<https://github.com/javaexpresschannel/springboot-hello/blob/master/Jenkinsfile> : Jenkin file URL

<https://www.youtube.com/watch?v=a6HwYMYmsls> Video URL

**Create Spring boot project**

With simple controller

**Main class**

package com;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com")

public class SimpleSpringBootWithAwsApplication {

public static void main(String[] args) {

SpringApplication.run(SimpleSpringBootWithAwsApplication.class, args);

}

}

**Controller class**

package com.controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class SampleController {

@GetMapping(value = "/")

public String greeting() {

return "Welcome to Spring boot with AWS host";

}

@GetMapping(value = "sayHello/{name}")

public String sayHello(@PathVariable("name") String name) {

return "Welcome user "+name+" to spring boot with AWS";

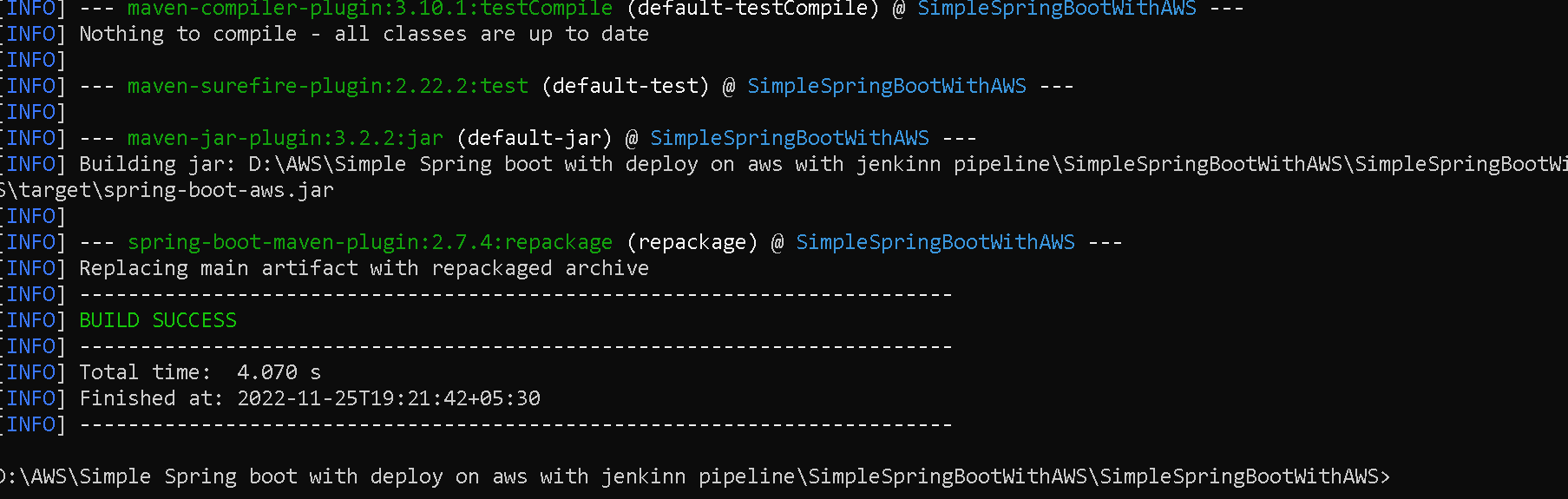
}

}

**application.properties**

serve.port = 9090

using mvn command please create the jar file for the project.



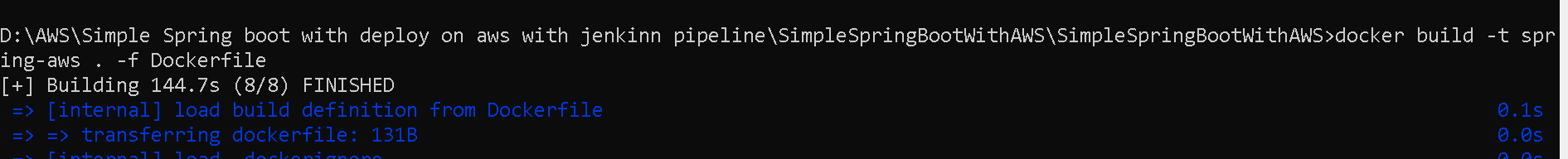
now we need to create the Dockerfile

**FROM** openjdk:11

**COPY** /target/spring-boot-aws.jar .

**CMD** ["java","-jar","spring-boot-aws.jar"]

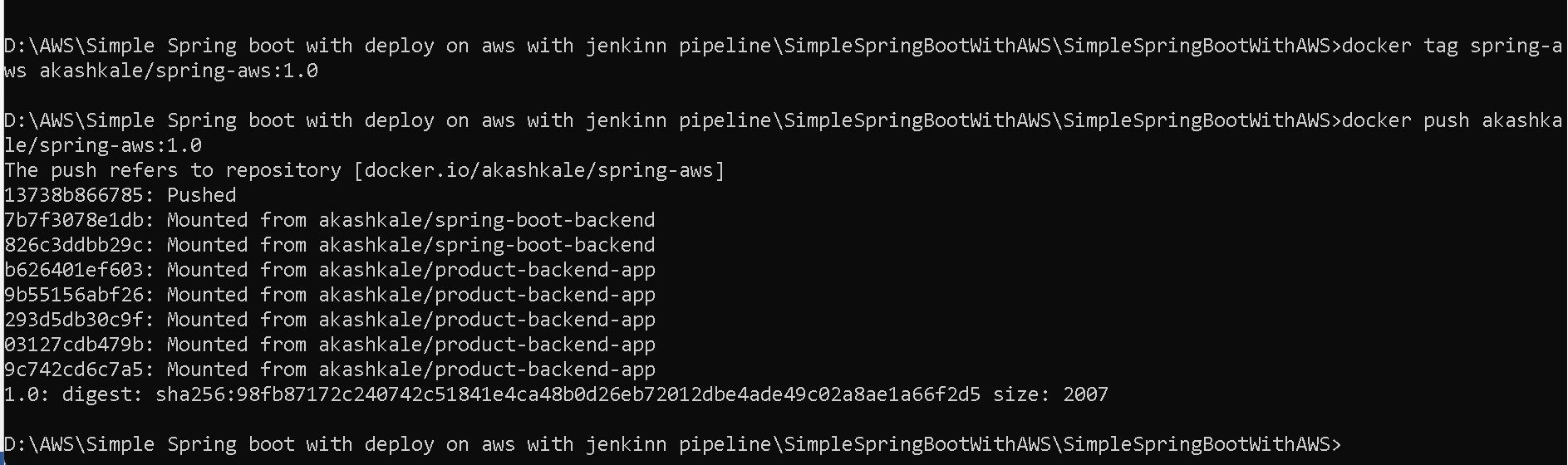
Now we need to build the Docker image



Now we need to create the tag for the image and push this image in docker hub account

**docker tag spring-aws akashkale/spring-aws:1.0**

**docker push akashkale/spring-aws:1.0**



How to deploy application in AWS Cloud (end to end development docker image jenkin pipeline, git, automatic deployment).

Now we will create the **Jenkinfile**

pipeline {

agent any

tools {

maven "M3"

}

stages {

stage('Compile and Clean') {

steps {

// Run Maven on a Unix agent.

sh "mvn clean compile"

}

}

stage('deploy') {

steps {

sh "mvn package"

}

}

stage('Build Docker image'){

steps {

echo "Hello Java Express"

sh 'ls'

sh 'docker build -t akashkale/spring-aws-jenkin:${BUILD\_NUMBER} .'

}

}

stage('Docker Login'){

steps {

withCredentials([string(credentialsId: 'DockerId', variable: 'Dockerpwd')]) {

sh "docker login -u akashkale -p ${Dockerpwd}"

}

}

}

stage('Docker Push'){

steps {

sh 'docker push akashkale/spring-aws-jenkin:${BUILD\_NUMBER}'

}

}

stage('Docker deploy'){

steps {

sh 'docker run -itd -p 9090:9090 akashkale/spring-aws-jenkin:${BUILD\_NUMBER}'

}

}

stage('Archving') {

steps {

archiveArtifacts '\*\*/target/\*.jar'

}

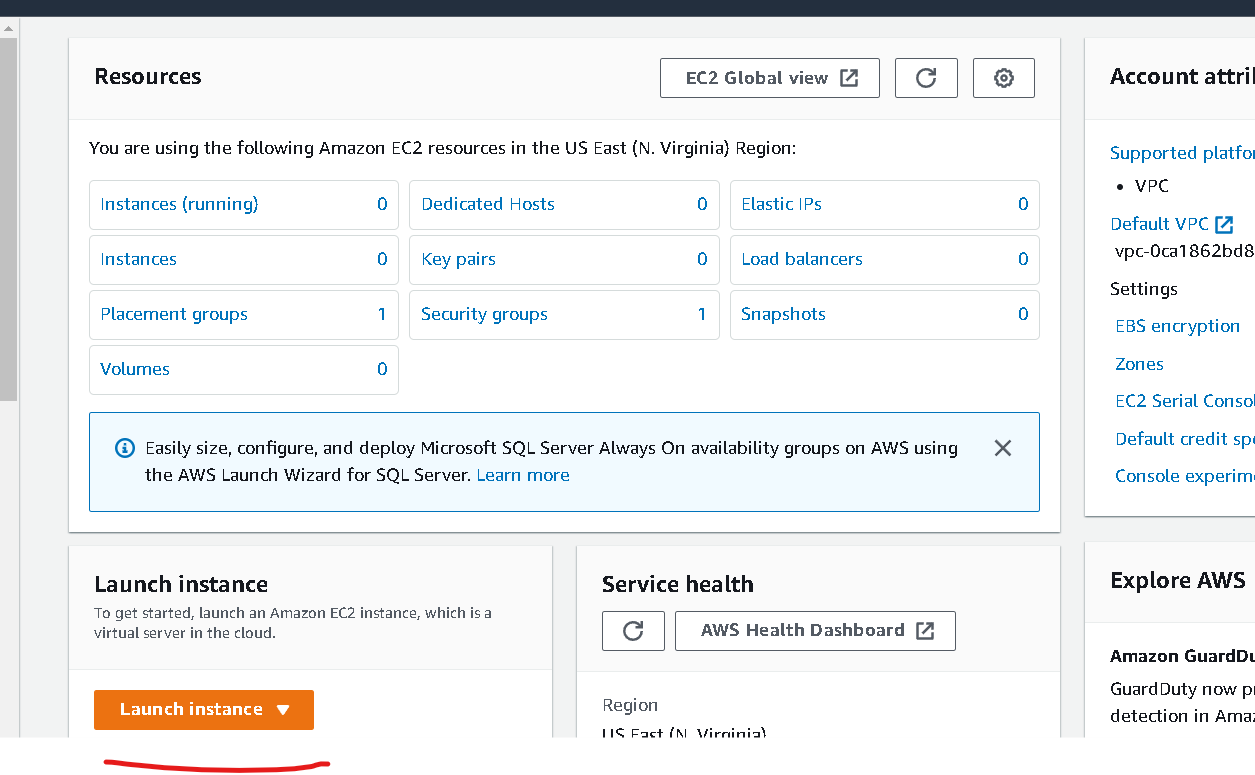
}

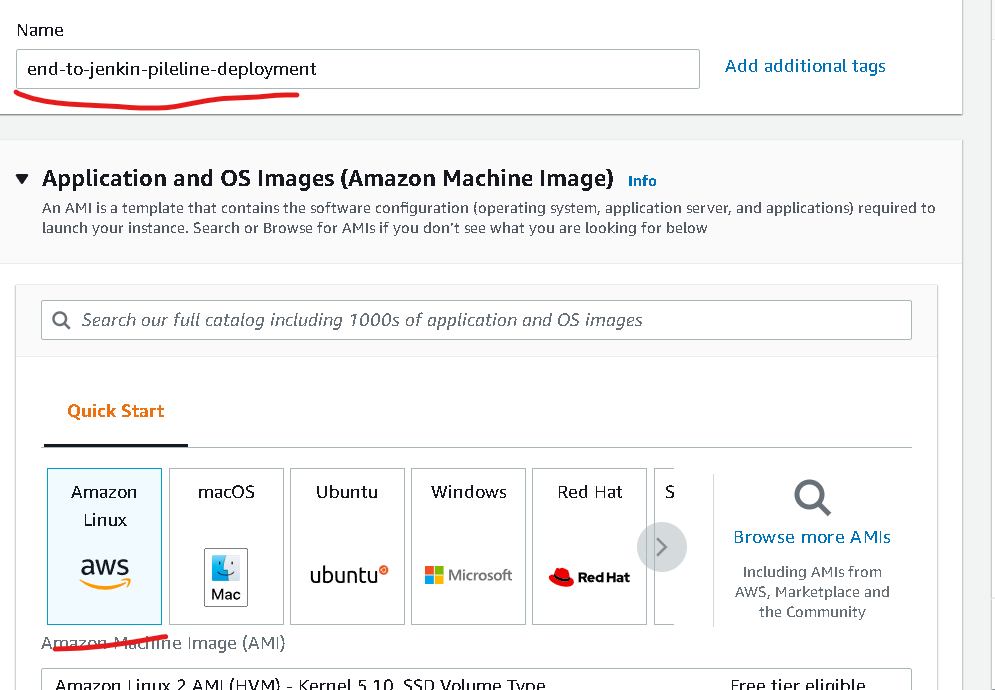
}

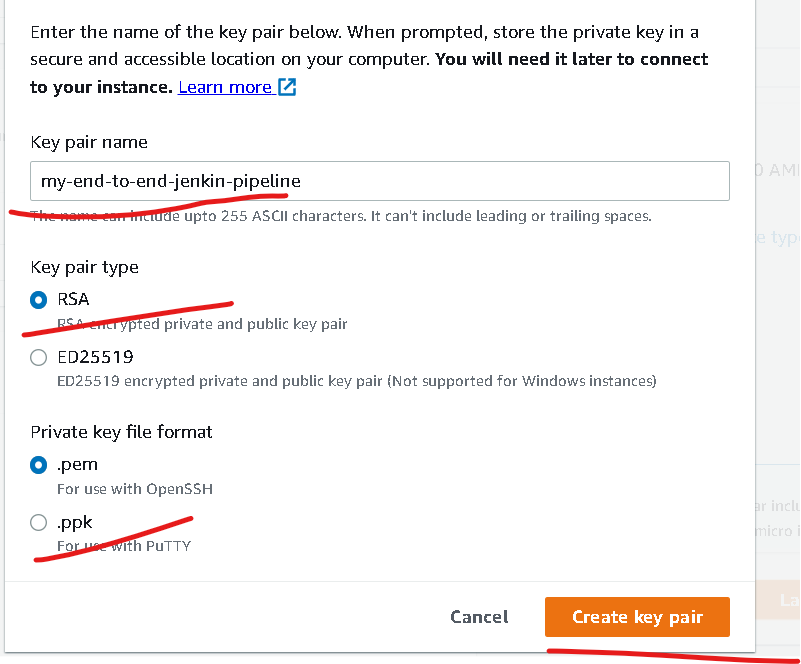
}

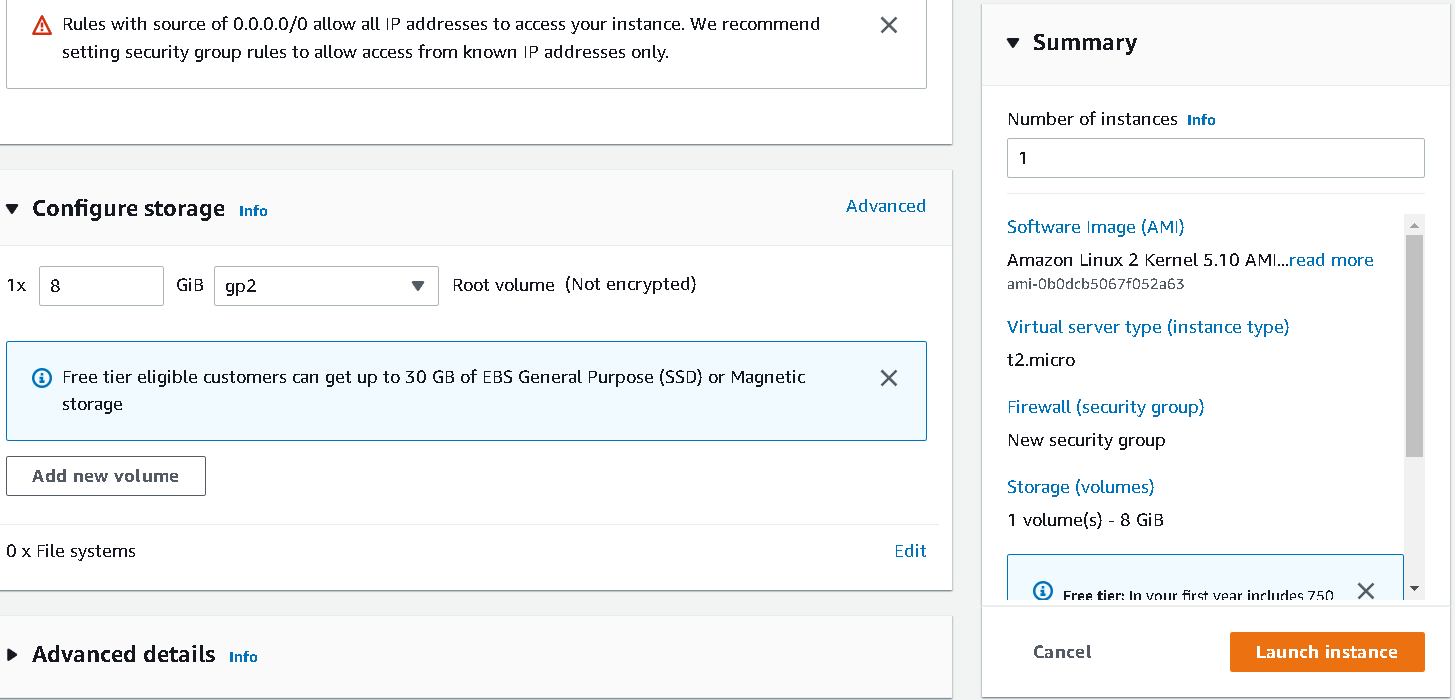
<https://github.com/Kaleakash/spring-boot-aws-jenkin-pipeline.git>

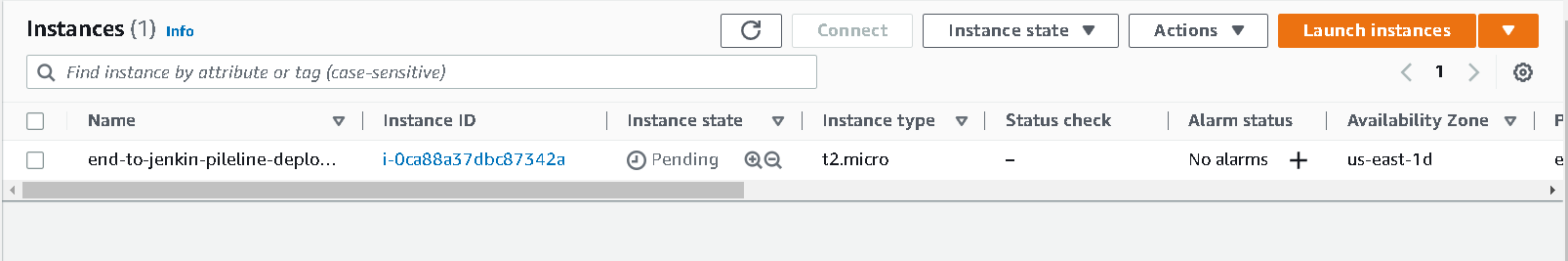
**First we will create the EC2 instance**



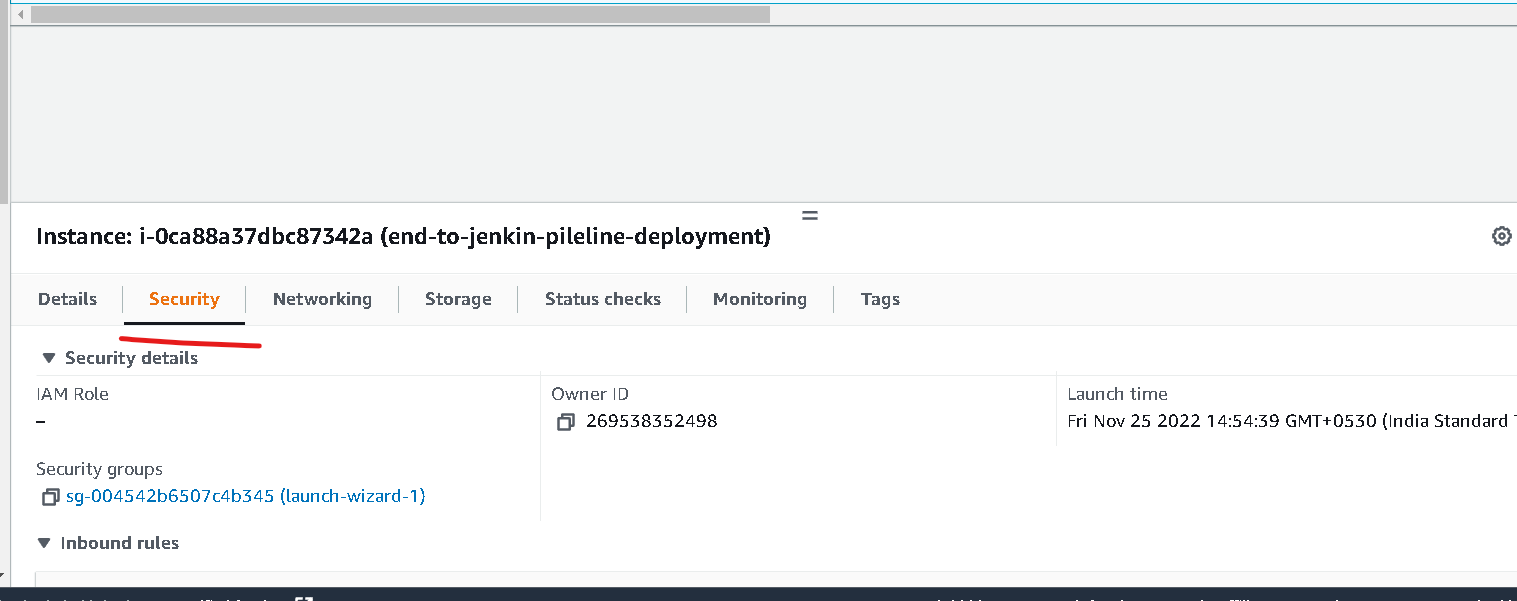


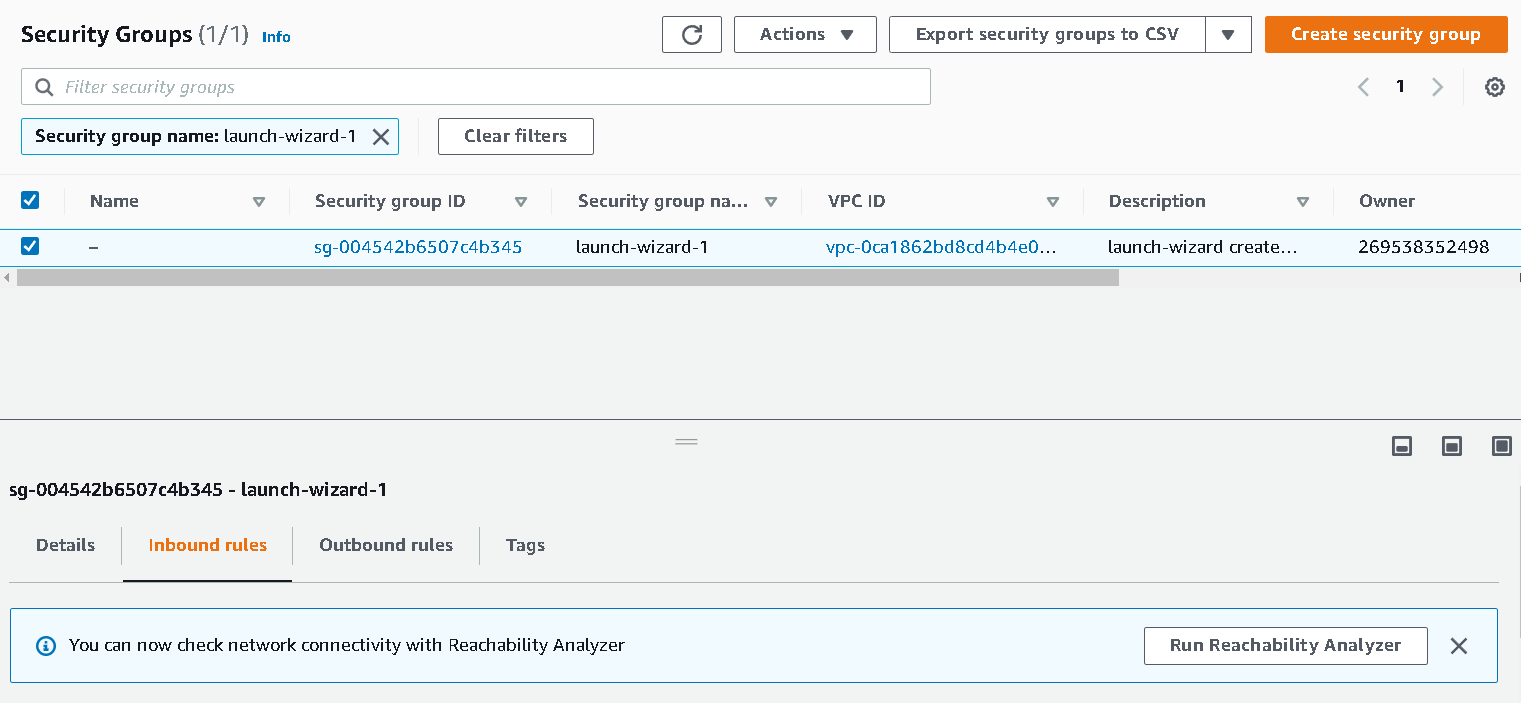


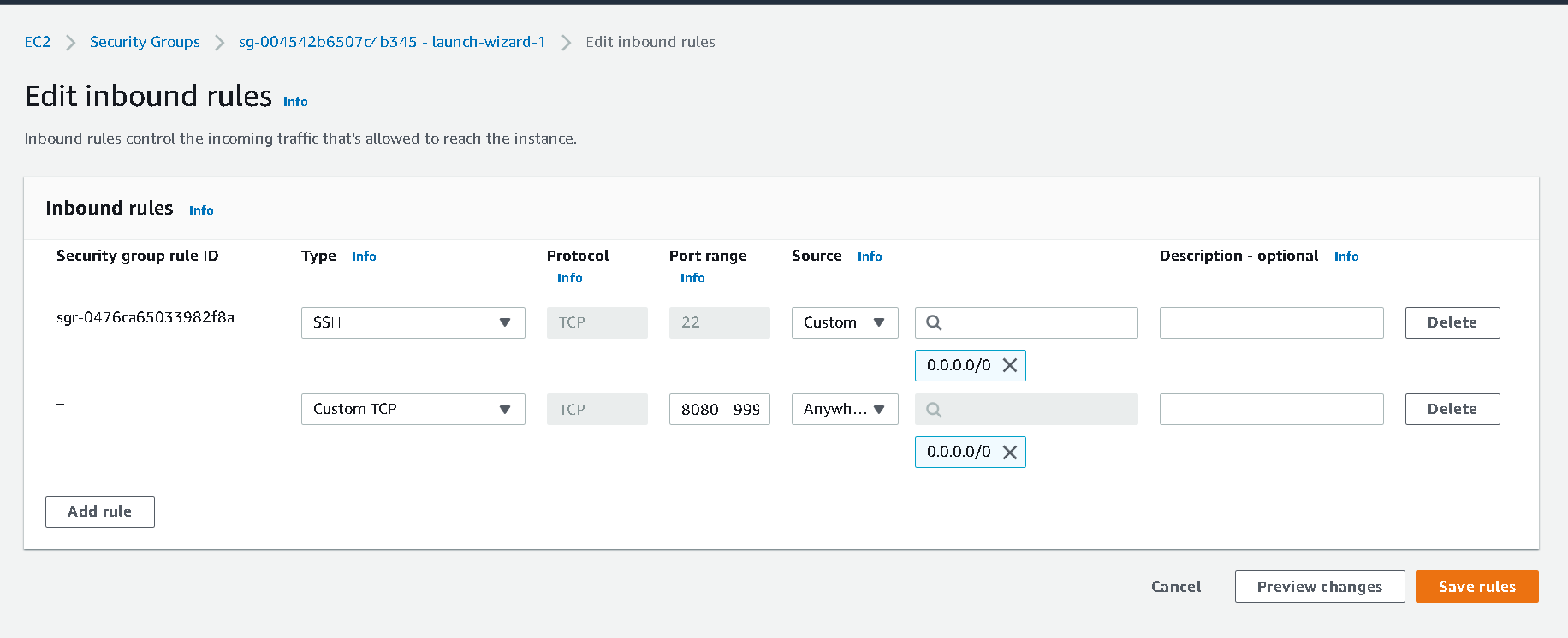




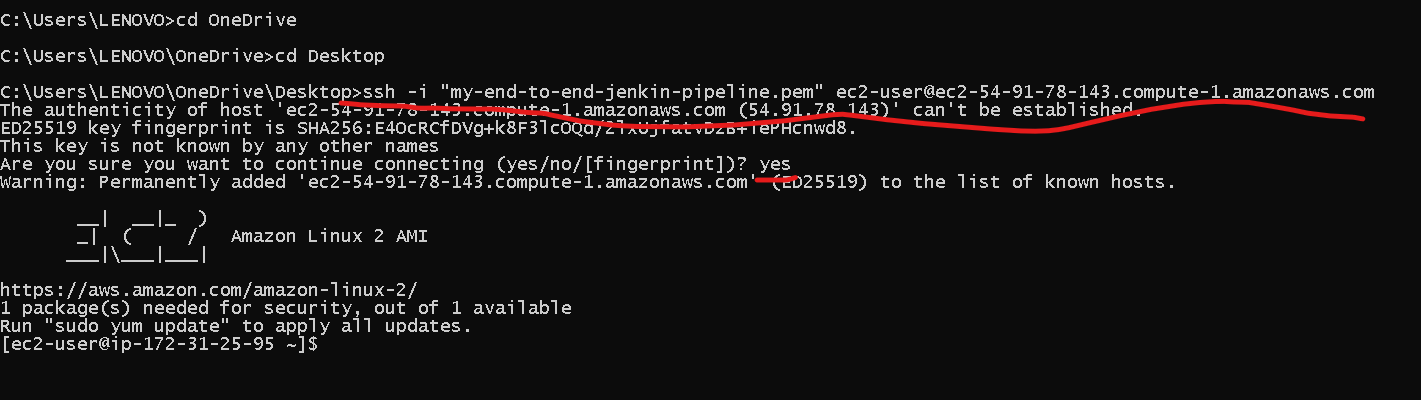
Enable in bond rules for the port number 9090







After open the port number range as 8080-9999 then connect to EC2 instance using GitBash terminal



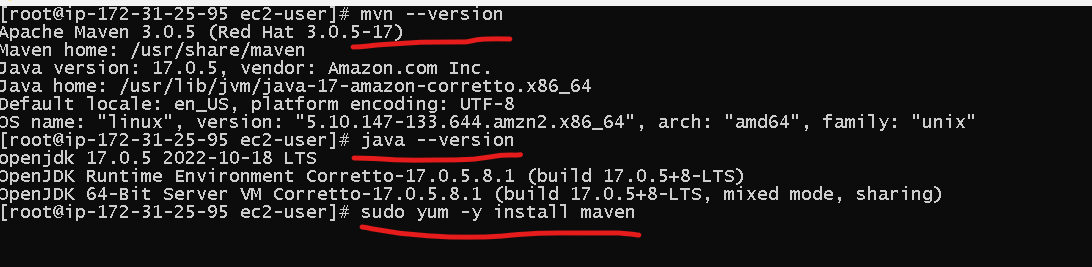
Now move to sudo user using command as

**sudo su**

Now we will install the maven using the command as

**sudo yum –y install maven**

This command is use to install the maven and openjdk



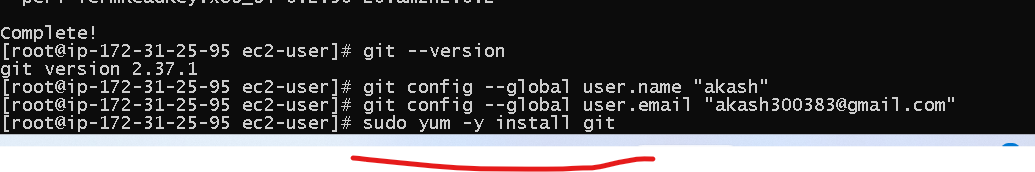
Now we need to install the git

**sudo yum –y install git**

**git --version**

git config --global user.name “akash”

git config --global user.email [akash300383@gmail.com](mailto:akash300383@gmail.com)



Now install the Jenkin using the command as

Now, to download the latest Jenkins package

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

\*\*iv. \*\*To enable the installation of the package, import the key file from Jenkins-CI:

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

Install Jenkins on the EC2 instance

sudo yum install jenkins

To start the Jenkins service

sudo service jenkins start

To check the status

sudo systemctl status jenkins

This one is use to get the default password

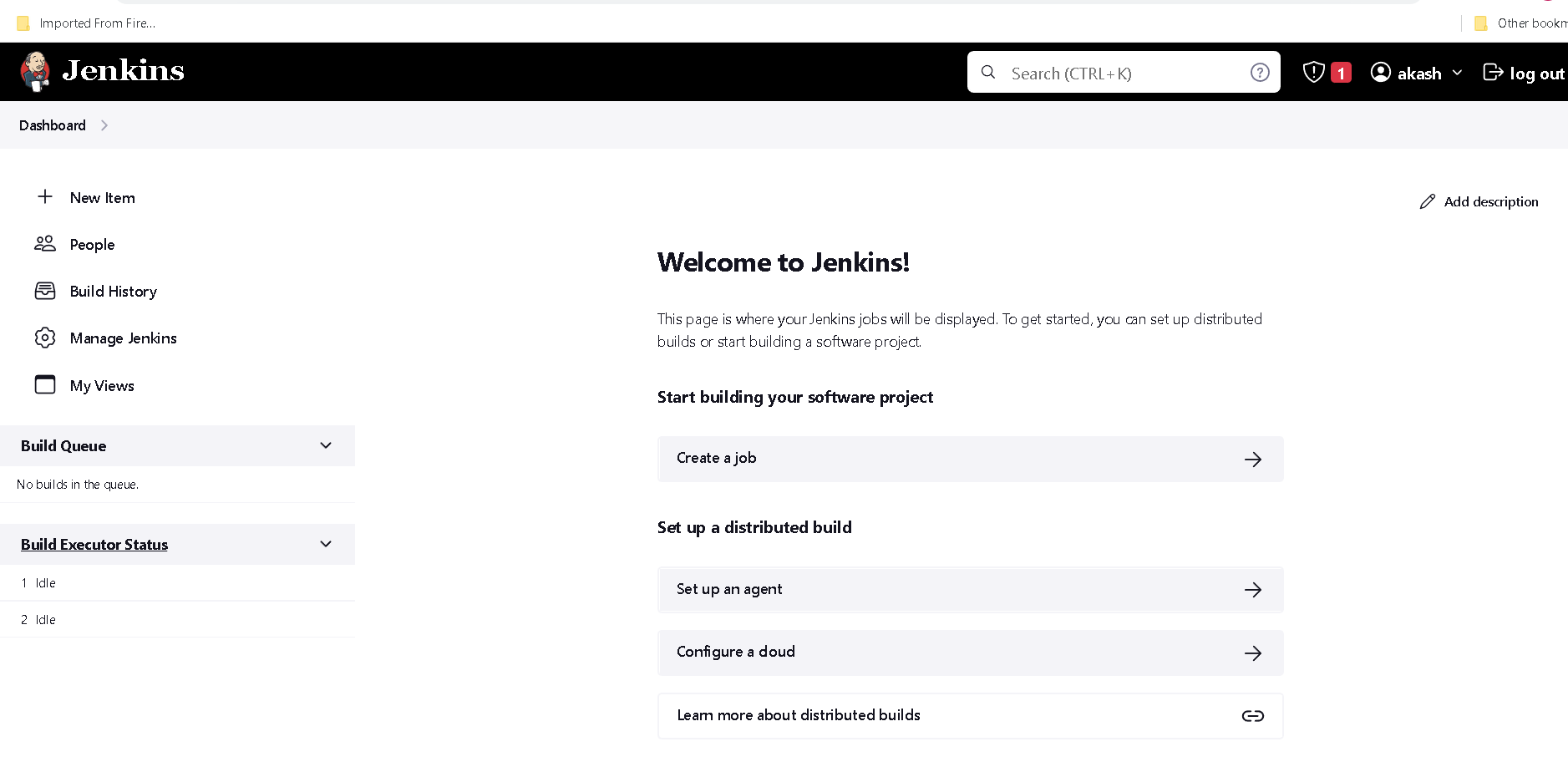
sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Then check the Public id address of Ec2 instance and using the ID Address

<http://54.91.78.143:8080/>

The copy and paste the password

Then download the plugin required.



Now install the docker

**install docker**

**sudo yum -y install docker**

**sudo service docker start**

**sudo docker info**

**sudo usermod -a -G docker ec2-user**

**sudo usermod -a -G docker Jenkins**

after added all groups please reboot ec2 instance using command as

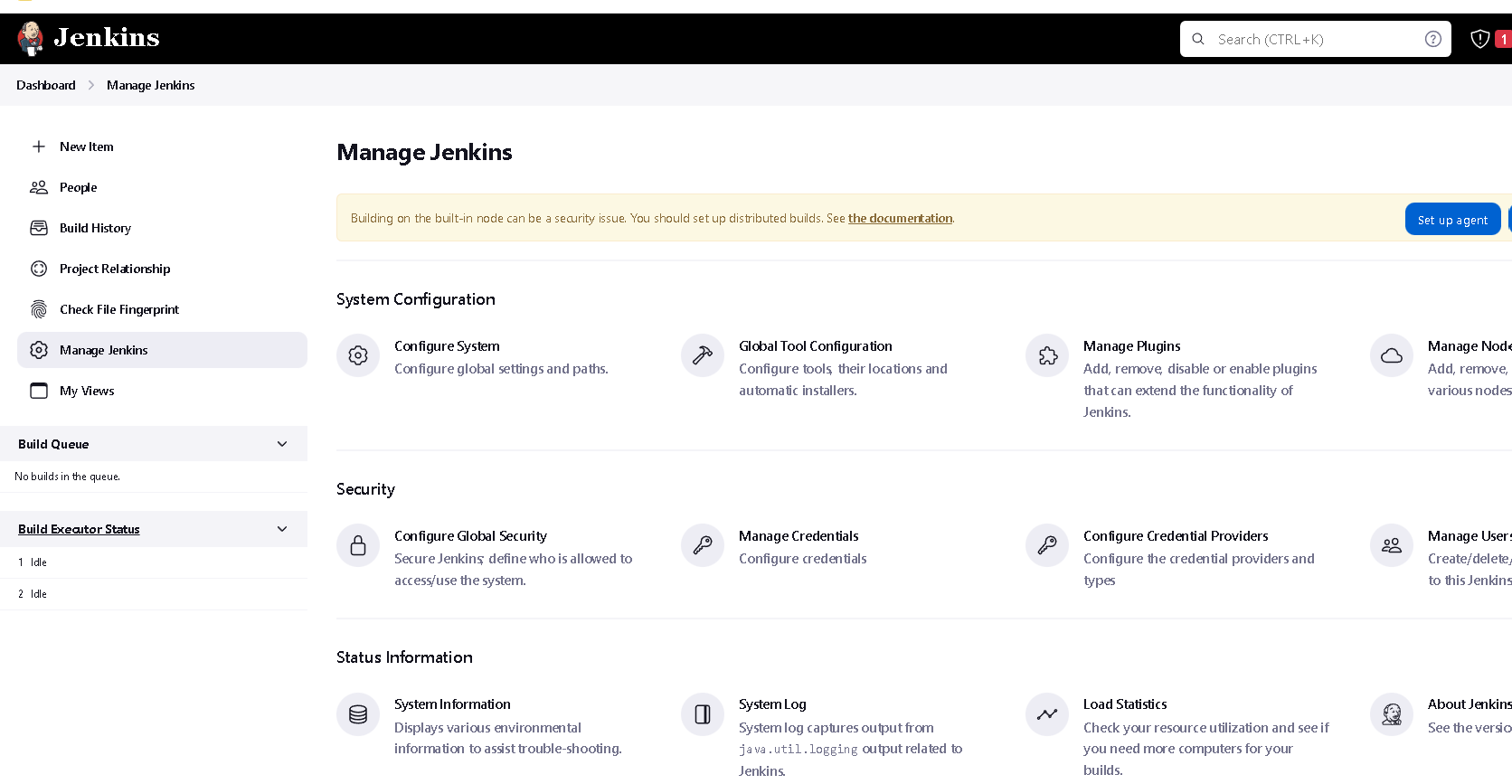
**sudo reboot**

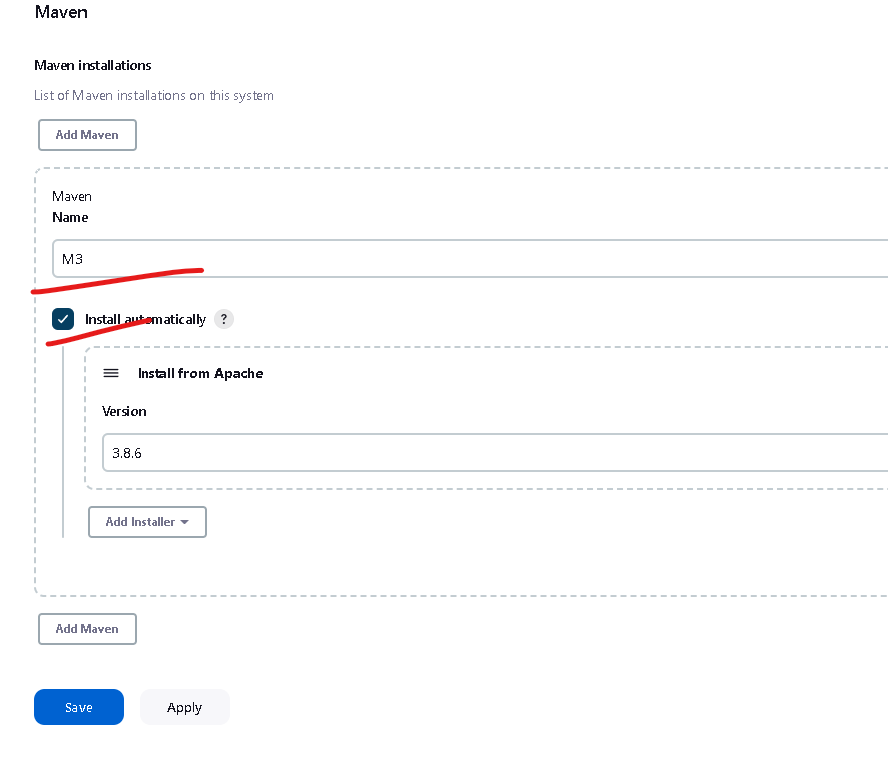
Now open the Jenkin

Make sure Jenkin service running in EC2 instance

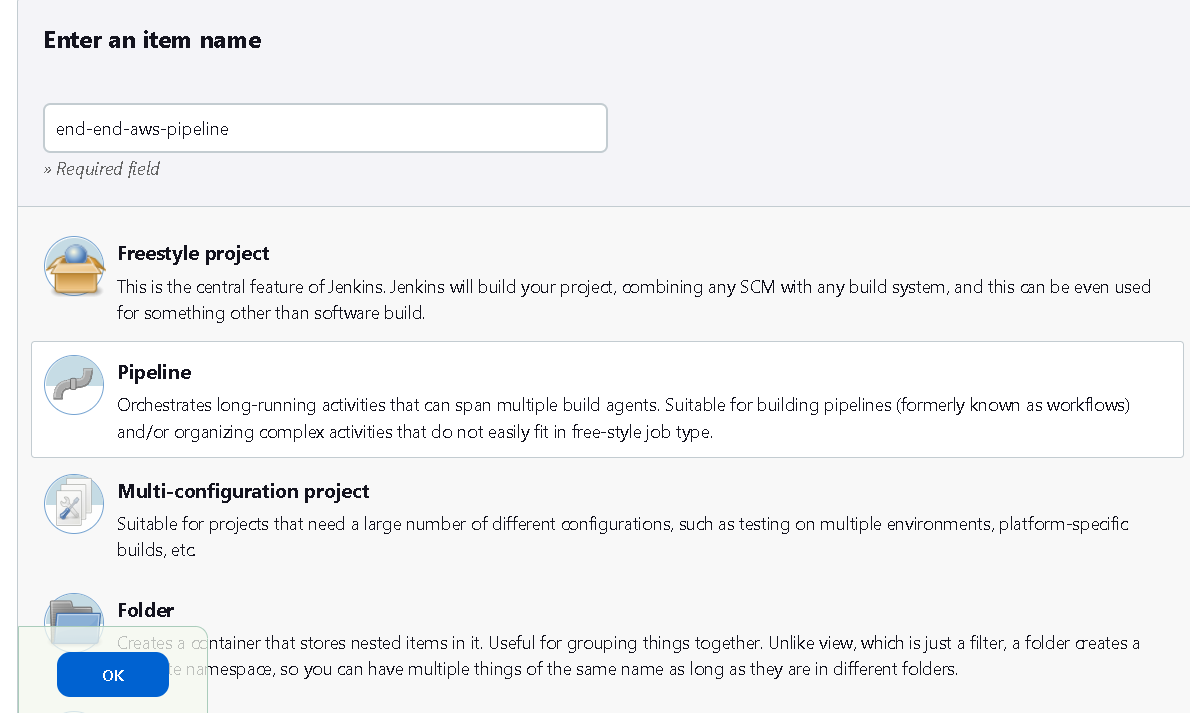
Before creating Jenking job we need to add

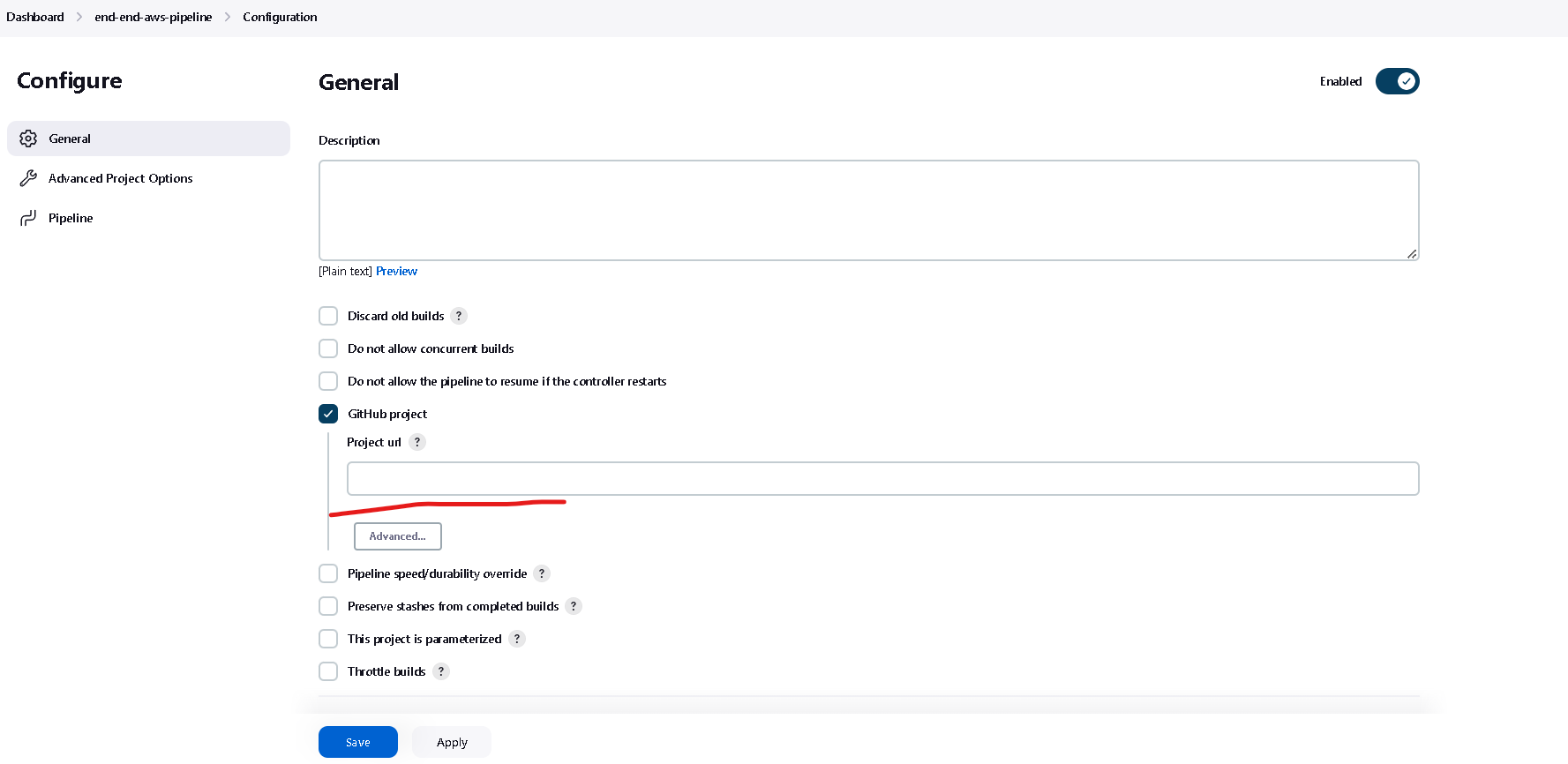
M3 Maven





Now we will create the Jenkin pipe line job

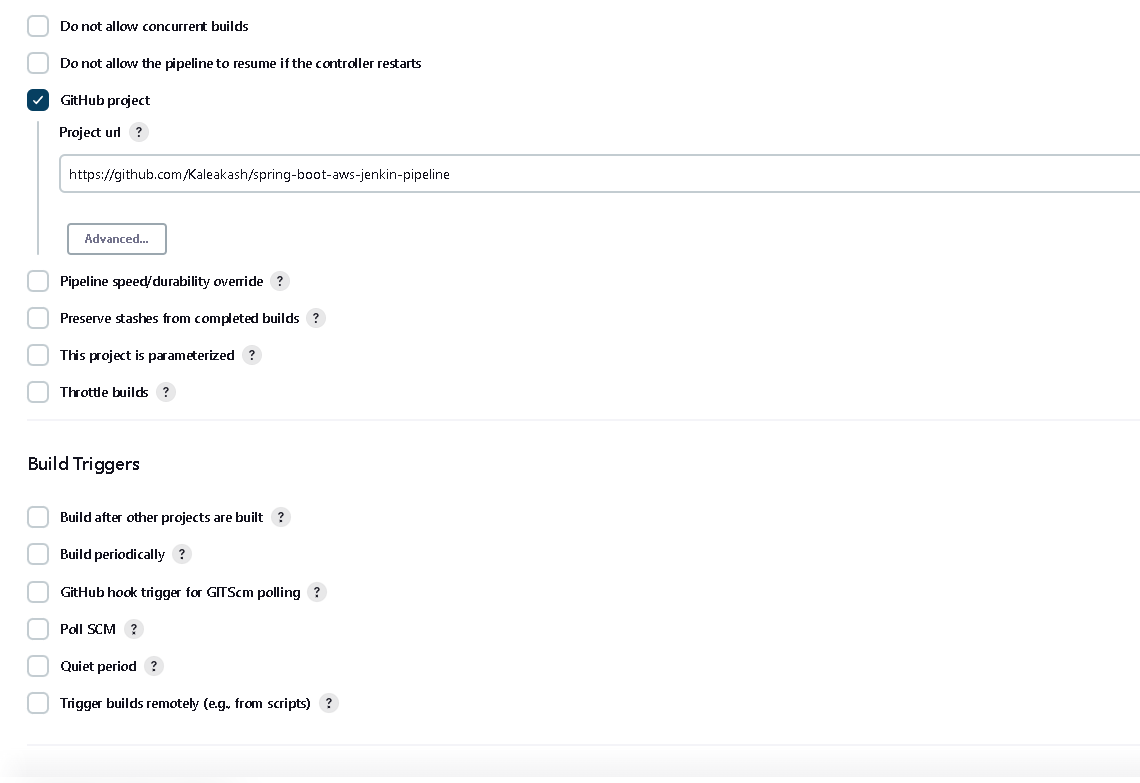


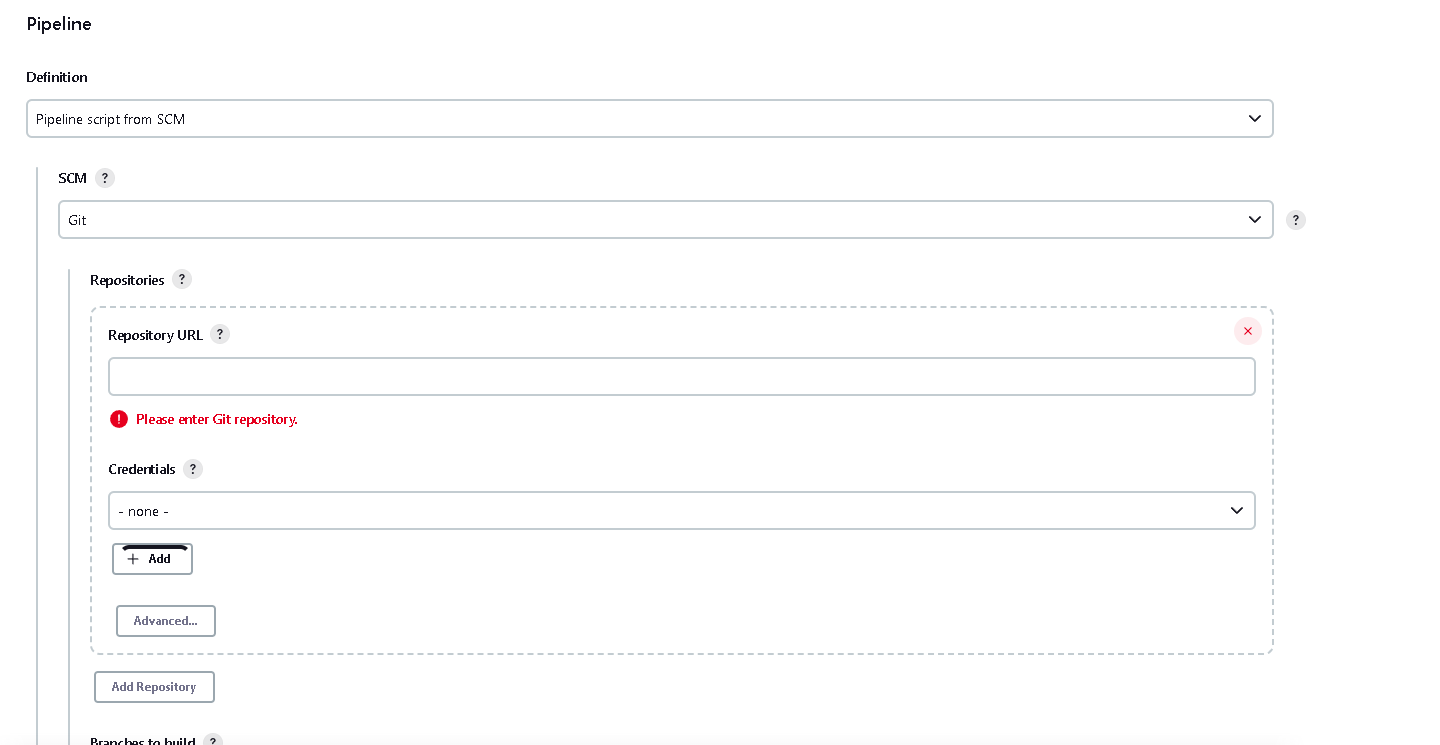


In Github project we now copy and paste GIT URL as

<https://github.com/Kaleakash/spring-boot-aws-jenkin-pipeline>

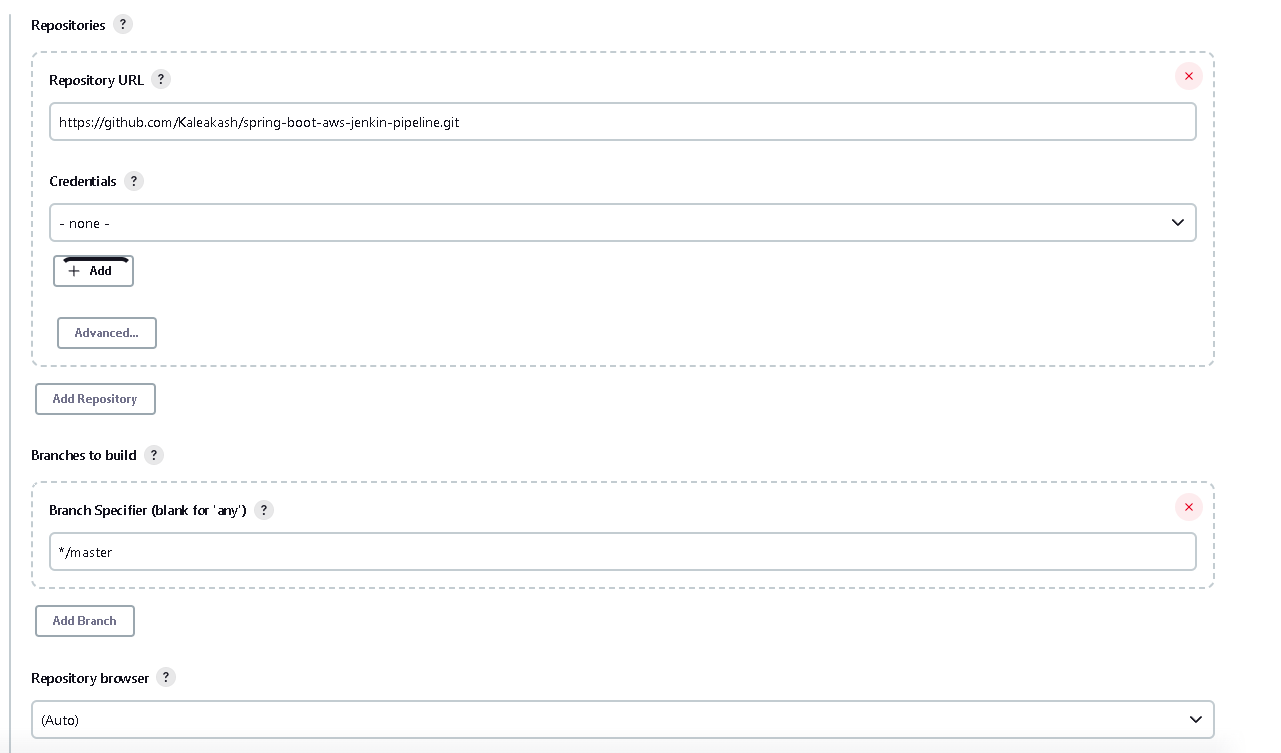
This is website URL

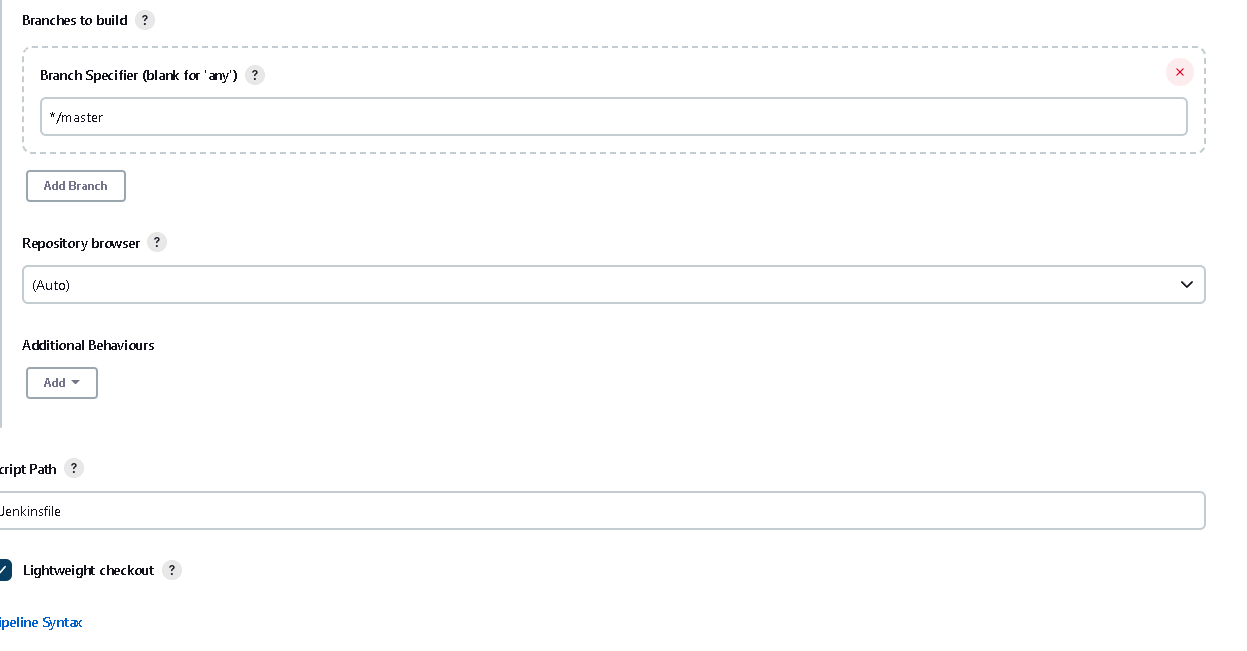




Now we need to provide the git repository URL

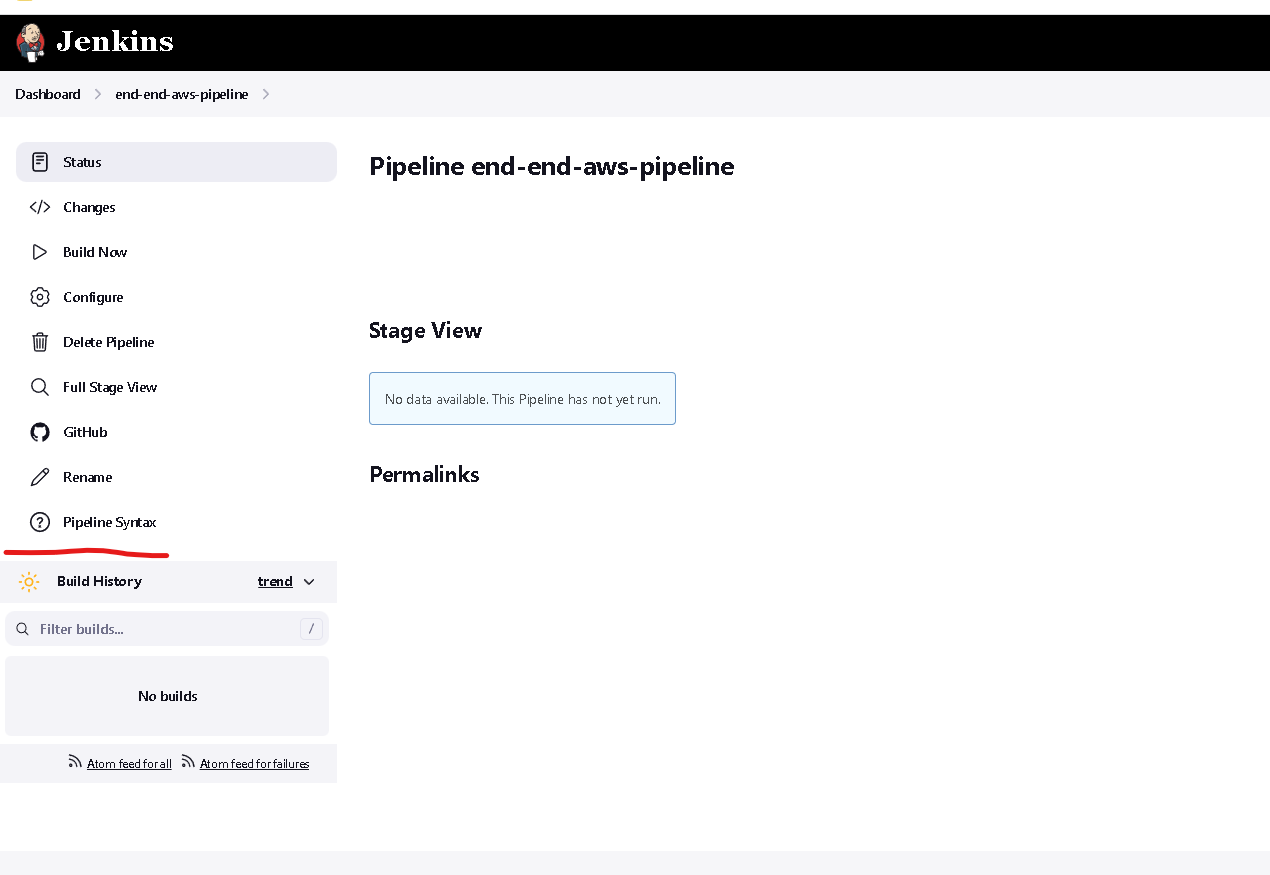
<https://github.com/Kaleakash/spring-boot-aws-jenkin-pipeline.git>

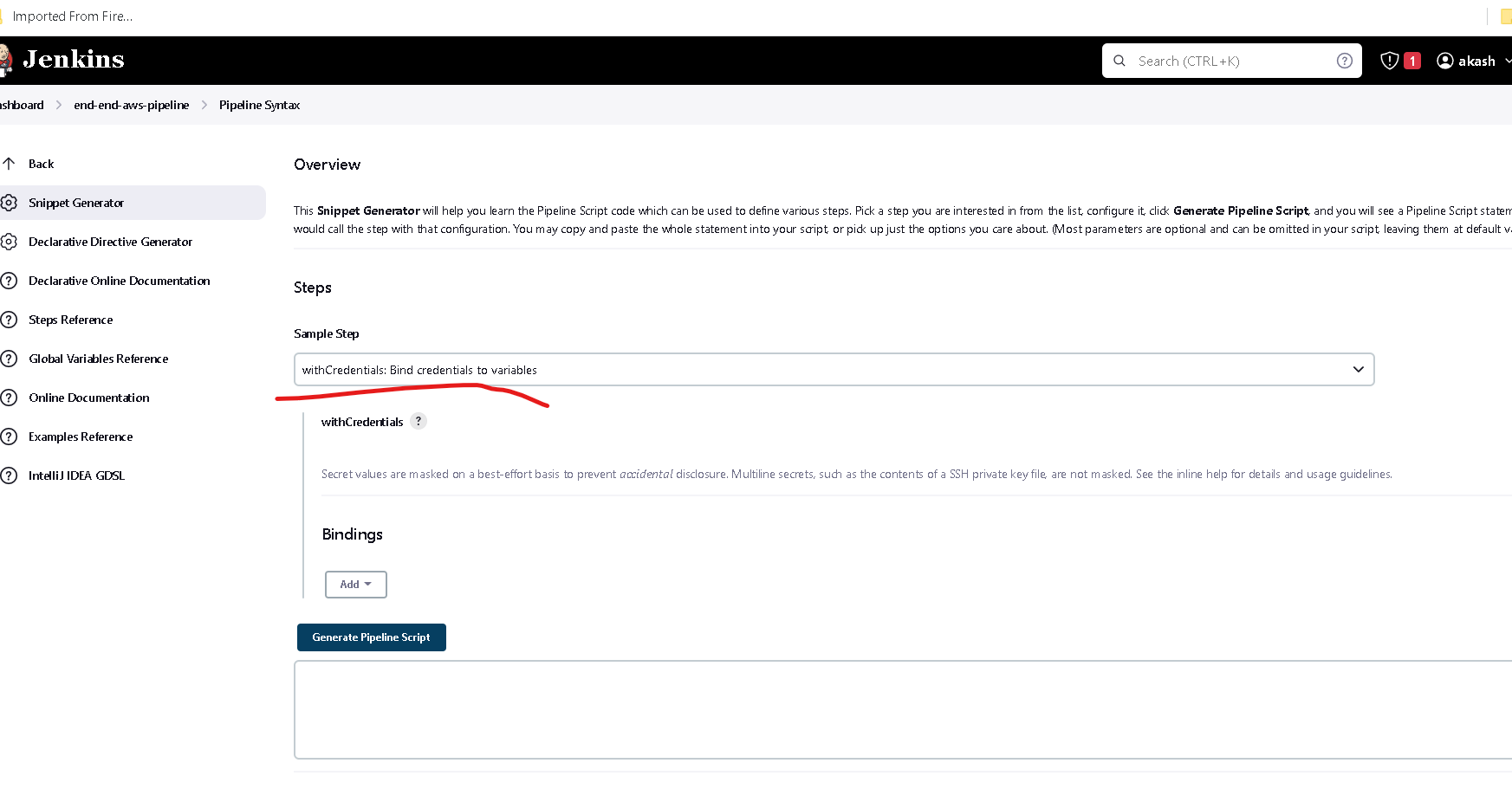




Now we need to save the project

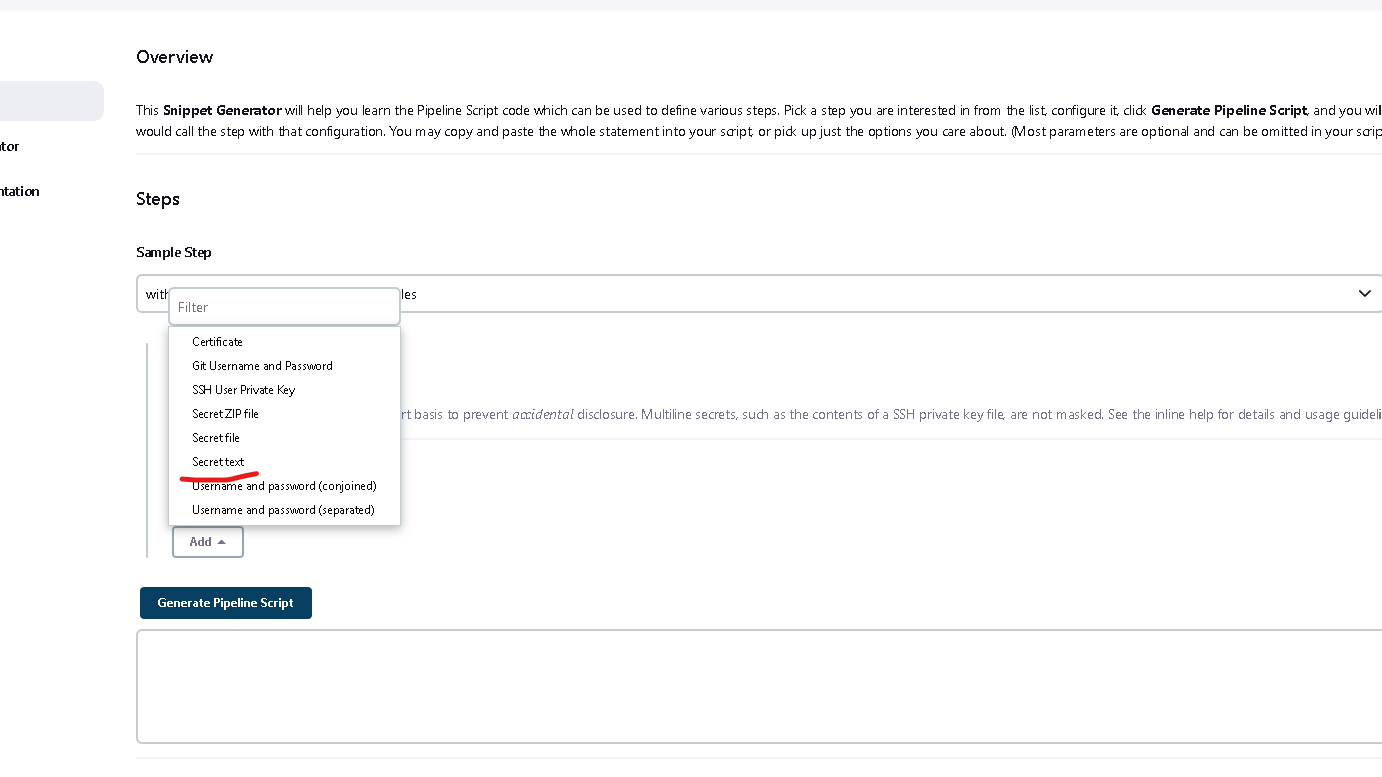
One we save now we need to configure our credentials

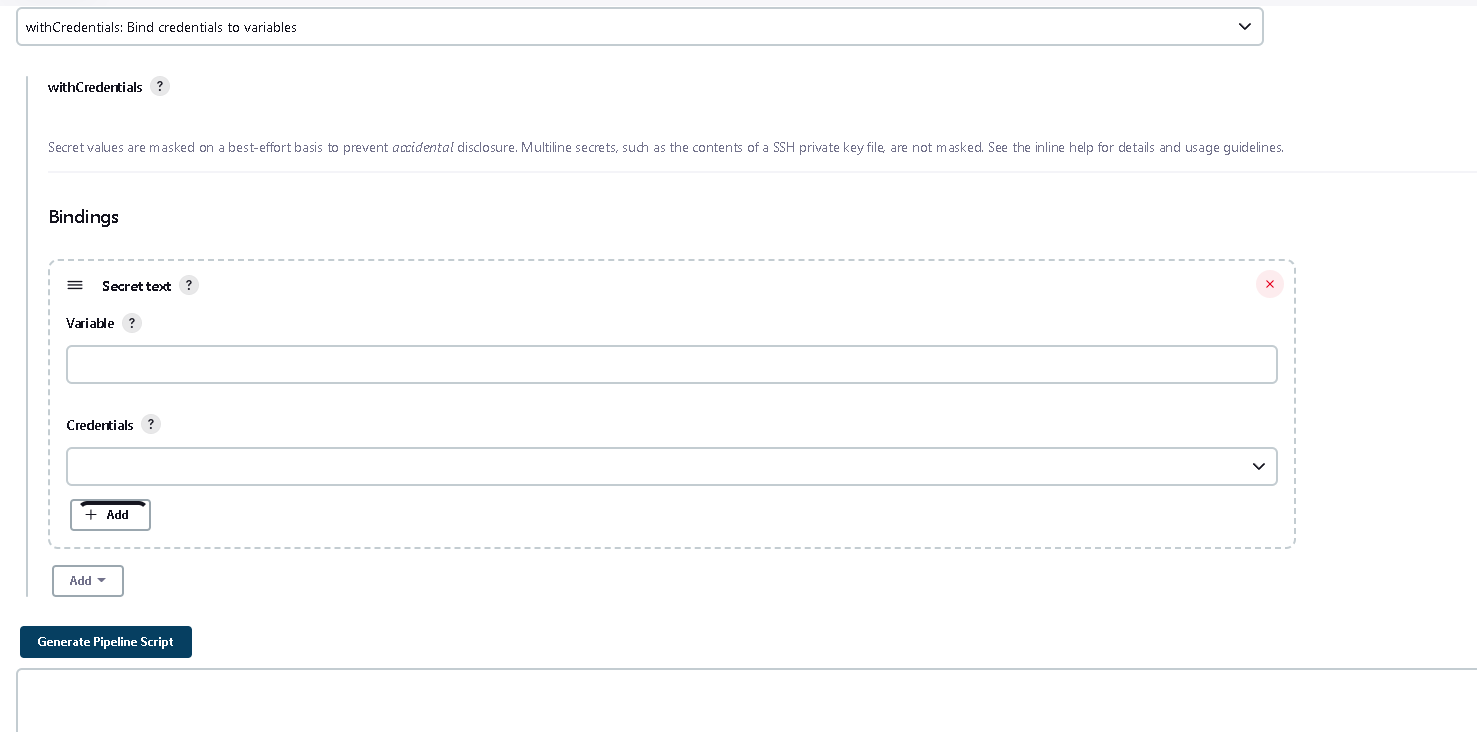


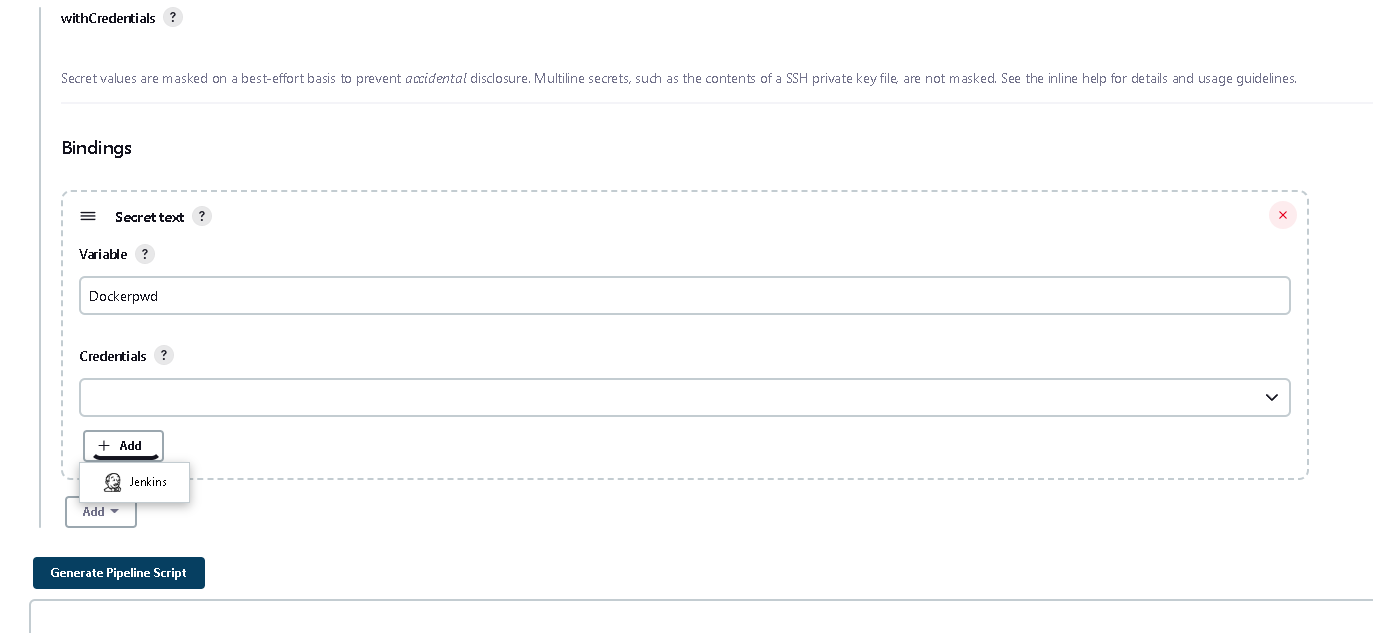


In Sample step we need to search the option as withCredential: bind credentials to variable options

Here search the option in binding as SecretText

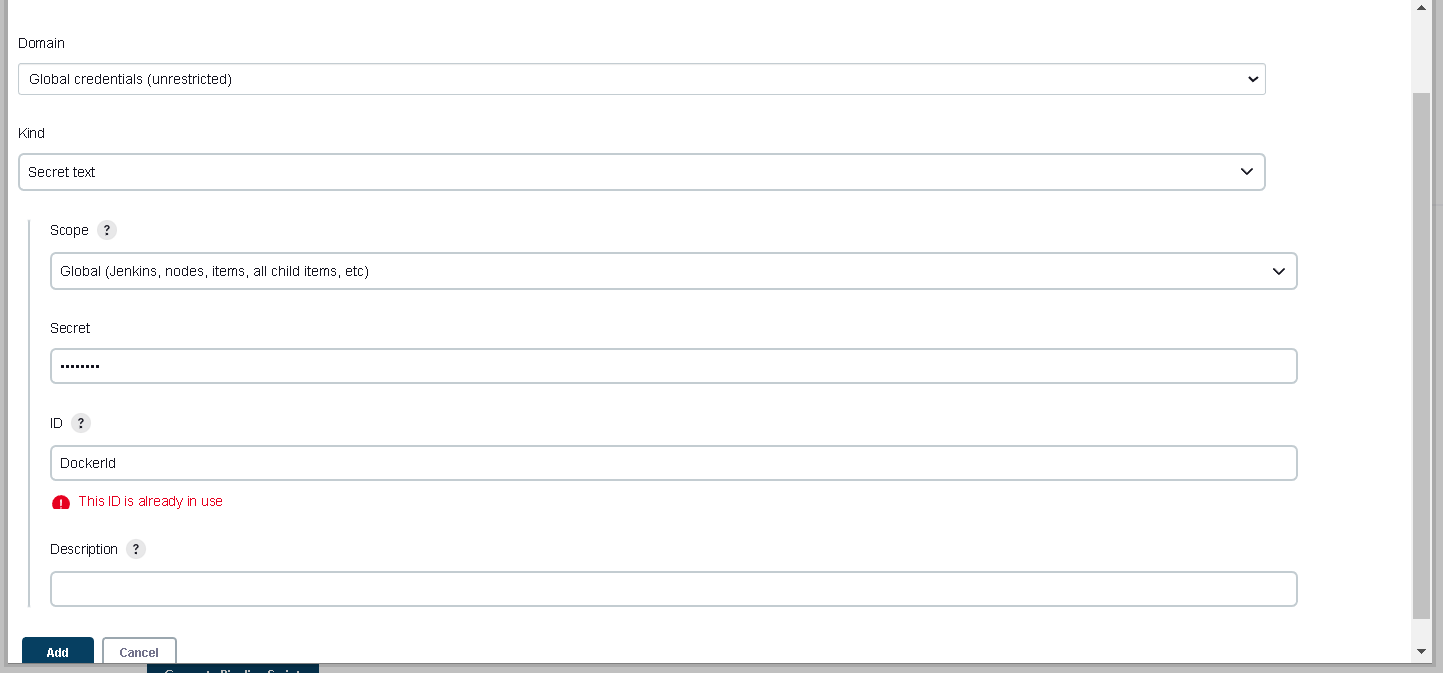






Here variable name present in Dockerfile in github

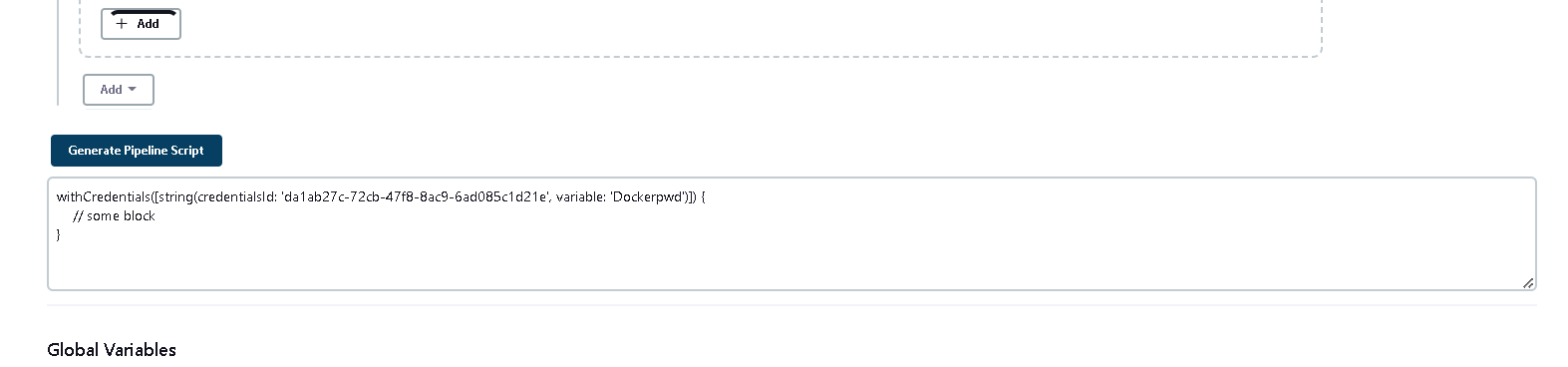
Then click add button



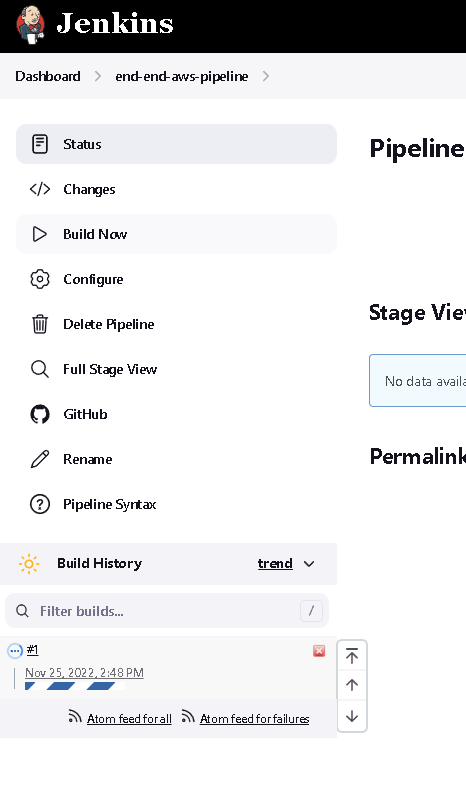
Select the option as Secret text and in secret please write out Docker hub account password ie Akash@83



Now then click on Generate pipeline script

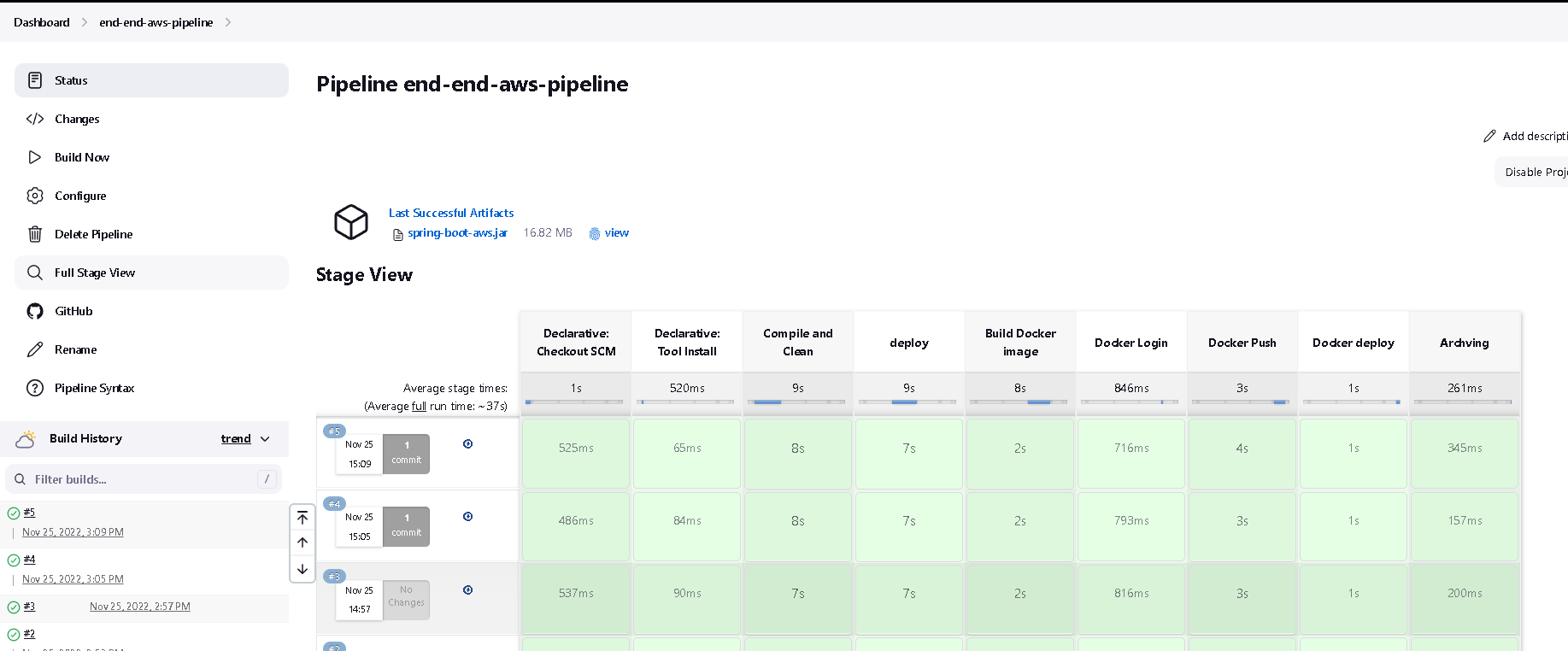


After every thing set now we click on build now

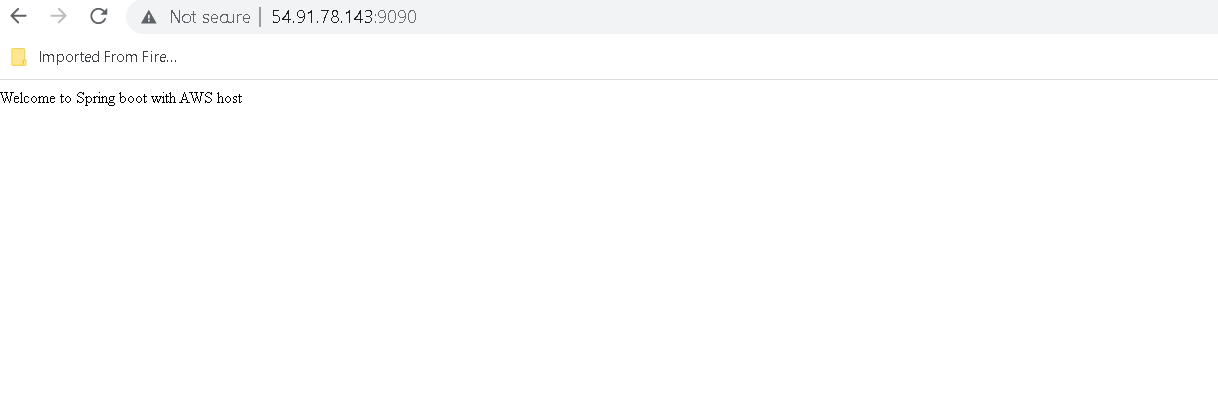


Now wait for few seconds

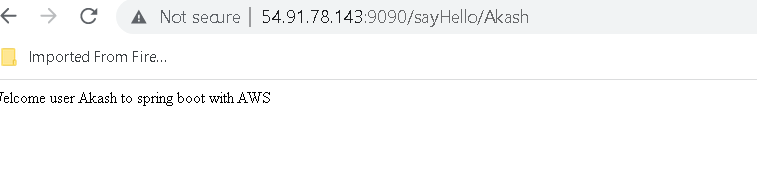
Project build successfully



Then open the application and check it



Another URL also check it



Both are working fine