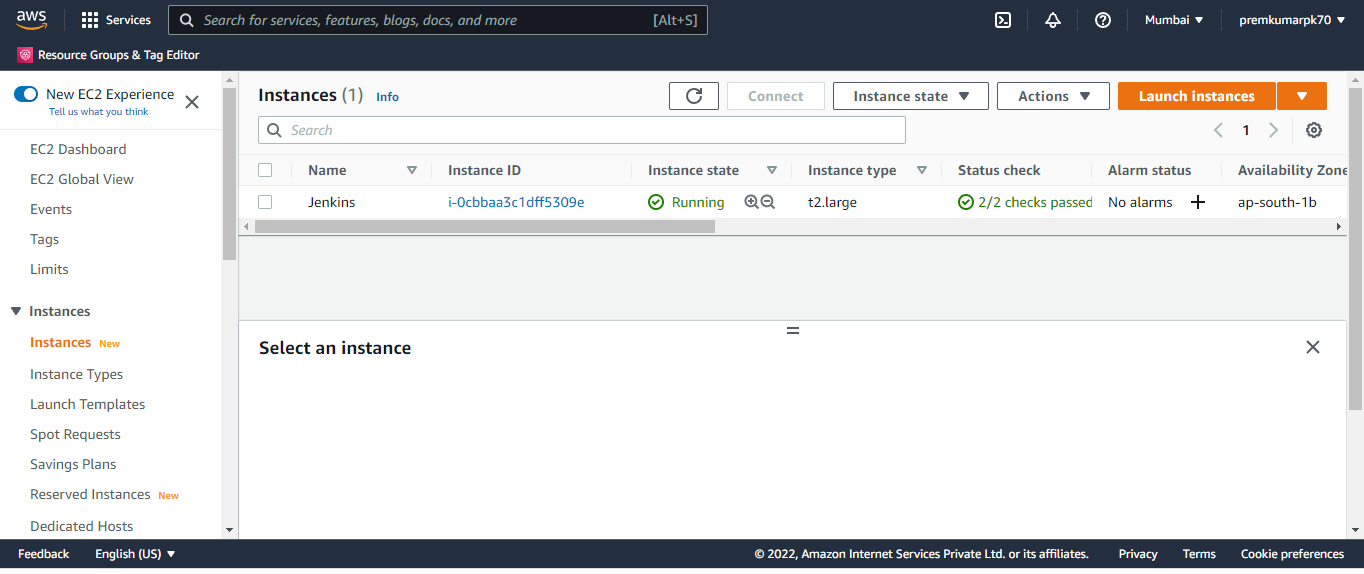
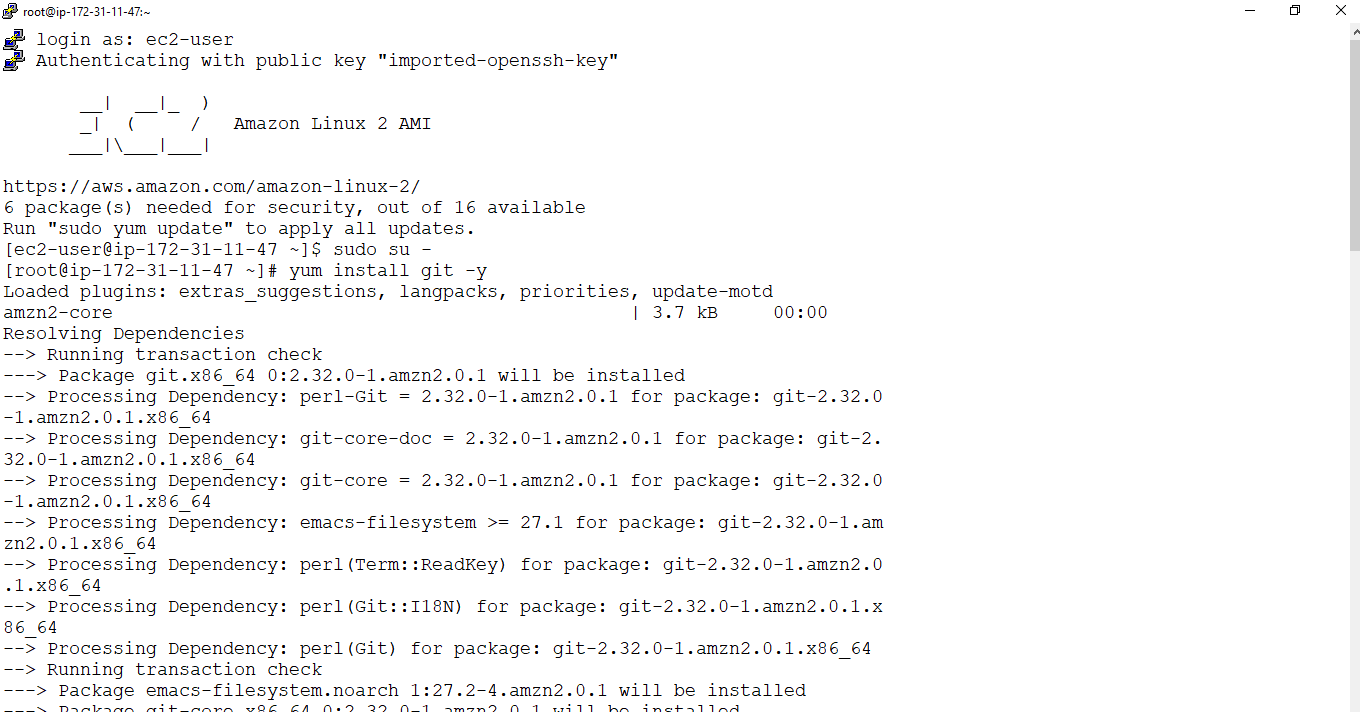
**Devops RealTime Project**

**VM created for Jenkins**



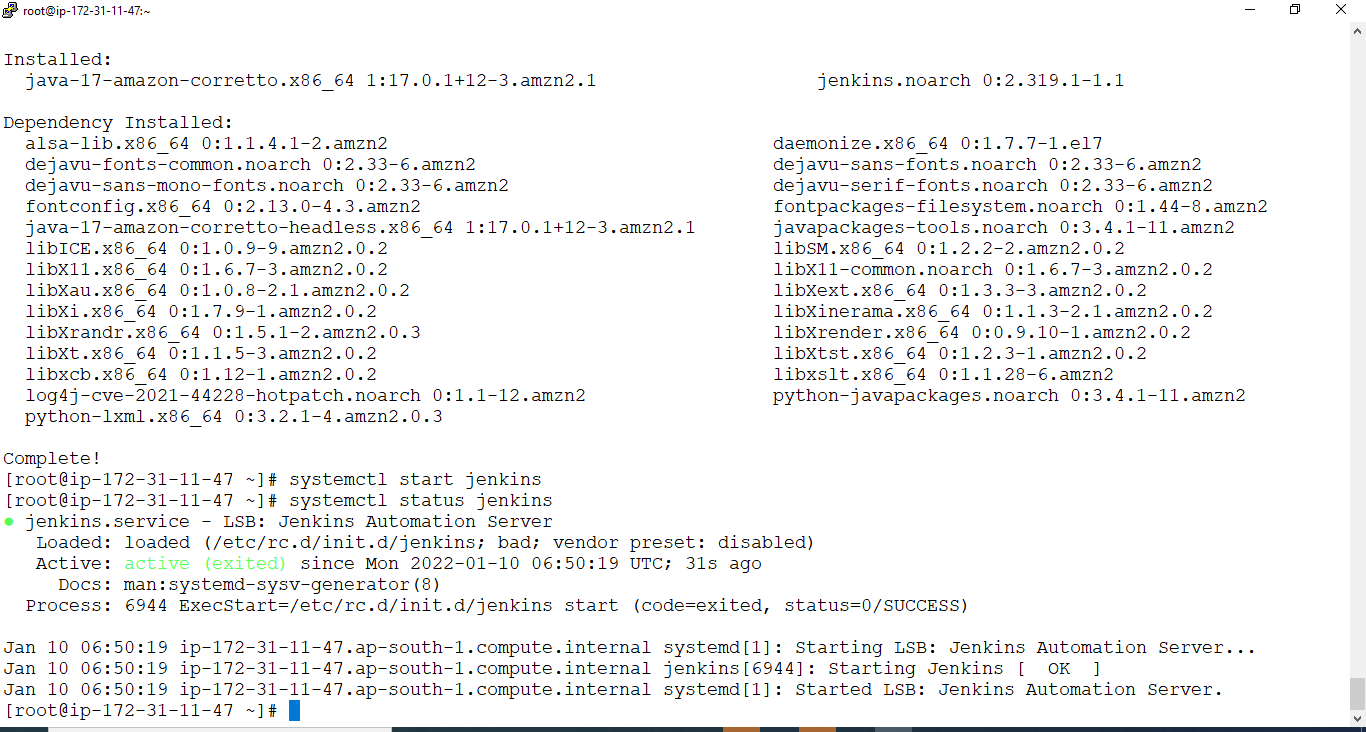
**After I l install git in Amazon Linux using putty in root user**



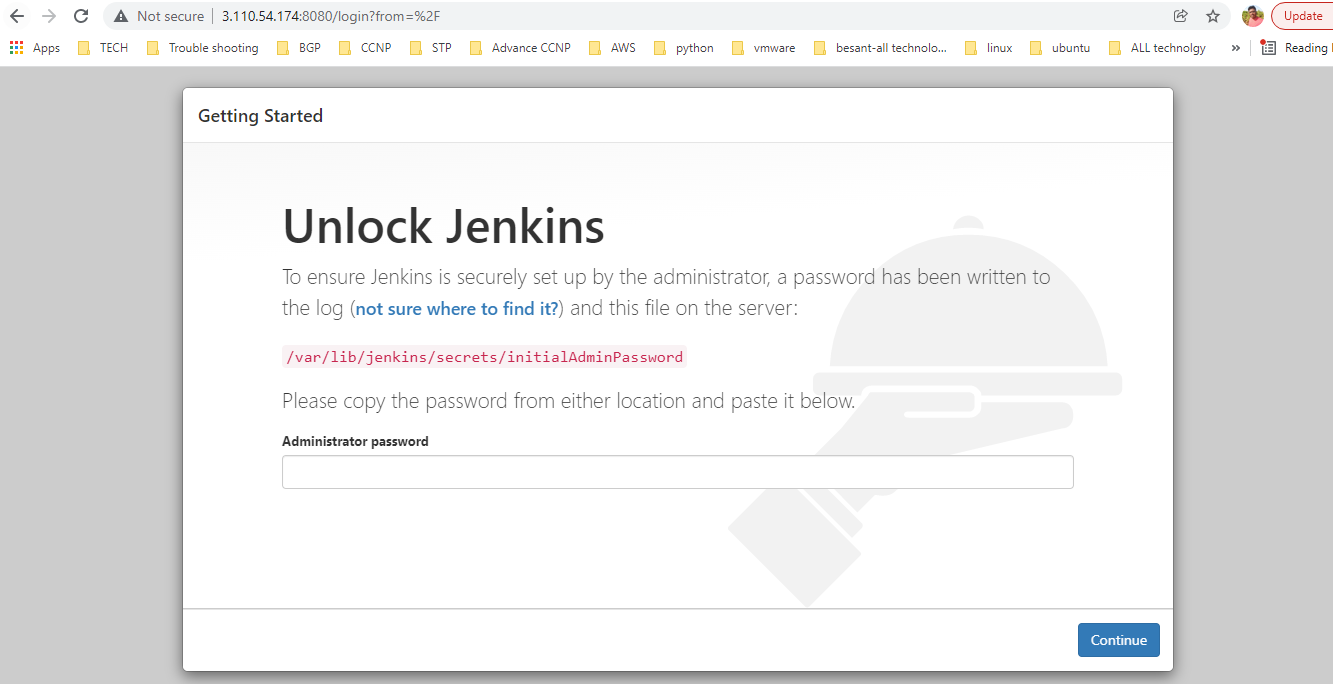
**Jenkins installation command run in putty as root user**



**Jenkins Start or not just verify**

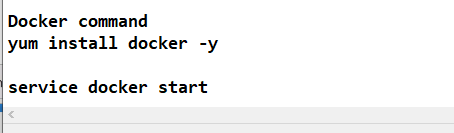


**Go to browser publicip:8080 enter it you get jenkin page open**

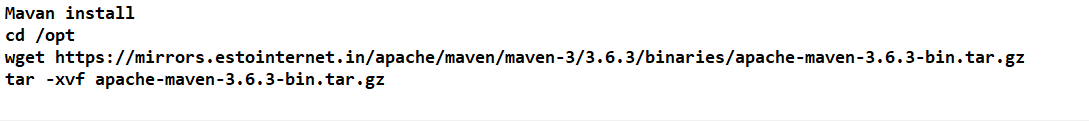


**Above command paste in putty as root user and get pass to paste above admin password then it show it two option select first select .Finally start installation in web console**

**Docker install and start service**



**Mavean Install run in putty as root user**

**.Mavean code get fork to my GIT account**

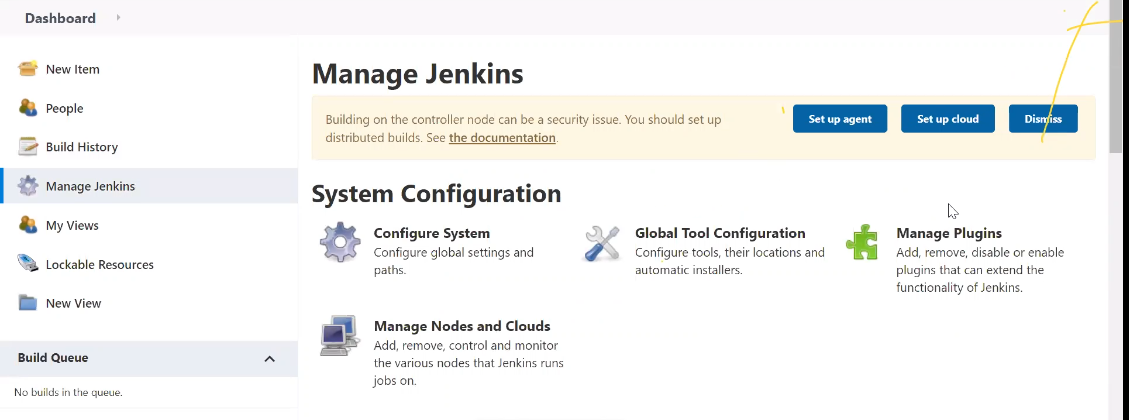
[https://github.com/damodaranj/my-app---> go](https://github.com/damodaranj/my-app--->go) browser click fork

It moves to your git hub account see below



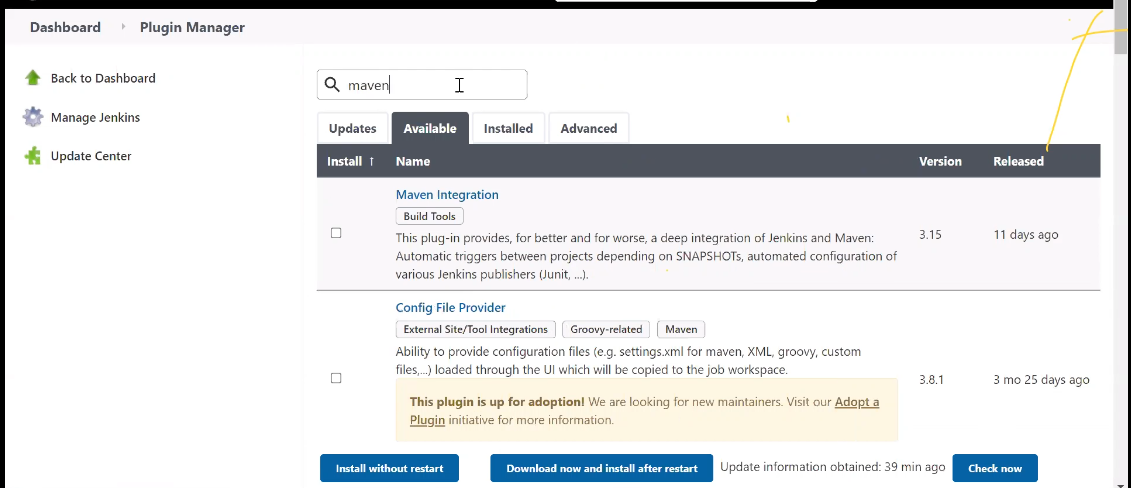
**Maven plugin install in Jenkins**

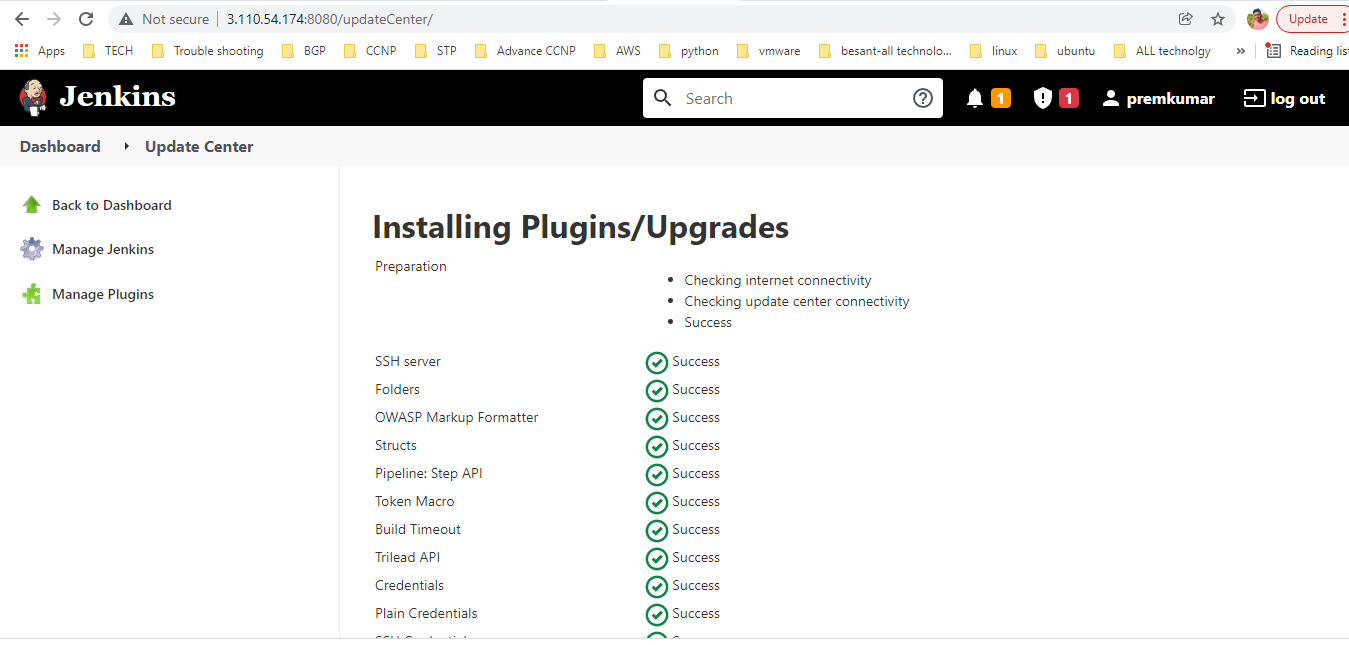
**Manage Jenkins--->Manage Plugin-->click**



**Go to Available Tab--->search Maven**

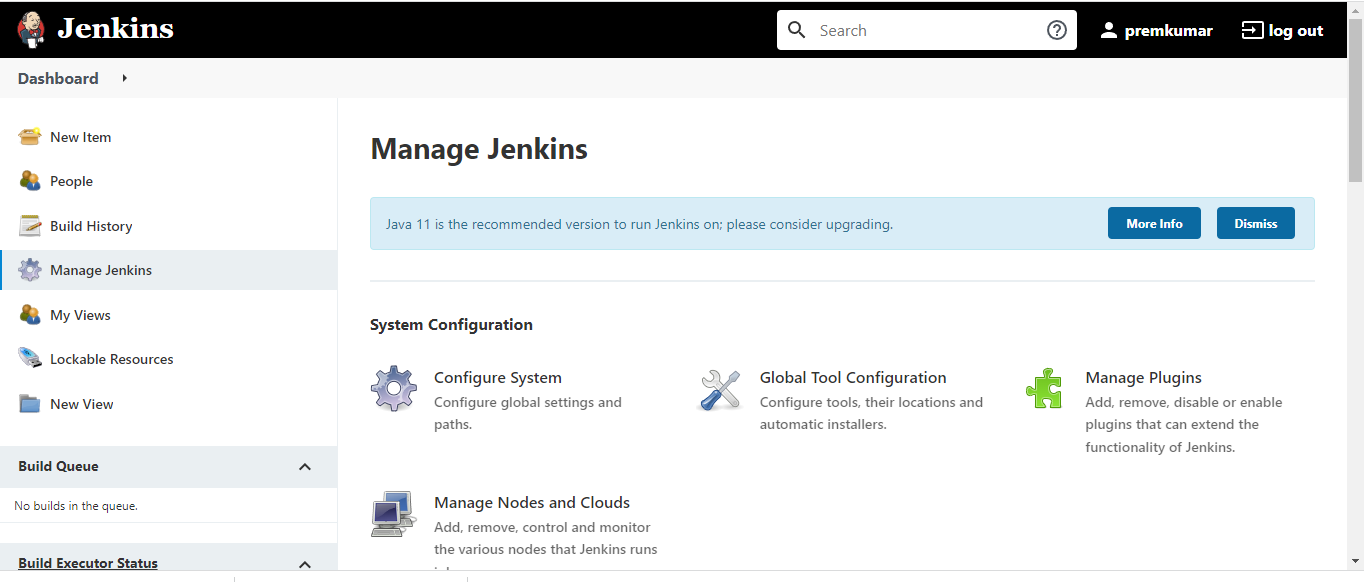
**Maven Integration click--->install without restart**

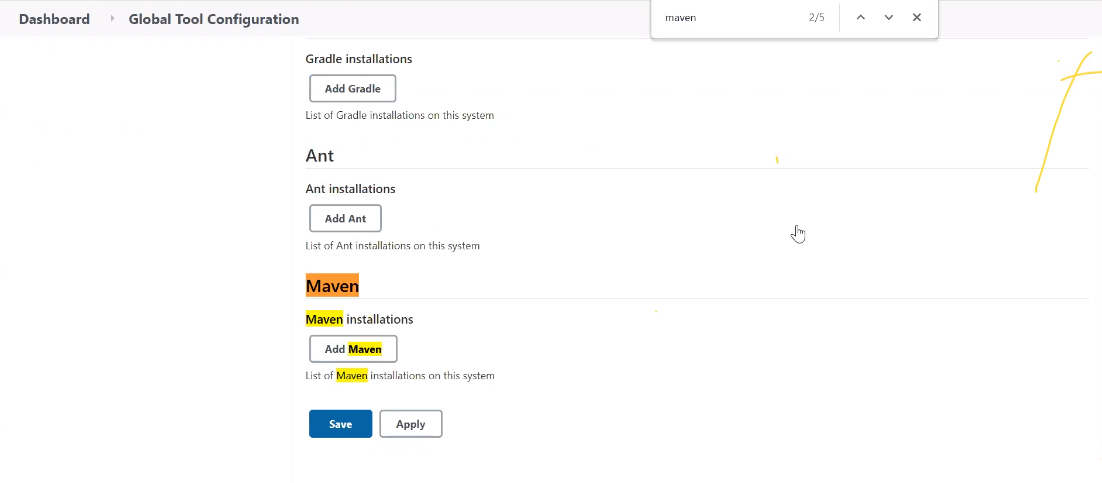


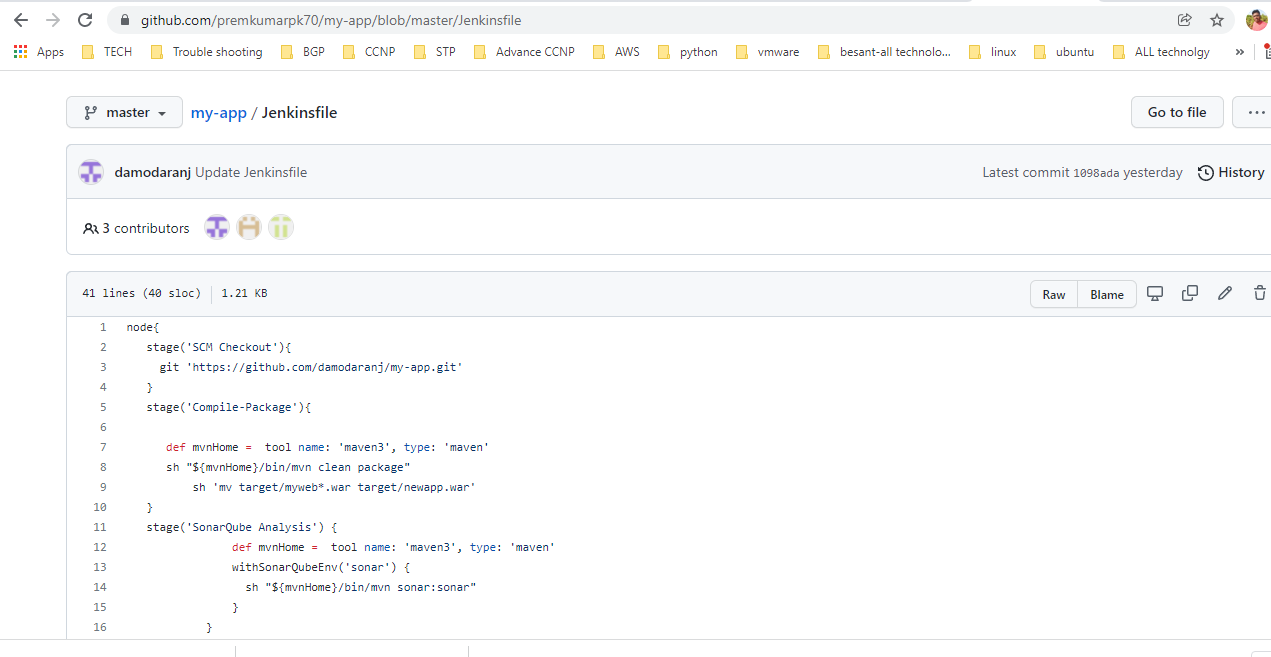


**Maven just inform where location to Jenkins**

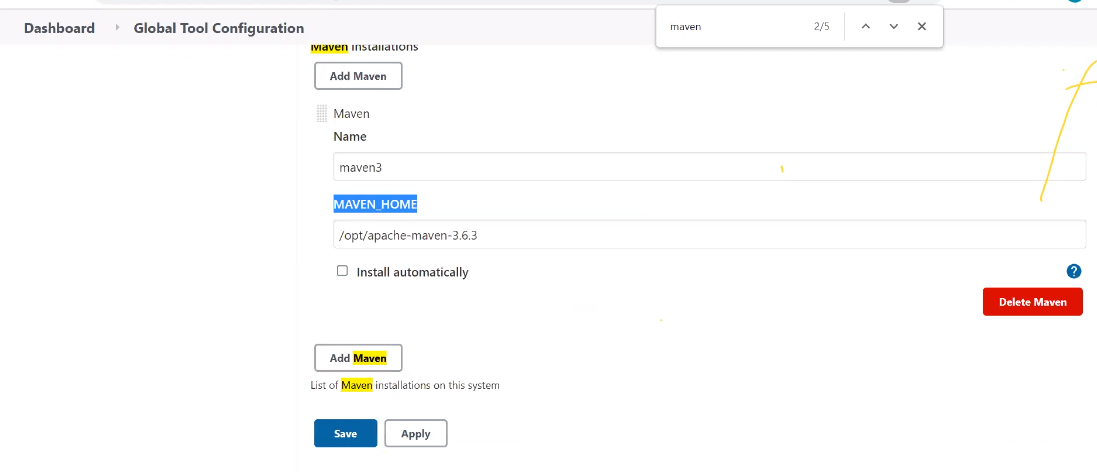
**Manage Jenkins---->Global Tool configuartion--->Ctrl + F--->search Maven**







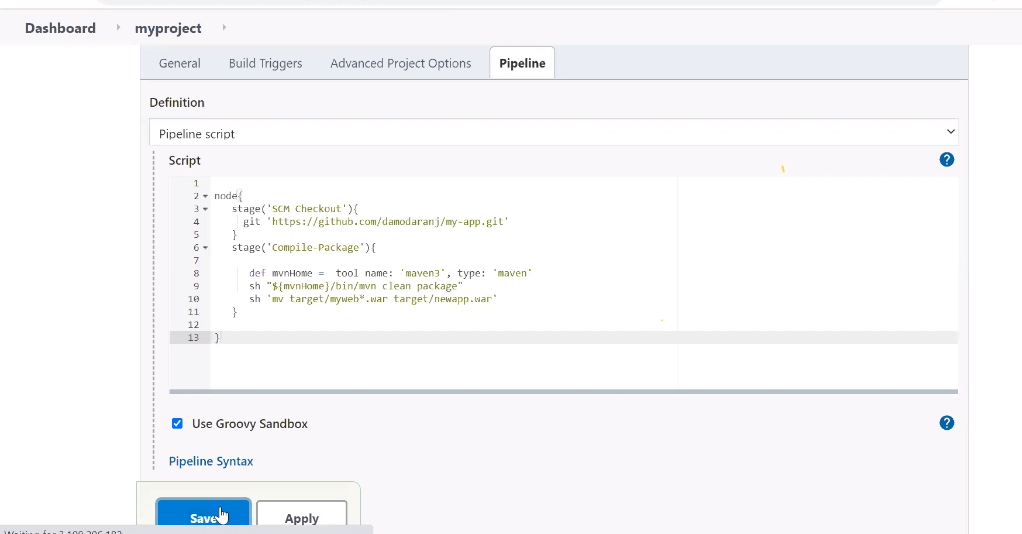
**Jenkins file--->maven3 that is alais name this is you want to give and maven location mention finally saved**

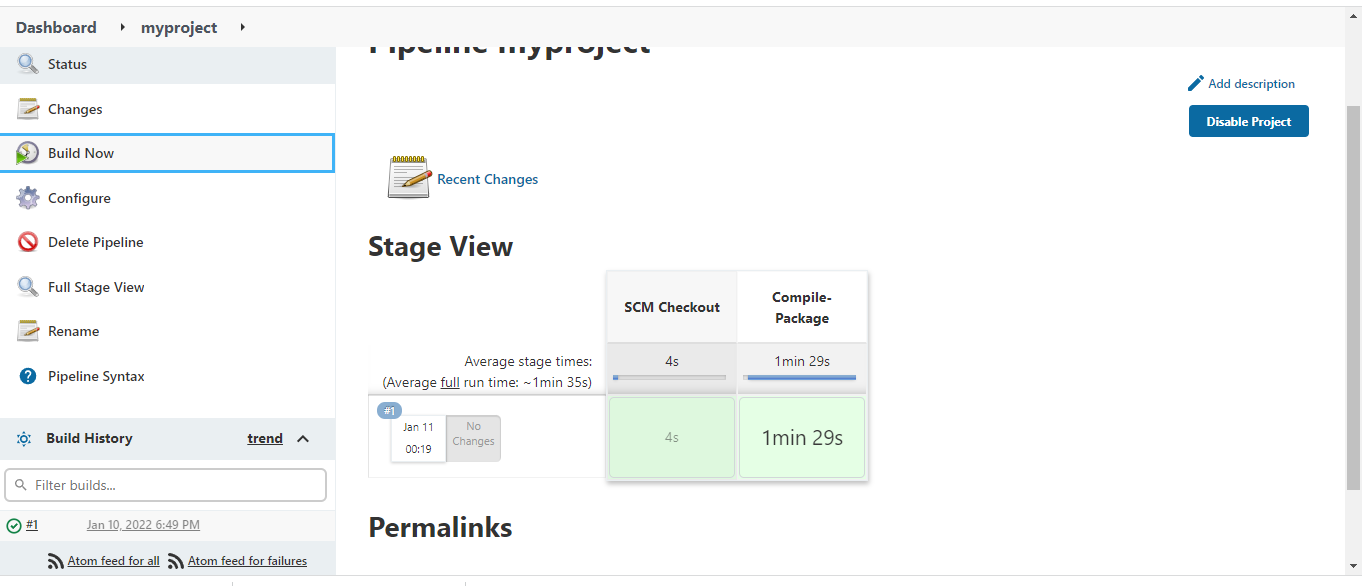


**Groovy script run so we create project --->New item--->select pipeline--->myproject name specfied--->ok**

**Go to Pipeline Tab-->paste script--->save --->Build now**

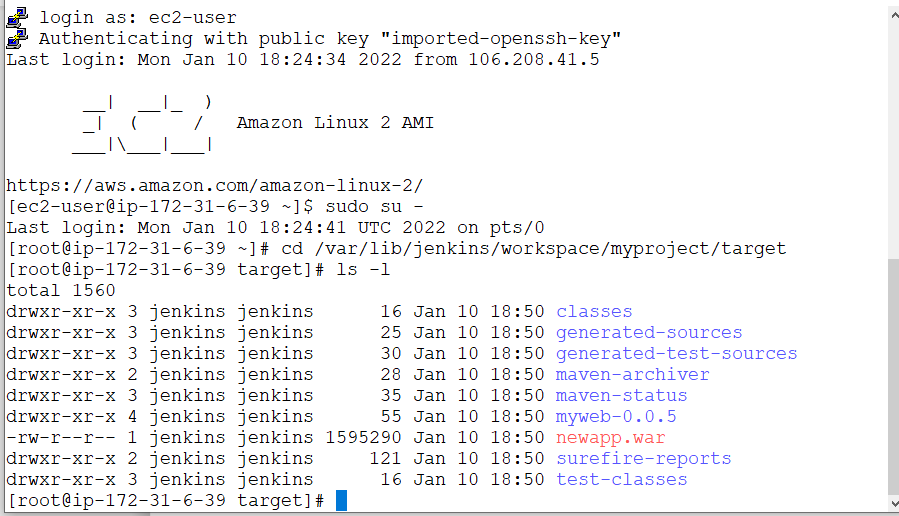






**Click green color it means sucess--->go console output-->ctrl+f--->search target--->war file build**





**First Github sigup--->login completed**

**Below I pass the github password varible --->>Manage Jenkins--->Manage credential--->Jenkins--->Global credential--->add credential**

**Below click secret text---->id- dockerpass,pass - my pass of dockerhub**

**Click ok**

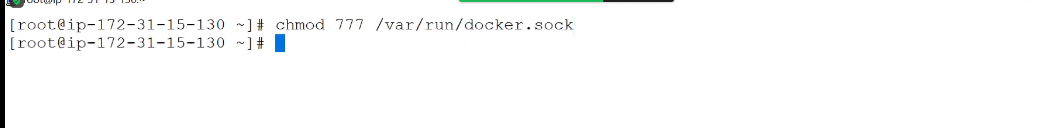
**Now paste groovy script in pipeline--->buildnow**

**It get error because file permission check from console output**

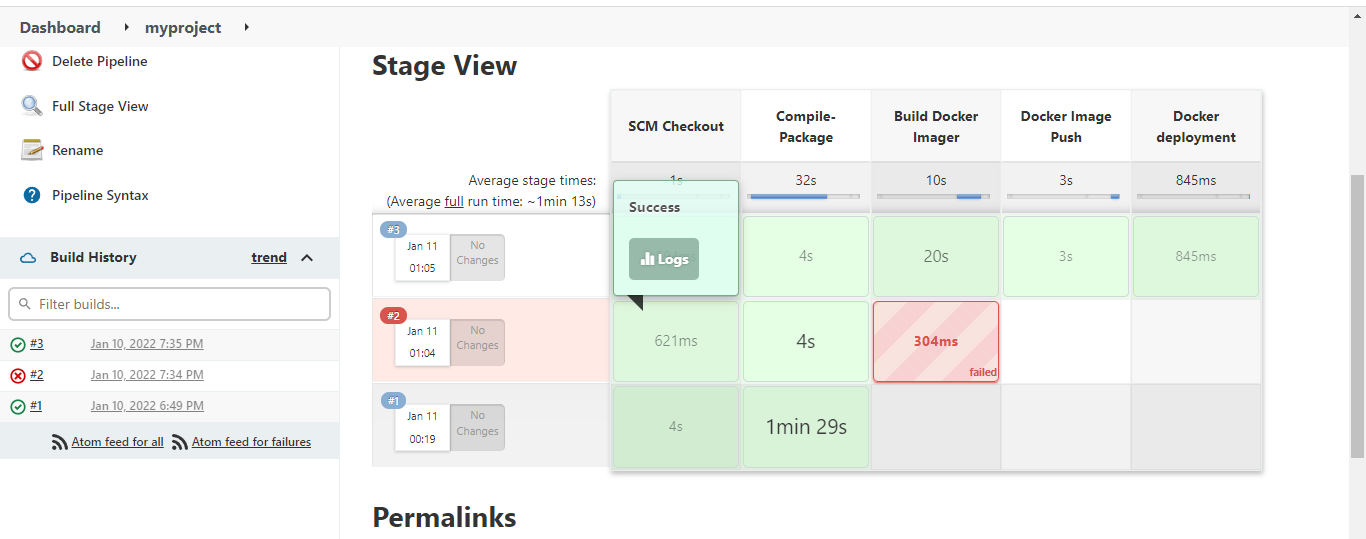




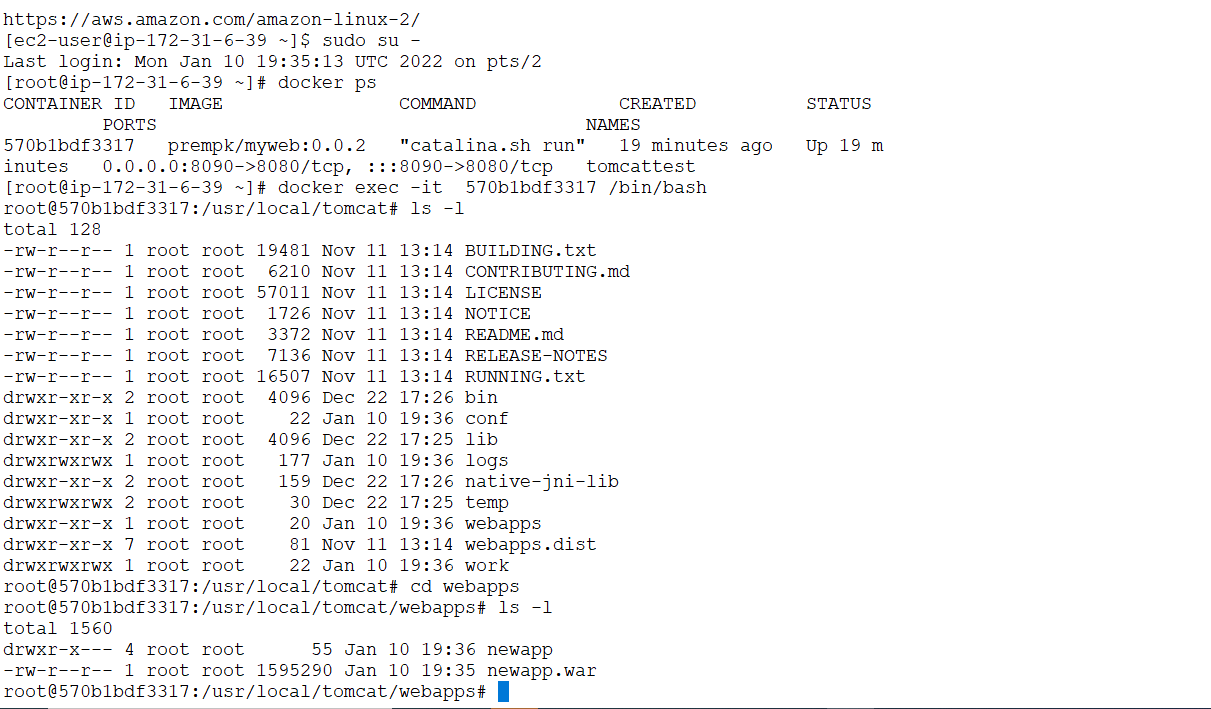




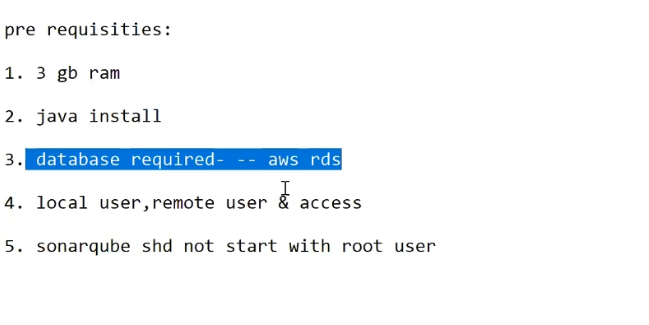
**Finally Buidnow click --->code sucessful deployment**

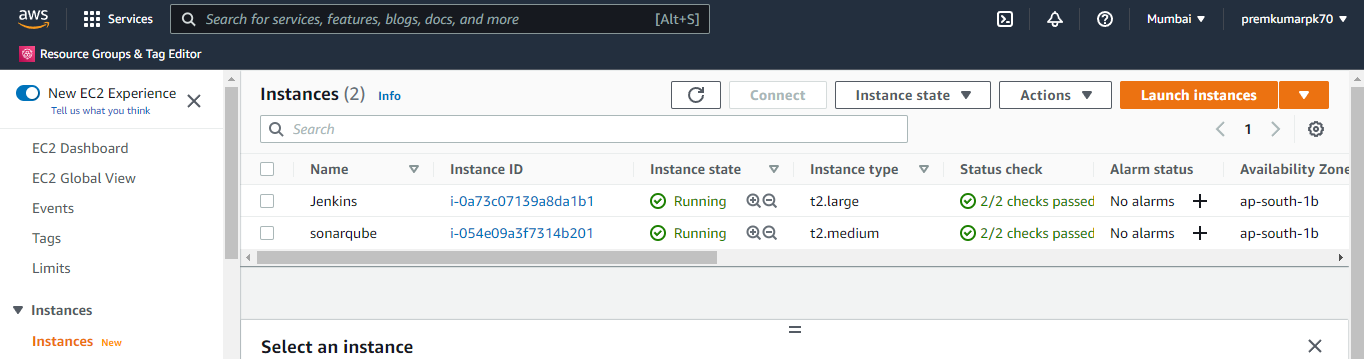






**Sonarqube installation**





**RDS-->crrate database-->mysql--->5.7.26-->free tier--->database name-->mysonar**

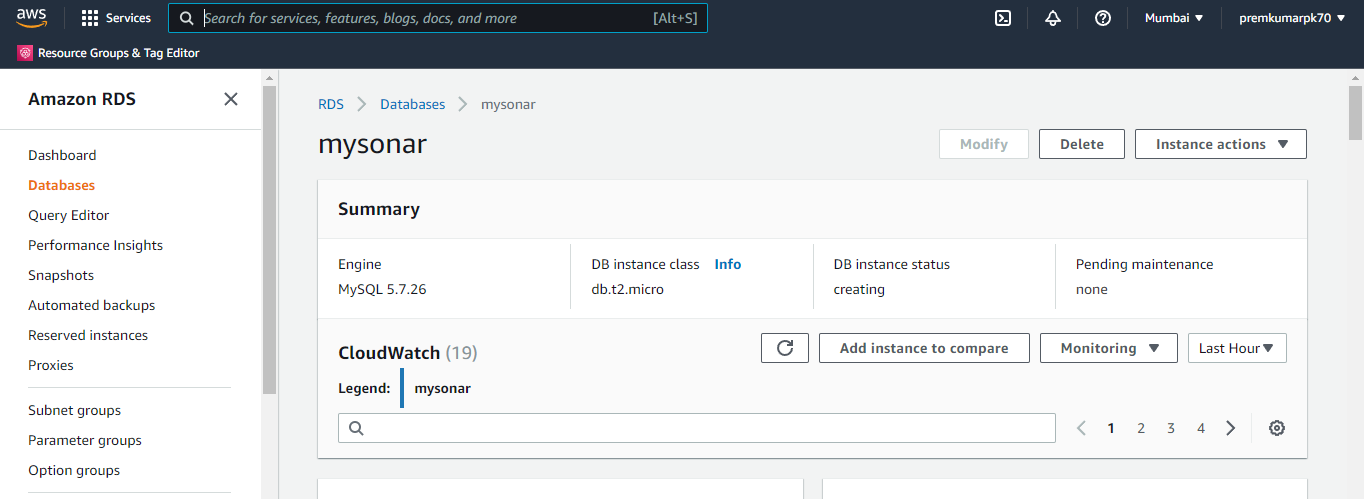
**Master -->admin**

**Password--->admin123**

**Security-->default,openall**

**Disable delete protection**

**Finally create database**





**Cd /opt**

**Yum install mysql -y run as root**

**mysql -h mysonar.cx9yvwbbtwjn.ap-south-1.rds.amazonaws.com -P 3306 -u admin -p**

**mysonar.cx9yvwbbtwjn.ap-south-1.rds.amazonaws.com--->endpoint get from rds datadase of mysonar**

**Put command and execute in mysql**

**mysql command**

**CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;**

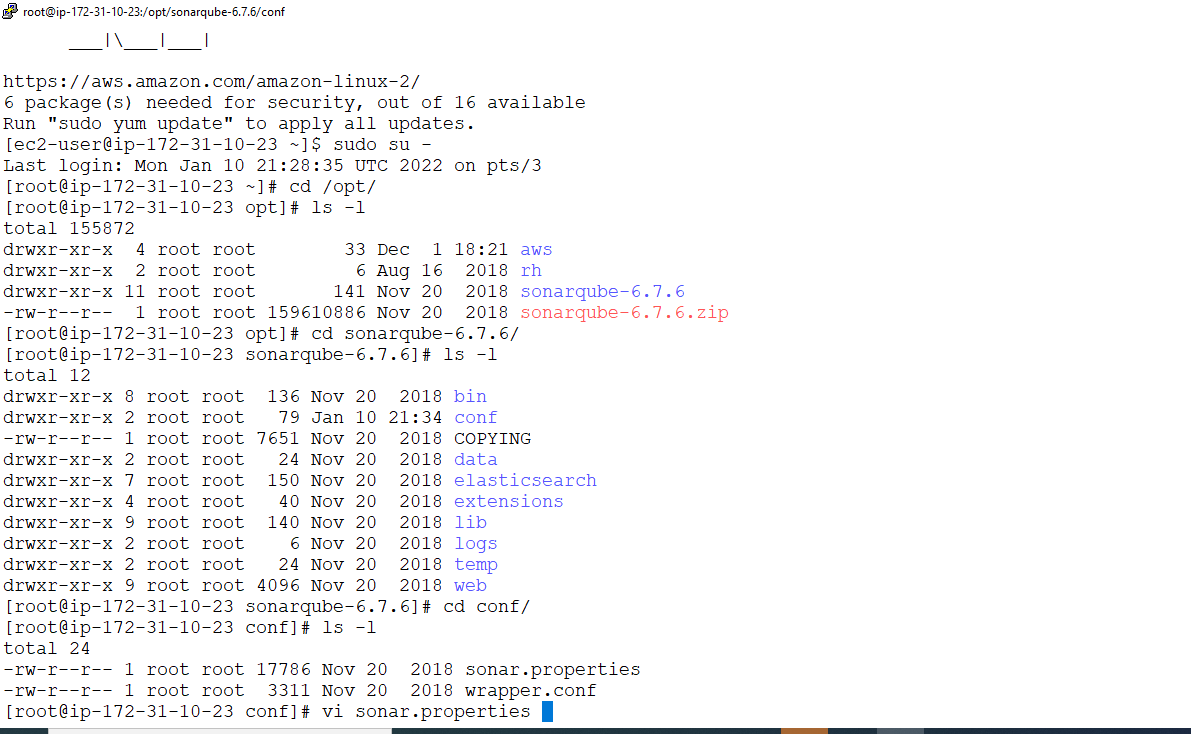
**CREATE USER sonar@localhost IDENTIFIED BY 'sonar';**

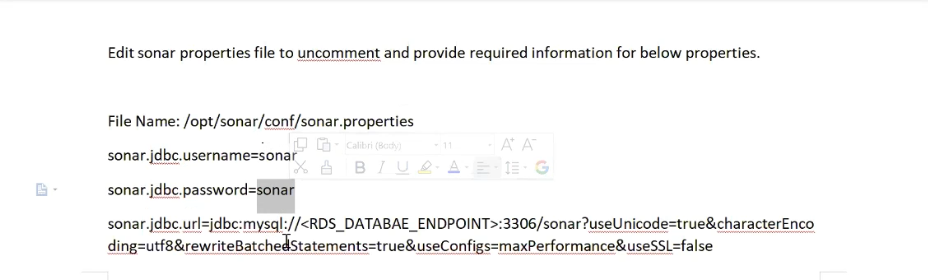
**CREATE USER sonar@'%' IDENTIFIED BY 'sonar';**

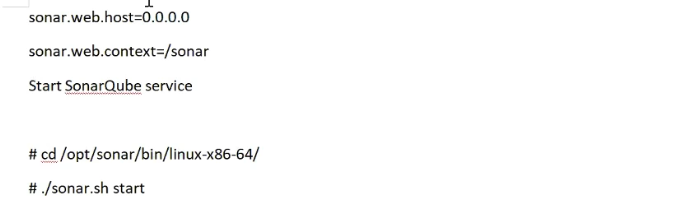
**GRANT ALL ON sonar.\* TO sonar@localhost;**

**GRANT ALL ON sonar.\* TO sonar@'%';**

**Change to uncomment in sonar properties**

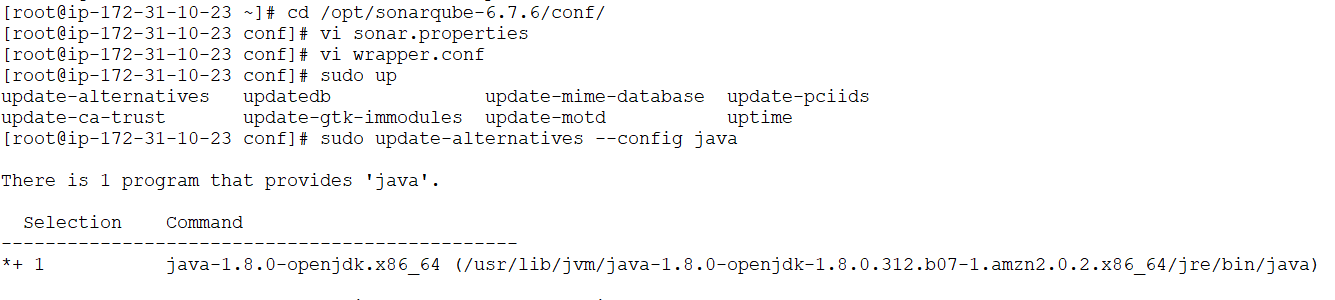






**To change wrapper file**

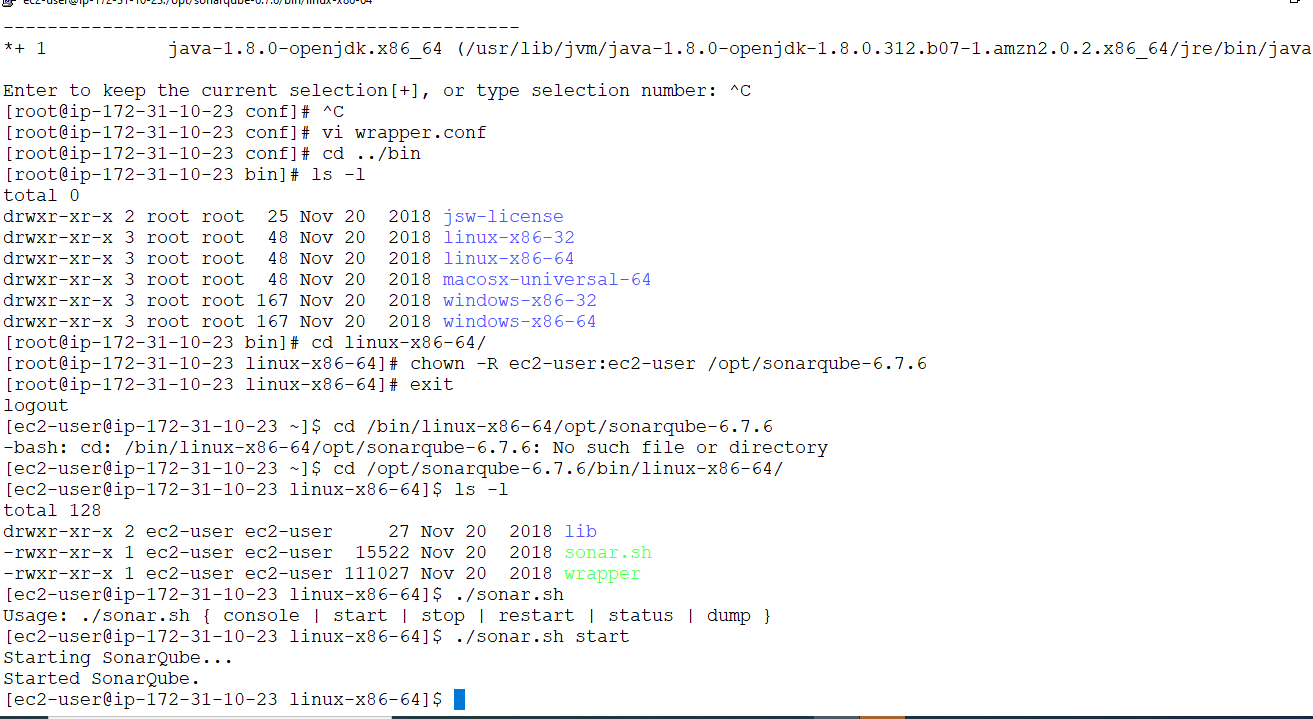
**Vi wrapper.conf**



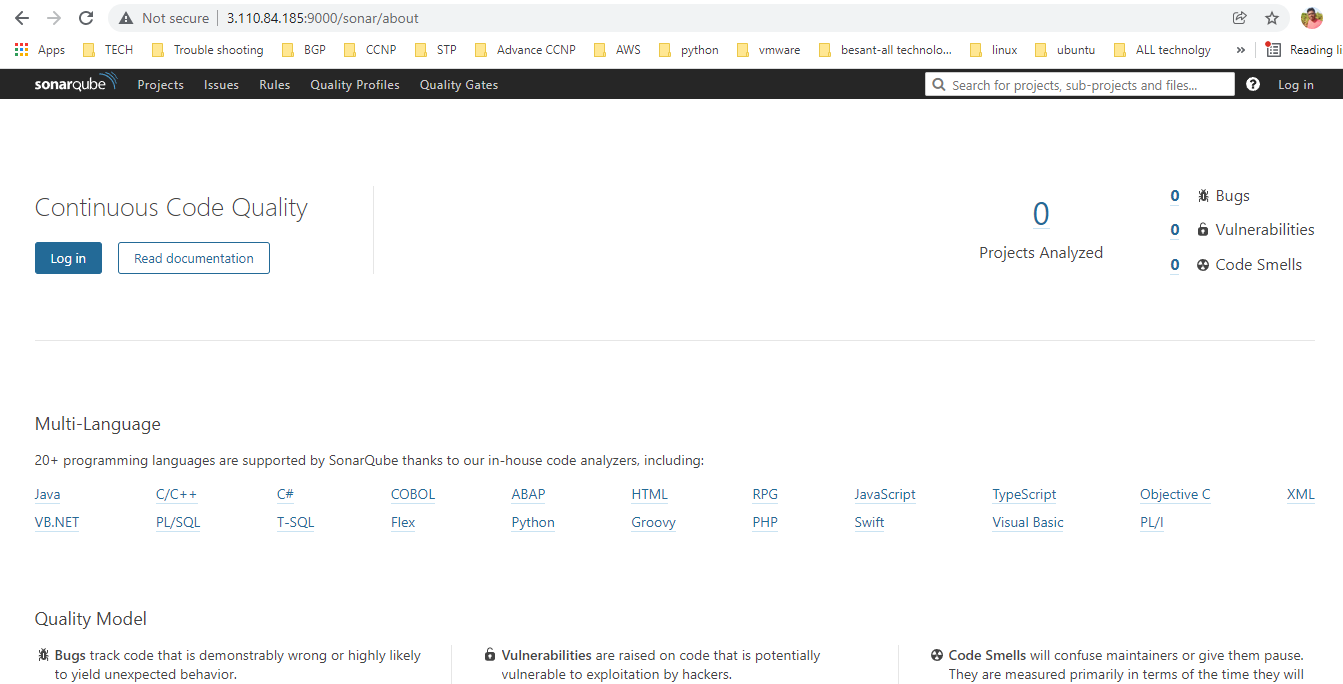


**Sonarcube don’t start in root we can change some command below screenshot**

**We can start in ec2-user**



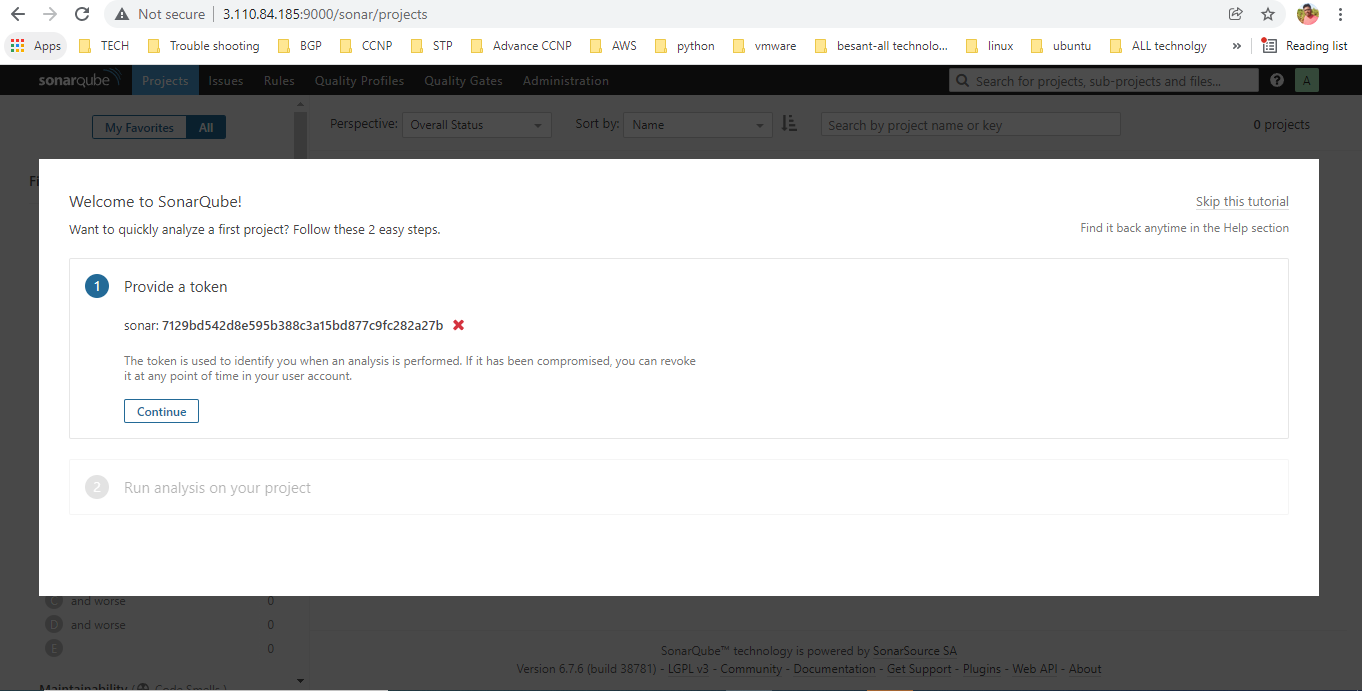
**Sonarcube open webbrowser**



**Give login--->username-admin**

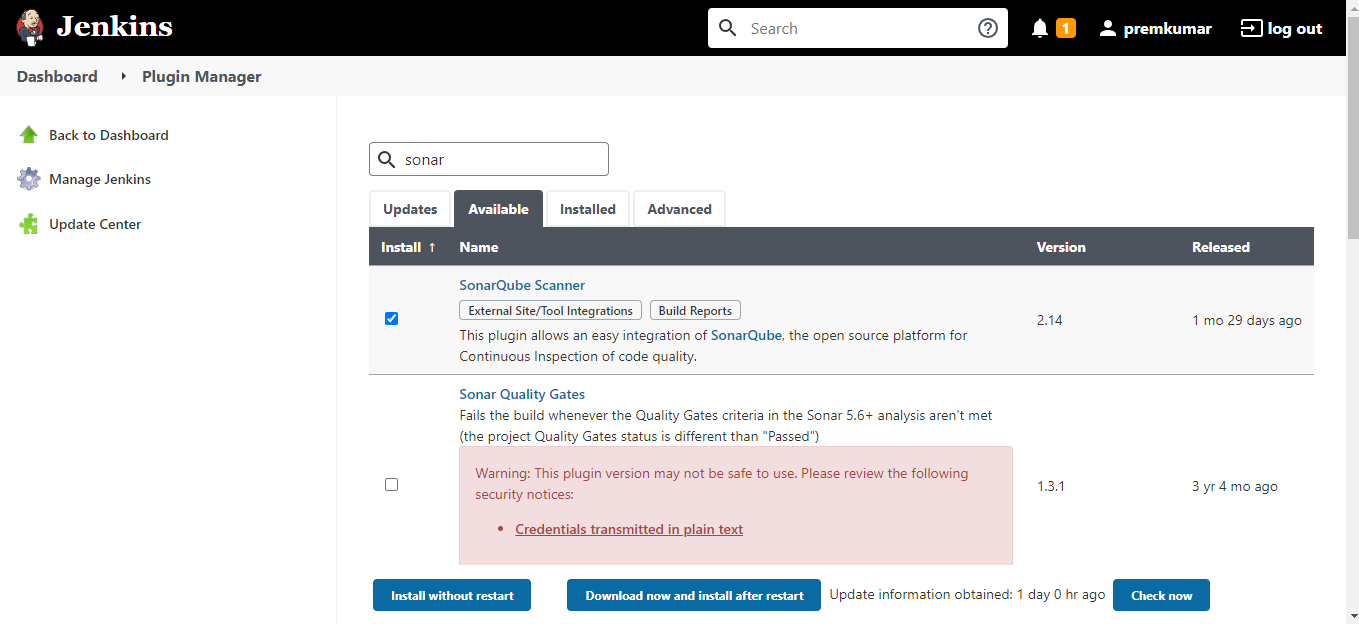
**Password-admin**

**It asking token name ---I give sonar--->it give password that see below**



**Suppose u foget token--->myaccount-->security-->create new token--->change**

**Now sonarqube scanner install in jenkins**

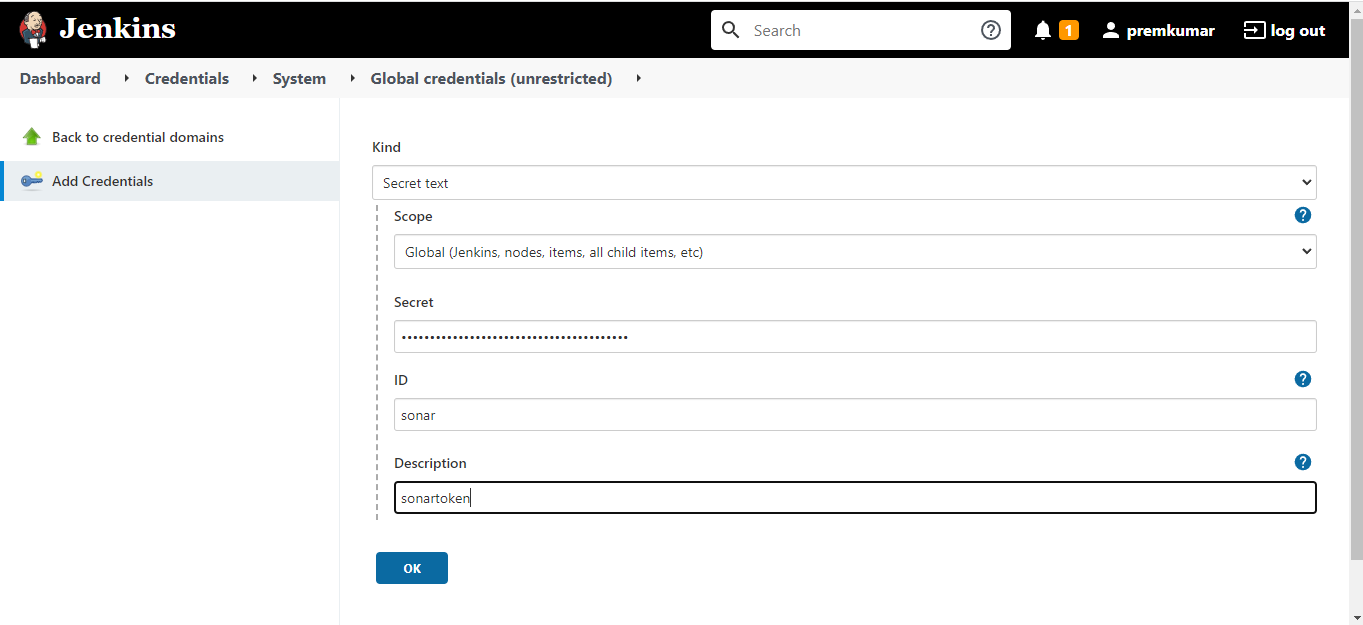
**Managejenkins--->manage plugin--->avialble--->sonarqube scaner-->click-->install without restart**

**Sonarcube password pass to jenkins**

**Sonar password --->give secret password**

**Id-->jenkins file mentioned--->sonar**

**ok**



**Manage jenkin-->configure system---**

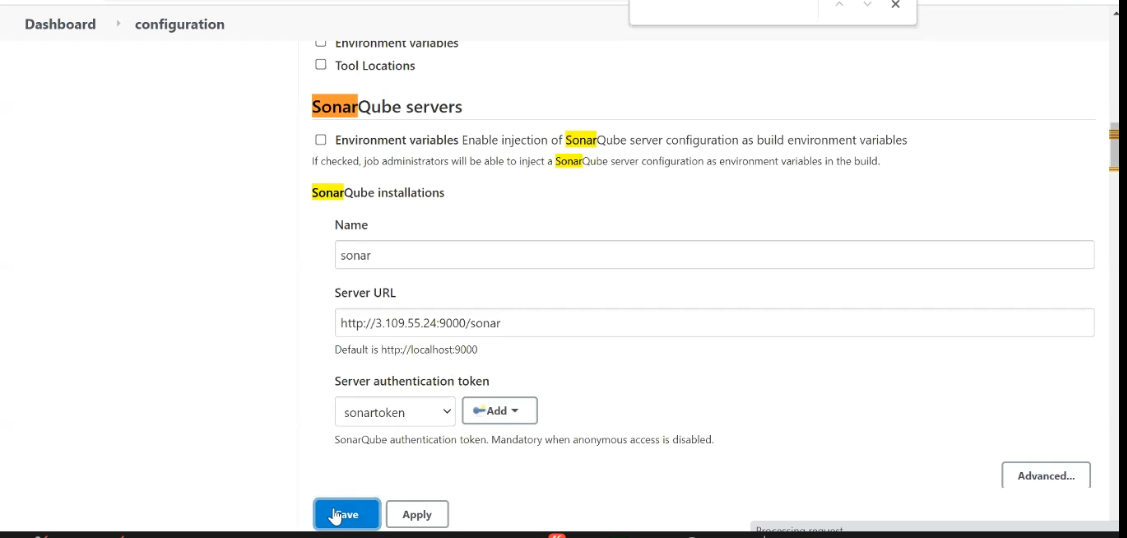
**.ctrl+f-->sonar--->search**

**Name-sonar**

**Url-->sonar url**

**Token-->sonar token**

**save**



**I add sonar cube and remove container in pipeline**

Groovy script

node{

stage('SCM Checkout'){

git 'https://github.com/damodaranj/my-app.git'

}

stage('Compile-Package'){

def mvnHome = tool name: 'maven3', type: 'maven'

sh "${mvnHome}/bin/mvn clean package"

sh 'mv target/myweb\*.war target/newapp.war'

}

stage('SonarQube Analysis') {

def mvnHome = tool name: 'maven3', type: 'maven'

withSonarQubeEnv('sonar') {

sh "${mvnHome}/bin/mvn sonar:sonar"

}

}

stage('Build Docker Imager'){

sh 'docker build -t prempk/myweb:0.0.2 .'

}

stage('Docker Image Push'){

withCredentials([string(credentialsId: 'dockerPass', variable: 'dockerPassword')]) {

sh "docker login -u prempk -p ${dockerPassword}"

}

sh 'docker push prempk/myweb:0.0.2'

stage('Remove Previous Container'){

try{

sh 'docker rm -f tomcattest'

}catch(error){

// do nothing if there is an exception

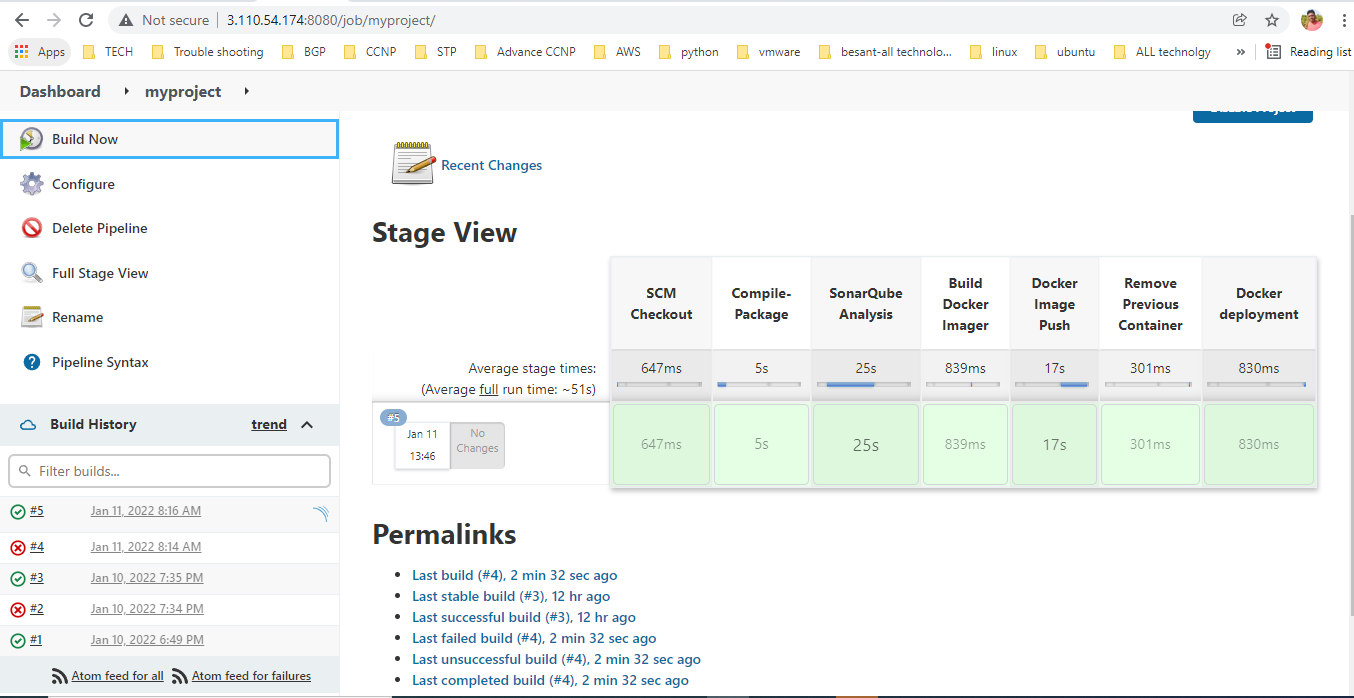
}

stage('Docker deployment'){

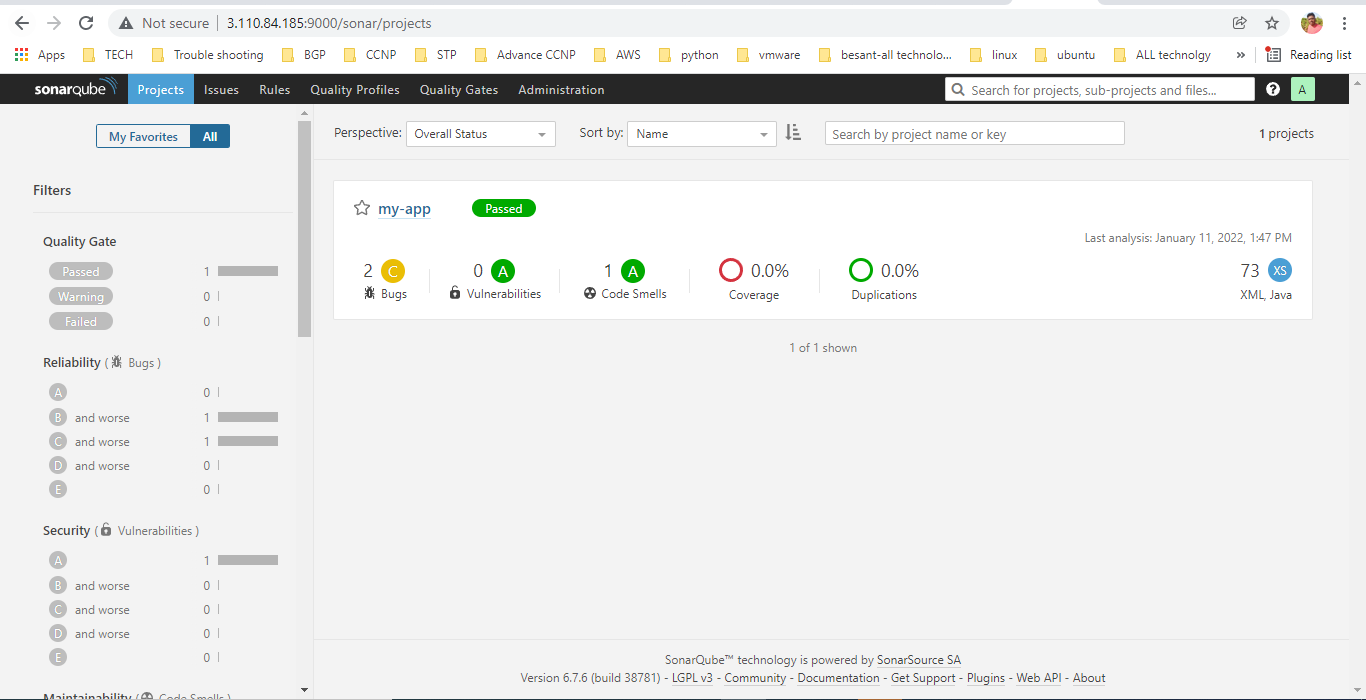
sh 'docker run -d -p 8090:8080 --name tomcattest prempk/myweb:0.0.2'

}

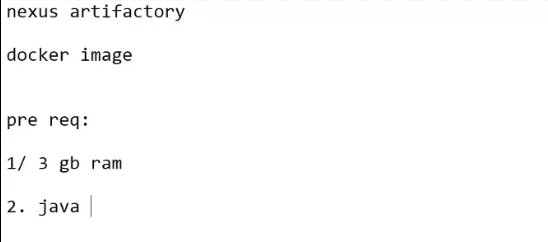
**Build now -->click-->all output sucess**

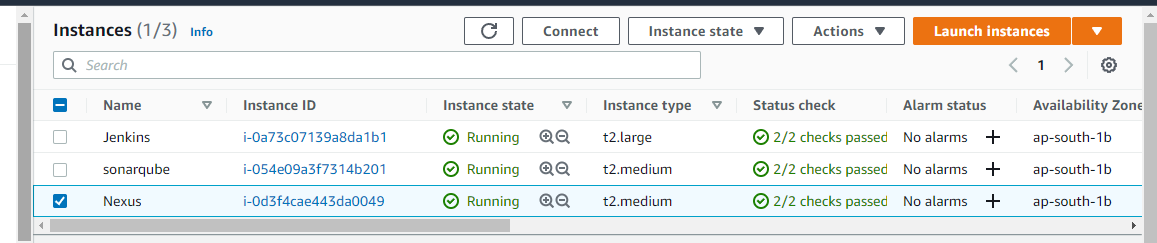


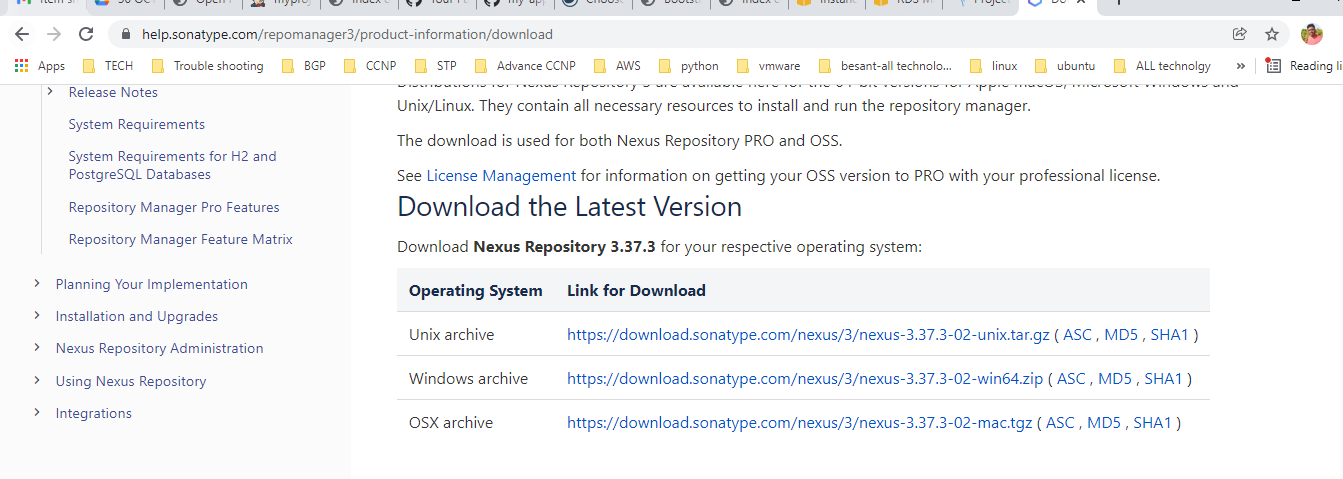
**Finally you can check application in sonarcube see below screenshot**



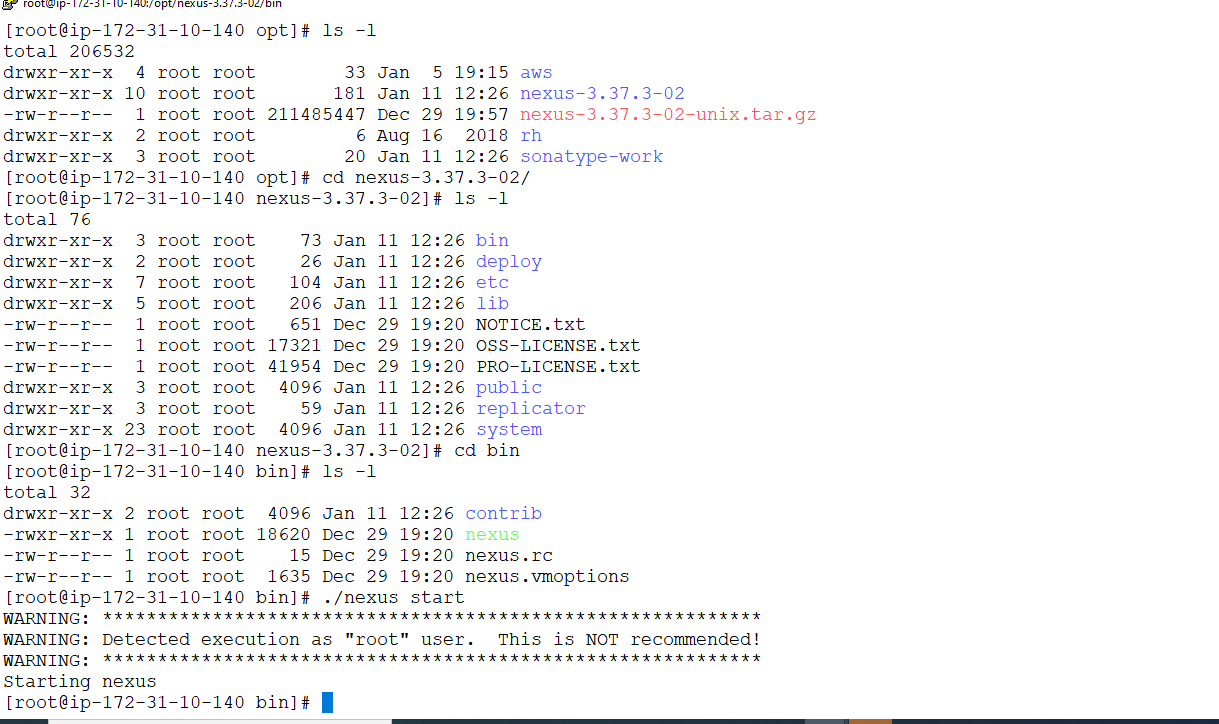
**Nexus installation**

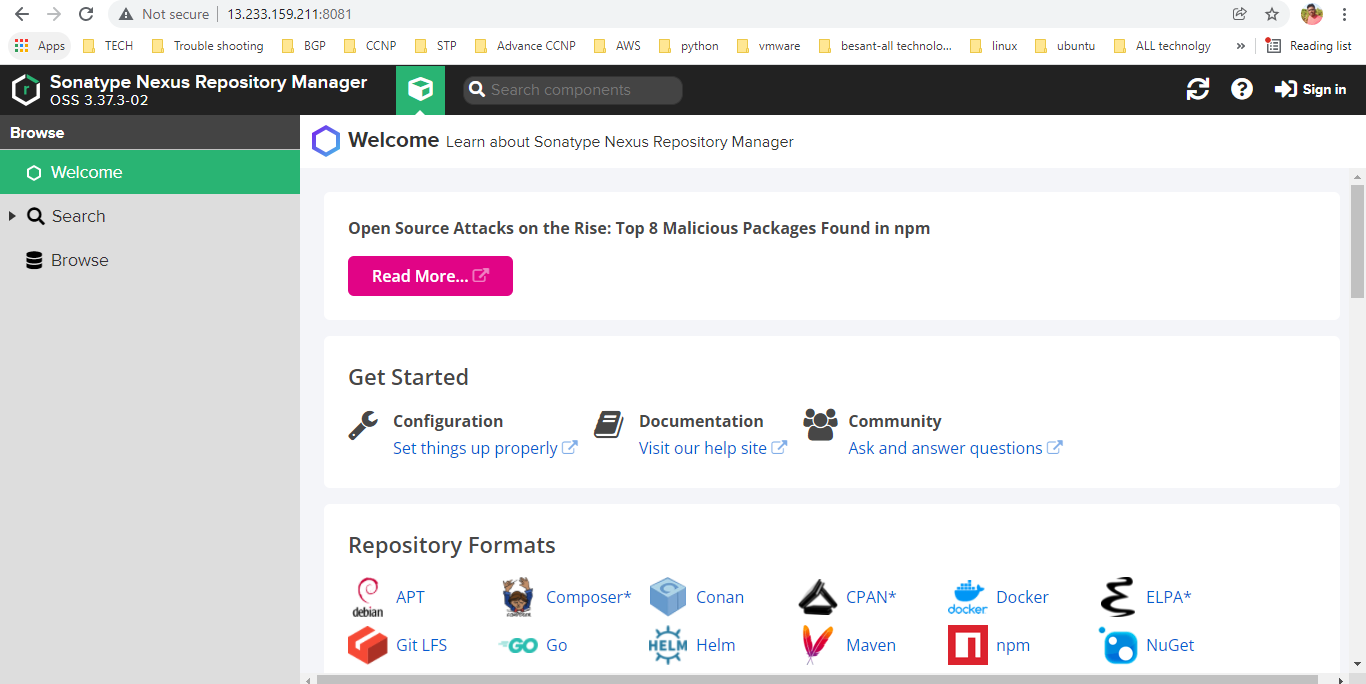




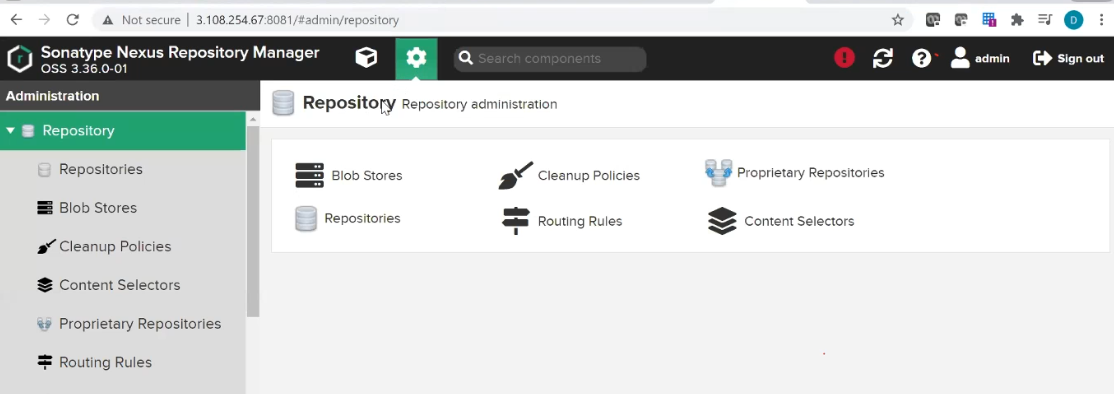


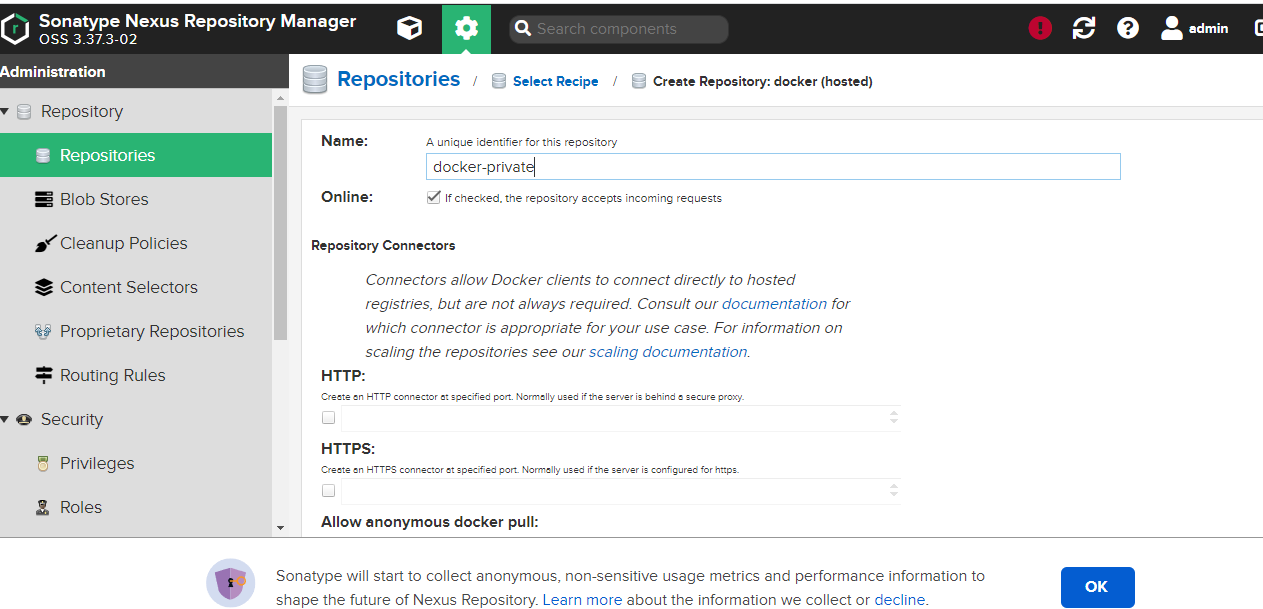


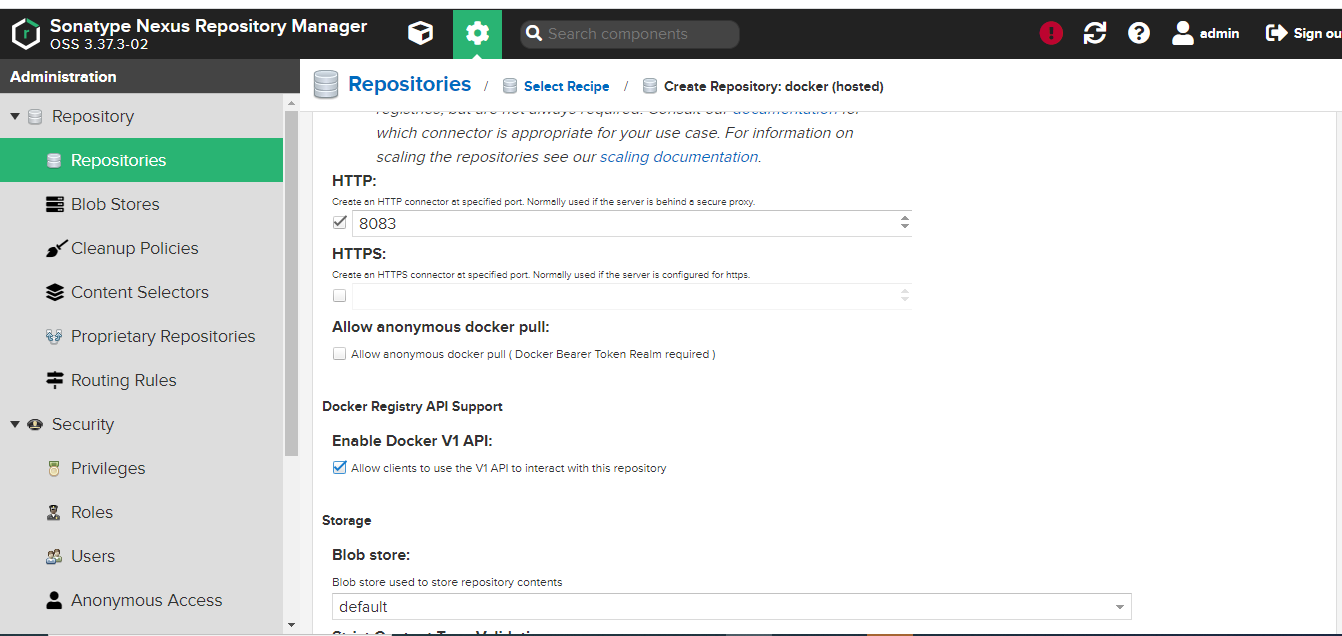




**Setting click--->repositories--->create repositories--->docker(hosted)**

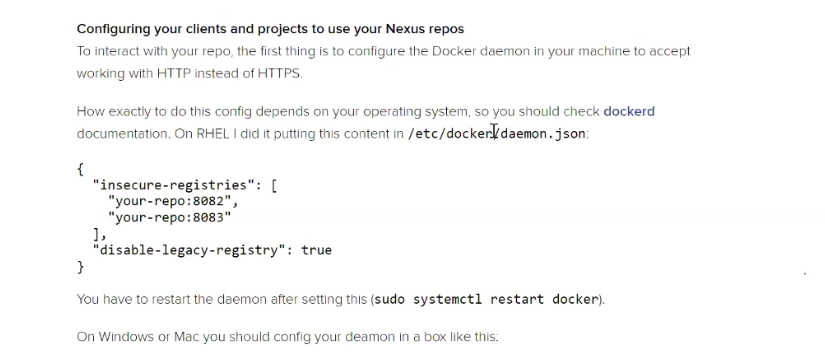




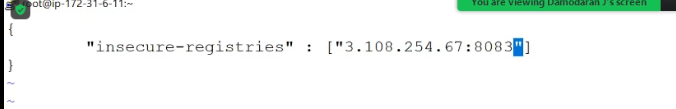


**Docker install in nexus ec2 as root user**

**Yum install docker -y**

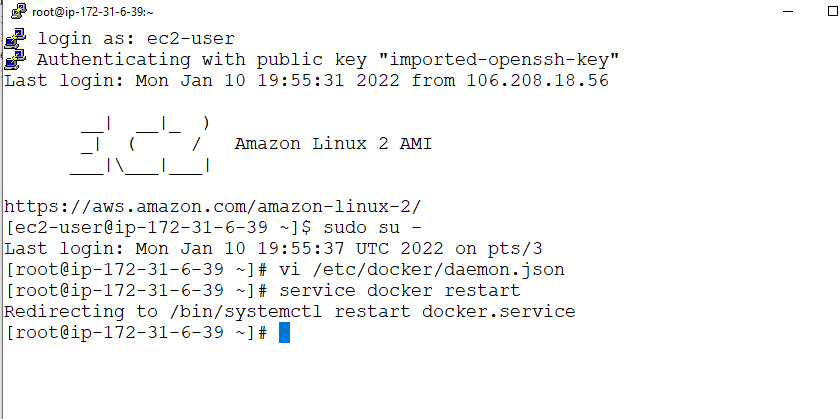


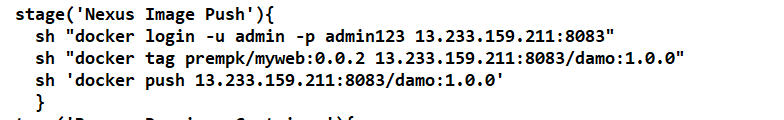
**Vi /etc/docker/daemon.json**



**systemctl restart docker**

**Jenkins interact to nexus**

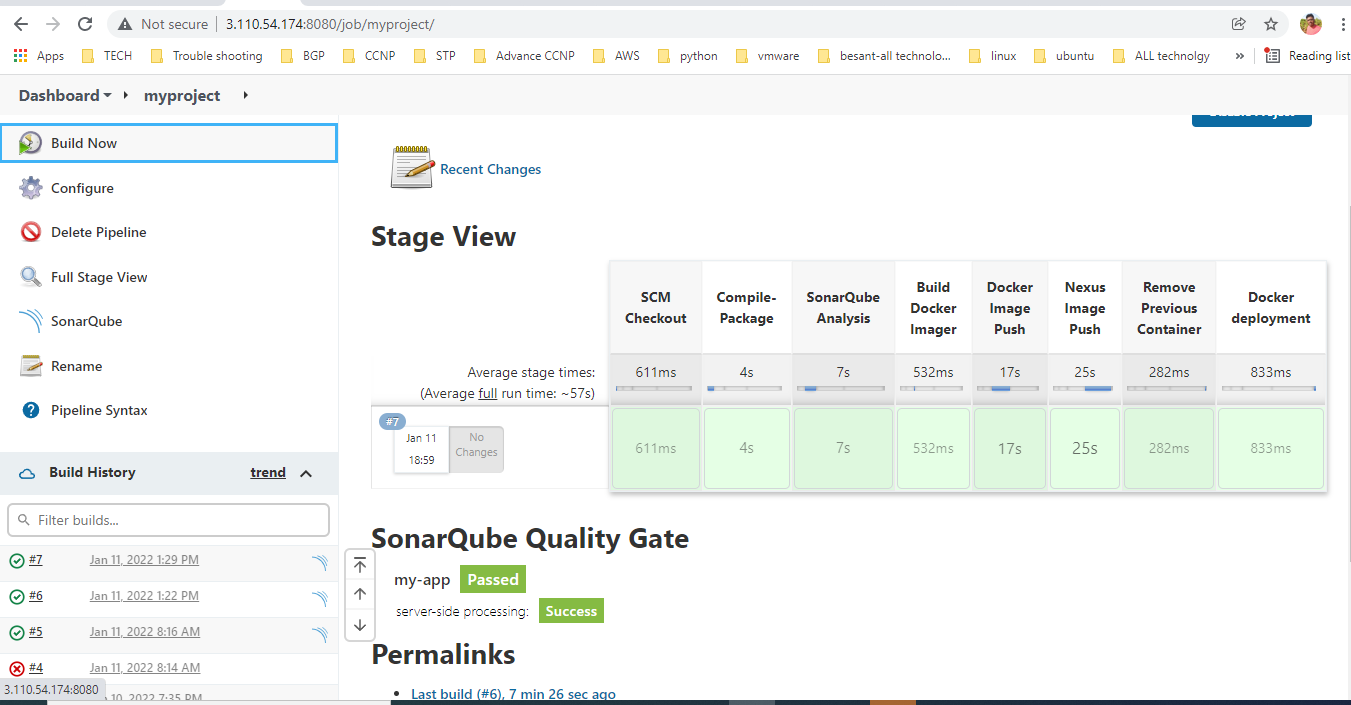




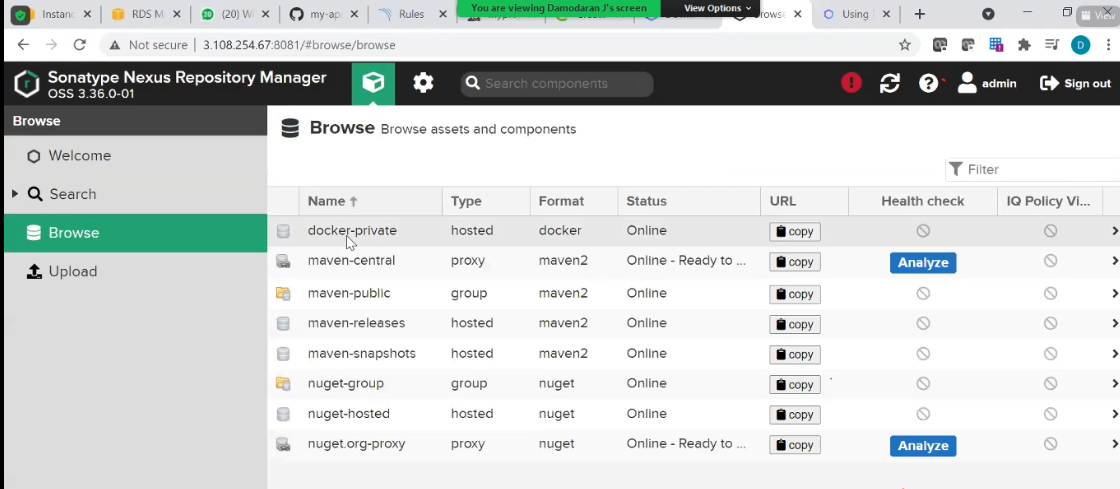
**Add groovy script in pipeline**

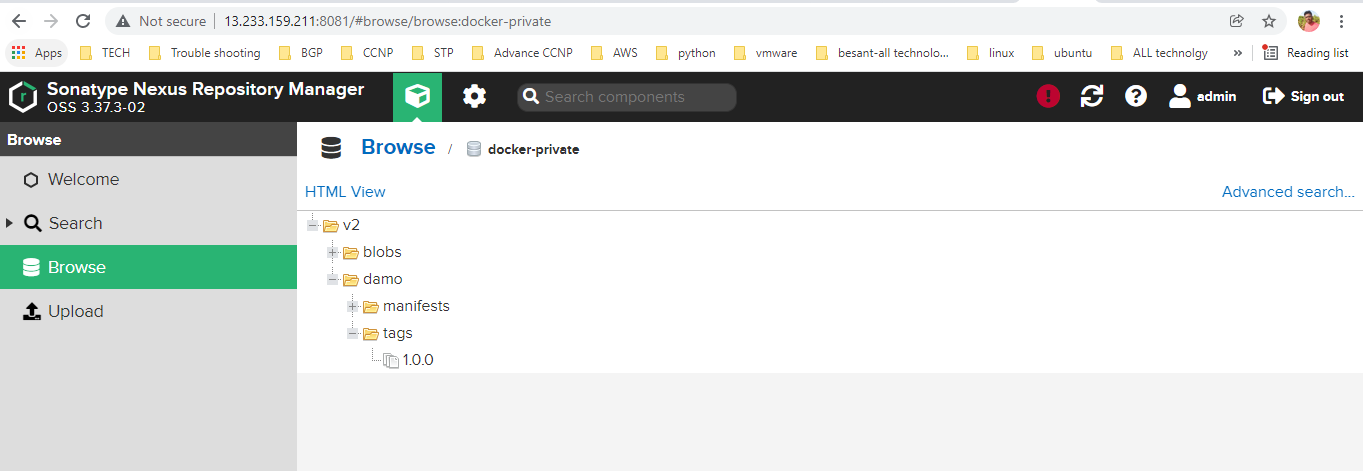
**Nexus private repositry docker image ip 13.233.159.211:8083**

**Now give buildnow in jenkins**



**Docker image moved nexus private repository**





**Promethus install in Nexus**



**vi /etc/prometheus/prometheus.yml---->copy below content and paste**

# my global configglobal:

scrape\_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.

evaluation\_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.

# scrape\_timeout is set to the global default (10s).

# Attach these labels to any time series or alerts when communicating with

# external systems (federation, remote storage, Alertmanager).

external\_labels:

monitor: 'codelab-monitor'

# Load rules once and periodically evaluate them according to the global 'evaluation\_interval'.rule\_files:

# - "first.rules"

# - "second.rules"

# A scrape configuration containing exactly one endpoint to scrape:# Here it's Prometheus itself.scrape\_configs:

# The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.

- job\_name: 'prometheus'

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'docker'

# metrics\_path defaults to '/metrics'

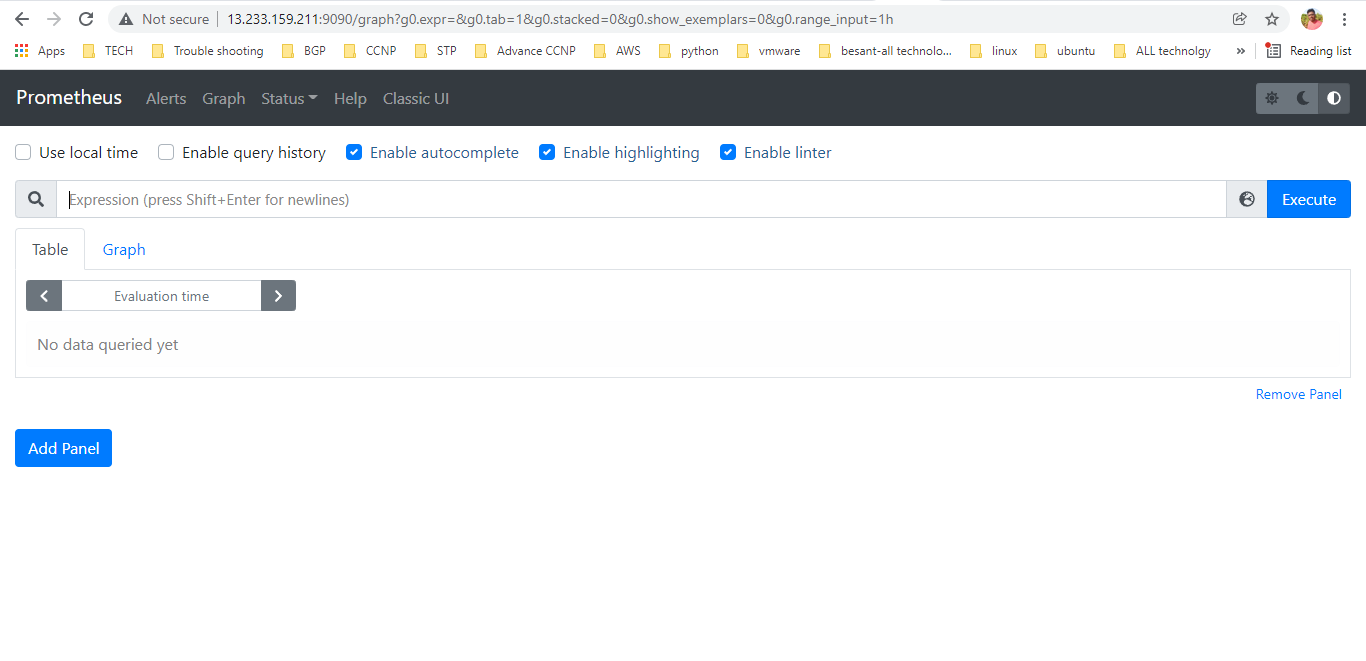
# scheme defaults to 'http'.

static\_configs:

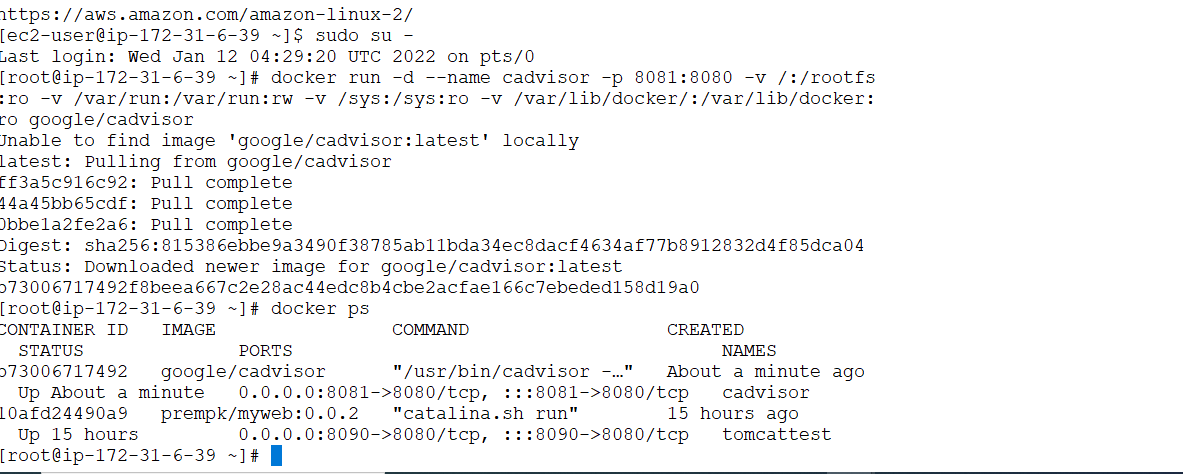
- targets: ['localhost:9323']

**After paste docker start**

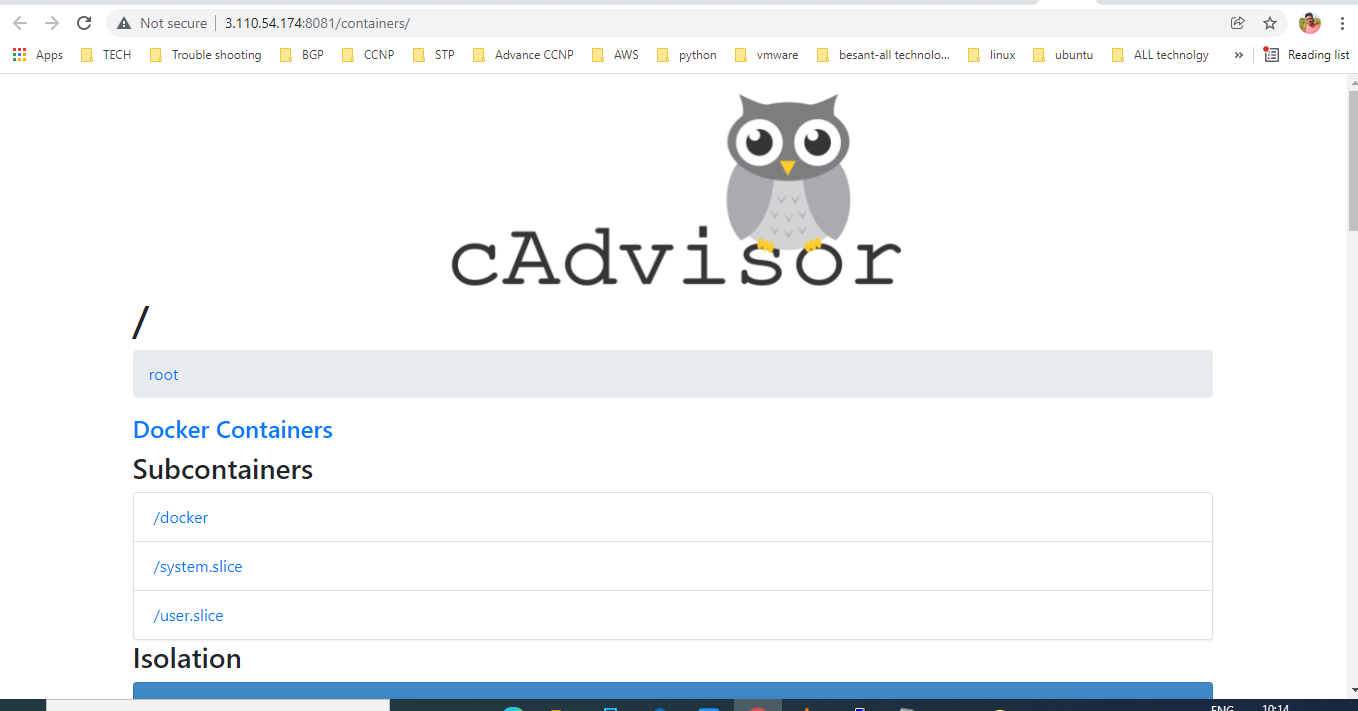
**docker start b682d7807fd4**



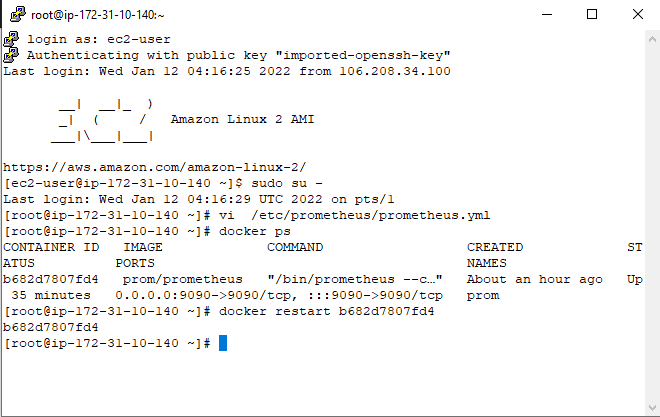
**Prometheus-Cadvisor agent tool install in jenkin for monitor running performance container**



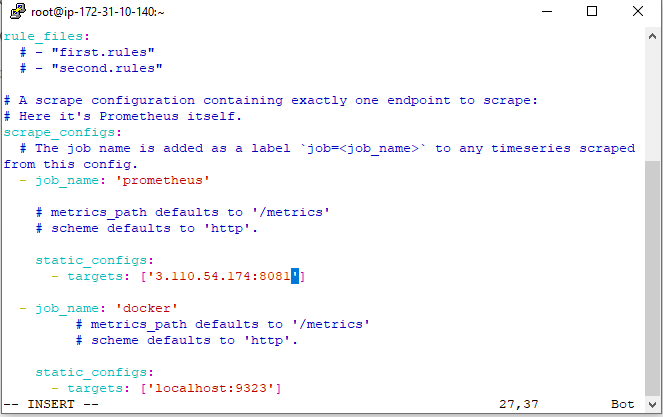
**Jenkins ip :8081**



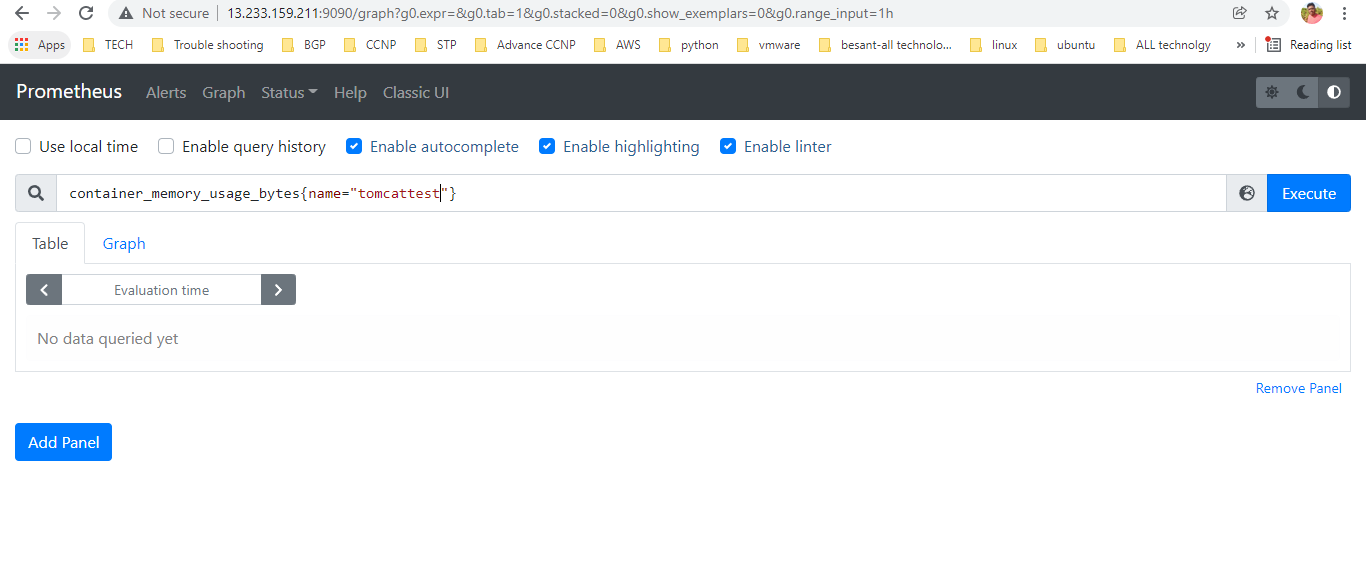
**Prometheus and cadvisor running separte machine so prometheus inform where running cadvisor**

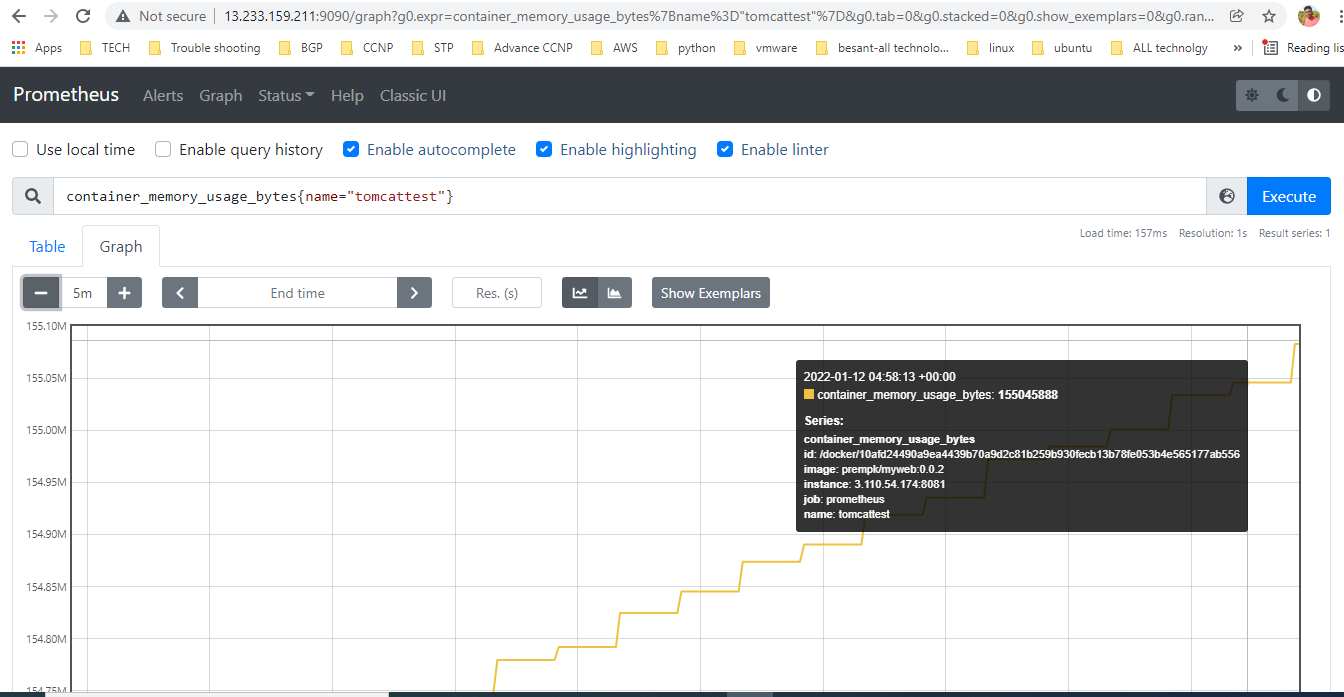


**Jenkins ip for cadvisor and 8081 add below file static\_config**



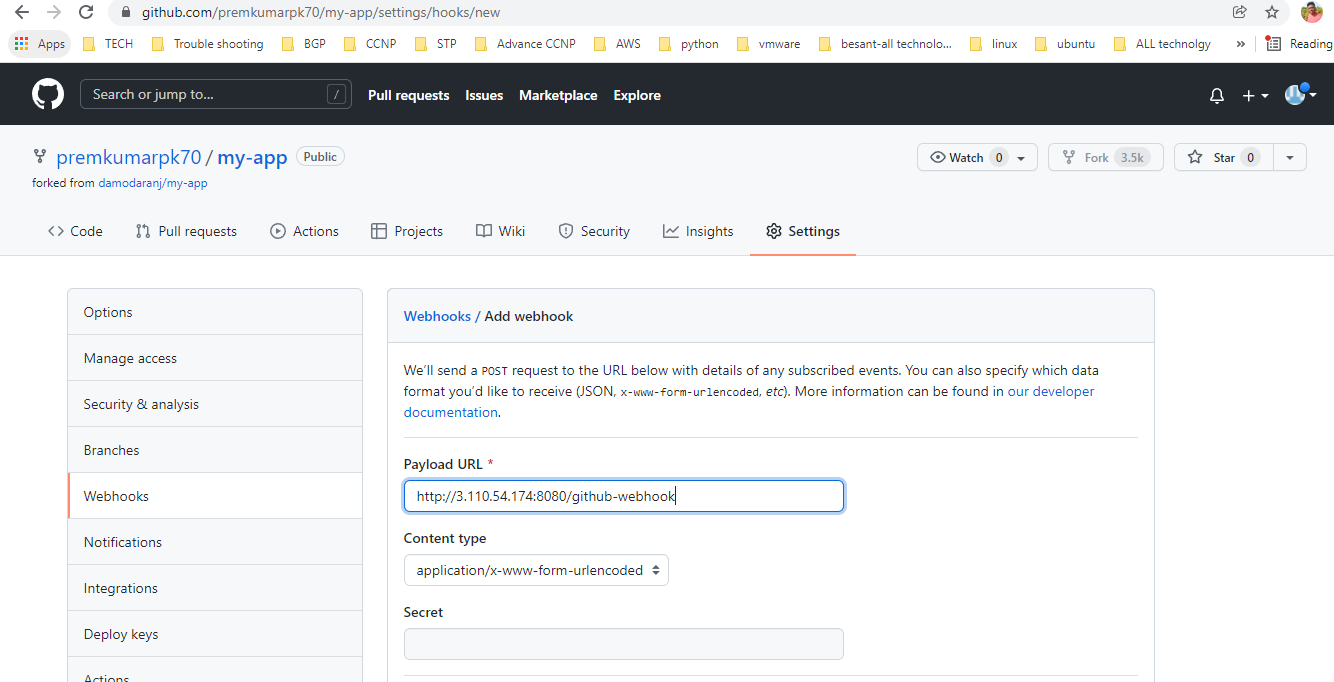
**Container memory check in prometheus--->name--container name---execute**



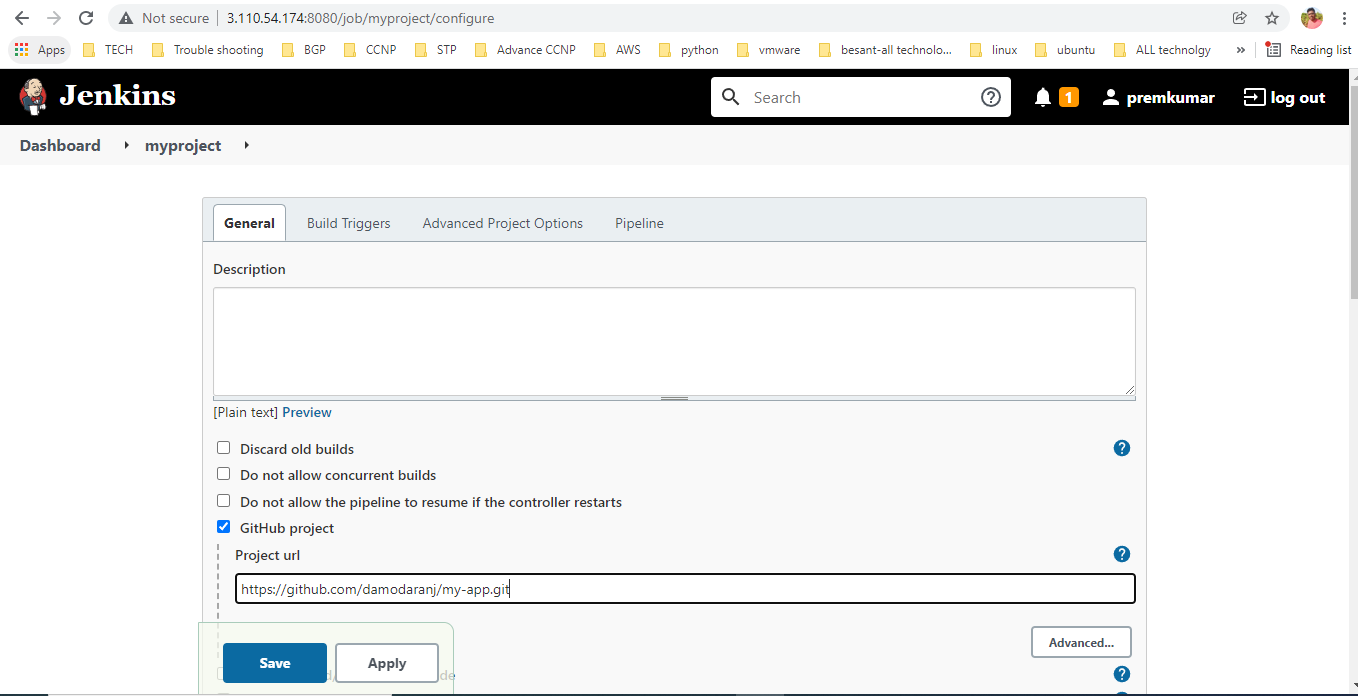


**Jenkins and github integration with webhook**

**Github--->setting--->webhook--->add webhook-->jenkinsip:8080/github-webhook**



**Github url give in jenkins**





**Github-->jenkins code any change--->commit--->automatically build now process happen without human**