CI CD DEPLOYMENT

Devops Project

DONE BY MANI SANKAR DIVI

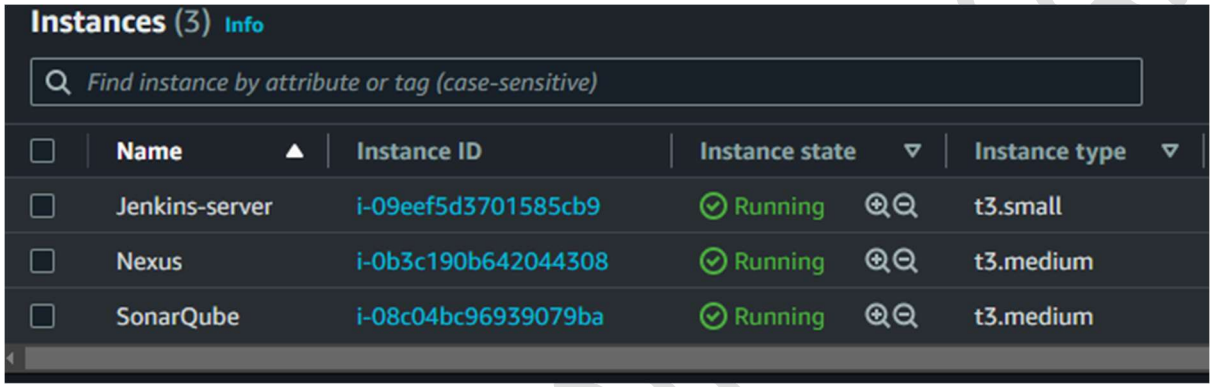
Introduction:-

In this Project I am Deploying a frontend-project, end-end deployment by using Jenkins, Maven, SonarQube, Nexus, Docker, Tomcat

- ➔ **In this we need 3 "Amazon linux-2 instances" and 2 Ubuntu instances**

Creation of instances:-

- ➔ **We need 1 instance for Jenkins-server (T3.small, 2cpu, 2gb ram)**
- ➔ **We need 2 instances for nexus, SonarQube (T3.medium, 2cpu, 4gb ram)**
- ➔ **We need 1 master k8s instance (T3.medium)**
- ➔ **We need 1 worker k8s instance (T3.micro)**



After installing 5 instances we need to connect SSH over Putty / Mobaxterm / Gitbash

Step-1: Jenkins-server

- ➔ **Connect Jenkins-server through putty/gitbash/mobaxterm**
- ➔ **We need to install Java, Maven, Jenkins, Docker**
- ➔ **Follow below commands**

$ sudo yum update -y

$ sudo -i

$ sudo yum install git tree -y

$ sudo yum install default java -y

$ java –version

- **➔ Now we need to install Maven.**

We can download maven manually latest version.

$ w

get https://dlcdn.apache.org/maven/maven-3/3.9.3/binaries/apache-maven-3.9.3-bin.tar.gz

$ ls

$ tar -xvzf apache-maven-3.9.3-bin.tar.gz

- **➔ Rename maven package after untar**

$ mv apache-maven-3.9.3 maven

$ mv maven /opt

- **➔ So we need to set JAVA_HOME_PATH & MAVEN_HOME_PATH**
- **➔ Usually java located in /usr/lib/jvm**

$ cd /usr/lib/jvm

$ ls

$ cd java-17-amazon-corretto.x86_64

$ pwd

➔ **Copy the java directory path**

$ /usr/lib/jvm/java-17-amazon-corretto.x86_64

$ cd

$ vi .bash_profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

JAVA_HOME=/usr/lib/jvm/java-17-amazon-corretto.x86_64
JAVA=/usr/lib/jvm/java-17-amazon-corretto.x86_64/bin
M2_HOME=/opt/maven
M2=/opt/maven/bin

PATH=$PATH:$HOME/bin:$JAVA_HOME:$JAVA:$M2_HOME:$M2

export PATH
~
```

$ logout

$ sudo -i

$ echo $JAVA_HOME

$ echo $M2_HOME

Now we need to install docker and Jenkins in Jenkins-server

➔ **To download Jenkins click on this link https://pkg.jenkins.io/redhat-stable/**

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

$ sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

$ sudo yum install jenkins -y

$ systemctl start jenkins

$ systemctl enable jenkins

$ systemctl status jenkins

$ jenkins --version

$ sudo yum install docker -y

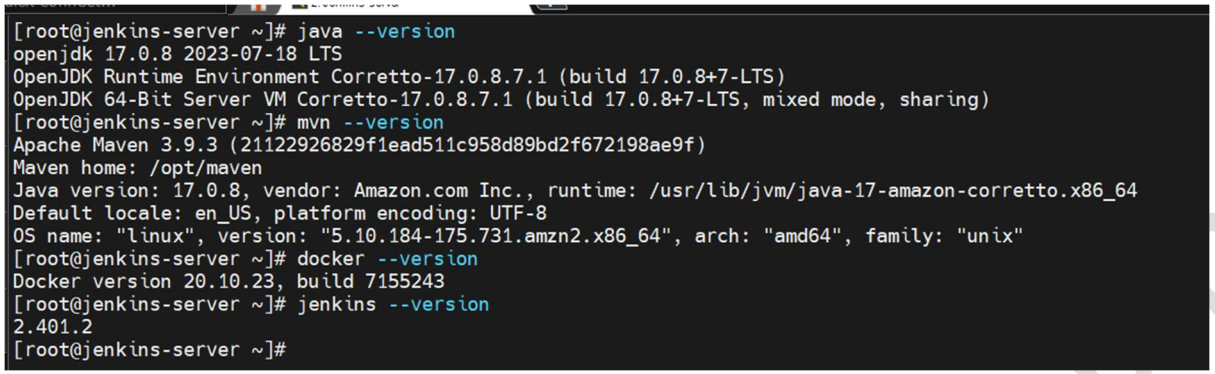
$ systemctl start docker

$ systemctl enable docker

$ sudo usermod -aG docker jenkins

$ chmod 777 /var/run/docker.sock

$ docker version



Step-2: sonarqube-server

Now connect sonarqube server through putty/mobaxtem/gitbash

$ sudo -i

$ sudo yum update -y

$ sudo yum install default java -y

$ java –version

- ➔ **Now we need to install sonarqube package, manually**
- ➔ **Click https://www.sonarsource.com/products/sonarqube/downloads/**
- ➔ **Click on this link https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.1.69595.zip**

$ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.9.1.69595.zip

$ unzip sonarqube-9.9.1.69595.zip

$ mv sonarqube-9.9.1.69595 sonar

$ mv sonar /opt

$ cd /opt/sonar

$ useradd sonar

$ passwd sonar

$ visudo

```
##
## The COMMANDS section may have other options
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
sonar ALL=(ALL)         NOPASSWD: ALL
## Allows members of the 'sys' group to run ne
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, S
```

$ sudo chown -R sonar:sonar /opt/sonar

$ sudo chmod -R 774 /opt/sonar

$ su - sonar

$ cd /opt/sonar

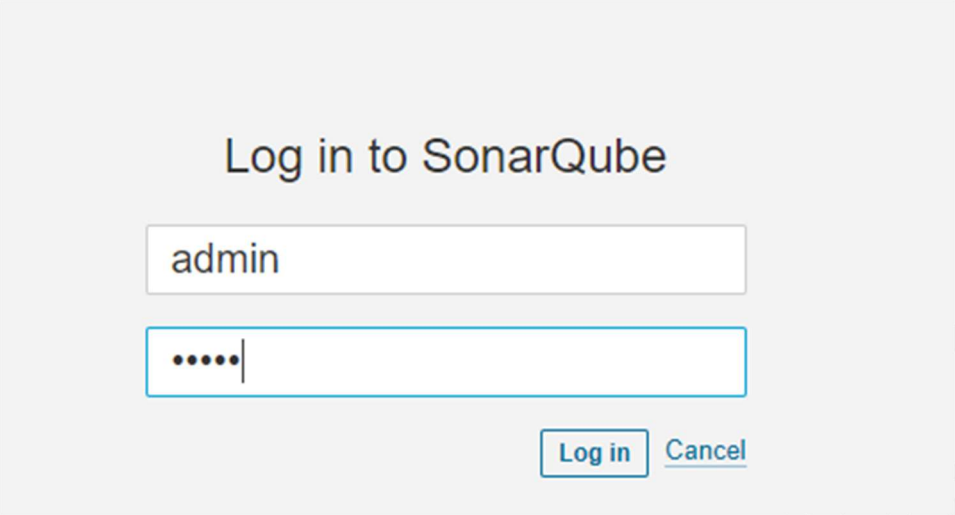
$ cd /opt/sonar/bin/linux-x86-64

$ ./sonar.sh start

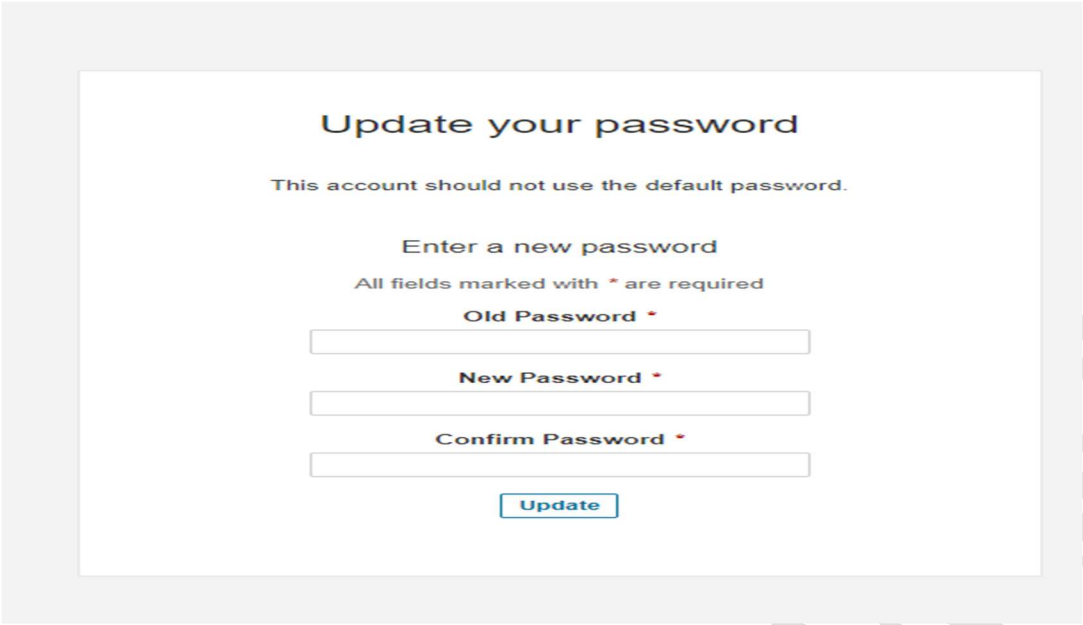
$ ./sonar.sh status

➔ **Sonarqube runs on port num : 9000**

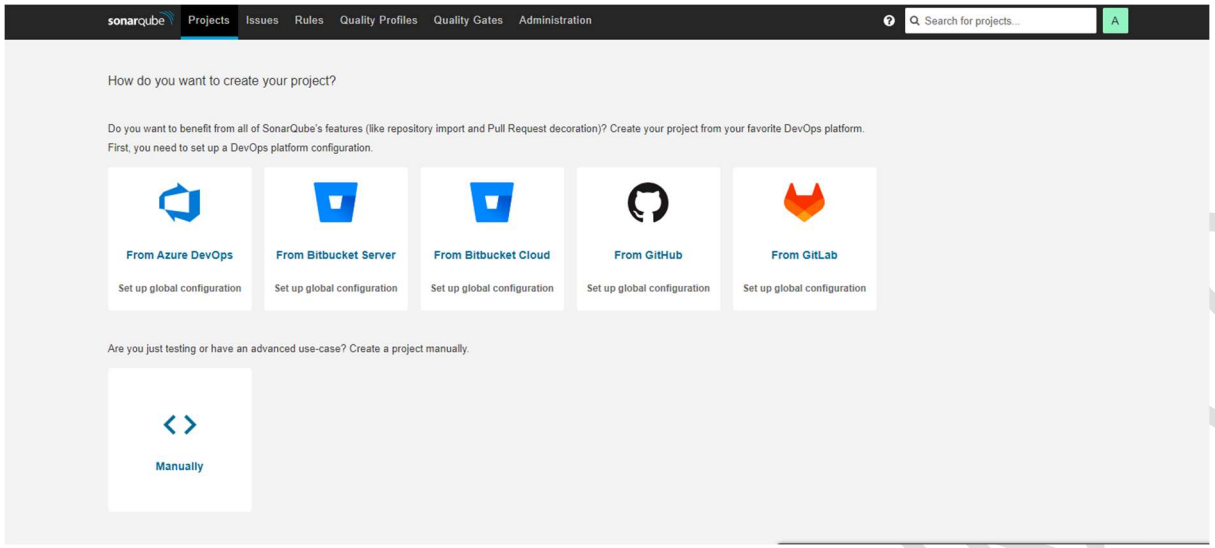
➔ http://sonar-serverpublic-ip:9000



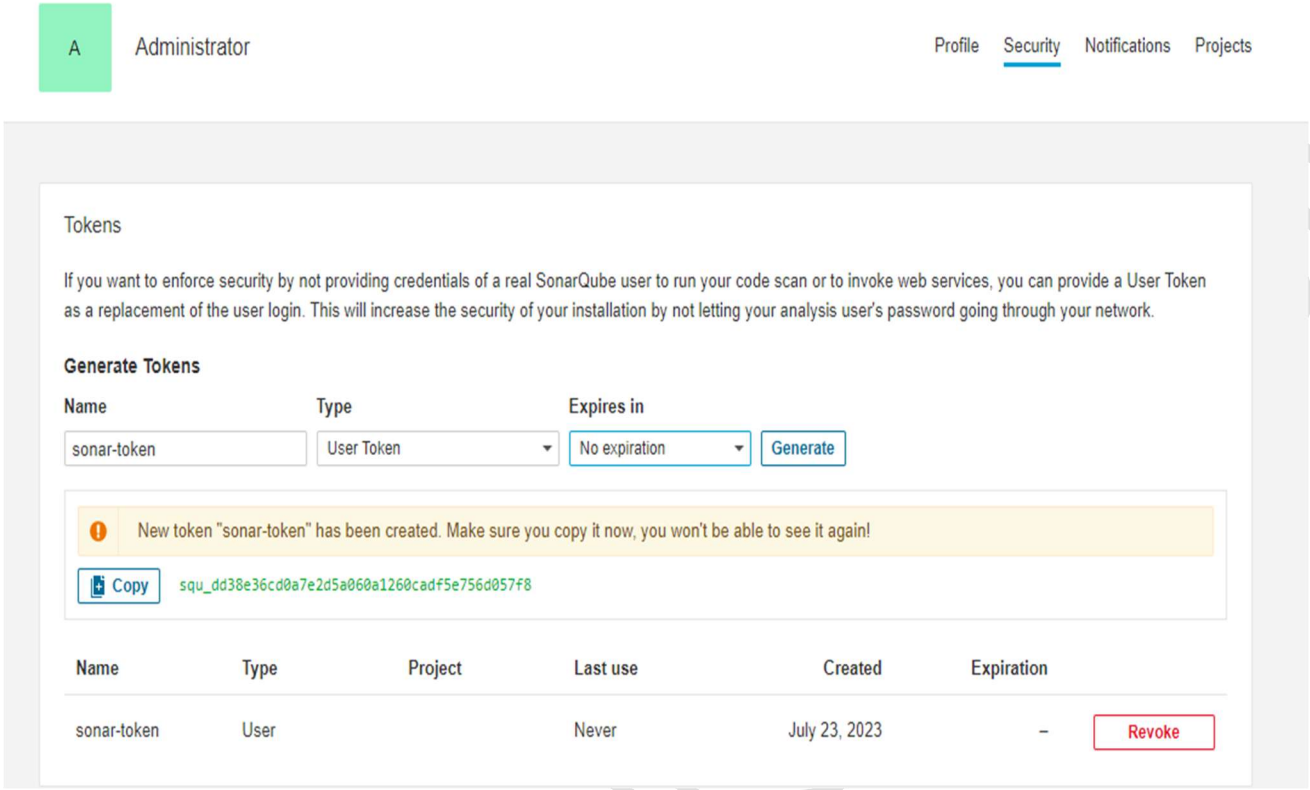
➔ **Username & password = admin**



➔ **Change to old to new password for sonarqube**



➔ **Click on Administrator >> Account >> security**



➔ **Copy the sonar-token id = squ_ed1f036e56b7ffc6e5f63bd1f6869c0fb3f58bed**

➔ **Below 2 lines should add on src code, pom.xml file**

➔ **<sonar.host.url>** http://18.60.85.230:9000**</sonar.host.url>**

➔ **<sonar.login>squ_ed1f036e56b7ffc6e5f63bd1f6869c0fb3f58bed</sonar.login>**

Step-3: Nexus-server

Now connect Nexus-server through putty/mobaxterm/gitbash

Nexus run on port num :8081

- ➔ **Nexus requires java-1.8**

$ sudo yum update -y

$ sudo -i

$ sudo yum install java-1.8.0-openjdk.x86_64 -y

$ java -version

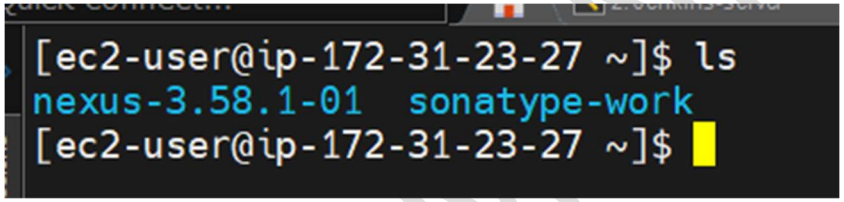
- ➔ **Now we need to download nexus package manually**
- ➔ **Click on https://help.sonatype.com/repomanager3/product-information/download**

$ cd /opt

$ wget https://download.sonatype.com/nexus/3/nexus-3.58.1-01-unix.tar.gz

$ tar -xvzf nexus-3.58.1-01-unix.tar.gz

$ ls



$ mv nexus-3.58.1-01 nexus

$ useradd nexus

$ passwd nexus

$ visudo

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
nexus   ALL=(ALL)       NOPASSWD: ALL
## Allows members of the 'sys' group to run n
## service management apps and more.
```

$ sudo chown -R nexus:nexus /opt/nexus

$ sudo chown -R nexus:nexus /opt/sonatype-work

$ su – nexus

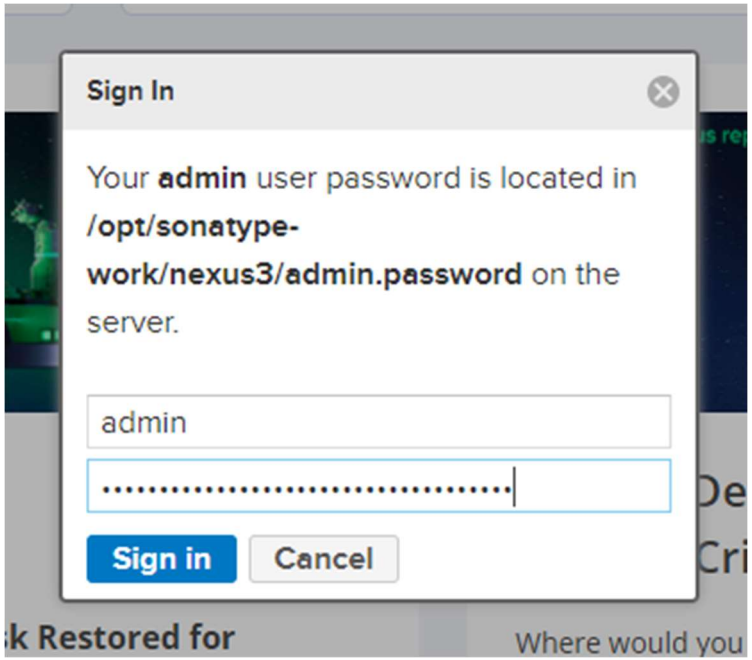
$ cd /opt/nexus/bin

$ ./nexus start

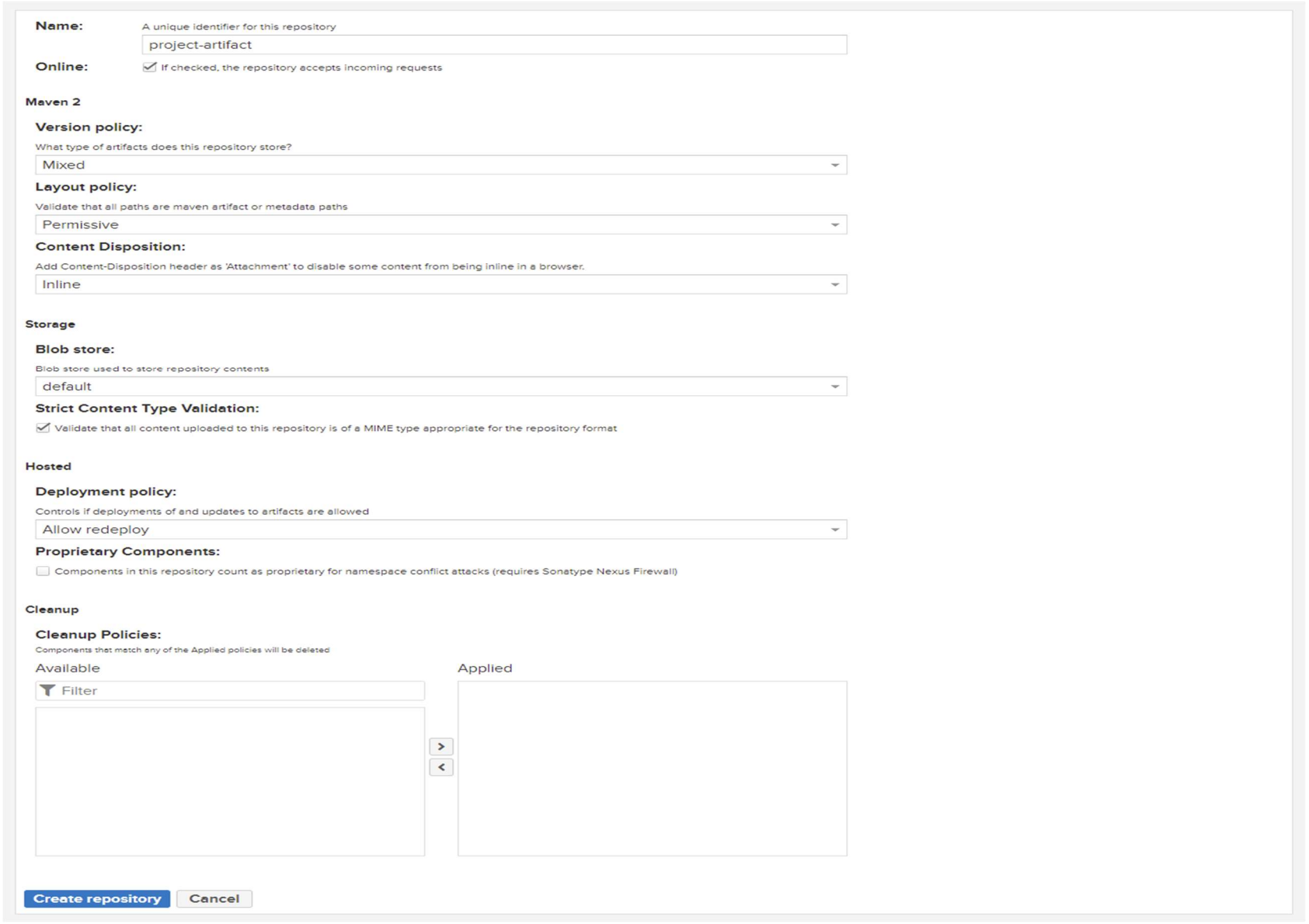
$ ./nexus status

- ➔ **http://nexus-server publicIP:8081**
- ➔ **Click on sign in >> username = admin, passwd = /opt/sonatype-work/nexus3/admin.password**

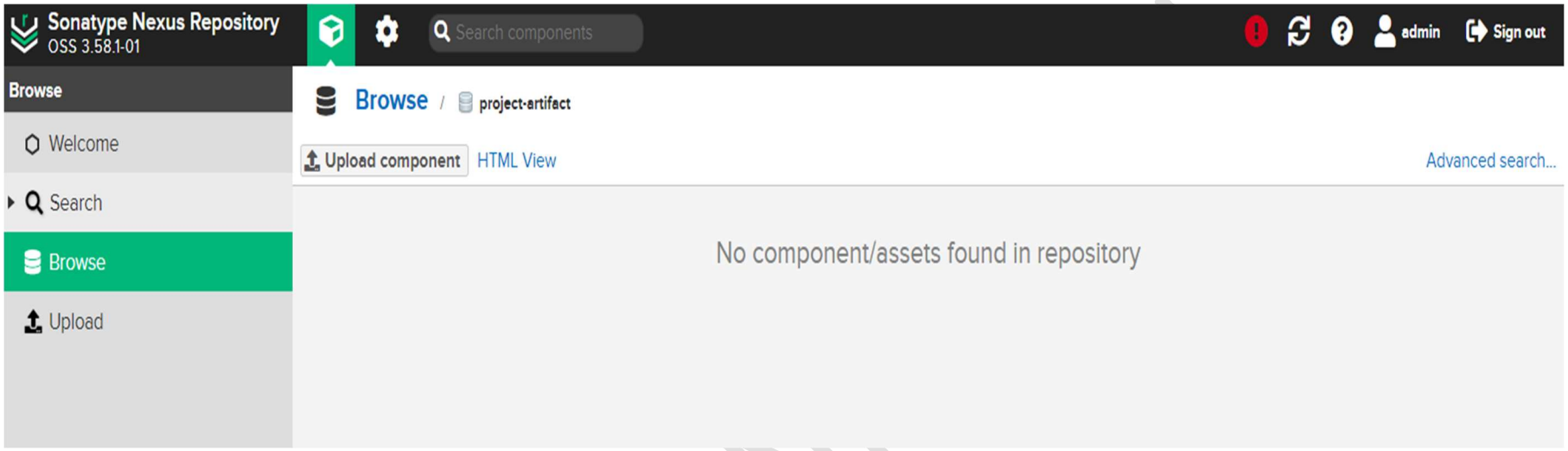
$ cat /opt/sonatype-work/nexus3/admin.password



- ➔ **We need to change password**
- ➔ **Click on settings icon >> click on repositories >> click on create repository >> choose maven (hosted)**
- ➔ **See below image for more clarity**


Name:
A unique identifier for this repository
project-artifact
Online:
If checked, the repository accepts incoming requests
Maven 2
Version policy:
What type of artifacts does this repository store?
Mixed
Layout policy:
Validate that all paths are maven artifact or metadata paths
Permissive
Content Disposition:
Add Content-Disposition header as 'Attachment' to disable some content from being inline in a browser.
Inline
Storage
Blob store:
Blob store used to store repository contents
default
Strict Content Type Validation:
Validate that all content uploaded to this repository is of a MIME type appropriate for the repository format
Hosted
Deployment policy:
Controls if deployments of and updates to artifacts are allowed
Allow redeploy
Proprietary Components:
Components in this repository count as proprietary for namespace conflict attacks (requires Sonatype Nexus Firewall)
Cleanup
Cleanup Policies:
Components that match any of the Applied policies will be deleted
Available
Filter
Applied
Create repository
Cancel

➔ **Click on browse >> click on your created repo**



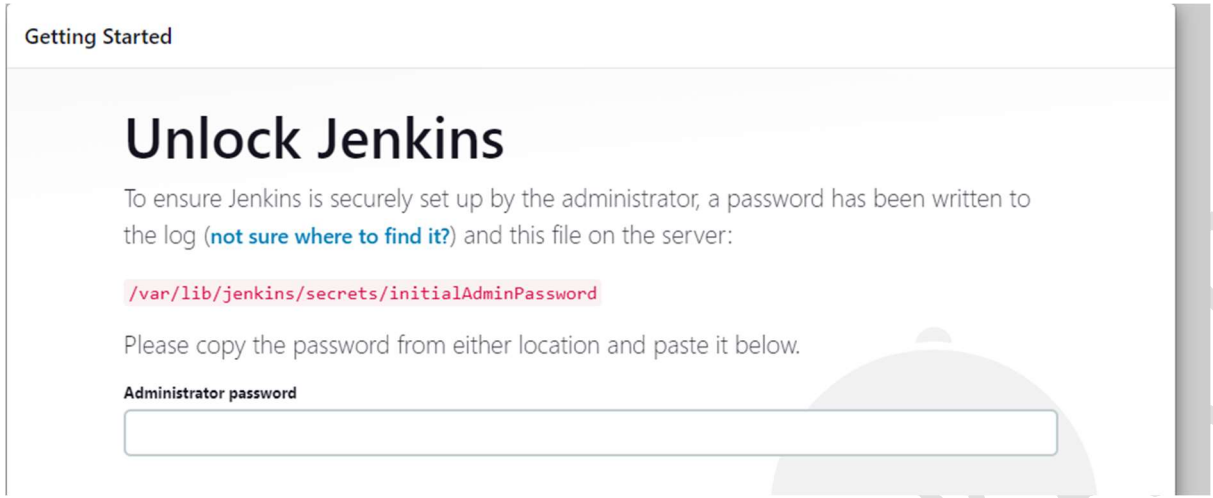
So Jenkins & Docker & SonarQube & nexus installed successfully.

- ➔ **We need to access Jenkins server**
- ➔ **Login Jenkins dashboard**
- ➔ **Install plugins and Tools configuration step by step process**

Step-4:

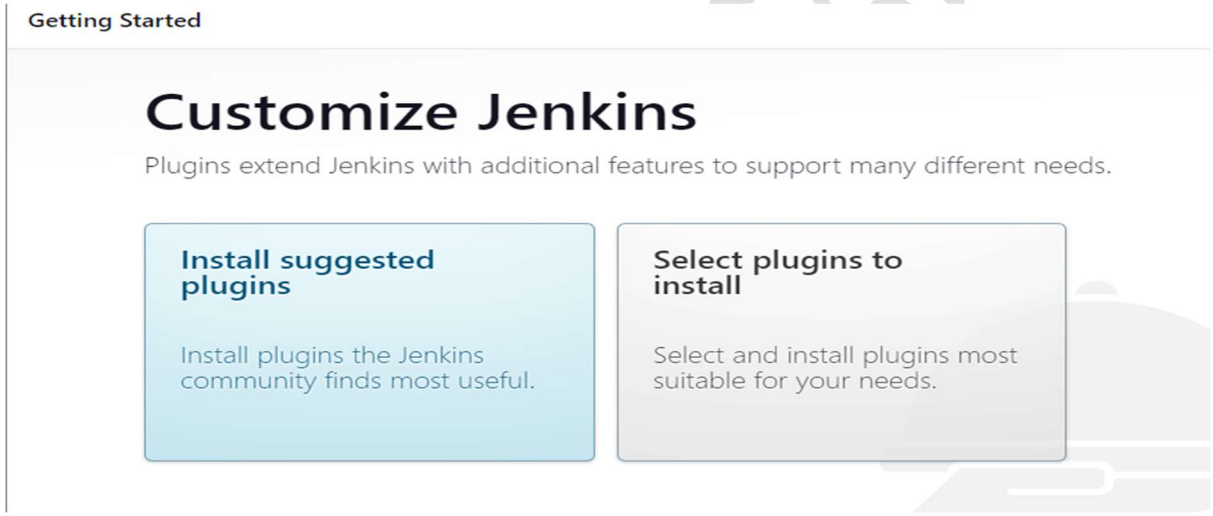
http://jenkins-server publicIP:8080 paste in your browser.

Jenkins runs under port num : 8080

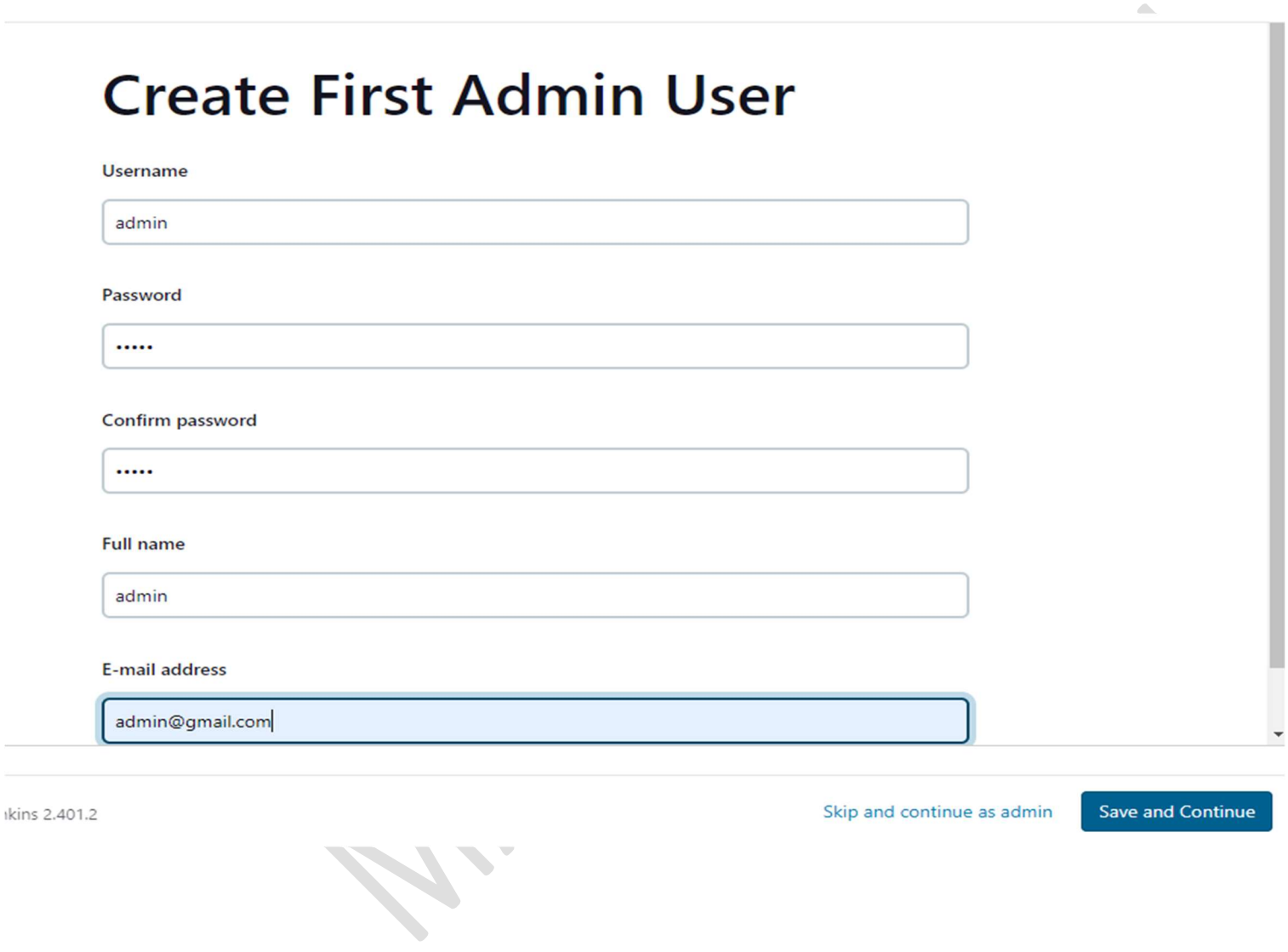


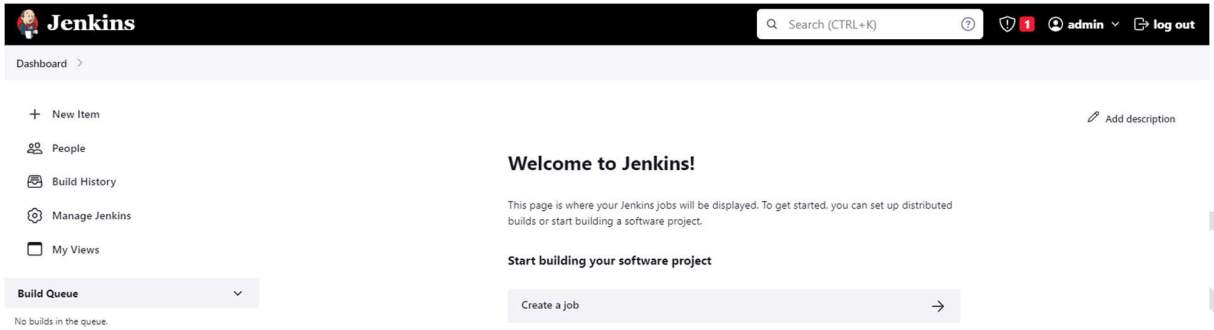
$ cat /var/lib/jenkins/secrets/initialAdminPassword

- ➔ **Copy key and paste in there**
- ➔ **Click on install suggested plugins**



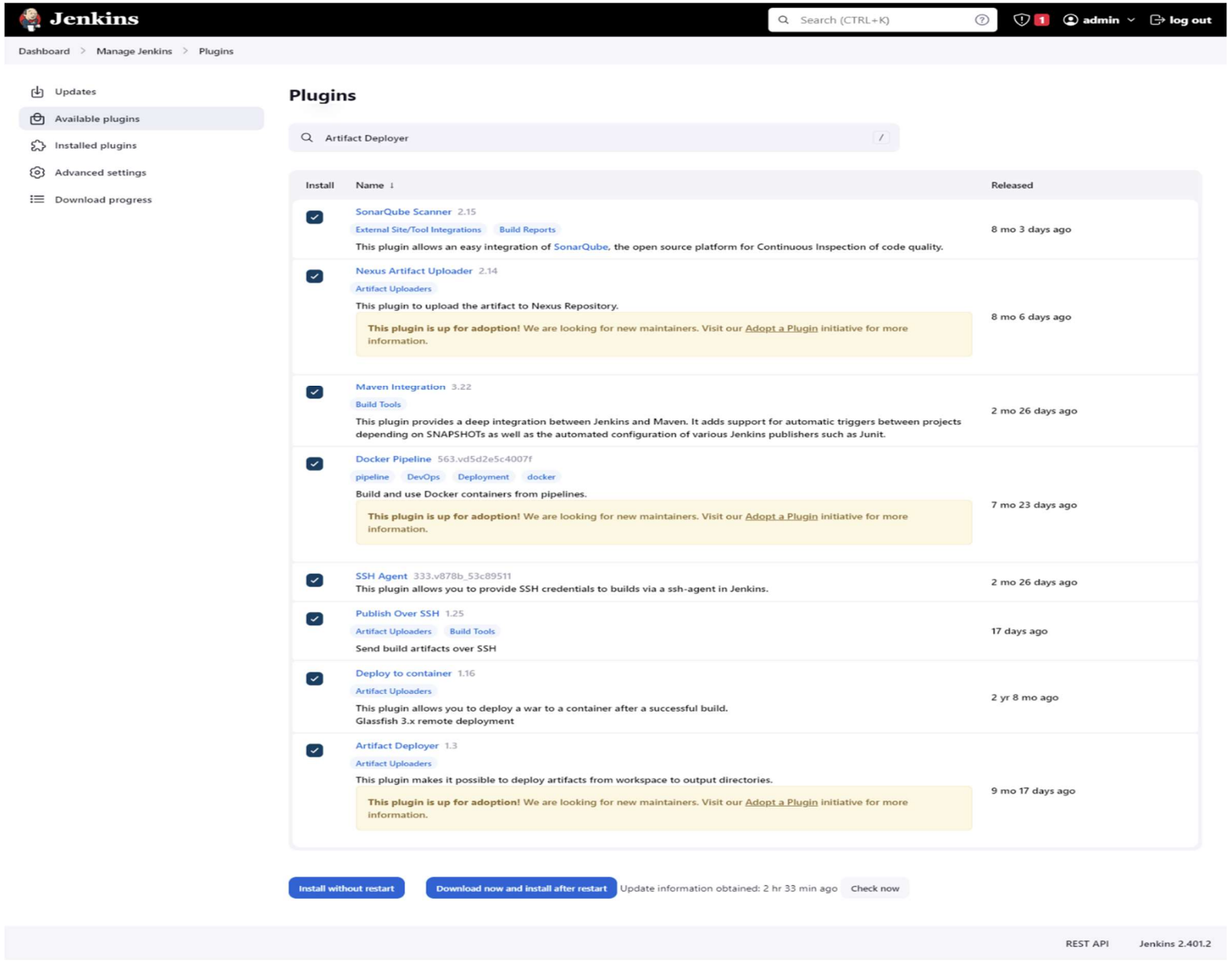
➔ **Create account username, password, email id.**





Now we need to install some plugins:

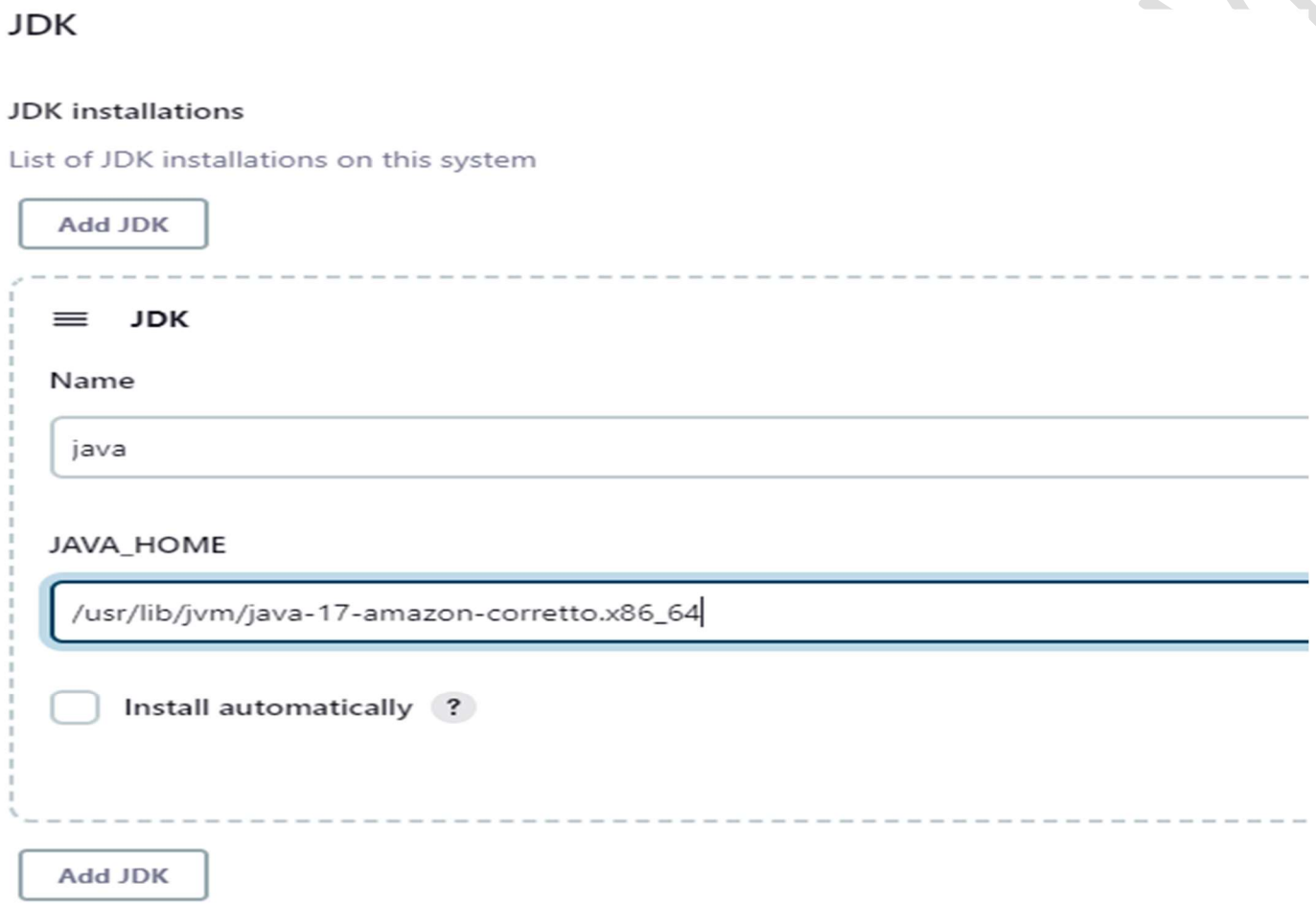
- ➔ **Click on manage jenkins >> click on plugins >> click on Available plugins**
- ➔ **Jenkins pulgins we need to download:**
- ➔ **SonarQube Scanner**
- ➔ **Nexus Artifact Uploader**
- ➔ **Maven Integration**
- ➔ **Docker Pipeline**
- ➔ **SSH Agent**
- ➔ **Publish Over SSH**
- ➔ **Deploy to container**
- ➔ **Artifact Deployer**
- ➔ **kubernetes**

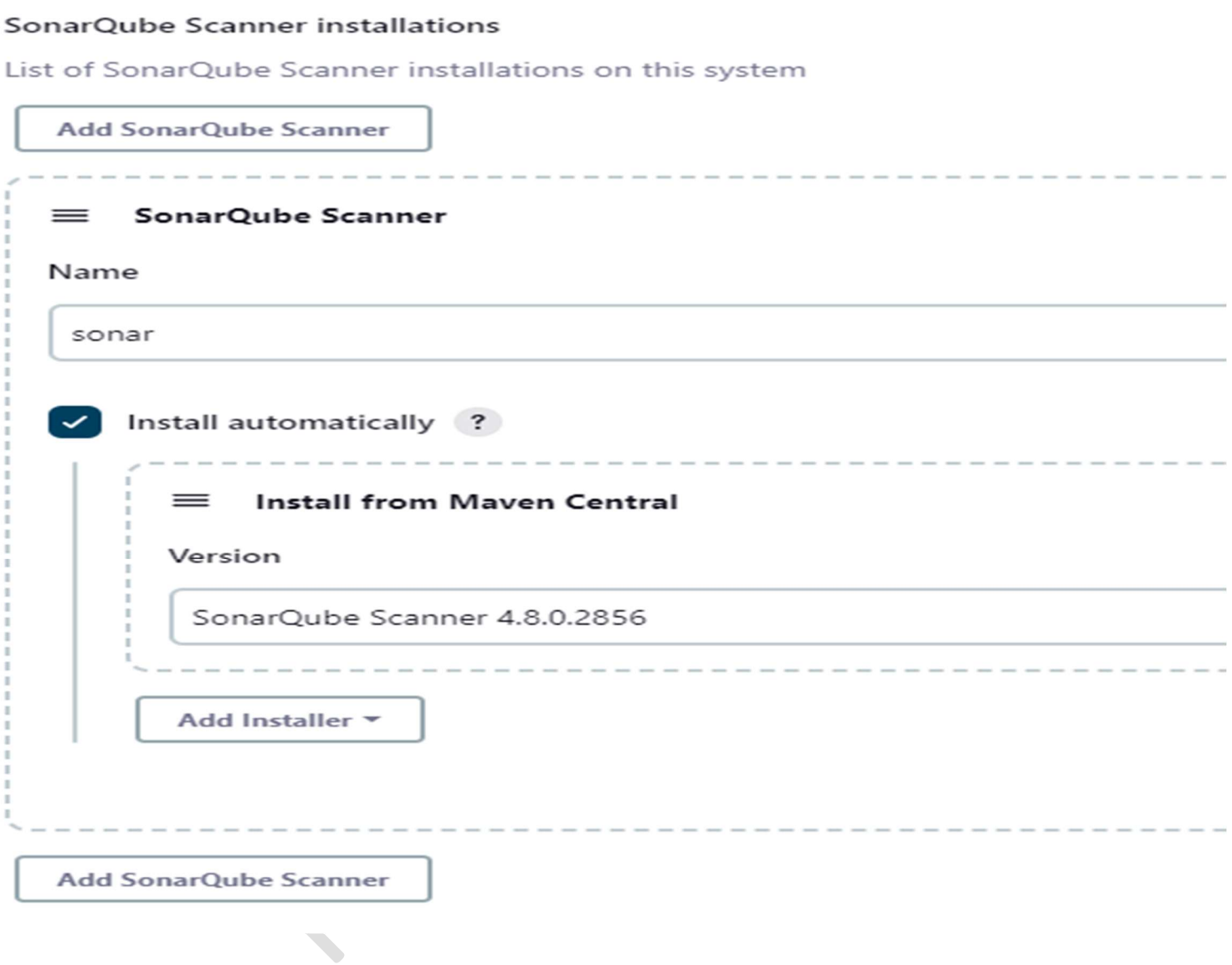

Jenkins
Search (CTRL+K)
admin
log out
Dashboard > Manage Jenkins > Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Plugins
Artifact Deployer
Install
Name
Released
SonarQube Scanner 2.15
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
8 mo 3 days ago
Nexus Artifact Uploader 2.14
Artifact Uploaders
This plugin to upload the artifact to Nexus Repository.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
8 mo 6 days ago
Maven Integration 3.22
Build Tools
This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.
2 mo 26 days ago
Docker Pipeline 563.vd5d2e5c4007f
pipeline
DevOps
Deployment
docker
Build and use Docker containers from pipelines.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
7 mo 23 days ago
SSH Agent 333.v878b_53c89511
This plugin allows you to provide SSH credentials to builds via a ssh-agent in Jenkins.
2 mo 26 days ago
Publish Over SSH 1.25
Artifact Uploaders
Build Tools
Send build artifacts over SSH
17 days ago
Deploy to container 1.16
Artifact Uploaders
This plugin allows you to deploy a war to a container after a successful build.
Glassfish 3.x remote deployment
2 yr 8 mo ago
Artifact Deployer 1.3
Artifact Uploaders
This plugin makes it possible to deploy artifacts from workspace to output directories.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
9 mo 17 days ago
Install without restart
Download now and install after restart
Update information obtained: 2 hr 33 min ago
Check now
REST API
Jenkins 2.401.2

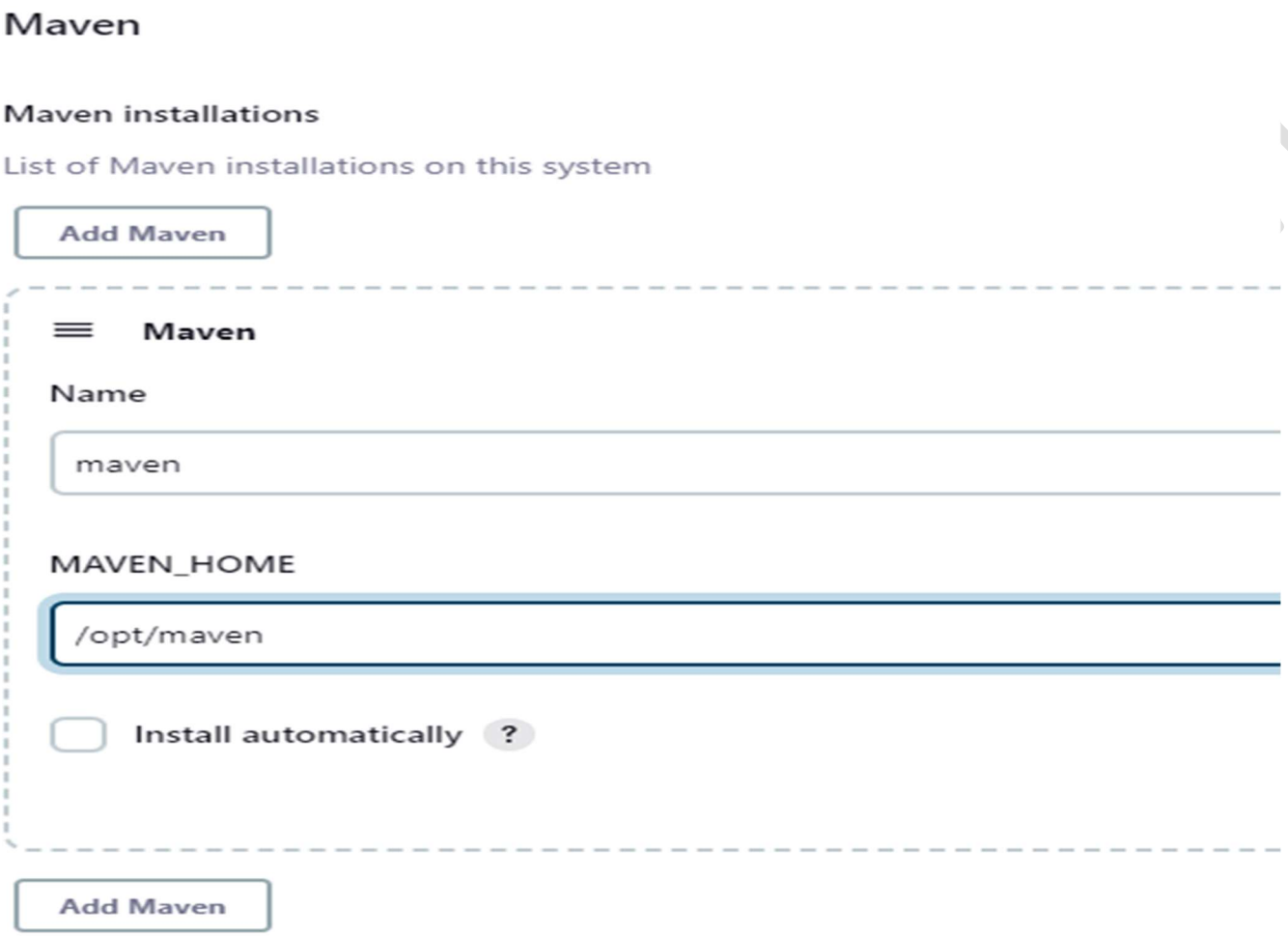
➔ **Click on install without restart**

➔ **Now we need to Configure tools like maven, java, nexus , sonar, Docker**

➔ **Click on manage jenkins >> click on Tools >>**




SonarQube Scanner installations
List of SonarQube Scanner installations on this system
Add SonarQube Scanner
SonarQube Scanner
Name
sonar
Install automatically
?
Install from Maven Central
Version
SonarQube Scanner 4.8.0.2856
Add Installer
Add SonarQube Scanner



➔ **Click on manage jenkins >> click on system >> select SonarQube servers**

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

Environment variables Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name

sonar

Server URL

Default is http://localhost:9000

http://18.60.251.203:9000

Server authentication token

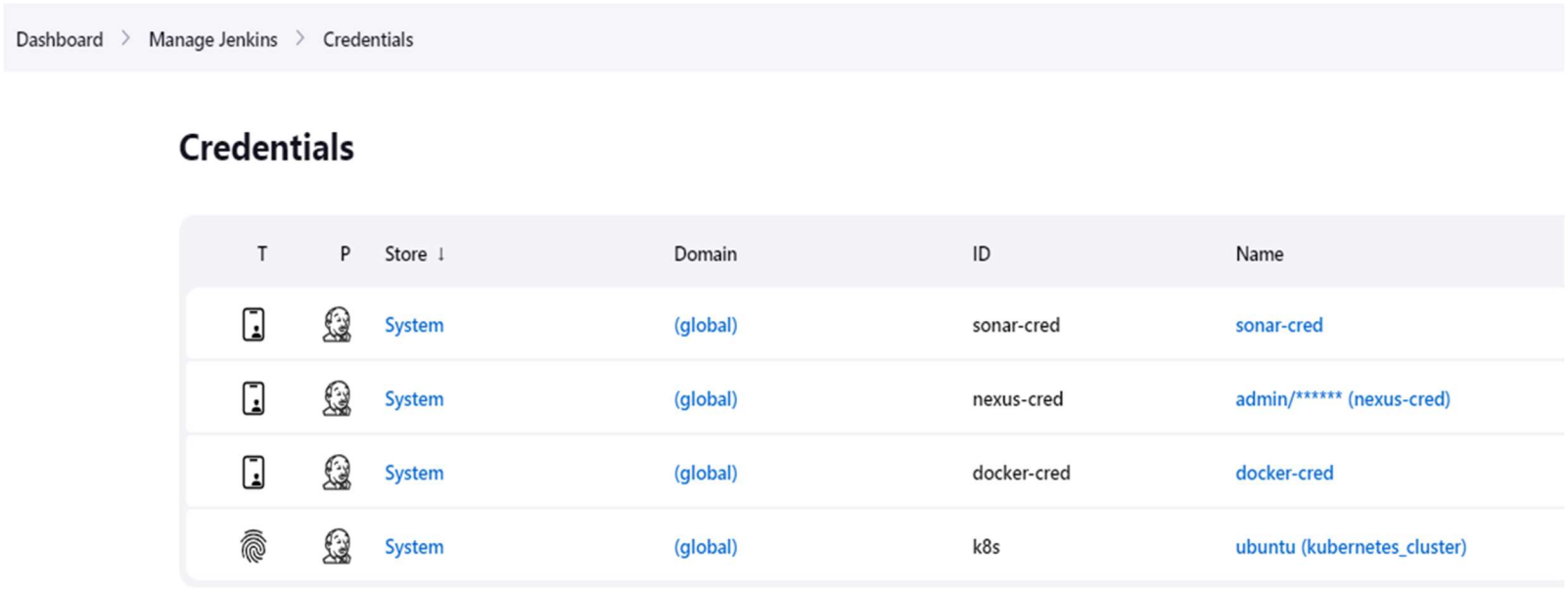
SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-cred

Add

Advanced

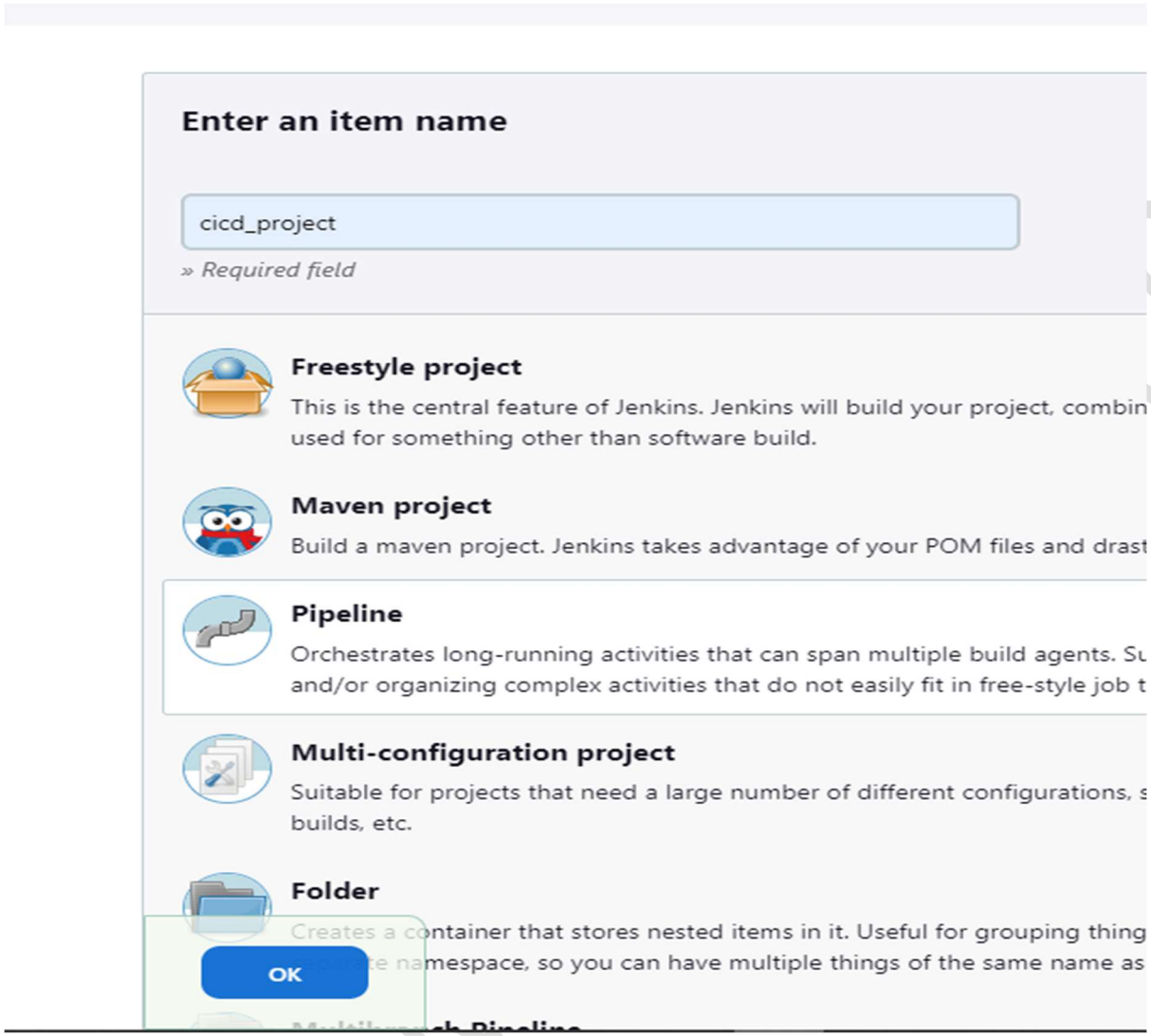
Add SonarQube



Step-5:

Now we need to Create a new job

➔ **Click on New item >> Enter any job-name>> select pipeline job >> click on ok.**


Enter an item name
cicd_project
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combin
used for something other than software build.
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drast
Pipeline
Orchestrates long-running activities that can span multiple build agents. Su
and/or organizing complex activities that do not easily fit in free-style job t
Multi-configuration project
Suitable for projects that need a large number of different configurations, s
builds, etc.
Folder
Creates a container that stores nested items in it. Useful for grouping thing
namespace, so you can have multiple things of the same name as
OK

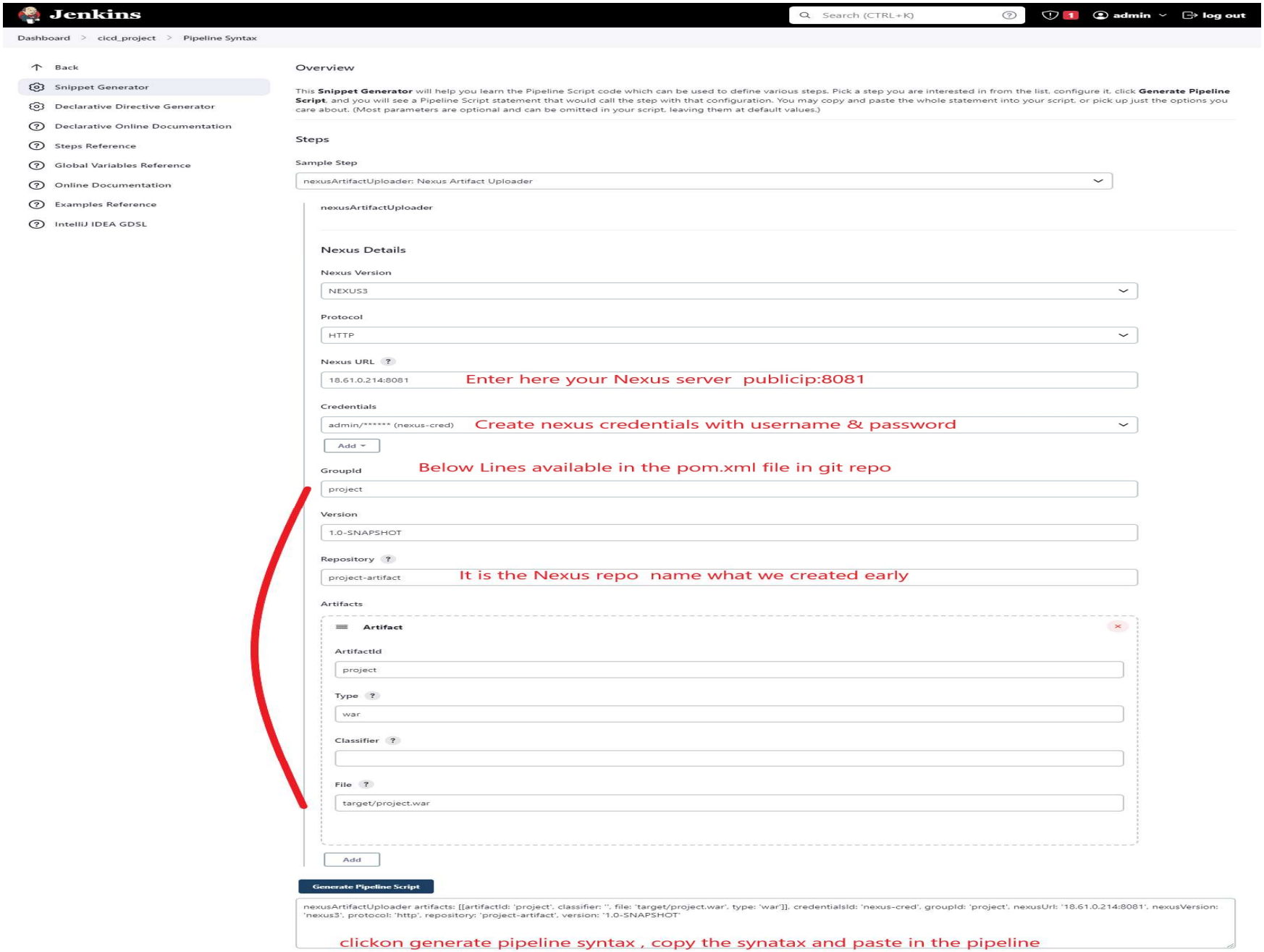
Github url: https://github.com/MANISANKARDIVI/Project-repo.git

Declarative pipeline:

```
pipeline{
   agent any
   environment {
      PATH = "$PATH:/opt/maven/bin"
   }
   stages{
      stage('get code'){
         steps{
            git branch: 'main', url: 'https://github.com/MANISANKARDIVI/project-repo.git'
         }
      }
      stage('Maven Build'){
         steps{
            sh 'mvn clean package'
         }
      }
      stage('code analysis'){
         steps{
            withSonarQubeEnv('sonarqube'){
               sh "mvn sonar:sonar"
            }
         }
      }

   }
}
```

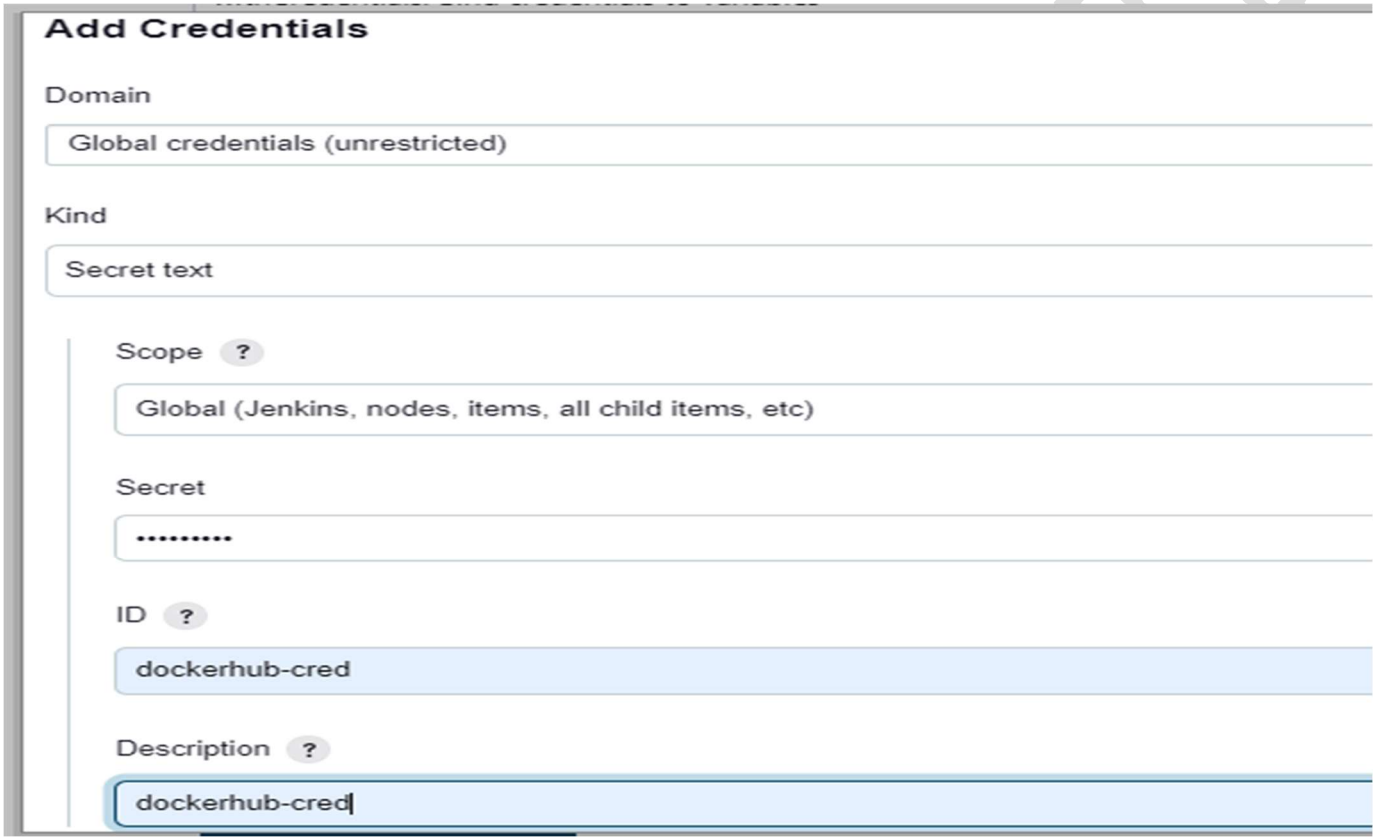
➔ **Next we need to add Nexus and Docker and K8s syntax follow below images and steps to generate syntax.**

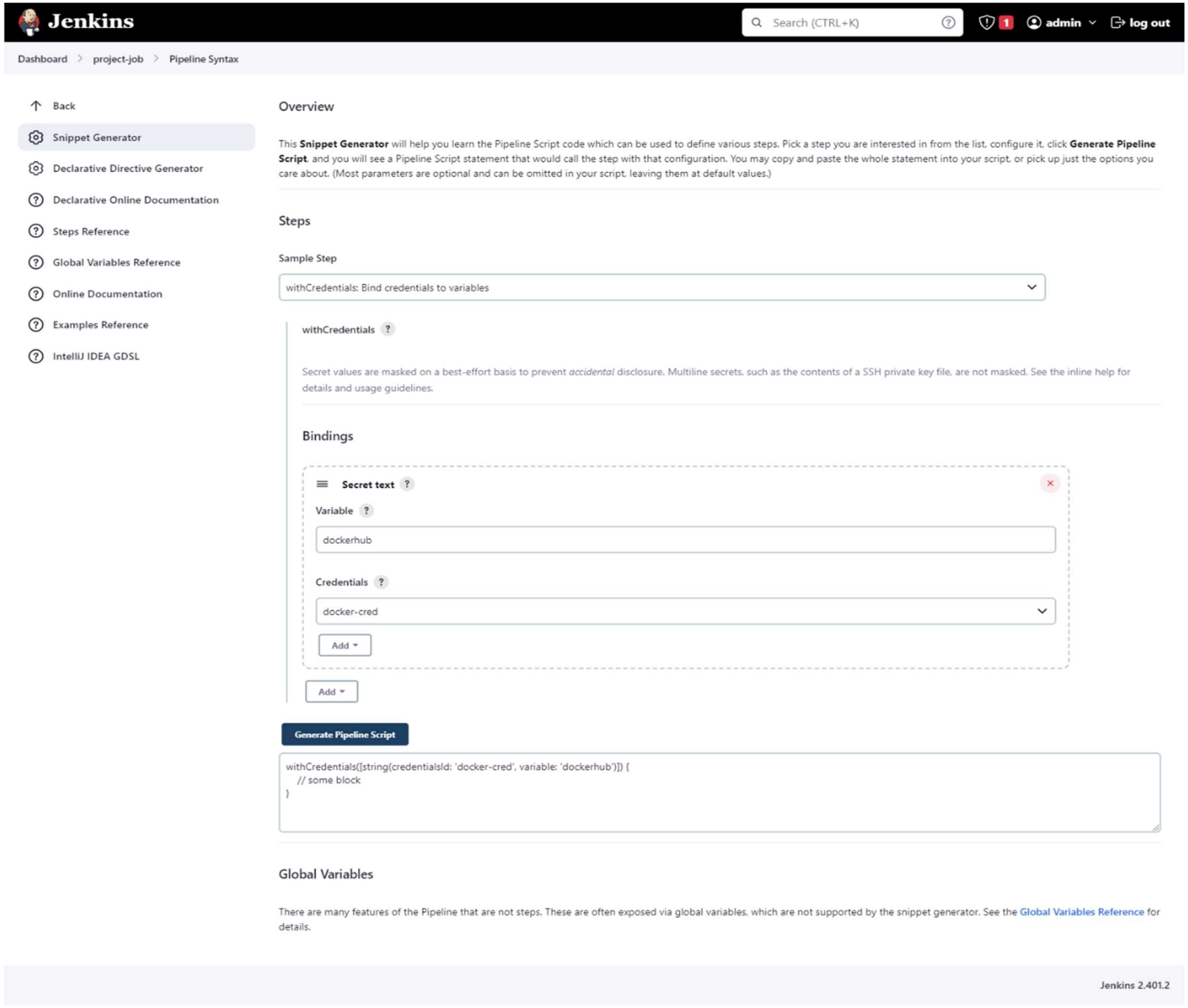

Jenkins
Search (CTRL+K)
admin
log out
Dashboard > cicd_project > Pipeline Syntax
Back
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)
Steps
Sample Step
nexusArtifactUploader: Nexus Artifact Uploader
nexusArtifactUploader
Nexus Details
Nexus Version
NEXUS3
Protocol
HTTP
Nexus URL
18.61.0.214:8081
Enter here your Nexus server publicip:8081
Credentials
admin/****** (nexus-cred)
Create nexus credentials with username & password
Add
GroupId
Below Lines available in the pom.xml file in git repo
project
Version
1.0-SNAPSHOT
Repository
project-artifact
It is the Nexus repo name what we created early
Artifacts
Artifact
ArtifactId
project
Type
war
Classifier
File
target/project.war
Add
Generate Pipeline Script
nexusArtifactUploader artifacts: [[artifactId: 'project', classifier: '', file: 'target/project.war', type: 'war']], credentialsId: 'nexus-cred', groupId: 'project', nexusUrl: '18.61.0.214:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'project-artifact', version: '1.0-SNAPSHOT'
clickon generate pipeline syntax , copy the synatax and paste in the pipeline
Global Variables
There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the Global Variables Reference for details.

```
stage('upload artifact in nexus'){
        steps{
           nexusArtifactUploader artifacts: [[artifactId: 'project', classifier: '', file: 'target/project.war', type: 'war']], credentialsId: 'nexus-cred', groupId:
'project', nexusUrl: '18.61.0.214:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'project-artifact', version: '1.0-SNAPSHOT'
        }
     }
```

Now we need to add docker hub credentials in variable format.

Click on pipeline syntax >> search for with Credentials : Bind credentials to variables




Jenkins
Search (CTRL+K)
1
admin
log out
Dashboard
project-job
Pipeline Syntax
Back
Snippet Generator
Declarative Directive Generator
Declarative Online Documentation
Steps Reference
Global Variables Reference
Online Documentation
Examples Reference
IntelliJ IDEA GDSL
Overview
This Snippet Generator will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click Generate Pipeline Script, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving them at default values.)
Steps
Sample Step
withCredentials: Bind credentials to variables
withCredentials
Secret values are masked on a best-effort basis to prevent accidental disclosure. Multiline secrets, such as the contents of a SSH private key file, are not masked. See the inline help for details and usage guidelines.
Bindings
Secret text
Variable
dockerhub
Credentials
docker-cred
Add
Add
Generate Pipeline Script
withCredentials([string(credentialsId: 'docker-cred', variable: 'dockerhub')]) {
// some block
}
Global Variables
There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the Global Variables Reference for details.
Jenkins 2.401.2

Click on Generate pipeline syntax:

```
withCredentials([string(credentialsId: 'docker-cred', variable: 'dockerhub')]) {
   // some block
}
```

- ➔ **Next we need to set Kubernetes deployment**
- ➔ **Create 2 Ubuntu instances**
- ➔ **Master = T3.medium**
- ➔ **Worker = T2.micro**

K8s installation commands:

-> Disable swap & add kernel settings
-> Make sure to run the following commands on all the master & worker nodes.

```
$ sudo swapoff -a
$ sudo sed -i '/ swap / s/^\(.*\)$/#\1/g' /etc/fstab
```

-> Load the following kernel modules on all the master & worker nodes

```
$ sudo tee /etc/modules-load.d/containerd.conf <<EOF
overlay
br_netfilter
EOF

$ sudo modprobe overlay
$ sudo modprobe br_netfilter
```

-> Set the following Kernel parameters for Kubernetes, run beneath tee command

```
$ sudo tee /etc/sysctl.d/kubernetes.conf <<EOF
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
net.ipv4.ip_forward = 1
EOF

$ sudo sysctl --system
```

-> Install containerd run time
-> we are using containerd run time for our Kubernetes cluster. So, to install containerd, first install its dependencies.

```
$ sudo apt install -y curl gnupg software-properties-common apt-transport-https ca-certificates
```

-> Enable docker repository

```
$ sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmour -o /etc/apt/trusted.gpg.d/docker.gpg

$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

-> Now, run following apt command to install containerd

```
$ sudo apt update -y
$ sudo apt install -y containerd.io
```

-> Configure containerd so that it starts using systemd as cgroup.

```
$ containerd config default | sudo tee /etc/containerd/config.toml >/dev/null 2>&1

$ sudo sed -i 's/SystemdCgroup \= false/SystemdCgroup \= true/g' /etc/containerd/config.toml
```

-> Restart and enable containerd service

```
$ sudo systemctl restart containerd
```

$ sudo systemctl enable containerd

-> Add apt repository for Kubernetes
-> Execute following commands to add apt repository for Kubernetes

$ curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

-> Install Kubernetes components Kubectl, kubeadm & kubelet
-> Install Kubernetes components like kubectl, kubelet and Kubeadm utility on all the nodes. Run following set of commands,

$ sudo apt update -y

$ sudo apt install kubeadm kubelet kubectl kubernetes-cni -y

$ sudo apt-mark hold kubeadm kubelet kubectl kubernetes-cni

--
Master node commands only
--
-> Initialize Kubernetes cluster with Kubeadm command
-> Now, we are all set to initialize Kubernetes cluster. Run the following Kubeadm command from the master node only.

$ sudo kubeadm init
$ sudo kubeadm init --control-plane-endpoint=master

-> To start interacting with cluster, run following commands from the master node,

$ mkdir -p $HOME/.kube

$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

$ sudo chown $(id -u):$(id -g) $HOME/.kube/config

-> Now, try to run following kubectl commands to view cluster and node status

$ kubectl cluster-info

$ kubectl get nodes

-> in master node token/key is generated copy that and paste in all worker nodes
ex: below key look like this, plz make sure sudo

```
$ sudo kubeadm join k8smaster.example.net:6443 --token vt4ua6.wcma2y8pl4menxh2 \
  --discovery-token-ca-cert-hash sha256:0494aa7fc6ced8f8e7b20137ec0c5d2699dc5f8e616656932ff9173c94962a36
```

#get worker token to add workers to cluster

-> $ kubeadm token create --print-join-command

-> Check the nodes status from master node using kubectl command,

$ kubectl get nodes

note: As we can see nodes status is 'NotReady', so to make it active. We must install CNI (Container Network Interface) or network add-on plugins like Calico, Flannel and Weave-net.
-> Install Calico Pod Network Add-on
-> Run following curl and kubectl command to install Calico network plugin from the master node,

$ kubectl apply -f https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/calico.yaml

 # to restart kublet service in all manager & worker nodes

-> $ sudo systemctl restart kubelet.service

-> Verify the status of pods in kube-system namespace,

$ kubectl get pods -n kube-system

-> check the nodes status as well.

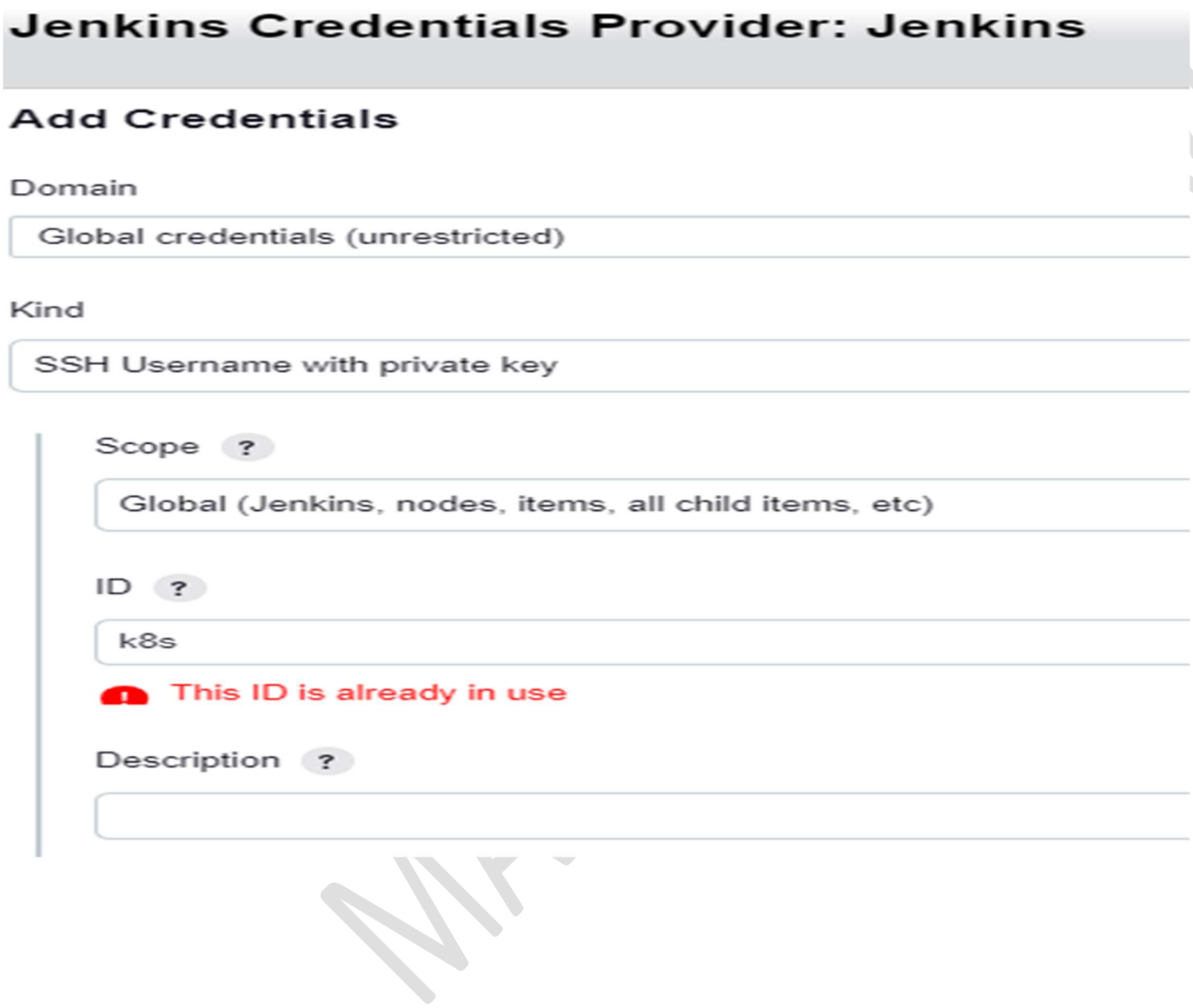
$ kubectl get nodes
$ kubectl get pods

$ kubectl get svc

Now we need to configure the k8s deployment

Go to pipeline syntax generator

Choose : SSh over agent:


Jenkins Credentials Provider: Jenkins
Add Credentials
Domain
Global credentials (unrestricted)
Kind
SSH Username with private key
Scope ?
Global (Jenkins, nodes, items, all child items, etc)
ID ?
k8s
This ID is already in use
Description ?

Username

ubuntu

Treat username as secret ?

Private Key

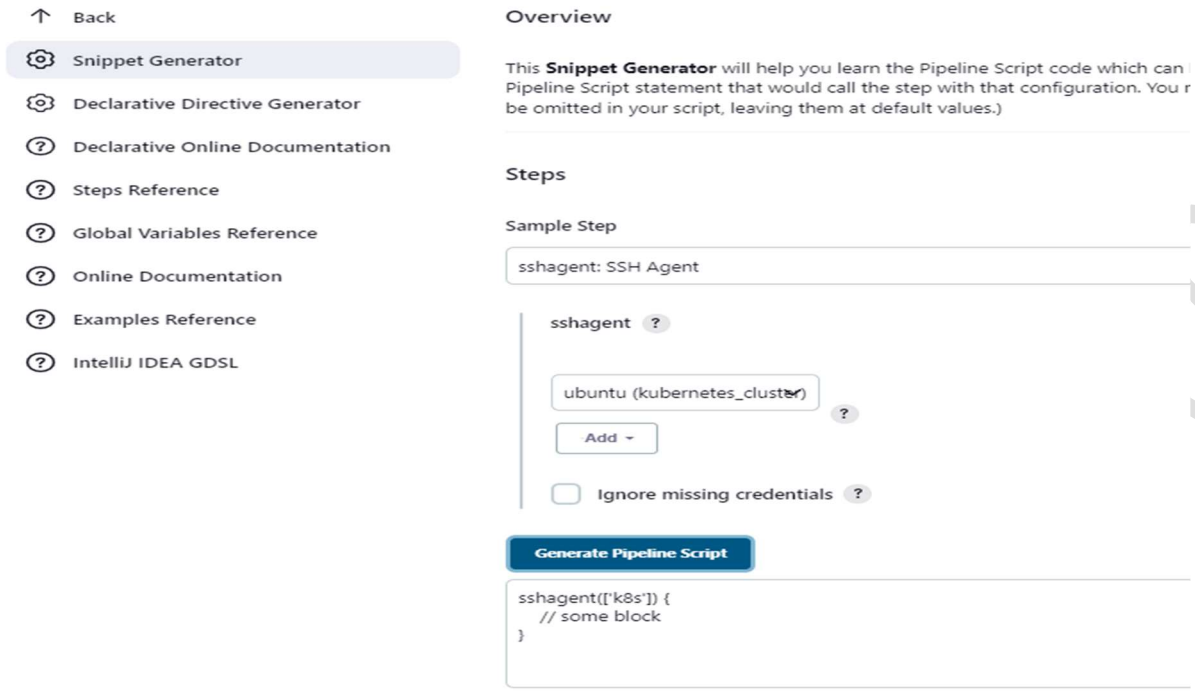
Enter directly

Key

Enter here Ubuntu instance pem.key copy and paste here

Passphrase

Add Cancel



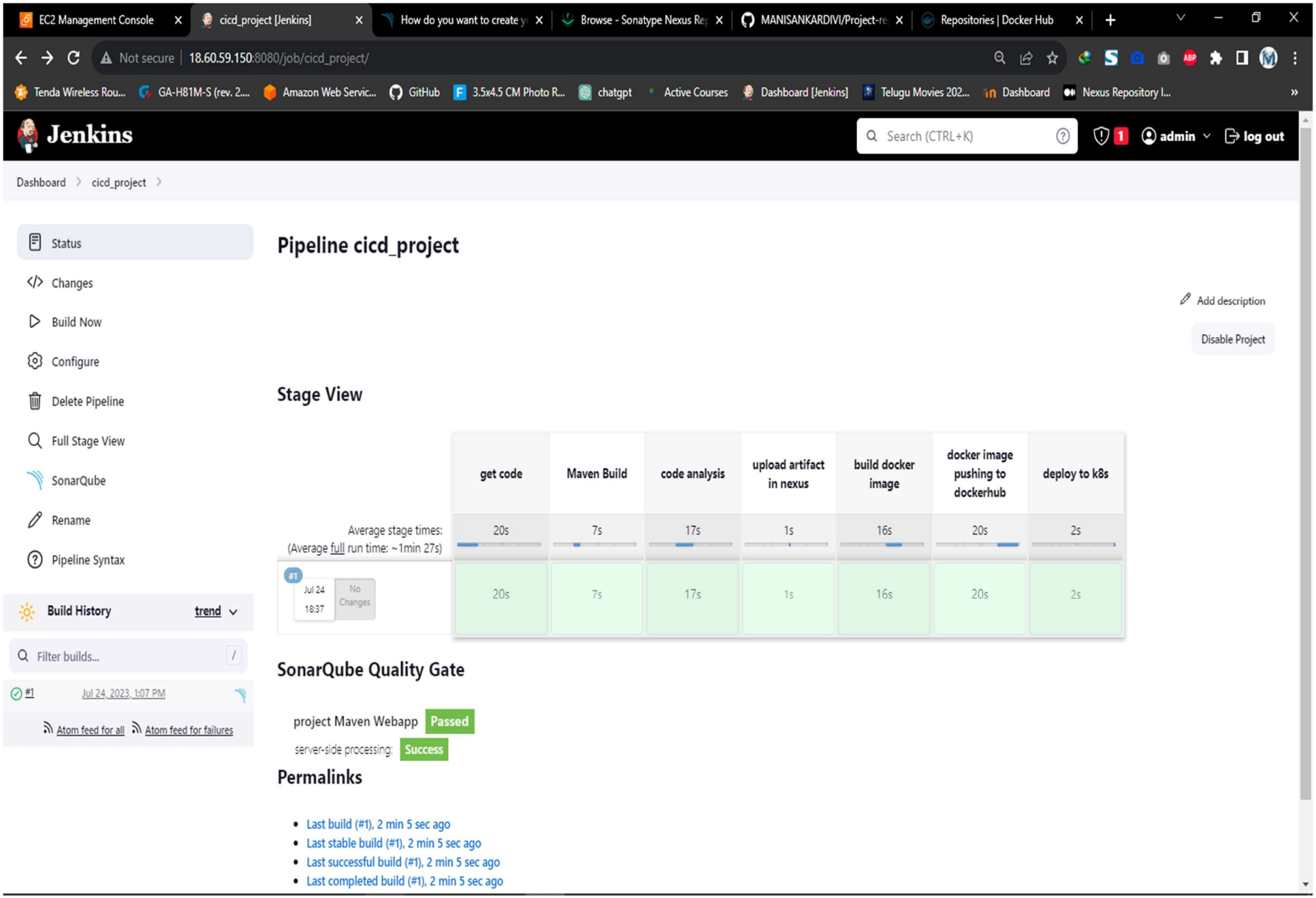
Copy the Generated syntax and we need to add some more line:

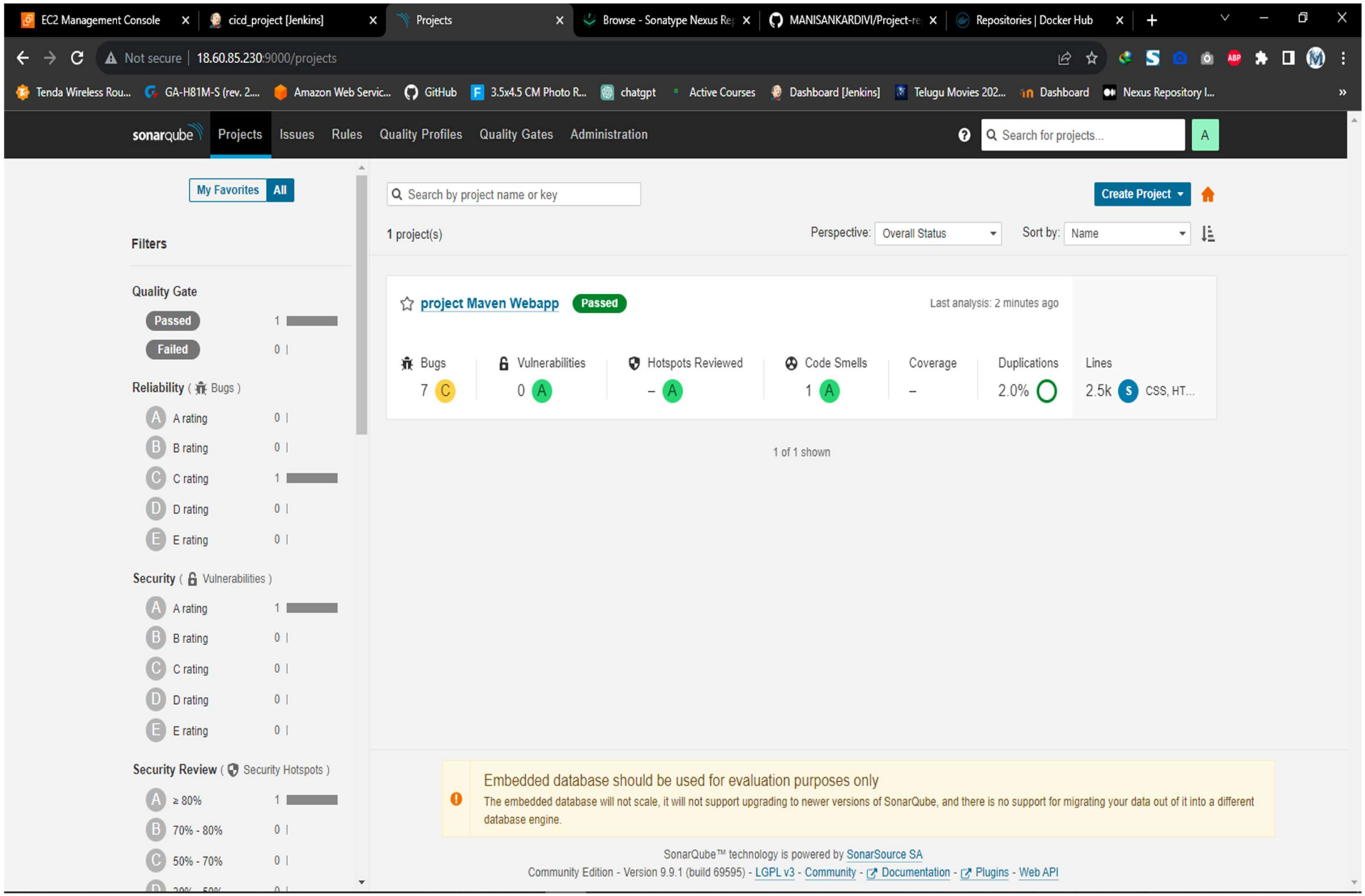
```
stage('deploy to k8s'){
        steps{
           sshagent(['k8s']) {
              sh 'scp -o StrictHostkeyChecking=no deploymentservice.yaml ubuntu@master PrivateIP:/home/ubuntu'
              sh 'ssh ubuntu@master privateIP kubectl apply -f deploymentservice.yaml'
           }
        }
     }
```

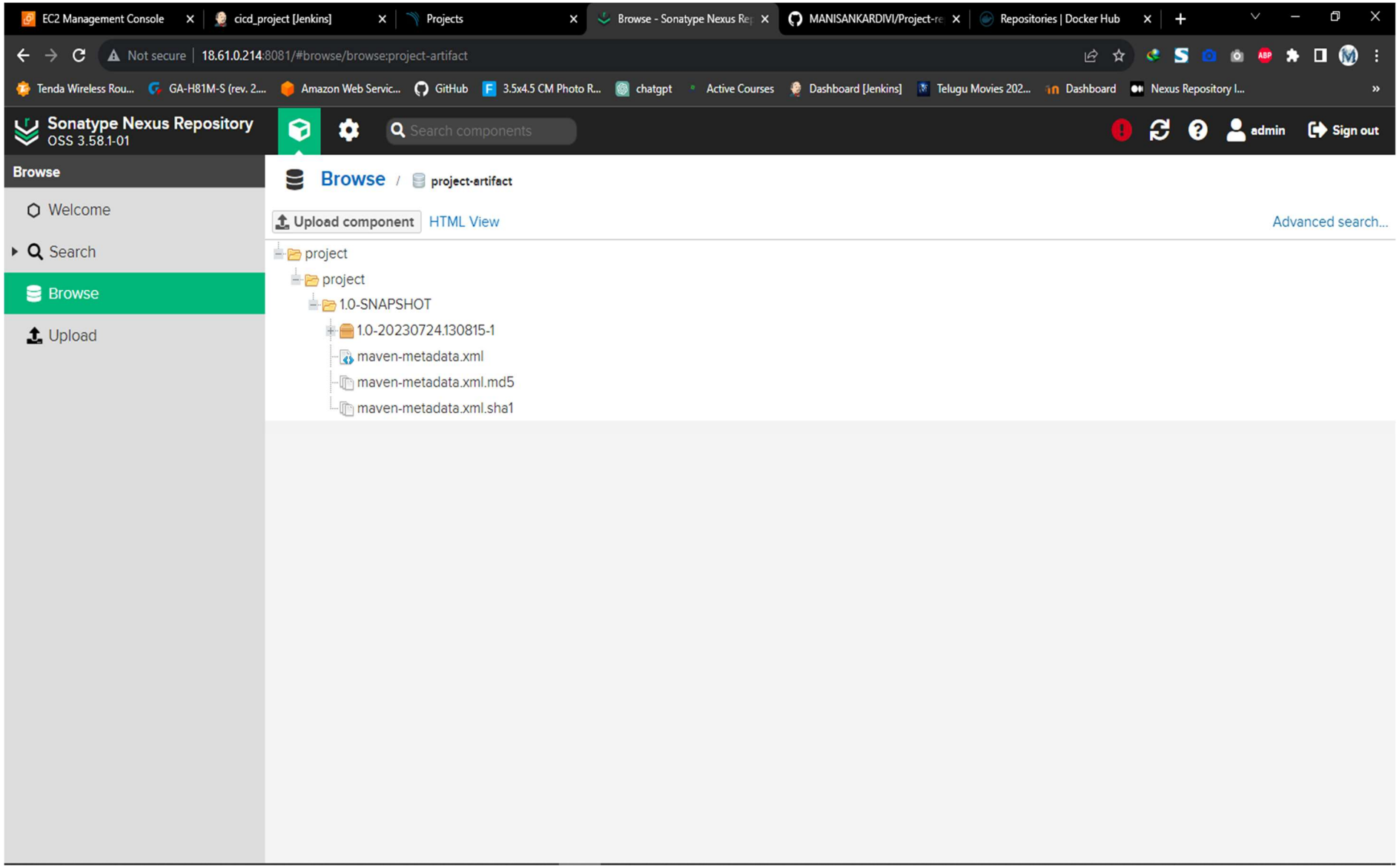
Jenkins Pipeline

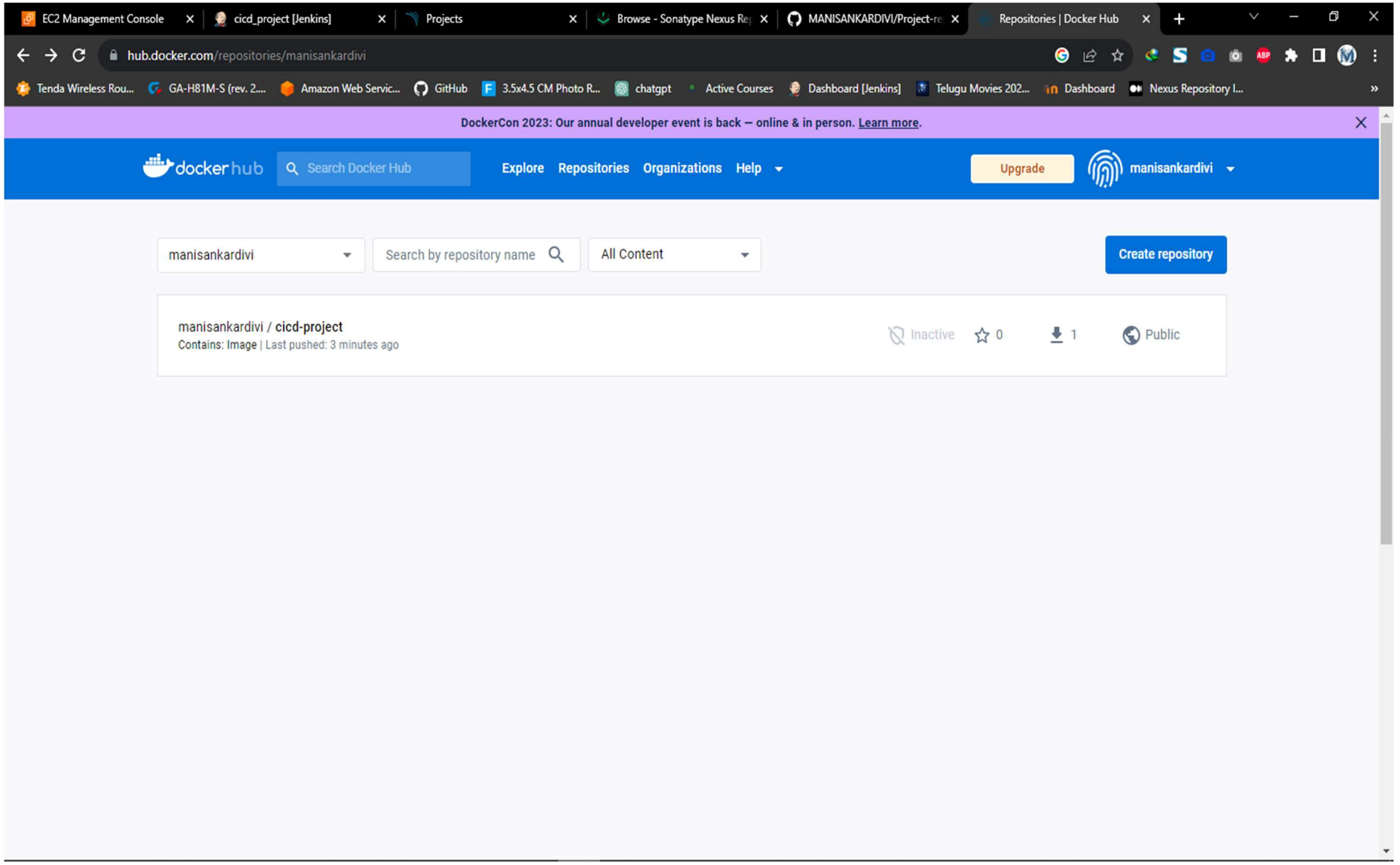
```
pipeline{
   agent any
   environment {
      PATH = "$PATH:/opt/maven/bin"
   }
   stages{
      stage('get code'){
         steps{
            git branch: 'main', url: 'https://github.com/MANISANKARDIVI/test-repo.git'
         }
      }
      stage('Maven Build'){
         steps{
            sh 'mvn clean package'
         }
      }
      stage('code analysis'){
         steps{
            withSonarQubeEnv('sonarqube'){
               sh "mvn sonar:sonar"
            }
         }
      }
      stage('upload artifact in nexus'){
         steps{
            nexusArtifactUploader artifacts: [[artifactId: 'project', classifier: '', file: 'target/project.war', type: 'war']], credentialsId: 'nexus-cred', groupId: 'project',
nexusUrl: '18.61.0.214:8081', nexusVersion: 'nexus3', protocol: 'http', repository: 'project-artifact', version: '1.0-SNAPSHOT'
         }
      }
      stage('build docker image'){
         steps{
            sh 'docker build -t cicd-project .'
         }
      }
      stage('docker image pushing to dockerhub'){
```

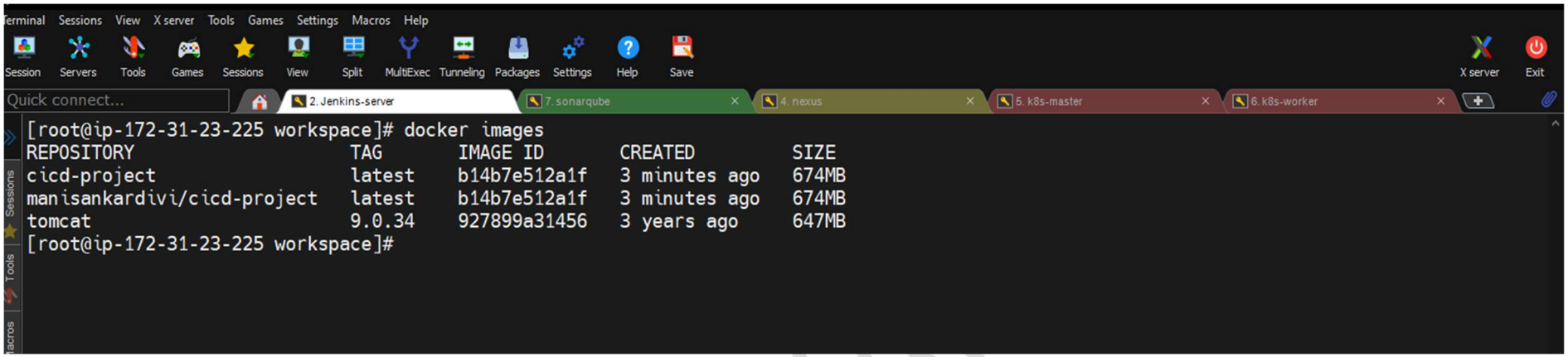
```
        steps{
            withCredentials([string(credentialsId: 'docker-cred', variable: 'dockerhub')]) {
                sh 'docker login -u manisankardivi -p ${dockerhub}'
                sh 'docker tag cicd-project manisankardivi/cicd-project'
                sh 'docker push manisankardivi/cicd-project'
            }
        }
    }
    stage('deploy to k8s'){
        steps{
            sshagent(['k8s']) {
                sh 'scp -o StrictHostkeyChecking=no deploymentservice.yaml ubuntu@172.31.24.66:/home/ubuntu'
                sh 'ssh ubuntu@172.31.24.66 kubectl apply -f deploymentservice.yaml'
            }
        }
    }
  }
}
```

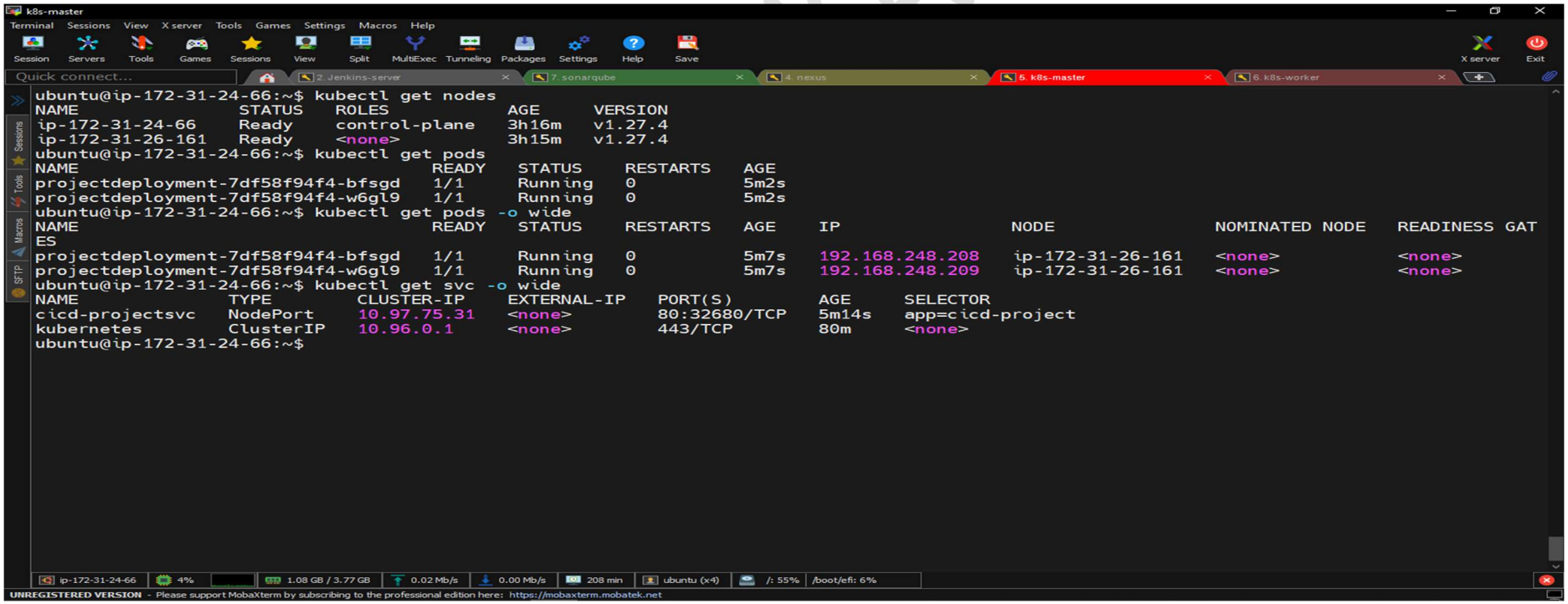

EC2 Management Console
cicd_project [Jenkins]
How do you want to create y
Browse - Sonatype Nexus Re
MANISANKARDIVI/Project-re
Repositories | Docker Hub
Not secure
18.60.59.150:8080/job/cicd_project/
Tenda Wireless Rou...
GA-H81M-S (rev. 2....
Amazon Web Servic...
GitHub
3.5x4.5 CM Photo R...
chatgpt
Active Courses
Dashboard [Jenkins]
Telugu Movies 202...
Dashboard
Nexus Repository I...
Jenkins
Search (CTRL+K)
admin
log out
Dashboard
cicd_project
Status
Changes
Build Now
Configure
Delete Pipeline
Full Stage View
SonarQube
Rename
Pipeline Syntax
Build History
trend
Filter builds...
#1
Jul 24, 2023, 1:07 PM
Atom feed for all
Atom feed for failures
Pipeline cicd_project
Add description
Disable Project
Stage View
get code
Maven Build
code analysis
upload artifact in nexus
build docker image
docker image pushing to dockerhub
deploy to k8s
Average stage times:
(Average full run time: ~1min 27s)
20s
7s
17s
1s
16s
20s
2s
#1
Jul 24
18:37
No Changes
SonarQube Quality Gate
project Maven Webapp
Passed
server-side processing:
Success
Permalinks
Last build (#1), 2 min 5 sec ago
Last stable build (#1), 2 min 5 sec ago
Last successful build (#1), 2 min 5 sec ago
Last completed build (#1), 2 min 5 sec ago


EC2 Management Console
cicd_project [Jenkins]
Projects
Browse - Sonatype Nexus Re
MANISANKARDIVI/Project-re
Repositories | Docker Hub
Not secure
18.60.85.230:9000/projects
Tenda Wireless Rou...
GA-H81M-S (rev. 2....
Amazon Web Servic...
GitHub
3.5x4.5 CM Photo R...
chatgpt
Active Courses
Dashboard [Jenkins]
Telugu Movies 202...
Dashboard
Nexus Repository I...
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
Search for projects...
A
My Favorites
All
Filters
Quality Gate
Passed 1
Failed 0
Reliability (Bugs)
A A rating 0
B B rating 0
C C rating 1
D D rating 0
E E rating 0
Security (Vulnerabilities)
A A rating 1
B B rating 0
C C rating 0
D D rating 0
E E rating 0
Security Review (Security Hotspots)
A ≥ 80% 1
B 70% - 80% 0
C 50% - 70% 0
Search by project name or key
Create Project
1 project(s)
Perspective: Overall Status
Sort by: Name
project Maven Webapp
Passed
Last analysis: 2 minutes ago
Bugs 7 C
Vulnerabilities 0 A
Hotspots Reviewed – A
Code Smells 1 A
Coverage –
Duplications 2.0%
Lines 2.5k S CSS, HT...
1 of 1 shown
Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.
SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 9.9.1 (build 69595) - LGPL v3 - Community - Documentation - Plugins - Web API


EC2 Management Console
cicd_project [Jenkins]
Projects
Browse - Sonatype Nexus Re
MANISANKARDIVI/Project-re
Repositories | Docker Hub
Not secure
18.61.0.214:8081/#browse/browse:project-artifact
Tenda Wireless Rou...
GA-H81M-S (rev. 2....
Amazon Web Servic...
GitHub
3.5x4.5 CM Photo R...
chatgpt
Active Courses
Dashboard [Jenkins]
Telugu Movies 202...
Dashboard
Nexus Repository I...
Sonatype Nexus Repository
OSS 3.58.1-01
Search components
admin
Sign out
Browse
Welcome
Search
Browse
Upload
Browse / project-artifact
Upload component
HTML View
Advanced search...
project
project
1.0-SNAPSHOT
1.0-20230724.130815-1
maven-metadata.xml
maven-metadata.xml.md5
maven-metadata.xml.sha1


EC2 Management Console
cicd_project [Jenkins]
Projects
Browse - Sonatype Nexus Re
MANISANKARDIVI/Project-re
Repositories | Docker Hub
hub.docker.com/repositories/manisankardivi
Tenda Wireless Rou...
GA-H81M-S (rev. 2....
Amazon Web Servic...
GitHub
3.5x4.5 CM Photo R...
chatgpt
Active Courses
Dashboard [Jenkins]
Telugu Movies 202...
Dashboard
Nexus Repository I...
DockerCon 2023: Our annual developer event is back — online & in person. Learn more.
docker hub
Search Docker Hub
Explore
Repositories
Organizations
Help
Upgrade
manisankardivi
manisankardivi
Search by repository name
All Content
Create repository
manisankardivi / cicd-project
Contains: Image | Last pushed: 3 minutes ago
Inactive
0
1
Public


Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help Save
X server Exit
Quick connect...
2. Jenkins-server
7. sonarqube
4. nexus
5. k8s-master
6. k8s-worker
[root@ip-172-31-23-225 workspace]# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
cicd-project latest b14b7e512a1f 3 minutes ago 674MB
manisankardivi/cicd-project latest b14b7e512a1f 3 minutes ago 674MB
tomcat 9.0.34 927899a31456 3 years ago 647MB
[root@ip-172-31-23-225 workspace]#


k8s-master
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help Save
X server Exit
Quick connect...
2. Jenkins-server
7. sonarqube
4. nexus
5. k8s-master
6. k8s-worker
ubuntu@ip-172-31-24-66:~$ kubectl get nodes
NAME STATUS ROLES AGE VERSION
ip-172-31-24-66 Ready control-plane 3h16m v1.27.4
ip-172-31-26-161 Ready <none> 3h15m v1.27.4
ubuntu@ip-172-31-24-66:~$ kubectl get pods
NAME READY STATUS RESTARTS AGE
projectdeployment-7df58f94f4-bfsgd 1/1 Running 0 5m2s
projectdeployment-7df58f94f4-w6gl9 1/1 Running 0 5m2s
ubuntu@ip-172-31-24-66:~$ kubectl get pods -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
projectdeployment-7df58f94f4-bfsgd 1/1 Running 0 5m7s 192.168.248.208 ip-172-31-26-161 <none> <none>
projectdeployment-7df58f94f4-w6gl9 1/1 Running 0 5m7s 192.168.248.209 ip-172-31-26-161 <none> <none>
ubuntu@ip-172-31-24-66:~$ kubectl get svc -o wide
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE SELECTOR
cicd-projectsvc NodePort 10.97.75.31 <none> 80:32680/TCP 5m14s app=cicd-project
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 80m <none>
ubuntu@ip-172-31-24-66:~$
ip-172-31-24-66 4% 1.08 GB / 3.77 GB 0.02 Mb/s 0.00 Mb/s 208 min ubuntu (x4) /: 55% /boot/efi: 6%
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

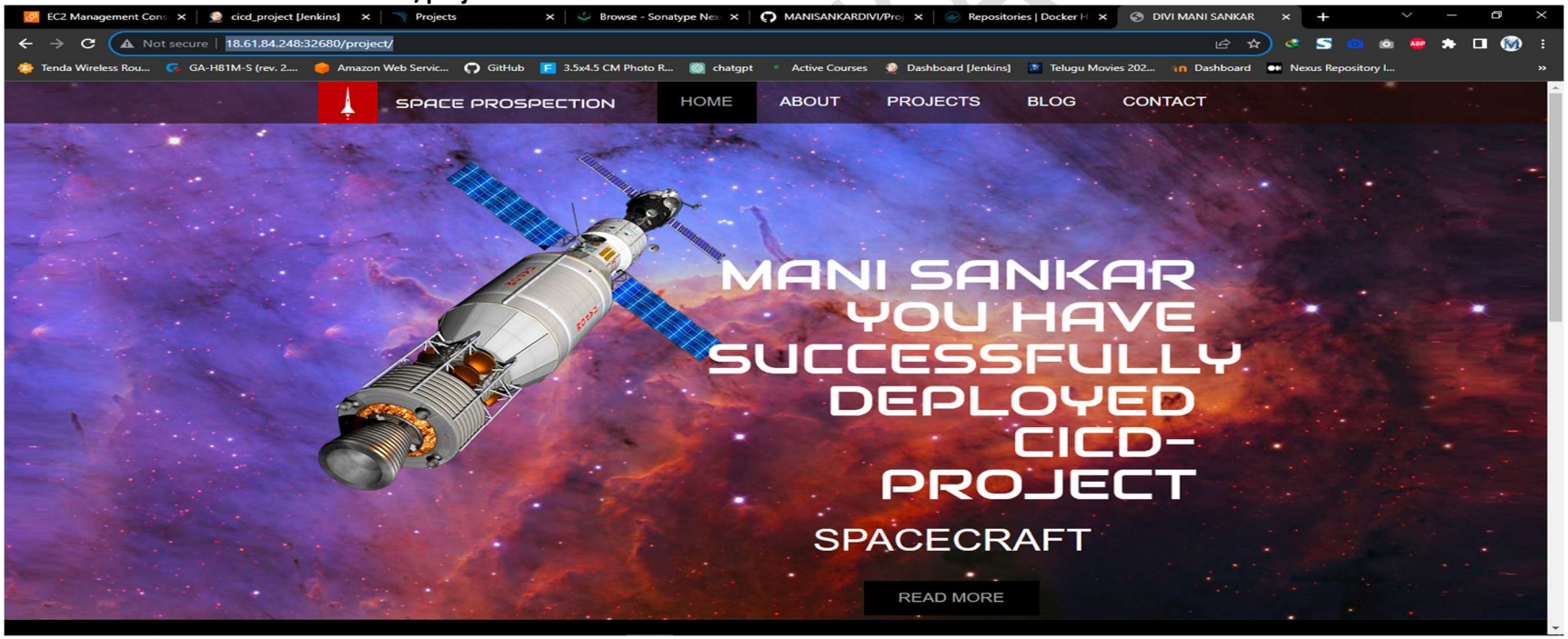
In Master k8s-server
$ kubectl get pods
$ kubectl get nodes
$ kubectl get svc
$ kubectl svc -o wide

➔ **To delete all nodes pods services use below command**

$ kubectl delete all --all

```
ubuntu@ip-172-31-24-66:~$ kubectl get svc -o wide
NAME             TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE     SELECTOR
cicd-projectsvc  NodePort    10.97.75.31    <none>        80:32680/TCP   5m14s   app=cicd-project
kubernetes       ClusterIP   10.96.0.1      <none>        443/TCP        80m     <none>
ubuntu@ip-172-31-24-66:~$
```

Copy the cicd-project Port no= 80:32680/TCP
Copy the PublicIP of Master or worker and paste in browser
Paste in Browser = 18.61.84.248:32680/project



THE END