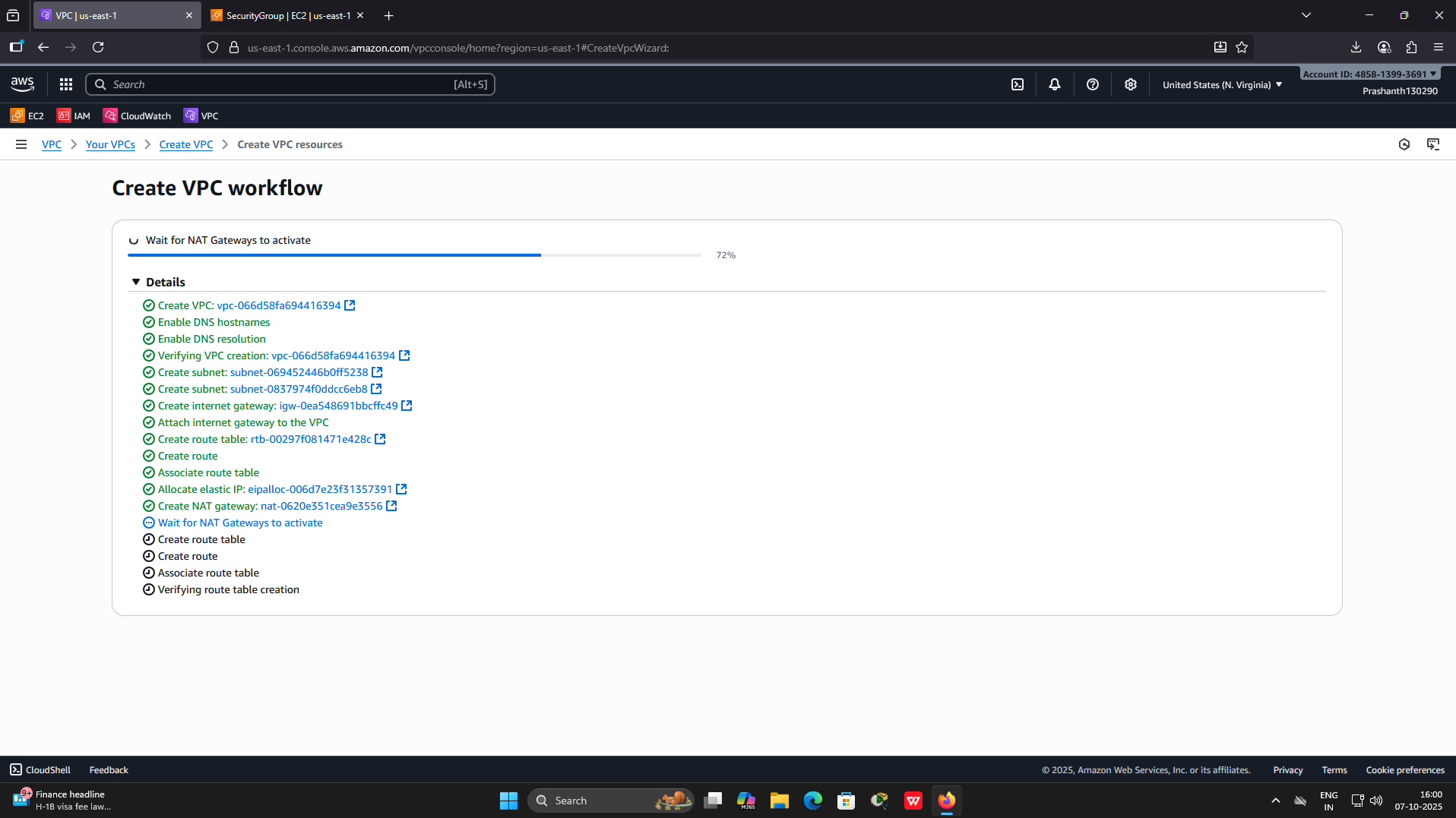
Create VPC

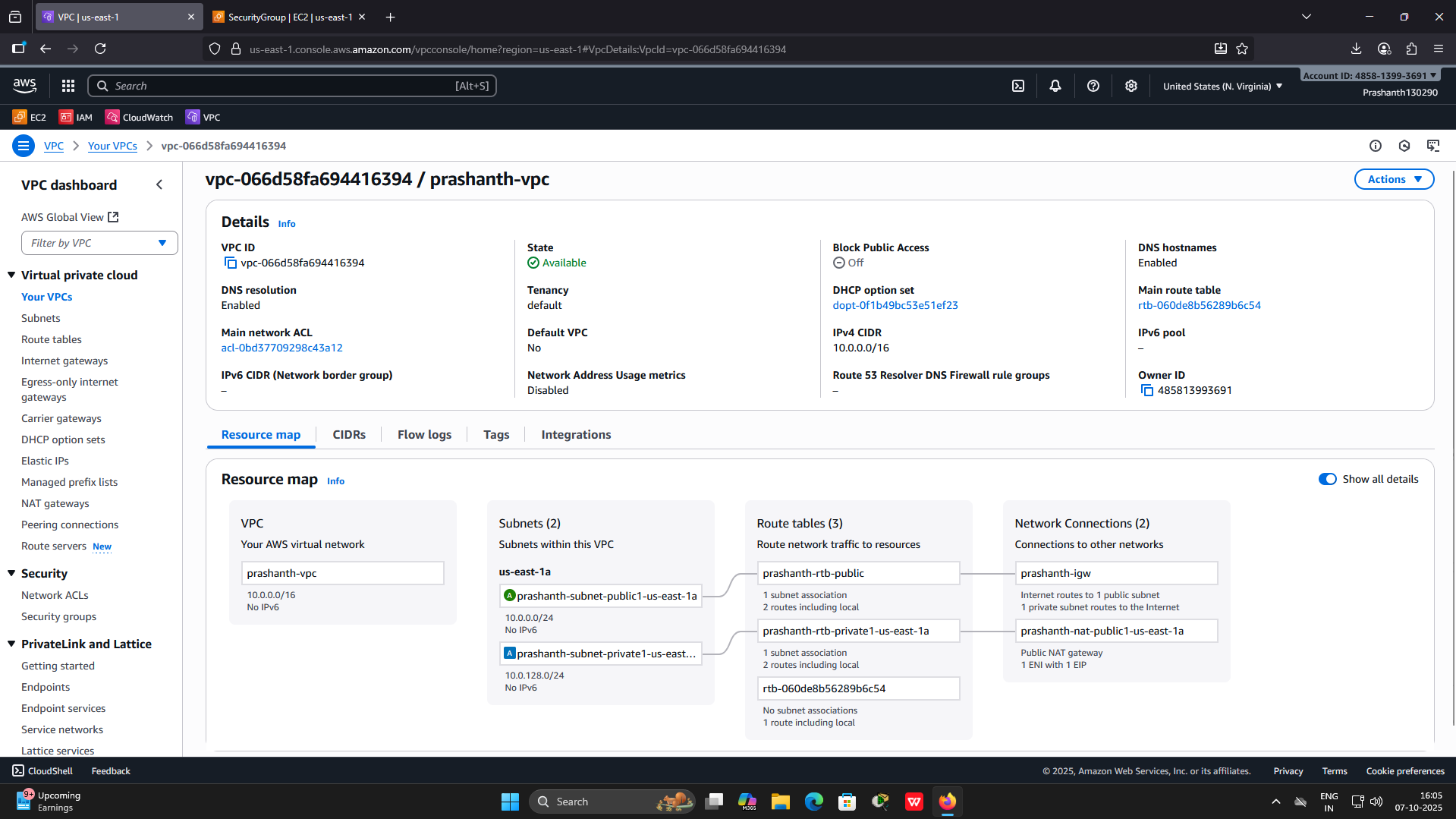
1. Launch VPC console and start creating VPC
2. choose Create VPC. Configure the VPC details in the VPC settings panel
   1. Under Name tag auto-generation, keep Auto-generate selected, however change the value from project to yourname.
   2. Keep the IPv4 CIDR block set to 10.0.0.0/16
   3. For Number of Availability Zones, choose 1.
   4. For Number of public subnets, keep the 1 setting.
   5. For Number of private subnets, keep the 1 setting.
   6. Expand the Customize subnets CIDR blocks section
   7. Change Public subnet CIDR block in us-east-1a to 10.0.0.0/24
   8. Change Private subnet CIDR block in us-east-1a to 10.0.1.0/24
   9. Set NAT gateways to In 1 AZ.
   10. Set VPC endpoints to None.
   11. Keep both DNS hostnames and DNS resolution enabled.
3. In the Preview panel on the right, confirm the settings you have configured.
   1. VPC: yourname-vpc
   2. Subnets: us-east-1a
   3. Public subnet name: lab-subnet-public1-us-east-1a
   4. Private subnet name: lab-subnet-private1-us-east-1a
   5. Route tables: lab-rtb-public, lab-rtb-private1-us-east-1a
   6. Network connections : lab-igw, lab-nat-public1-us-east-1a
4. Choose Create VPC and wait till all resources are created



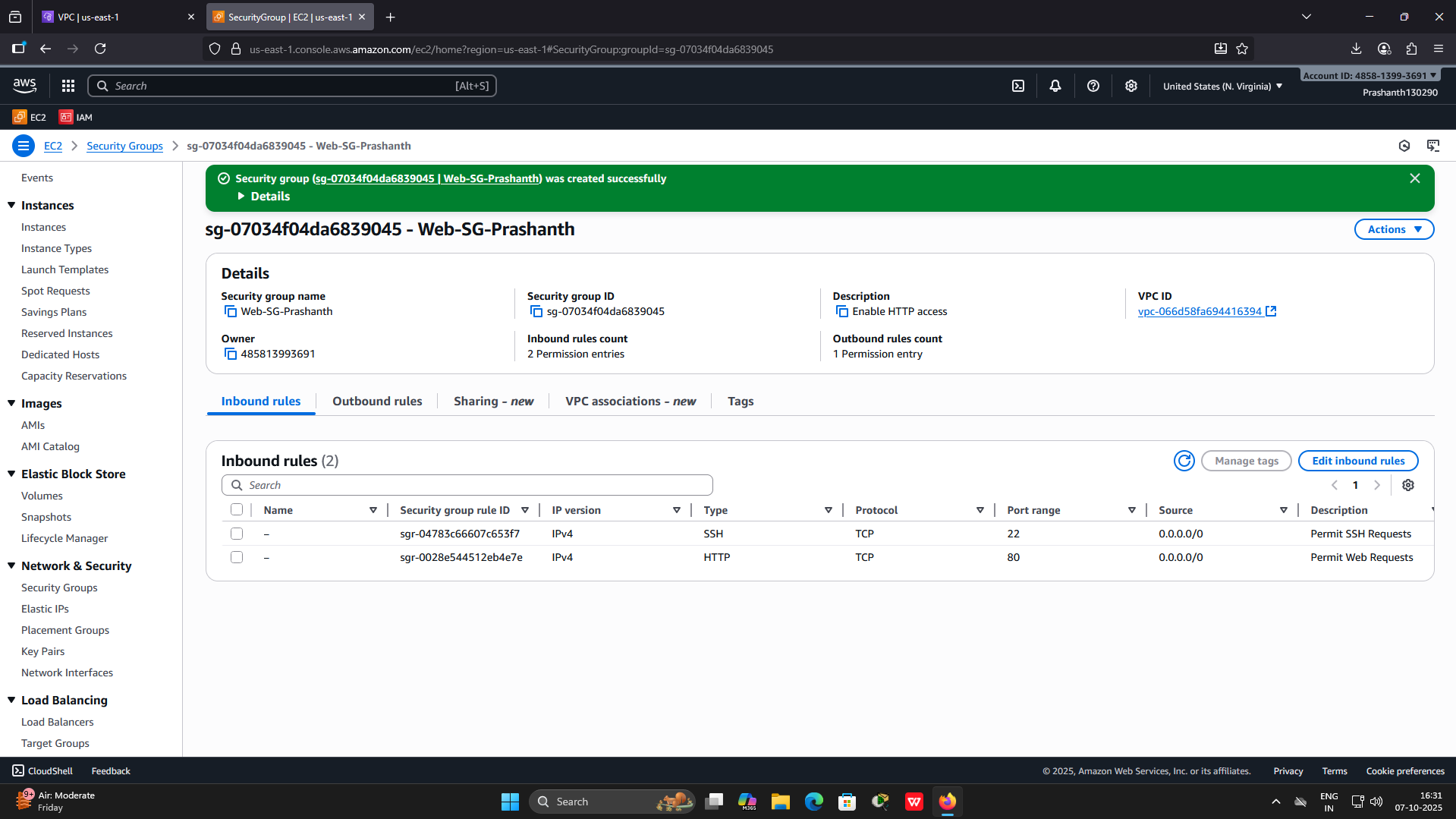
1. Choose View VPC

The wizard has provisioned a VPC with a public subnet and a private subnet in one Availability Zone with route tables for each subnet. It also created an Internet Gateway and a NAT Gateway.

Explore all settings.



1. Create a Security Group. Launch EC2 console . In the left navigation pane, choose Security groups.
2. Choose Create security group and then configure:
   1. Security group name: Web-SG-yourname
   2. Description: Enable HTTP access
   3. VPC: choose the X to remove the currently selected VPC, then from the drop down list choose the VPC you created
3. In the Inbound rules pane, choose Add rule:
   1. Type: HTTP
   2. Source: Anywhere-IPv4
   3. Description: Permit web requests
   4. Type: SSH
   5. Source: Anywhere IPv4
   6. Description : Permits SSH requests



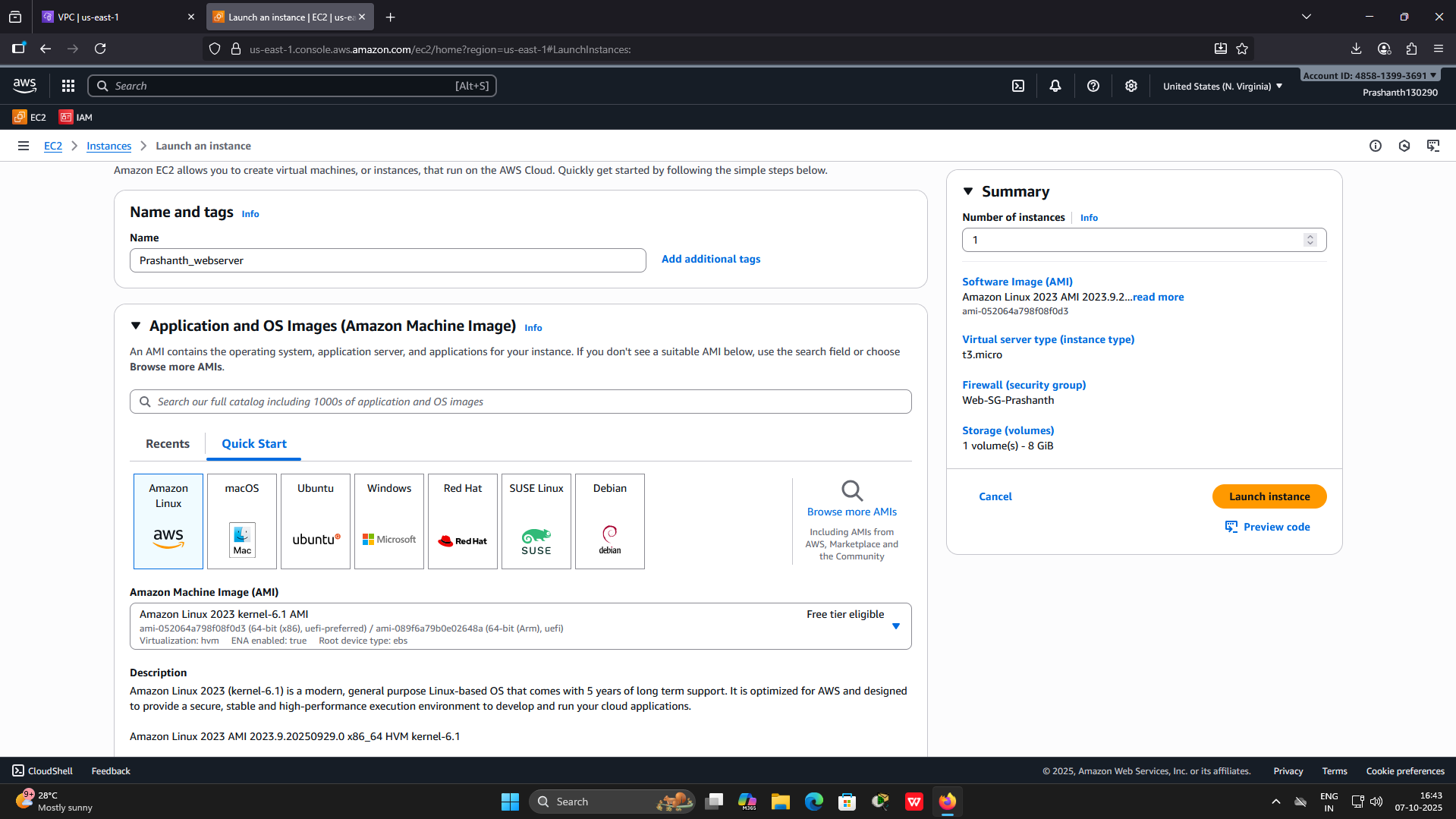
1. Scroll to the bottom of the page and choose Create security group

You will use this security group in the next task when launching an Amazon EC2 instance

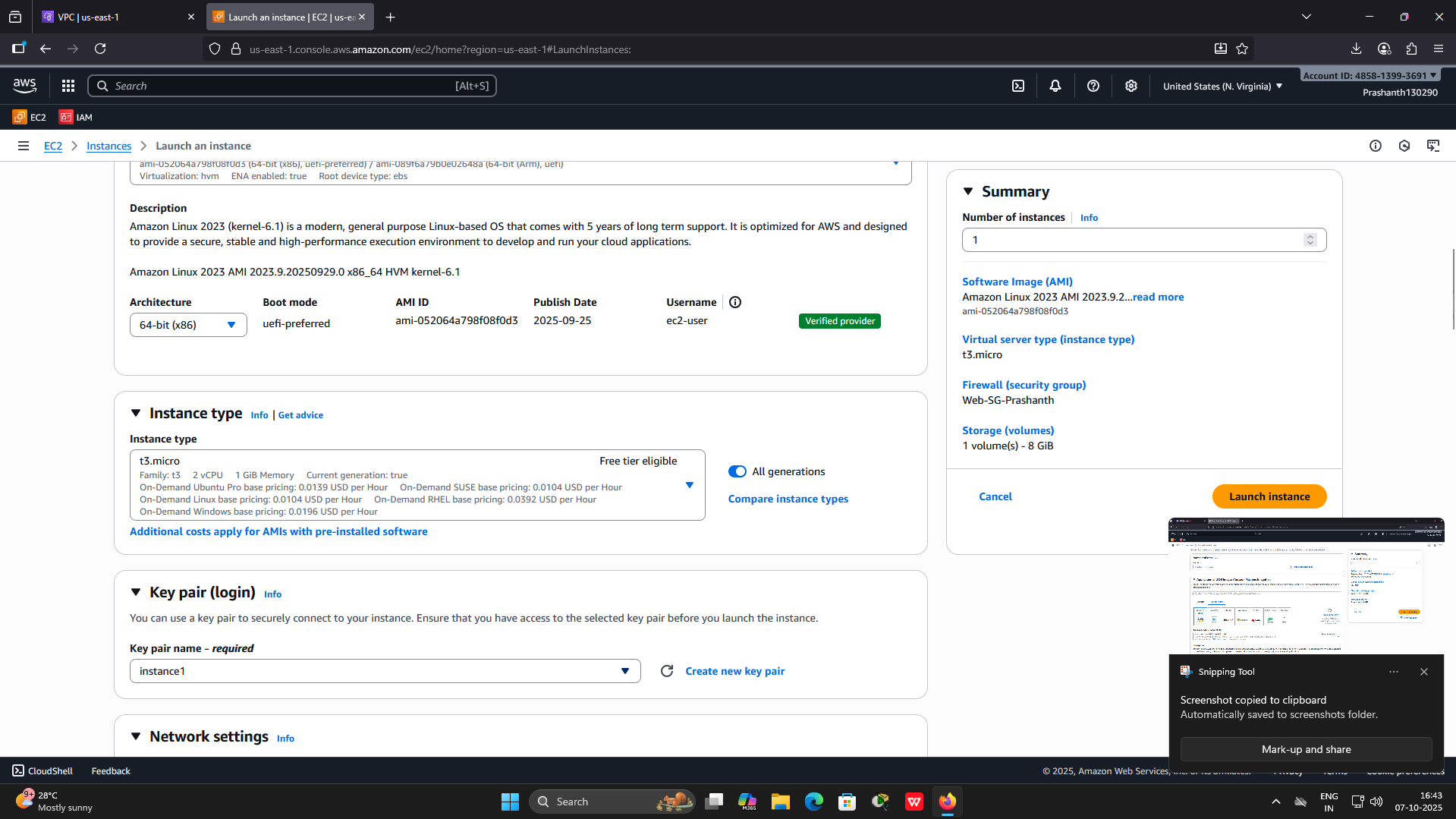
1. Create an EC2 instance which will be the Web Server. Launch EC2 console. Choose Launch instance.
2. Name the instance : yourname\_webserver
3. Choose an AMI from which to create the instance:

In the list of available Quick Start AMIs, keep the default Amazon Linux AMI selected. Also keep the default Amazon Linux 2 AMI (HVM) selected.

Choose an Instance type: t2.micro



1. Choose keypair : Use keypair created during demo
2. Configure the Network settings:
   1. Next to Network settings, choose Edit, then configure:
   2. Network: Select the VPC created earlier
   3. Subnet: lab-subnet-public2 (not Private!)
   4. Auto-assign public IP: Enable
   5. Next, you will configure the instance to use the Security Group created earlier.
   6. Under Firewall (security groups), choose Select an existing security group & select your security group
3. In the Configure storage section, keep the default settings.



1. Configure a script to run on the instance when it launches:
   1. Expand the Advanced details panel. Scroll to the bottom of the page and add the following script is user data box.

#!/bin/bash

yum update -y

sudo su

amazon-linux-extras install -y nginx1

systemctl start nginx

systemctl enable nginx

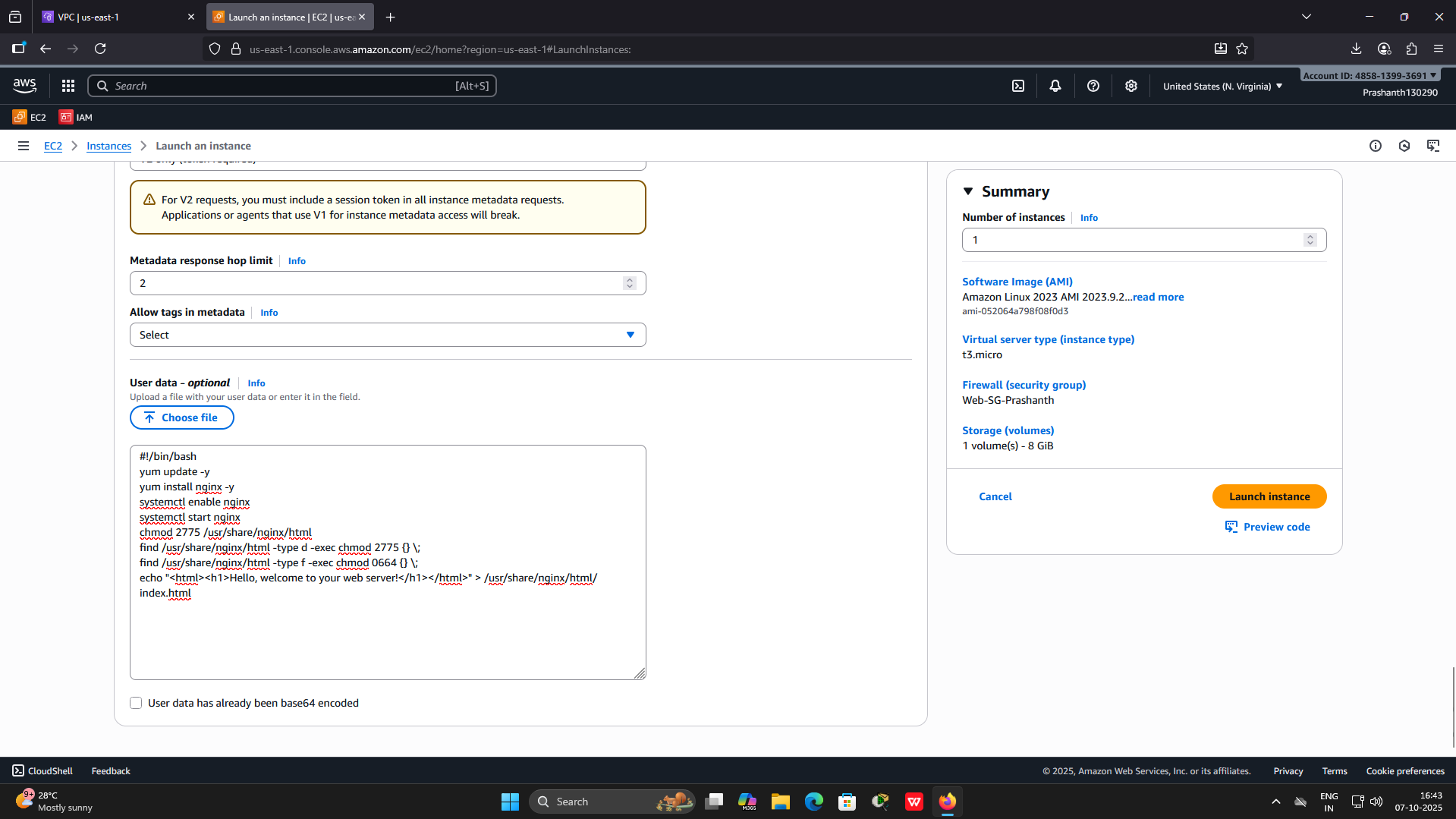
chmod 2775 /usr/share/nginx/html

find /usr/share/nginx/html -type d -exec chmod 2775 {} \;

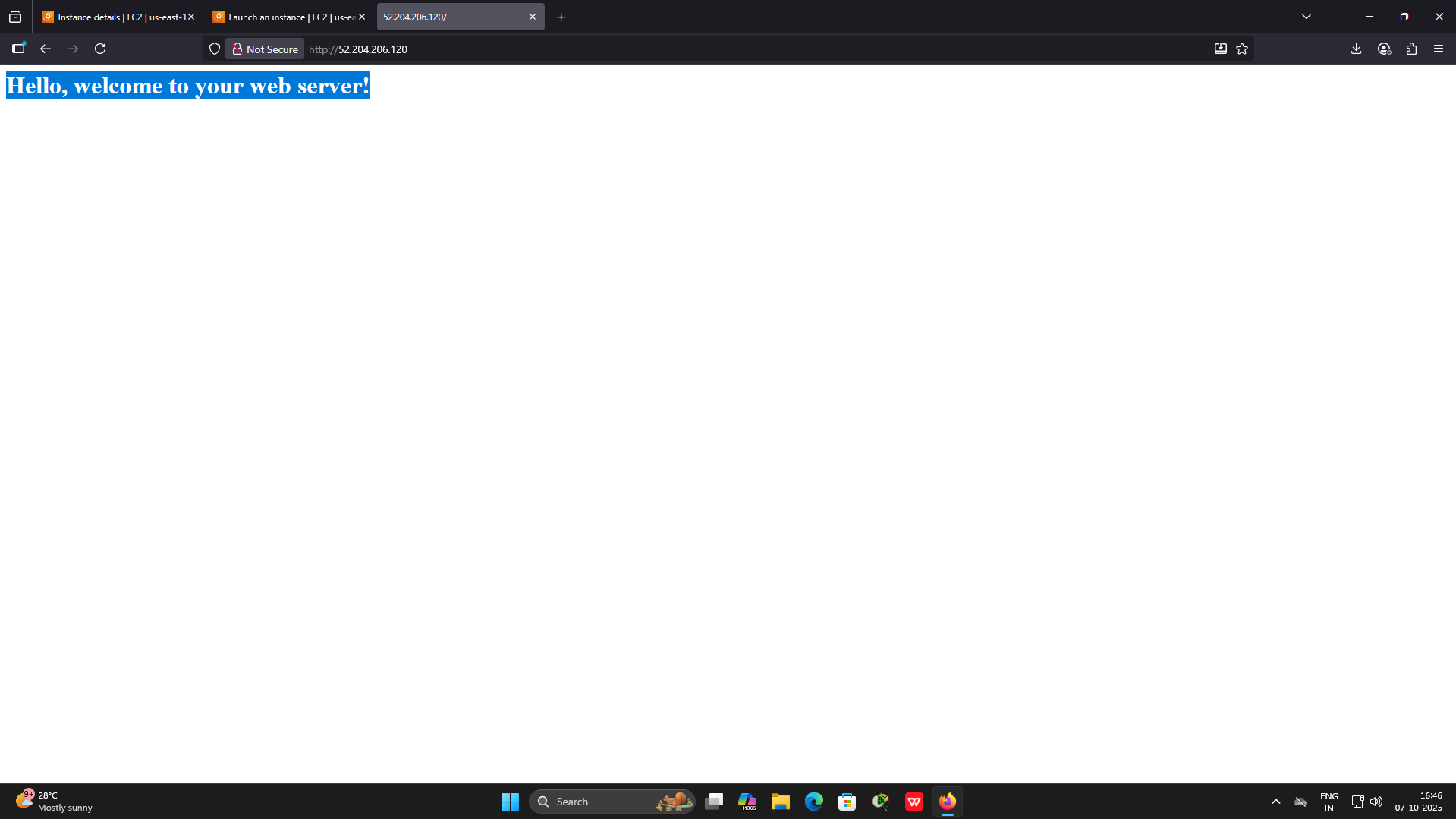
find /usr/share/nginx/html -type f -exec chmod 0664 {} \;

echo "<h1>It worked from CDK code of Anu</h1>" > /usr/share/nginx/html/index.html

1. Once the instance is launched, and status is running. Select the instance and explore the values/settings



1. Copy the Public IPv4 DNS value shown in the Details tab at the bottom of the page. Open a web browser tab, paste the Public DNS value and Check the output
2. Terminate the instance and delete VPC created



**Assignmnet** 2 - Lab 2 – EC2 Instance launching, resizing, managing, and monitoring

1. Launch EC2 console
   1. Give the instance the name : yourname-webserver
   2. keep the default Amazon Linux AMI selected
   3. In the Instance type panel, keep the default t2.micro selected
   4. In the Configure storage section, keep the default settings.
   5. Expand Advanced Details
   6. For Termination Protection, select Enable
   7. Scroll to User Data box and put the following script

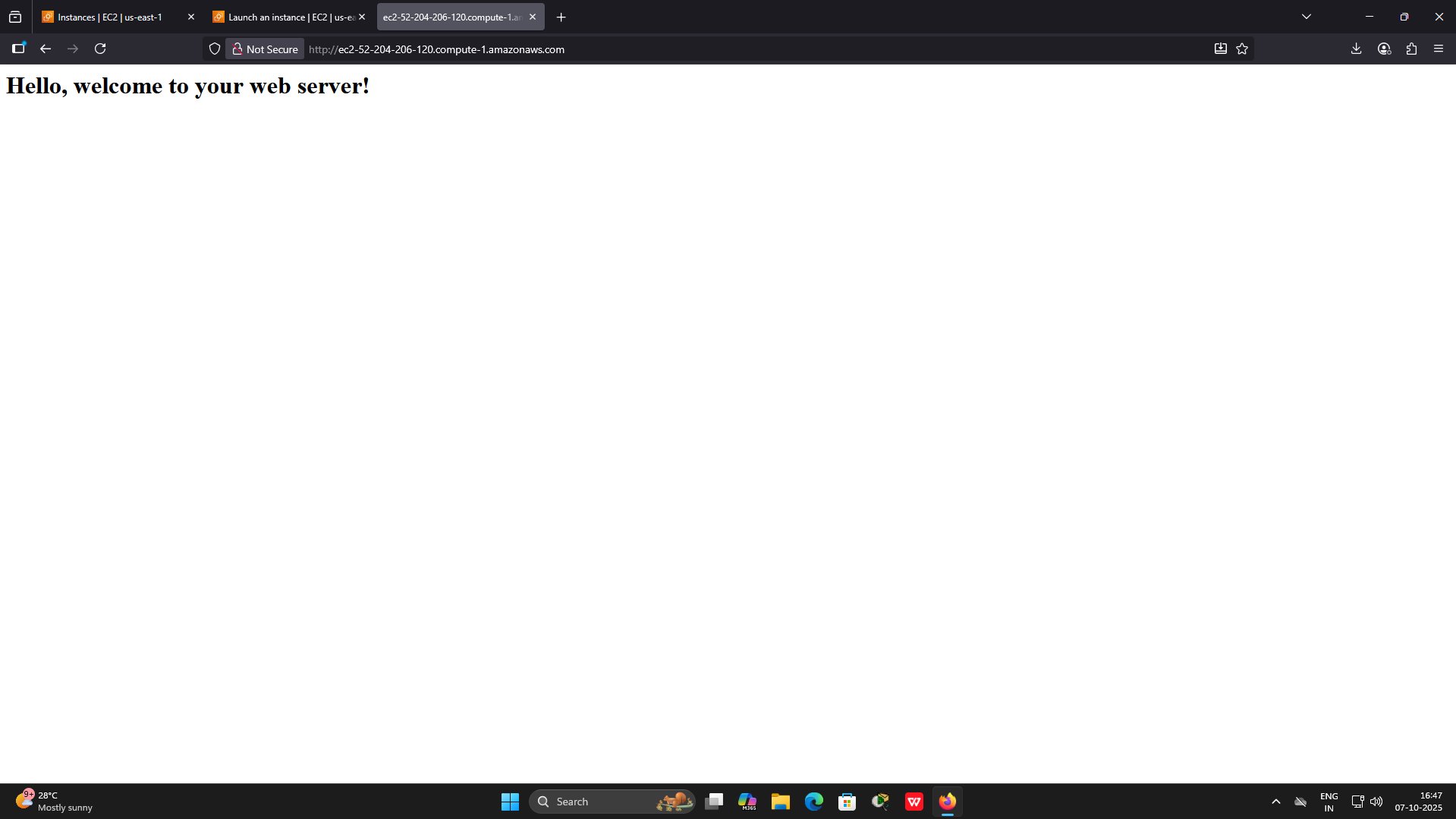
#!/bin/bash

yum -y install httpd

systemctl enable httpd

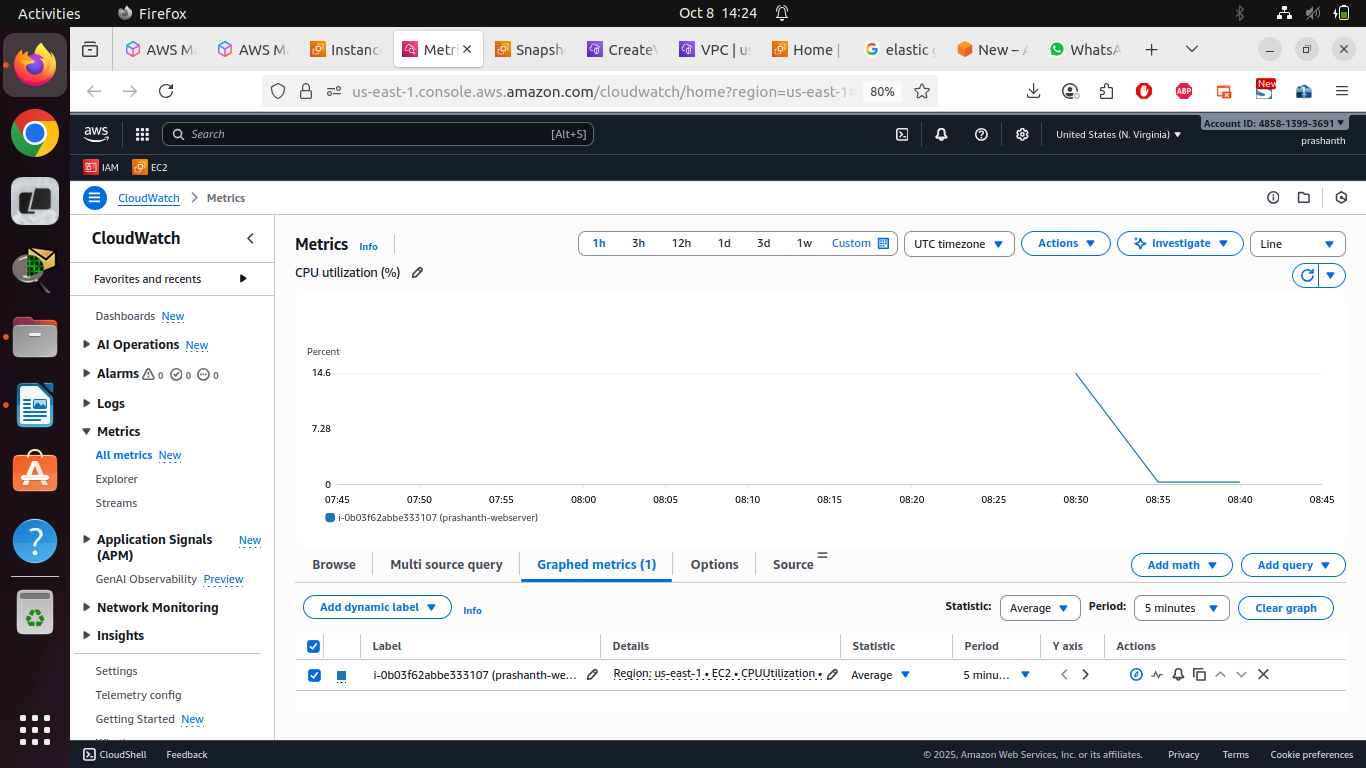
systemctl start httpd

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



1. Once the instance is launched and running, check out the details
2. Monitor the instance
   1. Notice that both the System reachability and Instance reachability checks have passed.
   2. Choose the Monitoring tab.

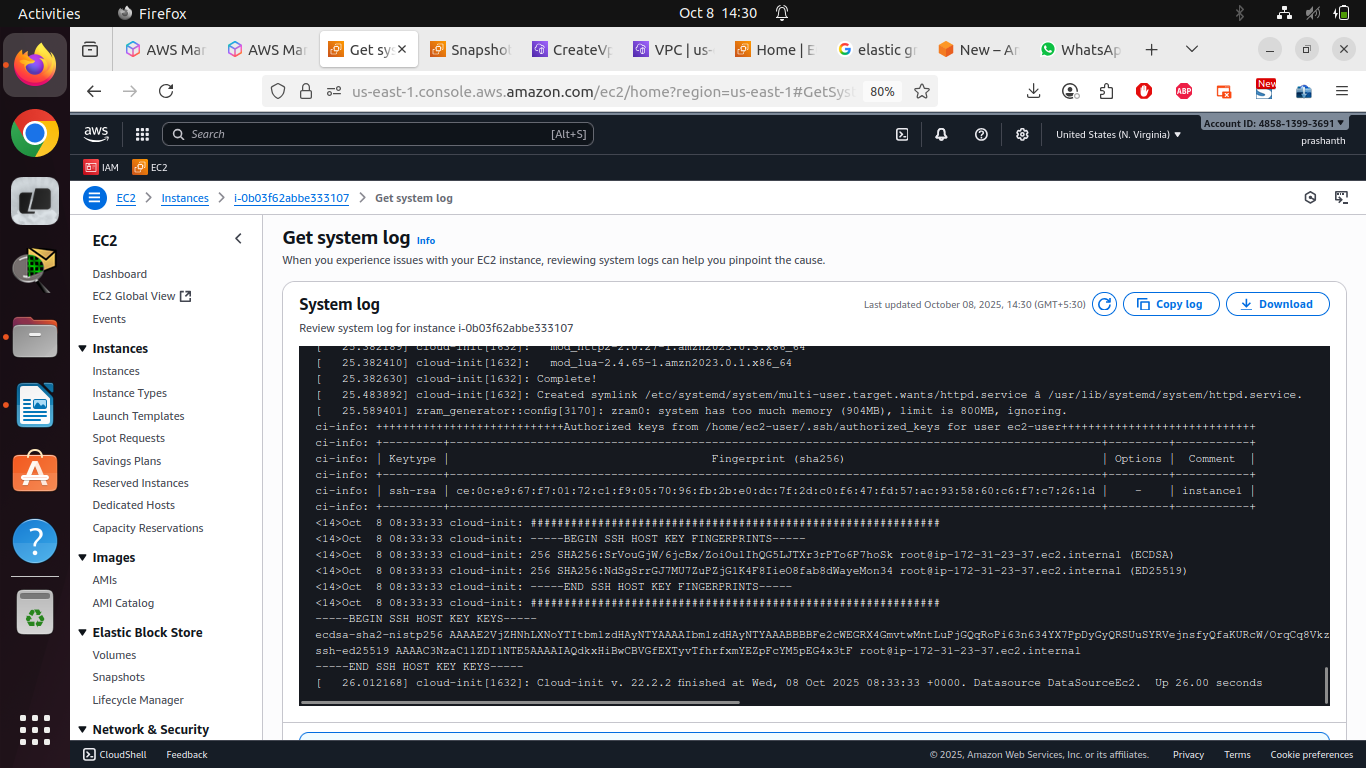
This tab displays Amazon CloudWatch metrics for your instance. Currently, there are not many metrics to display because the instance was recently launched. You can choose the three dots icon in any graph and select Enlarge to see an expanded view of the chosen metric. Amazon EC2 sends metrics to Amazon CloudWatch for your EC2 instances. Basic (five-minute) monitoring is enabled by default.



* 1. In the Actions menu towards the top of the console, 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. If you do not see a system log, wait a few minutes and then try again.

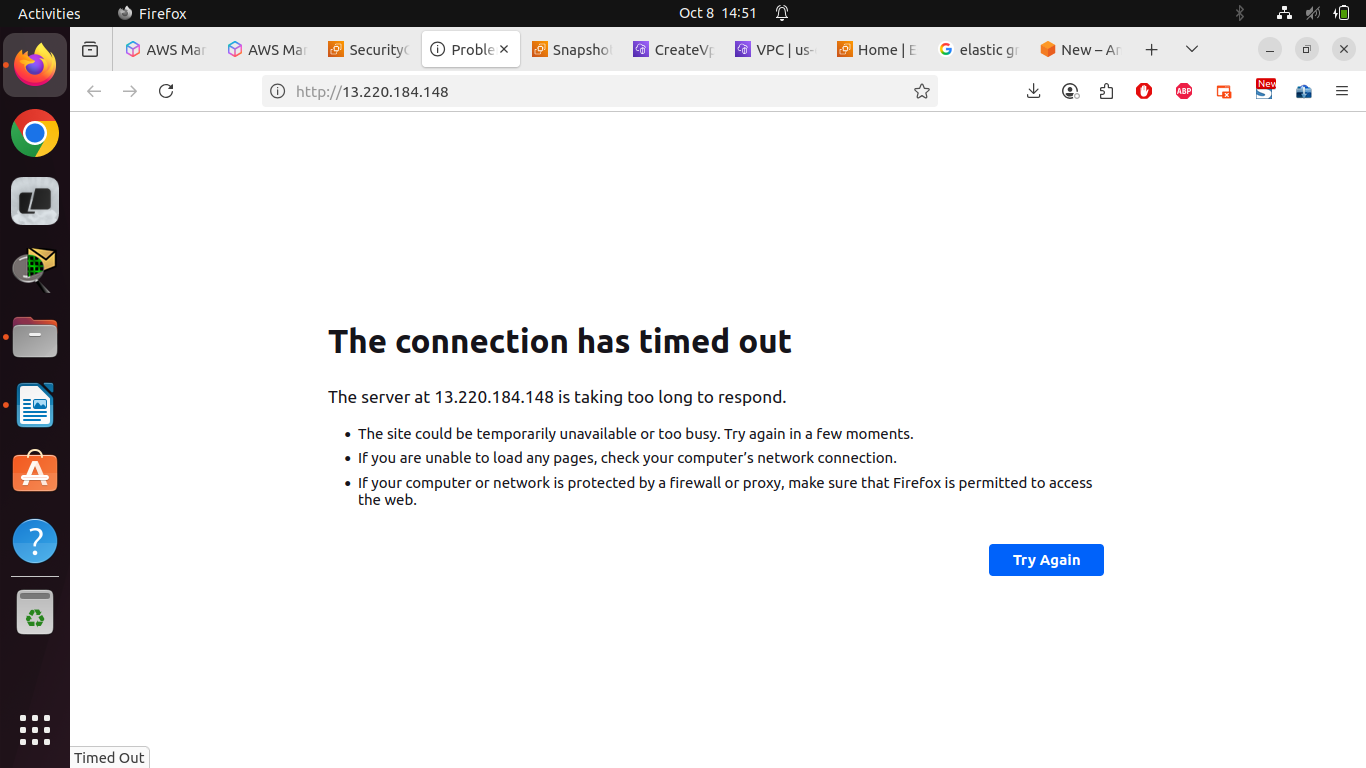
Scroll through the output and note that the HTTP package was installed from the user data that you added when you created the instance.

* 1. Choose Cancel



1. Update Security Group and check access to the instance
   1. From the left side pane select Security Group.
   2. Select your instance Security Group.
   3. Edit Inbound Rules, delete the HTTP rule. Save the changes
   4. Copy the Public IPv4 address of your instance to your clipboard.
   5. 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.

* 1. In the left navigation pane, choose Security Groups.
  2. Select Web Server security group. Choose the Inbound rules tab.
  3. Choose Edit inbound rules, select Add rule and then configure:

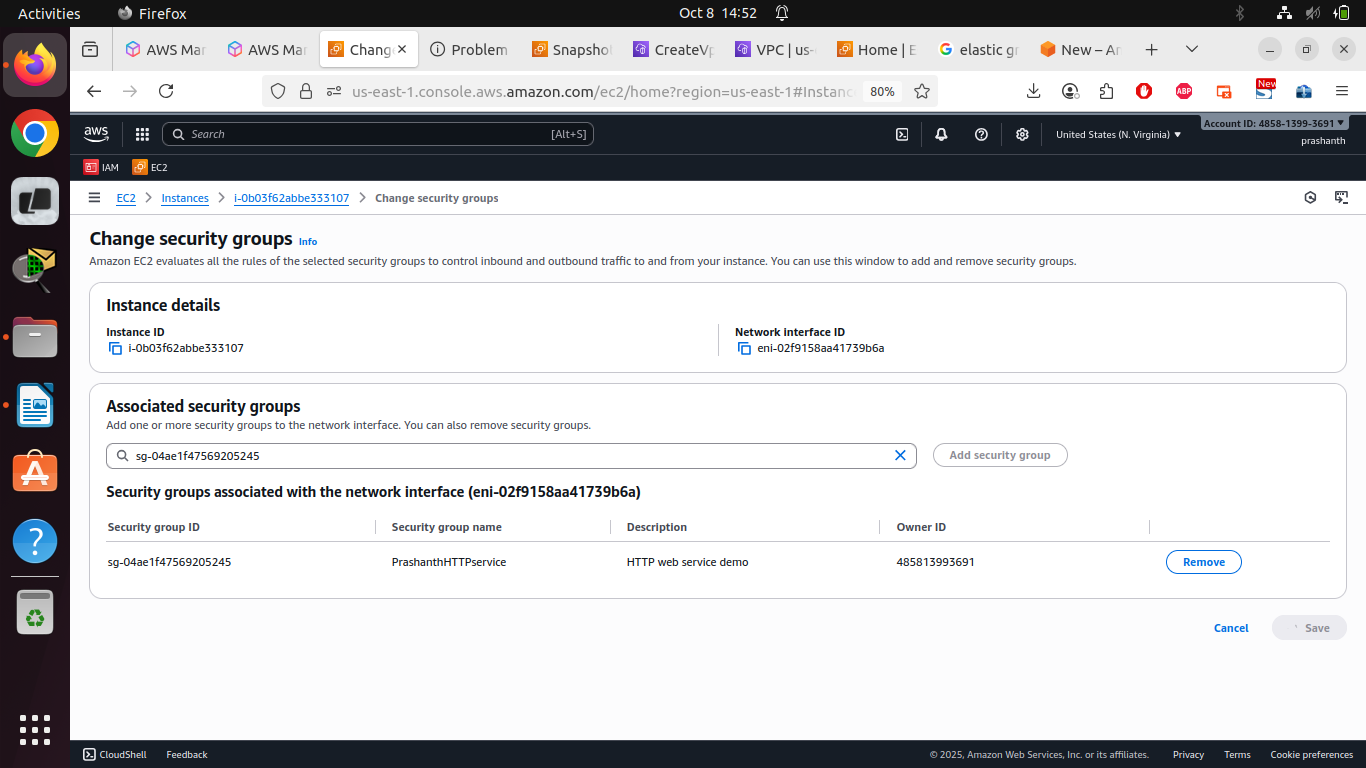
o Type: HTTP

o Source: Anywhere-IPv4

o Choose Save rules

* 1. Access the Public IPv4 address once again.

You should see the message Hello From Web Server!



1. Resize the instance : Instance Type and EBS volume
   1. Before resizing Stop the instance
   2. On the EC2 Management Console, in the left navigation pane, choose Instances.
   3. Choose your instance . In the Instance State menu, select Stop instance.Choose Stop
   4. Your instance will perform a normal shutdown and then will stop running Wait for the Instance State to display: Stopped
   5. Change the Instance Type

In the Actions menu, select Instance settings Change instance type, then configure:

Instance Type: t2.small

Choose Apply

* 1. Resize EBS volume

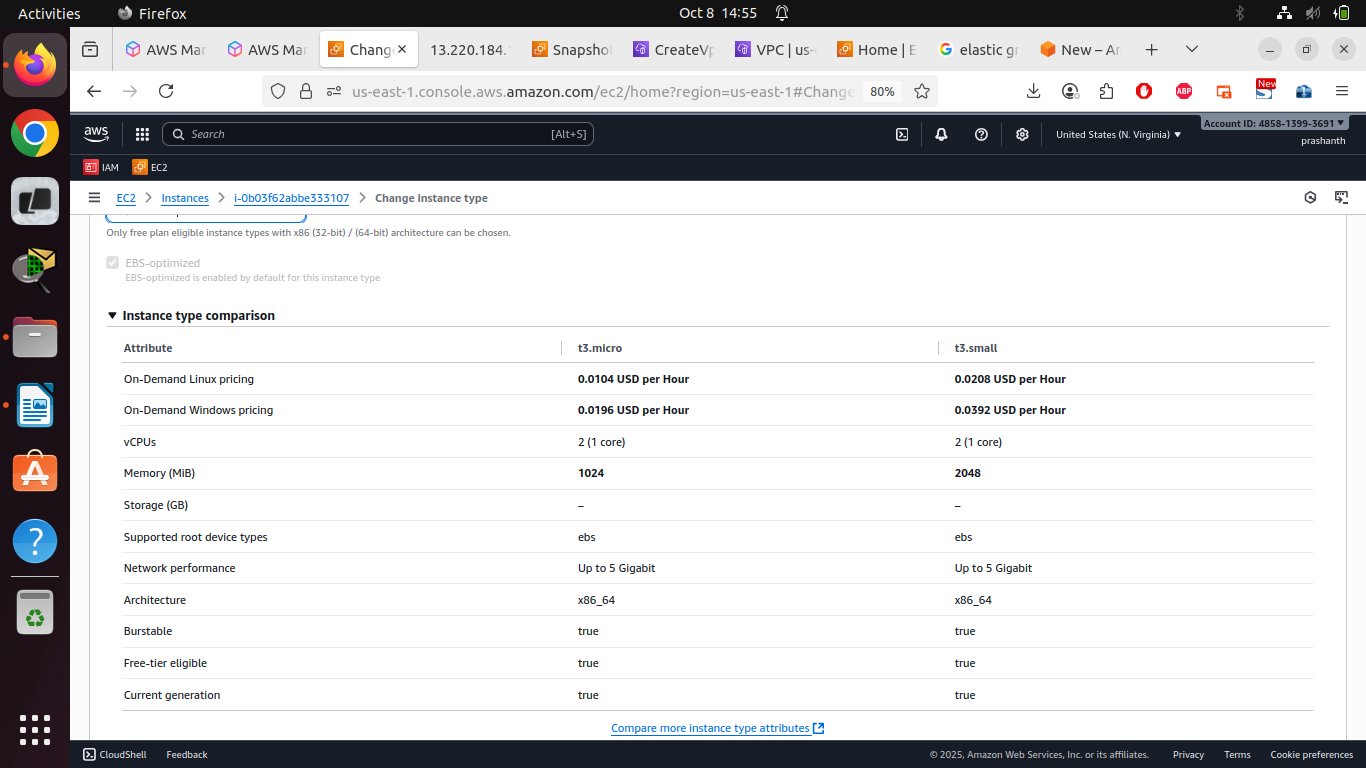
Choose the Storage tab, select the name of the Volume ID, then select the checkbox next to the volume that displays.In the Actions menu, select Modify volume.

Change the size to: 10

Choose Modify

Choose Modify again to confirm and increase the size of the volume.

* 1. In the Instance state menu, select Start Instance



1. Test Termination Protection
   1. In left navigation pane, choose Instances.

Select your instance and in the Instance state menu, select Terminate instance.

Then choose Terminate

Note the error message

* 1. In the Actions menu, select Instance Settings Change termination protection.

Remove the check next to Enable.

Choose Save

You can now terminate the instance.

Select your instance again and in the Instance state menu, select Terminate instance.

Choose Terminate. Now the instance should get terminated

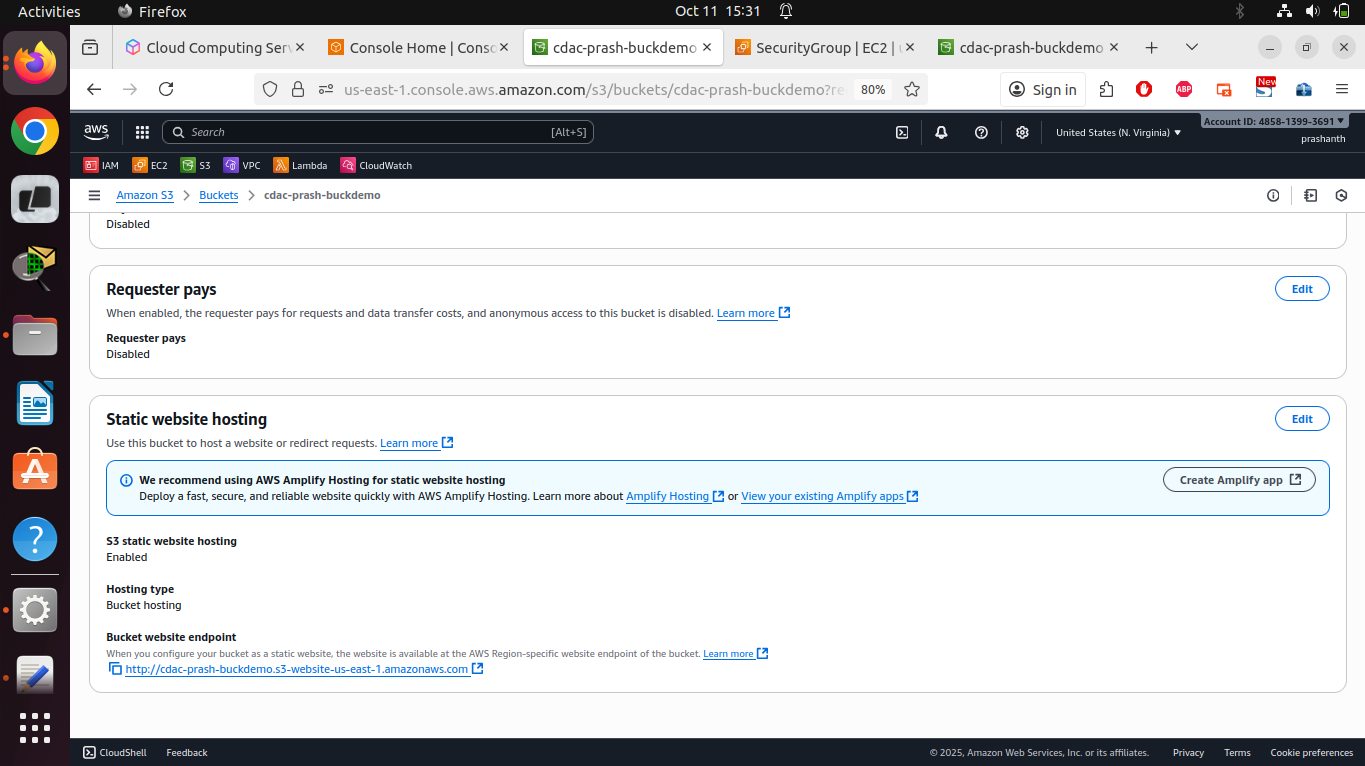
Labs on S3 Bucket

Lab 1: Creating and Configuring an S3 Bucket

1. Create a bucket
   * Launch the S3 service
   * Click "Create bucket"
   * Enter a globally unique name
   * Select region – us-east
   * Keep "Block all public access" enabled (we'll modify this later)
   * Keep the remaining default settings
   * Click "Create bucket"



1. Configure bucket properties
   * Select your newly created bucket
   * Go to the "Properties" tab
   * Enable "Static website hosting"
   * Select "Host a static website" and enter index.html as the index document
   * Save the settings



1. Upload files
   * Create a simple index.html file locally with the following content:

<!DOCTYPE html>

<html>

<head>

<title>My S3 Exercise</title>

</head>

<body>

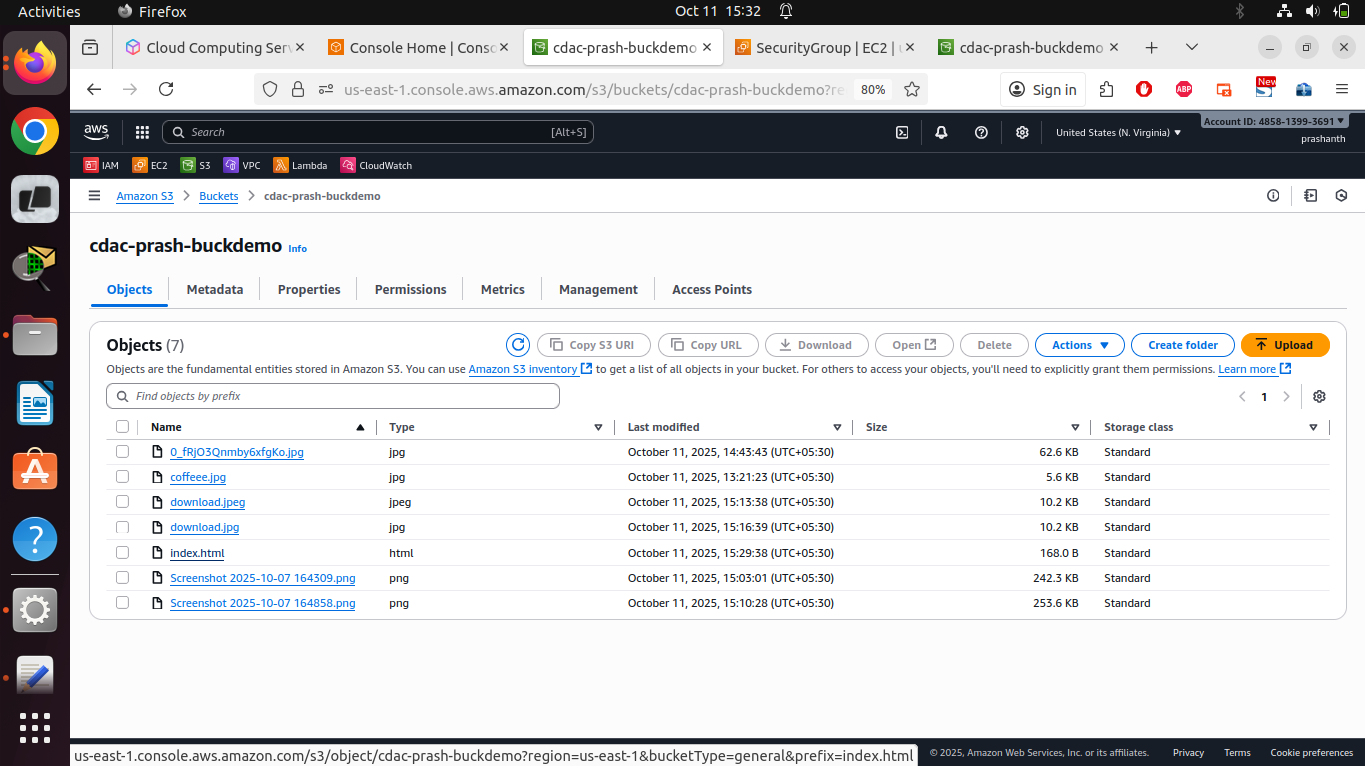
<h1>Welcome to my S3 Exercise!</h1>

<p>Date: [Current Date]</p>

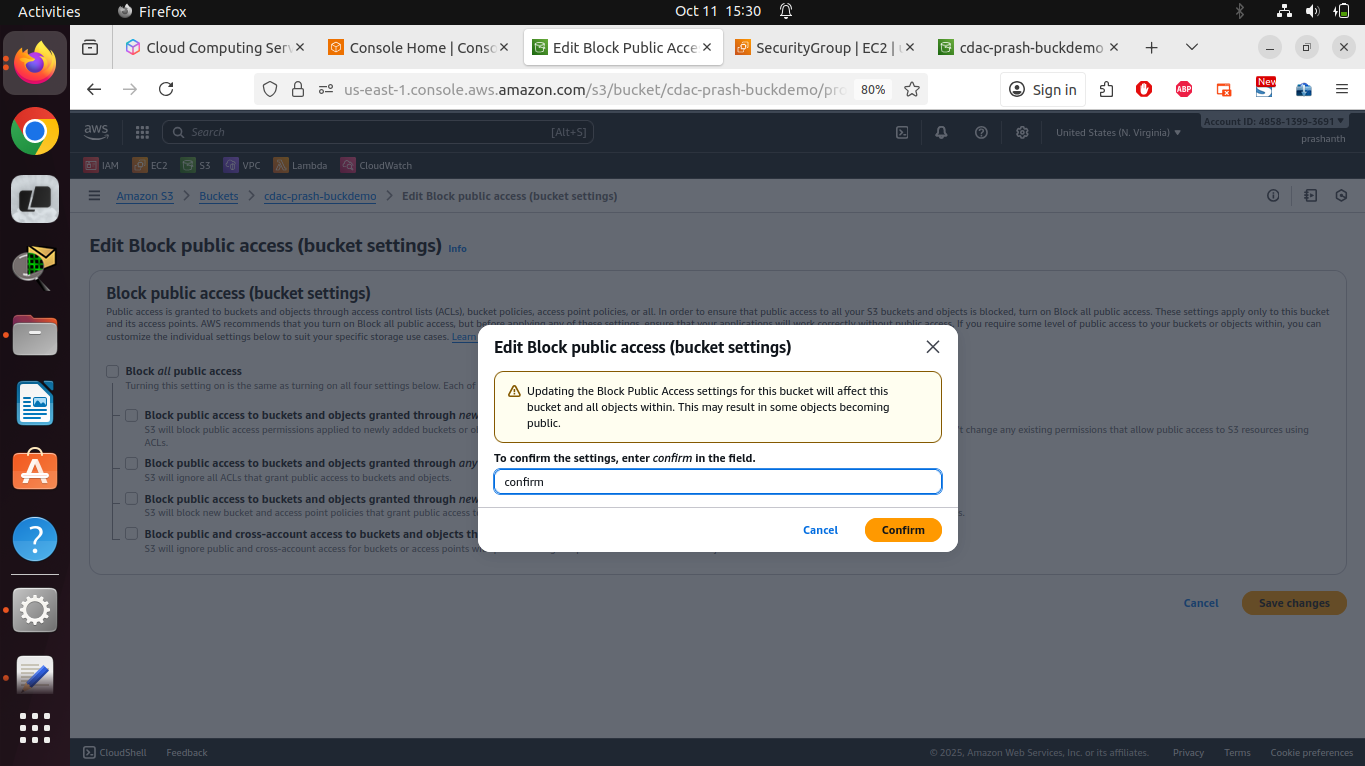
</body>

</html>

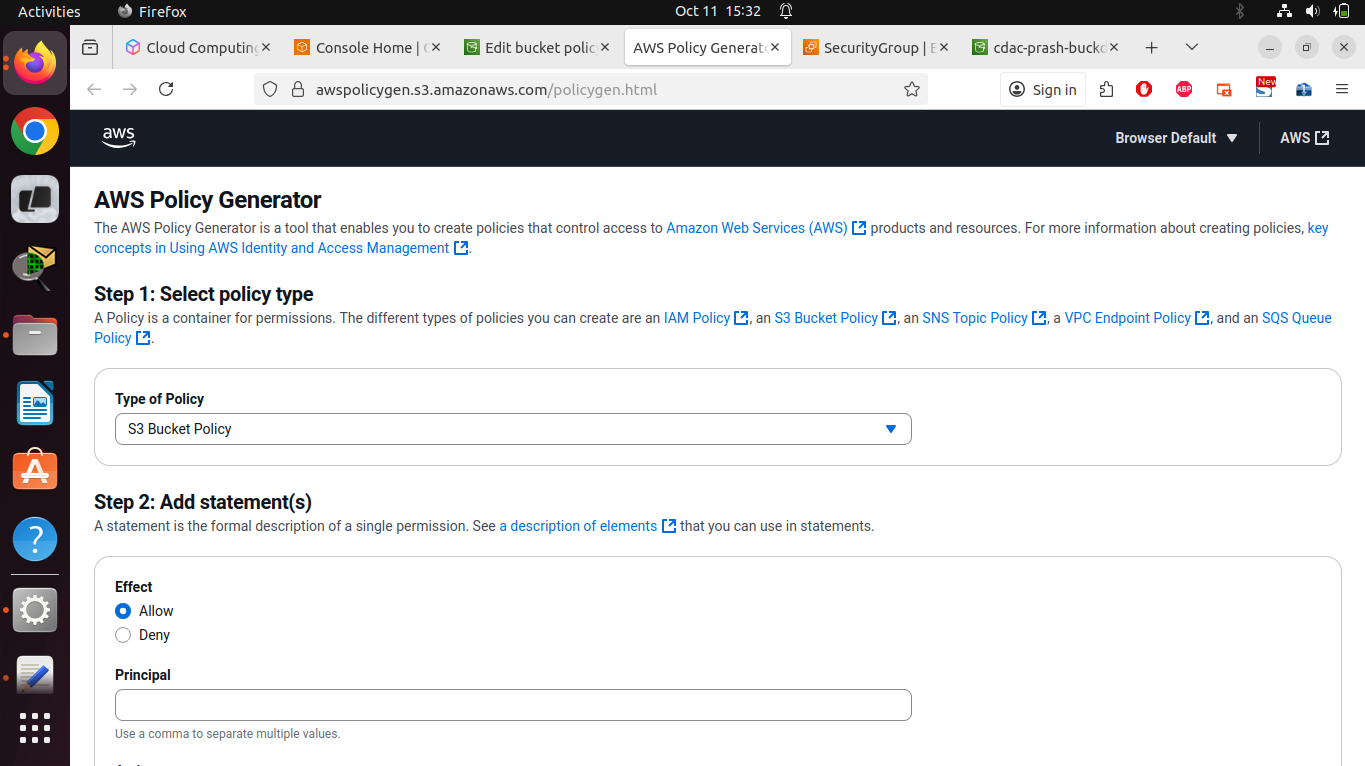
* + Go to the "Objects" tab and click "Upload"
  + Upload the index.html file
  + Also upload an image of your choice



1. Adjust bucket policy
   * Disable "Block all public access" in the bucket properties and confirm



* + Go to the "Permissions" tab
  + Click on "Bucket policy"
  + Generate the Bucket policy as follows (replace [YourBucketName] with your actual bucket name):



{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadForGetBucketObjects",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

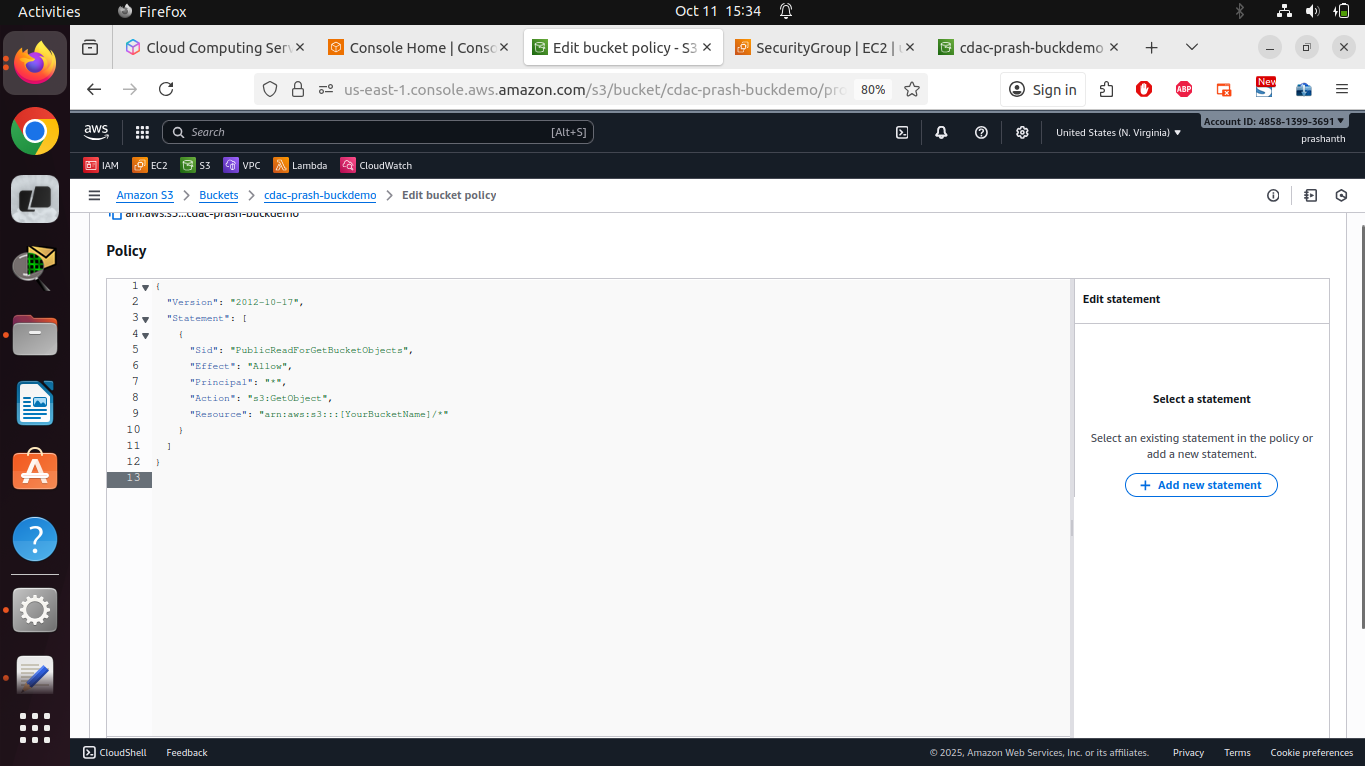
"Resource": "arn:aws:s3:::[YourBucketName]/\*"

}

]

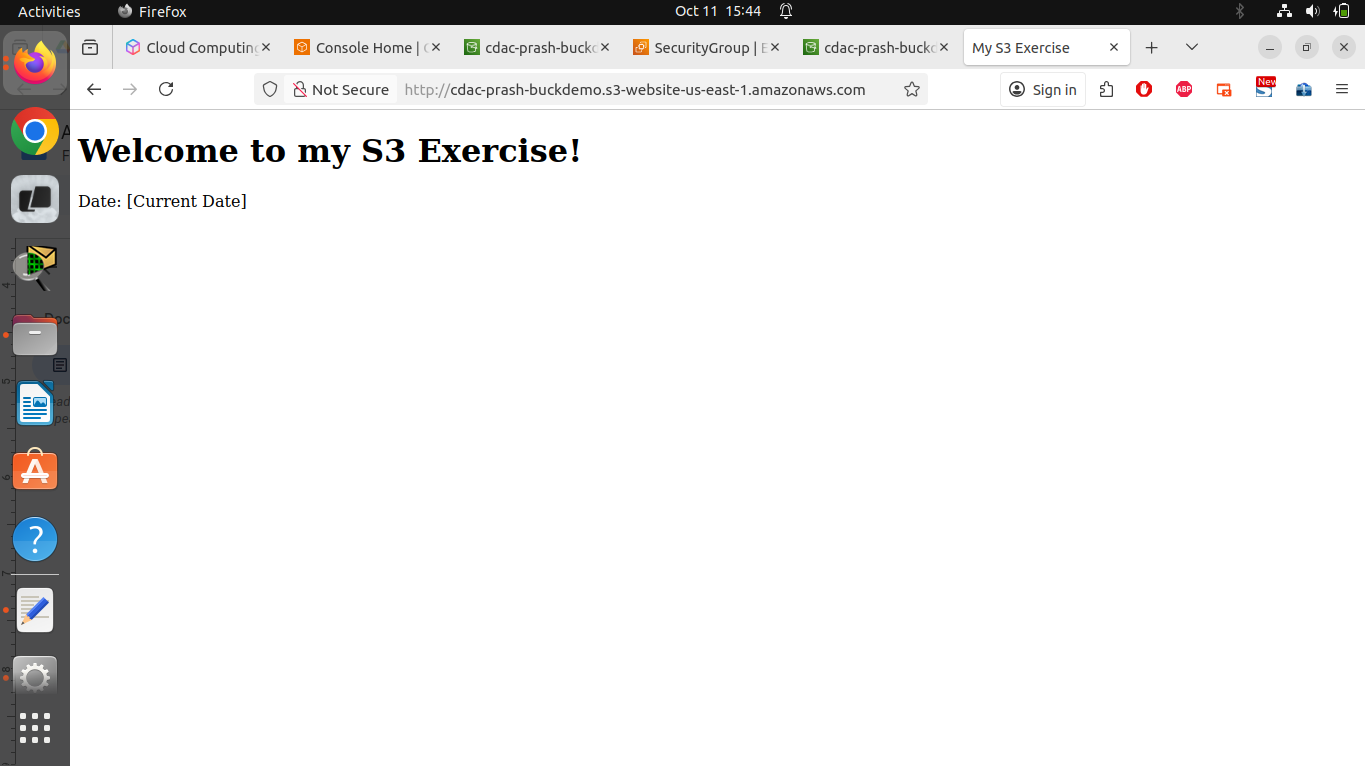
}

* + Save the policy

****

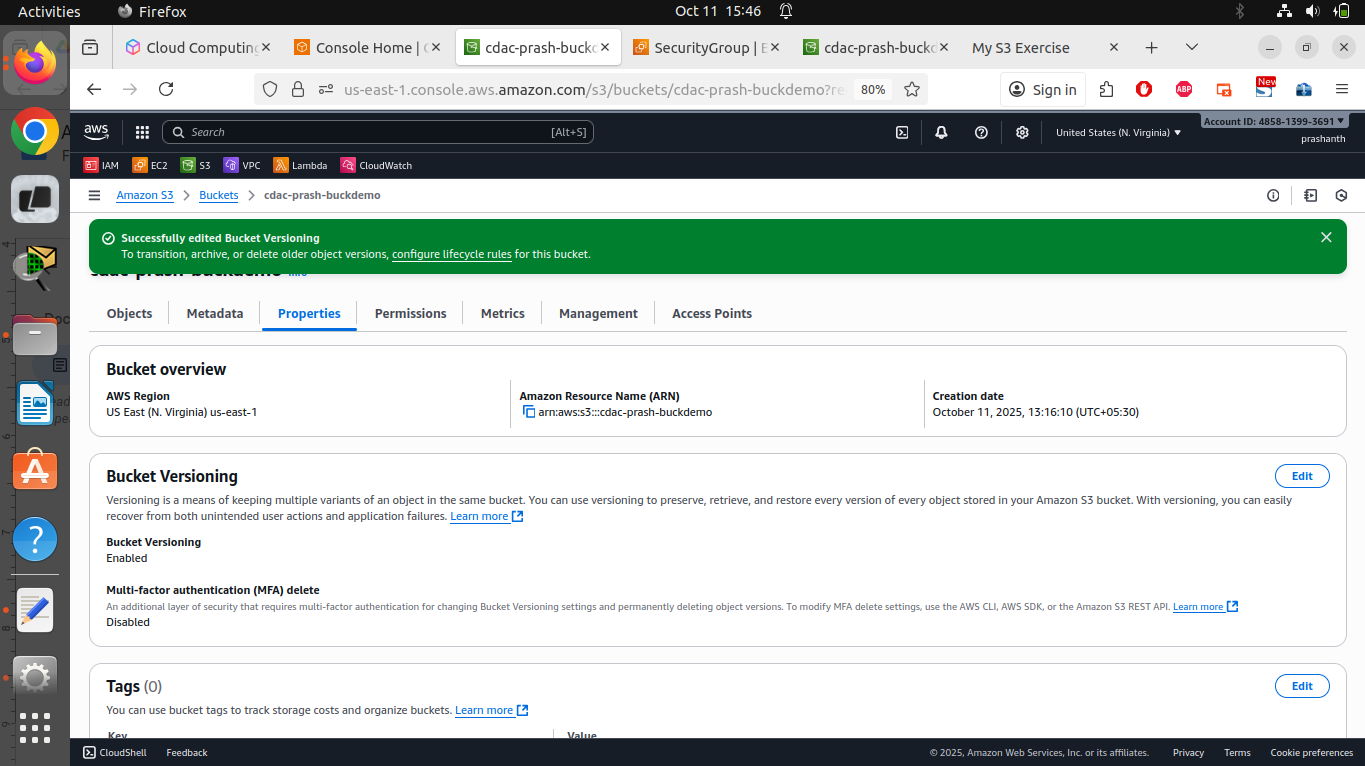
**NOTE**: Making S3 buckets publicly accessible as demonstrated in this exercise is NOT recommended for production environments.

1. Check the website
   * Return to the "Properties" tab
   * Scroll down to "Static website hosting"
   * Open the displayed website URL in a new browser tab
   * Verify that your website is displayed correctly



Lab 2: Versioning, Storage Classes, and Lifecycle Policies

1. Enable versioning
   * Navigate to your S3 bucket
   * Go to the "Properties" tab
   * Scroll to "Bucket Versioning"
   * Click "Edit" and enable versioning
   * Save the changes



1. Create multiple versions
   * Go to the "Objects" tab
   * Open your index.html file in a text editor and modify the content:

<!DOCTYPE html>

<html>

<head>

<title>My S3 Exercise - Version 2</title>

</head>

<body>

<h1>Welcome to my S3 Exercise!</h1>

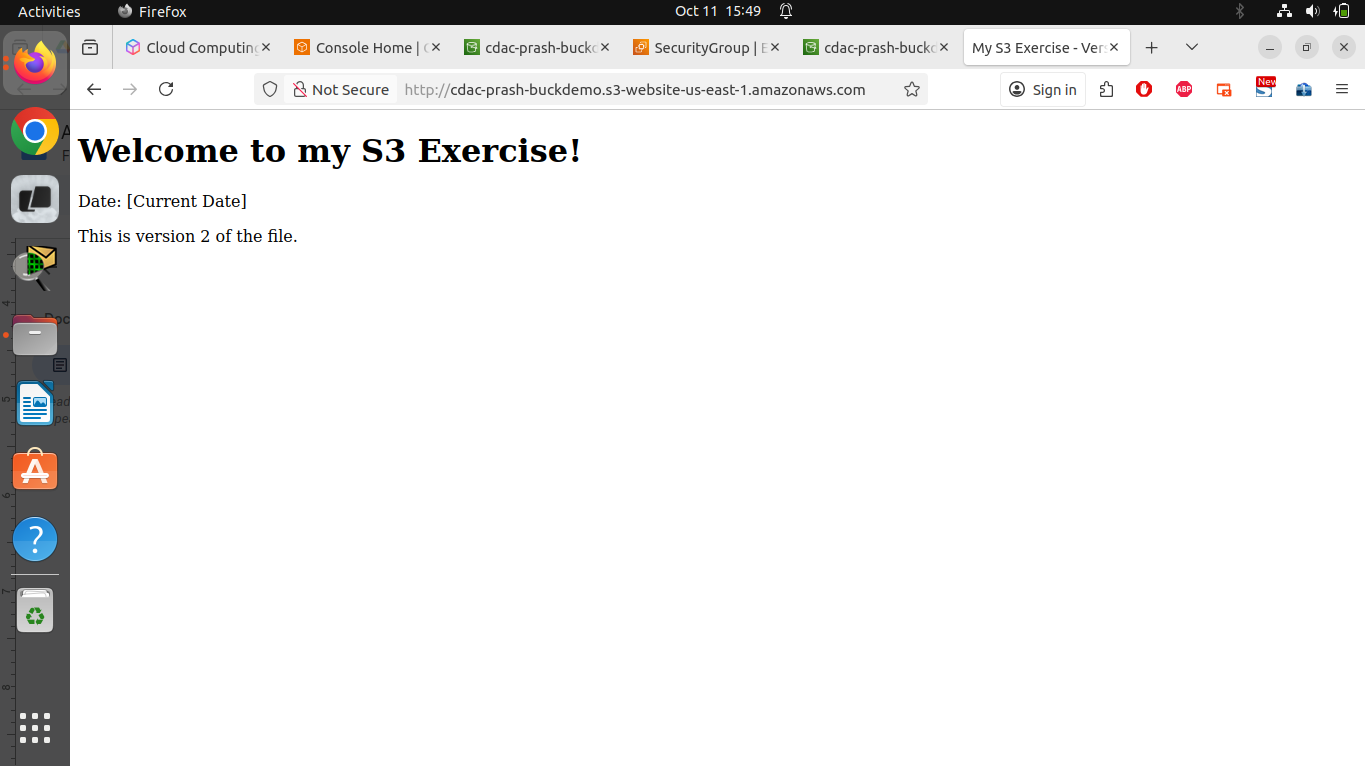
<p>Date: [Current Date]</p>

<p>This is version 2 of the file.</p>

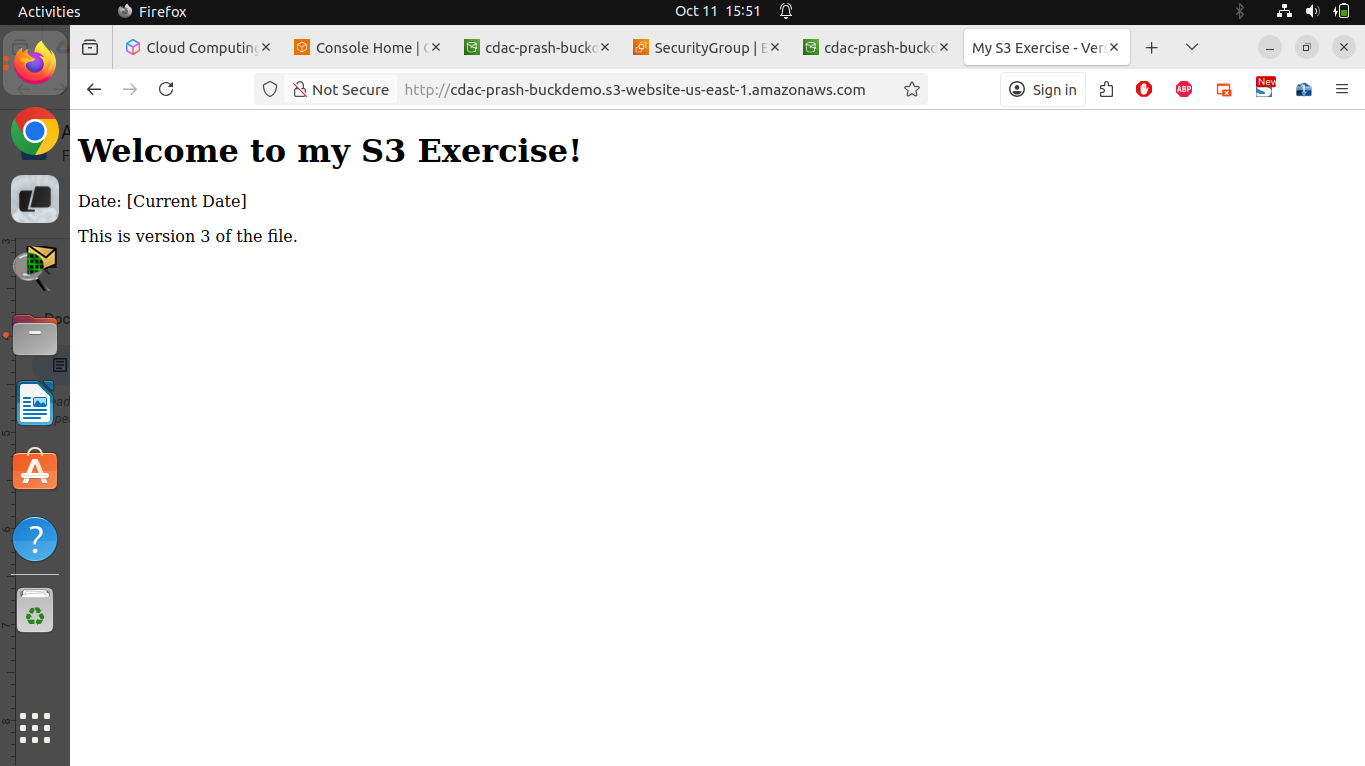
</body>

</html>

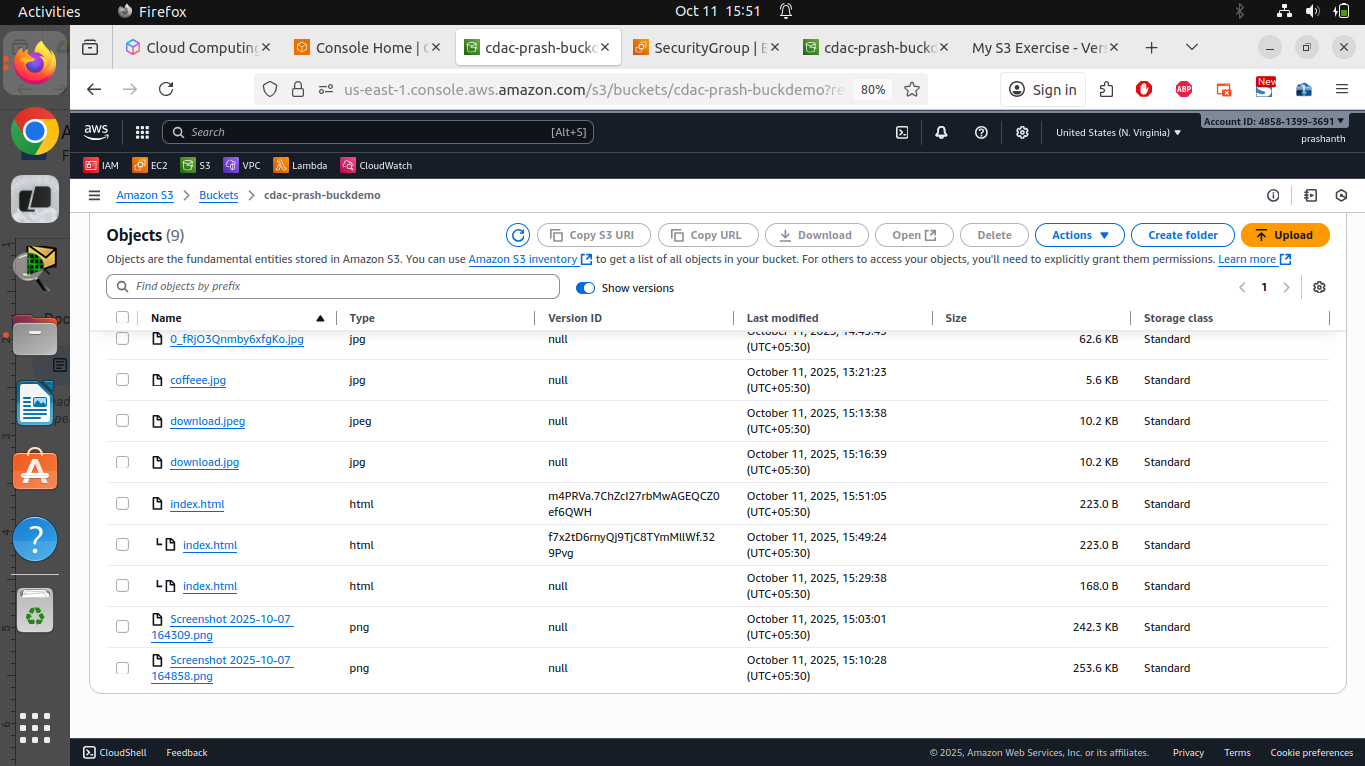
* + Upload the edited file with the same name

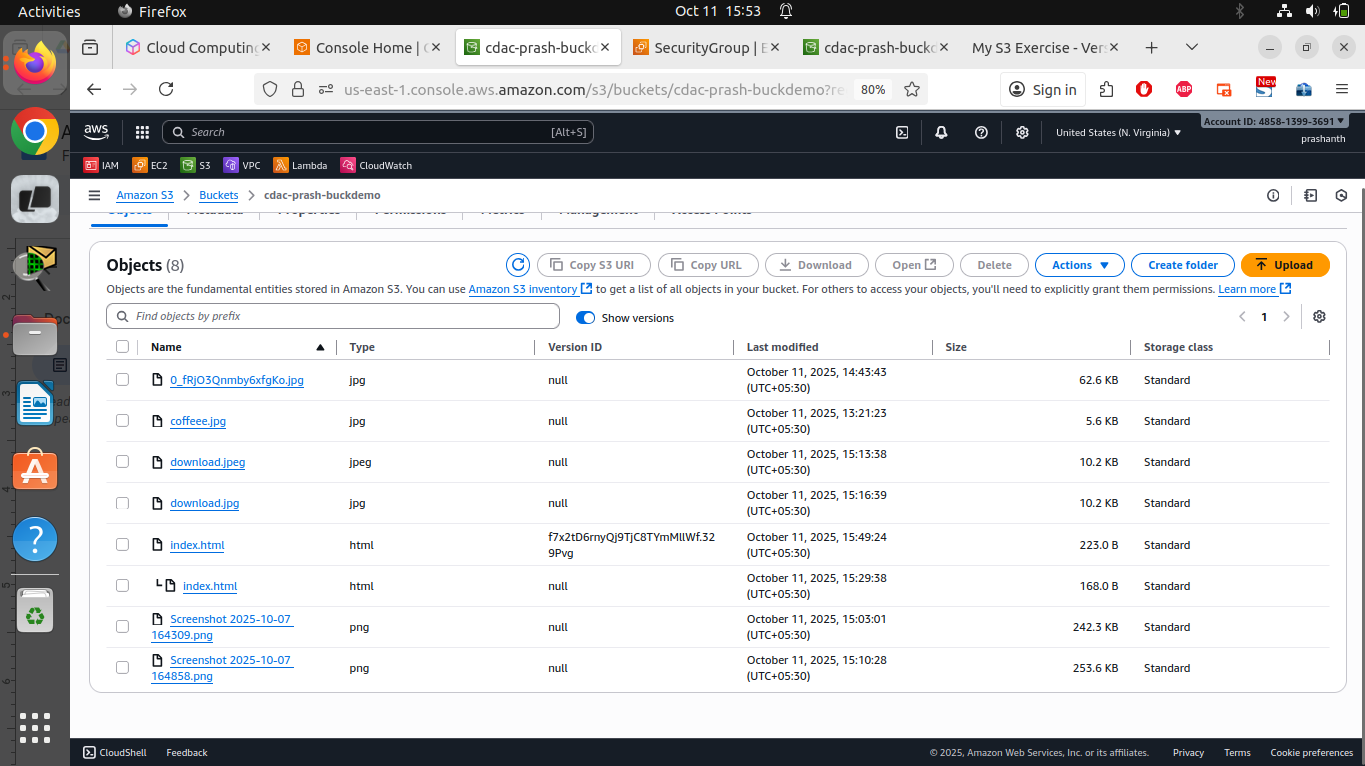


* + Repeat this process with another change for version 3

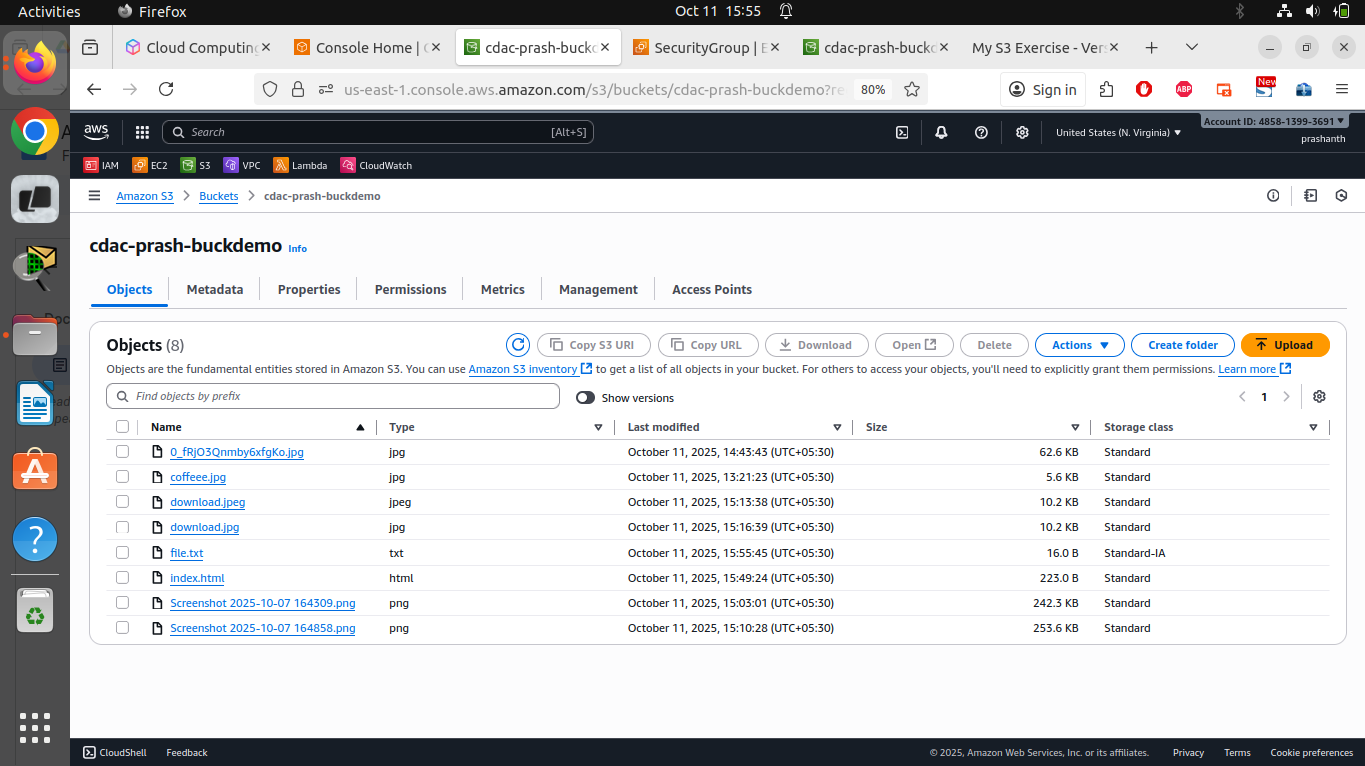


1. Manage versions
   * Enable "Show versions" in the Objects view
   * Verify that multiple versions of index.html are displayed
   * Download an older version to confirm that the original content is preserved
   * Restore an older version by selecting it and choosing "Delete" for the newest version

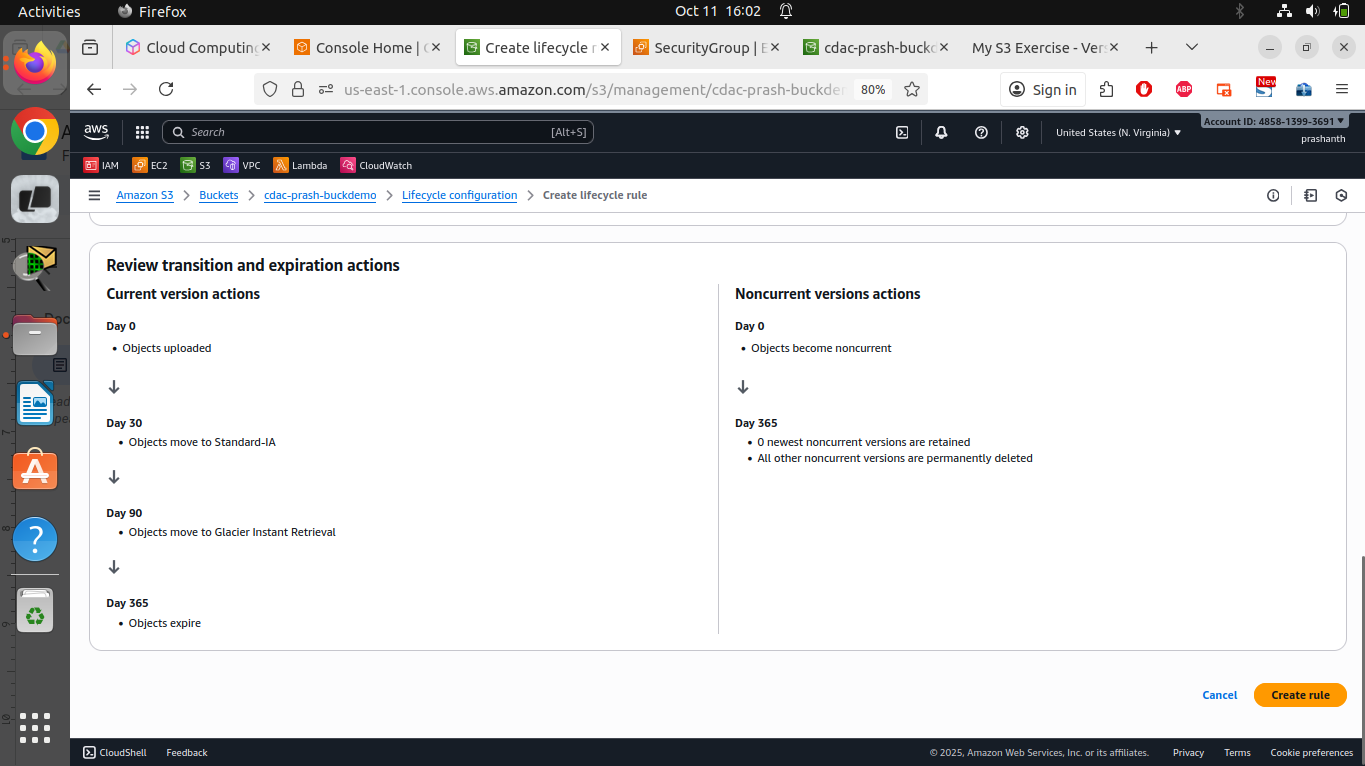




1. Configure storage classes
   * Upload a new file (e.g., test-file.txt) with the standard storage
   * Select the file and click "Actions" → "Edit storage class"
   * Change the storage class to "S3 Standard-IA" (Infrequent Access)
   * Save the change



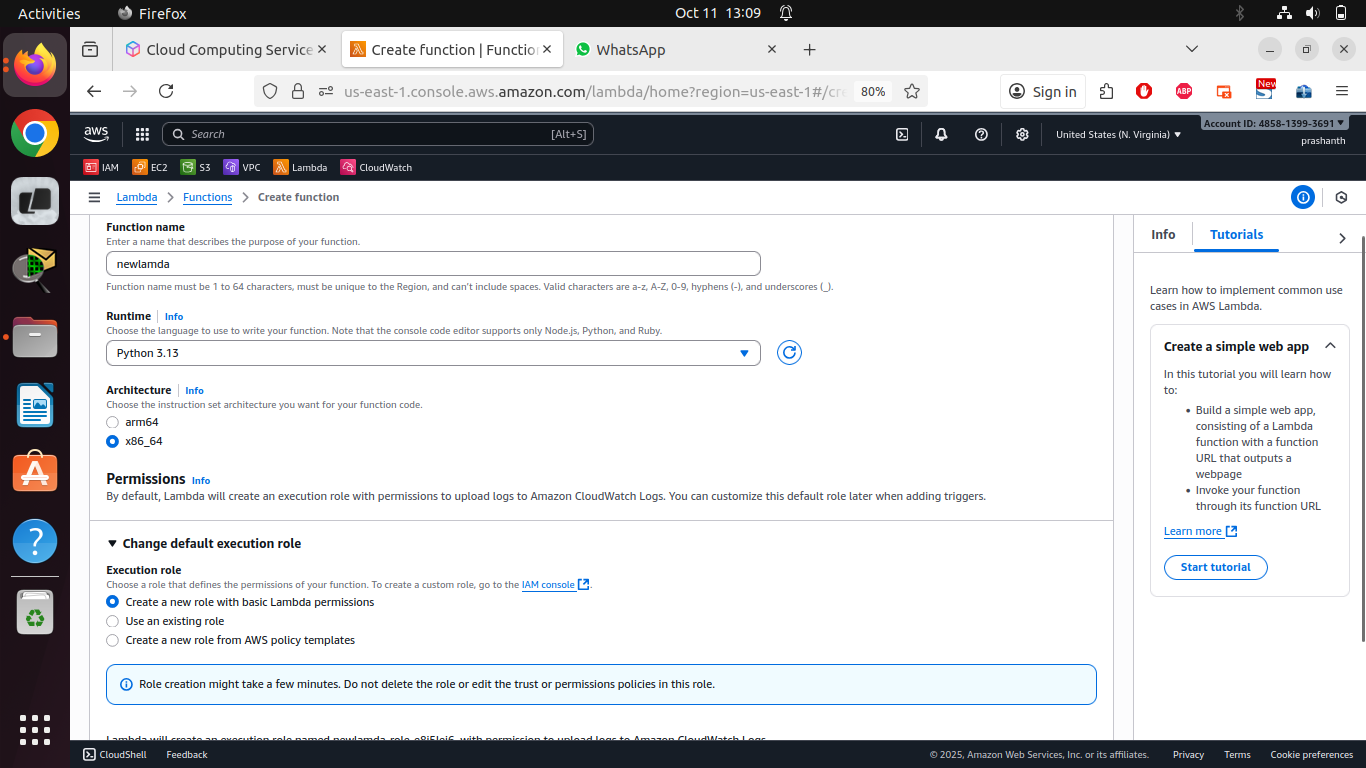
1. Create a lifecycle policy
   * Go to the "Management" tab
   * Click on "Create lifecycle rule"
   * Name the rule (e.g., "Archive-and-Delete-Rule")
   * Select "Apply to all objects in the bucket"
   * Configure the following actions:
     + After 30 days: Transition to S3 Standard-IA
     + After 90 days: Transition to S3 Glacier
     + After 365 days: Expire current versions of objects AND Permanently delete noncurrent versions of objects
   * Additionally enable "Delete expired object delete markers"
   * Create the rule



Labs on Lambda

Lab 1:- Creating a Lambda Function

1. Launch Lambda console
2. Create a new function
   1. Click Create Function.
   2. Choose “Author from Scratch.”
   3. Provide a name for your function.
   4. Select a runtime (e.g., Python 3.11).
   5. Click on the Create Function button.

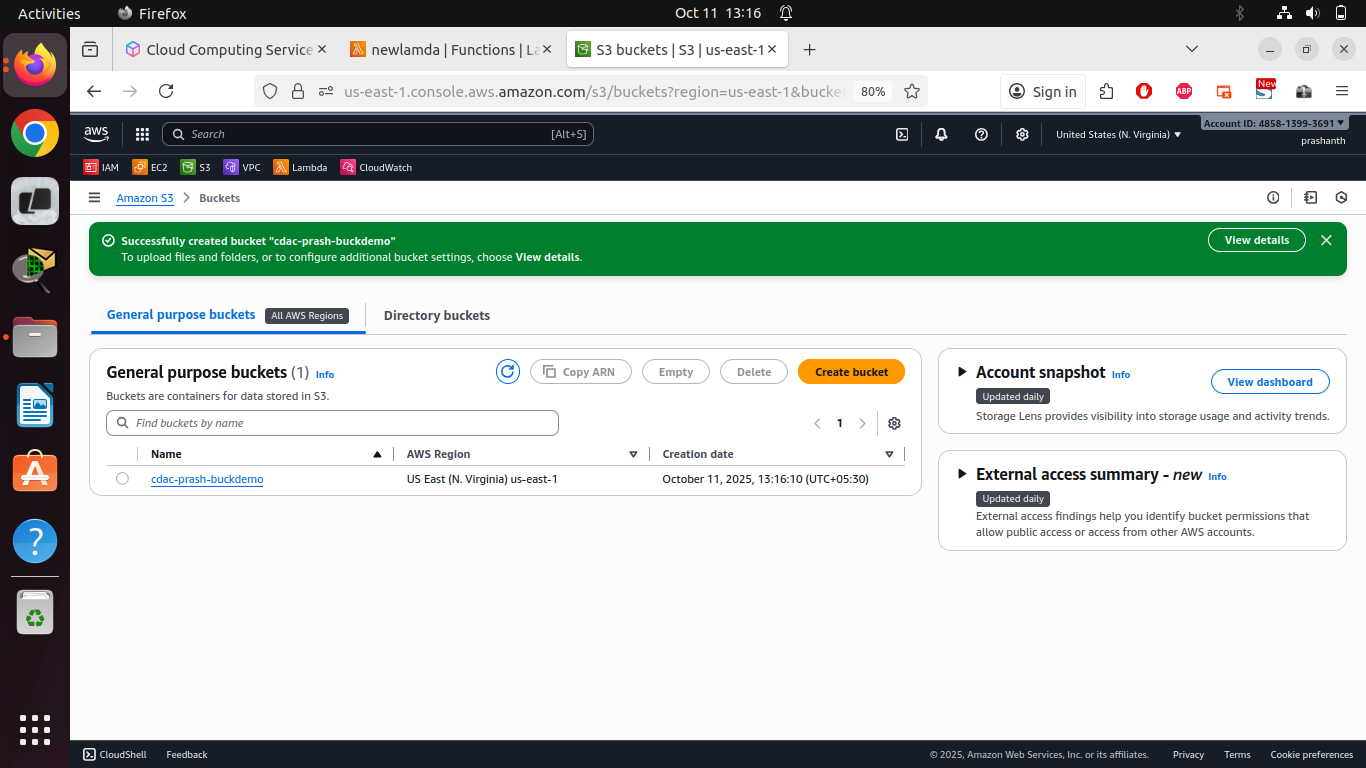


1. Edit the code for custom message. Test the Lambda function

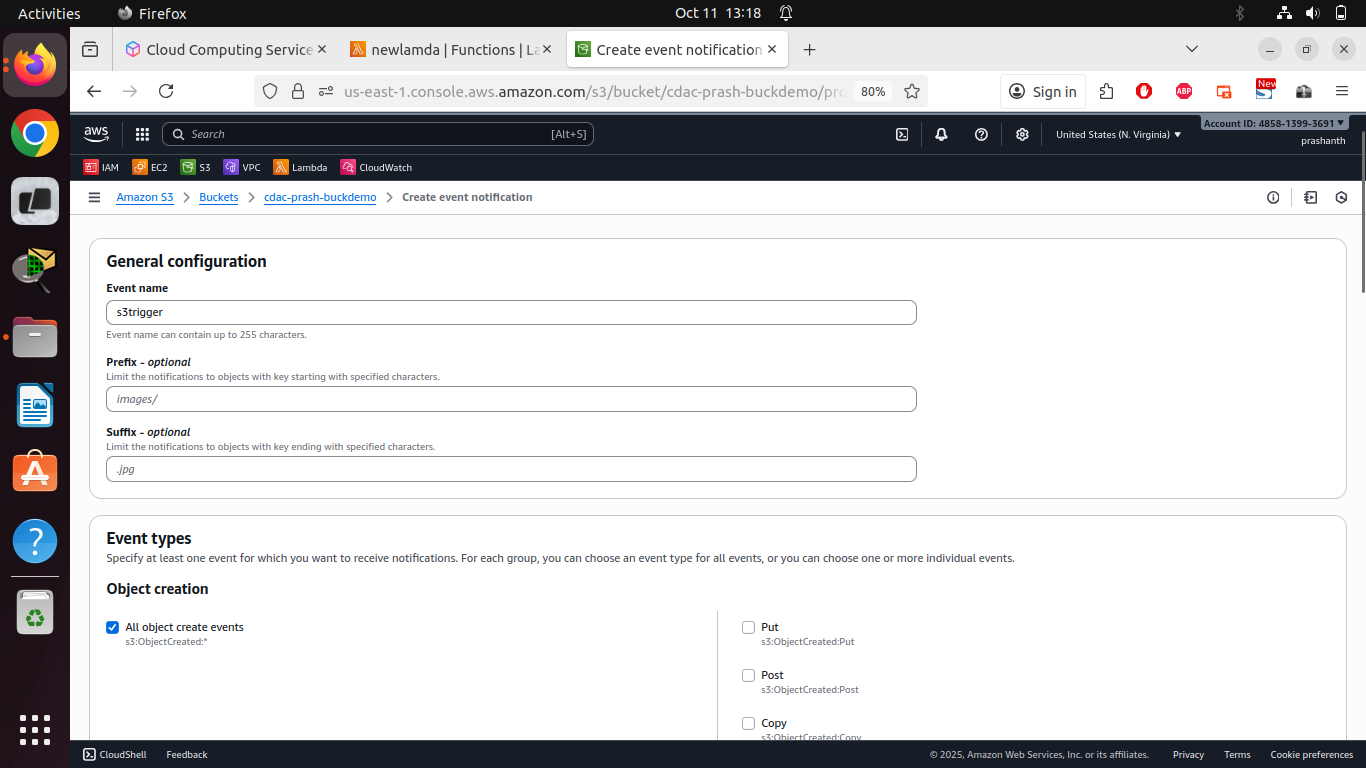


Lab 2: Triggering Lambda Function with events

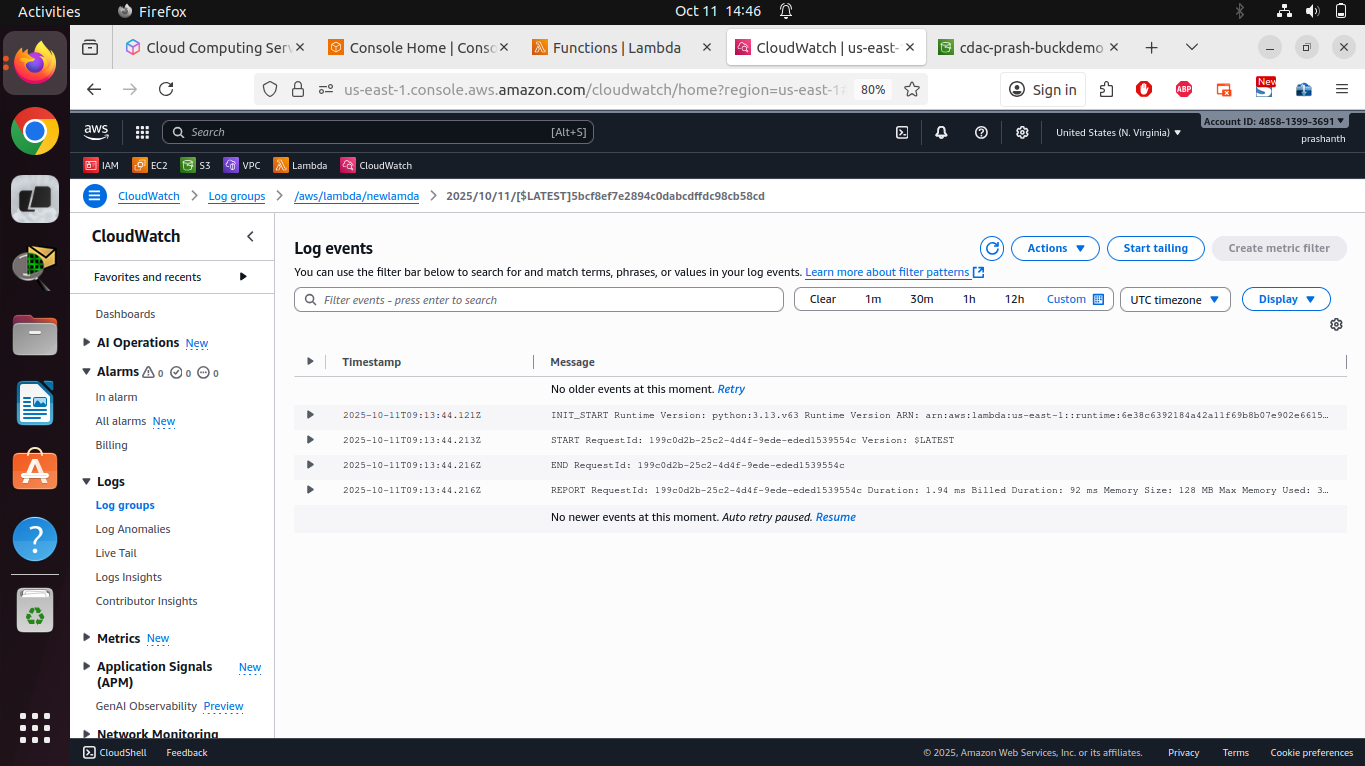
1. Launch the S3 console.
2. Create a bucket with all default settings
3. Select S3 bucket.



1. Under Properties, add an event notification to trigger your Lambda function when creating and removing object.
2. Scroll down to Event notification. Create event notification
3. Choose the events for triggering. Select the Lambda function to trigger in S3.

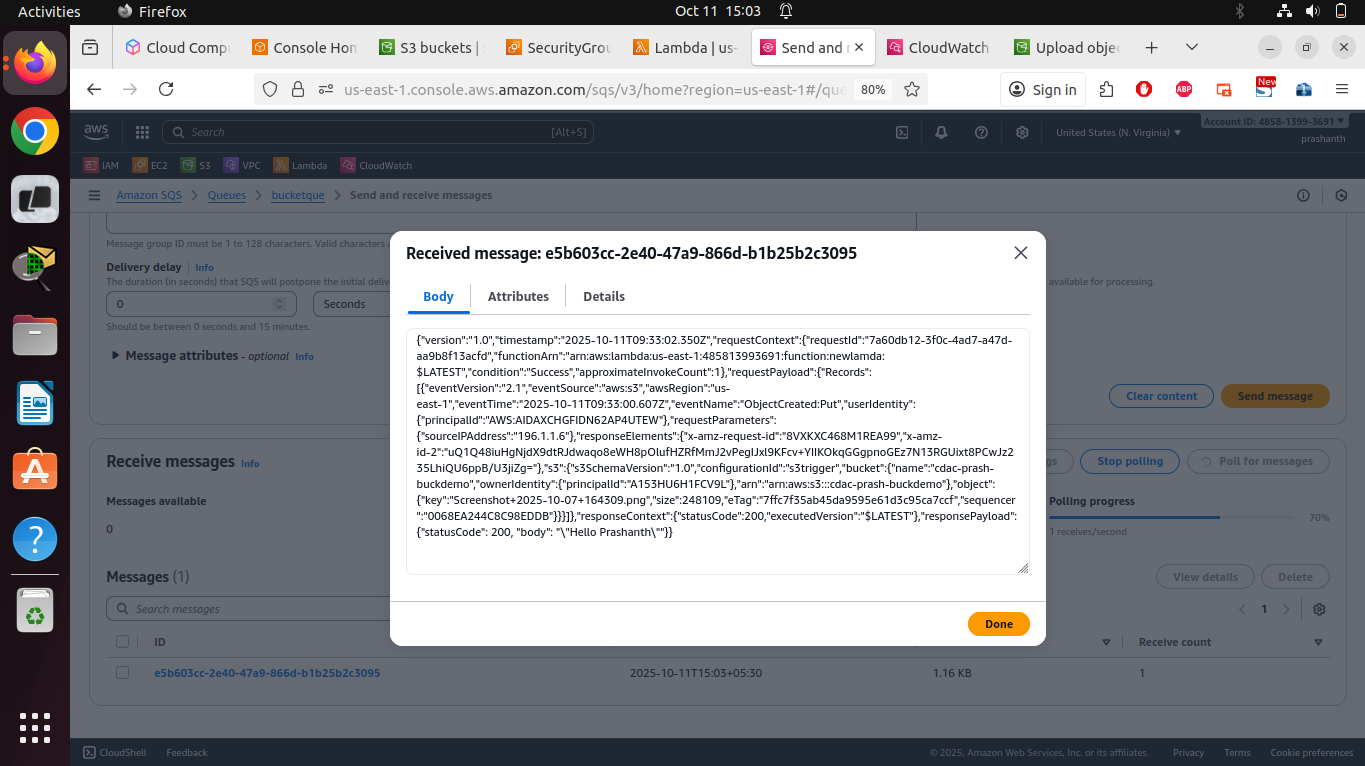


1. Upload/Delete an object in the S3 bucket. Verify if Lambda function is triggered.
2. Monitor the CloudWatch logs.



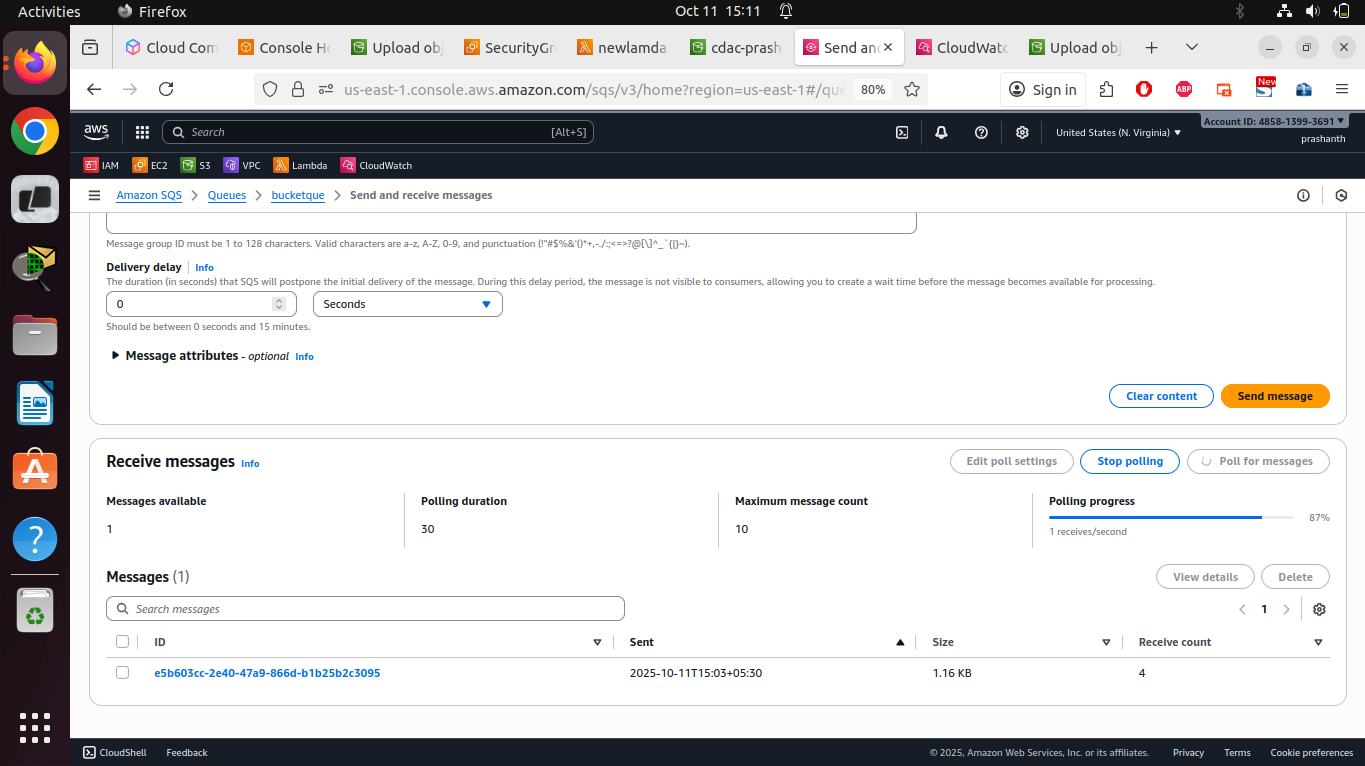
Lab 3: Add destination to the Lambda Function.

1. Create SNS topic to send a mail whenever the Lambda function is triggered based on S3 event.

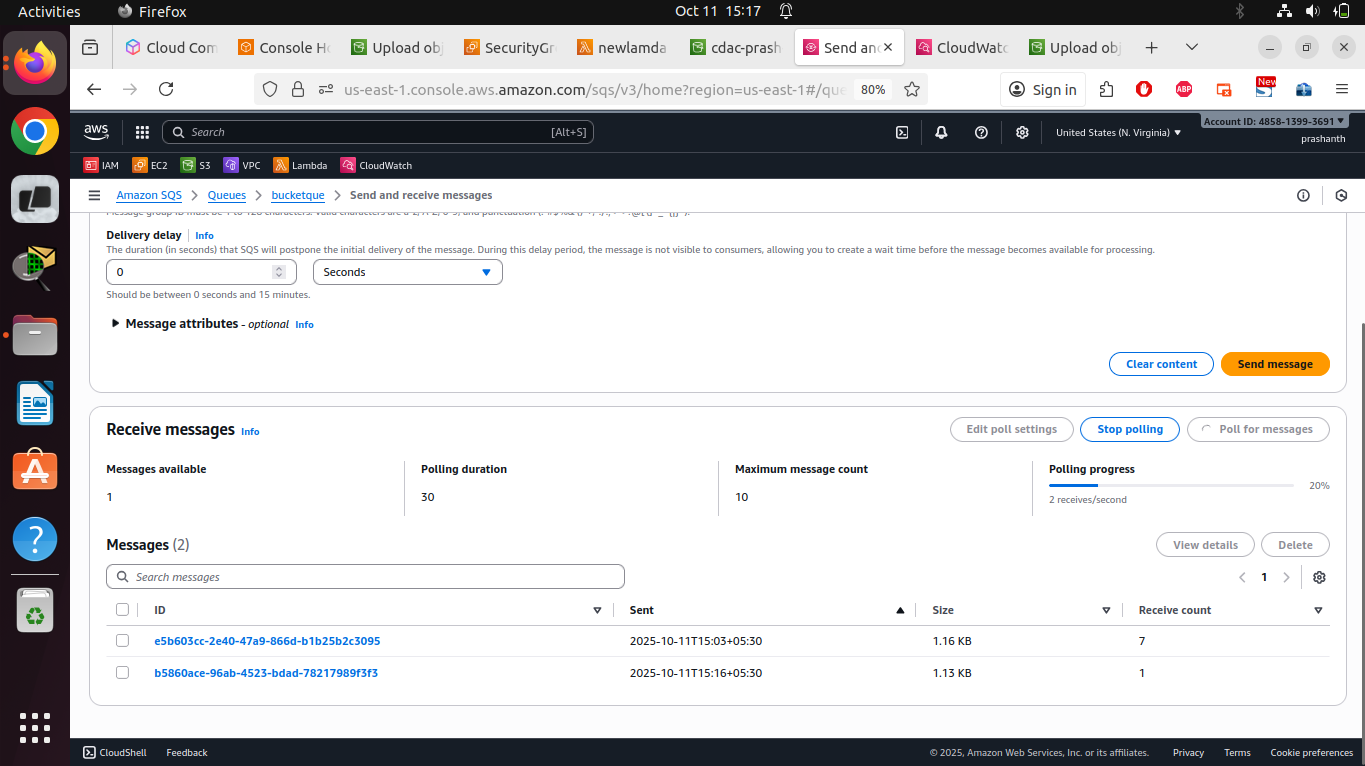


Lab 4: Modify in Lab2

1. Trigger the Lambda function only when .jpg files are uploaded.
2. Verify if the function does get triggered



**Not Triggered when uploaded .png file**

****

**Triggered when uploaded .jpg file**