22 July – 2024

Day 8

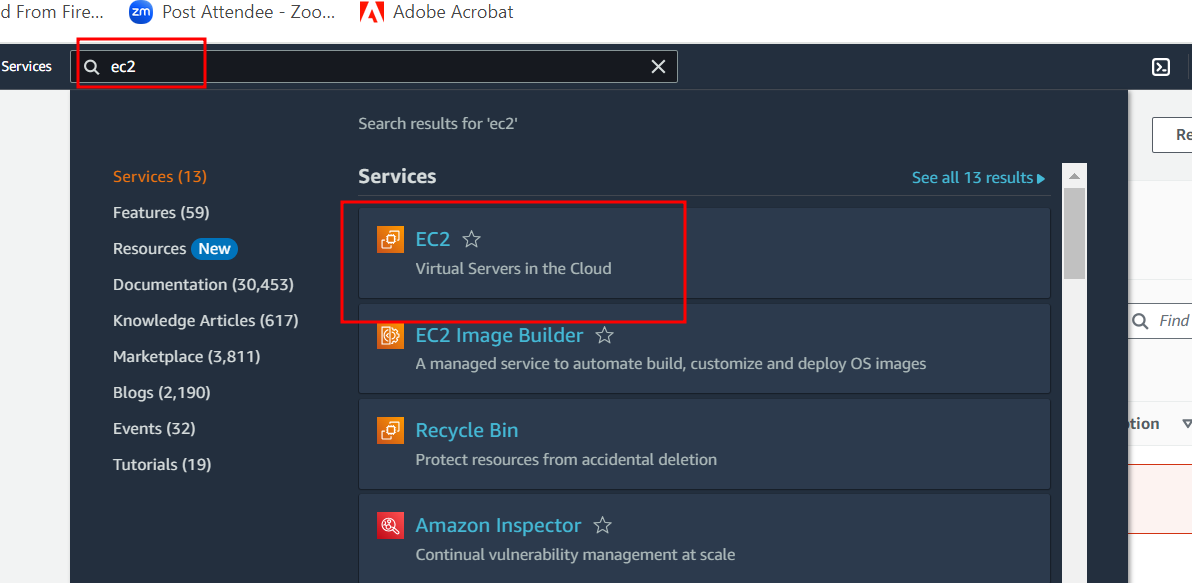
Description  
  
To build an infrastructure to host software on an AWS EC2 instance for a clinic to have online data access for pets, their owners, and consultation fees  
  
   
  
Prerequisites:  
  
Spring Boot  
Jenkins  
Docker  
AWS  
   
  
Problem Statement: This assignment is designed to help you understand how to plan and develop the back end of a given problem, gain hands-on experience building the CI/CD Pipeline using Jenkins, and containerize the application on the AWS cloud platform.

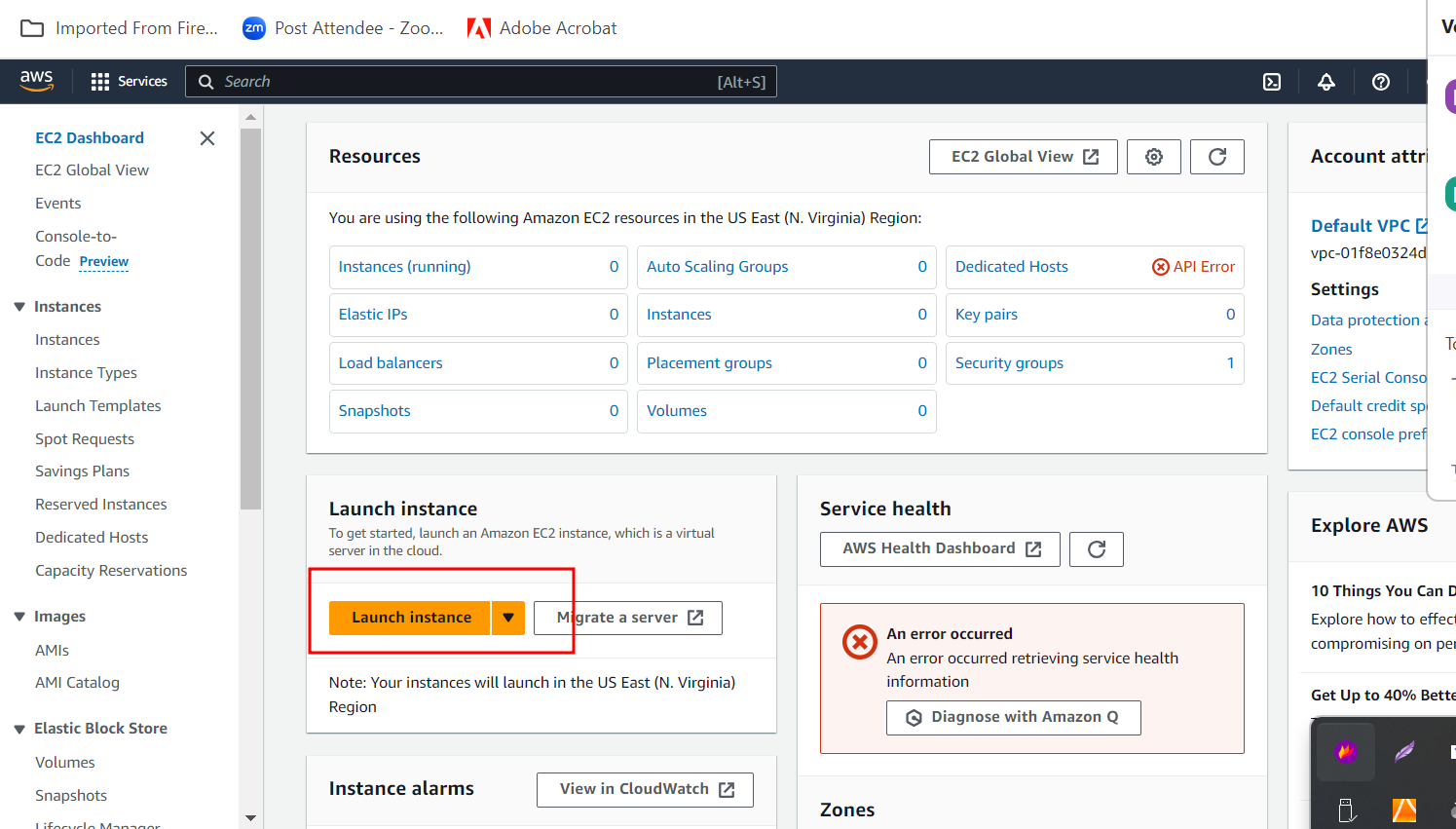
Task (Activities)  
1. Import the Spring Boot project with the generated code in Eclipse  
2. Configure the project with Dockerfile and Jenkinsfile  
3. Build the project using the Maven package  
4. Create and launch AWS EC2 instance  
5. Configure EC2 instance with JDK 11, Docker, and Jenkins  
6. Sync the given code to the Git repository  
7. Create Jenkins pipeline on EC2 with Git and GitHub  
8. Build the pipeline to dockerize the application

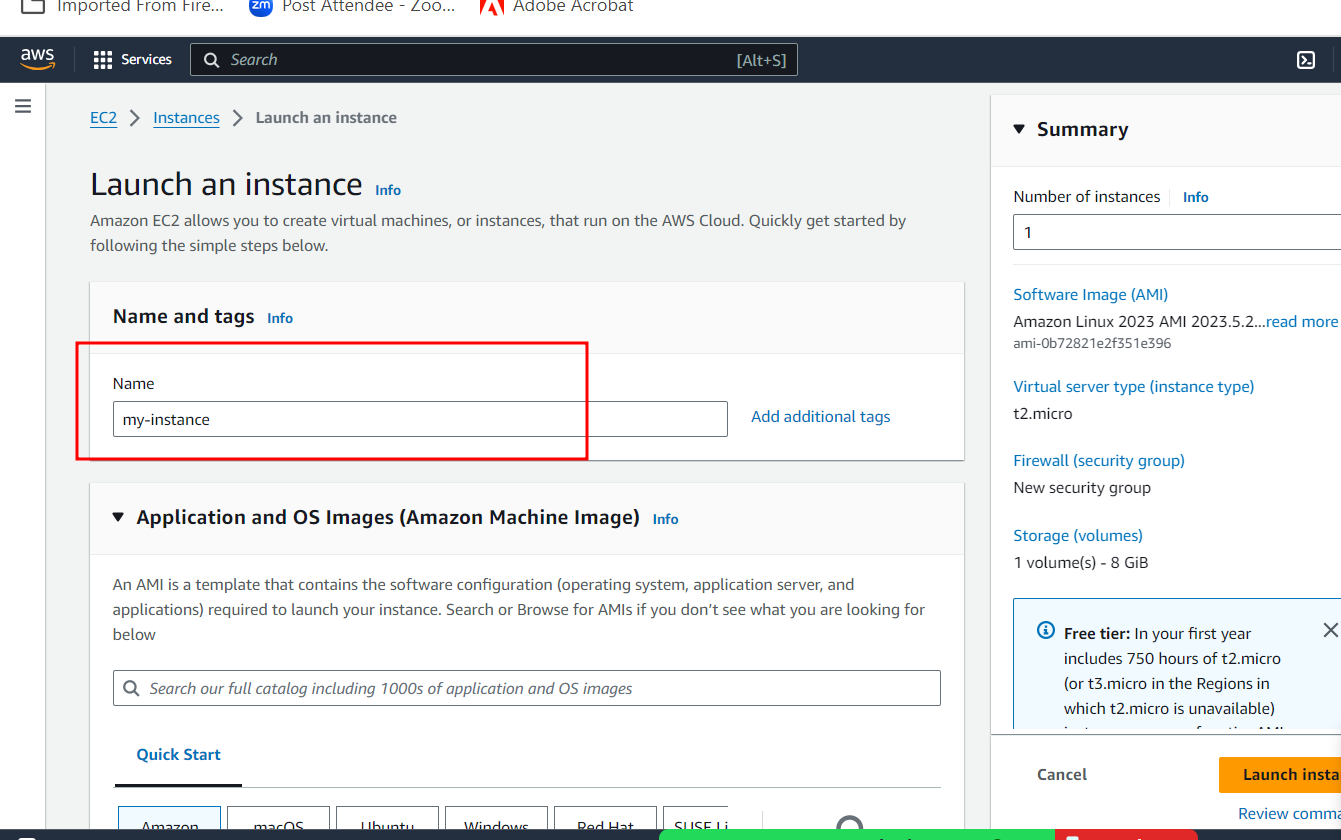
Steps for Problem Statements

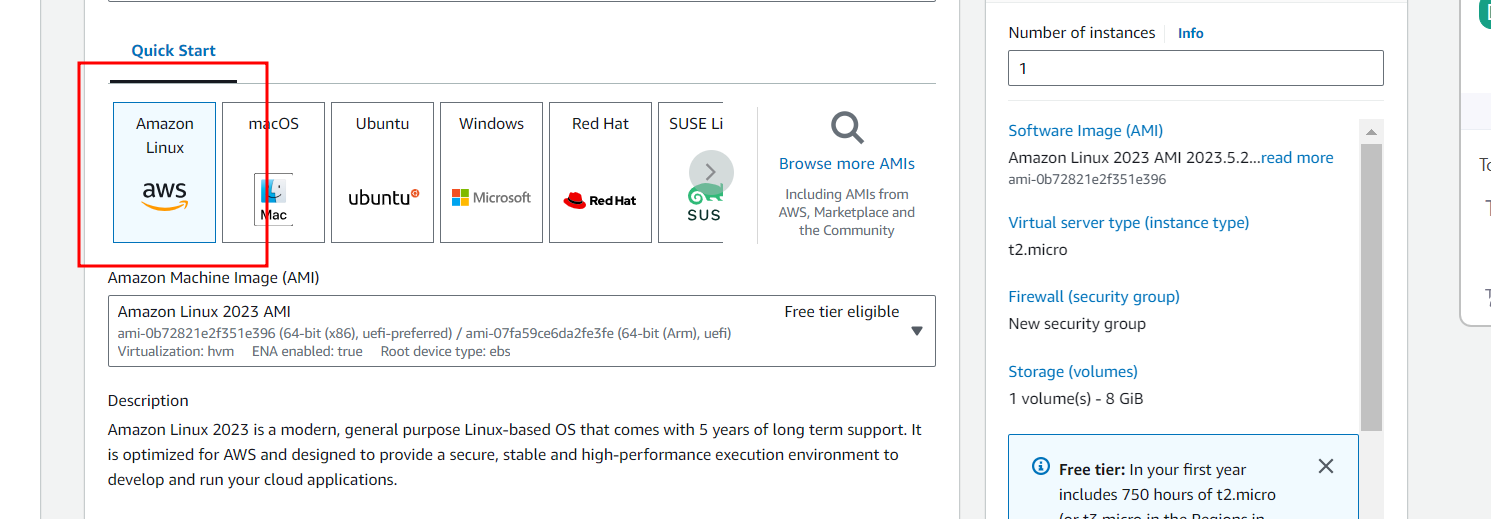
1. Creating spring boot project in local machine with starter as web starter and thymeleaf starter.
2. In index.html page write application details with message as well as image.
3. Please run this project with different port number 9090.
4. Then test the project in local machine working or not.
5. Then create the jar file using maven with eclipse or using mvn command.
6. Please create Dockerfile which is responsible to run spring boot project using docker.
7. In local machine please create docker images and run this image and verify container running or not in local machine or VM machine.
8. Now we need to create Jenkinsfile this Jenkins file responsible to run DockerImage
9. After creation please make this folder as local repository and push this code to remote repository. Remote repository contains spring boot project with Dockerfile and Jenkinsfile.
10. Now we need to logic for AWS account and create EC2 instance in AWS account.
11. Then connect EC2 instance using browser or terminal.
12. Need to install git, Java, jenkin, docker etc.
13. Open the port number 8080(jenkin), your application port number(9090) etc.
14. Then using pubic ip address with port number <http://publicidaddress:8080>
15. Then we need to create Jenkin pipe line job which is responsible to pull the project from git which contains Jenkinsfile which is responsible to run Docker image on EC2 instance.
16. Then we can view our project using <http://publicidaddress:9090>

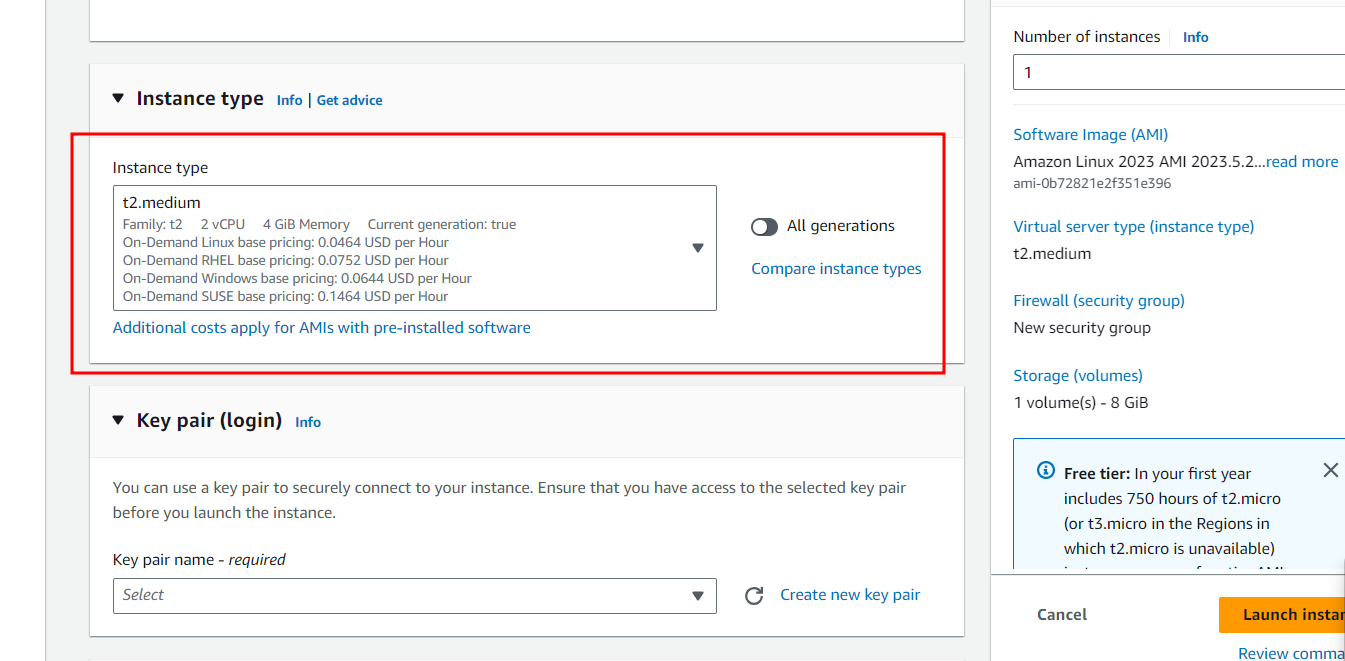
EC2 instance creation and installing required software.



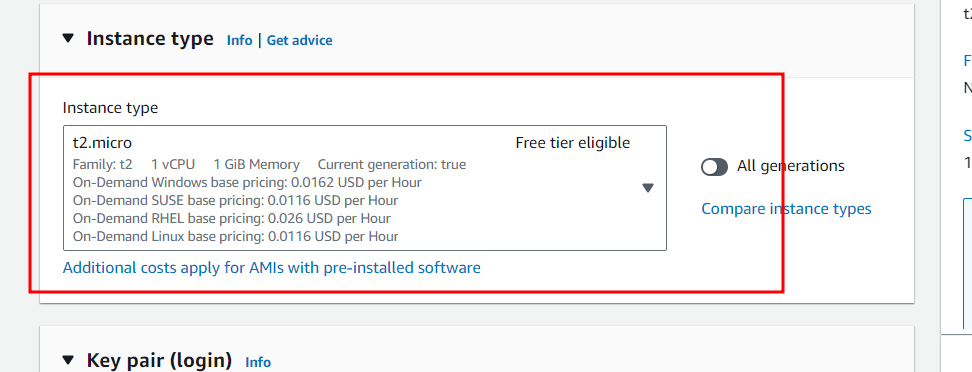


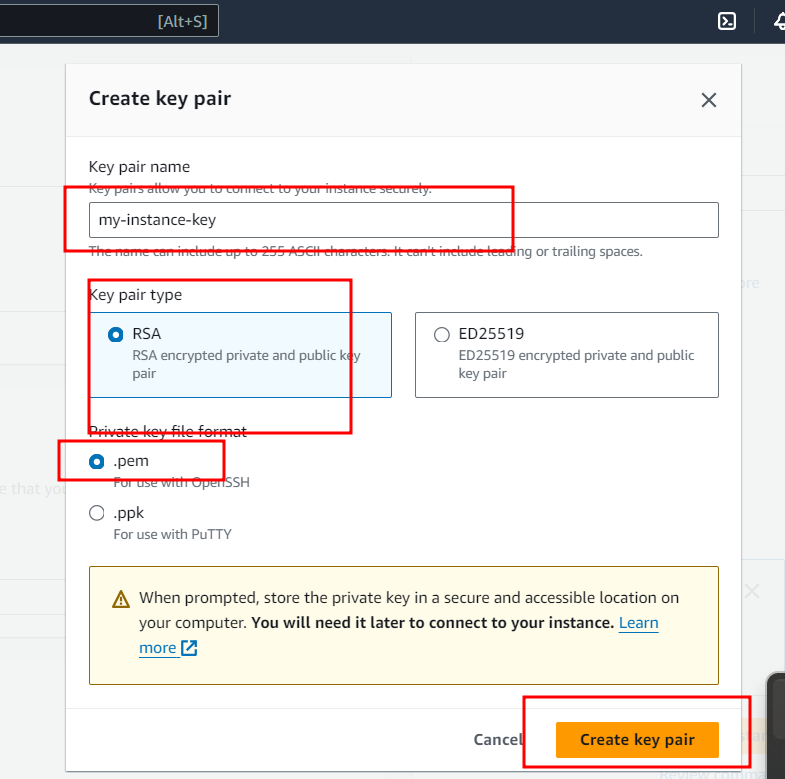


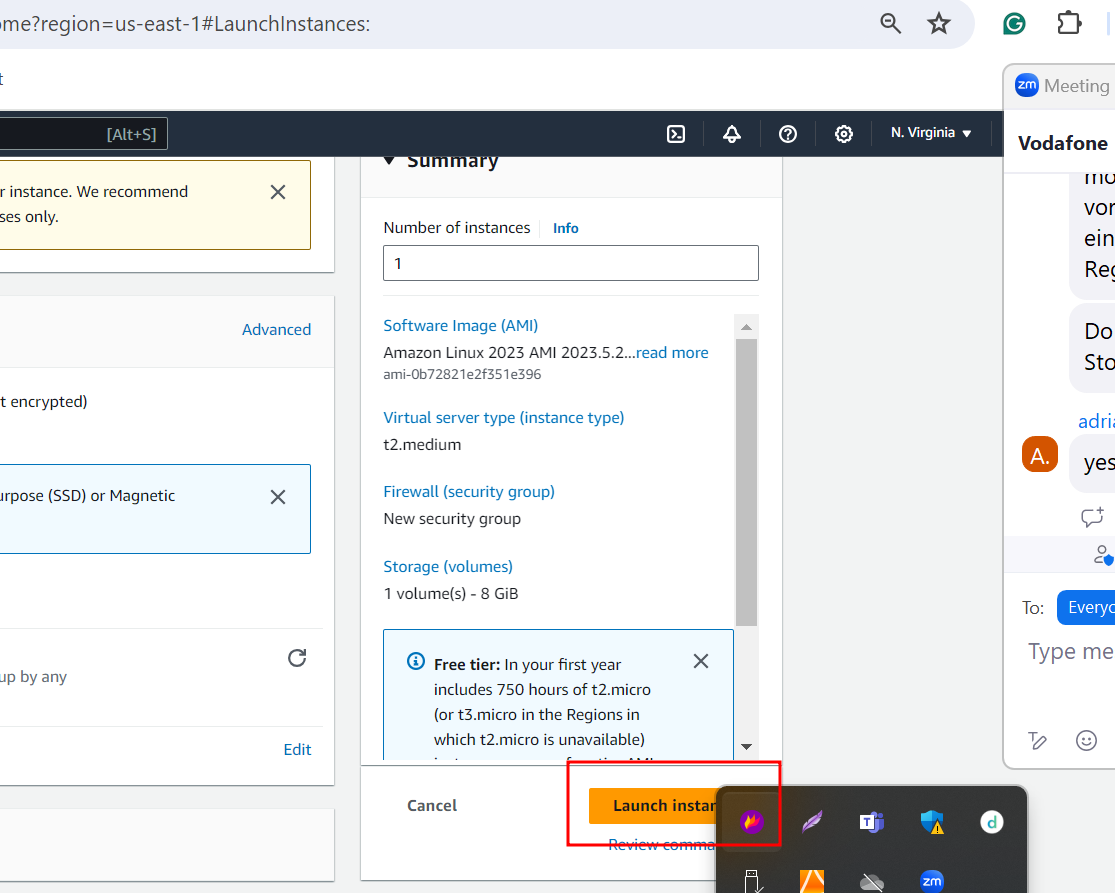




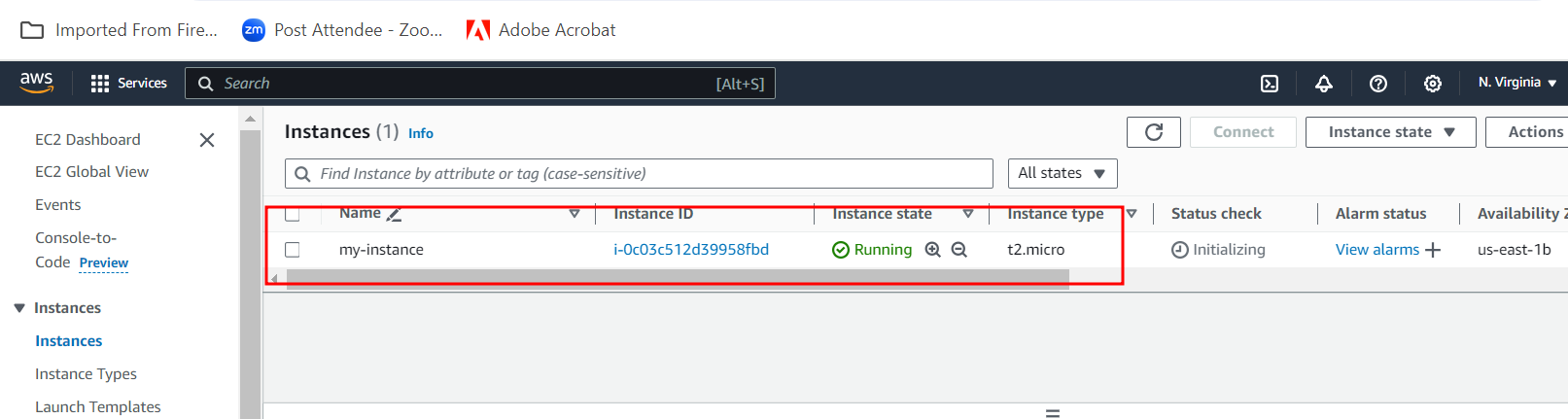
Or



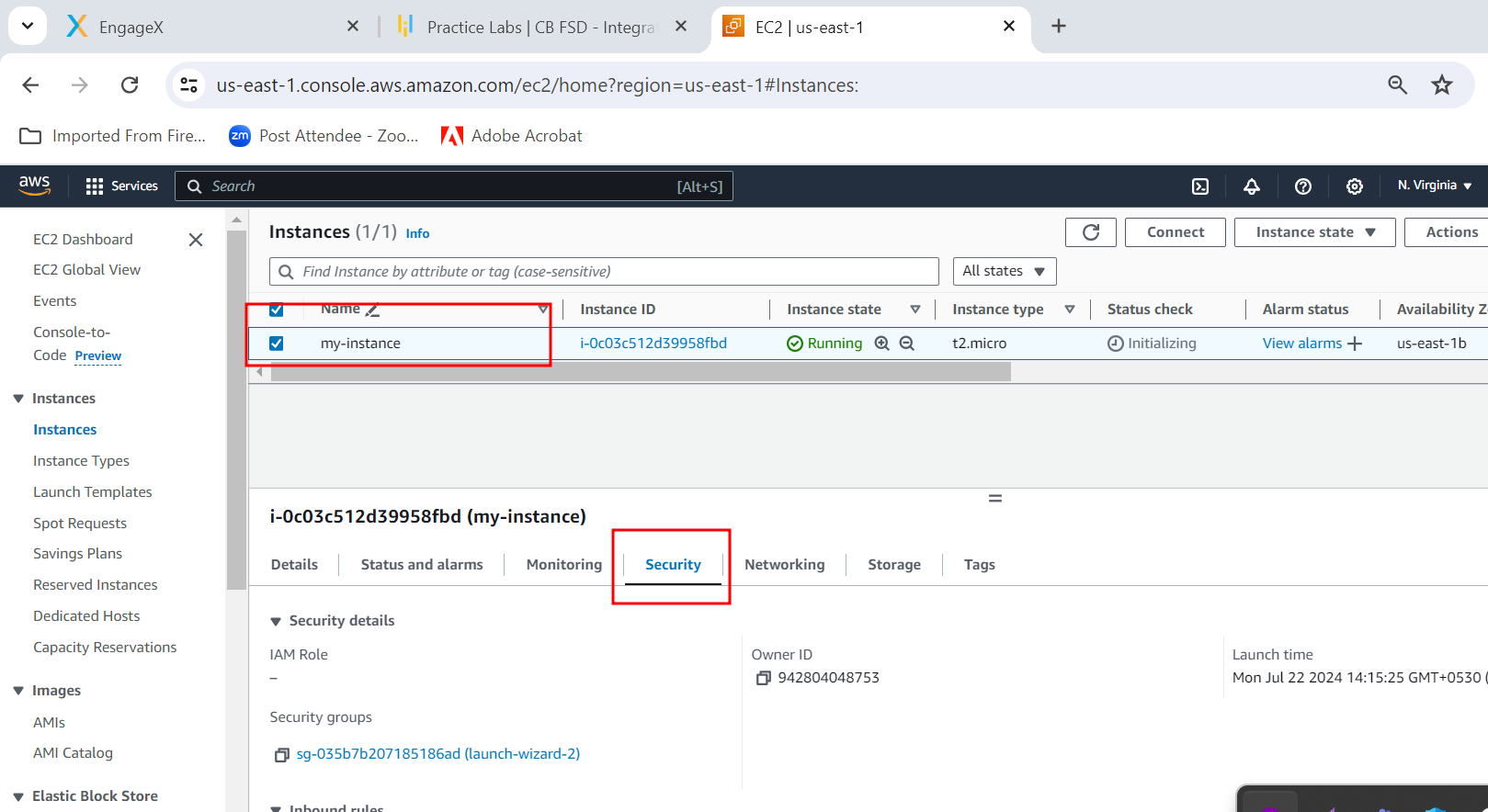


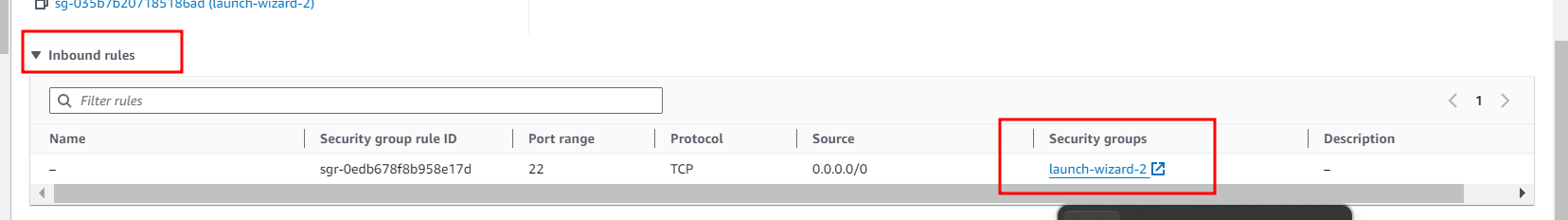


After created verify instance status running or not.

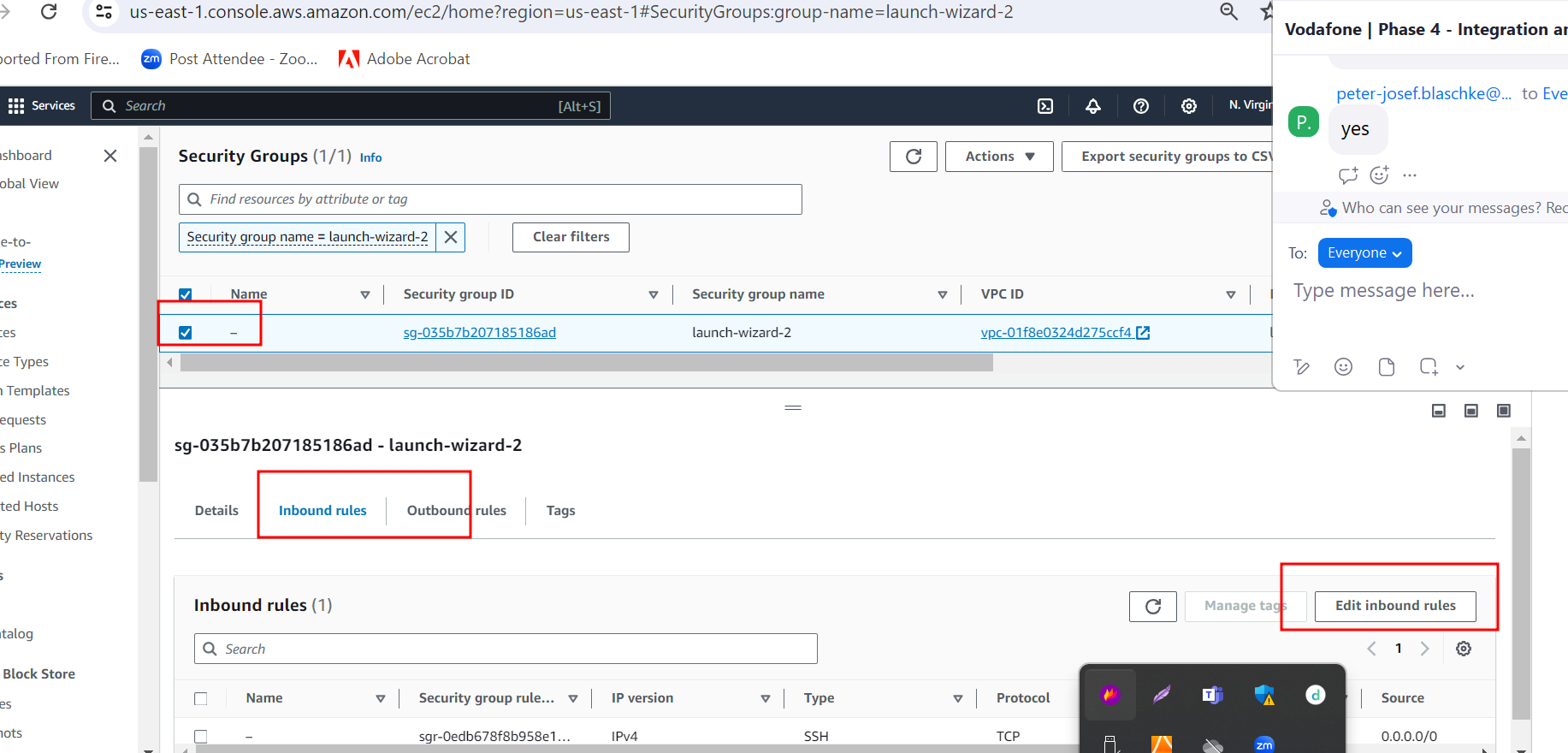


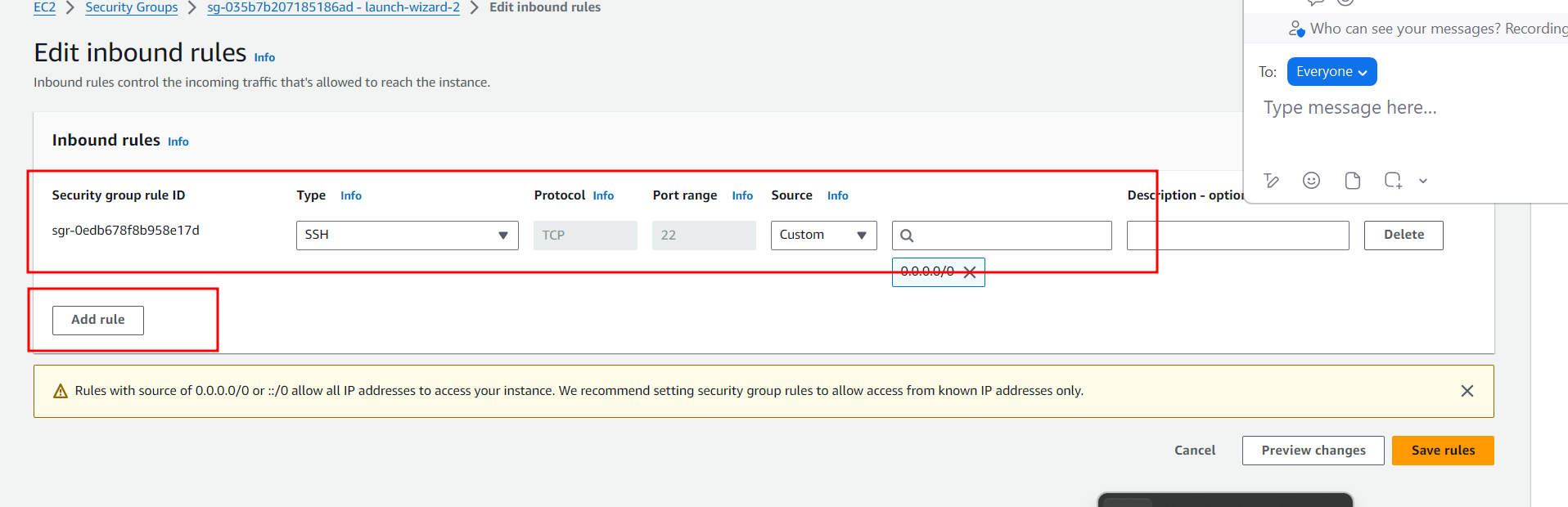
Open the port number depending upon the type of application your are running.

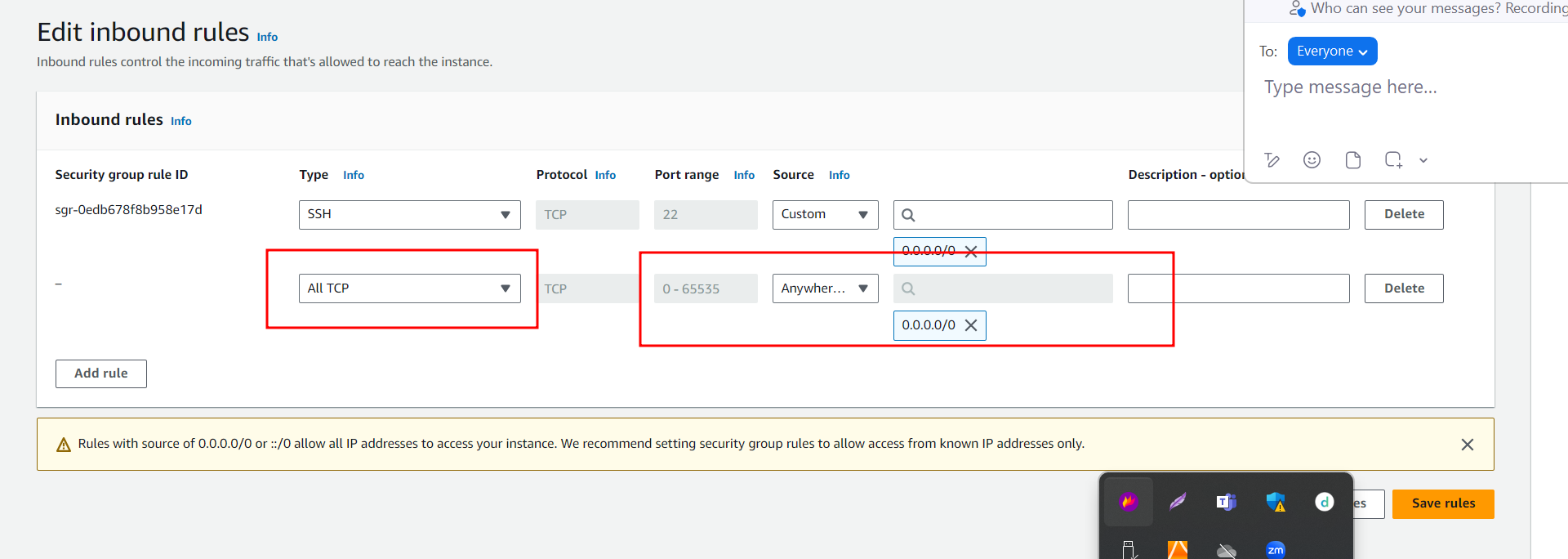




By default EC2 instance open default port number 22. Which help to connect EC2 instance SSH client.

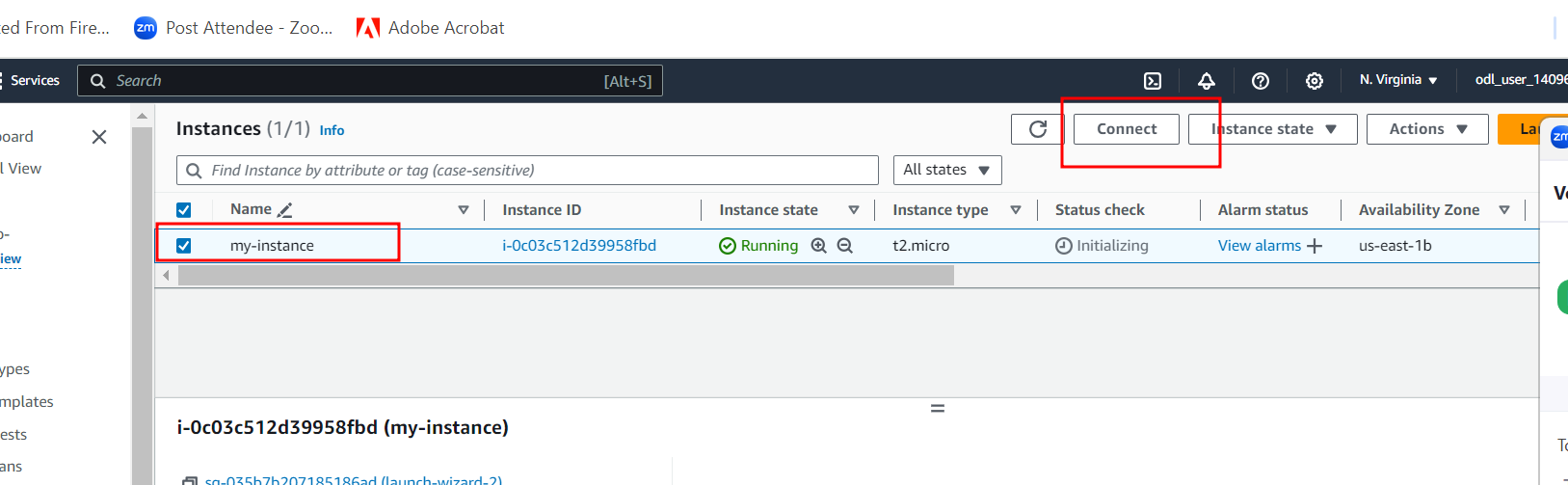


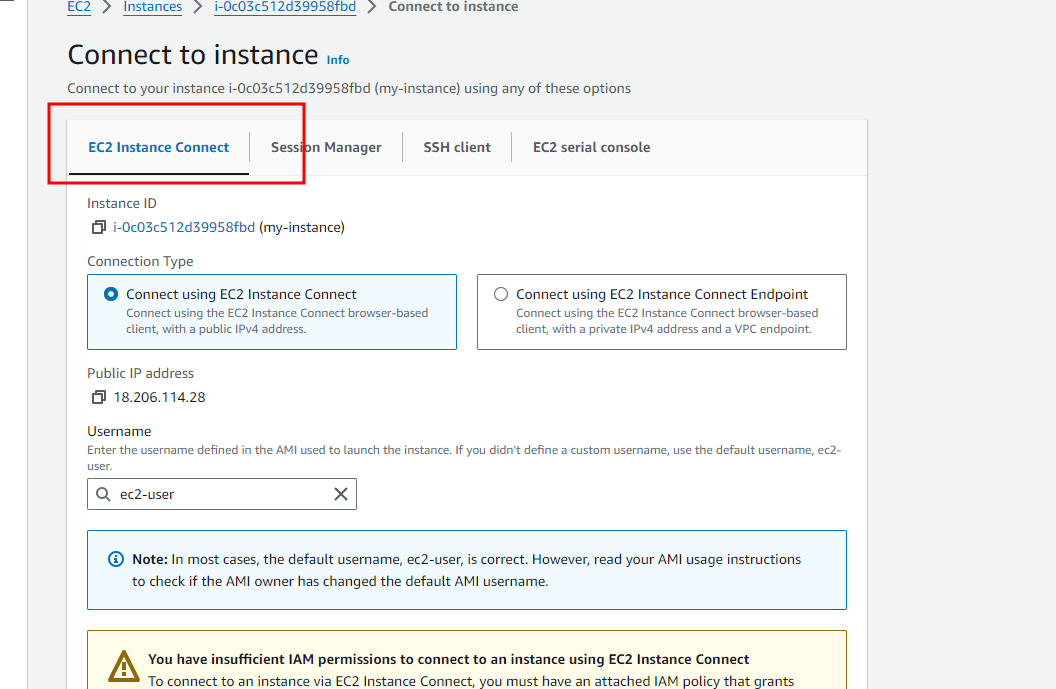


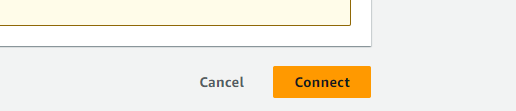


**Please save the rules.**

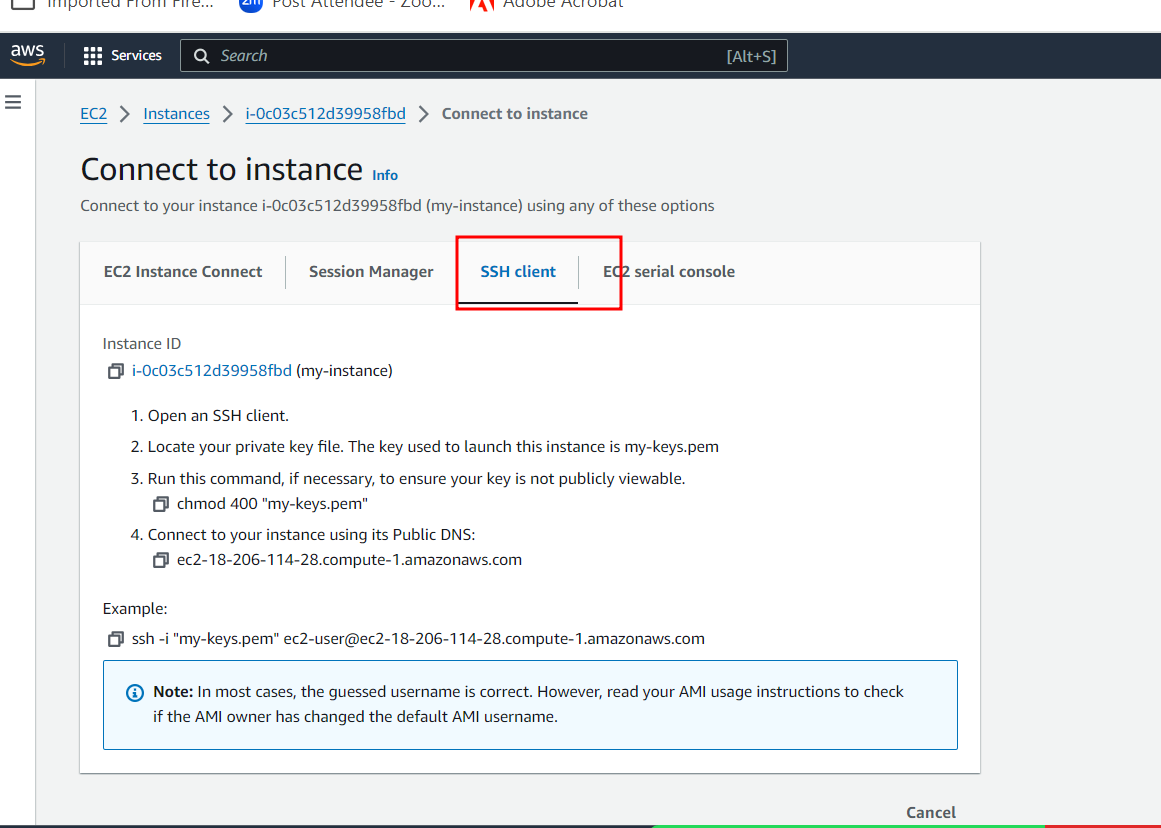
**Connecting EC2 instance using browser**



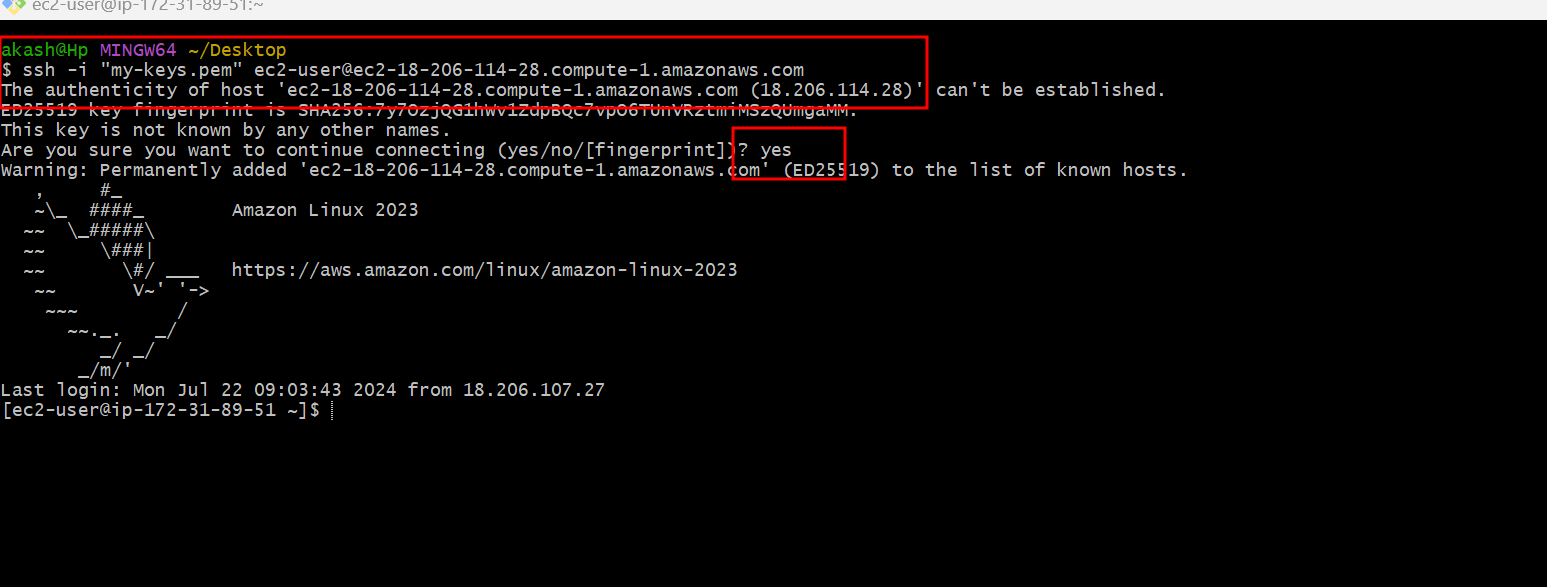




**Connection EC2 instance using Terminal (Non window user ) for window user (Gitbash terminal)**



Copy SSH command with option from Example then Open the terminal or gibash in the place where .pem file present and paste this command and hit enter key.



Installing required software

Below command is use to install git

sudo yum install git -y

below command is use to install java

sudo yum install java it installed version java installed. Don’t install

Please install java 17

sudo yum install java-17

using java version please verify default version of java

using below command we can switch from one version java to another version java

sudo alternatives --config java

below command is use to install maven software

sudo yum install maven

below command to install docker

sudo yum install docker

sudo service docker start

running docker image which published in docker hub in EC2 instance.

sudo docker run -d -p 80:80 akashkale/my-reactjs:a2

sudo docker images

sudo docker ps

<http://publicidaddress:80>

How to install Jenkin in EC2 instance.

First download Jenkin repository

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat/jenkins.repo

import/extract the keys

sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>

now we can install jenkin

**sudo yum install jenkins**

after installation we need to start the Jenkin

sudo service jenkins start this command to start

to check the Jenkin status

sudo systemctl status jenkins

to check admin jenkin password write below command as

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

open the browser <http://publicipaddress:8080>

it ask password and check the password in browser terminal.

Then installed suggested plugin and create the account login to Jenkins dashboard.

Updated Problem Statements

Steps for Problem Statements

1. Creating spring boot project in local machine with starter as web starter and thymeleaf starter.
2. In index.html page write application details with message as well as image.
3. Please run this project with different port number 9090 or 8080
4. Then test the project in local machine working or not.
5. Then create the jar file using maven with eclipse or using mvn command.
6. Please create Dockerfile which is responsible to run spring boot project using docker.
7. In local machine please create docker images and run this image and verify container running or not in local machine or VM machine. Optional
8. After creation please make this folder as local repository and push this code to remote repository. Remote repository contains spring boot project with Dockerfile.
9. Now we need to logic for AWS account and create EC2 instance in AWS account.
10. Then connect EC2 instance using browser or terminal.
11. Need to install git, Java, maven and docker etc.
12. Using Git clone please download remote repository code in EC2 instance terminal
13. Using CD command please move inside a project
14. Using mvn clean package command build the project which create jar file.
15. Using sudo docker build -d my-app . -f Dockerfile create the image
16. Using sudo docker run -d -p 9090:9090 my-app
17. Using public ip address of EC2 instance verify that project you can view or not.

<http://publicipaddress:9090>