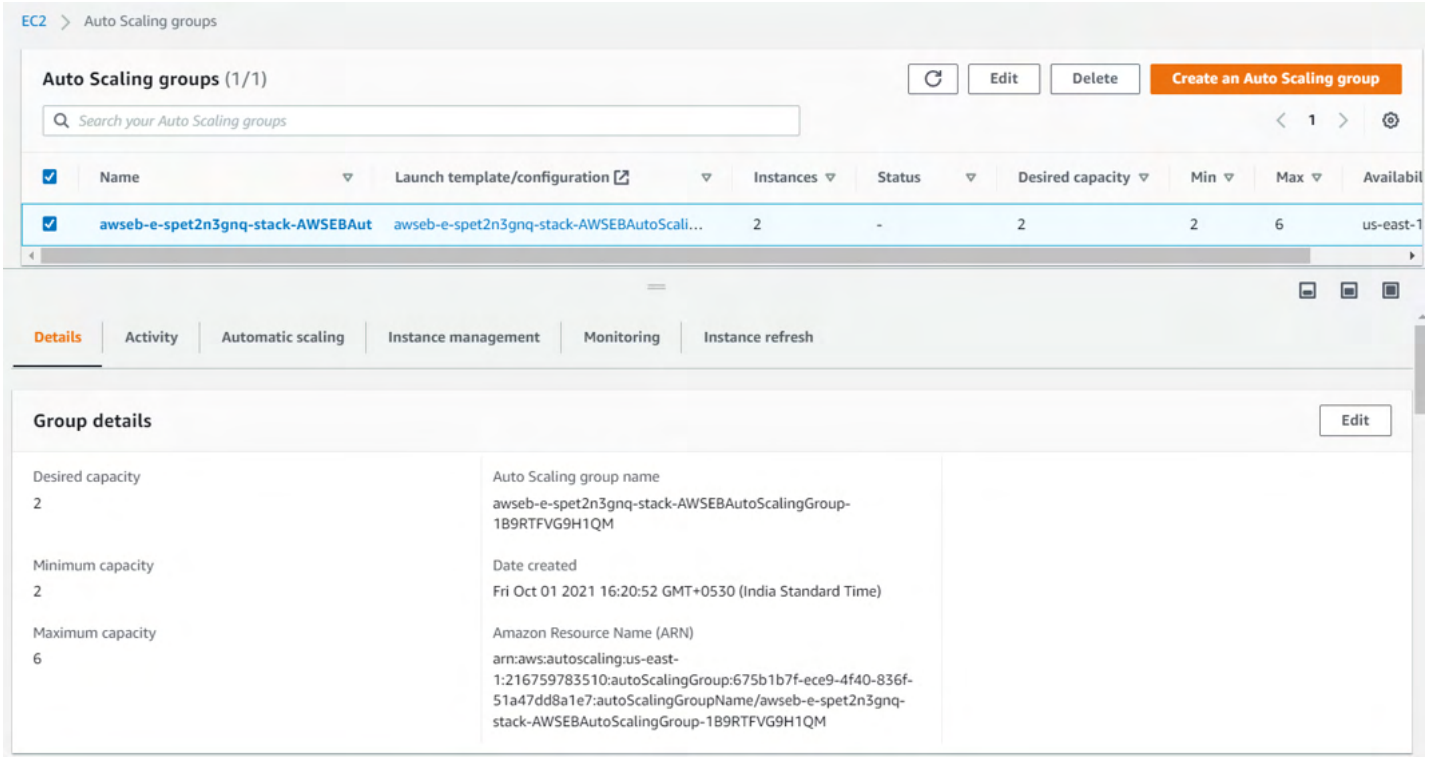
20. If you want to continue exploring the Amazon EC2 service resources that were created by Elastic Beanstalk, feel free to explore them. You will find:
    - A *security group* with port 80 open



- A *load balancer* that both instances belong to



- An *Auto Scaling group* that runs from two to six instances, depending on the network load



21. Though Elastic Beanstalk created these resources for you, you still have access to them.

Activity 3

AIM

To explore the sandbox environment

THEORY

This lab describes how to use the sandbox environment for ad-hoc exploration of Amazon Web Services (AWS) products and services.

You will be restricted to the following services and usage:

- Amazon Elastic Compute Cloud (Amazon EC2)
    - Amazon Machine Images (AMIs) – You can use Amazon provided Linux and Microsoft Windows AMIs
    - Instances – You can only launch the following instance types: t2.nano, t2.micro, t2.small, t2.medium, t3.nano, t3.micro, t3.small, t3.medium
    - Amazon Elastic Block Store (Amazon EBS) volumes – You can use volumes in sizes up to 35 GB and of type General Purpose SSD (gp2)
    - On-Demand Instances
- Amazon EC2 Auto Scaling
- Elastic Load Balancing
- AWS Lambda
- Amazon Simple Storage Service (Amazon S3)
- Amazon Simple Storage Service Glacier
- Amazon Relational Database Service (Amazon RDS)
    - DB Instance Class Types – You can use db.t2 and db.t3 instances of type db.t*.micro to db.t*.medium
    - EBS volumes – You can use volumes in sizes up to 100 GB and of type General Purpose SSD (gp2)
    - On-Demand DB Instance class types
    - DatabaseEngine – Amazon Aurora, MySQL, PostgreSQL, and MariaDB
    - Multi-AZ deployments are not supported – You can choose the Dev/Test or AWS Free Tier templates if prompted, and you can't create a standby instance
    - Enhanced monitoring is not supported – You must clear this default setting
- Amazon DynamoDB
- AWS Identity and Access Management (IAM) – You have read-only access
- AWS CloudFormation
- Amazon CloudWatch
- Amazon Simple Notification Service (Amazon SNS)
- Amazon Route 53
- Amazon CloudFront
- AWS CloudTrail
- AWS Key Management Service (AWS KMS) – You have list access
- Tagging

PROCEDURE

Using the terminal in the browser

A terminal window displays to the right of these instructions.

You can toggle the visibility of the terminal window by selecting or clearing the check box in the *Terminal* box at the top of the screen.

The terminal in the browser provides access to a Linux shell on a server that exists *outside* of the AWS account that you use when your lab is running.

Running AWS CLI commands

After you start the lab, the terminal will be preconfigured with the credentials that you need to use the AWS Command Line Interface (AWS CLI).

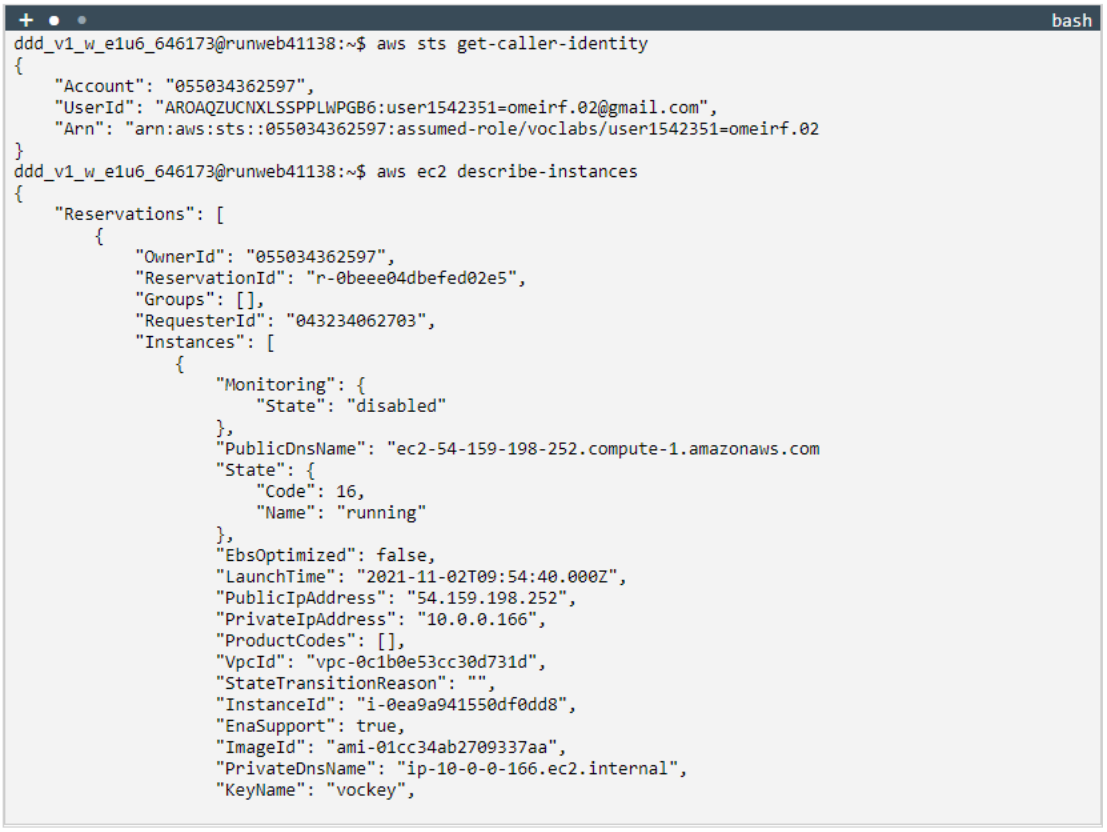
For example, to retrieve the account number and your user ID, run the following command:

```
aws sts get-caller-identity
```

If you have any EC2 instances that are running in the sandbox, this command provides information about them:

```
aws ec2 describe-instances
```

For details about how to use the AWS CLI, refer to the AWS CLI command reference documentation.



Using the AWS SDK for Python

Python V3 is installed in the terminal, with access to the AWS software development kit (SDK) for Python (Boto3) library. You can use this library to run AWS SDK for Python code. For example:

```
$ python3
>>> import boto3
>>> ec2 = boto3.client('ec2', region_name='us-east-1')
>>> ec2.describe_regions()
>>> exit()
$
```

```
ddd_v1_w_e1u6_646173@runweb41138:~$ python3
Python 3.6.4 (default, Feb  9 2018, 18:50:54)
[GCC 5.4.1 20160904] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> ec2 = boto3.client('ec2', region_name='us-east-1')
>>> ec2.describe_regions()
{'Regions': [{'Endpoint': 'ec2.eu-north-1.amazonaws.com', 'RegionName': 'eu-north-1'}, {'Endpoint': 'ec2.a
p-south-1.amazonaws.com', 'RegionName': 'ap-south-1'}, {'Endpoint': 'ec2.eu-west-3.amazonaws.com', 'Region
Name': 'eu-west-3'}, {'Endpoint': 'ec2.eu-west-2.amazonaws.com', 'RegionName': 'eu-west-2'}, {'Endpoint':
'ec2.eu-west-1.amazonaws.com', 'RegionName': 'eu-west-1'}, {'Endpoint': 'ec2.ap-northeast-3.amazonaws.com'
, 'RegionName': 'ap-northeast-3'}, {'Endpoint': 'ec2.ap-northeast-2.amazonaws.com', 'RegionName': 'ap-nort
heast-2'}, {'Endpoint': 'ec2.ap-northeast-1.amazonaws.com', 'RegionName': 'ap-northeast-1'}, {'Endpoint':
'ec2.sa-east-1.amazonaws.com', 'RegionName': 'sa-east-1'}, {'Endpoint': 'ec2.ca-central-1.amazonaws.com',
'RegionName': 'ca-central-1'}, {'Endpoint': 'ec2.ap-southeast-1.amazonaws.com', 'RegionName': 'ap-southeas
t-1'}, {'Endpoint': 'ec2.ap-southeast-2.amazonaws.com', 'RegionName': 'ap-southeast-2'}, {'Endpoint': 'ec2
.eu-central-1.amazonaws.com', 'RegionName': 'eu-central-1'}, {'Endpoint': 'ec2.us-east-1.amazonaws.com', '
RegionName': 'us-east-1'}, {'Endpoint': 'ec2.us-east-2.amazonaws.com', 'RegionName': 'us-east-2'}, {'Endpo
int': 'ec2.us-west-1.amazonaws.com', 'RegionName': 'us-west-1'}, {'Endpoint': 'ec2.us-west-2.amazonaws.com
', 'RegionName': 'us-west-2'}], 'ResponseMetadata': {'RequestId': '3b3a6414-4621-4fe8-a85c-4013f0034193',
'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': '3b3a6414-4621-4fe8-a85c-4013f0034193', 'cache-
control': 'no-cache, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDomains', 'vary'
: 'accept-encoding', 'content-type': 'text/xml;charset=UTF-8', 'content-length': '3875', 'date': 'Tue, 02
Nov 2021 10:01:00 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
>>> exit()
ddd_v1_w_e1u6_646173@runweb41138:~$
```

For details about how to use the AWS SDK for Python, refer to the Boto3 documentation page.

Accessing EC2 instances

When you launch EC2 instances in the sandbox environment, choose the option to use the existing key pair that's named *vockey* at the time of launch. Then:

- Above these instructions (that you are currently reading), choose the Details dropdown menu.
- Choose Show
  A **Credentials** window opens.

AWS CLI: Show

Cloud Labs

Remaining session time: 03:50:36(231 minutes)
Session started at: 2021-11-02T02:52:12-0700
Session to end at: 2021-11-02T06:52:12-0700

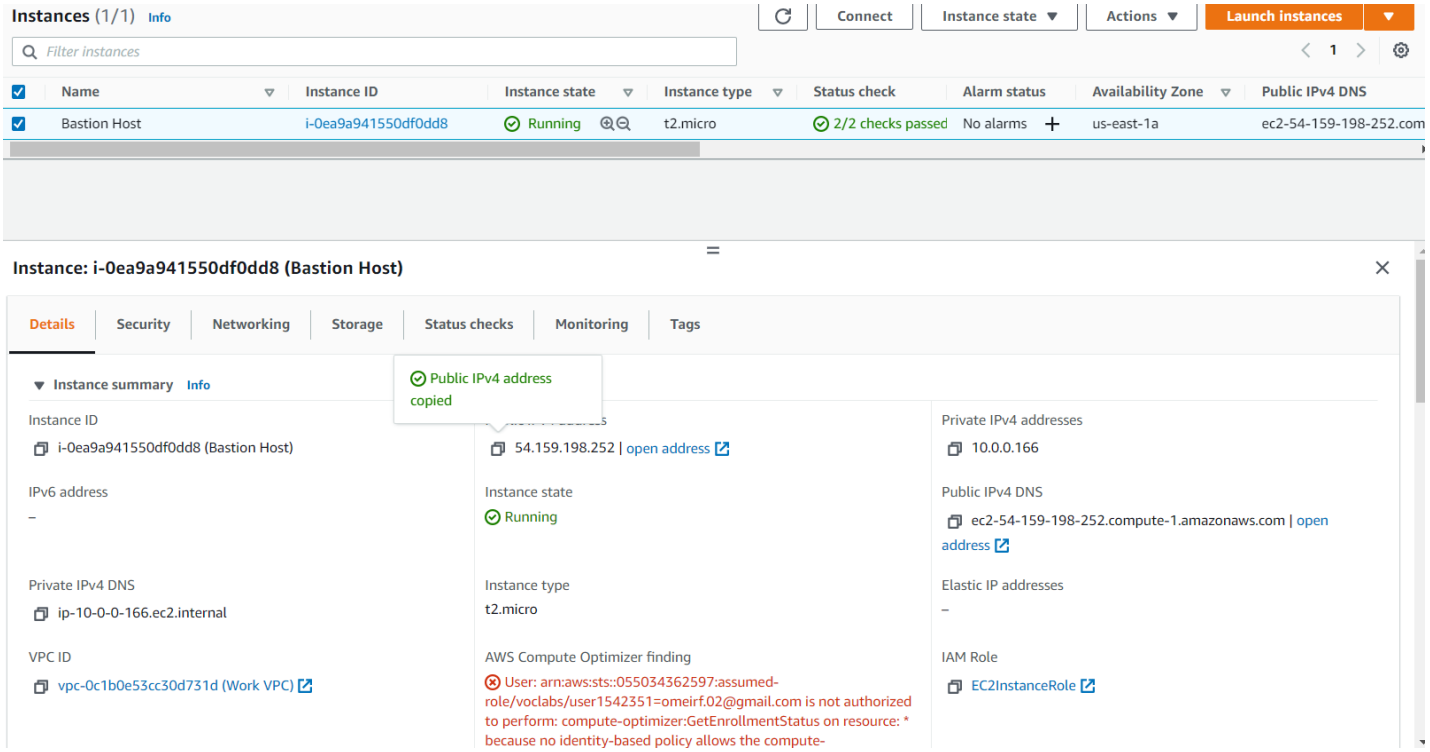
Accumulated lab time: 00:09:24 (10 minutes)

ips -- public:54.159.198.252, private:10.0.0.166

SSH key Show Download PEM Download PPK

AWS SSO Download URL

SecretKey	mQkyYxb1uqjZ4n/qKE53NaC23QxcKICl6FTYwIxa
BastionHost	54.159.198.252
Region	us-east-1
AccessKey	AKIAQZUCNXLSQLIYWXXR



- If you use a *Microsoft Windows computer* and want to connect to a Linux instance, or connect to a Microsoft Windows instance by using Secure Shell (SSH): Choose **Download PPK** and save the **labsuser.ppk** file.
- If you want to connect to a Windows instance desktop from any computer: Choose **Download PEM** and save the **labsuser.pem** file.
- If you use *macOS* and want to connect to a Linux instance by using SSH: Choose **Download PEM** and save the **labsuser.pem** file.

- Typically, your browser will save the labsuser.pem or the labsuser.ppk file to your **Downloads** directory.
- Exit the **Credentials** window by choosing the **X**.
- *To connect to the desktop of a Microsoft Windows instance*:

    - In the EC2 console, choose **Instances** and then choose the instance that you want to connect to.
    - From the **Actions** menu, choose **Get Windows Password**.
    - Next to *Key Pair Path*, choose **Browse**.
    - Browse to and select the labsuser.pem file that you downloaded earlier.
    - Choose **Decrypt Password**.
    - The connection information should display, including the instance's public Domain Name System (DNS), administrator user name, and the decrypted password.
    - Use a Remote Desktop Protocol (RDP) client to connect to the desktop of the EC2 instance by using these connection details.
- *To connect to a Linux instance by using SSH, see the next section*.

Using SSH to access any EC2 instances that you launch

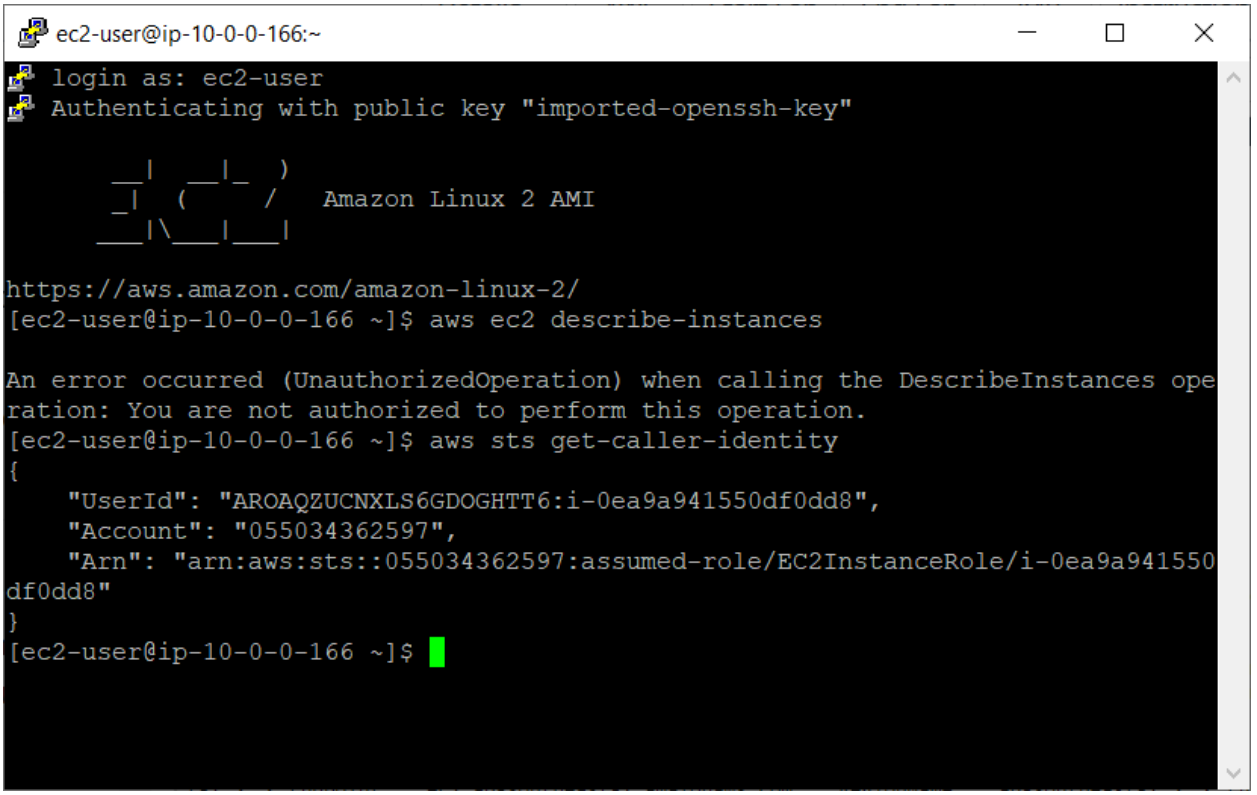
The following steps describe how to use the SSH key to connect to your instance.

Microsoft Windows users

These instructions are for Windows users only.

If you use macOS or Linux, skip to the next section.

6. Download the required software.
    - You will use *PuTTY* to use SSH to connect to EC2 instances. If you do not have PuTTY installed on your computer, download it here.
7. Open **putty.exe**
8. Configure PuTTY to not timeout:
    - Choose **Connection**
    - Set **Seconds between keepalives** to `30`
9. This setting enables you to keep the PuTTY session open for a longer time.
10. Configure your PuTTY session:
    - Choose **Session**
    - **Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the instance. To find it, return to the Amazon EC2 console and choose **Instances**. Select the box next to the instance and in the **Description** tab, copy the **IPv4 Public IP** value.
    - Back in PuTTY, in the **Connection** list, expand **SSH**
    - Choose **Auth** (don't expand it).
    - Choose **Browse**.
    - Browse to and select the .ppk file that you downloaded
    - Choose **Open** to select it
    - Choose **Open**
11. Choose **Yes**, to trust the host and connect to it.
12. When prompted to **login as**, enter: `ec2-user`
    This step will connect you to the EC2 instance.
13. Microsoft Windows users: Click here to skip ahead to the next task.



macOS and Linux Users - Using SSH to Connect

These instructions are only for macOS or Linux users.

13. Read through the three bullet points in this step before you start to complete the actions. You will not be able see these instructions when the **Details** panel is open.
    - Above these instructions you are currently reading, choose the Details dropdown menu, and then choose Show
14. A **Credentials** window will open.
    - Choose the **Download PEM** button and save the **labsuser.pem** file. Typically, your browser will save it to the **Downloads** directory.
    - Then, exit the Details panel by choosing the **X**.
15. Open a terminal window, and change directory (cd) to the directory where the .pem file was downloaded.
    For example, if it was saved to your **Downloads** directory, run this command:
16. `cd ~/Downloads`
17. Change the permissions on the key to be *read only* by running this command:
18. `chmod 400 <filename>.pem`
19. Return to the AWS Management Console, and in the EC2 service, choose **Instances**.
    Check the box next to the instance you want to connect to.
20. In the **Description** tab, copy the **IPv4 Public IP** value.
21. Return to the terminal window and run this command (replace ***<public-ip-address>*** with the actual public IP address that you copied):
22. `ssh -i <filename>.pem ec2-user@<public-ip-address>`
23. When you are prompted to allow a first connection to this remote SSH server, enter yes.
    Because you are using a key pair for authentication, you will not be prompted for a password.