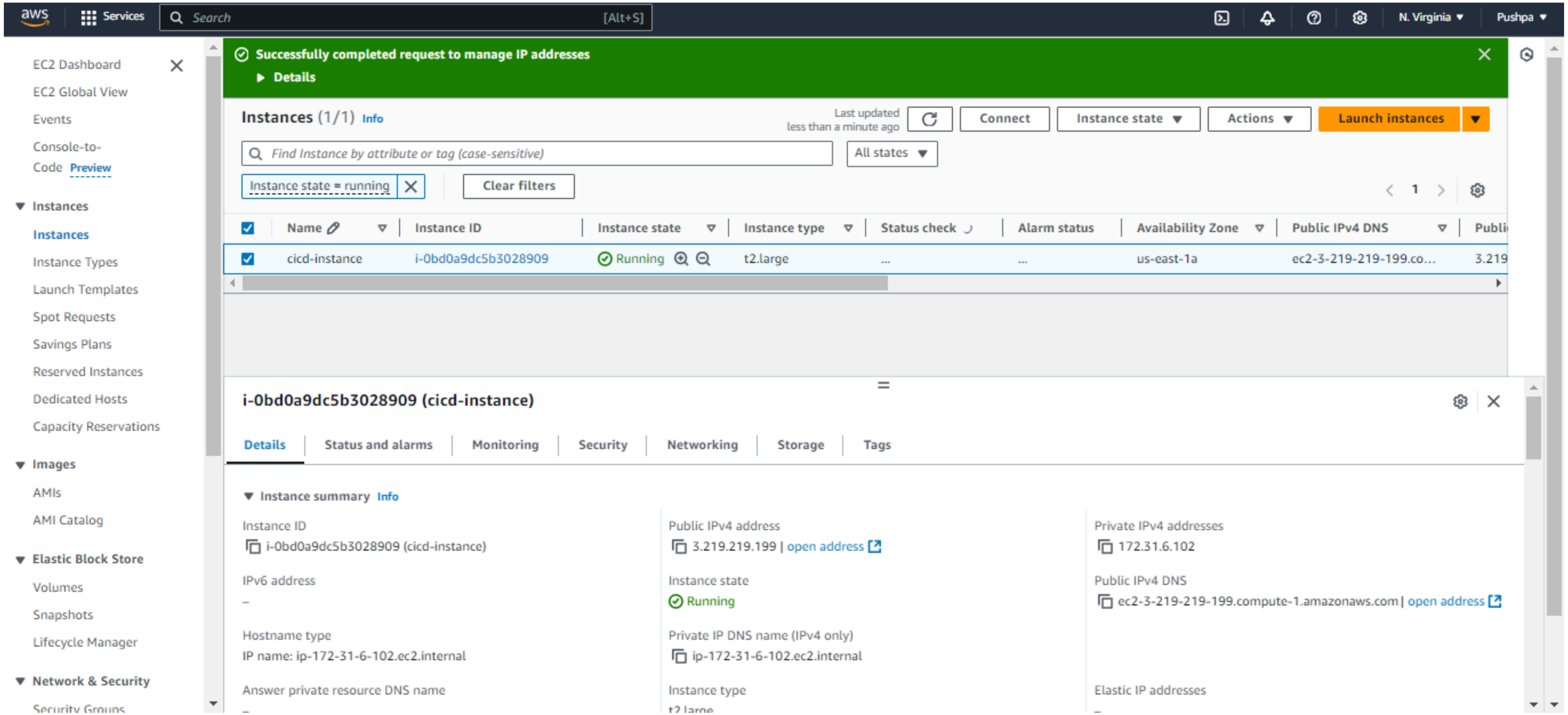
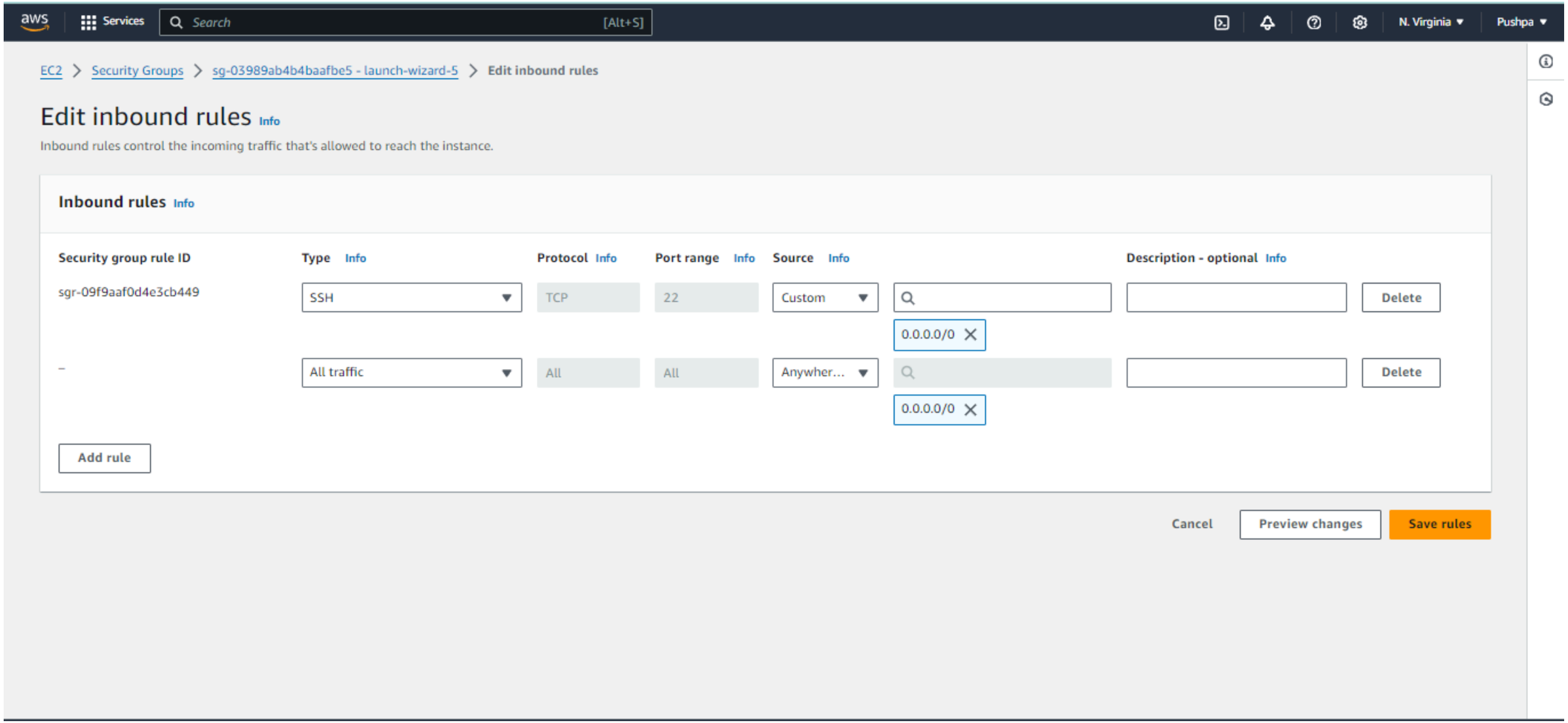
Steps for Continuous Integration

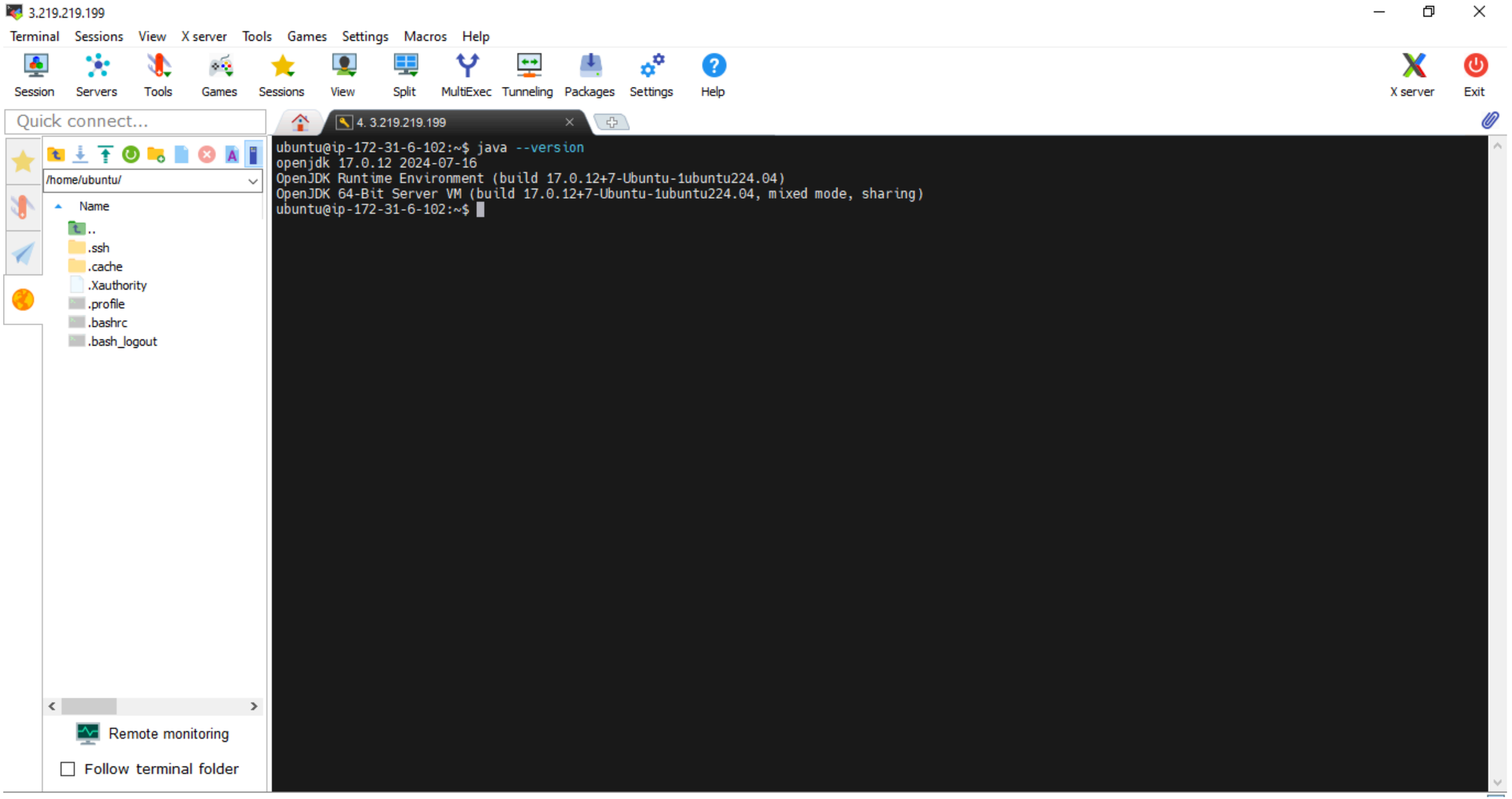
Step:1 Create an ec2 instance of type " t2 large "



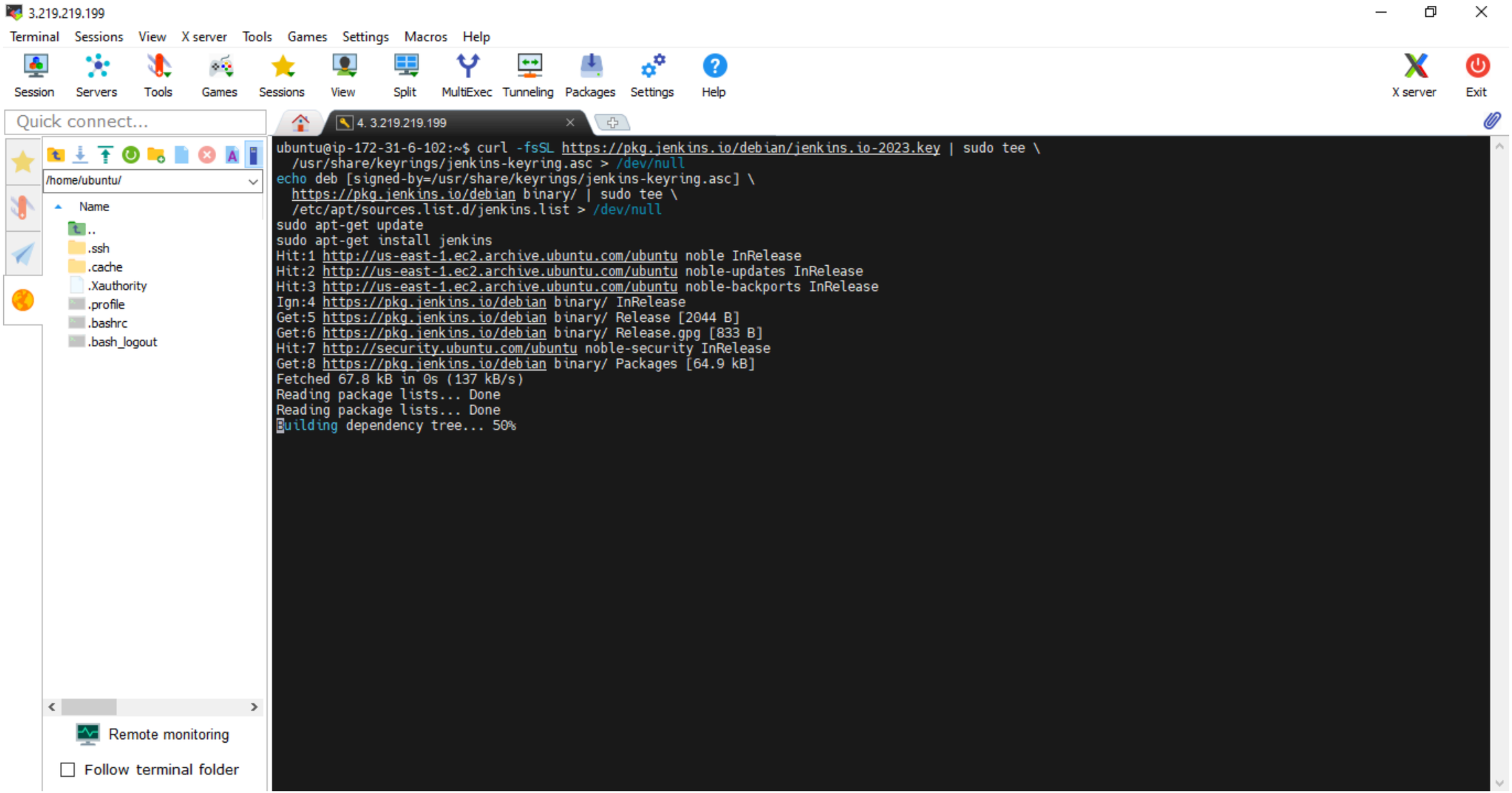
Step:2 Edit the inbound rules



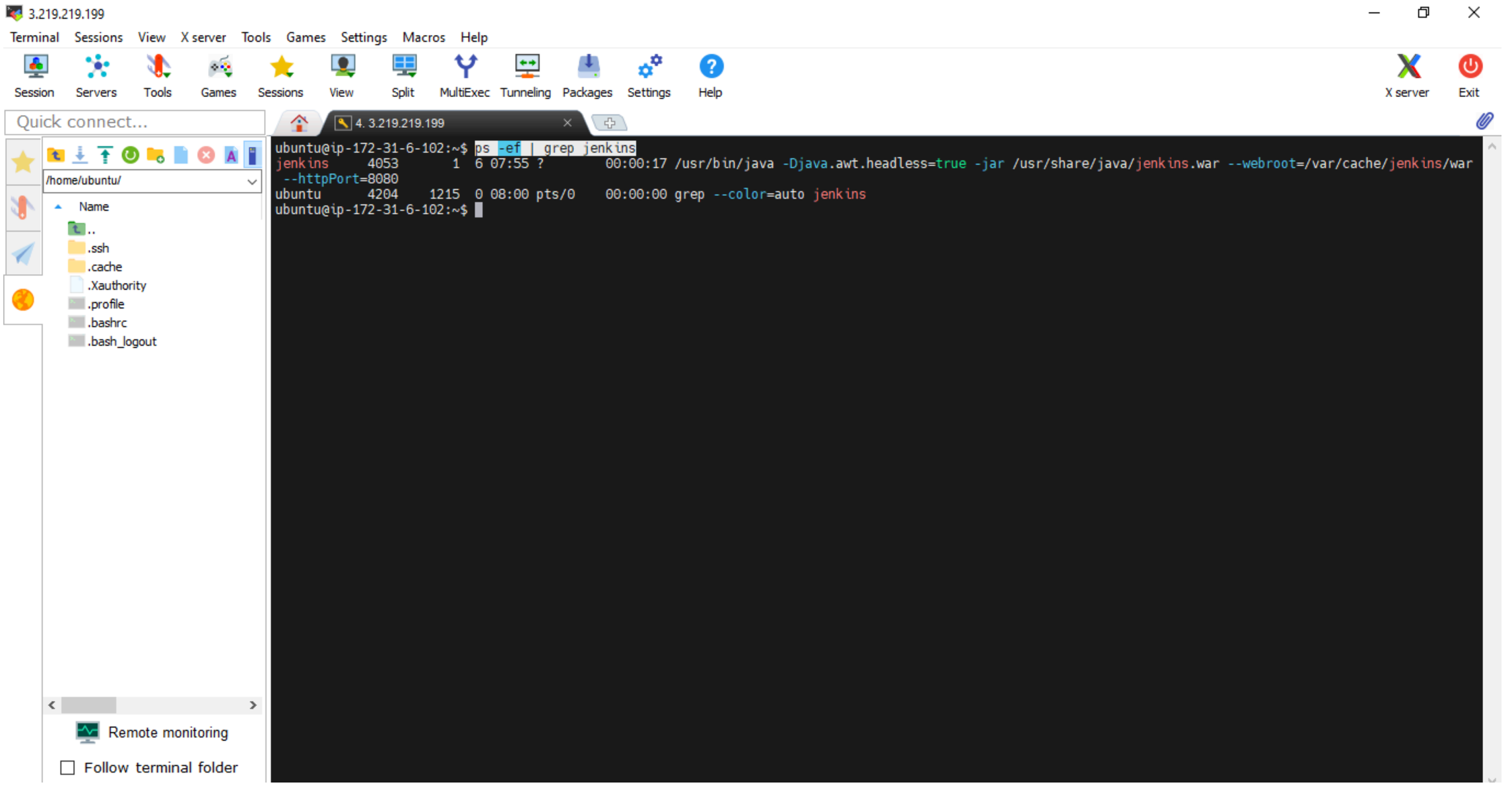
Step:3 connect to the ec2 instance and Install java



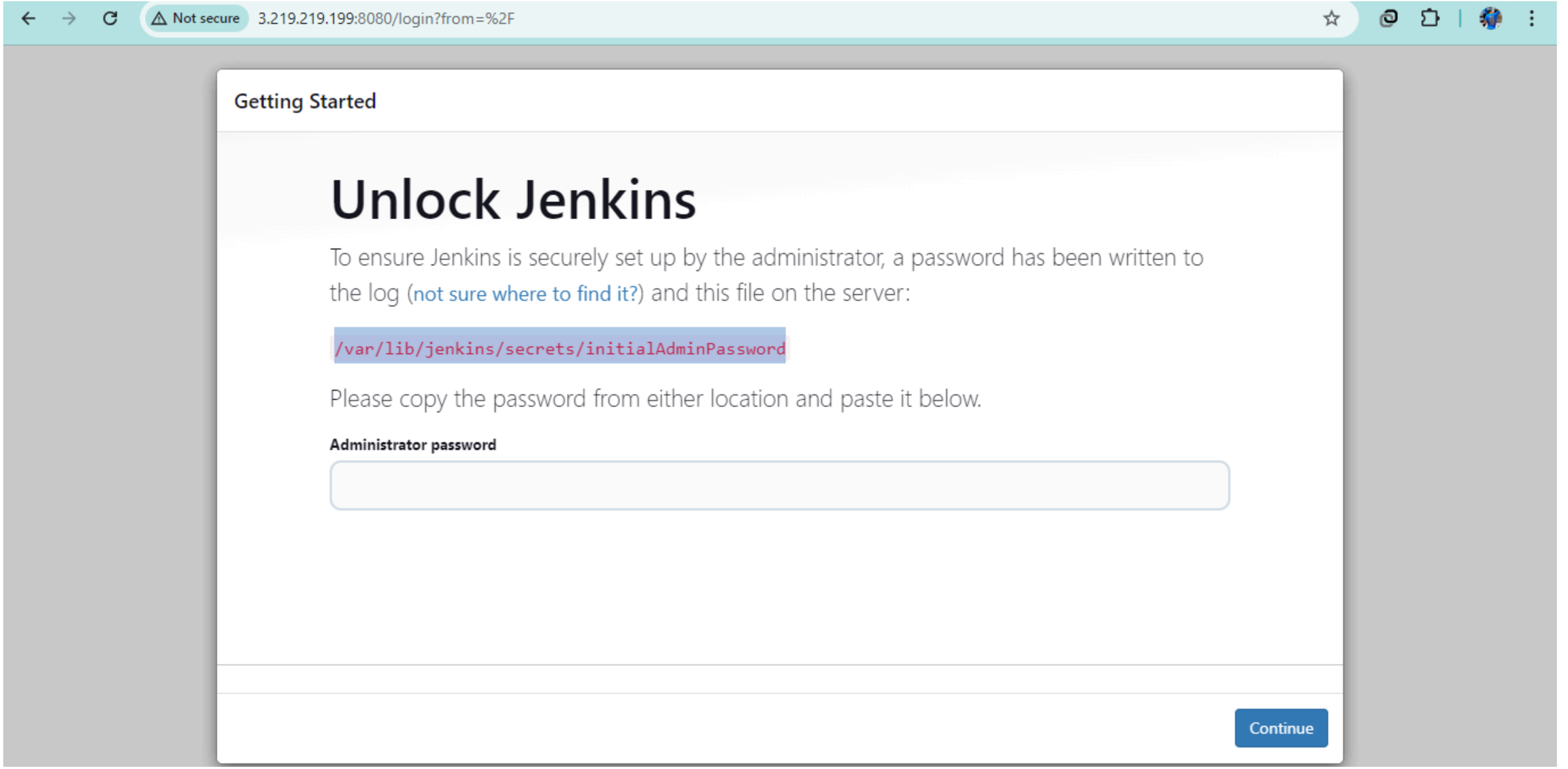
Step:4 Install jenkins



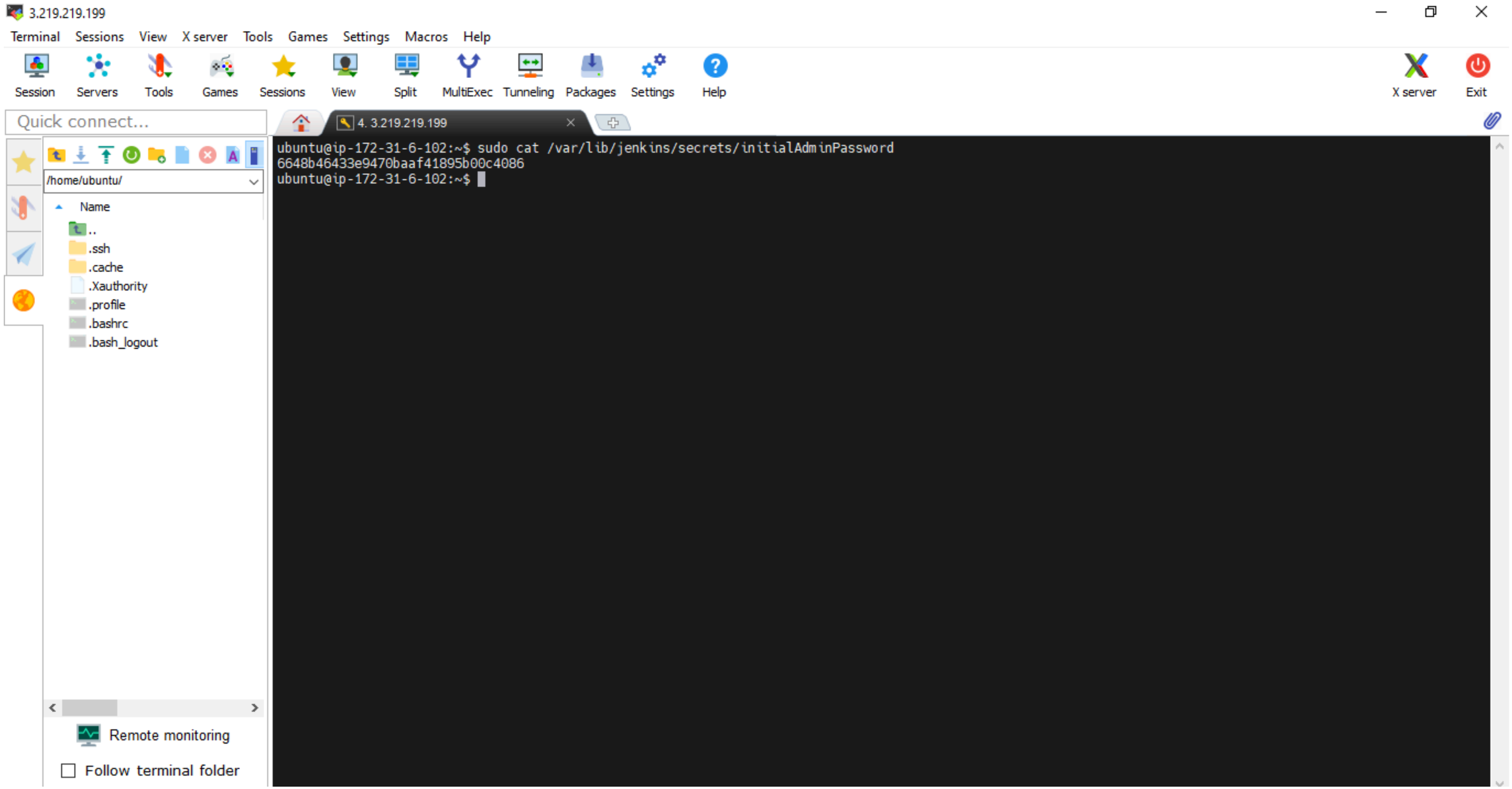
Verify if jenkins is installed or not



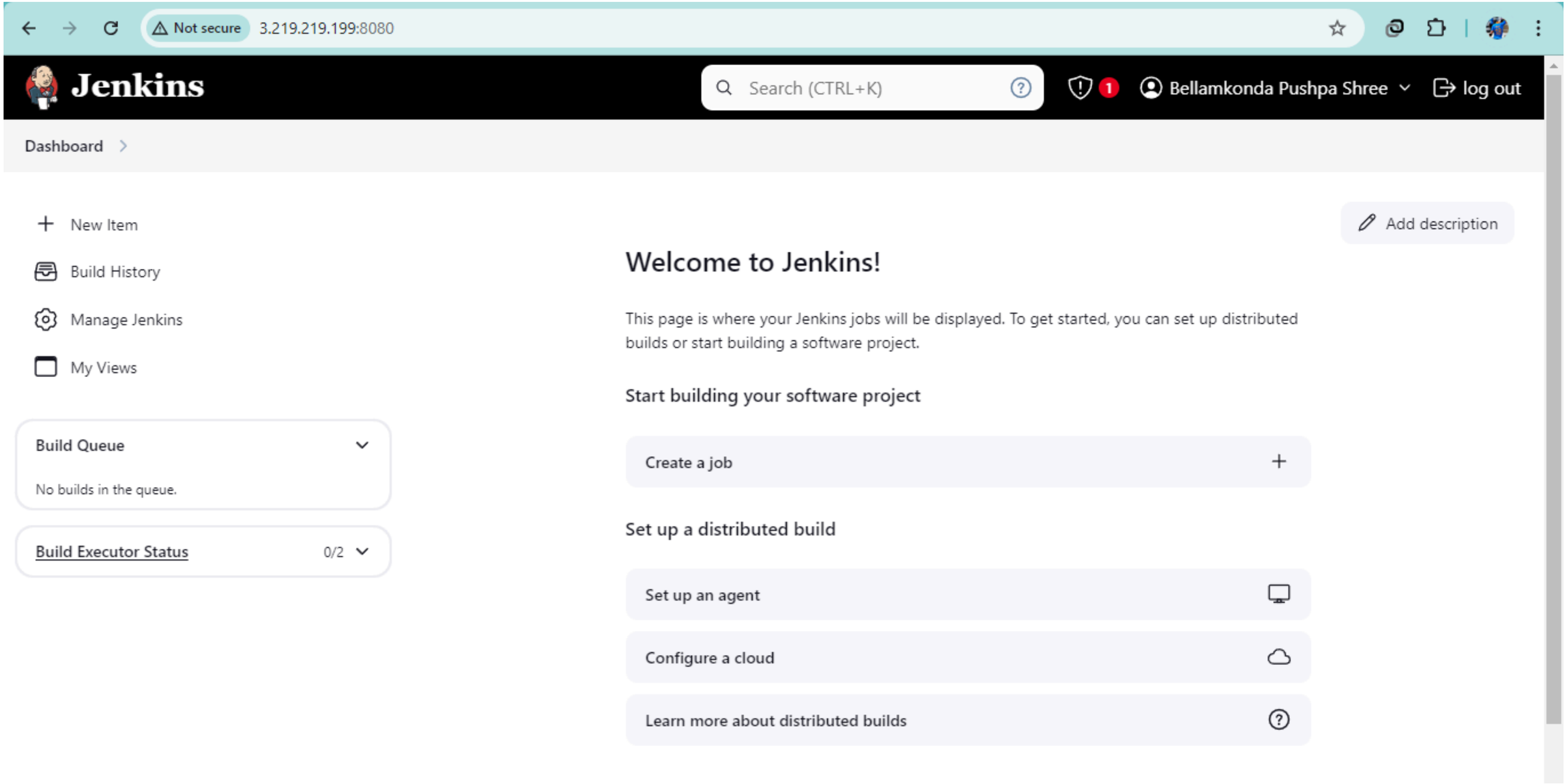
step:5 Browse the public IP of the ec2 instance with port 8080 to access jenkins



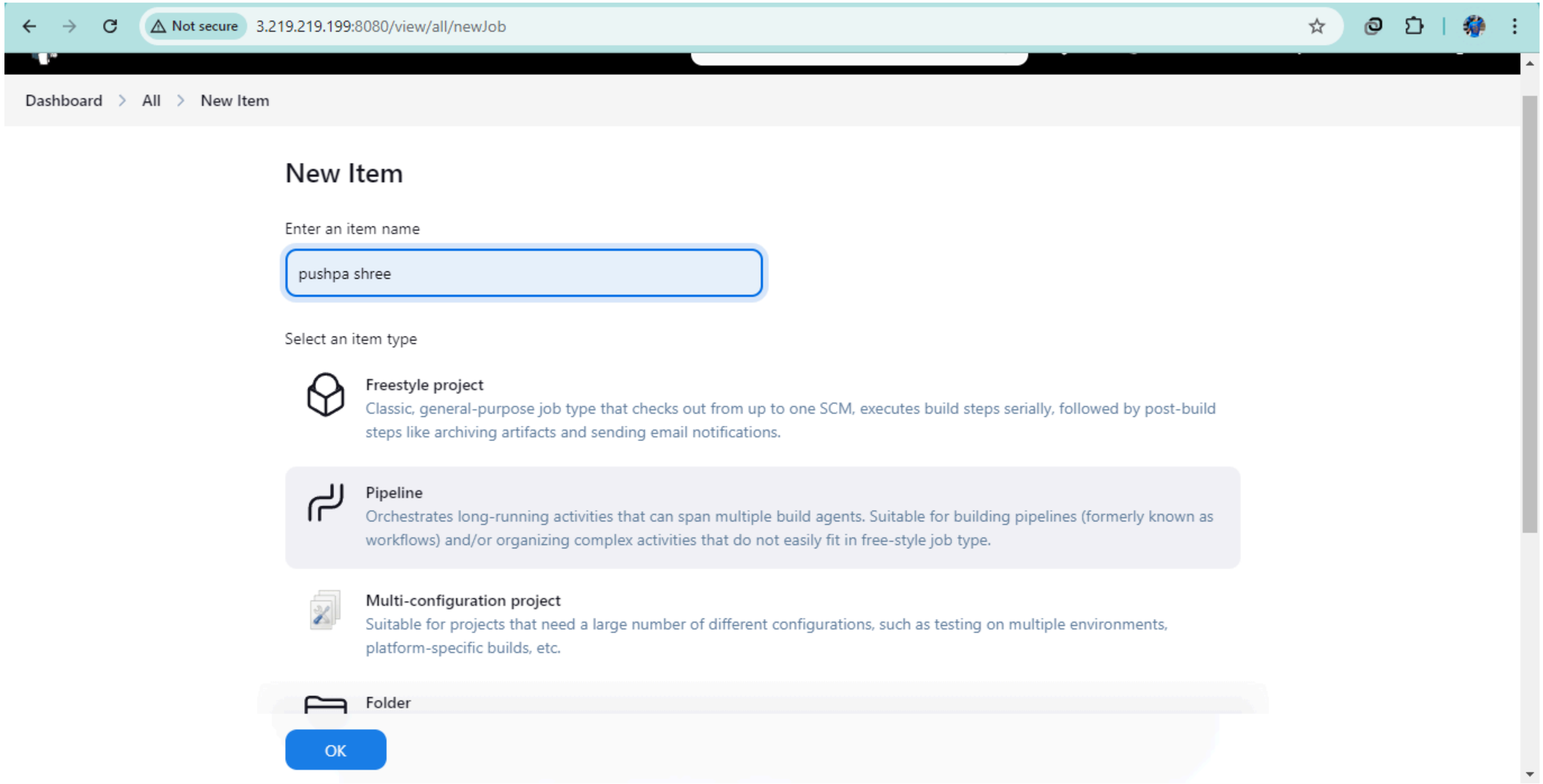
step:6 Copy the url and say "sudo cat url" to get the password for jenkins



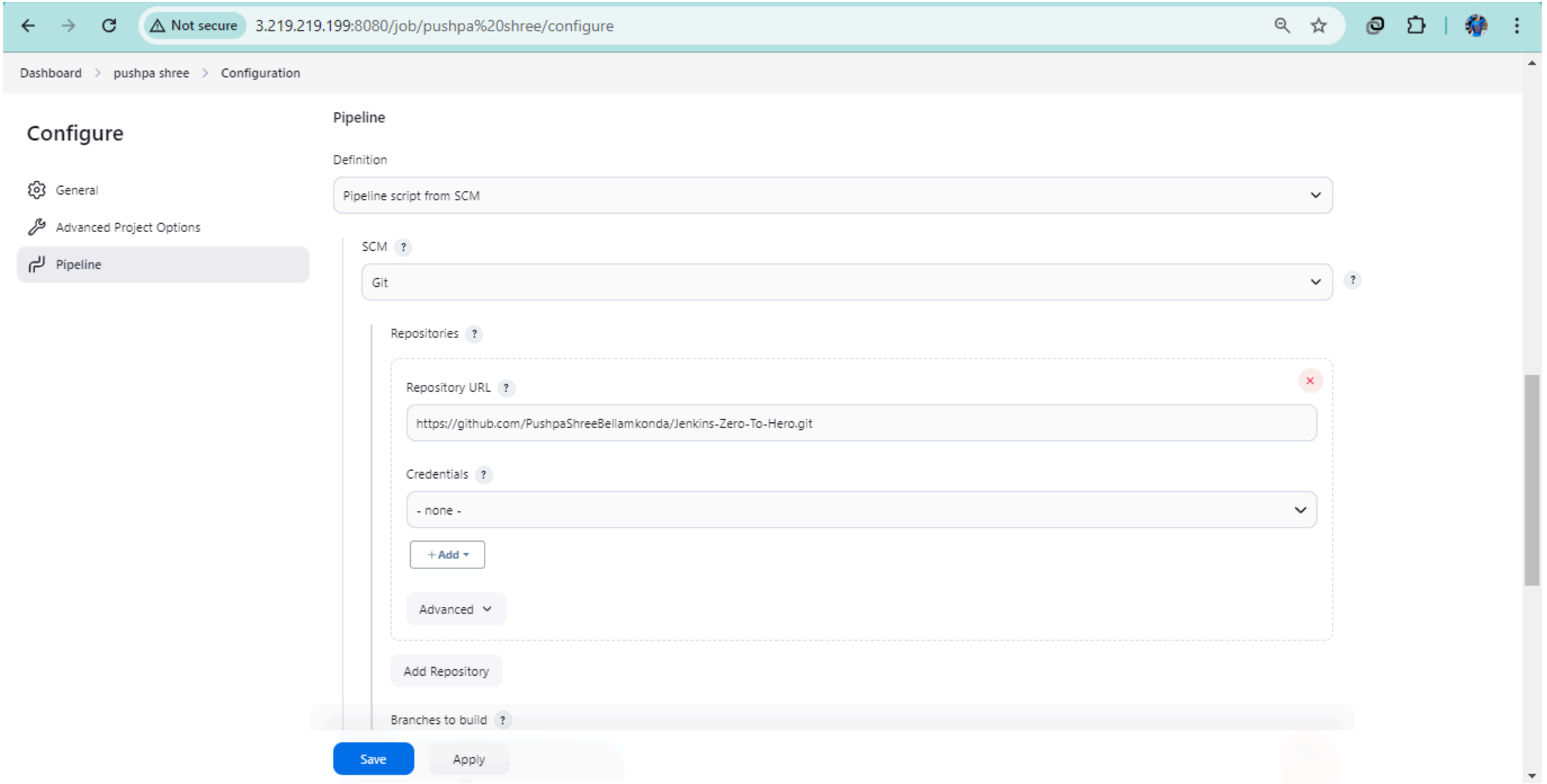
step:7 Now login to the jenkins and install the plugin



step:8 Now select new item , give a name , and select pipeline and ok



step:9 Now in the pipeline select "pipeline script from SCM" and give "github repo url" the "branch" and "Script path"



Not secure 3.219.219.199:8080/job/pushpa%20shree/configure

Dashboard > pushpa shree > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

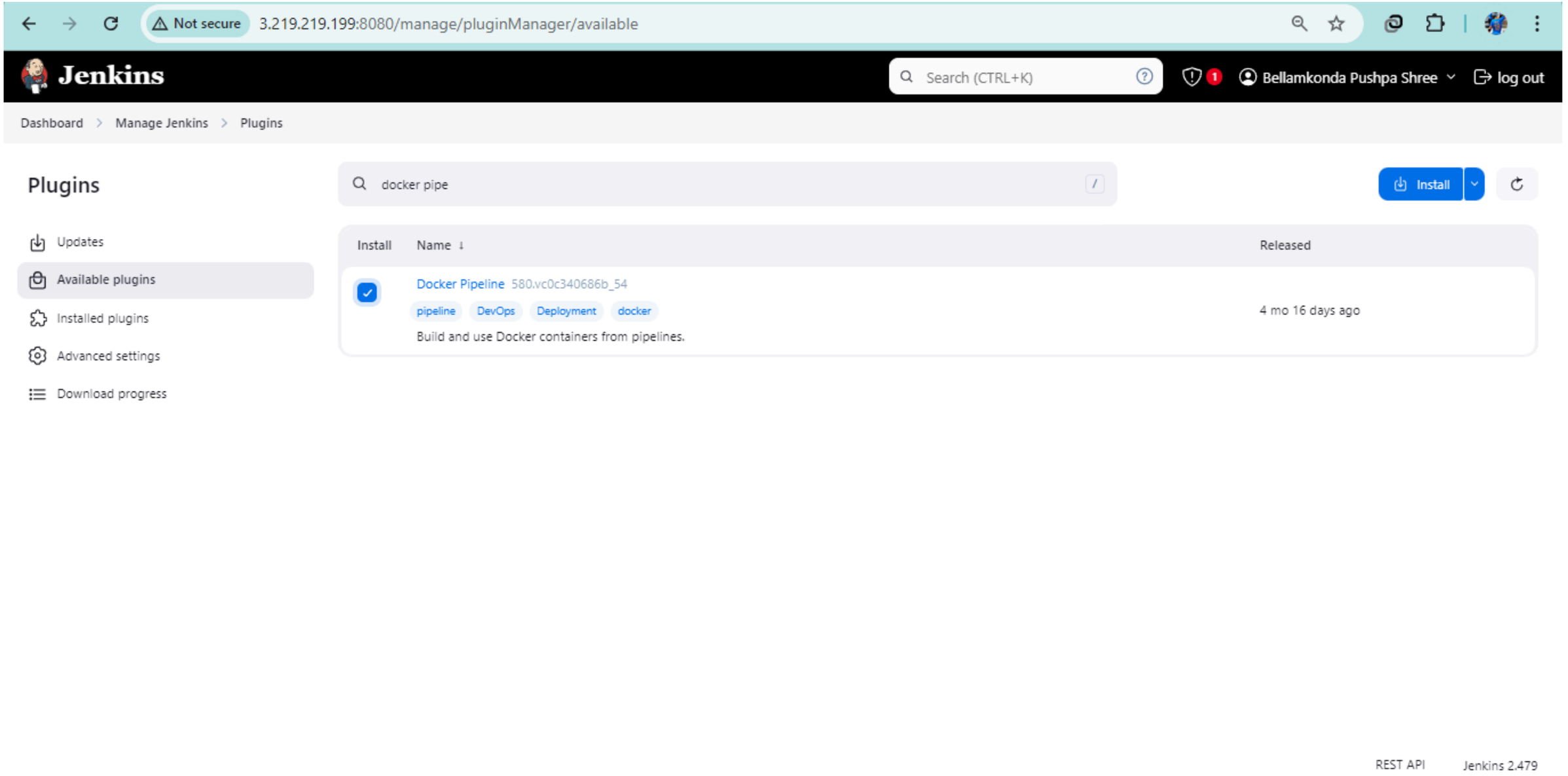
java-maven-sonar-argocd-helm-k8s/spring-boot-app/JenkinsFile

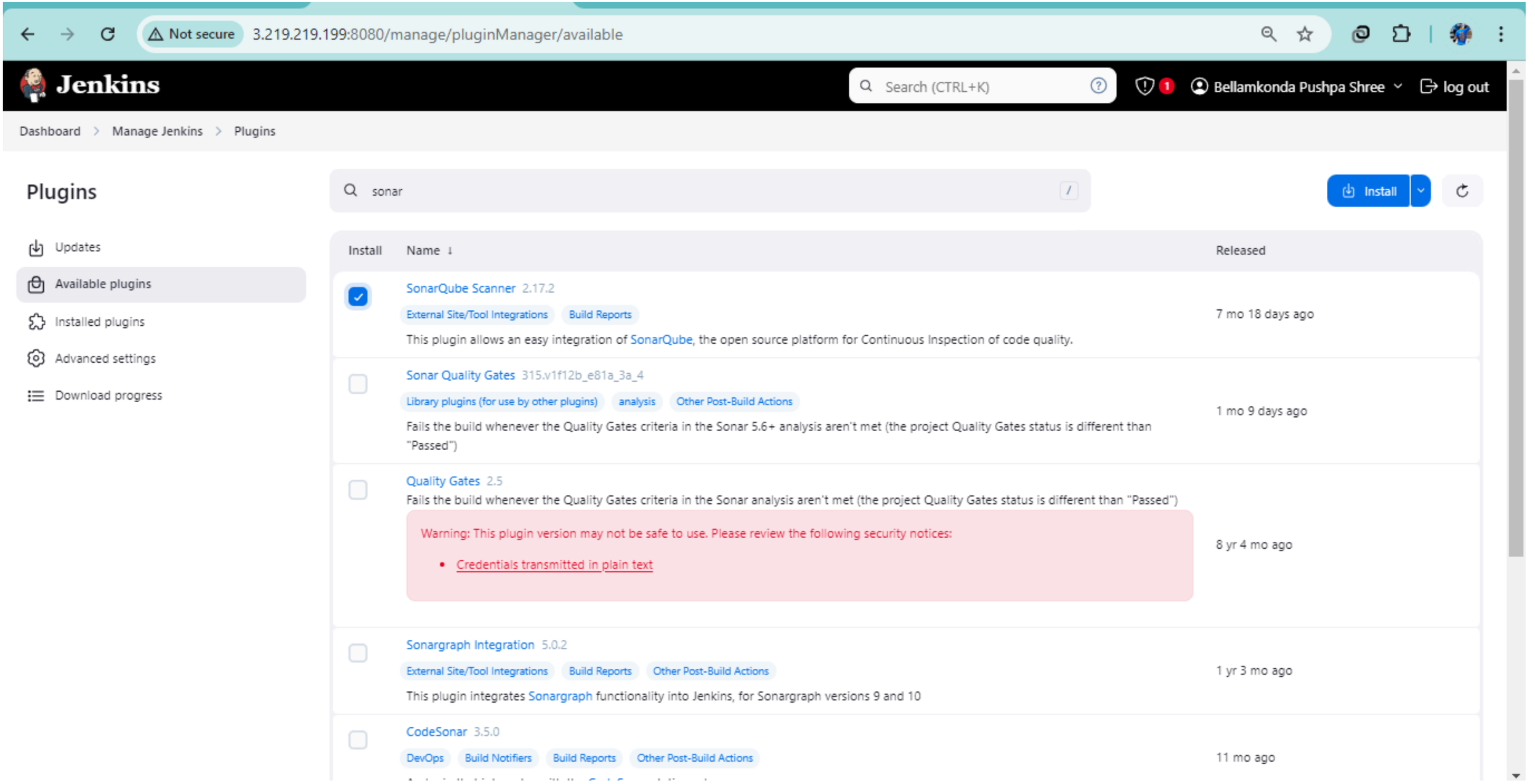
Lightweight checkout ?

Pipeline Syntax

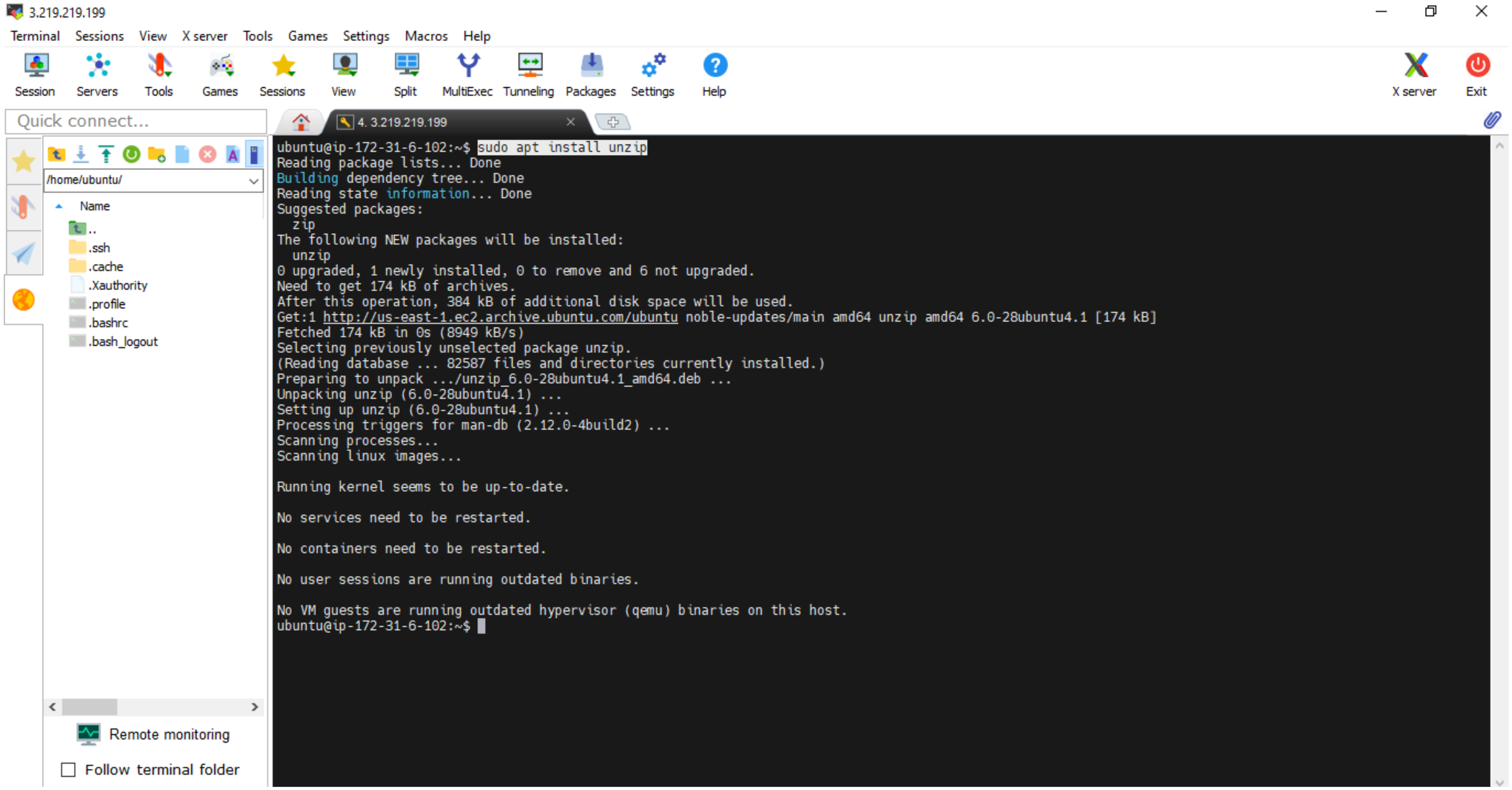
Save Apply

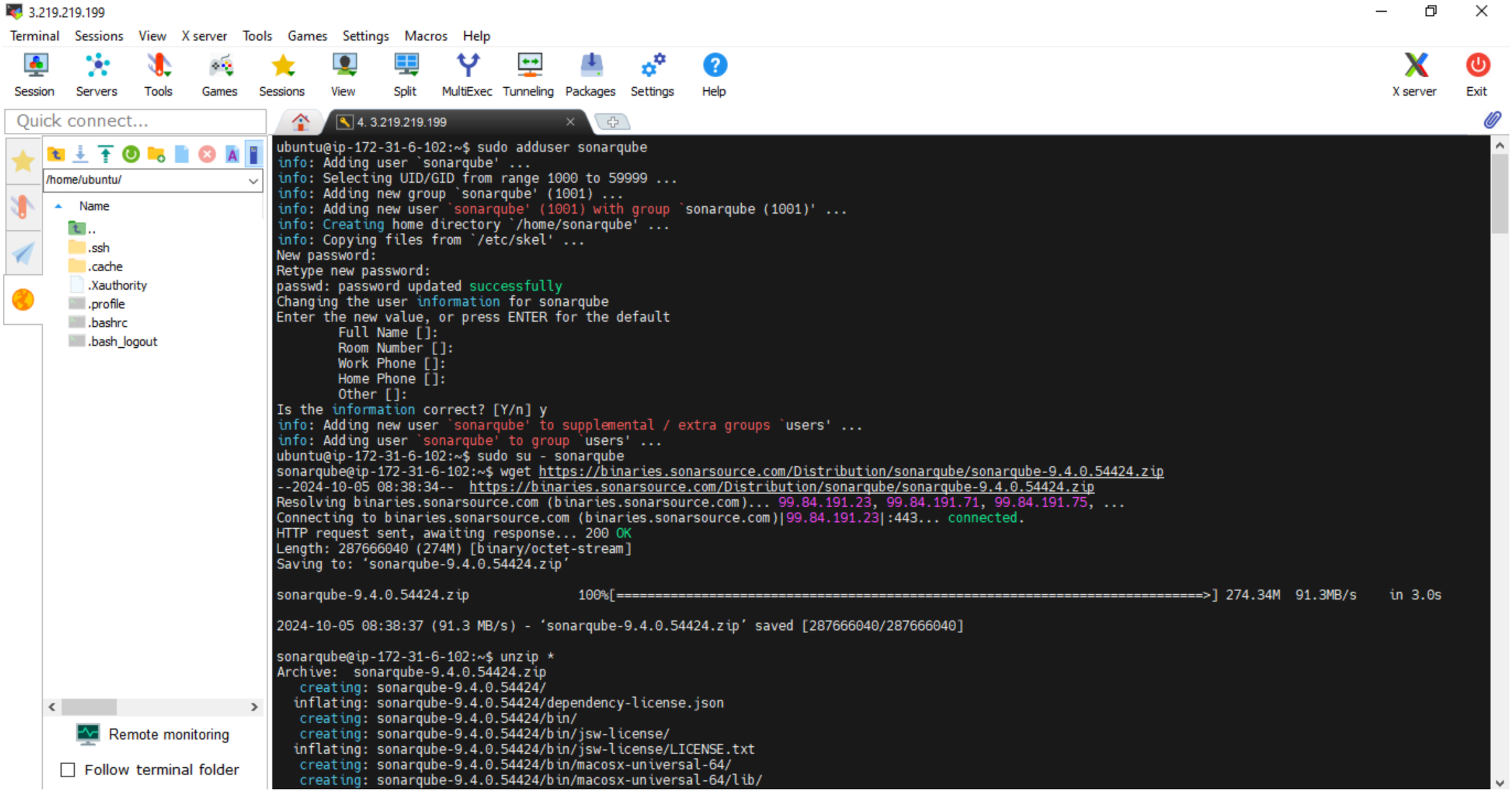
step:10 Install required plugins , " Docker Pipeline " and " Sonarqube Scanner "

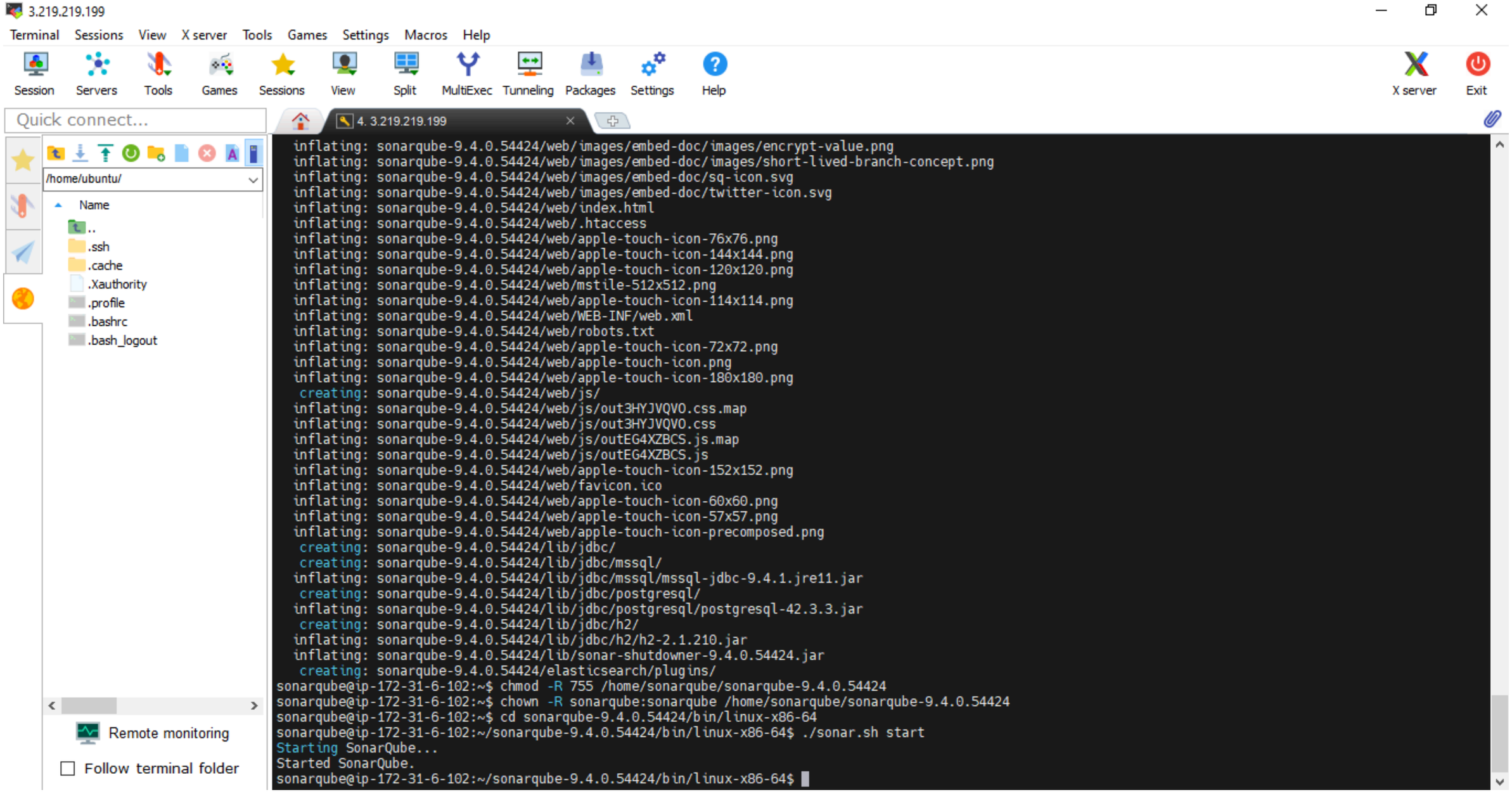



Not secure
3.219.219.199:8080/manage/pluginManager/available
Jenkins
Search (CTRL+K)
Bellamkonda Pushpa Shree
log out
Dashboard > Manage Jenkins > Plugins
Plugins
sonar
Install
Updates
Available plugins
Installed plugins
Advanced settings
Download progress
Install
Name
Released
SonarQube Scanner 2.17.2
External Site/Tool Integrations
Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
7 mo 18 days ago
Sonar Quality Gates 315.v1f12b_e81a_3a_4
Library plugins (for use by other plugins)
analysis
Other Post-Build Actions
Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed")
1 mo 9 days ago
Quality Gates 2.5
Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")
Warning: This plugin version may not be safe to use. Please review the following security notices:
Credentials transmitted in plain text
8 yr 4 mo ago
Sonargraph Integration 5.0.2
External Site/Tool Integrations
Build Reports
Other Post-Build Actions
This plugin integrates Sonargraph functionality into Jenkins, for Sonargraph versions 9 and 10
1 yr 3 mo ago
CodeSonar 3.5.0
DevOps
Build Notifiers
Build Reports
Other Post-Build Actions
11 mo ago

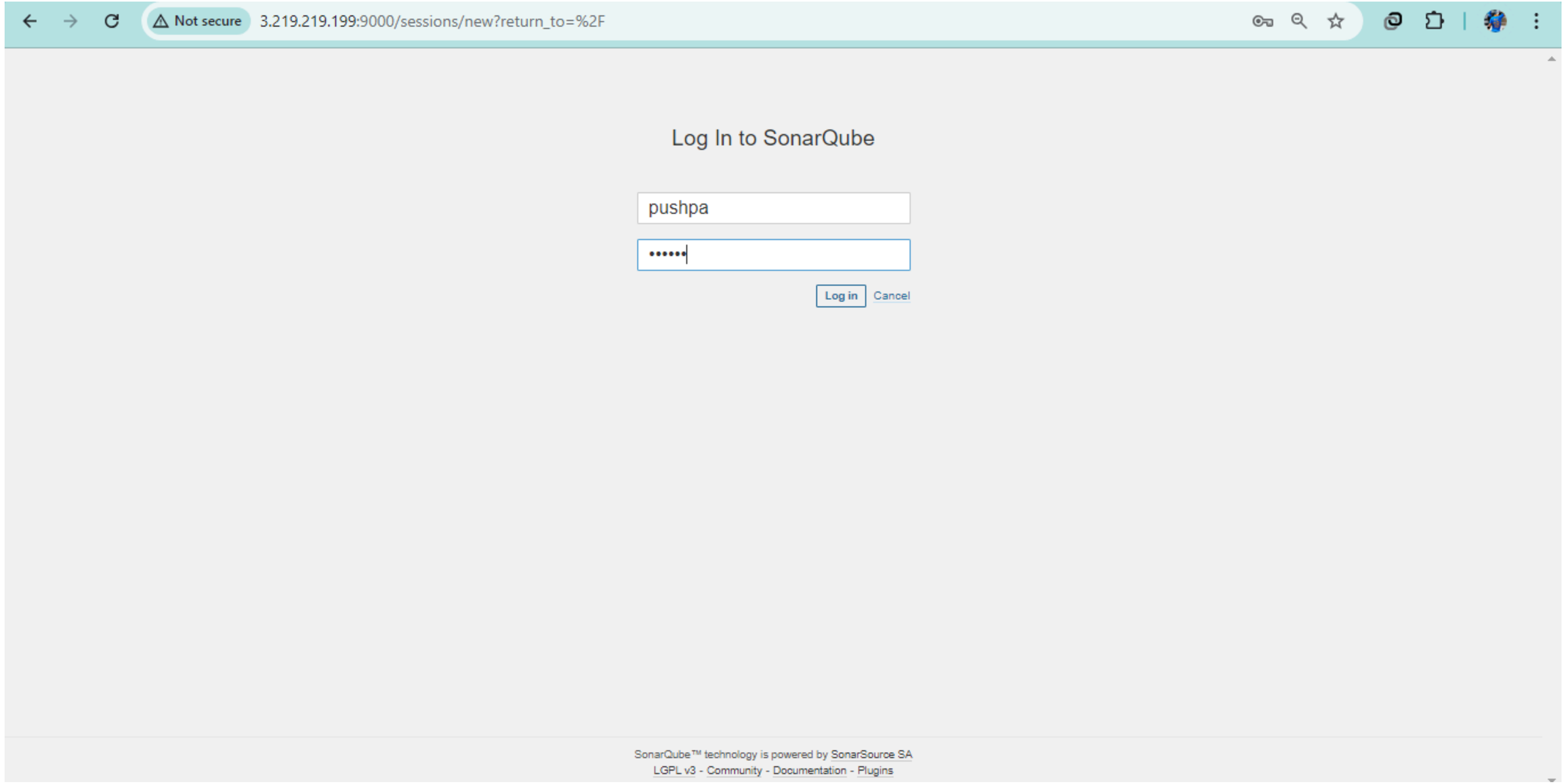
step:11 Install sonarqube in the ec2 instance



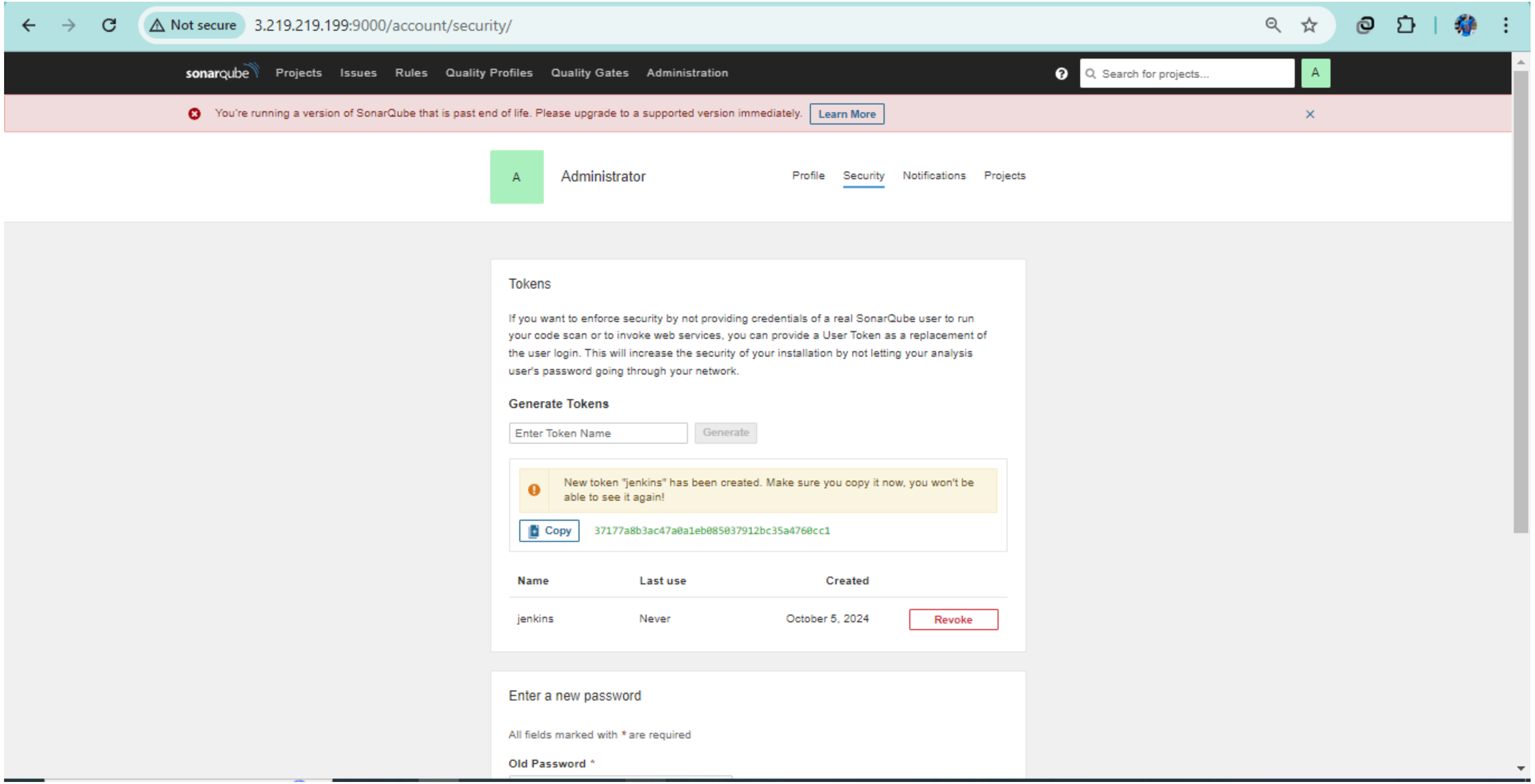

3.219.219.199
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
4. 3.219.219.199
/home/ubuntu/
Name
..
.ssh
.cache
.Xauthority
.profile
.bashrc
.bash_logout
Remote monitoring
Follow terminal folder
ubuntu@ip-172-31-6-102:~$ sudo adduser sonarqube
info: Adding user `sonarqube' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `sonarqube' (1001) ...
info: Adding new user `sonarqube' (1001) with group `sonarqube (1001)' ...
info: Creating home directory `/home/sonarqube' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sonarqube
Enter the new value, or press ENTER for the default
Full Name []:
Room Number []:
Work Phone []:
Home Phone []:
Other []:
Is the information correct? [Y/n] y
info: Adding new user `sonarqube' to supplemental / extra groups `users' ...
info: Adding user `sonarqube' to group `users' ...
ubuntu@ip-172-31-6-102:~$ sudo su - sonarqube
sonarqube@ip-172-31-6-102:~$ wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip
--2024-10-05 08:38:34-- https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-9.4.0.54424.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.191.23, 99.84.191.71, 99.84.191.75, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|99.84.191.23|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 287666040 (274M) [binary/octet-stream]
Saving to: 'sonarqube-9.4.0.54424.zip'
sonarqube-9.4.0.54424.zip 100%[==========>] 274.34M 91.3MB/s in 3.0s
2024-10-05 08:38:37 (91.3 MB/s) - 'sonarqube-9.4.0.54424.zip' saved [287666040/287666040]
sonarqube@ip-172-31-6-102:~$ unzip *
Archive: sonarqube-9.4.0.54424.zip
creating: sonarqube-9.4.0.54424/
inflating: sonarqube-9.4.0.54424/dependency-license.json
creating: sonarqube-9.4.0.54424/bin/
creating: sonarqube-9.4.0.54424/bin/jsw-license/
inflating: sonarqube-9.4.0.54424/bin/jsw-license/LICENSE.txt
creating: sonarqube-9.4.0.54424/bin/macosx-universal-64/
creating: sonarqube-9.4.0.54424/bin/macosx-universal-64/lib/


3.219.219.199
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
Quick connect...
4. 3.219.219.199
/home/ubuntu/
Name
..
.ssh
.cache
.Xauthority
.profile
.bashrc
.bash_logout
Remote monitoring
Follow terminal folder
inflating: sonarqube-9.4.0.54424/web/images/embed-doc/images/encrypt-value.png
inflating: sonarqube-9.4.0.54424/web/images/embed-doc/images/short-lived-branch-concept.png
inflating: sonarqube-9.4.0.54424/web/images/embed-doc/sq-icon.svg
inflating: sonarqube-9.4.0.54424/web/images/embed-doc/twitter-icon.svg
inflating: sonarqube-9.4.0.54424/web/index.html
inflating: sonarqube-9.4.0.54424/web/.htaccess
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-76x76.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-144x144.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-120x120.png
inflating: sonarqube-9.4.0.54424/web/mstile-512x512.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-114x114.png
inflating: sonarqube-9.4.0.54424/web/WEB-INF/web.xml
inflating: sonarqube-9.4.0.54424/web/robots.txt
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-72x72.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-180x180.png
creating: sonarqube-9.4.0.54424/web/js/
inflating: sonarqube-9.4.0.54424/web/js/out3HYJVQVO.css.map
inflating: sonarqube-9.4.0.54424/web/js/out3HYJVQVO.css
inflating: sonarqube-9.4.0.54424/web/js/outEG4XZBCS.js.map
inflating: sonarqube-9.4.0.54424/web/js/outEG4XZBCS.js
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-152x152.png
inflating: sonarqube-9.4.0.54424/web/favicon.ico
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-60x60.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-57x57.png
inflating: sonarqube-9.4.0.54424/web/apple-touch-icon-precomposed.png
creating: sonarqube-9.4.0.54424/lib/jdbc/
creating: sonarqube-9.4.0.54424/lib/jdbc/mssql/
inflating: sonarqube-9.4.0.54424/lib/jdbc/mssql/mssql-jdbc-9.4.1.jre11.jar
creating: sonarqube-9.4.0.54424/lib/jdbc/postgresql/
inflating: sonarqube-9.4.0.54424/lib/jdbc/postgresql/postgresql-42.3.3.jar
creating: sonarqube-9.4.0.54424/lib/jdbc/h2/
inflating: sonarqube-9.4.0.54424/lib/jdbc/h2/h2-2.1.210.jar
inflating: sonarqube-9.4.0.54424/lib/sonar-shutdowner-9.4.0.54424.jar
creating: sonarqube-9.4.0.54424/elasticsearch/plugins/
sonarqube@ip-172-31-6-102:~$ chmod -R 755 /home/sonarqube/sonarqube-9.4.0.54424
sonarqube@ip-172-31-6-102:~$ chown -R sonarqube:sonarqube /home/sonarqube/sonarqube-9.4.0.54424
sonarqube@ip-172-31-6-102:~$ cd sonarqube-9.4.0.54424/bin/linux-x86-64
sonarqube@ip-172-31-6-102:~/sonarqube-9.4.0.54424/bin/linux-x86-64$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
sonarqube@ip-172-31-6-102:~/sonarqube-9.4.0.54424/bin/linux-x86-64$

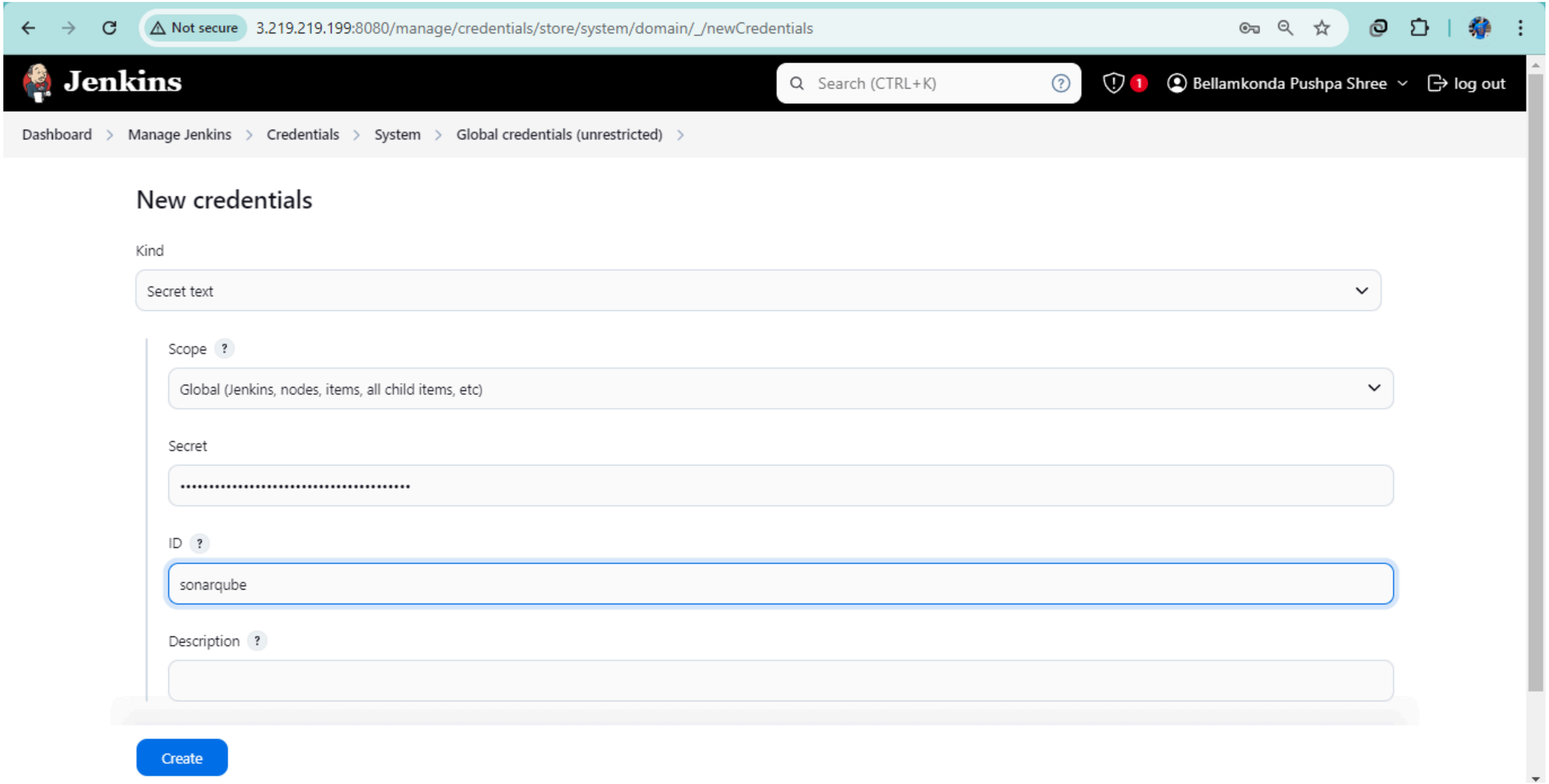
step:12 Now browse the public IP of instance with port 9000 to access sonarqube and login to it



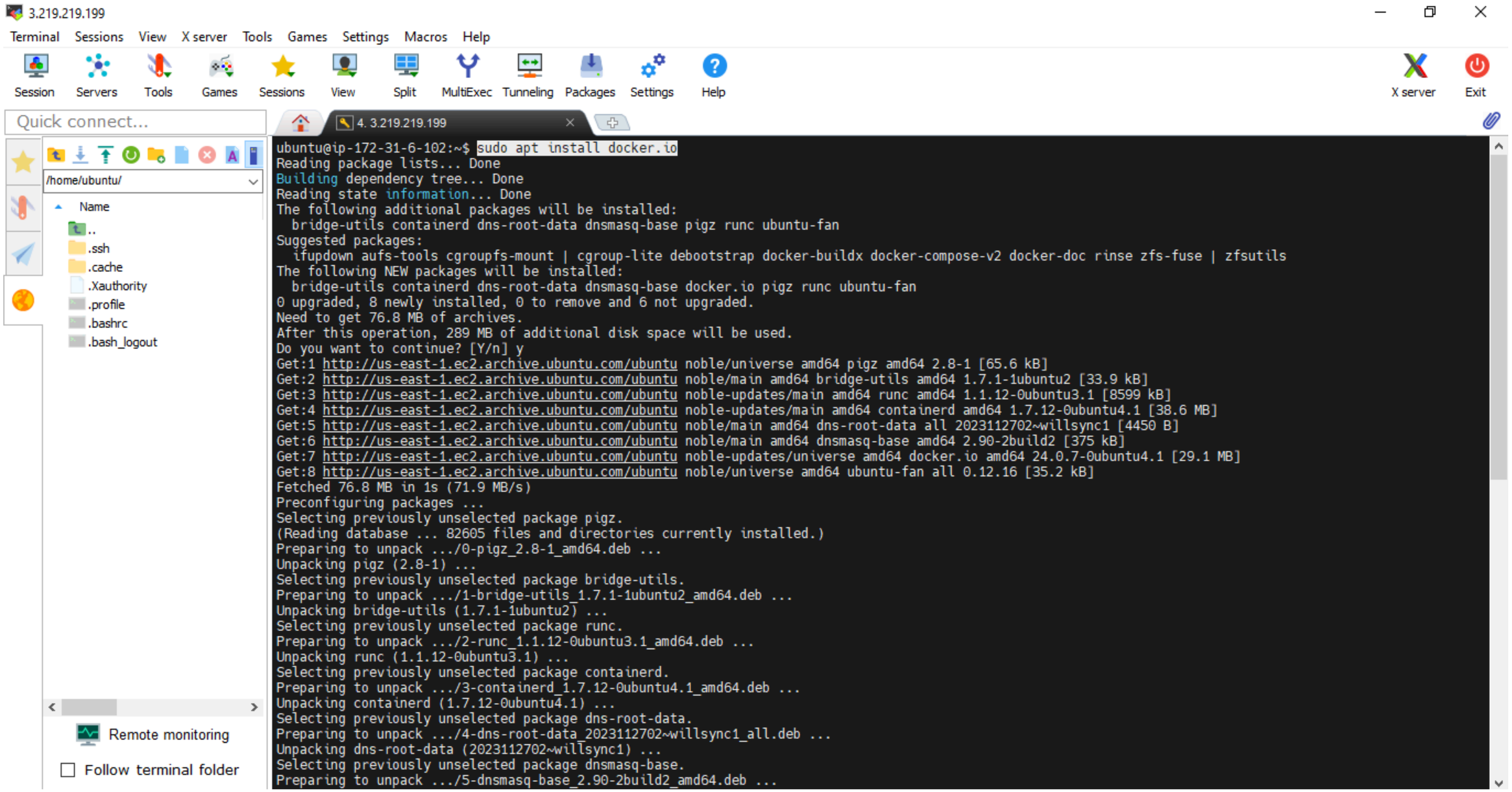
step:13 Now create a token and by using this token we can make jenkins to interact with sonarqube



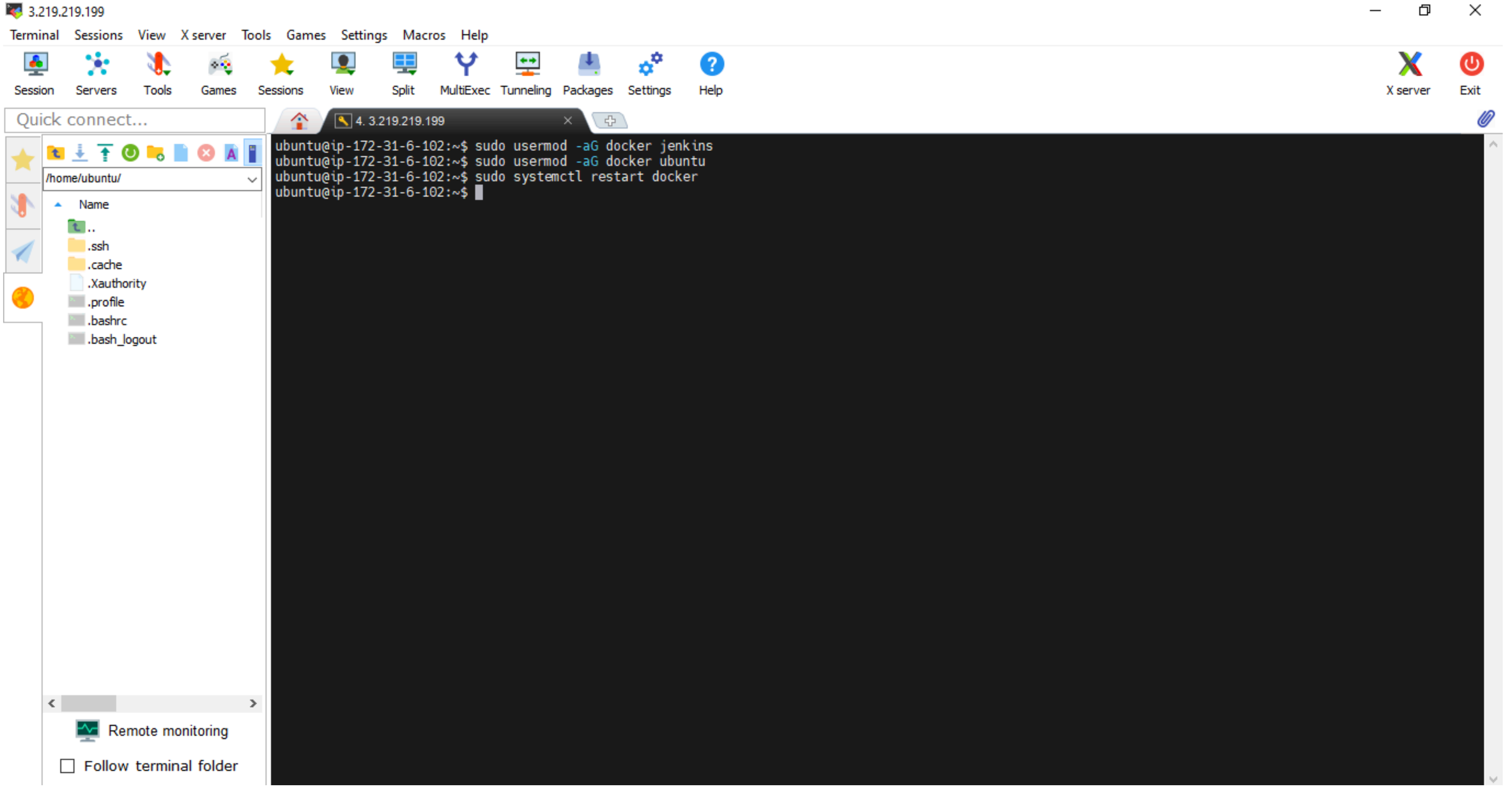
step:14 Now in jenkins create sonarqube credentials in order to make both jenkins and sonarqube interact with eachother



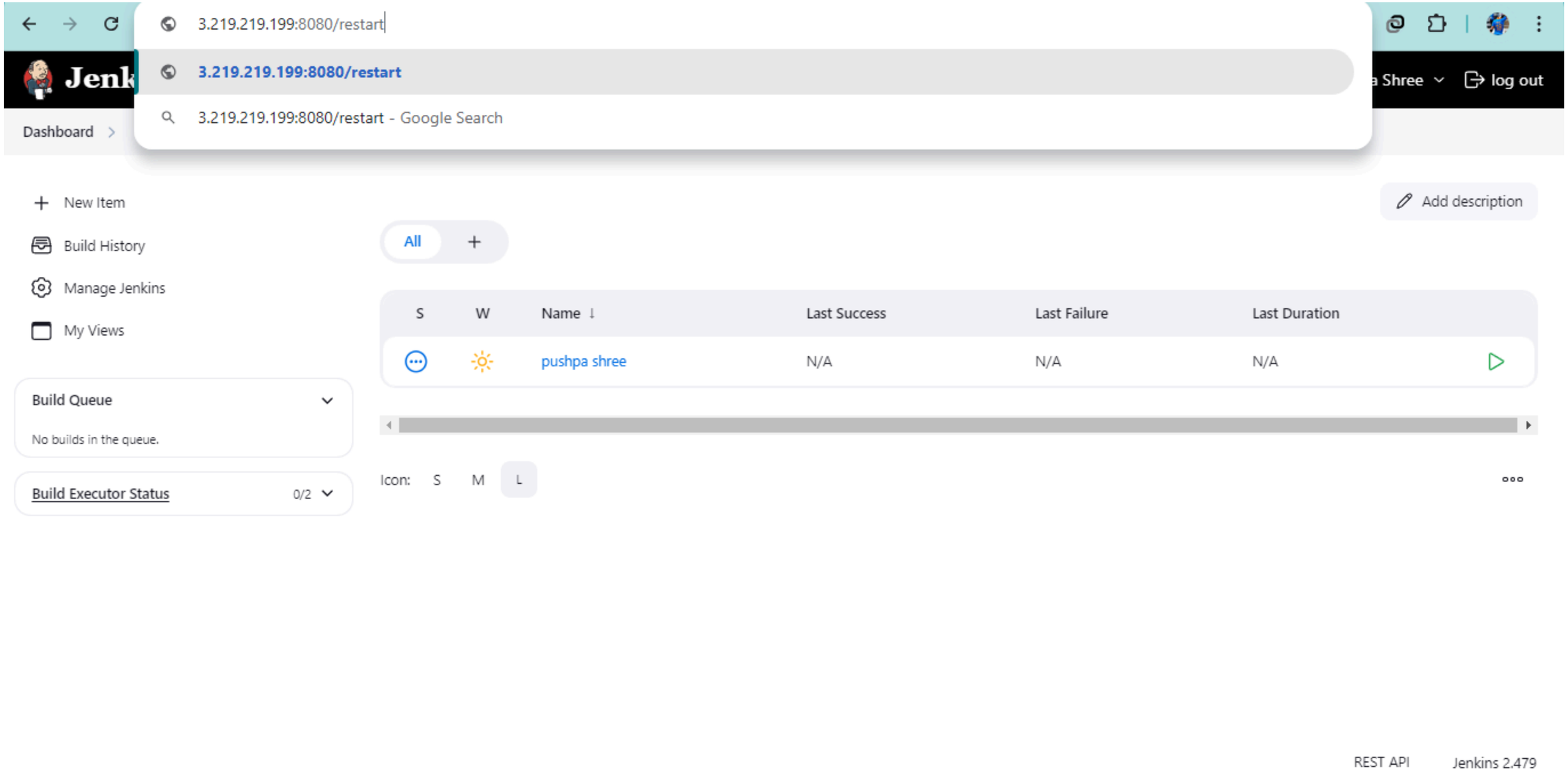
step:15 Now install Docker inside ec2 instance



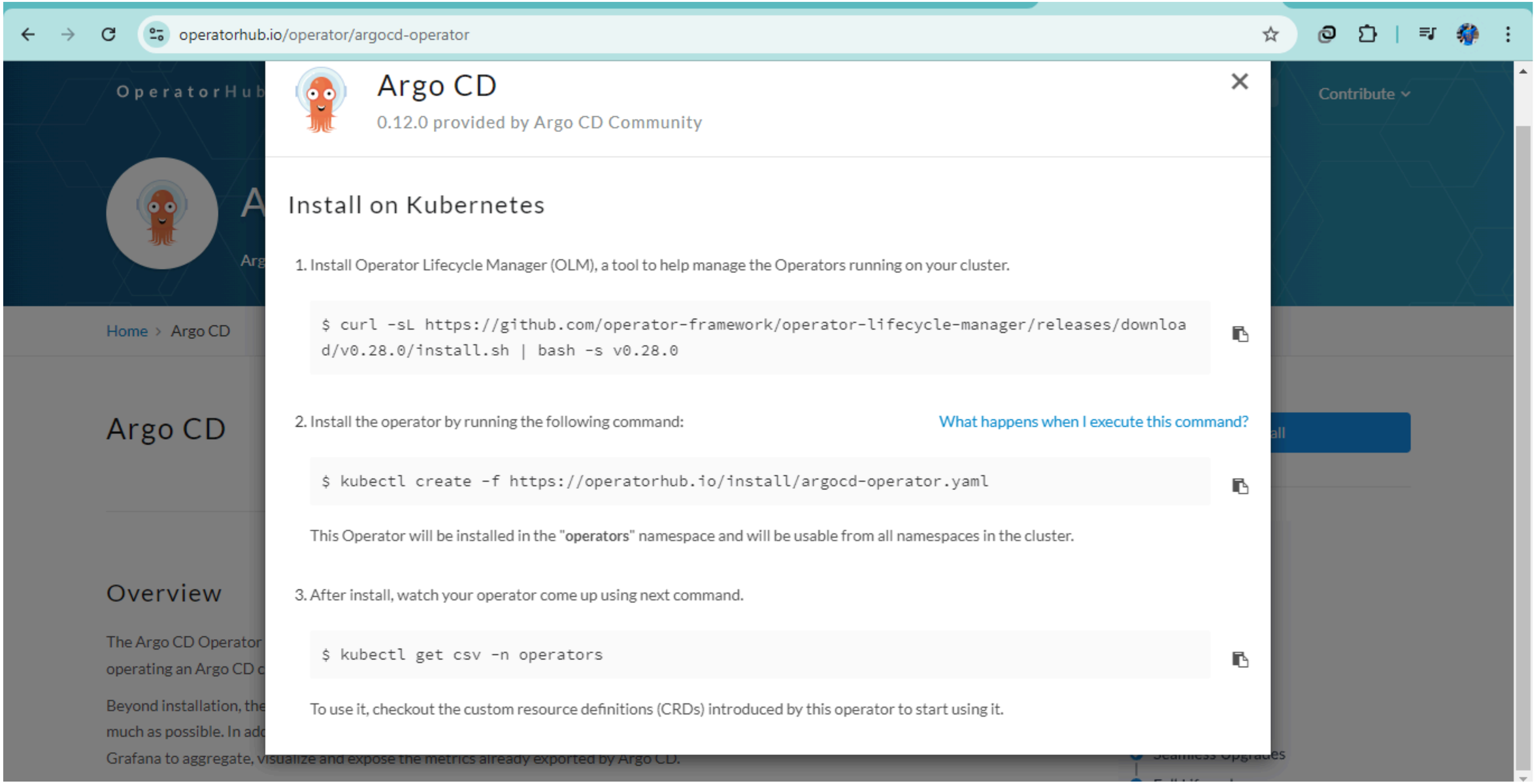
Now run these commands to give previleges and restart docker



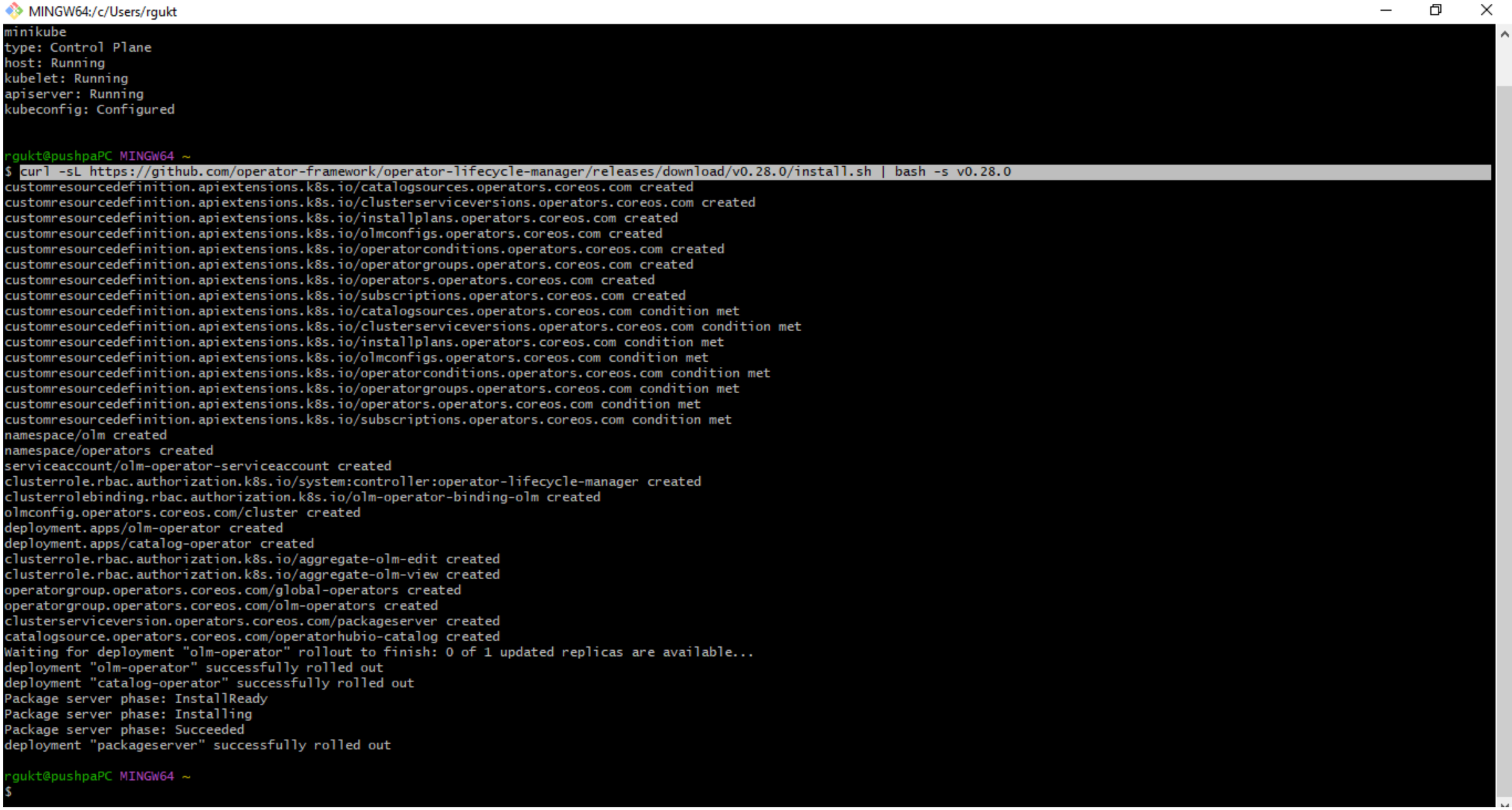
Now restart jenkins



Now browse "operatorhub argocd install" to install argocd operator



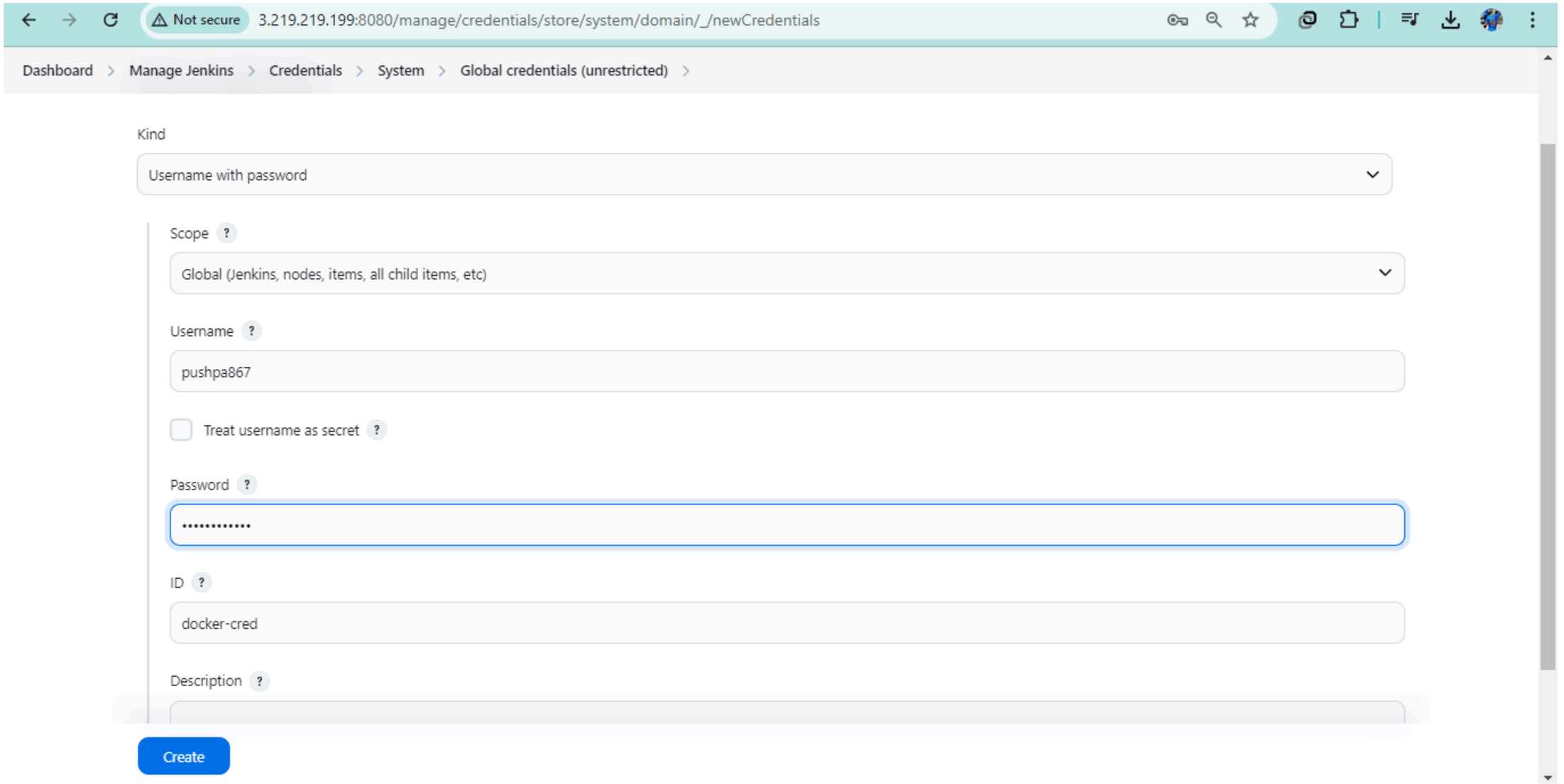
Now run those commands one by one



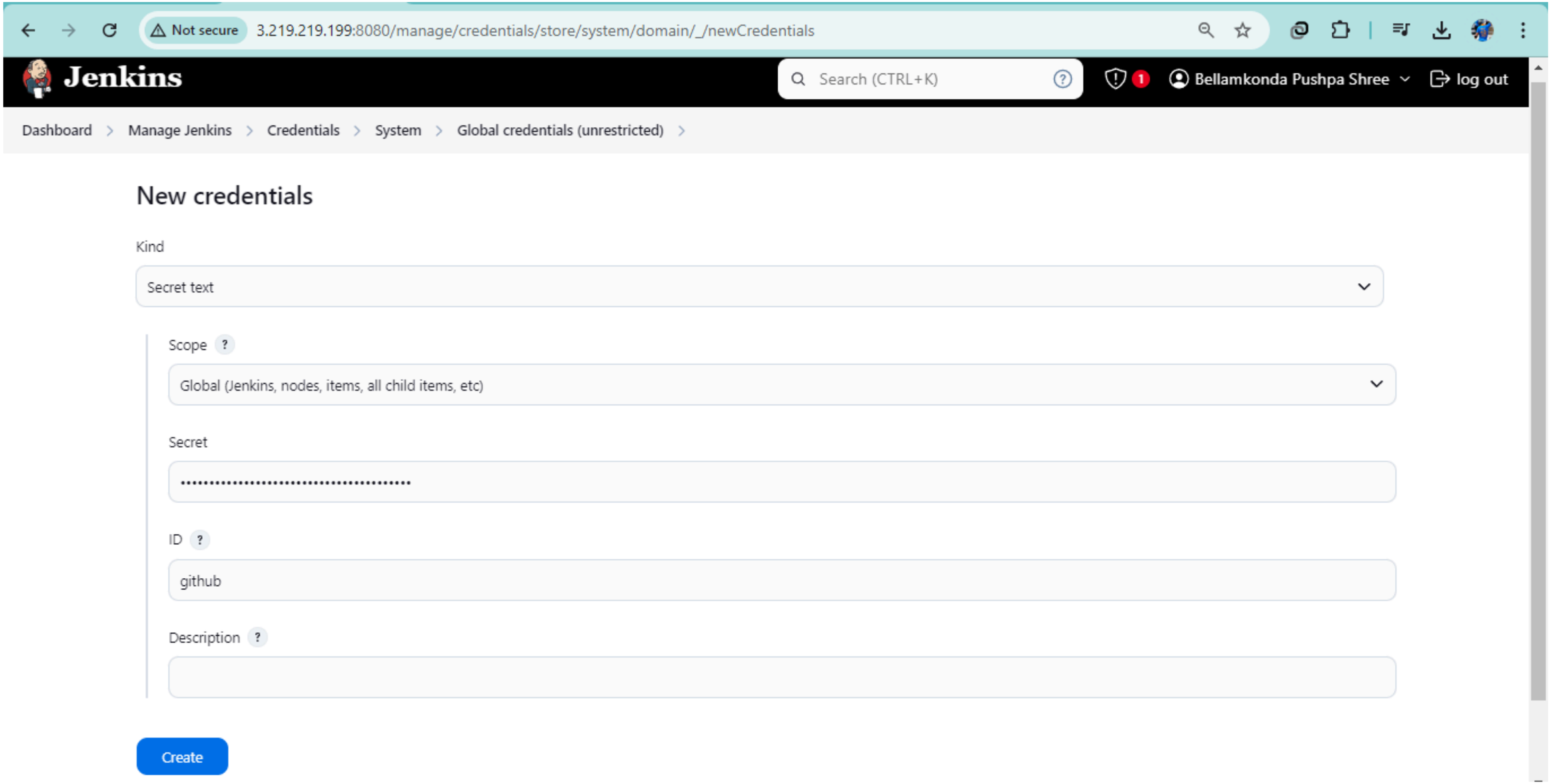
```
rgukt@pushpaPC MINGW64 ~
$ kubectl create -f https://operatorhub.io/install/argocd-operator.yaml
subscription.operators.coreos.com/my-argocd-operator created

rgukt@pushpaPC MINGW64 ~
$
```

```
rgukt@pushpaPC MINGW64 ~
$ kubectl get csv -n operators
NAME                         DISPLAY   VERSION   REPLACES                     PHASE
argocd-operator.v0.12.0      Argo CD   0.12.0    argocd-operator.v0.11.0      Installing

rgukt@pushpaPC MINGW64 ~
$
```

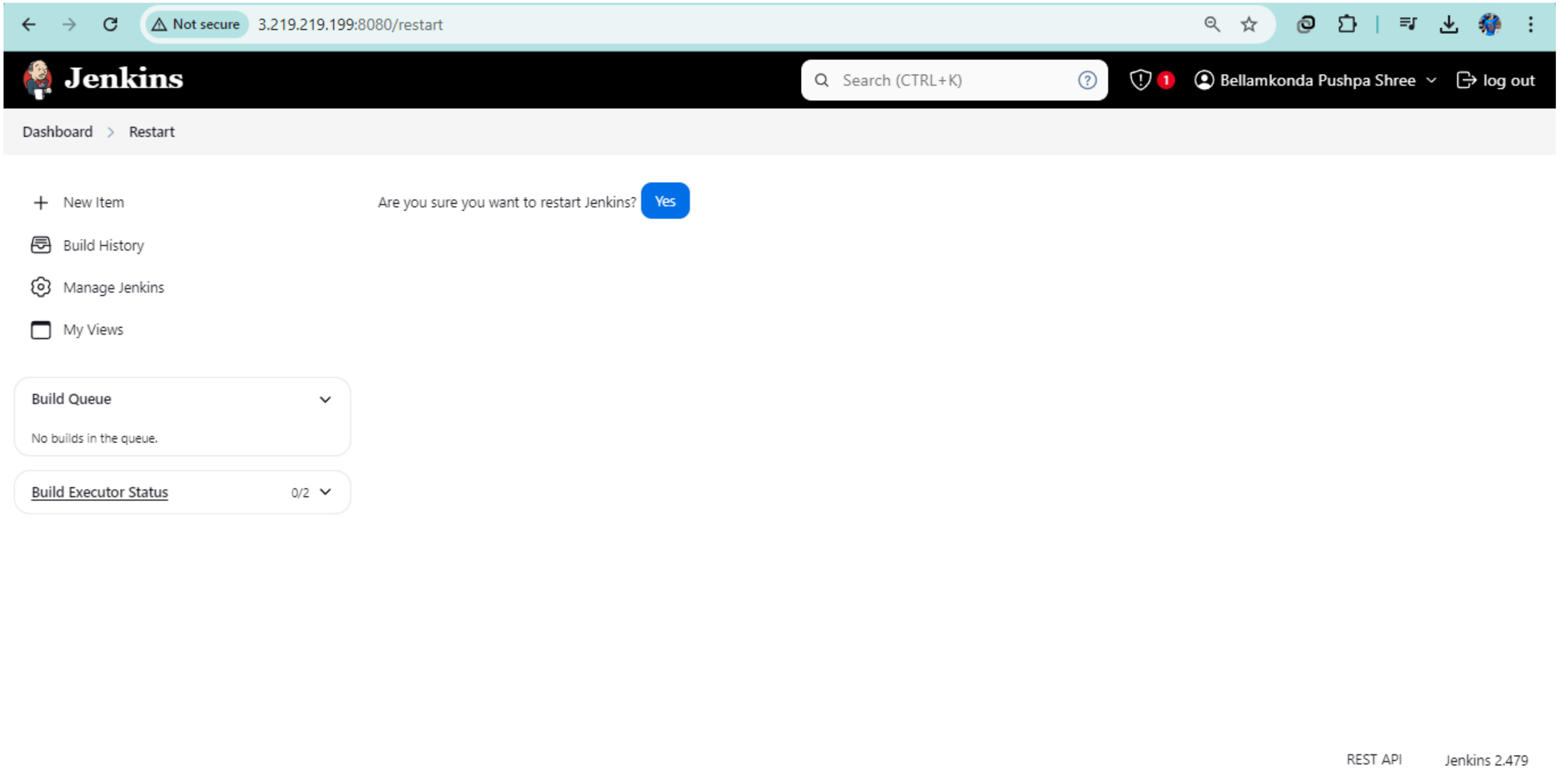
Now add the dockerhub credentials in the jenkins so that the docker image will be stored inside the dockerhub of the respective repository that we gave in the pipeline syntax



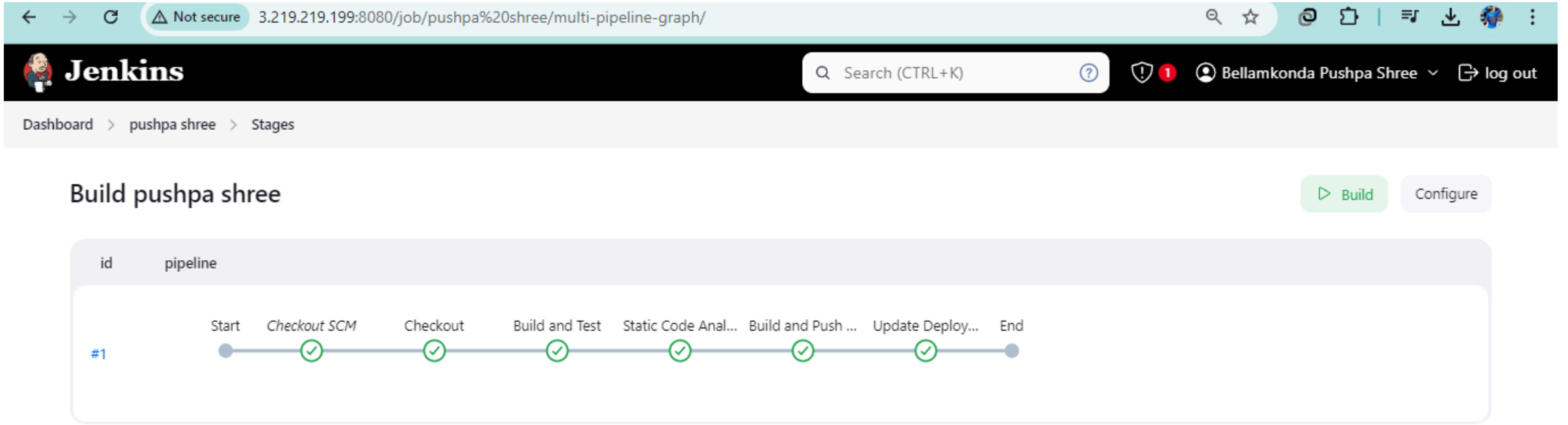
Now provide the github credentials in the jenkins , create a token in the github



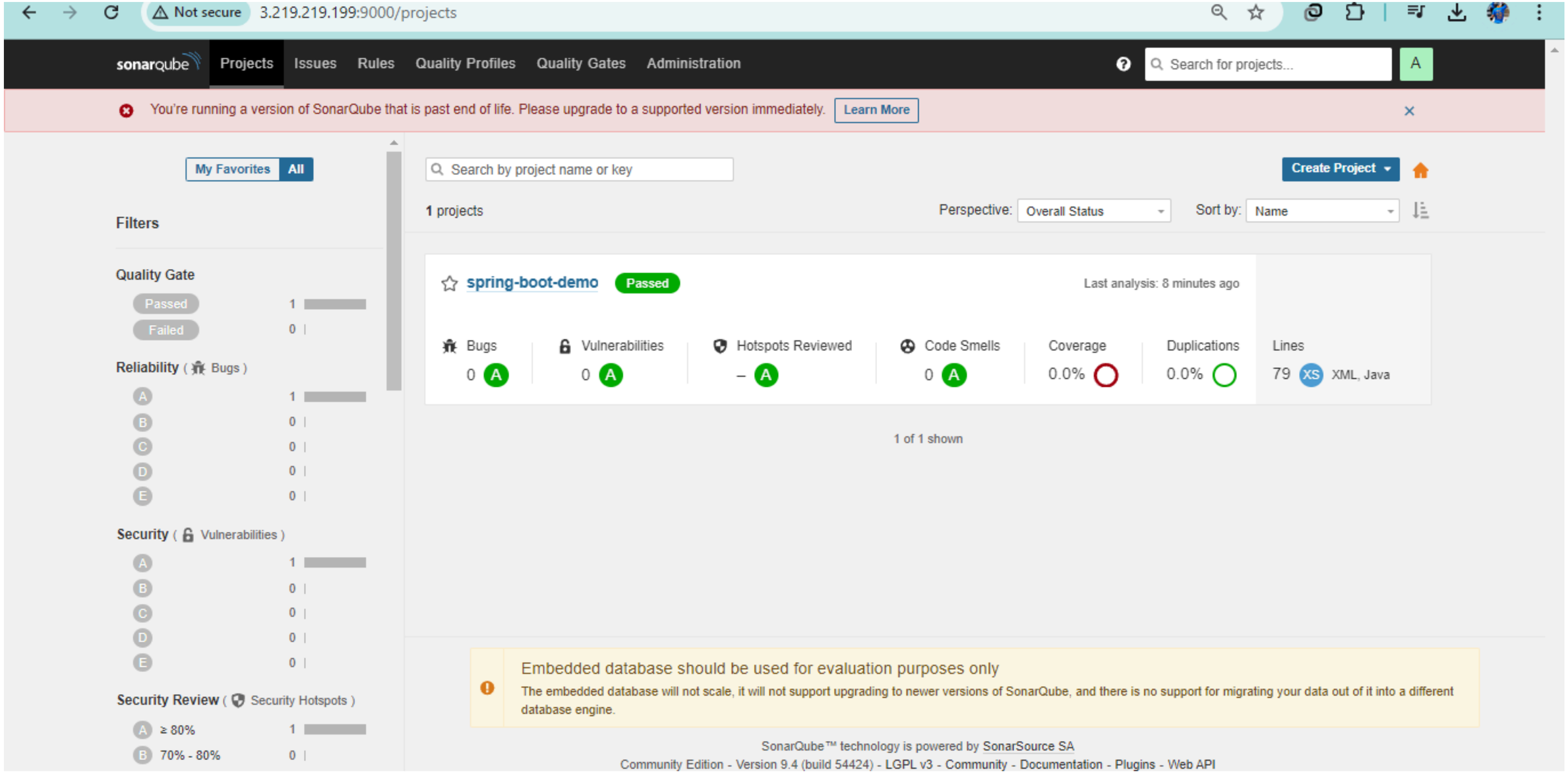
Now again restart the jenkins interface



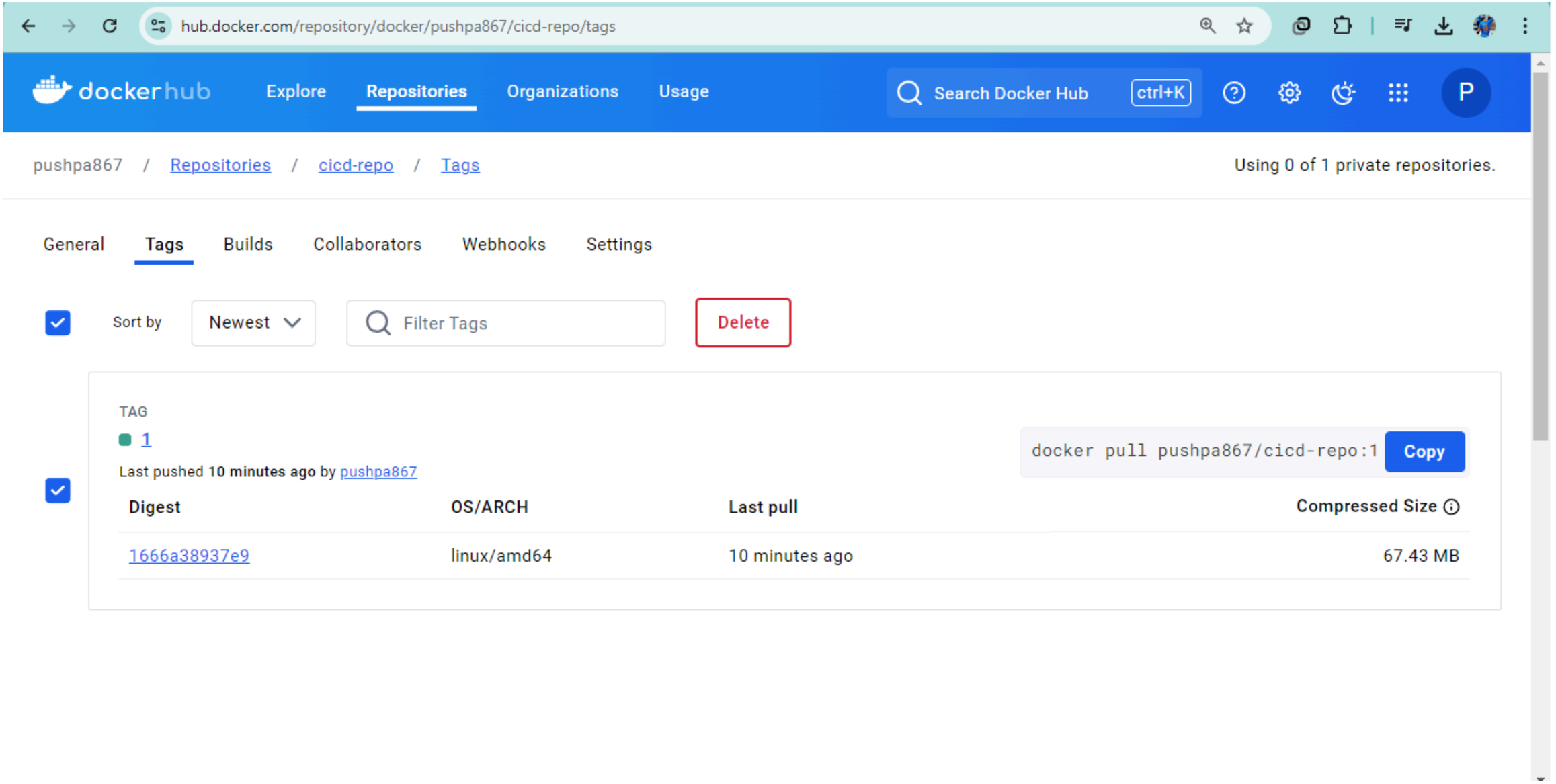
step:16 Now build the pipeline



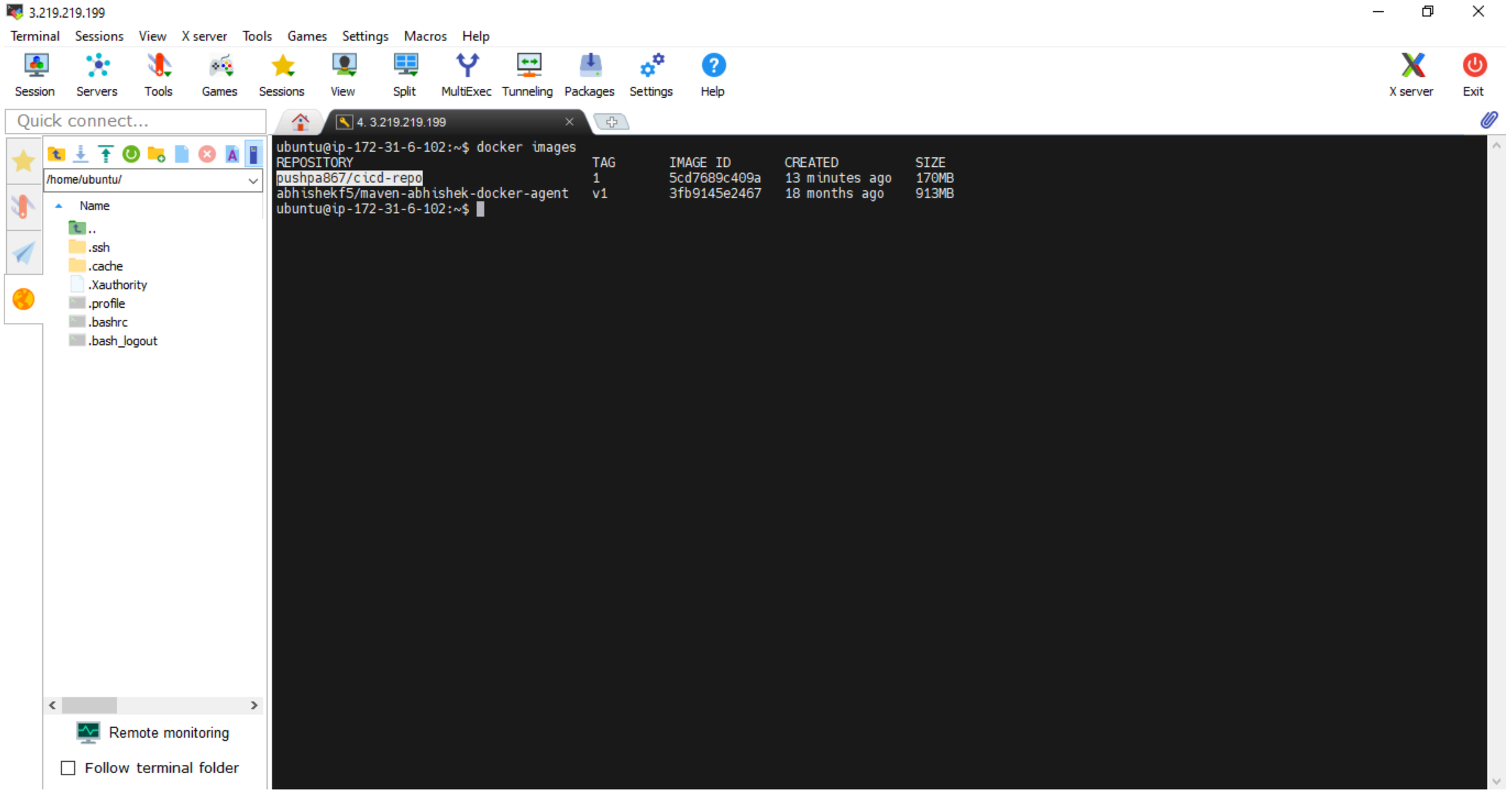
We can observe that our application is deployed in the sonarqube server



And our docker image is also pushed inside our dockerhub repository



Run this command to see the docker image in the ec2 instance



Steps for Continuous Deployment

step:1 create a new Argo CD cluster with the default configuration

-Create a yaml file called "argocd-basic.yaml" which is used to create argocd cluster

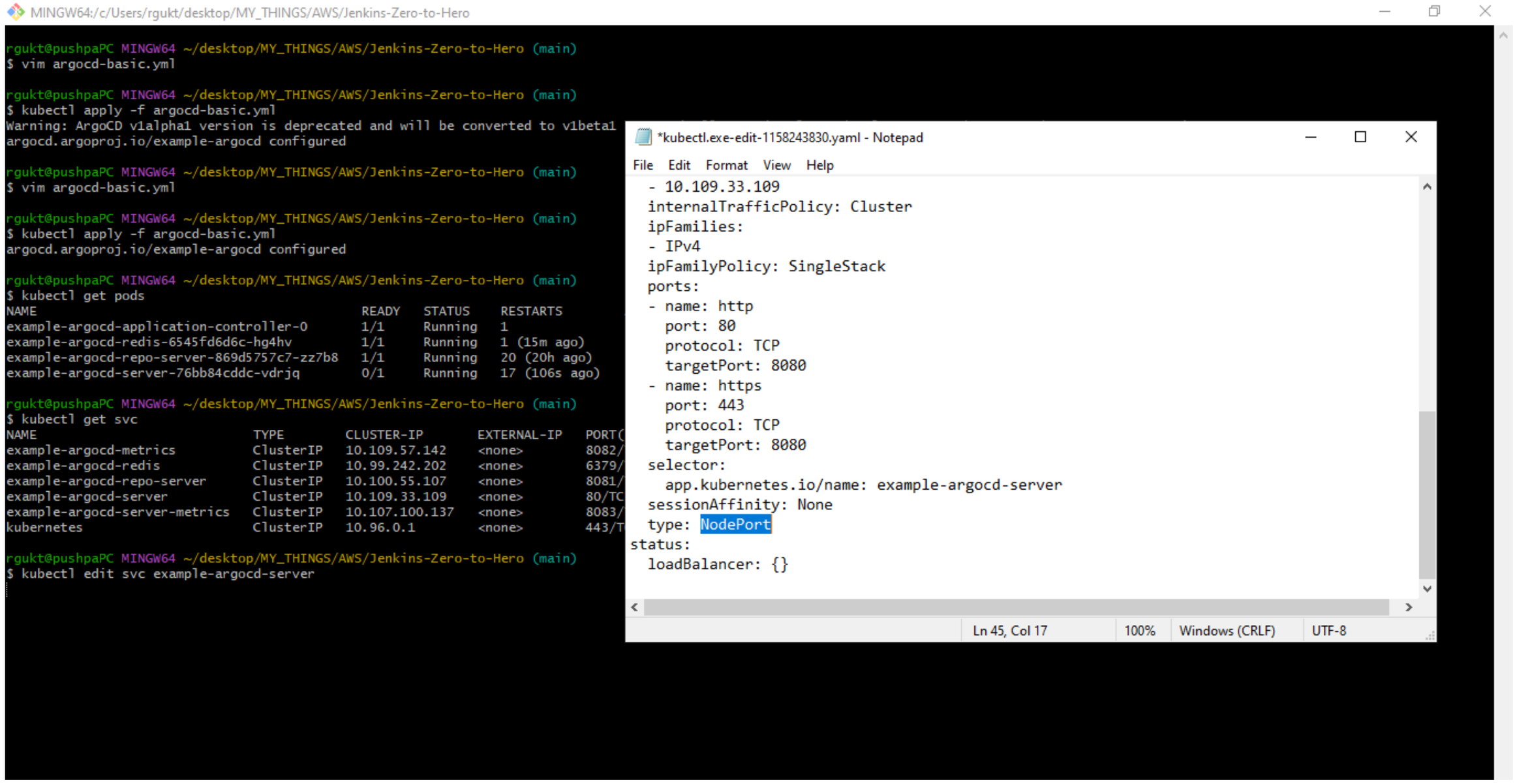
-Configure this yaml file i.e we are giving previliges to the file

-We can observe that our "argocd-basic" service is in "ClusterIP" mode , it means only people who has access with cluster can access this service so to make it accessable to our friends or collegues we will make it into "NodePort" mode

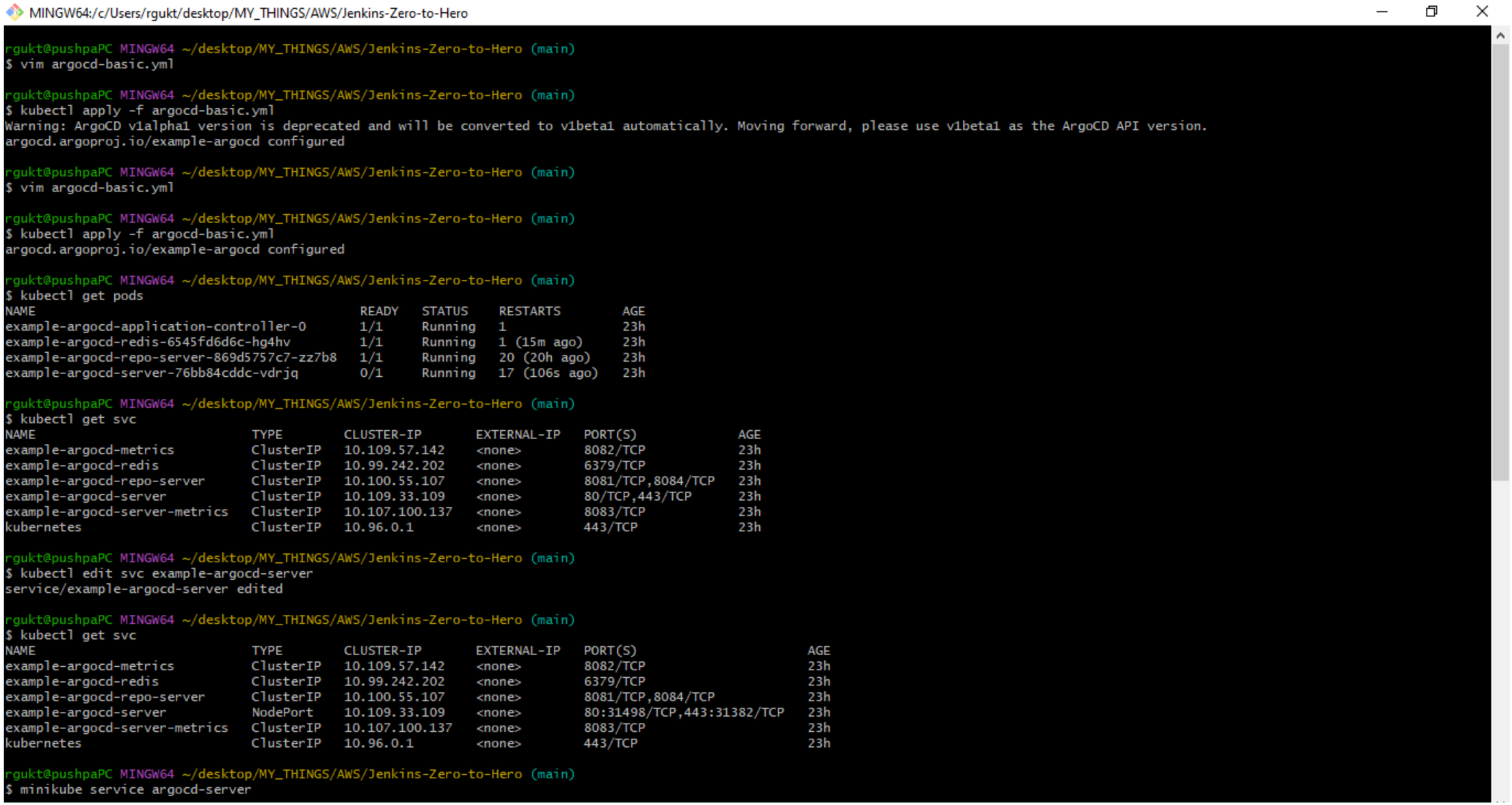
-For that

kubectl edit svc example-argocd-server

**Now in the editor , change the type from ClusterIP to "NodePort"

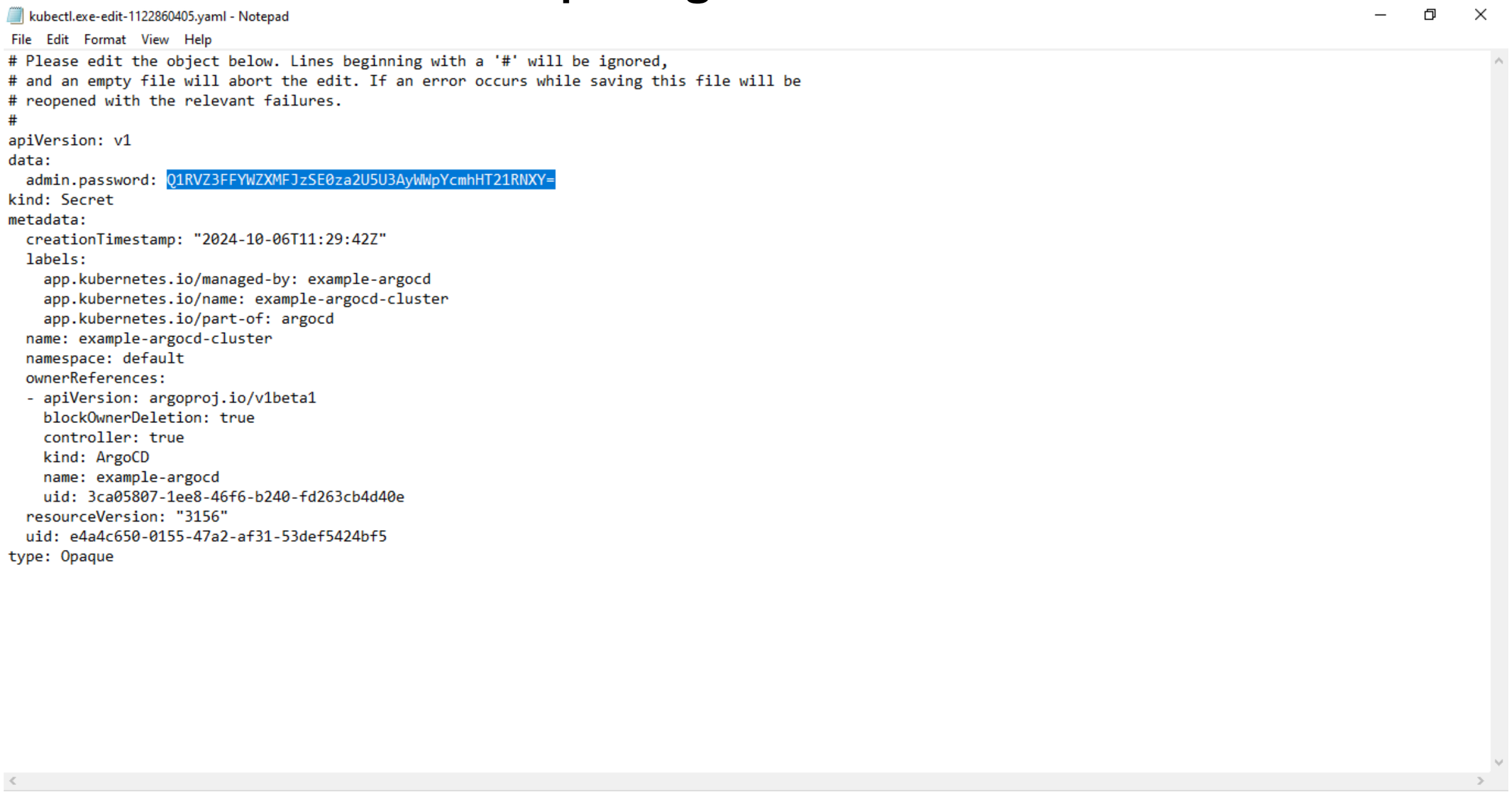


Now we can observe that our service is changed to NodePort mode



Run the following command to get secret code in the editor

kubectl edit secret example-argocd-server



Now use the command to get the encrypted code of this secret

echo "your_code" | base64 -d

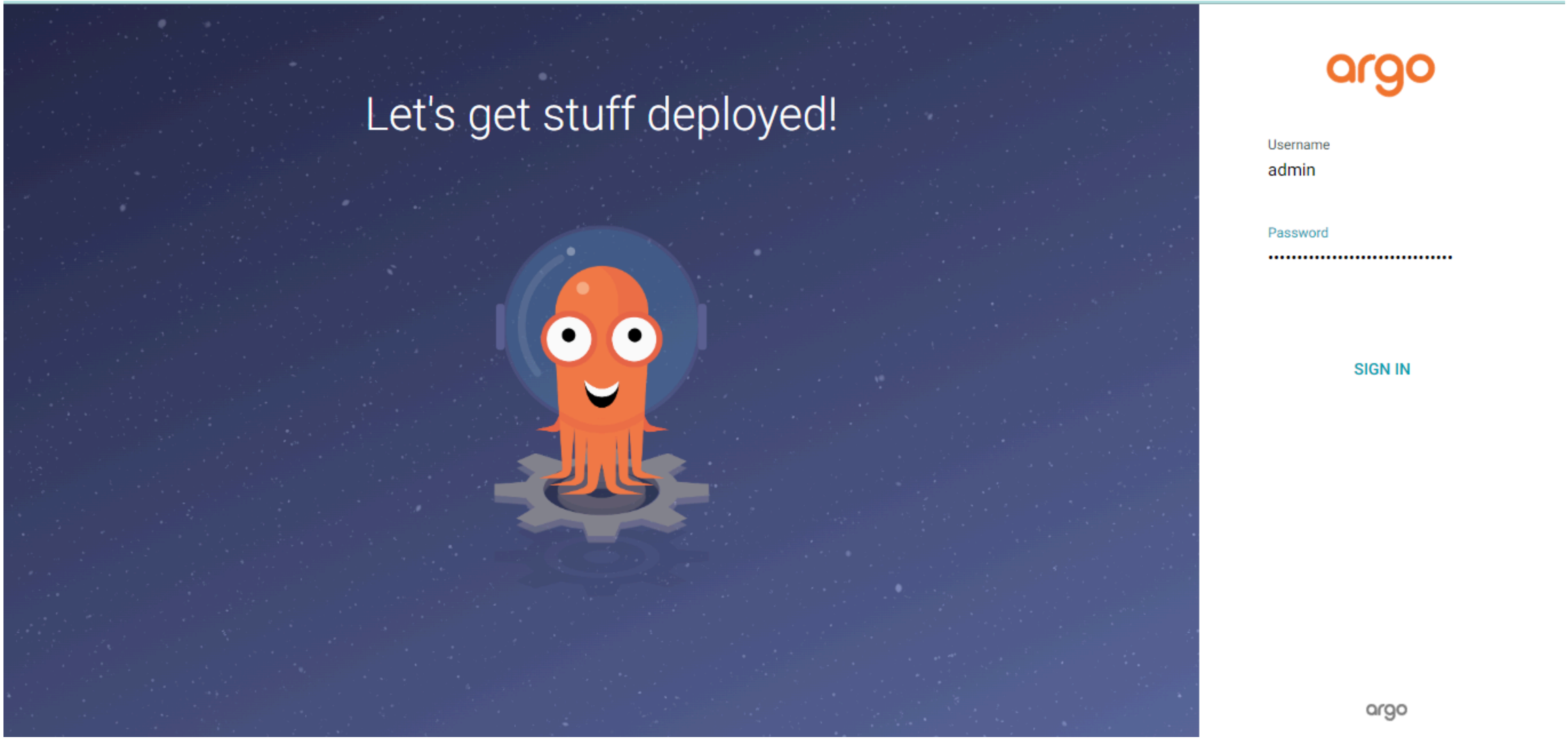
```
rgukt@pushpaPC MINGW64 ~/desktop/MY_THINGS/AWS/Jenkins-Zero-to-Hero (main)
$ kubectl edit secret example-argocd-cluster
Edit cancelled, no changes made.

rgukt@pushpaPC MINGW64 ~/desktop/MY_THINGS/AWS/Jenkins-Zero-to-Hero (main)
$ echo Q1RVZ3FFYWZXMFJzSE0za2U5U3AyWWpYcmhHT21RNXY= | base64 -d
CTUgqEafW0RsHM3ke9Sp2YjXrhGOmQ5v
rgukt@pushpaPC MINGW64 ~/desktop/MY_THINGS/AWS/Jenkins-Zero-to-Hero (main)
$
```

step:2 Now run the below command , so that we get default url to access our cluster

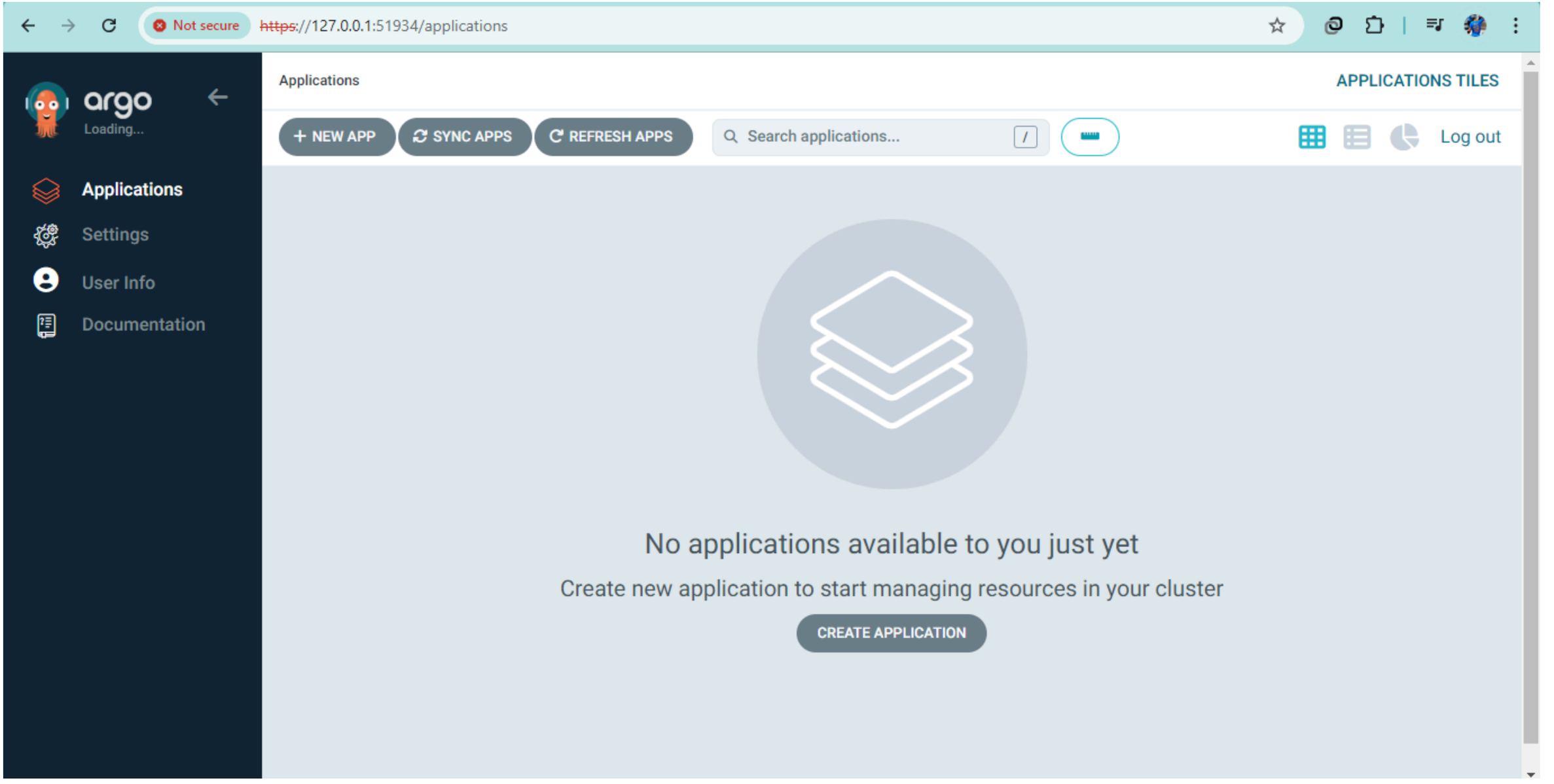
minikube service example-argocd-server

```
rgukt@pushpaPC MINGW64 ~/desktop/Jenkins-Zero-to-Hero (main)
$ minikube service example-argocd-server
|-----------|------------------------|-------------|---------------------------|
| NAMESPACE |          NAME          | TARGET PORT |            URL            |
|-----------|------------------------|-------------|---------------------------|
| default   | example-argocd-server  | http/80     | http://192.168.49.2:31498 |
|           |                        | https/443   | http://192.168.49.2:31382 |
|-----------|------------------------|-------------|---------------------------|
* Starting tunnel for service example-argocd-server.
|-----------|------------------------|-------------|-------------------------|
| NAMESPACE |          NAME          | TARGET PORT |           URL           |
|-----------|------------------------|-------------|-------------------------|
| default   | example-argocd-server  |             | http://127.0.0.1:51934  |
|           |                        |             | http://127.0.0.1:51935  |
|-----------|------------------------|-------------|-------------------------|
[default example-argocd-server  http://127.0.0.1:51934
http://127.0.0.1:51935]
! Because you are using a Docker driver on windows, the terminal needs to be open to run it.
```

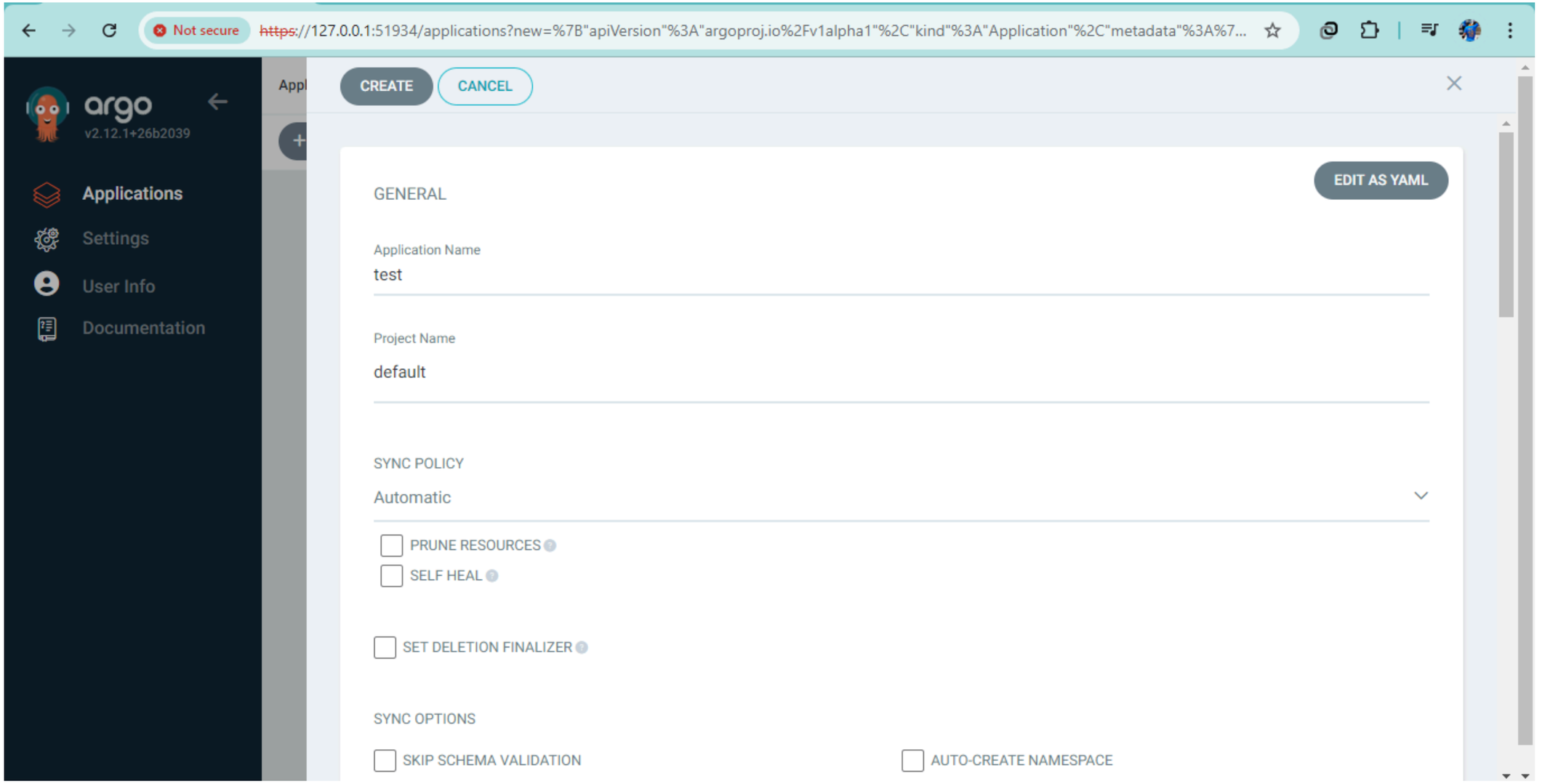
step:3 Now edit one of the url and provide the password that we got and give username as your choise

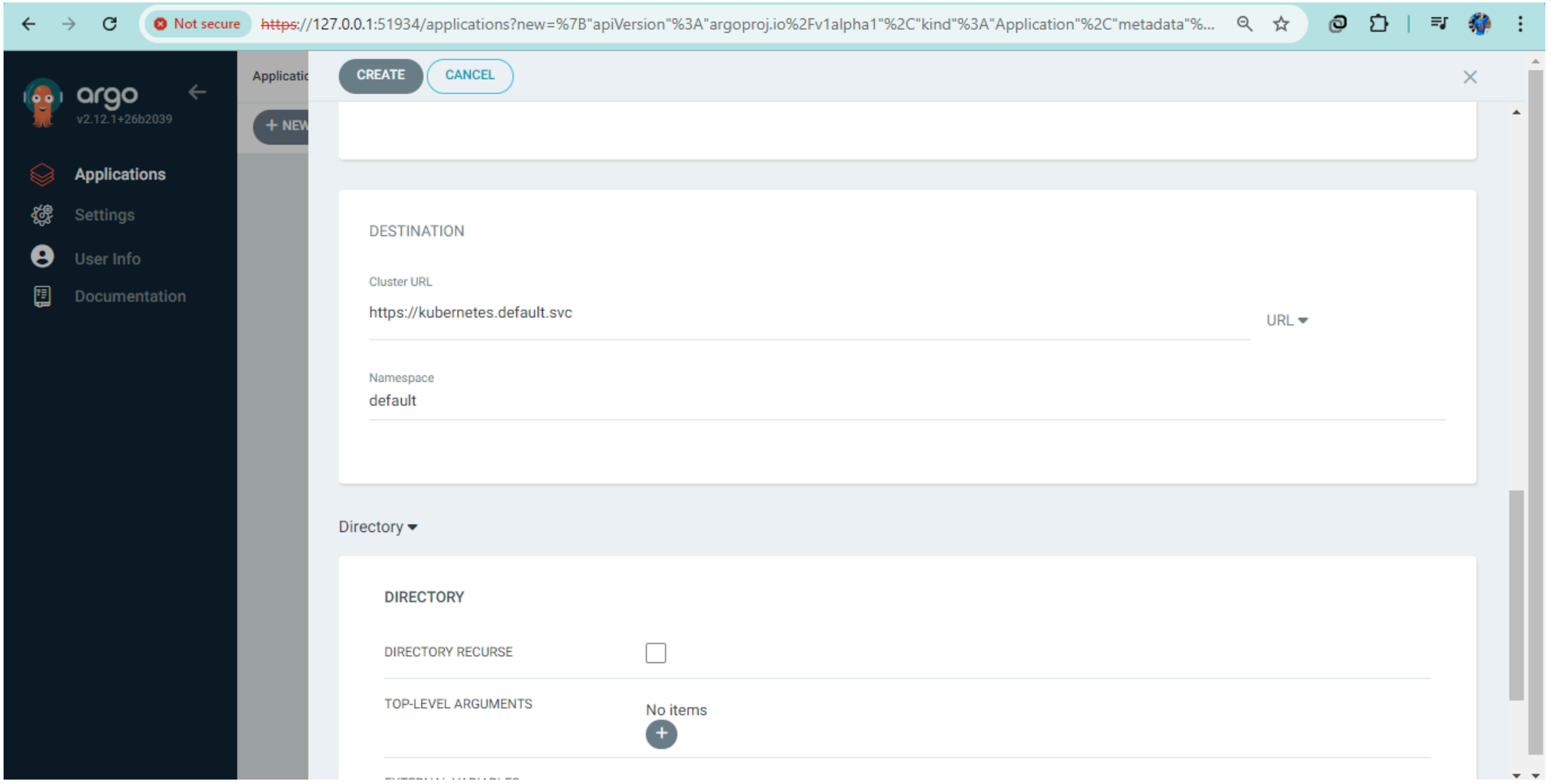


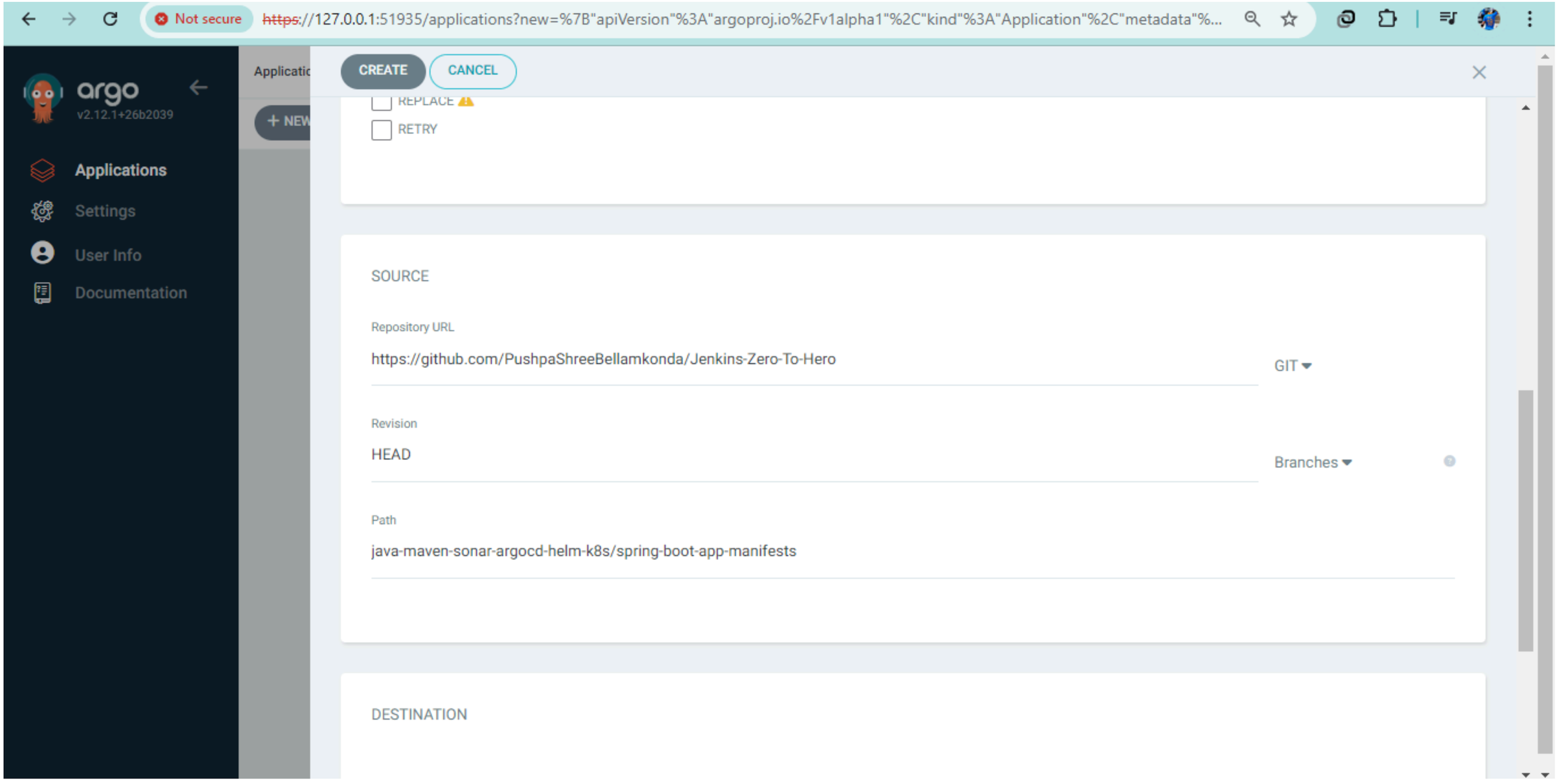
After logging in this is the interface we see



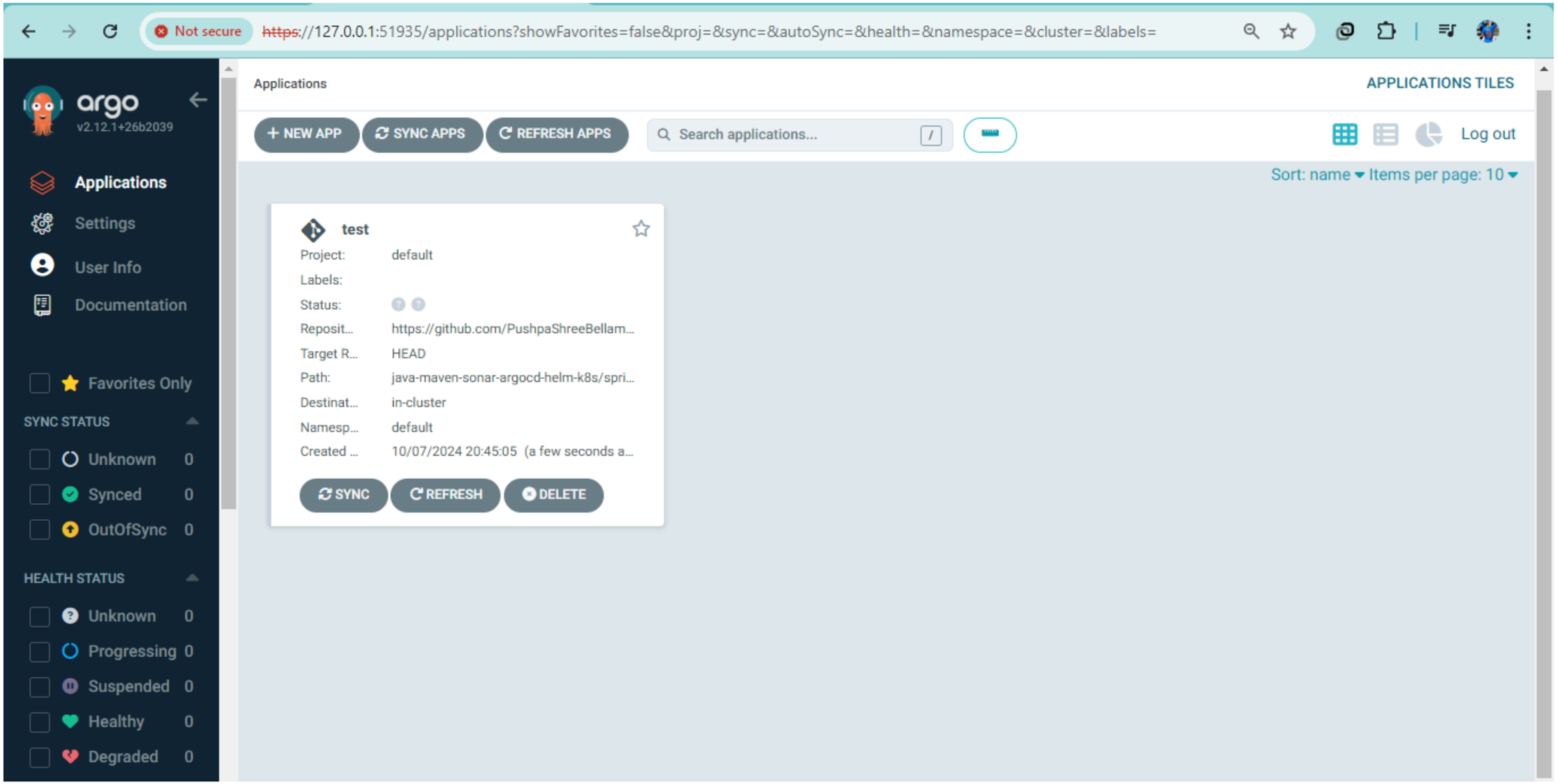
step:4 Now click on "new app" and click on "create" and give application name , project name , destination url , namespace , repository url and dockerfile path and click on create



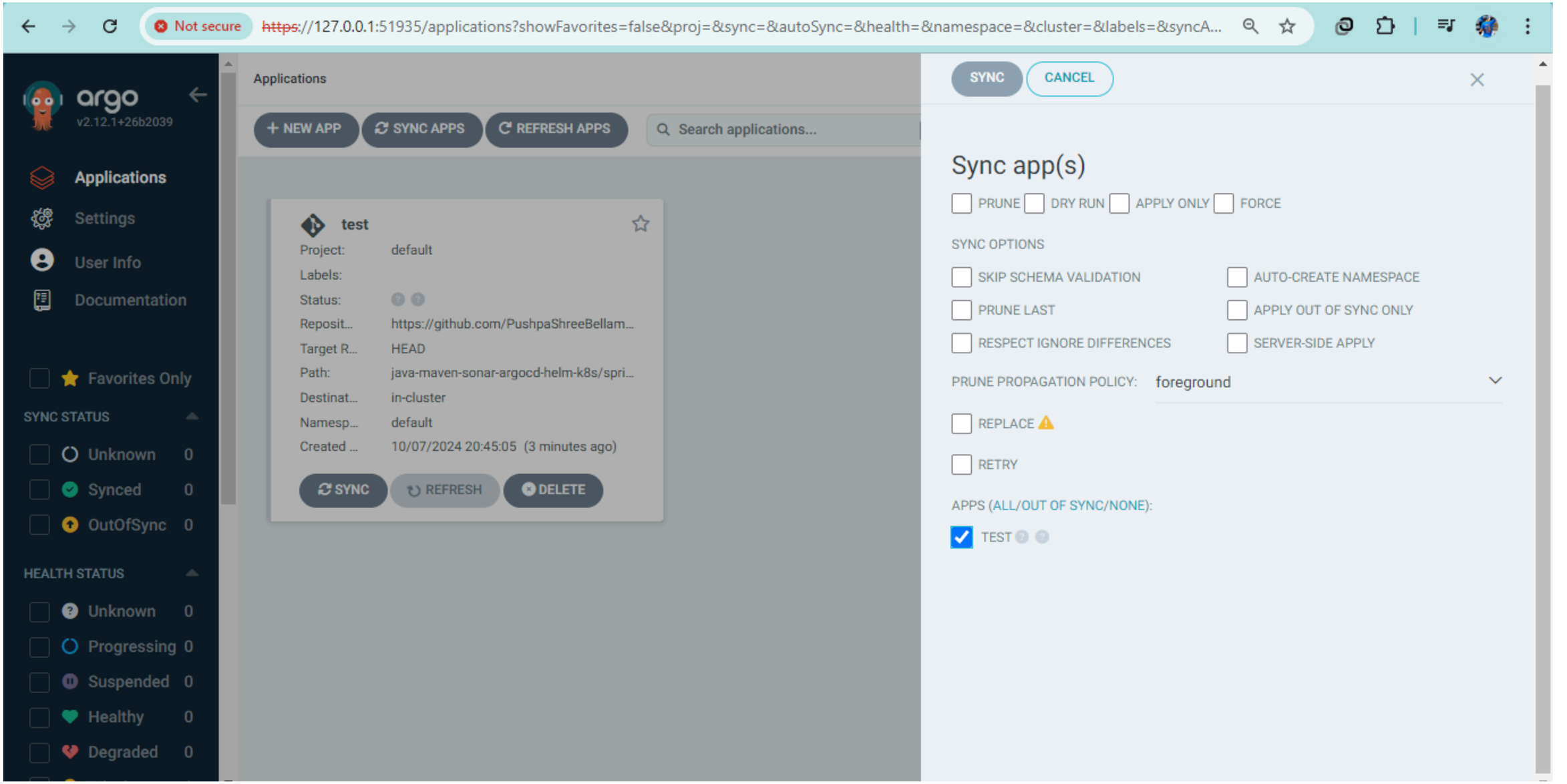

Not secure
https://127.0.0.1:51934/applications?new=%7B"apiVersion"%3A"argoproj.io%2Fv1alpha1"%2C"kind"%3A"Application"%2C"metadata"%...
argo
v2.12.1+26b2039
Applications
Settings
User Info
Documentation
Applicatic
+ NEW
CREATE
CANCEL
DESTINATION
Cluster URL
https://kubernetes.default.svc
URL
Namespace
default
Directory
DIRECTORY
DIRECTORY RECURSE
TOP-LEVEL ARGUMENTS
No items

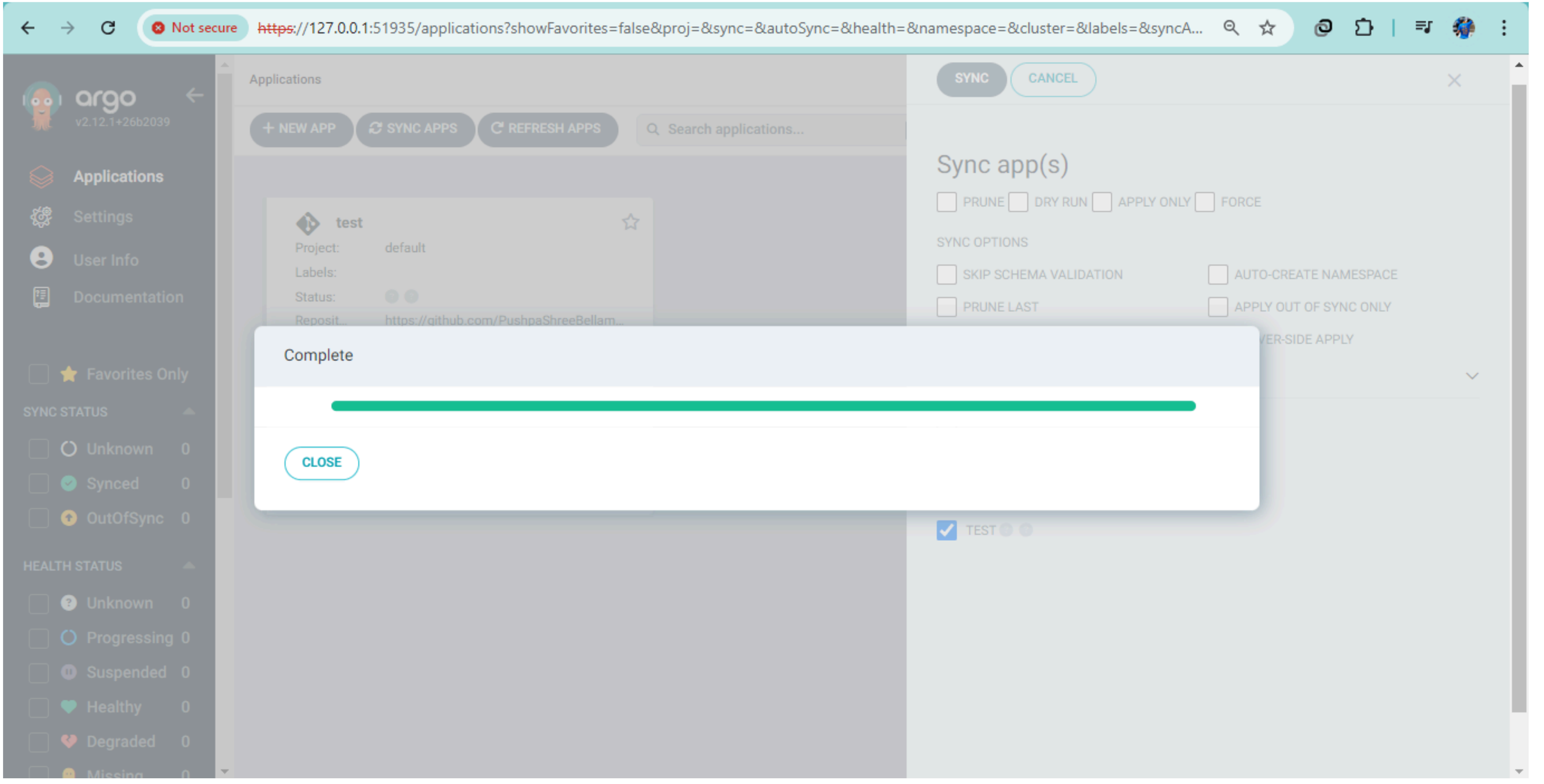

Not secure
https://127.0.0.1:51935/applications?new=%7B"apiVersion"%3A"argoproj.io%2Fv1alpha1"%2C"kind"%3A"Application"%2C"metadata"%...
argo
v2.12.1+26b2039
Applications
Settings
User Info
Documentation
Applicatic
+ NEW
CREATE
CANCEL
REPLACE
RETRY
SOURCE
Repository URL
https://github.com/PushpaShreeBellamkonda/Jenkins-Zero-To-Hero
GIT
Revision
HEAD
Branches
Path
java-maven-sonar-argocd-helm-k8s/spring-boot-app-manifests
DESTINATION

This is the application we have created



step:5 Now click on "sync apps" and check the application "test" and sync it




Not secure
https://127.0.0.1:51935/applications?showFavorites=false&proj=&sync=&autoSync=&health=&namespace=&cluster=&labels=&syncA...
argo
v2.12.1+26b2039
Applications
Settings
User Info
Documentation
Favorites Only
SYNC STATUS
Unknown 0
Synced 0
OutOfSync 0
HEALTH STATUS
Unknown 0
Progressing 0
Suspended 0
Healthy 0
Degraded 0
Applications
+ NEW APP
SYNC APPS
REFRESH APPS
Search applications...
test
Project: default
Labels:
Status:
Reposit... https://github.com/PushpaShreeBellam...
Complete
CLOSE
SYNC
CANCEL
Sync app(s)
PRUNE
DRY RUN
APPLY ONLY
FORCE
SYNC OPTIONS
SKIP SCHEMA VALIDATION
AUTO-CREATE NAMESPACE
PRUNE LAST
APPLY OUT OF SYNC ONLY
TEST

step:6 Now check the pods and deployment , we see that our pods are deployed and running

```
rgukt@pushpaPC MINGW64 ~
$ kubectl get pods
NAME                                          READY   STATUS    RESTARTS        AGE
example-argocd-application-controller-0       0/1     Error     1               43m
example-argocd-redis-6545fd6d6c-hg4hv         0/1     Error     7               47h
example-argocd-repo-server-869d5757c7-zz7b8   0/1     Pending   44              47h
example-argocd-server-76bb84cddc-vdrjq        0/1     Error     44              47h
spring-boot-app-7f67db6bf-ggsnt               1/1     Running   2 (4m10s ago)   18h
spring-boot-app-7f67db6bf-mr2bz               1/1     Running   2               18h

rgukt@pushpaPC MINGW64 ~
$ kubectl get deploy
NAME                      READY   UP-TO-DATE   AVAILABLE   AGE
example-argocd-redis      1/1     1            1           47h
example-argocd-repo-server   0/1     1            0           47h
example-argocd-server     0/1     1            0           47h
spring-boot-app           2/2     2            2           18h

rgukt@pushpaPC MINGW64 ~
$
```