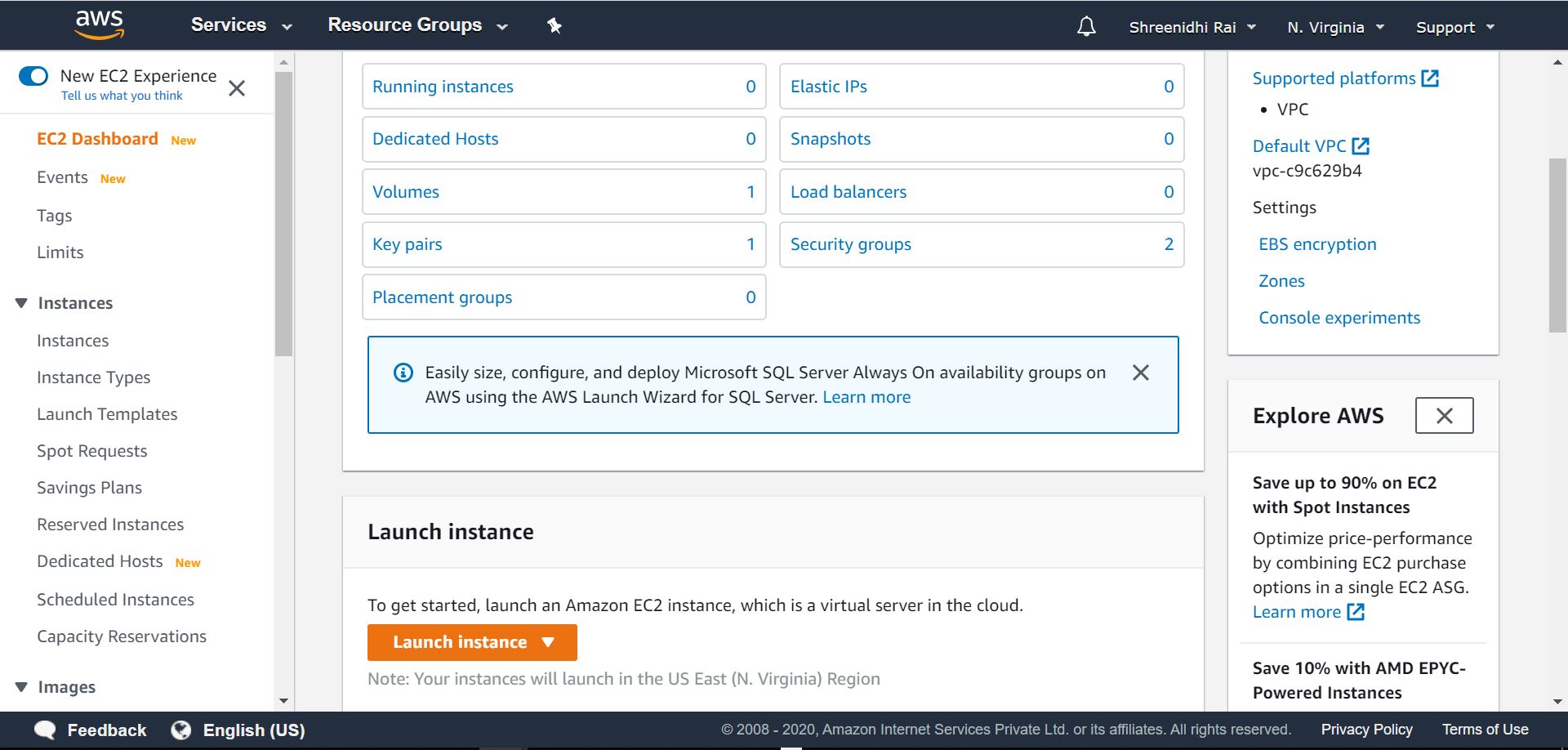
Deploy Spring-boot application in AWS EC2 server

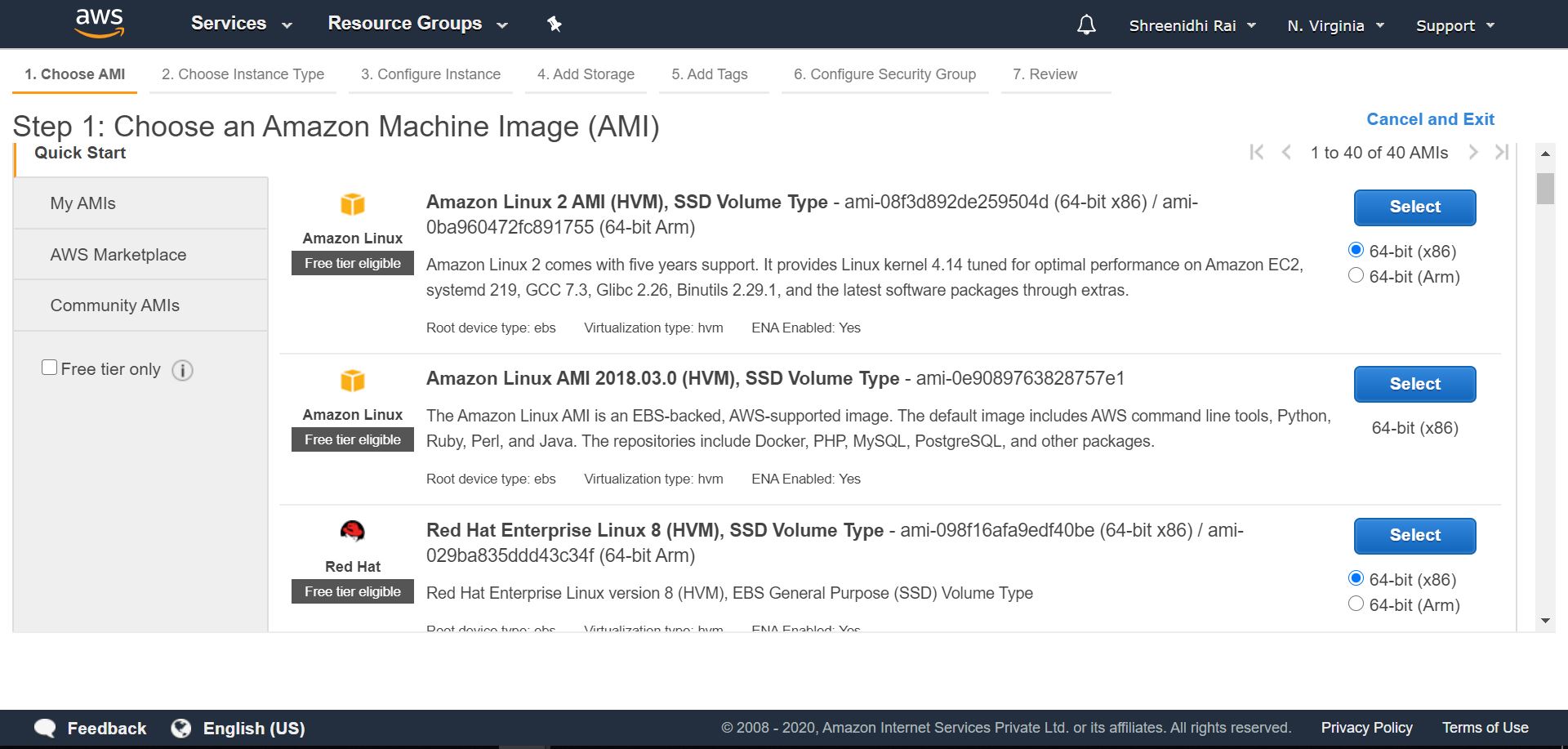
**Step 1 : Creating .jar file from existing Spring boot project.**

**Step 2 : Create EC2 instance and deploy the application.**

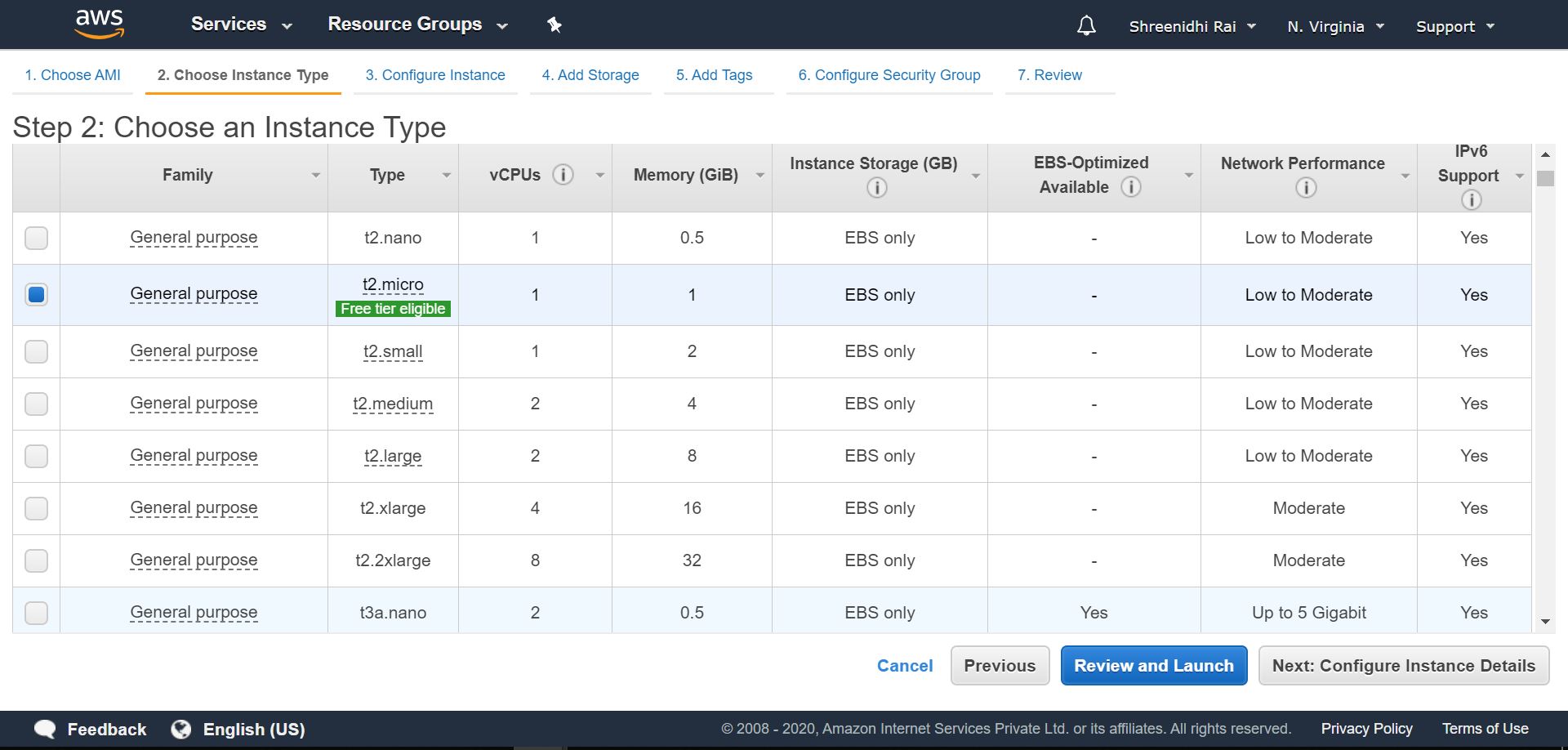
1. Login to AWS console and select EC2 service.
2. Now create Launch Instance button from the EC2 landing page.



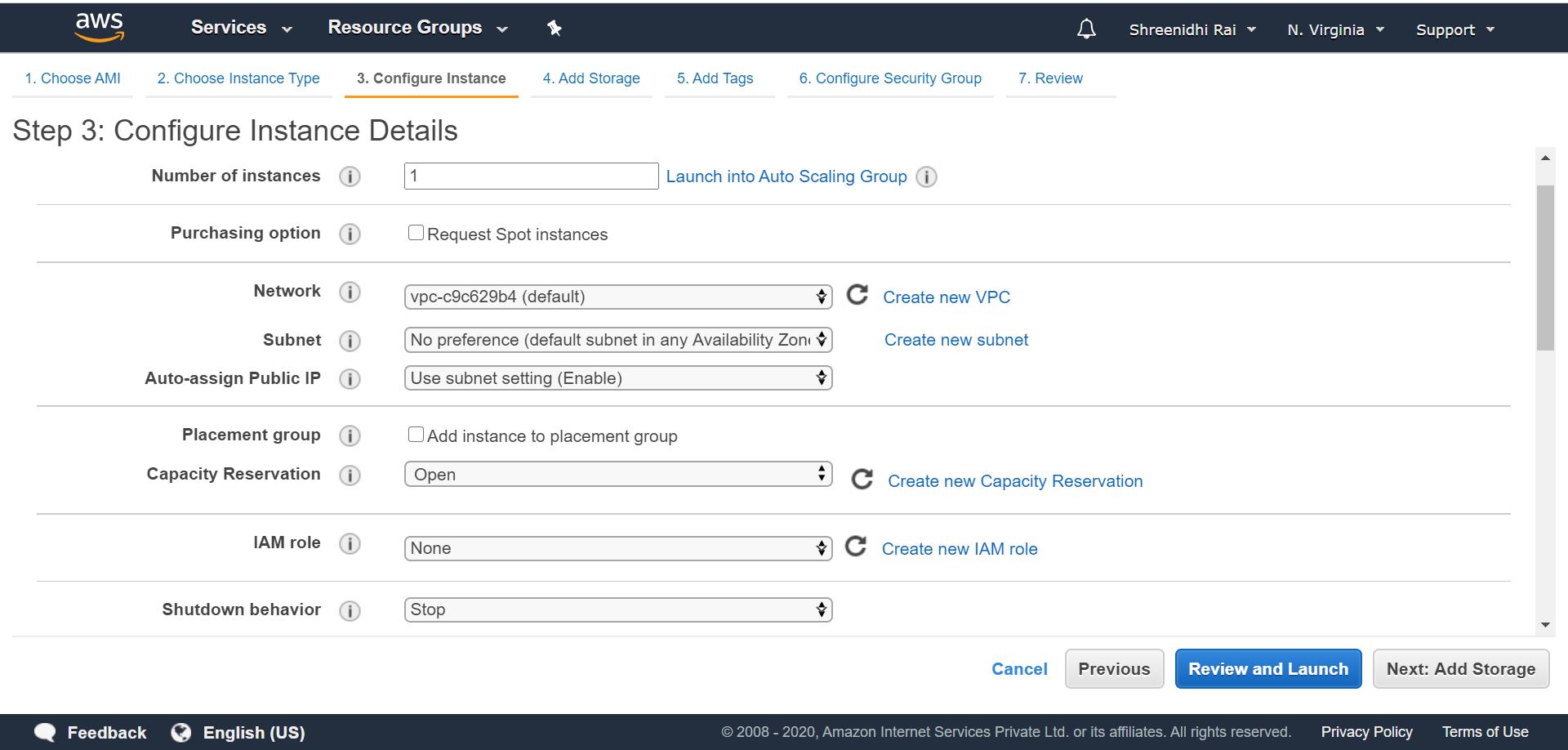
1. Then select the AMI and go to the next page.



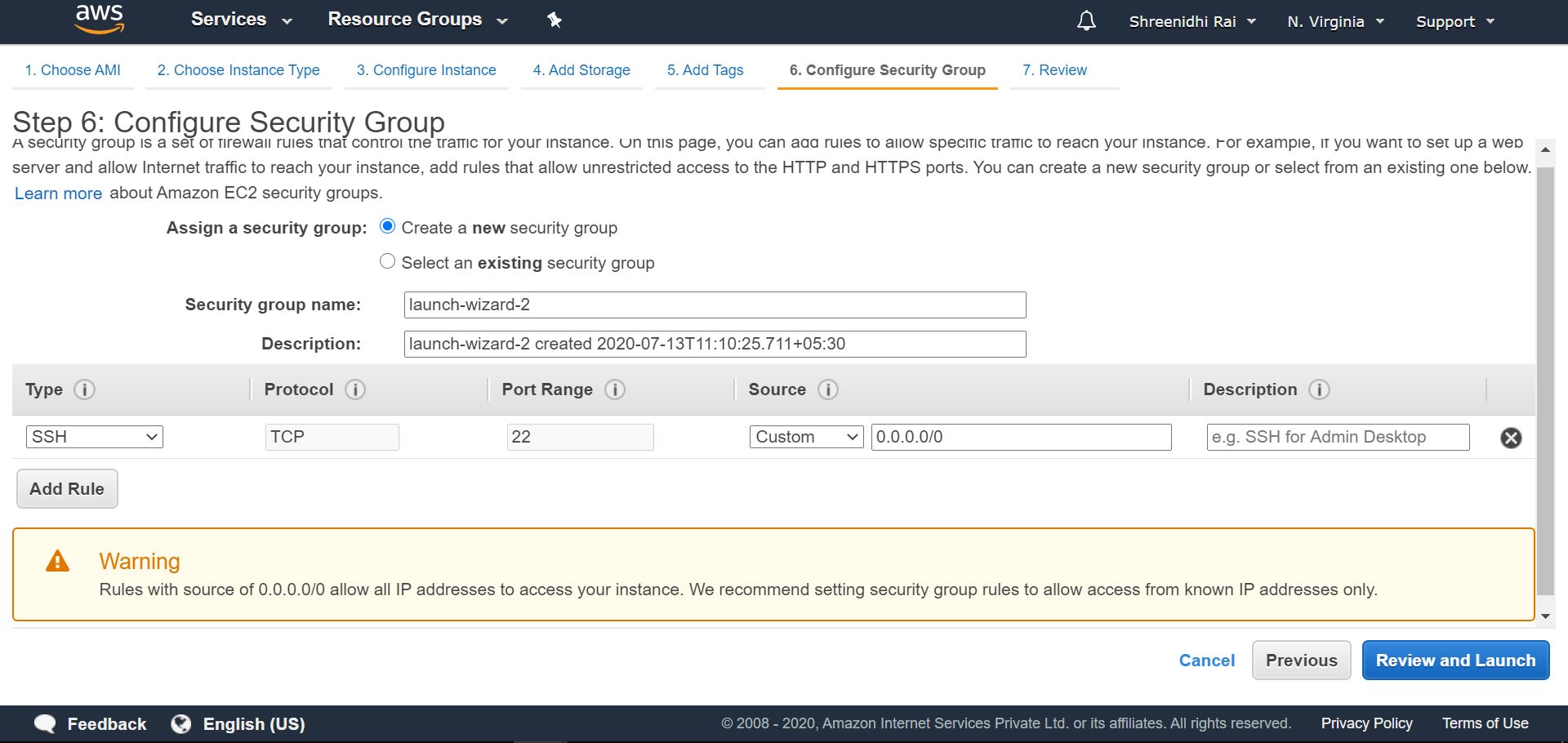
1. Now select the instance type (Choose only the free tire eligible, else charges will apply).



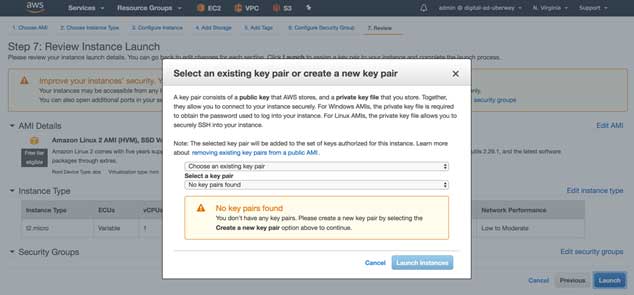
1. Now in instance details page we need to select two important things, rest of the things you can keep as default like VPC, subnet etc.



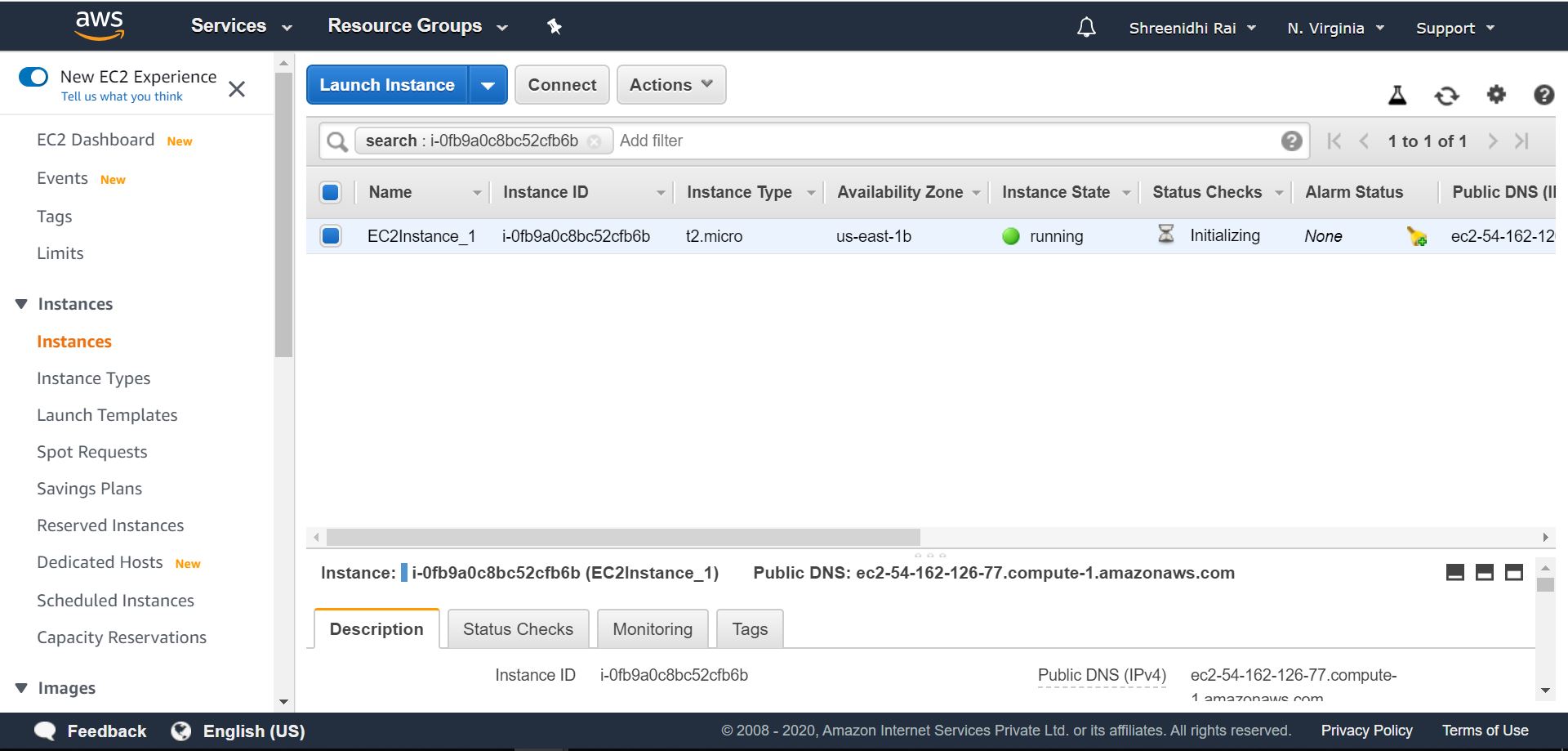
1. Now choose/edit the security group to allow HTTP traffic to port 80



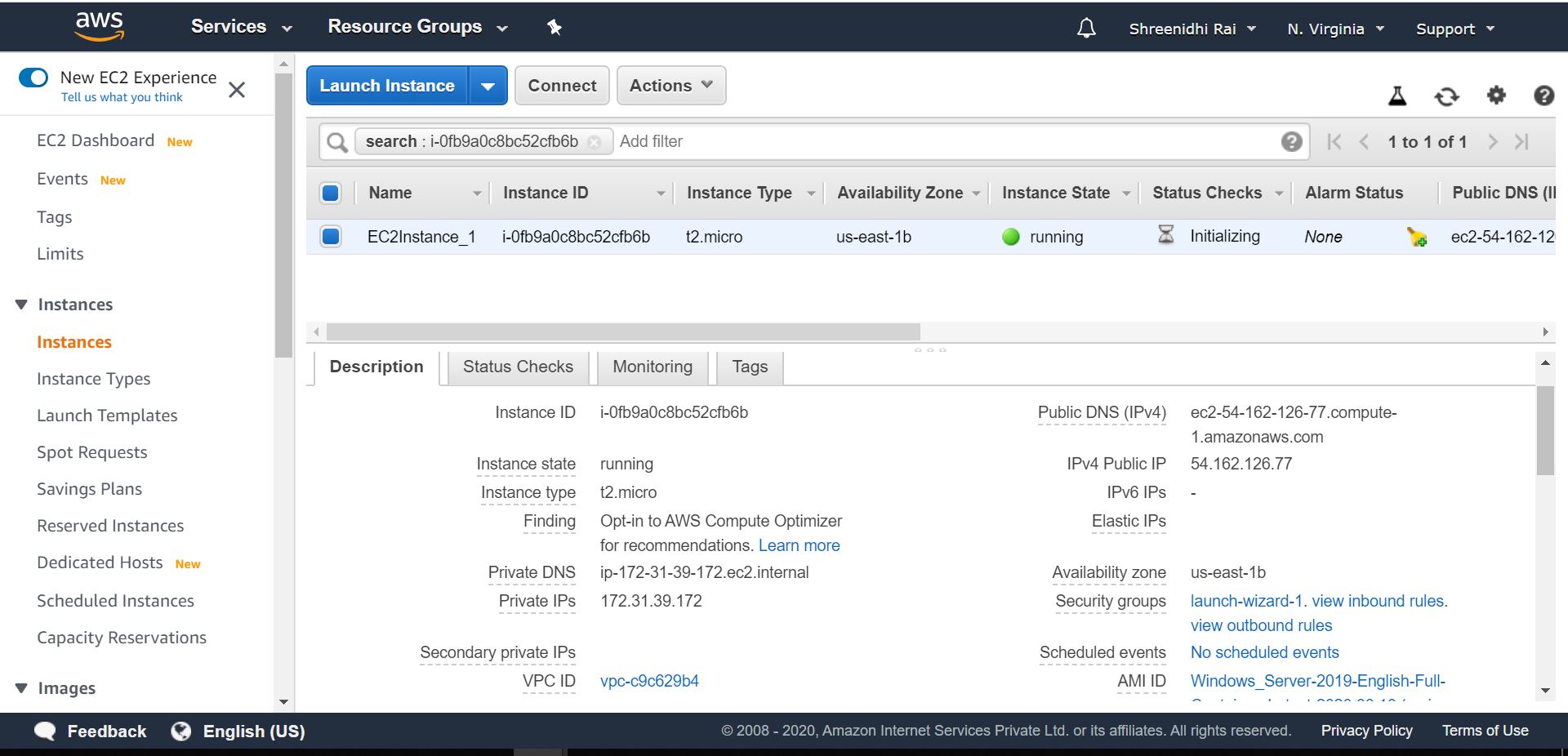
1. Finally choose one key pair and finish the instance creation step by clicking Launch Instance button.



1. Now go back to the EC2 landing page, you will see instance is being launched. Wait till instance state is running and all status checks are completed.

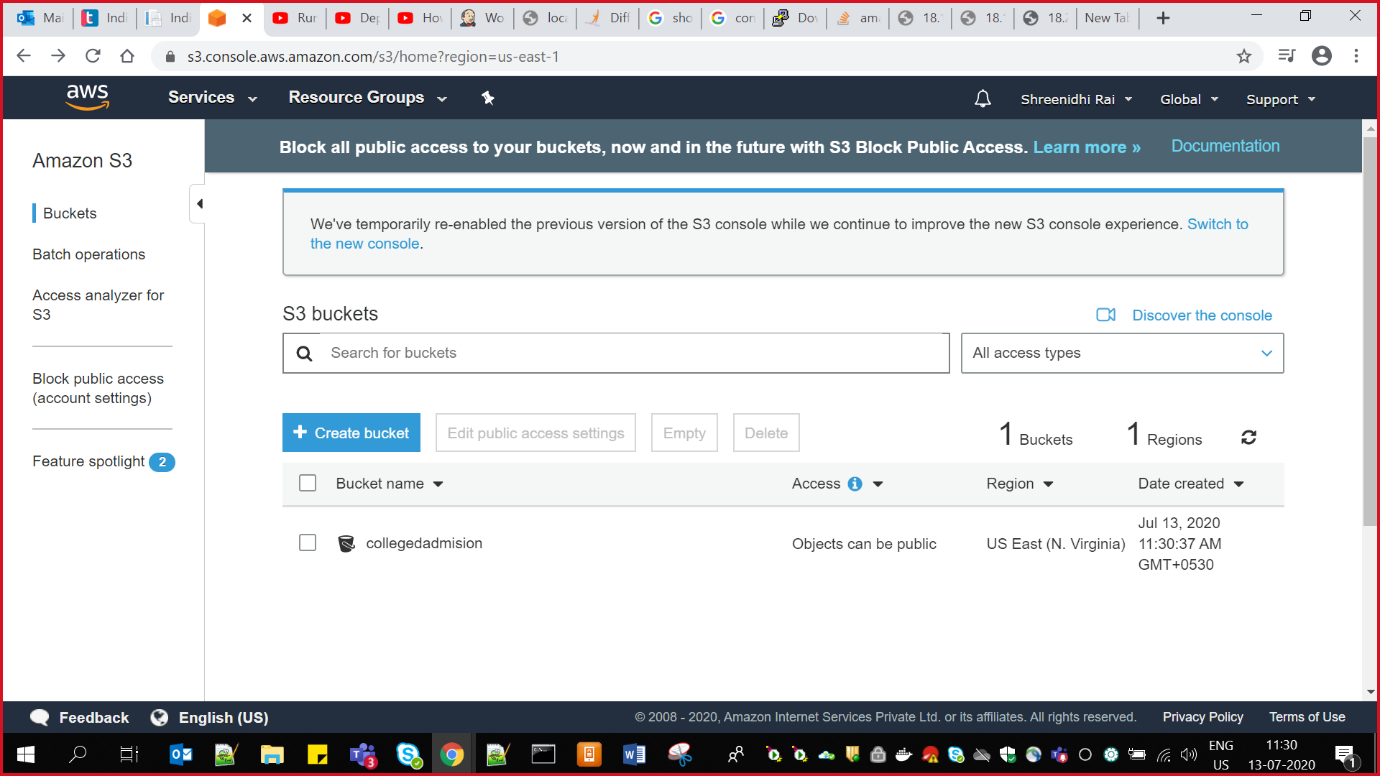


1. Now note down the public DNS/IP address from the instance description tab of the same page, we will use this DNS/IP to access the application.

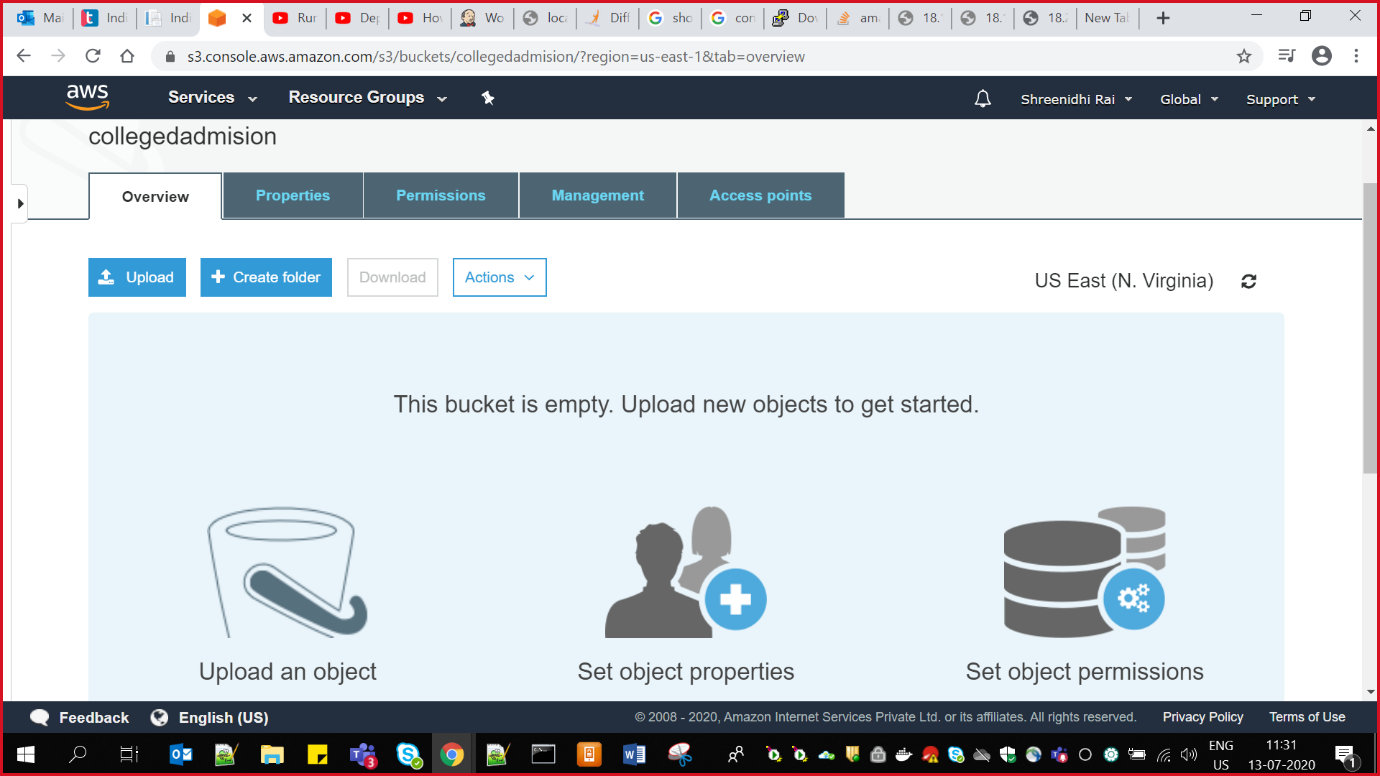


**Step 3 : create S3 bucket and upload jar file.**

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Bucket name** list, choose the name of the bucket that you want to upload your files to.

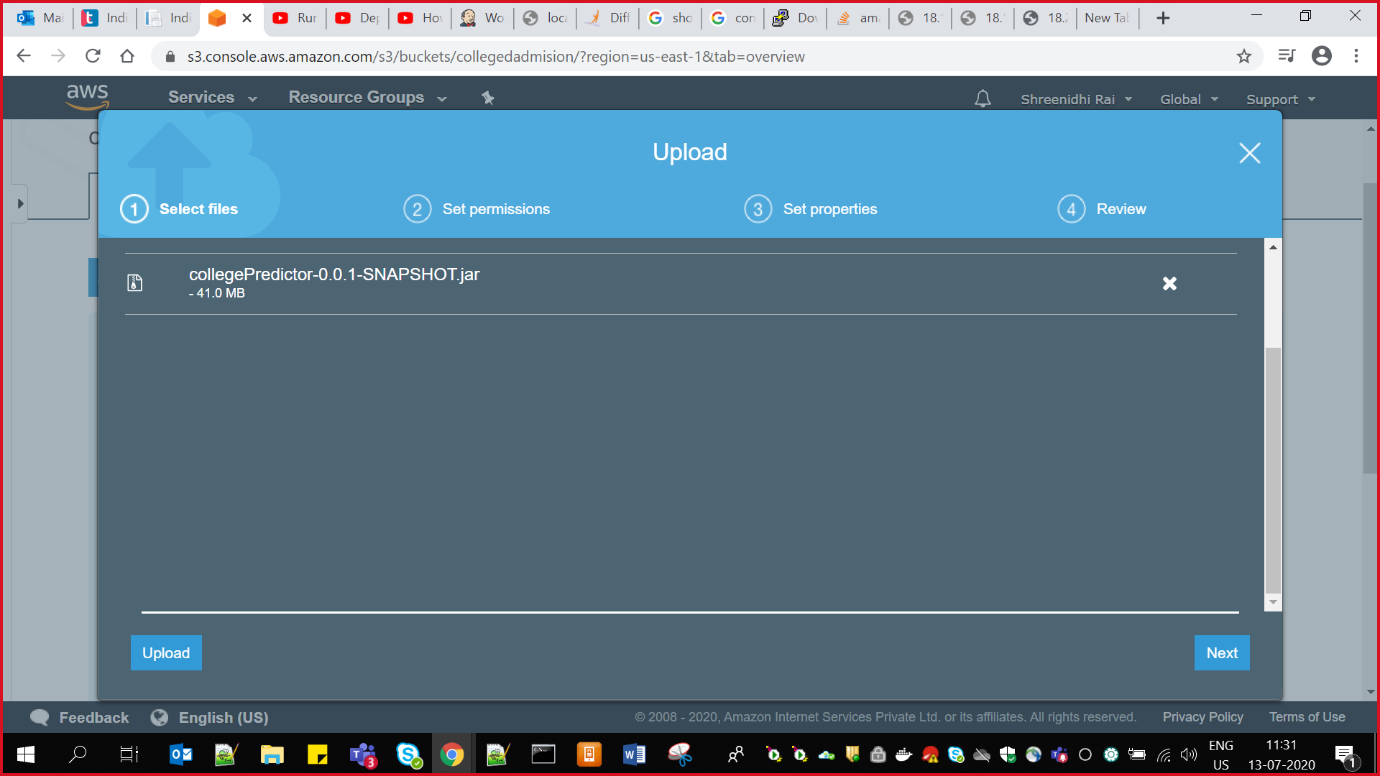


1. Choose **Upload**.
2. In the **Upload** dialog box, choose **Add files**.



1. Choose one or more files to upload, and then choose **Open.**
2. After you see the files that you chose listed in the **Upload** dialog box, do one of the following:

* To add more files, choose **Add more files**.
* To immediately upload the listed files, choose **Upload**.
* To set permissions or properties for the files that you are uploading, choose **Next**.



**Step 4 : Virtual installation of java, postgreSql.**

To install JAVA:

sudo apt update

java - version

sudo apt install default-jre

java - version

java -version

sudo apt install default-jdk

javac -version

**To Install PostgreSQL Commands:**

# Create the file repository configuration:

sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

# Import the repository signing key:

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Update the package lists:

sudo apt-get update

# Install the latest version of PostgreSQL.

# If you want a specific version, use 'postgresql-12' or similar instead of 'postgresql':

sudo apt-get install postgresql

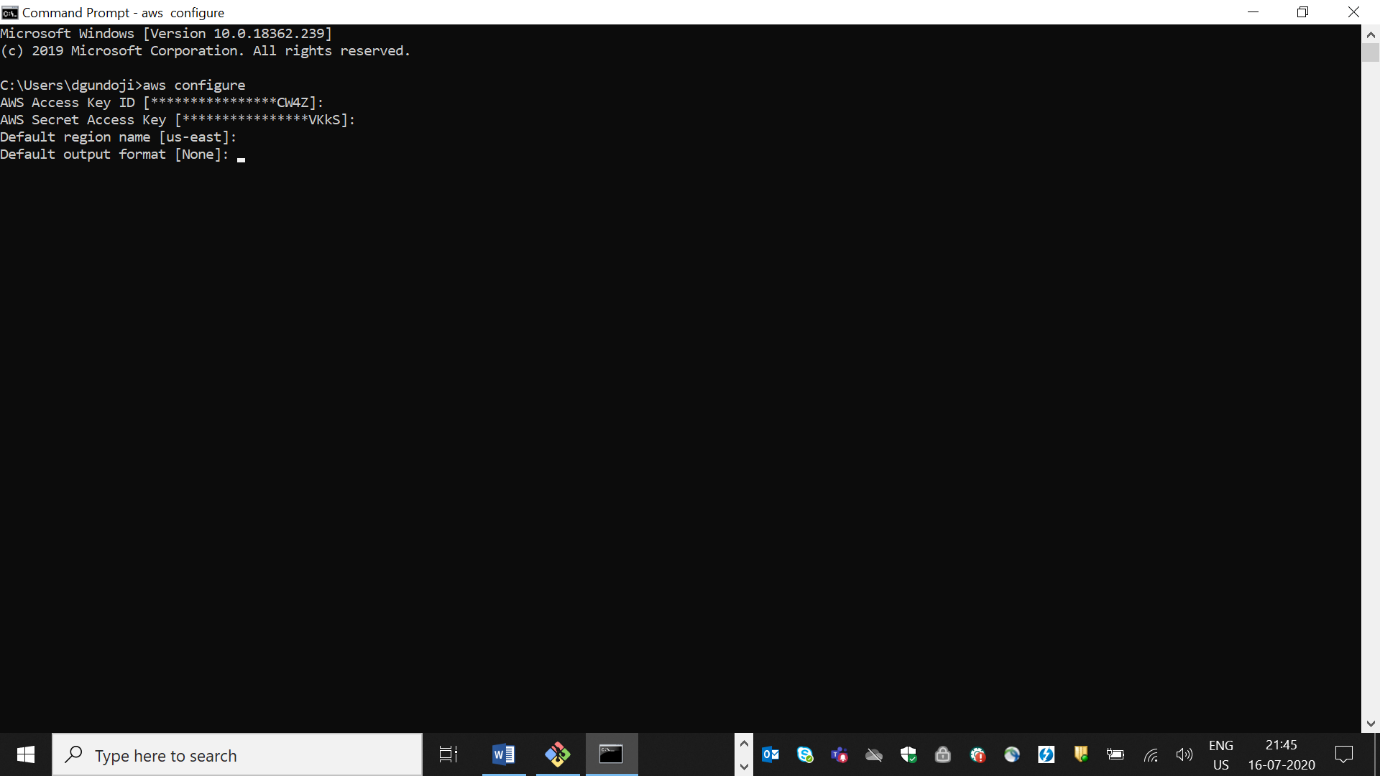
**Step 5: Getting jar file in to your virtual machine**

First we need to upload the jar in a bucket(s3), here main thing is if our jar file size is below 50mb you can go ahead otherwise we need to install aws cli and need to configure it with our aws account:

In order to download that go to : <https://awscli.amazonaws.com/AWSCLIV2.msi>

Then configure it with commands:

“aws configure” then you will see screen like:



We need to give those fields , where you can get AWS ACCESS KEY ID fromnew\_user\_credentials.csvfrom aws IAM then we need to create a role so we will get this new\_user\_credentials.csv there.

new\_user\_credentials.csv consists of AWS ACCESS KEY ID, AWS SECRET ACCESS KEY. Then we need to give region and output as per the requirement.

So that our local AWS cli will be connected with aws account. Then we can move our jar from local to aws bucket.

aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt

or

aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt

Then we can pull the bucket from s3 to virtual by using the command wget and pass resource URL.

i.e. wget <https://collegedadmision.s3.amazonaws.com/collegePredictor-0.0.1-SNAPSHOT.jar>

**Step 6: Run the application.**

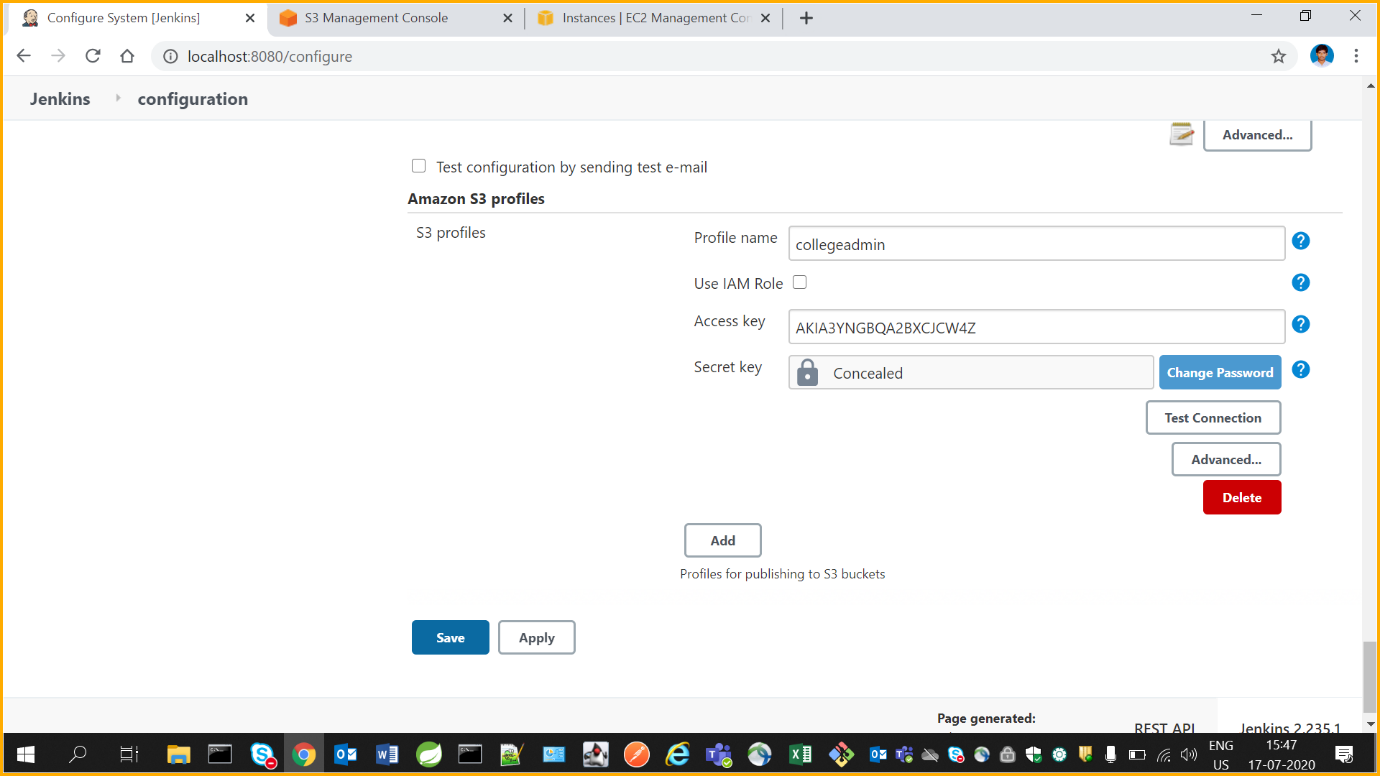
Use command – > java –jar collegePredictor-0.0.1-SNAPSHOT.jar (jar file name)

**Step 7 : Application is up, and deployed in AWS successfully.**

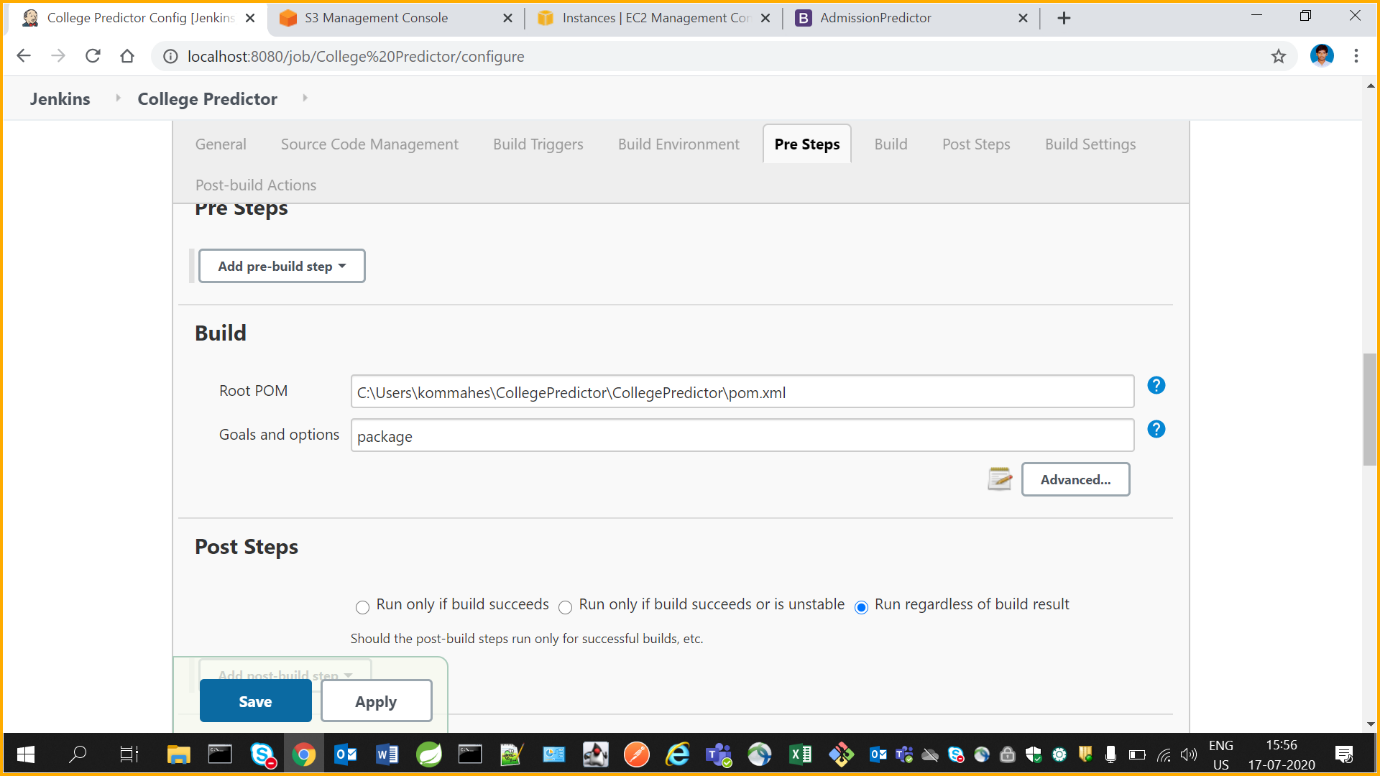


Deployment in AWS using Jenkins:

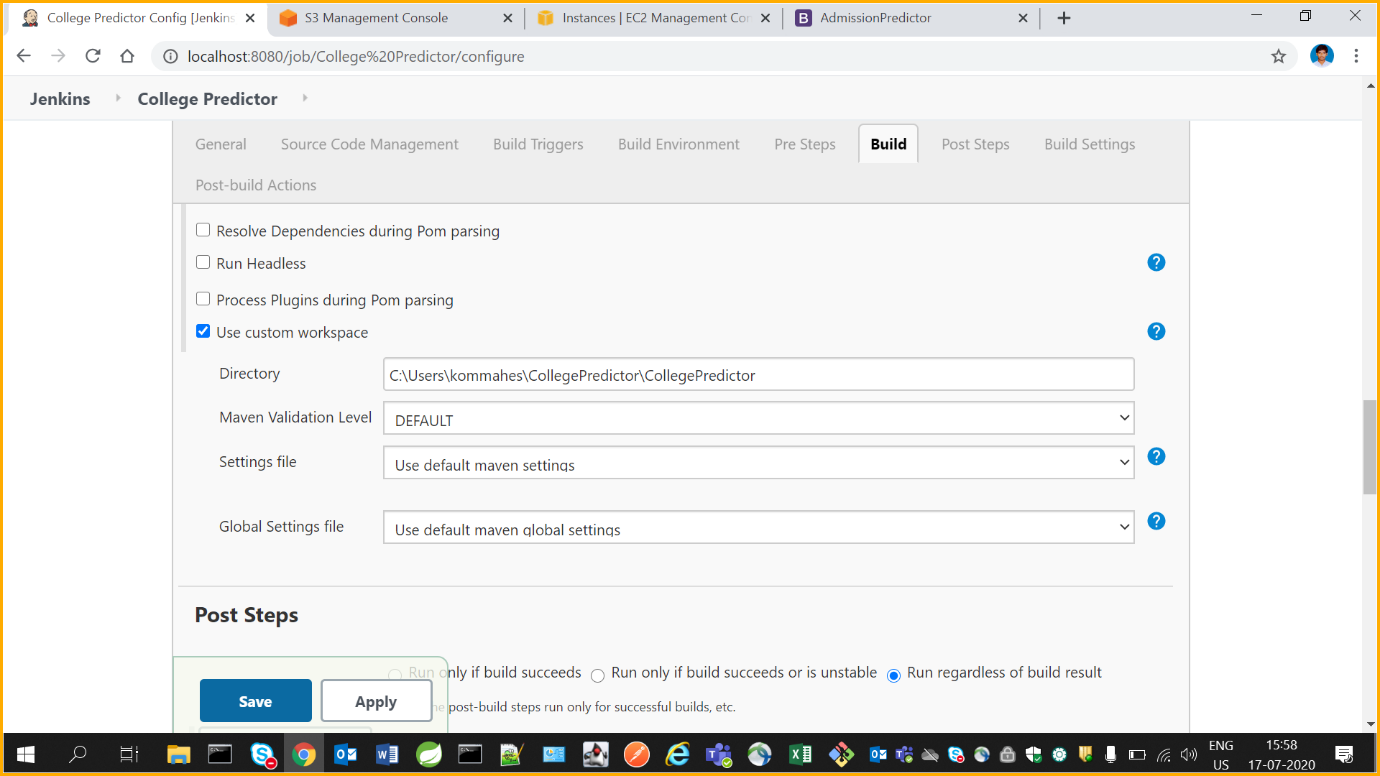
* Install s3 publisher plugin in Jenkins
* Go to manage Jenkins -> configure system -> **Amazon S3 profiles**
* Give Access key and secret key, profile name will be taken from IAM in aws users.



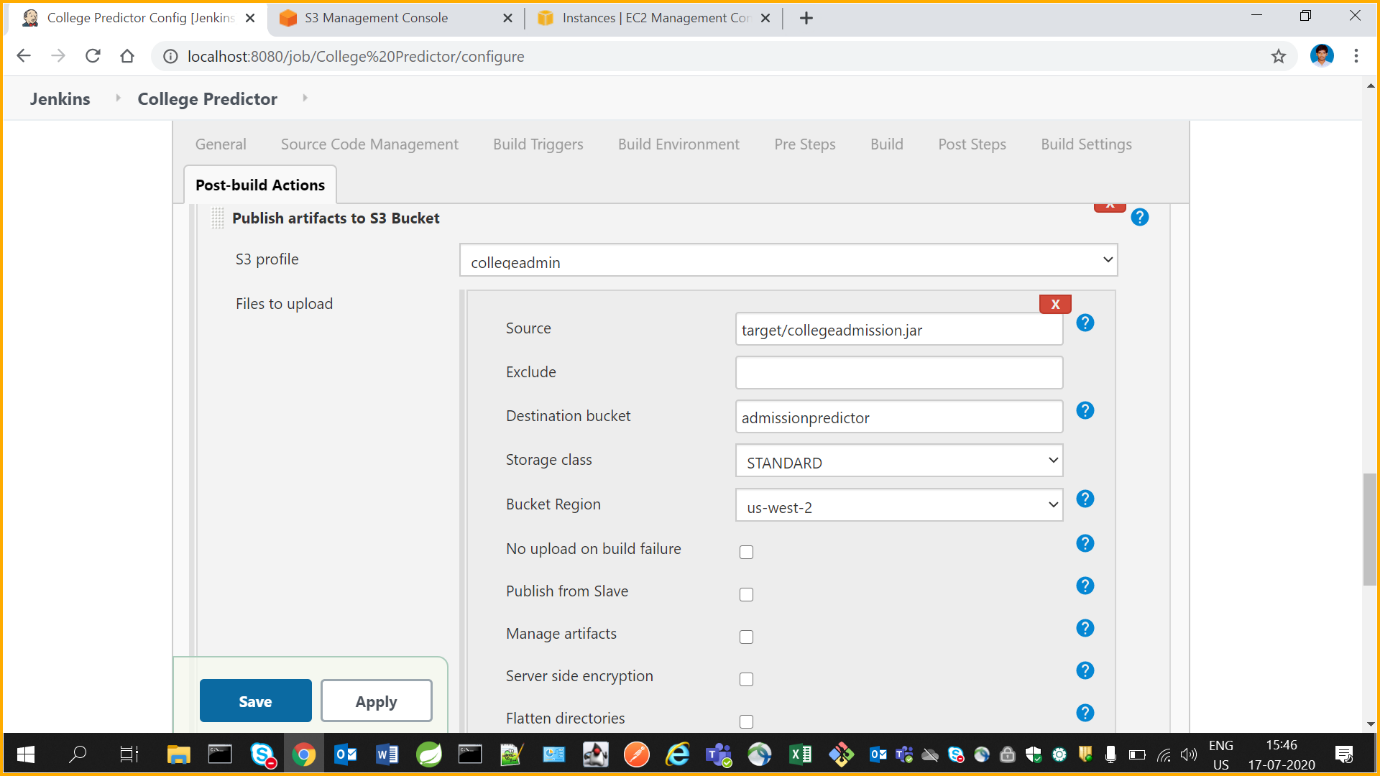
* Create a new Maven project
* Give POM.xml file path in Build and in goals give as package



* Click on advanced and check use custom workspace and give workspace path



* In Post build actions



* Give build
* After successful build jar will be deployed into S3 bucket
* Then we can pull the bucket from s3 to virtual by using the command wget and pass resource URL.
* i.e. wget <https://collegedadmision.s3.amazonaws.com/collegePredictor-0.0.1-SNAPSHOT.jar>
* Use command – > java –jar collegePredictor-0.0.1-SNAPSHOT.jar (jar file name)
* Then we can copy the instance public IP: port number of our application and run in browser

