Assisted Practice: 3.6 Launch and Connect to an EC2 Linux Instance

This section will guide you to:

- Launch and connect to an EC2 instance

This lab has three subsections, namely:

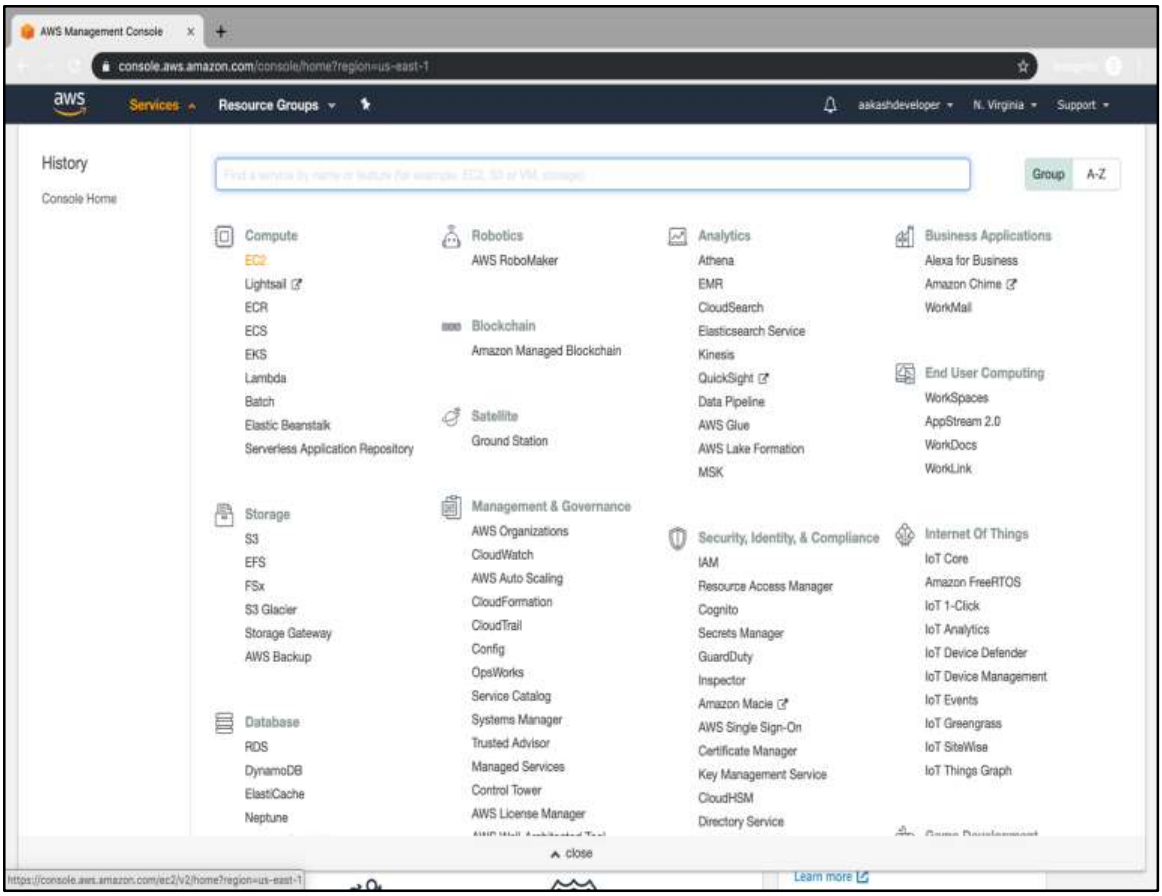
3.1.1 Launching an EC2 instance

3.1.2 Connecting to the EC2 instance

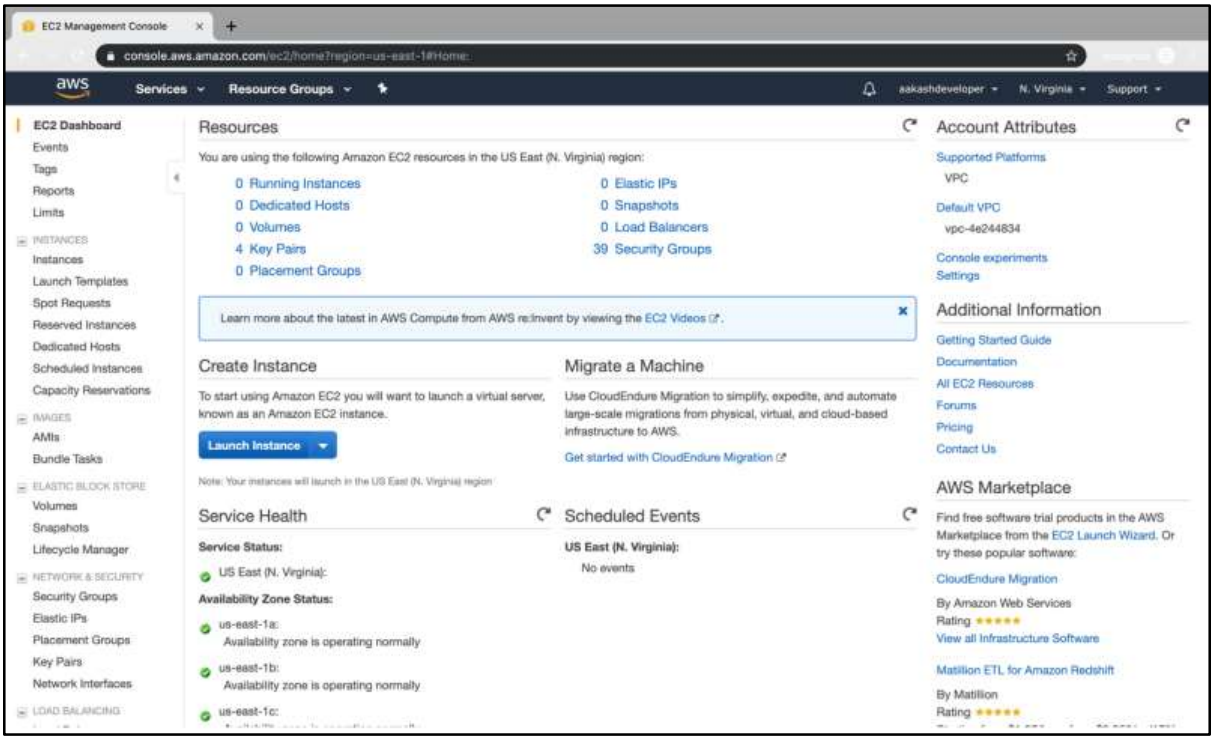
3.1.3 Pushing the files to GitHub repositories

Step 3.1.1: Launching an EC2 instance

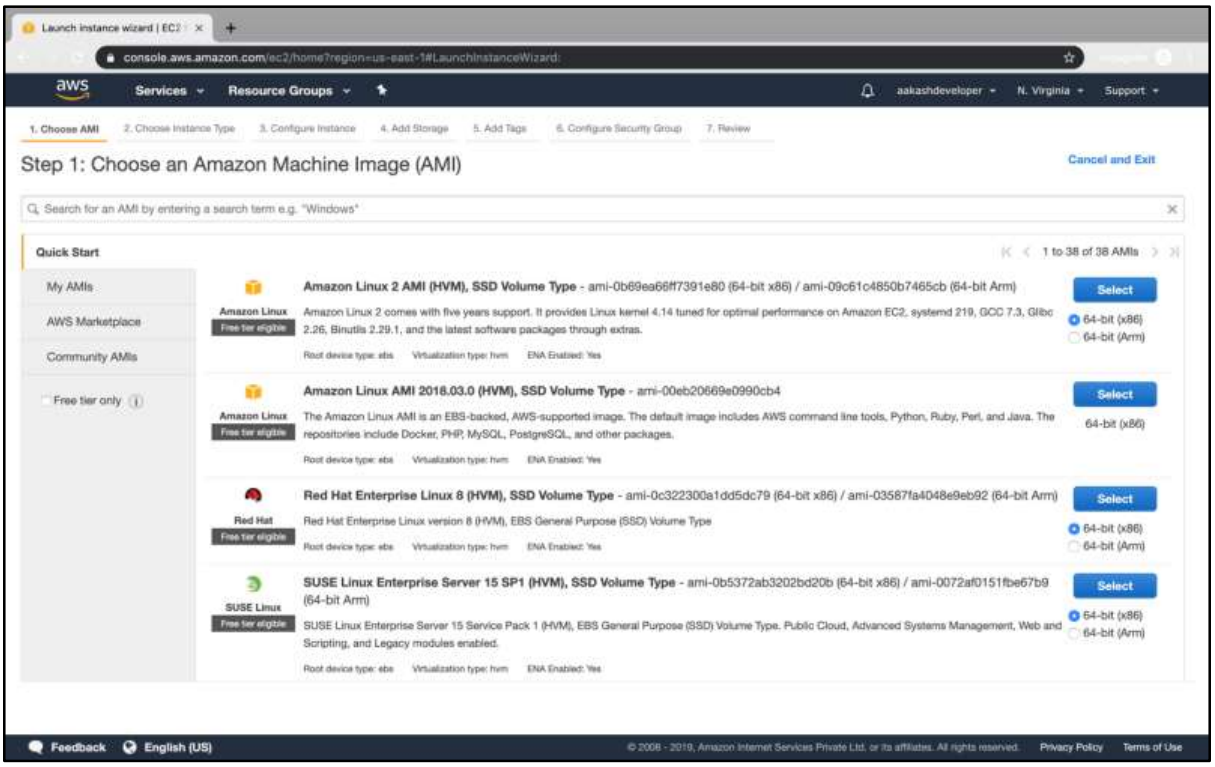
- Go to Amazon dashboard
- Select EC2



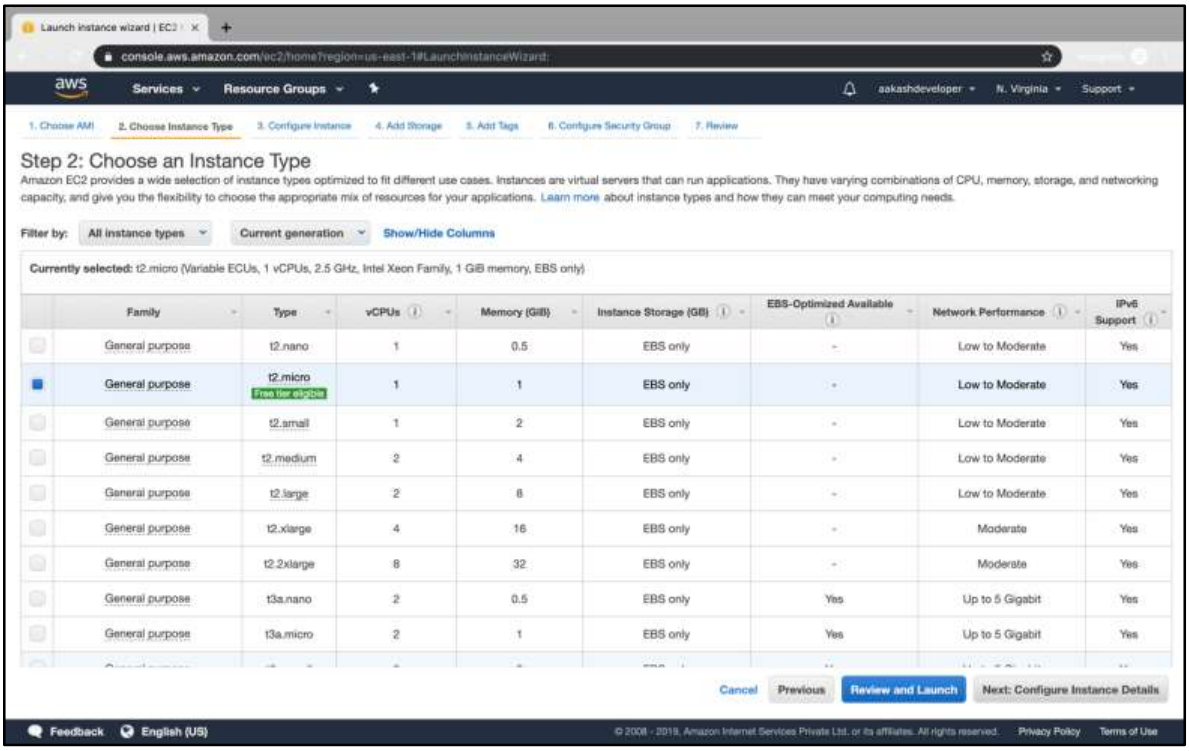
- Click on *launch instance* to run any instance



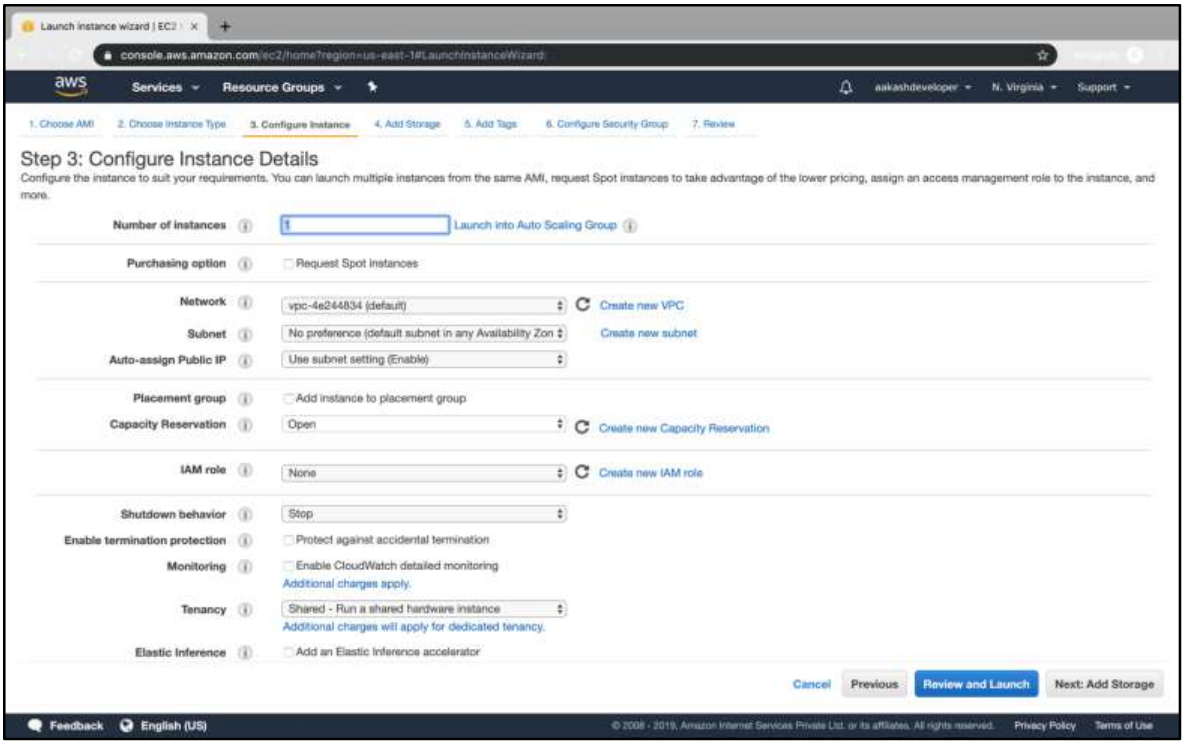
- Select the AMI



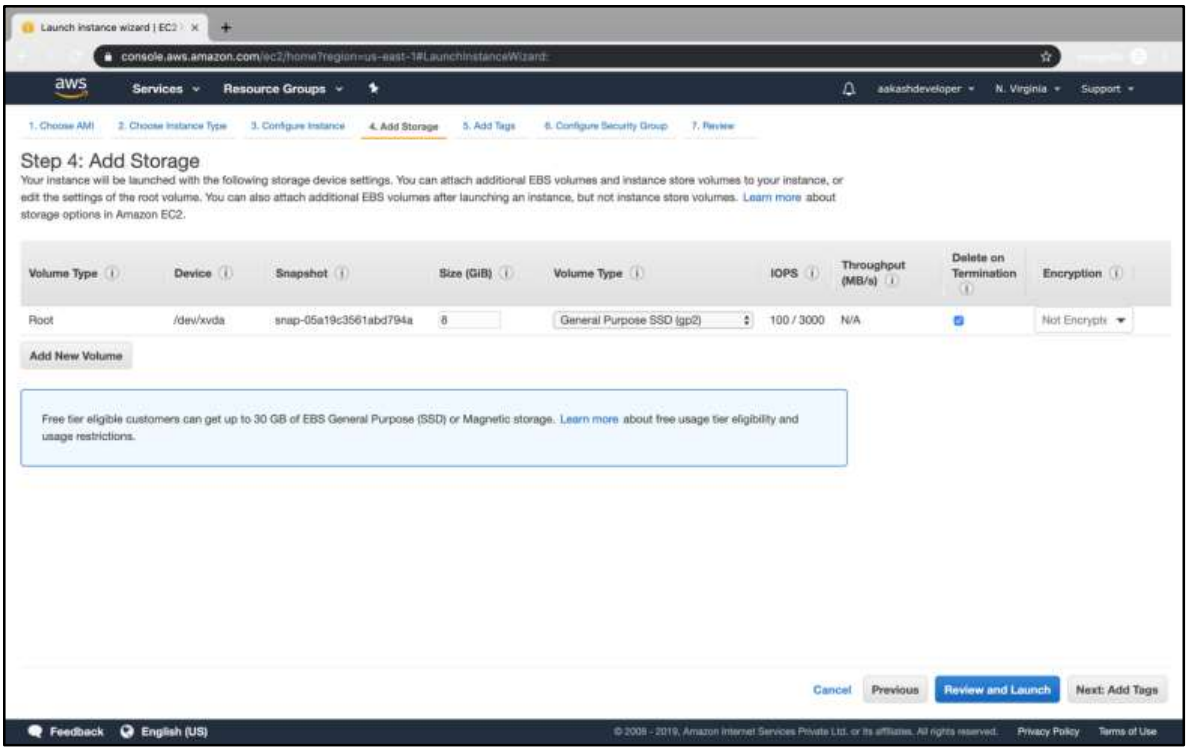
- Select t2.micro as the instance type



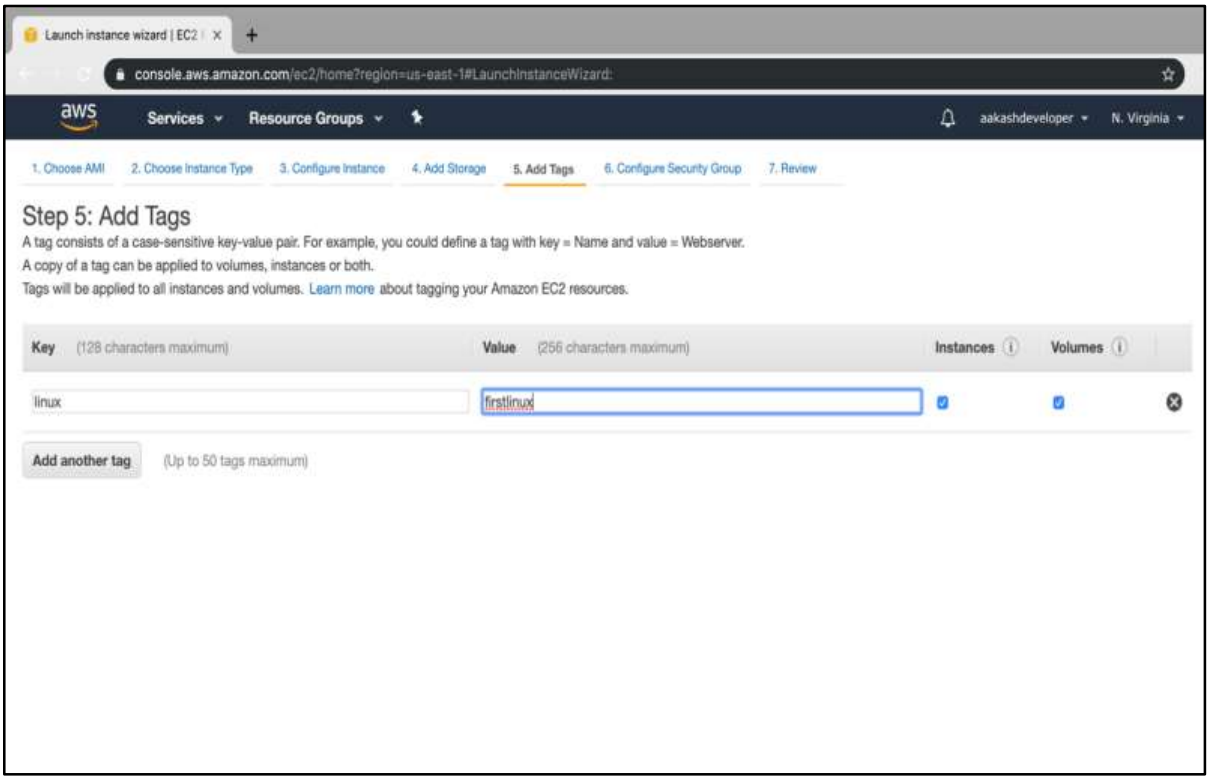
- Specify the number of instances, networks, placement groups, and IAM roles and click *Next*

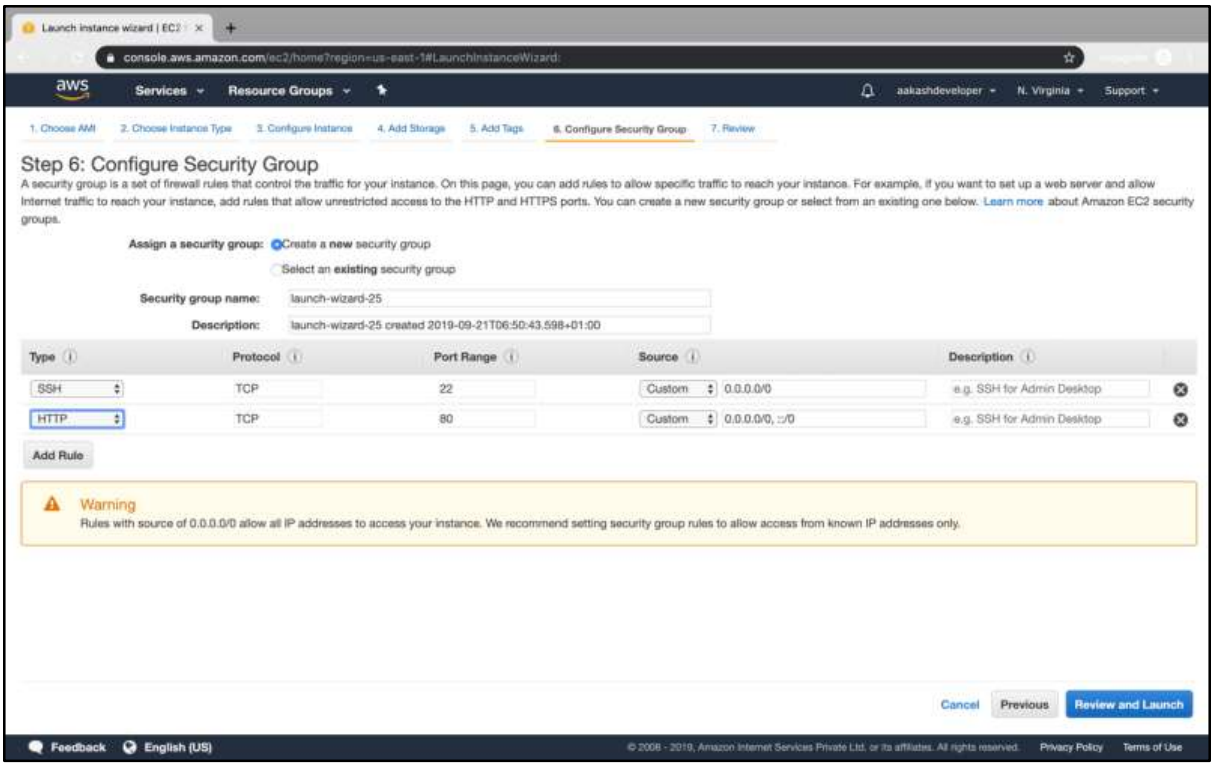


- There is one volume attached to the instance by default
- In Linux, the default volume attached is 8 GB
- You can add more volume if required

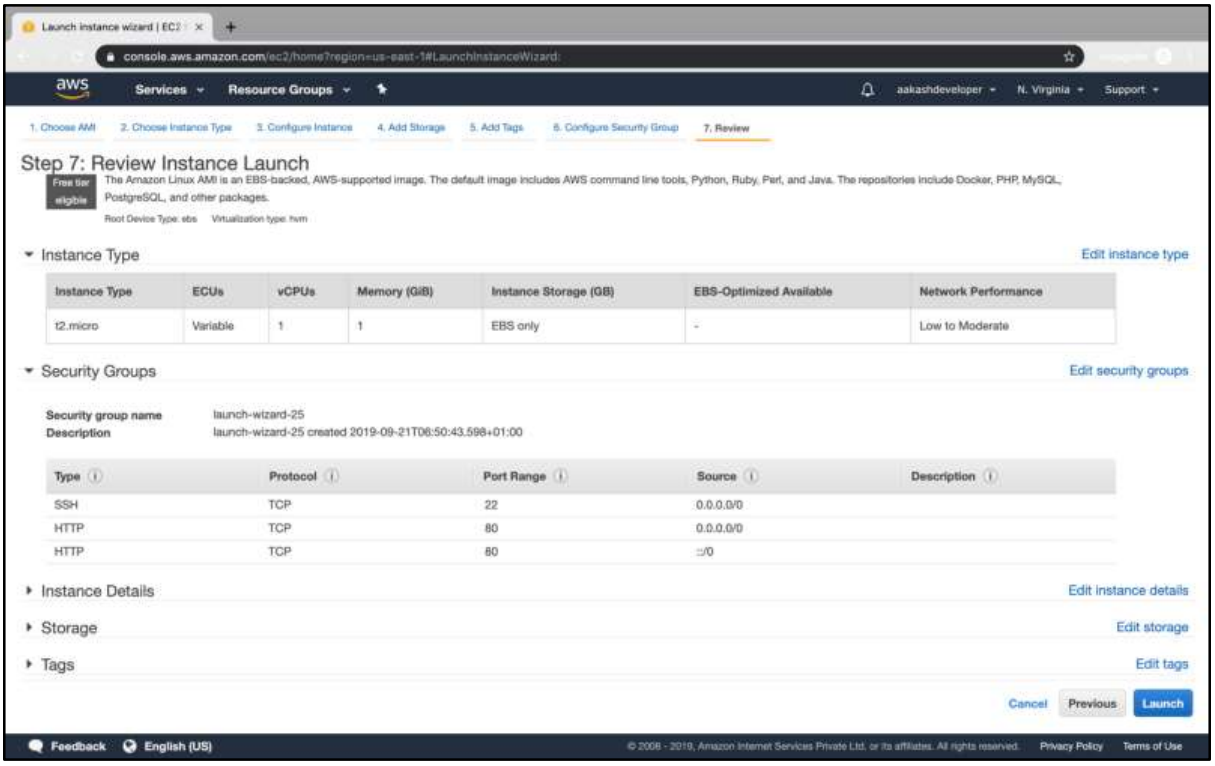


- You can add a key-value pair to the instance

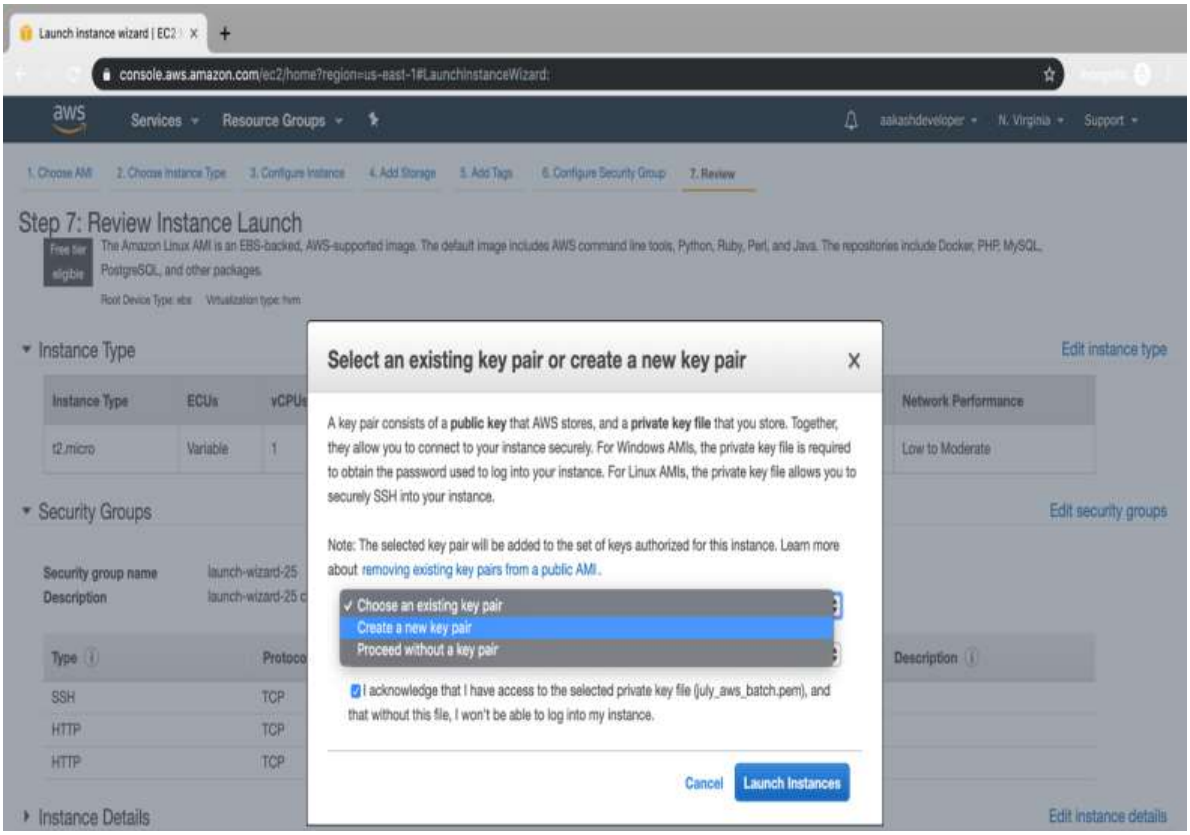


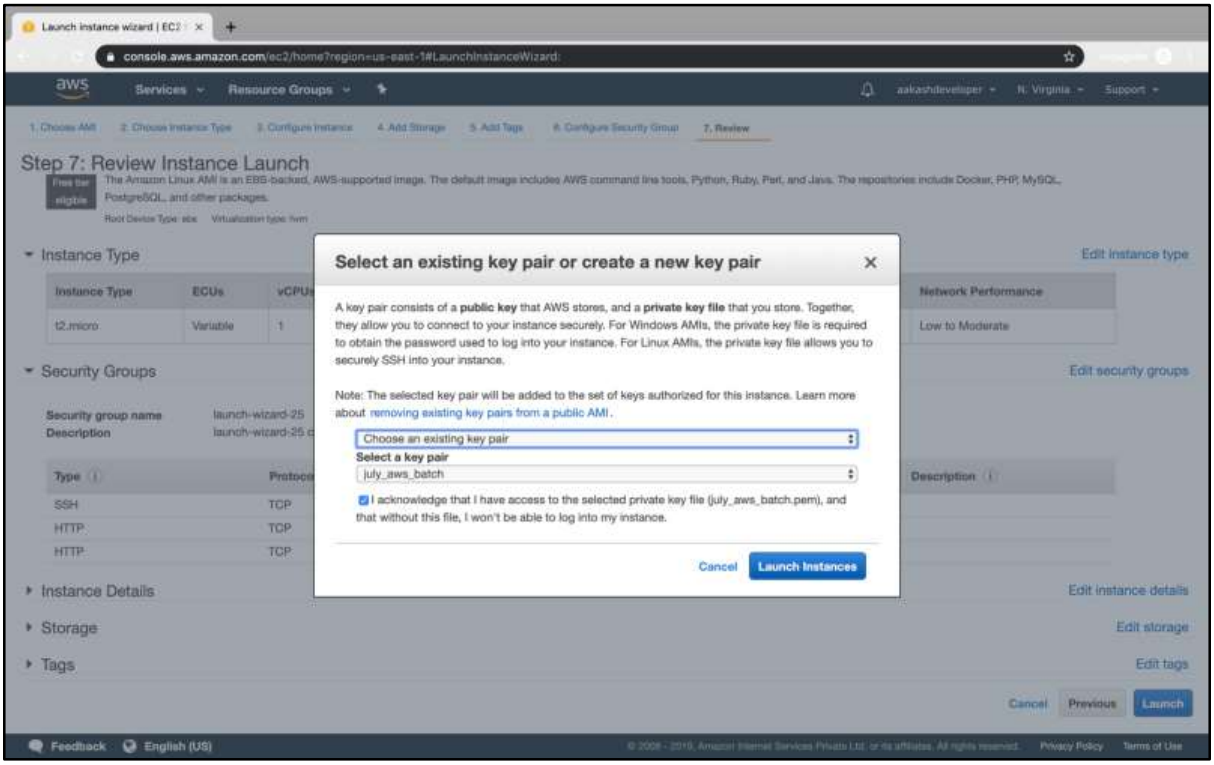


- Attach the three policy groups depending on the type of access required

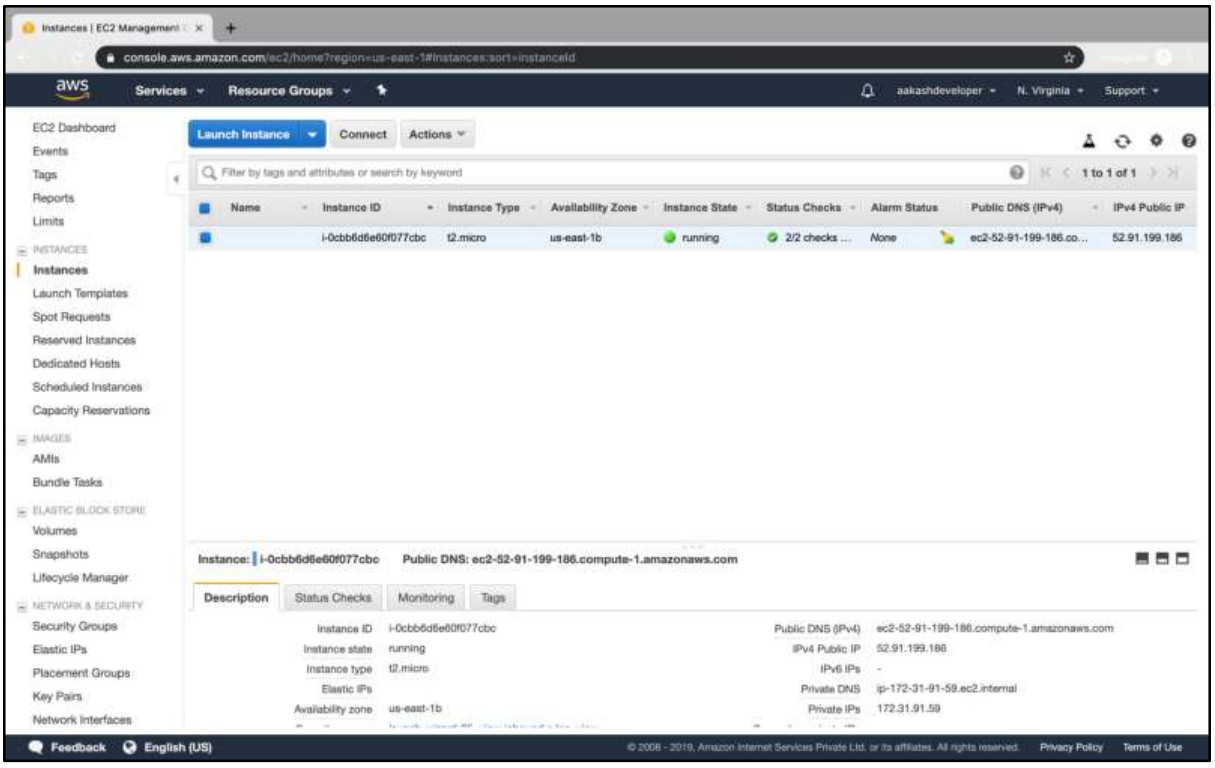


- Select an existing key-value pair to launch the instance



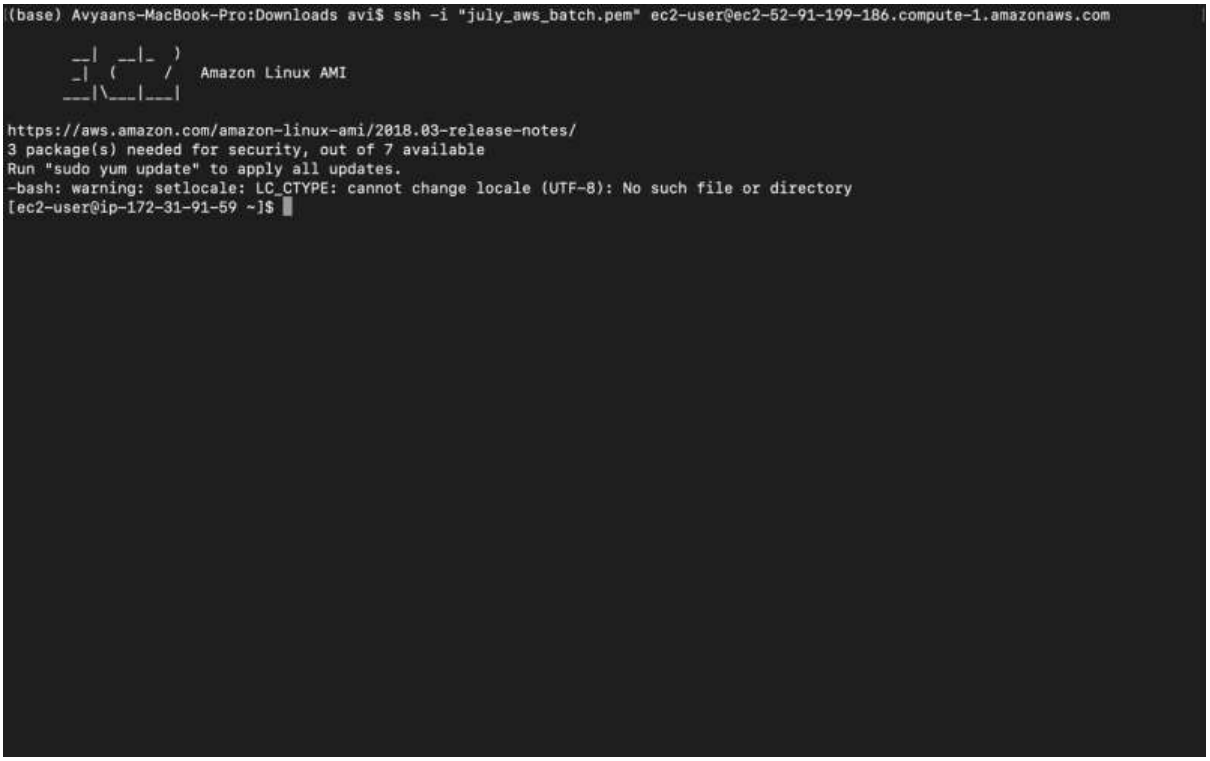


- The instance is ready to use



- Click on *Connect* on EC2 dashboard
- Run the ssh command provided

```
Last login: Fri Sep 20 17:34:39 on ttys001
(base) Avyaans-MacBook-Pro:~ avi$ ssh -i "july_aws_batch.pem" ec2-user@ec2-52-91-199-186.compute-1.amazonaws.com
```



Step 3.1.2: Connecting to an EC2 instance

- Run the following command to launch a website over EC2:

  yum install httpd -y

```
          [command]
usage: sudo [-AbEHknPS] [-r role] [-t type] [-C fd] [-D level] [-g groupname|#gid] [-p prompt] [-u user name|#uid] [-g
          groupname|#gid] [VAR=value] [-i|-s] [<command>]
usage: sudo -e [-AknS] [-r role] [-t type] [-C fd] [-D level] [-g groupname|#gid] [-p prompt] [-u user name|#uid] file ...
[ec2-user@ip-172-31-91-59 ~]$ sudo su -
[root@ip-172-31-91-59 ~]# yum install httpd -y
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                                                                          | 2.1 kB  00:00:00
amzn-updates                                                                                       | 2.5 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.2.34-1.16.amzn1 will be installed
--> Processing Dependency: httpd-tools = 2.2.34-1.16.amzn1 for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: apr-util-ldap for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.2.34-1.16.amzn1.x86_64
--> Running transaction check
---> Package apr.x86_64 0:1.5.2-5.13.amzn1 will be installed
---> Package apr-util.x86_64 0:1.5.4-6.18.amzn1 will be installed
---> Package apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1 will be installed
---> Package httpd-tools.x86_64 0:2.2.34-1.16.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                    Arch                   Version                        Repository                             Size
================================================================================================================================
Installing:
 httpd                      x86_64                 2.2.34-1.16.amzn1              amzn-main                             1.2 M
Installing for dependencies:
 apr                        x86_64                 1.5.2-5.13.amzn1               amzn-main                             118 k
 apr-util                   x86_64                 1.5.4-6.18.amzn1               amzn-main                              99 k
 apr-util-ldap              x86_64                 1.5.4-6.18.amzn1               amzn-main                              19 k
 httpd-tools                x86_64                 2.2.34-1.16.amzn1              amzn-main                              80 k

Transaction Summary
================================================================================================================================
```

- Navigate to /var/www/html and create index.html file using command

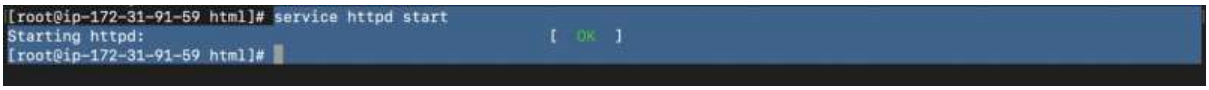
  vi index.html

```
Install  1 Package (+4 Dependent packages)

Total download size: 1.5 M
Installed size: 3.6 M
Downloading packages:
(1/5): apr-util-ldap-1.5.4-6.18.amzn1.x86_64.rpm                                                   |  19 kB  00:00:00
(2/5): apr-1.5.2-5.13.amzn1.x86_64.rpm                                                             | 118 kB  00:00:00
(3/5): apr-util-1.5.4-6.18.amzn1.x86_64.rpm                                                        |  99 kB  00:00:00
(4/5): httpd-tools-2.2.34-1.16.amzn1.x86_64.rpm                                                    |  80 kB  00:00:00
(5/5): httpd-2.2.34-1.16.amzn1.x86_64.rpm                                                          | 1.2 MB  00:00:00
--------------------------------------------------------------------------------------------------------------------------------
Total                                                                              1.9 MB/s | 1.5 MB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : apr-1.5.2-5.13.amzn1.x86_64                                                                              1/5
  Installing : apr-util-1.5.4-6.18.amzn1.x86_64                                                                         2/5
  Installing : httpd-tools-2.2.34-1.16.amzn1.x86_64                                                                     3/5
  Installing : apr-util-ldap-1.5.4-6.18.amzn1.x86_64                                                                    4/5
  Installing : httpd-2.2.34-1.16.amzn1.x86_64                                                                           5/5
  Verifying  : httpd-tools-2.2.34-1.16.amzn1.x86_64                                                                     1/5
  Verifying  : apr-util-1.5.4-6.18.amzn1.x86_64                                                                         2/5
  Verifying  : httpd-2.2.34-1.16.amzn1.x86_64                                                                           3/5
  Verifying  : apr-1.5.2-5.13.amzn1.x86_64                                                                              4/5
  Verifying  : apr-util-ldap-1.5.4-6.18.amzn1.x86_64                                                                    5/5

Installed:
  httpd.x86_64 0:2.2.34-1.16.amzn1

Dependency Installed:
  apr.x86_64 0:1.5.2-5.13.amzn1              apr-util.x86_64 0:1.5.4-6.18.amzn1      apr-util-ldap.x86_64 0:1.5.4-6.18.amzn1
  httpd-tools.x86_64 0:2.2.34-1.16.amzn1

Complete!
[root@ip-172-31-91-59 ~]# cd /var/www/html/
```

- Enter *I* and start creating HTML file content

- Once done, type *:wq!*

- Using public IP of EC2 instance, you can see your app running on the browser



Step 3.1.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.7 Change the Volume Size of an Instance

This section will guide you to:

- Change the volume size of an instance

This lab has seven subsections, namely:

3.2.1 Creating the new volume which we want to add to an instance and clicking on **create volume**

3.2.2 Validating the Availability zone of the EC2 instance with which you want to add volume

3.2.3 Providing type, size, availability zone, and snapshot name on the **create volume** tab

3.2.4 Selecting newly created volume and clicking on the action to attach the volume to the instance

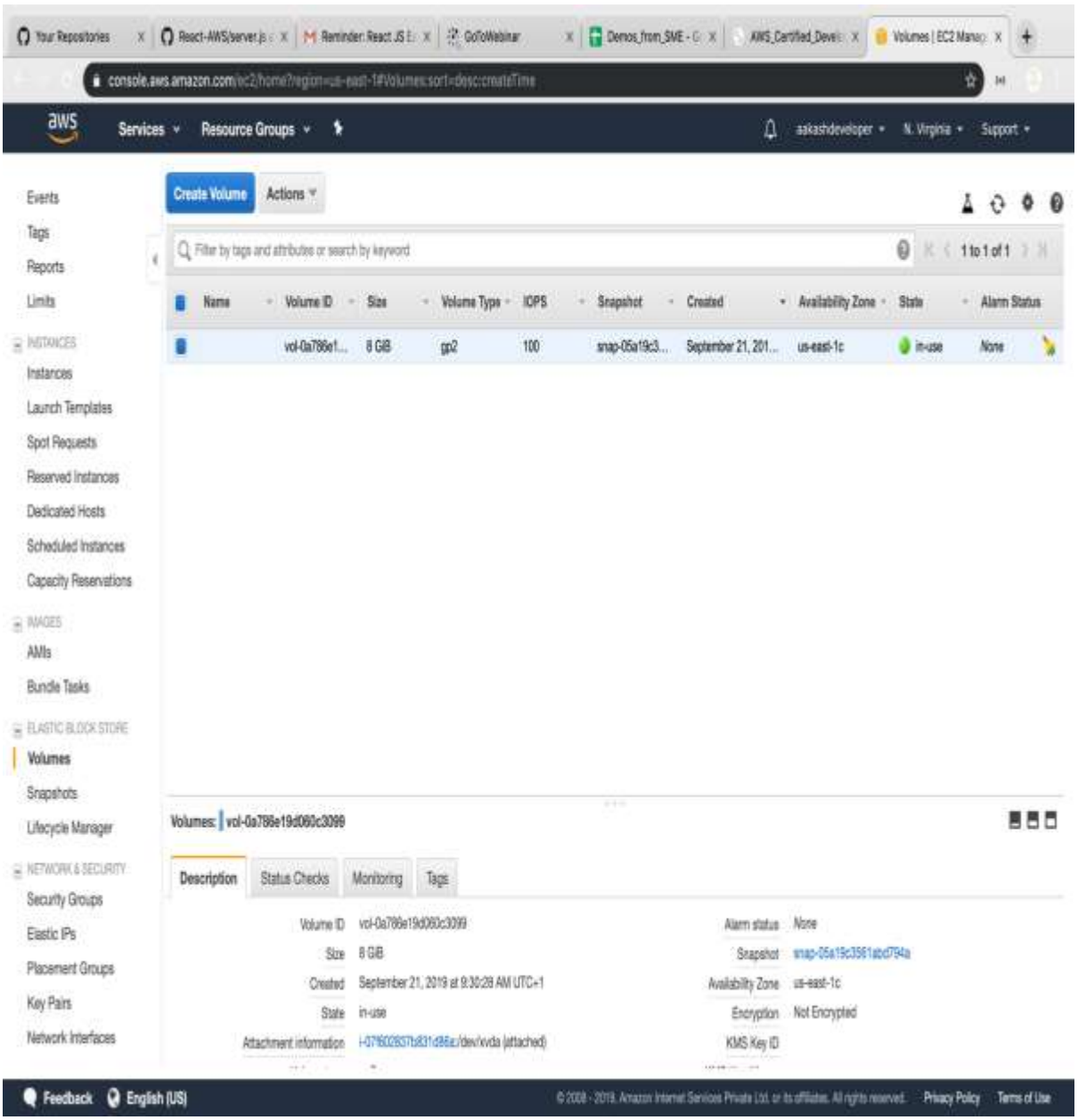
3.2.5 Selecting the instance of the same availability zone and attaching the volume in pop

3.2.6 Verifying the instance by adding one more device to the block device

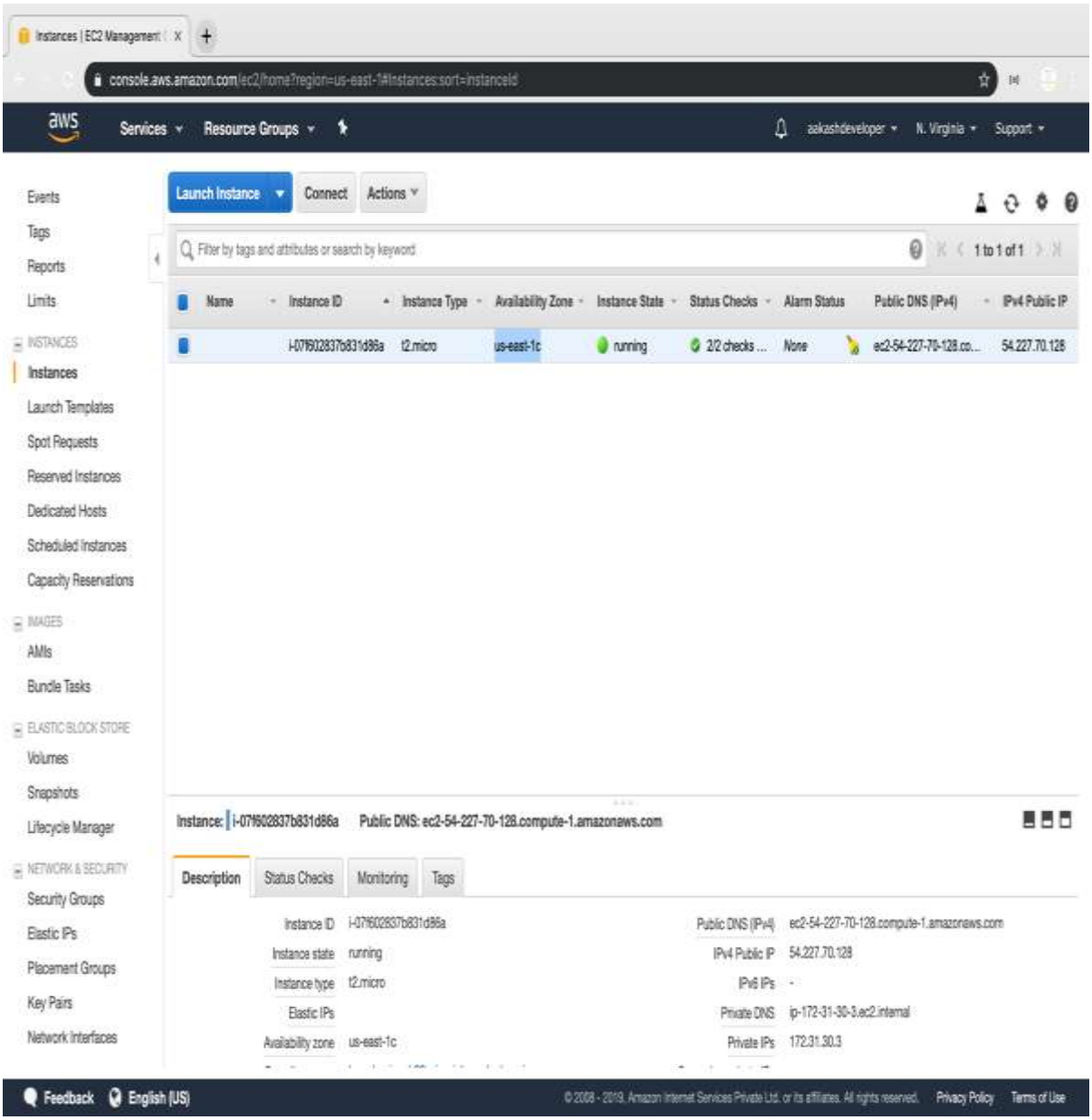
3.2.7 Pushing the code to GitHub repositories

Note: By default, when we create a new instance, we need to attach the minimum volume to the instance depending on the OS or AMI. But we can also add or remove the volume.

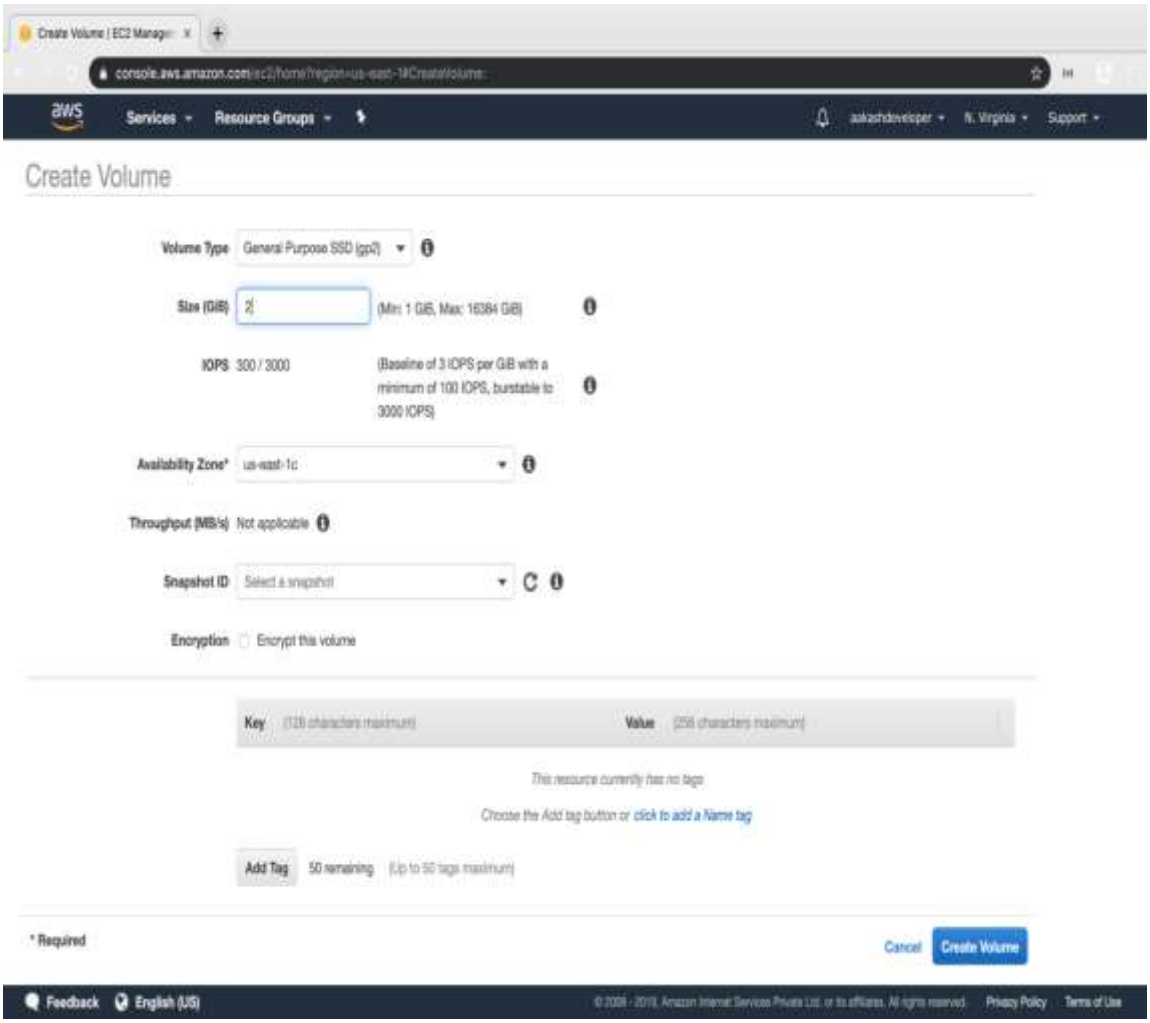
Step 3.2.1: Creating the new volume which we want to add to an instance and clicking on **create volume**



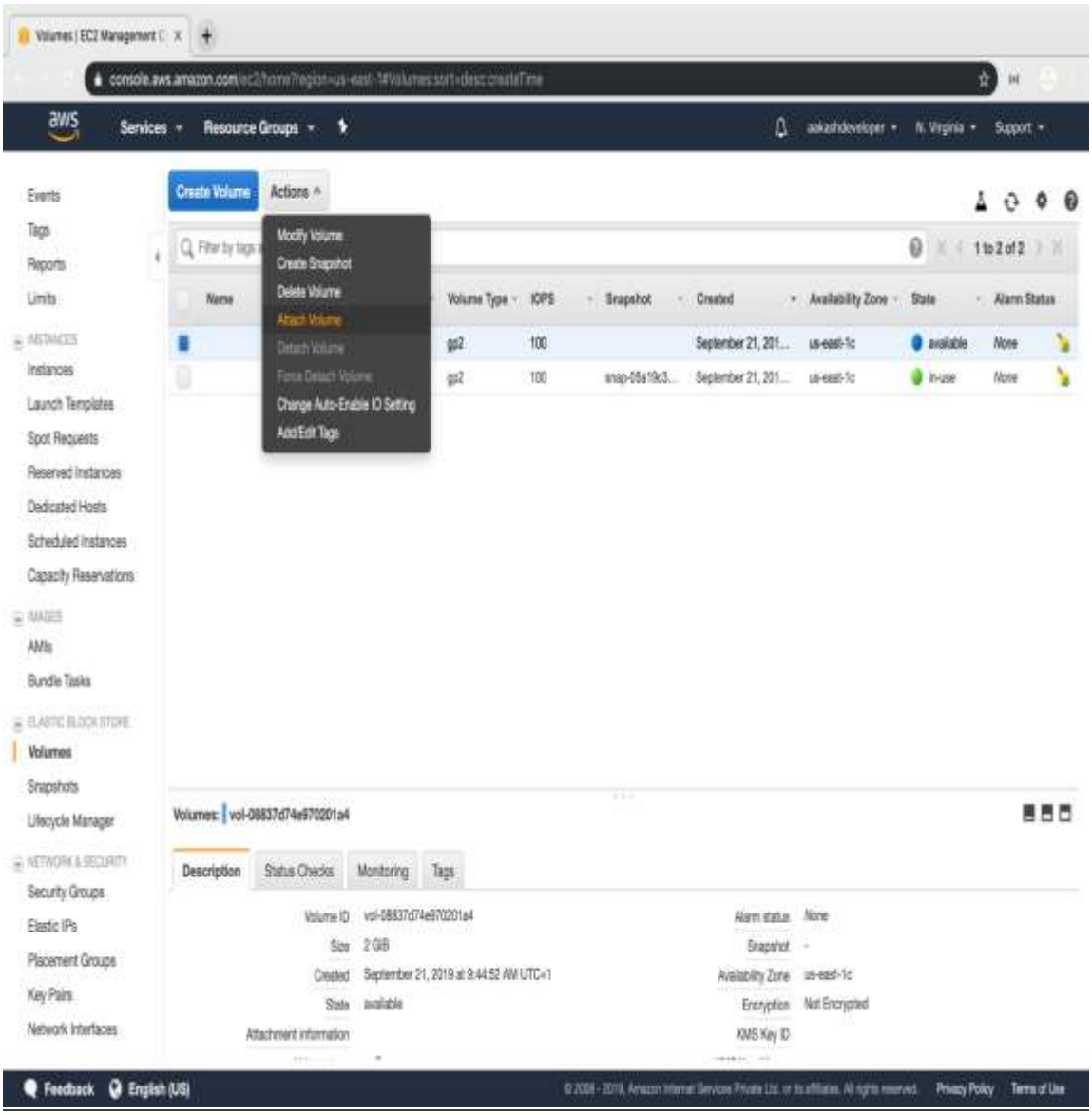
Step 3.2.2: Validating the Availability zone of the EC2 instance with which you want to add volume



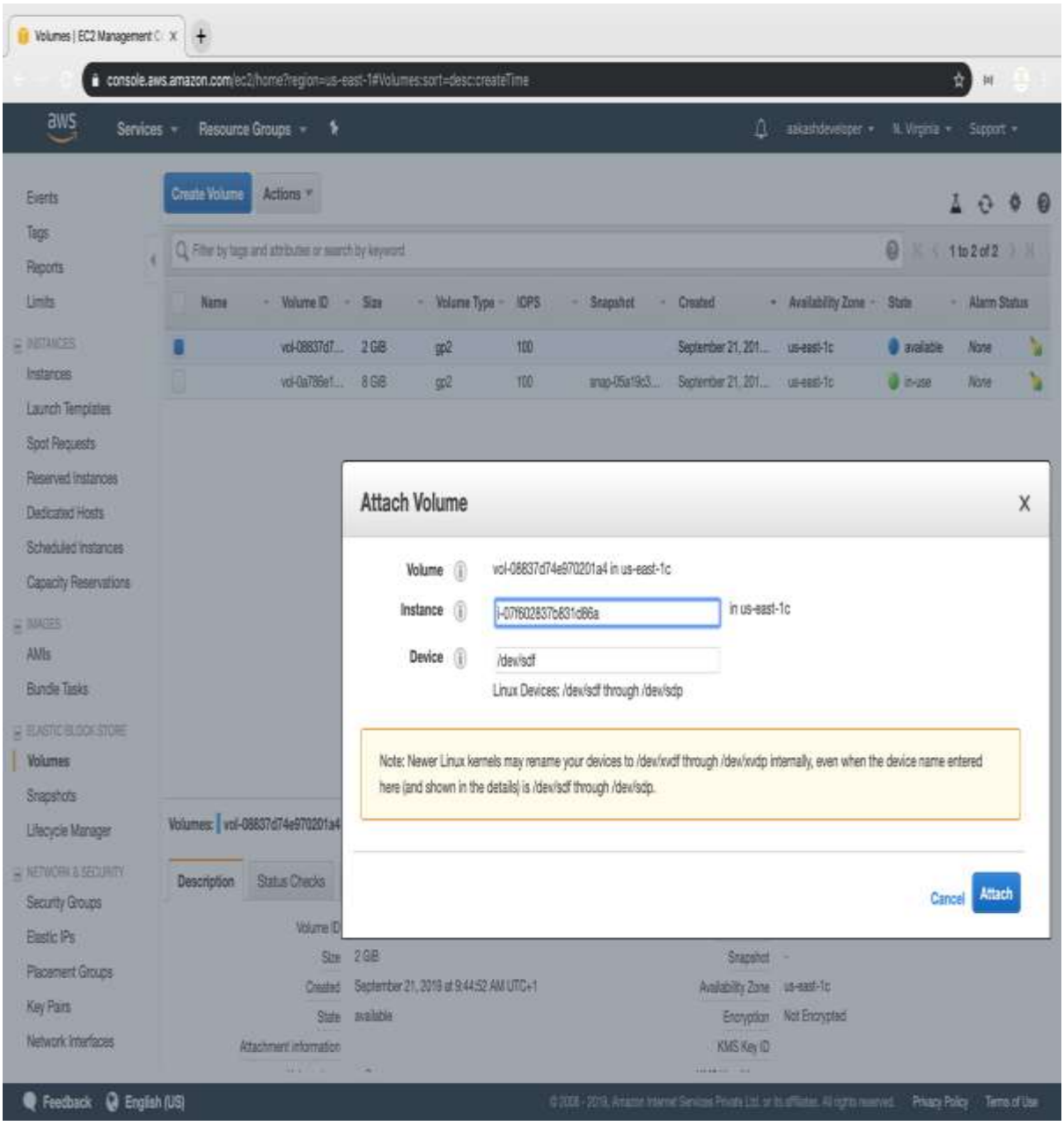
Step 3.2.3: Providing type, size, availability zone, and snapshot name on the **create volume** tab



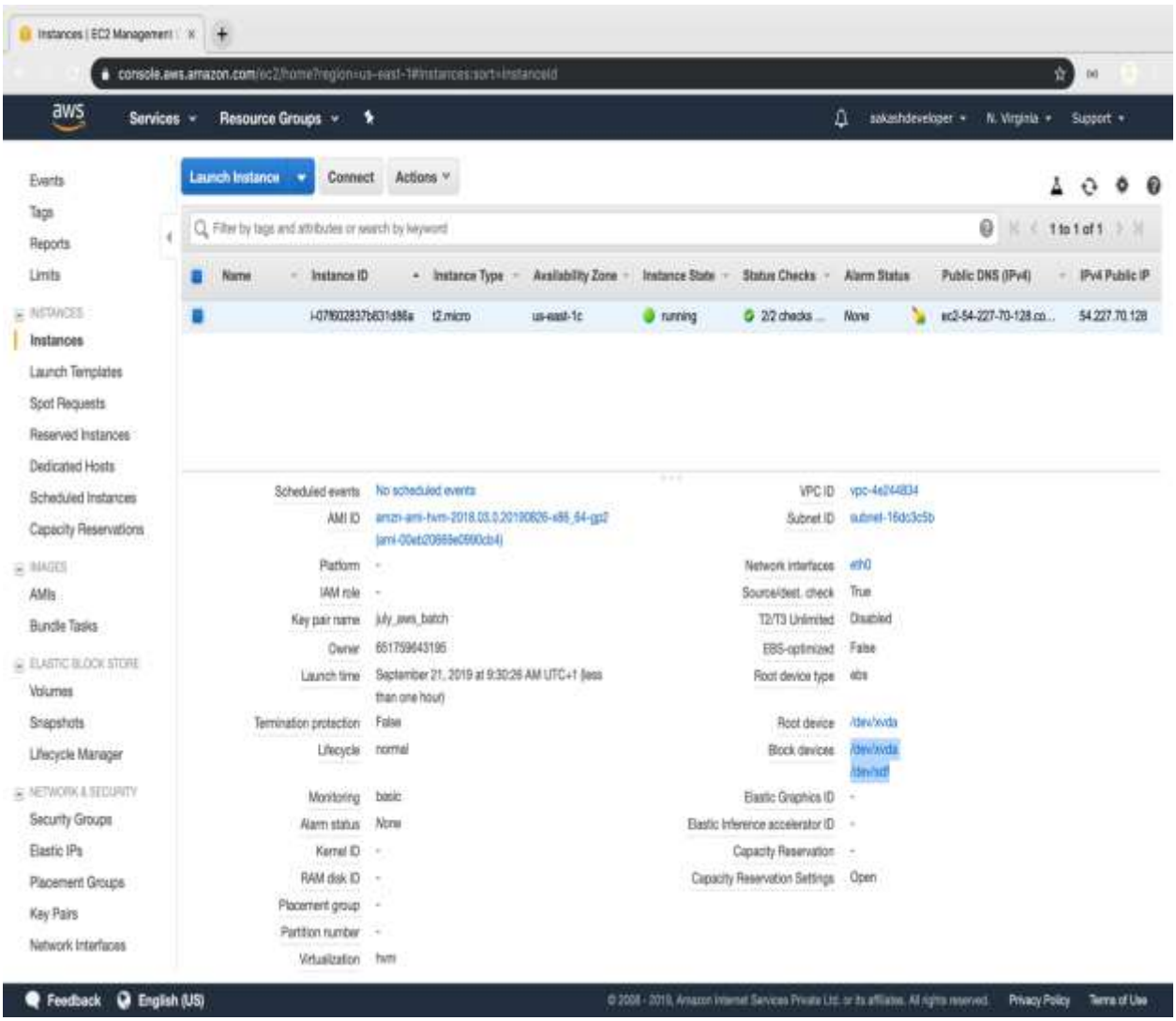
Step 3.2.4: Selecting newly created volume and clicking on the action to attach the volume to the instance



Step 3.2.5: Selecting the instance of the same availability zone and attaching the volume in pop



Step 3.2.6: Verifying the instance by adding one more device to the block device



Step 3.2.7: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

3.9 Launch an Instance in a Placement Group

This section will guide you to:

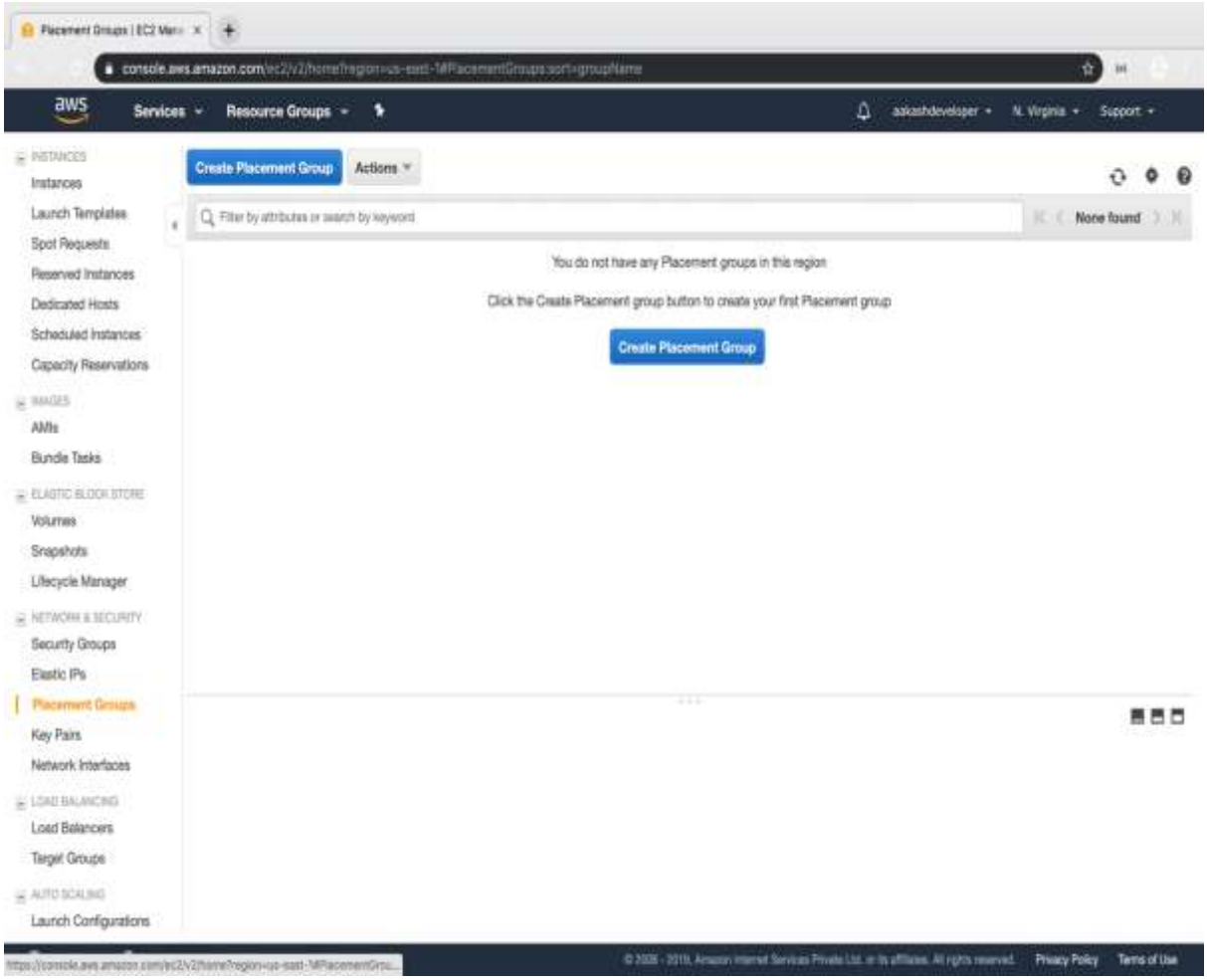
- Launch an instance in a placement group

This lab has two subsections, namely:

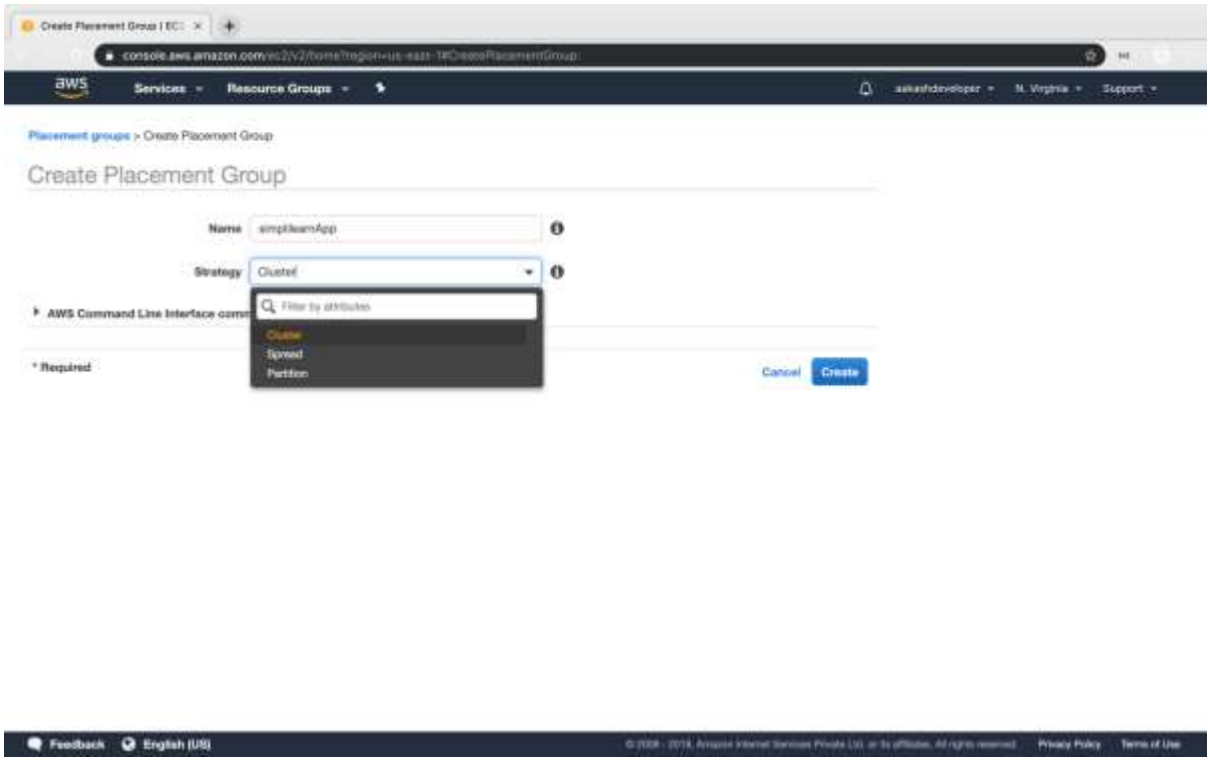
4.3.1 Creating a placement group
4.3.2 Launching an instance in the placement group

Step 4.3.1: Creating a placement group

- Click on *Create Placement Group*

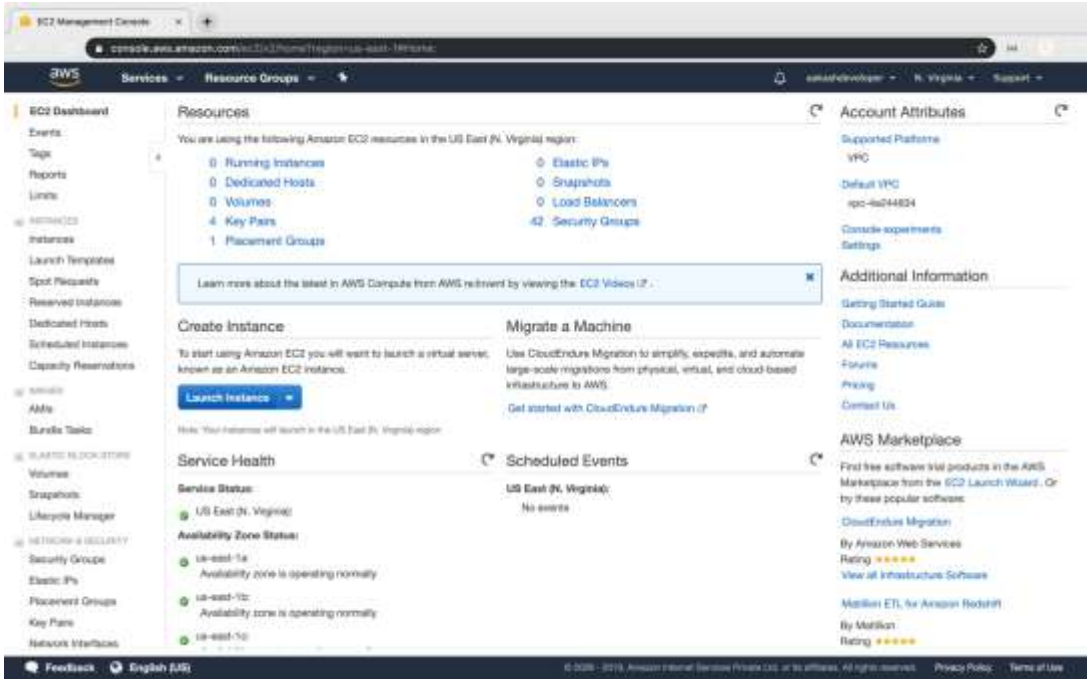


- Provide a name and strategy

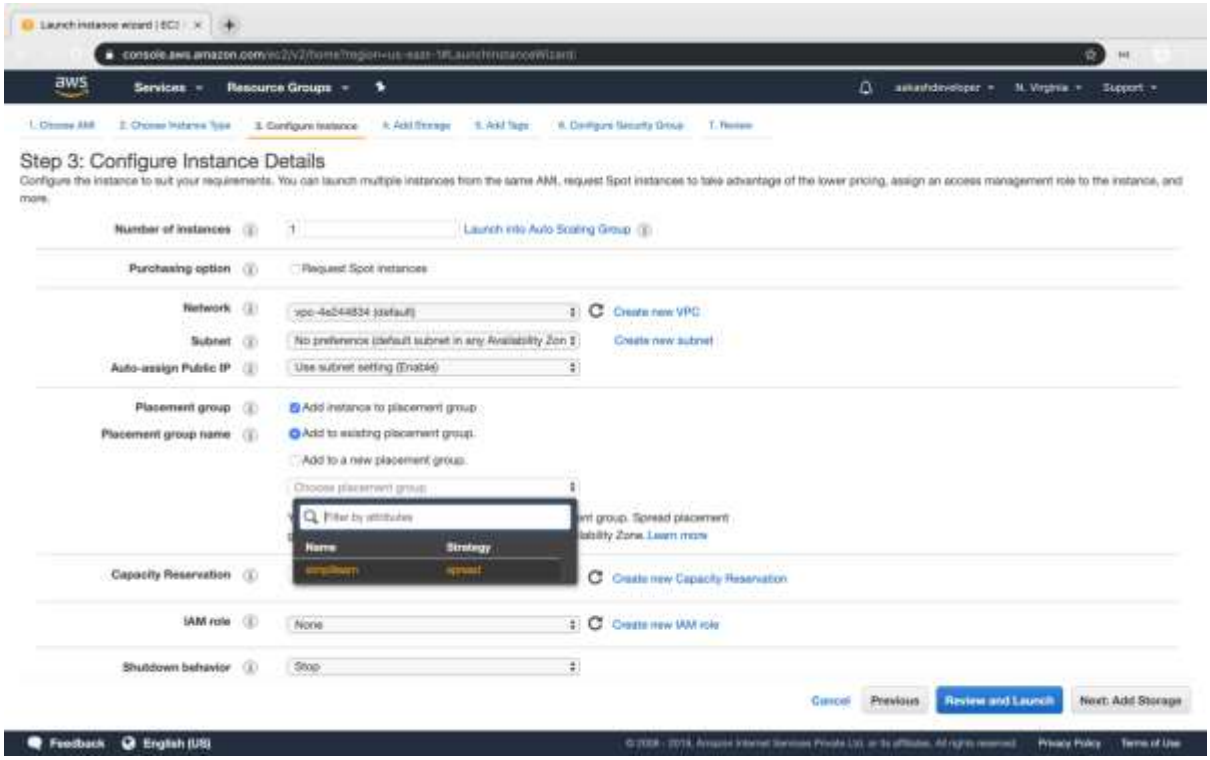


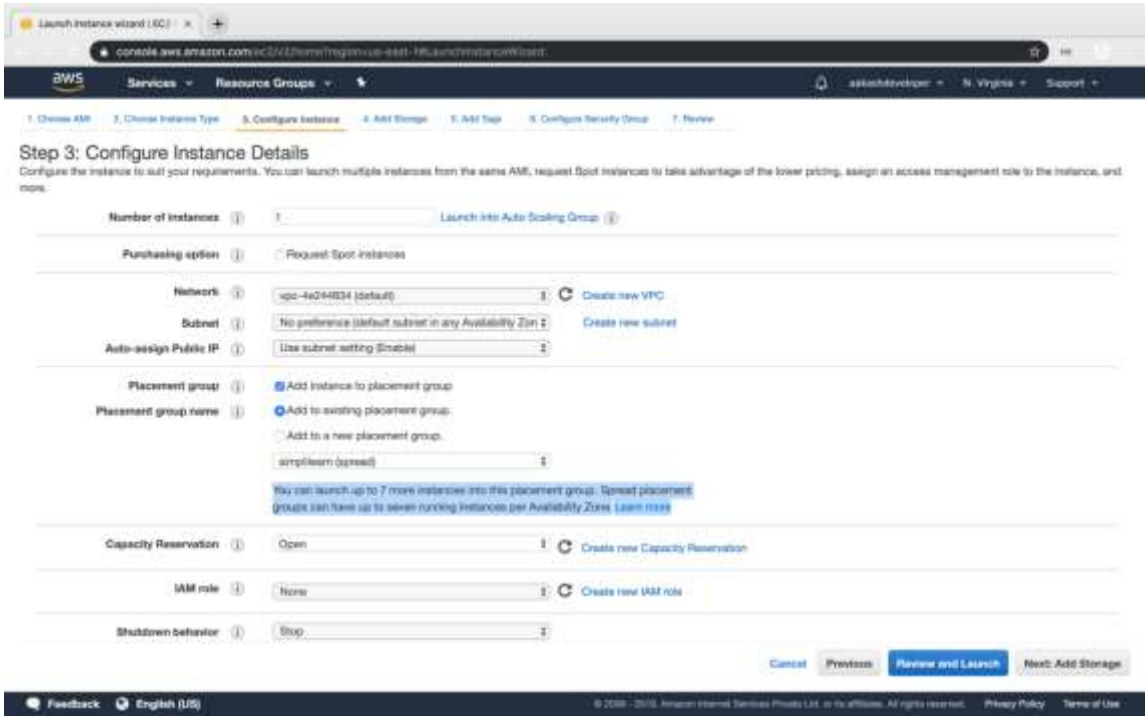
Step 4.3.2: Launching an instance in the placement group

- Launch the new EC2 instance

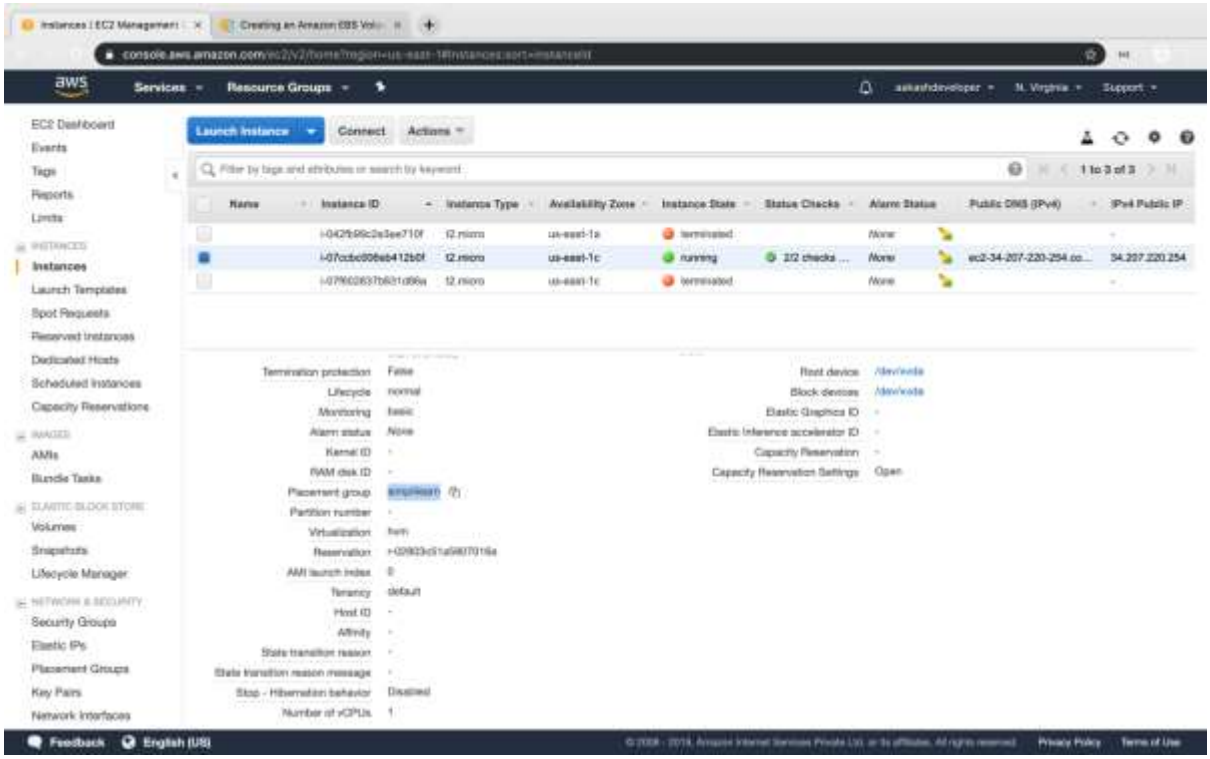


- Select the placement group





- Verify the placement group of the instance


Launch Instance
Connect
Actions
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
IPv4 Public IP
i-0425d9c2a3ee710f
t2.micro
us-east-1a
terminated
None
i-07cdc606eb412b0f
t2.micro
us-east-1c
running
2/2 checks ...
None
ec2-34-207-220-254.co...
34.207.220.254
i-07f602837b631d96a
t2.micro
us-east-1c
terminated
None
Termination protection
False
Lifecycle
normal
Monitoring
basic
Alarm status
None
Kernel ID
RAM disk ID
Placement group
simplilearn
Partition number
Virtualization
hvm
Reservation
AMI launch index
0
Tenancy
default
Host ID
Affinity
State transition reason
State transition reason message
Stop - Hibernation behavior
Disabled
Number of vCPUs
1
Root device
Block devices
Elastic Graphics ID
Elastic Inference accelerator ID
Capacity Reservation
Capacity Reservation Settings
Open
Feedback
English (US)
Privacy Policy
Terms of Use

Assisted Practice: 3.13 Create an EBS Volume

This section will guide you to:

- Change the volume size of an instance

This lab has six subsections, namely:

3.4.1 Selecting the **volume** tab in the EC2 dashboard

3.4.2 Creating a new volume by clicking on **Create**

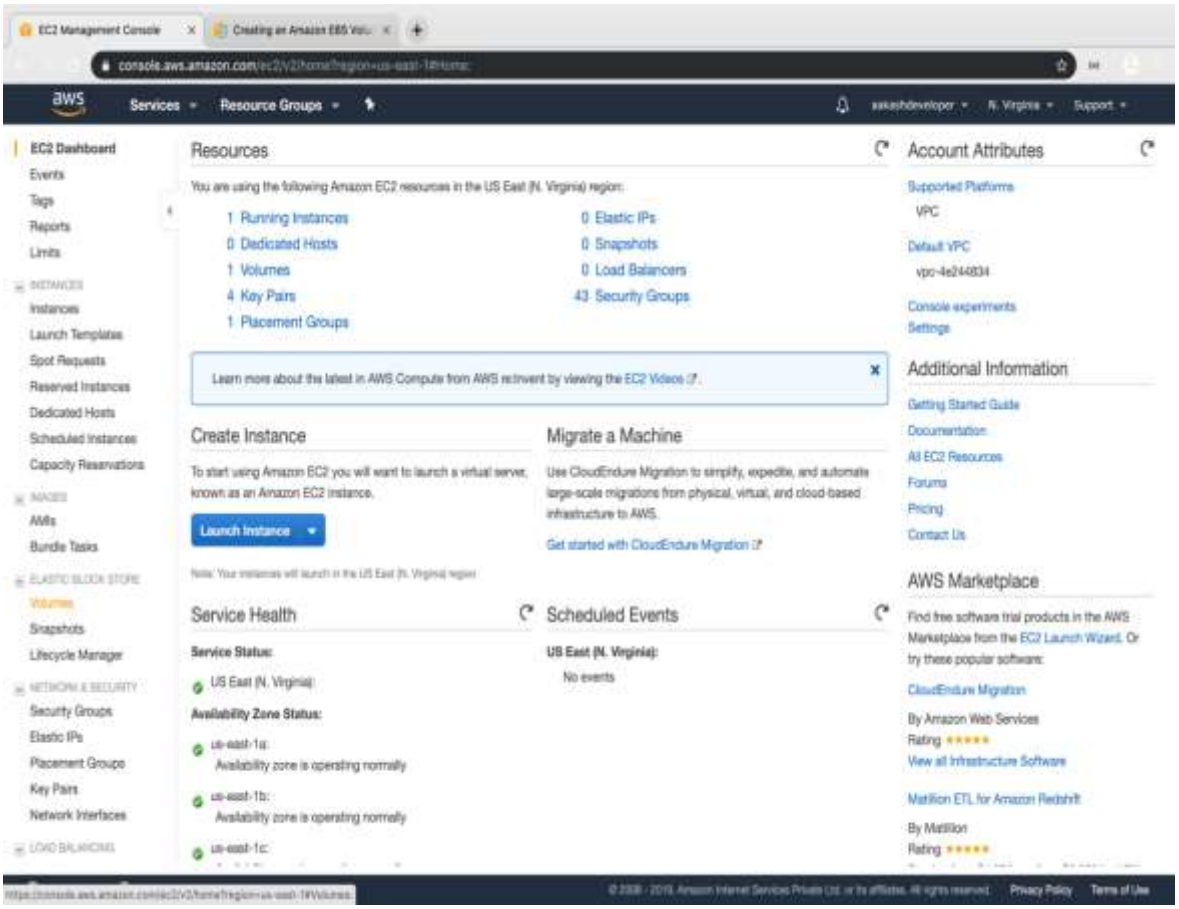
3.4.3 Selecting the volume type according to the project requirement

3.4.4 Creating a new volume by specifying the size and defining the zone

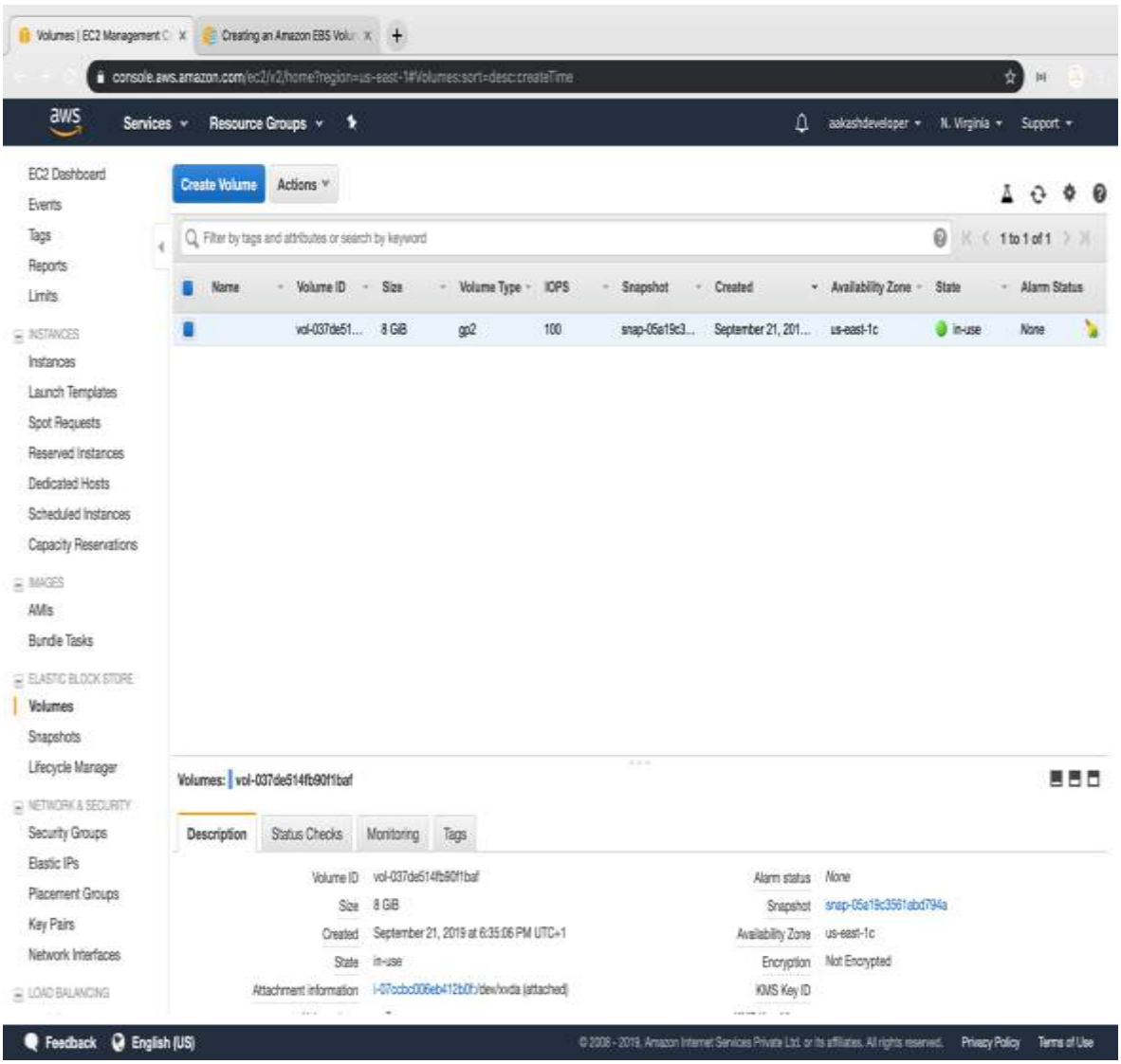
3.4.5 Verifying the new volume created on the volume console and its availability

3.4.6 Pushing the code to GitHub repositories

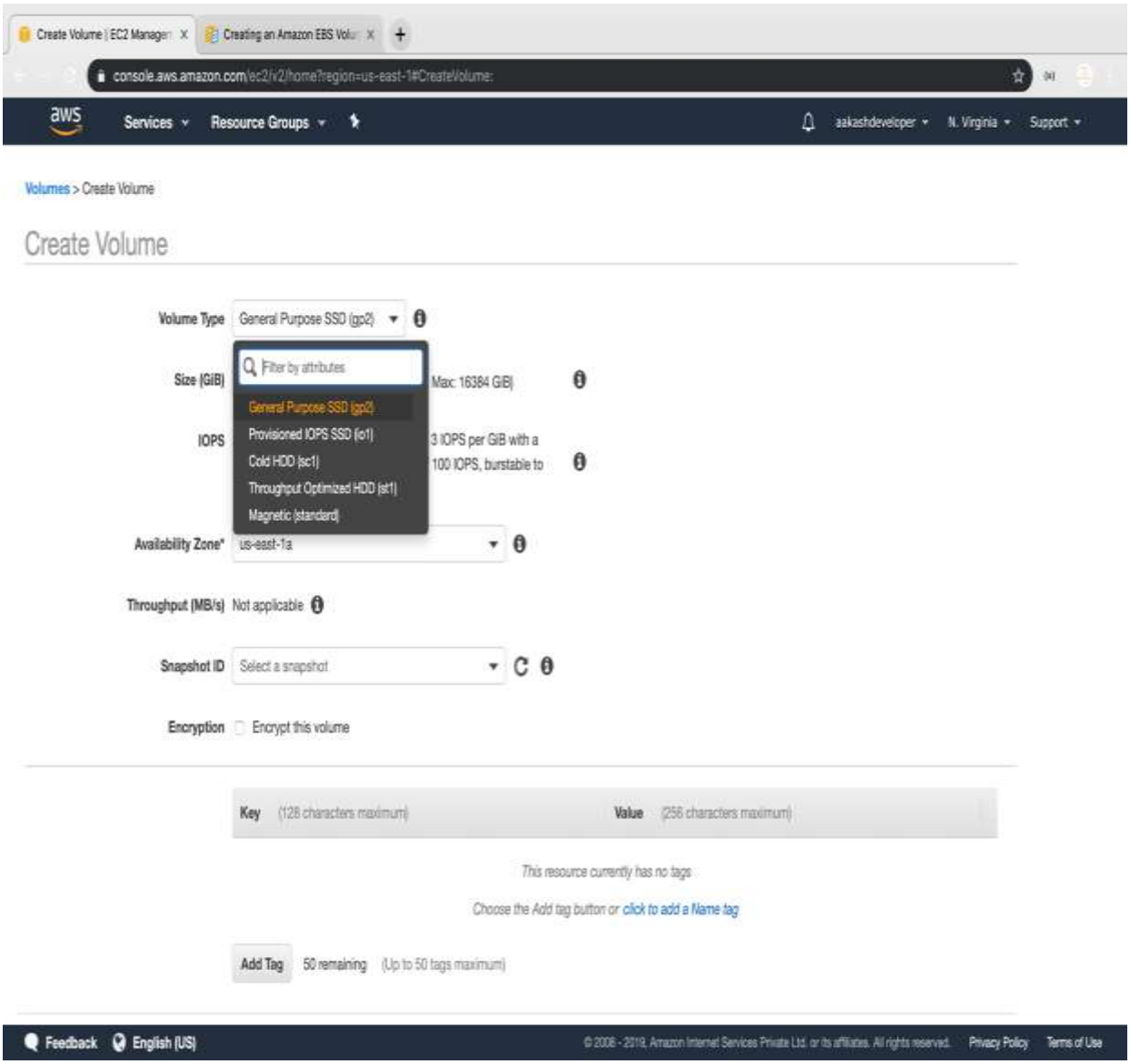
Step 3.4.1: Selecting the **volume** tab in the EC2 dashboard



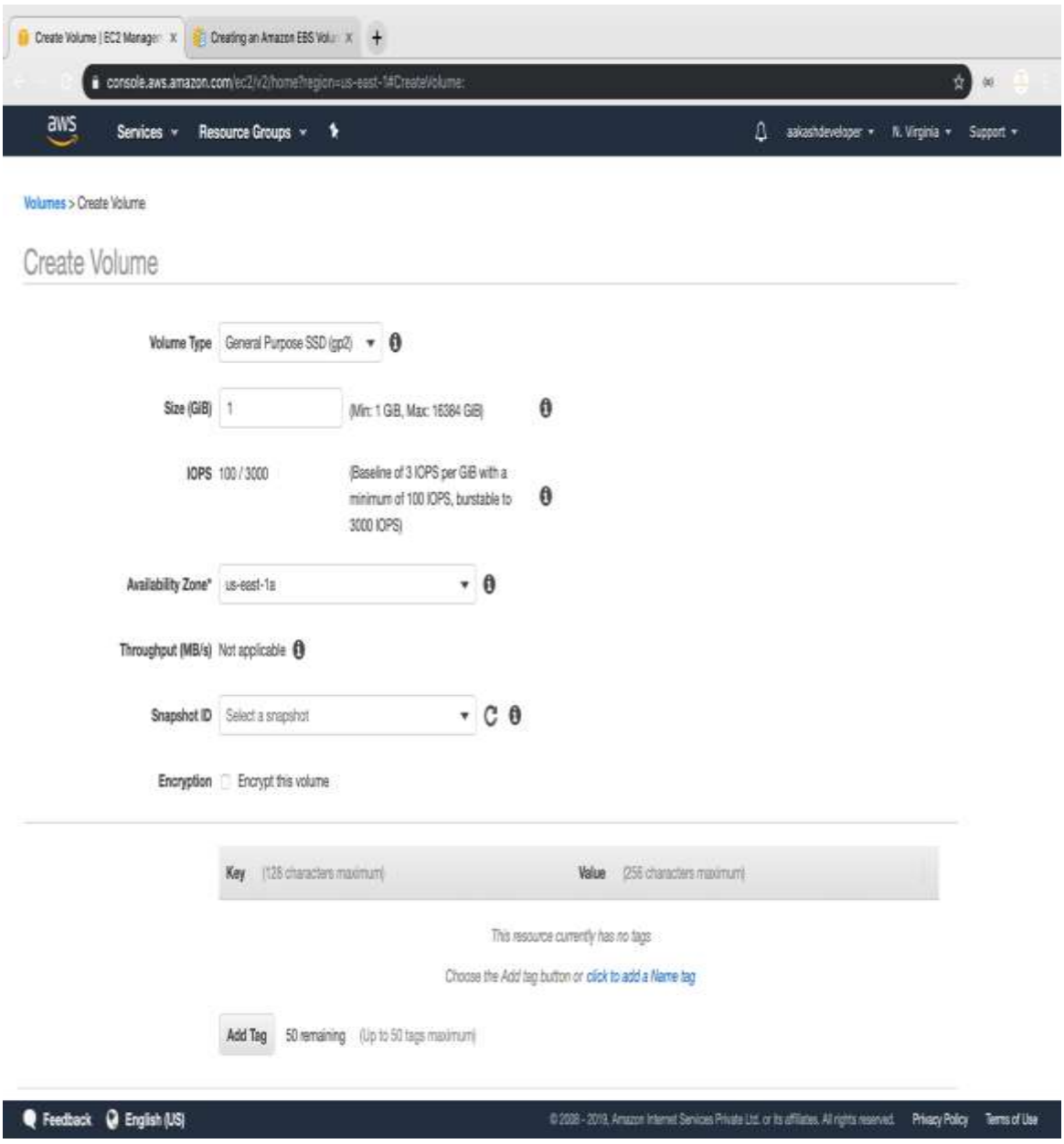
Step 3.4.2: Creating a new volume by clicking on **Create**



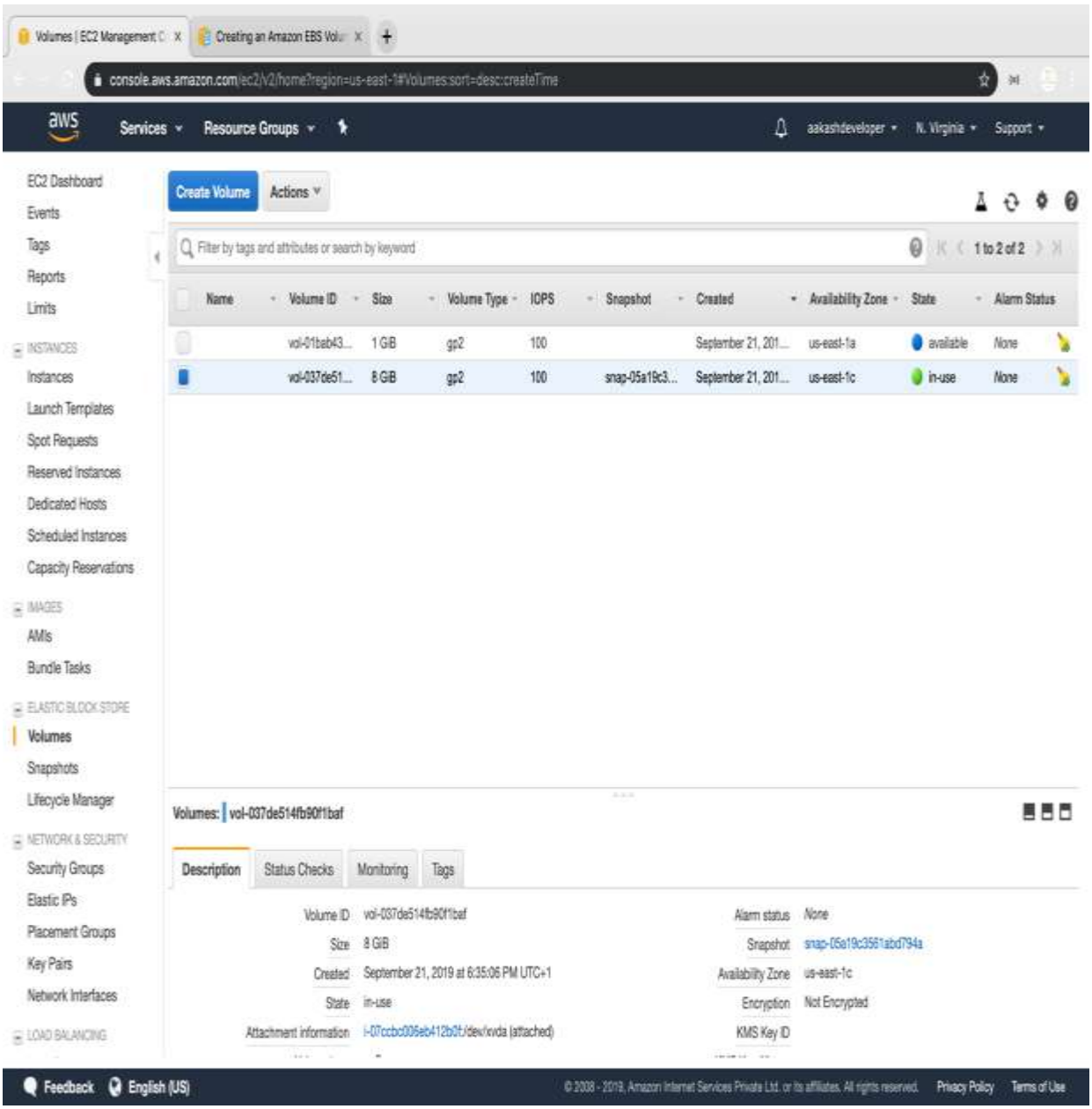
Step 3.4.3: Selecting the volume type according to the project requirement



Step 3.4.4: Creating a new volume by specifying the size and defining the zone



Step 3.4.5: Verifying the new volume created on the volume console and its availability



Step 3.4.6: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.15 Format and Mount an EBS Volume

This section will guide you to:

- Format and mount an EBS volume

This lab has four subsections, namely:

3.5.1 Attaching an existing EBS volume

3.5.2 Mounting the EBS volume

3.5.3 Unmounting the EBS volume

3.5.4 Pushing the files to GitHub repositories

Step 3.5.1: Attaching an existing EBS volume

- Select an existing volume and click on *Attach Volume*
- Select the instance

Step 3.5.2: Mounting the EBS volume

- Login to your EC2 instance and list the available disks using the following command:

  ldblk

- Use the following command to check if the volume has any data:

  sudo file -s /dev/xvdf

- Use the following command to format the volume to ext4 filesystem:

  sudo mkfs -t ext4 /dev/xvdf

- Use the following command to create a directory to mount the volume:

  sudo mkdir /newvolume

- Use the following command to mount the volume:

  sudo mount /dev/xvdf /newvolume/

- Check the disk space using the following command:

  cd /newvolume

  df -h .

Step 3.5.3: Unmounting the EBS volume

- Use the following command to unmount the volume:

  umount /dev/xvdf

Step 3.5.4: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.16 Detach an EBS Volume

This section will guide you to:

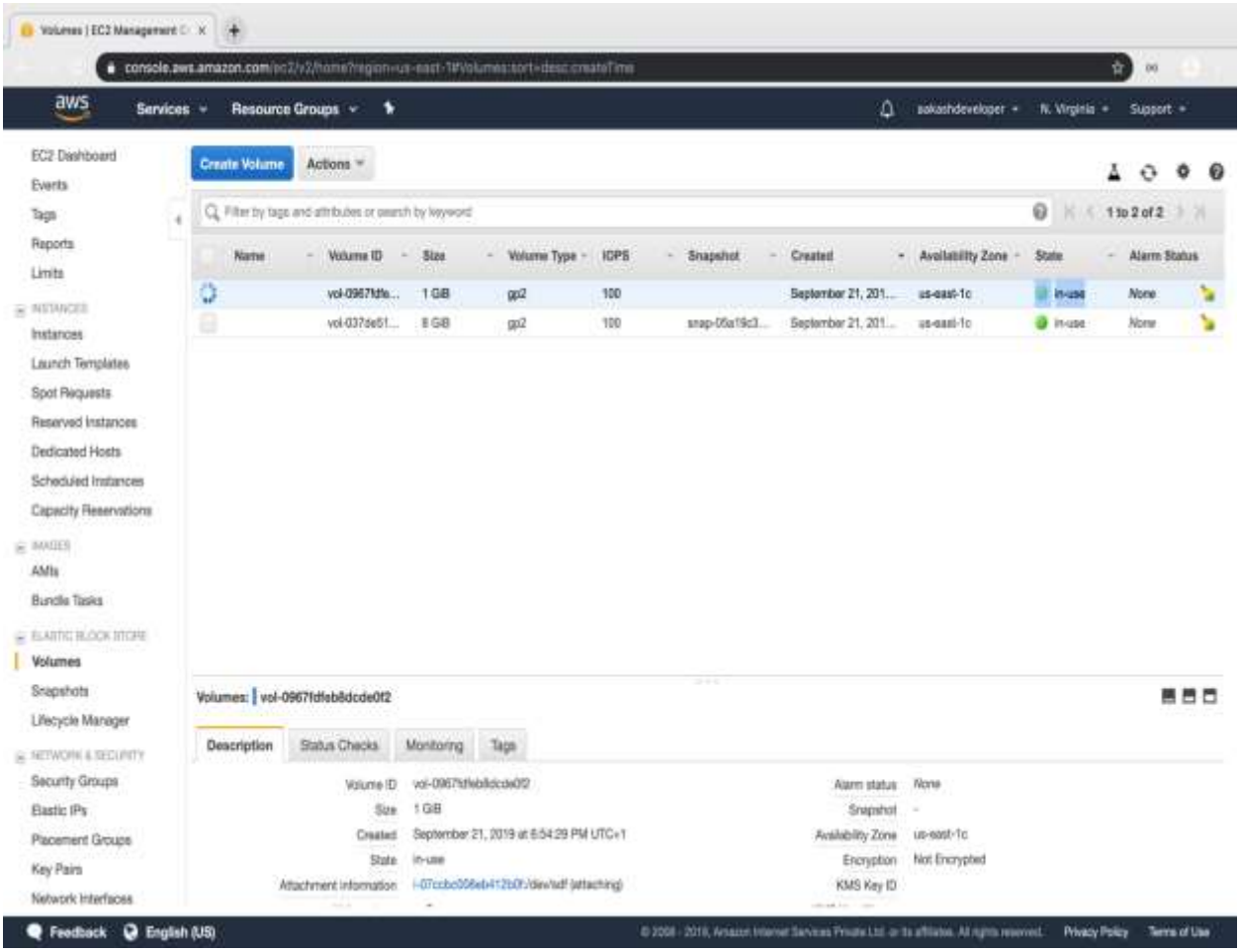
- Detach an EBS volume from an instance

This lab has three subsections, namely:

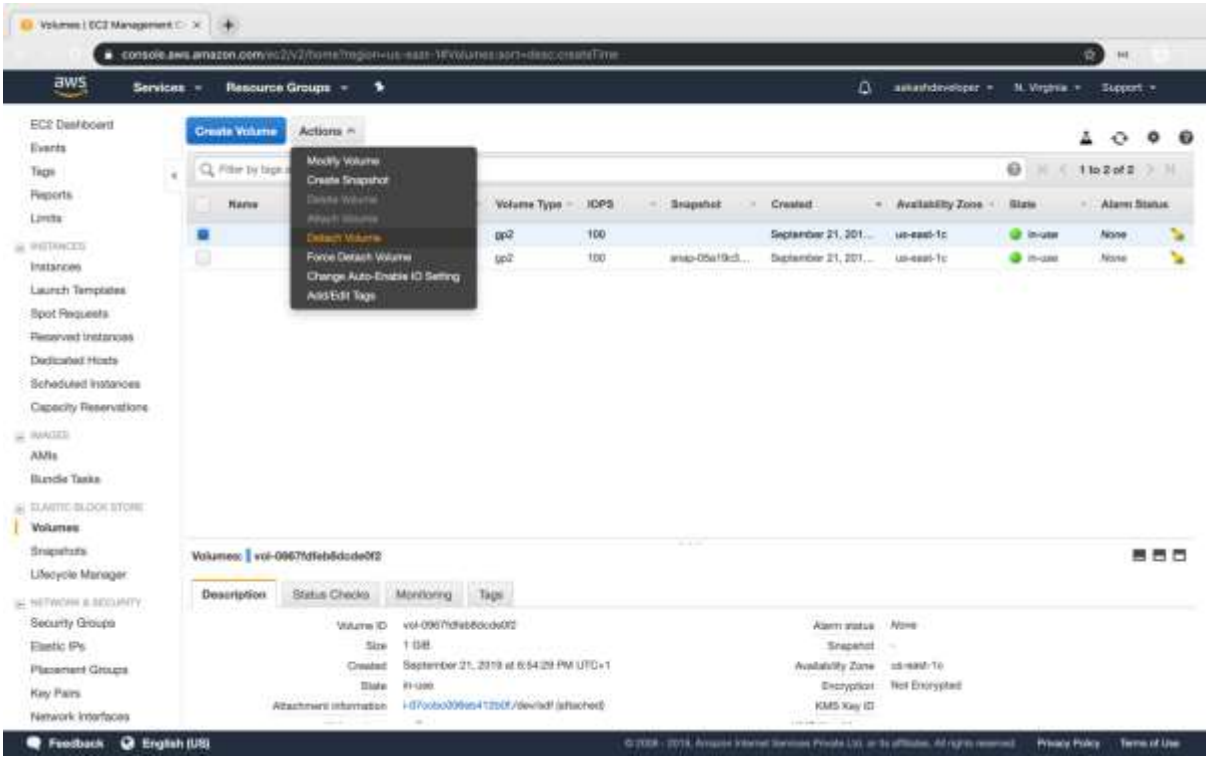
3.6.1 Selecting the EBS volume you want to detach
3.6.2 Detaching the volume
3.6.3 Pushing the files to GitHub repositories

Step 3.6.1: Selecting the EBS volume you want to detach

- Choose the bucket

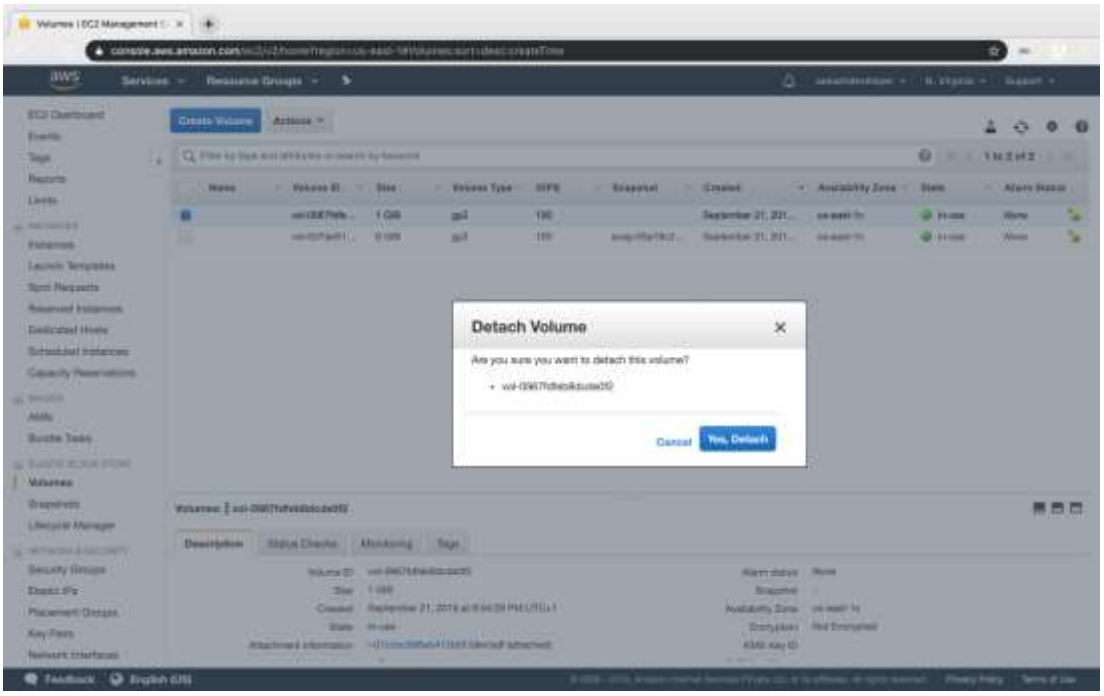


- Select *Detach Volume* option

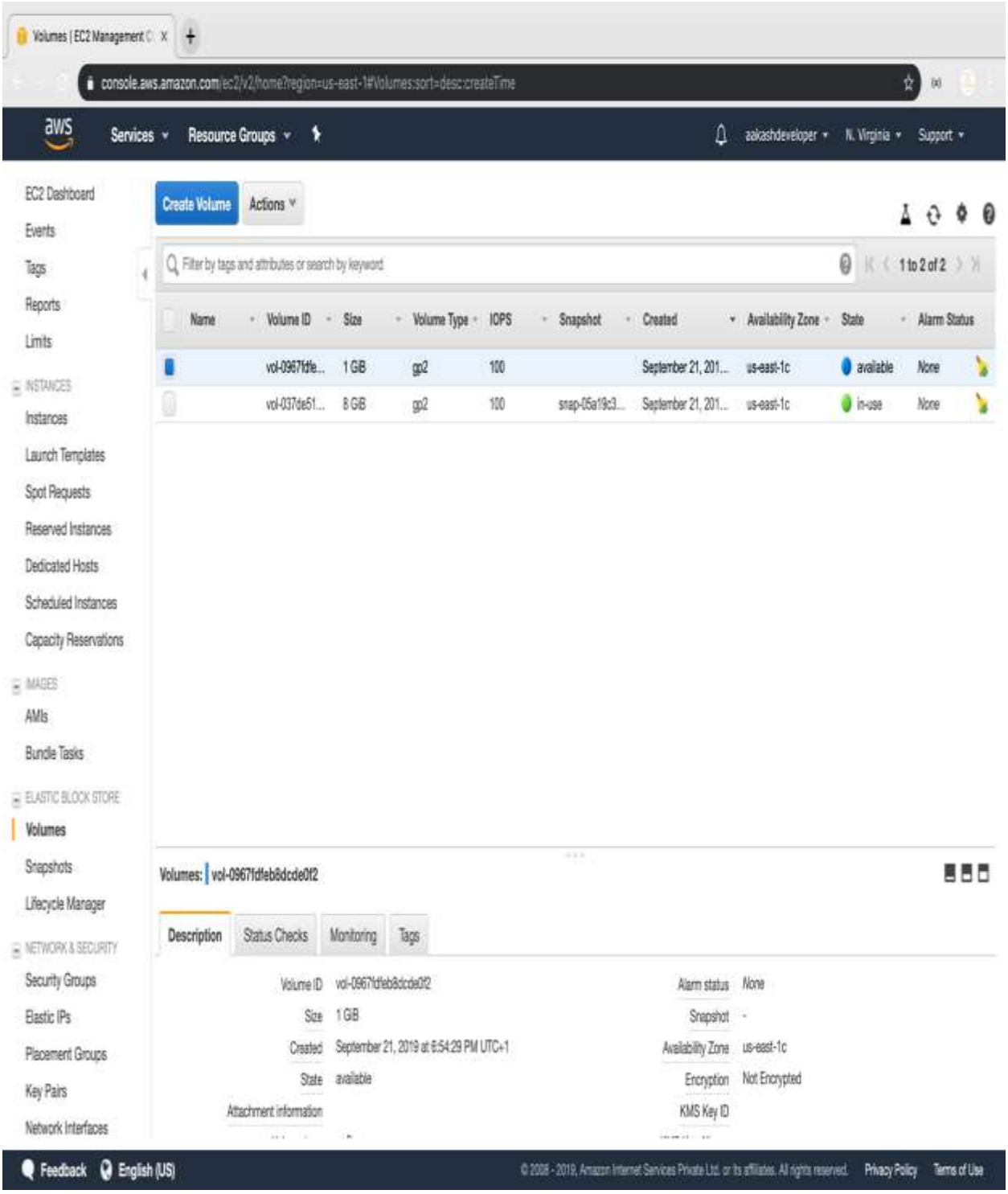


Step 3.6.2: Detaching the volume

- Select *Yes* once the confirmation pop-up appears



- Once detached, the volume will be labeled as *Available*



Step 3.6.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.17 Delete an EBS Volume

This section will guide you to:

- Delete an EBS volume

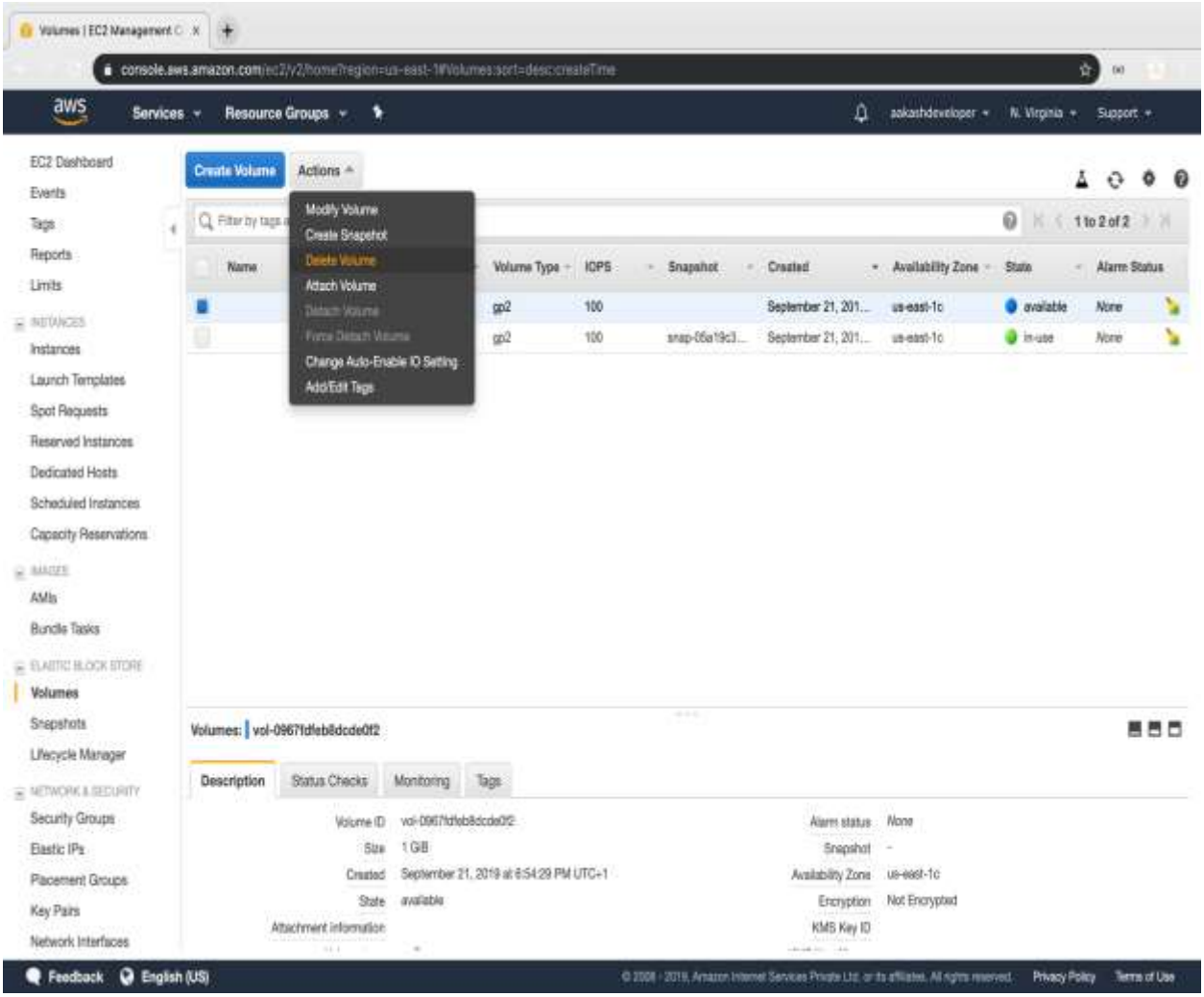
This lab has three subsections, namely:

3.7.1 Selecting the EBS volume you want to delete

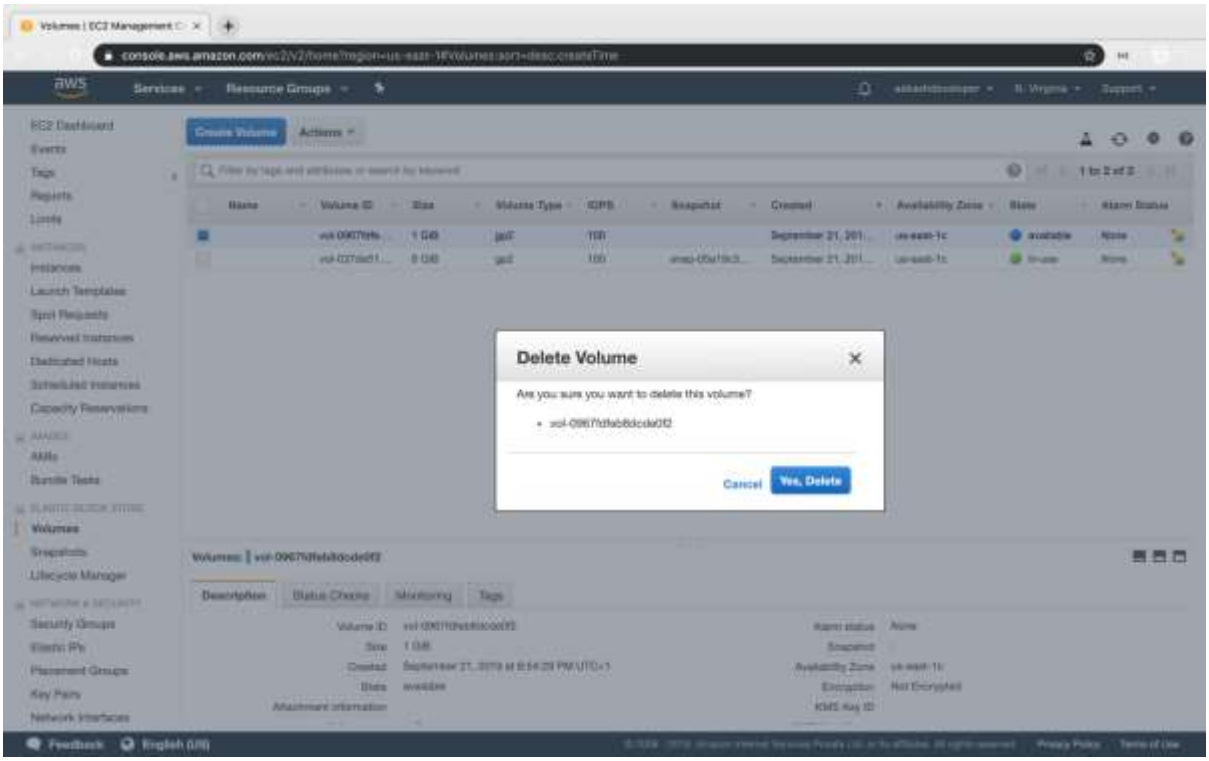
3.7.2 Deleting the volume

3.7.3 Pushing the file to GitHub repositories

Step 3.7.1: Selecting the EBS volume you want to delete



Step 3.7.2: Deleting the volume



Step 3.7.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.20 Create an EBS Snapshot

This section will guide you to:

- Create an EBS snapshot

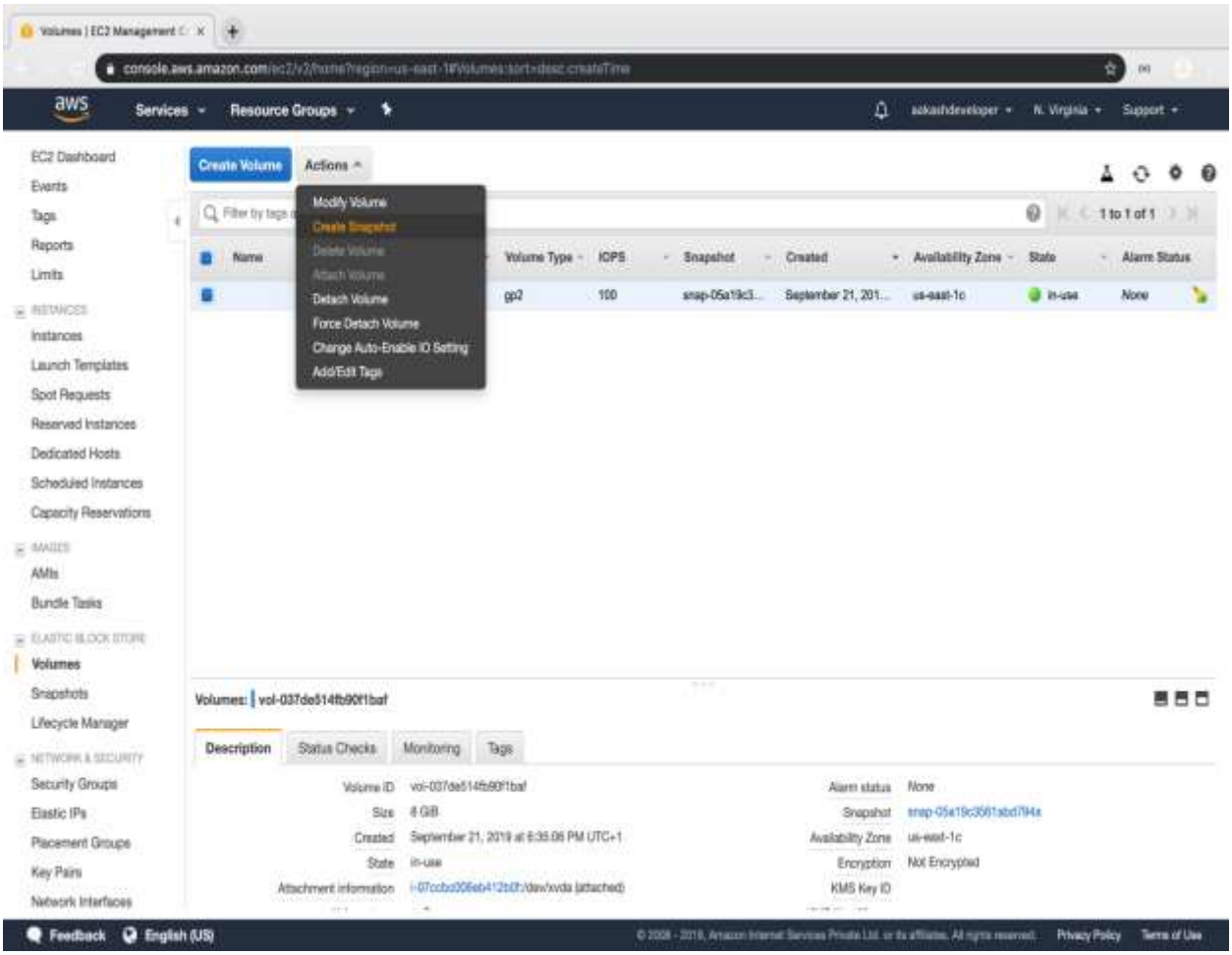
This lab has two subsection, namely:

3.8.1 Creating an EBS snapshot

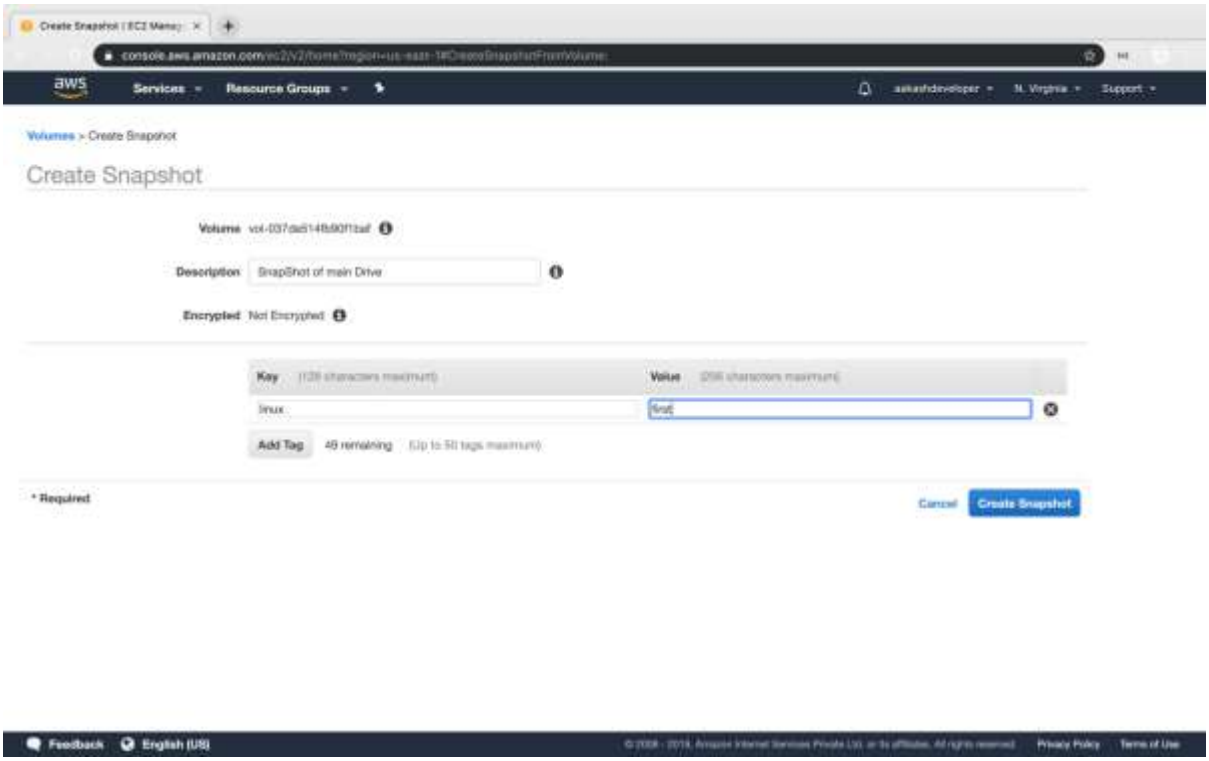
3.8.2 Pushing the file to GitHub repositories

Step 3.8.1: Creating an EBS snapshot

- Select *Volume* in EC2 dashboard
- Click on *Action -> Create Snapshot*



- Add snapshot name, key, and value



Step 3.8.2: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

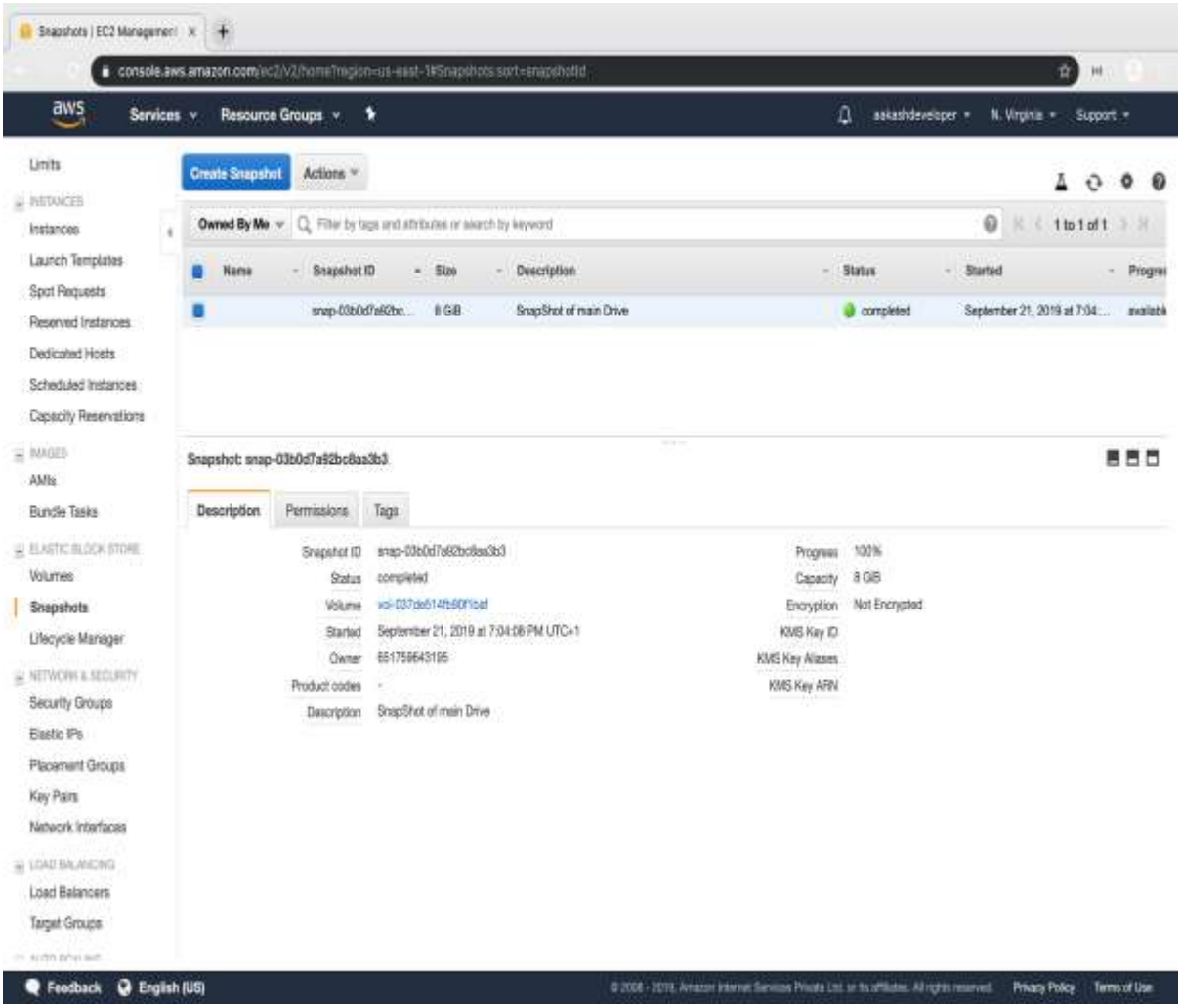
- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.21 View snapshot

This section will guide you to view a snapshot in your AWS account.

Step 1: To view snapshots, you need to navigate to the EC2 **dashboard** available on the panel on the left. Click on **snapshot** to view all available snapshots.



Step 2: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.22 Initialize a Volume Restored from a Snapshot on Linux

This section will guide you to:

- Initialize a volume restored from a snapshot

This lab has three subsections, namely:

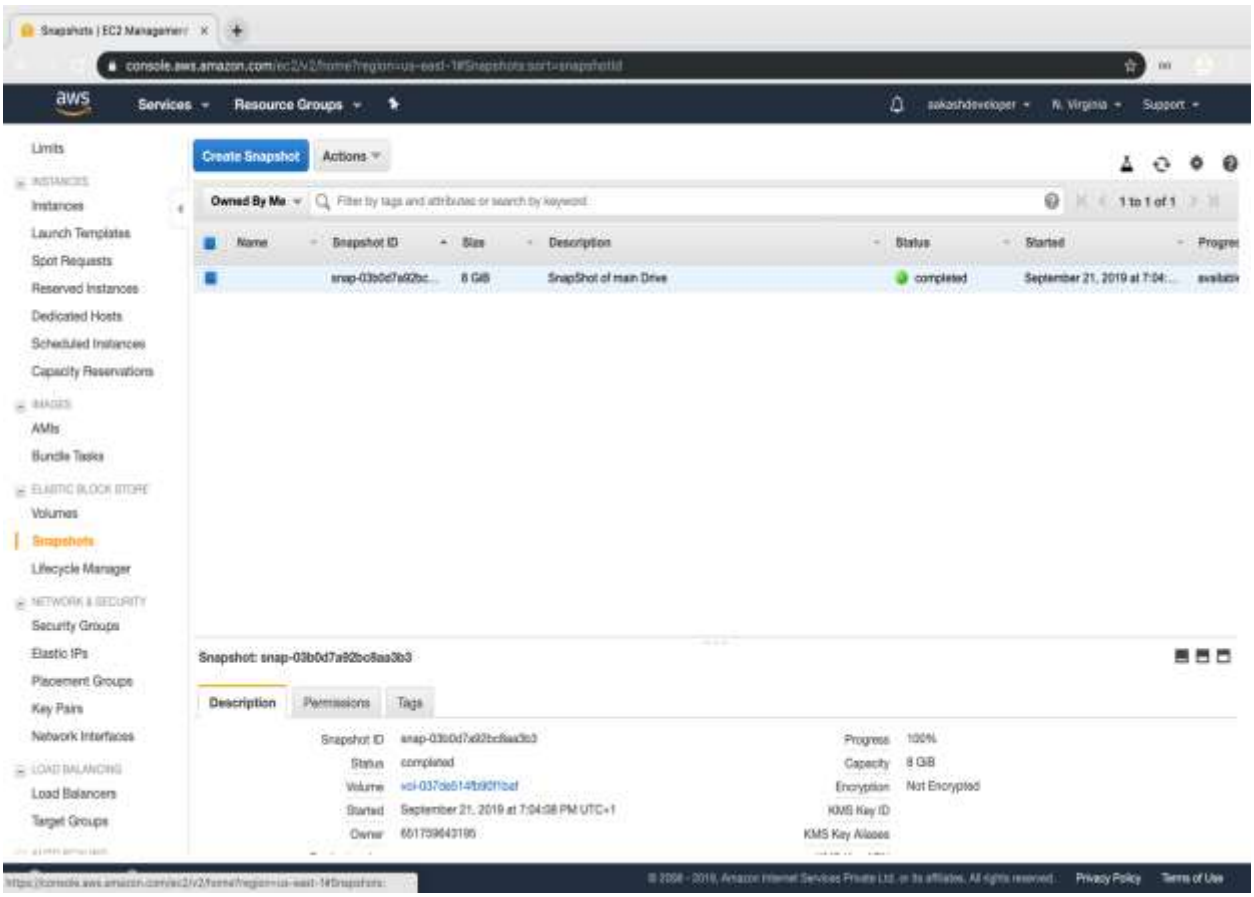
3.10.1 Selecting the snapshot

3.10.2 Restoring volume from the snapshot

3.10.3 Pushing the files to GitHub repositories

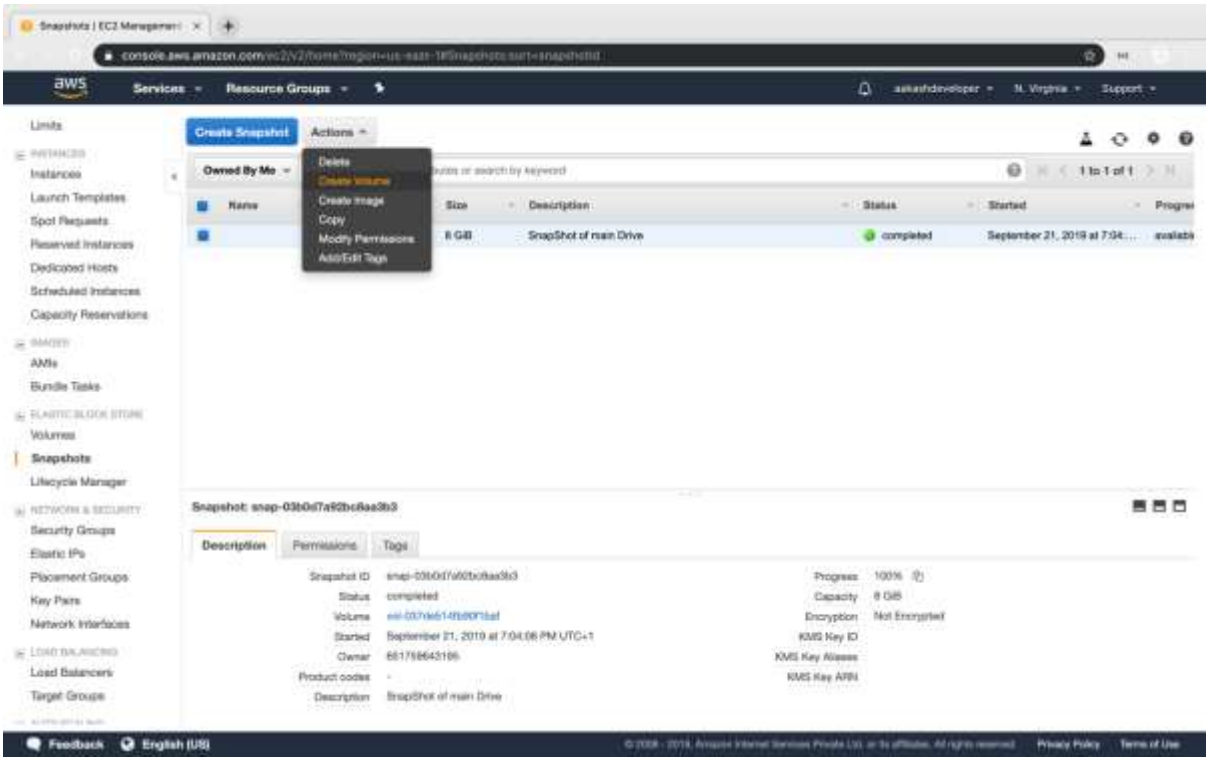
Step 3.10.1: Selecting the snapshot

- Go to EC2 dashboard in EBS and click on *Snapshot*

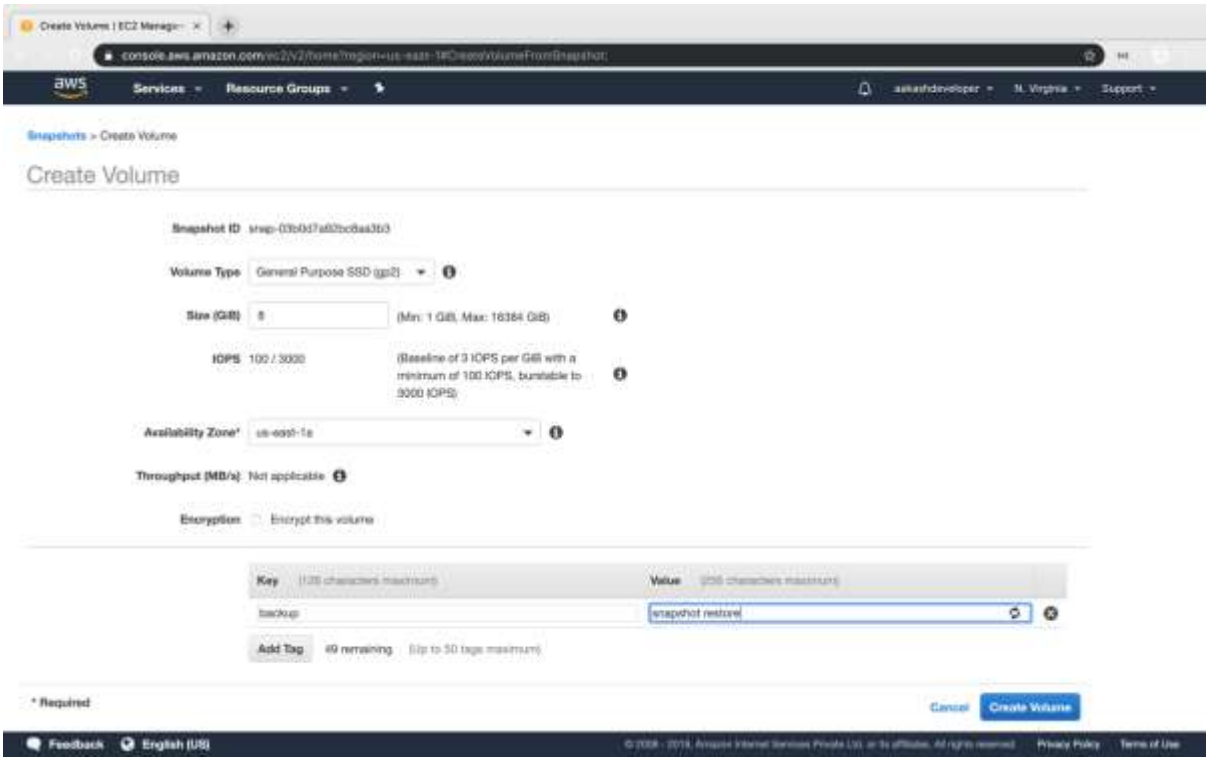


Step 3.10.2: Restoring volume from the snapshot

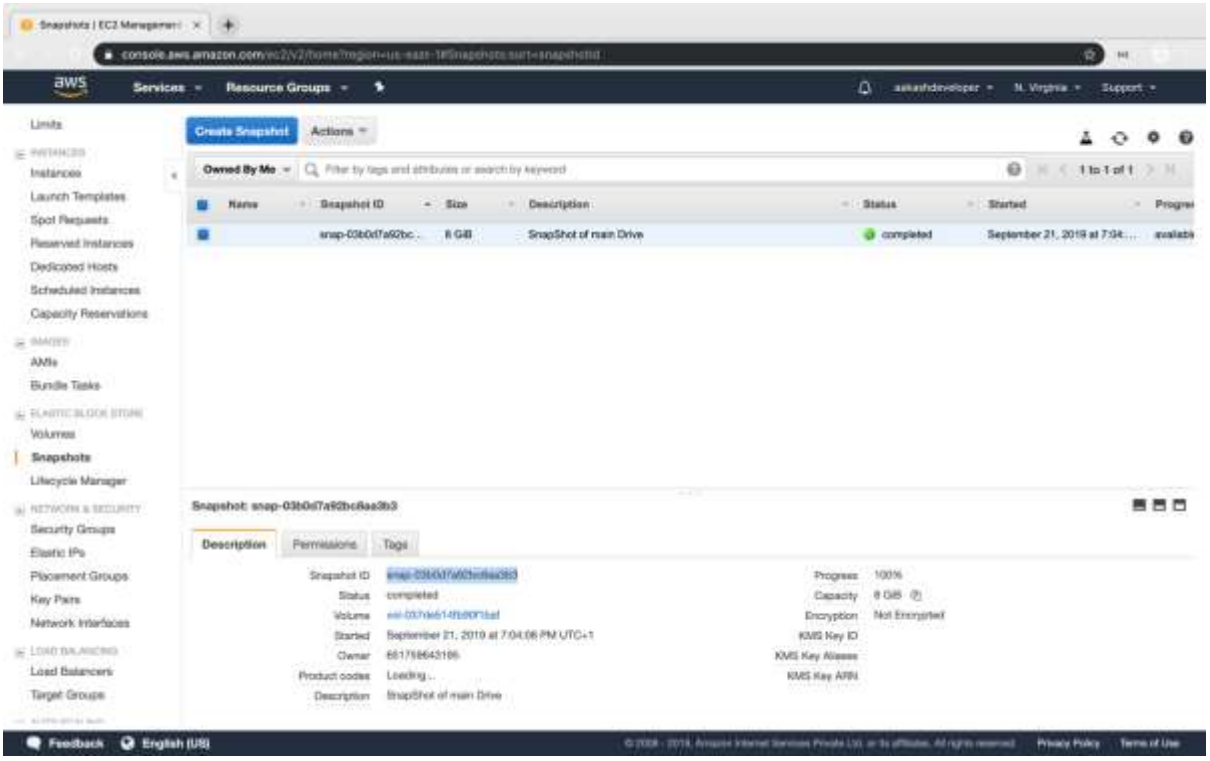
- Select the snapshot and click on *Action*



- You can specify the key and value to identify the volume



- Restore the volume from the snapshot



Step 3.10.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.25 Create a Bucket

This section will guide you to:

- Create a bucket

This lab has eight subsections, namely:

3.11.1 Selecting S3 from the storage of the AWS service panel

3.11.2 Verifying the button to create a bucket

3.11.3 Providing a bucket name and region

3.11.4 Entering key and value for identification and tracking the bucket

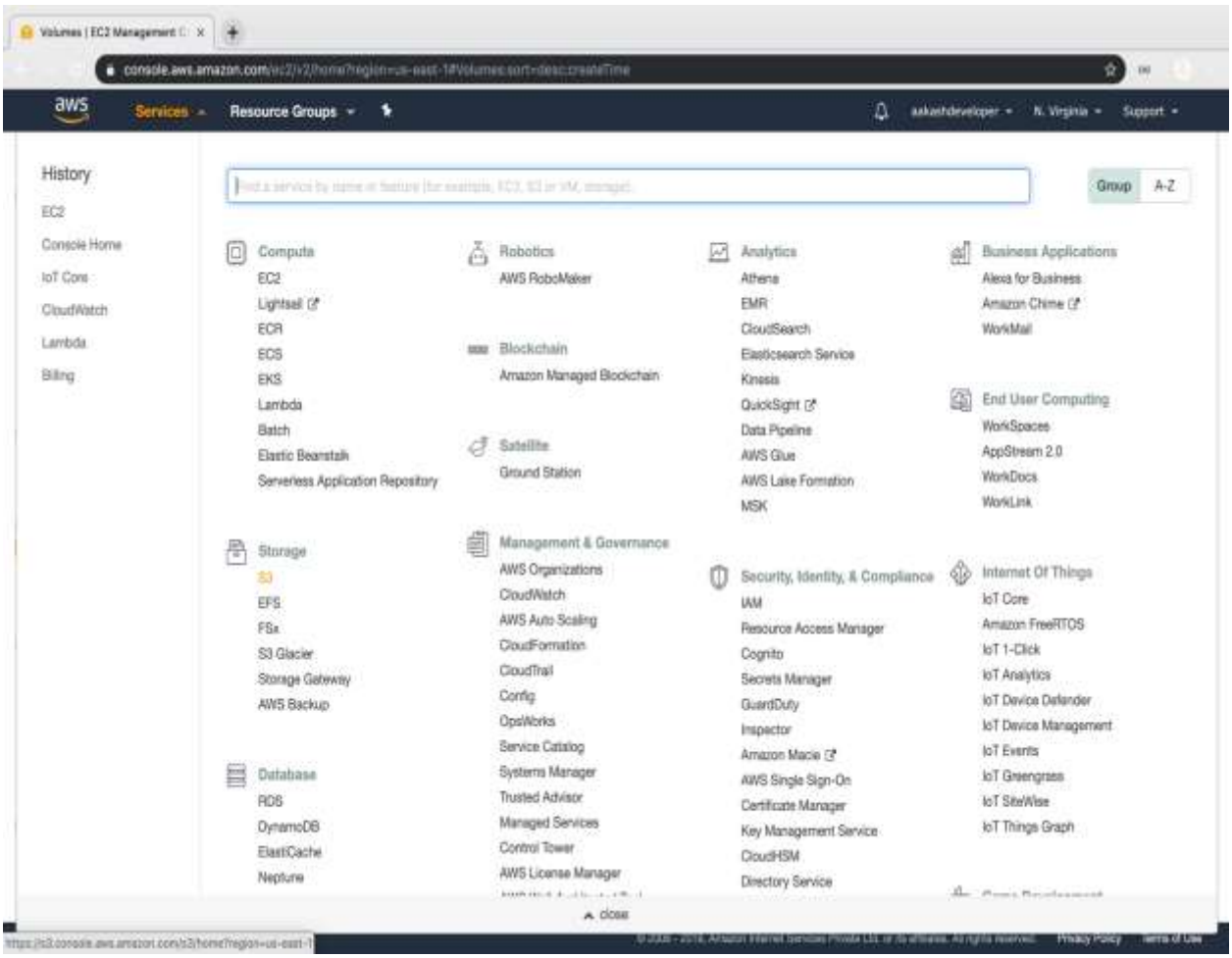
3.11.5 Blocking all the public access

3.11.6 Reviewing all parameters and creating the bucket

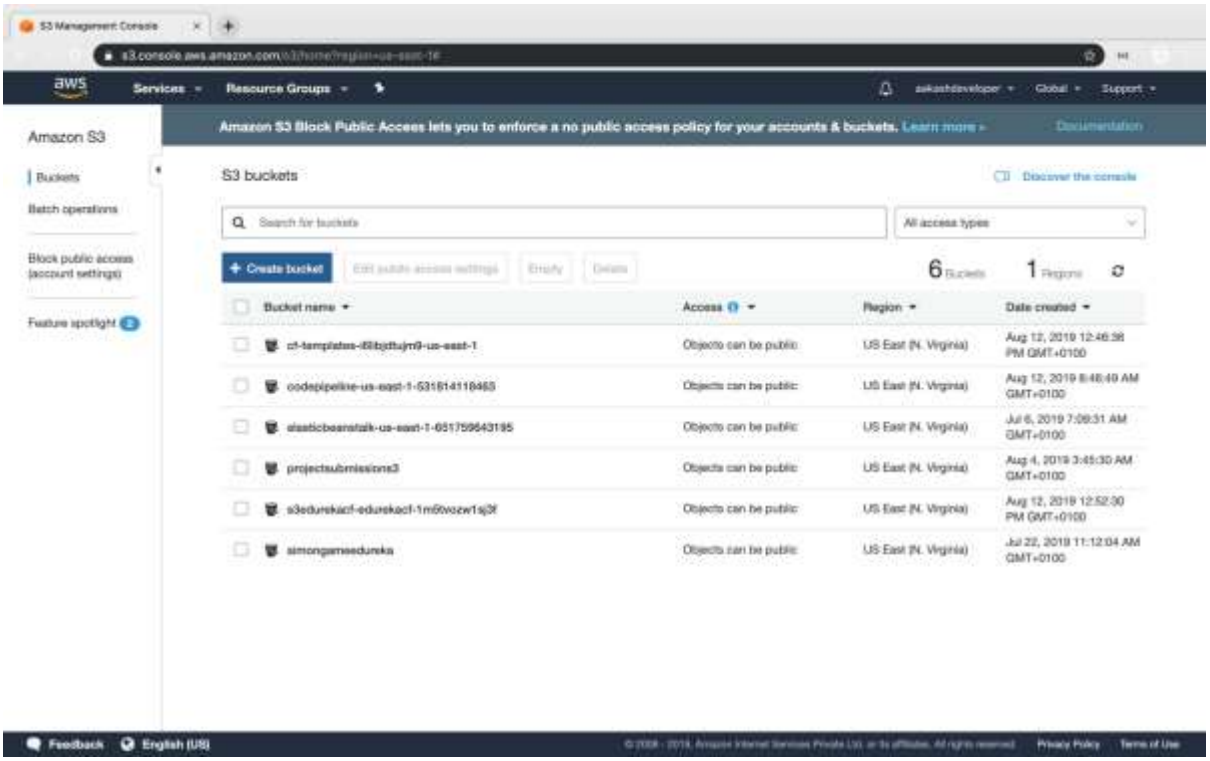
3.11.7 Verifying the newly created bucket on the panel of S3

3.11.8 Pushing the files to GitHub repositories

Step 3.11.1: Selecting S3 from the storage of the AWS service panel

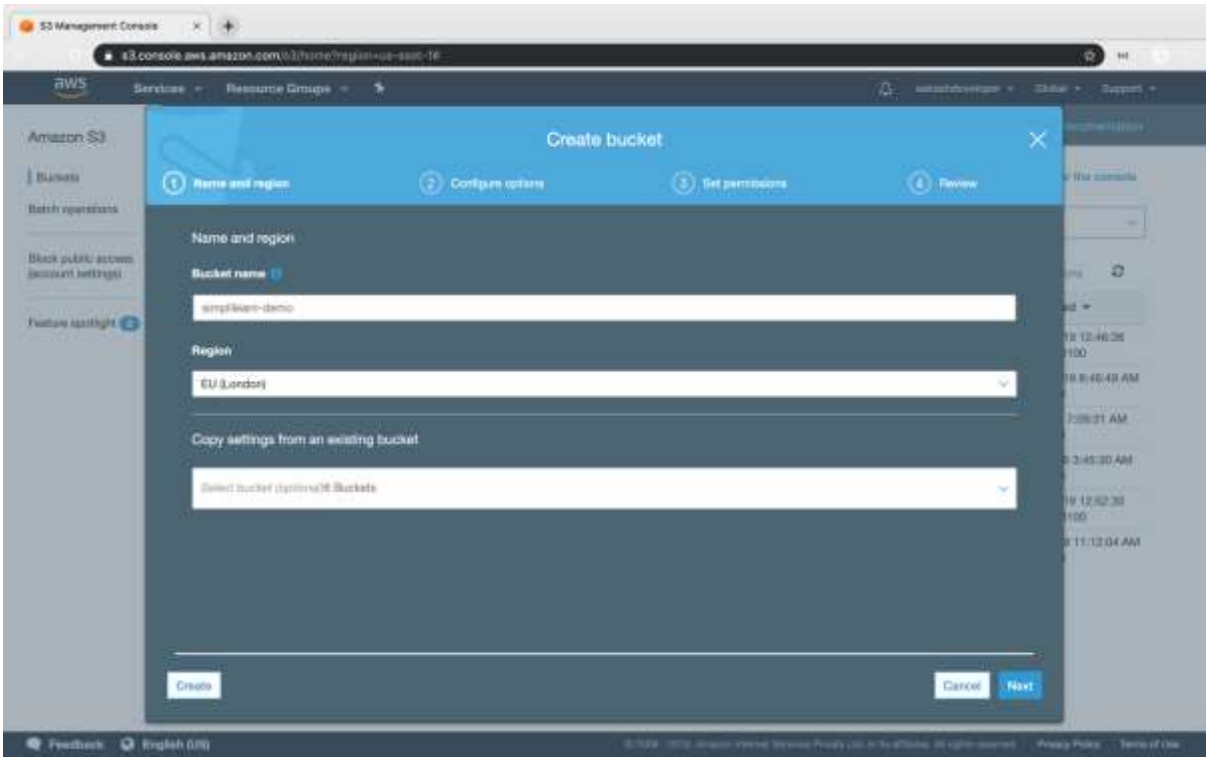


Step 3.11..2: Verifying the button to create a bucket

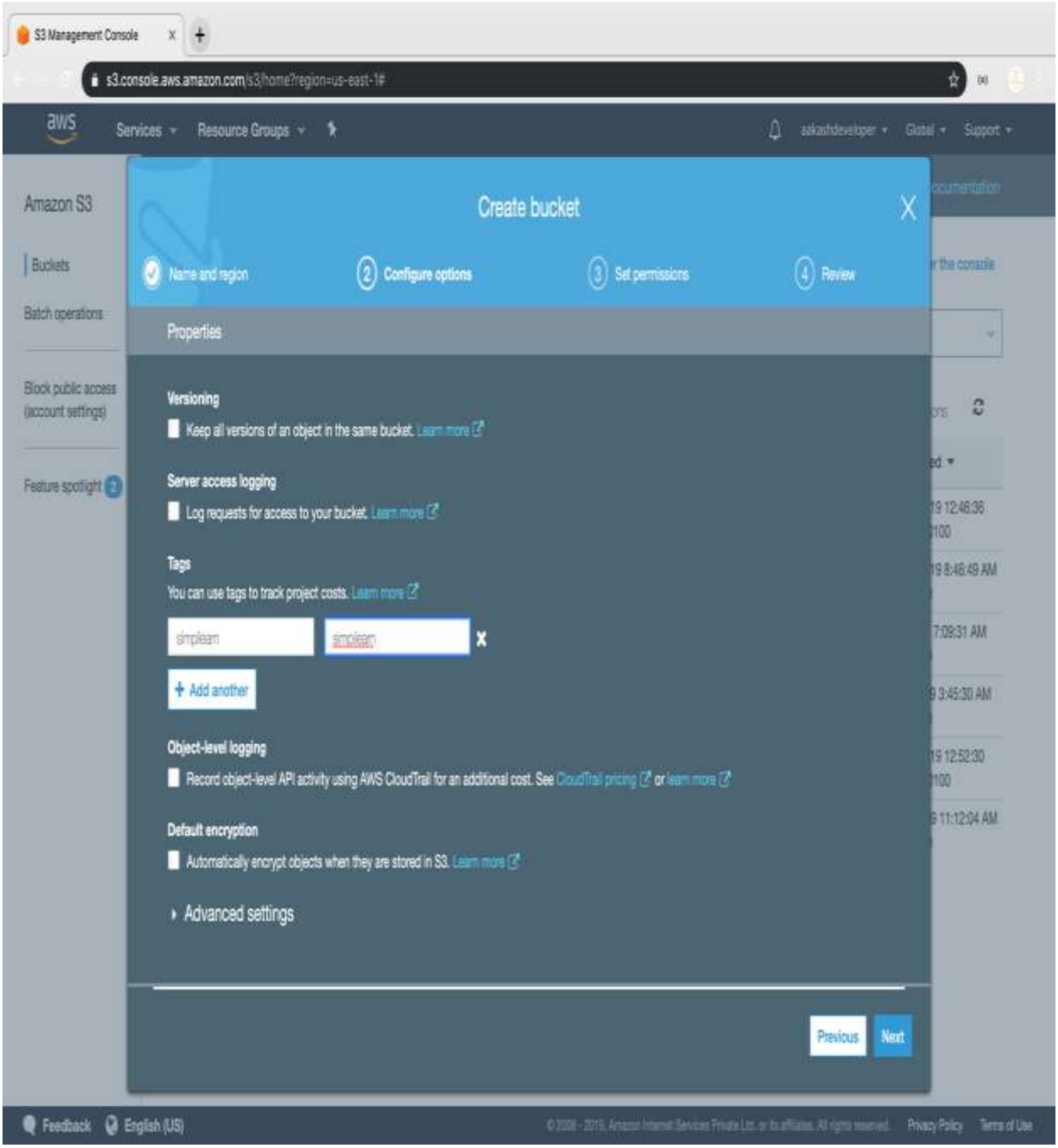


Step 3.11.3: Providing a bucket name and region

- Make sure that the bucket name is unique

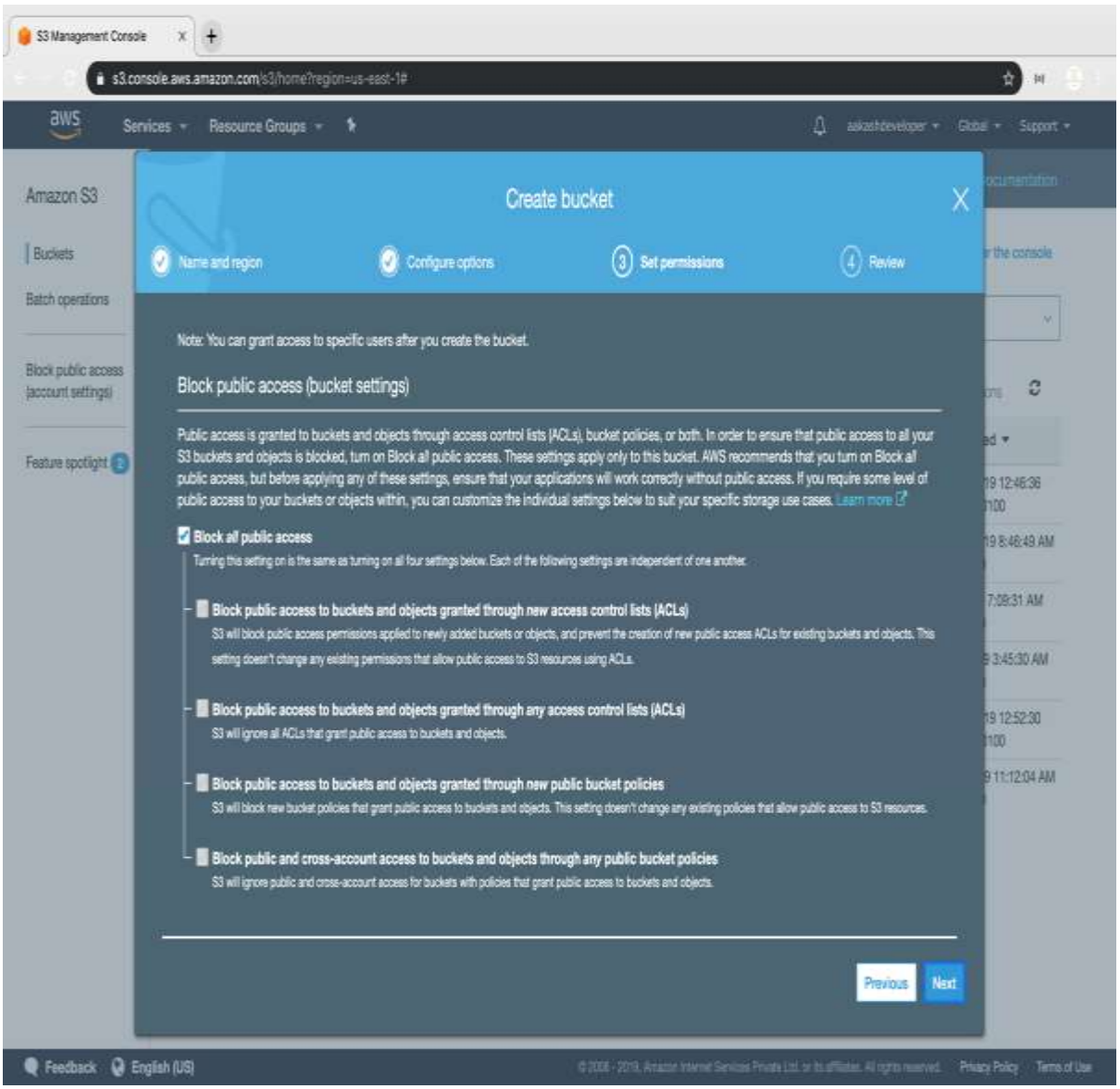


Step 4.11.4: Entering key and value for identification and tracking the bucket

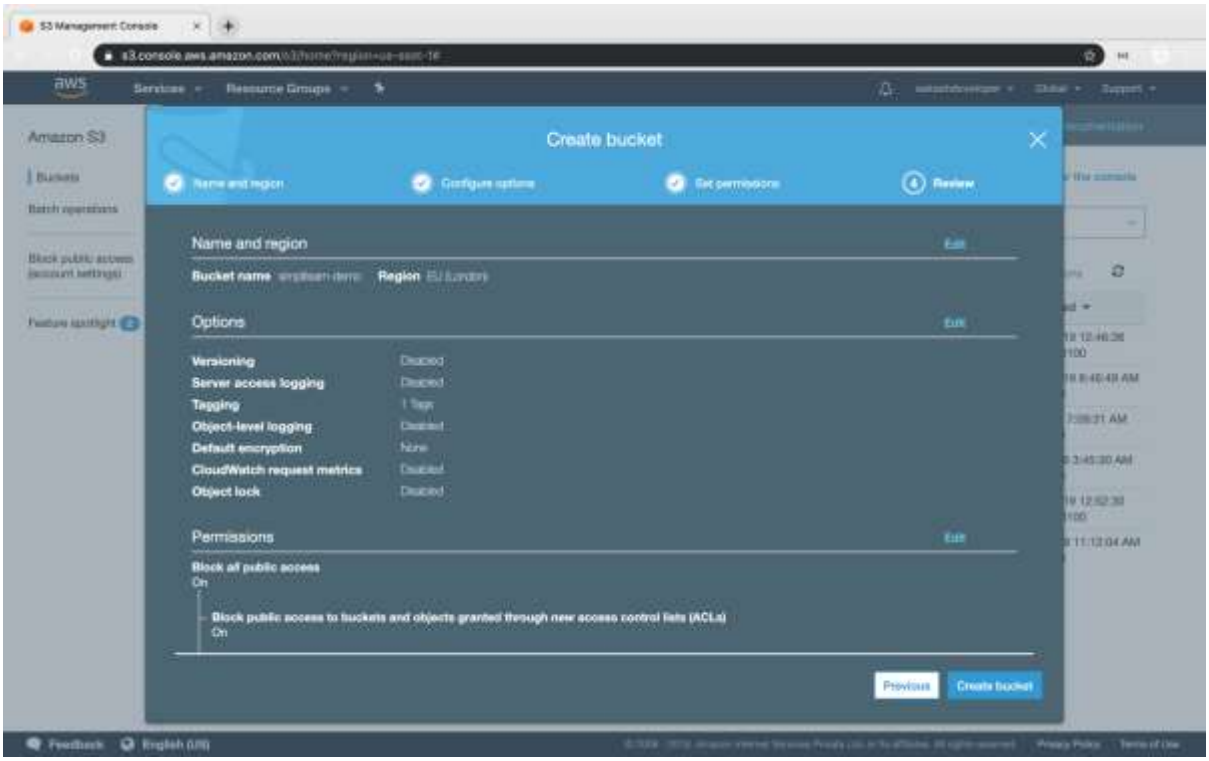


Step 3.11.5: Blocking all the public access

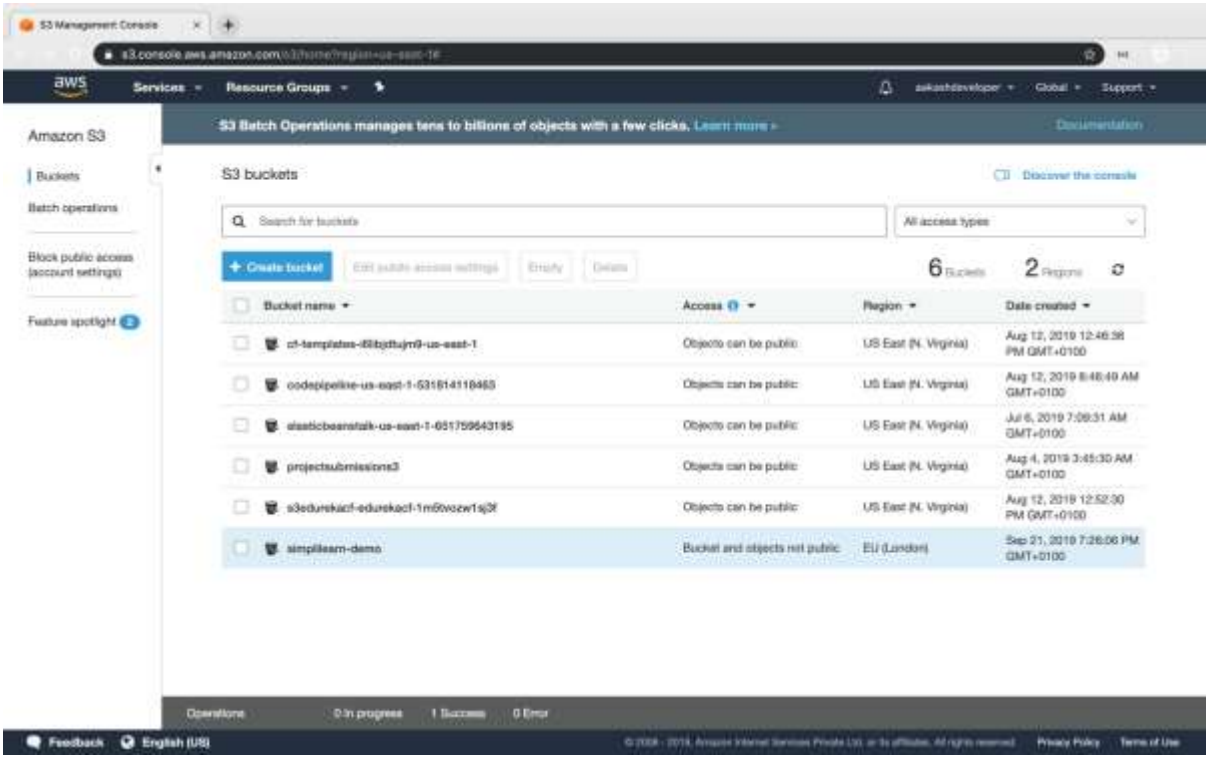
- If required, you can configure it later



Step 3.11.6: Reviewing all parameters and creating the bucket



Step 3.11.7: Verifying the newly created bucket on the panel of S3



Step 3.11.8: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.28 Delete an S3 Bucket

This section will guide you to:

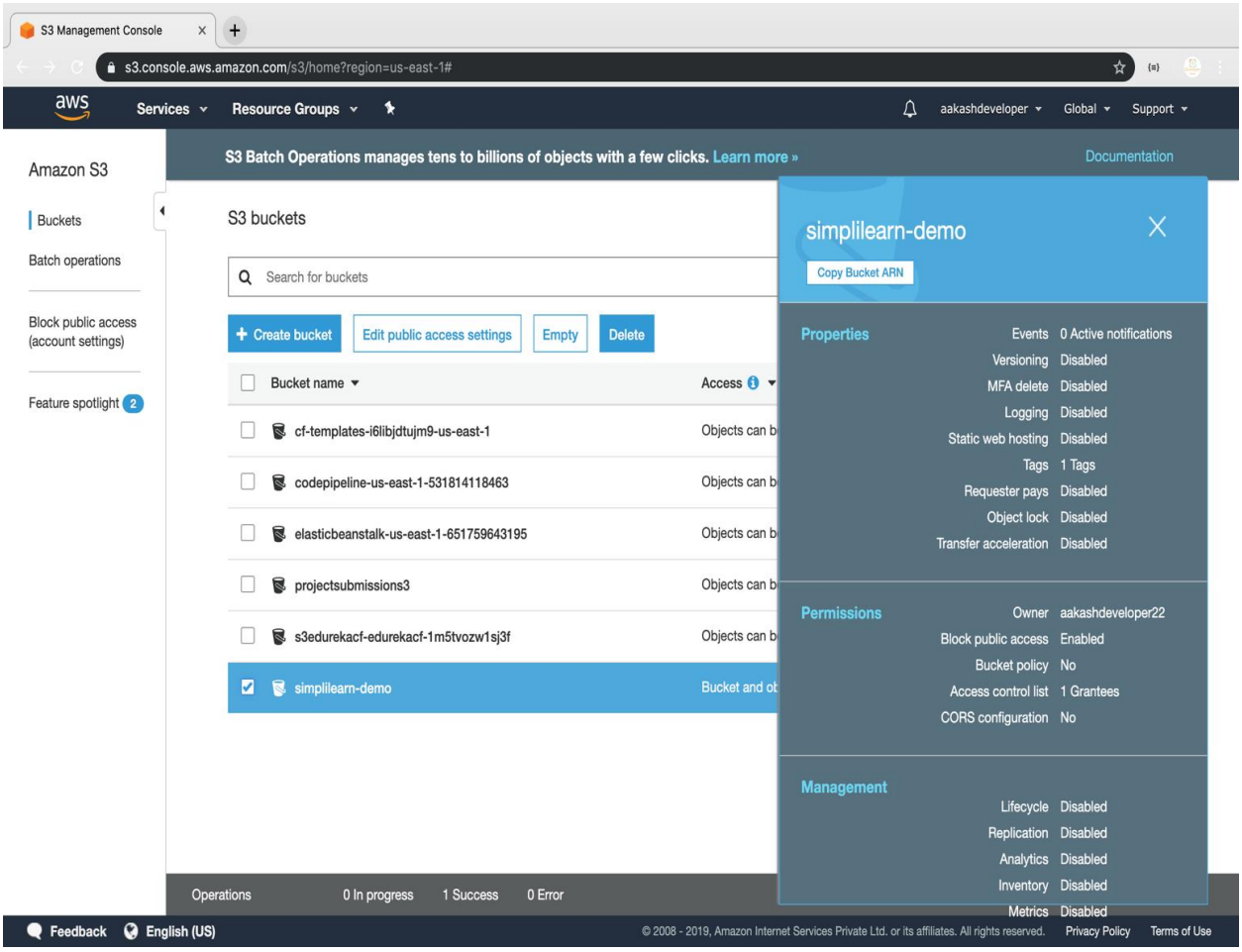
- Delete an S3 bucket

This lab has three subsections, namely:

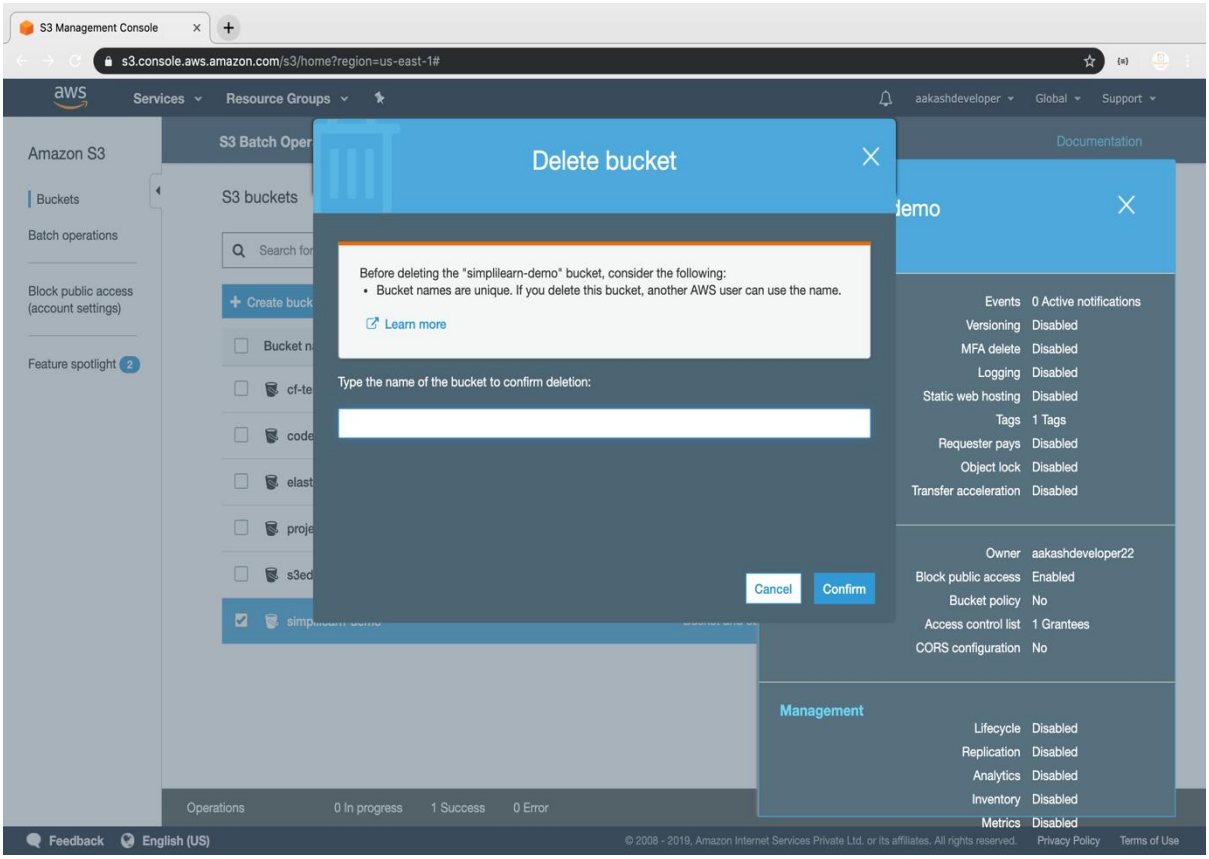
3.12.1 Selecting the S3 bucket you want to delete
3.12.2 Deleting the bucket
3.12.3 Pushing the files to GitHub repositories

Step 3.12.1: Selecting the S3 bucket you want to delete

- Choose the bucket



Step 3.12.2: Deleting the bucket



Step 3.12.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.28 Delete an S3 Bucket

This section will guide you to:

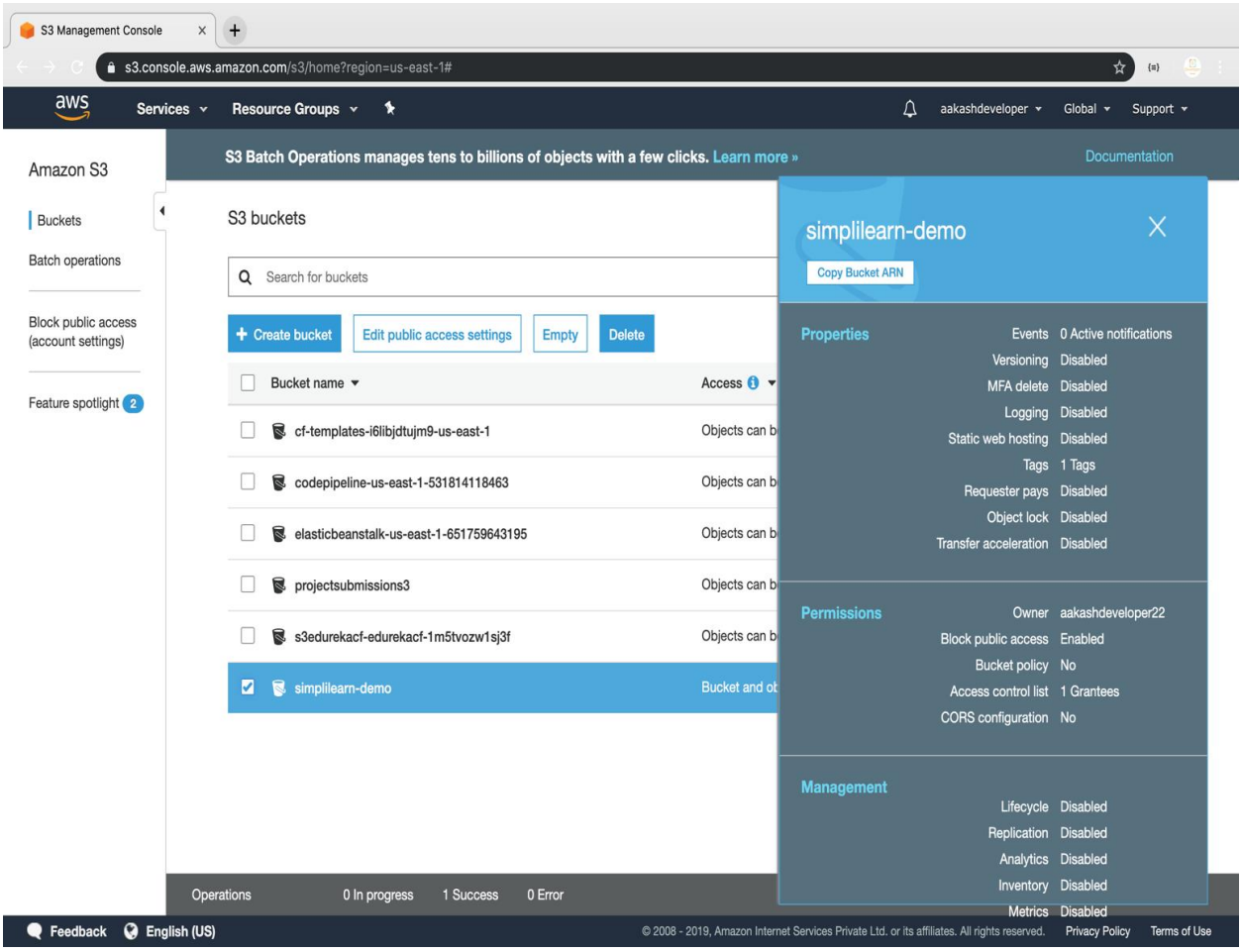
- Delete an S3 bucket

This lab has three subsections, namely:

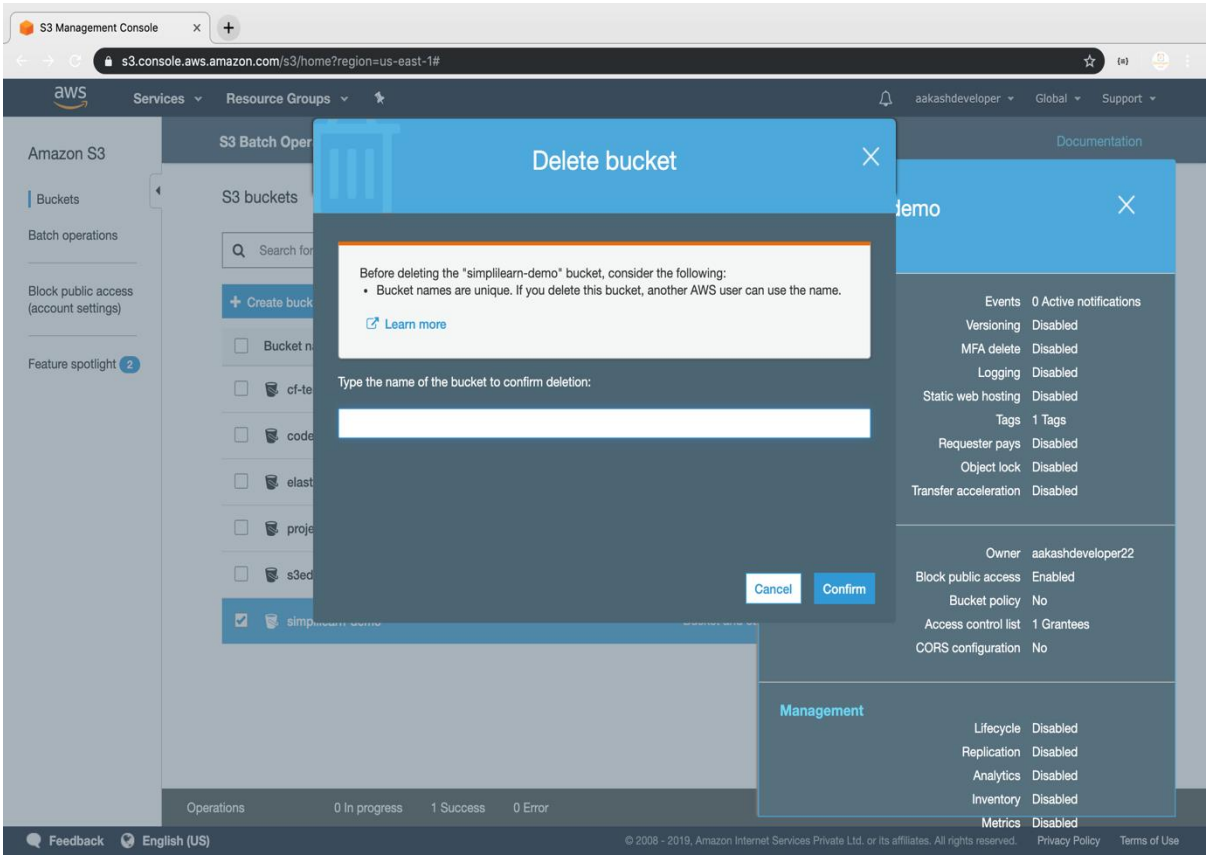
3.12.1 Selecting the S3 bucket you want to delete
3.12.2 Deleting the bucket
3.12.3 Pushing the files to GitHub repositories

Step 3.12.1: Selecting the S3 bucket you want to delete

- Choose the bucket



Step 3.12.2: Deleting the bucket



Step 3.12.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.32 Set the Storage Class of an Object

This section will guide you to:

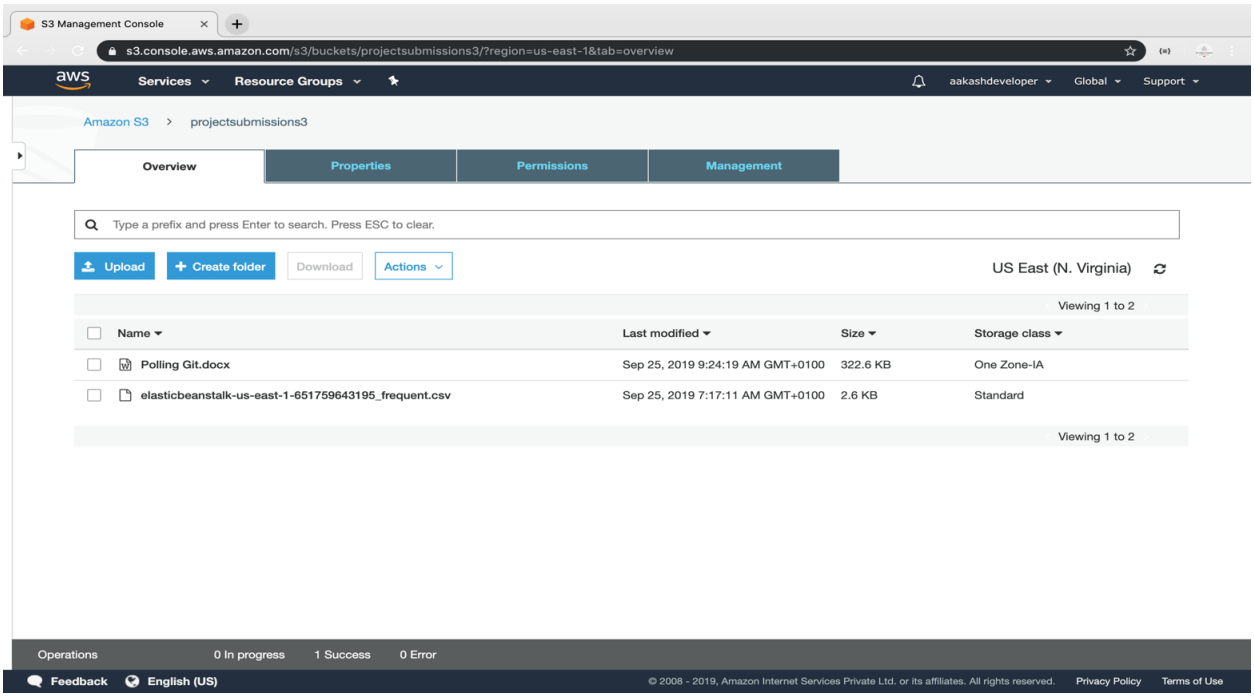
- Set the storage class of an object

This lab has three subsections, namely:

3.14.1 Selecting the bucket
3.14.2 Setting the storage class of an object
3.14.3 Pushing the files to GitHub repositories

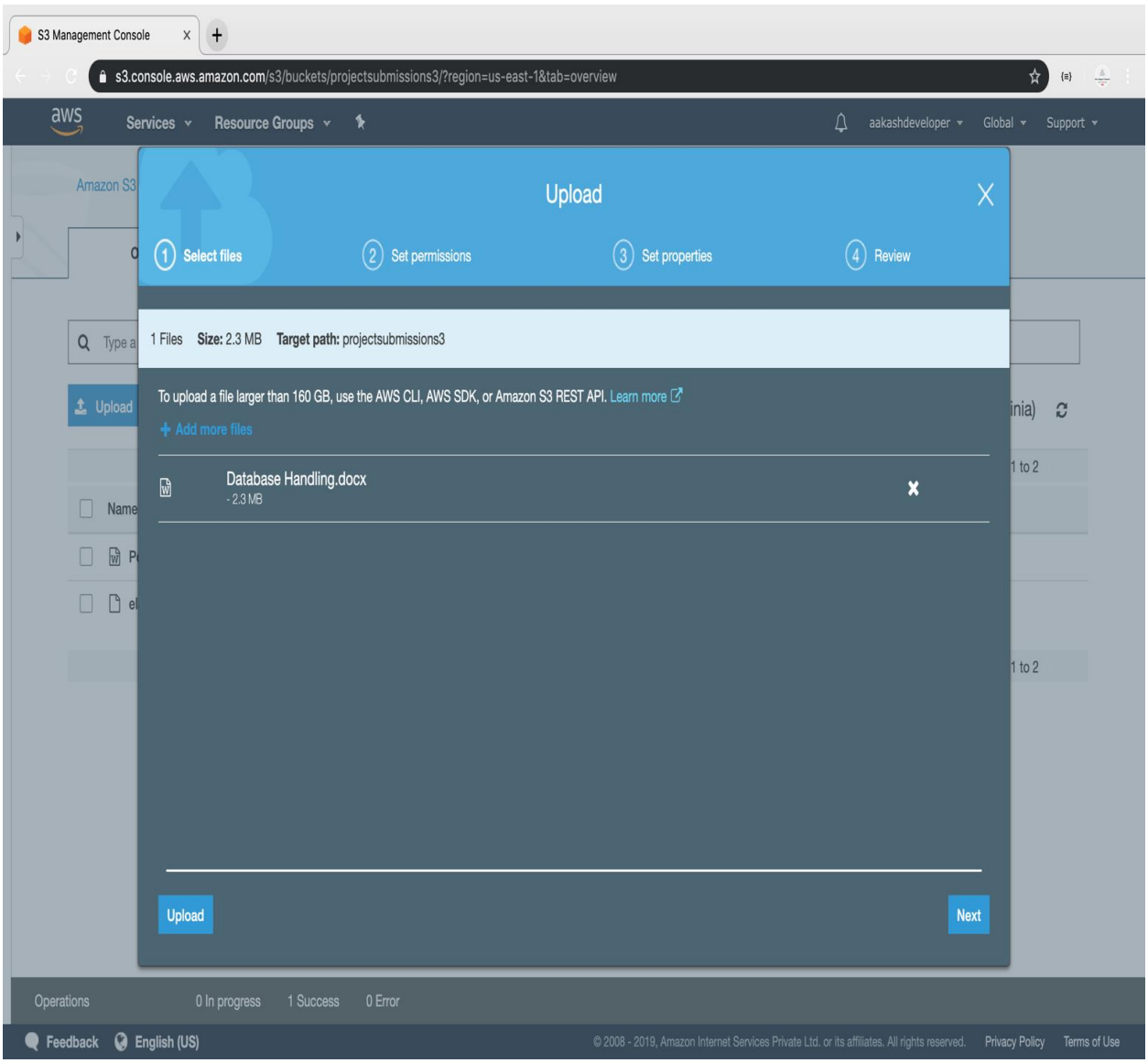
Step 3.14.1: Selecting the bucket

- Select the bucket and open the list of items

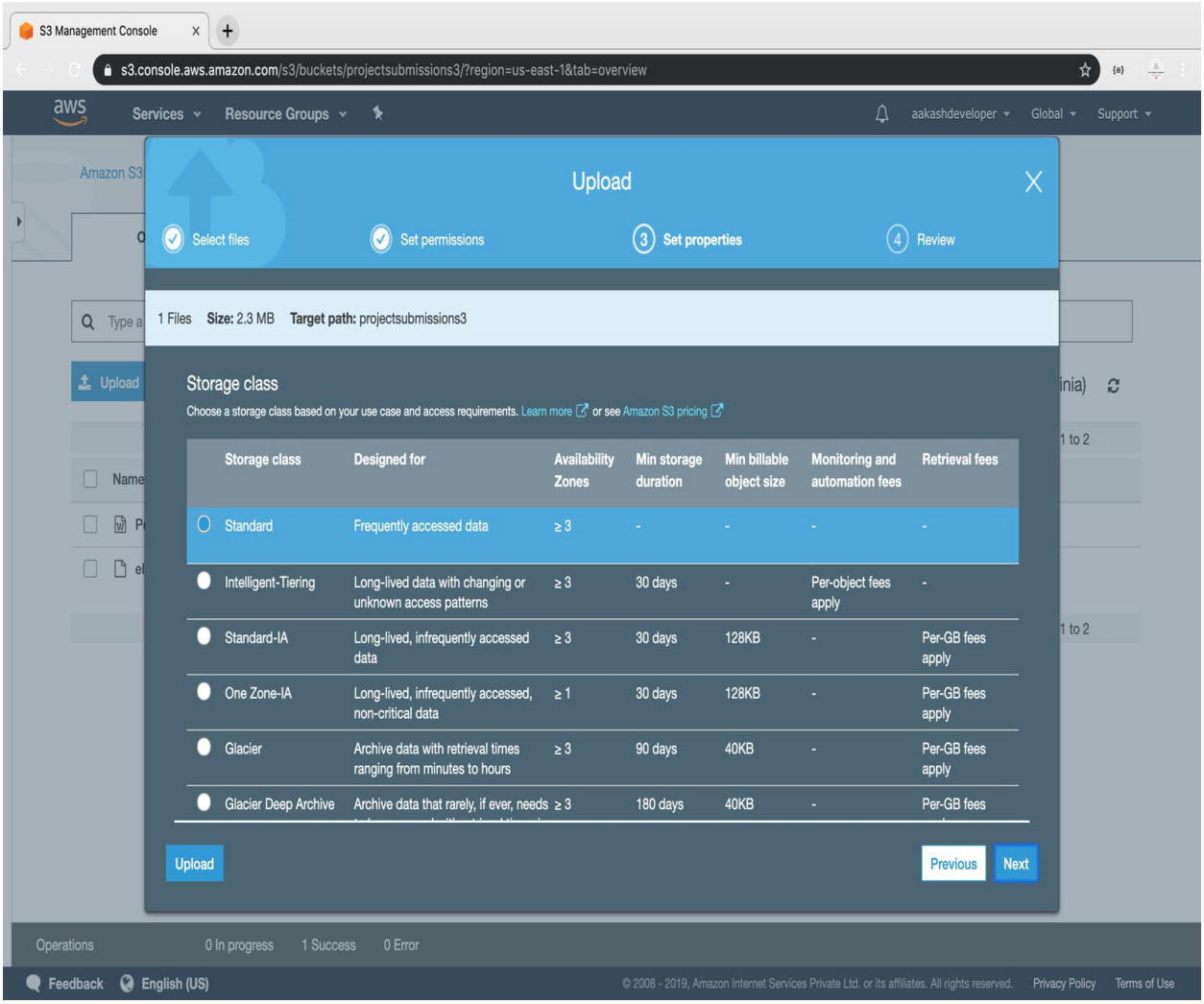


Step 3.14.2: Setting the storage class of an object

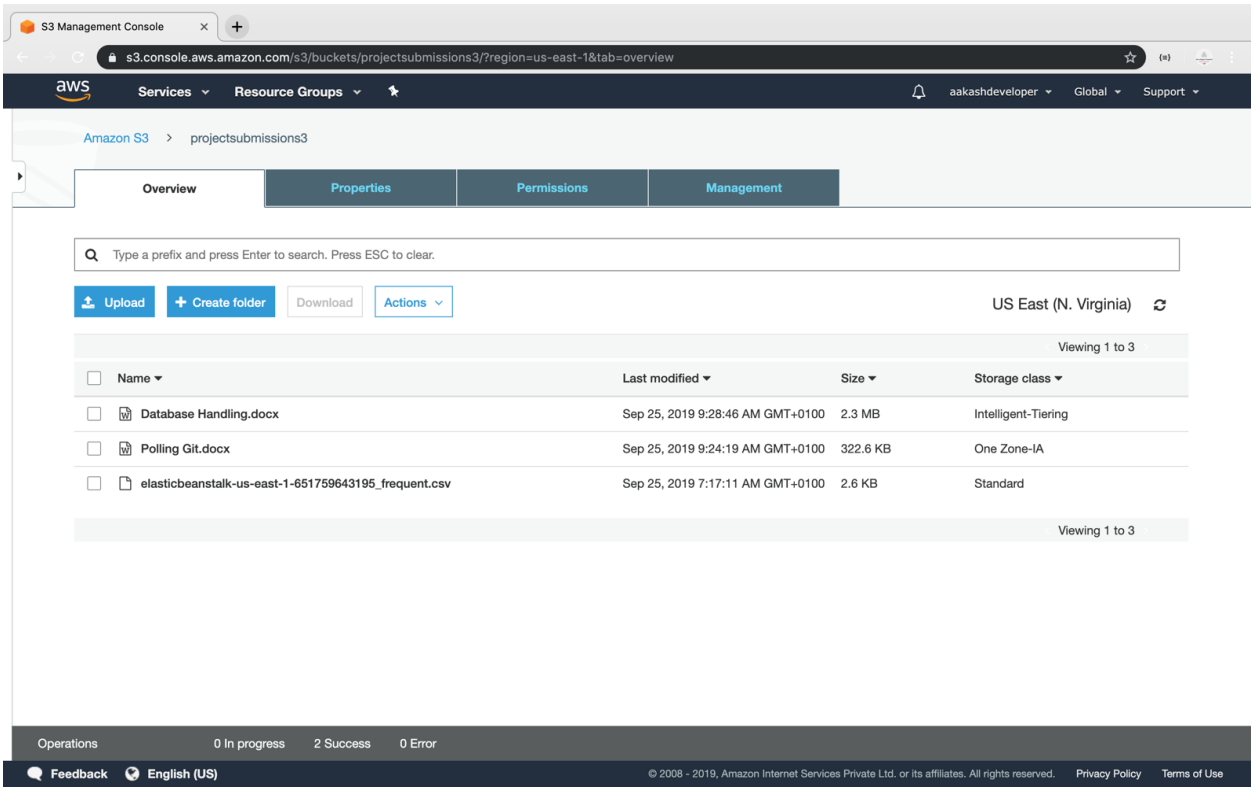
- Click on *Upload* and select the file and click *Next*



- Select the class and click *Next*



- The class will get assigned to the uploaded object



Step 3.14.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.36 Create an IAM User

This section will guide you to:

- Create an IAM user
- Validate the newly created IAM user

This lab has five subsections, namely:

3.15.1 Selecting the IAM from the AWS Console

3.15.2 Navigating to the user creation page

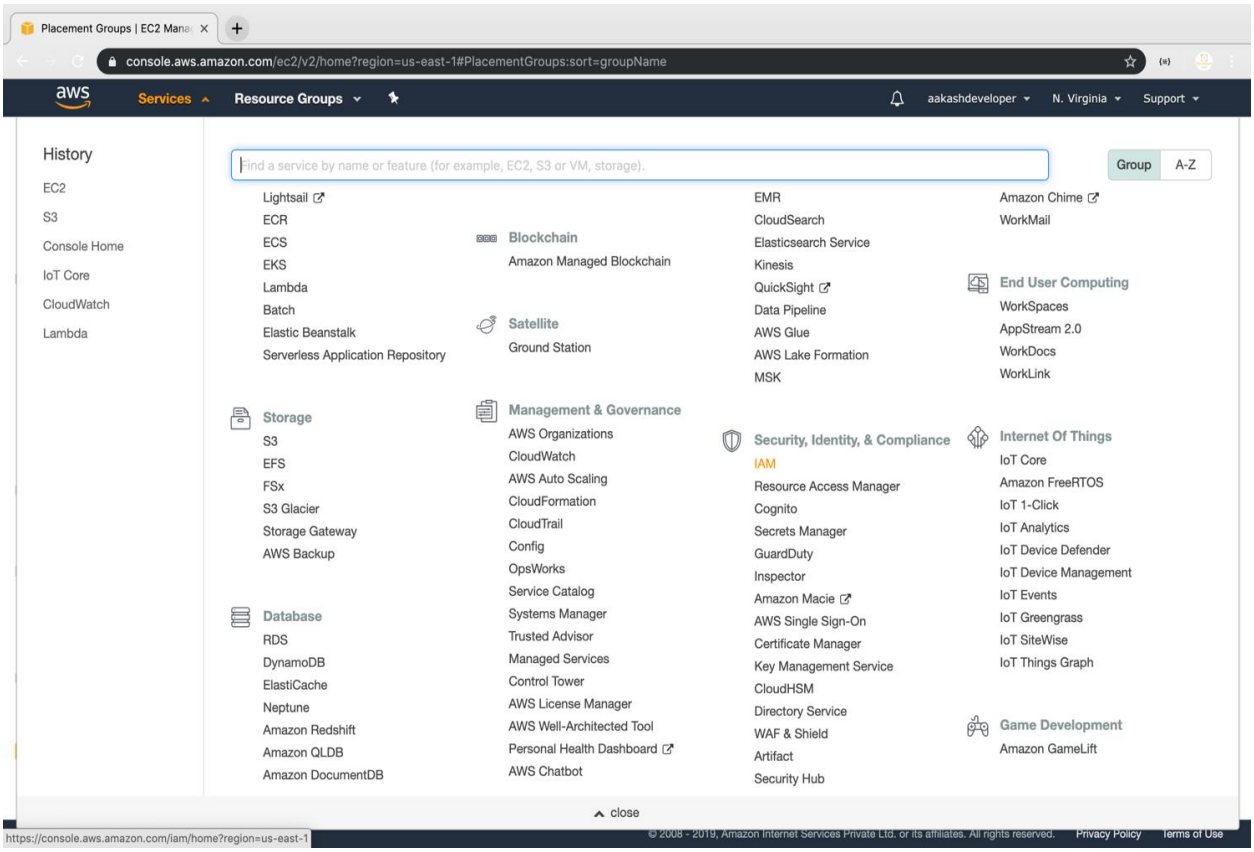
3.15.3 Changing access permissions

3.15.4 Adding an identifier

3.15.5 Pushing the files to GitHub repositories

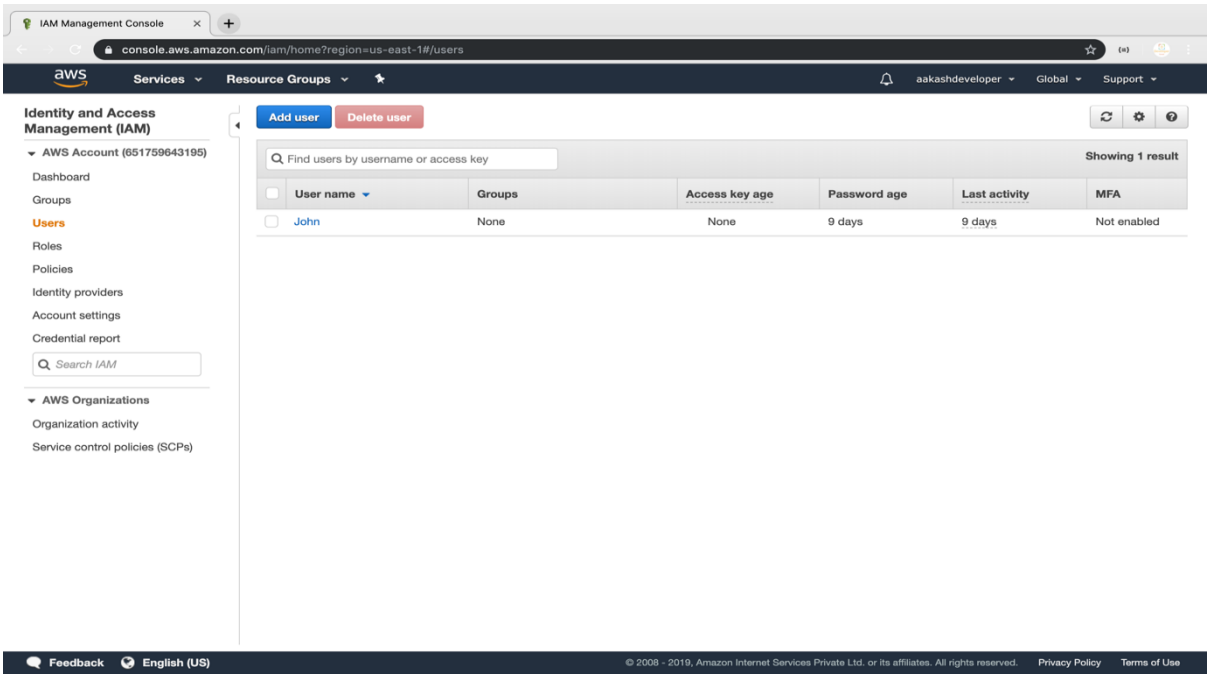
Step 3.15.1: Selecting the IAM from the AWS console

- Login to your AWS console and search for IAM



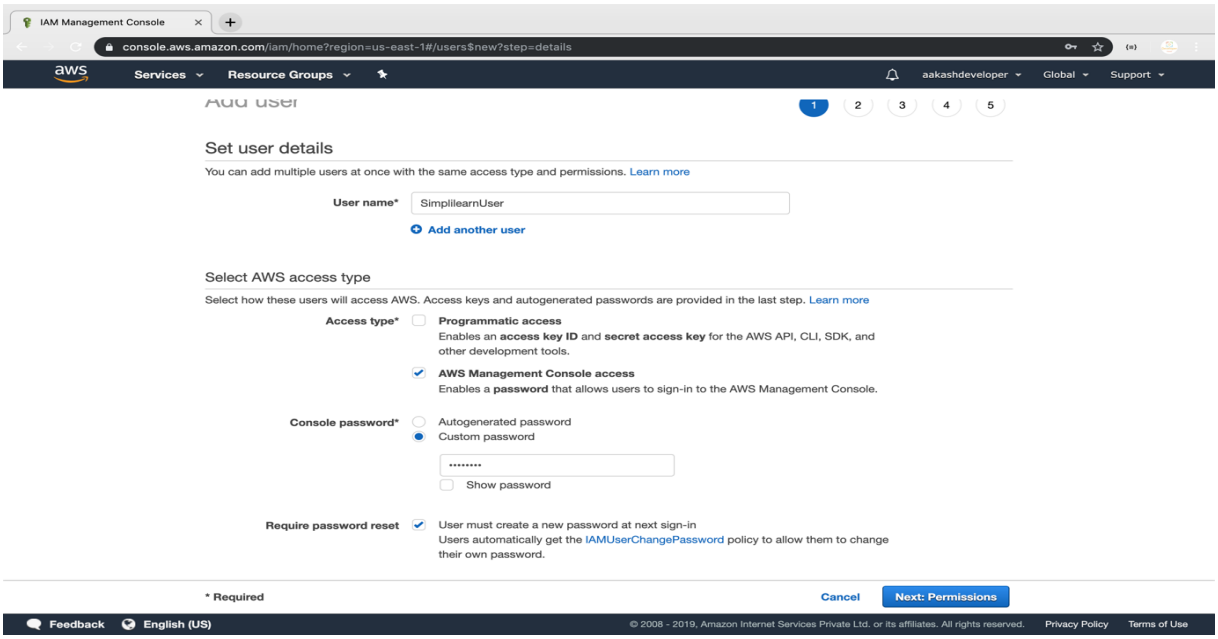
Step 3.15.2: Navigate to the user creation page

- Click on **Users** to redirect to the user creation page to create IAM users

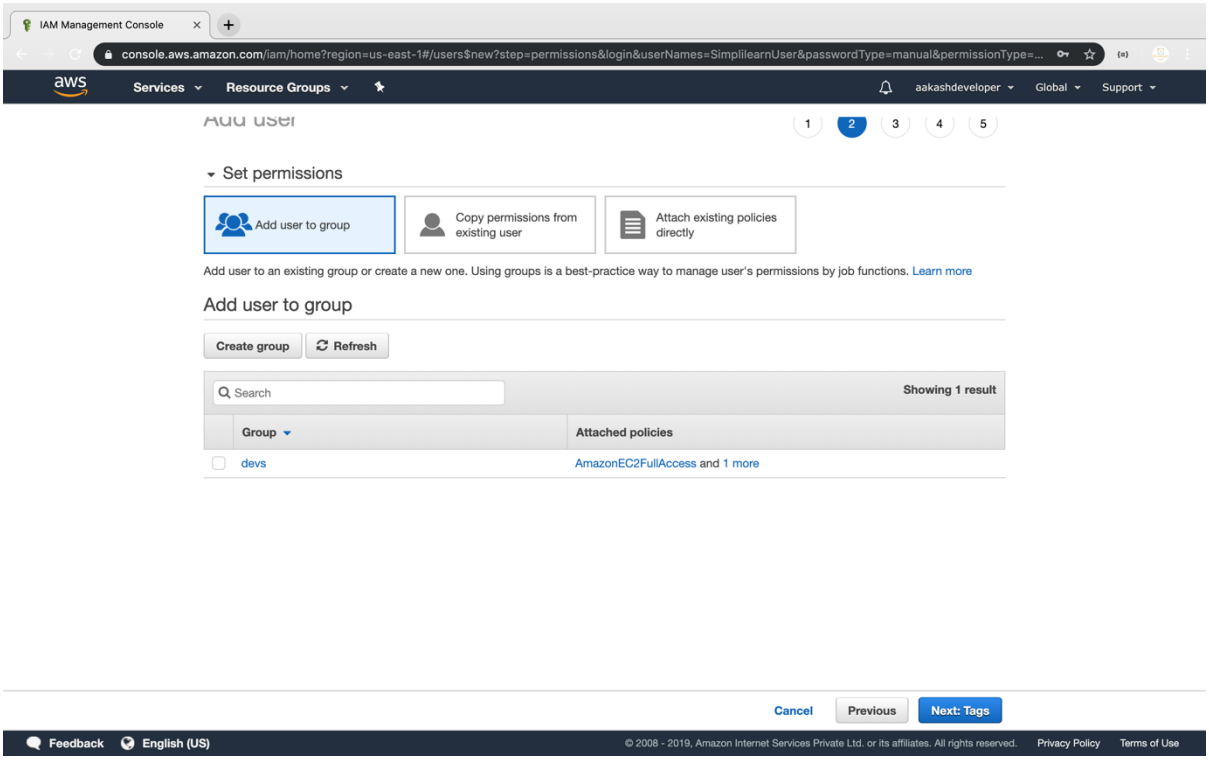


Step 3.15.3: Changing access permissions

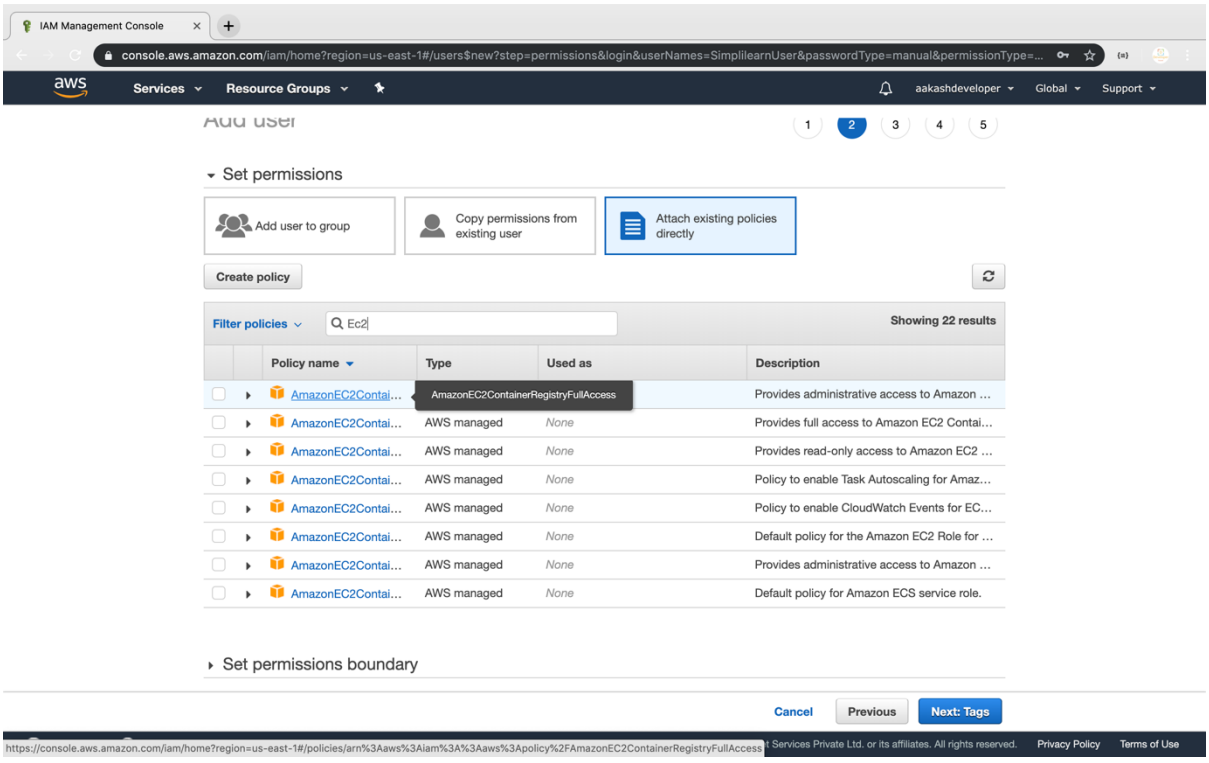
- Add username in the **User name*** section. Select **AWS Management Console access** and select one of the ways to configure **console password**



- Either we can add user to the created group or we can add policy as shown below:

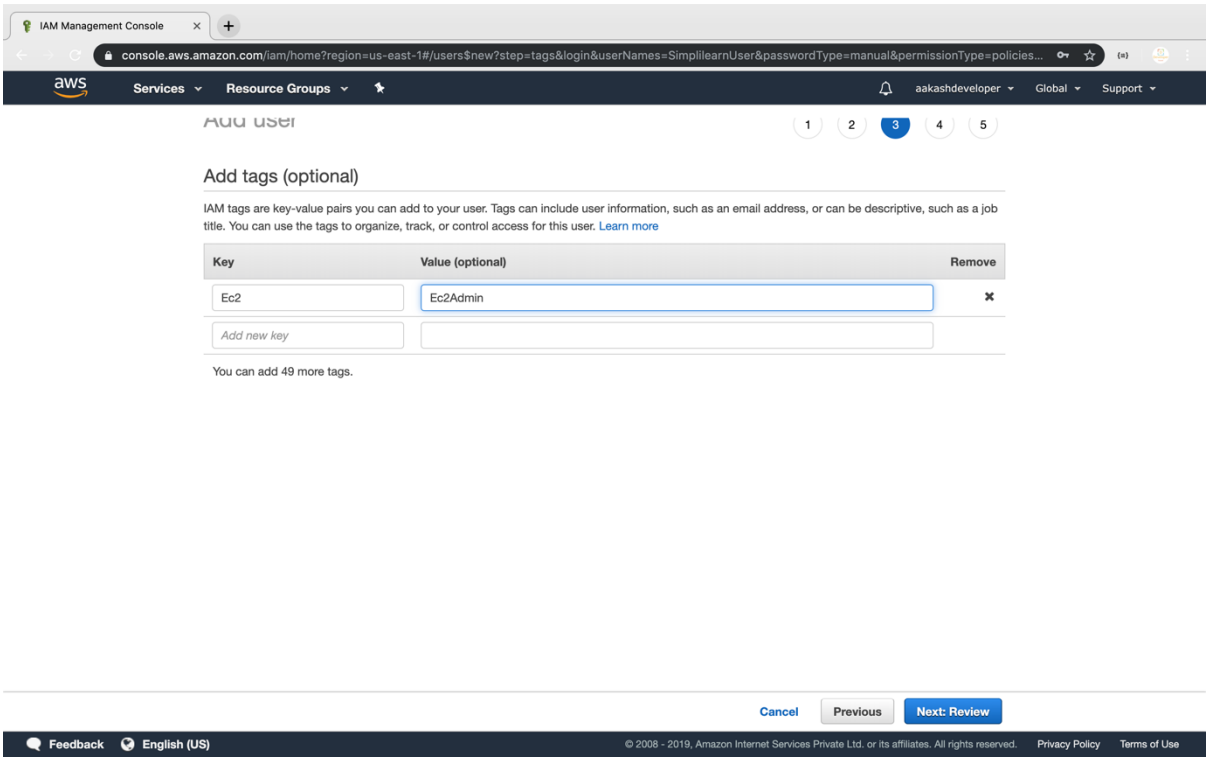


- In case a user needs to add a policy first, then he or she needs to search and then add policy to the user

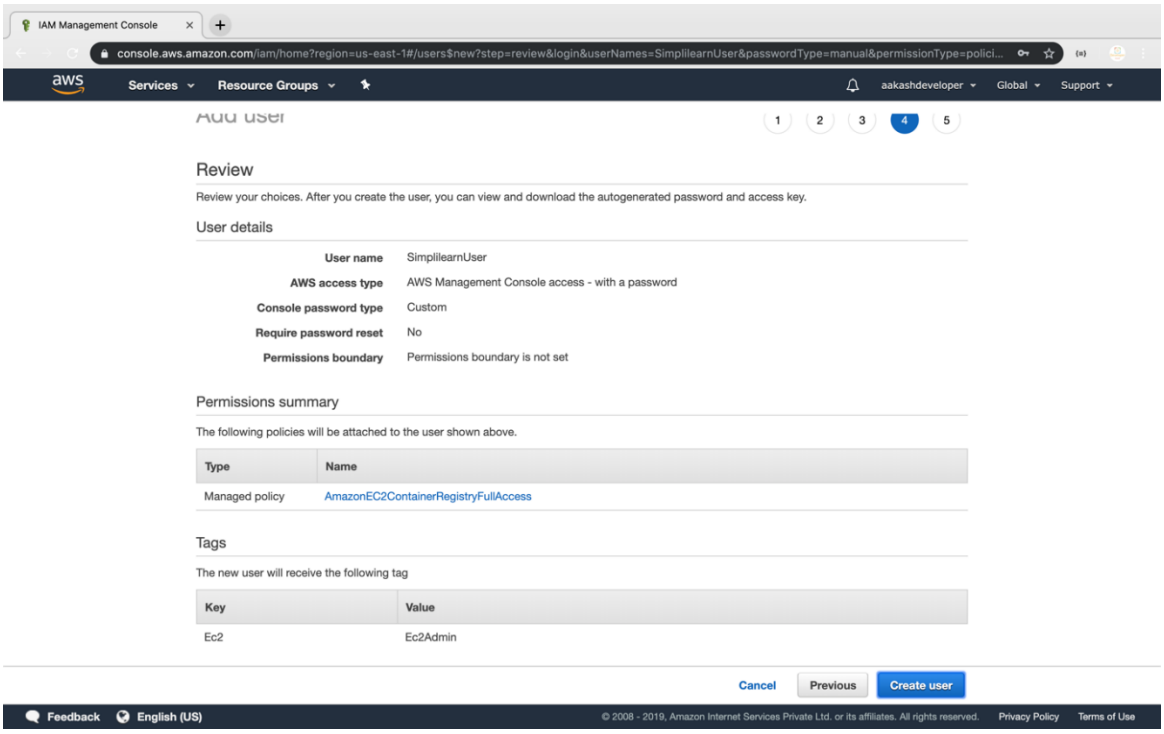


Step 3.15.4: Adding an identifier

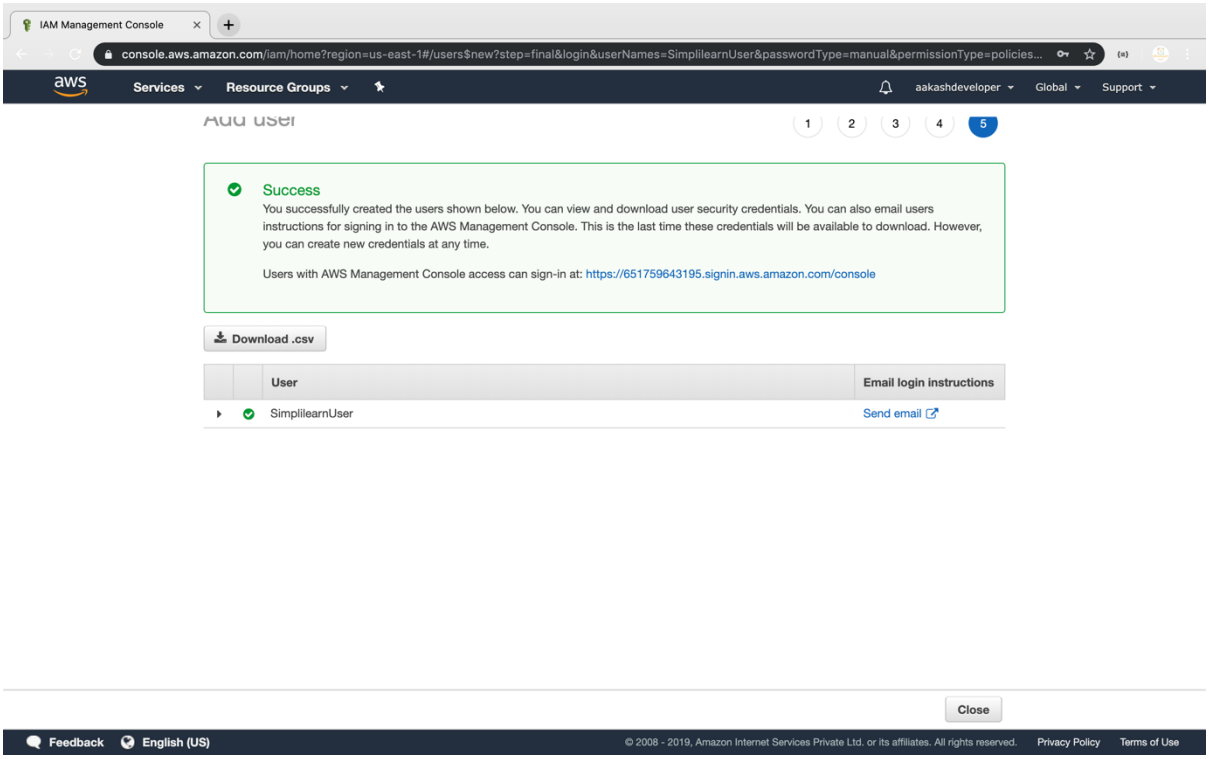
- Add an identifier by adding key and value to tag each user as shown below:



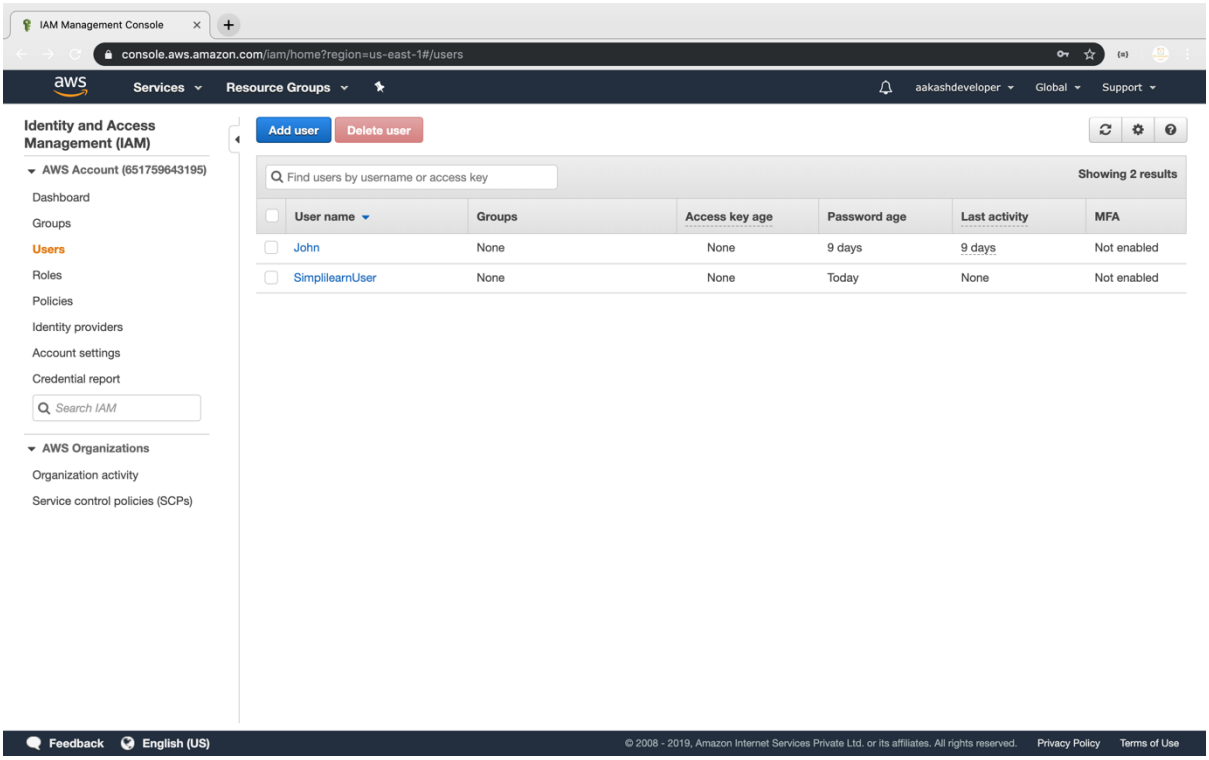
- Finally, it will create a user with a group or an added policy



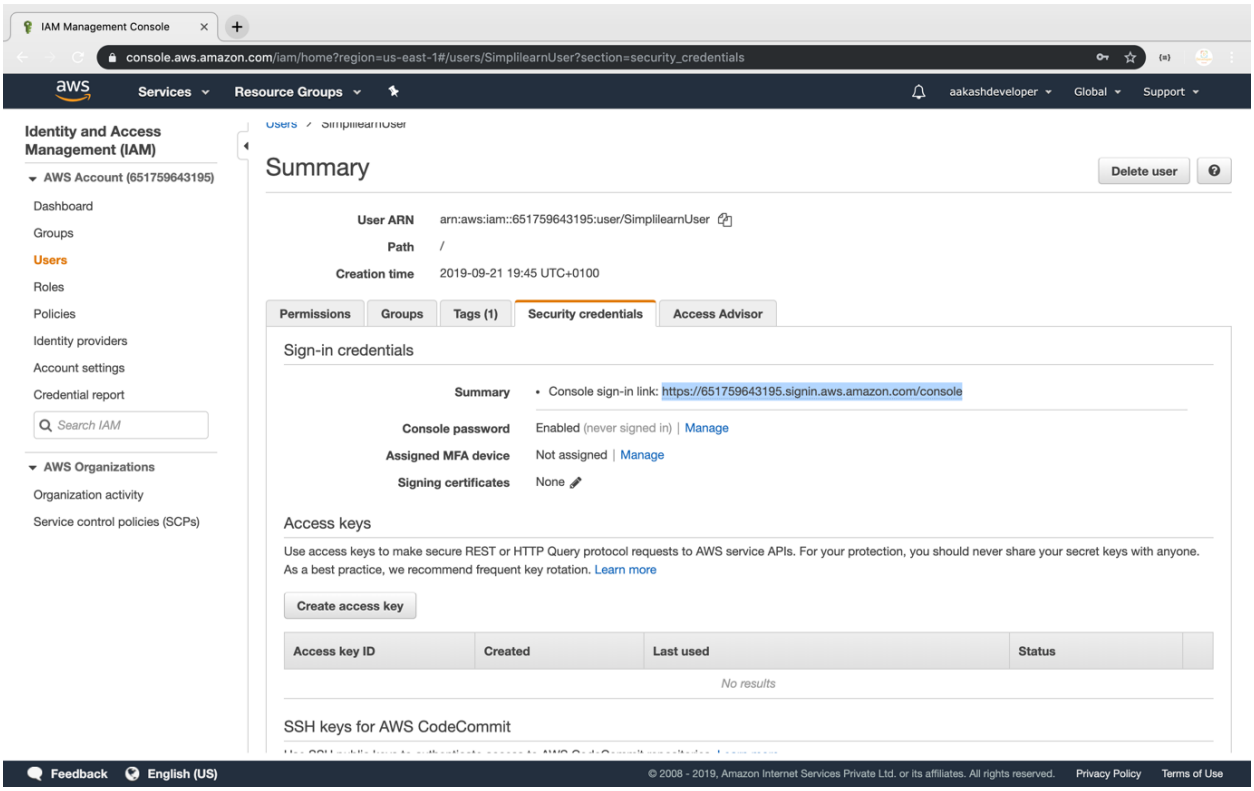
- On successful user creation, a URL will be created with which a user can login as an IAM user



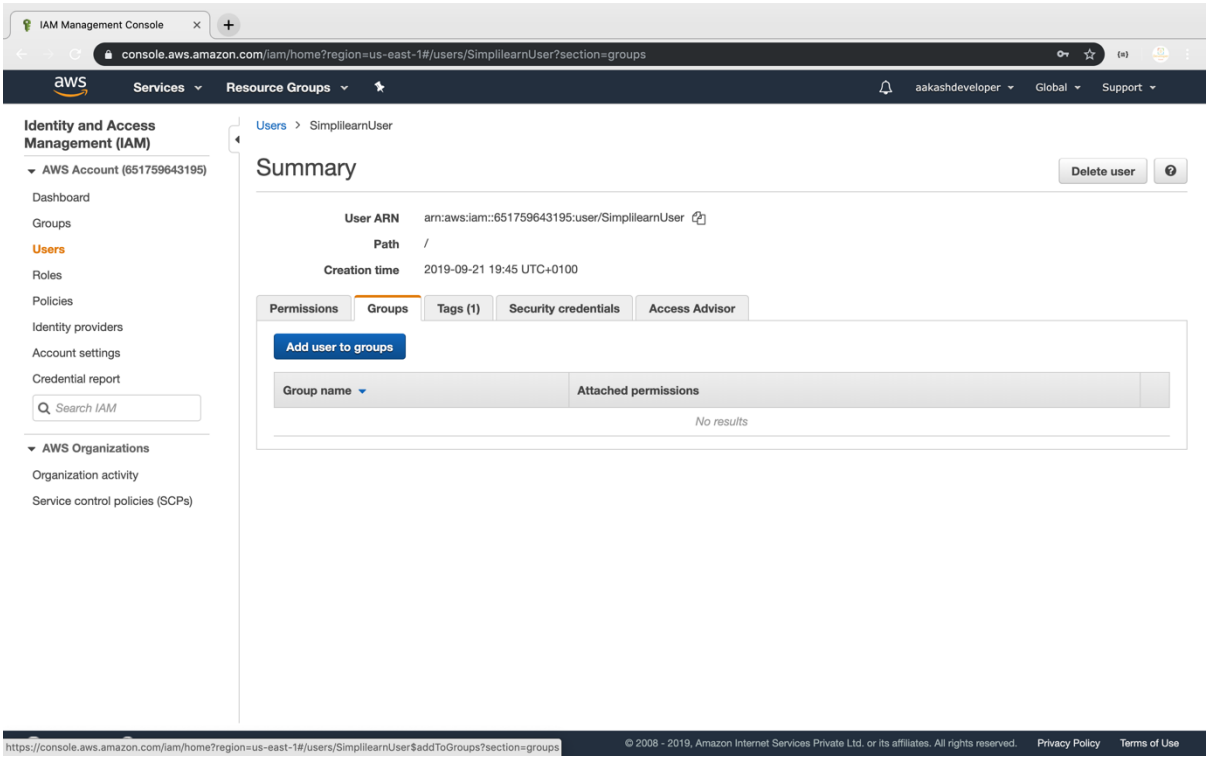
- The newly created user will be available on the dashboard as shown below:



- By clicking on each user, you can see the key sign in the URL



- By selecting a user, you can add that user to a particular group



Step 3.15.5: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.37 Create an IAM Role

This section will guide you to:

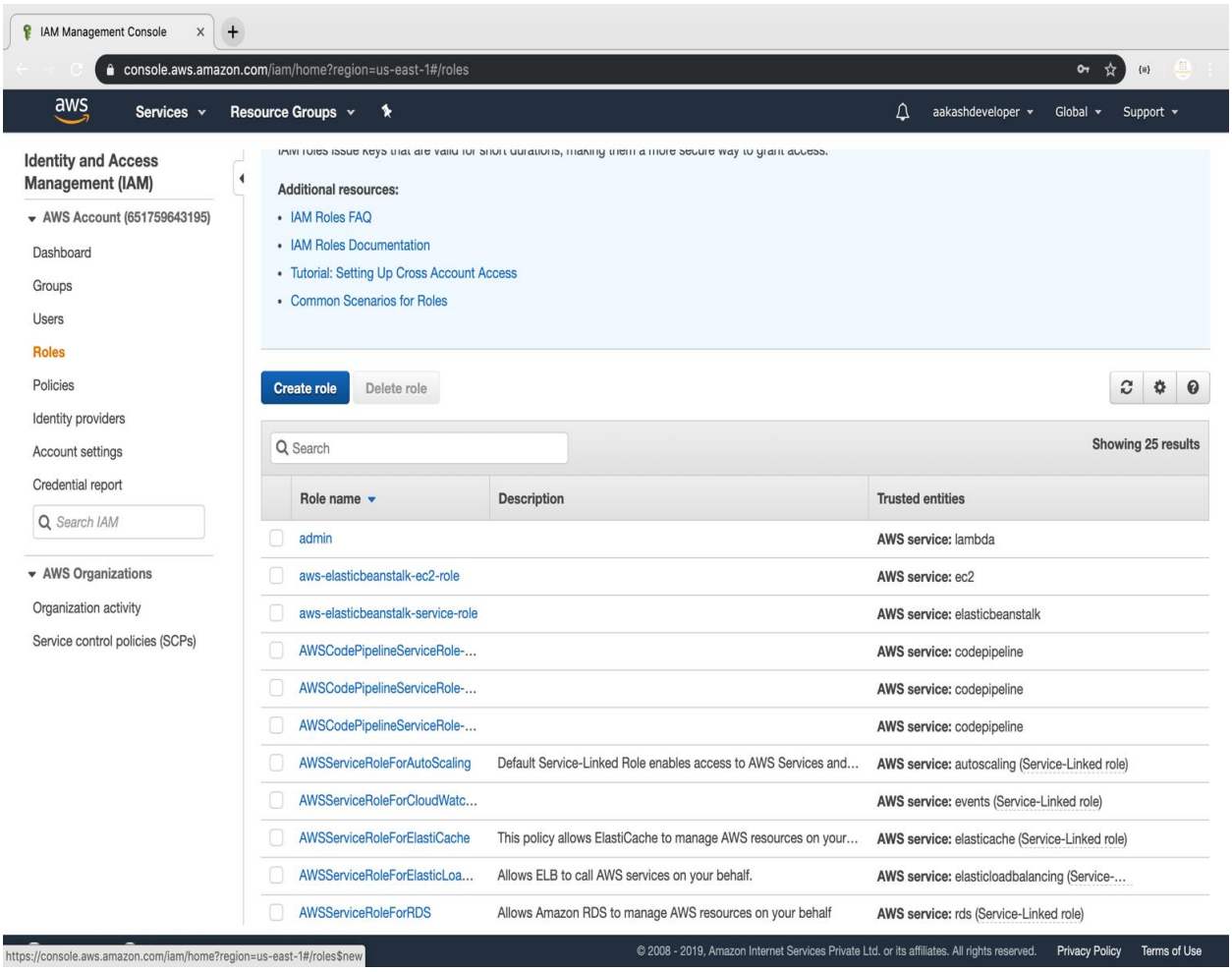
- Create an IAM role

This lab has four subsections, namely:

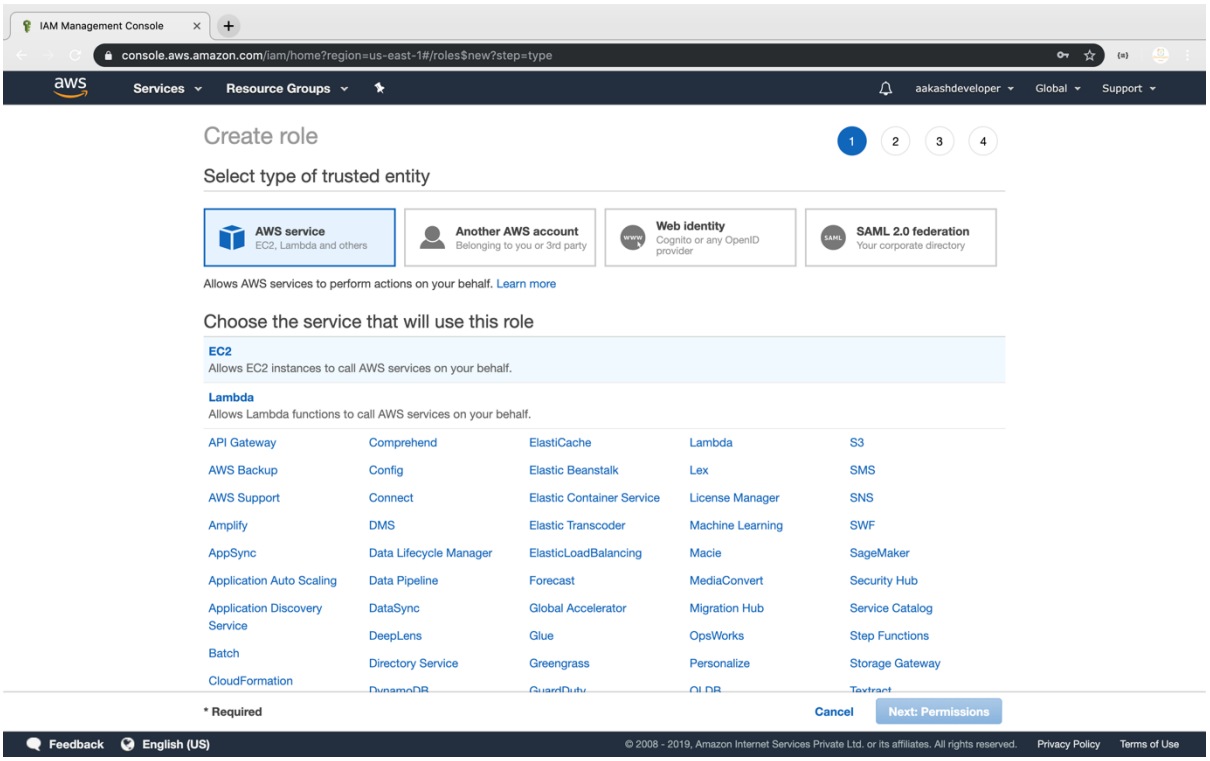
3.16.1 Creating an IAM role
3.16.2 Adding AWS policy to the role
3.16.3 Providing a key and a value for the role
3.16.4 Pushing the files to GitHub repositories

Step 3.16.1: Creating an IAM role

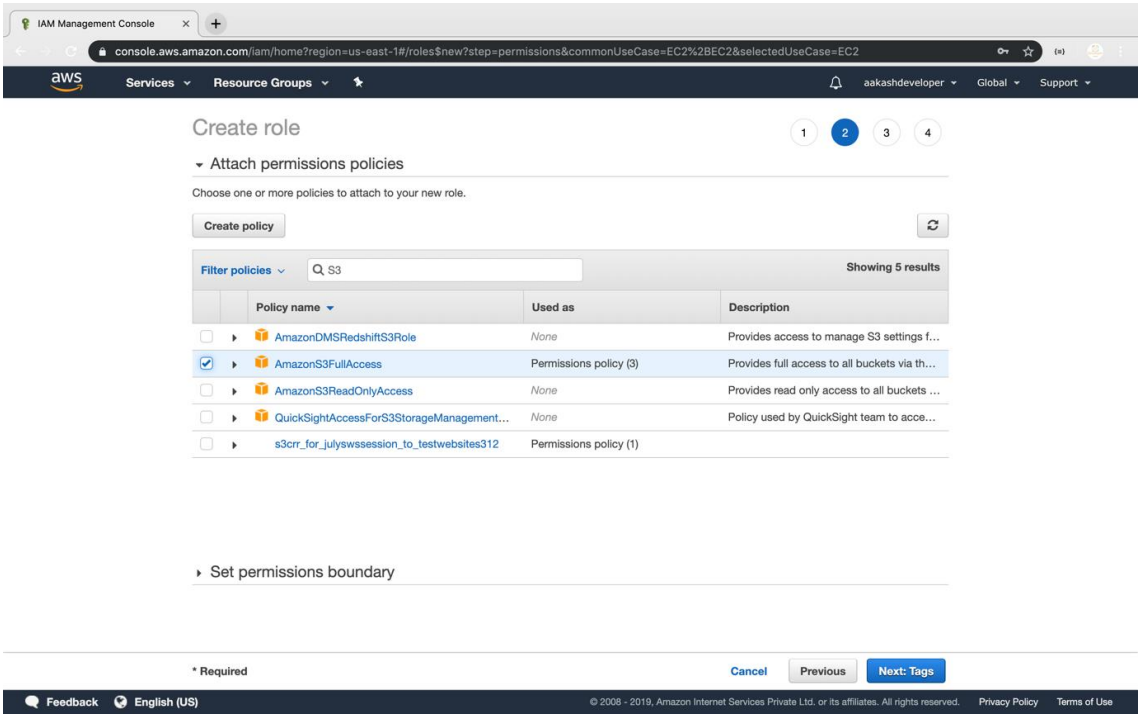
- In the IAM panel, click on *Roles* -> *Create role*



- Select the service

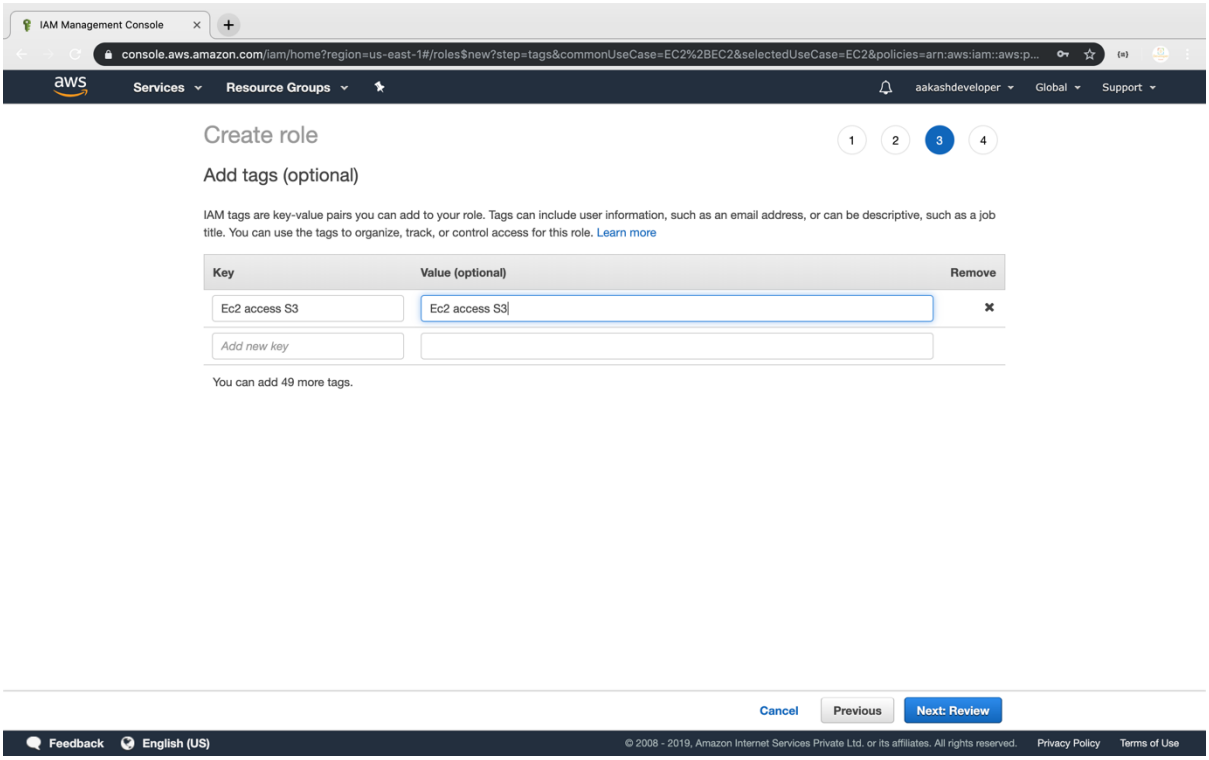


Step 3.16.2: Adding AWS policy to the role

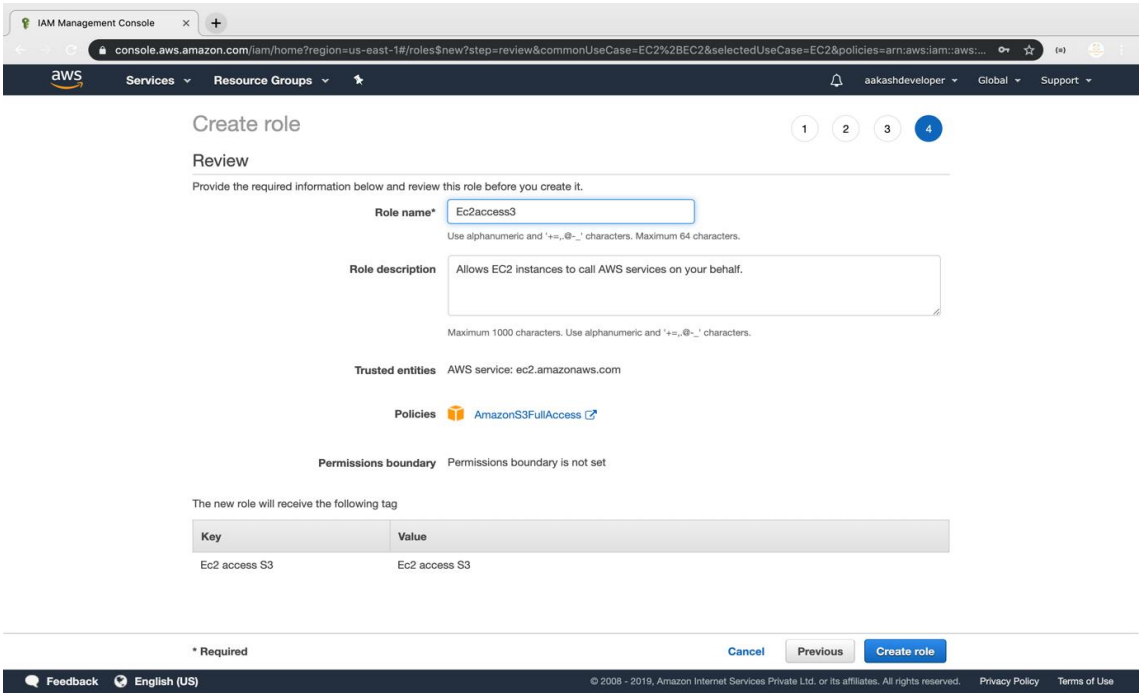


Step 3.16.3: Providing a key and a value for the role

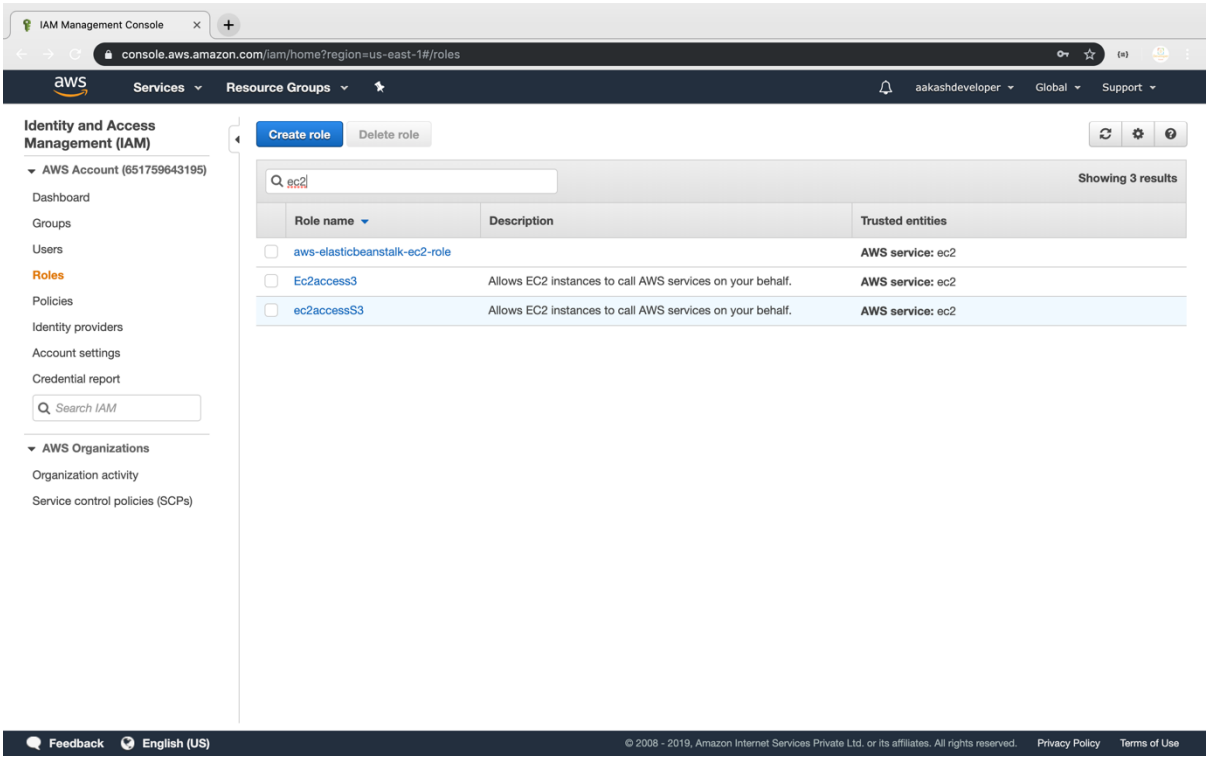
- Add a key and a value to the role



- Add a role:



- Once the role is created, you can find it in the IAM panel



Step 3.16.4: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.38 Create an IAM Group

This section will guide you to:

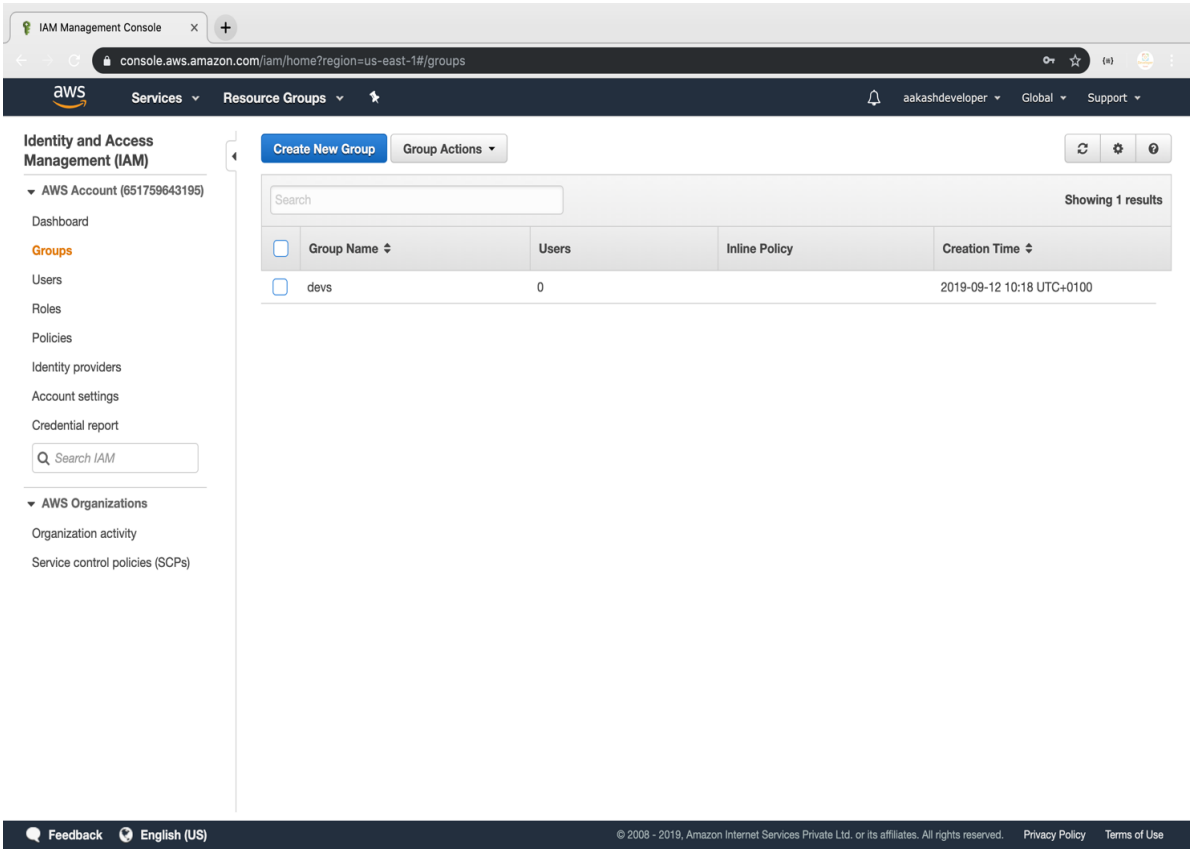
- Create an IAM group

This lab has three subsections, namely:

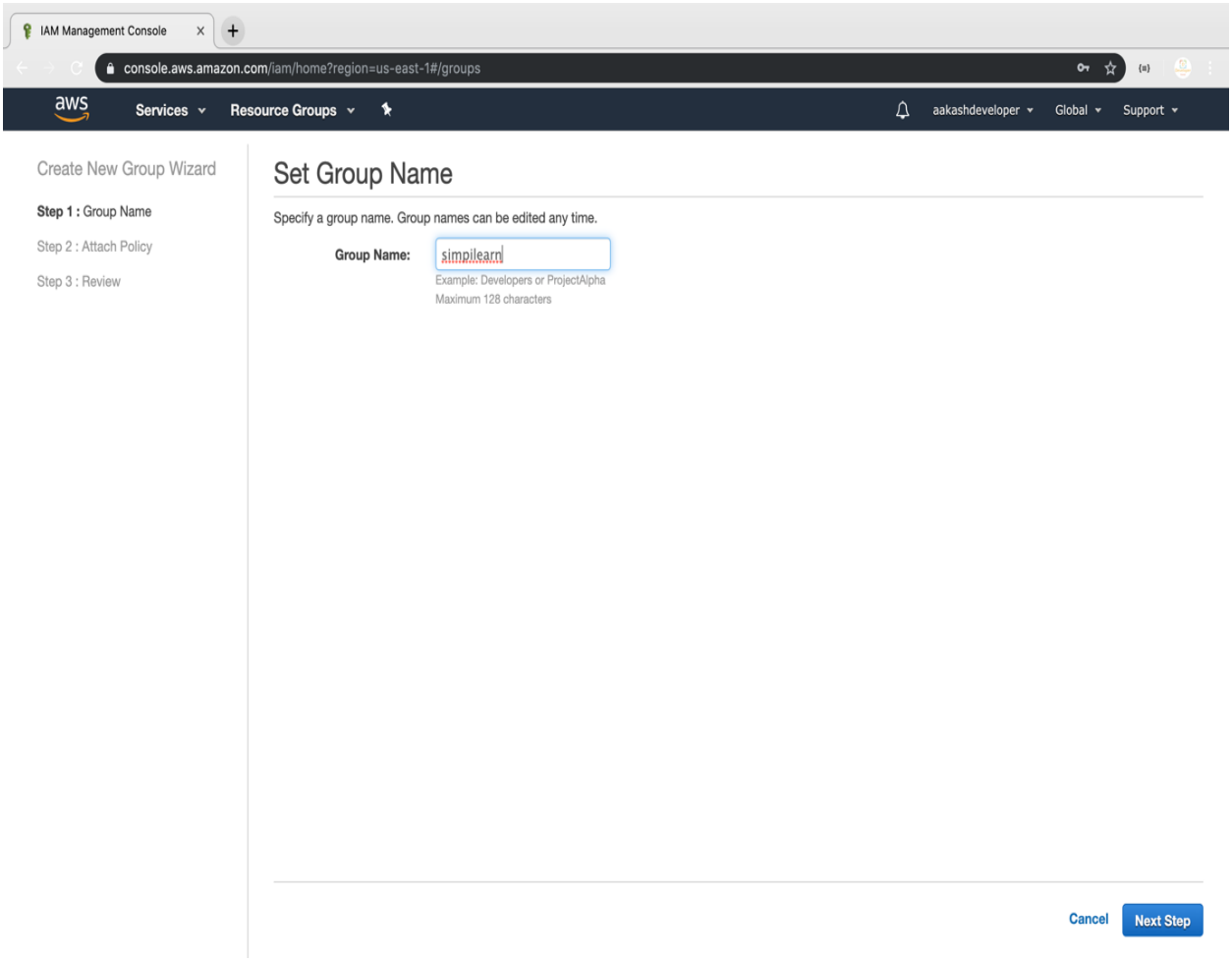
3.17.1 Creating an IAM group
3.17.2 Assigning the required policy to the group
3.17.3 Pushing the files to GitHub repositories

Step 3.17.1: Creating an IAM group

- In the IAM panel, click on *Groups*, and you will land on the *Create New Group* page

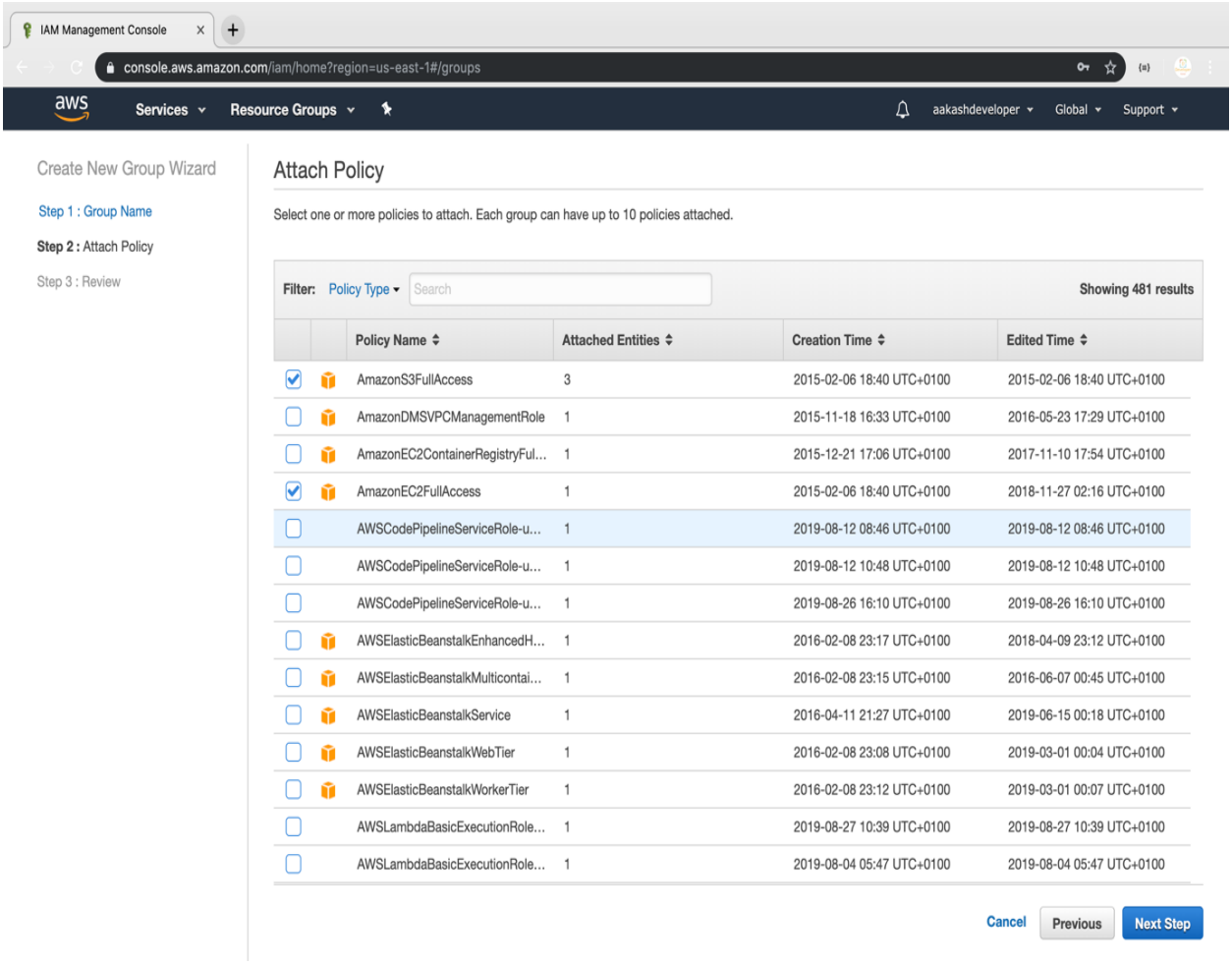


- Enter the group name

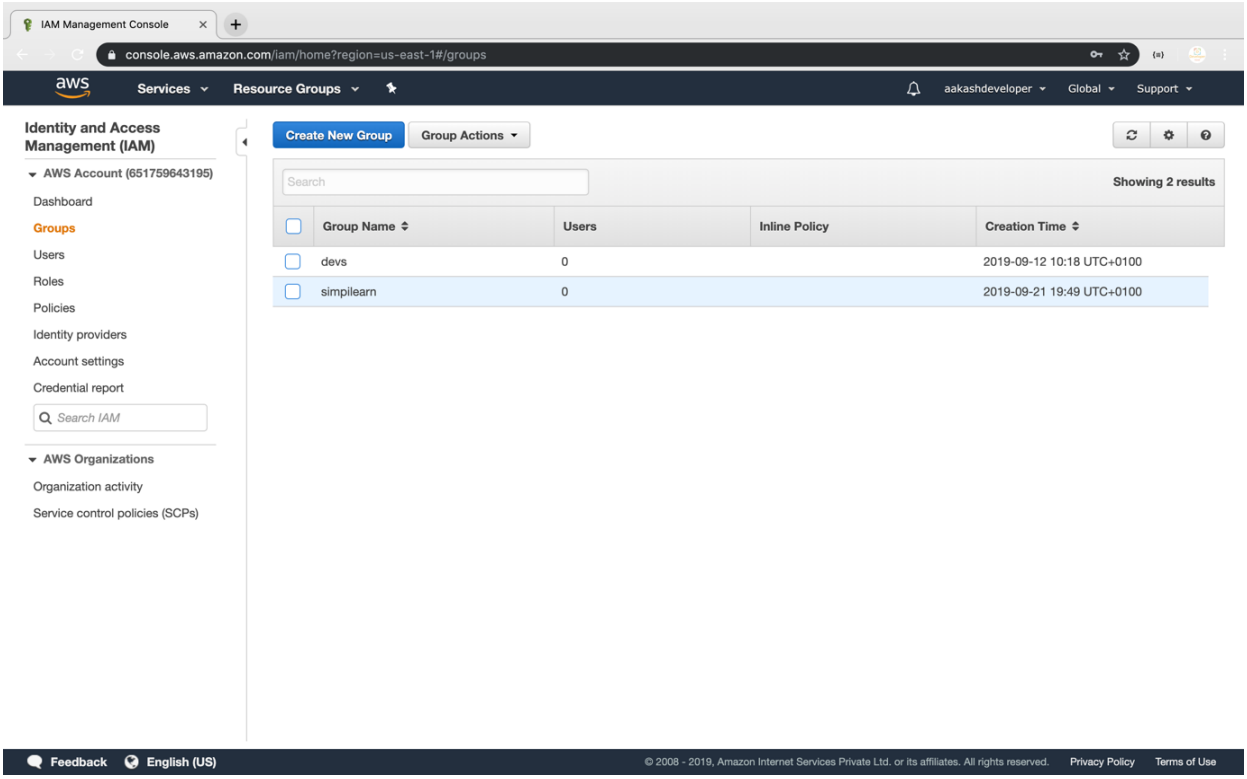


Step 3.17.2: Assigning the required policy to the group

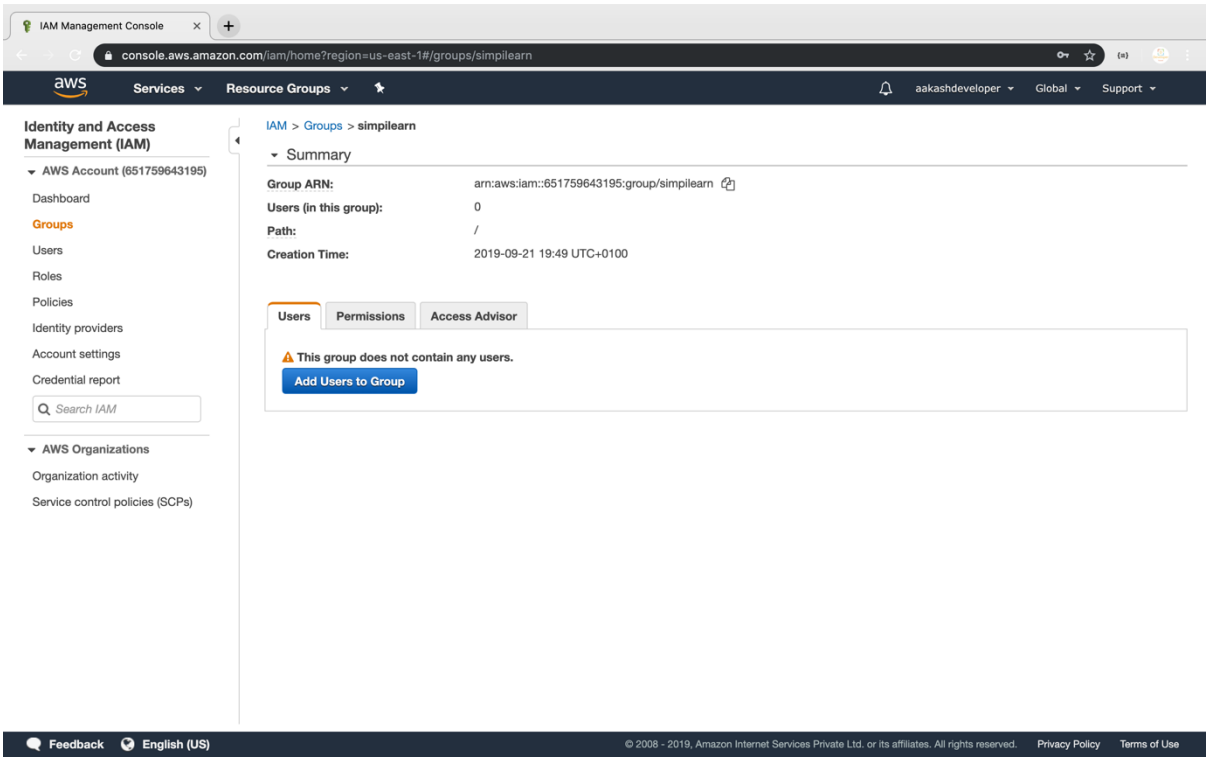
- Assign the policy provided by AWS



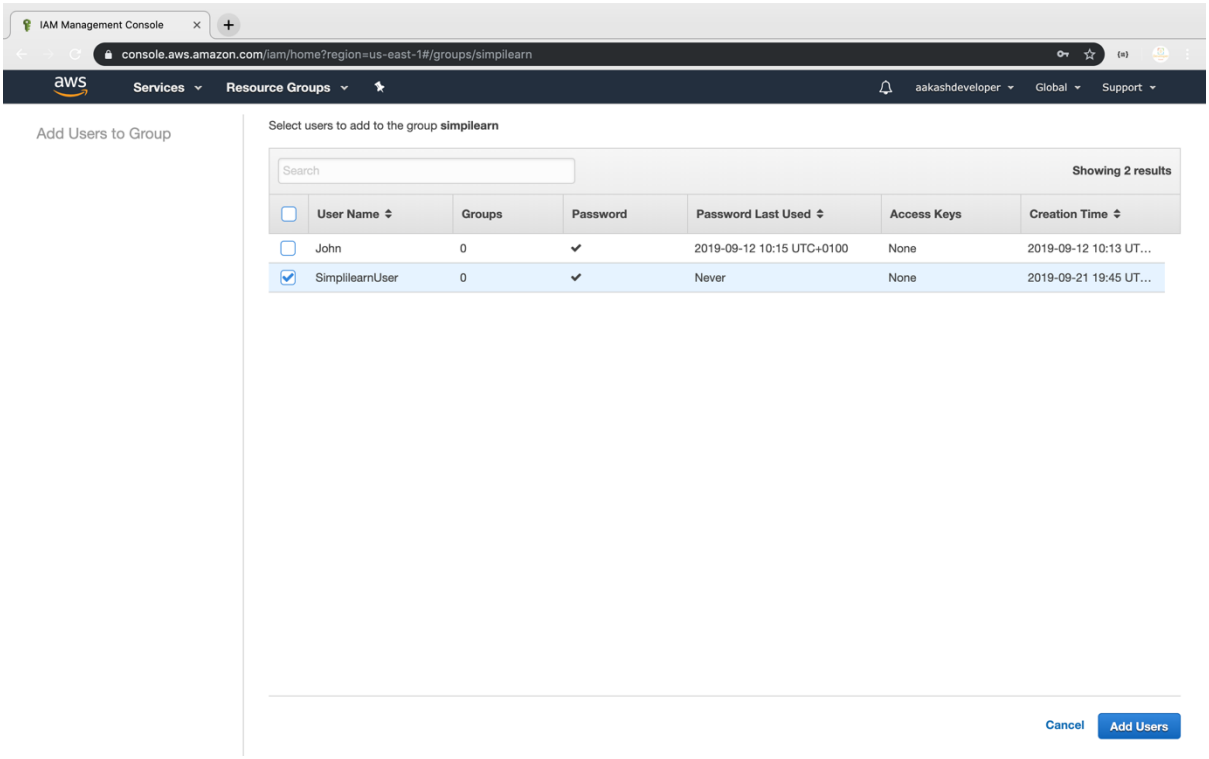
- Once the group is created, you can see the new group in the console



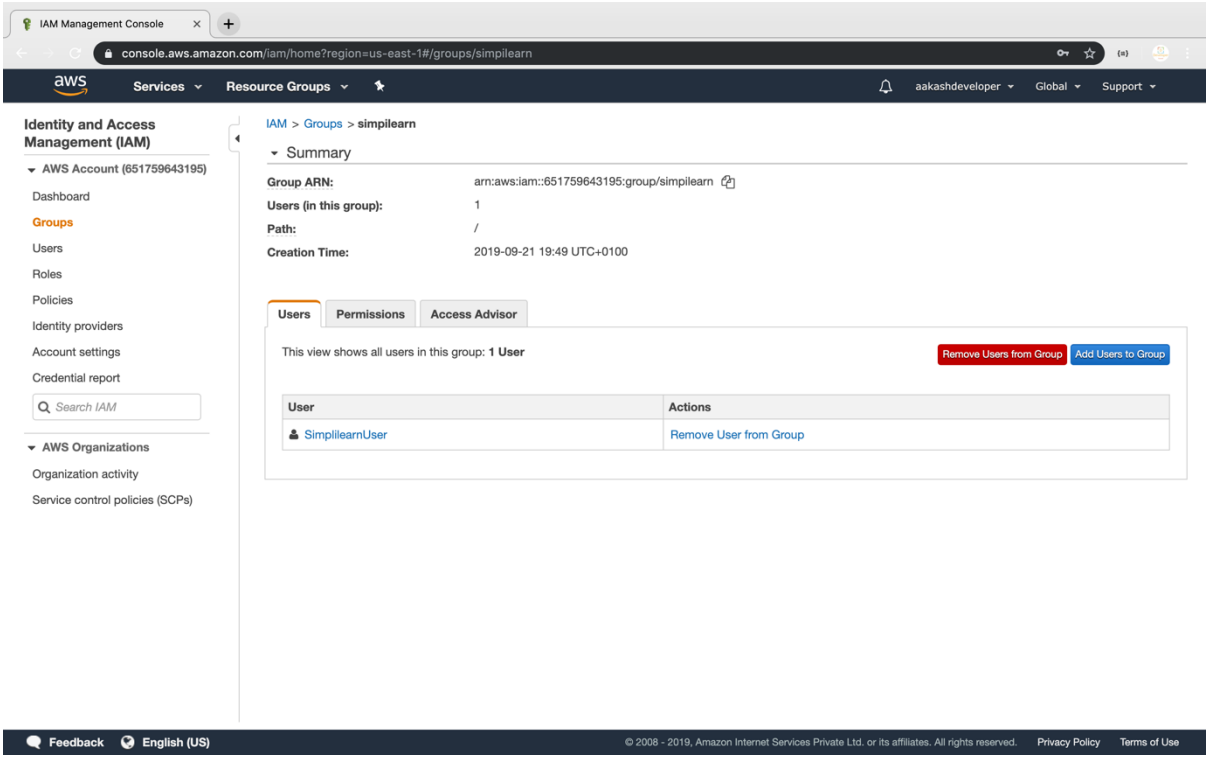
- You can add a user or a policy to the group any time



- You can see the user list by clicking on *Add User*



- You can also find all the policies added to the group



Step 3.17.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.39 Policies and Permissions

This section will guide you to:

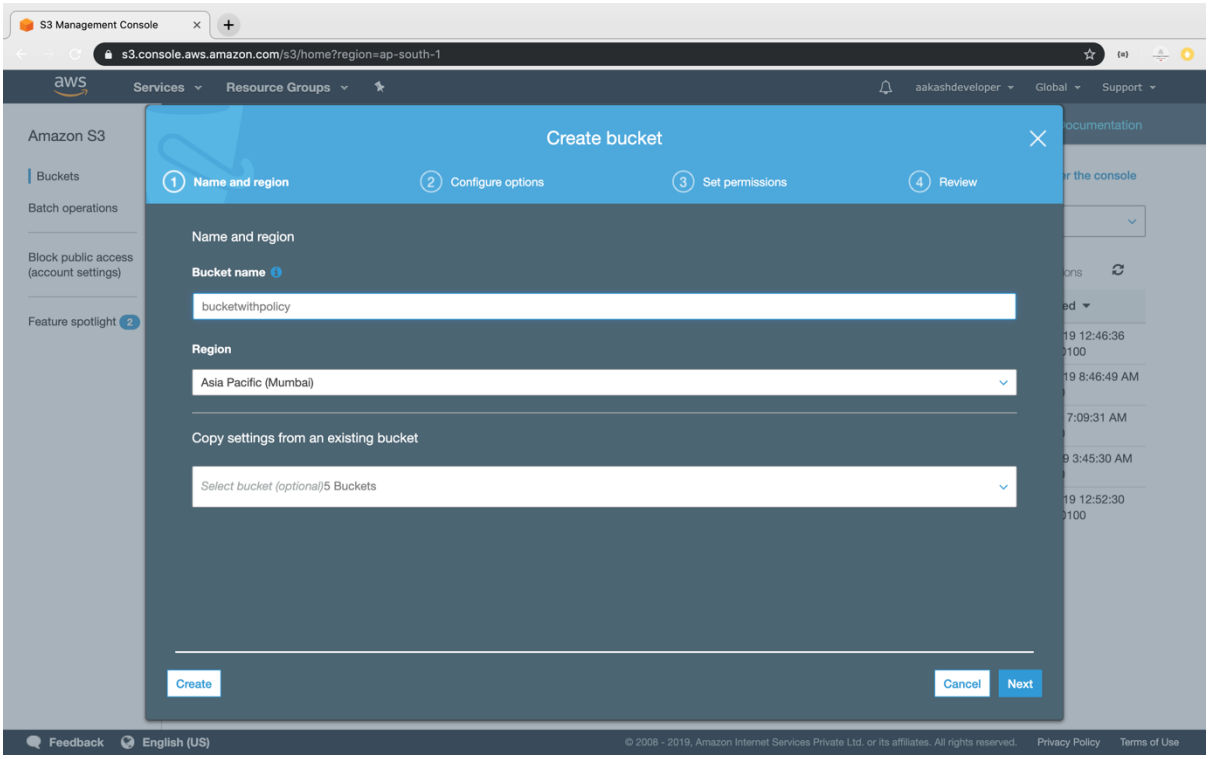
- Implement policies and permissions on an S3 bucket

This lab has three subsections, namely:

3.18.1 Creating a bucket
3.18.2 Adding policies
3.18.3 Pushing the files to GitHub repositories

Step 3.18.1: Creating a bucket

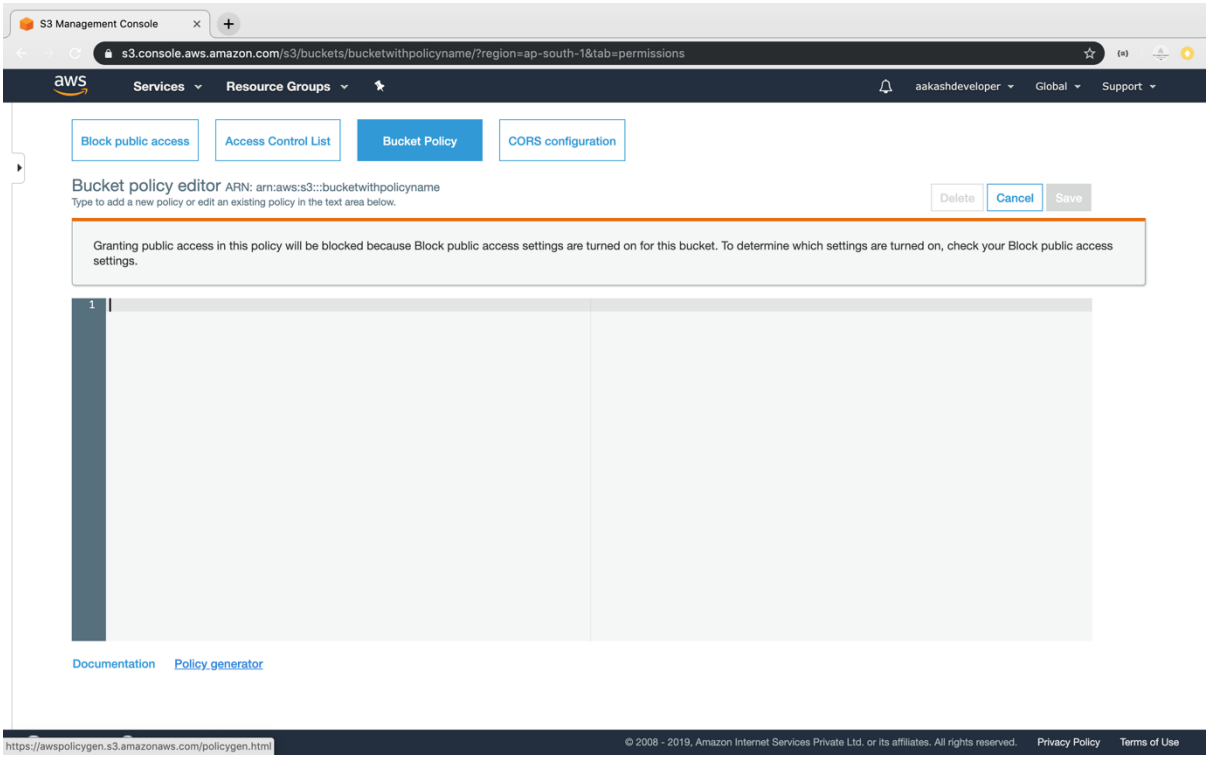
- Create a new bucket in which we want to add policy with all default settings.



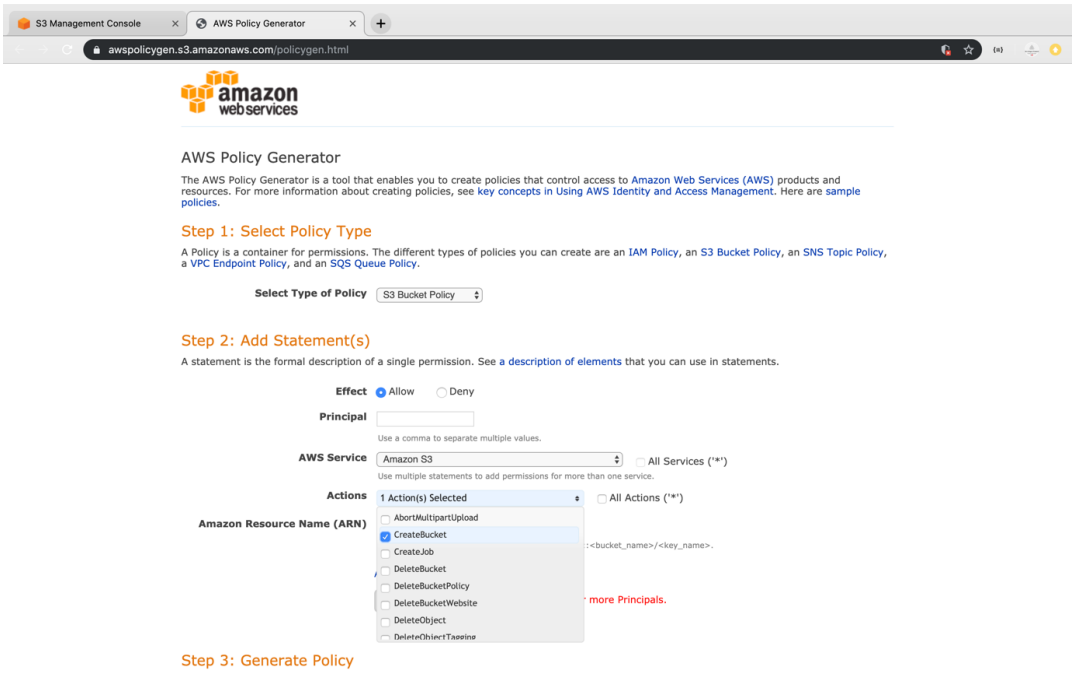
- Click on the *Permission* section.
- Go to *Bucket Policy*.
- It will open the console where you can add policy.

Step 3.18.2: Adding policies

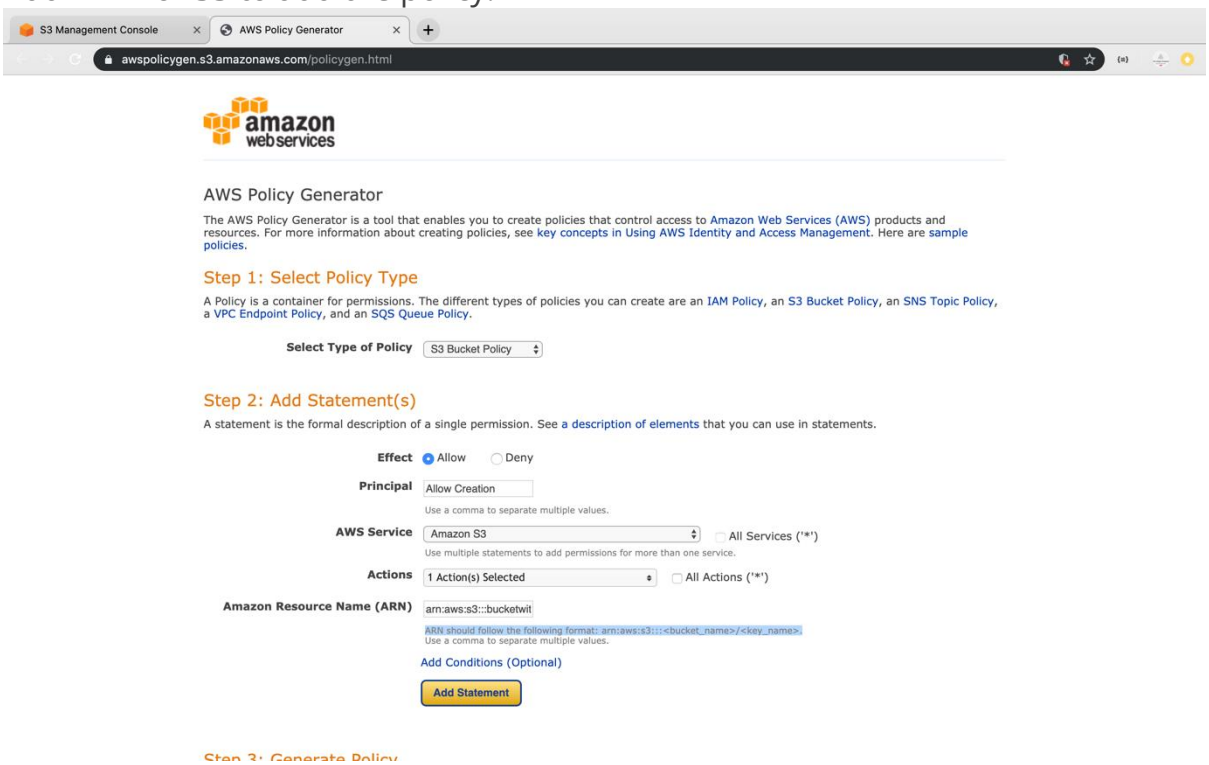
- Click on the *Policy Generator*.



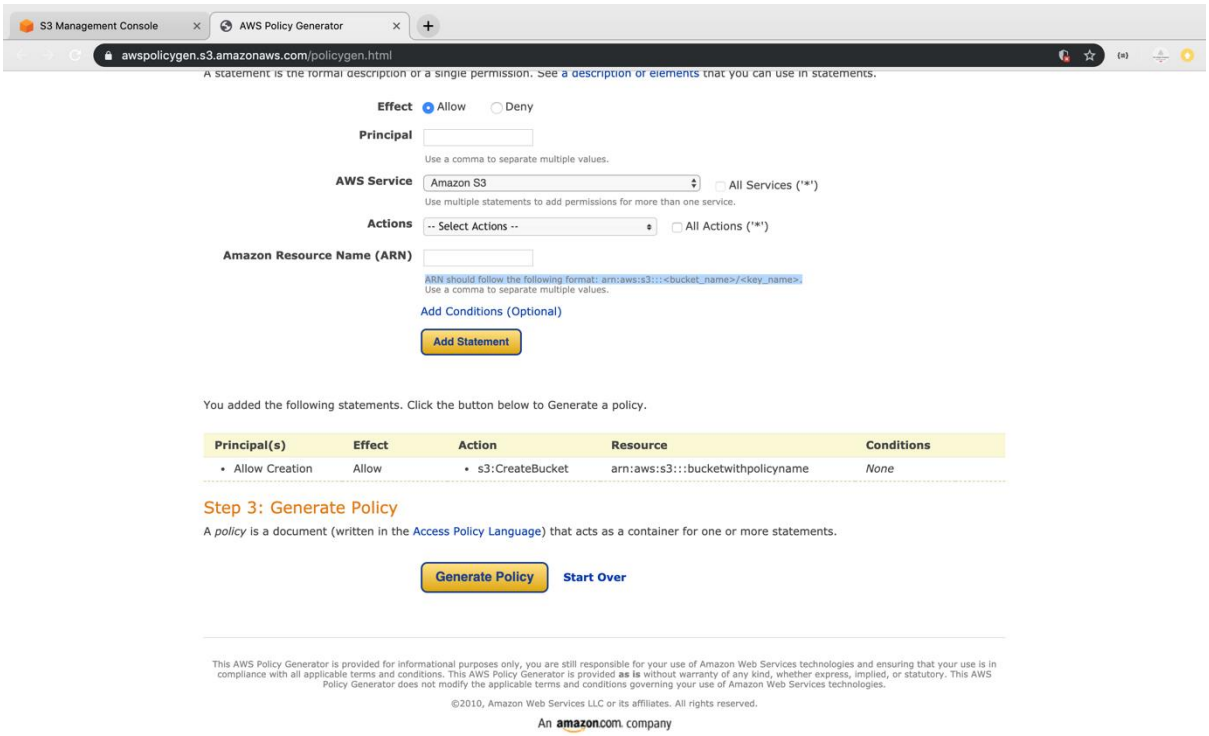
- Select the type of policy and services required.



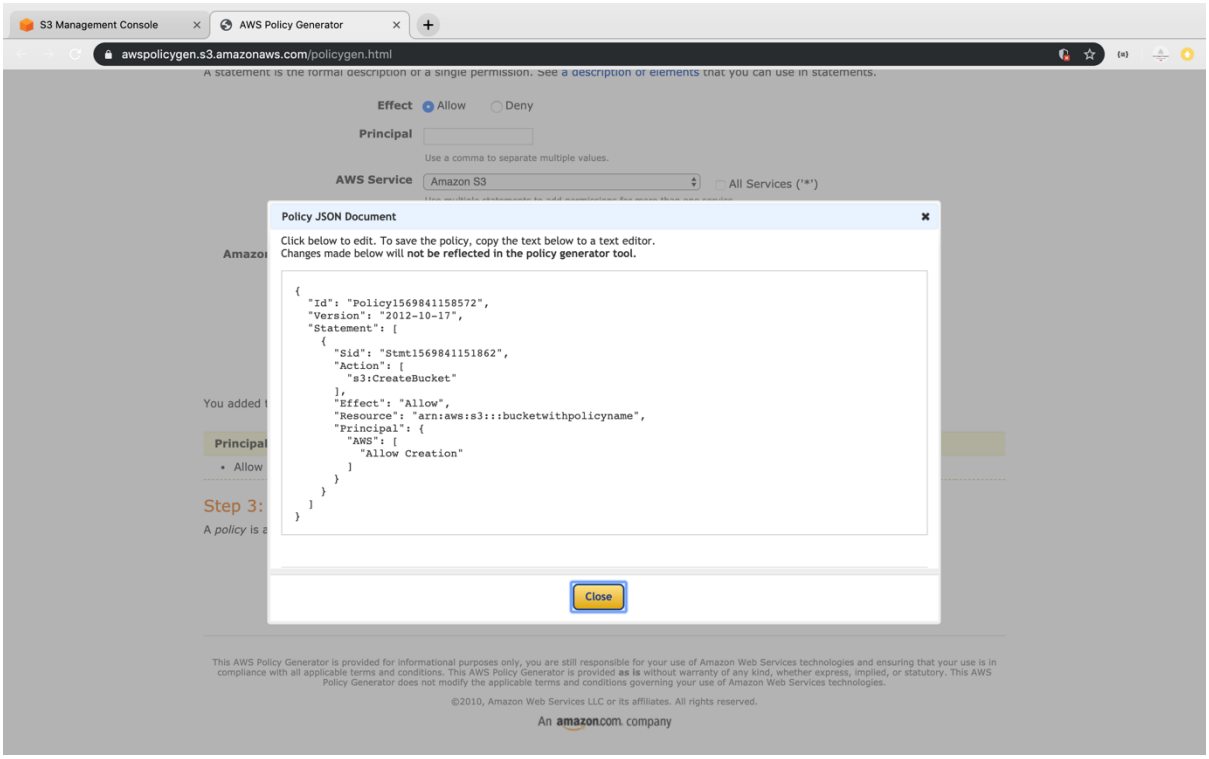
- Add ARN for S3 to add the policy.



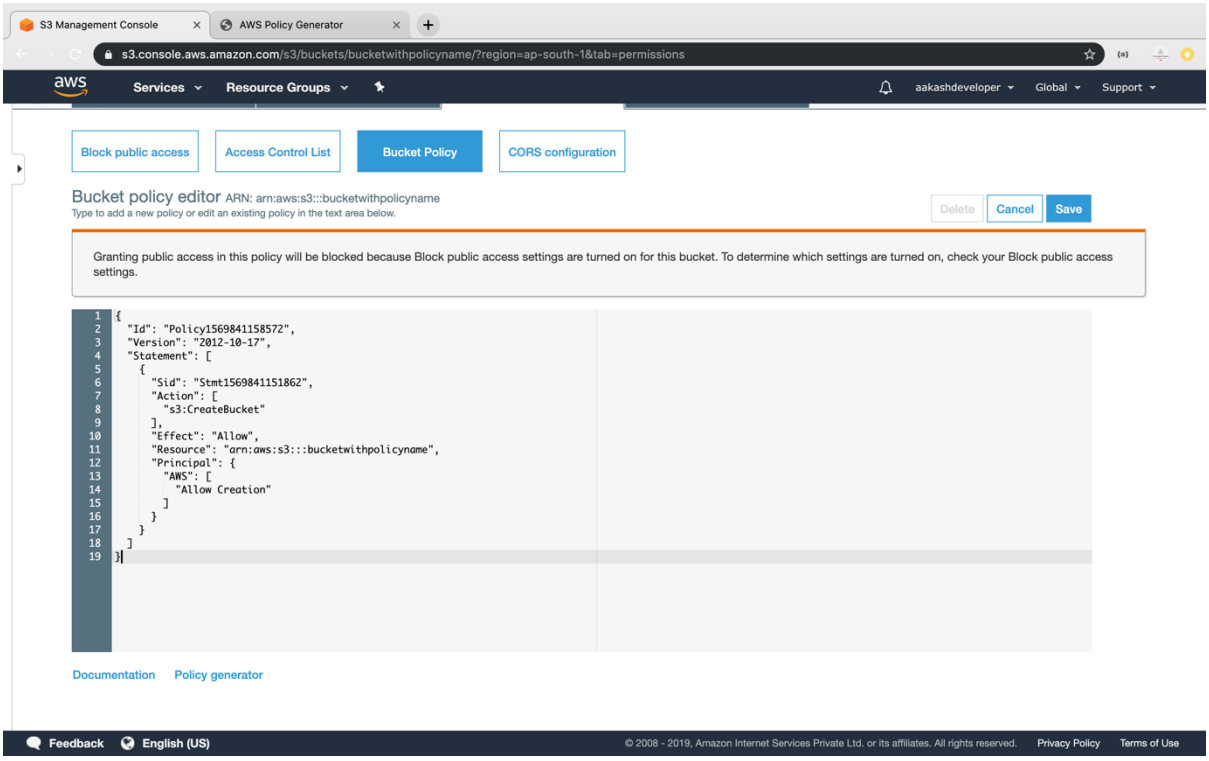
- Click on policy generator to create the JSON of the policy.



- Copy the generated JSON.



- Paste the JSON in the policy console of bucket and click on *Save*.



Step 3.18.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

  git push -u origin master

Assisted Practice: 3.41 Empty an S3 Bucket

This section will guide you to:

- Empty an S3 bucket

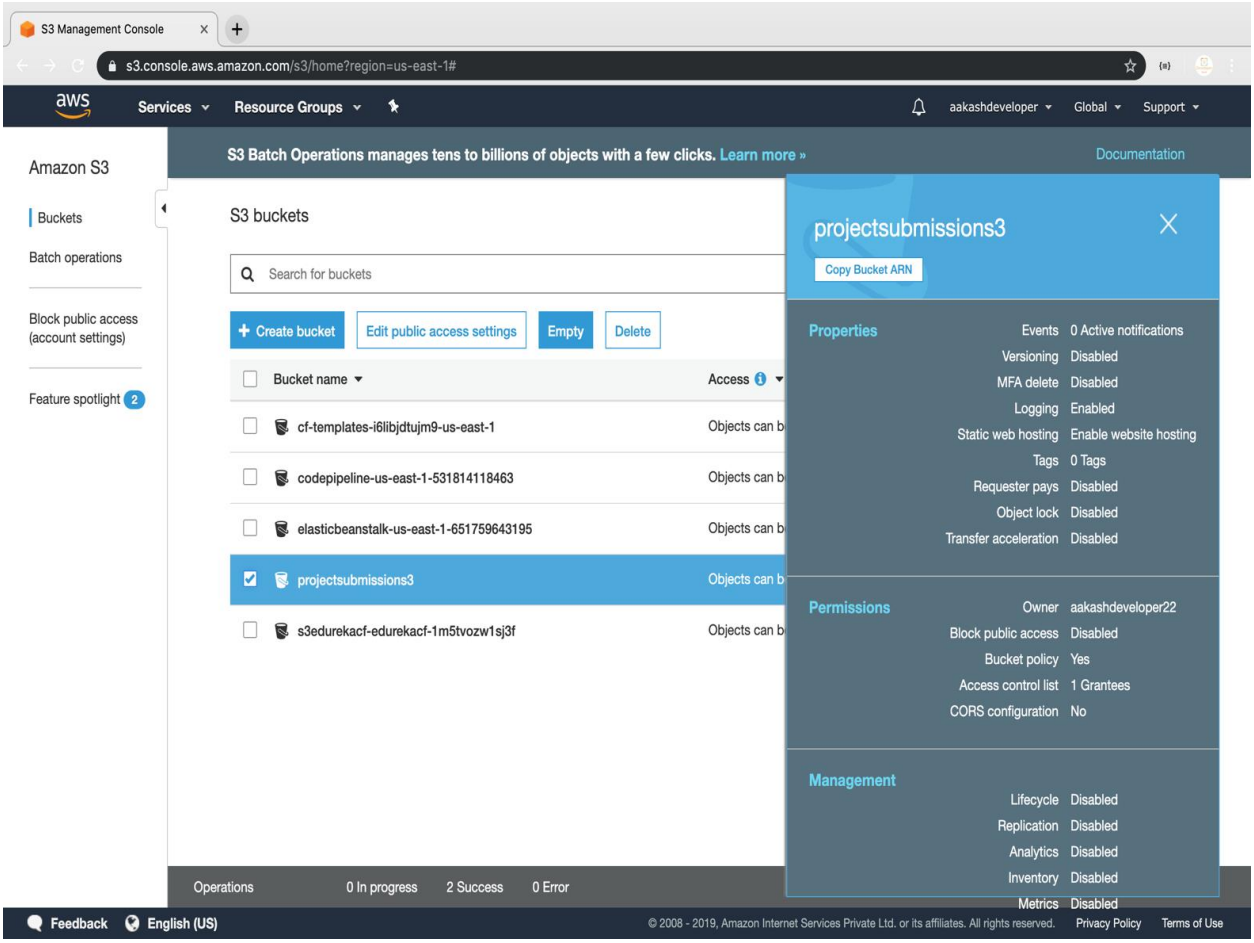
This lab has three subsections, namely:

3.13.1 Selecting the S3 bucket you want to empty

3.13.2 Emptying the bucket

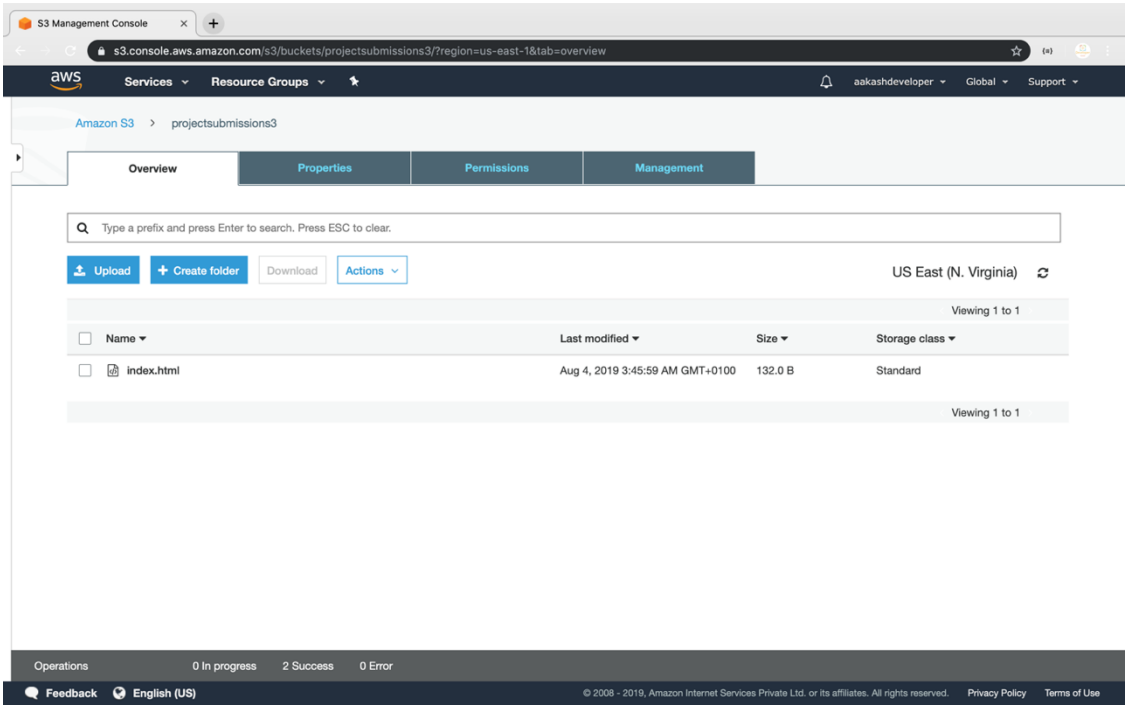
3.13.3 Pushing the files to GitHub repositories

Step 3.13.1: Selecting the S3 bucket you want to empty

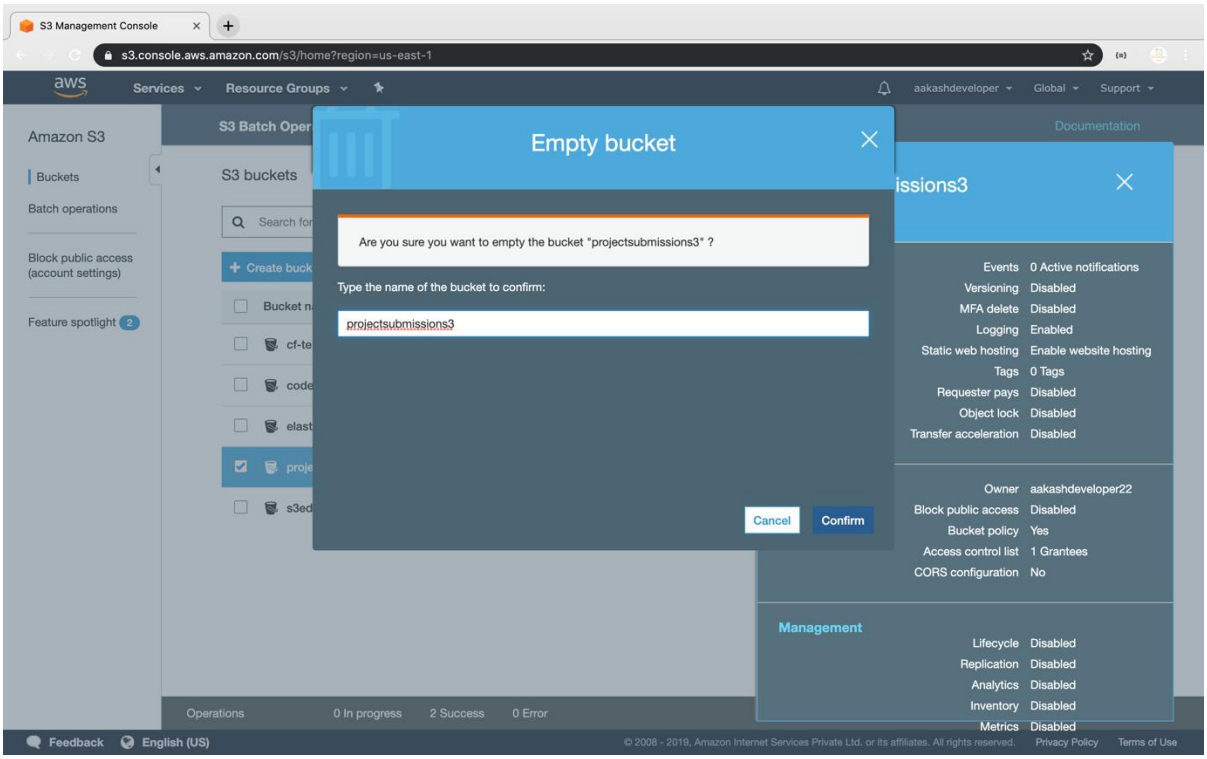


Step 3.13.2: Emptying the bucket

- Verify the content present in the bucket



- Empty the bucket



Step 3.13.3: Pushing the code to your GitHub repositories

- Open your command prompt and navigate to the folder where you have created your files.

  cd <folder path>

- Initialize your repository using the following command:

  git init

- Add all the files to your git repository using the following command:

  git add .

- Commit the changes using the following command:

  git commit . -m “Changes have been committed.”

- Push the files to the folder you initially created using the following command:

git push -u origin master