**Phase 5 :**

**Day 1 : 06/26/2023**

Junit Testing

Testing and Deployment using devops life cycle

Test NG Open Source framework for testing java programs.( Unit testing and Integration testing)

Selenium Testing with Test NG Framework (Automation Testing )

Introduction to AWS (EC2, S3 and EBS) etc

Git

Maven or Gradle

Docker Devops

CI and CD tool using Jenkin

Overview of Kubernetes

Phase end project

Testing : Testing is use to find the defects or error or bugs in the application.

Running the program develop in any language in directly we are testing that program.

class Operation {

public int add(int x, int y) {

int sum = x+y;

return sum;

}

}

We are testing java programs without main method also known as testing the programs.

Testing mainly divide into 2 types.

1. Black box testing :

Input ------------------🡪Process ------------------🡪Output

1. White box testing

Input ------------------🡪Process ------------------🡪Output

Unit Testing : Unit testing is a type of white box testing which help to test function functionality working or not properly. Unit is smallest code which work independently. Like in a method or function etc.

To do this type of unit testing

Java use 2 framework

jUnit : junit testing open source framework which help to do unit testing

Test NG (next generation) : Test NG is base upon Junit testing. Light weighted which help to do

Unit testing as well as integration testing.

JavaScript -🡪 Jasmine is an open source framework which help to do unit testing for JavaScript programs.

React JS 🡪 JEST is an third party framework provided by Facebook which help to do the unit testing for React JS programs.

We will do unit testing using Junit

Junit 3.x without annotation

Junit 4.x with annotation

jUnit 5.x with more than third party library and support java8 features with annotation

test case : test case is a type of class which contains more than one test function which help to test function functionality.

test suite : test suite is use to run more than one test case classes.

Junit hook :

4 hook

@Before : this function get called before each @Test function

@After : this function get called after each @Test function

@BeforeClass :this function get called only once before one or all @Test function

@AfterClass : this function get called only once after one or all @Test function

TestSuite : Test suite is use to execute more than one test case classes.

TestNG : Test NG another testing framework inspired by junit testing framework. It is an open source testing framework similar to junit test framework. In Test NG (next generation).

Test NG is combination of junit and nunit

It is designed better than Junit testing especially when we do the integration testing.

Features of Test NG framework

1. It generate report in html format (by default).
2. It provided few more annotation which were missing in jUnit.
3. It allow use to do group as well as priority base testing.
4. It support parallel testing features. jUnit do sequentially testing.

Unit testing using Test NG

Phase 3 project discussion

**Phase 5 :**

**Day 2 : 06/27/2023**

Unit : we create separate test suite which help to run more than one test case.

In Unit is it is a type of class.

But in TestNG that class replace by xml file. This file hold the details about all TestNG class.

In Test Ng all test function execute by default in alphabetical order.

In Test NG inside @Test annotation we can use priority property to run the test base upon the priority.

TestNG hook.

@BeforeSuite : it will call only once when we run through suite file. It will call only once

Inside suite we can configure more than one TestNG class. each class contains

More than one test function

Before all test function part of same TestNG class or different TestNG class

@BeforeTest hook get called only once.

@BeforeClass : it will call each class level only once.

@BeforeMethod : it will call each @Test function. This will call again and again.

**Day 3 : 06/28/2023**

Selenium : Selenium is one of the most widely used open source Web UI(user interface) automation tool.

Selenium is platform independent, language independent and browser independent Web UI tool.

Selenium support by Java, C#, Python, JavaScript etc.

Selenium can be use to automate functional test (actual business code) non function test (it will support to do the task).

It can be integrate with DevOps tool like Git, Jenkin, Docker and Kubernetes.

We can integrate Selenium with jUnit or TestNG framework.

We can do Selenium code using Java as well as Selenium provided OWN IDE which we can configure with browser or we can download to do the Testing for UI Component doesn’t matter that application develop using any language.e

We need to download the browser specific driver.

To connect mysql or oracle database we downloaded jar file manually or using maven tool.

In Selenium we need to download the browser driver in the form of .exe or other format base upon OS support.

In Web Application all contents are known as DOM. (Document Object Model).

If we want to load the tag or dom elements contents then we need to refer those tags using

Tagnaname, classname, idname etc.

First we need to refer to dom tags base up selector and Selenium provided WebElement API which help to read, write and update DOM content using Java.

**Phase 5 :**

**Day 5 : 06/30/2023**

RDBMS (Relational Database management System)

Limitation of MySQL or Oracle or db2 etc.

All RDBMS Databases are schema base database. Means before storing any record in database first we need to create table with number of column as well as column data types.

Structure data

Semi structure

Un structure

Emp

Id Name Salary age city

1 Ravi 12000.50 null null

2 Ramesh 14000.50 null null

3 Lex 16000.75 21 null

4 Neeta 18000.00 null Bangalore

No SQL : In No SQL Database we can store the data in any format apart from Table format. Like Json, xml, graph etc.

Mongo DB : Mongo DB is a type of open source no sql database which help to store the data using document in the format of json.

Mongo db is cross platform high performance no sql database.

RDBMS Mongo DB

Database Database

Table Collection

Record (same types) document (can be same type or different types)

Table format using cell each document hold information in json format

Each cell allow single value document can be same or different types.

Open mongo terminal using command as

mongo

show databases

or

show dbs

use databasename;

like

use phase5; if database present it will move to existing database else it will create to move inside that database.

show colletions;

Or

show tables; this command is use to show all collection present in current database.

Mongo db provided pre defined object ie db which contains lot of pre defined functions which help

Create collection, store document, delete, document, update document and retrieve documents.

db.createCollection(“CollectionName”);

To store the document in collection

db.CollectionName.insert({key1:value1,key2:value2,key3:value3});

View the documents from collection

db.collectionName.find({});

mongo db internally create \_id pre defined field for each document to maintain unique ness between two document and it will generate unique random alpha numerical value.

If we want we can pass the value but we can’t change the field name it is consider as primary key in mongo db.

Emp

\_id name salary age city deptId

In mongo db while inserting document in json format if collection present it will insider that document inside that document else it will create.

> db.Emp.insert({\_id:1,name:"Ravi",age:21,salary:240000,city:"Bangalore",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:2,name:"Ramesh",age:24,salary:280000,city:"Bangalore",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:3,name:"Lokesh",age:27,salary:210000,city:"Delhi",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:4,name:"Neena",age:21,salary:180000,city:"Delhi",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:5,name:"Leena",age:22,salary:190000,city:"Bangalore",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:6,name:"Teena",age:23,salary:210000,city:"Delhi",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:7,name:"Heena",age:24,salary:220000,city:"Mumbai",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:8,name:"Seena",age:26,salary:260000,city:"Mumbai",deptId:101});

WriteResult({ "nInserted" : 1 })

db.Emp.find({}); : it retrieve all documents from Emp collection

db.Emp.find({})[0]; : it retrieve 0 position index position document from Emp collection

db.Emp.find({})[1].name : it retrieve particular index position field document values.

Retrieve specific fields value from document.

db.Emp.find({condition},{fieldname:1});

db.Emp.find({},{name:1}); display name and \_id

db.Emp.find({},{name:1,city:1}); display name, city and \_id

db.Emp.find({},{name:1,city:1,\_id:0}); display name and city no \_id

retrieve the document from collection with conditions.

db.Emp.find({\_id:1});

db.Emp.find({city:"Bangalore"});

db.Emp.find({salary:24000});

db.Emp.find({salary:{$gt:200000}});

db.Emp.find({salary:{$gte:200000}});

db.Emp.find({salary:{$lt:200000}});

db.Emp.find({salary:{$lte:200000}});

db.Emp.find({salary:{$eq:200000}});

db.Emp.find({salary:{$ne:260000}});

and/or operator

db.Emp.find({$and:[{salary:{$gt:150000}},{salary:{$lt:200000}}]});

db.Emp.find({$or:[{salary:{$gt:250000}},{city:"Bangalore"}]});

sort the documents.

db.Emp.find().sort({age:1}); sort age by document in asc order

db.Emp.find().sort({age:-1}); sort age by document in desc order

update document

update salary using \_id field

db.Emp.update({\_id:1},{$set:{salary:250000}});

update salary and age using \_id field

db.Emp.update({\_id:1},{$set:{salary:200000,age:26}});

this query can update more than one document if conditions meet.

db.Emp.updateMany({city:"Delhi"},{$set:{city:"New Delhi"}});

remove documents

it will remove using \_id field

db.Emp.remove({\_id:1})

it will remove using city field

db.Emp.remove({city:"Bangalore"});

MySQL

Student details

Sid(pk) sname age, marks

1 Leena 21 67,88,98,65

Student

SID(PK) Sname Age

1 Leena 21

2 Meeta 25

Marks

Mid(PK with auto increment) sid(fk) marks

100 1 67

101 1 88

102 1 98

103 1 65

104 2 78

Mongo db relationship

One to One

One to many

Trainer

PK

Tid TName tech

1 Raj Java

Student

PK FK

Sid SName age tsid

100 Neena 21 1

101 Meena 22 1

In mongo db we can achieve relationship using two ways

1. Embedded style relationship : single collection
2. Linking style relationship : more than one collection

Embedded style relationship

Employee has one Address : one to one relationship

Employee working more than one project at same time : one to many relationship

db.Employee.insert({\_id:1,name:"Ravi",age:21,address:{city:"Bangalore",state:"Kar"}});

db.Employee.insert({\_id:2,name:"Rajesh",age:22,address:{city:"Mumbai",state:"Mh"},project:[{pid:1122,tech:"Java"}]});

db.Employee.insert({\_id:3,name:"Neena",age:23,address:{city:"Mumbai",state:"Mh"},project:[{pid:1122,tech:"Java"},{pid:1123,tech:"Python"}]});

This code display the output in proper or pretty format.

db.Employee.find().pretty();

Linking Style

Trainer

\_id(PK) TName tech

1 Ravi Java

db.Trainer.insert({\_id:1,tname:"Ravi",tech:"Java"});

db.Trainer.insert({\_id:2,tname:"Rajesh",tech:"Python"});

Student1

\_id(PK) SName age tsId

100 Leena 21 1

101 Veena 22 1

db.Student1.insert({\_id:100,tname:"Meeta",age:21,tid:db.Trainer.find()[0].\_id});

db.Student1.insert({\_id:101,tname:"Leeta",age:22,tid:db.Trainer.find()[0].\_id});

db.Student1.insert({\_id:102,tname:"Keeta",age:23,tid:db.Trainer.find()[1].\_id});

Student2

db.Student2.insert({\_id:100,tname:"Meeta",age:21,tid:db.Trainer.find()[0]});

\_id(PK) SName age tsdetails

100 Leena 21 {\_id:1,tname:”ravi”,tech:”Java”}

**Phase 5 :**

**Day 6 : 07/03/2023**

Cloud Computing using AWS

Cloud the term refer to a network or the internet.

Machine

System software window, linux, mac etc.

Application software : java, python, .net, node js

SAP, Sales force, MQ/MB etc

My sql, oracle, db2 etc

Tomcat, web logic, jboss, glashfish etc

In Simple in Cloud computing rather than running any application or software or tool or database in local machine we are running in cloud or virtual machine

There are certain services and model working behind the scene making the cloud computing feasible and accessible to the end user (developer, programmer, tester, etc).

Cloud mainly divided into four types (Deployment model).

Public cloud :public cloud allow system and services to be easily accessible to the general public with the login details.

Private cloud : private cloud allow to access specific people within their organization.

Hybrid cloud : it is a combination of public and private cloud.

Community cloud : this cloud maintain by more than one organization.

Service Model

IaaS (HaaS): Instructure as a Service : The main advantage of using IaaS/HaaS is that it helps to avoid the cost and complexity of purchasing and managing physical server, tool or product.

Storage, networking, load balancing etc.

PaaS : Platform as a Service : PaaS cloud computing platform created for programmer to deploy, test and manage ie build the project.

SaaS : Software as a service : SaaS also known as on-demand-software etc. it is a software in which application are hosted by cloud service provider. User can access this application with the help of login details.

Cloud computing provider

1. AWS
2. Azure
3. Google cloud
4. Oracle cloud

Java or MEAN Stack or MERN Stack + AWS cloud

Java + Azure

.net + Azure

Cloud S3 (Simple Storage Service)

It is like a google drive which help to share the data . Data can be any type, with high security, 24/7 availability.

EC2: Amazon Elastic Compute Cloud :EC2: EC2 help us to create virtual lab.

Base upon our own configuration like OS, RAM, number of instance etc.

In EC2 we can install required software which help to deploy and test the application.

If we install required software In local machine those software we can access through localhost.

But if we run all application on EC2 instance AWS provide the unique ip address for that virtual machine.

<http://localhost:9090/>

<http://127.0.0.1:9090>

**Phase 5 :**

**Day 7 : 07/4/2023**

This command is use to download jar file present in S3 bucket into ES3 instance terminal

wget <https://akash123123.s3.amazonaws.com/spring-boot-with-aws-0.0.1-SNAPSHOT.jar>

ls this command is use to find the list of files and folder in current path.

create-react-app react-test

open this project in vs code

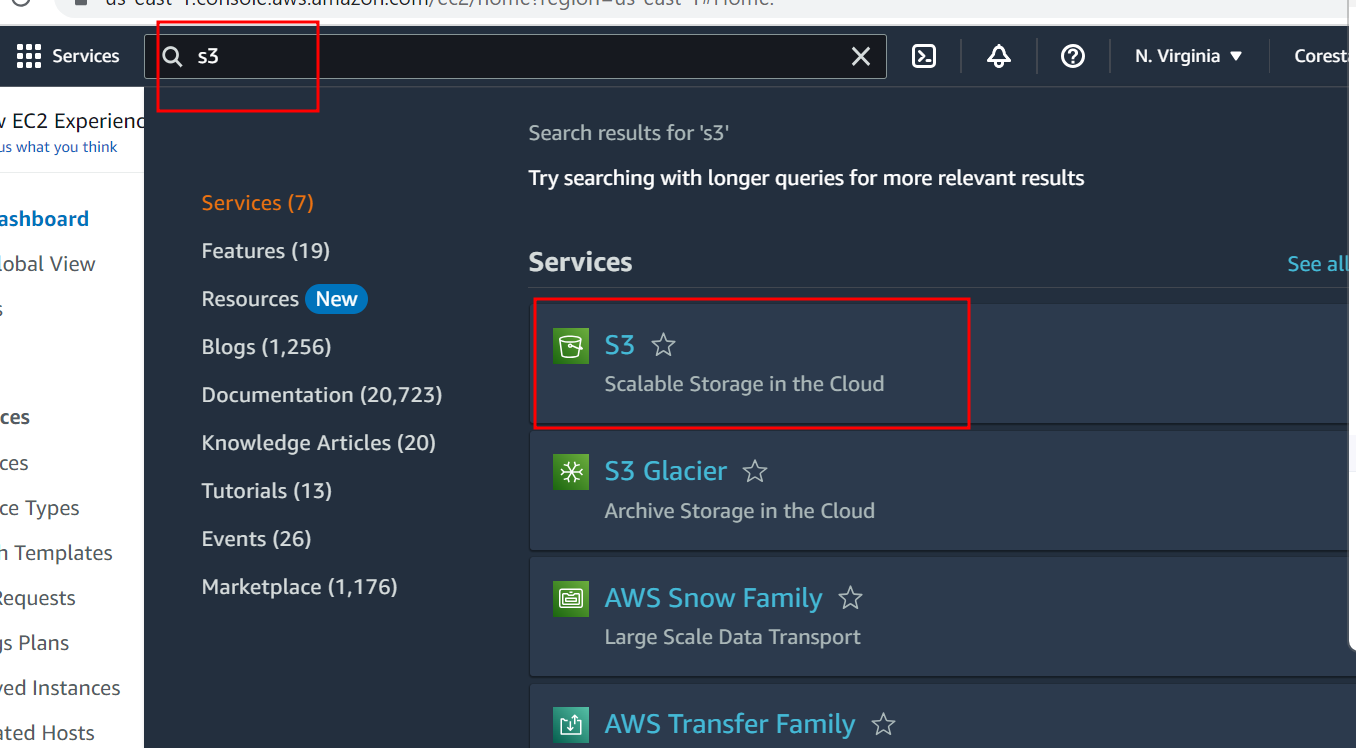
in app.js file write your name.

Welcome to React js project created by Akash Kale

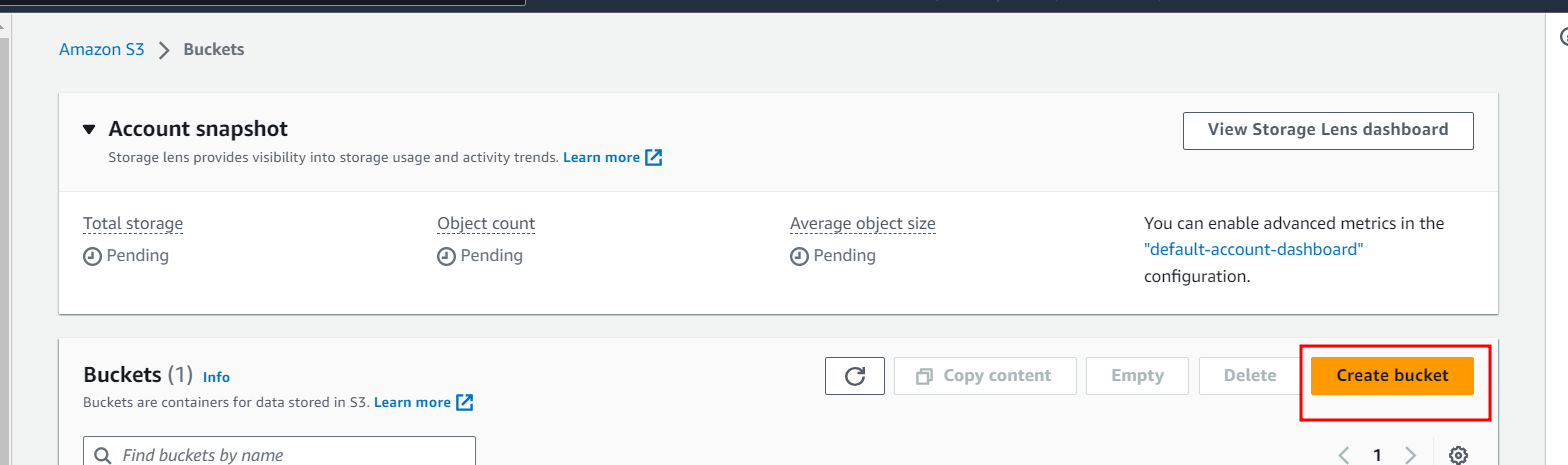
Creating S3 bucket and adding Java Jar file in S3 Bucket

First login to AWS Account

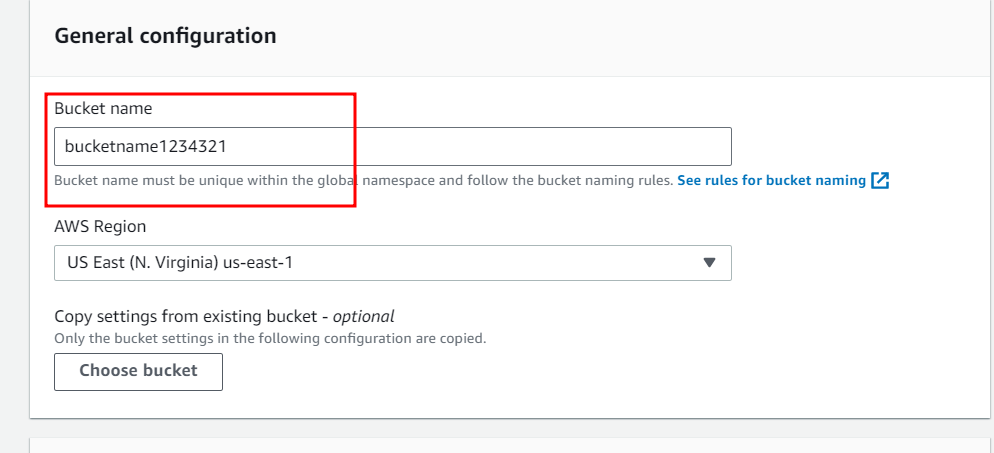
And in search box please write S3

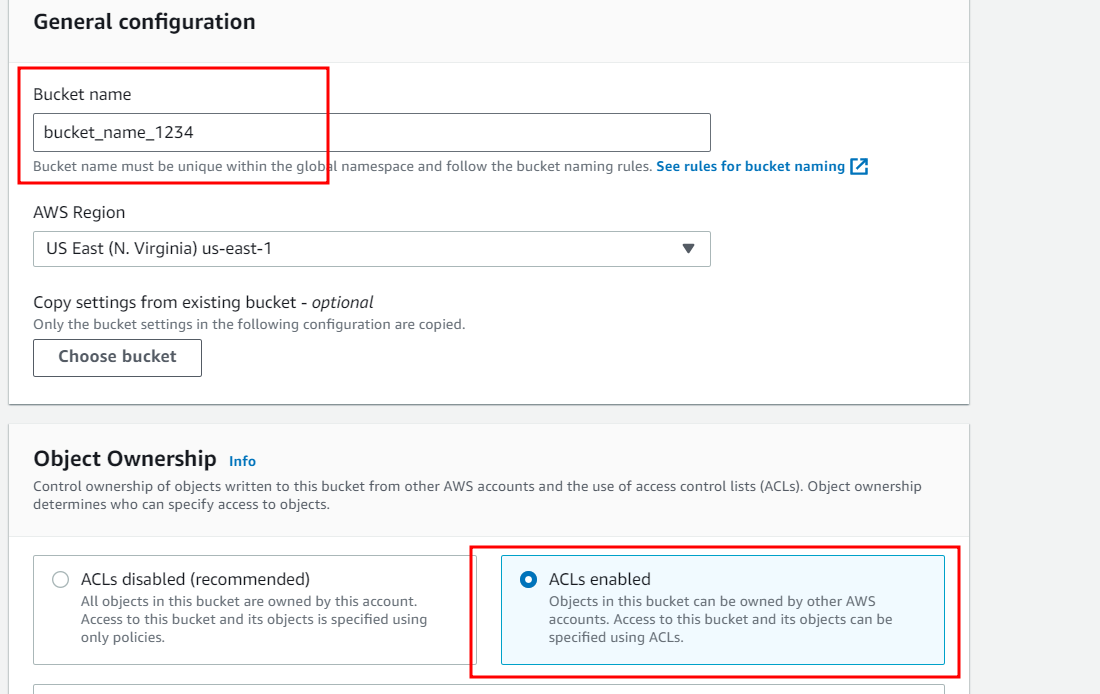


In right corner search create button option



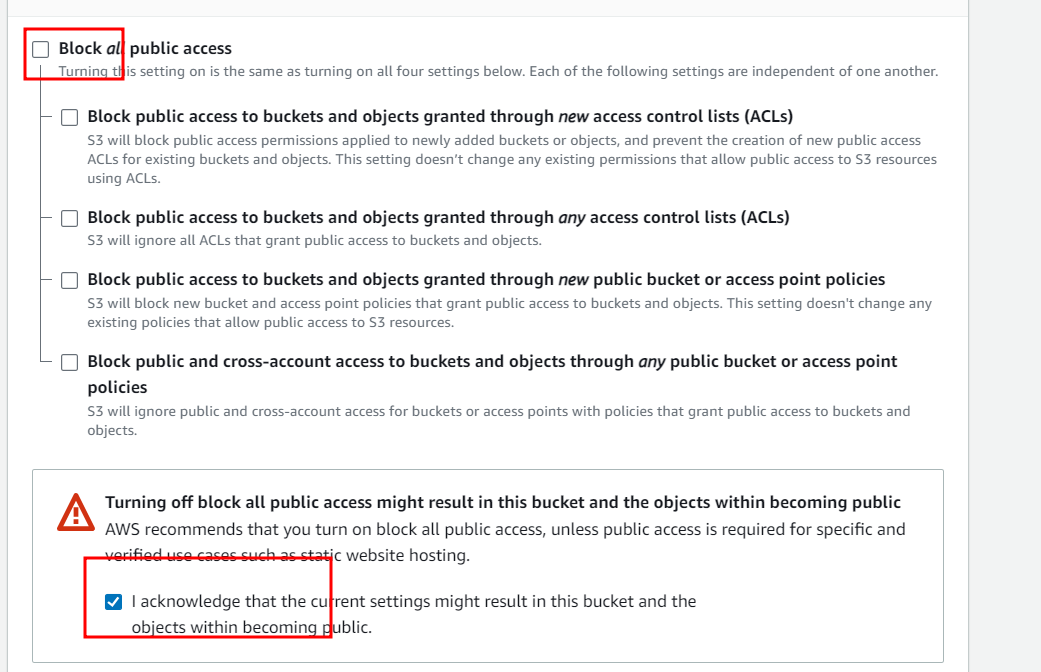
Plz provide bucket name and enable ACL radio button



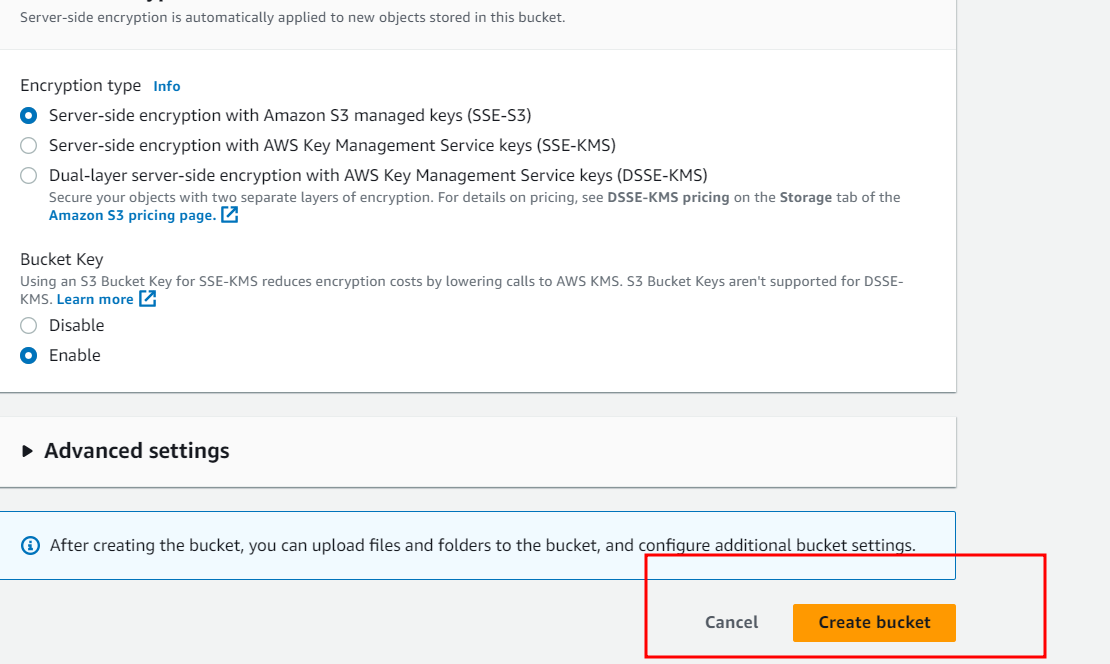


Deselect block all public option

And acknowledged.

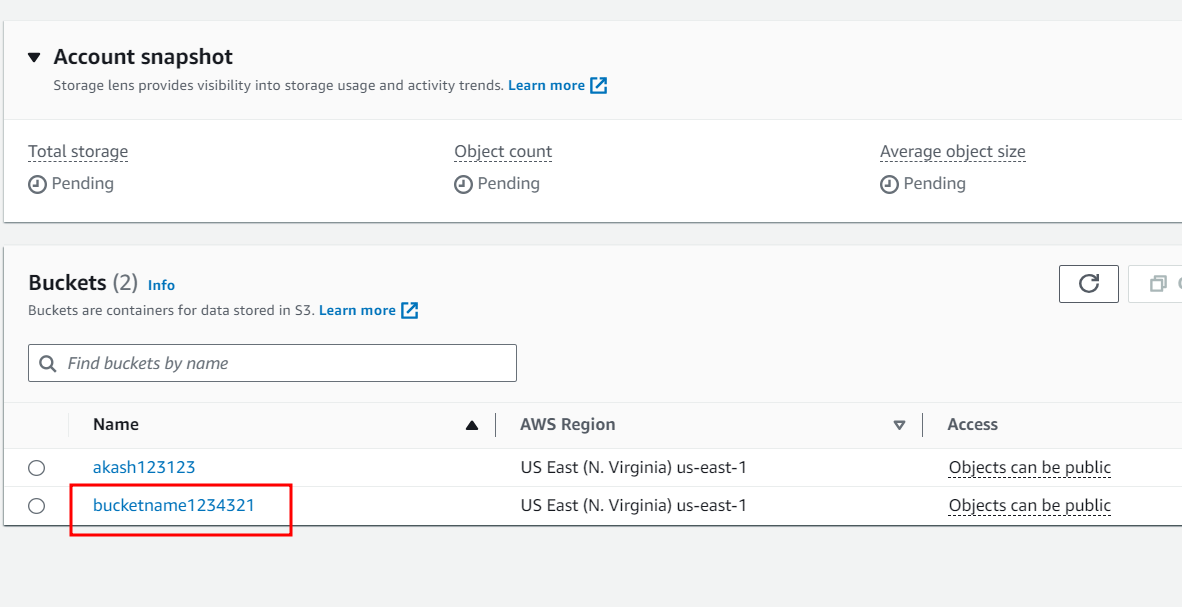


Then click on create bucket option

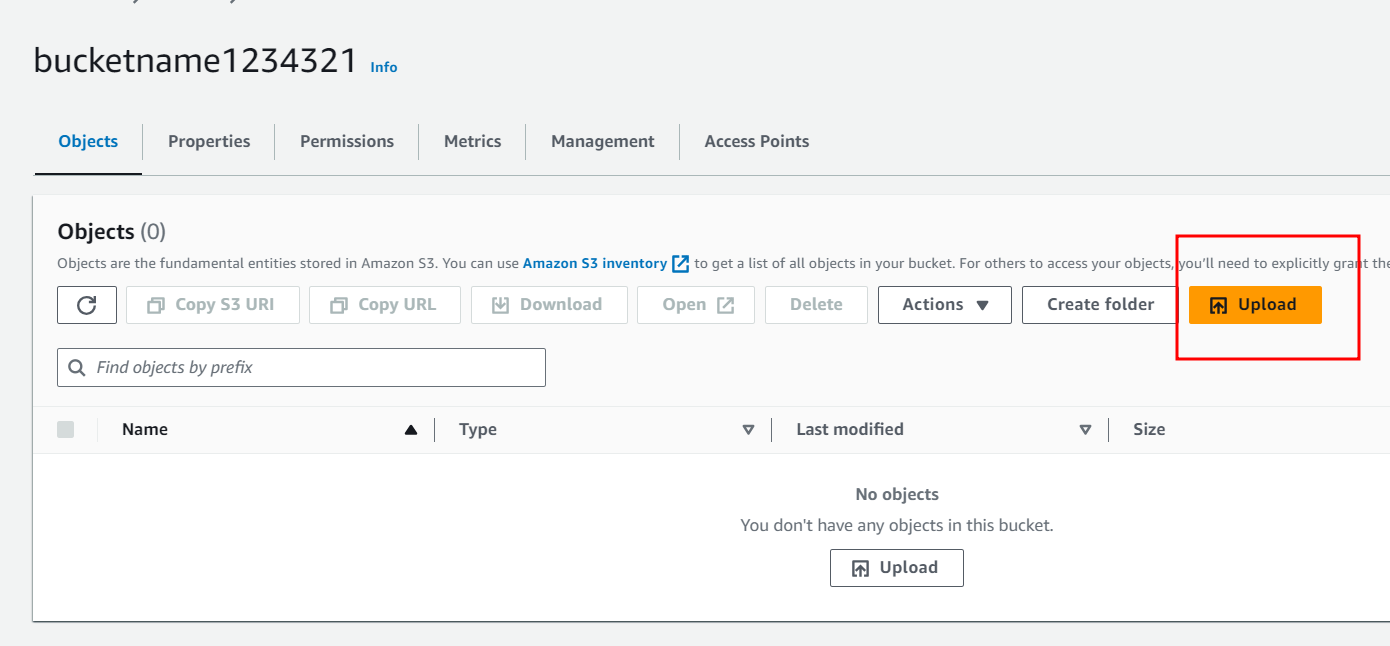


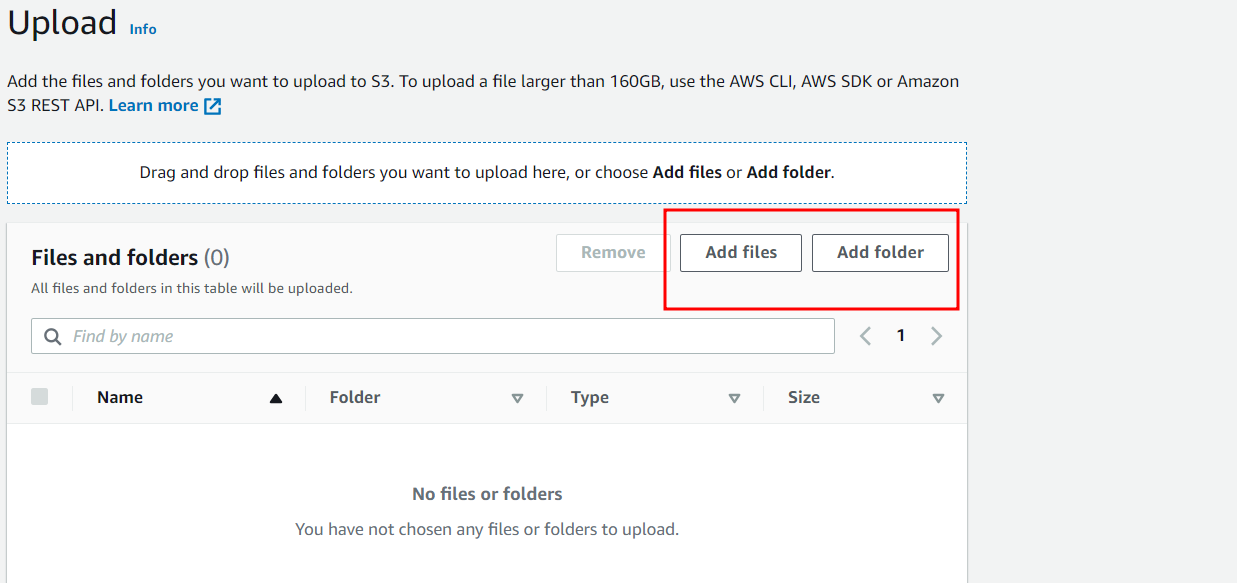
Now adding files or jar file inside a bucket

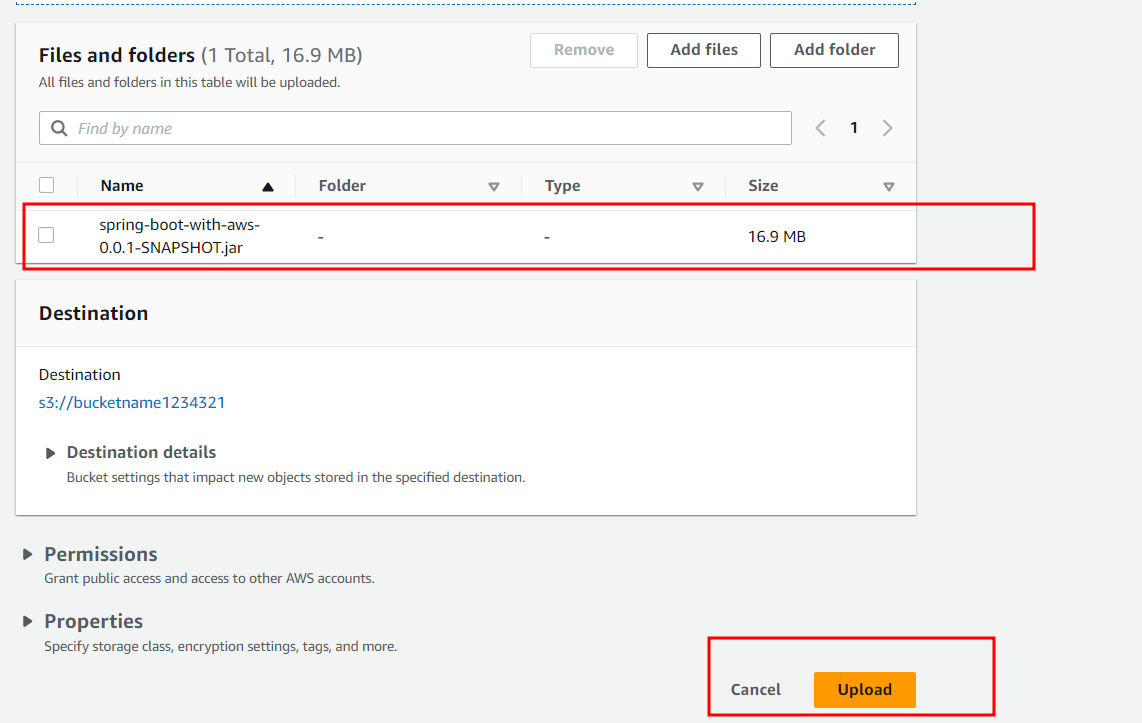
First move inside a bucket



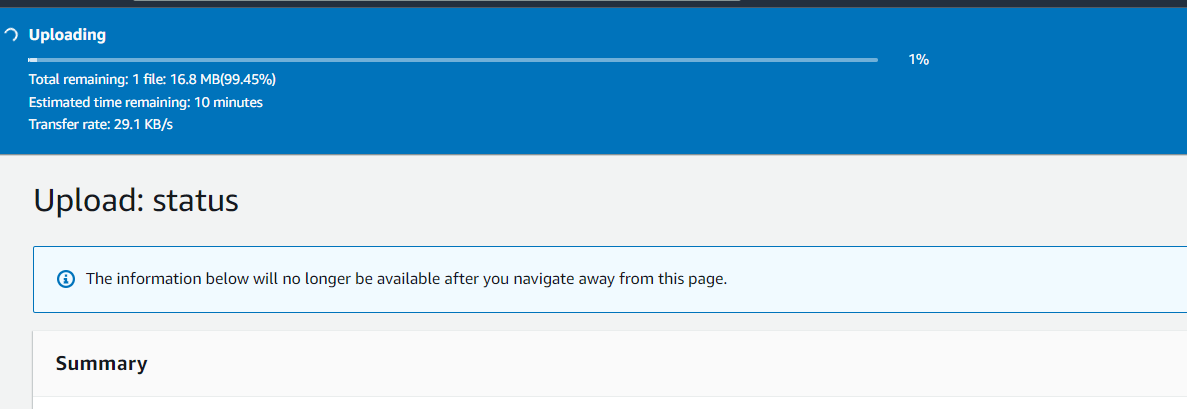
Click on upload button to upload the jar file or pdf file





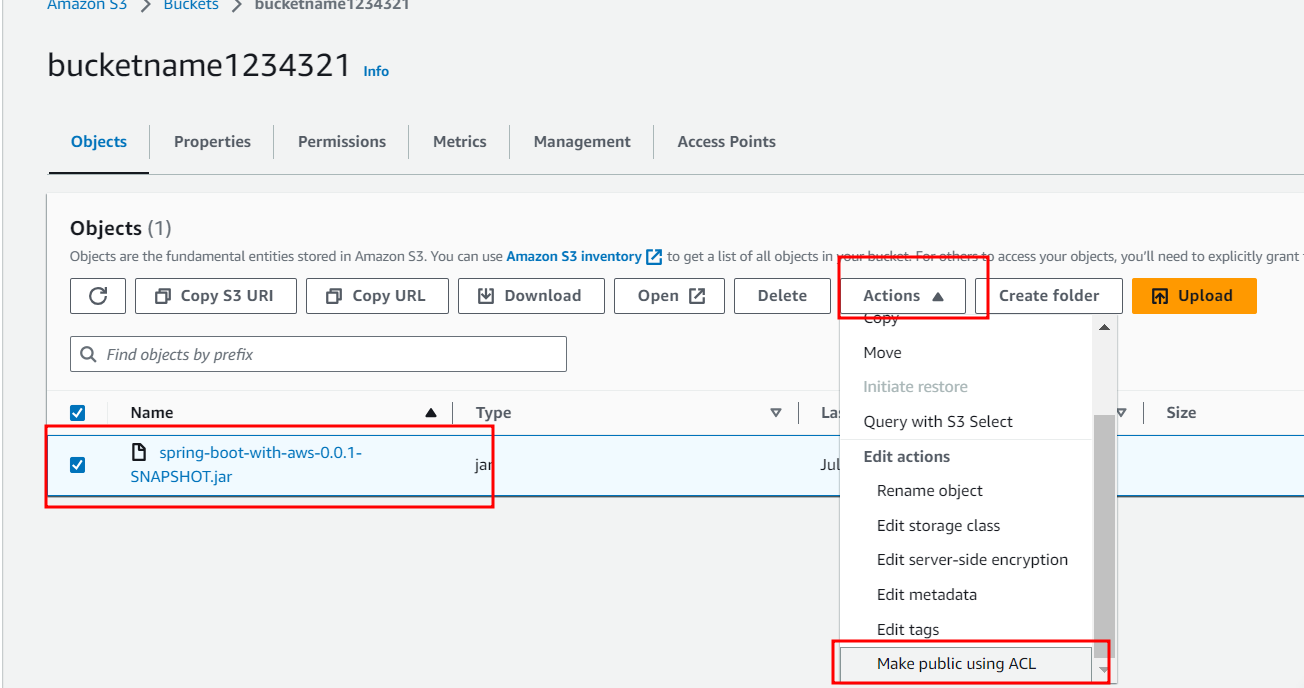


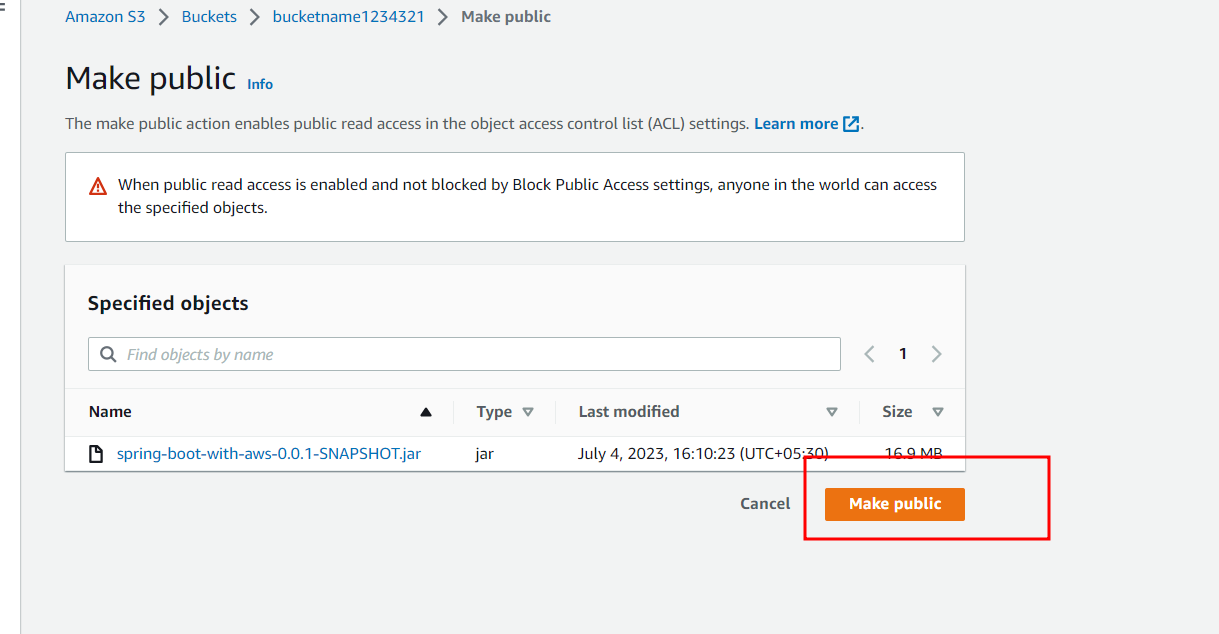
Click on upload

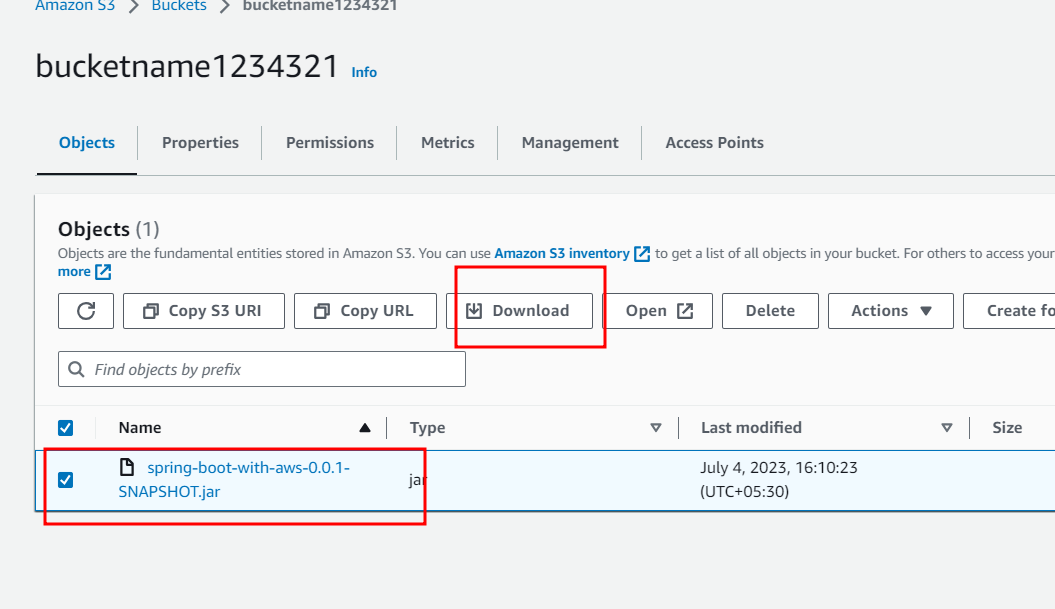


To allow to access this file for anyone please select the file and in action dropdown option

Make public ACL enable





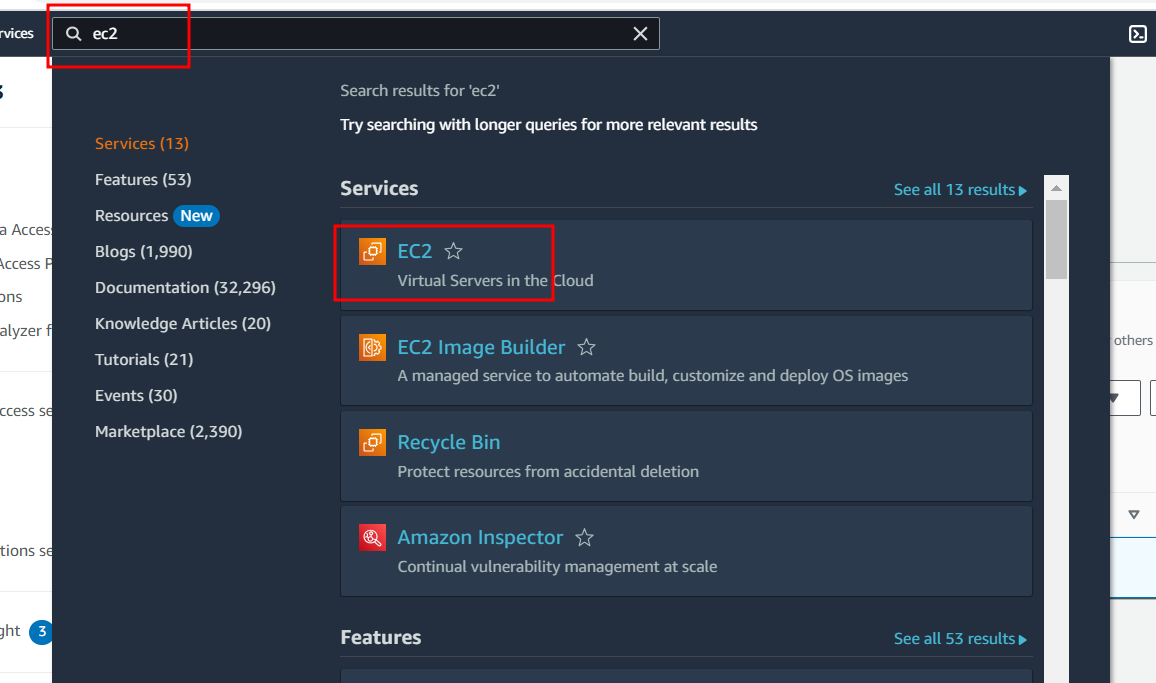


You can download as well as we can provide URL for that jar file which we uploaded in S3

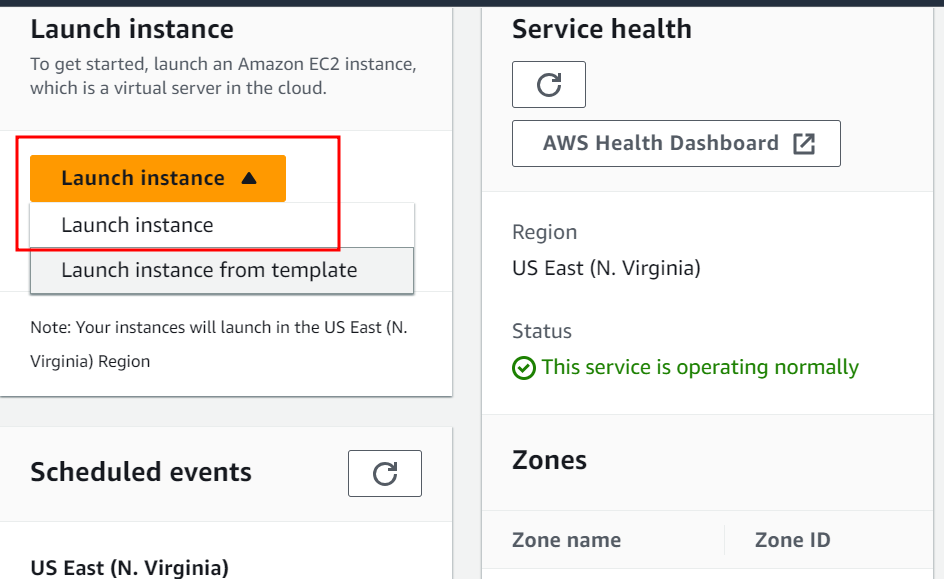
<https://bucketname1234321.s3.amazonaws.com/spring-boot-with-aws-0.0.1-SNAPSHOT.jar>

now we will create ES2 instance

in Search write EC2 instance



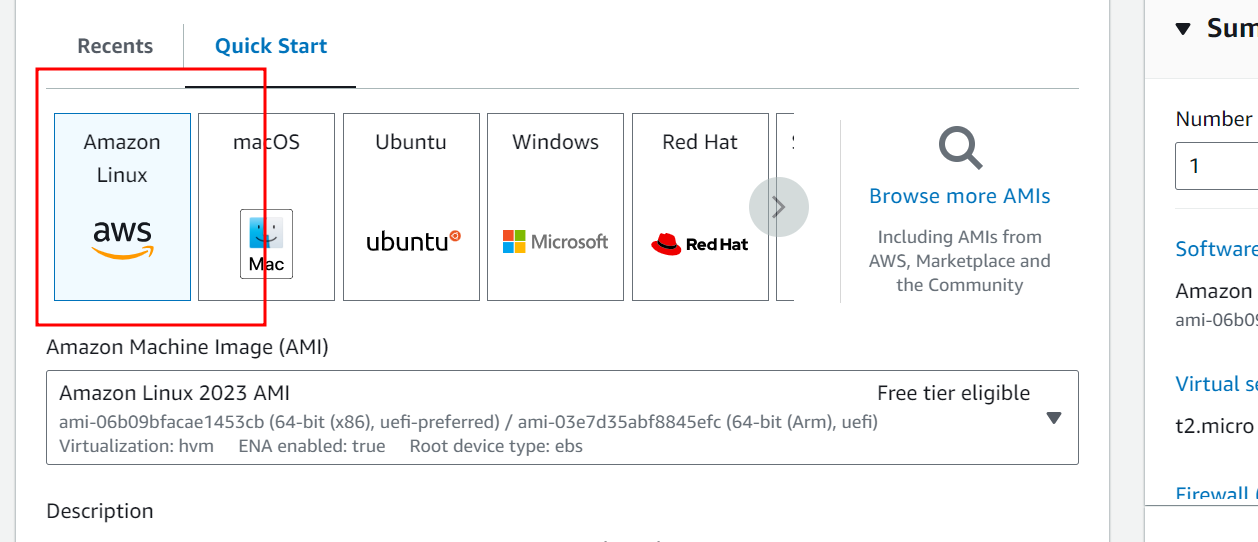
Click on launch instance



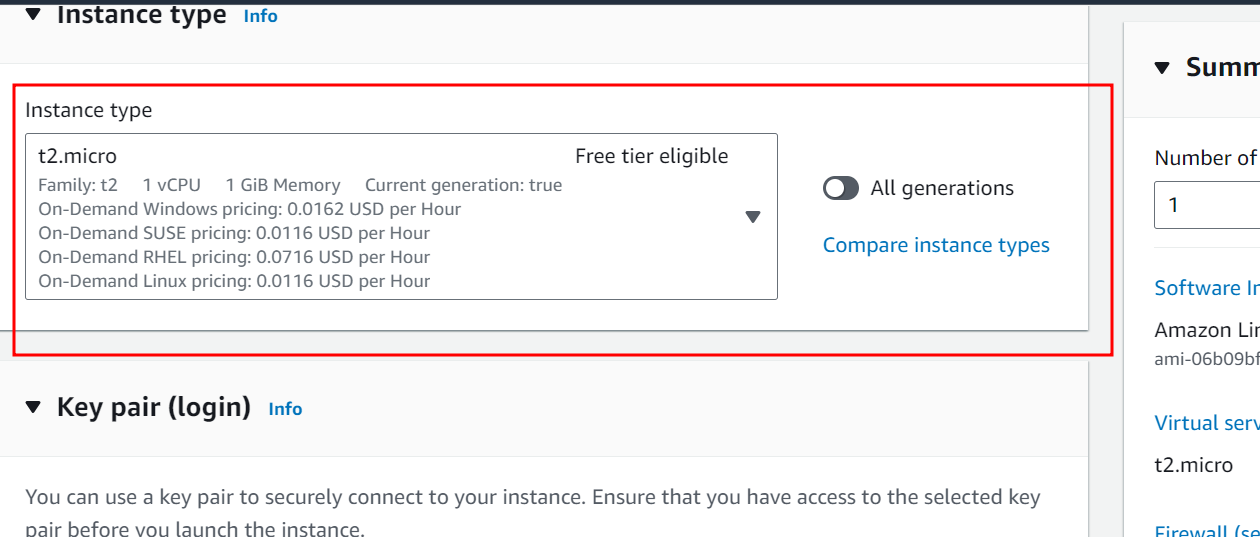
Provide the ec2 instance name



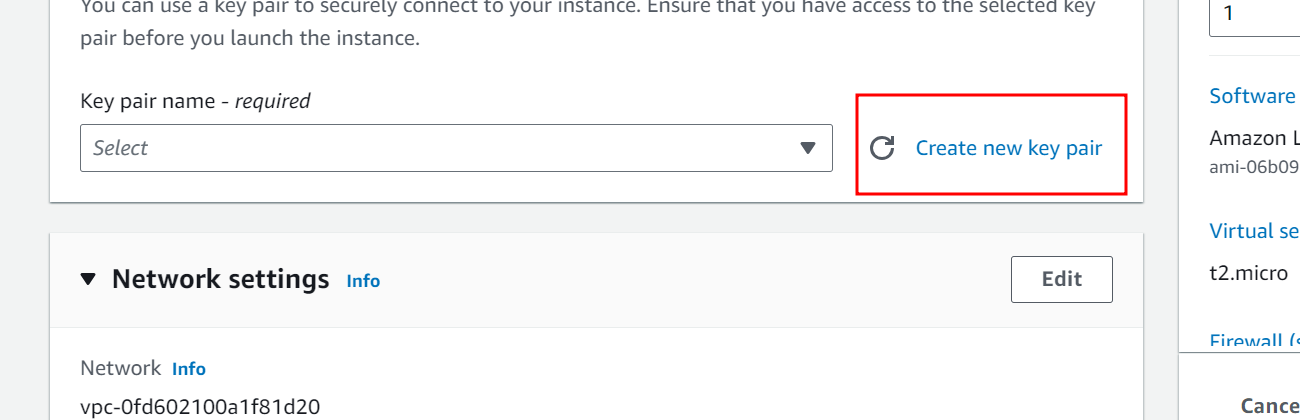
OS Amazon Linux

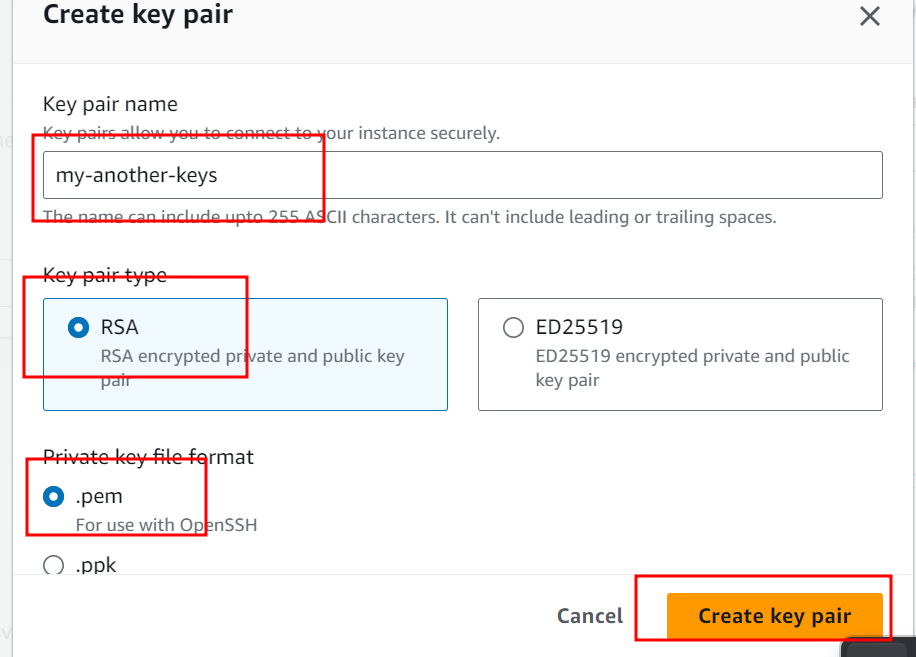


Instance type : free tier

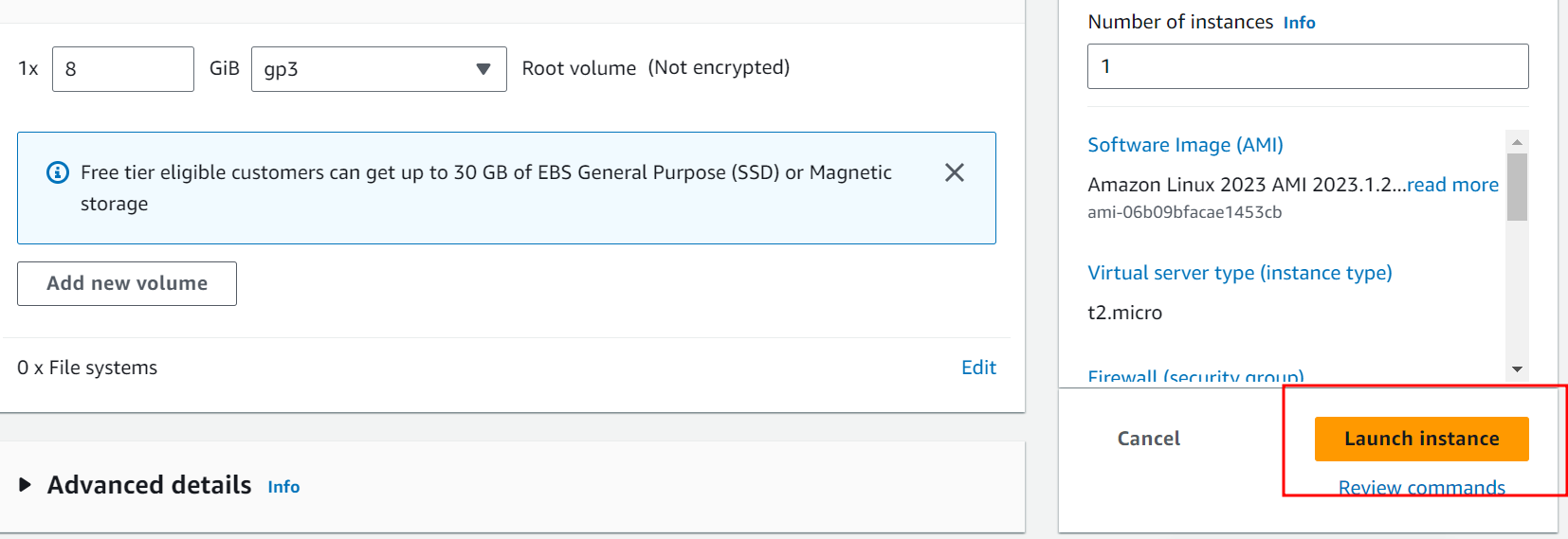


Click on new key pairs

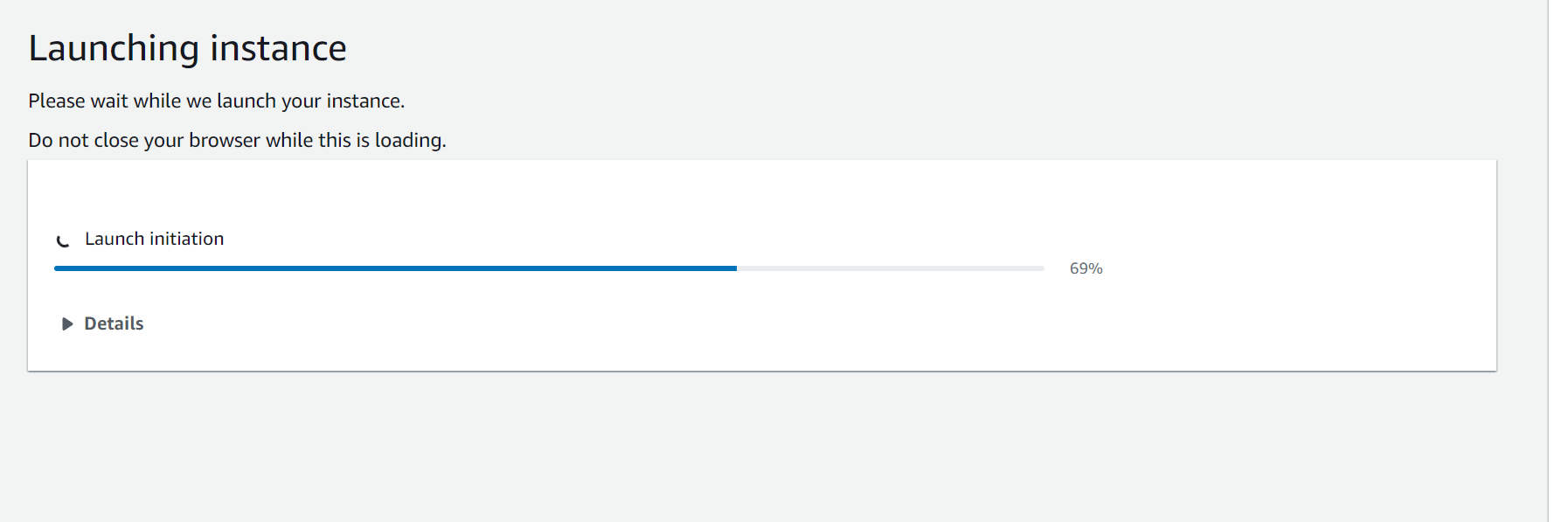




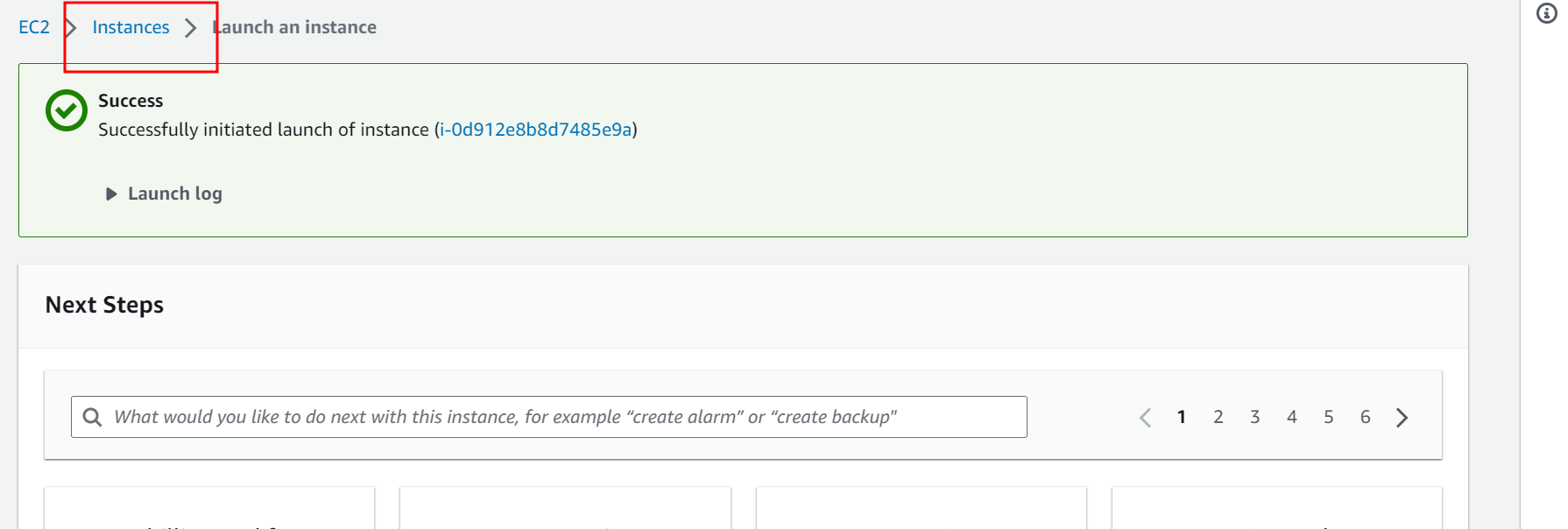
Remaining option are default don’t do any changes



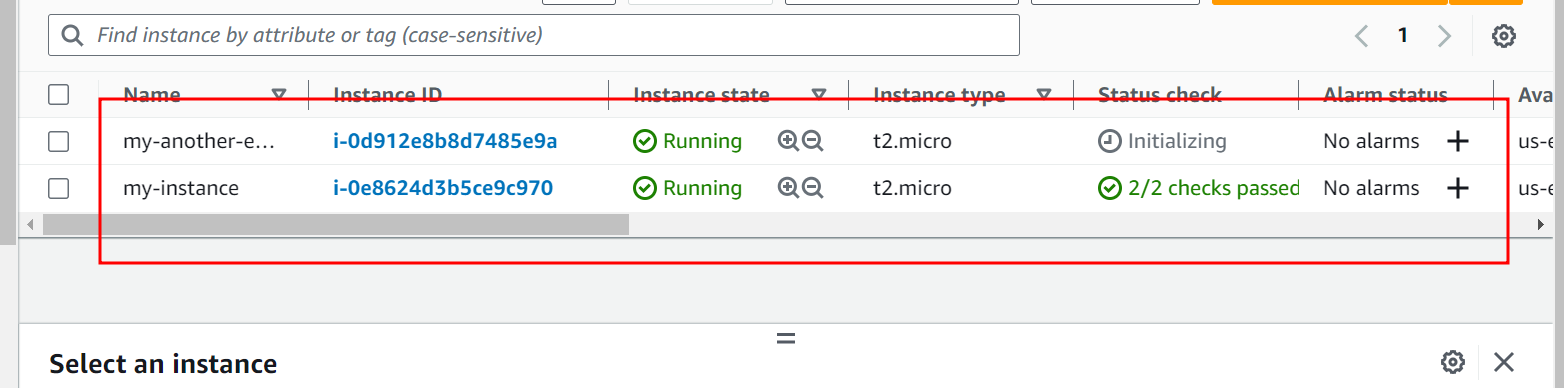
And click on launch instance



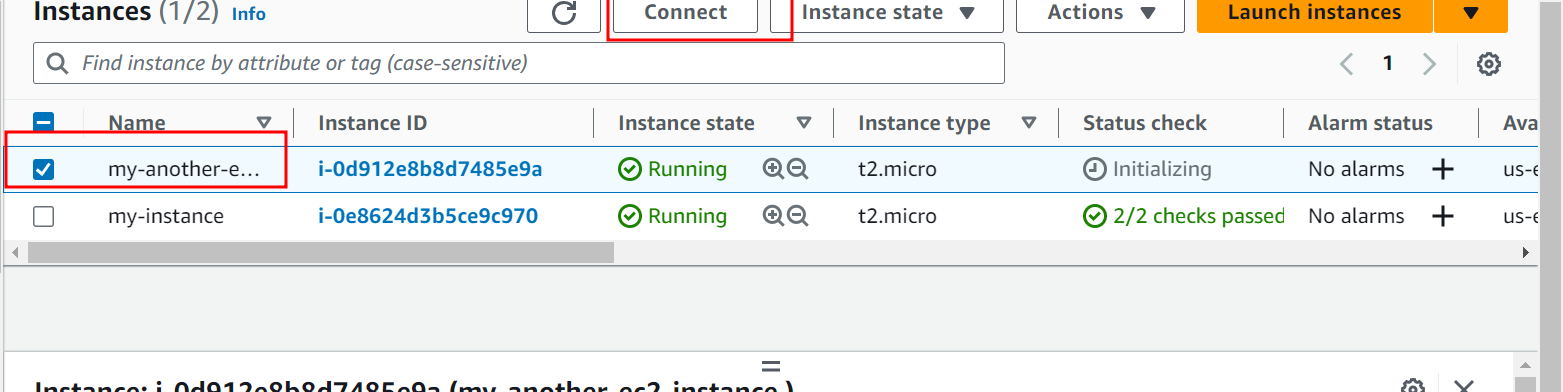
After successfully instance created please click on instance option



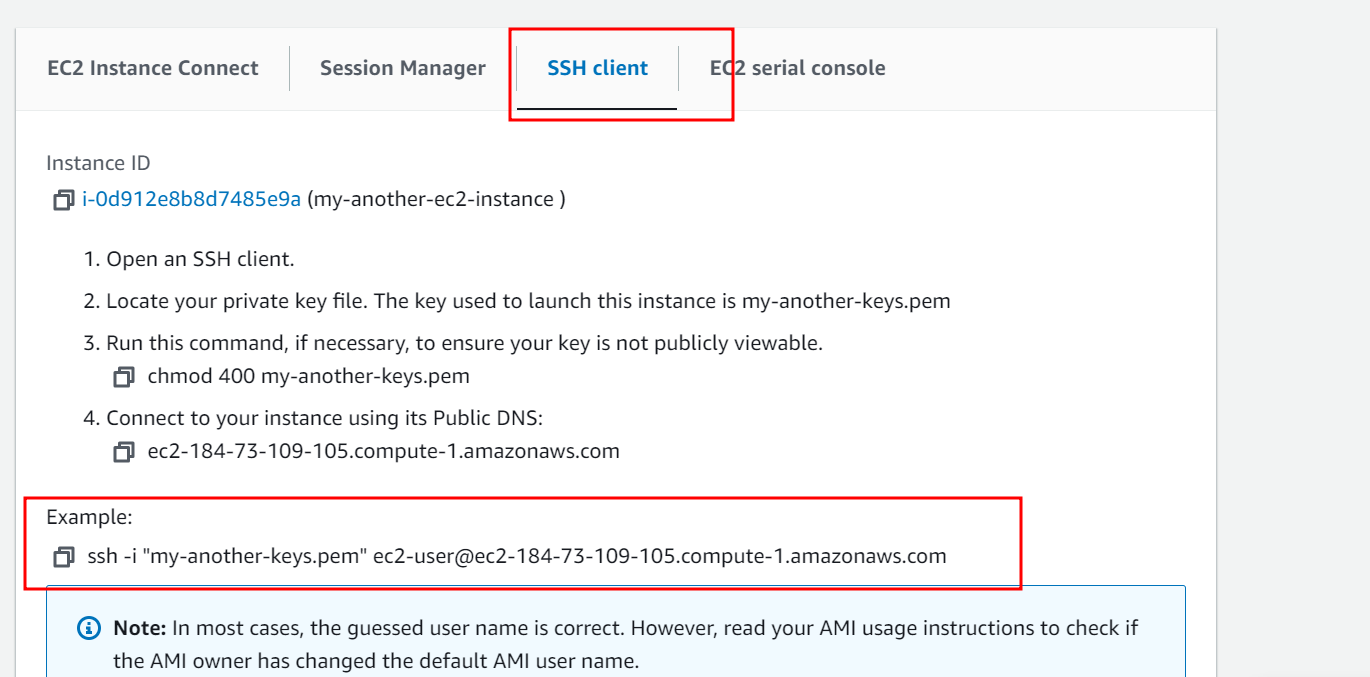
Wait for few minutes to start the instance



Now select instance check box and click on connect button

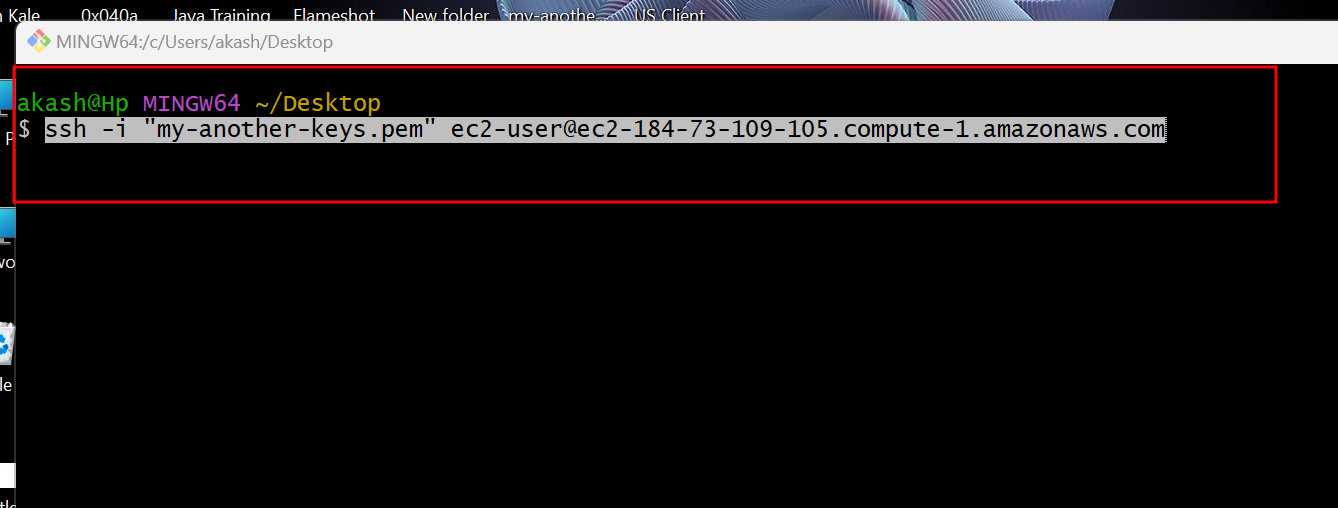


In pop up window select ssh option and copy example command

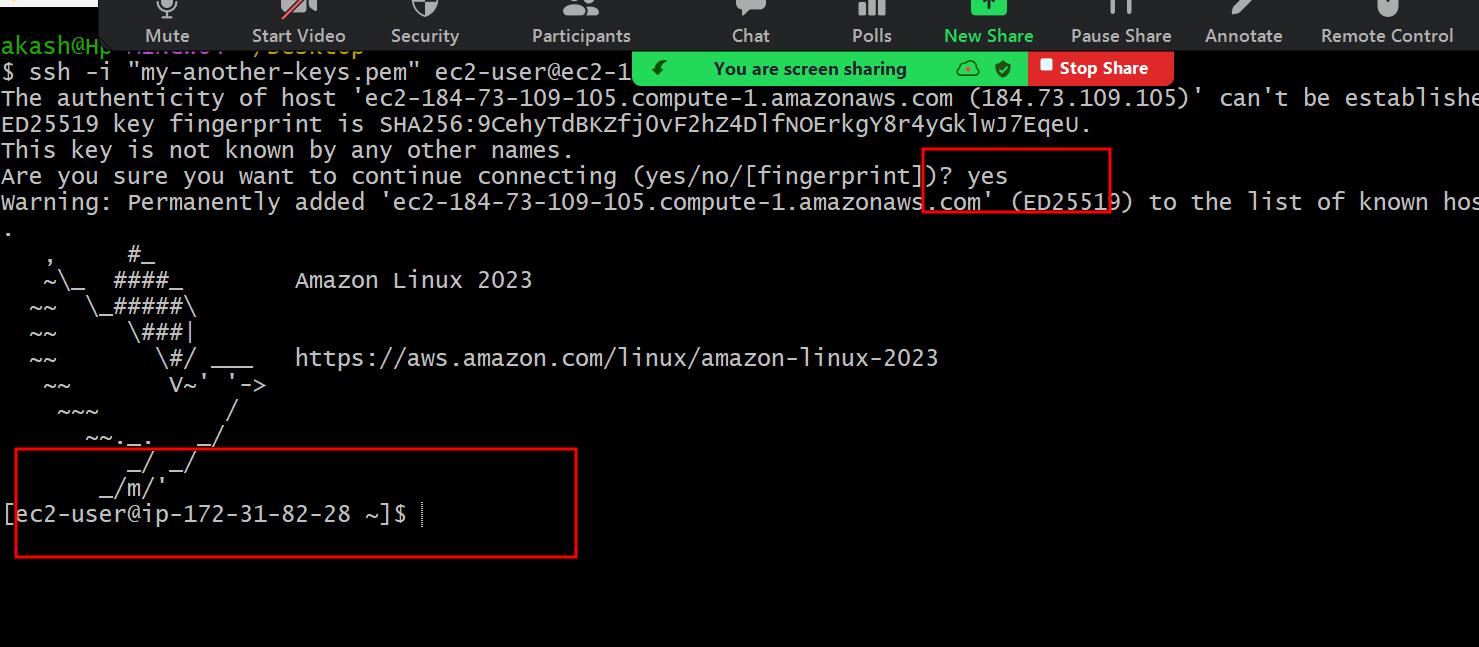


Then window user open git terminal in the place where you download .pem key file

Non window user open terminal

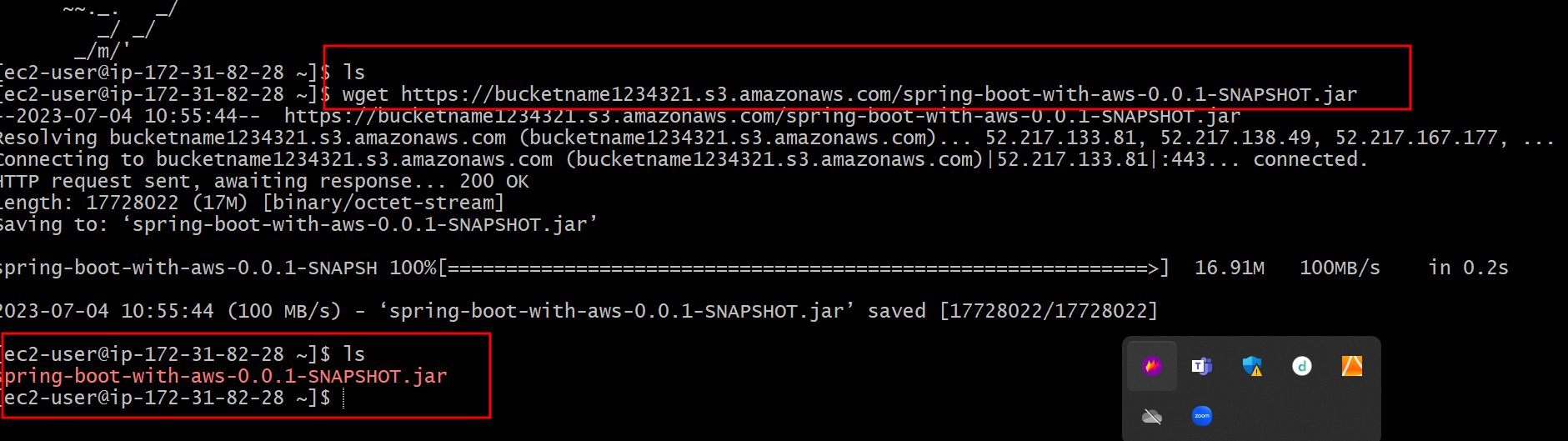


If everything go file then we can connect successfully

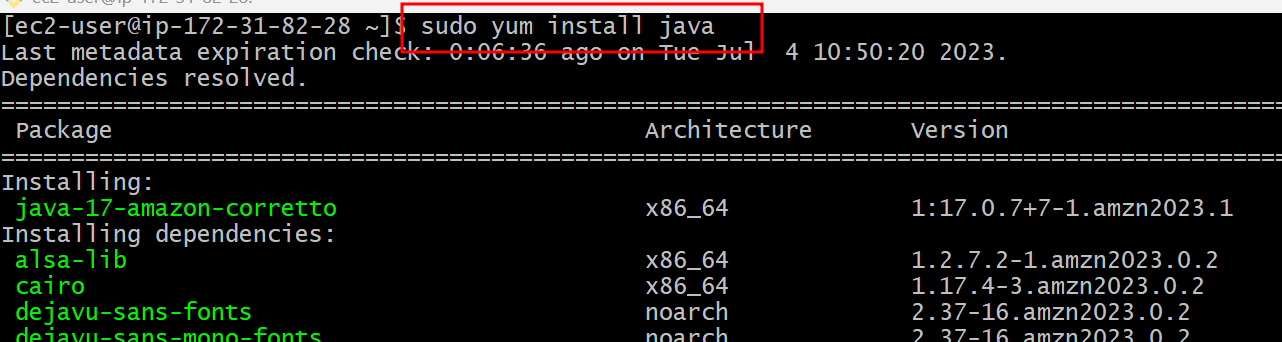


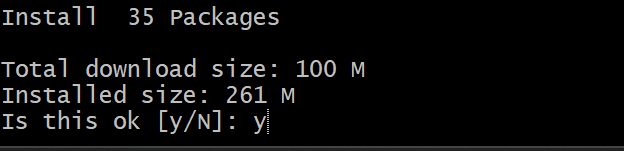
Now we will download jar file present in S3 into EC2 instance using command as wget

**wget URL**

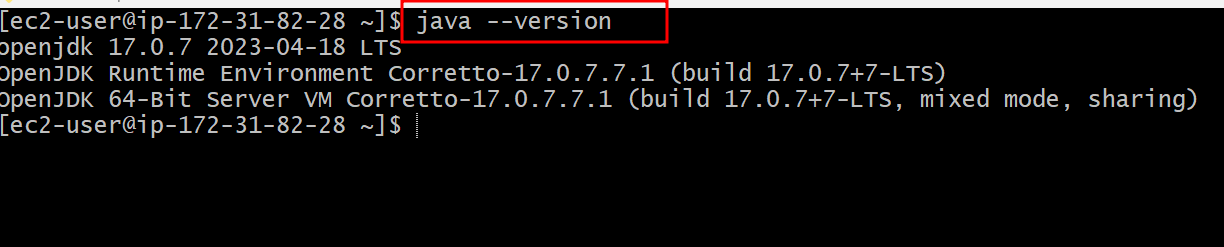


Now to run this jar file we need to install java software



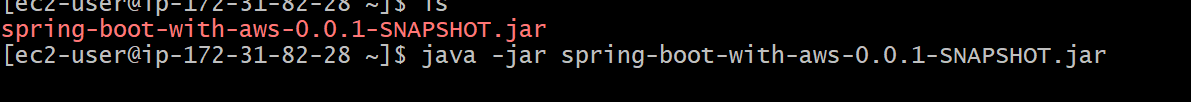


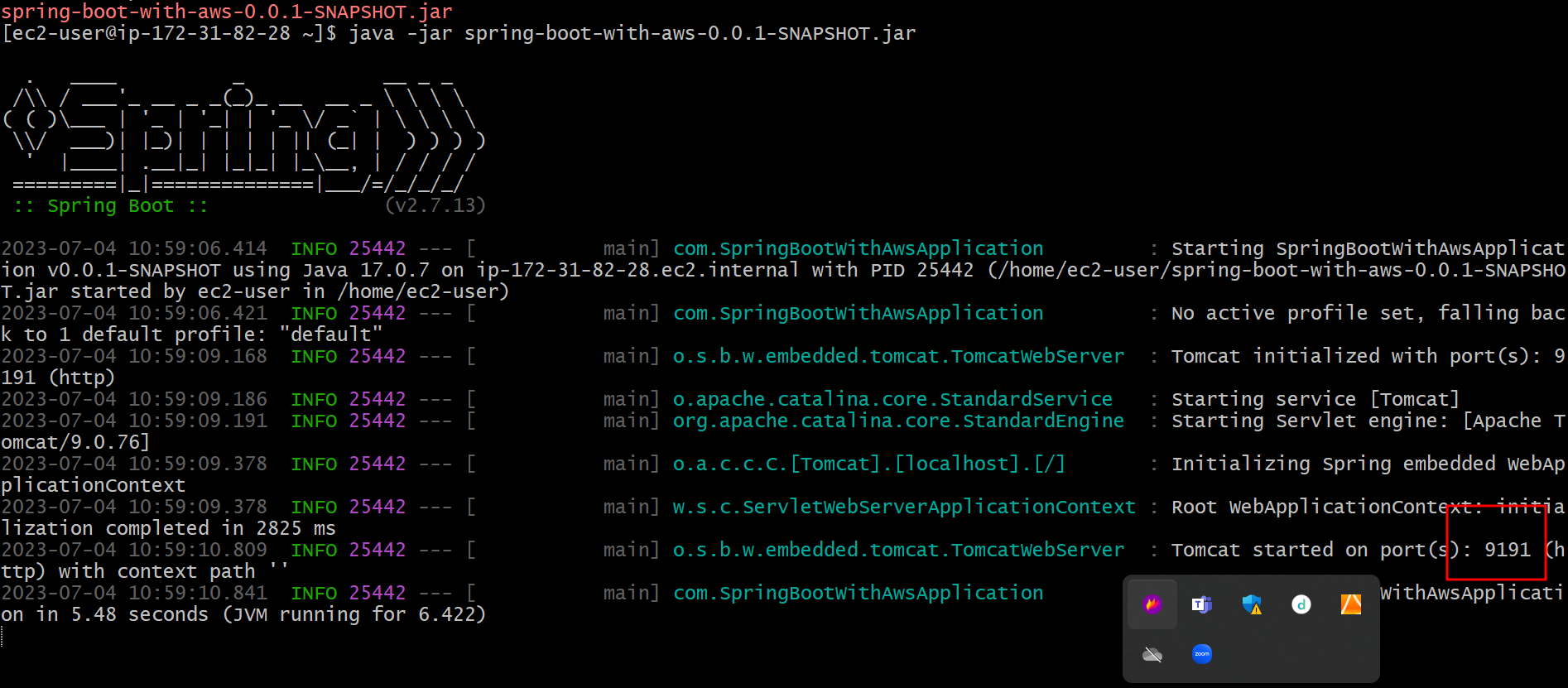
After installation please verify java installed or not.



Now you can run jar file using command as

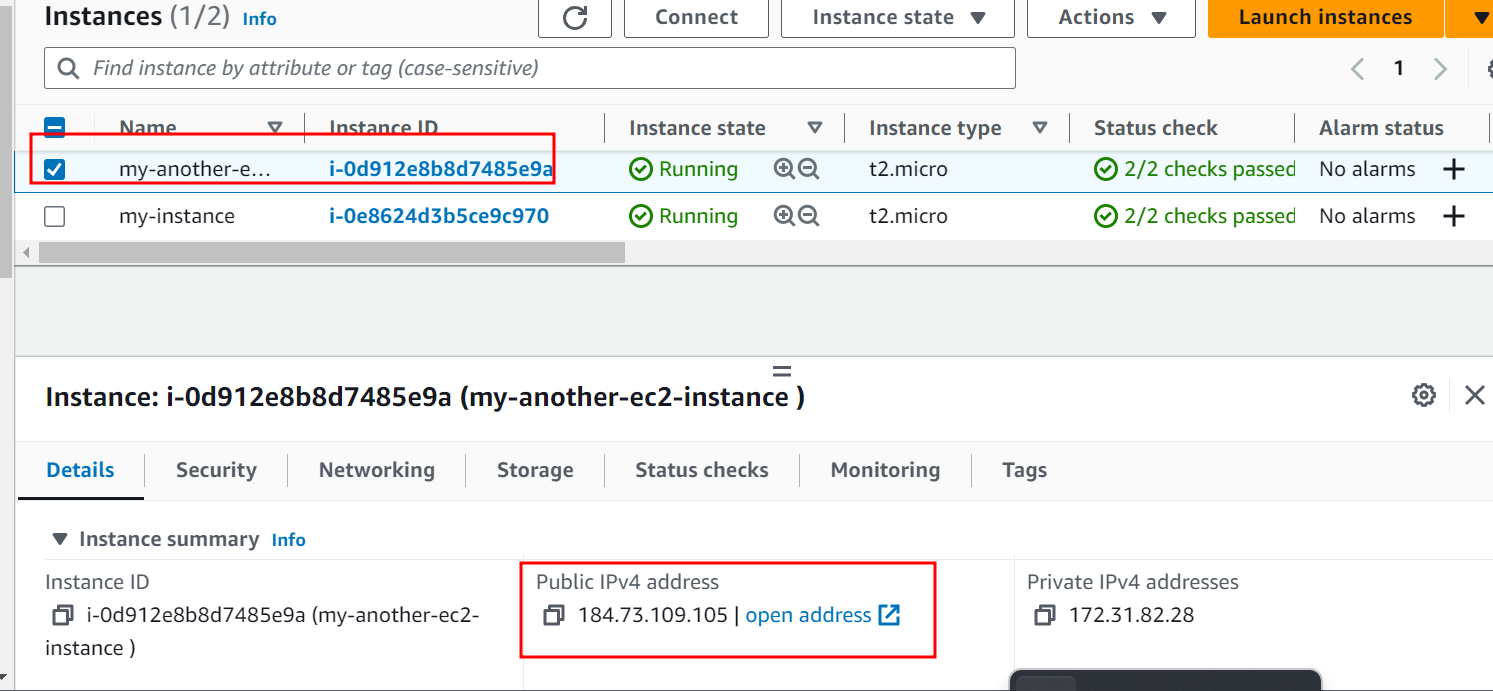
**java -jar jarfilename.jar**





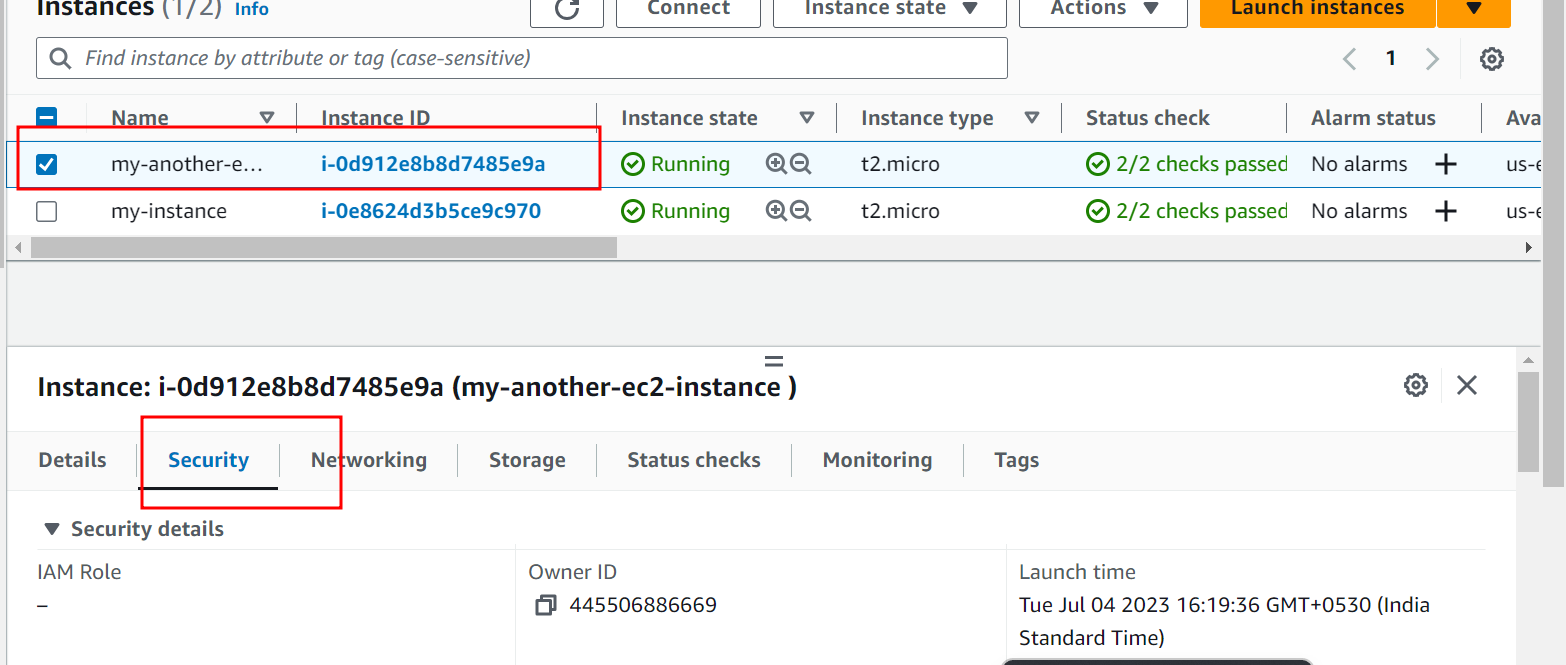
This jar file running on port number 9191

Every ES2 instance provide unique ip address to access the application.

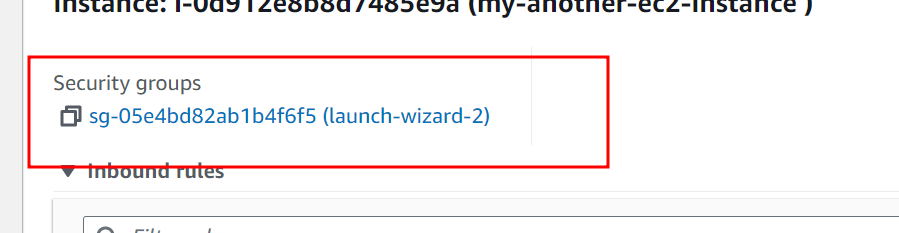


Now we need to open the 9191 port number

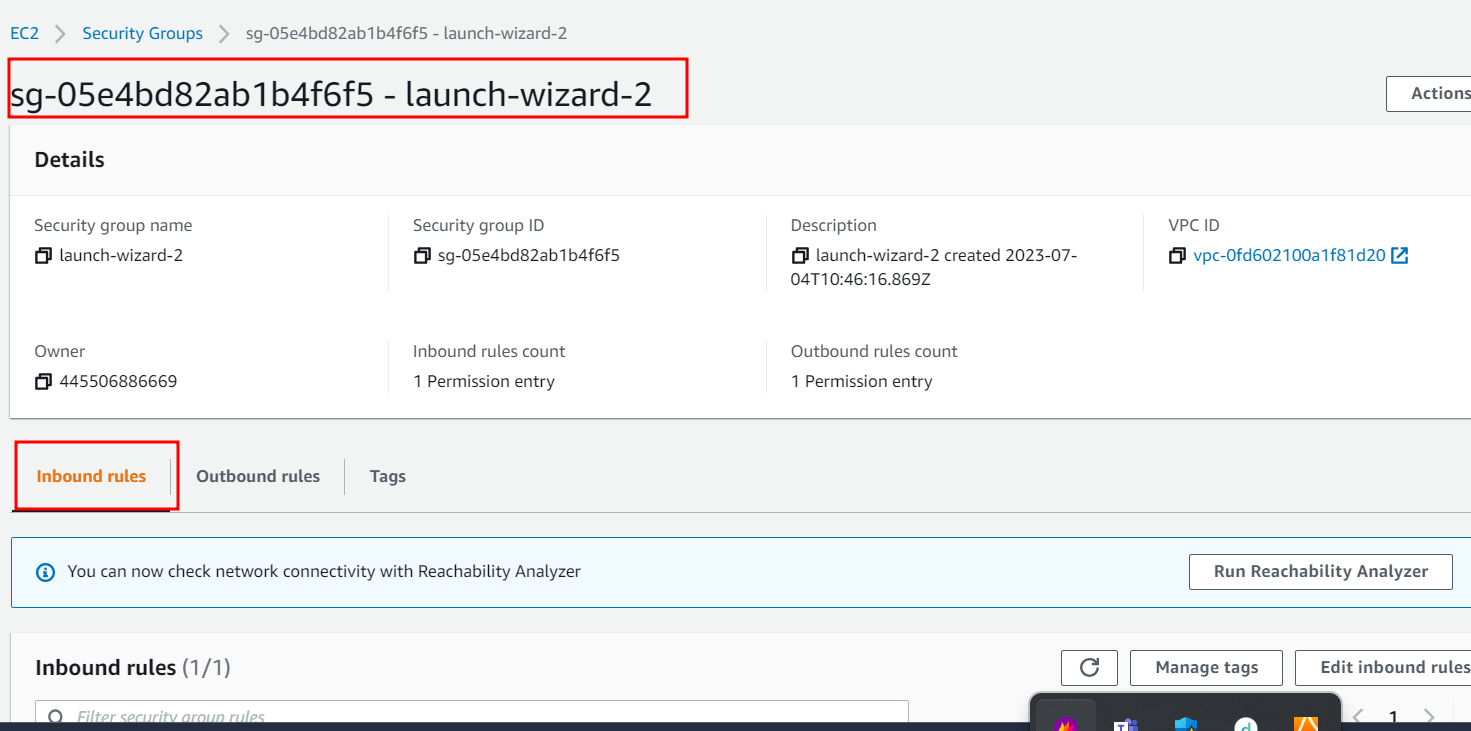
To access the spring boot project in ES2 instance.



In Security option plz search button security wizard option

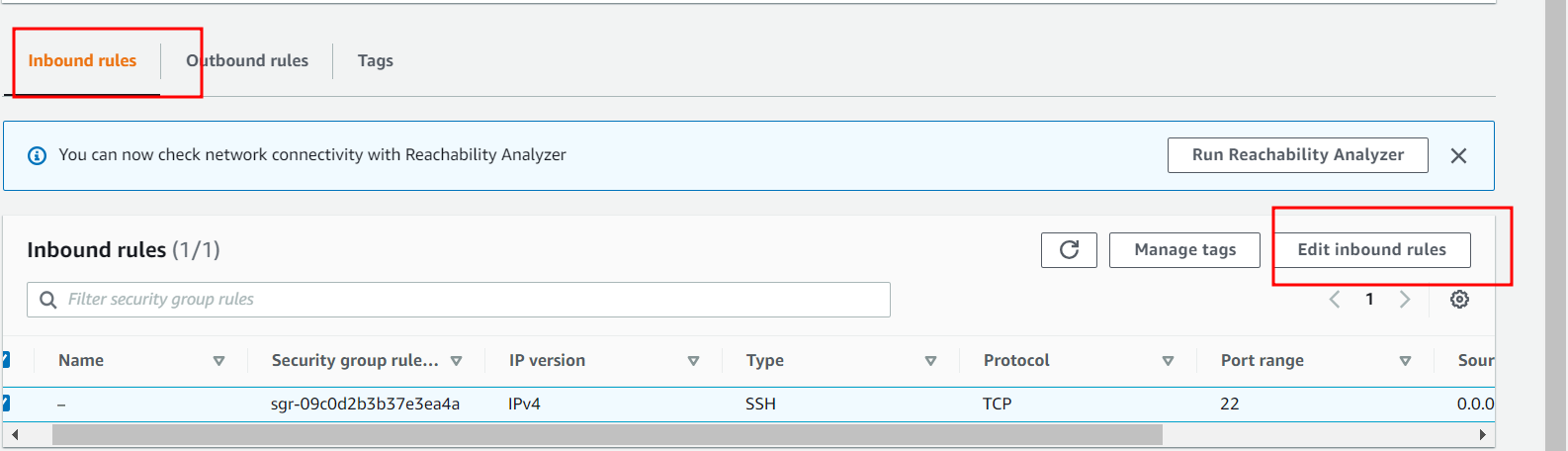


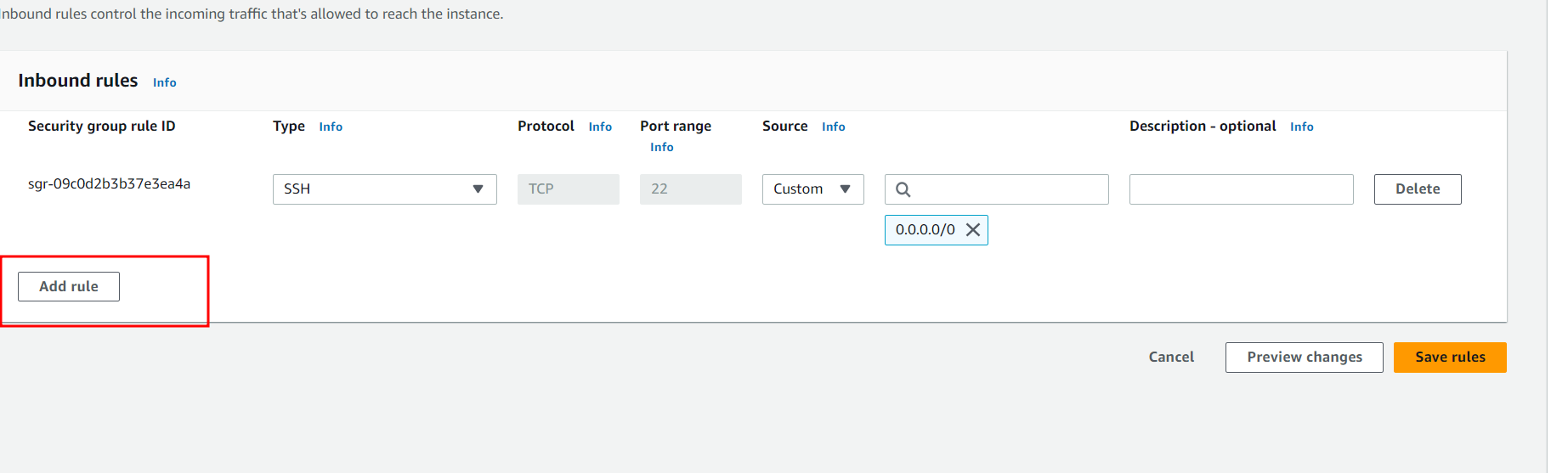
On click

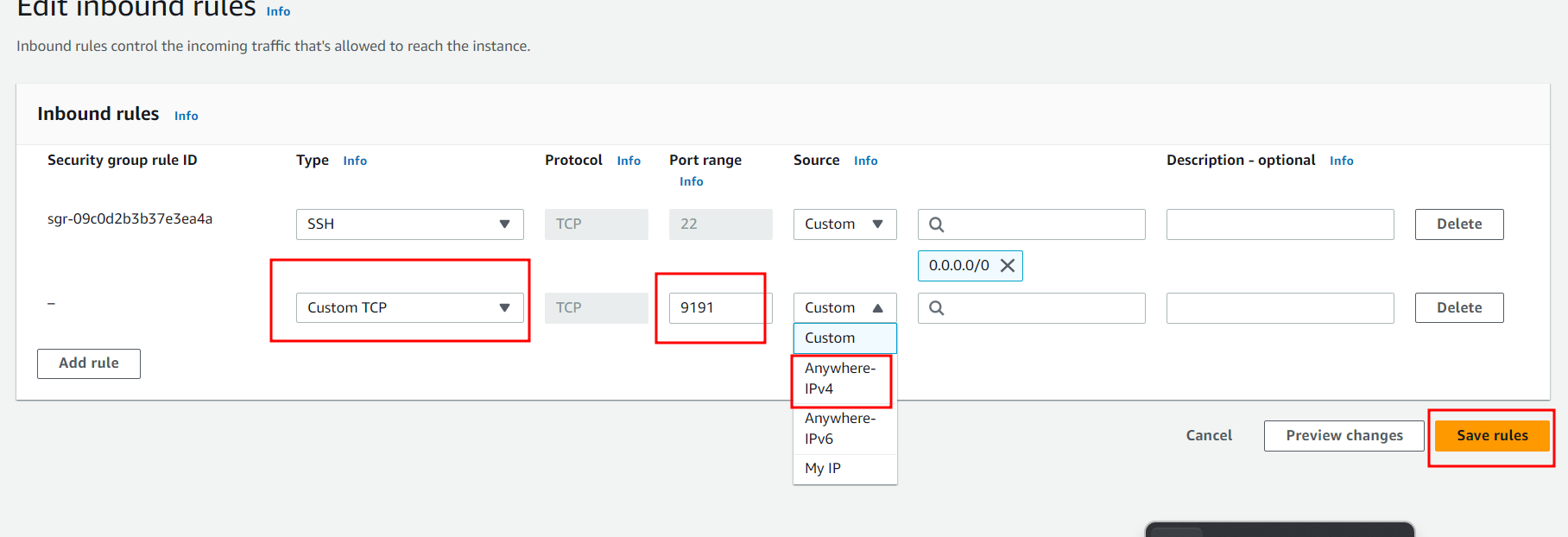


Click on inbond rules

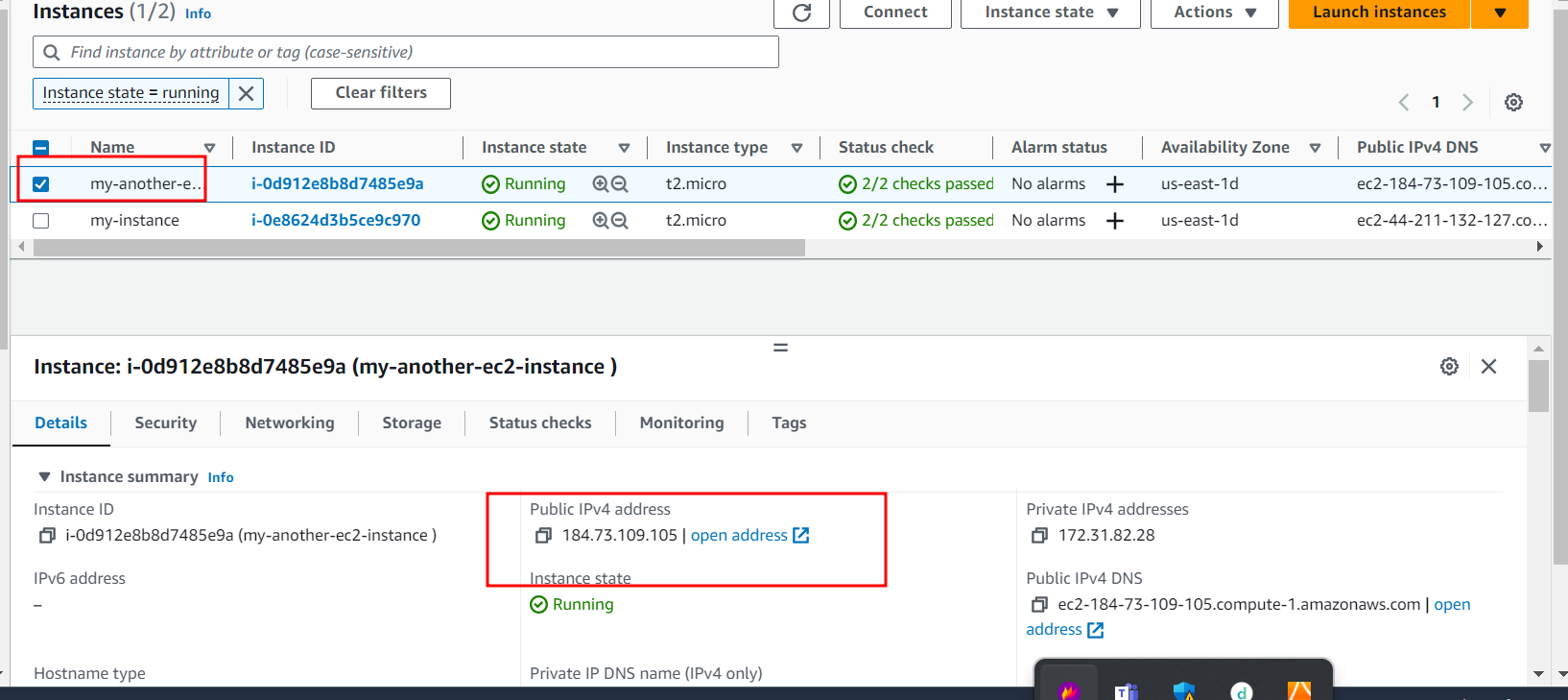
In inbound rules select edit inbound rules







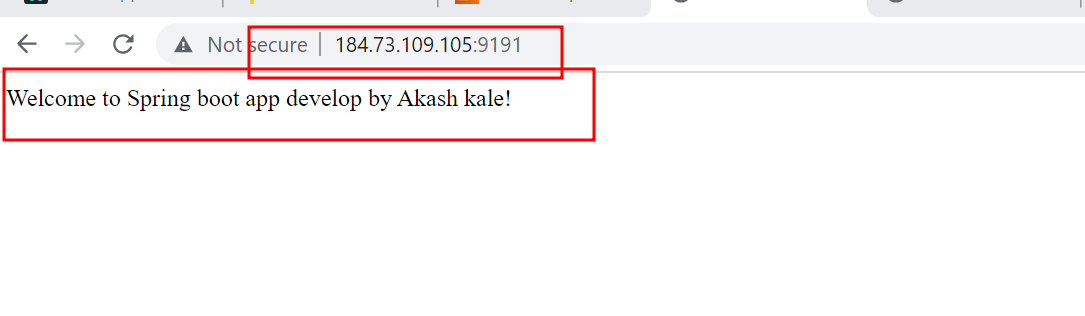
After saved rules successfully open the running instance



And find the IP Address of that specific instance

And open the browser

<http://IPAddress:portNumber>



**Make sure no https**

Deploy React JS Project in EC2 instance

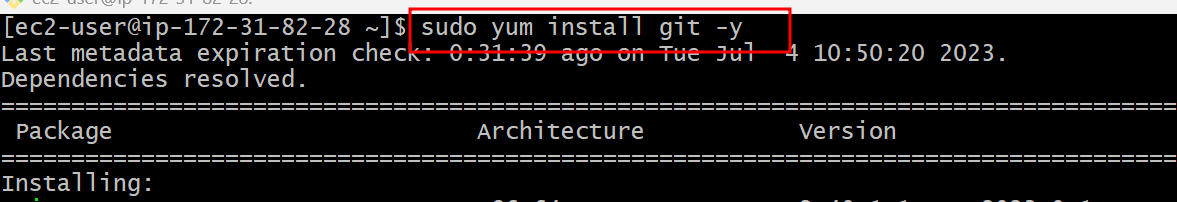
First create the react js project in local machine or virtual lab.

Do the Edit in App.js or do the coding depending upon application requirement.

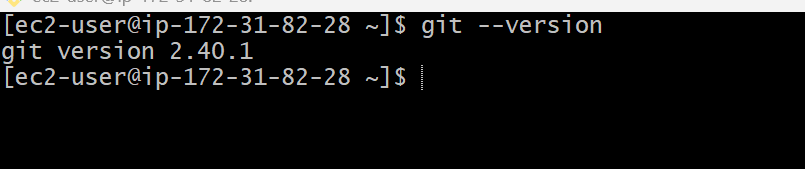
Then push the project in Git hub account.

Then in EC2 instance we need to install

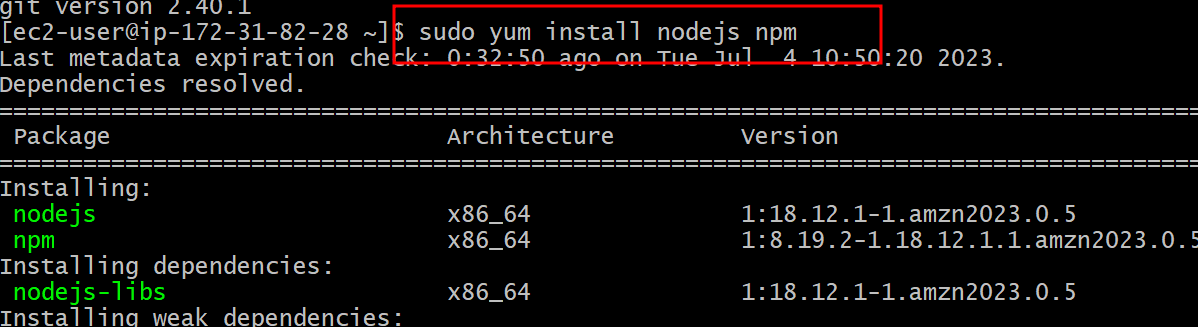
git and node js



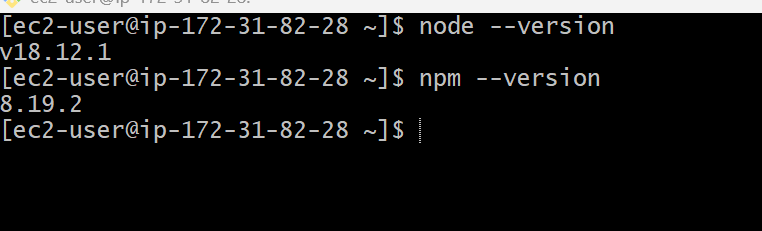
Then verify git installed or not



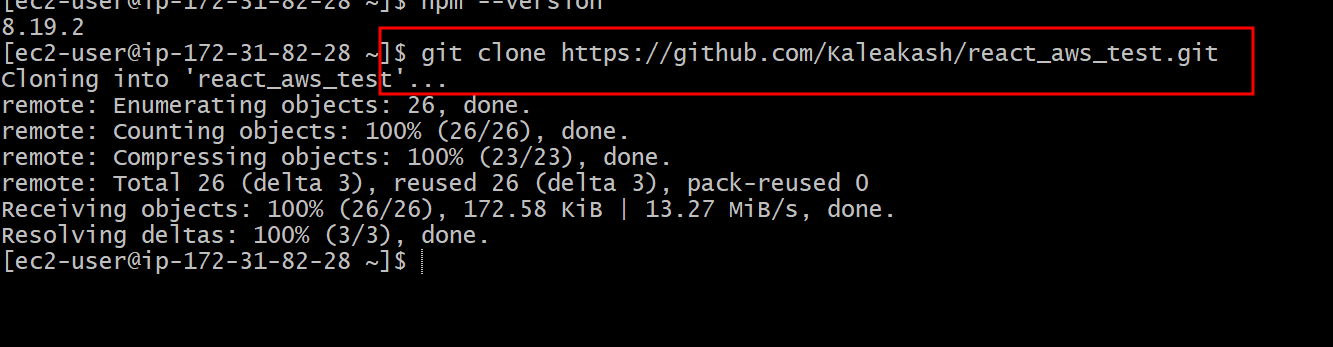
Now we will install node js and npm command



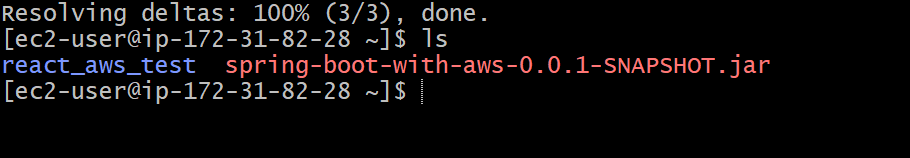
Then verify node and npm command



Using git clone the project from git hub account in EC2 terminal.

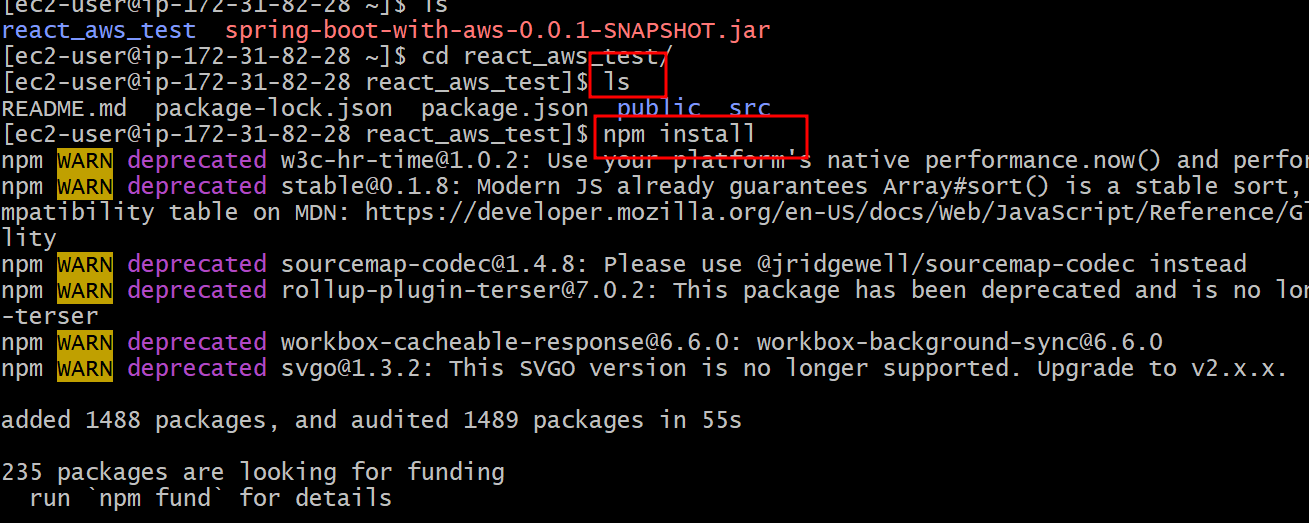


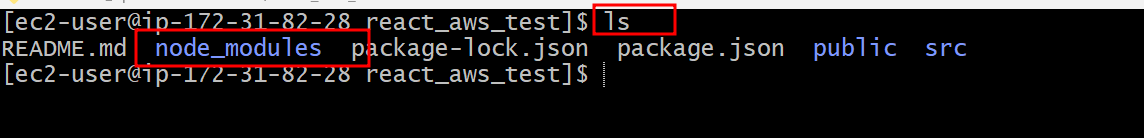
Using ls command please verify project present or not in EC2 instance.



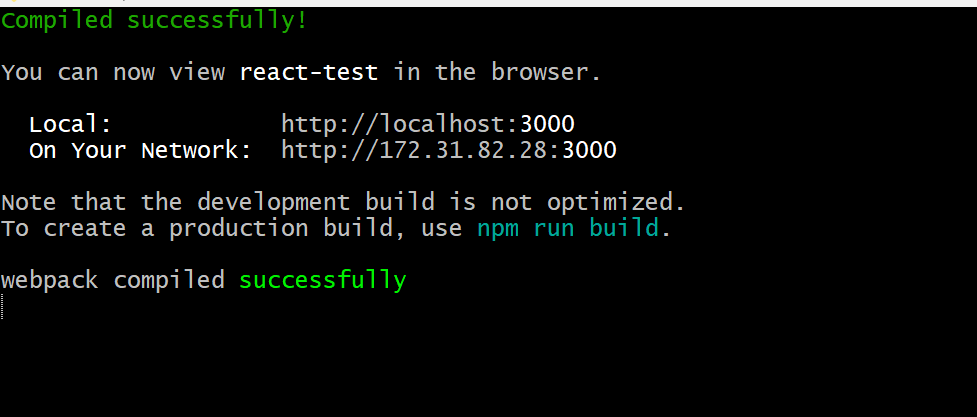
Now move inside a react js project

Then using npm install command install node\_model folder.



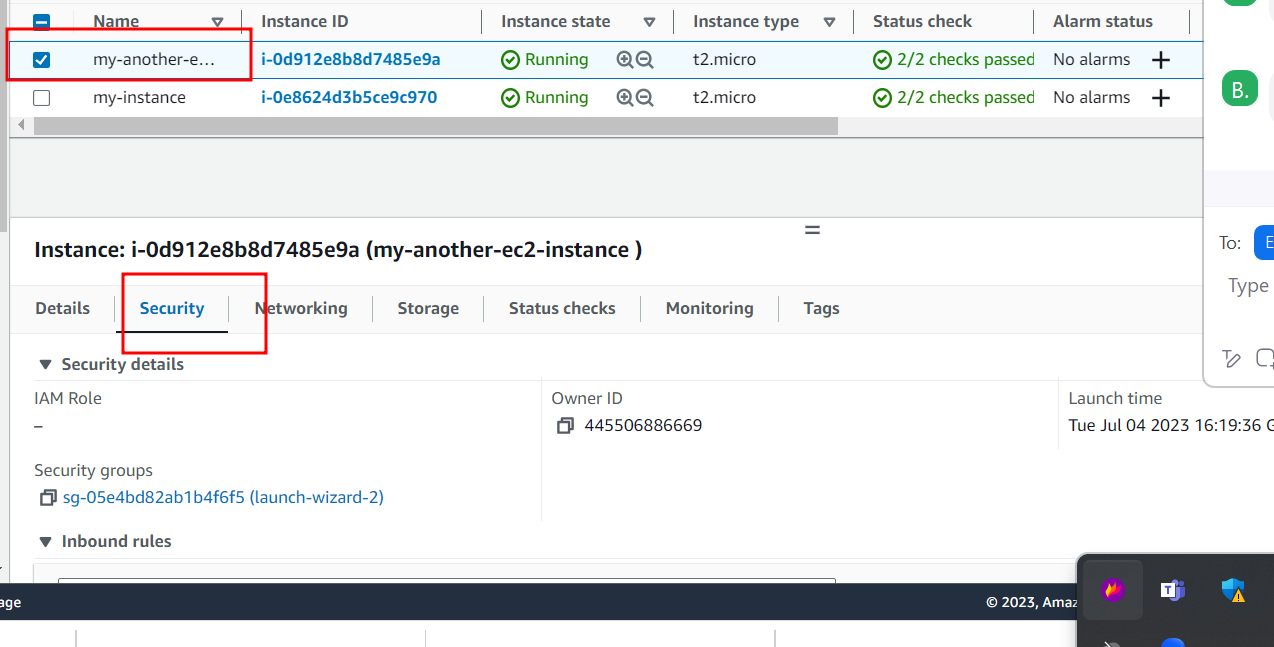


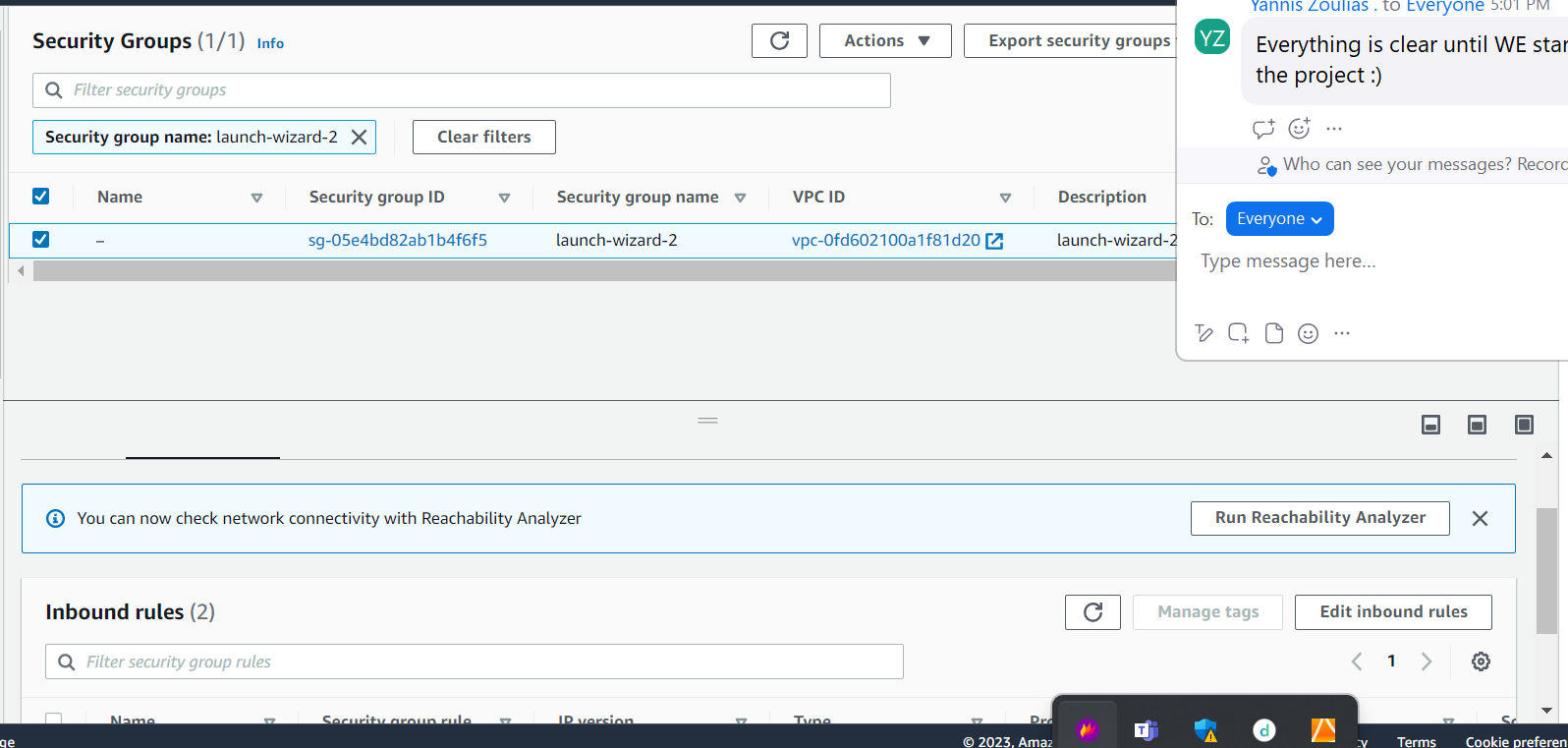
Then run **npm start**

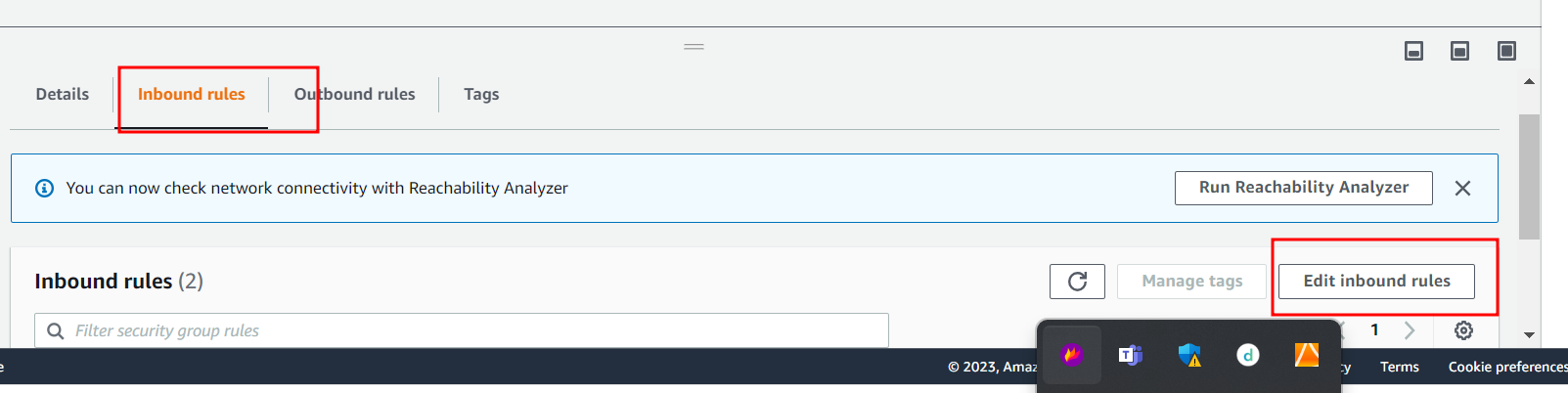


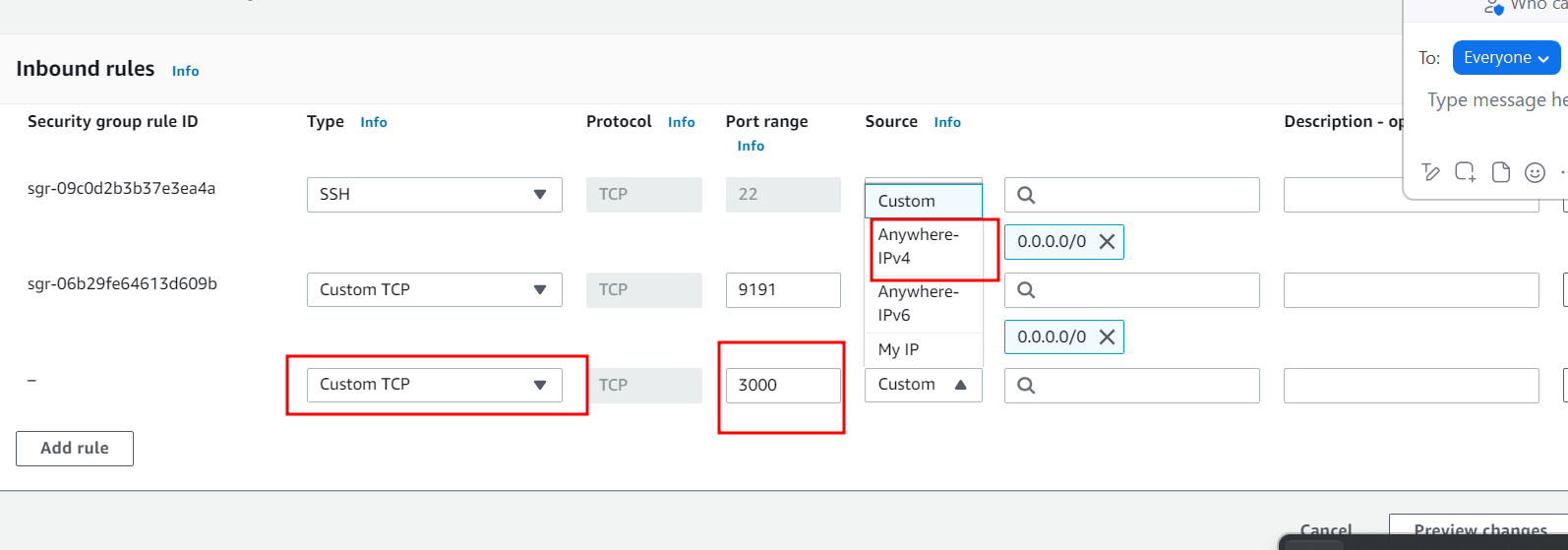
To access react js project we need to open 3000 port number.

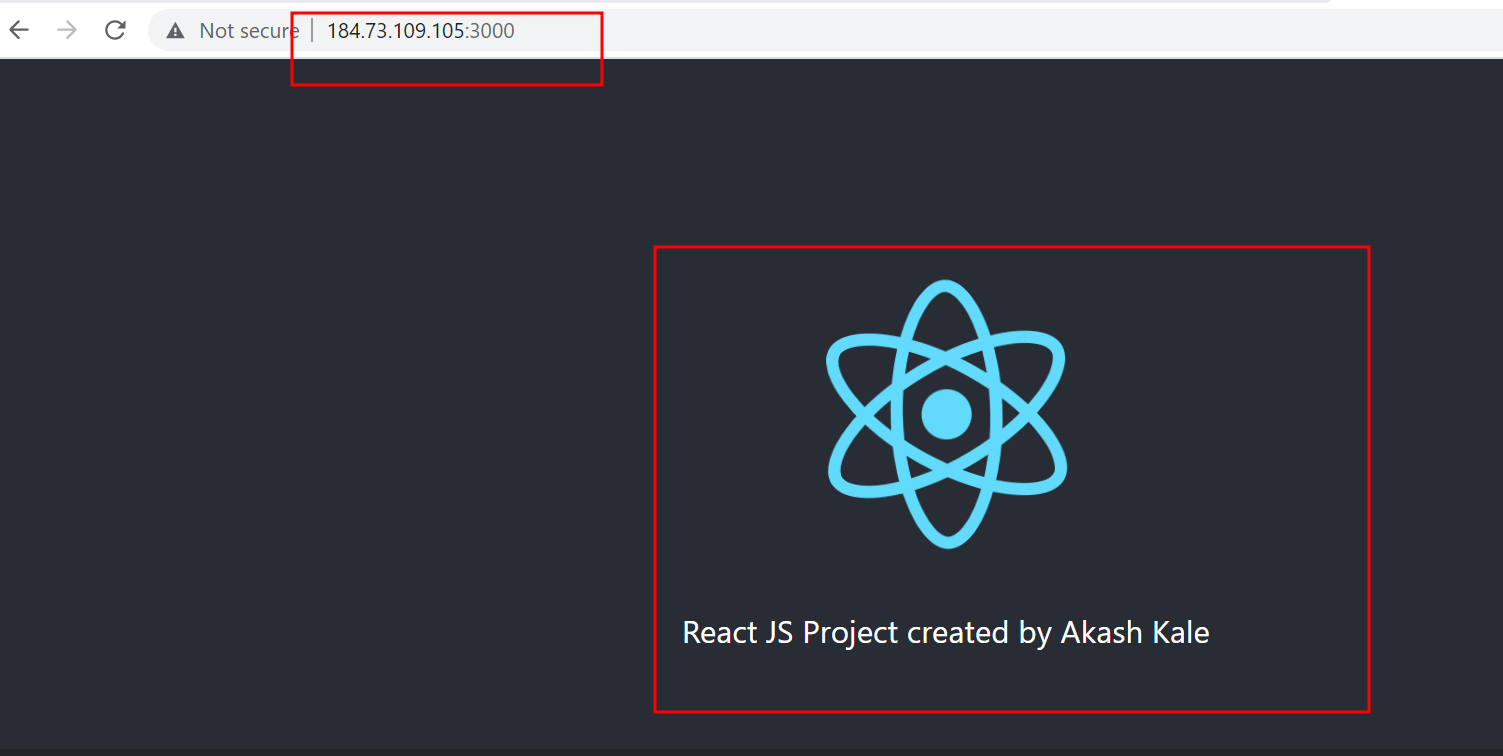
In security group.











**Phase 5 :**

**Day 8 : 07/05/2023**

Docker : Docker is an open source platform for developing, shipping and running an application.

Mostly is it known as a container manager.

Container : Container also know as engine. It encapsulate environment which run on top of very shallow level of abstraction providing by virtual machine.

Event Docker also known as Advanced OS Virtualization software platform that makes it easier to create, deploy and run an application in Docker Container.

If we want to run the application software develop in any language like Java, Python, Node or React JS or spring boot to run that application we need system software ie OS and in that OS we need all required software which help to run that application.

VMWare software :

Oracle virtual BOX.

With help of VM ware software we can run multiple OS on base machine.

My base machine : window 10 with RAM 16GB 1 TB

If we are planning run the Guest OS

Then we need to provide RAM 4GM ram and external memory 100gb,

If we wan to run 10 Guest OS

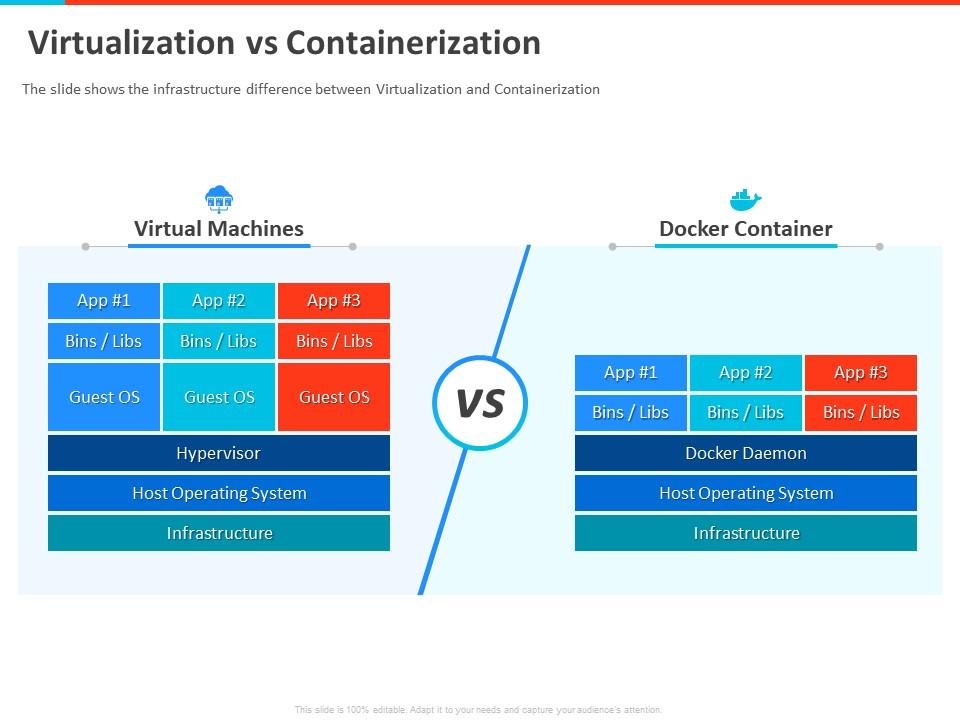
Then we need to provide for each guest os ram and external memory.

In Virtualization we need to share the resource from base machine to virtual machine.

Using Docker we can create the Containerization application.

Virtualization is an abstract version of physical machine OS.

Containerization in an abstraction version of an application.



Please login to Virtual lab

Open terminal

sudo docker --version

sudo docker images this command is use to check current images present in base machine

sudo docker pull hello-world : this command is use to pull the hello-world images from container hub repository.

sudo docker run imageName/ImageId

sudo docker run hello-world : this command is use to run the image

please create docker hub account with personal emailed and password.

Busy box : it is a small tiny 5mb unix utilities OS.

sudo docker pull busybox pull busy box images.

Docker Container : This is a running process or instance of images.

A docker container contains everything which help to run our application in any environment with help of docker engine.

Program in execution

Docker Image : The file system and configuration details of our application which are used to create the container as well as to run the container. Image is use to run the application with help of container.

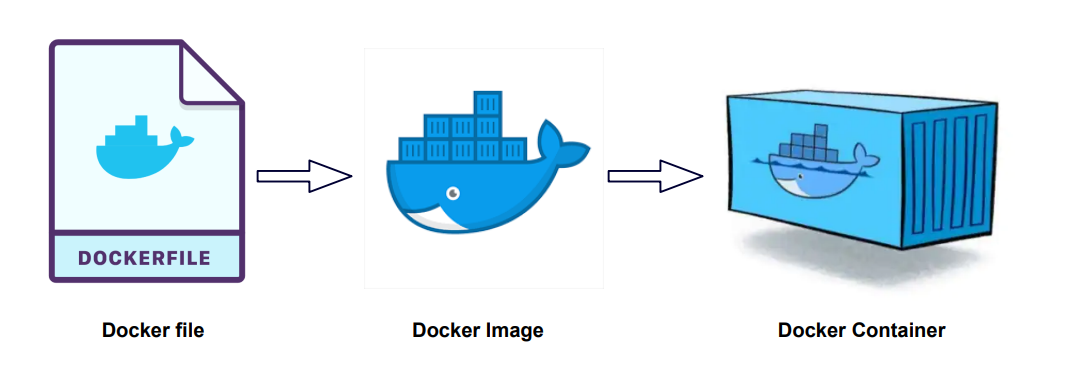
Class Demo {

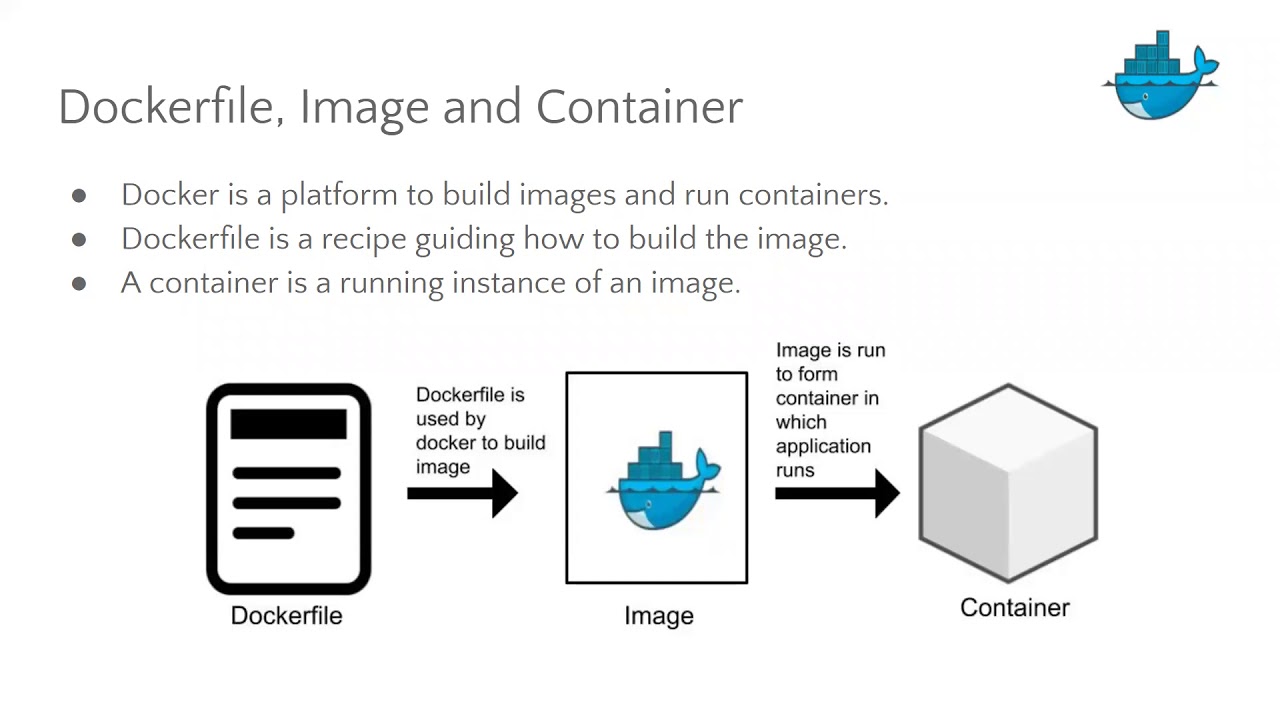
}

Demo obj = new Demo();

Docker file : A docker file is a blue print or set of instruction which help to create the images.

Demo.java





We will create Docker Images

Create one folder as

DockerImages

* WelcomeImages

To create the image we need to create the docker file with name Dockerfile without extension.

Dockerfile

FROM busybox:latest

CMD ["echo","Welcome to Docker Concept"]

Open the terminal in that location where Dockerfile present

To create the image

sudo docker build -t my-busybox . -f Dockerfile

sudo docker images

docker run my-busybox

creating image to run simple java program

Demo.java

public class Demo {

    public static void main(String args[]){

        System.out.println("Welcome to Java Program runnign through Docker");

    }

}

Dokerfile

FROM openjdk:11

COPY Demo.java .

RUN javac Demo.java

CMD ["java","Demo"]

sudo docker build -t my-java . -f Dockerfile

sudo docker run my-java

creating image to run spring boot application

copy spring boot created jar file in folder

Dockerfile

FROM openjdk:11

COPY spring-boot-with-aws-0.0.1-SNAPSHOT.jar .

CMD ["java","-jar","spring-boot-with-aws-0.0.1-SNAPSHOT.jar"]

Create the images using command as

sudo docker build -t my-spring . -f Dockerfile

if images is responsible to run the web application on server then we need to run as

docker run -d -p 9191:9191 imageName;

right side port number are actual port number

left side port number are expose port number can be same or different.

docker run -d -p 9191:9191 my-spring

docker run -d -p 9292:9191 my-spring

Creating Image for React JS Application

create one folder as creating image for react js project

create-react-app react-with-docker

After project created open App.js file and write your name

After coding in react js project

Terminate the application means stop the react js project

npm run build

after build command it will create build folder which contains

all build file of react js project

react js internally provided web server which run on default port number 3000.

One we build the project we need to take external server to deploy project.

Docker hub provided lot of open source server images which help to run

Angular or html/css/js or react js project.

Nginx server

Nginx default port number is 80

Create Dockerfile in react js project

Dockerfile

FROM nginx

COPY /build/ /usr/share/nginx/html

Then create the image

**sudo docker build -t my-react . -f Dockerfile**

**docker run -d -p 80:80 my-react**

we will publish this image in docker hub account

first run the command as

docker login

username/emailed : dockerhubaccountemailid

password : password of dockerhub accont

create the tag for local image before push this image in docker hub registry

**docker tag my-react akashkale/my-react:1.0**

now we can push the image

docker push akashkale/my-react:1.0

if you want to pull my image

**sudo docker pull akashkale/my-react:1.0**

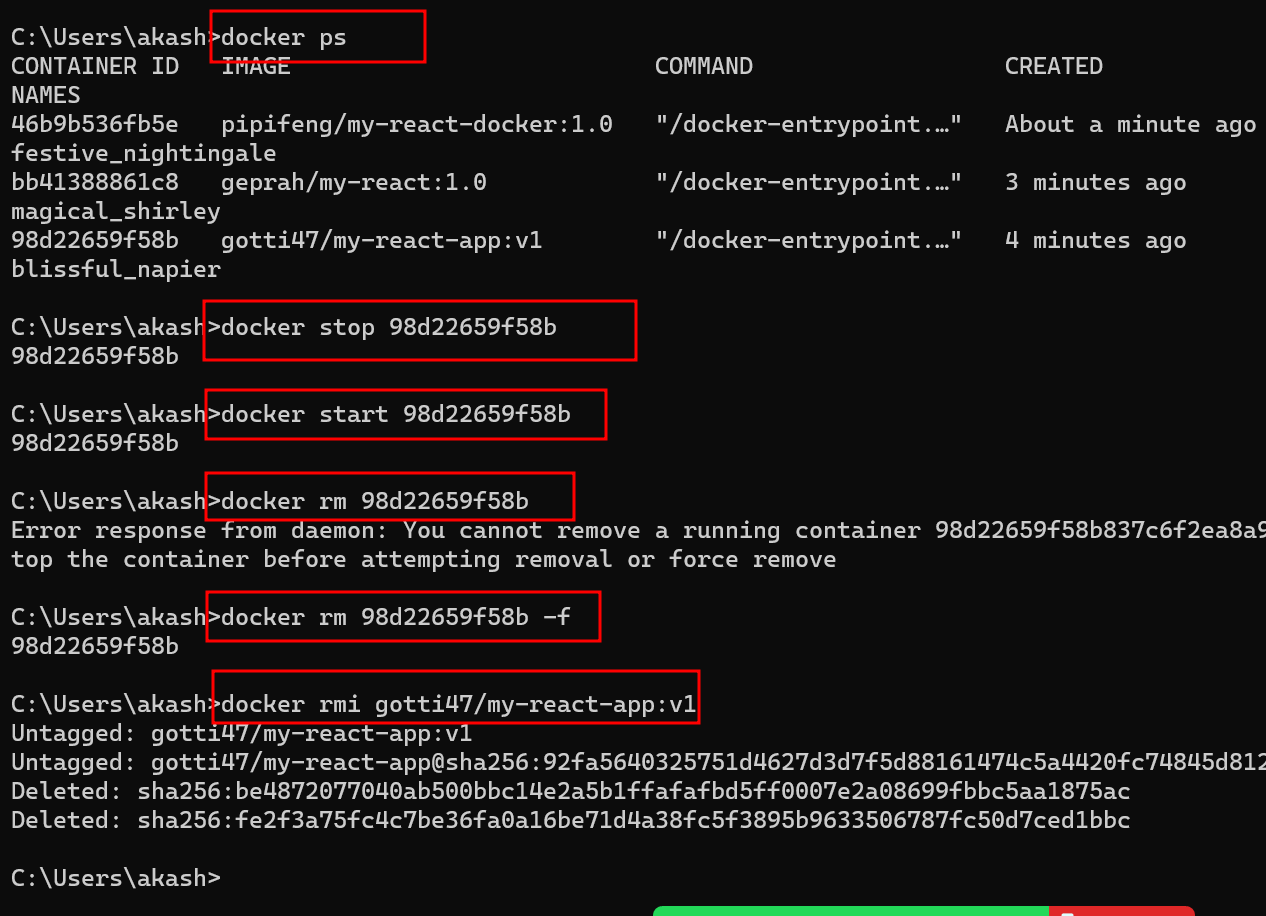
docker run -d -p 81:80 imageName/imageId

docker run -d -p 82:80 imageName/imageId

docker run -d -p 83:80 imageName/imageId

**Day 9**

**07/06/2023**



docker ps it will display only running container

docker ps -a it will display all container doesn’t matter they are running or stopped mode.

CI and CD tools

Continuous integration and Continuous delivery or deployment.

Dev1

Login page

Dev2 Remote Repository ) 🡨---CI/CD tool 🡪other team or production server.

App page

Dev3 build the project

Feedback page may be success

May be failure

Post build we will send

Notification

Jenkin is an open source light weighted java based, plugin based CI and CD tool.

This tool is use build the application like compile, run, test, creating jar or war files.

To do the setup for Jenkin

1. We need to download war file . In local machine we need web server like tomcat to run the war file.
2. Download Jenkin software with respective OS. Like window or linux or mac.
3. We can pull Jenkin image to run that image using docker.

<http://localhost:8080>

if we run Jenkin Image using Docker in EC2 instance then we can run Jenkin application using

<http://ipAddress:8080>

EC2 instance :

**sudo yum install java**

**sudo yum install docker**

**sudo docker --version**

This command execute in Virtual Lab or EC2 instance

Which pull jenkin image and run Jenkin software on port numbere 8080

sudo docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk11

Window User

docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk11

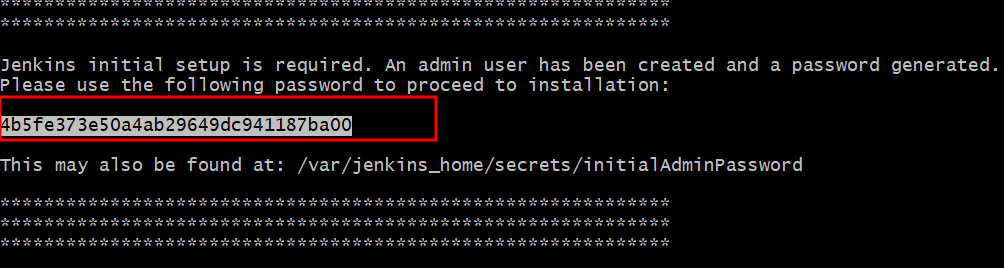
plz open Jenkin on

<http://IpAddress:8080>

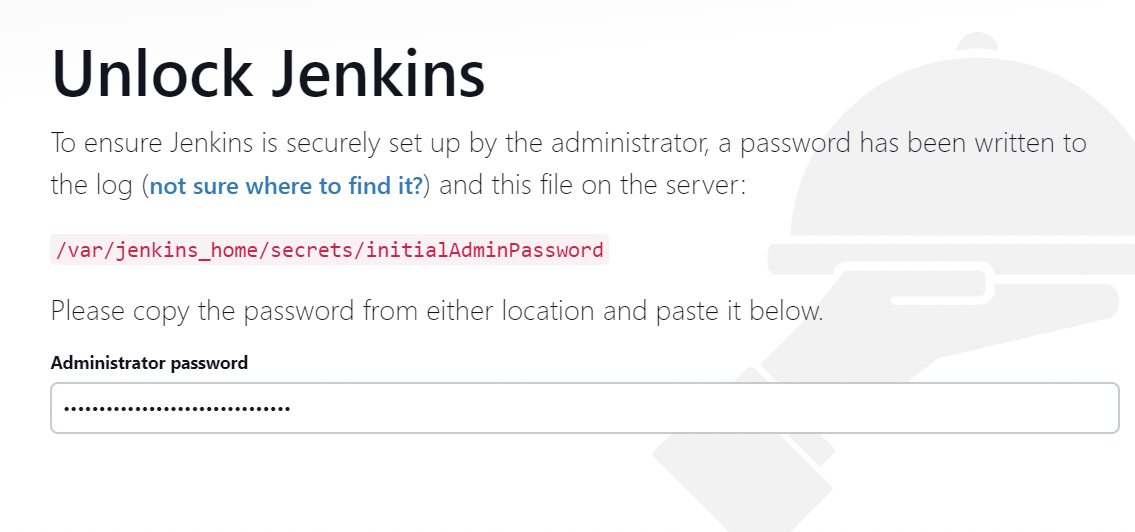
or

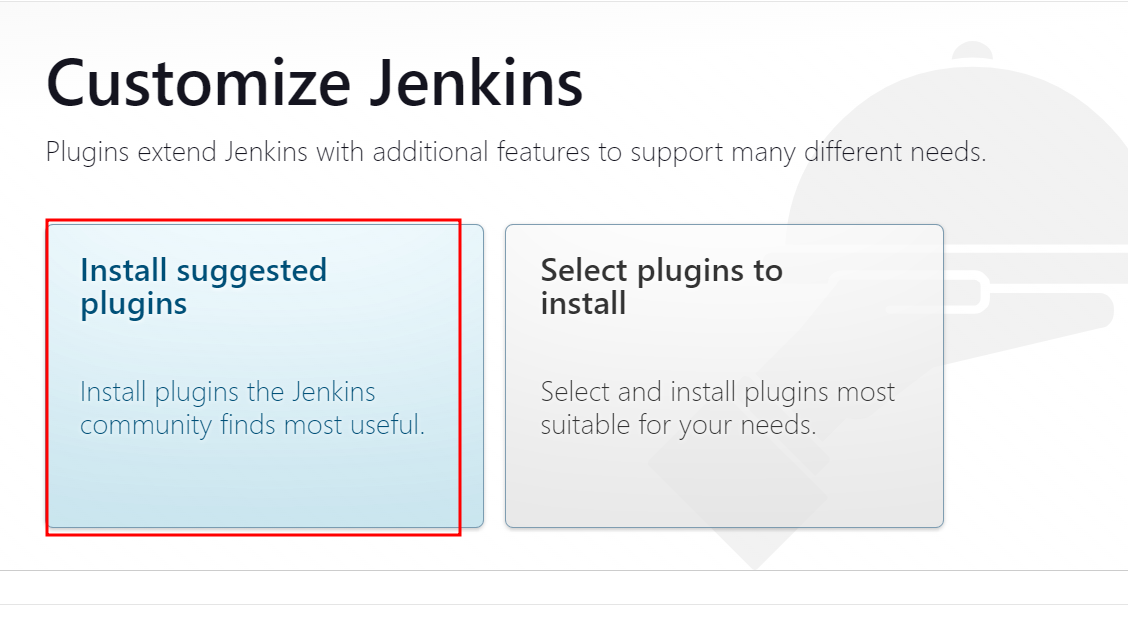
<http://localhost:8080>

it will ask the password



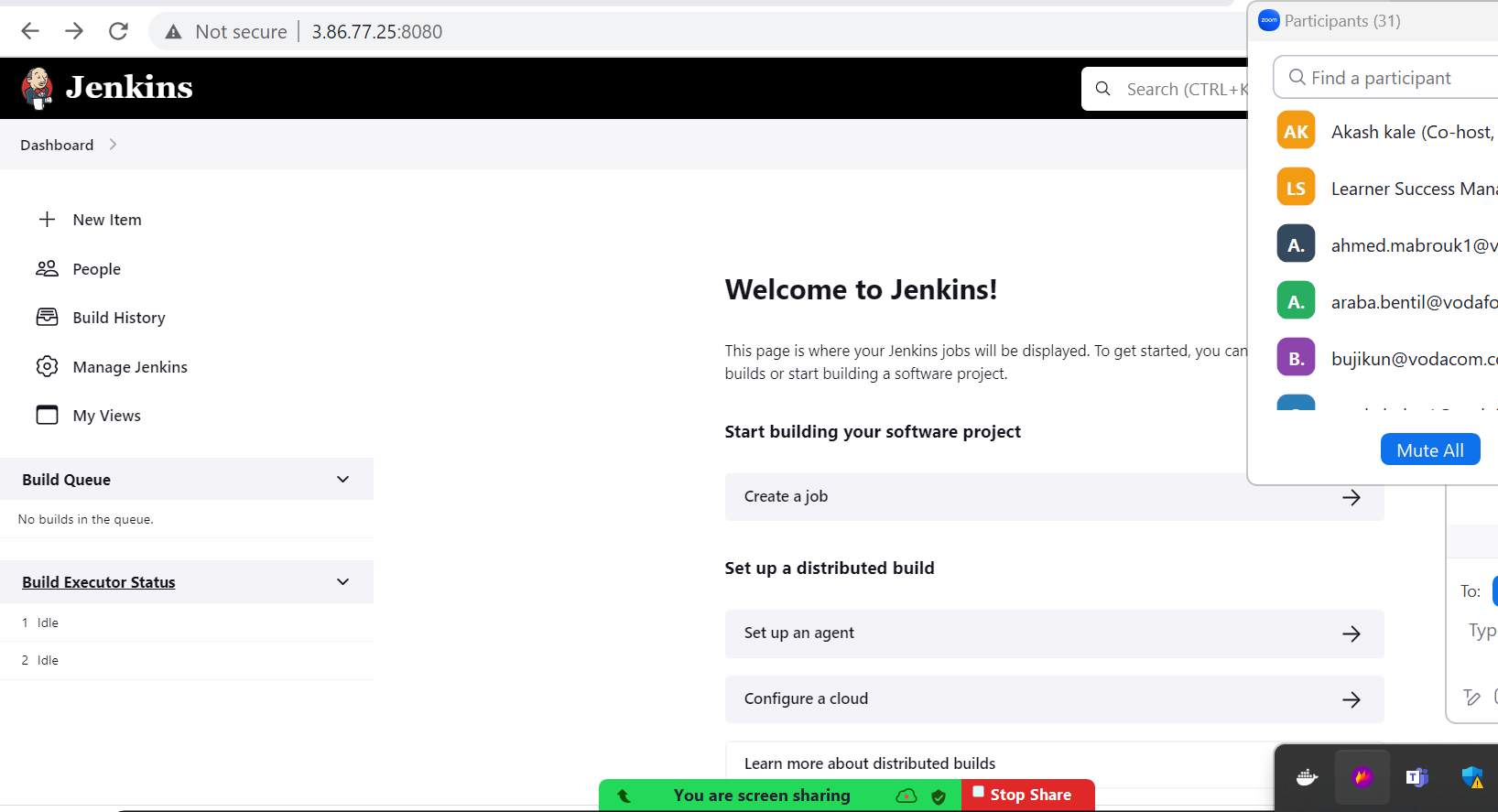
Plz copy password





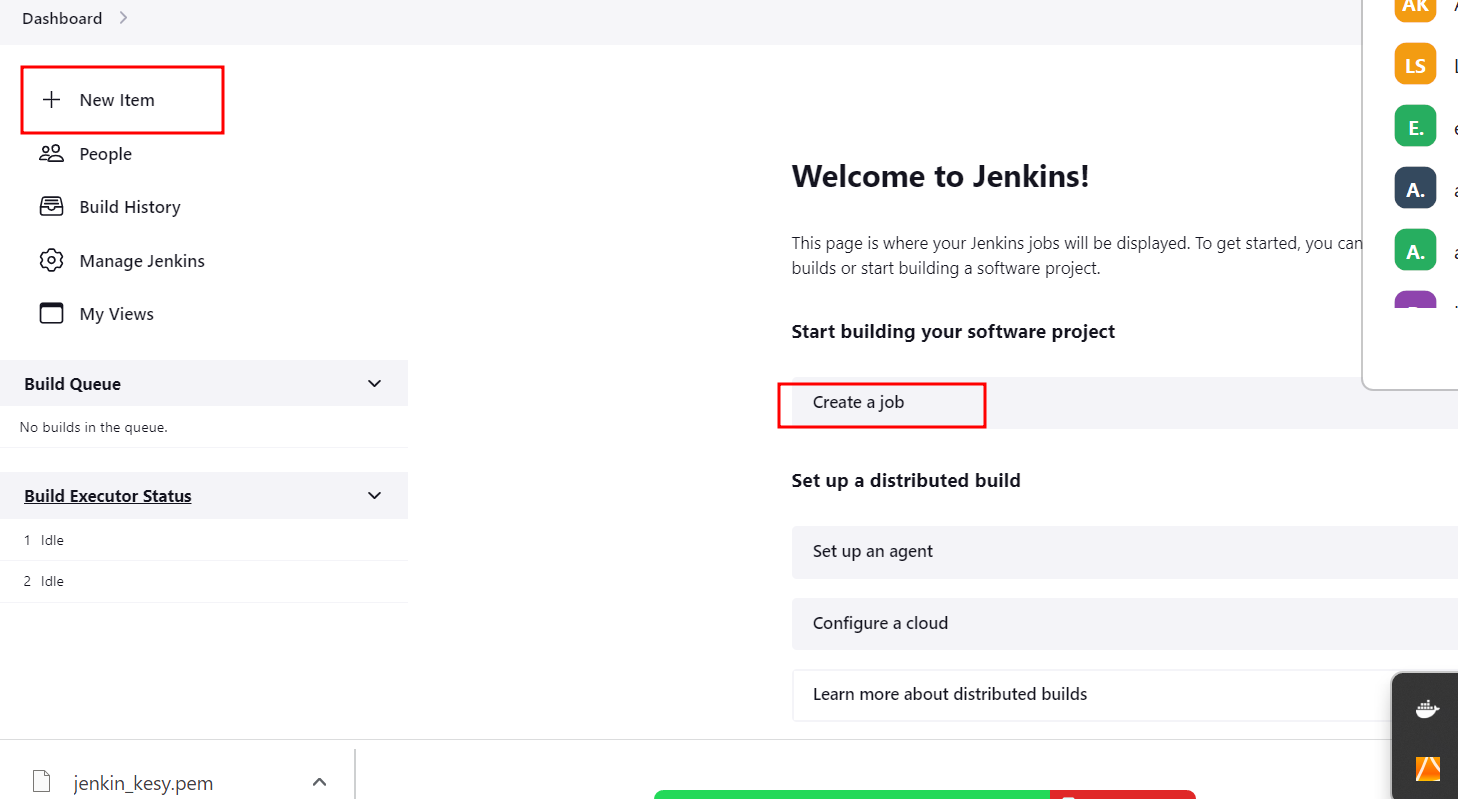
Plz instal suggested plugin

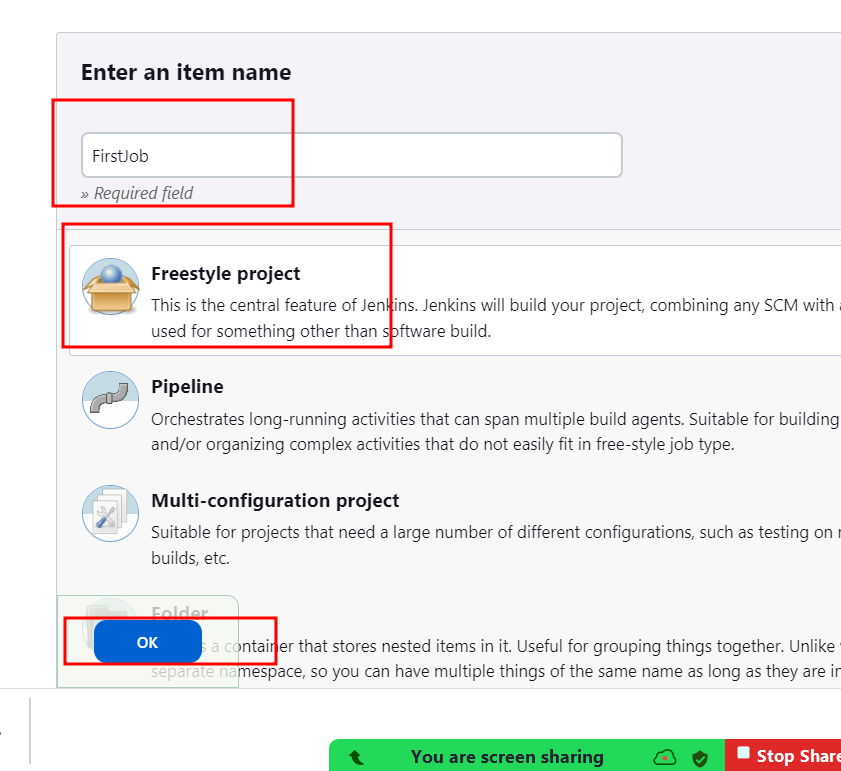
After installation please Create Jenkin with user name and password



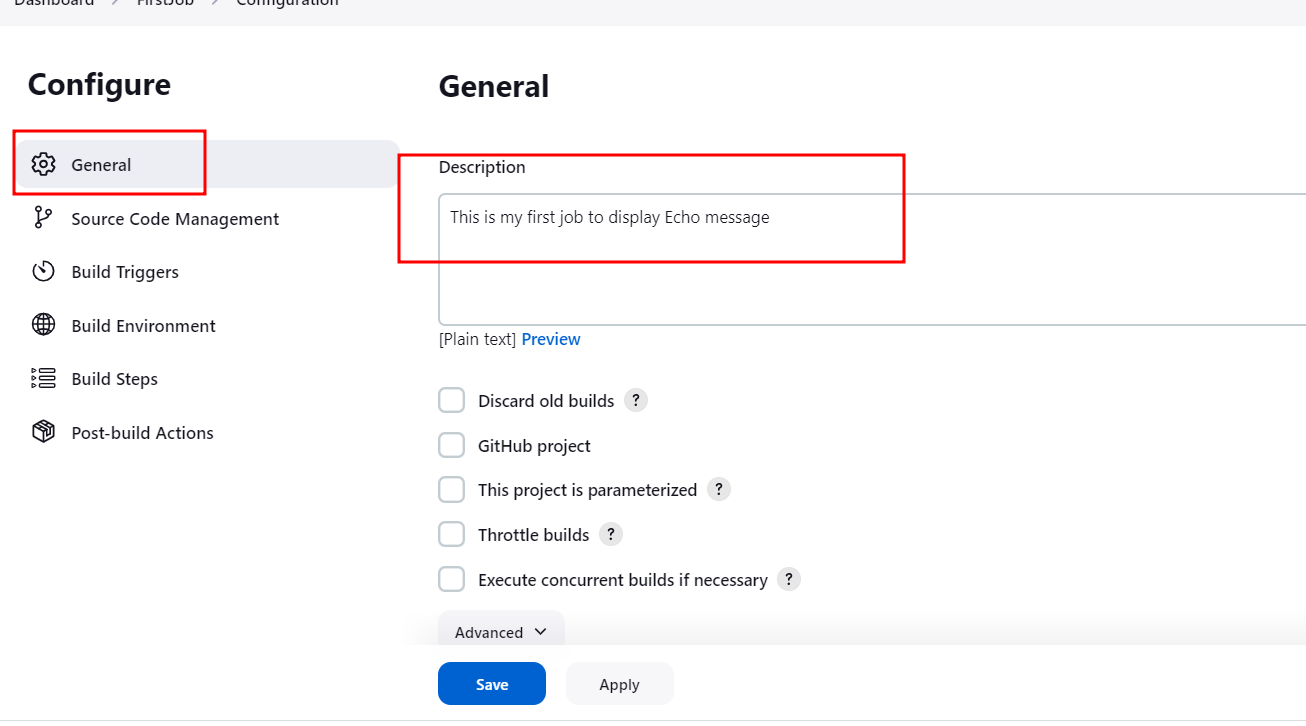
Then move inside a Jenkin dashboard.

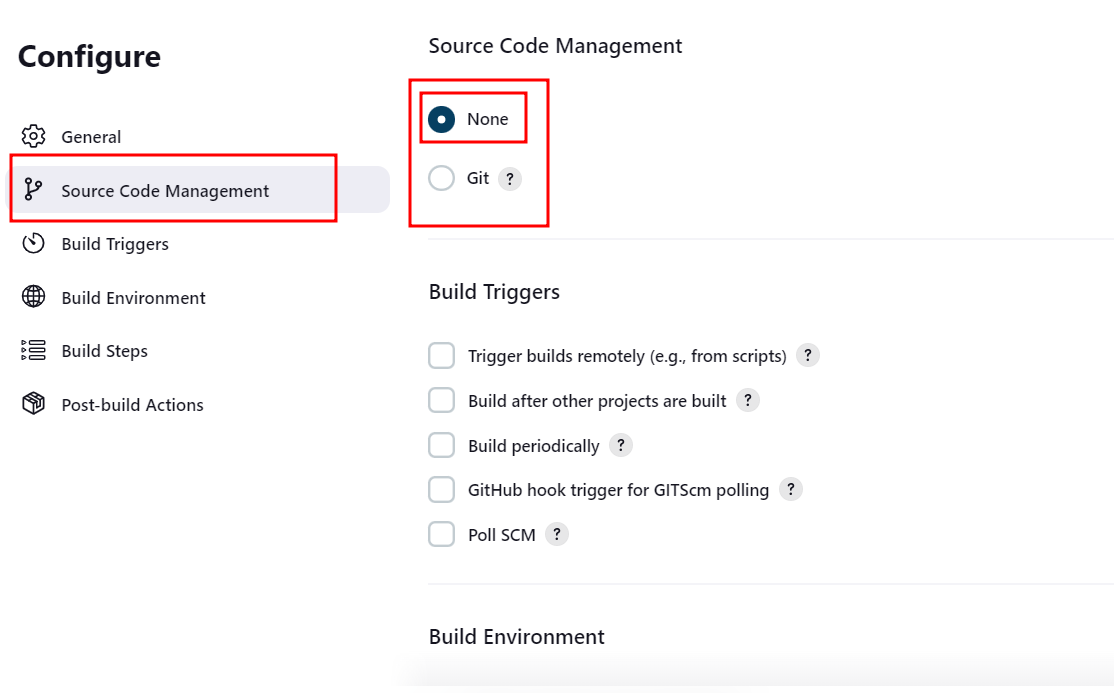
Now we need to create Jenkin Jobs.



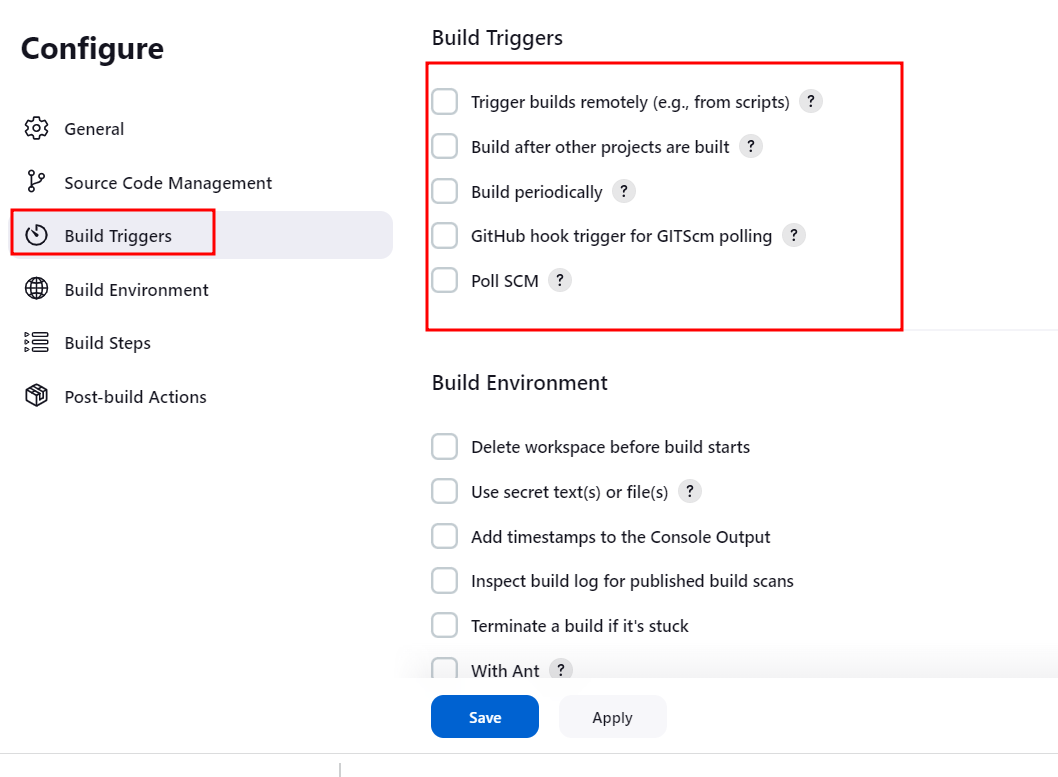


General Information



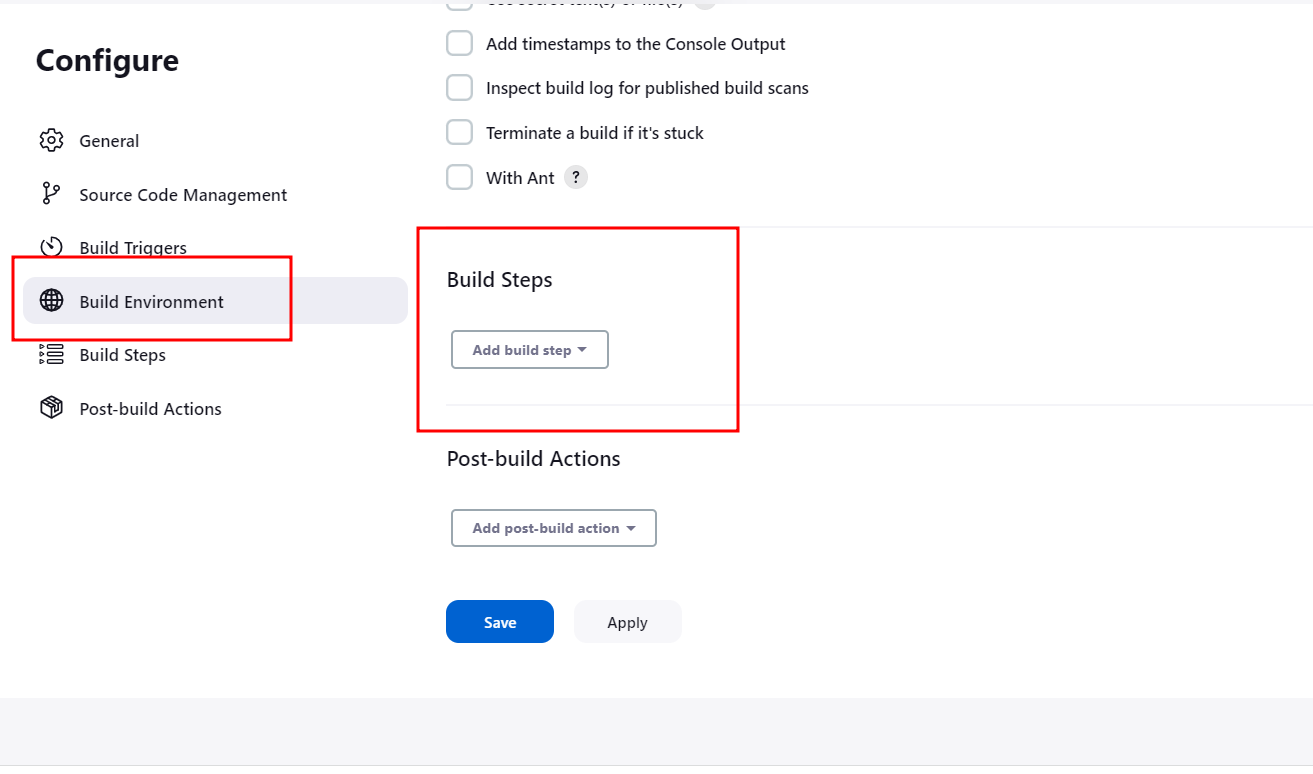


Trigger

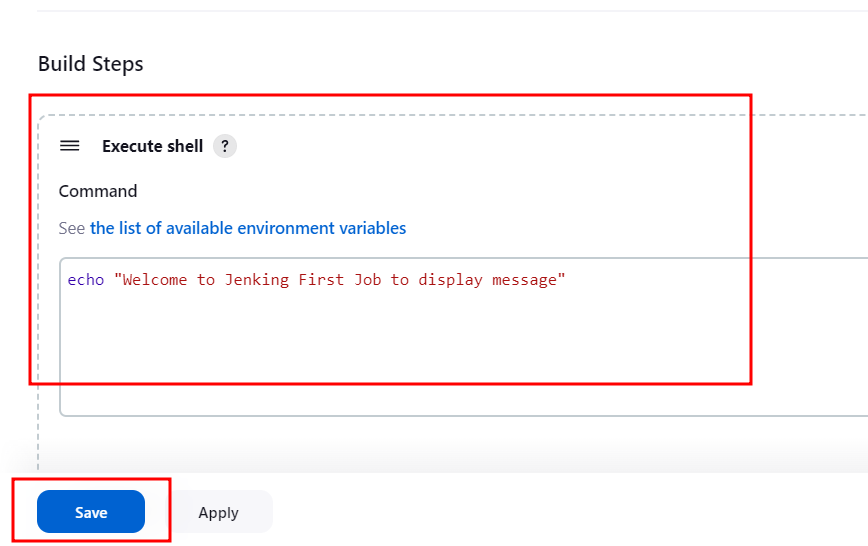


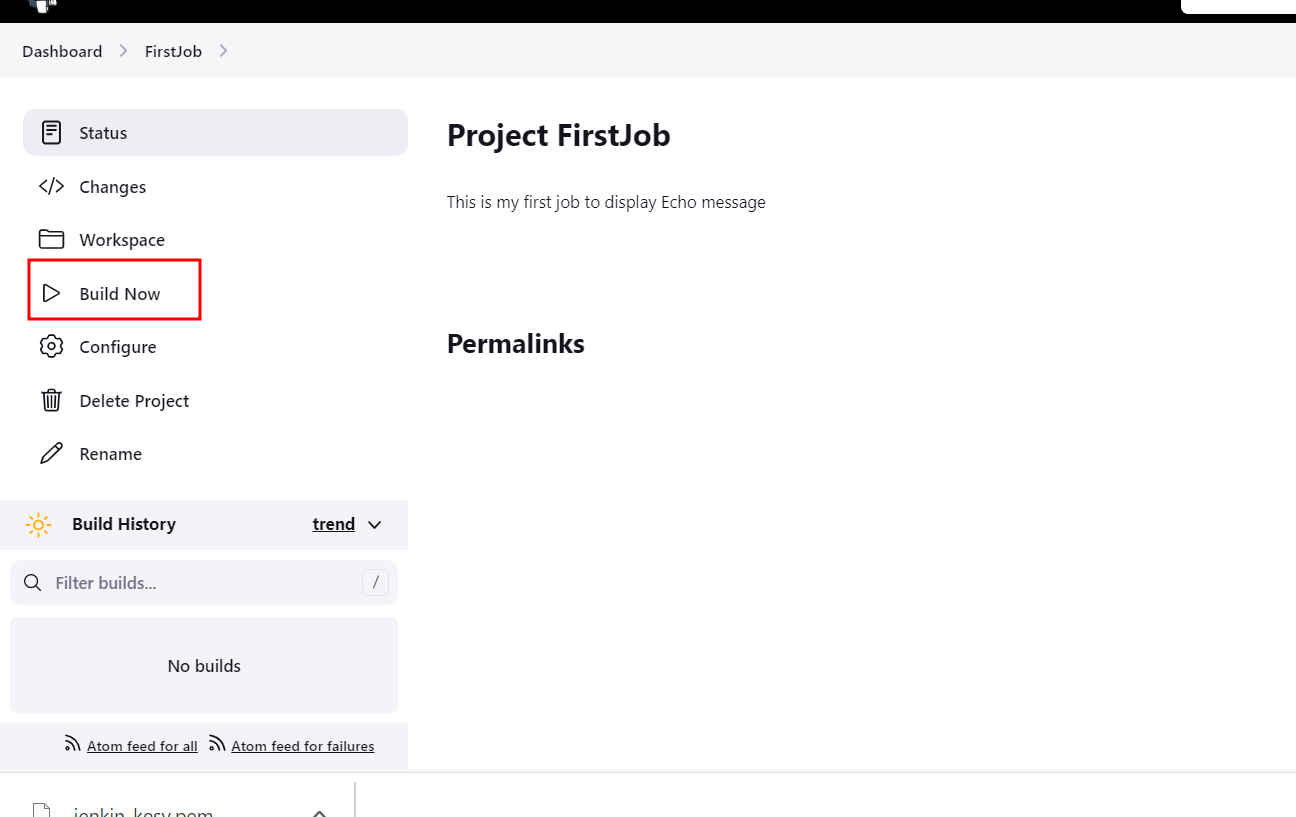
Build Environment

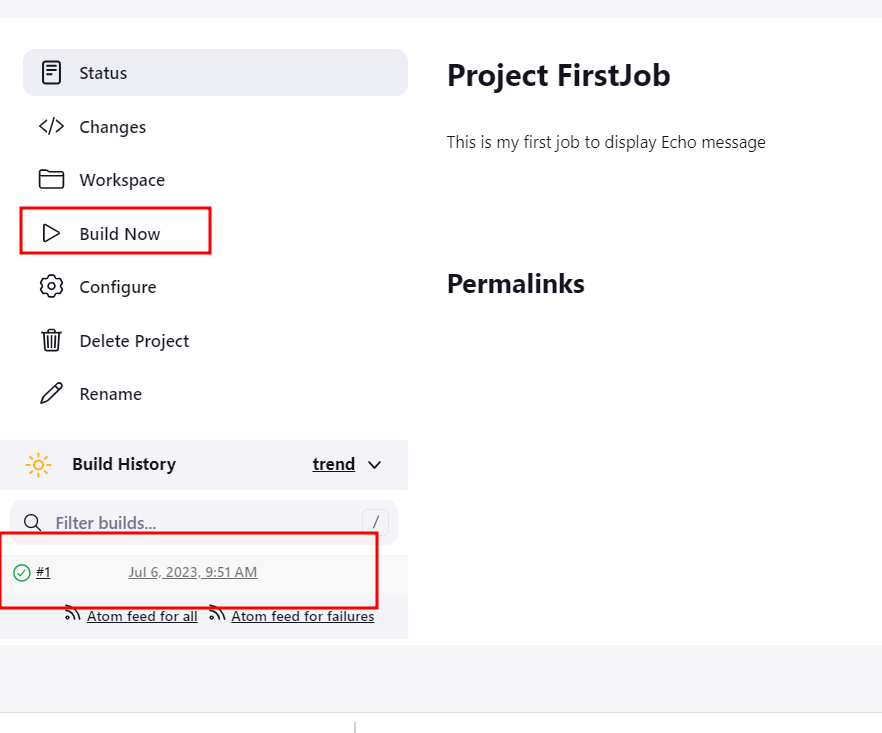
To do the setup for the software to build that application.

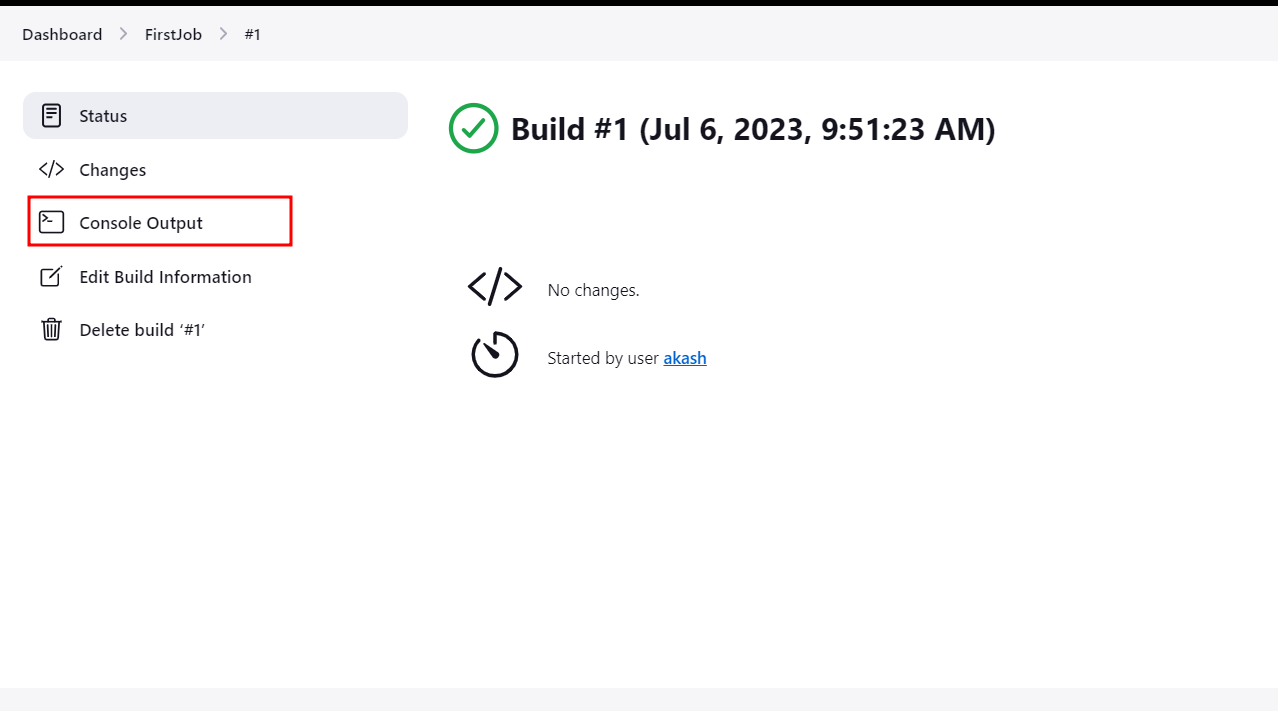


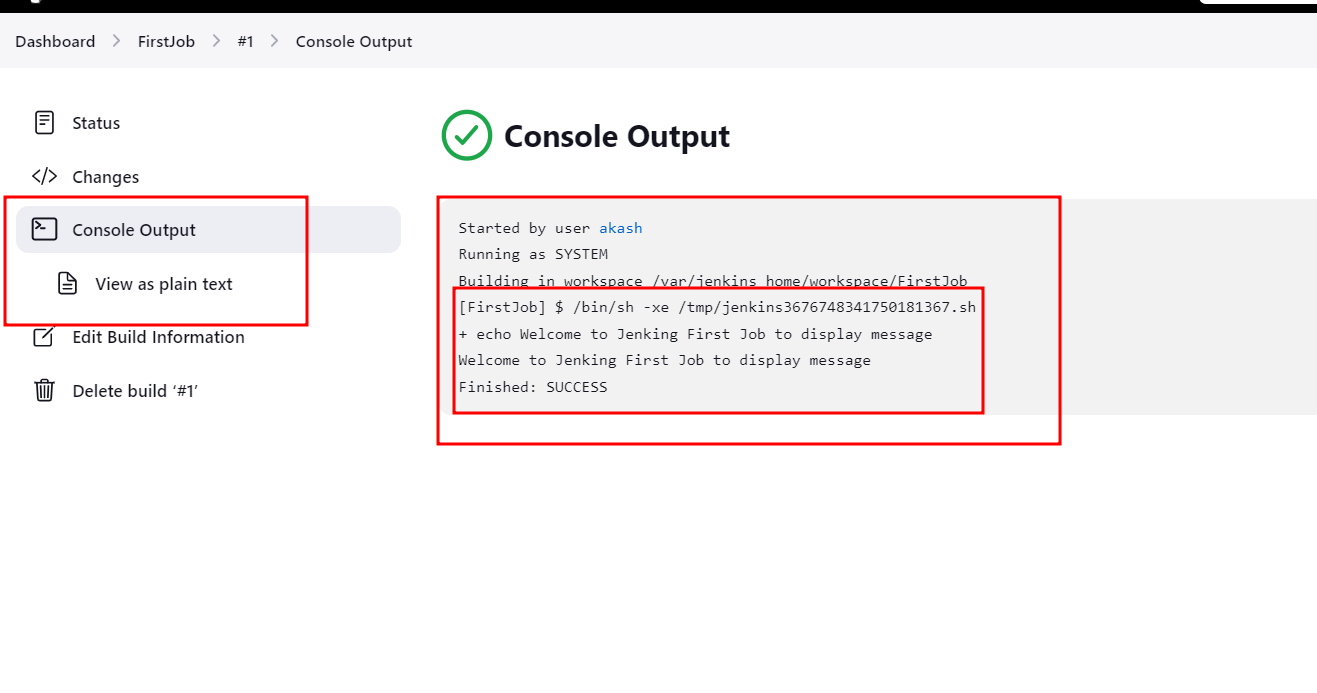
Build : In this option we are going to execute actual command to build the project.



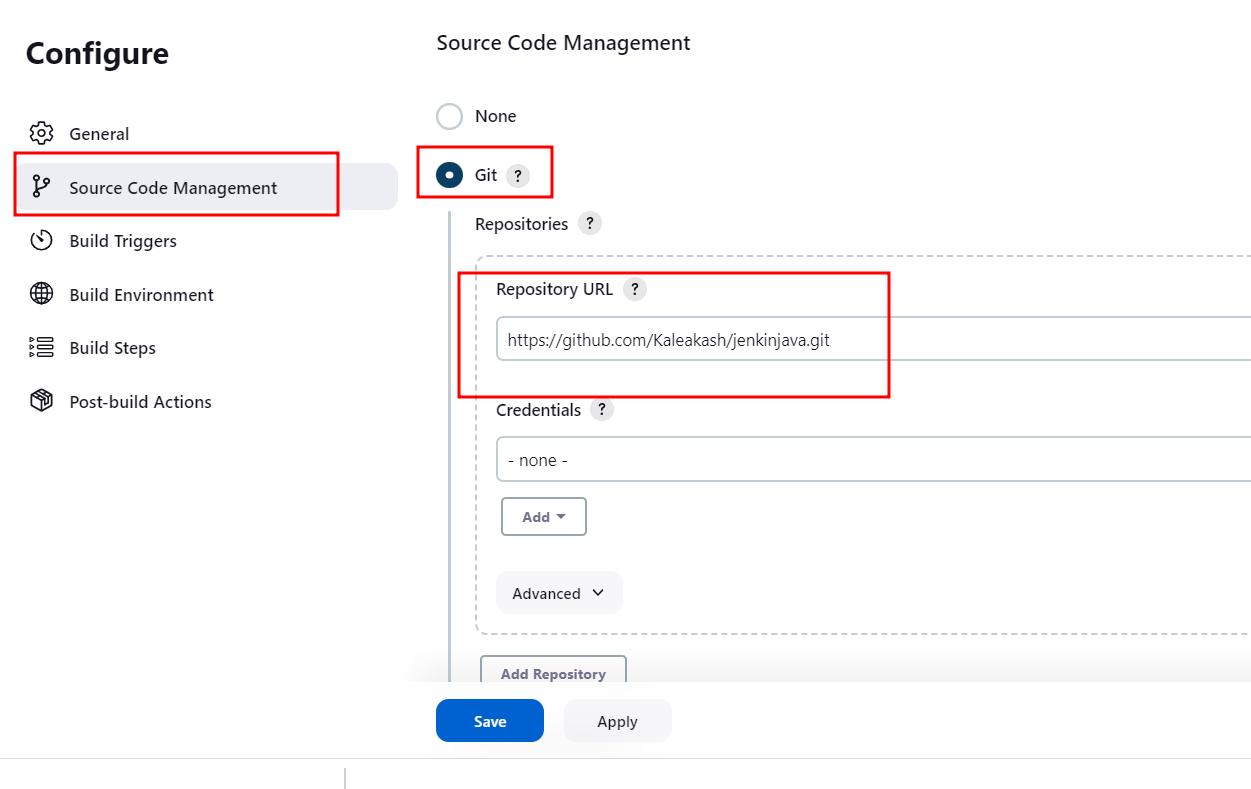








To assign Git repository in Jenkin Job



Create Spring boot project with one or more Rest API

@RestController {

@GetMapping(value=”/”)

public String sayHello() {

return “Welcome to Spring boot deploy application using jenking ci and cd tool with docker in EC2 instance by Akash Kale”;

}

}

Spring boot project run on different port number apart from 8080.

Or

Create React JS project

Then create new repository with name as yourName\_phase5\_project

git init

git add .

git commit -m “done”

git push -u origin main/master

Now login to AWS account

Create EC2 instance

Connect to EC2 instance using git bash or terminal

Then open port number 8080 or 3000 and 9090.

Then install Docker

Then pull Jenkin Image

Then run Jenkin image on port number 8080.

In Jenkin please create job which pull the project from Git hub account and run the Spring boot jar file or maven project or ./mvnw command

This command we write in Build option.

<http://ipAddress:9090>

Screen shot