# **Jenkins Declarative pipeline Script**

**Pre-requisites**

Step 1 — Launch an Ubuntu(22.04) T2 Large Instance

Install java:

sudo apt update

sudo apt install fontconfig openjdk-17-jre

java -version

openjdk version "17.0.8" 2023-07-18

OpenJDK Runtime Environment (build 17.0.8+7-Debian-1deb12u1)

OpenJDK 64-Bit Server VM (build 17.0.8+7-Debian-1deb12u1, mixed mode, sharing)

Install Jenkins:

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt-get update

sudo apt-get install jenkins

sudo systemctl start jenkins

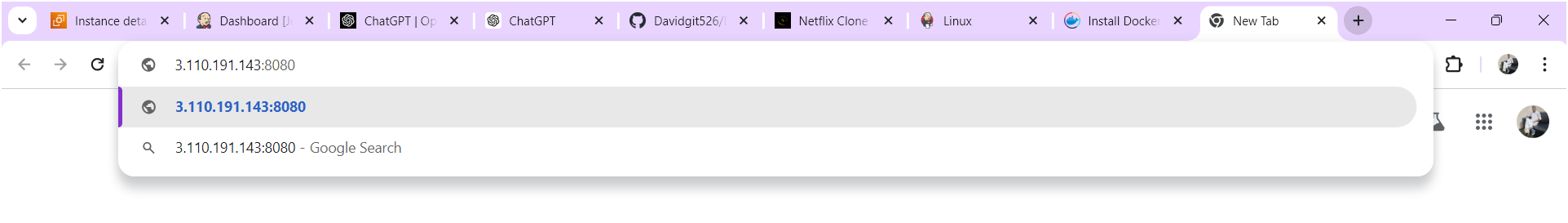
sudo systemctl enable jenkins

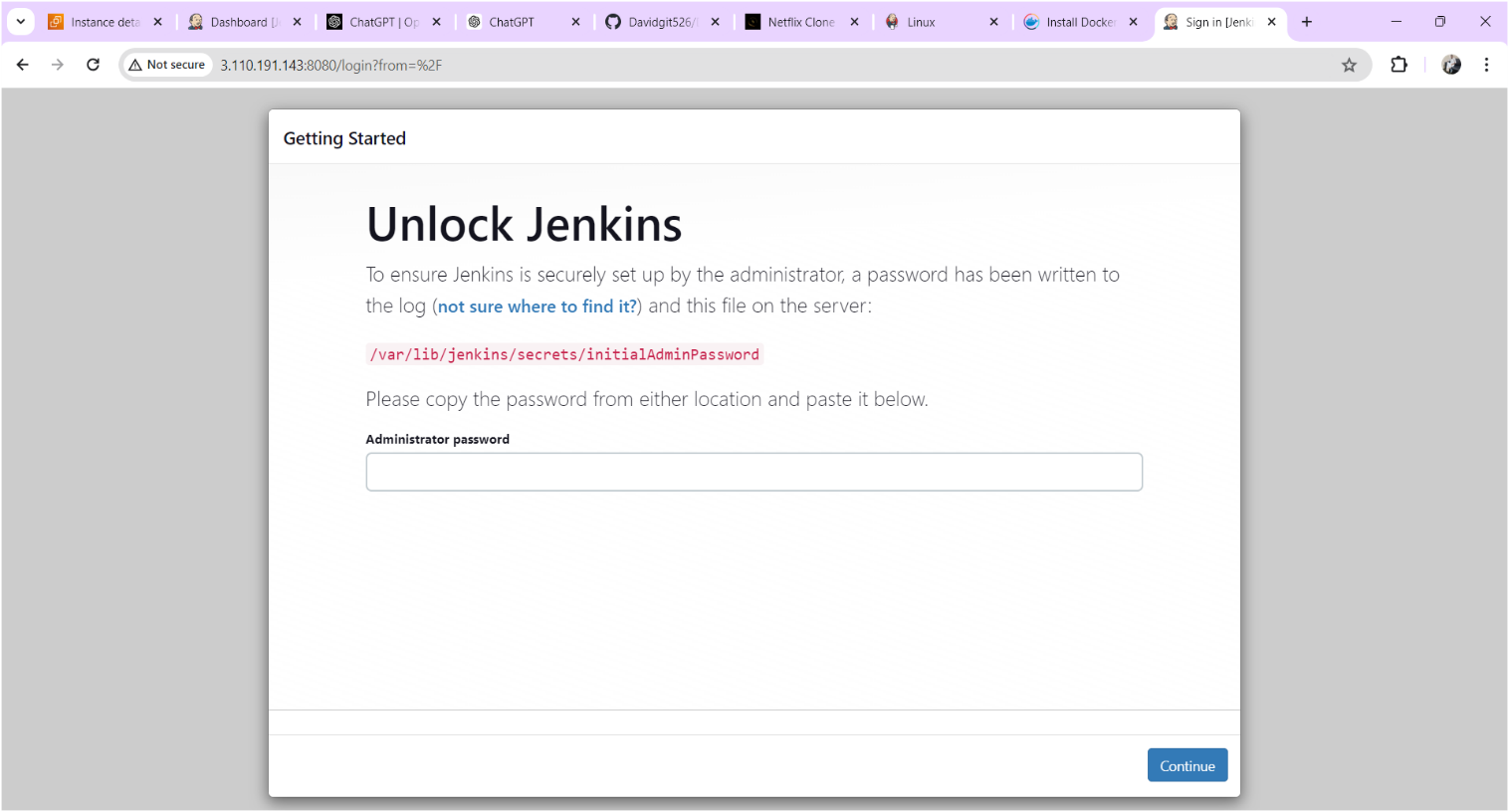
sudo systemctl status jenkins

Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

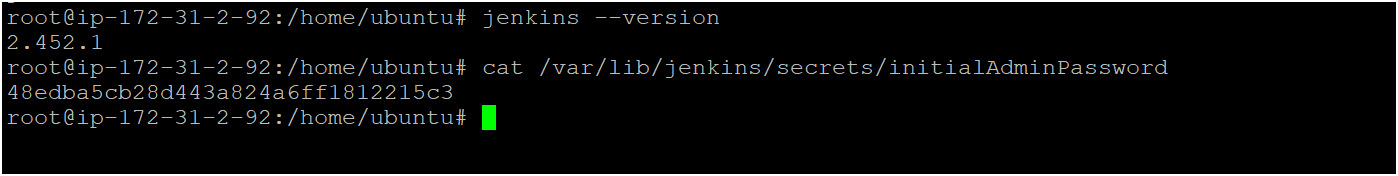
Now, grab your Public IP Address

<EC2 Public IP Address:8080>

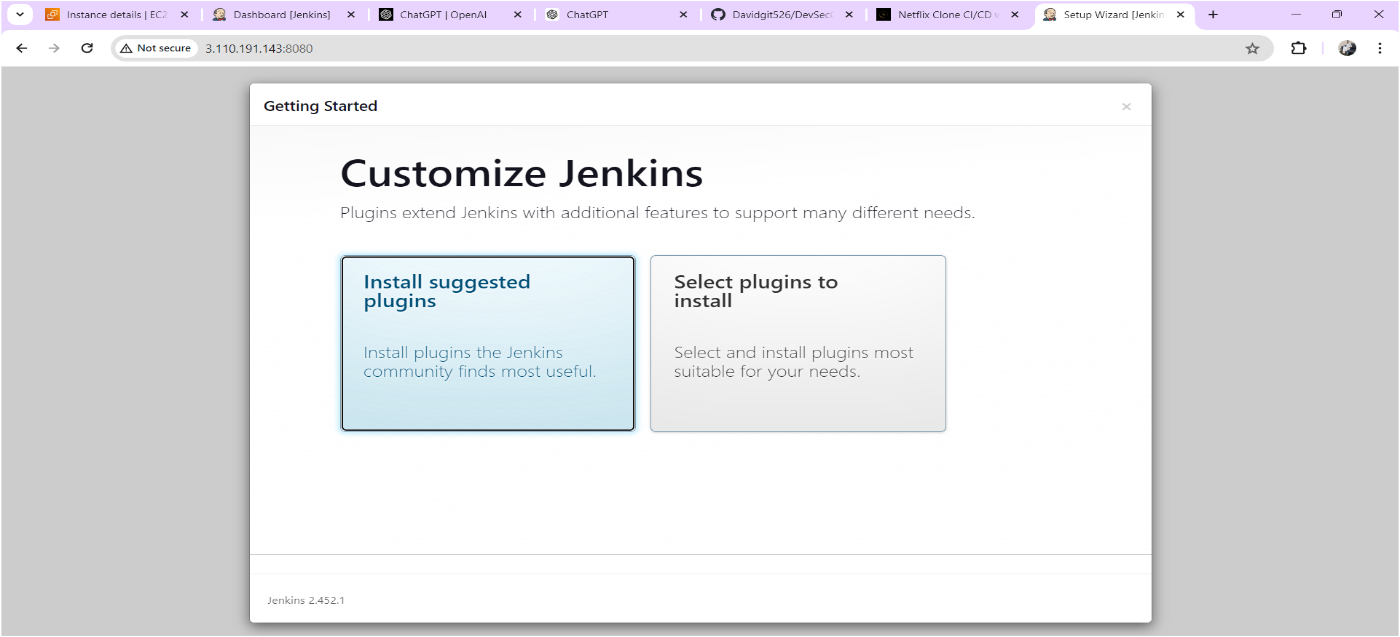




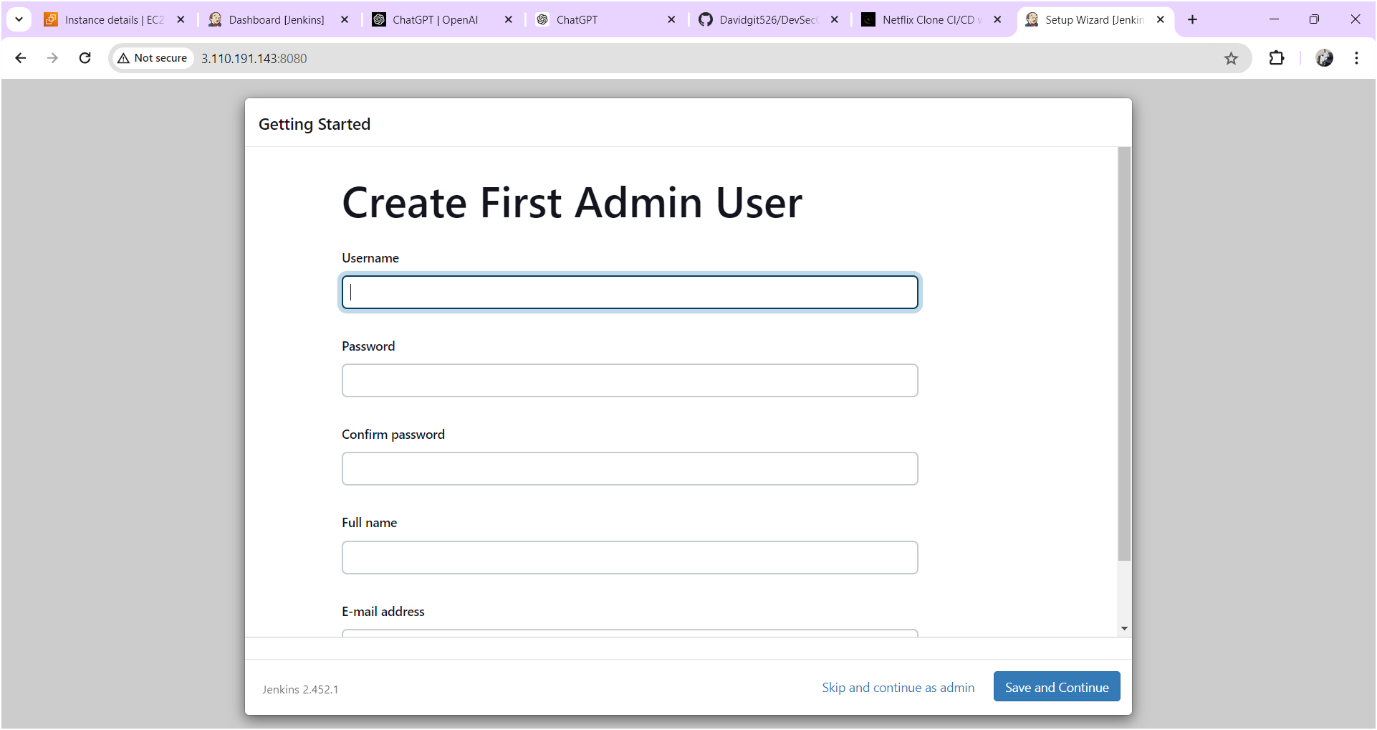
Copy the text which is mentioned in yellow colour box above and paste it using command “cat /var/lib/jenkins/secrets/initialAdminPassword” on your terminal after you will get below output.



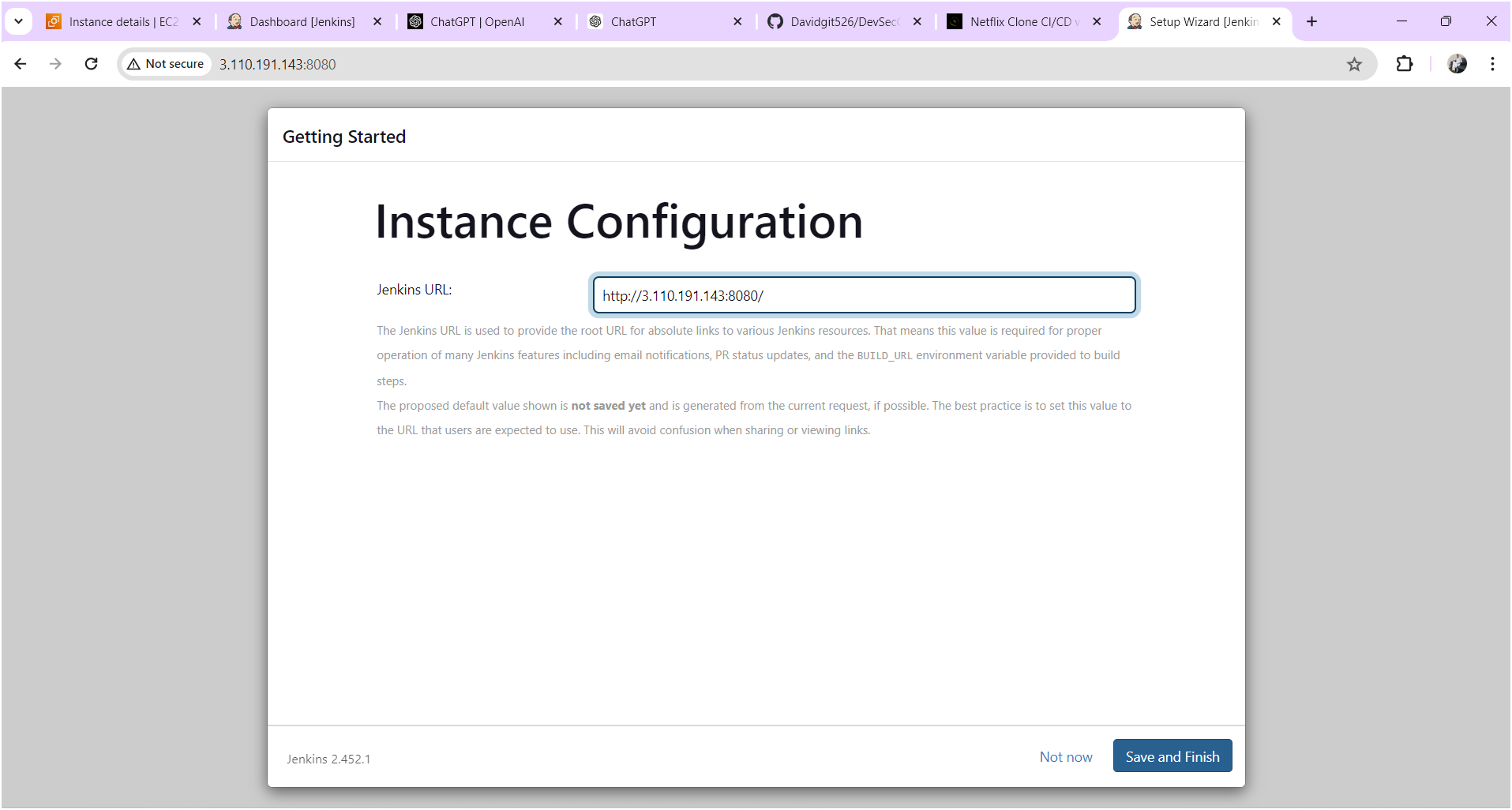
Copy the password for login to jenkins as shown above image



Click on Suggested plugins as shown above image after you will get output as shown below image



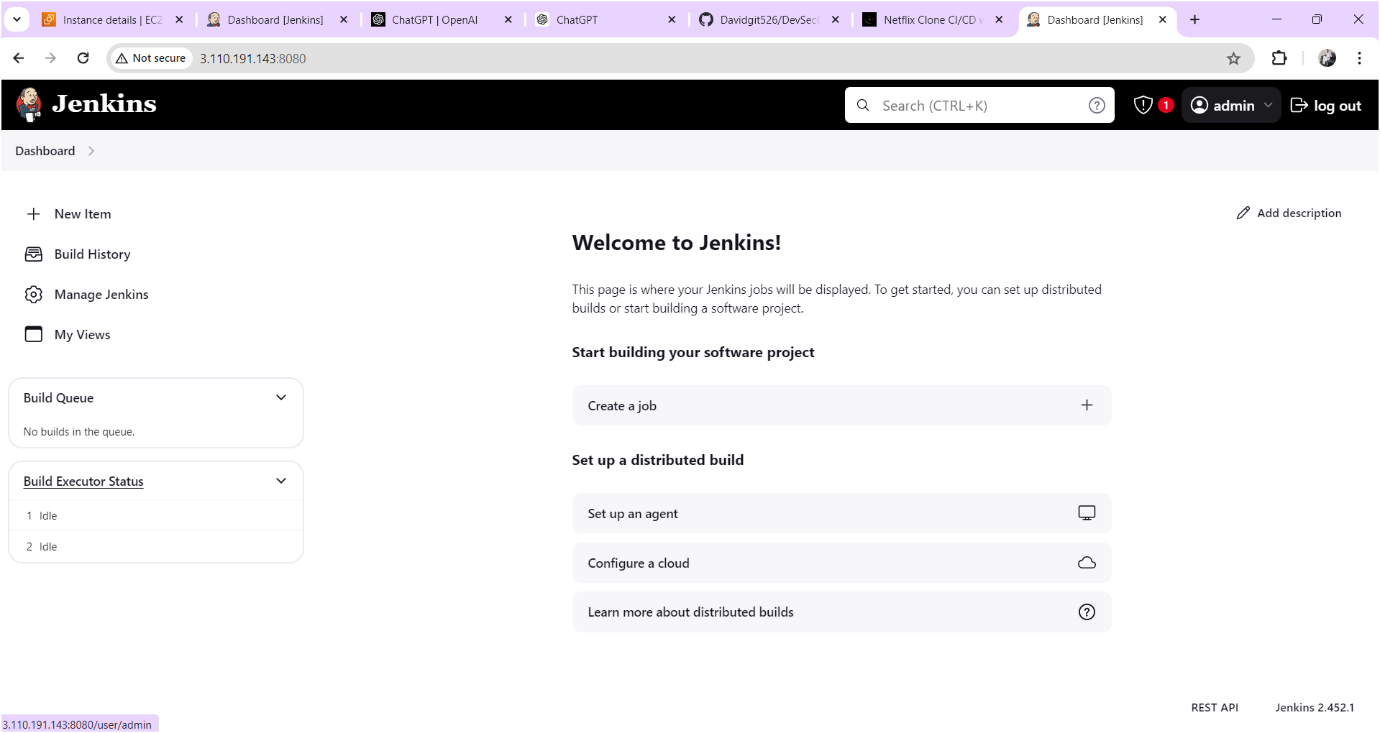
Click on skip and continue as admin



Click on save and finish

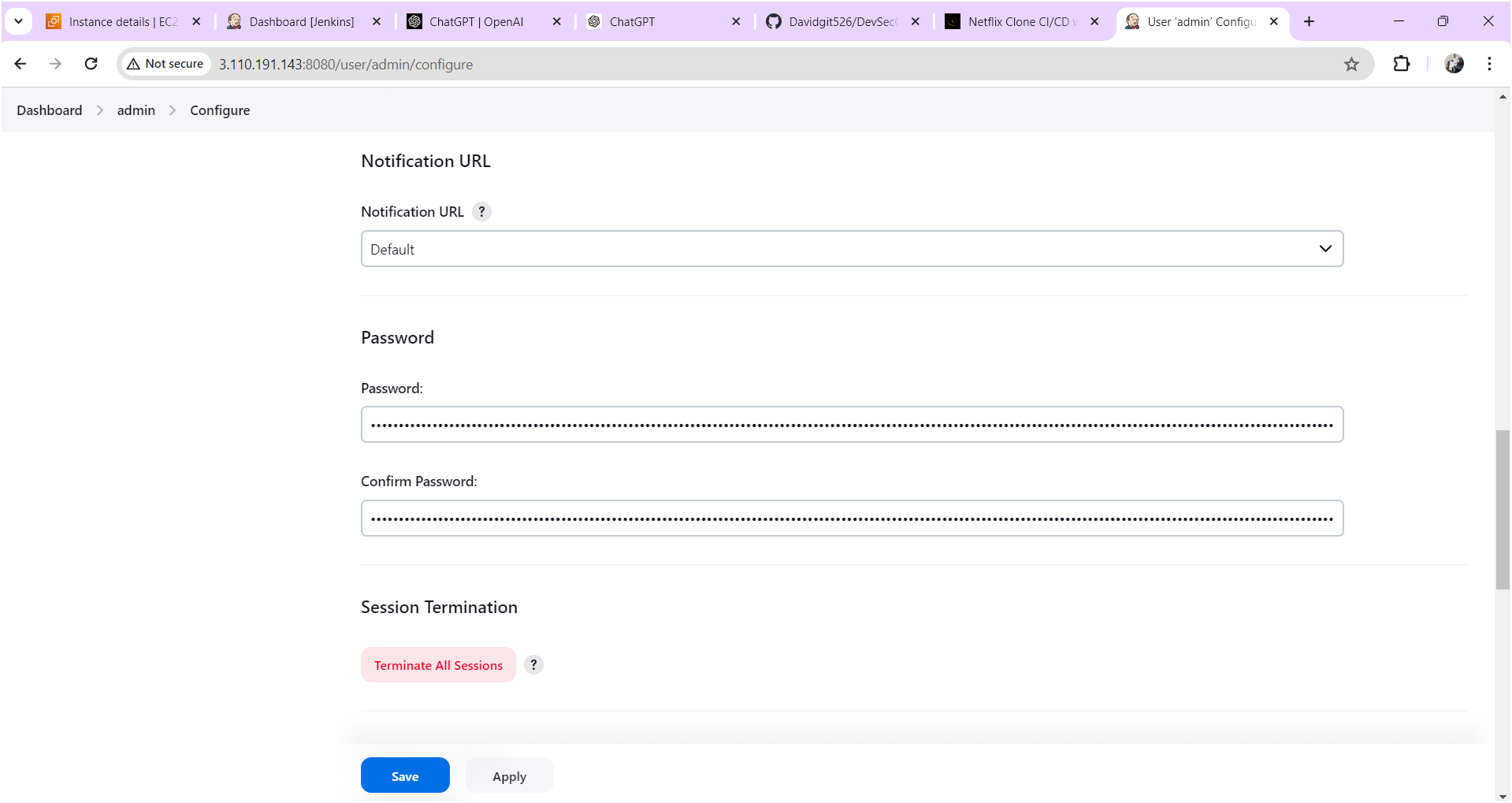


Click on start using jenkins

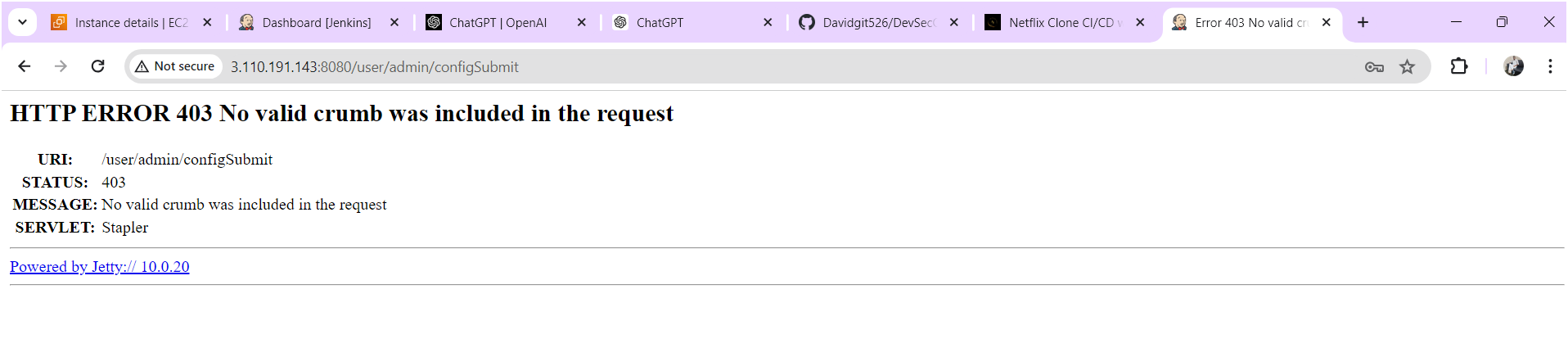


Now you should change the password for that

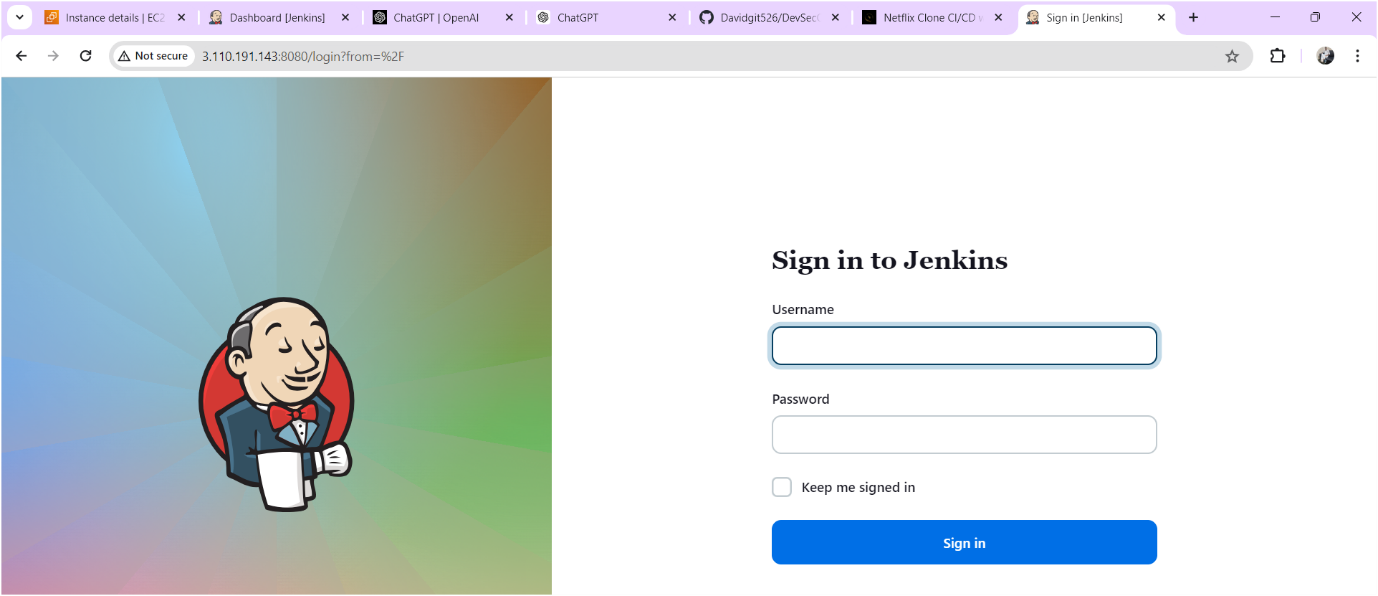
Click on admin🡪click on configure find password there you can change the password whatever you want. Output sown as below image



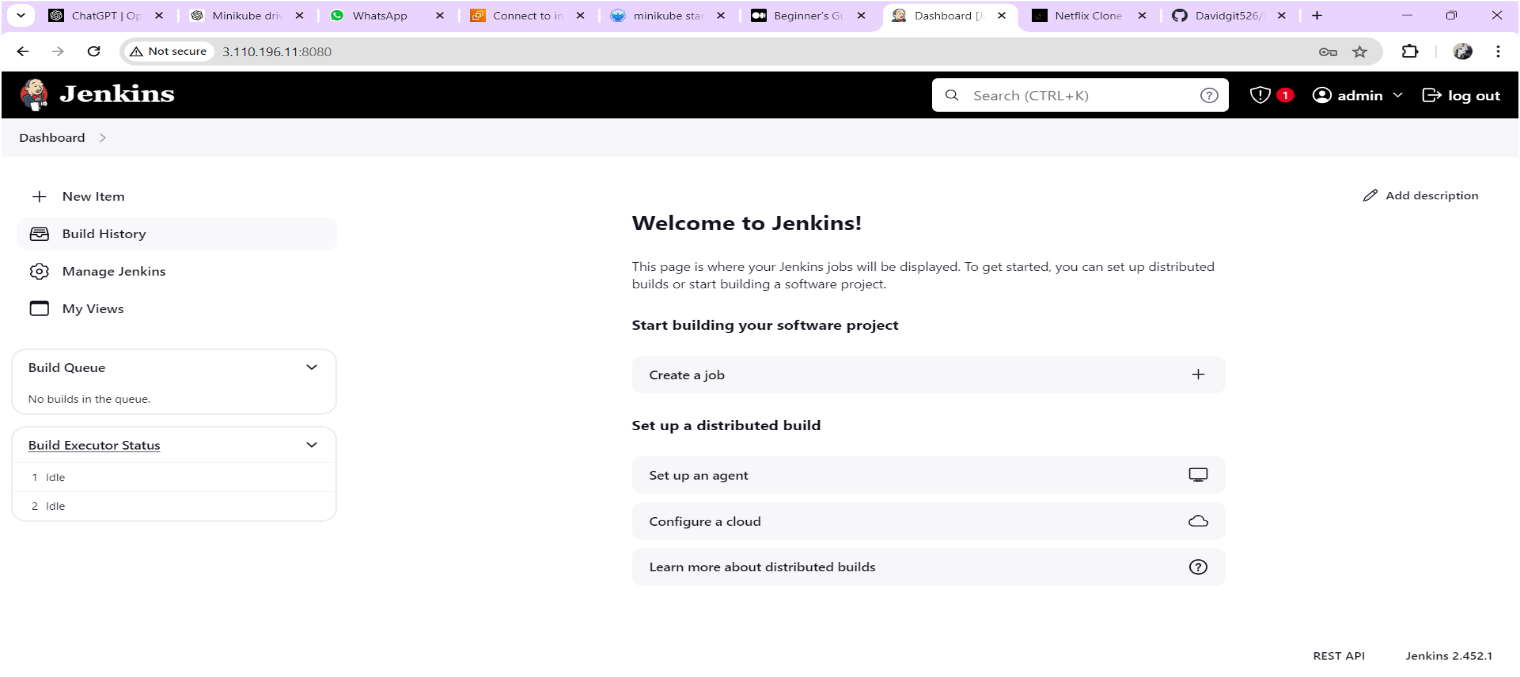
After changing the password click on **apply** and **save** after output will come as shown below image.



Now refresh the page after output will come as shown below image



Login to the jenkins with your updated credentials after you will get jenkins home page as shown below.



**Install SonarQube:**

Launch an Ubuntu(22.04) T2 medium Instance or you can install on jenkins server only (Note: If jenkins server running slowly you need to create another instance and install sonarqube.)

**Set up Docker on the EC2 instance:**

sudo apt-get update

sudo apt-get install docker.io -y

sudo usermod -aG docker $USER # Replace with your system's username, e.g., 'ubuntu'

newgrp docker

sudo chmod 777 /var/run/docker.sock

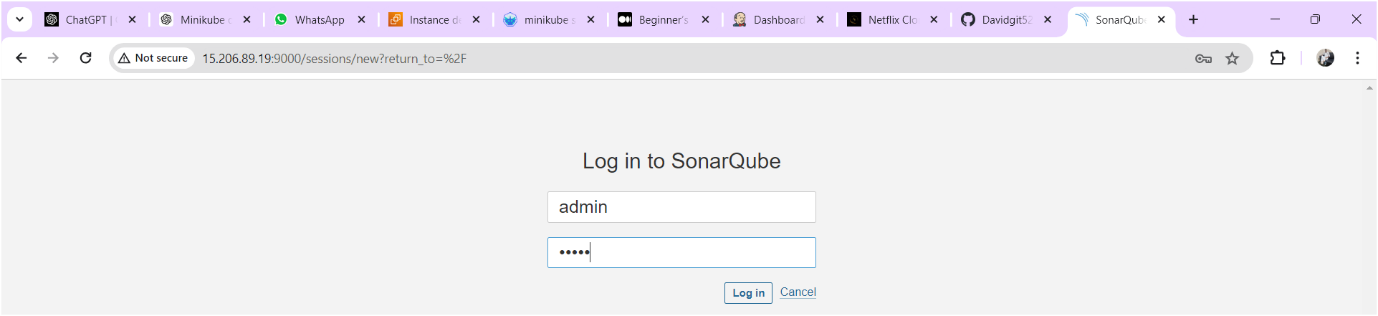
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community



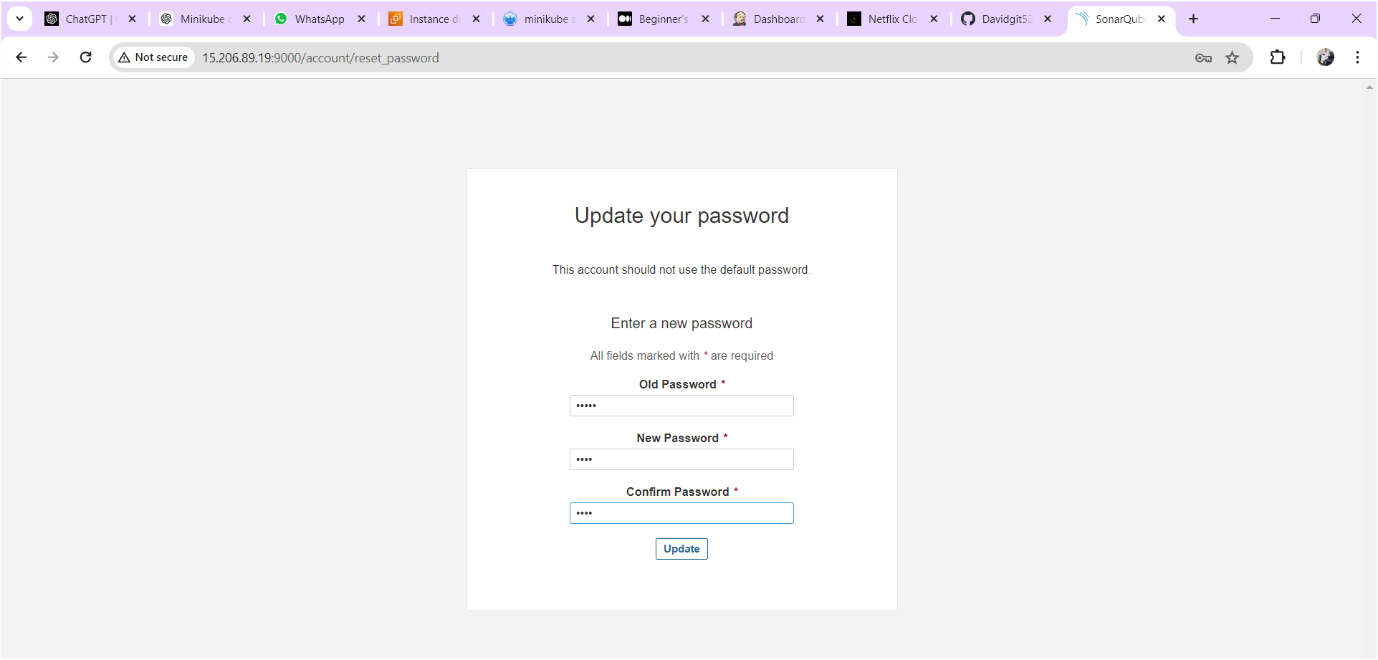
To access:

Need to open port number 9000 in security groups

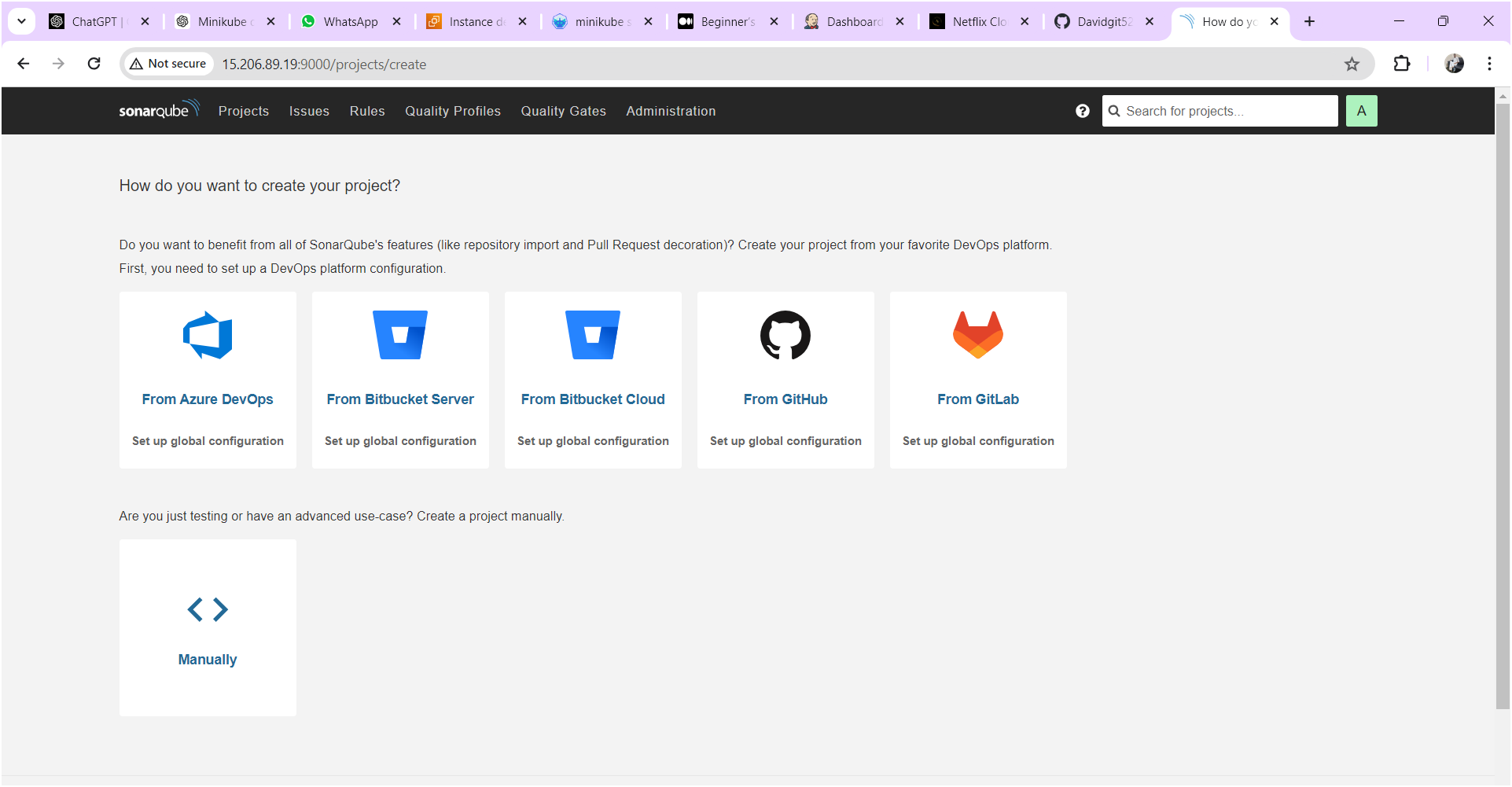
publicIP:9000 (by default username & password is admin)



After login you need to change the password as shown below image

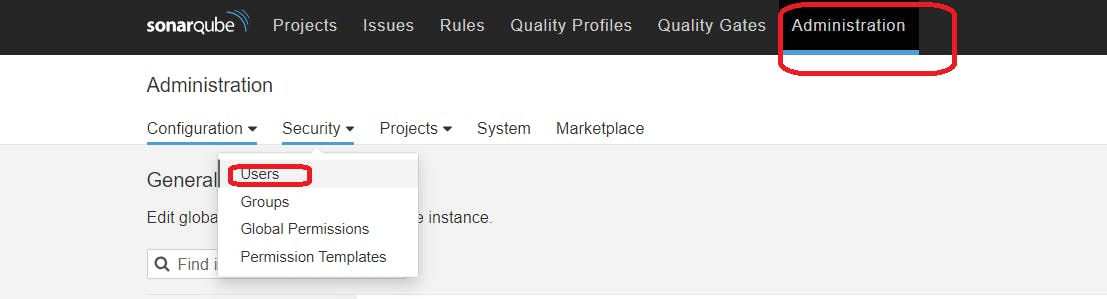


Sonarqube home page is as shown below

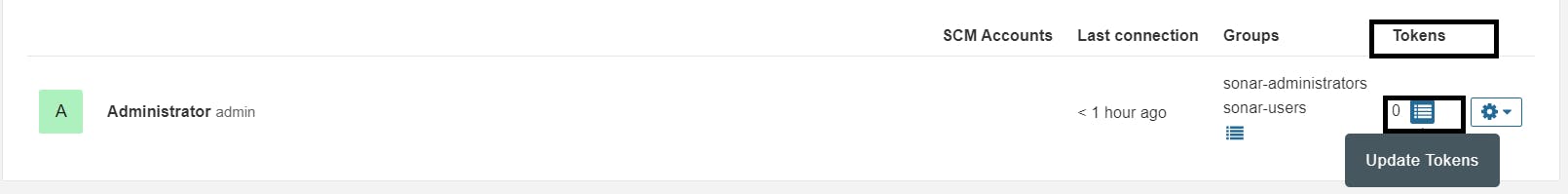


## **Configure Sonar Server in Manage Jenkins**

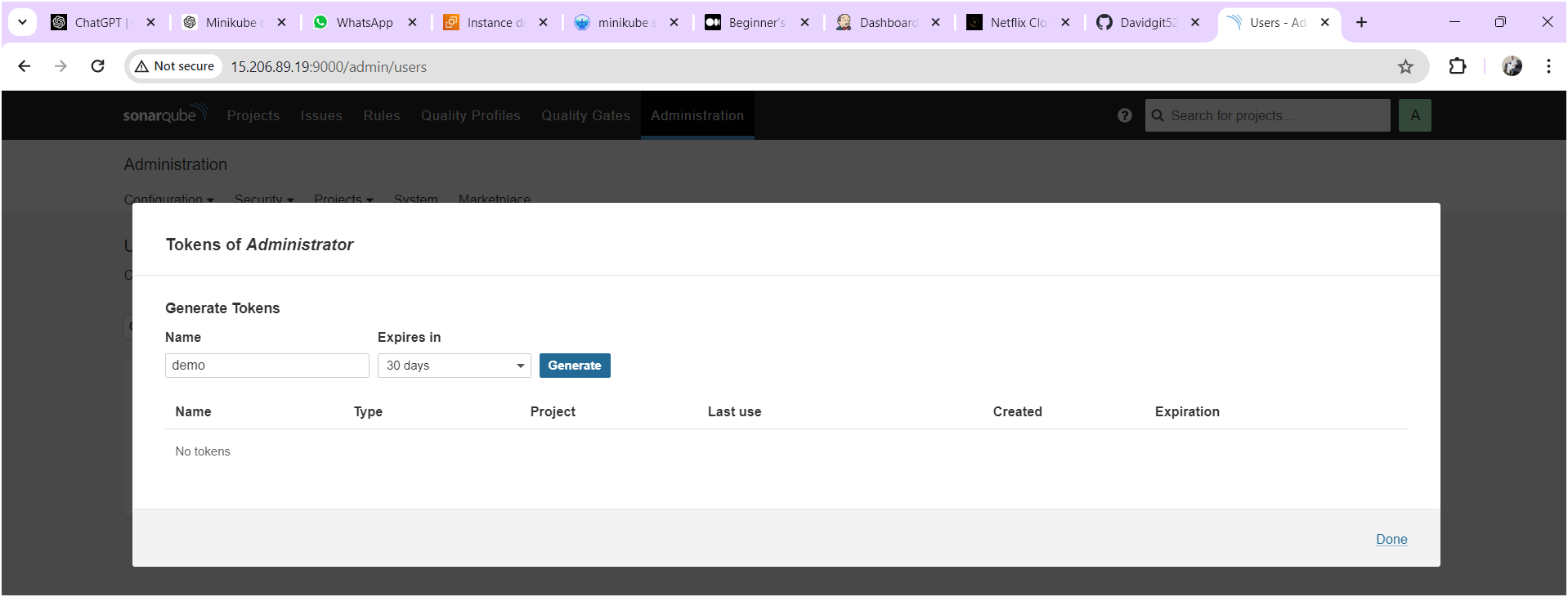
Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token



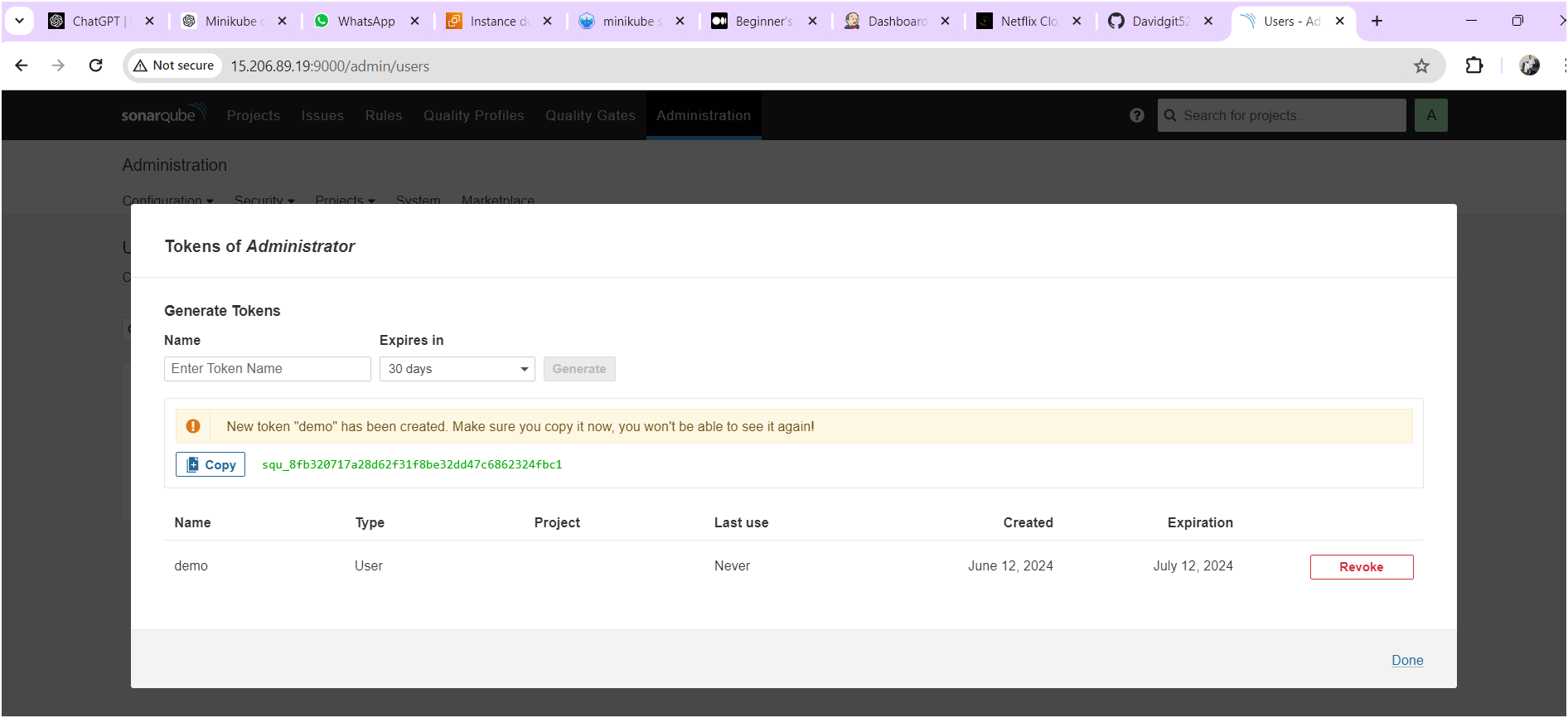
click on update Token



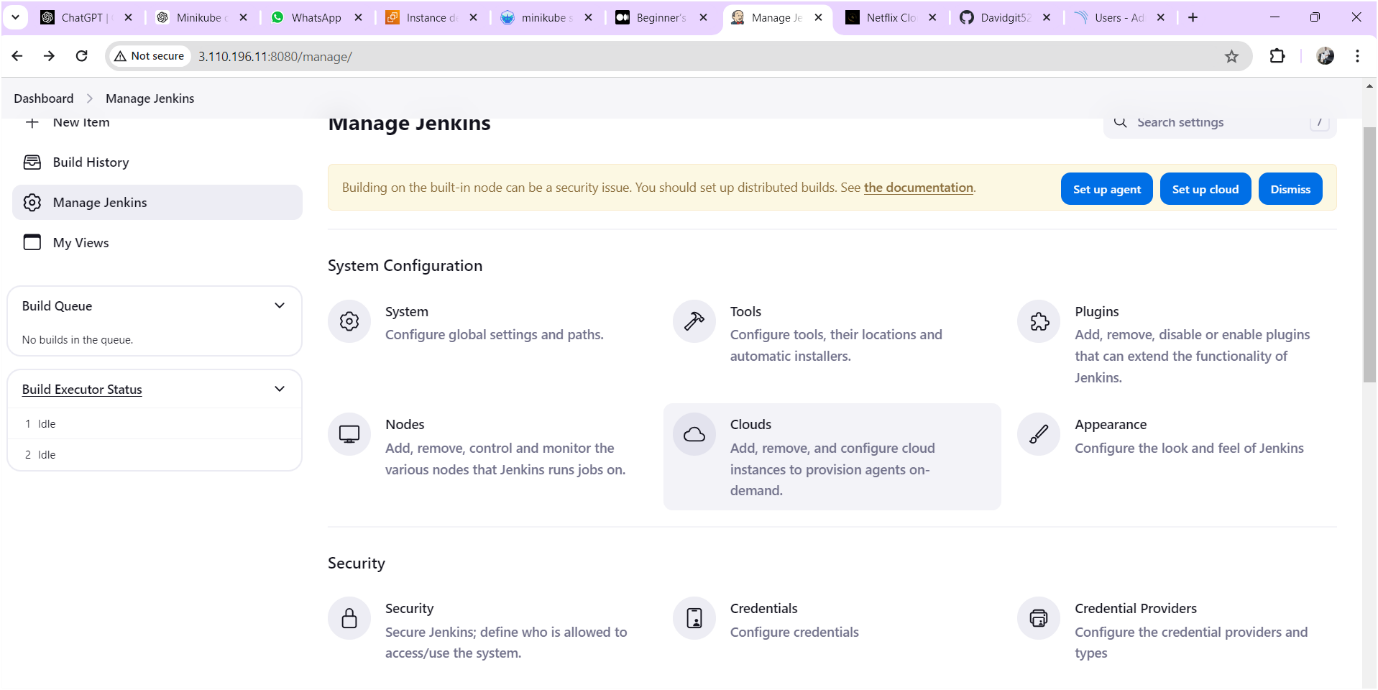
Give name for token and click on generate then token will generate and keep it save some other place for future use.



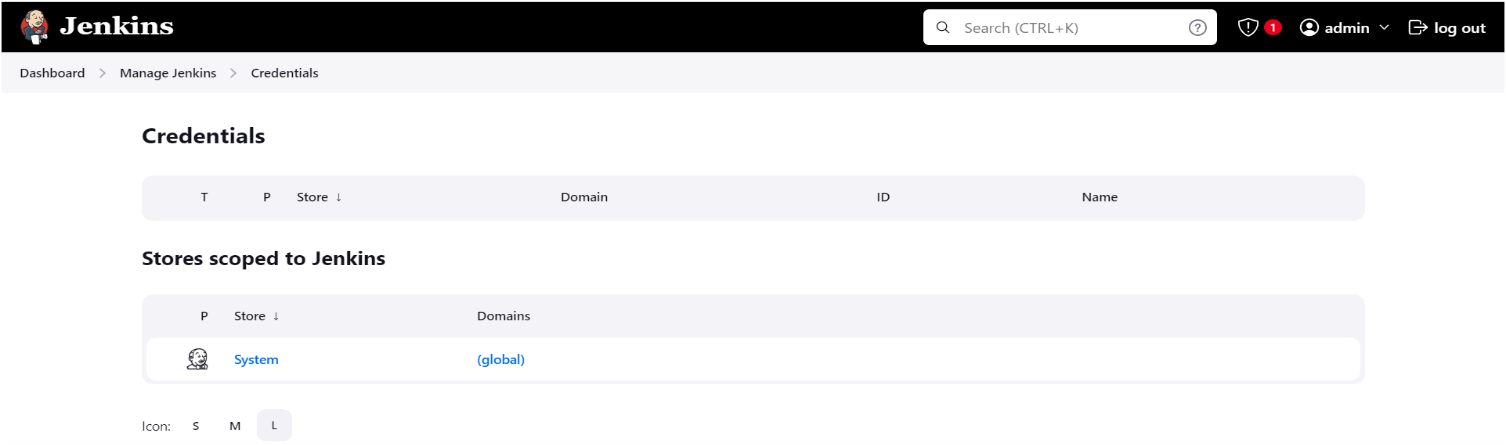
Copy the token save it at somewhere



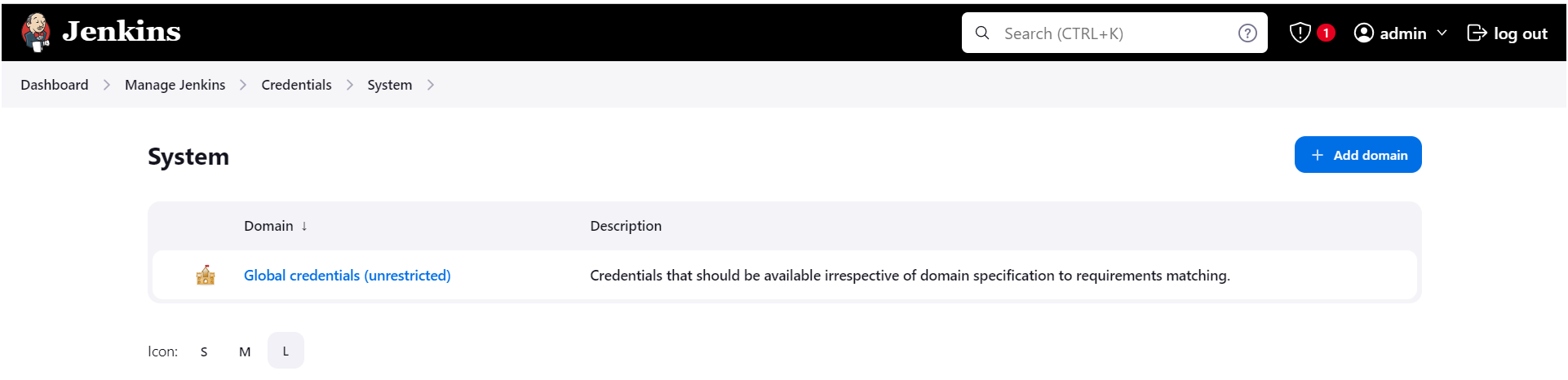
Goto Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this.



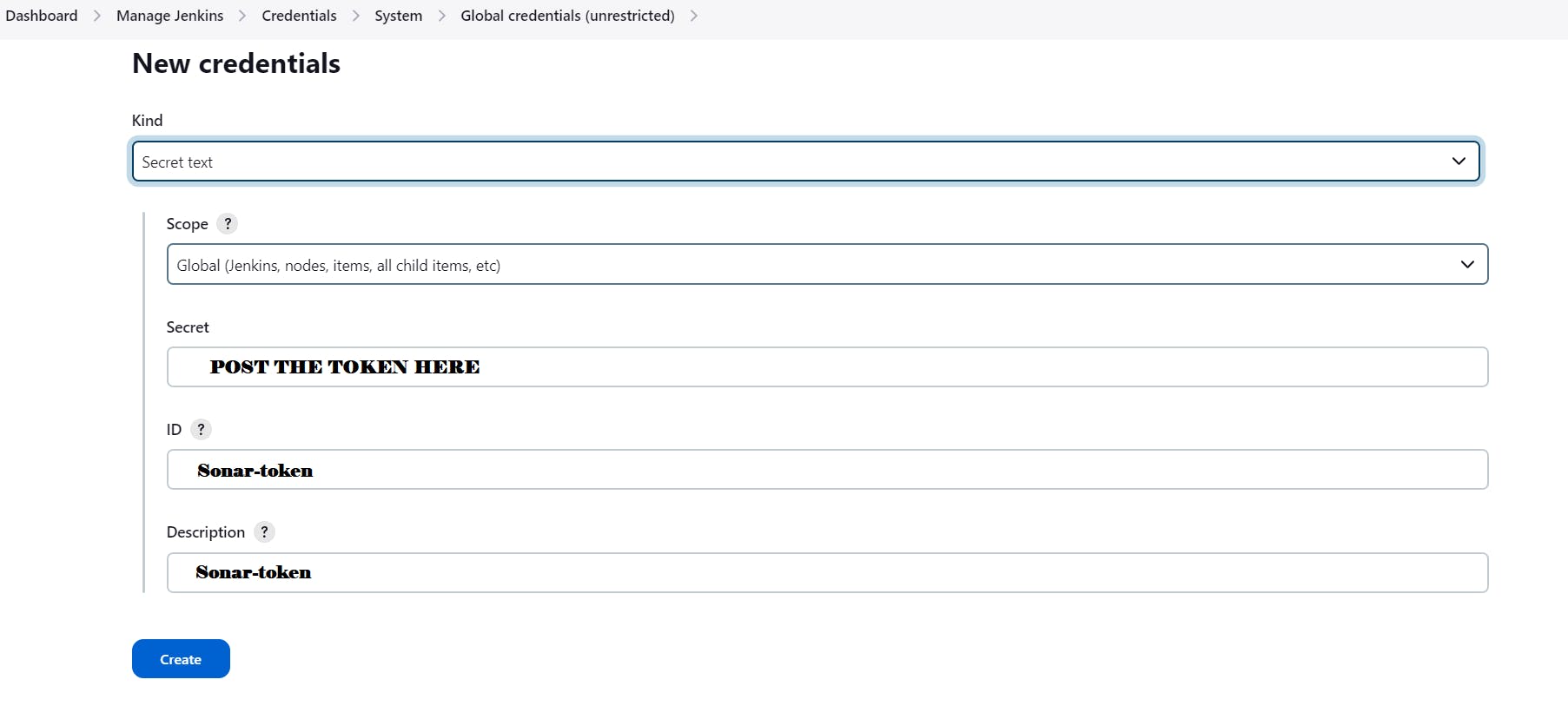
Click on “system” as shown below image



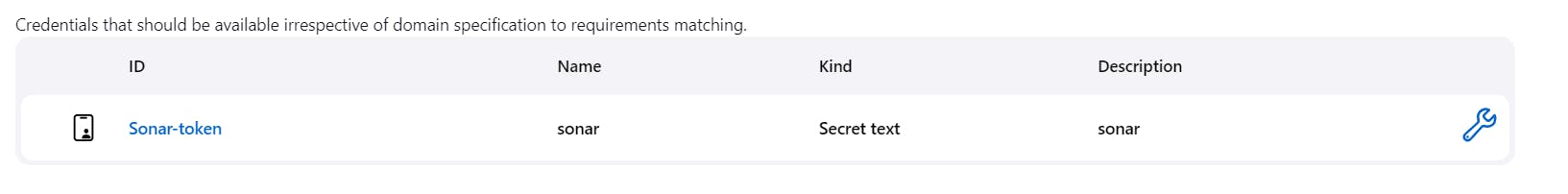
Click on “Global Credentials” as shown below image



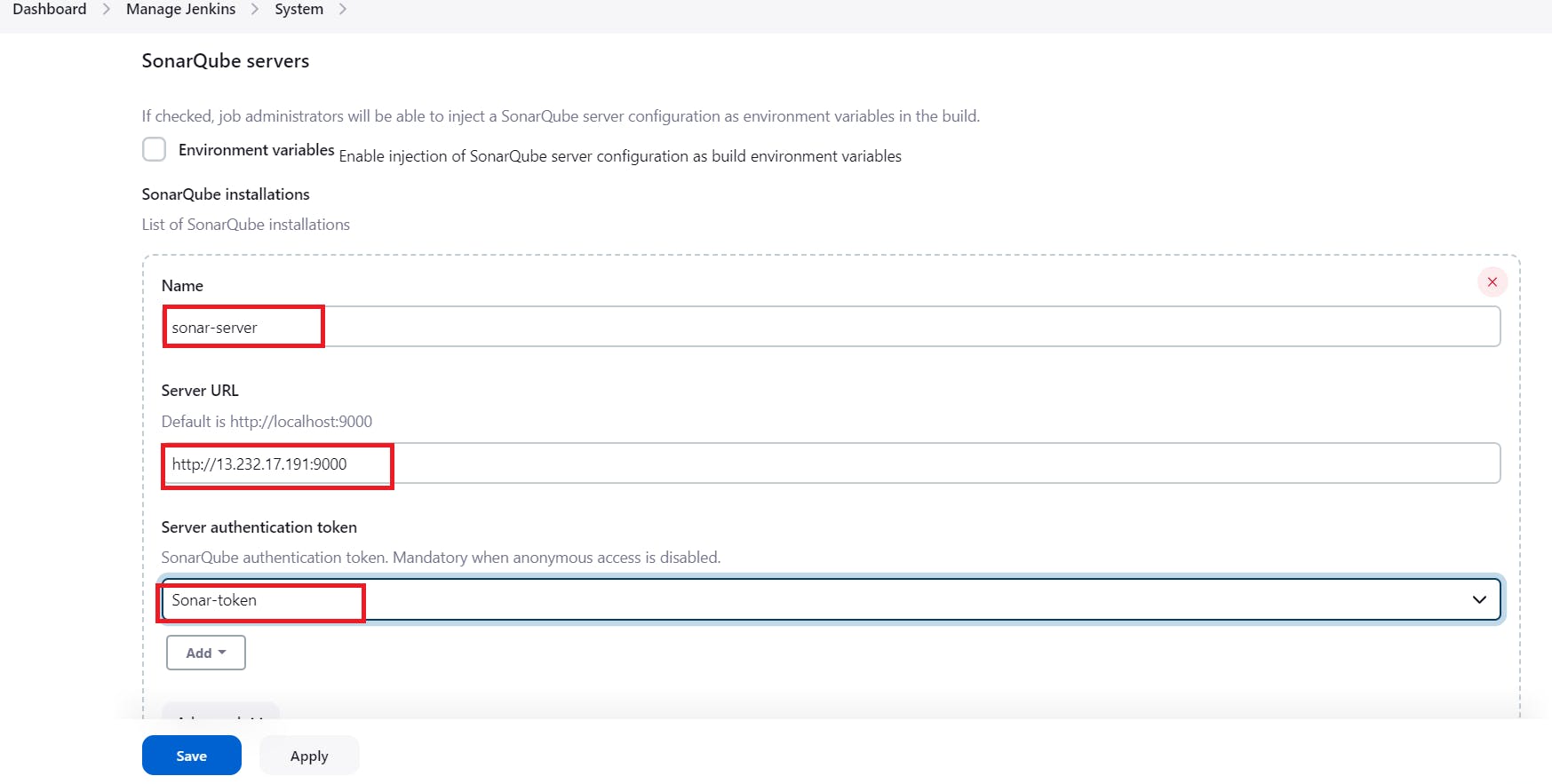
Here paste the token and give id and description as “sonar-token like below image.



You will this page once you click on create



Now, go to Dashboard → Manage Jenkins → System and Add like the below image.

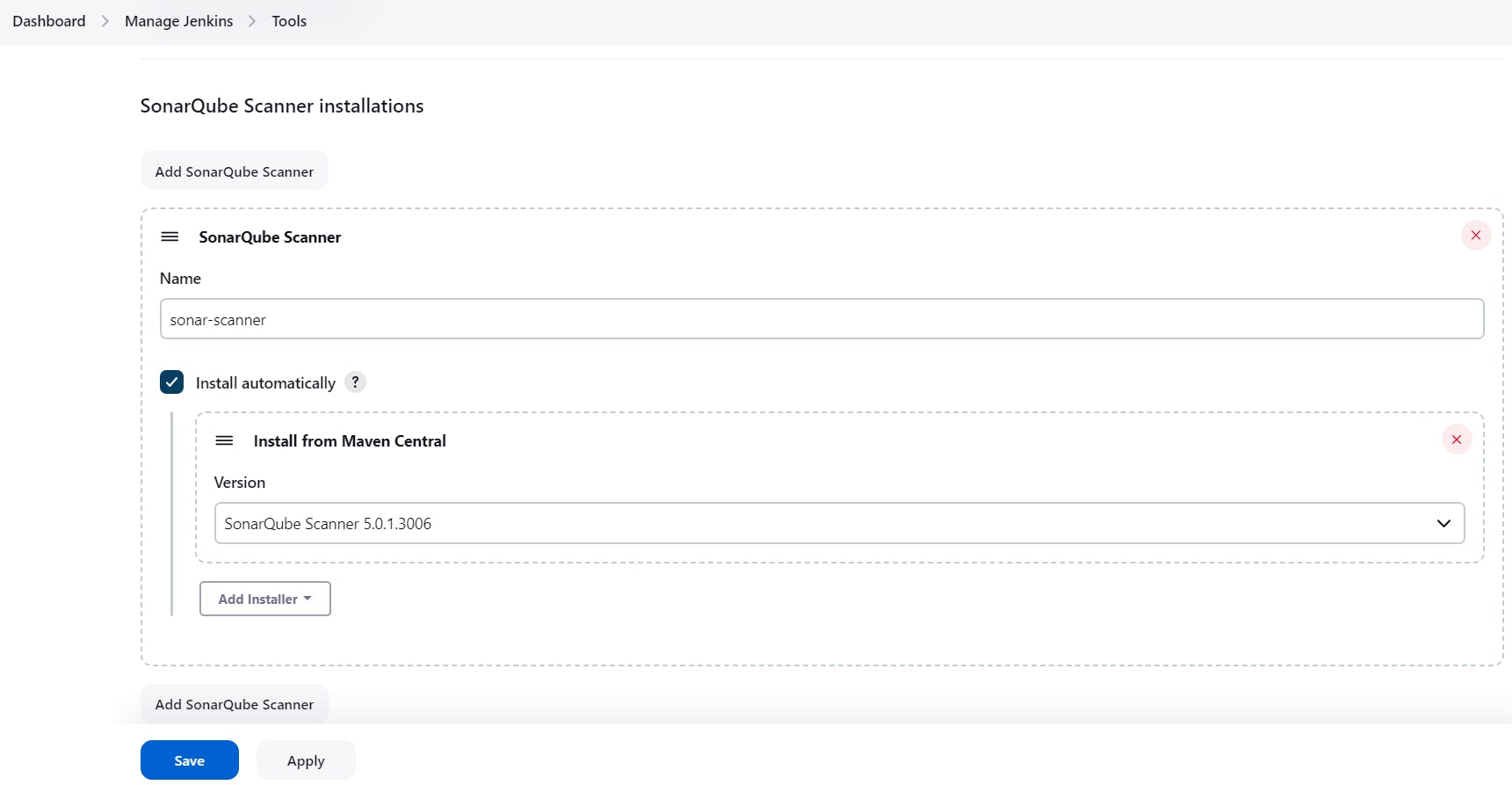


Click on Apply and Save

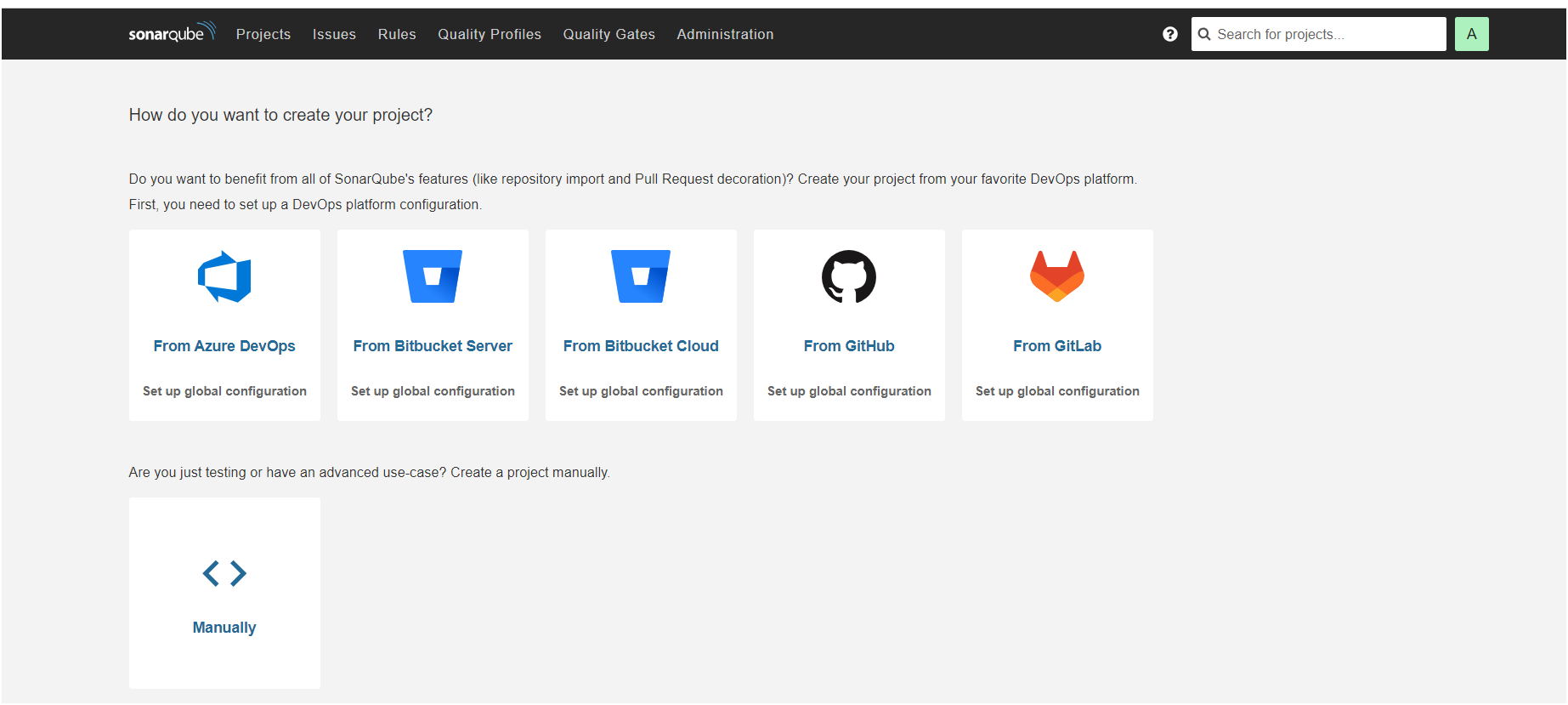
**The Configure System option** is used in Jenkins to configure different server

**Global Tool Configuration** is used to configure different tools that we install using Plugins

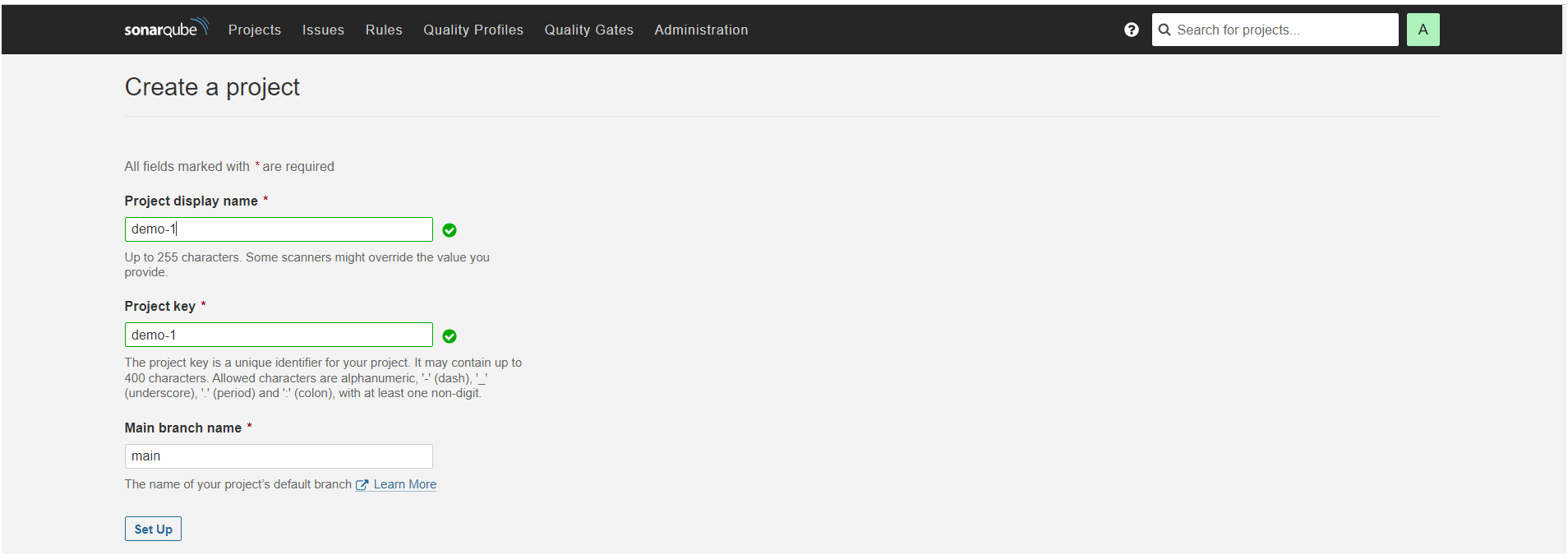
We will install a sonar scanner in the tools.



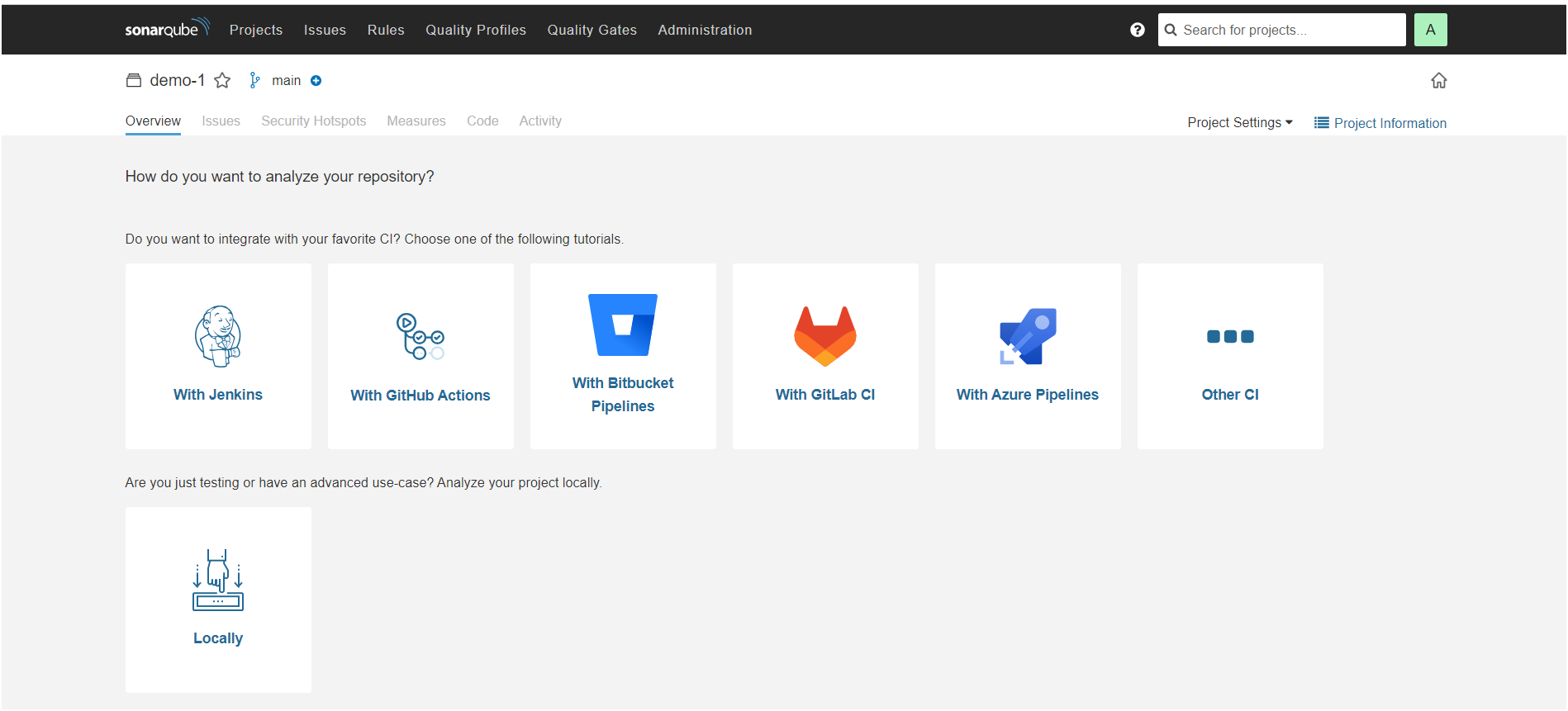
Next create a project in sonarqube server go to projects🡪manually as shown below image.



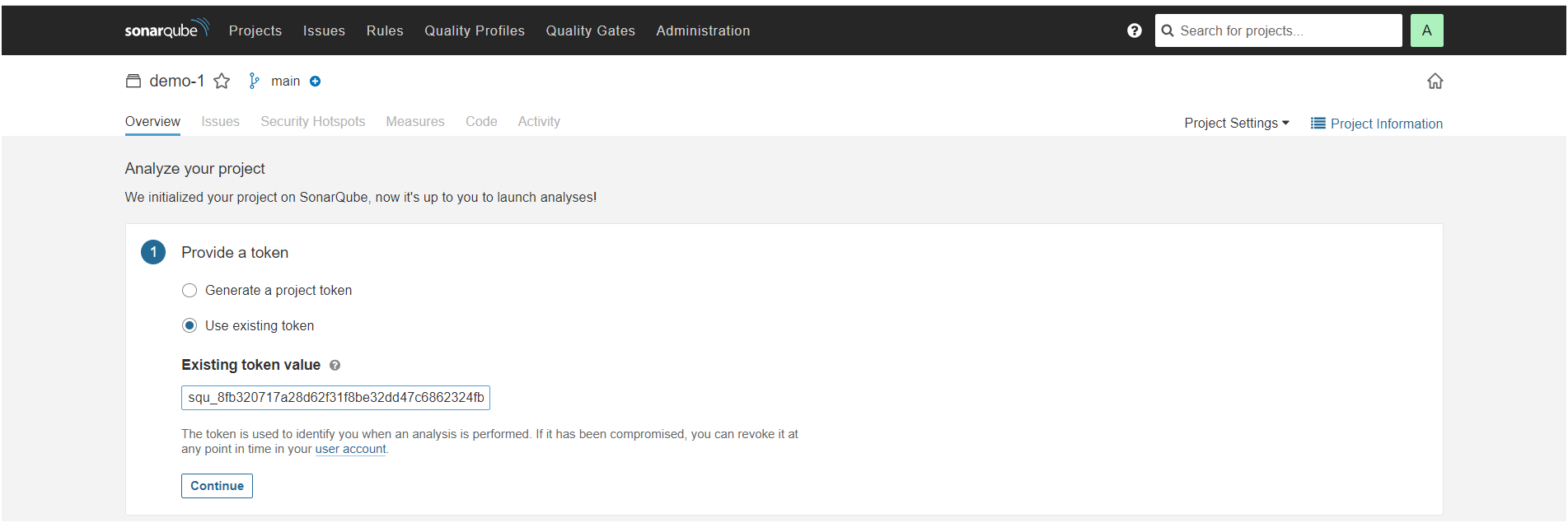
After clicking on “manually” you will get as shown below image



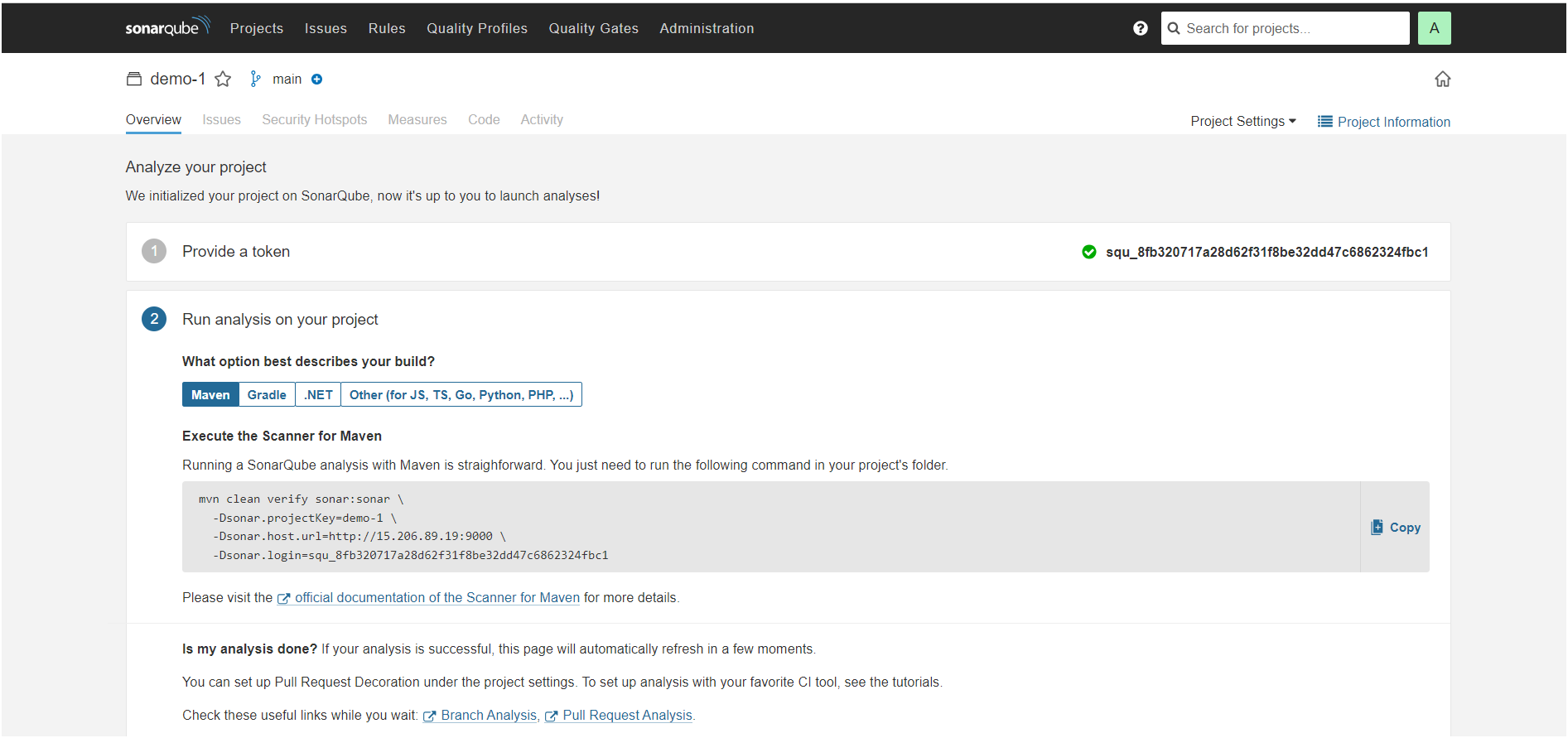
After click on “set up” you will get as shown below image.



Once clicked on “locally” you will get next page as shown below after click on existing token and paste token here which we have generated a token after click on “continue”.



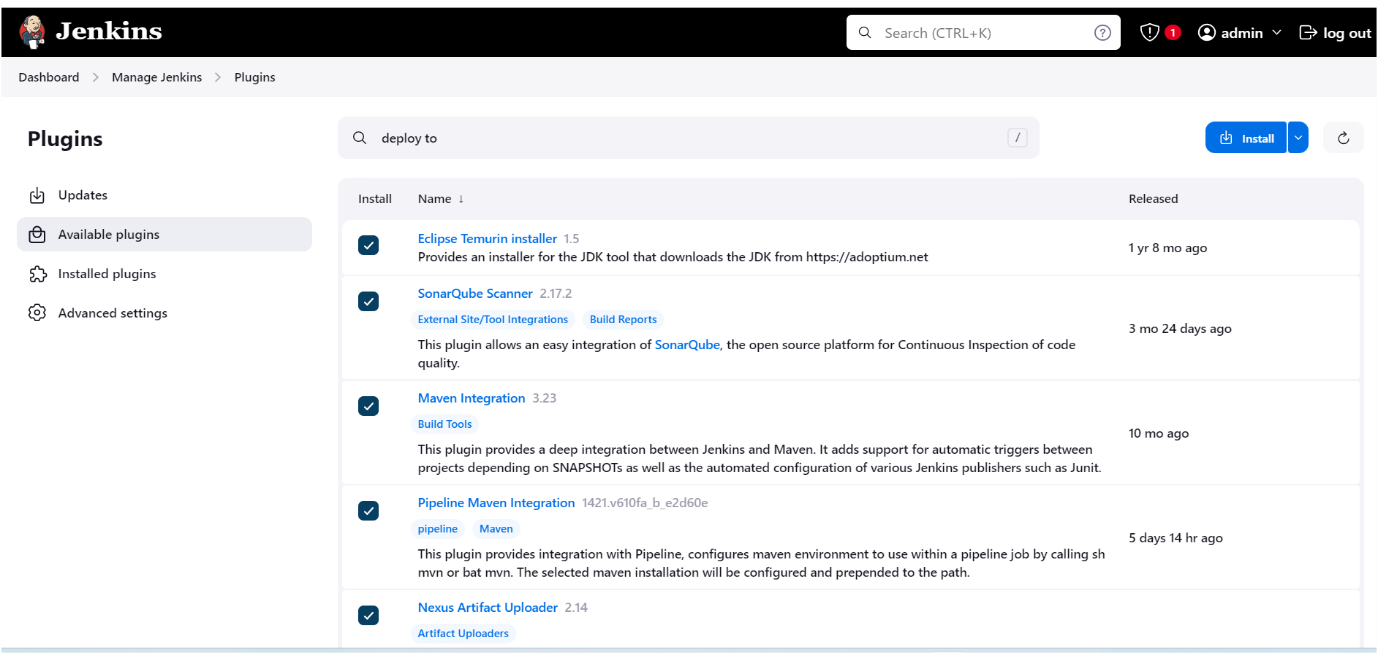
Next you will get page as shown below image

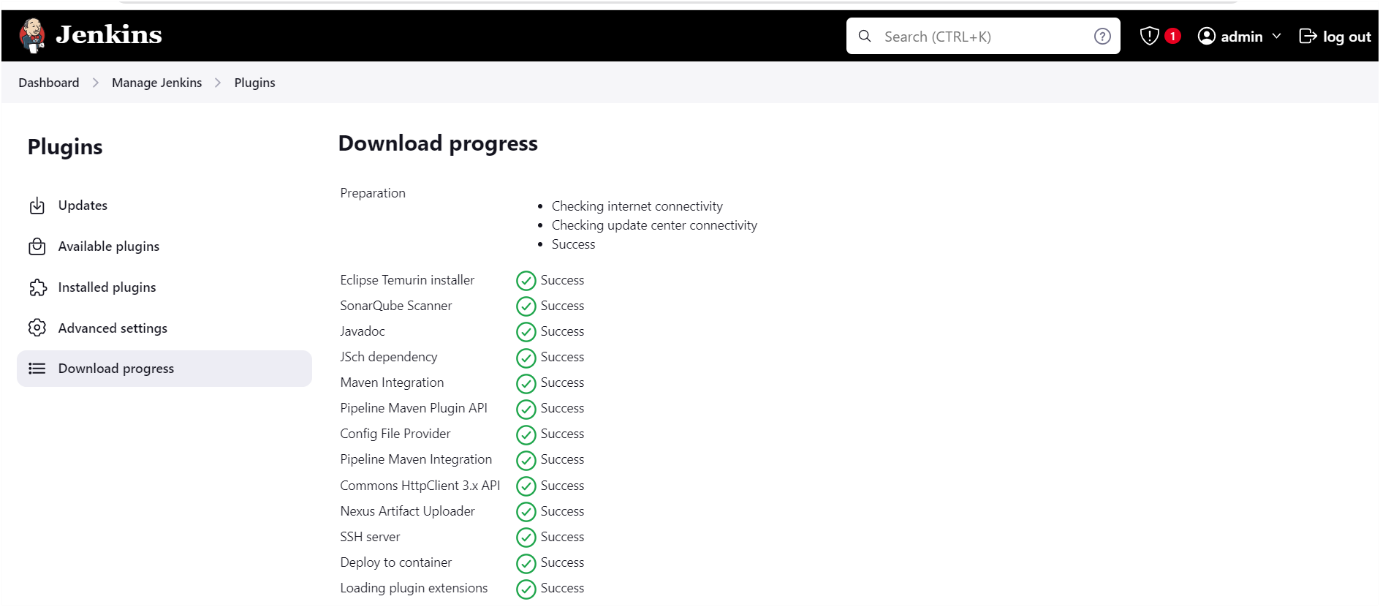


## **Install Plugins**

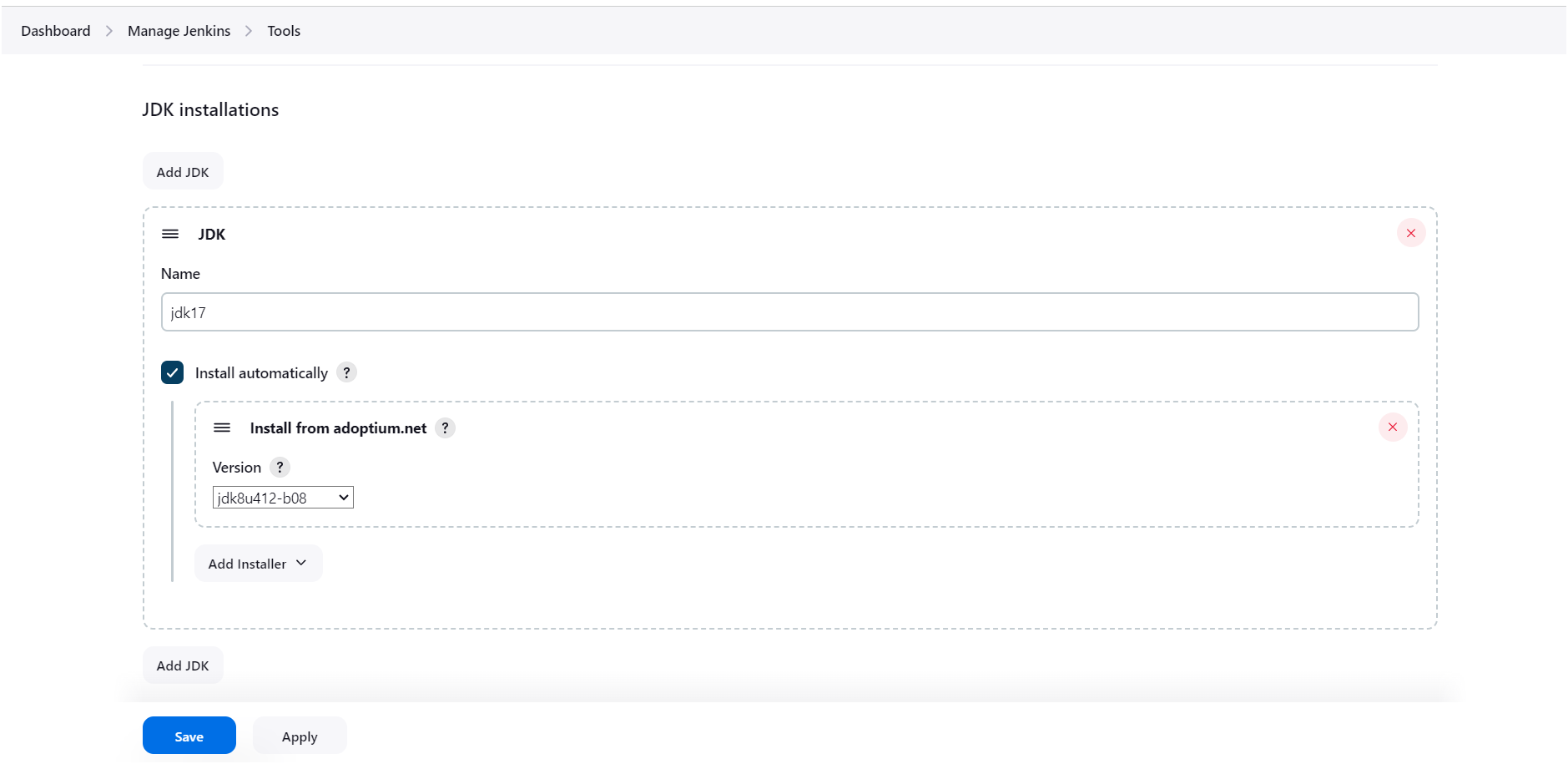
1. Eclipse Temurin Installer
2. SonarQube Scanner
3. Maven Integration
4. Pipeline Maven Integration
5. Deploy to Container
6. Nexus Artifact Uploader

**To install plugins, go to Dashboard🡪Manage Jenkins🡪Available Plugins**

****

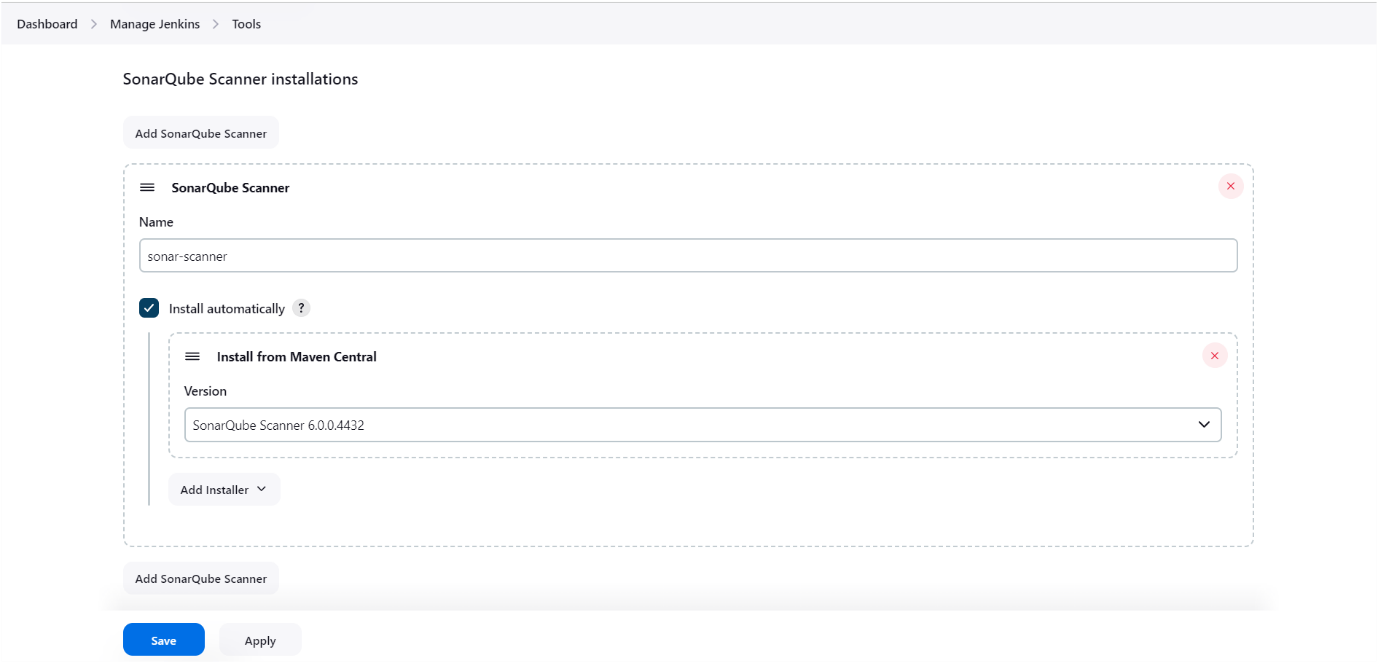
****

Configure **jdk** go to **manage Jenkins🡪tools** as shown below images



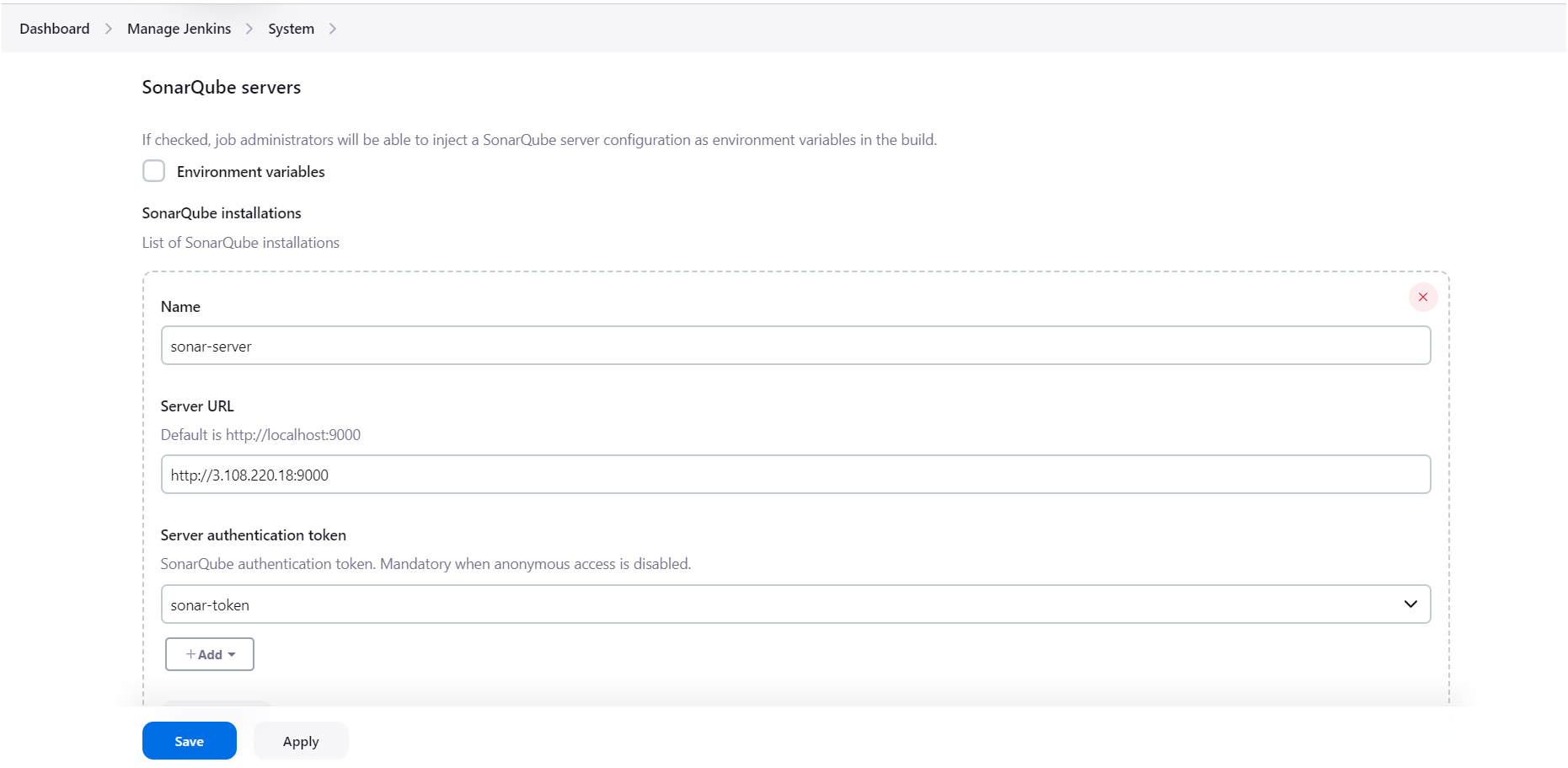
Click on apply and save☝️

Configure **Sonarqube go to manage Jenkins🡪tools** as shown below image👇



Click on apply and save☝️

Configure **Sonarqube go to manage Jenkins🡪systems** as shown below image👇, in server URL you need to provide your sonarqube server URL

****

Click on apply and save☝️

Configure **maven** go to **manage jenkins🡪tools** as shown below image👇

****

Click on apply and save☝️

Before Configure nexus you have to deploy nexus application inside the container as shown below👇

**Install nexus server**

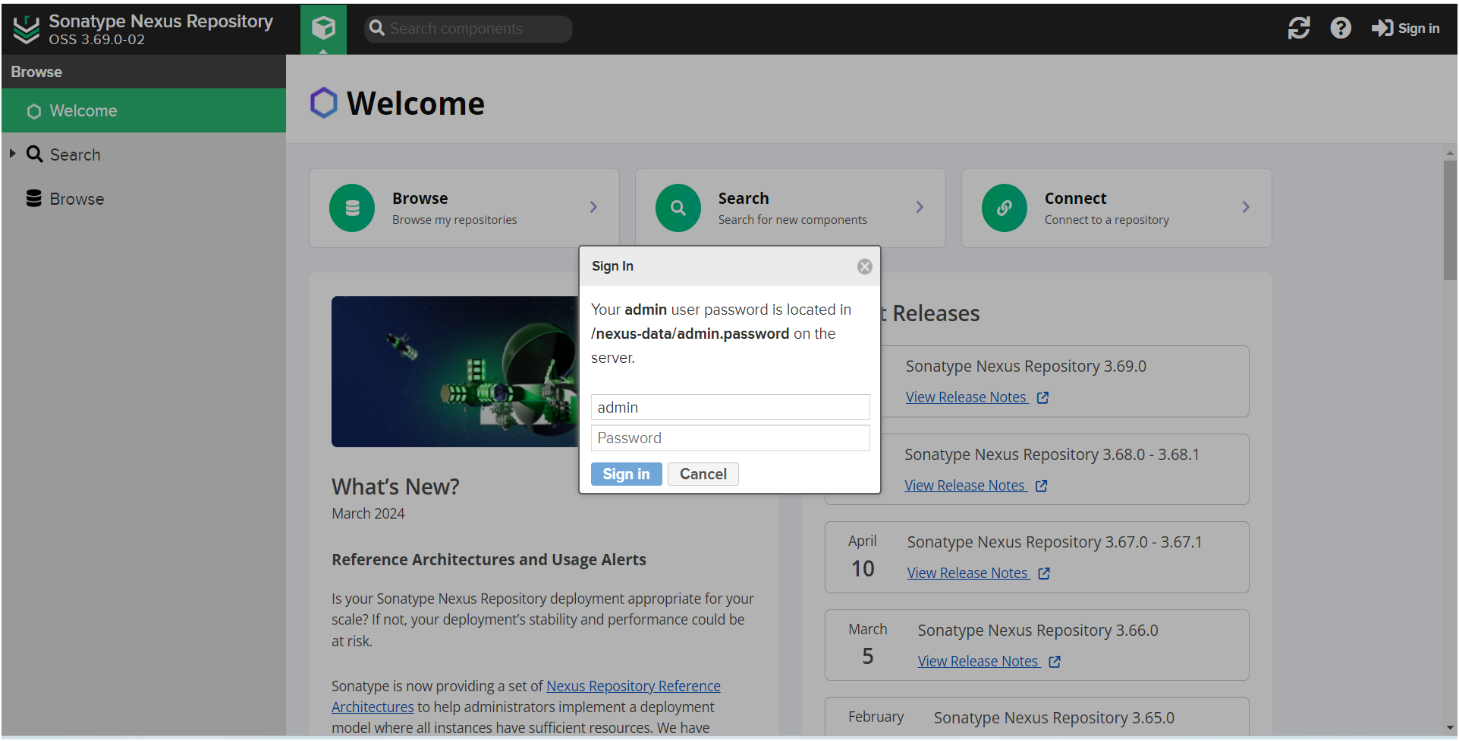
docker run -d --name nexus -p 8081:8081 sonatype/nexus3



To access:

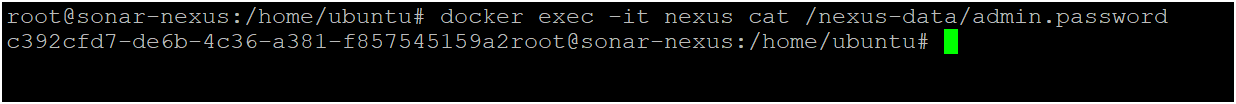
Need to open port number 8081 in security groups

publicIP:9000 by default username is admin and for password follow below steps 👇

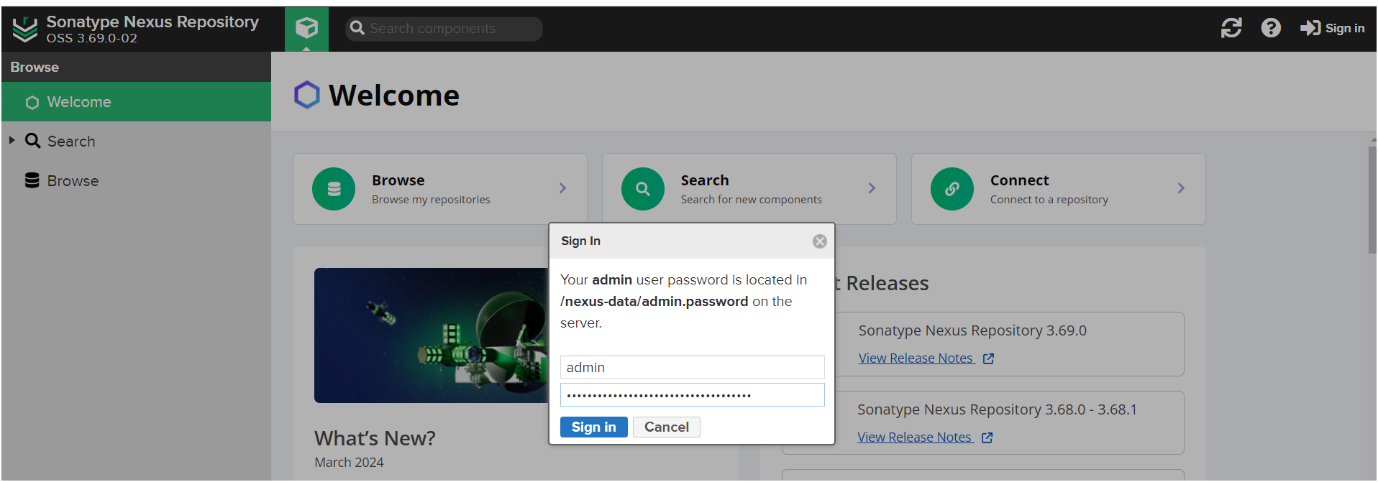


To get password execute follow commands as shown below👇

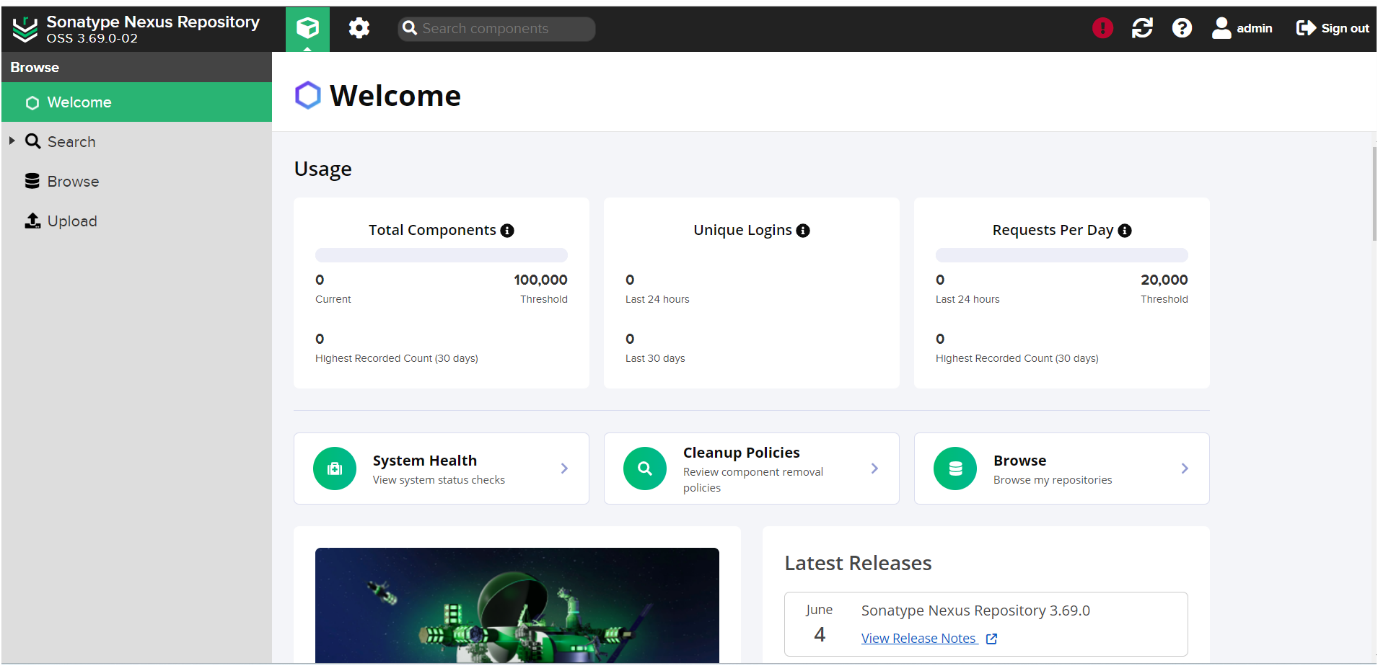
docker exec -it nexus cat /nexus-data/admin.password🡪 run that commands in your server where you have deployed nexus server as shown below👇



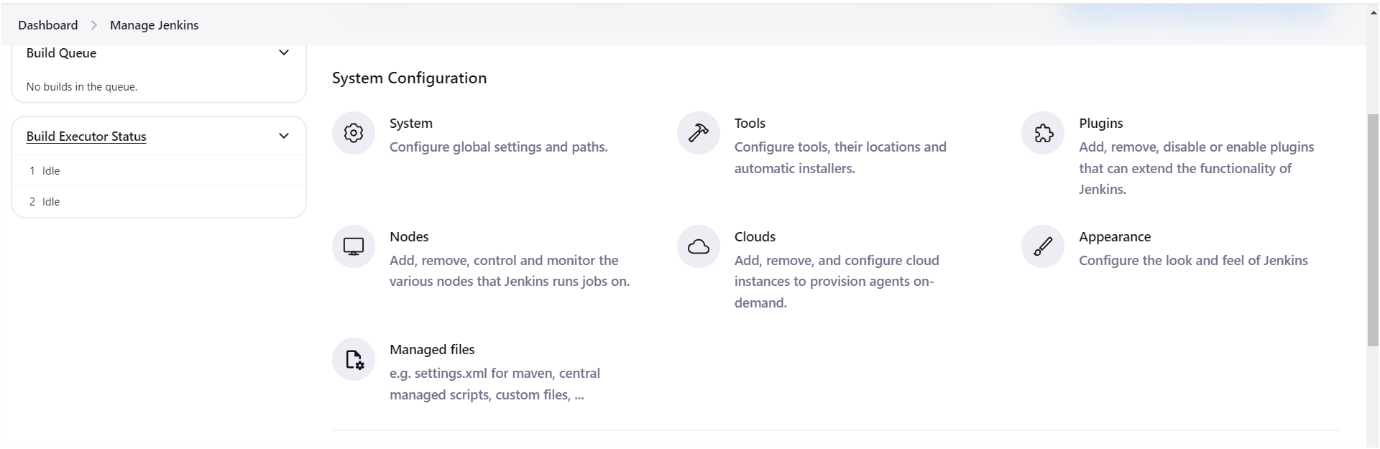
This is the password for nexus copy and login to the nexus server as shown below image



Nexus home page will appear as shown below



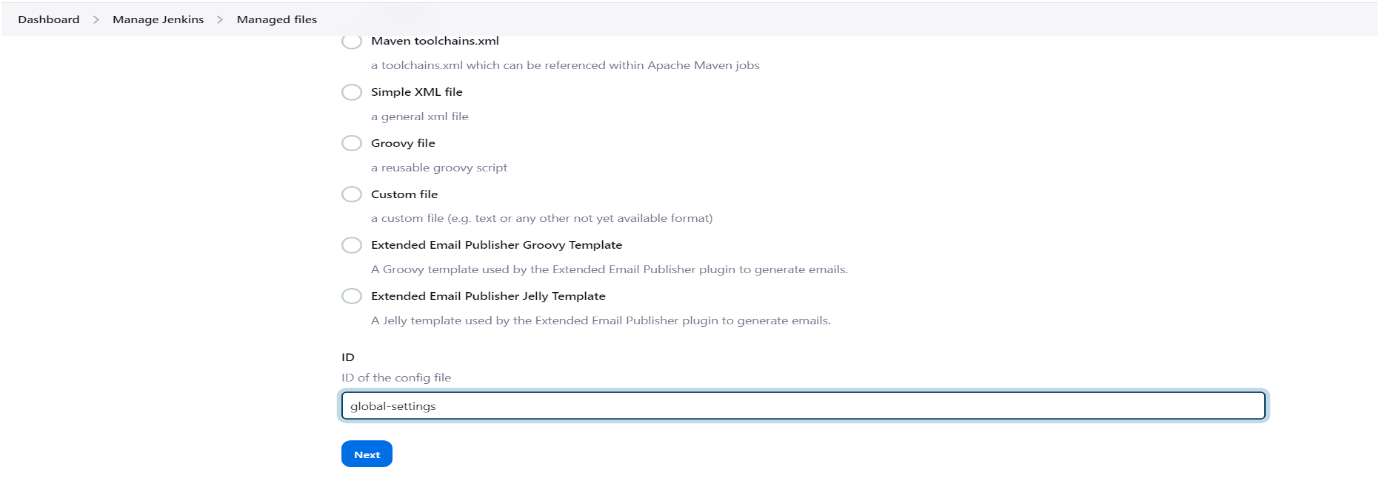
Configure or set up nexus in jenkins go to **manage jenkins🡪go to manged files** as shown below image👇



Click on **Add a new config**🡪click on **Global Maven settings.xml** as shown below image

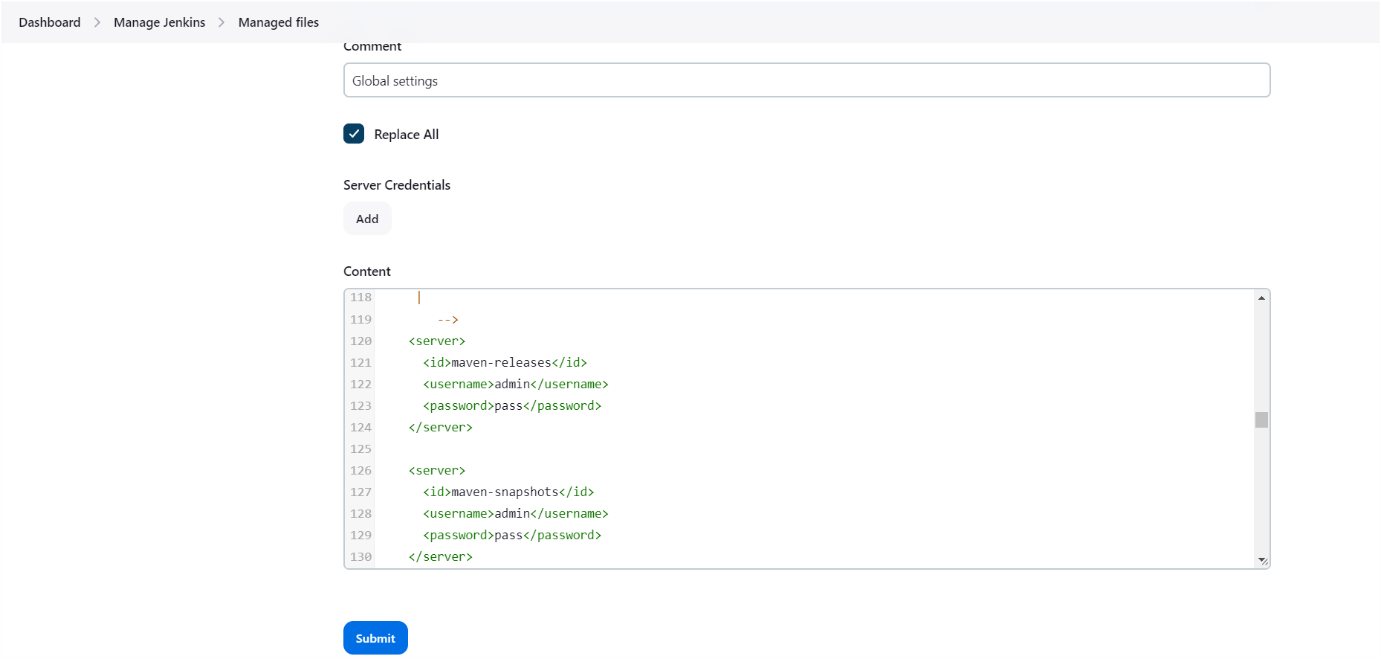


Next scroll down go to ID 🡪 give name as “global-settings” as shown below image👇



Click on next☝️

Next provide **credentials of nexus server** (give your nexus credentials) as shown below image👇



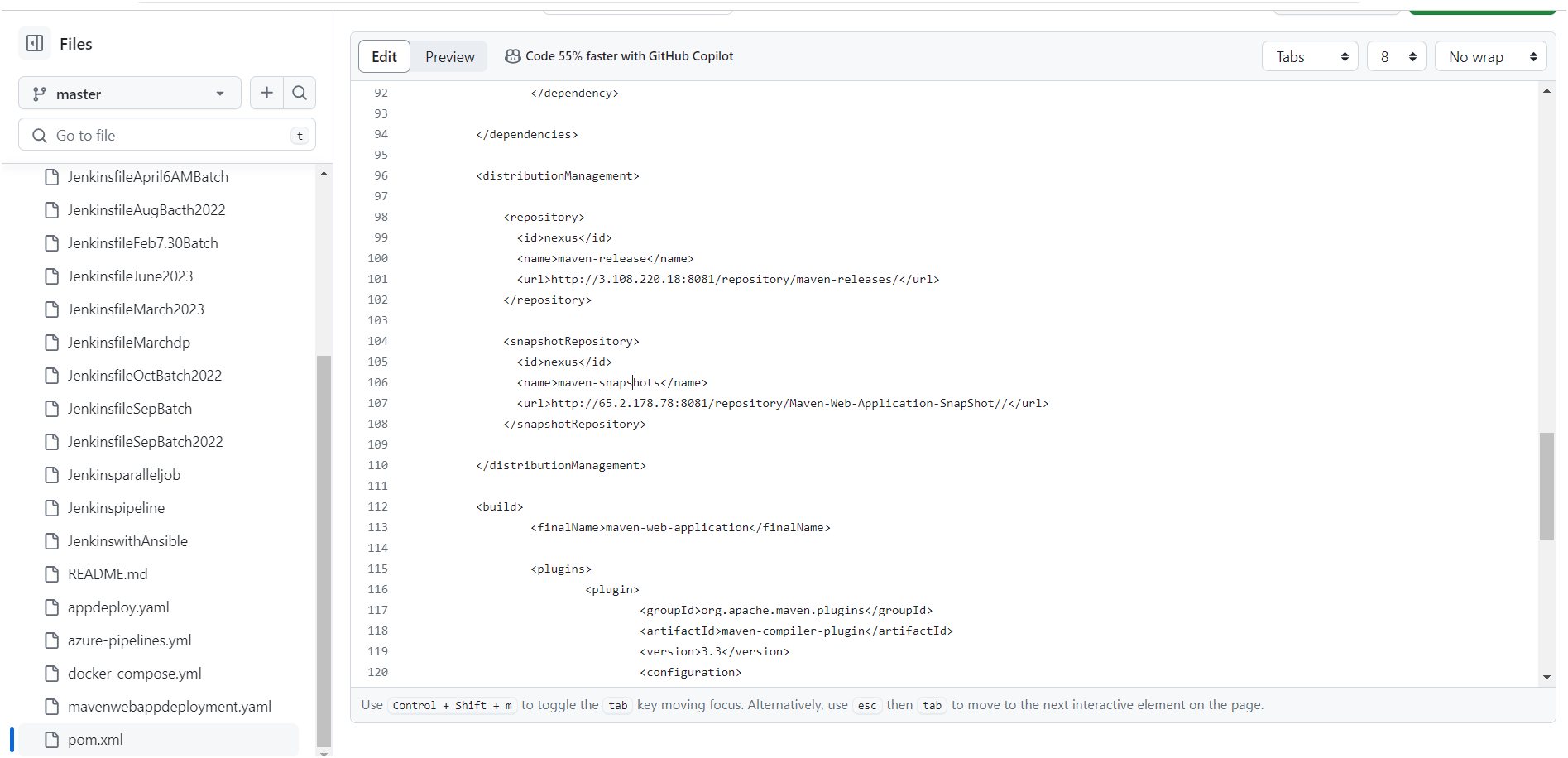
Click on submit☝️

Next need to update maven-snapshots and maven-releases URLs in your pom.xml file as shown below image👇

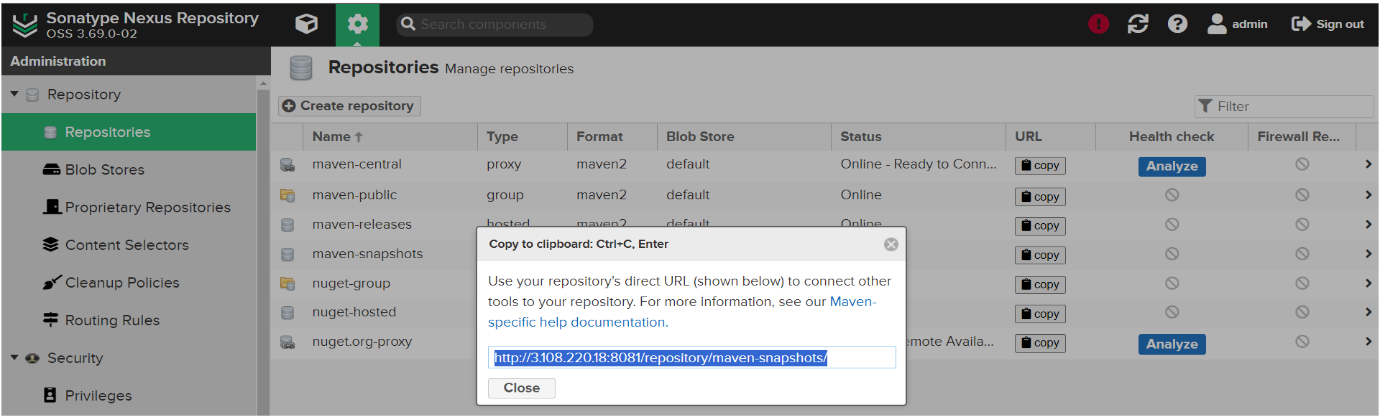
For getting maven-releases url follow below steps👇

****

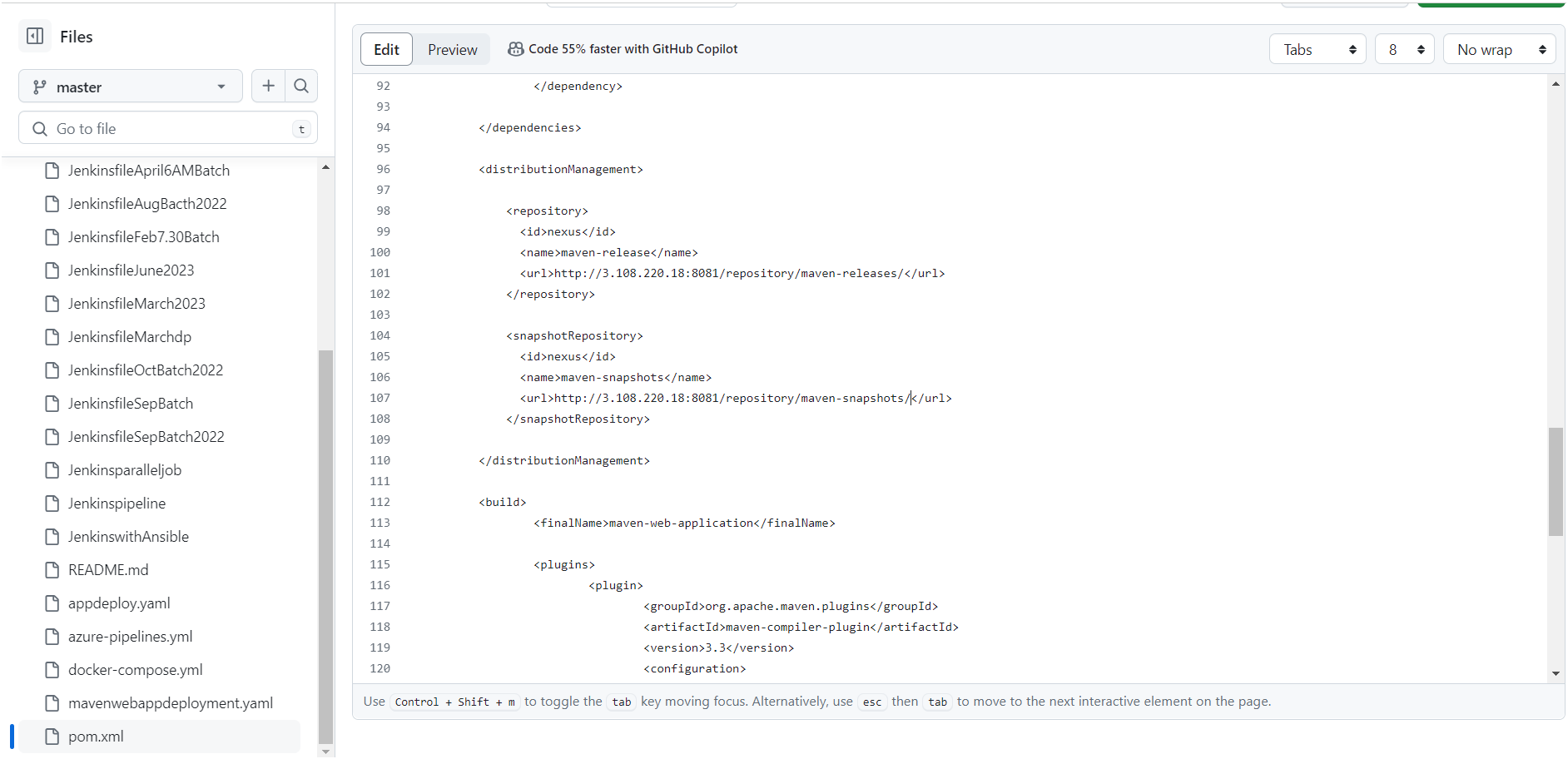
Update the above url in your pom.xml as shown below image 👇



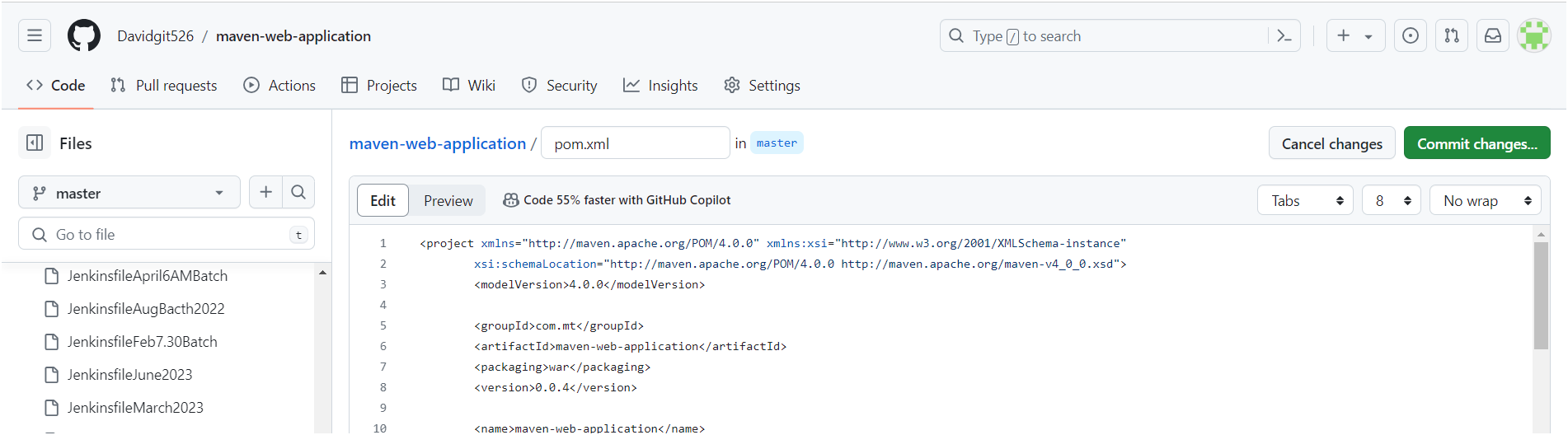
For getting maven-snapshots URL follow below steps👇



Update the above URL in your pom.xml as shown below image 👇



Next you need to commit the changes as shown below image at your GitHub repository👇



**Install Tomcat on Ubuntu 24.04 LTS**

### Step #1: Install Java on Ubuntu 24.04 LTS

To install Tomcat, you should have Java installed. Tomcat relies on Java to function.

First update the package repository.

sudo apt update

Install Openjdk (Java).

sudo apt install openjdk-17-jdk

### Step #2: Create a Tomcat User

For security reasons, it’s best not to run Tomcat as the root user. Let’s create a dedicated user and group.

sudo useradd -m -U -d /opt/tomcat -s /bin/false tomcat

This command creates a user named “tomcat” with a home directory at **/opt/tomcat**. The **/bin/false** shell prevents login access.

### Step #3:Install Tomcat on Ubuntu 24.04 LTS

Now let’s install Tomcat. First navigate to the Tomcat official website and download the latest version of Tomcat 10. You can use **wget**for this purpose.

sudo wget https://www-eu.apache.org/dist/tomcat/tomcat-10/v10.1.24/bin/apache-tomcat-10.1.24.tar.gz -P /tmp

extract it to the **/opt/tomcat** directory.

sudo tar -xvf /tmp/apache-tomcat-10.1.24.tar.gz -C /opt/tomcat

### Step #4:Update Permissions of Tomcat

Change ownership of the Tomcat directory to the **tomcat**user and group:

sudo chown -R tomcat:tomcat /opt/tomcat

### Step #5: Configure Tomcat as a Service

A systemd unit file tells systemd how to manage a service. Create a file named **tomcat.service** under the **/etc/systemd/system** directory using a text editor.

navigate to the **/etc/systemd/system**.

cd /etc/systemd/system

Create a tomcat.service file using nano command.

sudo nano tomcat.service

add the following content into it.

[Unit]

Description=Tomcat Server

After=network.target

[Service]

Type=forking

User=tomcat

Group=tomcat

Environment="JAVA\_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

WorkingDirectory=/opt/tomcat/apache-tomcat-10.1.24

ExecStart=/opt/tomcat/apache-tomcat-10.1.24/bin/startup.sh

[Install]

WantedBy=multi-user.target

Save and close the file.

Adjust the **JAVA\_HOME**environment variable path based on your OpenJDK installation location.

### Step #6:Reload systemd and Start Tomcat

Reload the systemd daemon to apply the changes.

sudo systemctl daemon-reload

Start the Tomcat service.

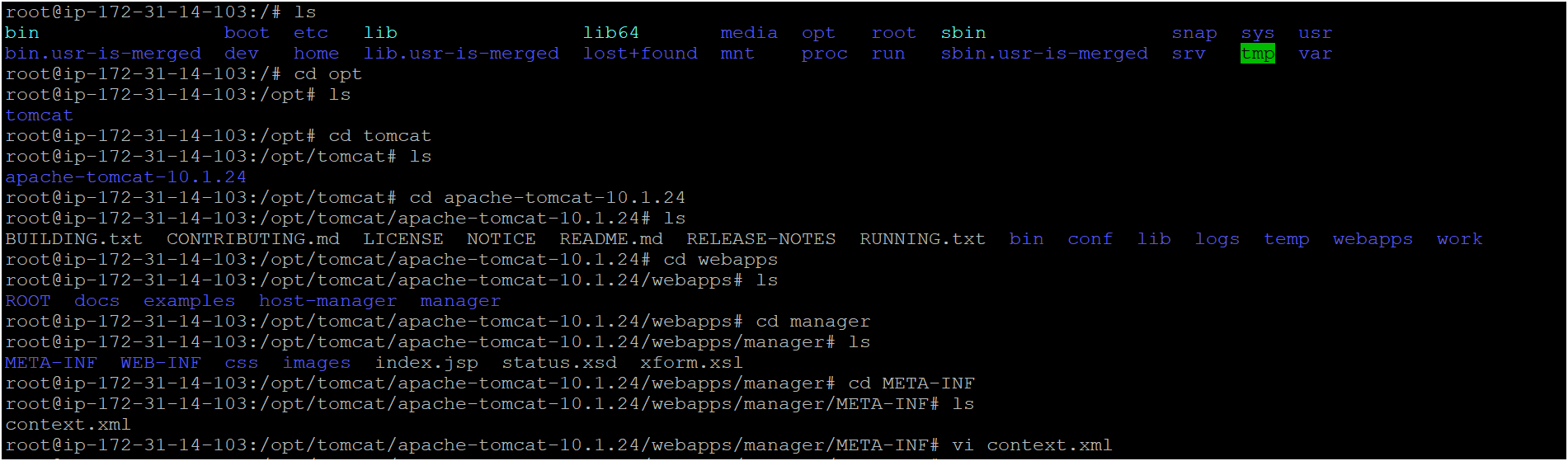
sudo systemctl start tomcat

Enable Tomcat to start on boot.

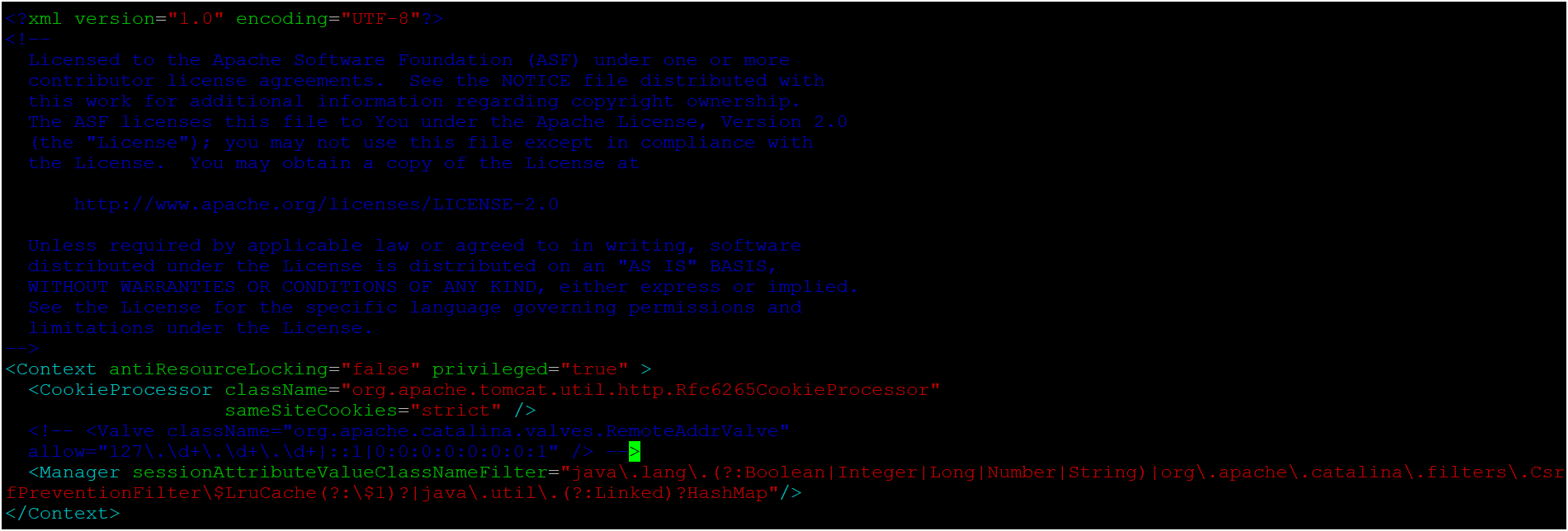
sudo systemctl enable tomcat

Configure **tomcat credentials** in your tomcat server as shown below image👇

Follow the commands as shown below image👇



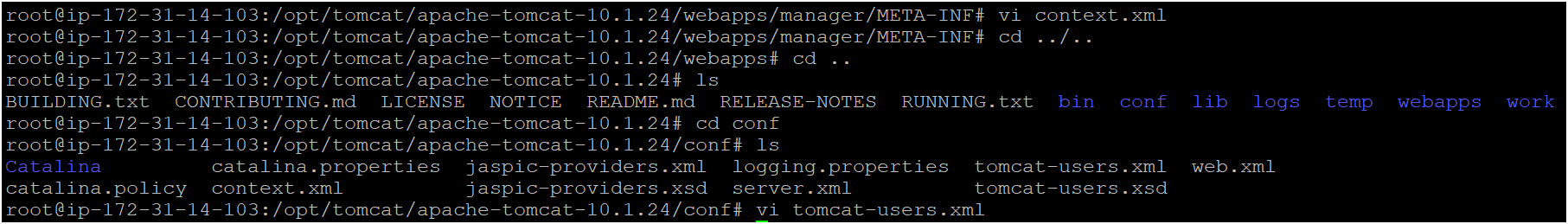
After it will open like the below image 👇

****

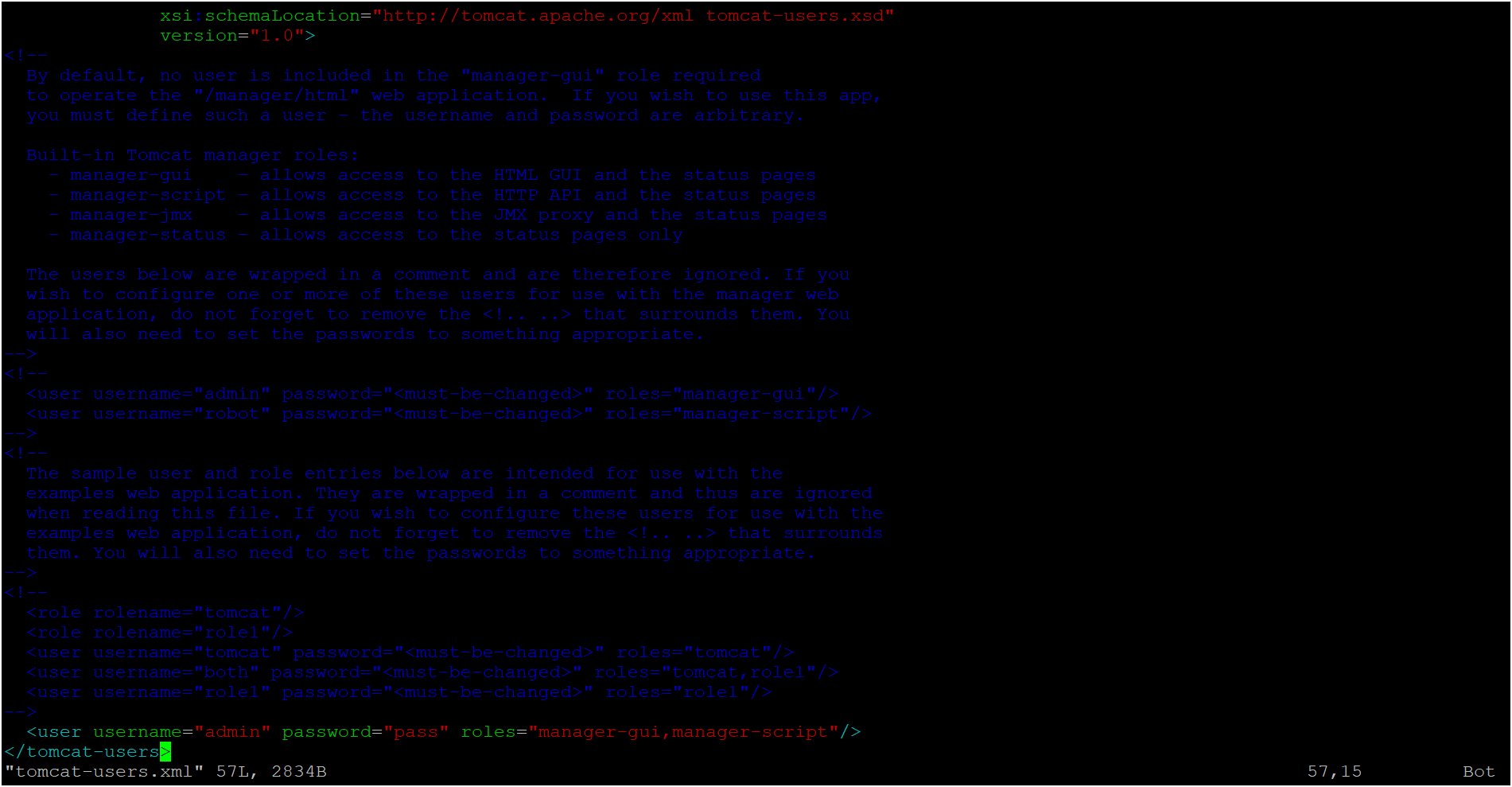
You need to edit this file as shown above image☝️

Next **Configure credentials for tomcat server** as shown below image👇

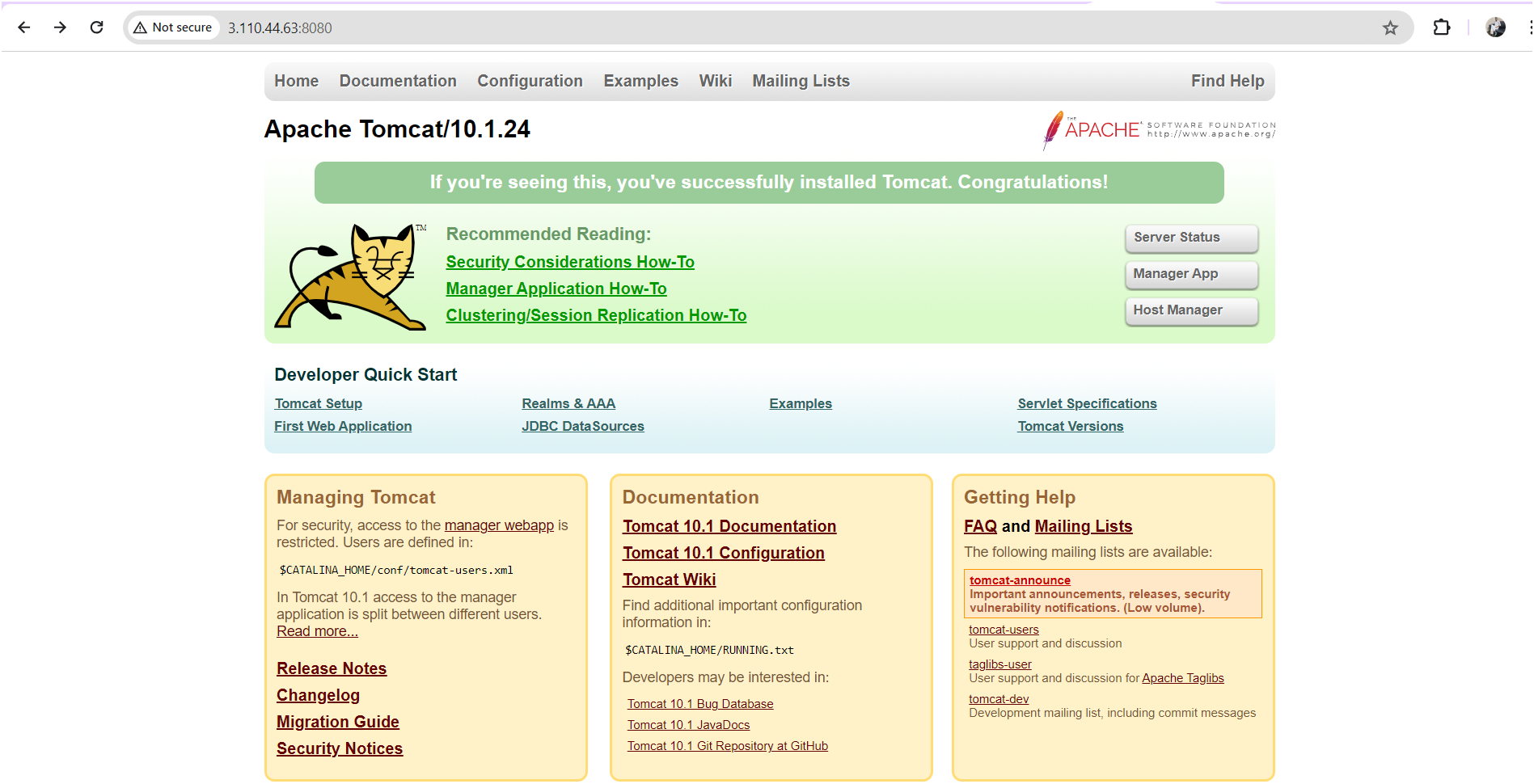
Follow the commands as shown below image👇



After it will open like the below image 👇

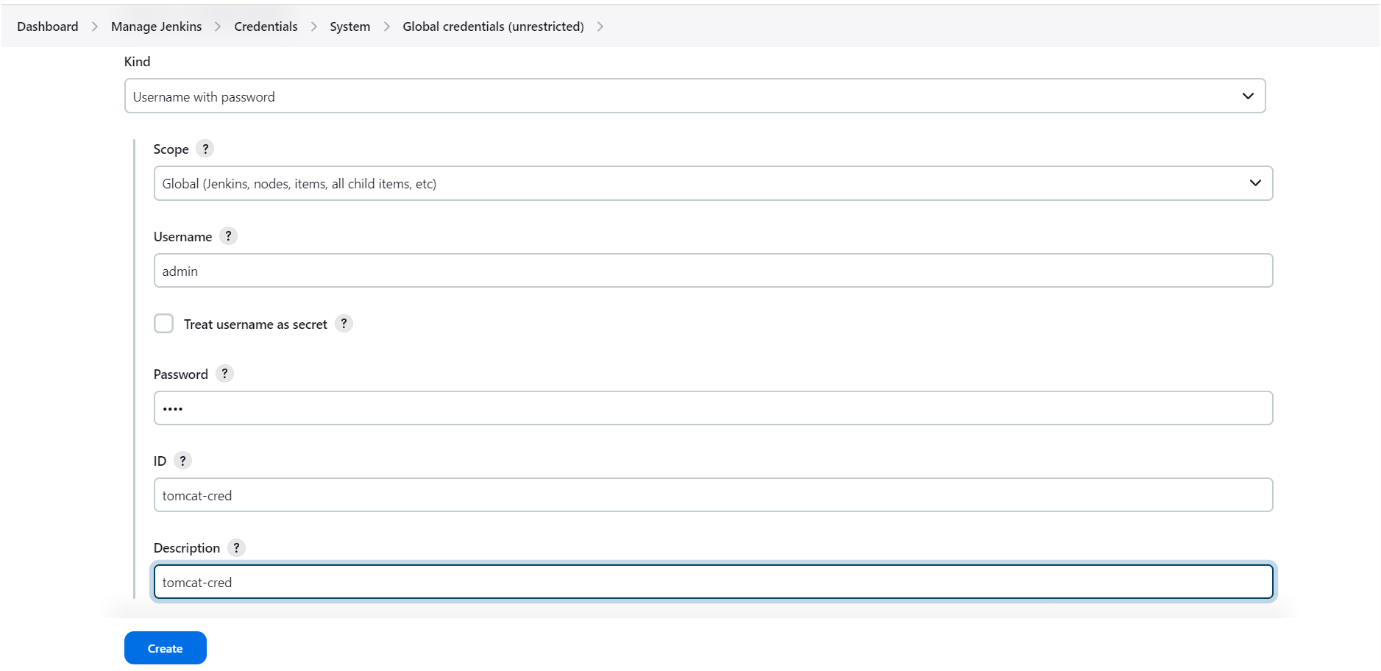


Now you can verify if your service is running properly or not by running your public ip address with port 8080 which is default port for Tomcat in url. You should see the Tomcat welcome page if everything is set up correctly.

****

To login to tomcat server click on **Manage App** after provide credentials

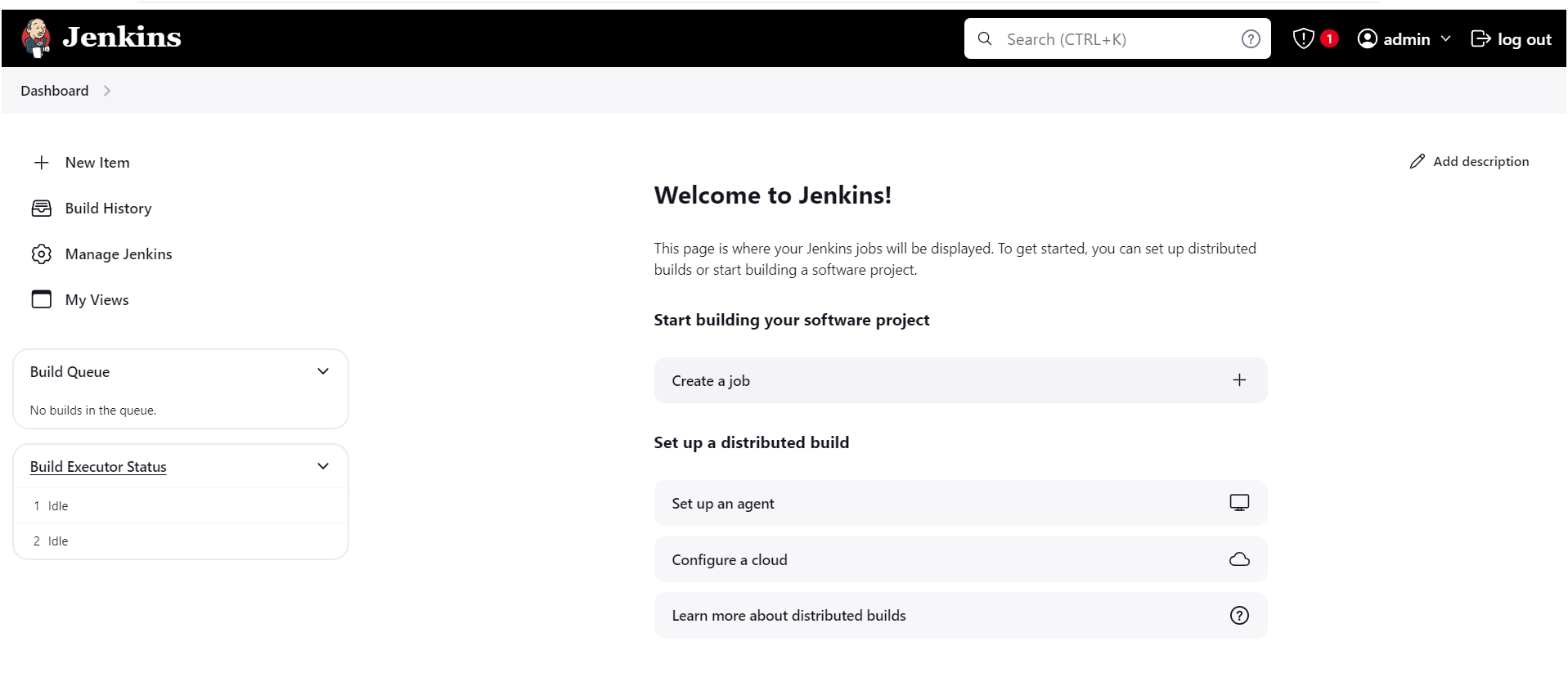
Now configure tomcat server in Jenkins go to **manage Jenkins🡪go to credentials** as shown below image👇



Click on create ☝️

**Let’s create a pipeline**

Firstly, we need to create a job or item by clicking on “new item” as shown below image

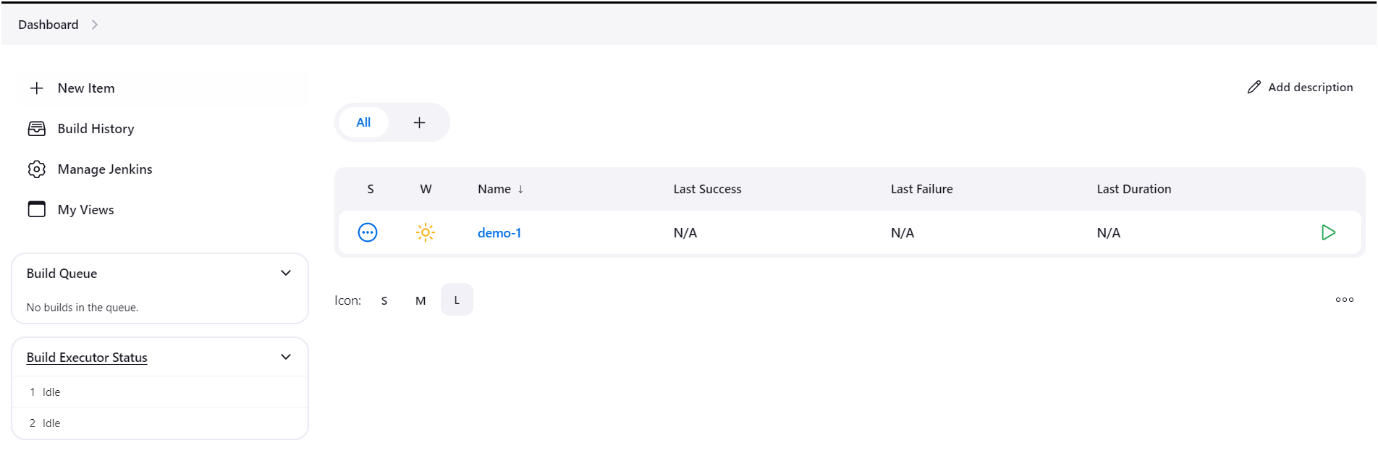


Next, we need to select pipeline type as shown below iamge.

Give name🡪select “pipeline”🡪click on “ok”

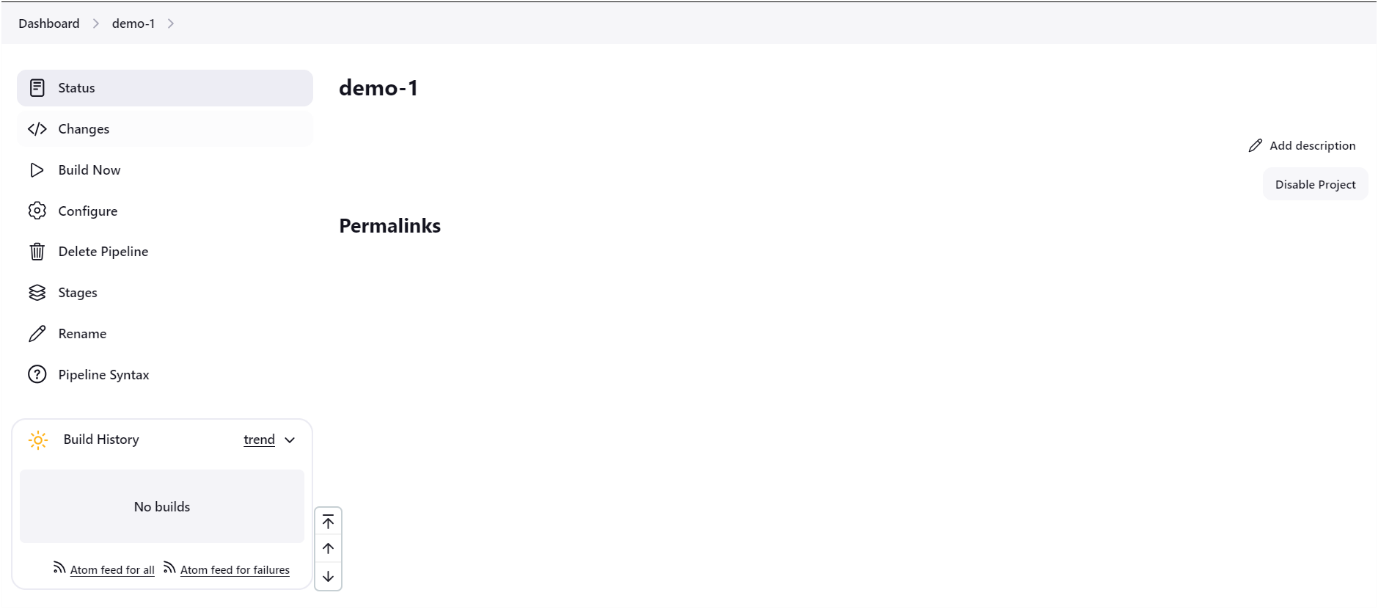


After page will appear as shown below 👇



Next click on project name “**demo-1**”

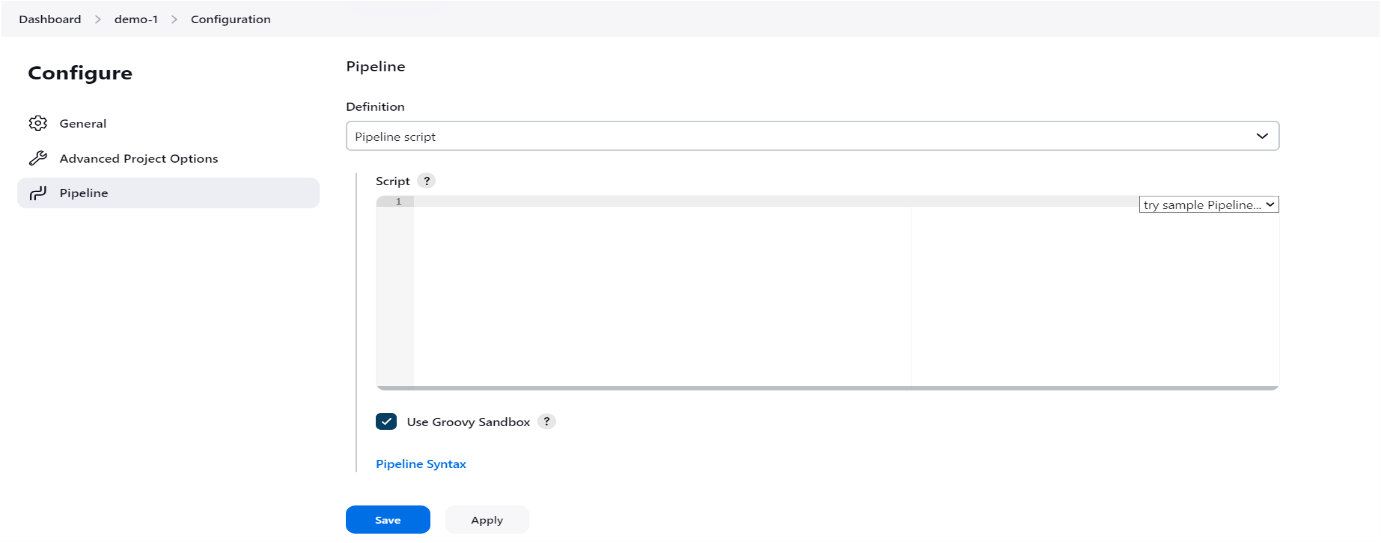
**Let’s start to write declarative pipeline script 👇**



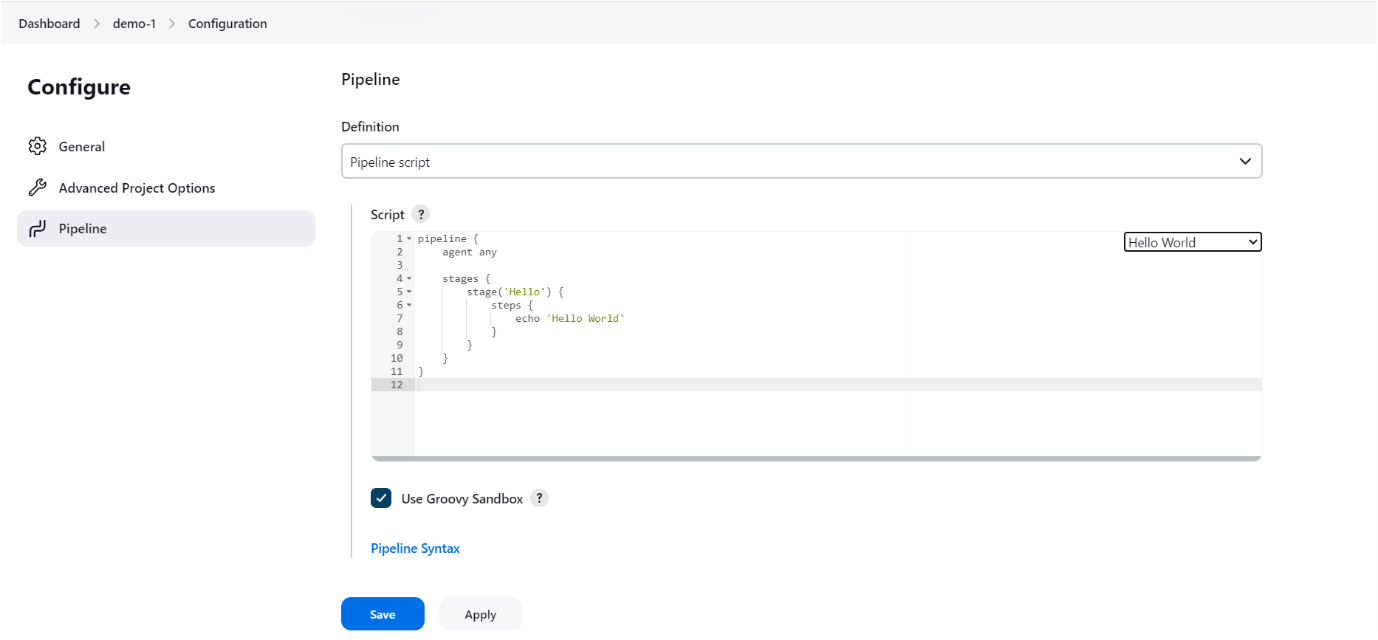
Click on **configure** ☝️



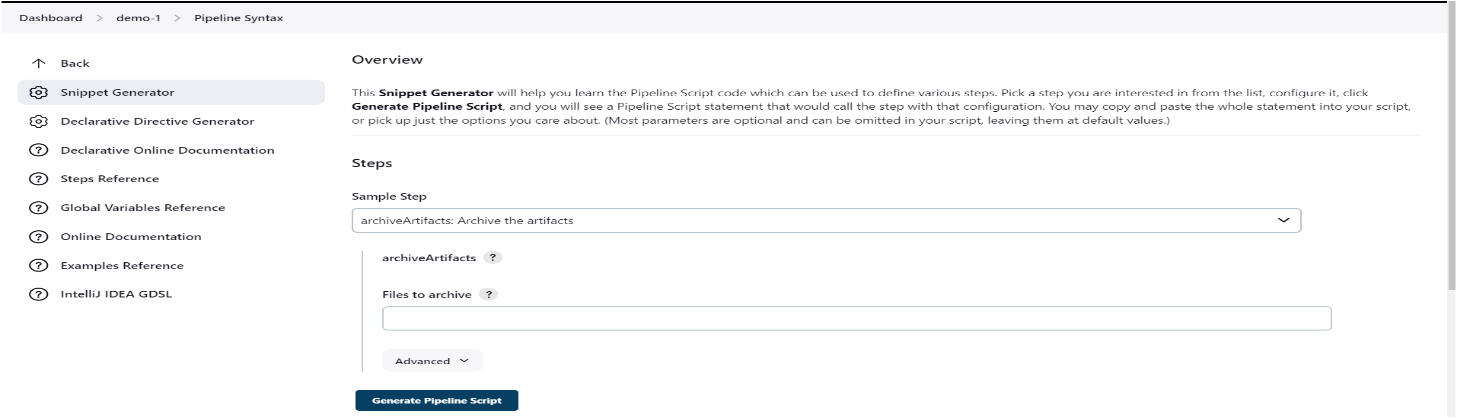
Click on **pipeline** ☝️after below page visible 👇



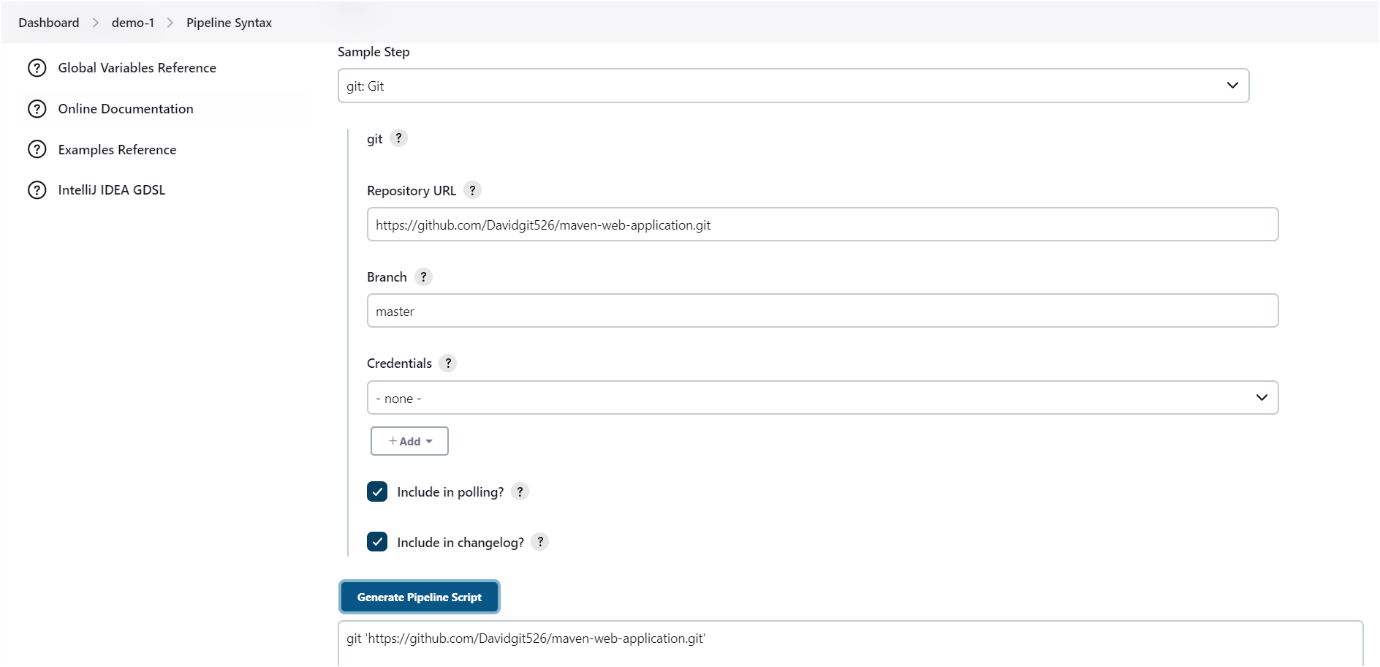
After below page visible 👇



We can write pipeline script by using **pipeline syntax** after clicking on pipeline sysntax below page will visible 👇



1. Generate Git checkout script using pipeline syntax 👇, provide your repository url in Repository URL👇



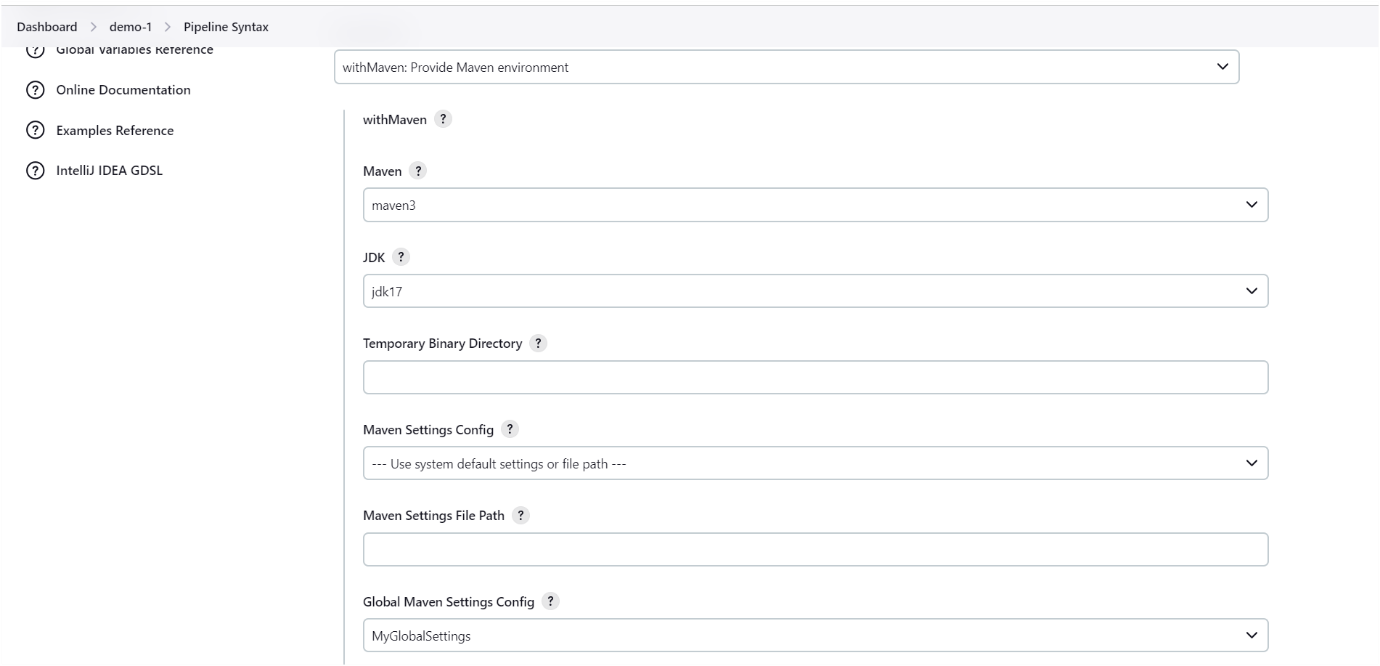
Git checkout script 👇

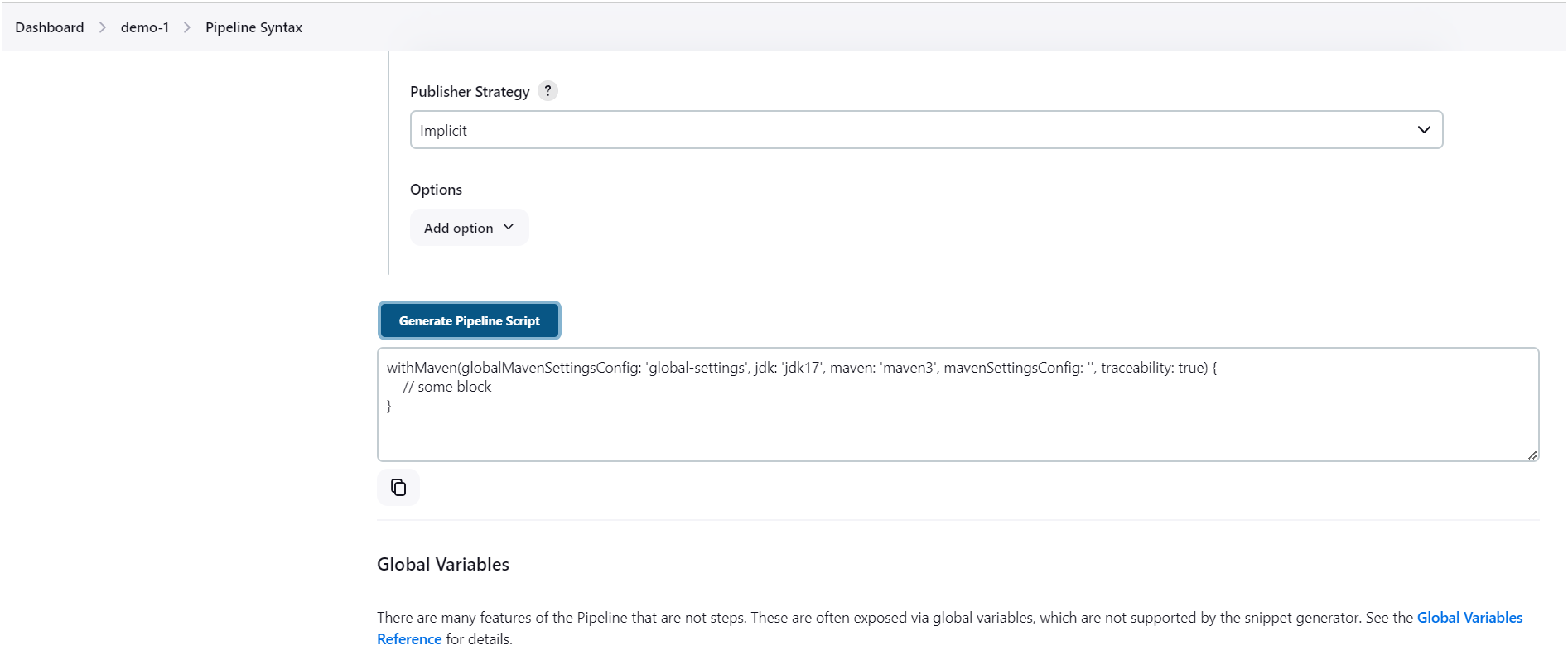


Maven script 👇we need to write manually



Nexus script generate by using pipeline syntax👇

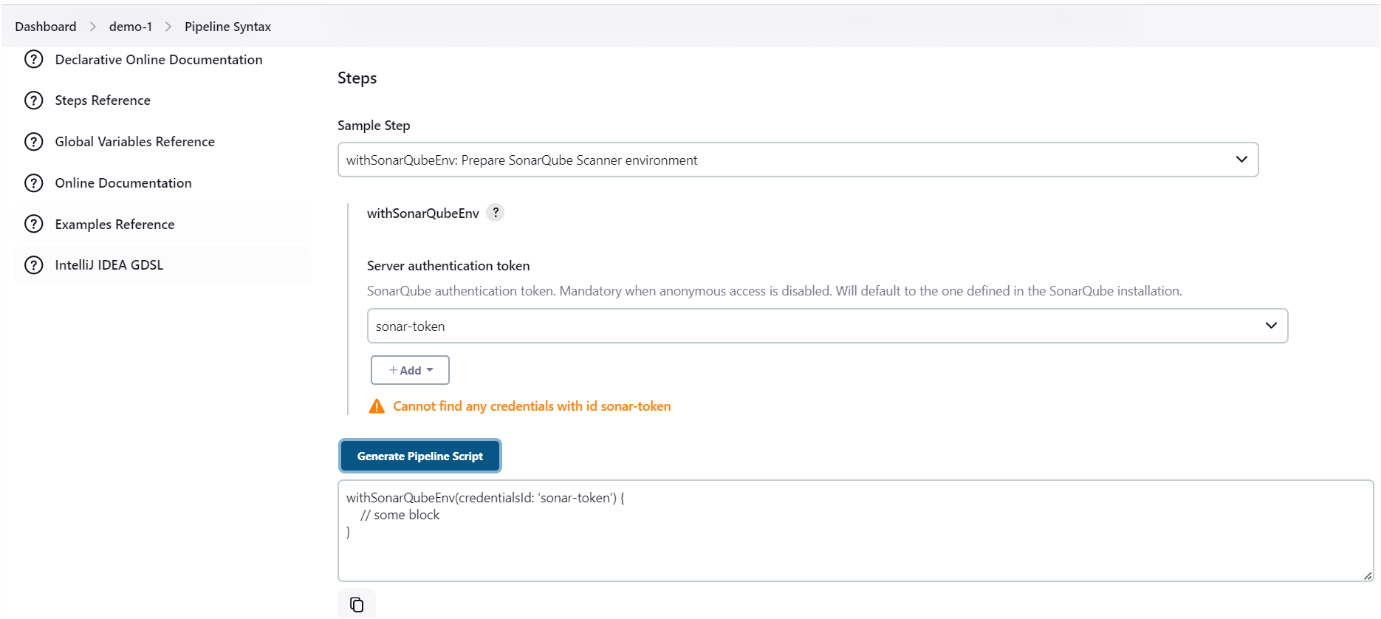


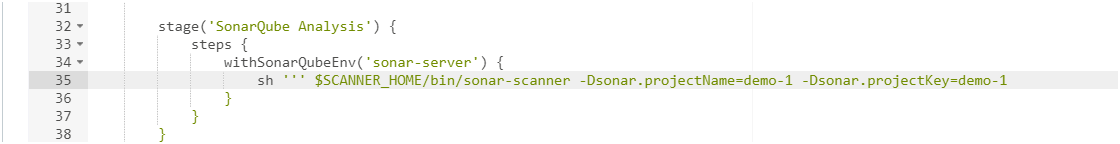


Nexus script 👇



SonarQube script generate 👇





Now we have to generate ssh-key on our Jenkins server for authenticating our Tomcat Apache server follow the below steps👇

**On Jenkins server:**

**Generate and verify SSH keys on Jenkins server:**

ssh-keygen -t rsa

(Press Enter to accept the default file location and leave the passphrase empty.)

cat ~/.ssh/id\_rsa.pub

(copy the public and need to paste it on remote server inside authorized keys file)

**On remote server: (Tomcat server)**

mkdir -p ~/.ssh

chmod 700 ~/.ssh

nano ~/.ssh/authorized\_keys

“Ensure the public key is correctly pasted in one line. Save and exit.”

chmod 600 ~/.ssh/authorized\_keys

**Check SSHD Configuration**

Ensure the SSH daemon configuration on the remote server is correct.

**On the remote server:**

sudo nano /etc/ssh/sshd\_config

Ensure the following settings are enabled:(If not remove **#** symbole)

**PubkeyAuthentication yes**

**AuthorizedKeysFile .ssh/authorized\_keys**

Save and exit.

**Restart SSH Service:**

sudo systemctl restart ssh

**Test connection with verbose mode:(On Jenkins server)**

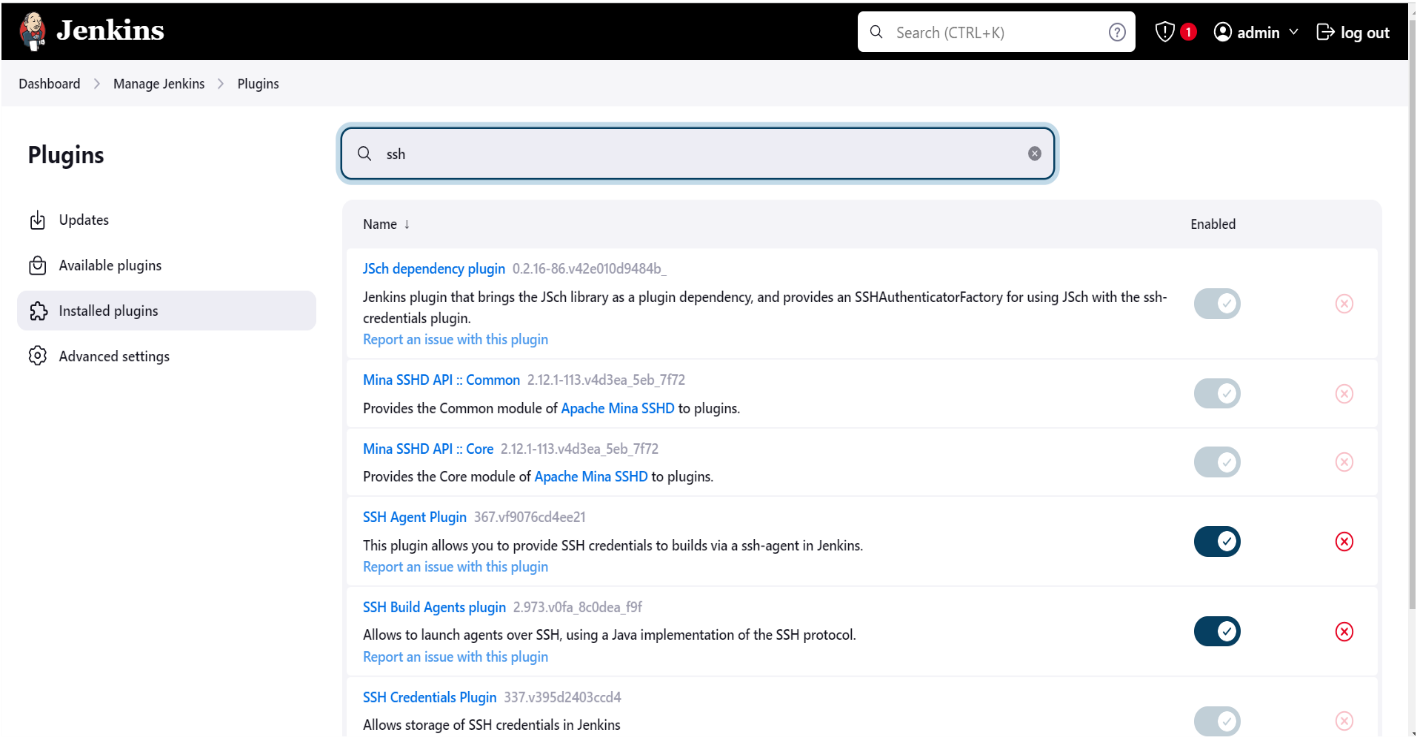
ssh -vvv -i ~/.ssh/id\_rsa [user@13.234.87.245](mailto:user@13.234.87.245)

(user-username, remote server ip address)

Now you can check are you able access your tomcat server from Jenkins , if are you able to access then come out of it.

Now need to install **ssh agent plugin**

Go to Dashboard🡪go to manage plugins 🡪 available plugins🡪 select **ssh agent plugin**



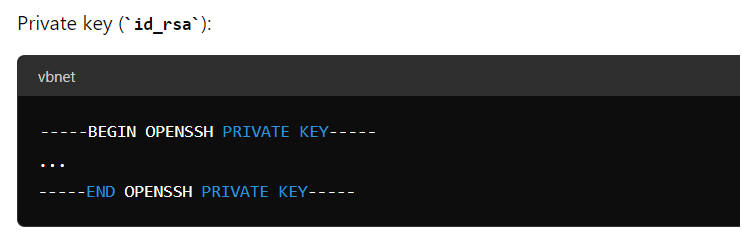
Now configure ssh-key (pvt key) in Jenkins server in credentials section as follow below image 👇

Now copy your private key as shown below 👇

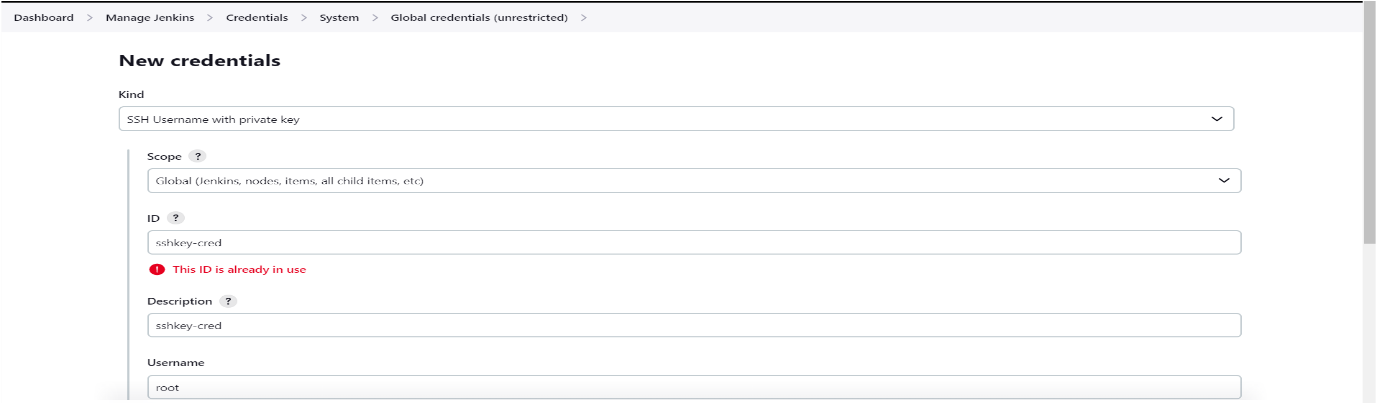




Example of private 👇

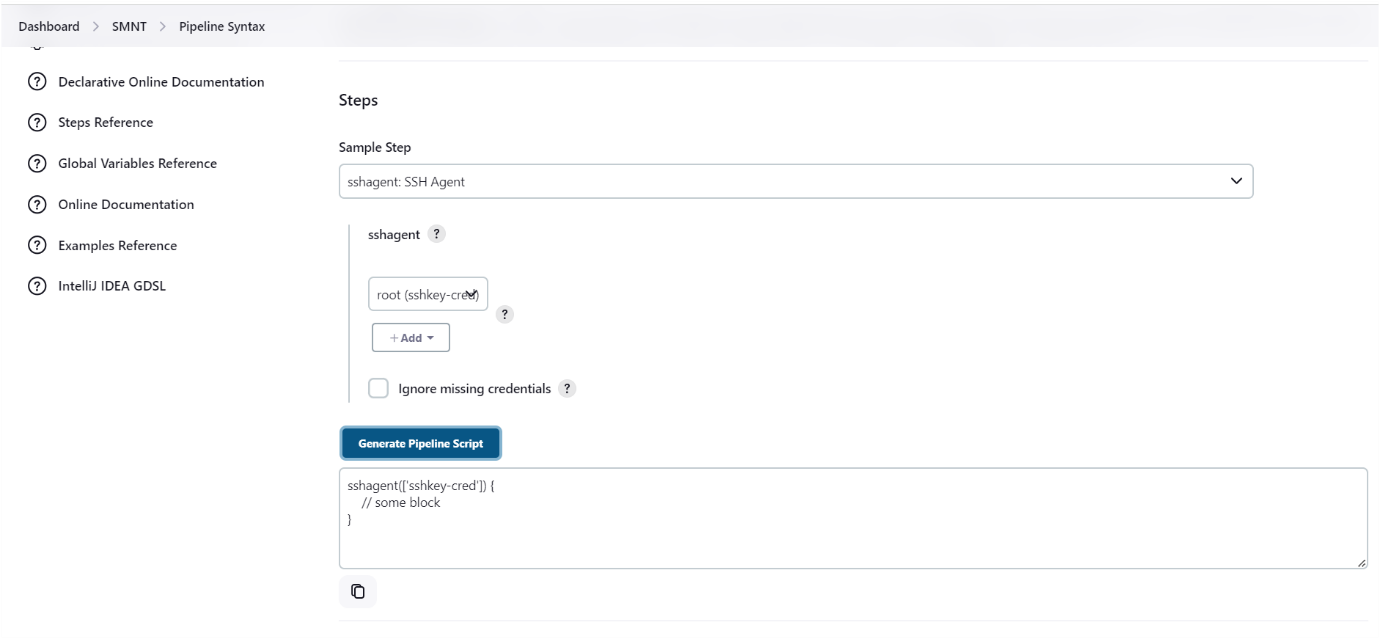


Go to manage plugins 🡪 go to credentials and configure private key of ssh here 👇





Tomcat ssh-key script generate 👇



Now we can run entire pipeline script 👇

pipeline {

agent any

tools {

jdk 'jdk17'

maven 'maven3'

}

environment {

SCANNER\_HOME = tool 'sonar-scanner'

}

stages {

stage ('git checkout') {

steps {

git branch: 'main', url: 'https://github.com/Davidgit526/Petclinic.git'

}

}

stage('Build') {

steps {

sh 'mvn clean package'

}

}

stage ('Publish To Nexus') {

steps {

withMaven(globalMavenSettingsConfig: 'my-global-settings', jdk: 'jdk17', maven: 'maven3', mavenSettingsConfig: '', traceability: true) {

}

}

}

stage('SonarQube Analysis') {

steps {

withSonarQubeEnv('sonar-server') {

sh ''' $SCANNER\_HOME/bin/sonar-scanner -Dsonar.projectKey=demo-1 -Dsonar.projectName=demo-1 '''

}

}

}

stage('Deploy to tomcat') {

steps {

sshagent(['sshkey-cred']) {

sh 'scp -o StrictHostKeyChecking=no target/petclinic.war root@172.31.38.31:/opt/tomcat/apache-tomcat-10.1.24/webapps'

}

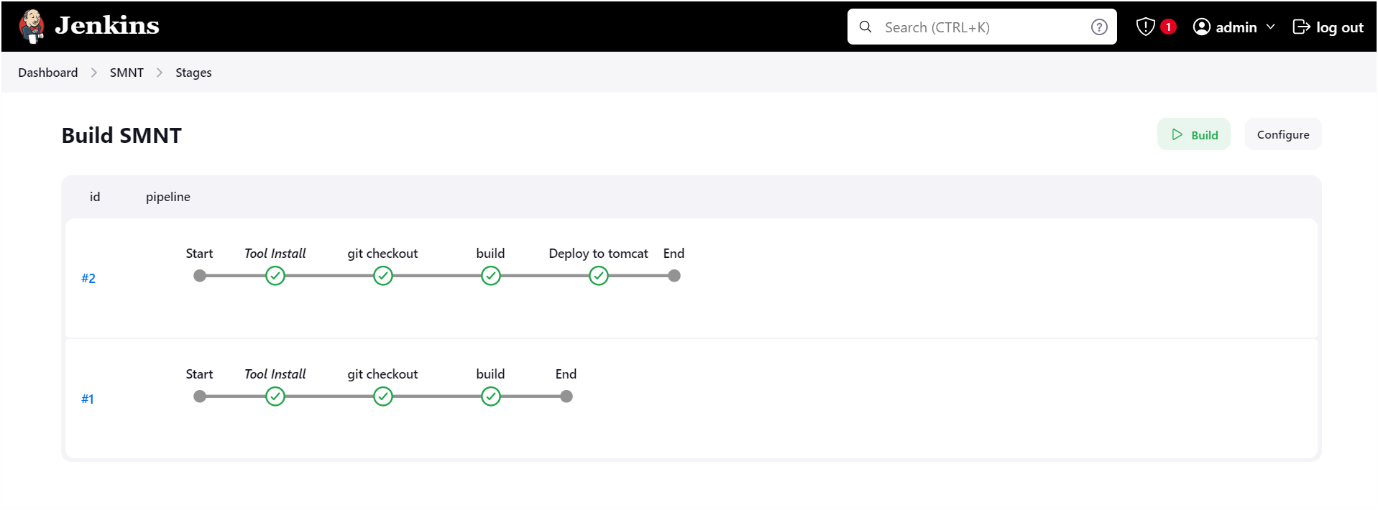
}

}

}

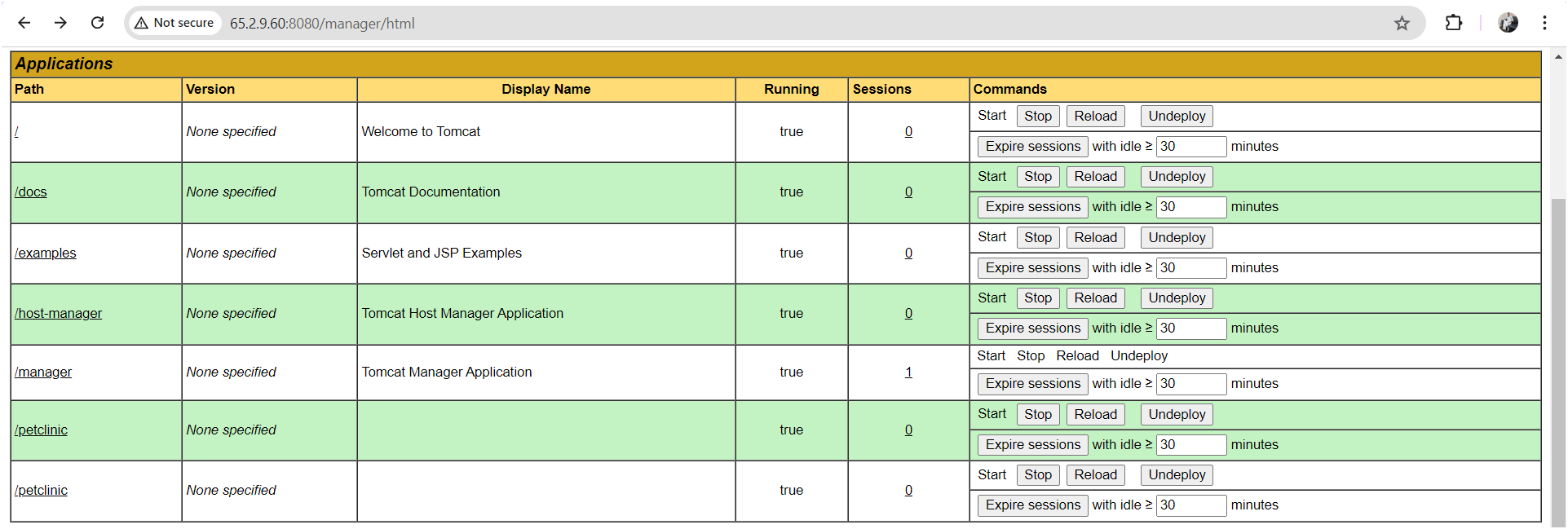
}

Now we can see pipeline status 👇



Now we can see the application deployed into the tomcat server

👇



This is for how to integrate SonarQube, maven, nexus and tomcat server in declarative script.

**THE END**