# SONARQUBE JENKINS INTEGRATION

**What is Sonarqube?**

SonarQube is an open-source platform that is used to inspect code quality continuously. It performs automated code reviews by scanning the code for possible issues like bugs, vulnerabilities, bad code practices, security risks. and way more. SonarQube supports a vast number of programming languages and plugs into CI/CD pipelines with the help of which the quality of the code is continuously checked, and necessary actions are done to improve it.

**Some Key Features In Sonarqube:**

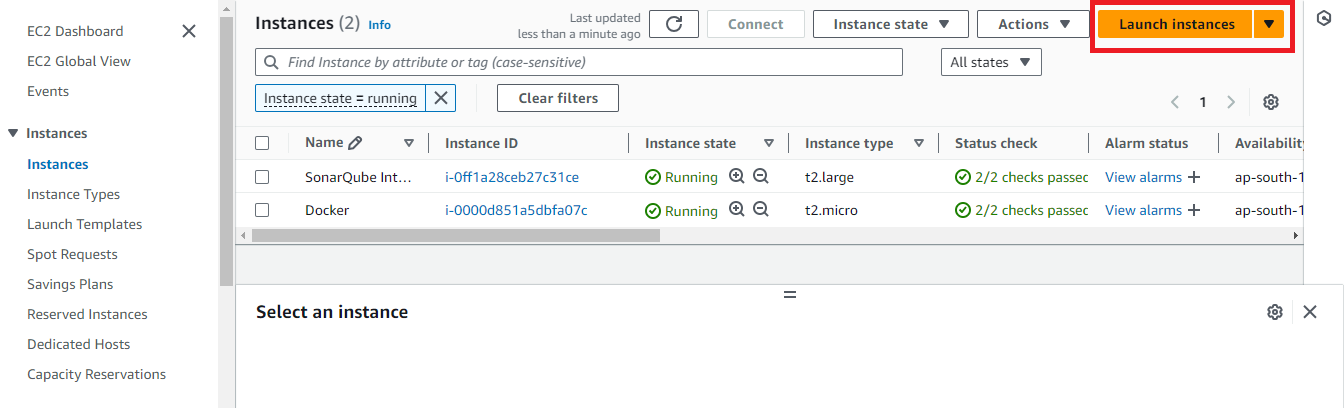
* Static Code Analysis
* Code Quality Metrics
* Security Vulnerabilities
* Language Support
* Integration with CI/CD Tools

**Sonarqube Jenkins Integration:**

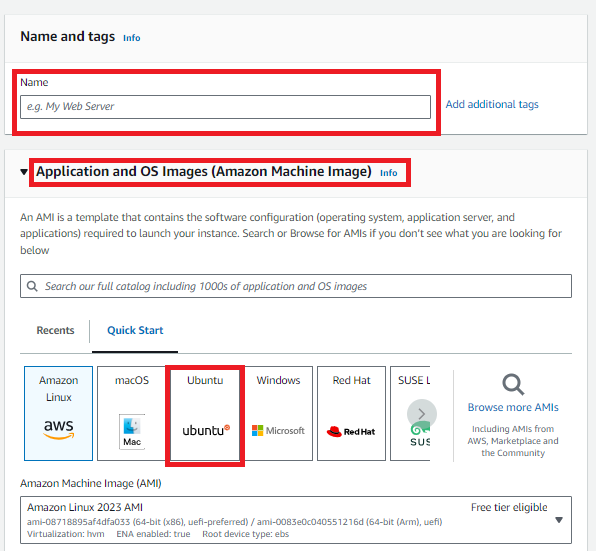
**Step-1:**

First Need to Create one EC2 instance in AWS Console

1. First click on launch instance in EC2 dashboard



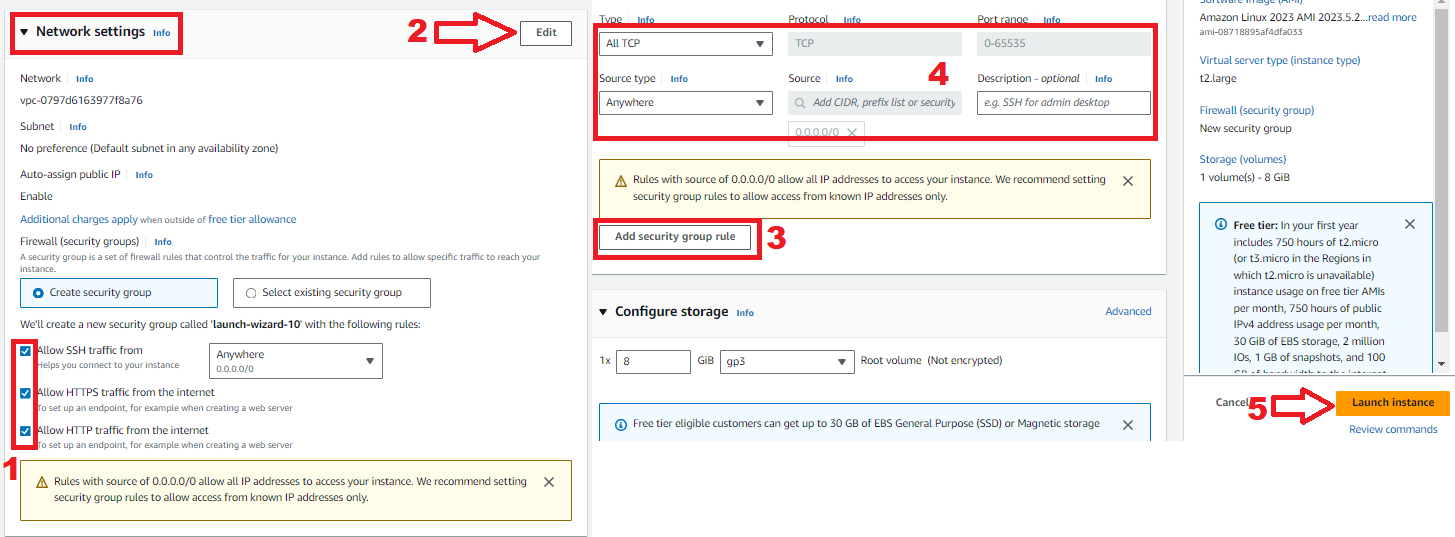
2.After click on the Launch Instance Need to provide Name and select AMI (like Linux,Ubuntu etc…,)



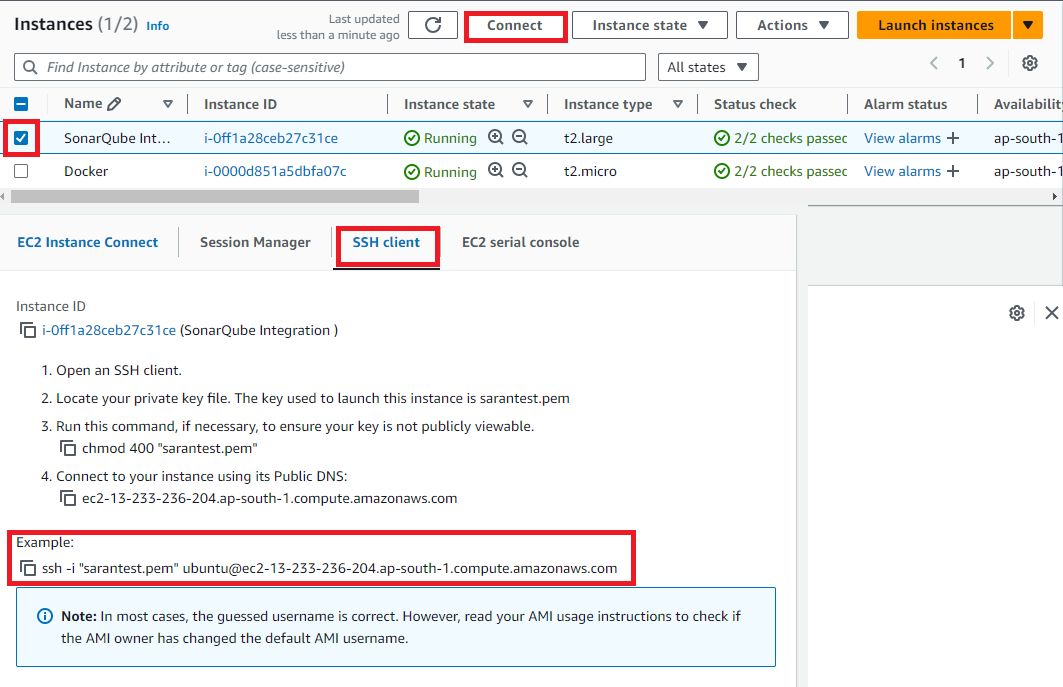
3.Then select Instance Type as t2.large and select select any existing key pair or create new key by click create new key pair.



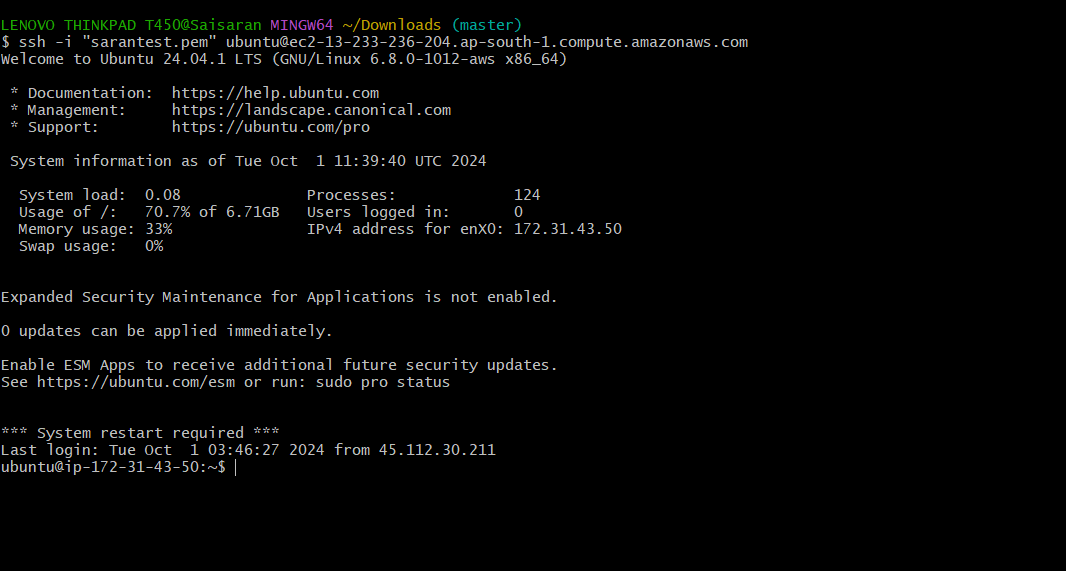
1. Then go the Network settings and then Allow the HTTP & HTTPS then click on Edit then add security group with All TCP then click on Launch Instance.



1. After the Instance Created then Connect the instance with any tool(GIT,Mobaxterm etc..,) with SSH Key



1. After Copy the SSH link paste that in any tool (GIT,Mobaxterm etc..,) to connect with AWS.



**What is Jenkins?**

Jenkins is an open source automation tool written in java-programming language that allows continous integration.

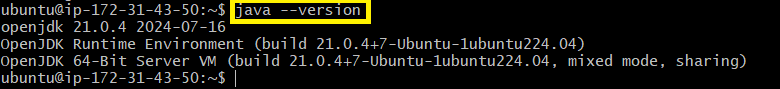
Jenkins will build and test the code.It supports continuous integration (CI) and continuous delivery (CD) pipelines, allowing teams to quickly integrate changes into a project and automate the deployment process.

**Step-2:**

Now install Java,Maven,Jenkins,sonarqube and Docker in the instance for Integration.

To Install Jenkins follow below commands

1. sudo apt update -y
2. sudo apt upgrade -y
3. wget -O - <https://packages.adoptium.net/artifactory/api/gpg/key/public> | tee /etc/apt/keyrings/adoptium.asc
4. echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] <https://packages.adoptium.net/artifactory/deb> $(awk -F= '/^VERSION\_CODENAME/{print$2}' /etc/os-release) main" | tee /etc/apt/sources.list.d/adoptium.list
5. Sudo apt update -y
6. Now need to Install Jdk in Instance for that use **sudo apt install temurin-17-jdk -y**
7. After Installing jdk check jdk version by using **java --version**



1. Then need to Install Maven In the Instance using

**sudo apt install maven -y**

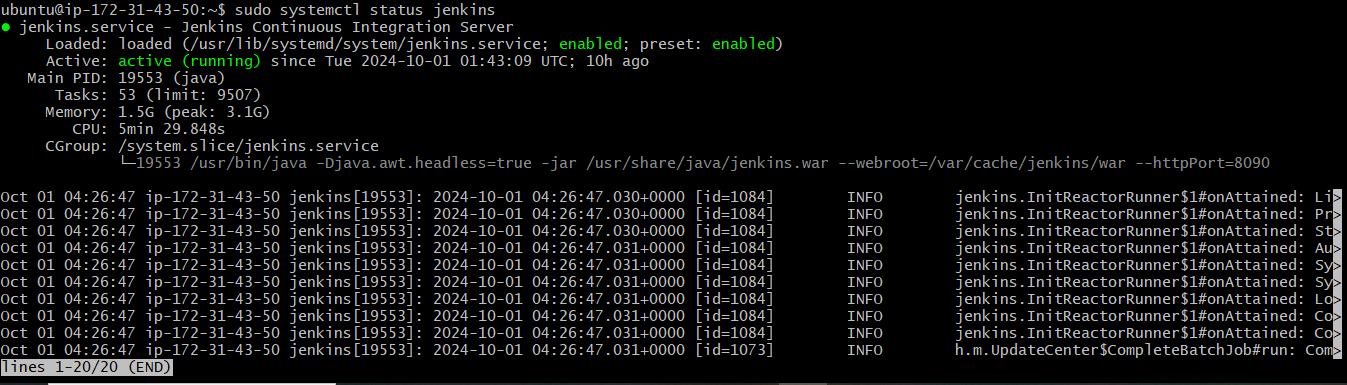
1. Then need to Install Jenkins before install need to add Keys for Jenkins

* **curl -fsSL** [**https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key**](https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key) **| sudo tee \ /usr/share/keyrings/jenkins-keyring.asc > /dev/null**
* **echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \** [**https://pkg.jenkins.io/debian-stable binary/**](https://pkg.jenkins.io/debian-stable%20binary/) **| sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null**

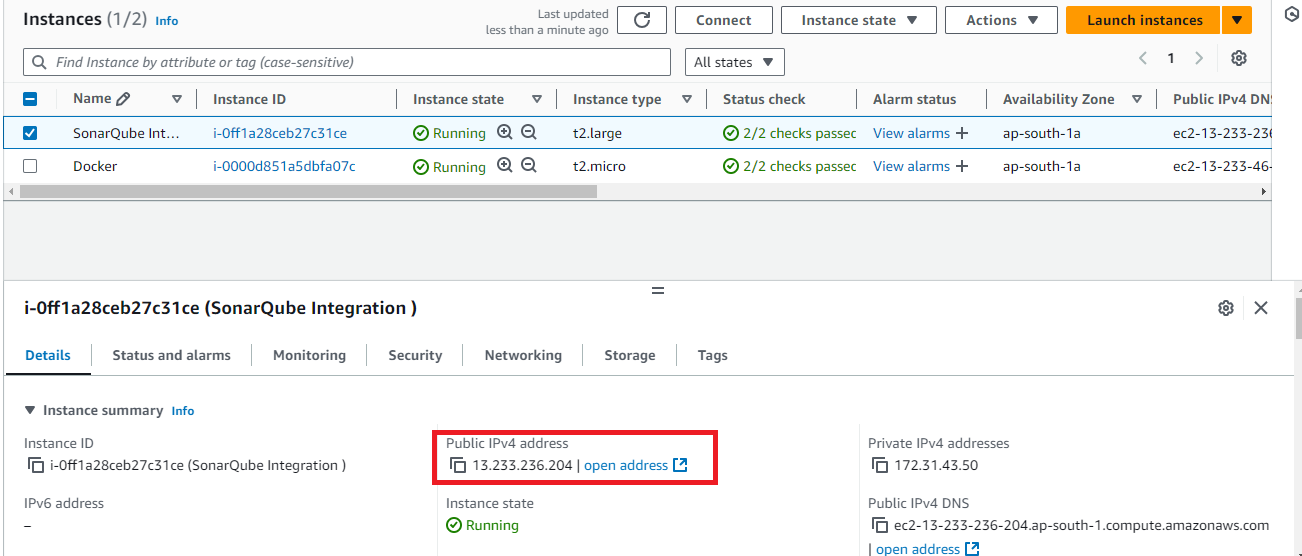
1. After Adding the keys need to install the Jenkins by using **sudo apt-get install jenkins -y**
2. After Jenkins Installed need to Jenkins Jenkins version



1. Then start the jenkins by using **sudo systemctl start jenkins**
2. After starting the Jenkins need to checkstatus of Jnekins it is active or not by using **sudo systemctl status jenkins**



1. Then connect to the Jenkins dashboard by using **Instance public IPV4 address:8080**

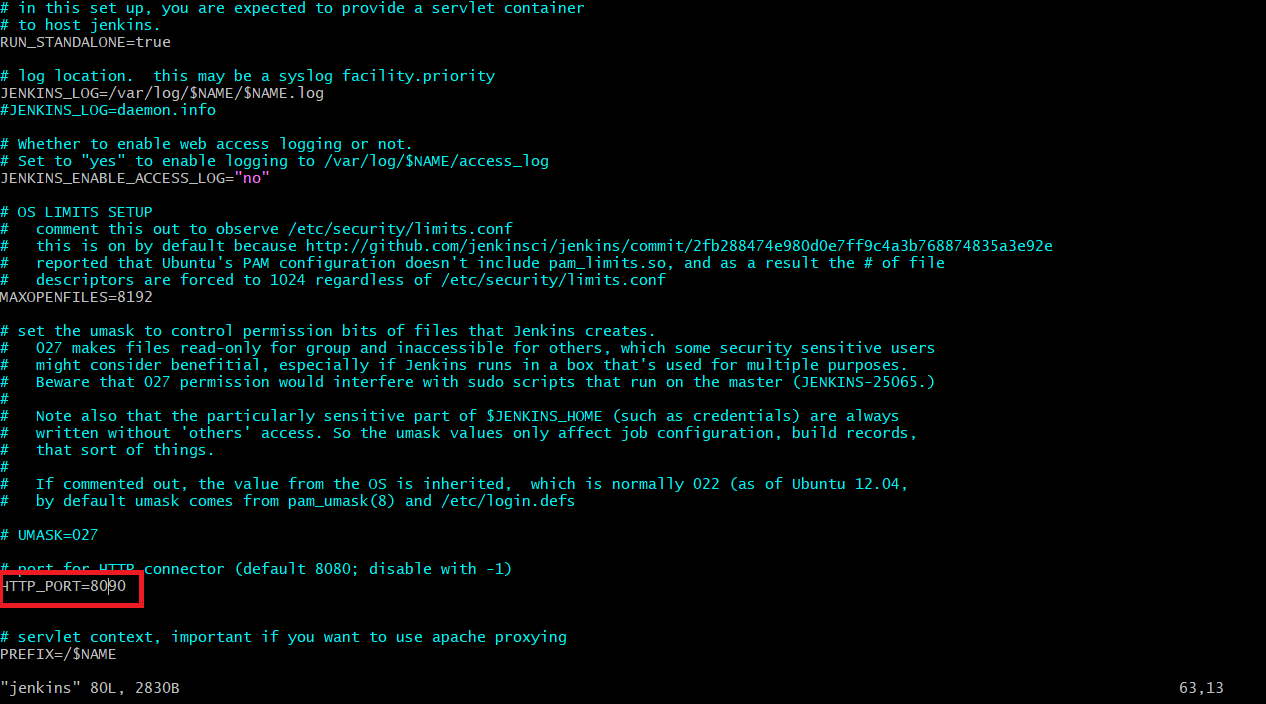


16.Once Jenkins is installed and connected to dashboard then for this Application case, we are running Jenkins on another port. so change the port to 8090

For changing to 8090 port first need to stop Jenkins by using **Sudo systemctl stop jenkins**

Then check the status of jenkins it is inactive or not by using **sudo systemctl status jenkins**

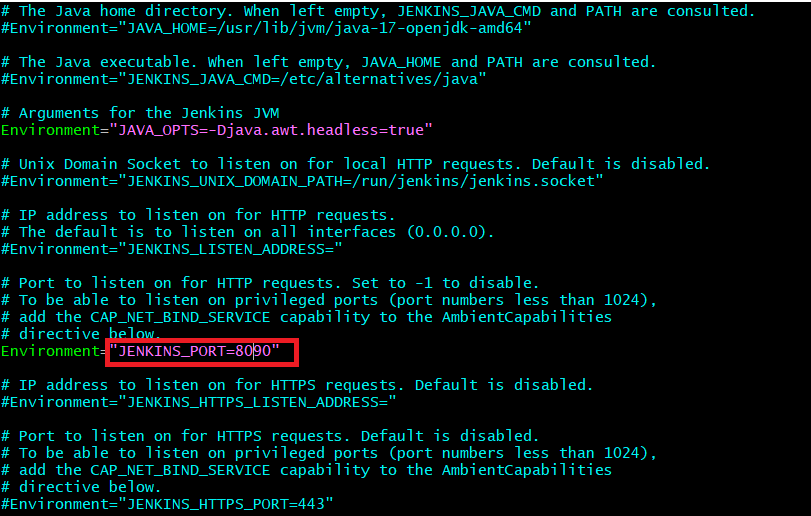
1. Now we need to change the port to 8090 for that need to change the directory by using **cd /etc/default**
2. After changing into directory need to go vi editor to change the port number by using **sudo vi jenkins**



To exit form vi editor need to enter **:wq**

19.After exit from vi editor need to change to another directory by using **cd /lib/systemd/system**

20.After changing into above directory need to got vi editor again by using **sudo vi jenkins.service** in this need to change the jenkins port to 8090 and then exit form vi editor



1. After exit from vi editor need to reload the jenkins server by using **sudo systemctl daemon-reload**

**Step-3:**

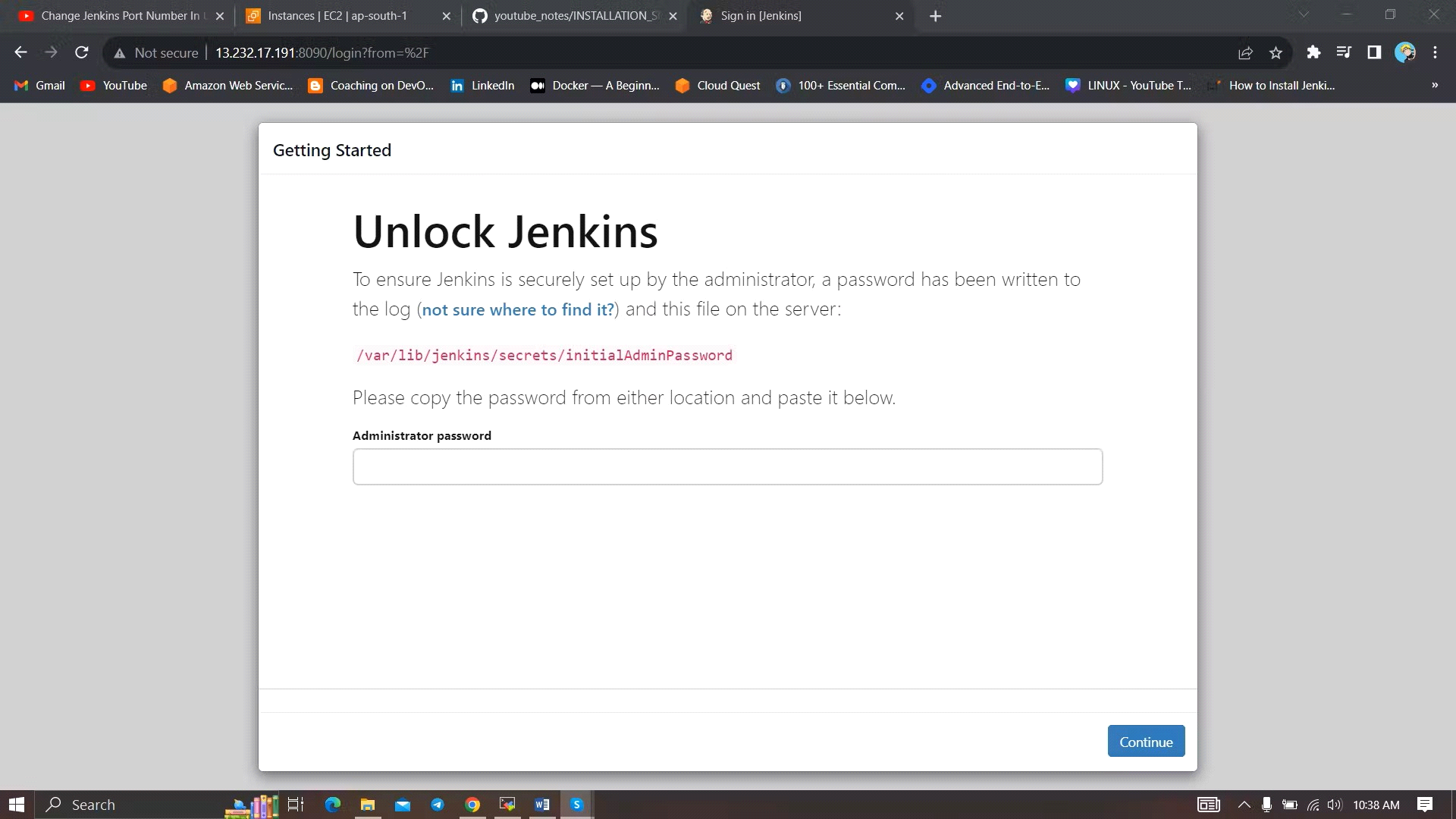
Jenkins dashboard creation

1.Then restart the jenkins server by using **sudo systemctl restart jenkins**

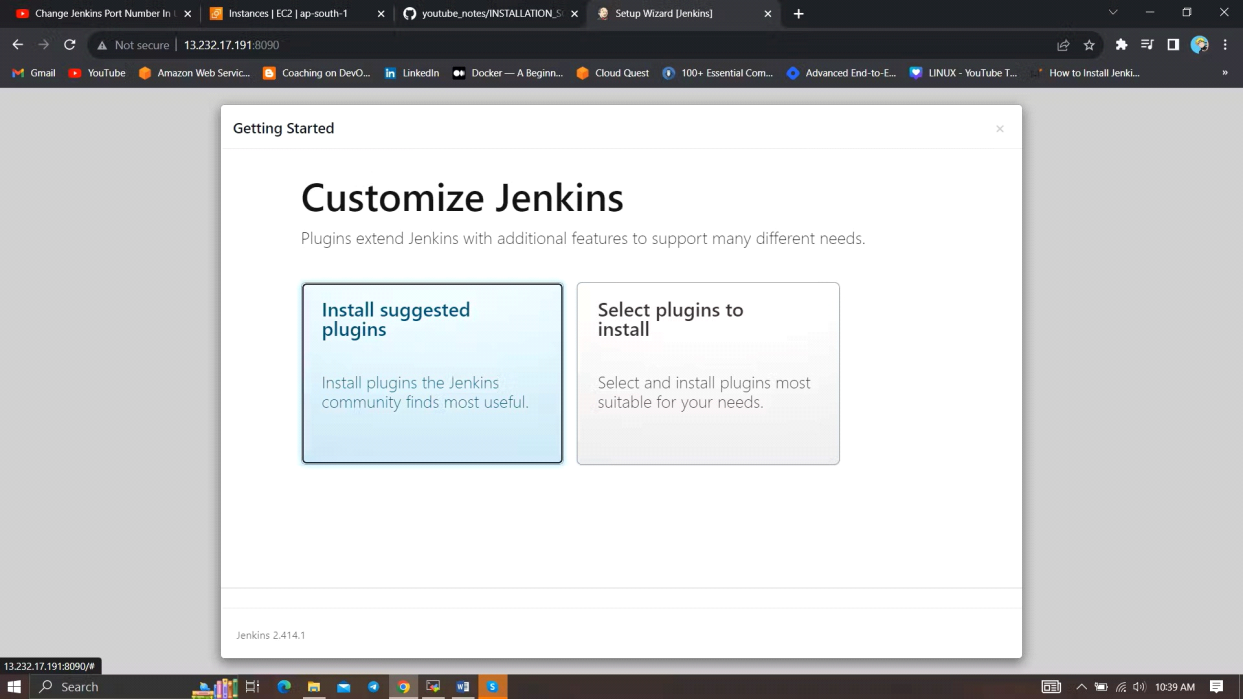
2.After restarting the service then connect the with 8090 port

3.Now Unlock Jenkins using an administrative password

4.To get need to copy var path and paste in Git by using **sudo cat var path** to get administrative password

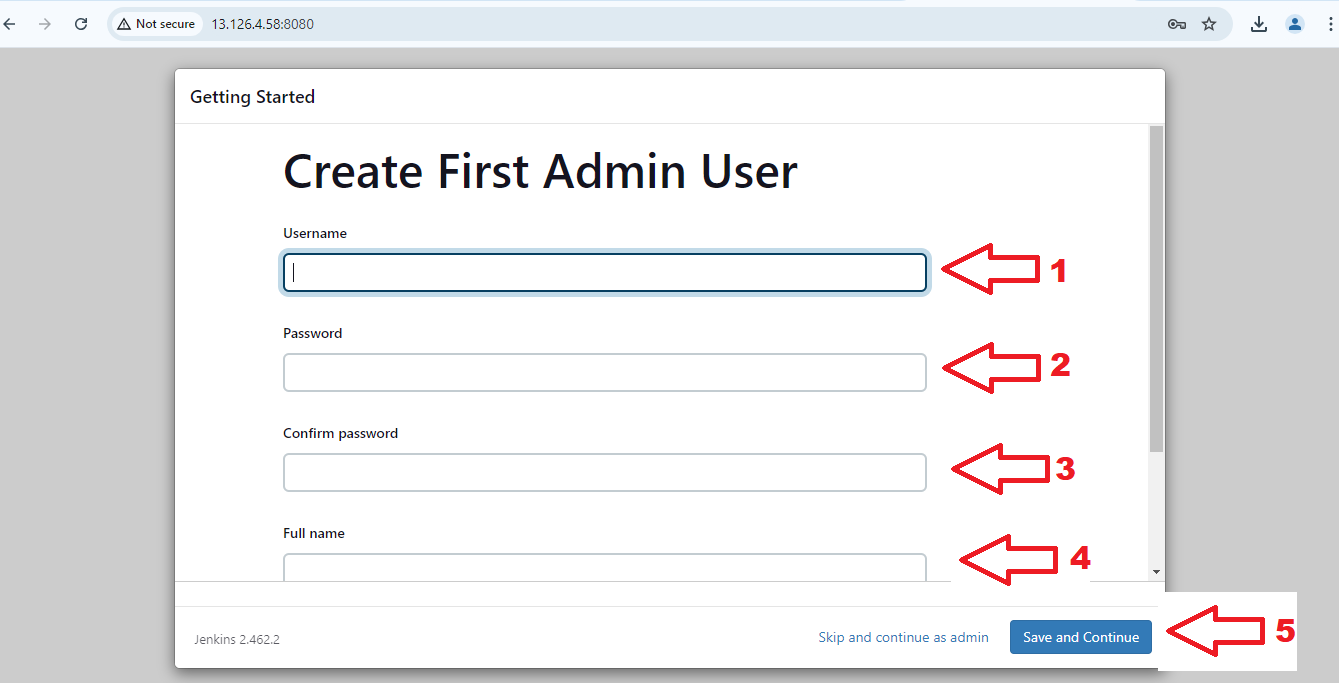


5.After unlocking the jenkins click on install the suggested plugins.

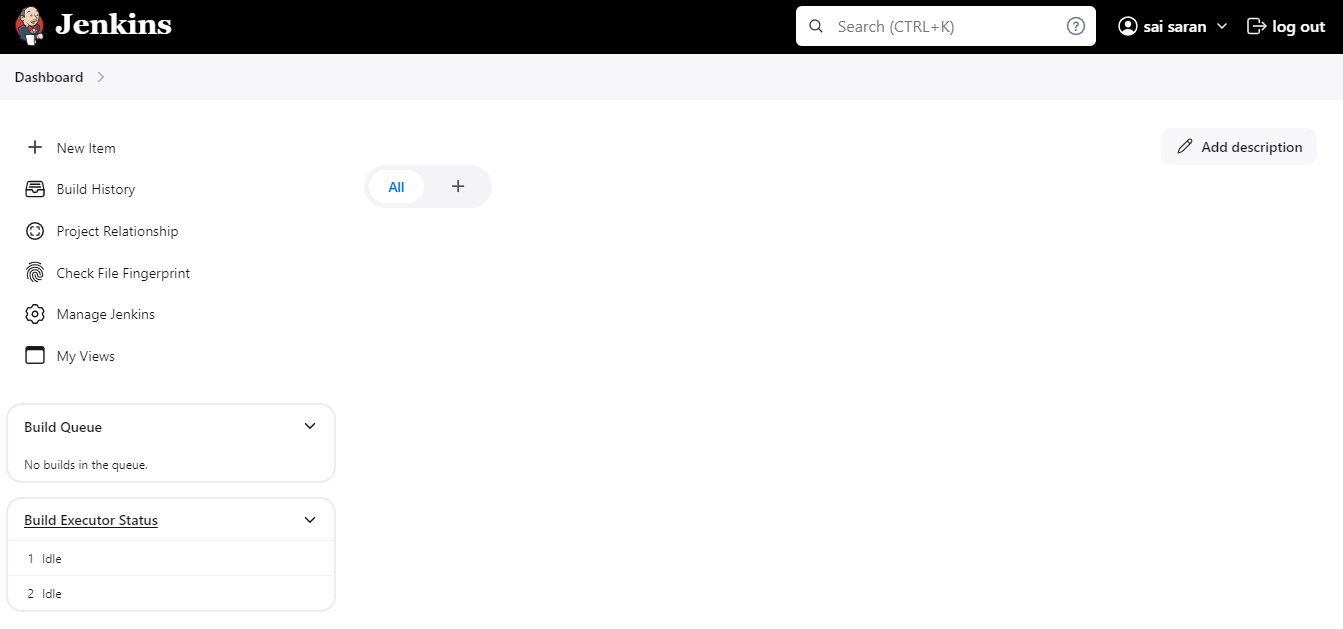


6.After successfully installation suggested we will get login page in that need to the enter below details

* User name
* Password
* Conform Password
* Full name
* Email
* After the all details need to
* Click on save and continue



1. After click on save and continue Jenkins dashboard page will open



8.After created Jenkins Dashboard again got the Instance and install the docker

**What is Docker?**

Docker is an open-source platform that allows developers to automate the deployment, scaling, and management of applications inside lightweight, portable containers. These containers package an application with all its dependencies, ensuring that it runs consistently across different environments, such as development, testing, and production.

**Step-4:**

1.Need to Install Docker Instance and connect to the sonarQube with Docker Container by using below commands

2.**sudo apt-get update**

3.After update the tools need install Docker by using **sudo apt-get install docker.io -y**

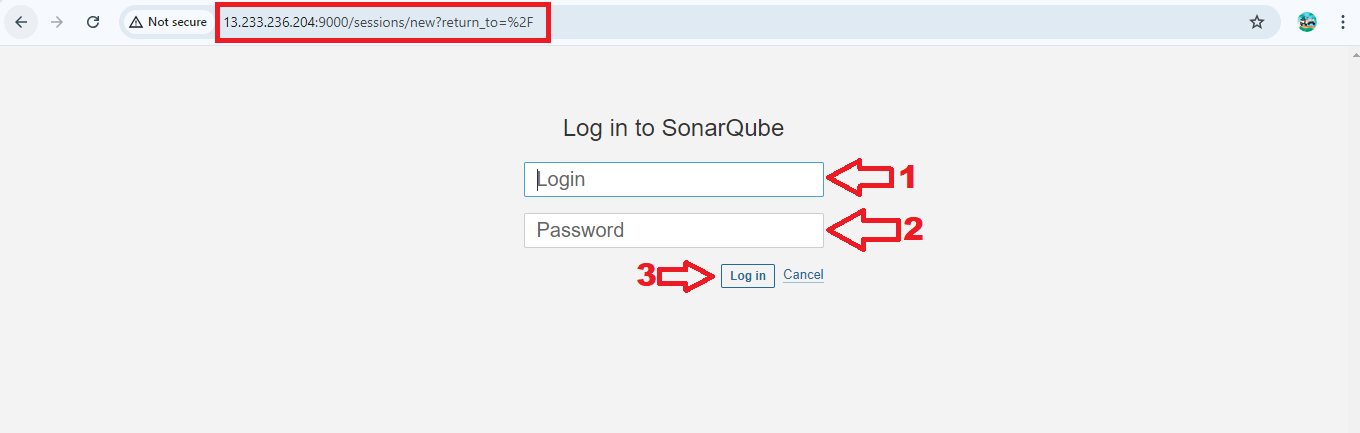
4.After Installing the docker create user in usermod by using **sudo usermod -aG docker $USER #my case is ubuntu**

5.Then New group for docker by using **newgrp docker**

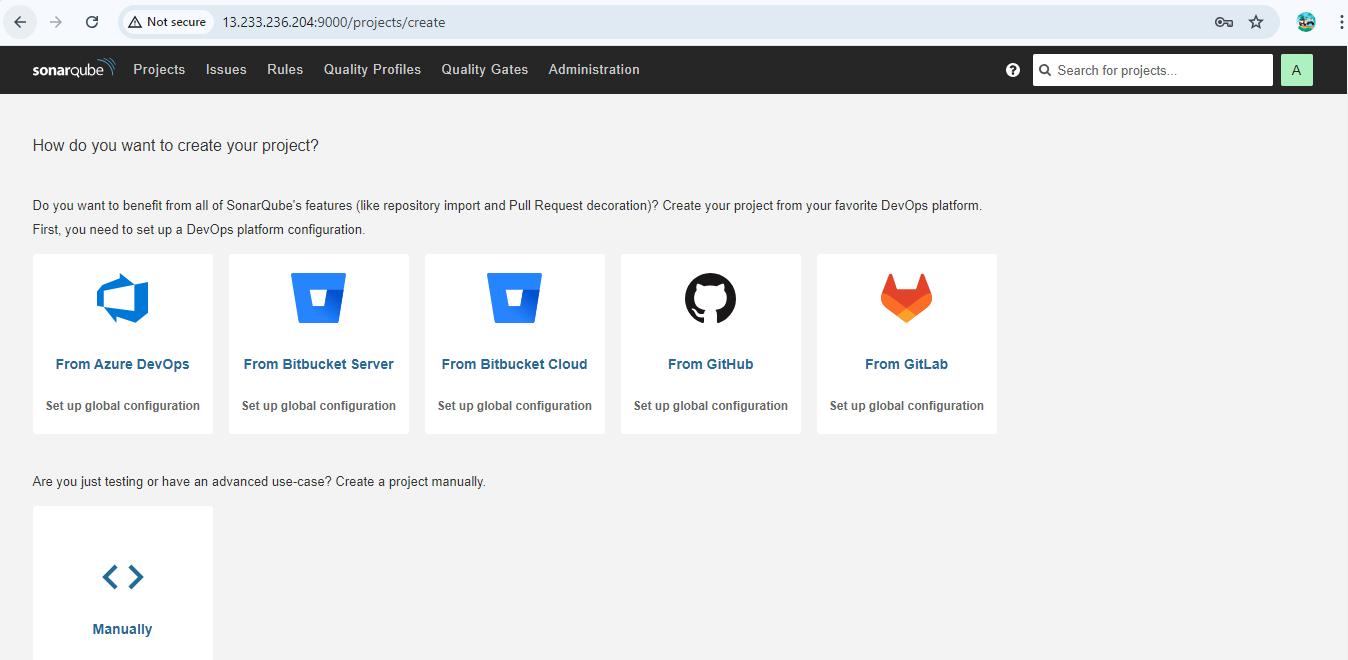
6.After the docker installation, we create a sonarqube container by using **docker run -d --name sonar -p 9000:9000 sonarqube:lts-community** Then connect the SonarQube by using Instance **public IPV4address:9000** (SonarQube default port number)

Then login with login

1. Enter username as **admin**
2. Enter password as **admin**
3. Then click on **login**



After click on the Login the SonarQube dashboard will open



**Step-5:**

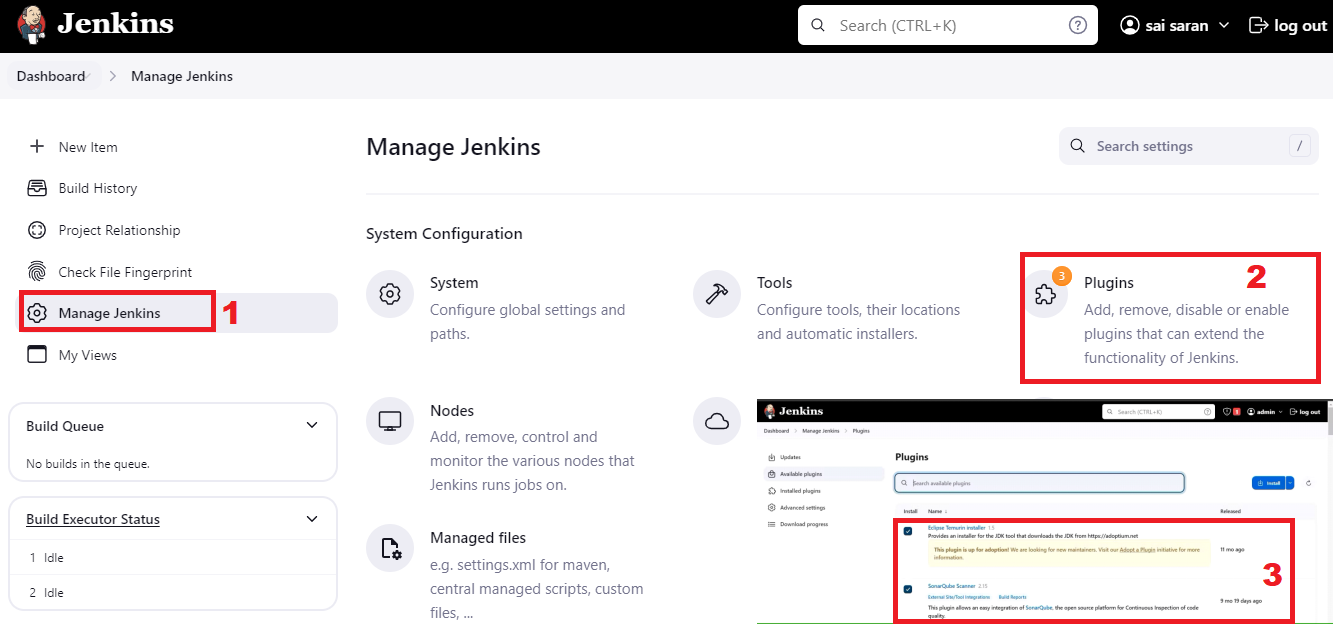
After completing of SonarQube connectivity need to go to the jenkins dashboard and start build the project

**Step-6:**

First login into Jenkins dashboard and the go to the manage Jenkins and select plugins in plugins need to select available plugins and install the required plugins

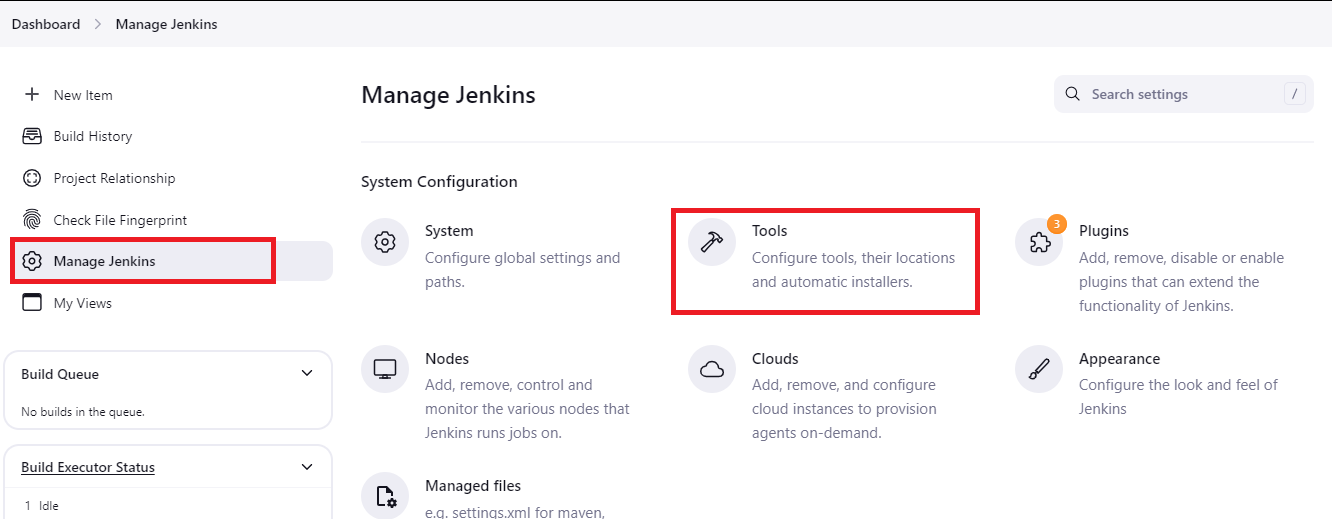
1.Eclipse Temurin Installer (Install without restart)

2.SonarQube Scanner (Install without restart)



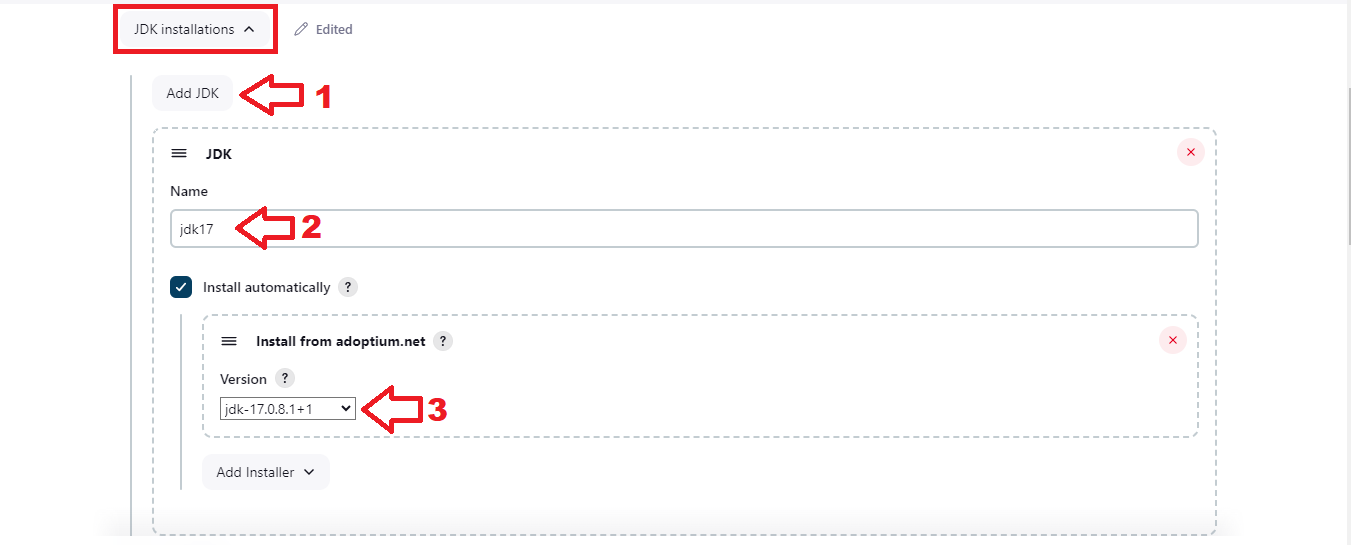
**Step-9:**

Then again go to the Manage Jenkins and select Tools.



In Tools need to add jdk installation and maven installation

1. Add jenkins Installation
2. Name
3. Java Version (jdk-17.0.8.1+1)



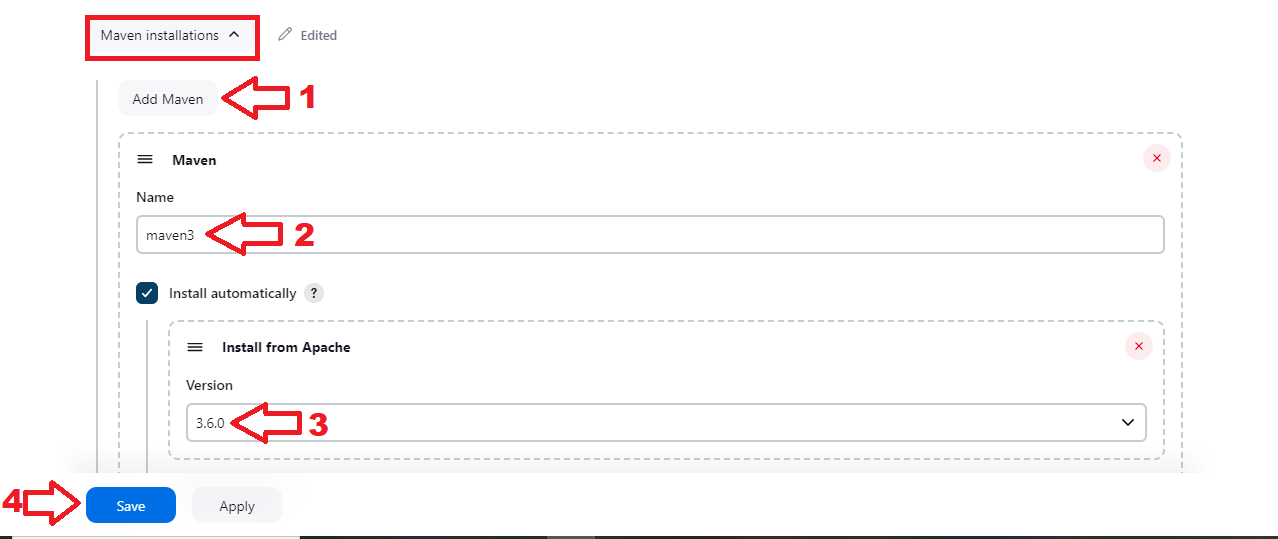
Then again In Tools need to add maven installation and maven installation

1.Add maven

2.Name

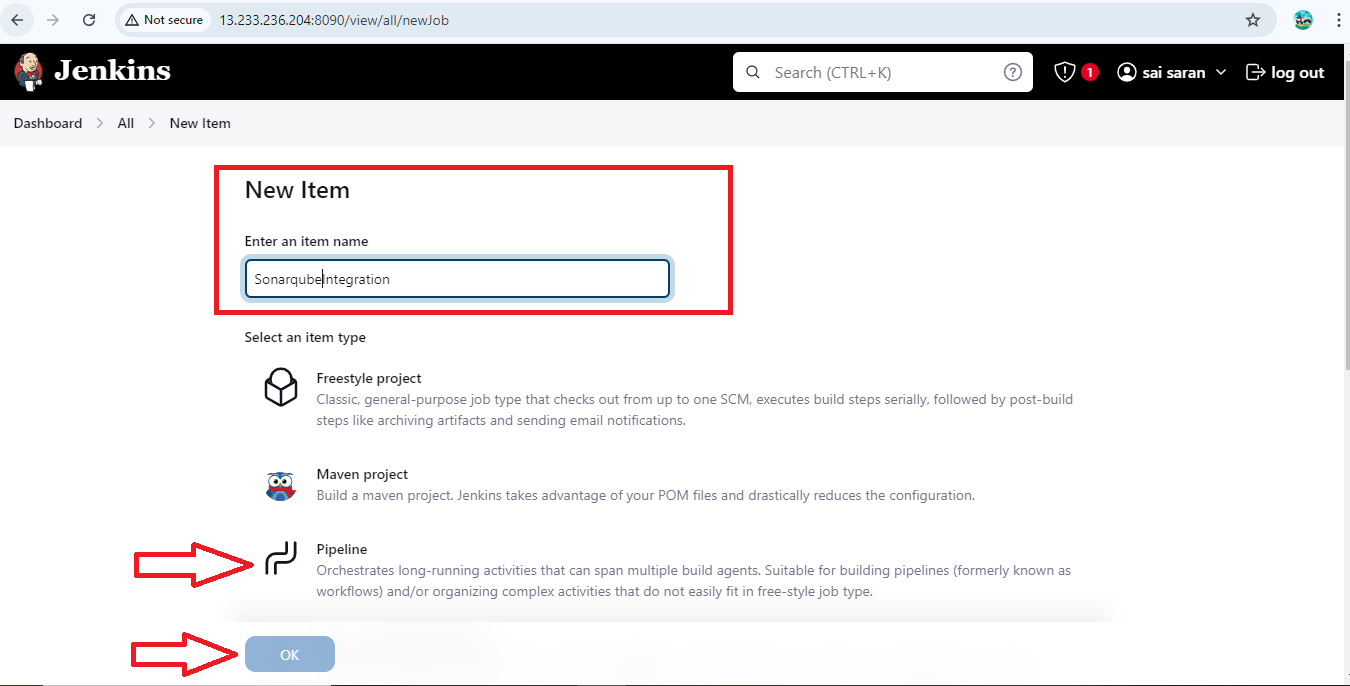
3.Maven Version (3.6.0)

4.Then Apply and Save



**Step-10:**

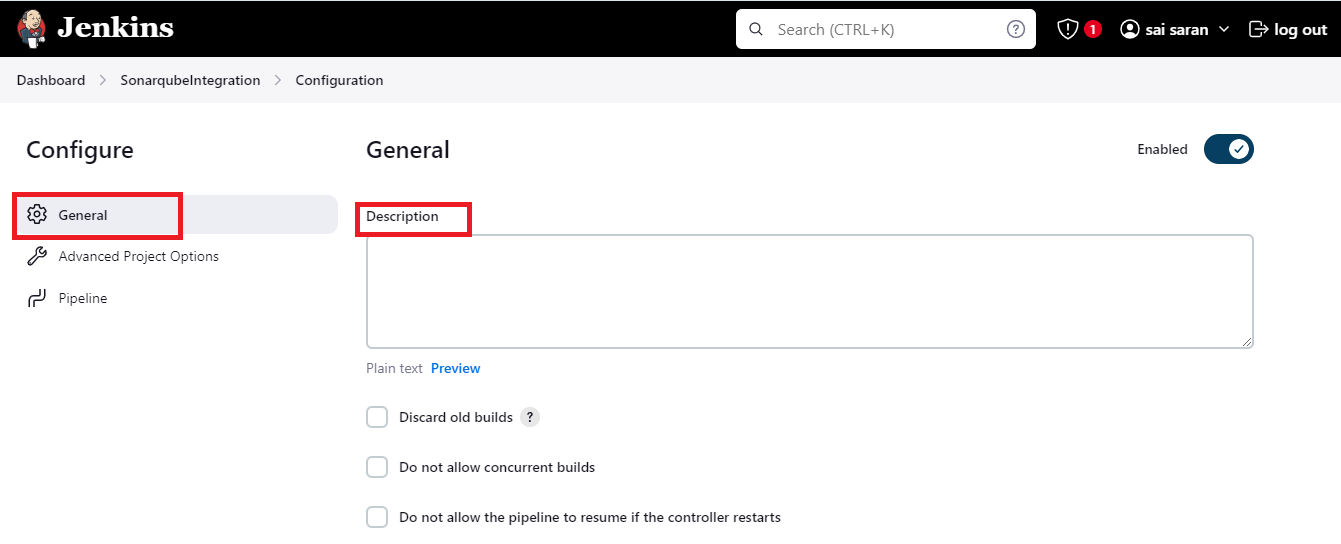
After providing all the information need to Create a new project with Pipeline



**Step-11:**

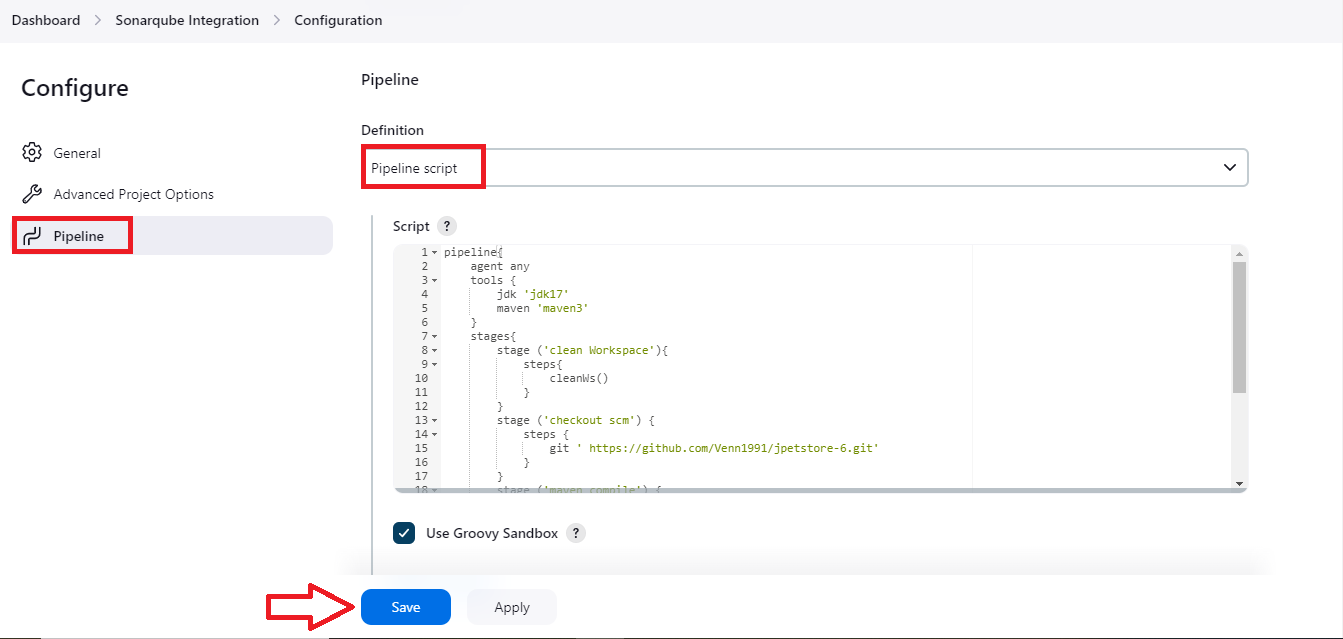
After Click on Ok the configure page will open

1. In General Tab need to provide the decesription



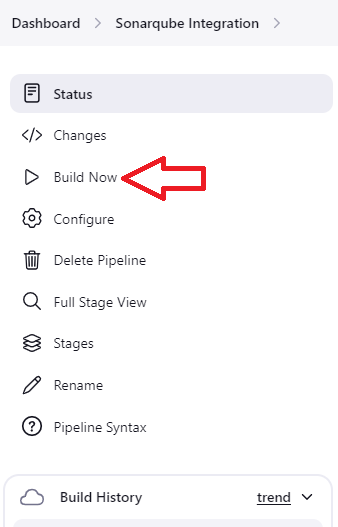
**Step-12:**

Then go the Pipeline in Configure and then provide the pipeline script Then Apply and Save.



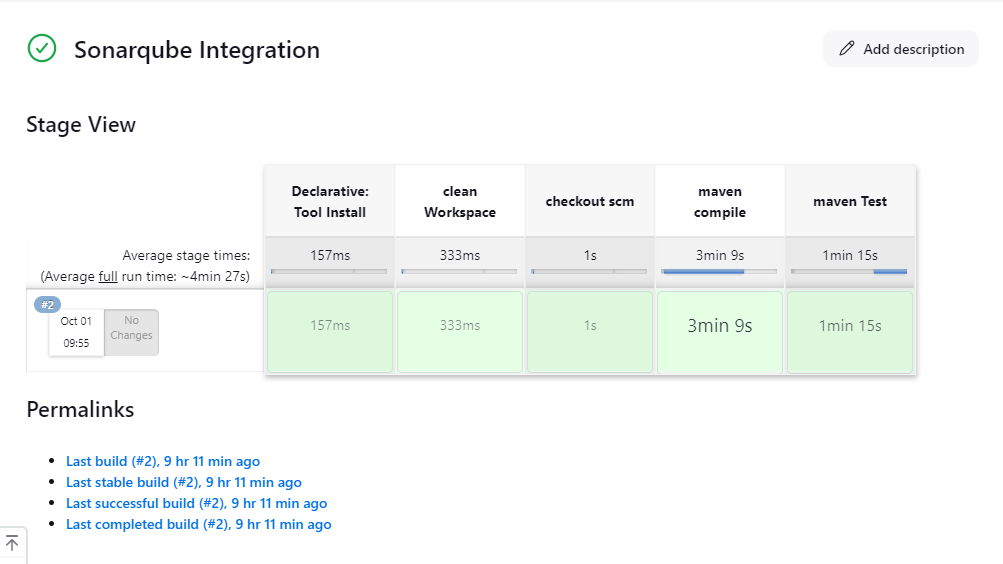
**Step-13:**

After save the configure page then click on Build Now.



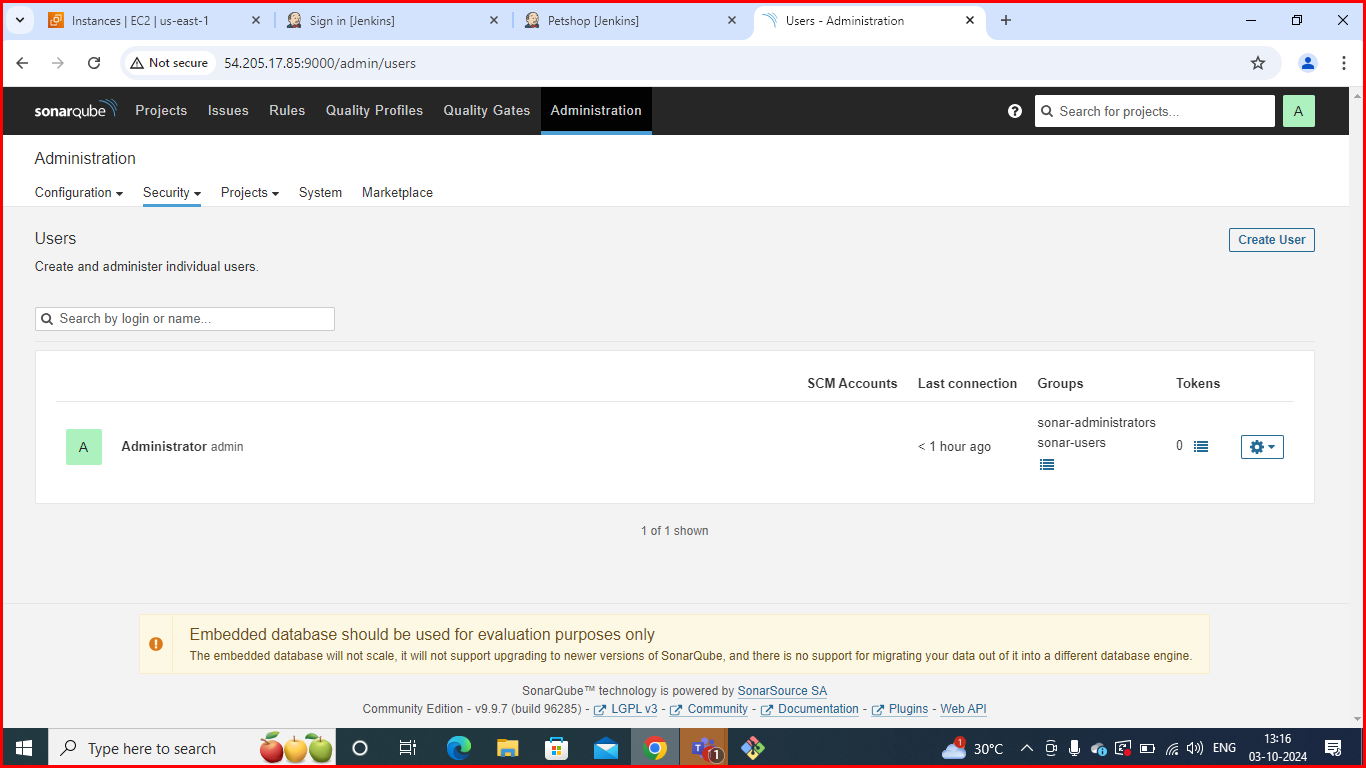
**Step-14:**

After Click on Build Now you will get success output.

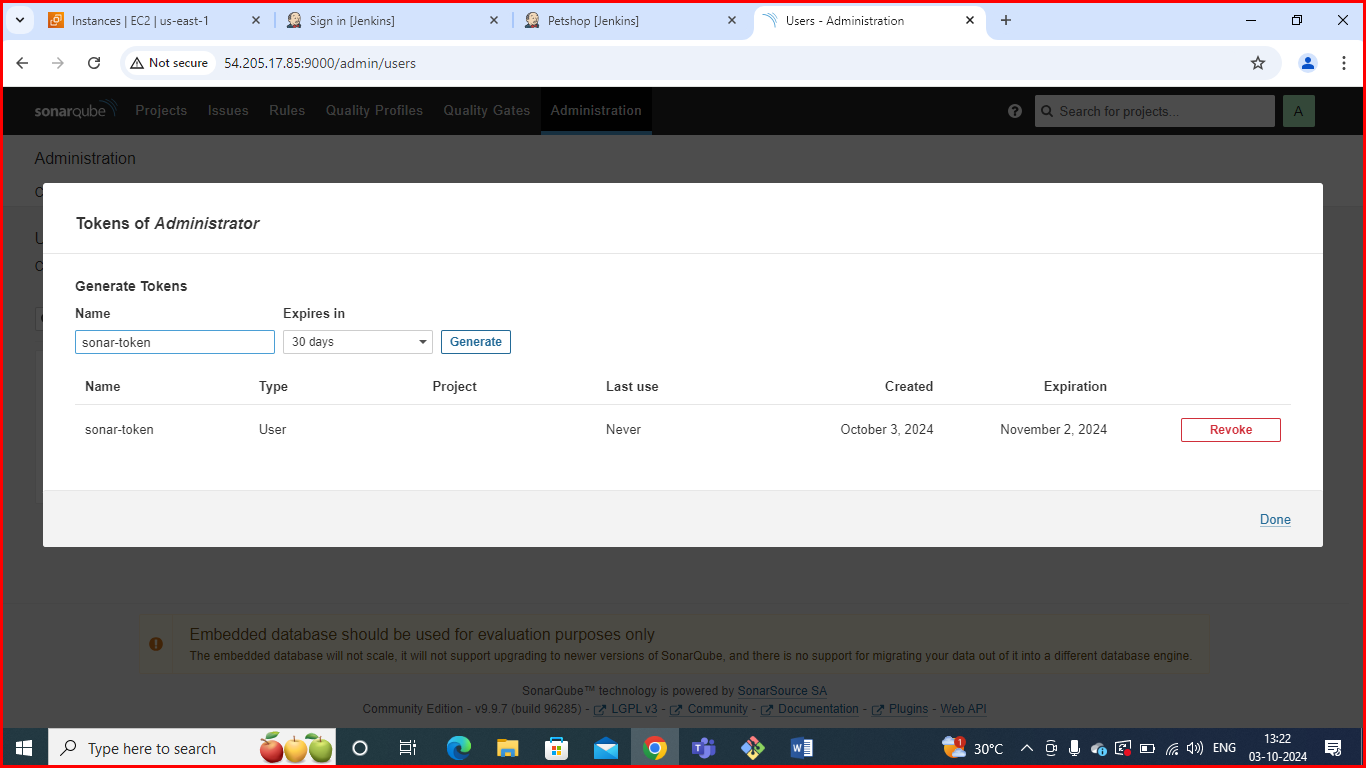


**Configure Sonar Server in Manage Jenkins**

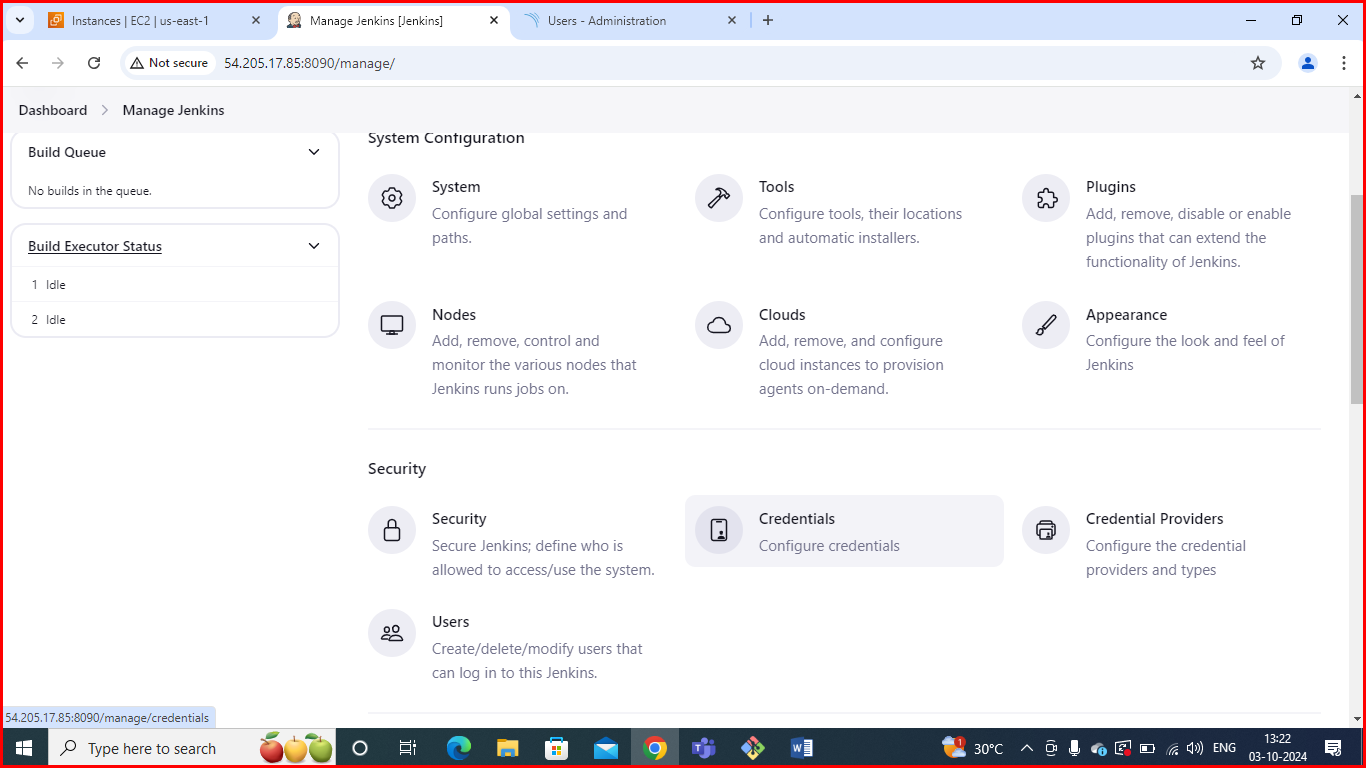
Create a token with a name and generate:



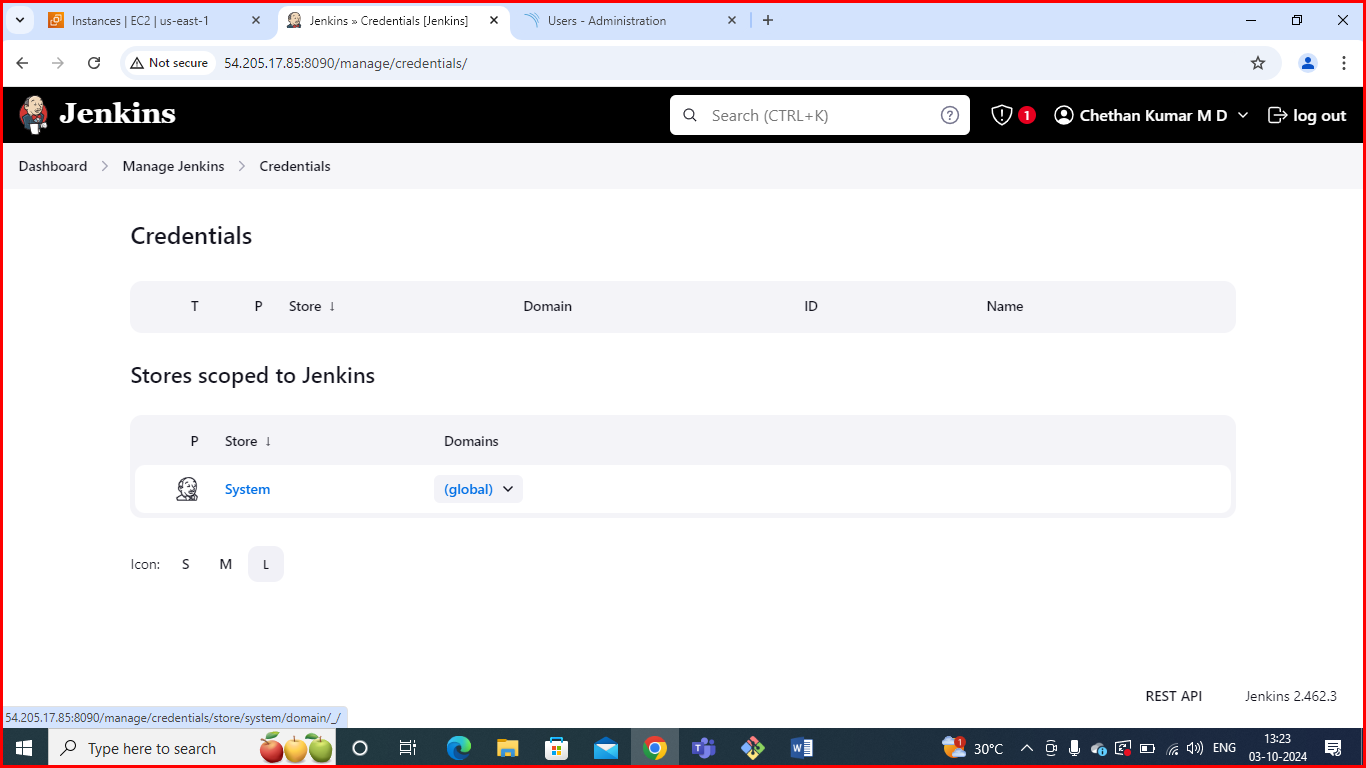
**Copy token:**



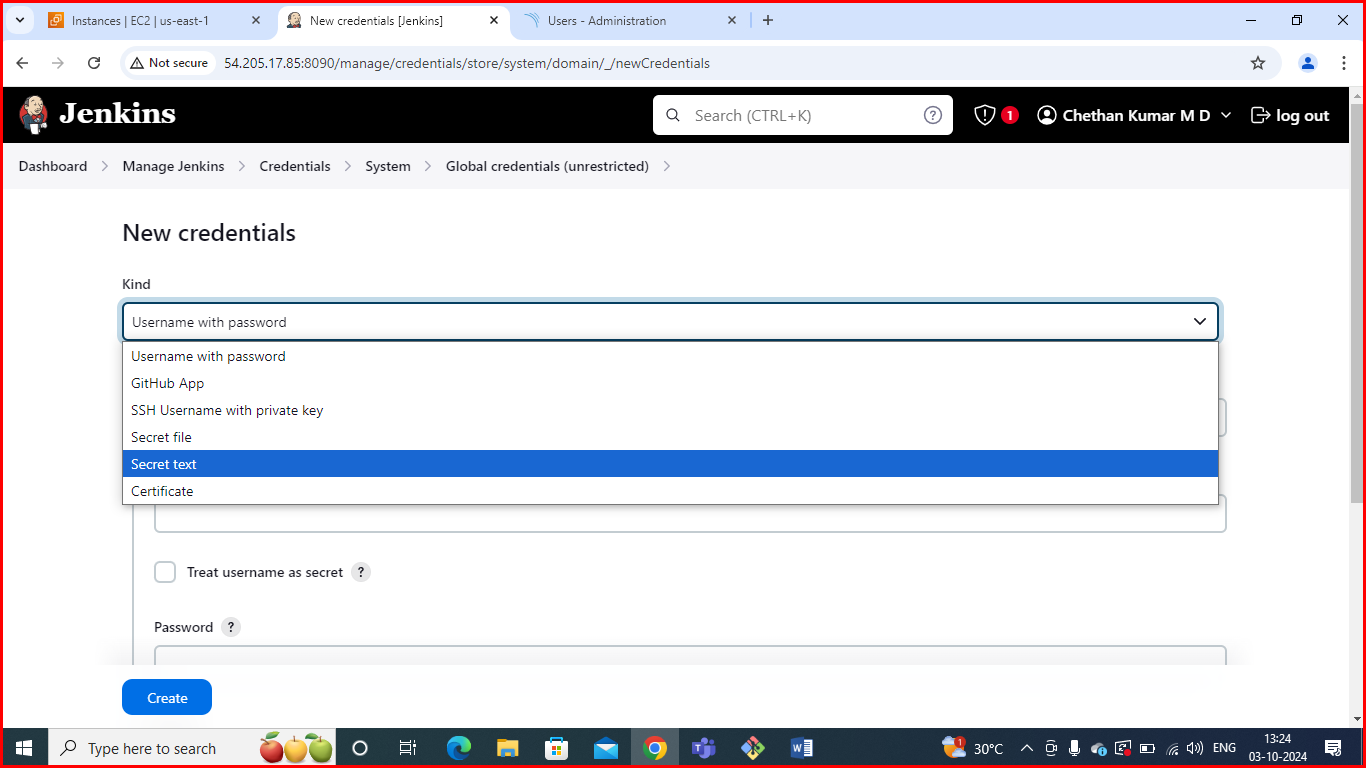
**Go to Jenkins Dashboard → Manage Jenkins → Credentials**



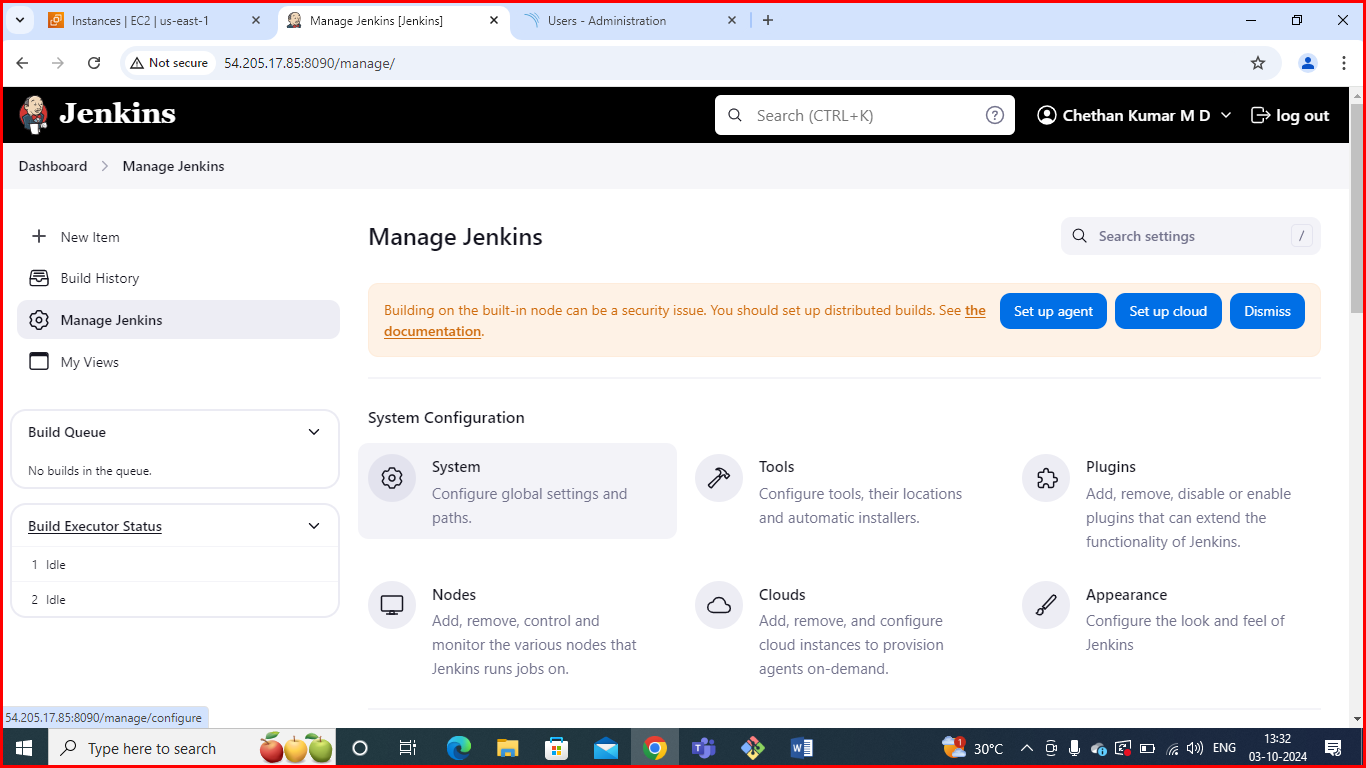
**In credentials click on global :**



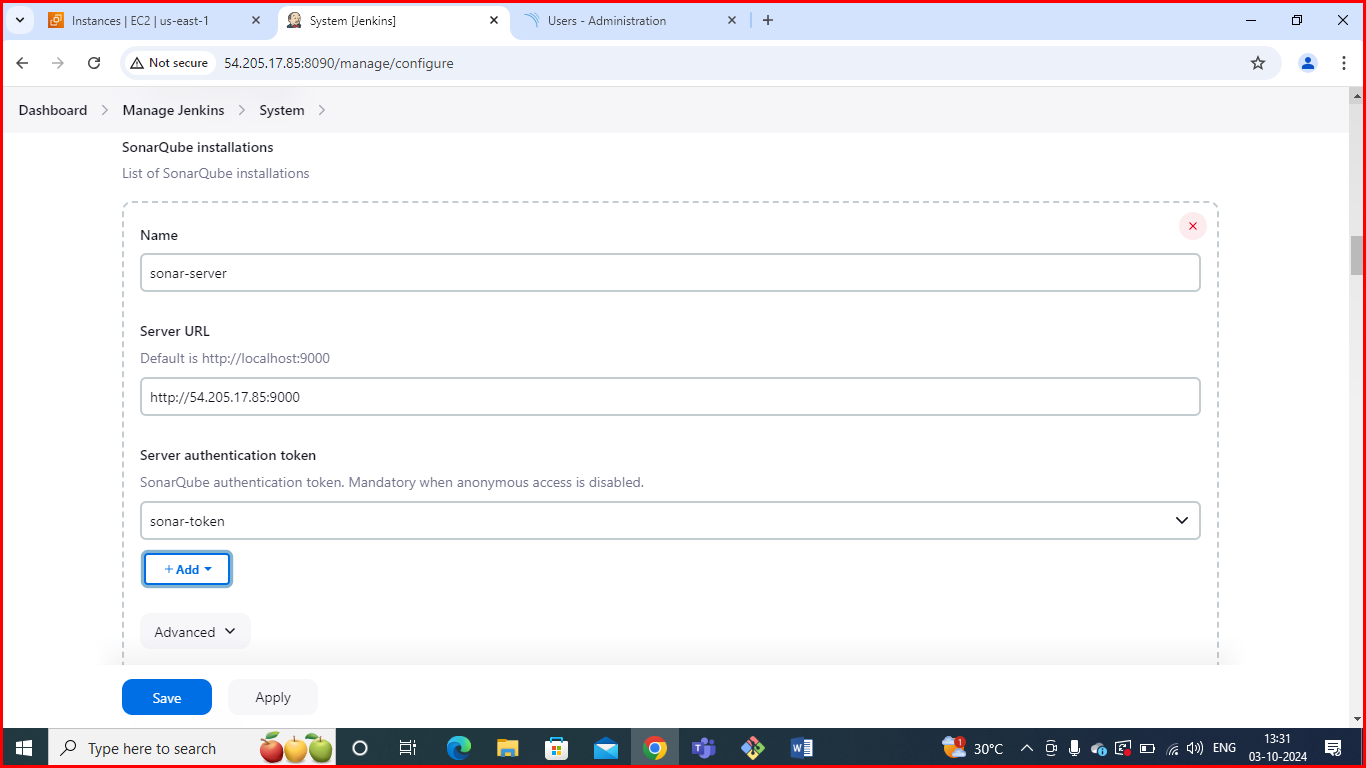
**Add Secret Text:**



**Now, go to Dashboard → Manage Jenkins → System**



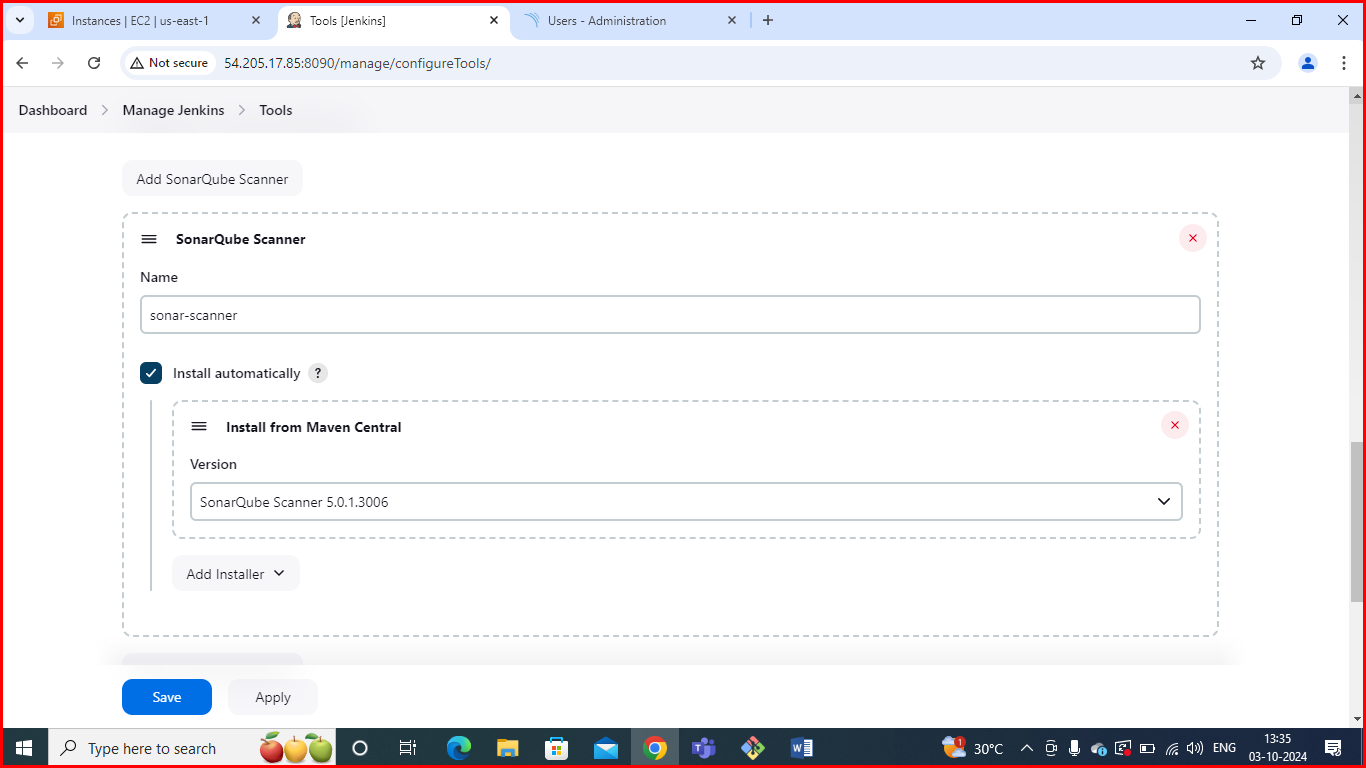
**Here give name and paste url of sonar cube.**



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

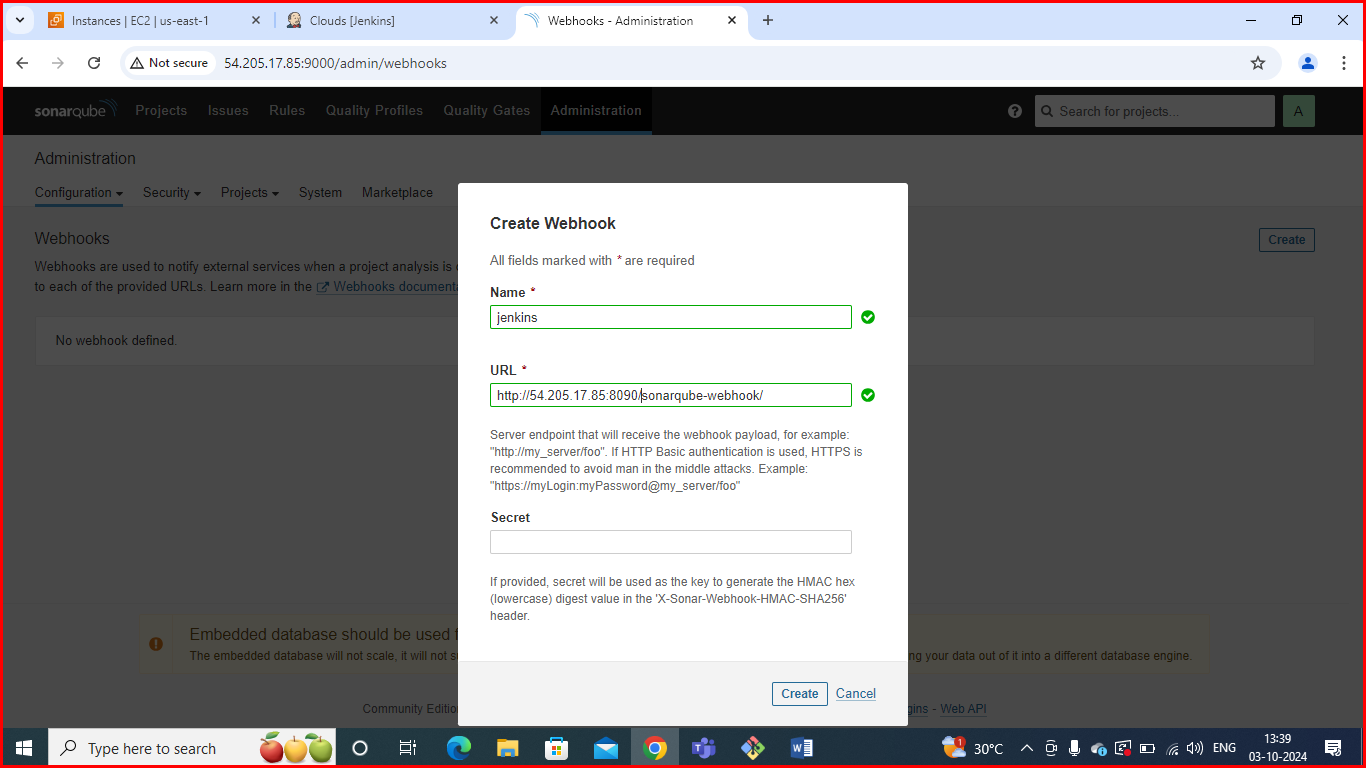
**Global Tool Configuration** is used to configure different tools that we install using Plugins

We will install a sonar scanner in the tools.

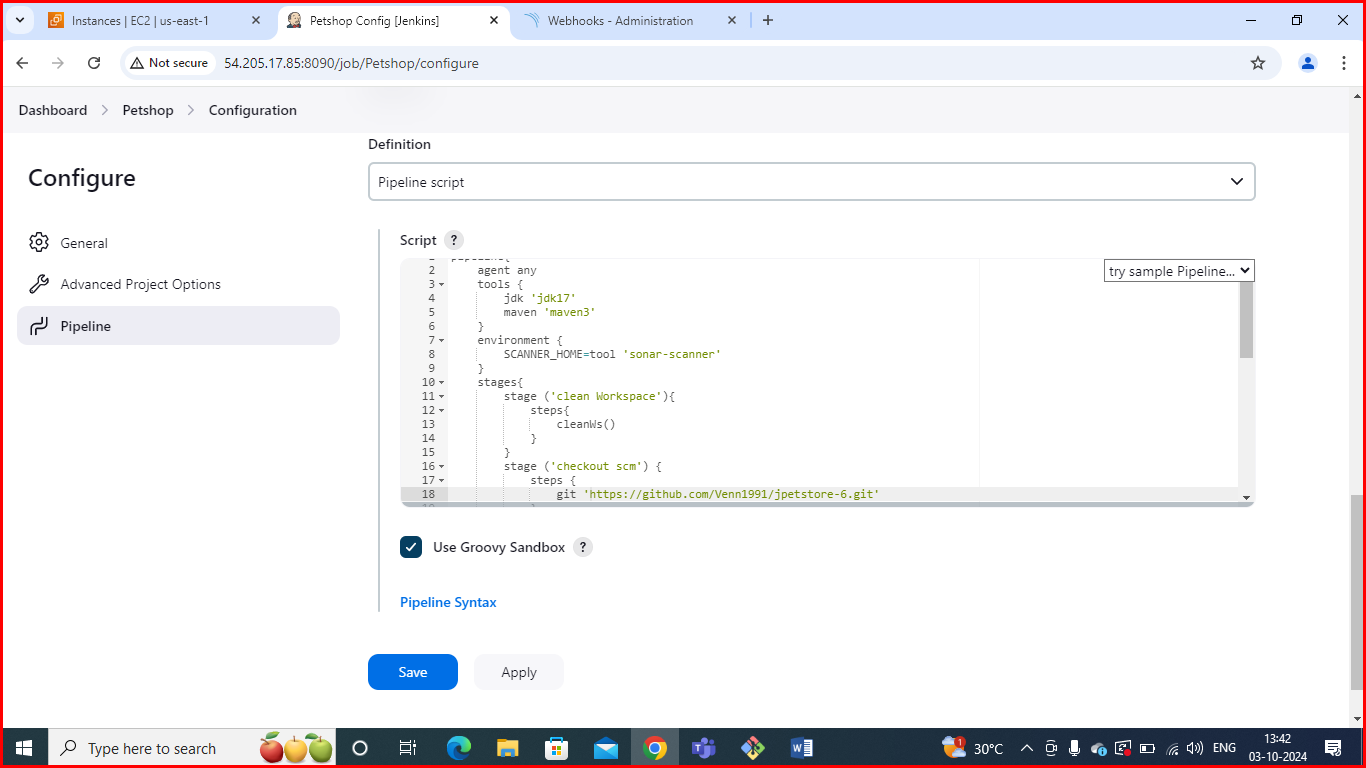


In the Sonarqube Dashboard add a quality gate also

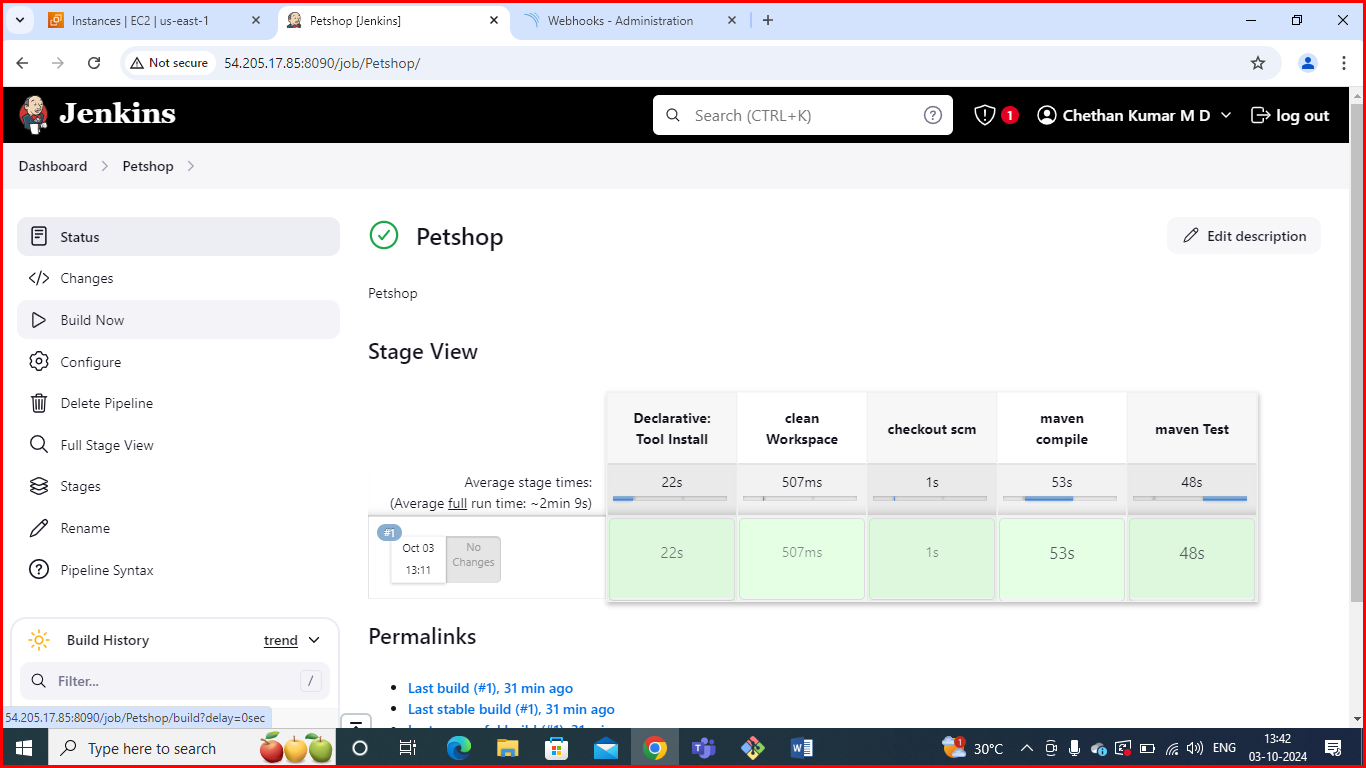
**Administration--> Configuration-->Webhooks**



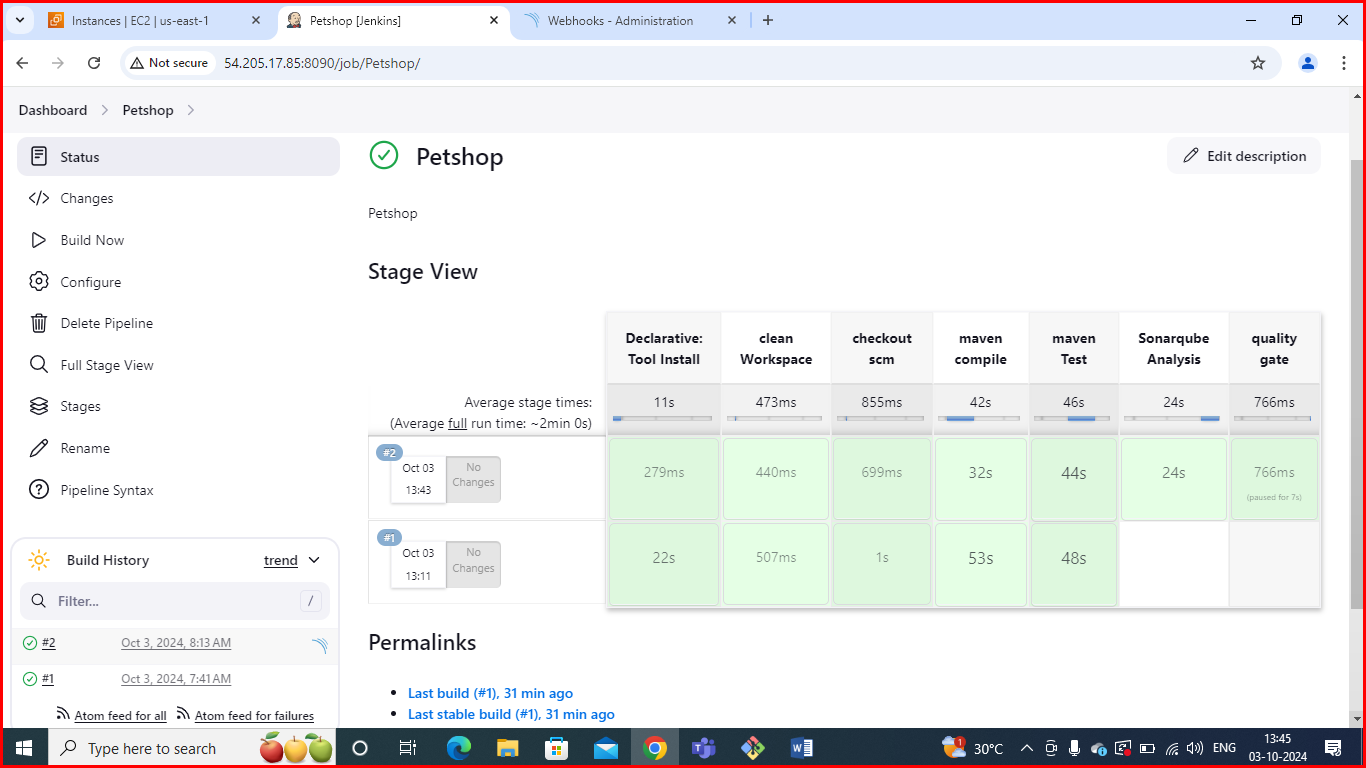
**Pipeline script**: In Jenkins, a pipeline script is used to define the steps and processes for continuous integration and continuous delivery (CI/CD).



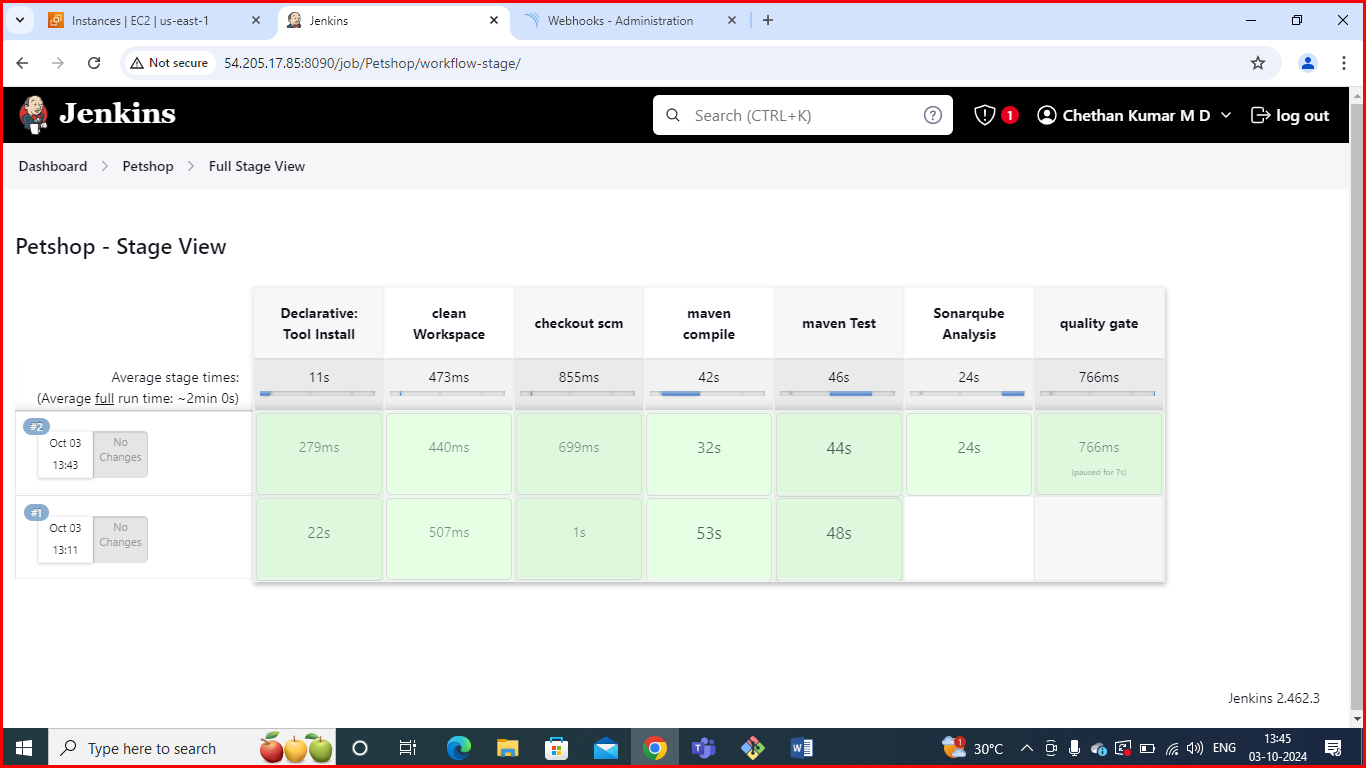
**Build now**: In Jenkins, the "Build Now" option allows you to manually trigger a build for a job or pipeline. Here’s how to use it and what to expect:



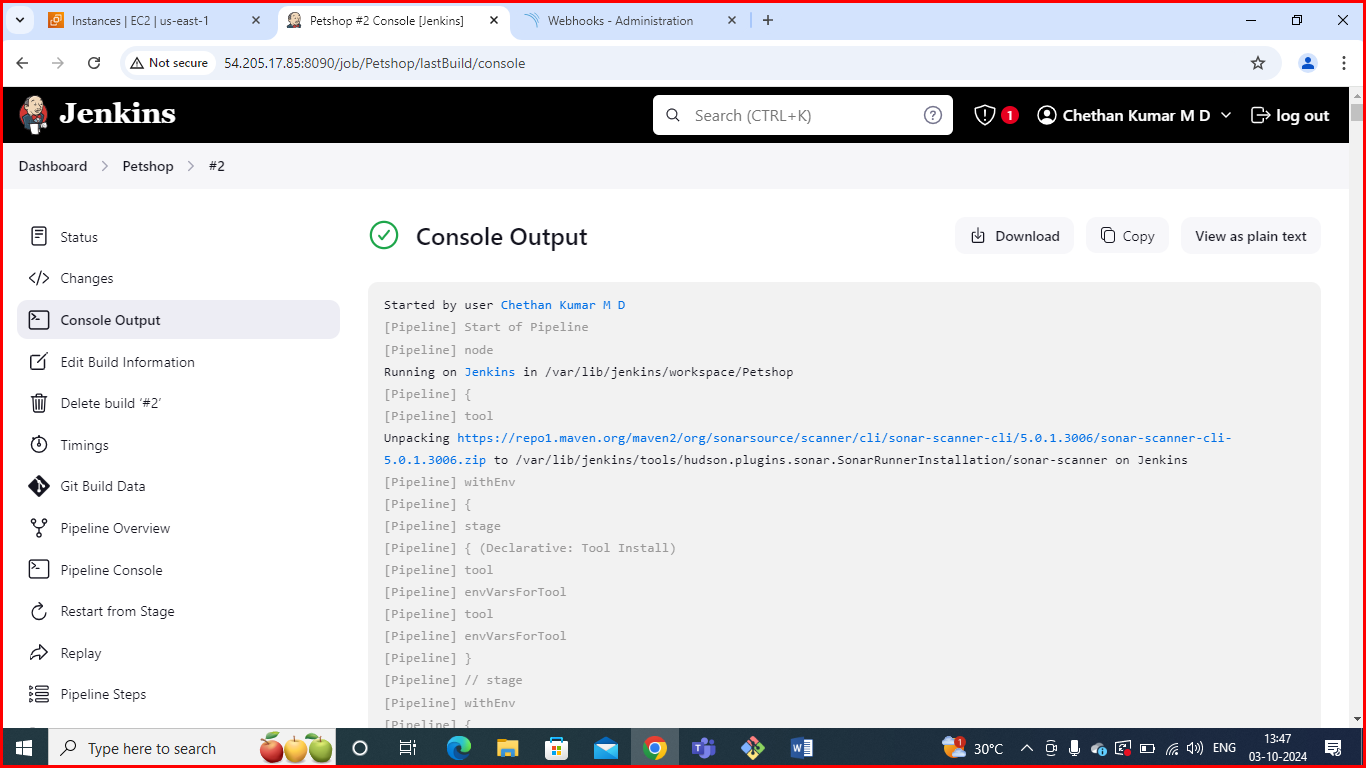
**Build success:** In Jenkins, a build is considered successful when all the defined stages and steps in the pipeline complete without errors. Here's what you need to know about successful builds in Jenkins:

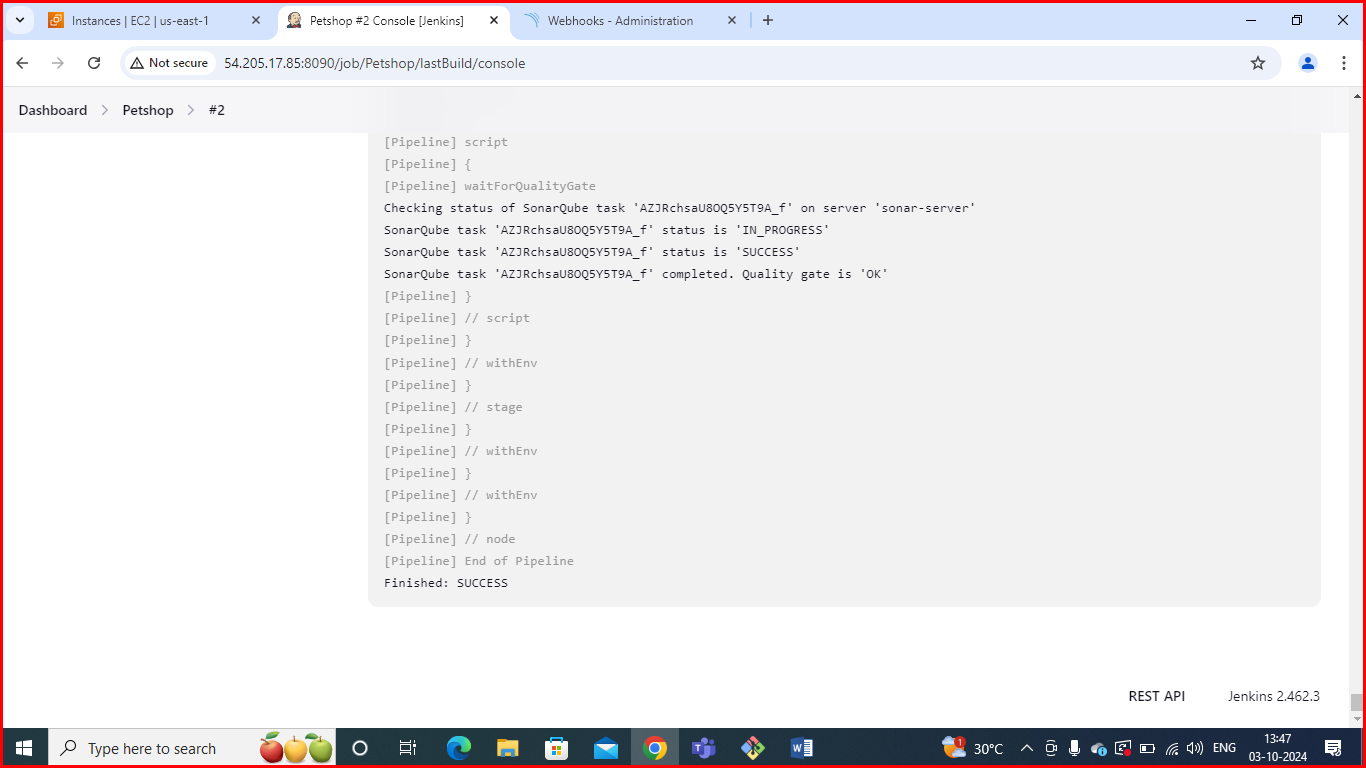


**Stage view:** The Jenkins Stage View provides a visual representation of the stages of a pipeline job. It helps you monitor the execution flow, see which stages have completed, and identify any failures in a more user-friendly way than raw console output.



**Console output:** Jenkins console output provides a real-time view of the build process for a job. It includes logs from each step of the build, allowing you to monitor progress, see error messages, and debug issues.





To see the report, you can go to Sonarqube Server and go to Projects.

You can see the report has been generated and the status shows as passed. You can see that there are 6.7k lines. To see a detailed report, you can go to issues. 