**CI/CD Deployment for Springboot Application.**

1. **CI/CD Deployment for Spring boot for Repositories:**

**This document contains sections for:**

* Project Description
* [Core concepts used in project](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Core_concepts)
* [Flow of the Application](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Flow).
* Project Users Stories : ( Agile and Scrum )
* Git Repositories
* How to run project
* [Demonstrating the product capabilities, appearance, and user interactions.](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Product_capability)
* [Unique Selling Points of the Application](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#USP)
* [Conclusions](file:///C:\Users\Asus\Downloads\LockedMe%20-%20Virtual%20Key%20for%20Repositories.docx#Conclusions)

The code for this project is hosted at :

<https://github.com/Nachikethrayapura/Phase-V>

* 1. **Project Description:**

**Project Objective:**

As a Full Stack Developer, you have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

**Background of the problem statement:**

As the project is in the final stage, management has asked you to automate the integration and deployment of the web application. You are required to set up an environment where the application will be hosted and accessed by users. The source code is supposed to be fetched from a GitHub repository.

**You must use the following:**

* Eclipse
* GitHub
* Jenkins
* AWS EC2/ Virtual machine

**Following requirements should be met:**

* A part of the source code should be tracked on the GitHub repository. You need to document the tracked files that are ignored during the final push to the GitHub repository.
* The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document.
* The step-by-step process involved in completing this task should be documented.
  1. **Core concepts used in project:**

Launching and connect to an EC2 Instance

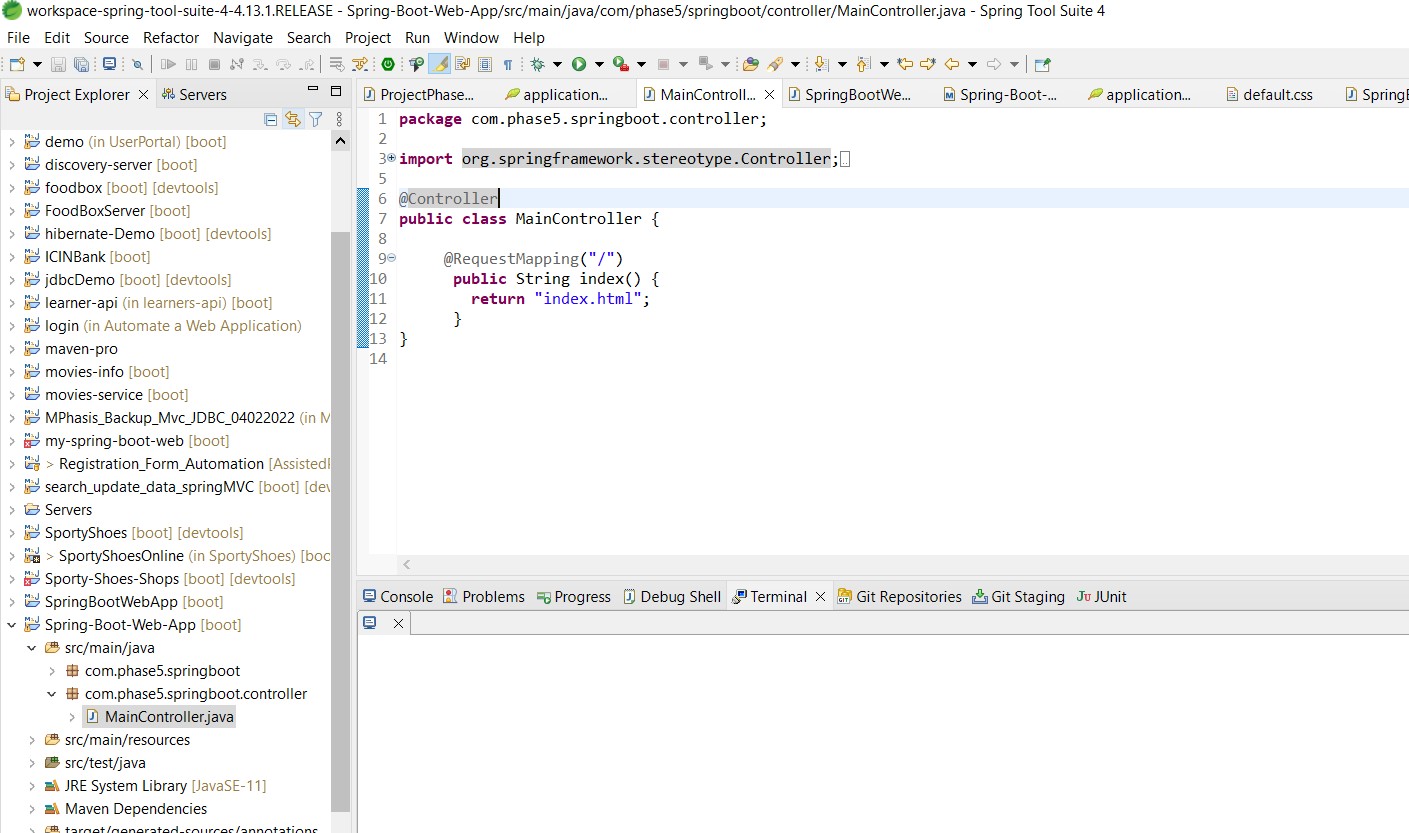
This lab has three subsections, namely:

1. Write spring boot program.
2. Launching an EC2 instance
3. Connecting to the EC2 instance
4. Creating S3 Bucket.
5. Added jar file in bucket.
6. Pushing the files to GitHub repositories

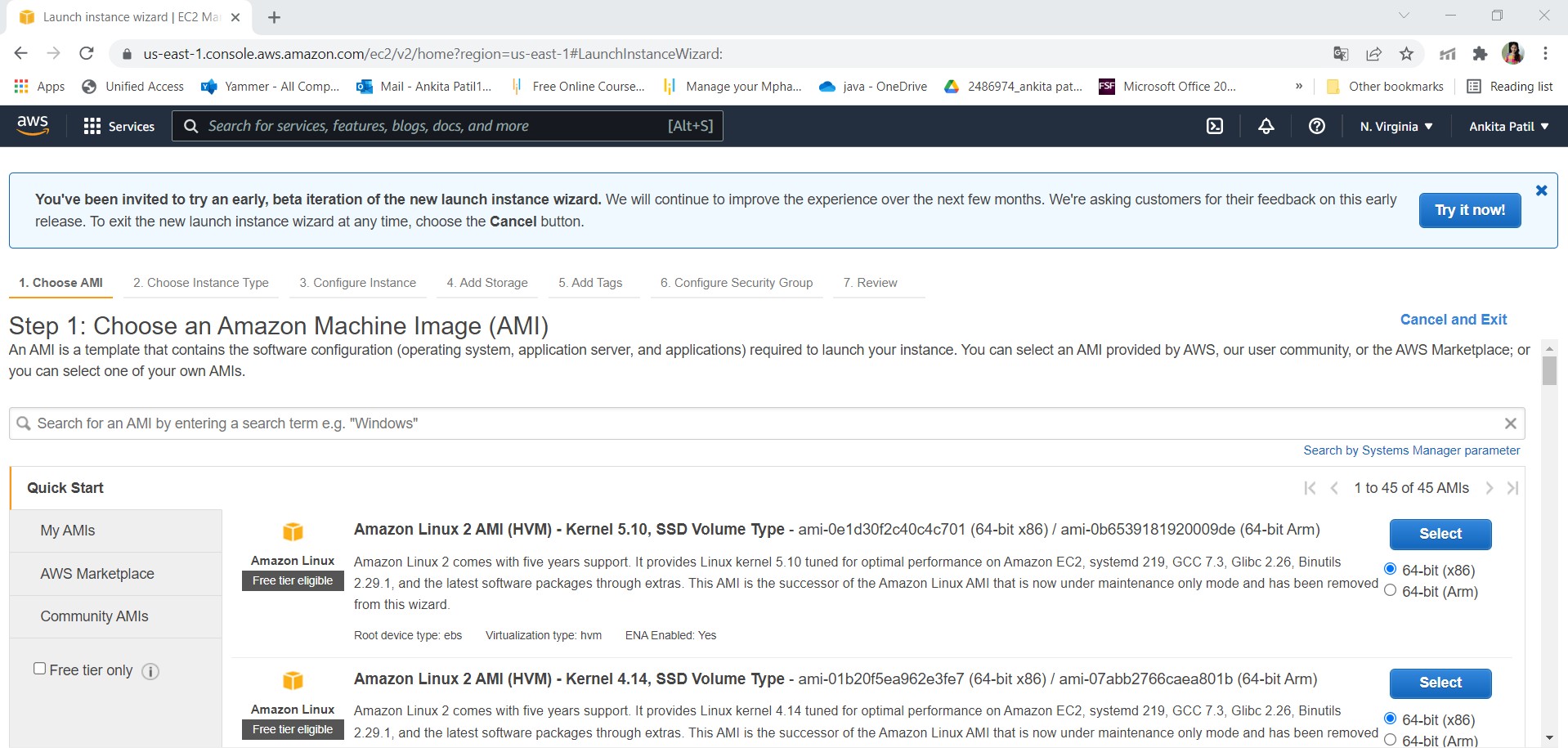
**Flow chart:**

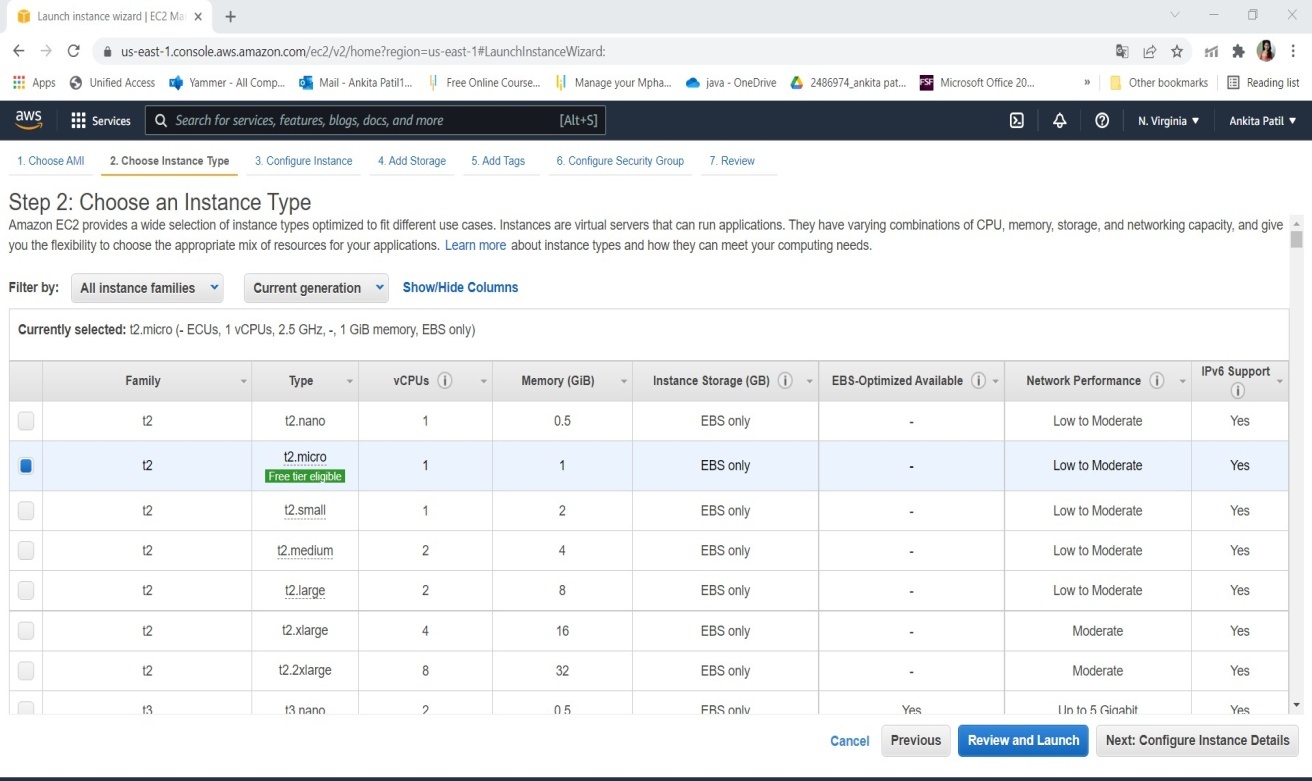


**Step 1: Write spring boot program:**

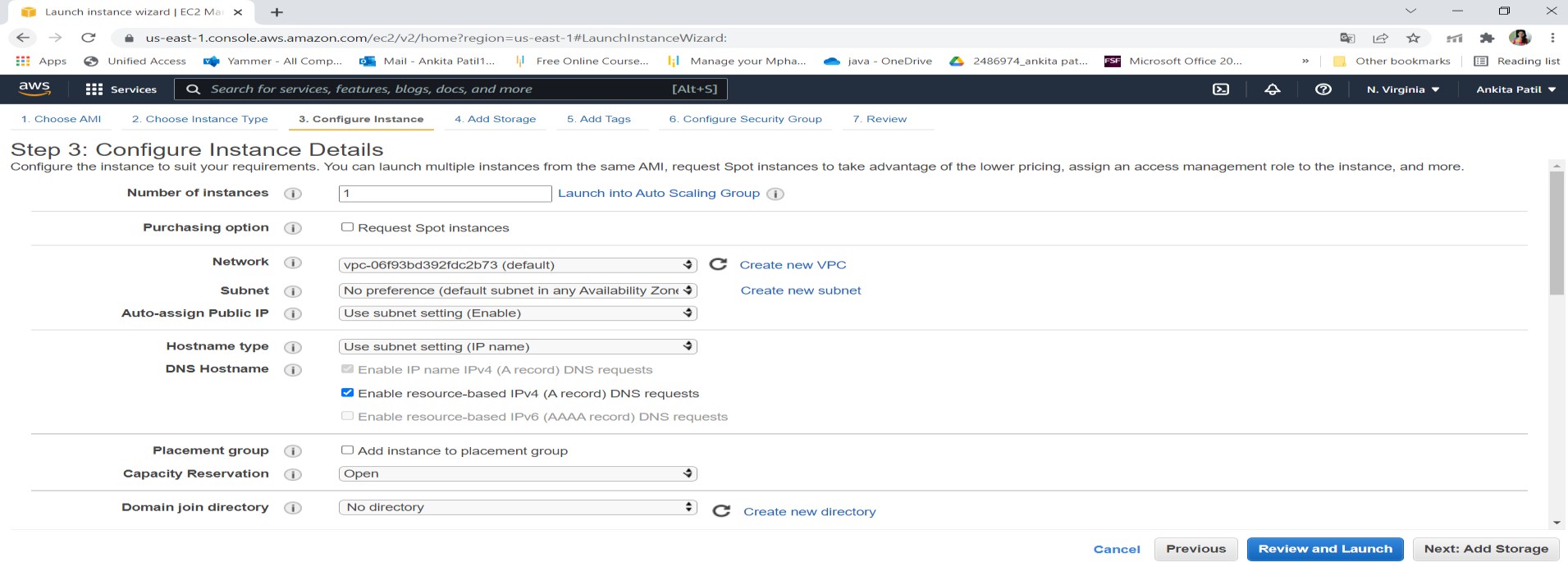


* + **Step 2: Launching EC2 instance:**

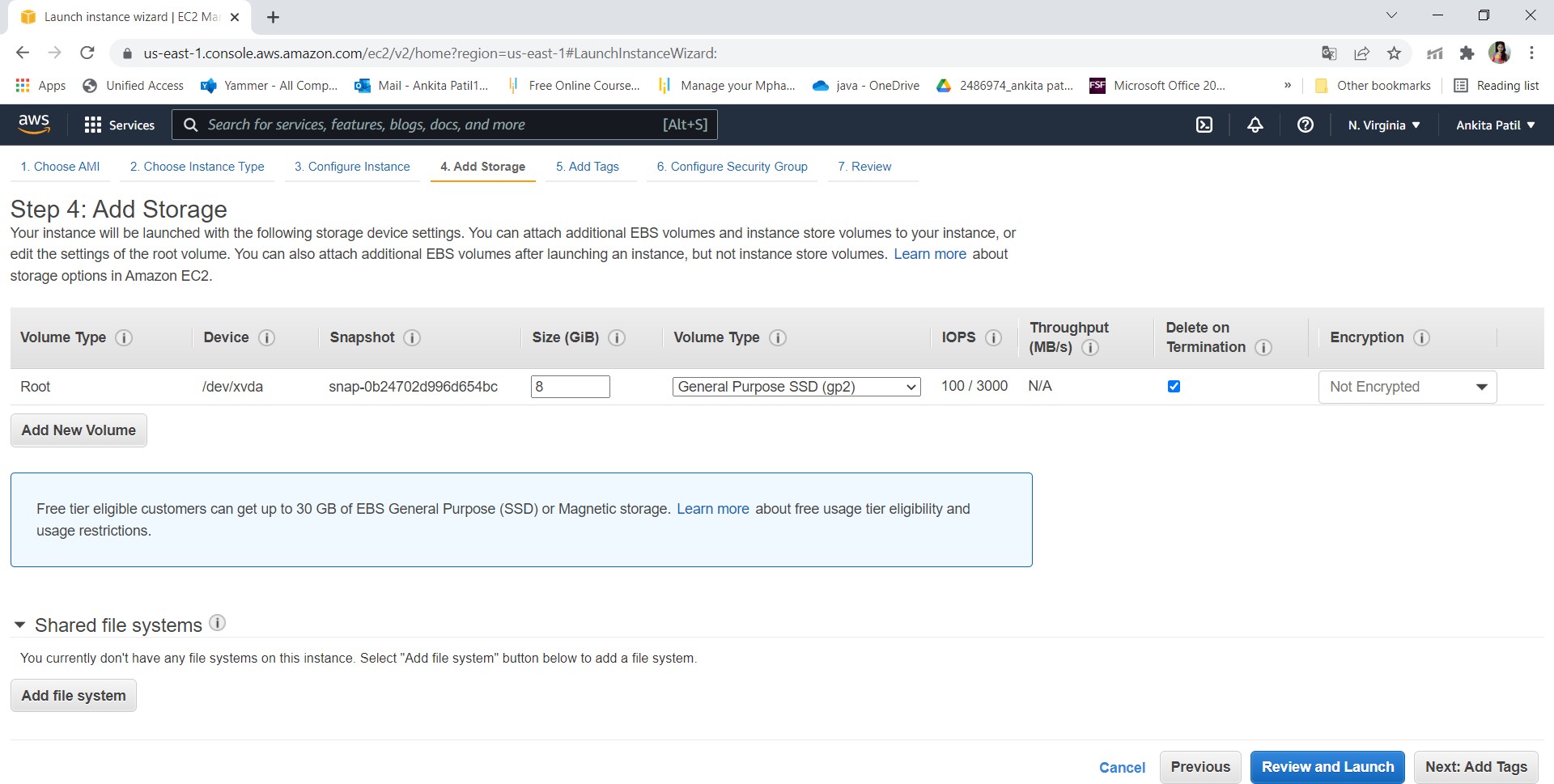
1. Click on launch instance to run any instance
2. Select the AMI
3. Select t2.micro as the instance type



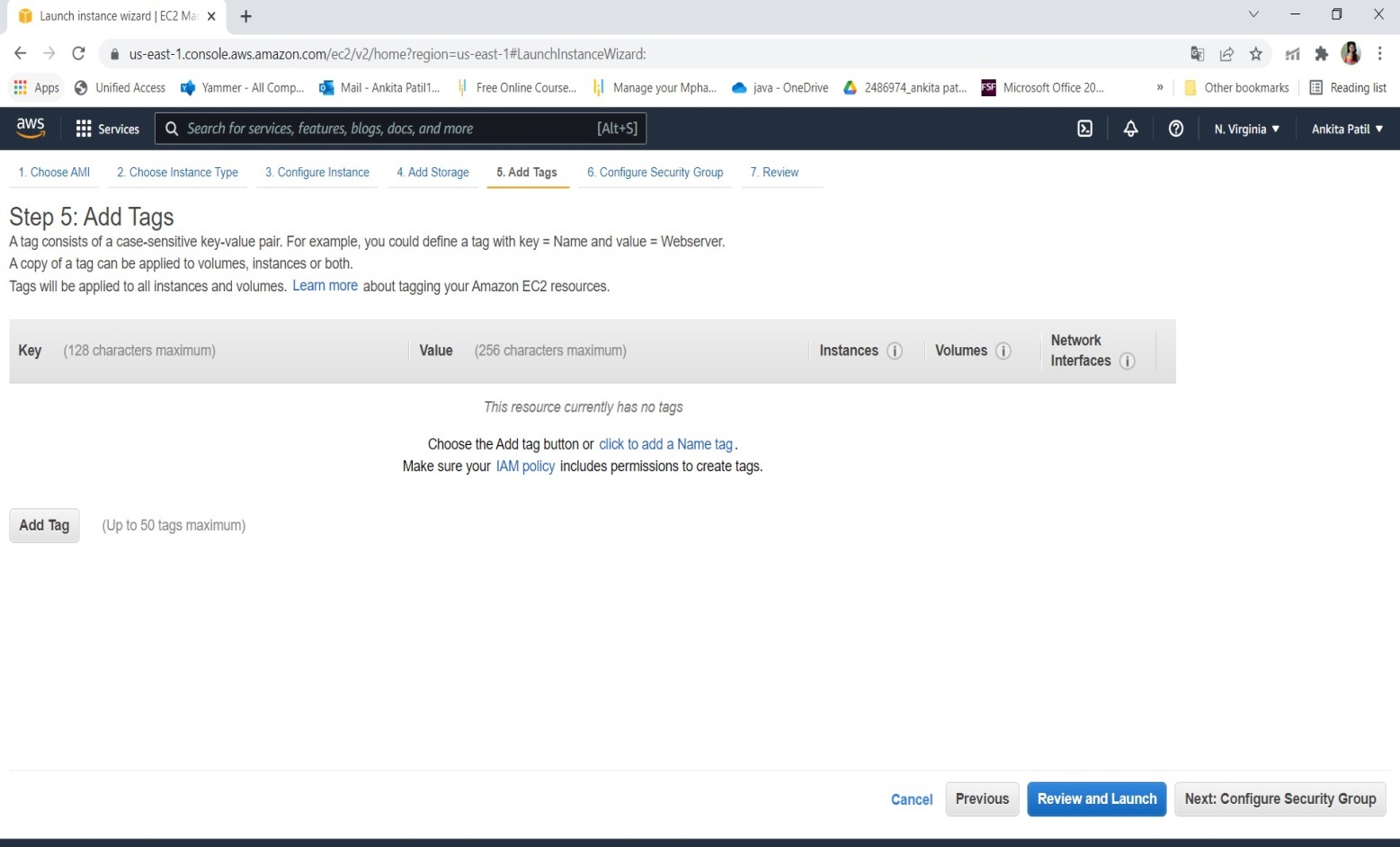
1. Specify the number of instances, networks, placement groups, and IAM roles and click Next

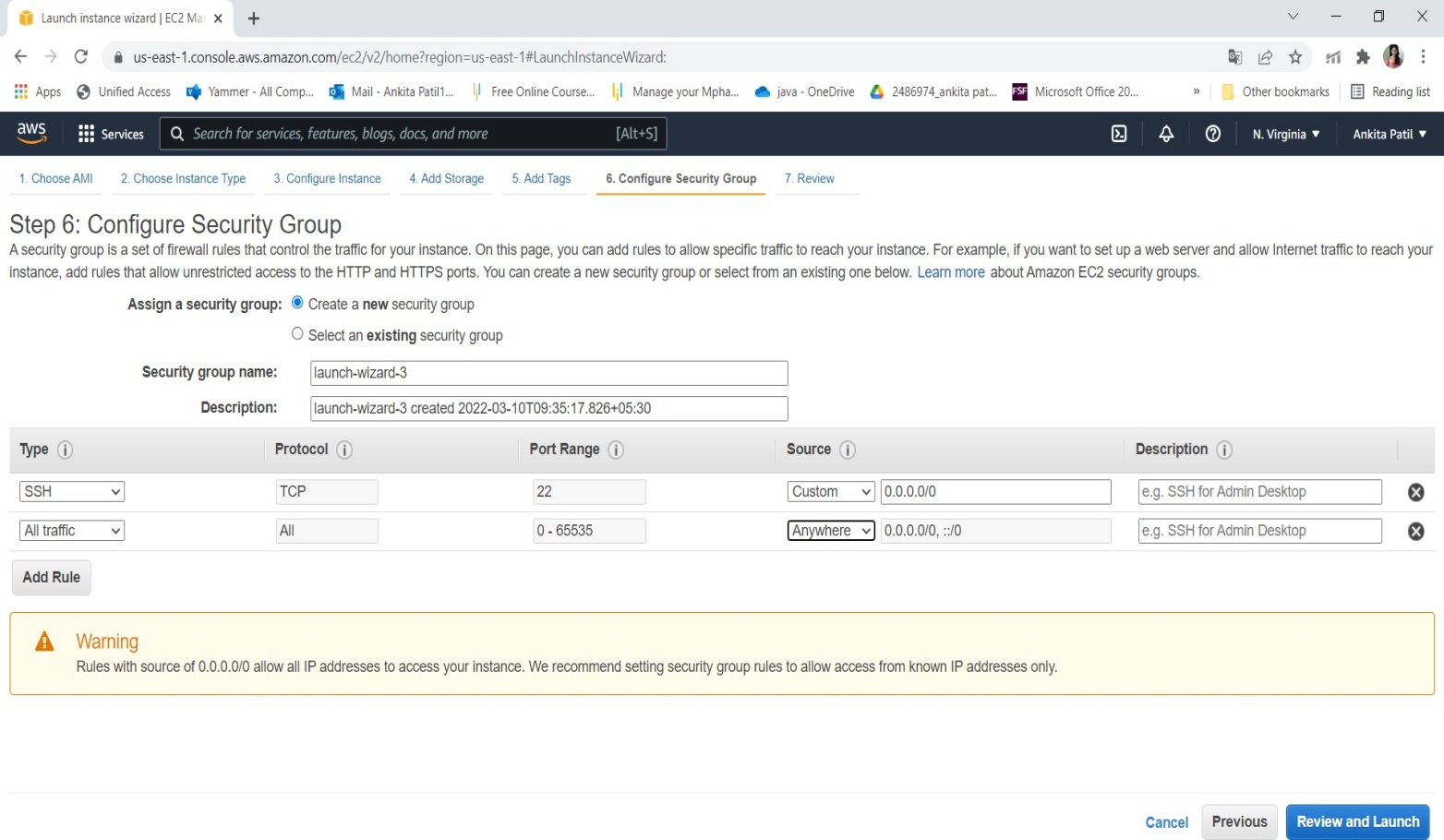


1. Add storage

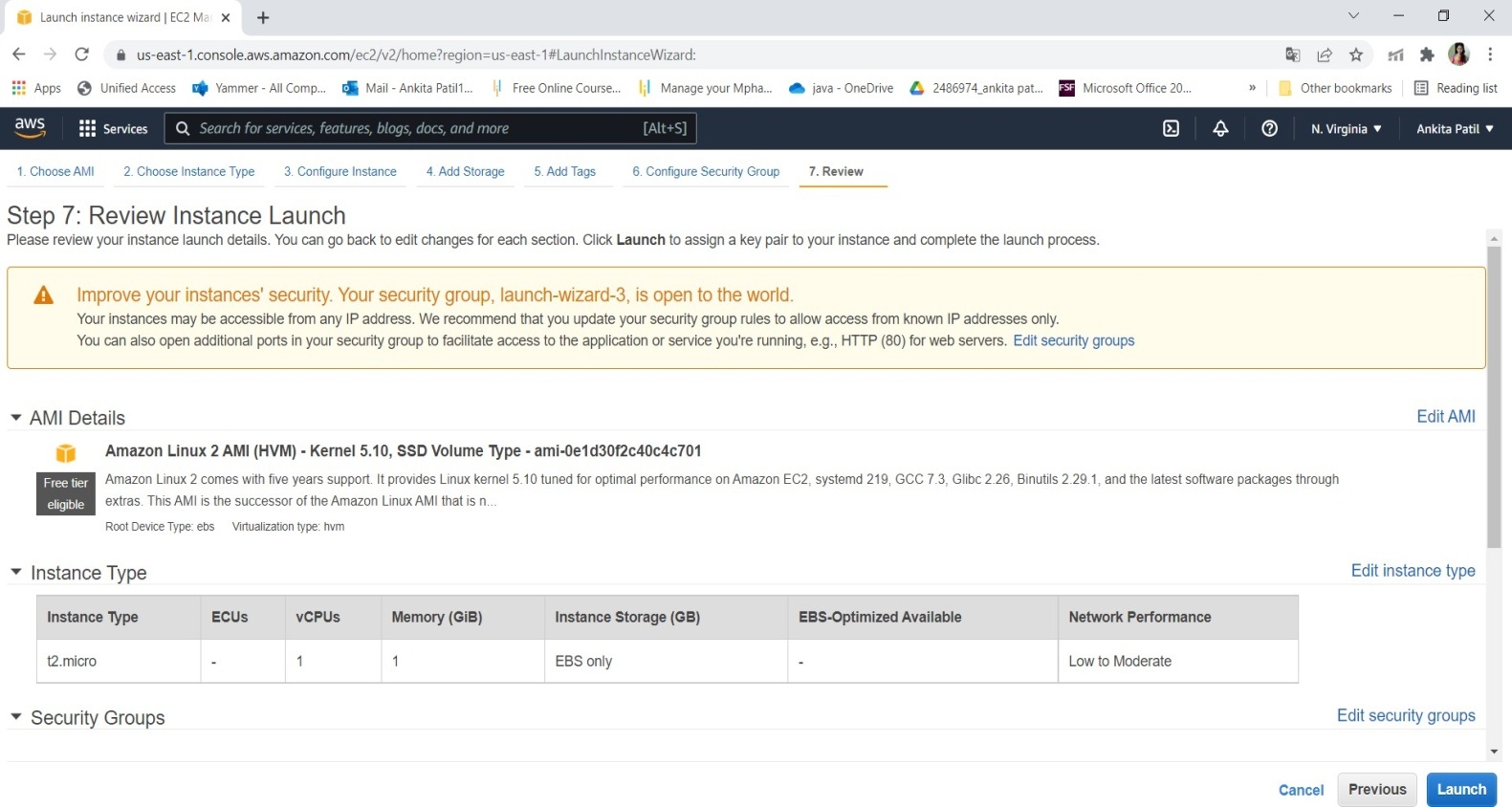


1. You can add a key-value pair to the instance

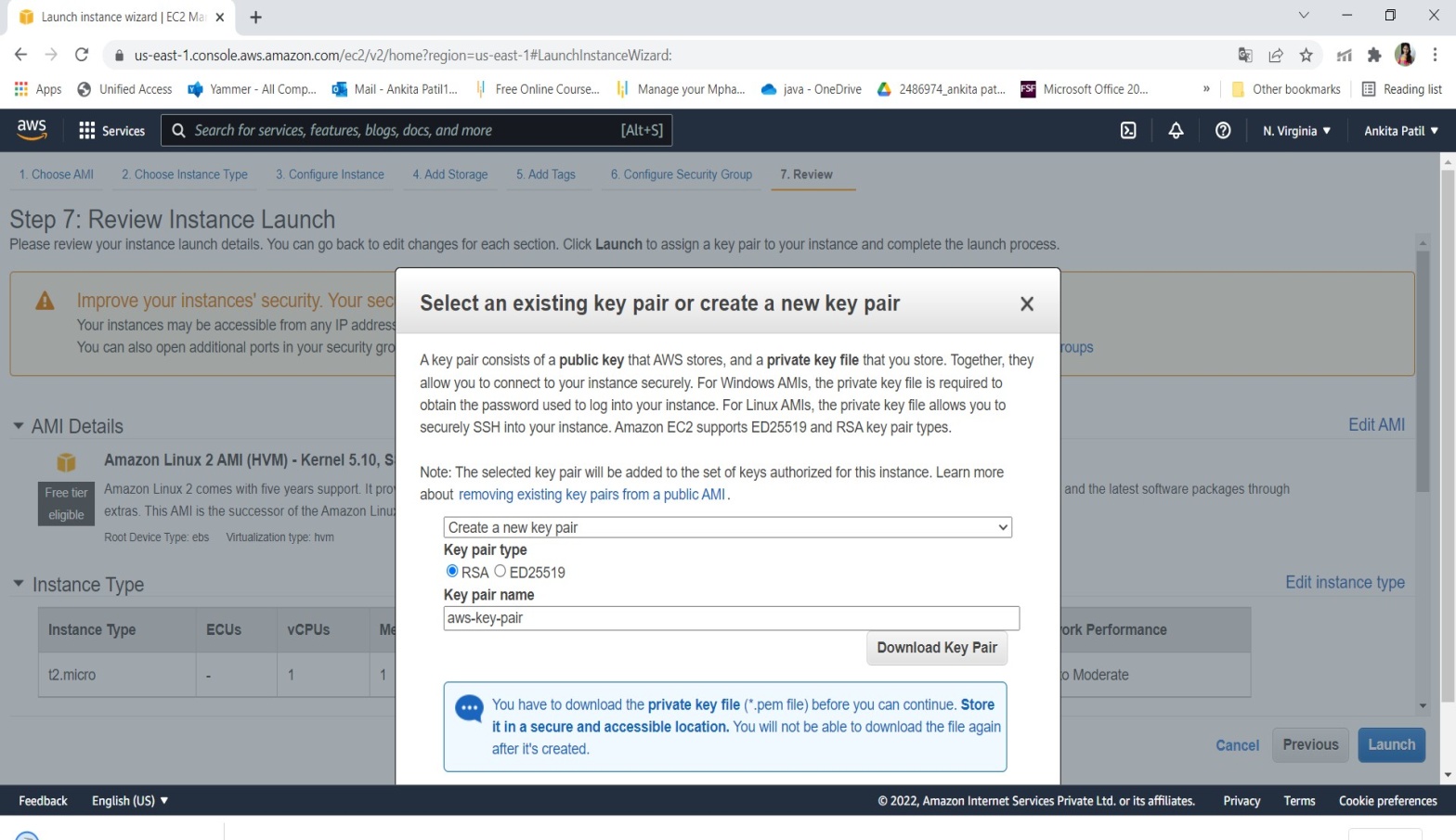




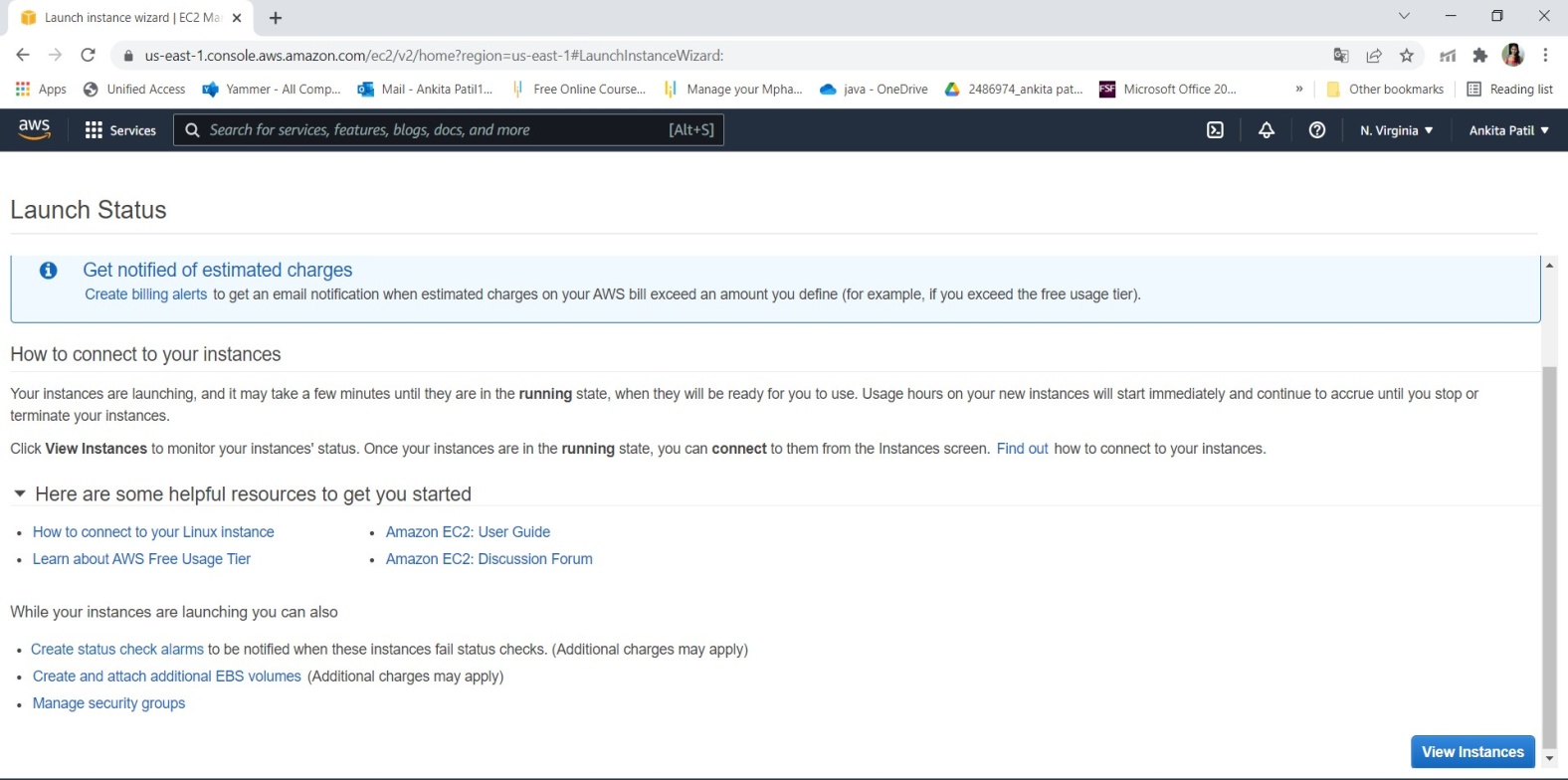
1. Click on launch



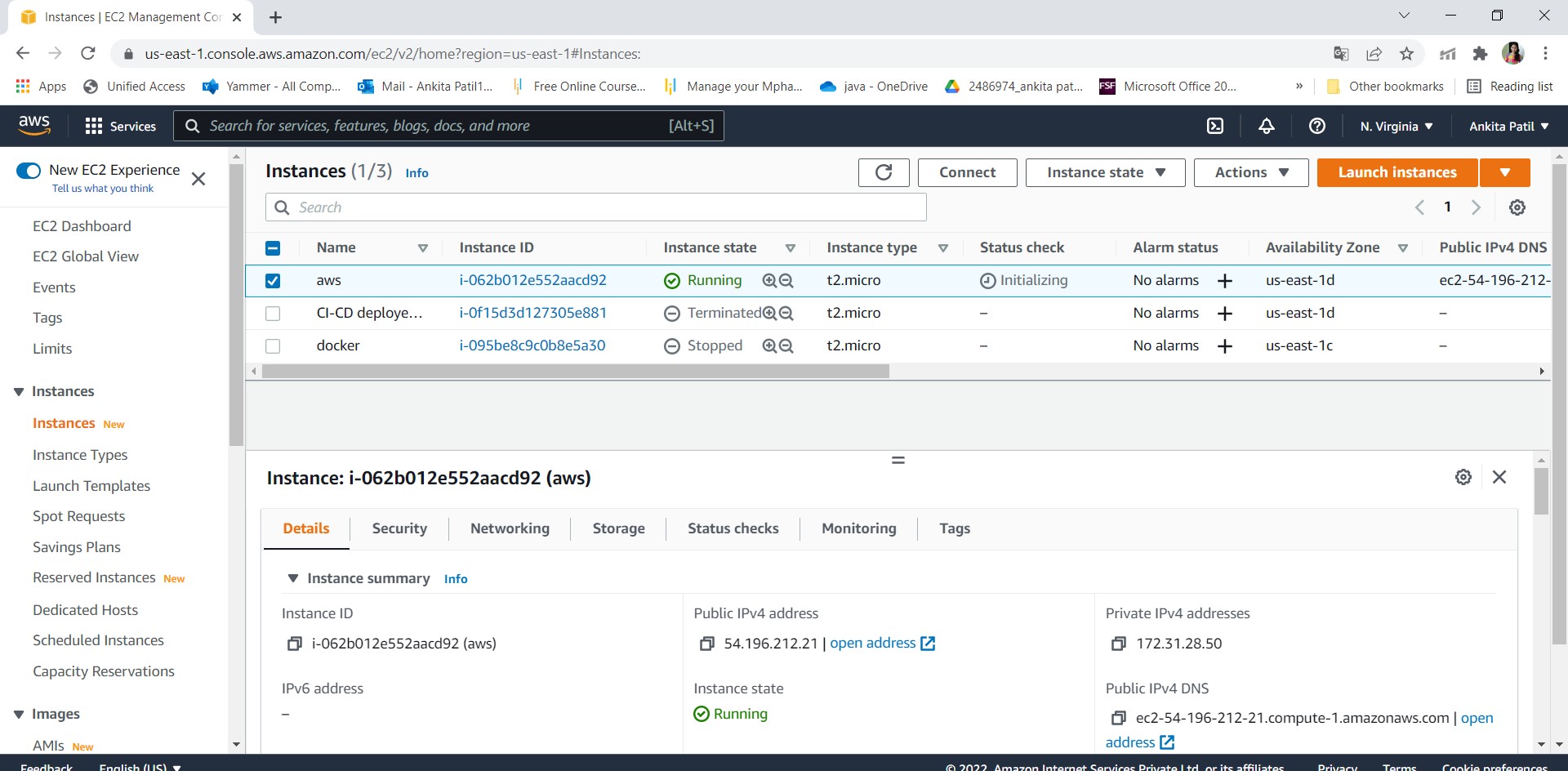
1. Create key pair

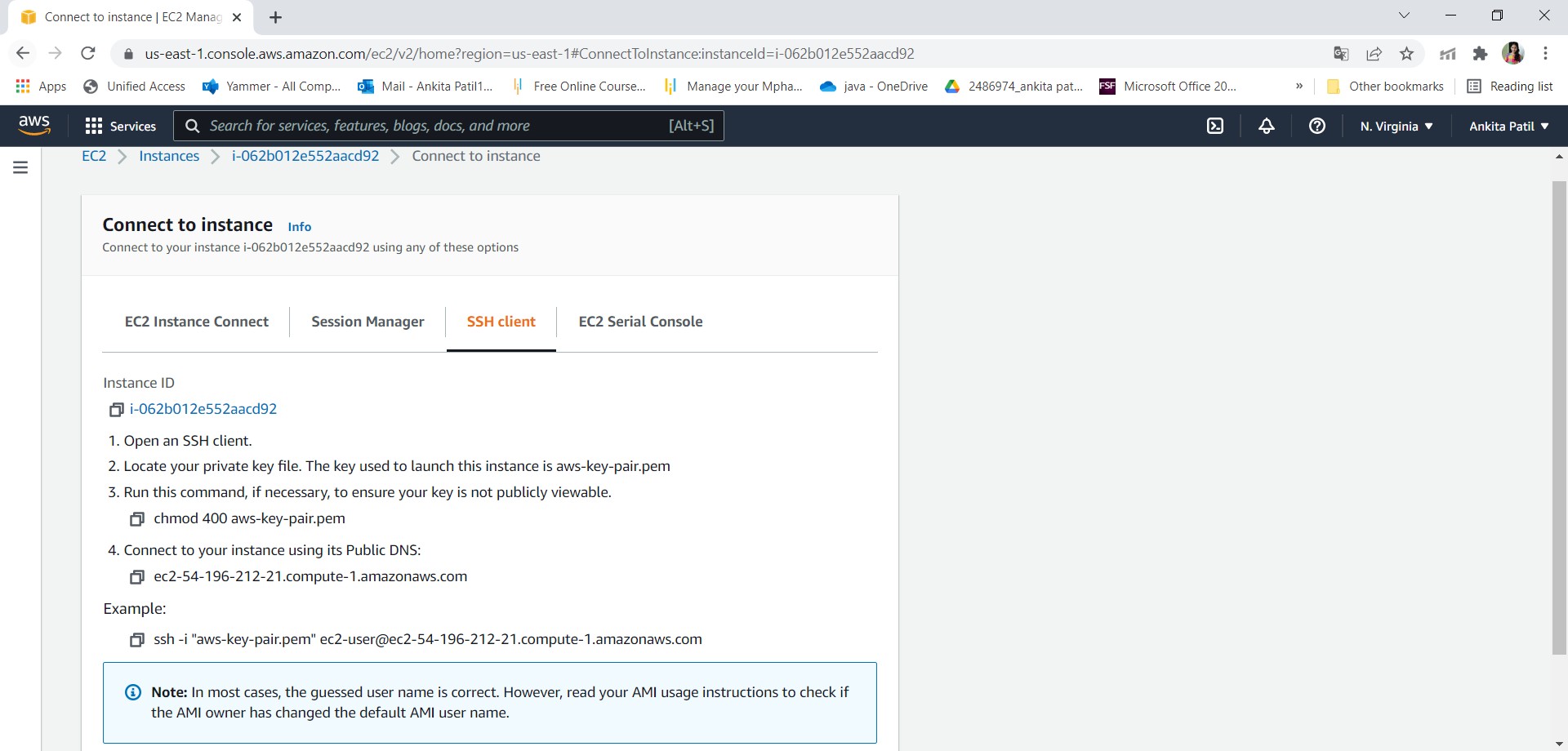


1. Click on view instance

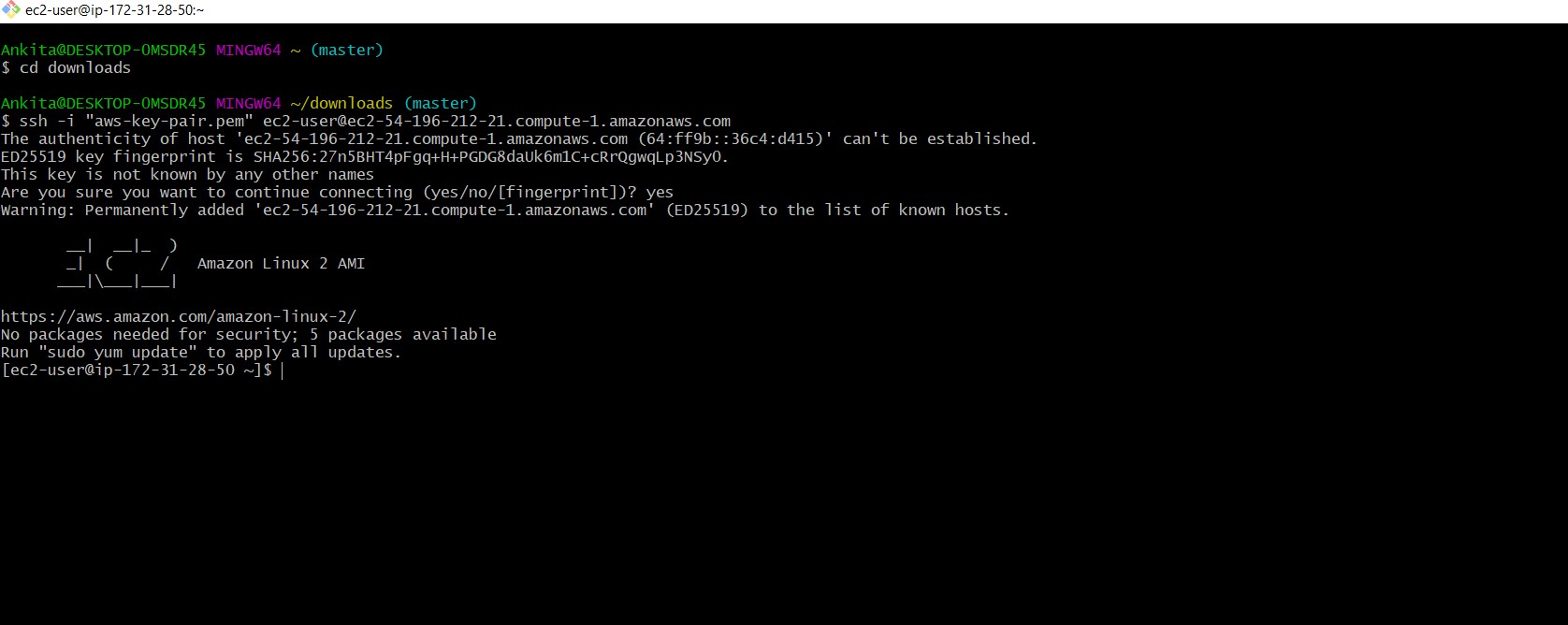


1. Connect the instance



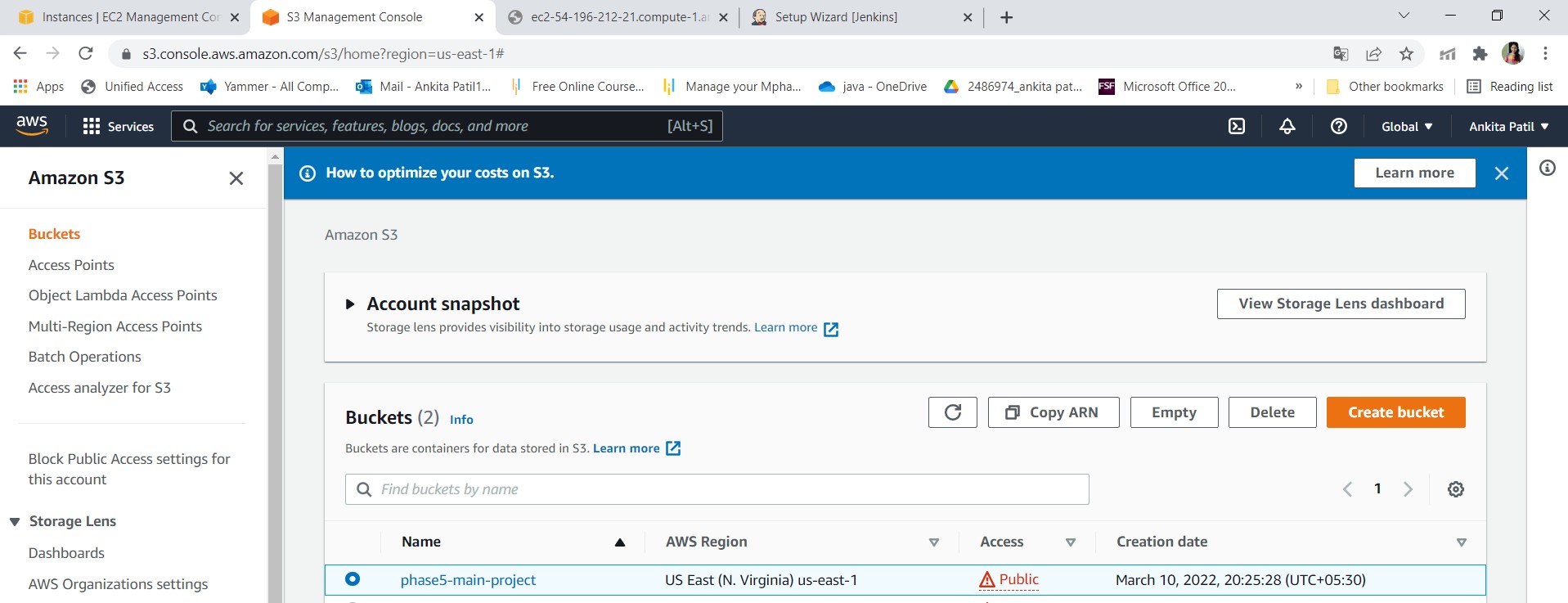
* + **Step 3: Connect to EC2 instances:**

1. Click on Connect on EC2 dashboard & Run the ssh command provided

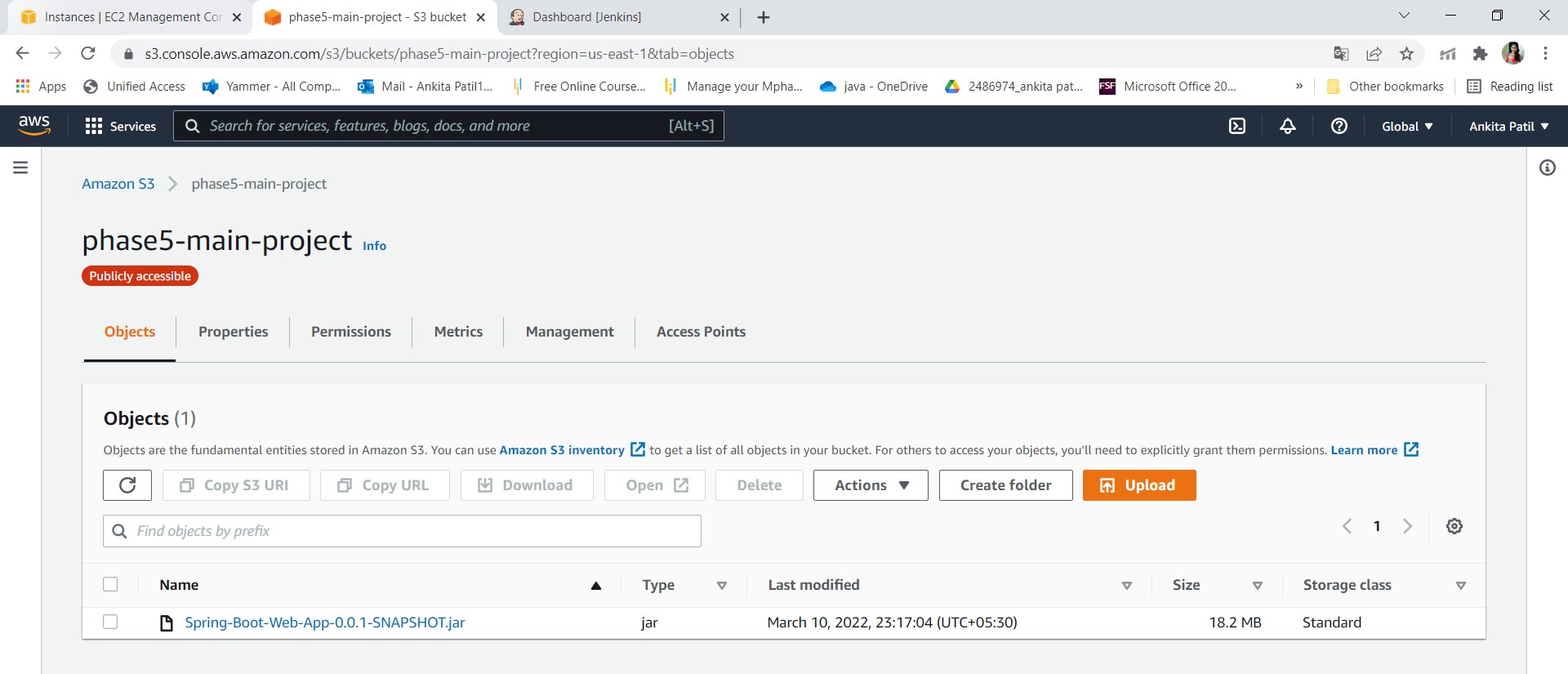


**Step 4: Creating S3 Bucket:**

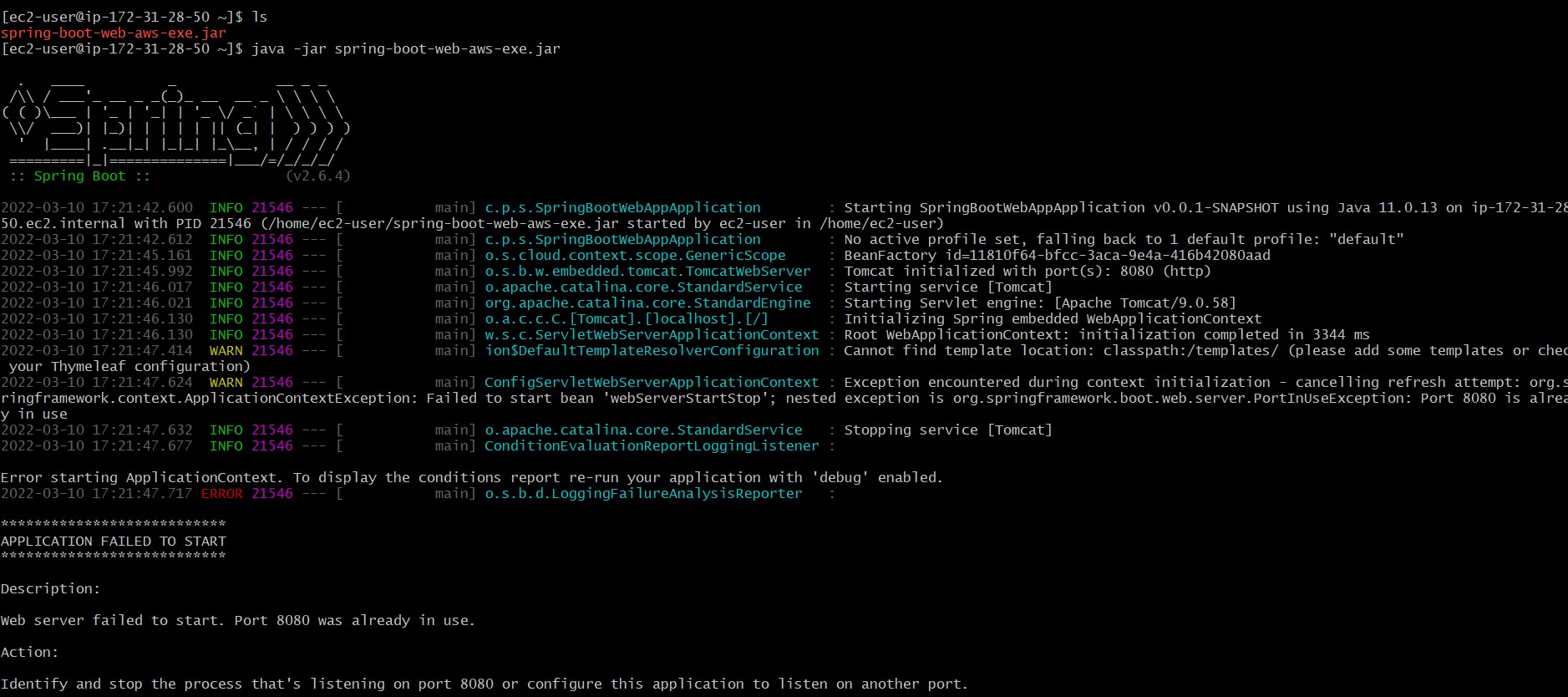
* 1. Create Bucket to store jar file:



Step 5:Adding jar file:

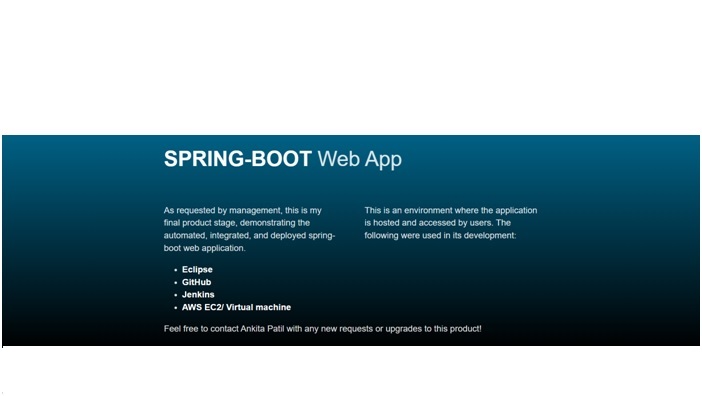


* + Run program through instance:



* + Output:

Screen shot :



**Step 6:** Pushing the code to GitHub repository

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

*cd <folder path>*

* Initialize 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 <commit message>*

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

*git push -u origin master*