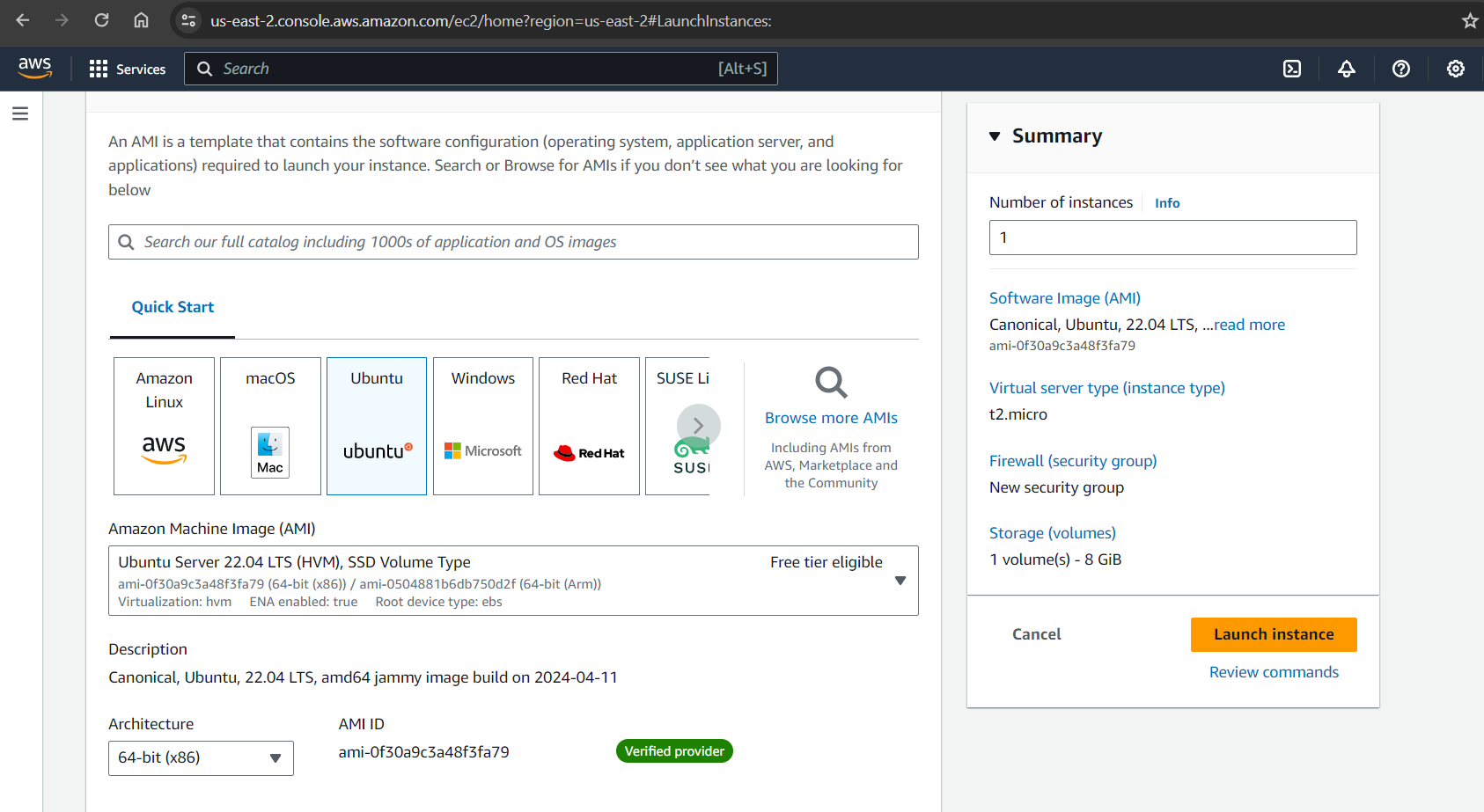
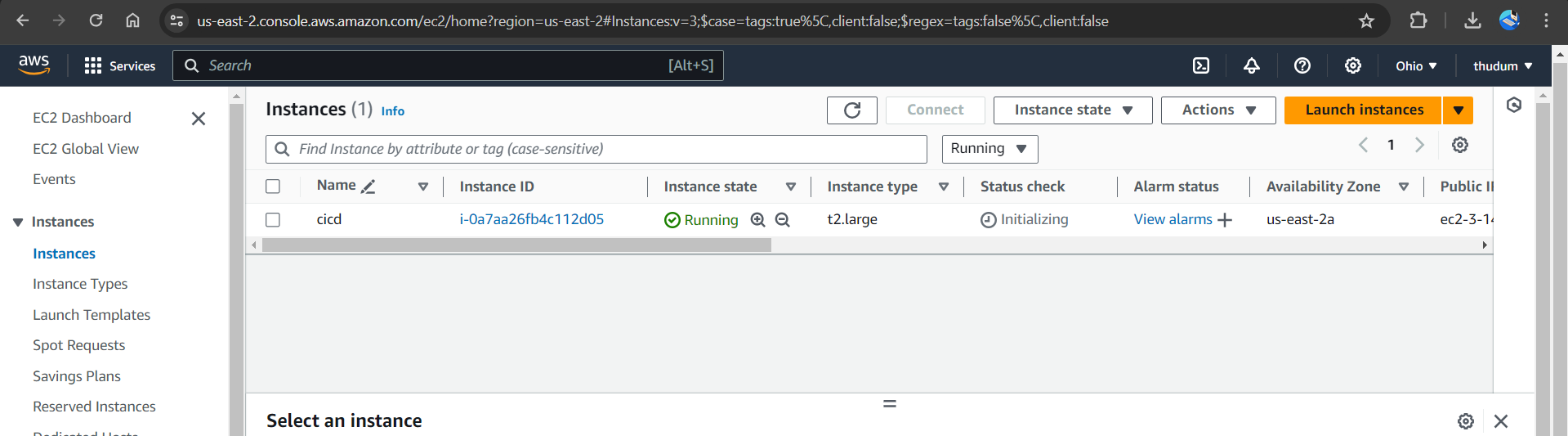
**AWS & DEVOPS INTERNSHIP PROJECT-6**

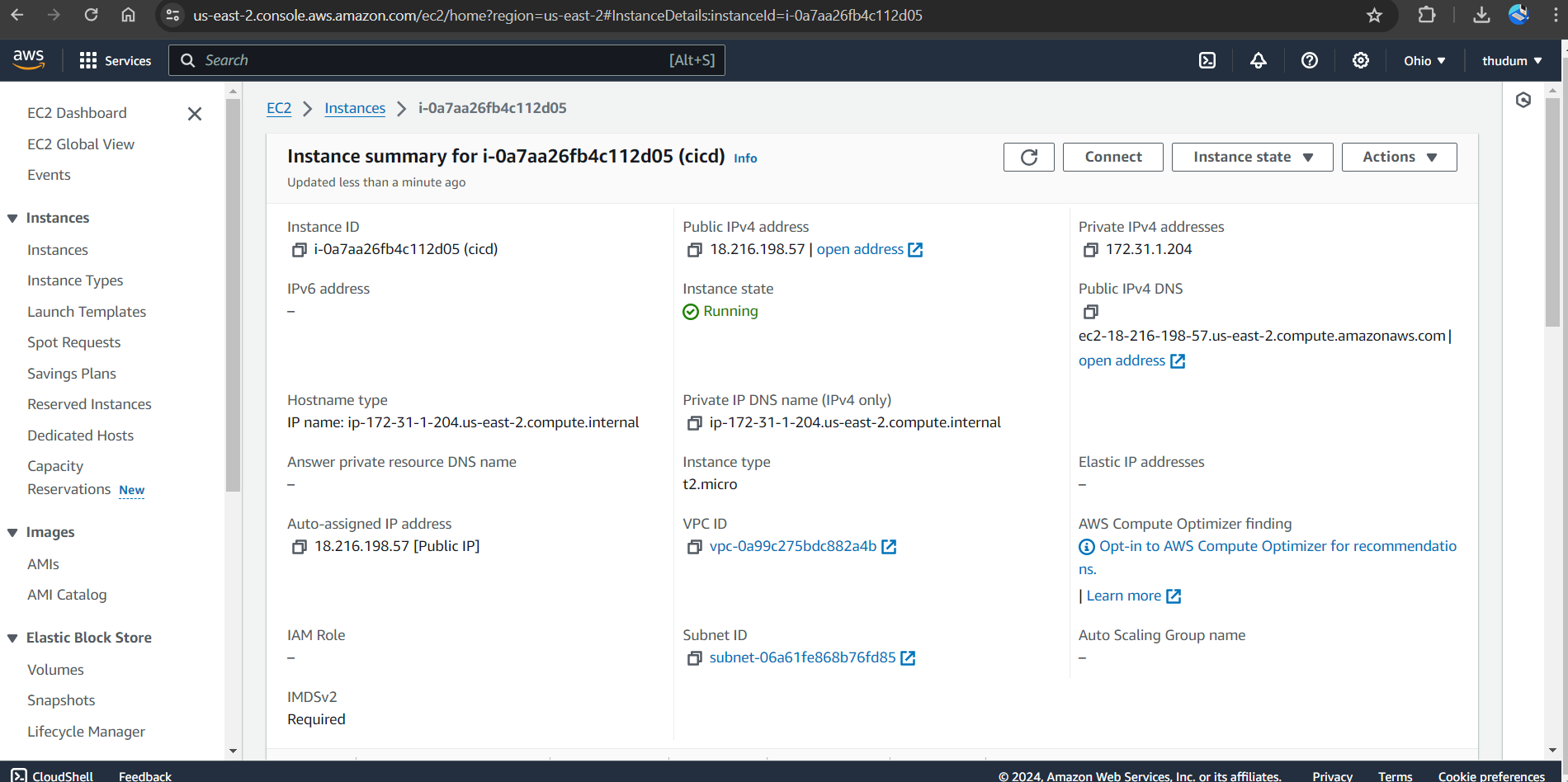
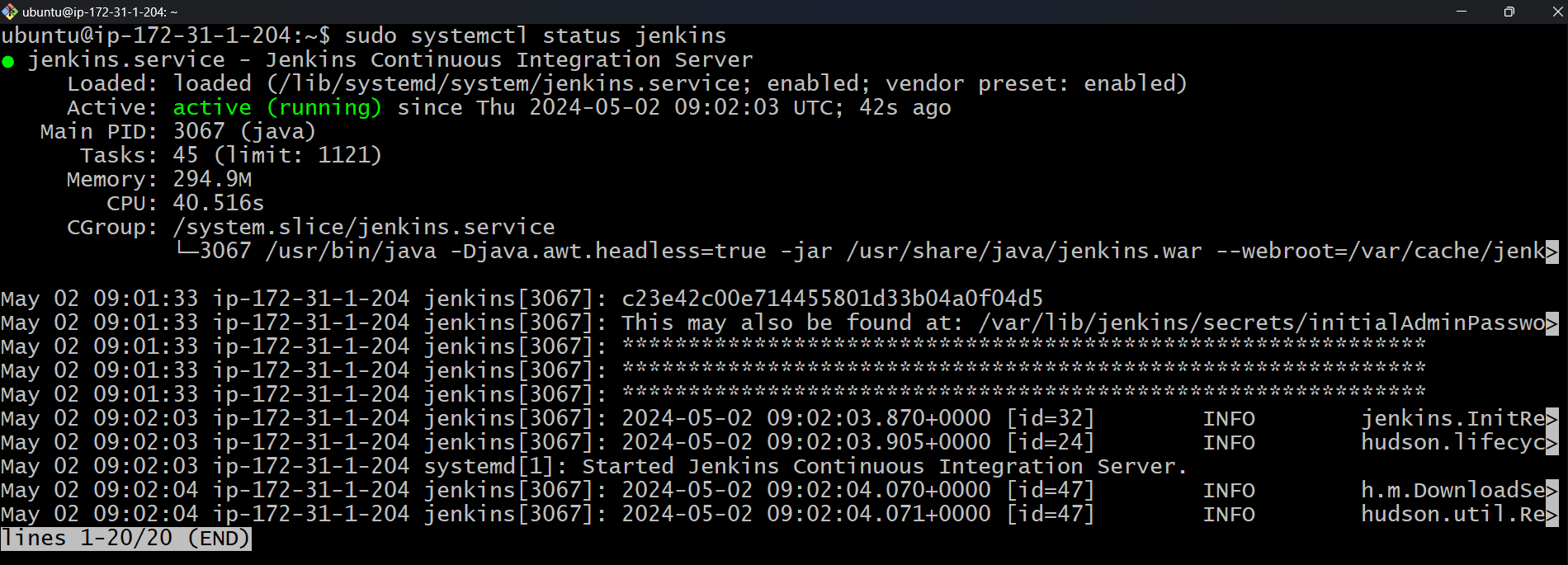
**JENKINS CI-CD PIPELINE WITH ANSIBLE**

* Create an Ubuntu (22.04) T2 large instance.

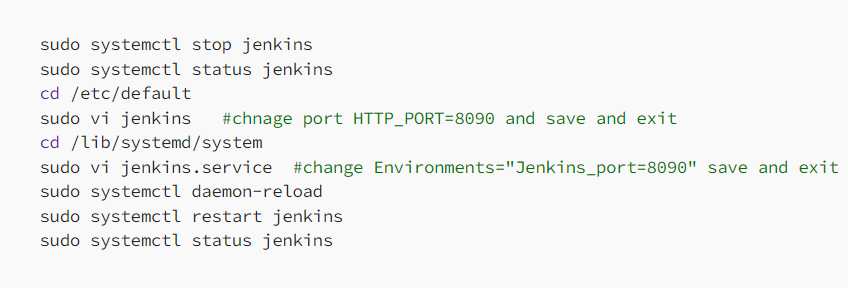


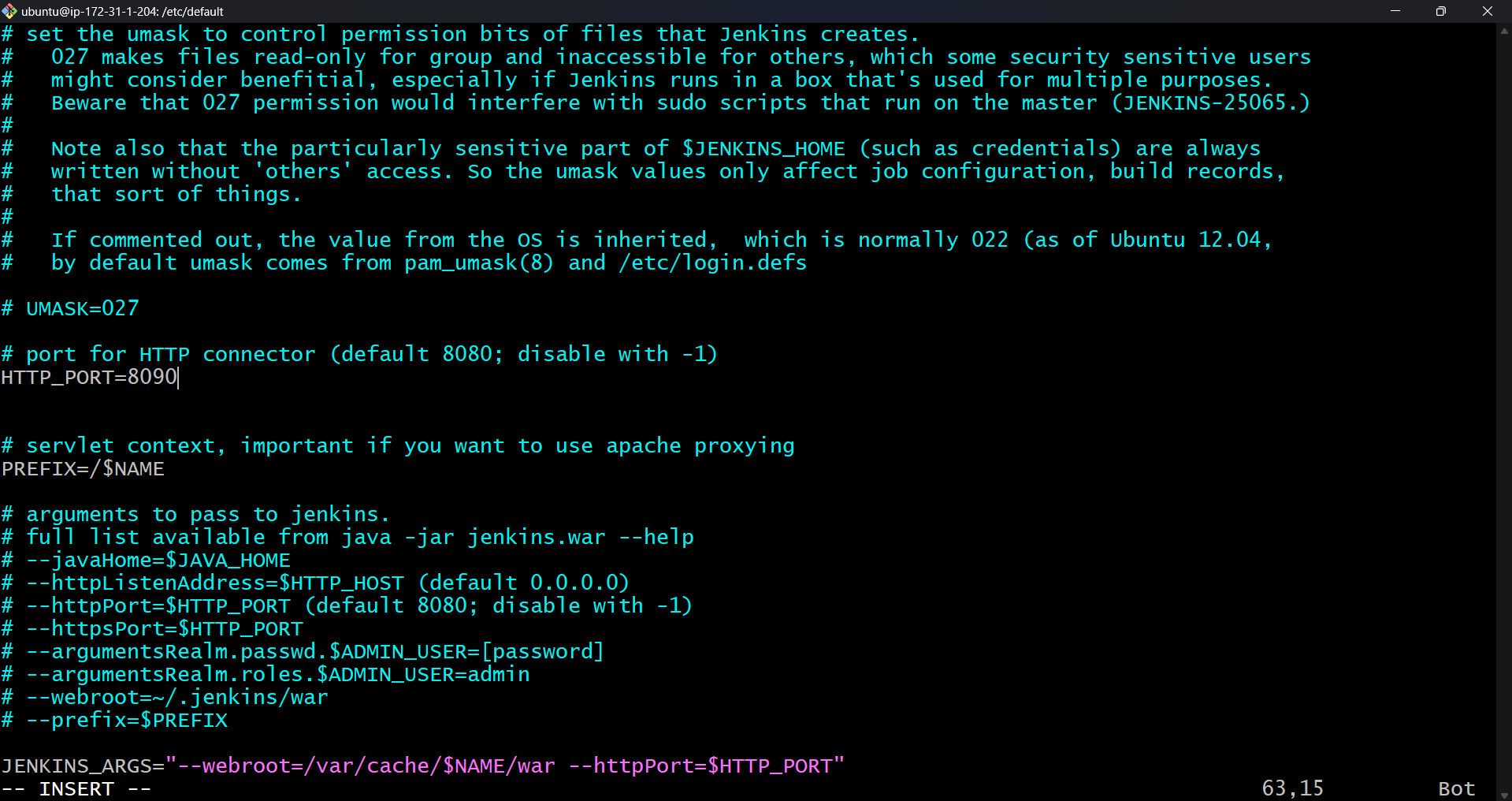
* Connect to EC2 instance, and enter below commands to install Jenkins.
* Once the Jenkins is installed, you will need to go to your AWS EC2 security group and open Inbound port 8080 and 8090, 9000 for SonarQube.

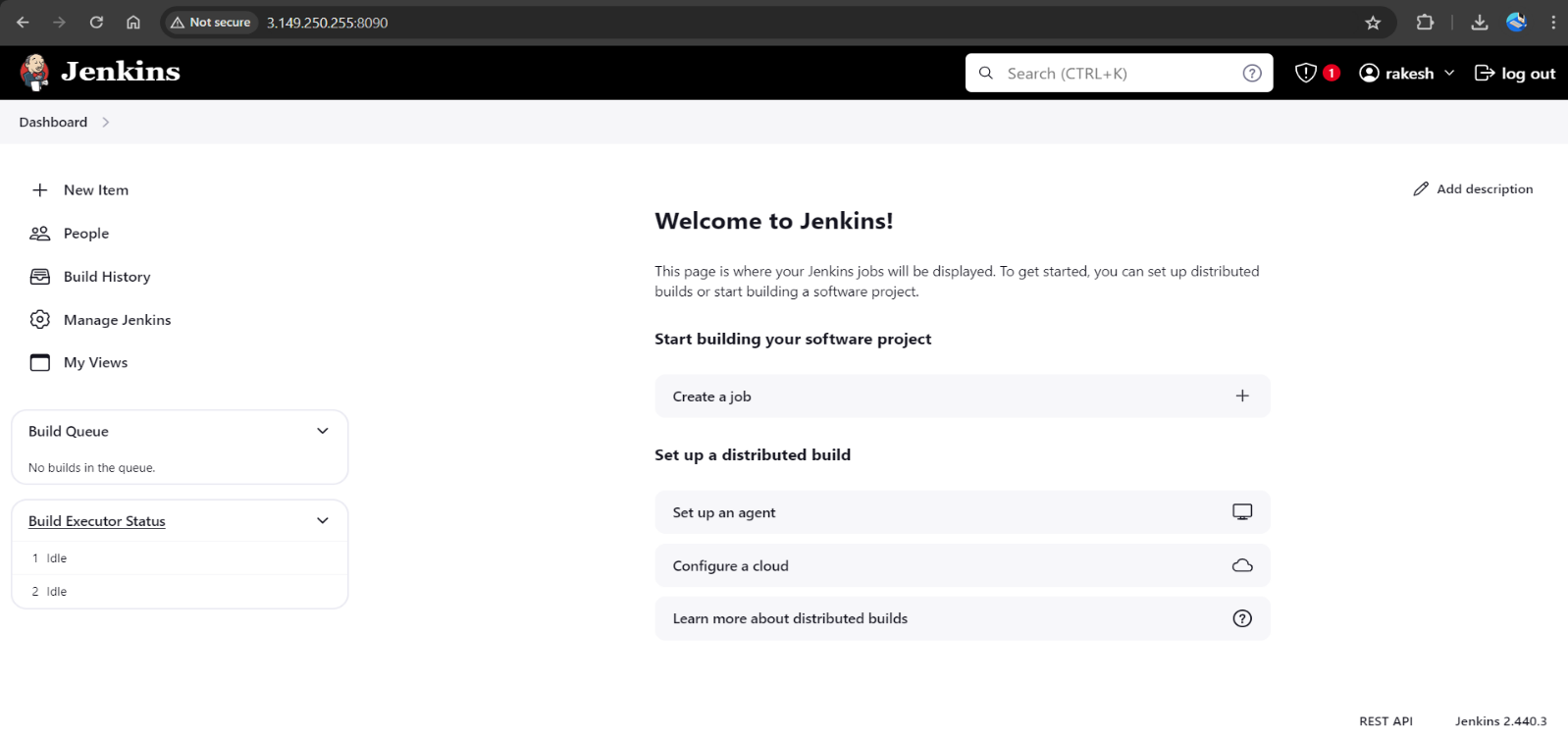




* But, for this Application case, we are running Jenkins on port 8090 using below commands.





* Now, grab your Public IP Address:8090, Unlock Jenkins using an administrative password and install the suggested plugins. Jenkins will now get installed and install all the libraries.

# Install Docker

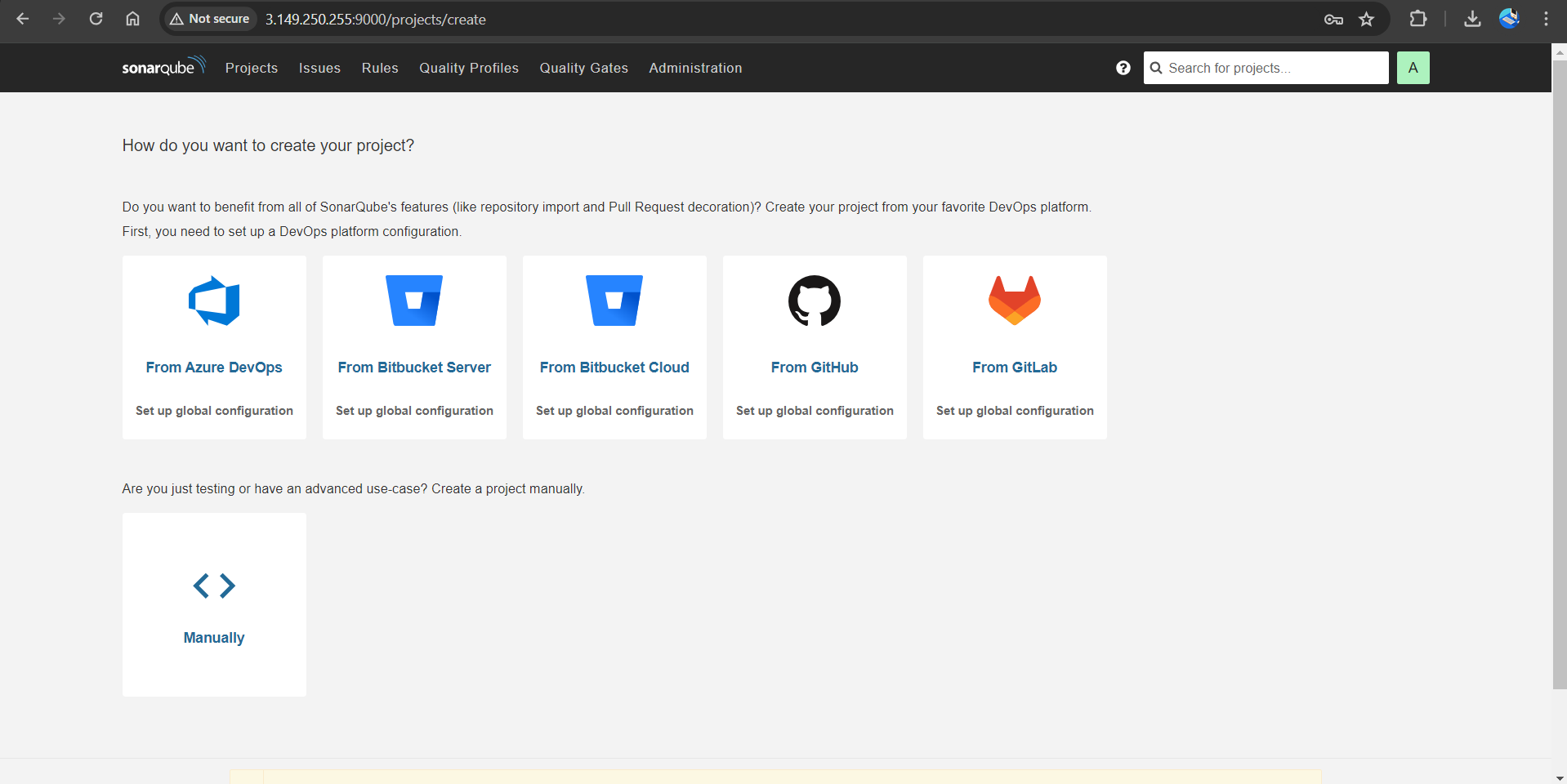
# Execute below commands in terminal.

# After the docker installation, we create a SonarQube container (Remember added 9000 ports in the security group).

# Now, run the command < docker run -d --name sonar -p 9000:9000 sonarqube:lts-community>.

# Now our SonarQube is up and running.

# Update New password, this is Sonar Dashboard.



# Install Trivy

# Create trivy.sh file with the commands.

# sudo apt-get install wget apt-transport-https gnupg lsb-release -y wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb\_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list sudo apt-get update sudo apt-get install trivy -y

# And give permissions to execute the .sh file and execute .sh with commands <vi trivy.sh>, <sudo chmod 777 trivy.sh>, <./trivy.sh>.

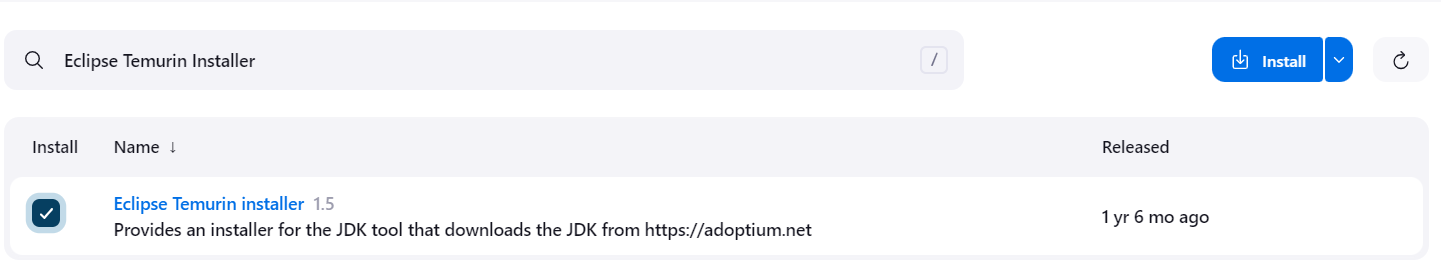
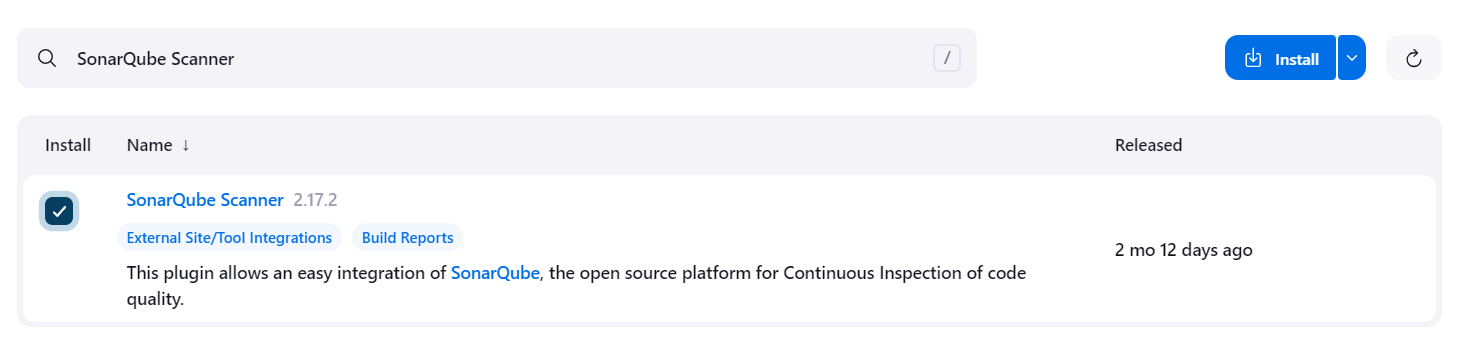
# 

# Next, we will log in to Jenkins and start to configure our Pipeline in Jenkins

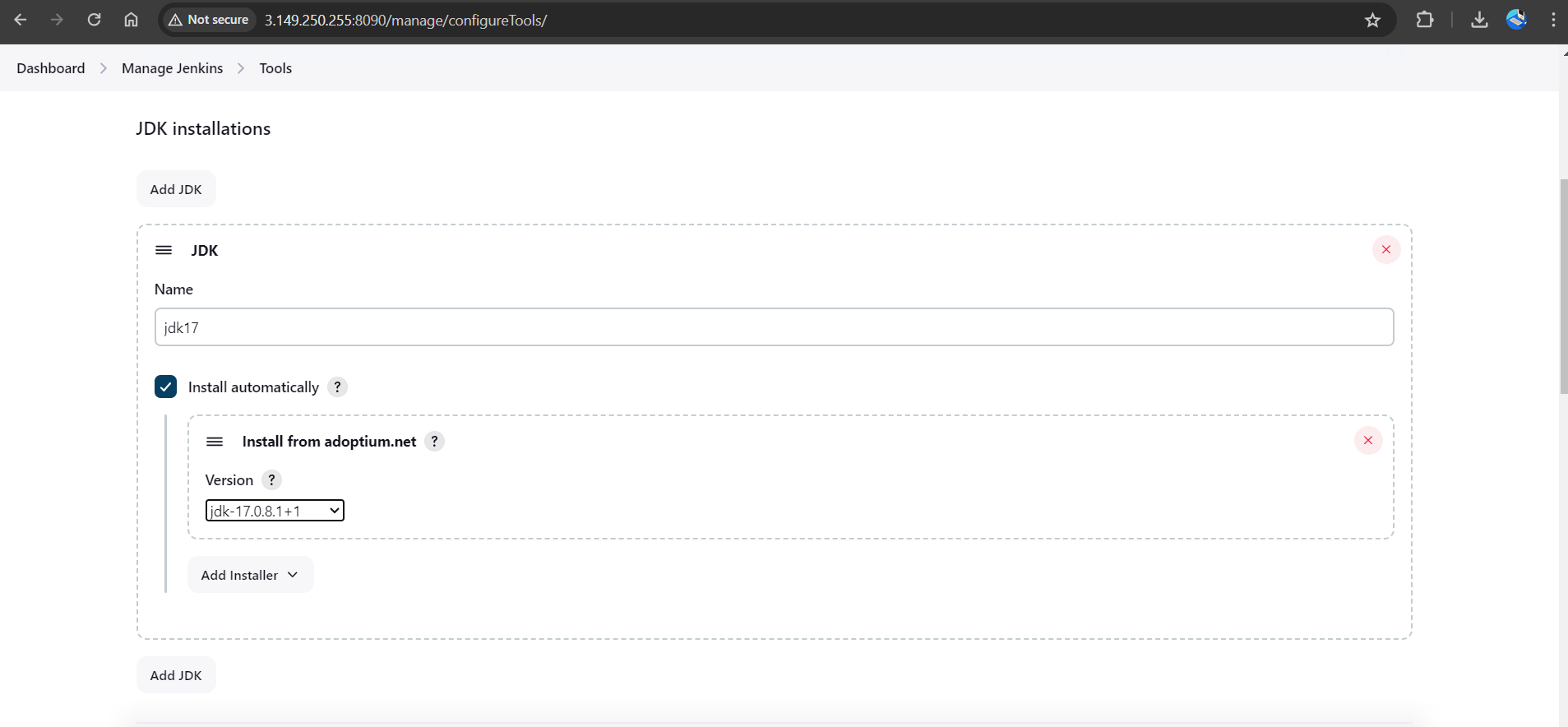
# Install Plugins like JDK, SonarQube Scanner, Maven, OWASP Dependency Check.

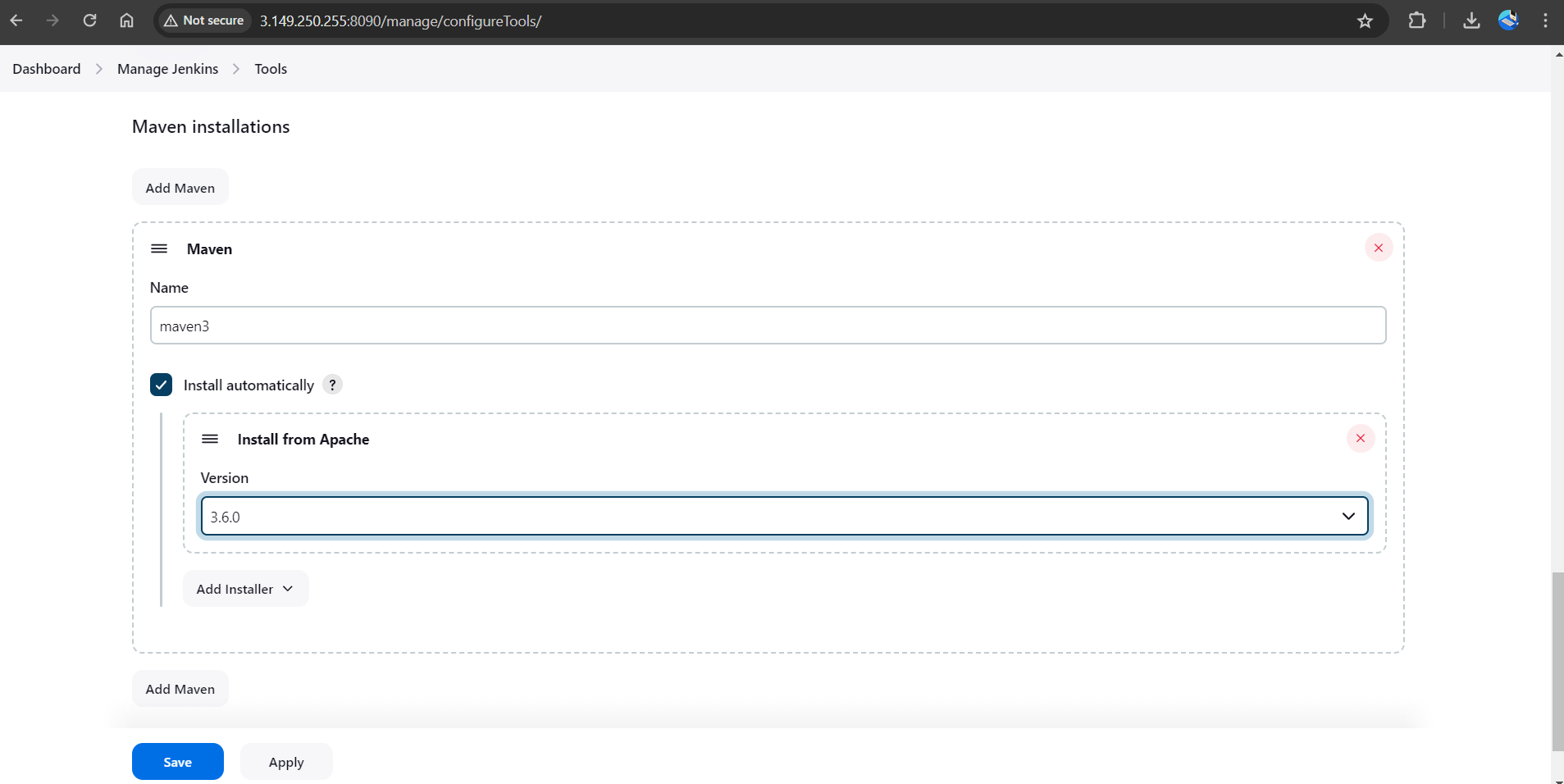
* Goto Manage Jenkins →Plugins → Available Plugins →

**Install below plugins**

* Eclipse Temurin Installer (Install without restart)
* SonarQube Scanner (Install without restart)

# Configure Java and Maven in Global Tool Configuration

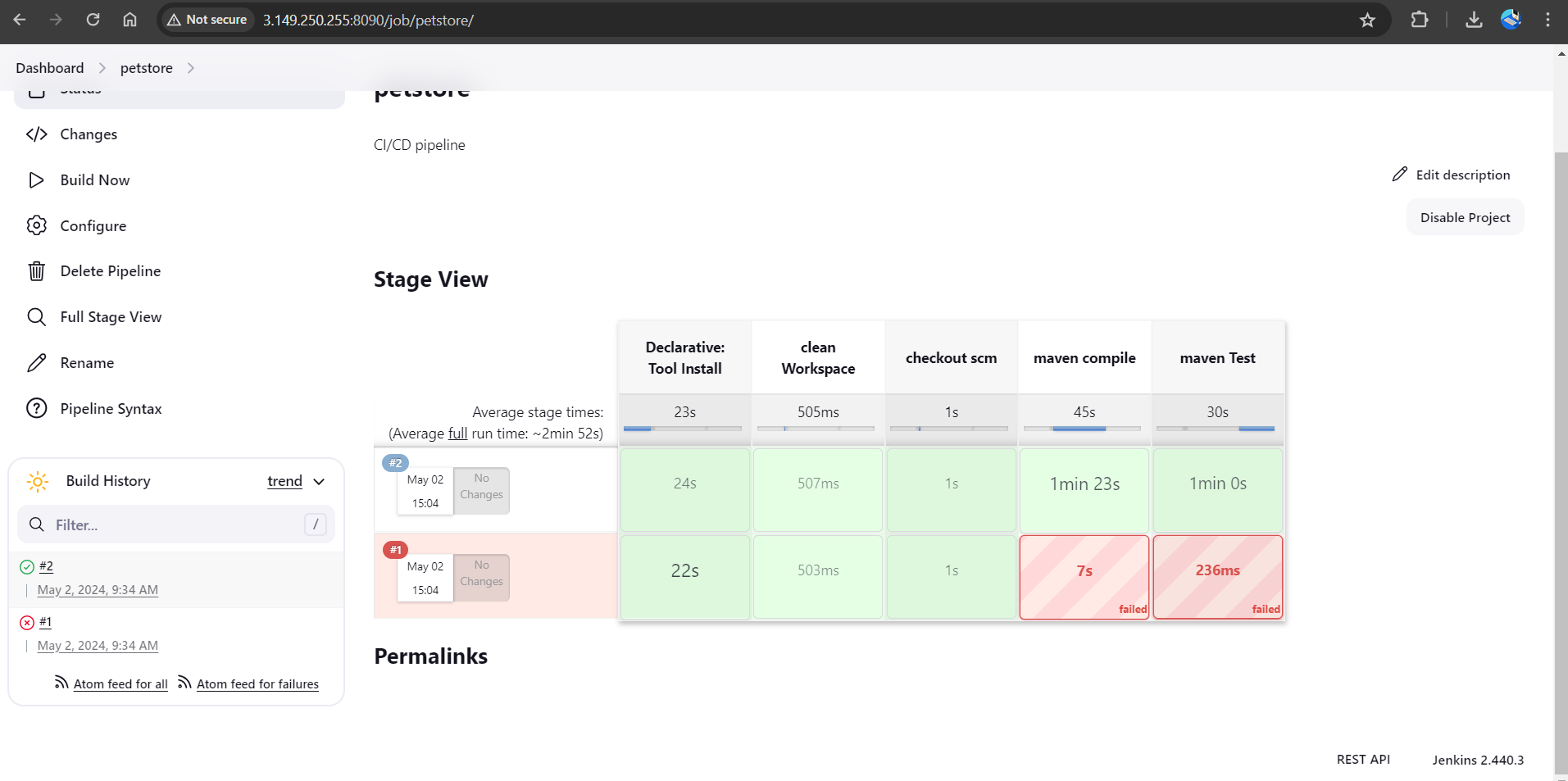
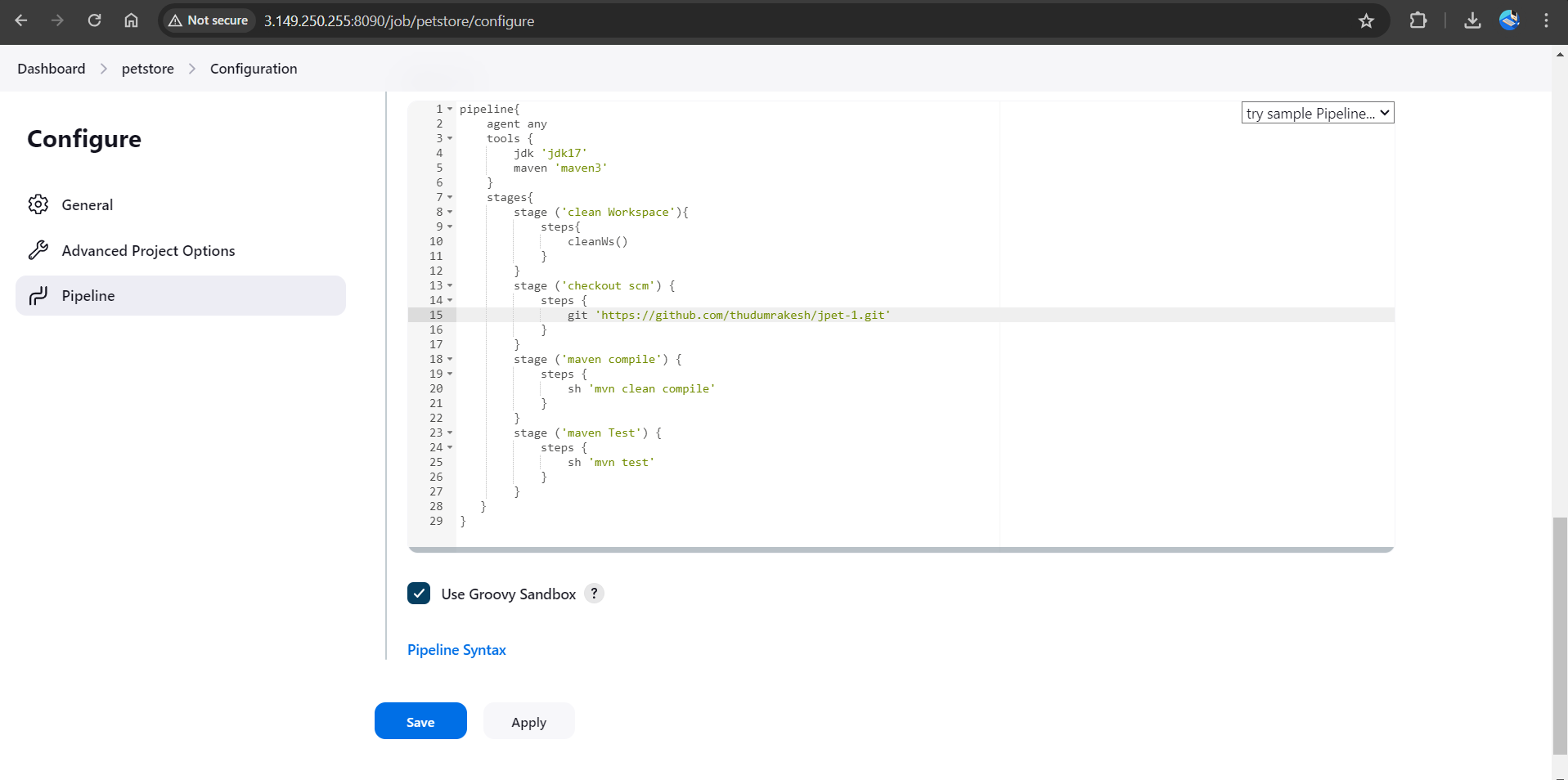
* Goto Manage Jenkins → Tools → Install JDK(17) and Maven3(3.6.0) → Click on Apply and Save.



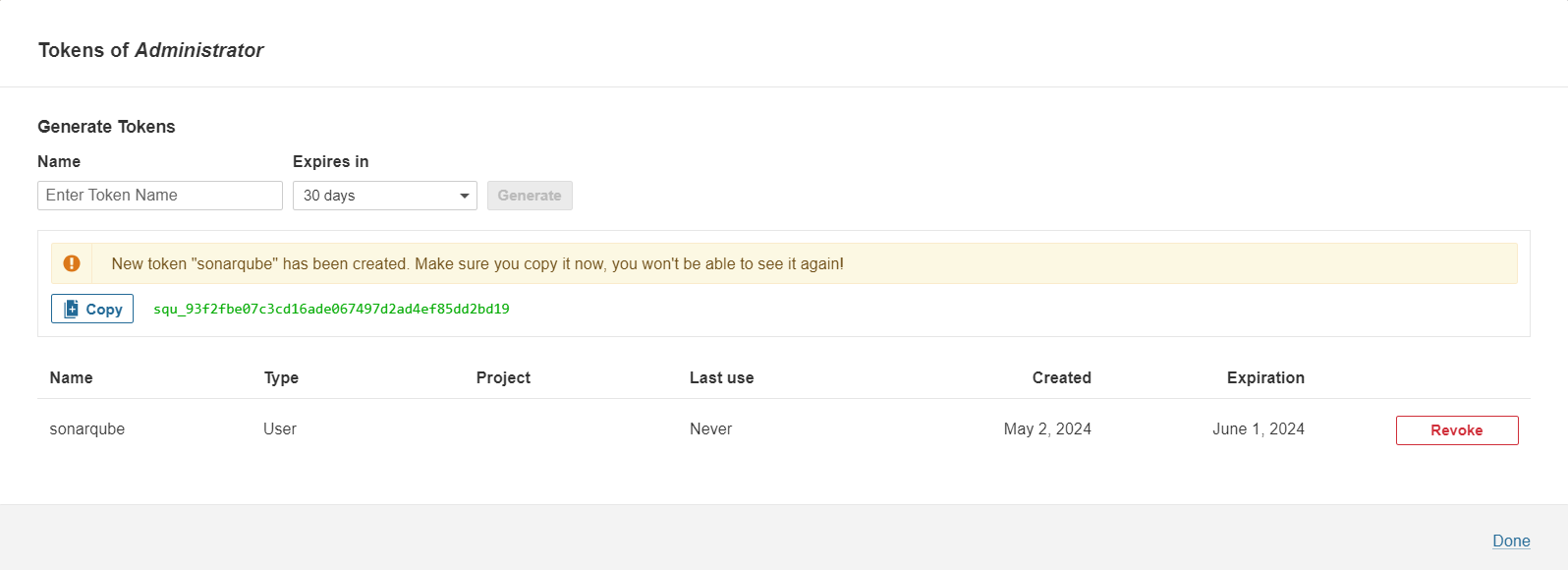
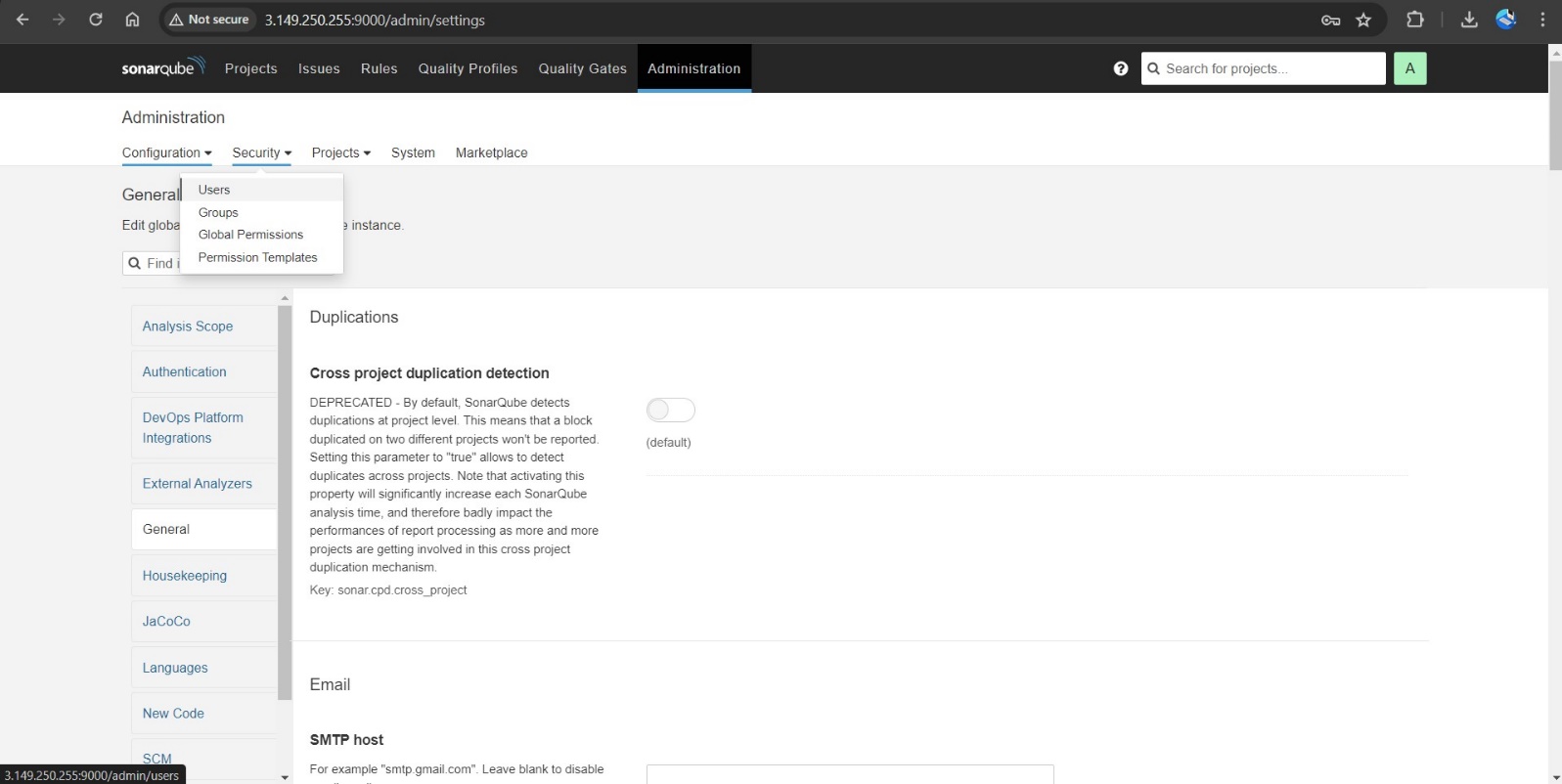
# Create a Job

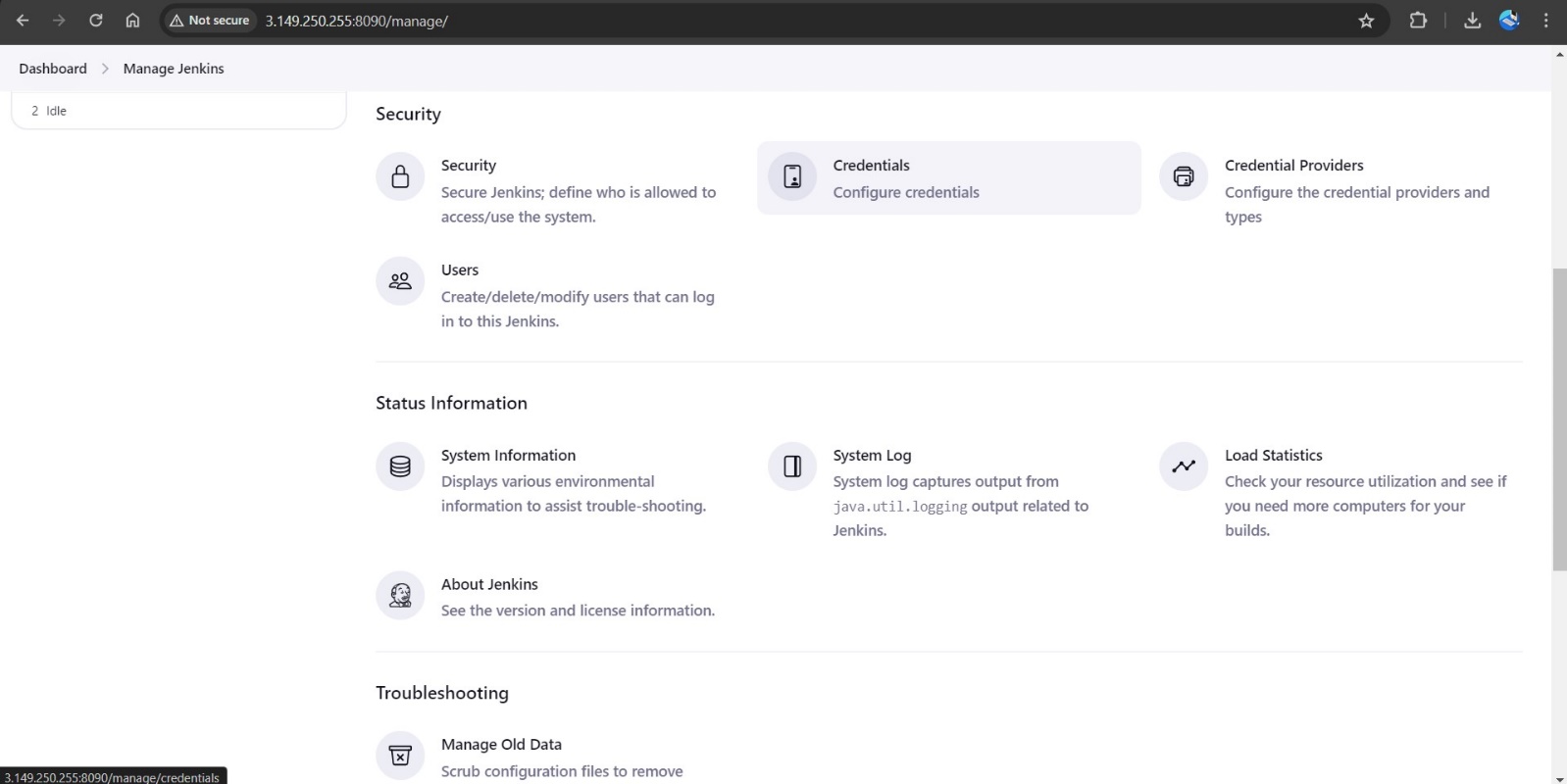
# Label it as petstore, click on Pipeline and OK.

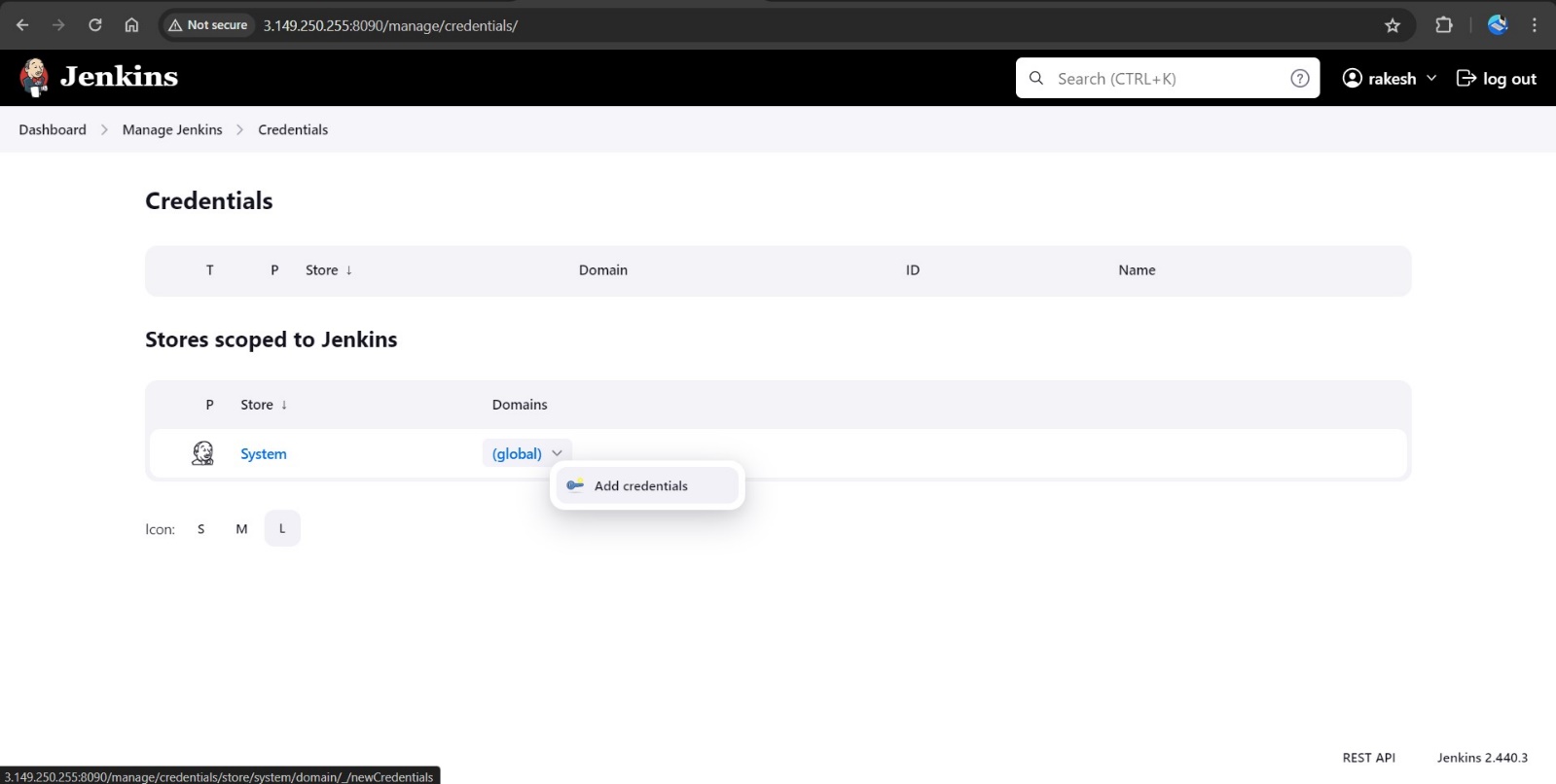
# Enter this in Pipeline Script, The stage view would look like below.

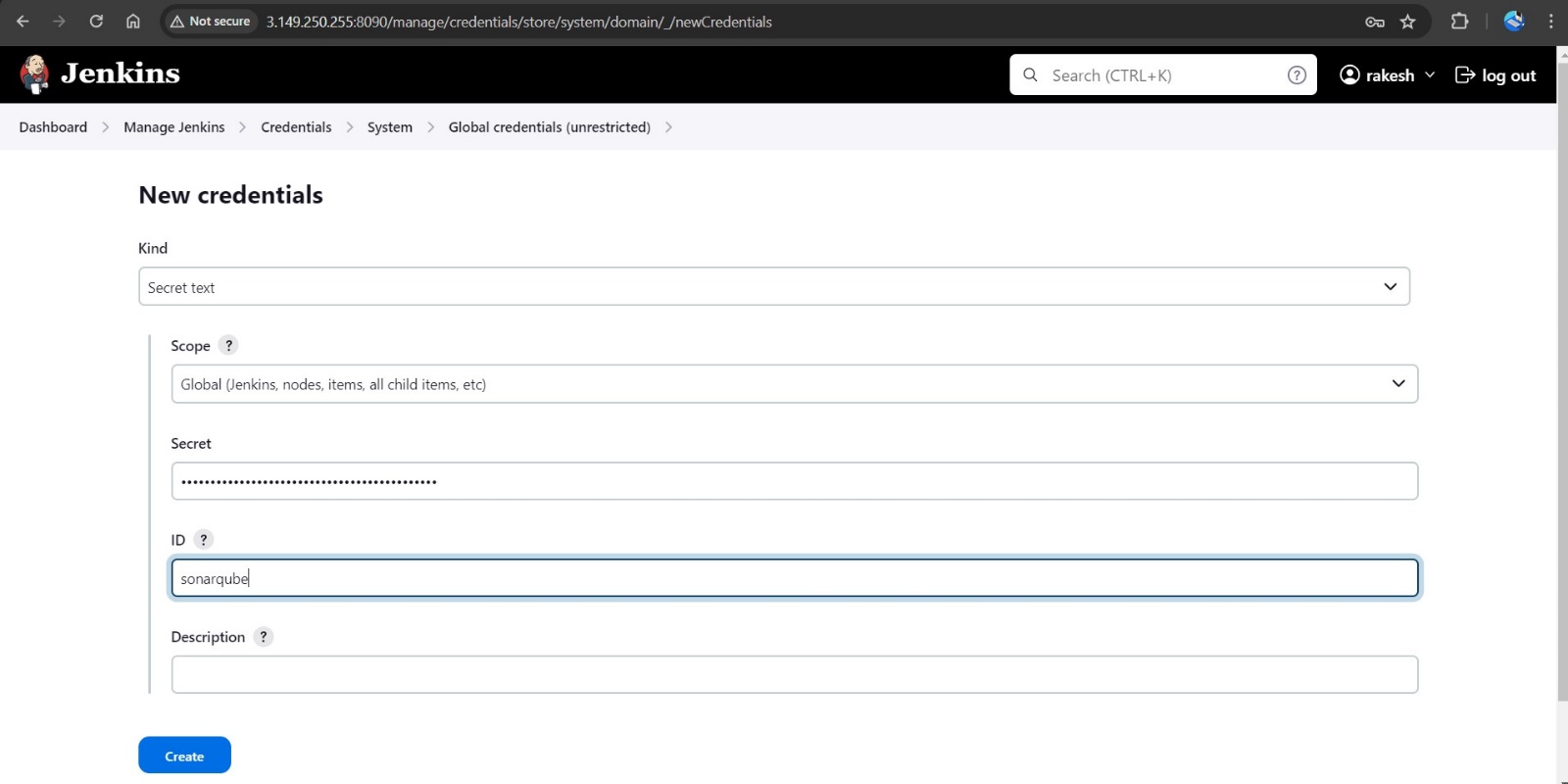


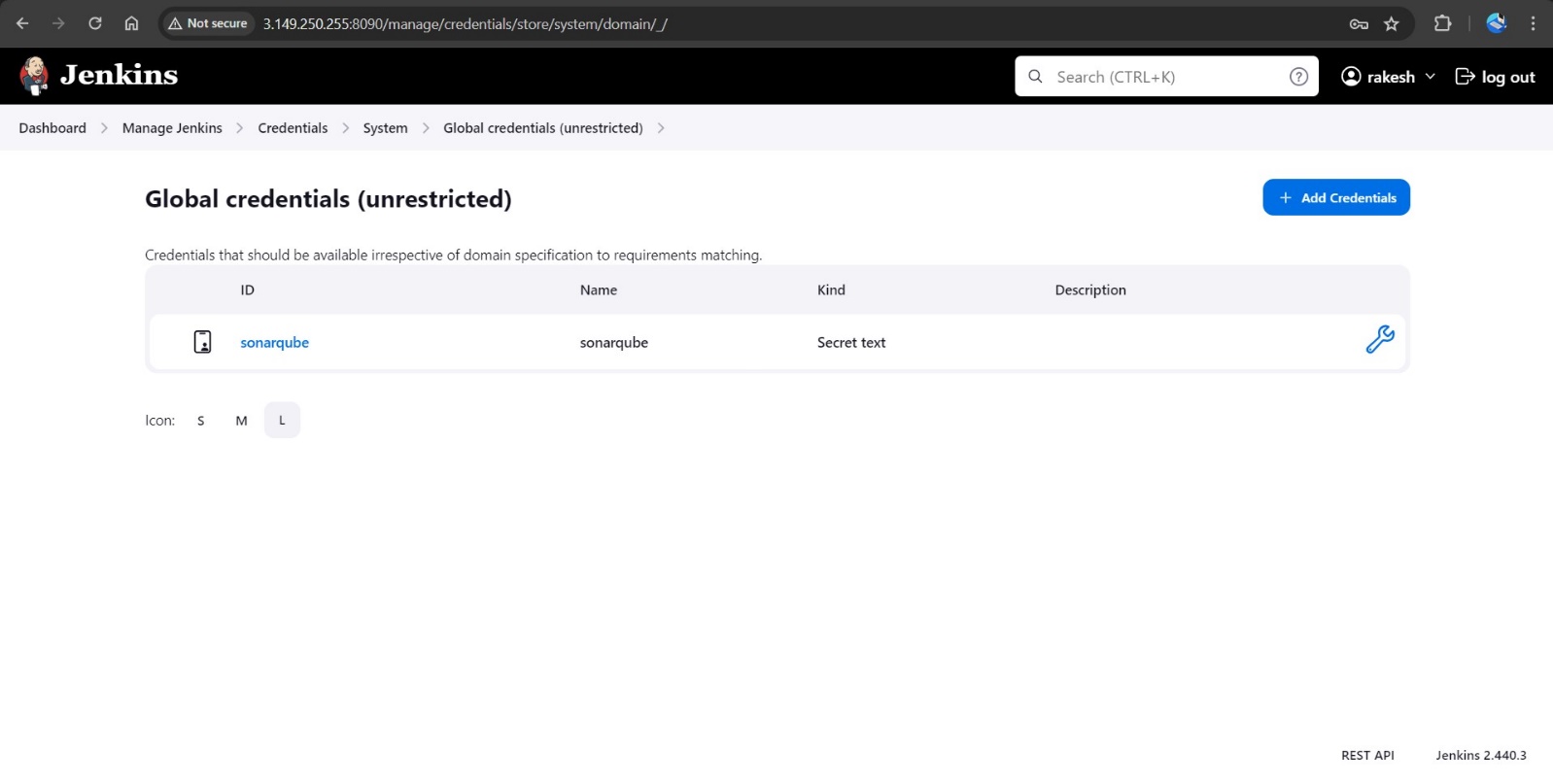
**Configure Sonar Server in Manage Jenkins**

* Grab the Public IP Address of your EC2 Instance, SonarQube works on Port 9000, so <Public IP>:9000. Go to your SonarQube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token. And copy Token.
* Go to Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this

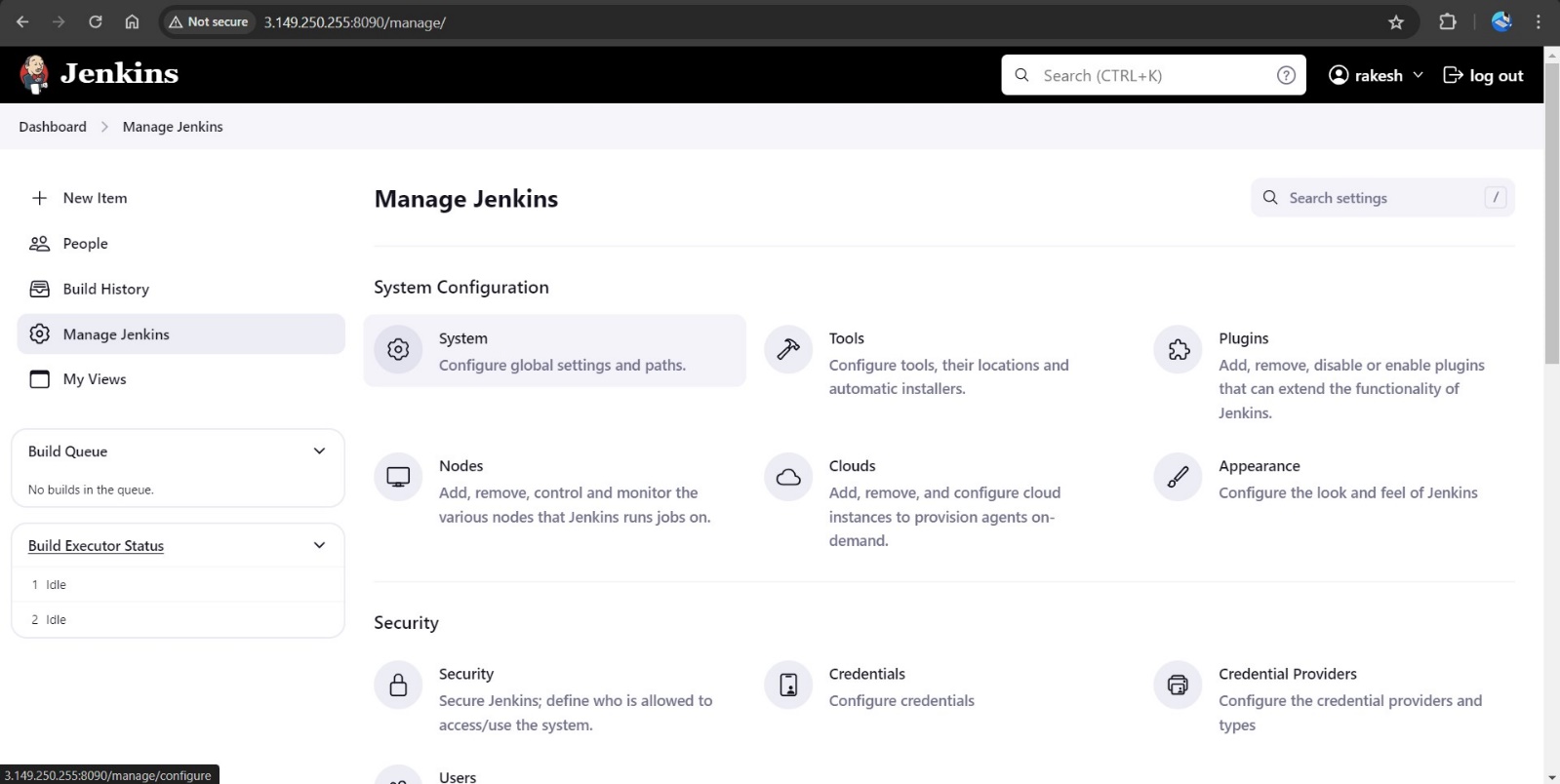


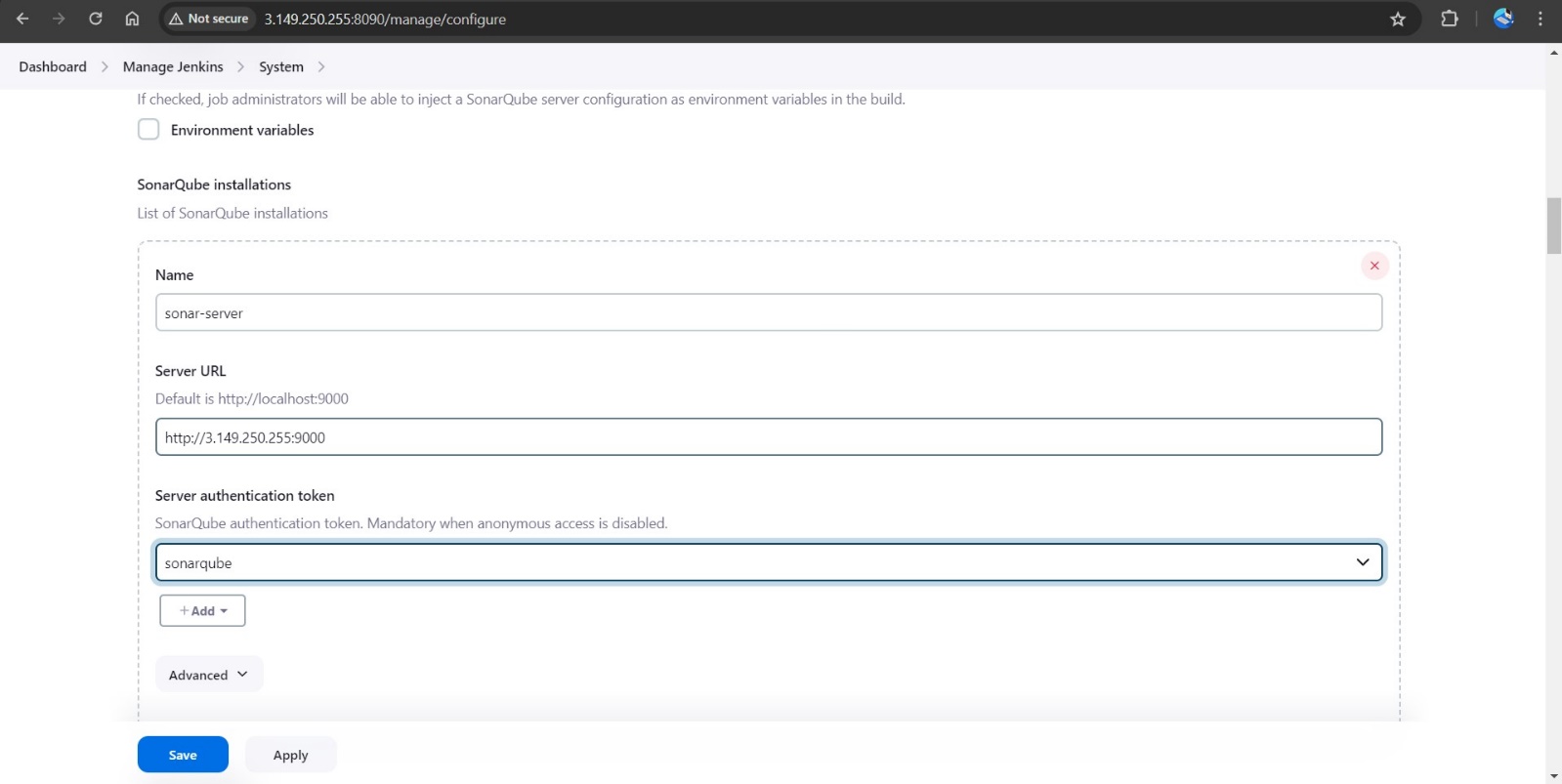


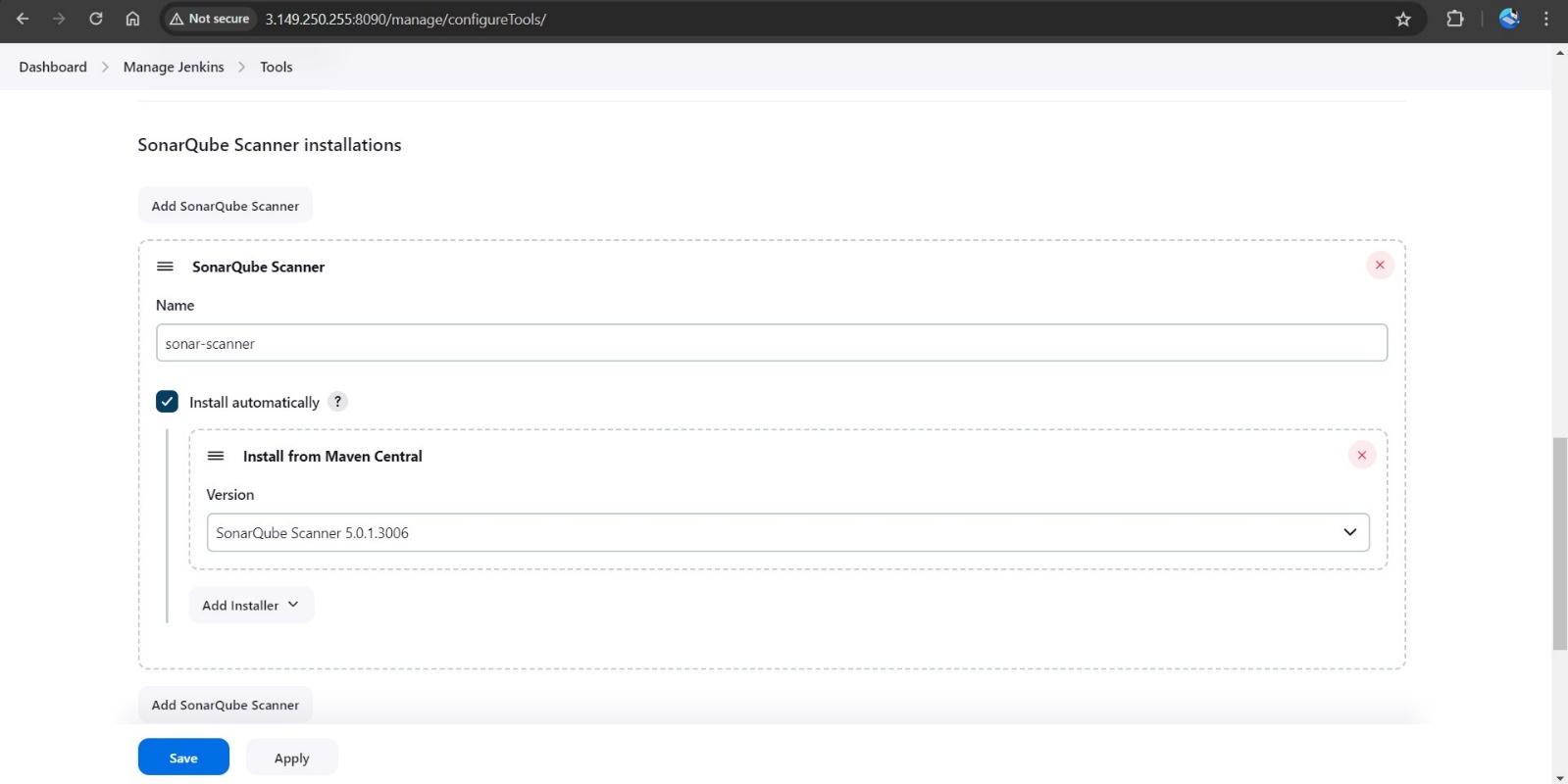




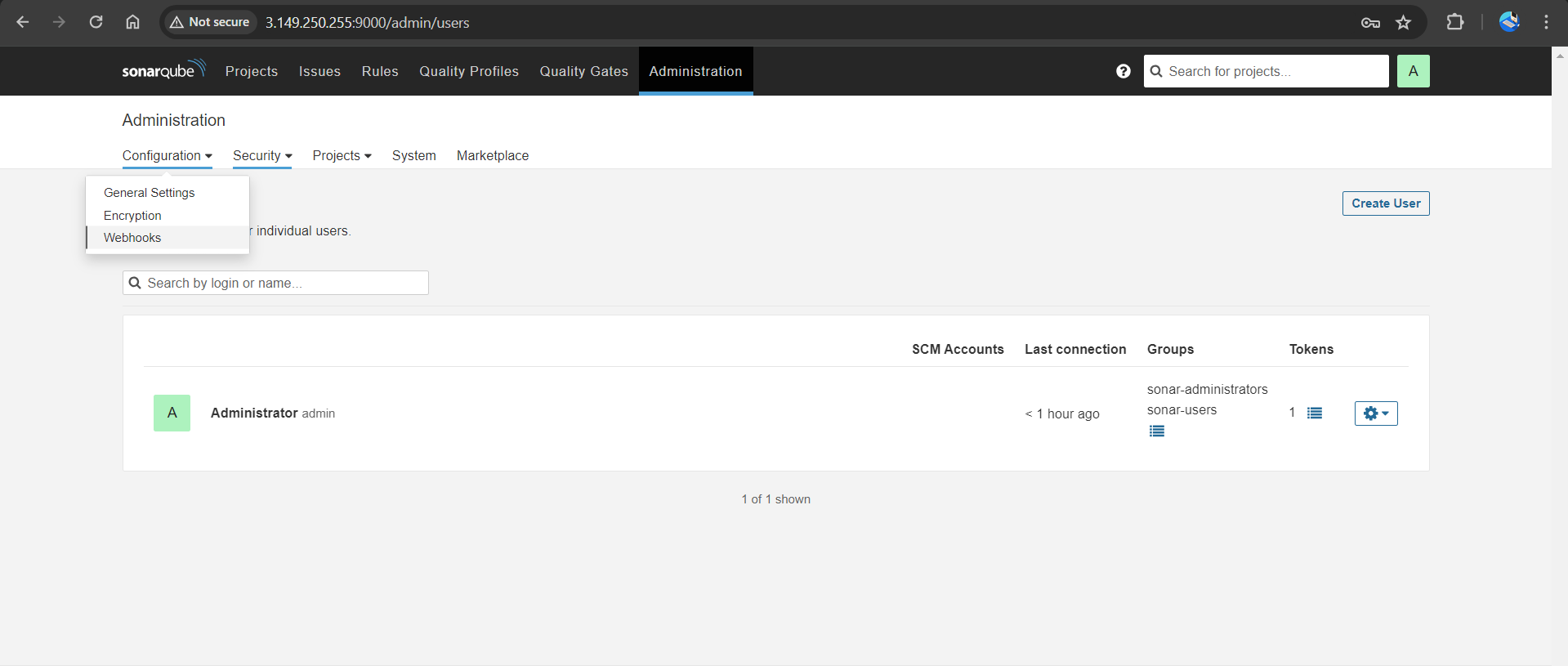
* Now, go to Dashboard → Manage Jenkins → System and Add like the below image.

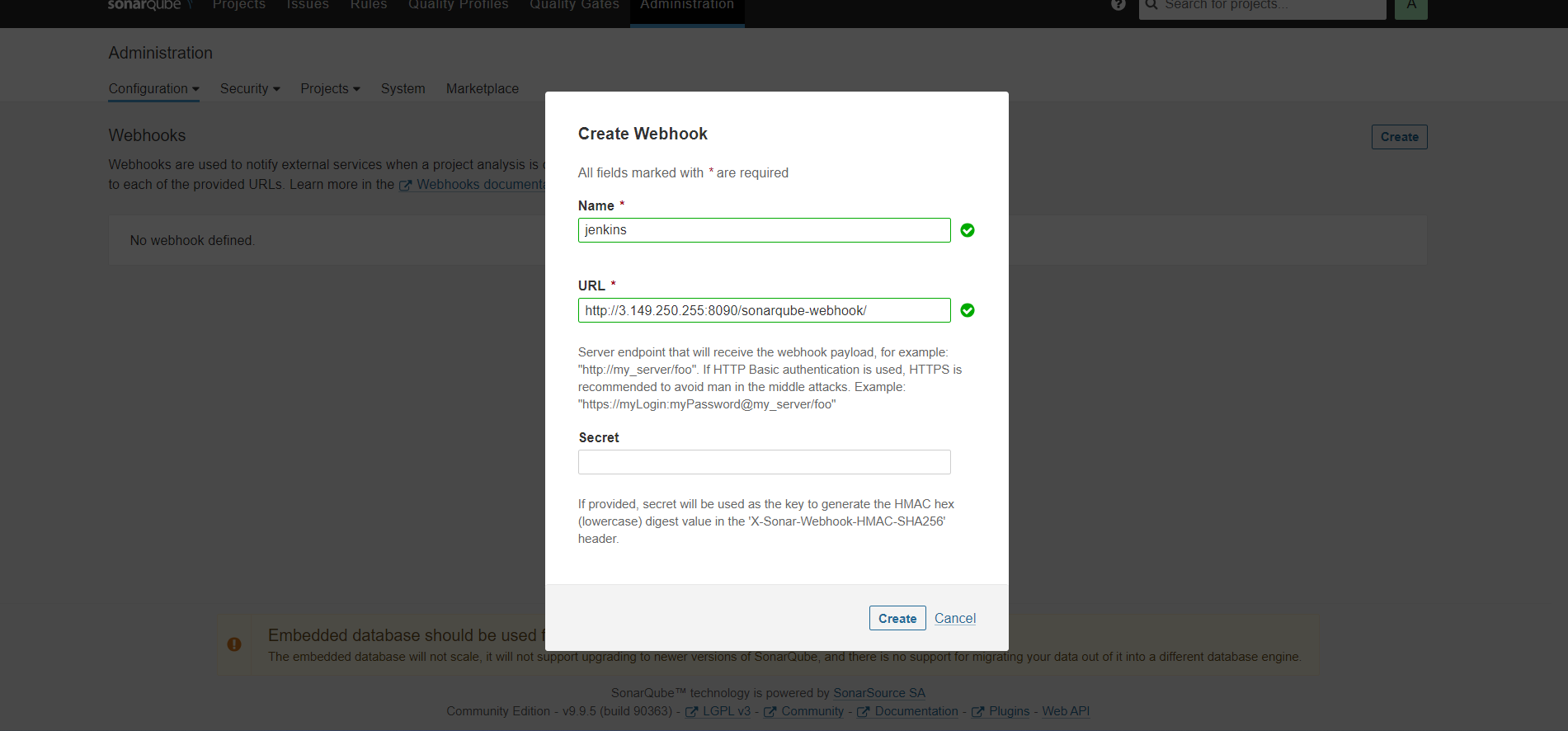






* Click on Apply and Save
* The Configure System option is used in Jenkins to configure different server
* Global Tool Configuration is used to configure different tools that we install using Plugins
* We will install a sonar scanner in the tools.
* In the SonarQube Dashboard add a quality gate also
* Administration → Configuration →Webhooks





Let’s go to our Pipeline and add SonarQube Stage in our Pipeline Script.

pipeline {

agent any

tools {

jdk 'jdk17'

maven 'maven3'

}

environment {

SCANNER\_HOME = tool name: 'sonar-scanner', type: 'hudson.plugins.sonar.SonarRunnerInstallation'

}

stages {

stage ('Clean Workspace') {

steps {

cleanWs()

}

}

stage ('Checkout SCM') {

steps {

git branch: 'main', url: 'https://github.com/Shaik-DevOpsLucky/petstore.git'

}

}

stage ('Maven Compile') {

steps {

sh 'mvn clean compile'

}

}

stage ('Maven Test') {

steps {

sh 'mvn test'

}

}

stage ("Sonarqube Analysis") {

steps {

withSonarQubeEnv('sonar-server') {

script {

def scannerHome = tool 'sonar-scanner'

sh "${scannerHome}/bin/sonar-scanner -Dsonar.projectName=Petshop -Dsonar.java.binaries=. -Dsonar.projectKey=Petshop"

}

}

}

post {

always {

script {

def qg = waitForQualityGate()

if (qg.status != 'OK') {

error "Pipeline aborted due to quality gate failure: ${qg.status}"

}

}

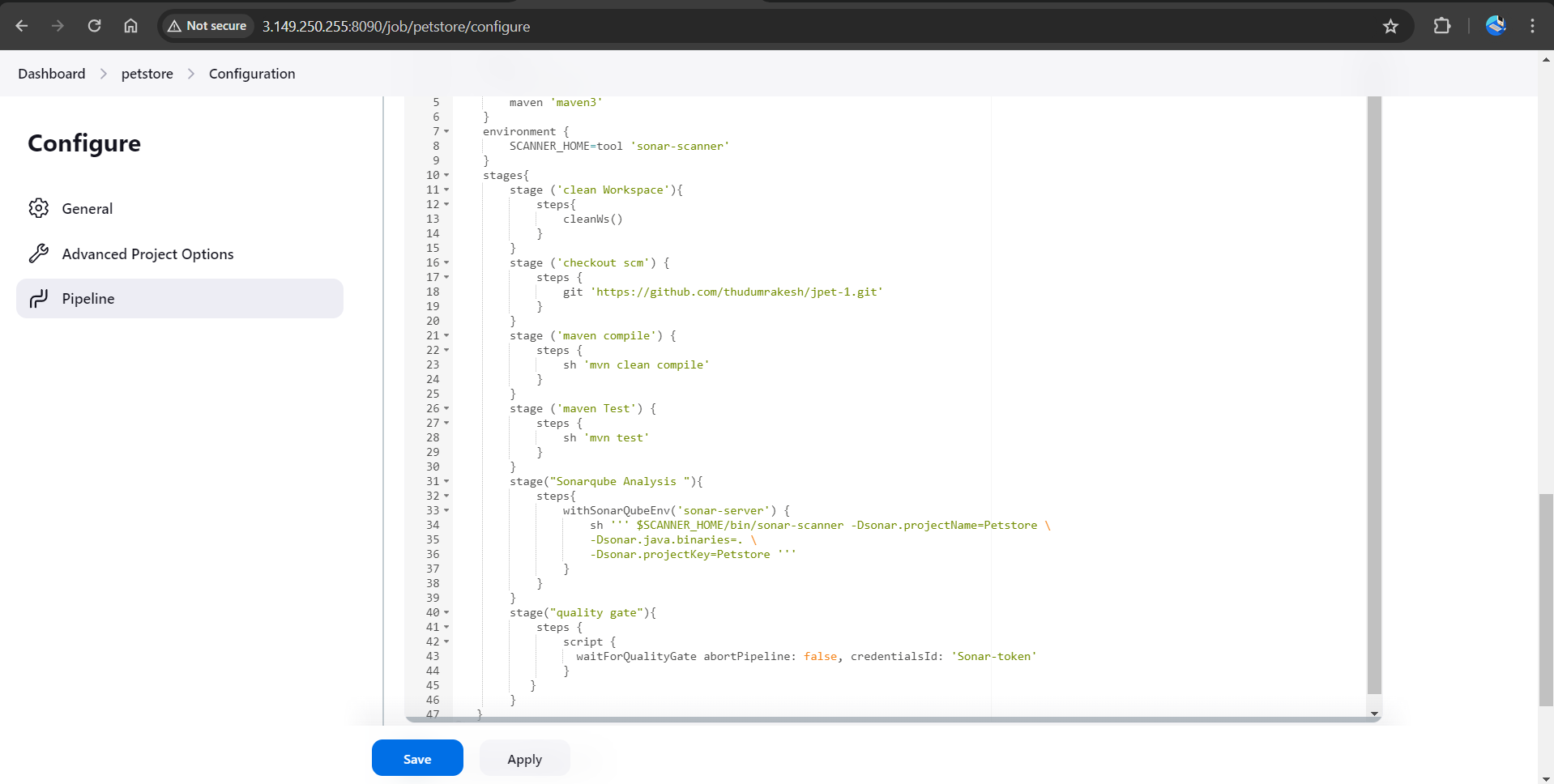
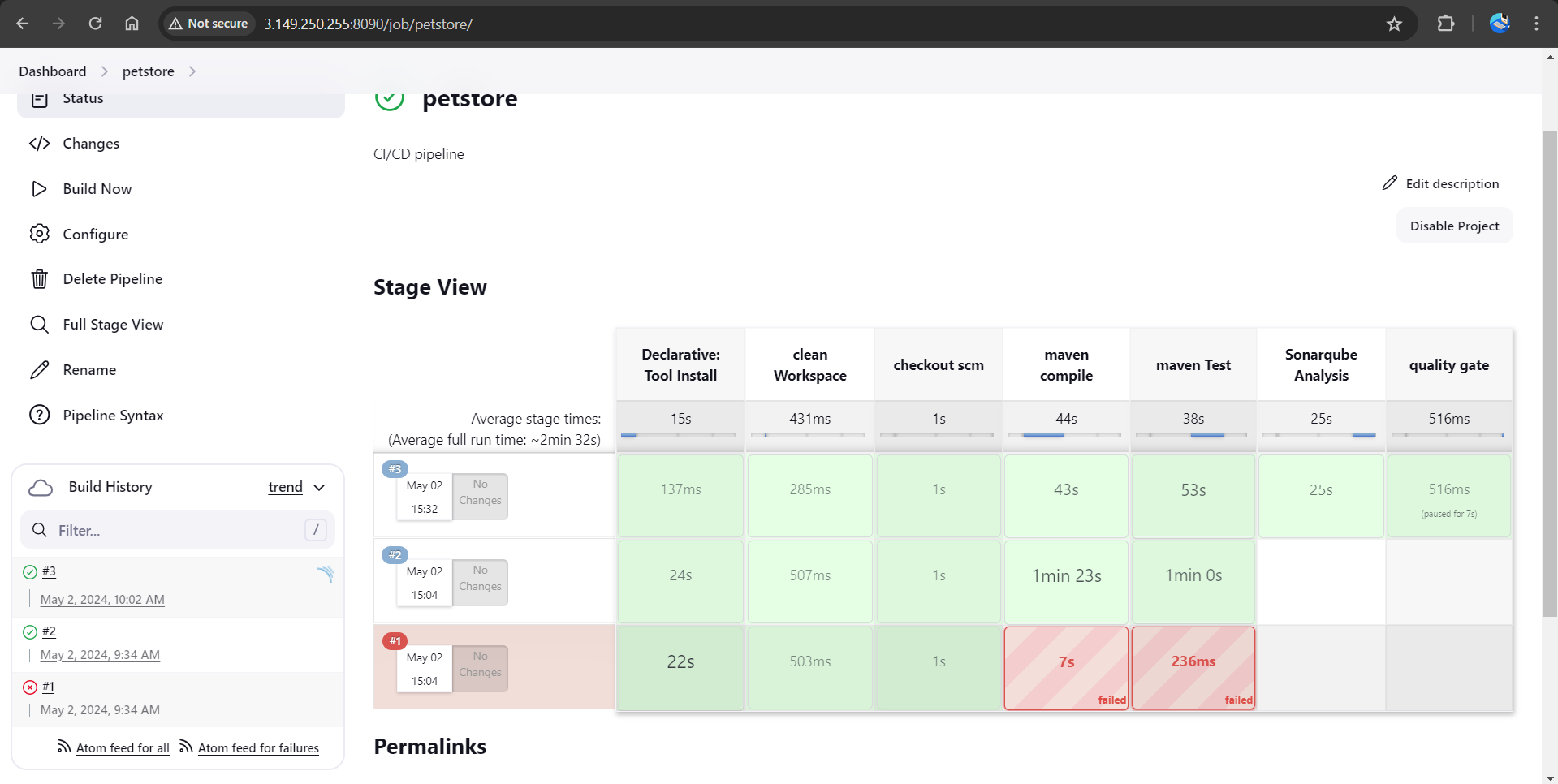
}

}

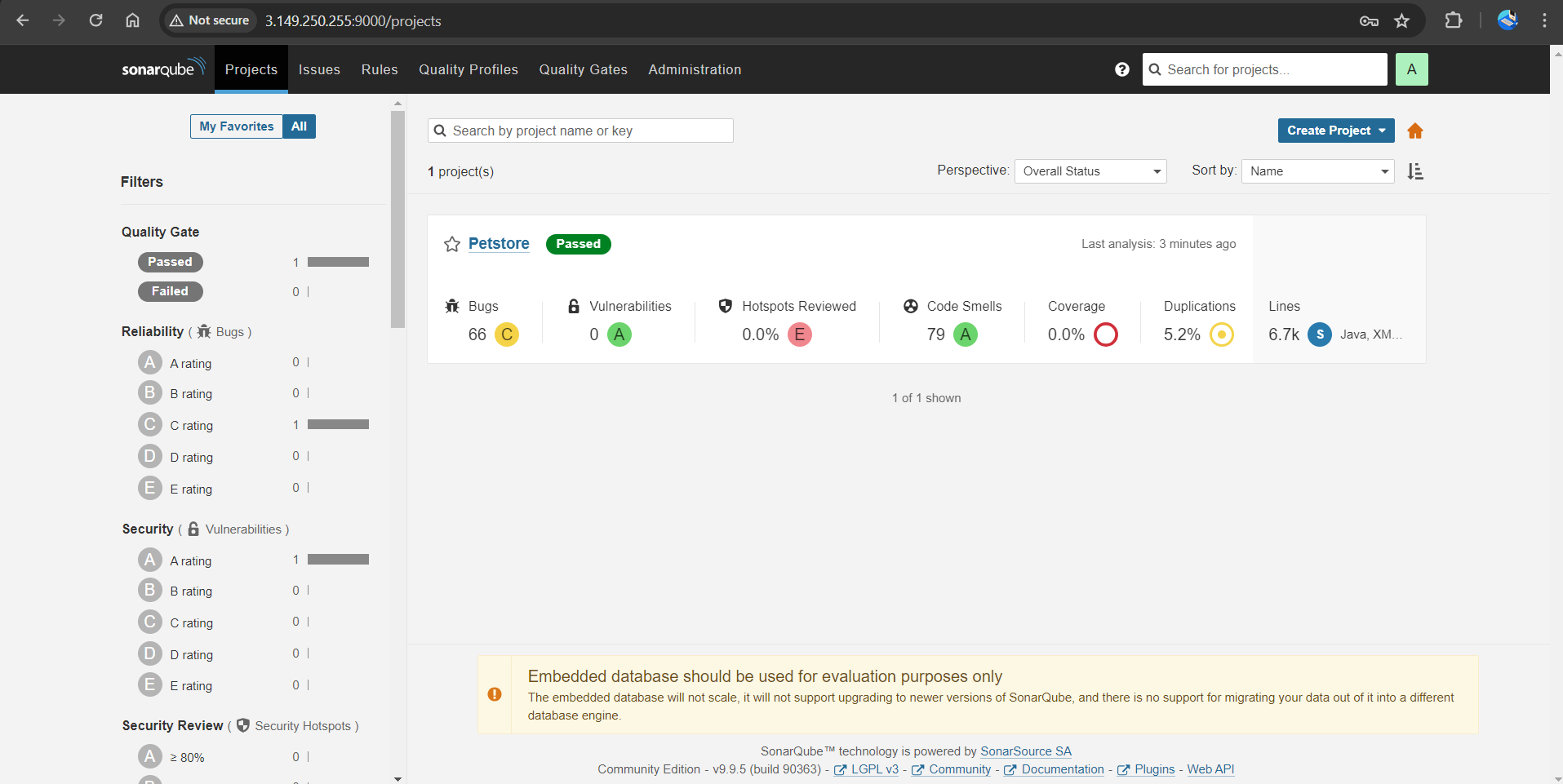
}

}

}

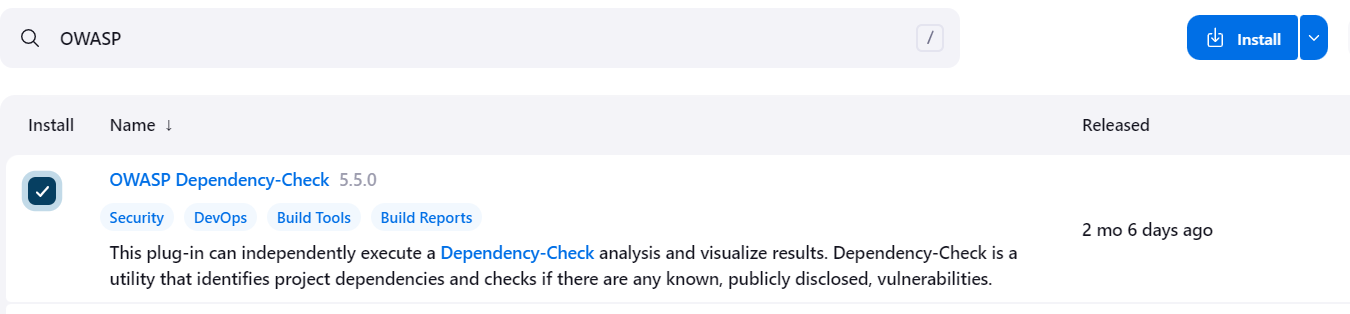


* Click on Build now, you will see the stage view like this
* To see the report, you can go to SonarQube Server and go to Projects.

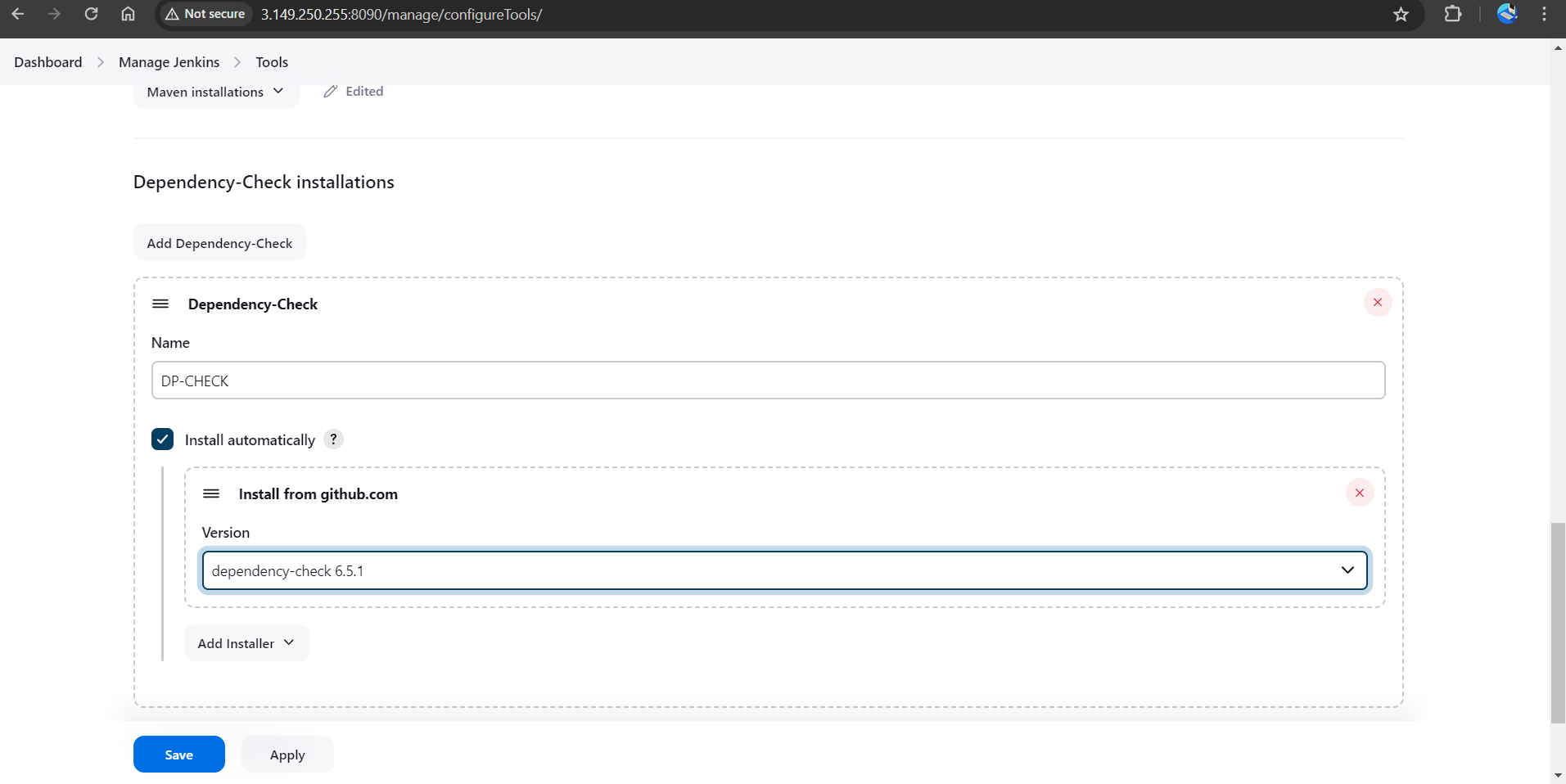


# Install OWASP Dependency Check Plugins

# Go to Jenkins-Dashboard → Manage Jenkins → Plugins → OWASP Dependency-Check. Click on it and install it without restart.



* First, we configured the Plugin and next, we had to configure the Tool
* Goto Dashboard → Manage Jenkins → Tools →
* Click on Apply and Save here.
* Now go configure → Pipeline and add this stage to your pipeline.

stage ('Build war file') {

steps {

sh 'mvn clean install -DskipTests=true'

}

}

stage("OWASP Dependency Check") {

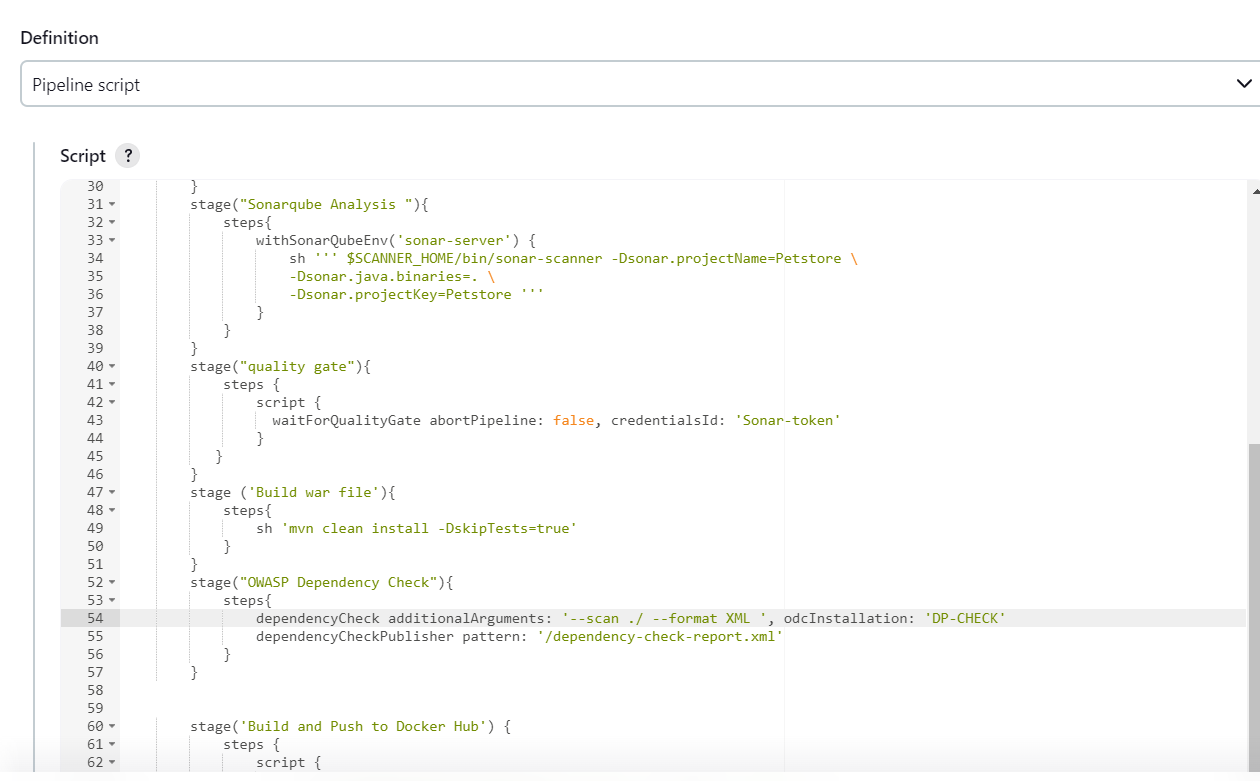
steps {

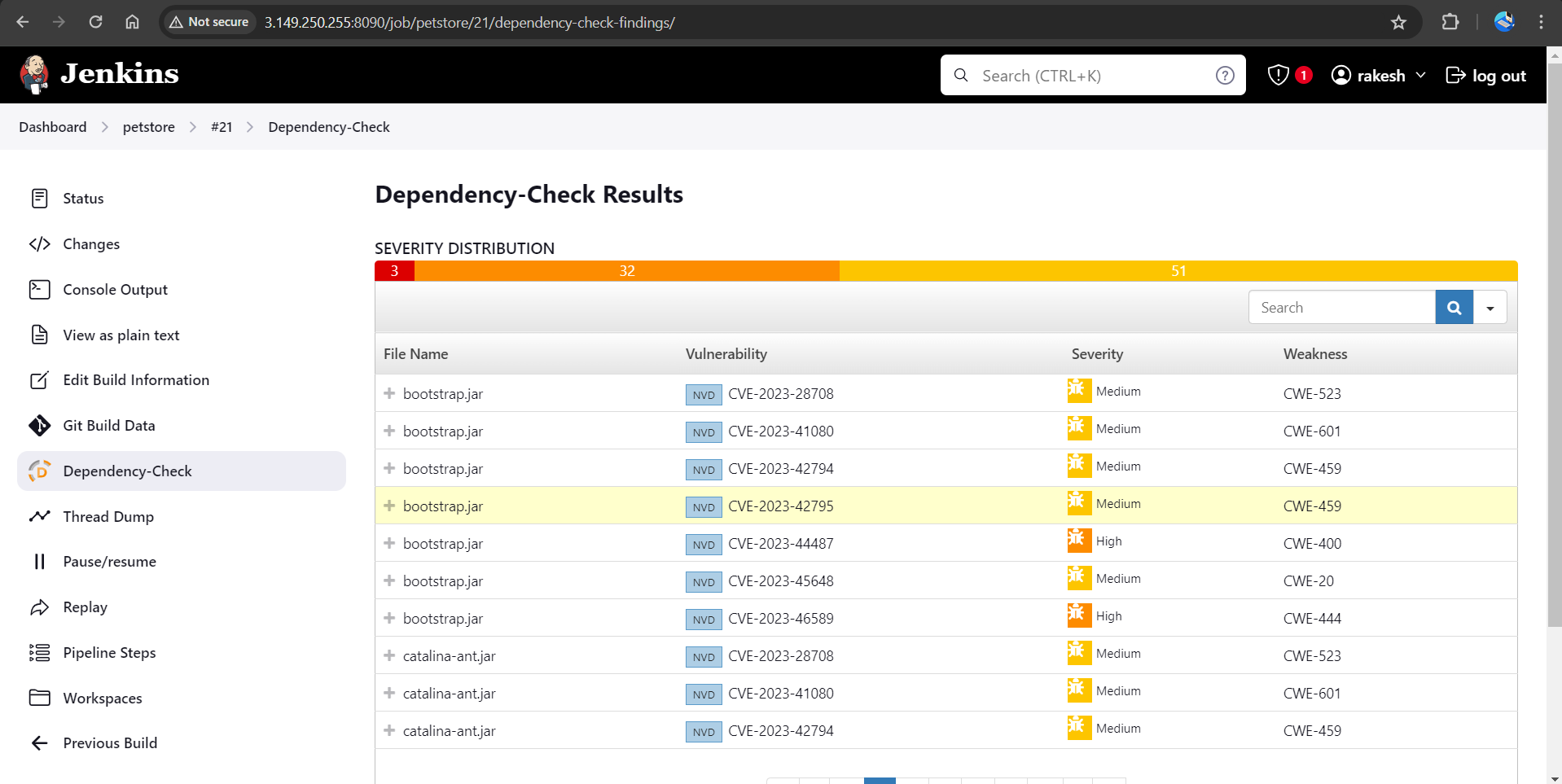
dependencyCheck additionalArguments: '--scan ./ --format XML ', odcInstallation: 'DP-Check'

dependencyCheckPublisher pattern: '\*\*/dependency-check-report.xml'

}

}

 }

* Click on Build now, you will see the stage view like this.

# Docker plugin and credential Setup

* We need to install the Docker tool in our system, Goto Dashboard → Manage Plugins → Available plugins → Search for Docker and install these plugins

Docker

Docker Commons

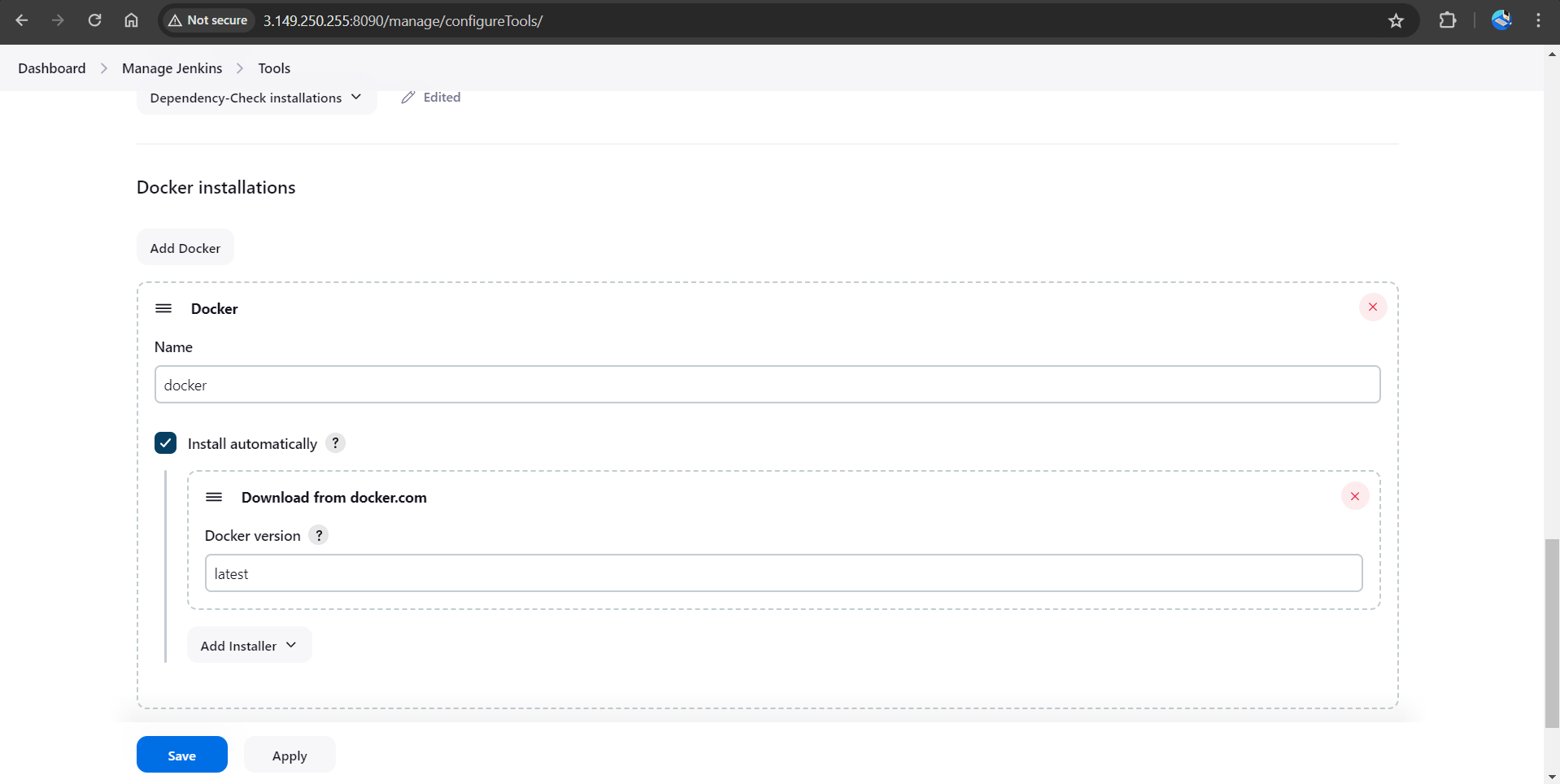
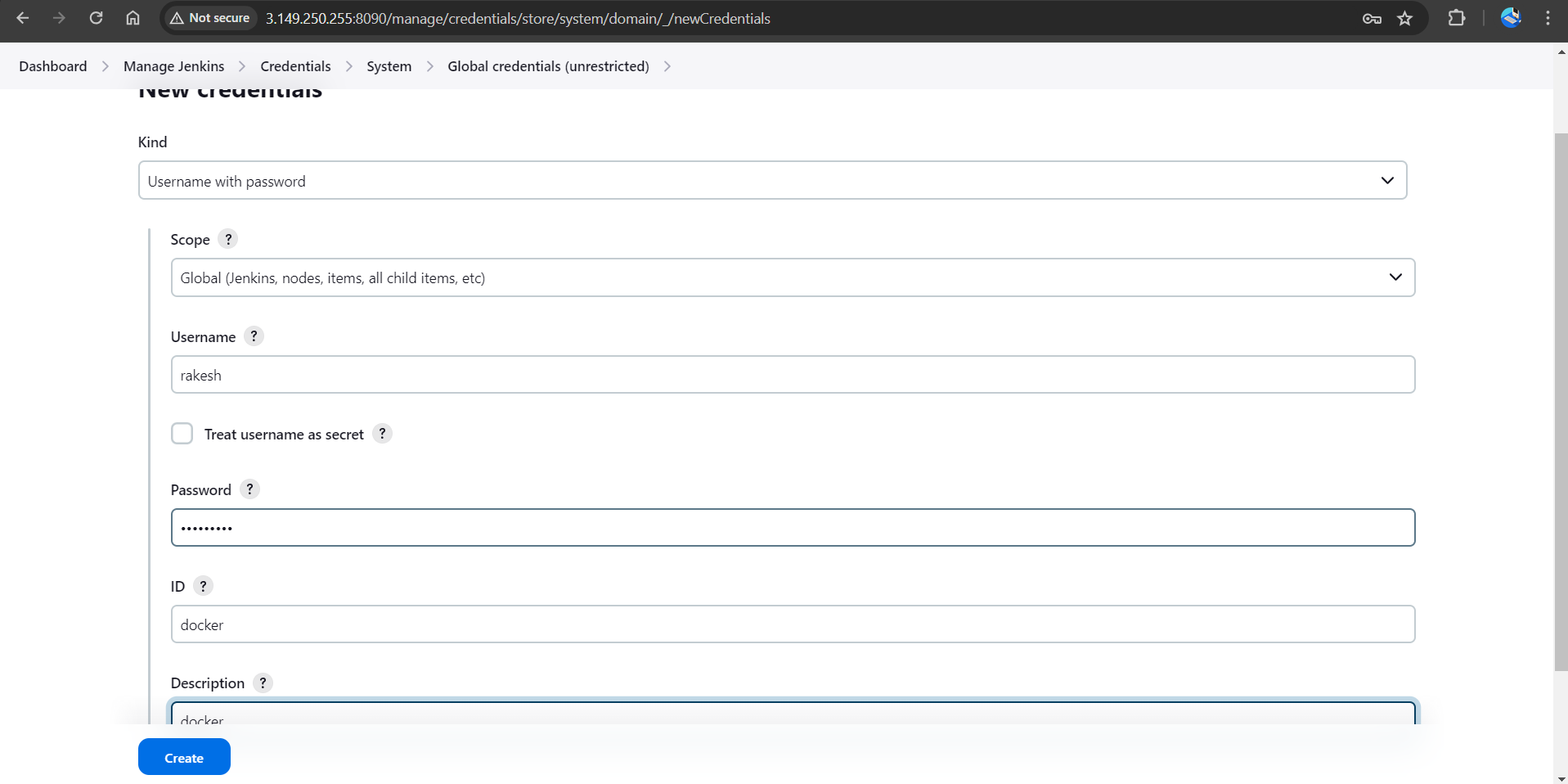
Docker Pipeline

Docker API

docker-build-step

* And click on install without restart

# 

* Now, goto Dashboard → Manage Jenkins → Tools →
* Add Docker-Hub Username and Password under Global Credentials.

 **Adding Ansible Repository in Ubuntu**

* Now we are going to run the below commands on the Ansible server
* Update your system packages

<sudo apt-get update>, <sudo apt install software-properties-common>,

* Add the ansible repository via PPA

<sudo add-apt-repository --yes --update ppa:ansible/ansible>, <sudo apt install python3>.

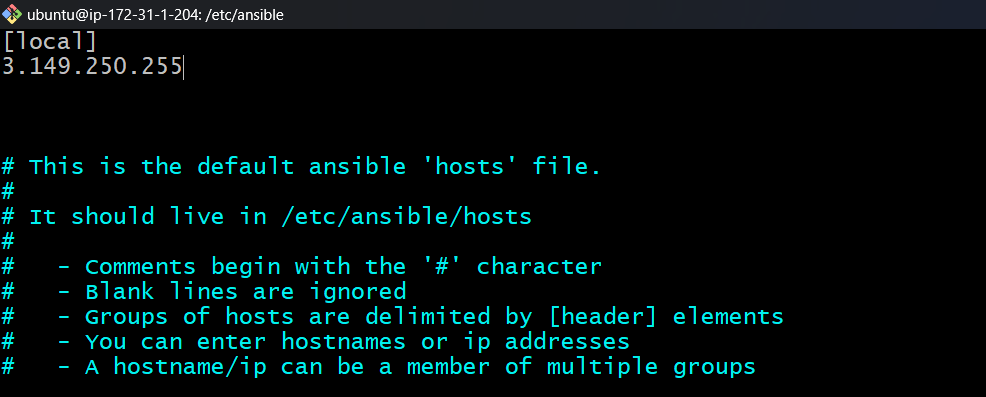
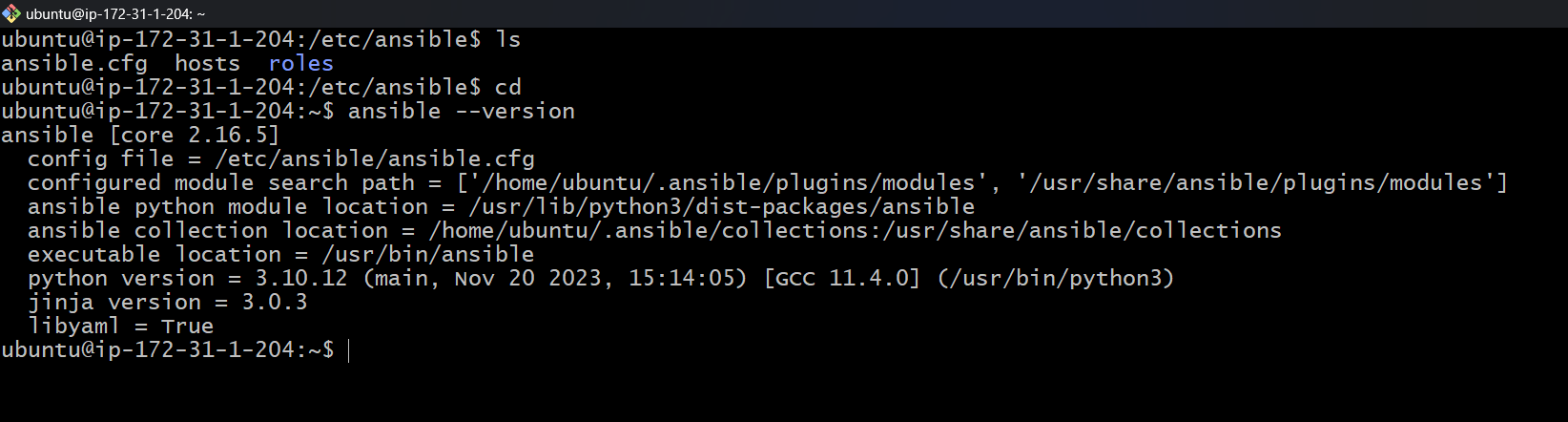
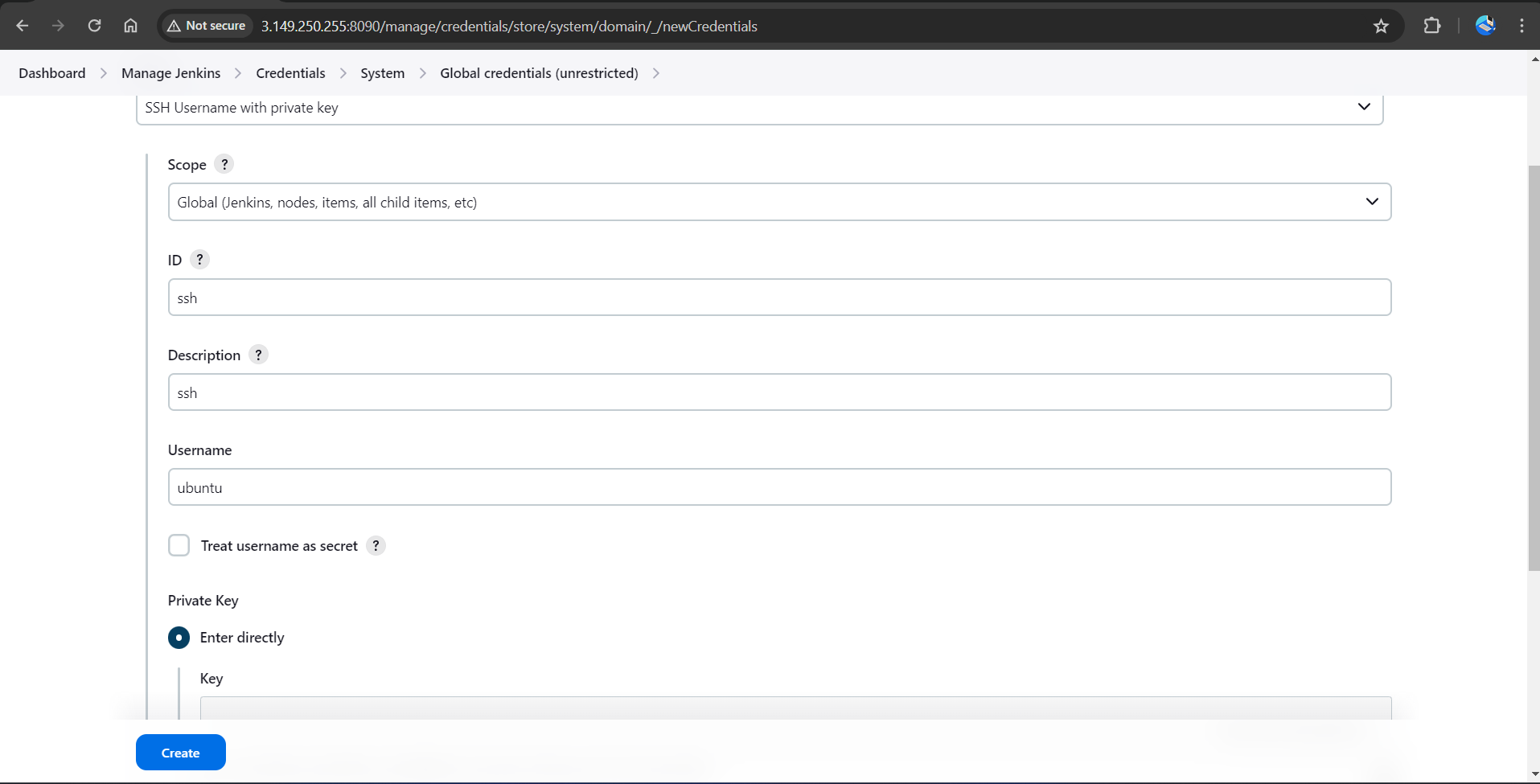
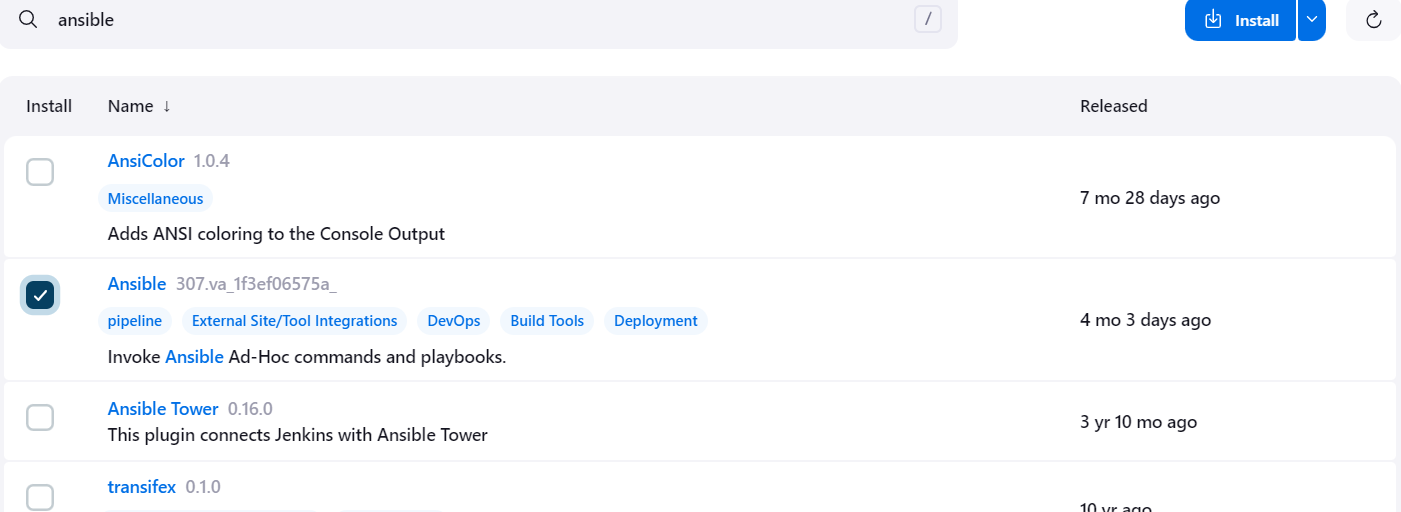
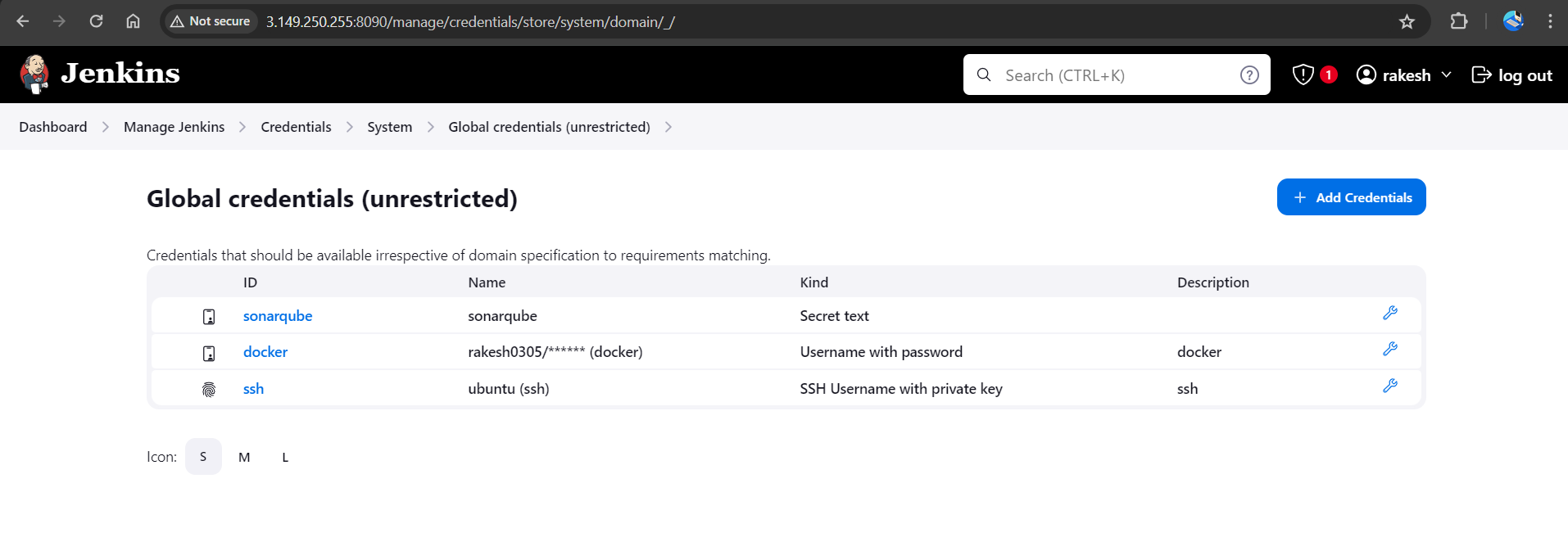
* Install Ansible on Ubuntu 22.04 LTS

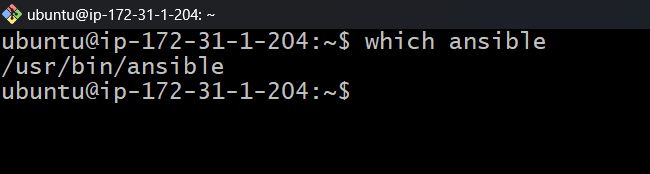
**<**sudo apt install ansible -y> <sudo apt install ansible-core -y>, <ansible –version>.

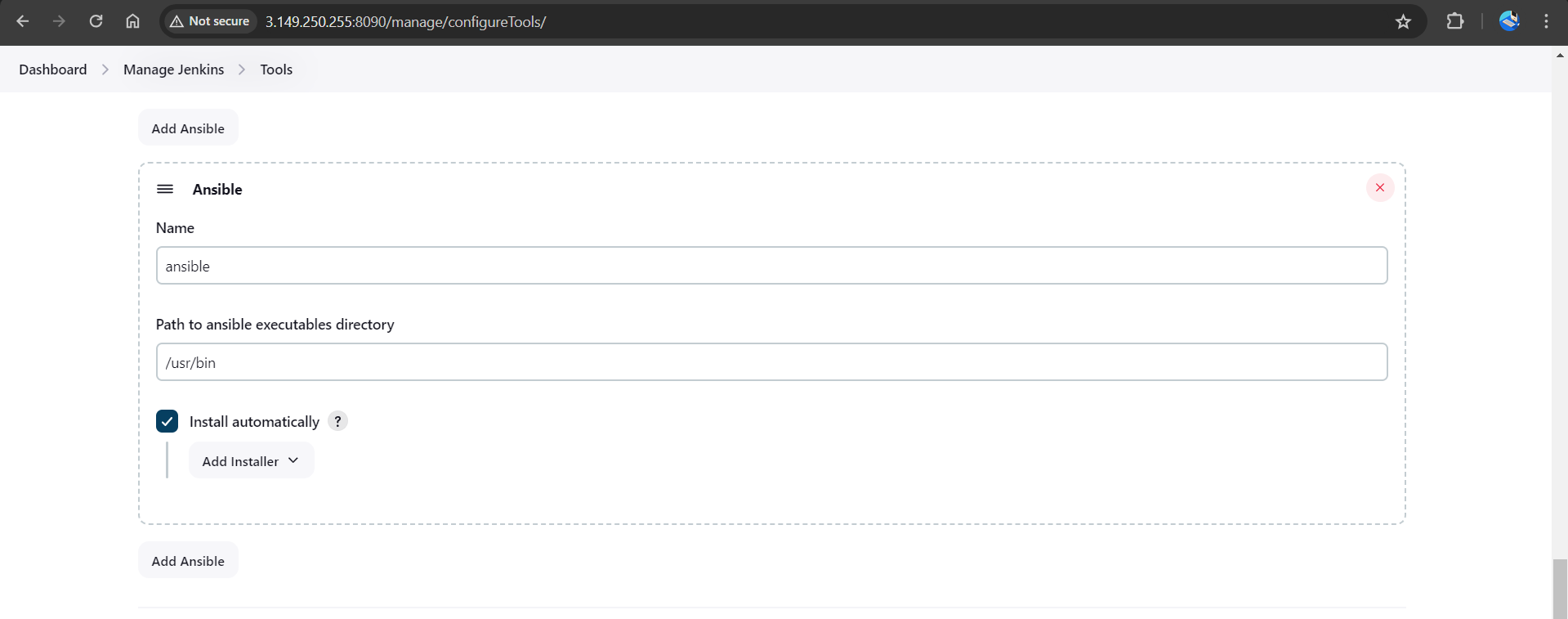
**Create an Inventory file in Ansible**

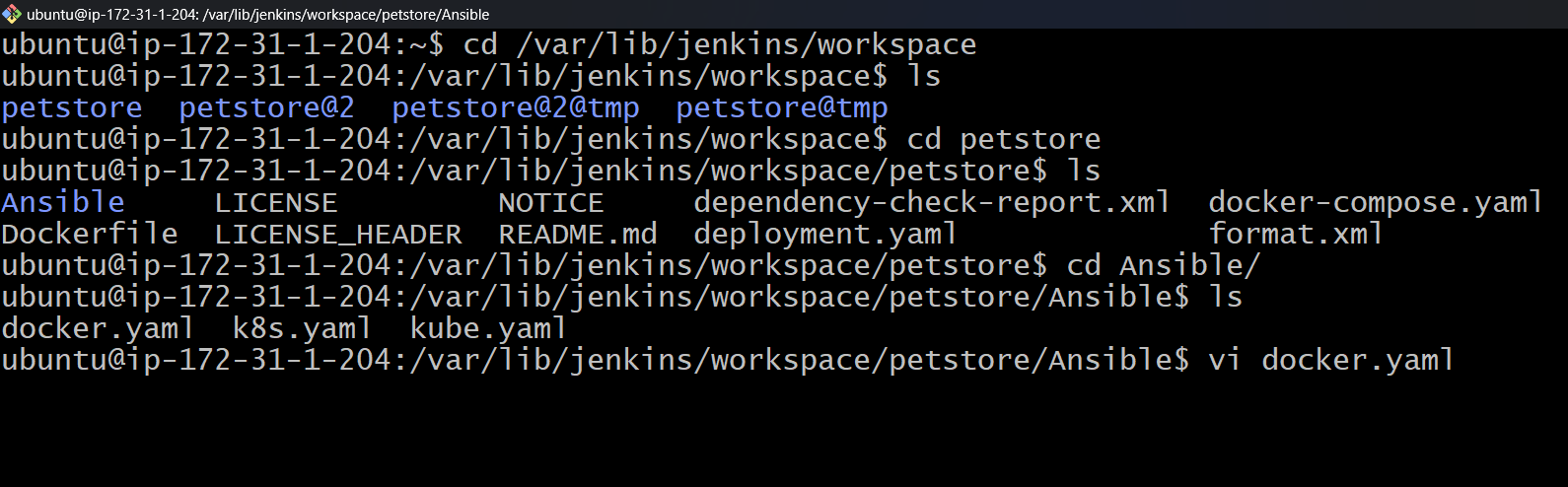
* To add inventory you can create a new directory or add in the default Ansible hosts file

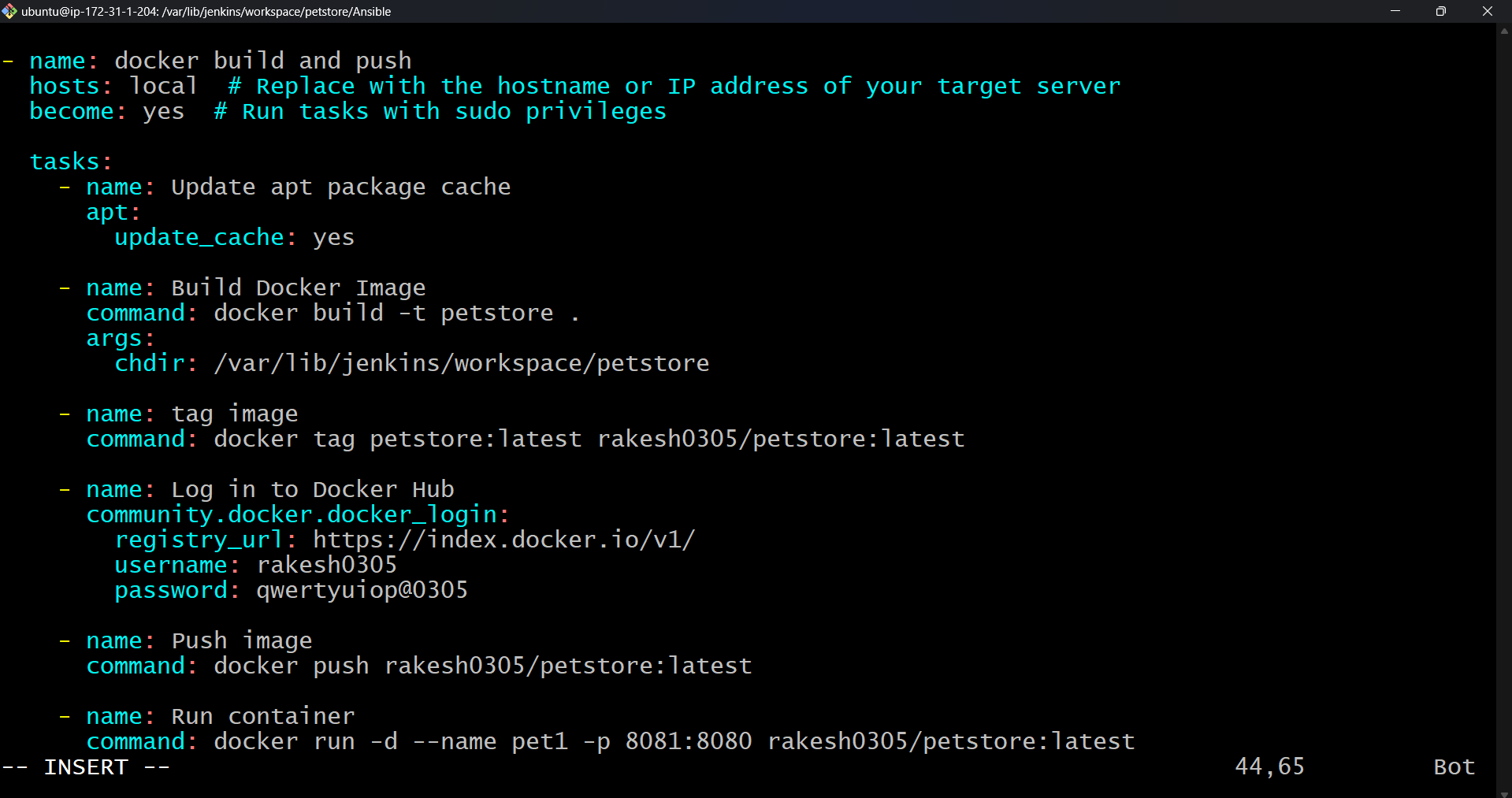
cd /etc/ansible  
sudo vi hosts

* Now go to the host file inside the Ansible server and paste the public IP of the Jenkin, save and exit from the file.
* Let's install The Ansible plugin to integrate with Jenkins.
* Now add Credentials to invoke Ansible with Jenkins.
* In the private key section, Select Enter directly and add your Pem file for the key. And finally, click on Create.
* Give this command in your Jenkins machine to find the path of your ansible which is used in the tool section of Jenkins.

 <which ansible>

* Copy that path and add it to the tools section of Jenkins at ansible installations.
* Now write an Ansible playbook to create a docker image, tag it and push it to the docker hub, and finally, we will deploy it on a container using Ansible.

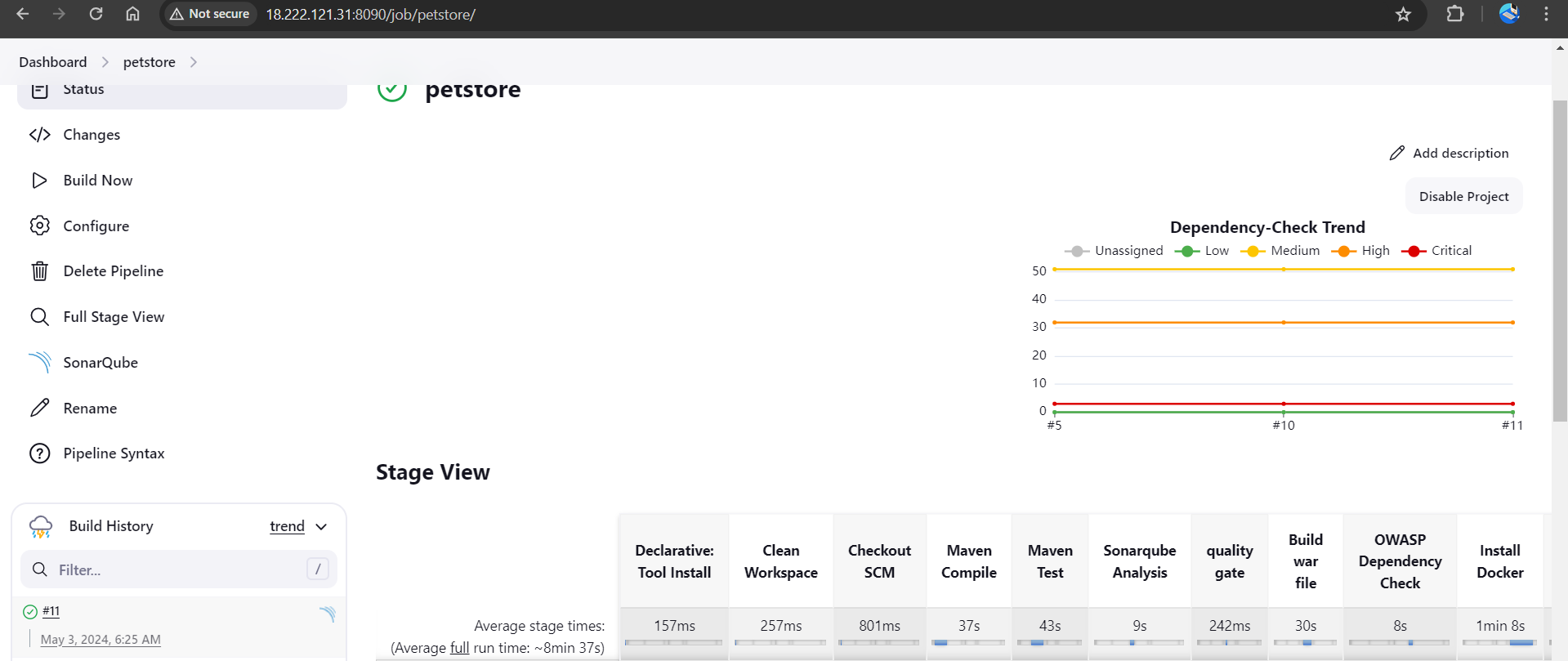


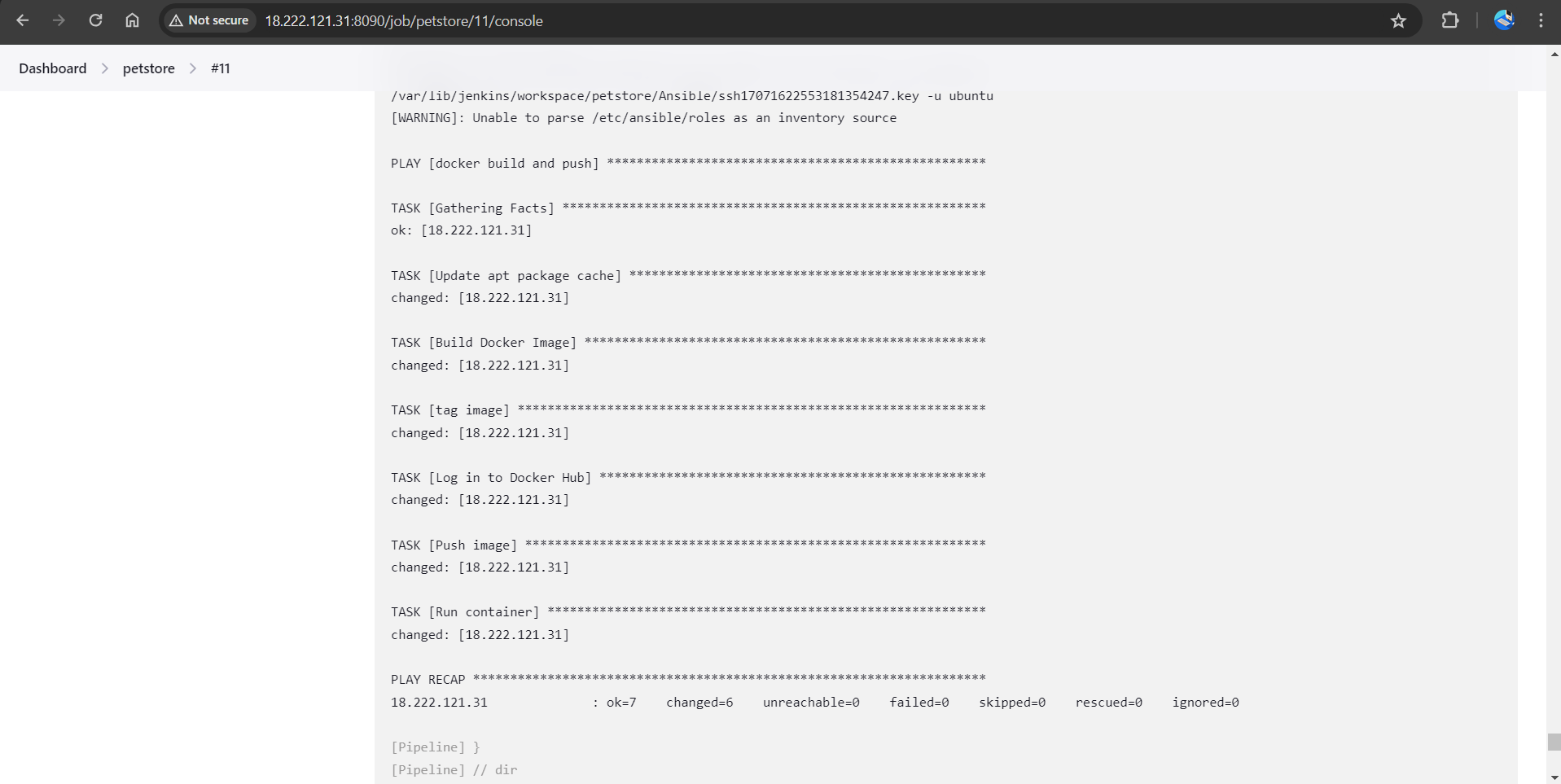




* Add this stage to the pipeline to build and push it to the docker hub, and run the container.

stage('Install Docker') {  
 steps {  
 dir('Ansible'){  
 script {  
 ansiblePlaybook credentialsId: 'ssh', disableHostKeyChecking: true, installation: 'ansible', inventory: '/etc/ansible/', playbook: 'docker-playbook.yaml'  
 }   
 }   
 }  
 }





* Output of pipeline

# 

* Now copy the IP address of Jenkins and paste it into the browser

<jenkins-ip:8081>/petstore

