Kubernetes pull docker image using jenkins

- Requirements:
- java executable program
- machine with jenkins & docker
- and machine with kubernetes
- kubernetes connected nodes

Step1:

First we create java runnable programs and pull in to your git
And create pom.xml file to create war or jar file
i get it from my old git java file. Here link for this

https://github.com/cmmani/java-app

Step2:

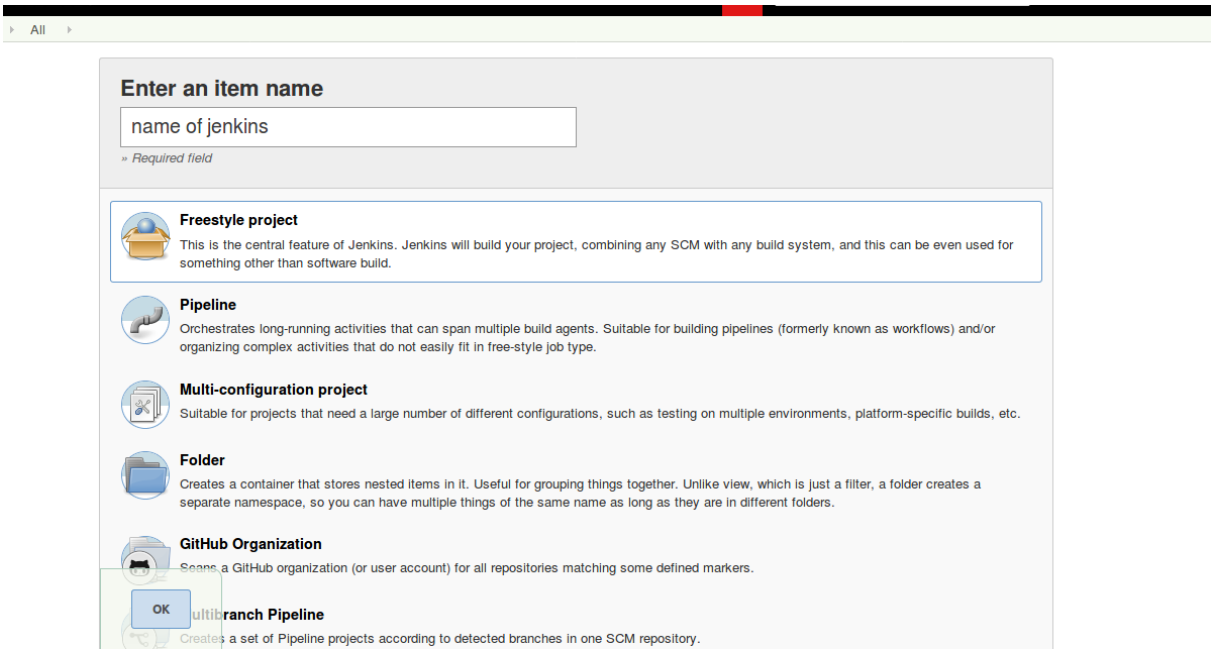
Here our project start

first we install docker from jenkins machine or any bare machine.
But i install docker in jenkins machine.

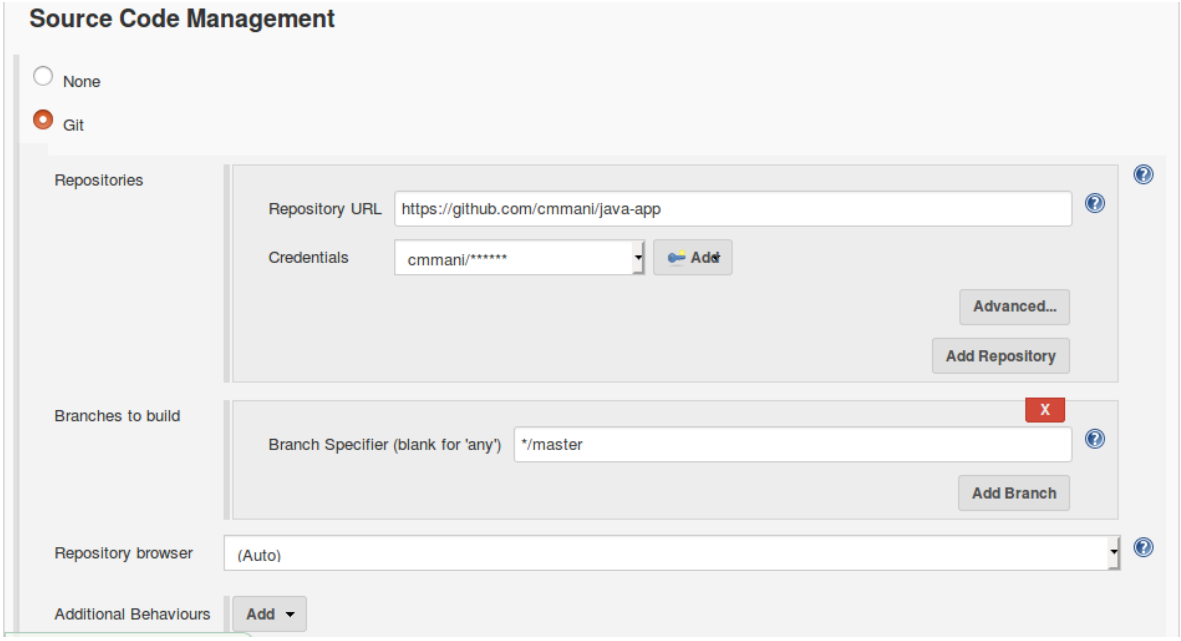
Step3:

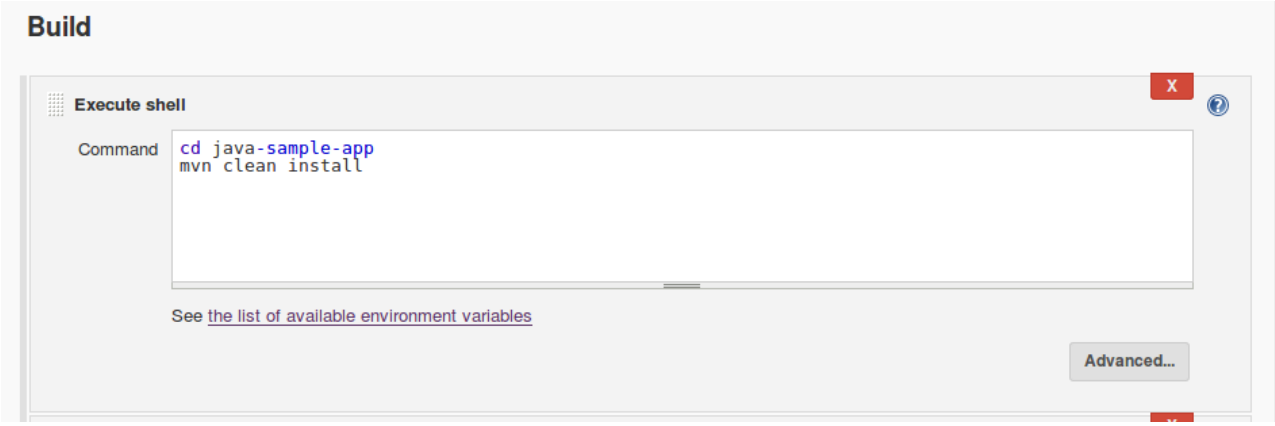
create new jenkins directory to create war file in local jenkins machine

Create jenkins in freestyle project:



Get a java application from git and add credential:





Step4:

Then build now the project to deploy War file in jenkins workspace

Step5:

Go to the jenkins path

$ cd /var/lib/jenkins/workspace/[define your jenkins projectname]/target$ ls

$ ls

You can see the war file in that folder

JAVA_APP.war

Step 6:

Create docker tomcat container to deploy war file

$ vi Dockerfile

FROM tomcat

here you run a tomcat container;

And tomcat need auntheticate to enter in to the app folder

So we need to set root password and username for tomcat

target>>

$ vi contexts.xml

```
<Context antiResourceLocking="false" privileged="true" >
   <!-- <Valve className="org.apache.catalina.valves.RemoteAddrValve"
       allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" /> -->
   <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|
String)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Linked)?
HashMap"/>
</Context>
```

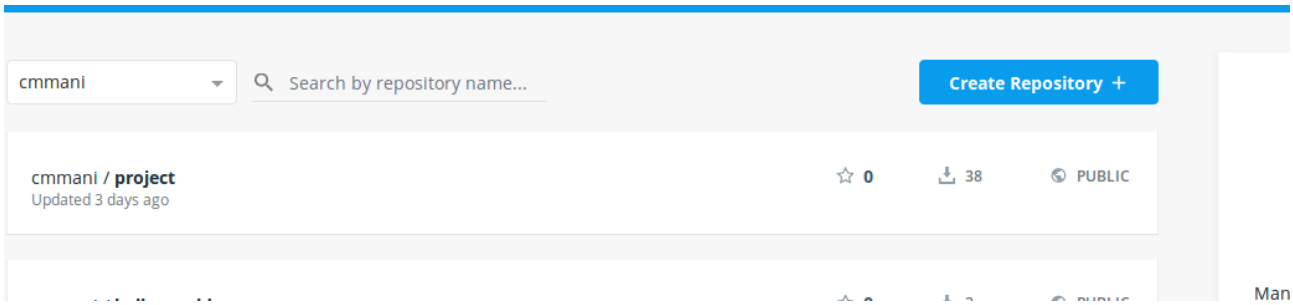
$ vi tomcat-user.xml

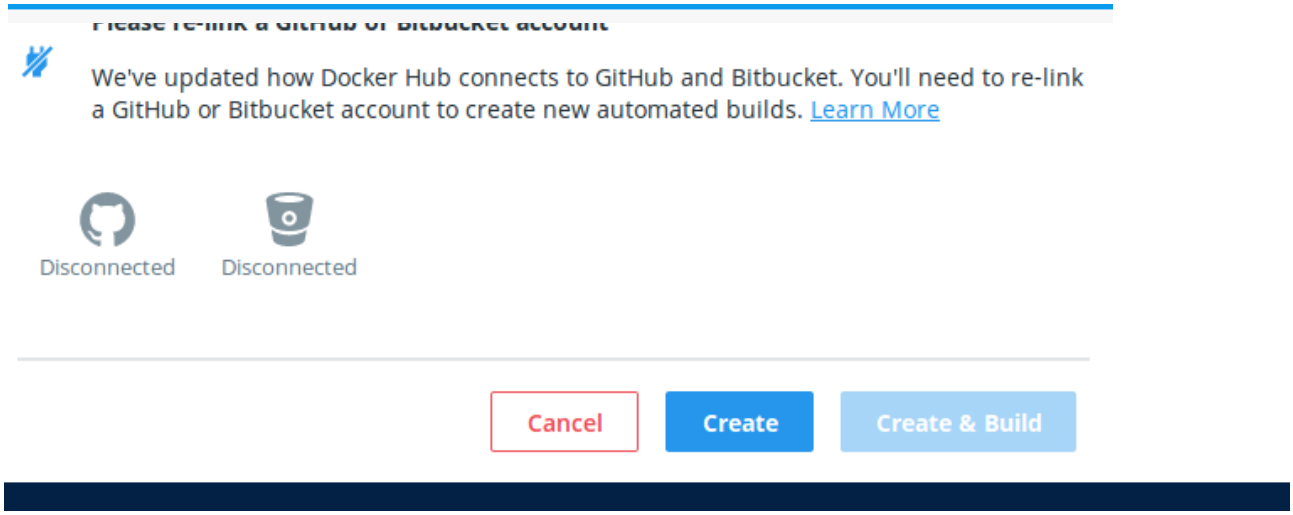
```
<tomcat-users xmlns="http://tomcat.apache.org/xml"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd"
            version="1.0">
  <user username="admin" password="secret" roles="manager-gui"/>
```

</tomcat-users>

Step 6:

Create docker repo in your docker hub account





$ vi Dockerfile

```
FROM tomcat
COPY tomcat-users.xml /usr/local/tomcat/conf/tomcat-users.xml
COPY context.xml /usr/local/tomcat/webapps/manager/META-INF/context.xml
COPY java-sample-app-1.0.0.war /usr/local/tomcat/webapps/
```

Then pull three file into Git in the same GIT repo

https://github.com/cmmani/java-app

(In this repo had that files)

Step 7:

GO to jenkins :

Manage jenkins >>>> Manage plugin >>>

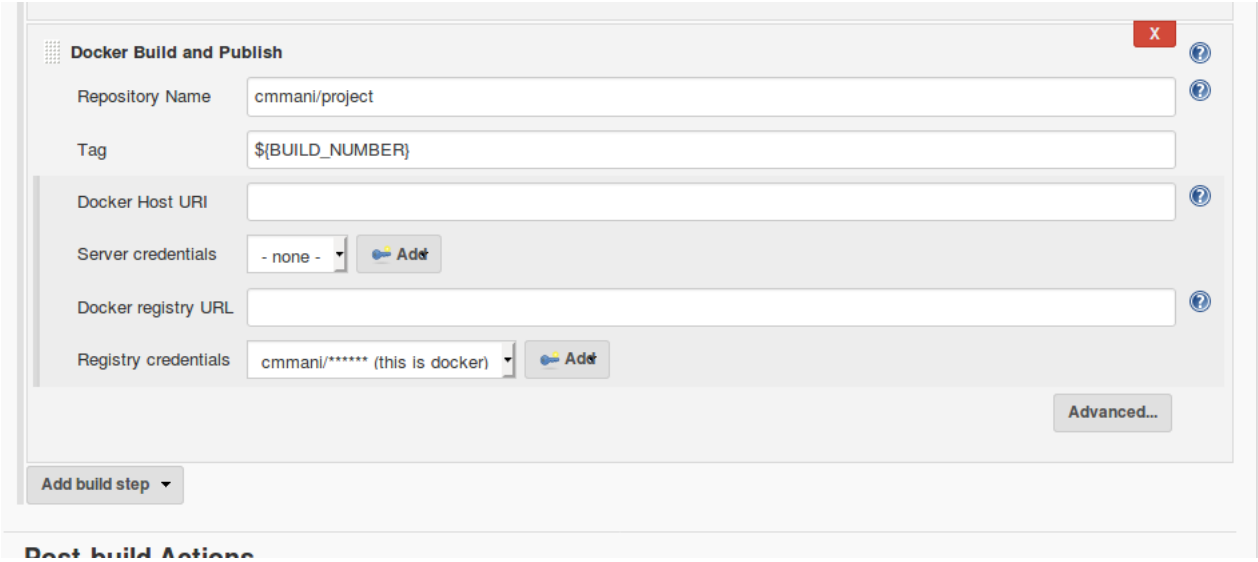
Go to availables in plugin

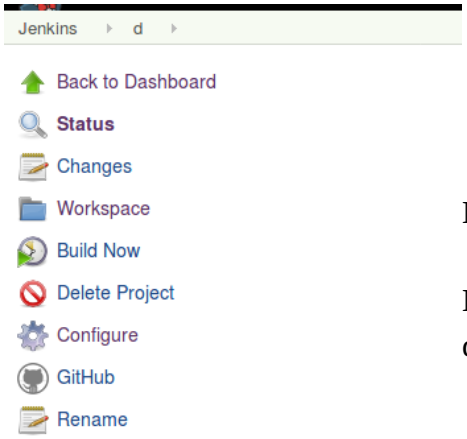
Search plugin Docker and Publish

And install the plugin **Docker and publish**

Then go to jenkinsproject >> configure

And go to build environment



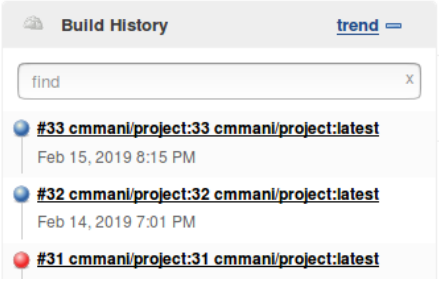


Here we build our jenkins project;

Now our file pull in to the docker hub.

Next we have to pull image run in kubernetes

My idea is to run the kubernetes by connect with SSH and run as a downstream project.



Kubernetes work:

Requirement:

- kubernetes master
- and its node

Create Yaml file to run docker pull image.

I create

$ vi index.yml

```
kind: Deployment
metadata:
  name: angular-deployment
spec:
  selector:
    matchLabels:
      app: angular
  replicas: 2
  template:
    metadata:
      labels:
        app: angular
    spec:
      containers:
      - name: angular
        image: cmmani/project:latest
        ports:
        - containerPort: 8080
---
networking/service/#defining-a-service
kind: Service
apiVersion: v1
metadata:
  name: angular-service
spec:
  selector:
    app: angular
  ports:
  - protocol: TCP
    port: 8080
    targetPort: 8080
    nodePort: 31001
  type: NodePort
```

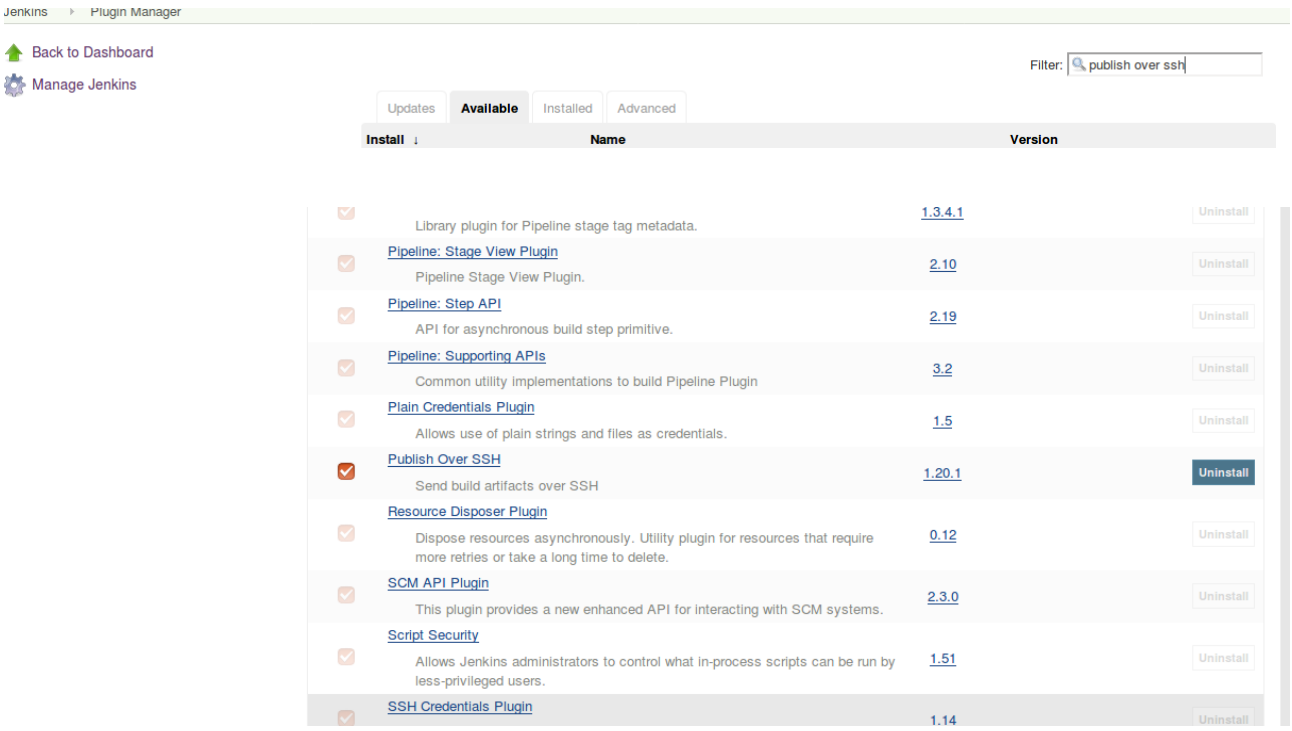
Those above file is in /root path in kubernetes master

Step1:

Then I connect kubernetes master with ssh plugin

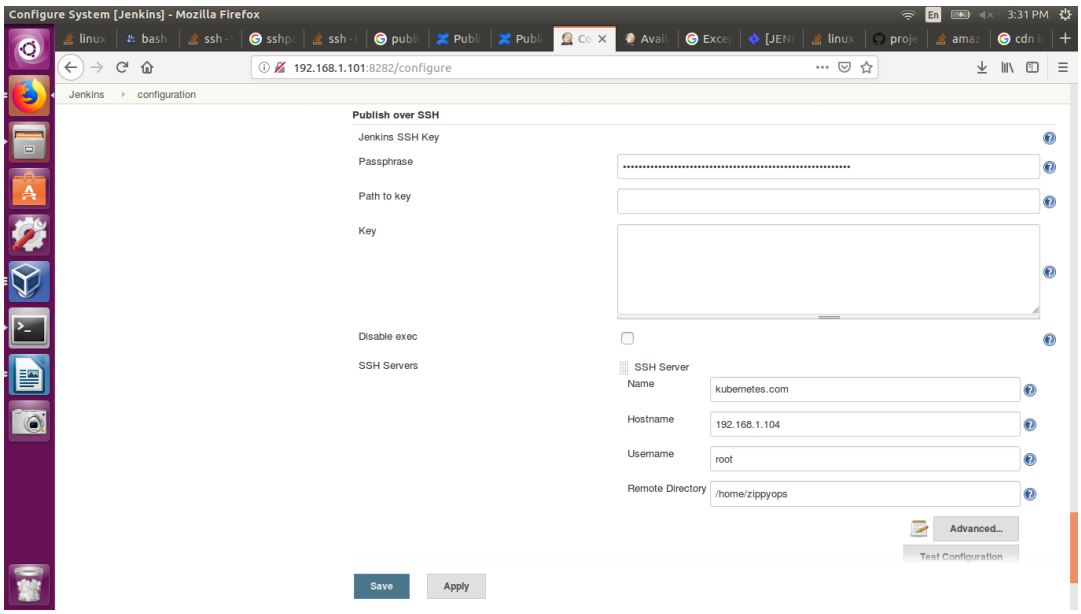
Managejenkins>>>manage plugins

Publish Over SSH

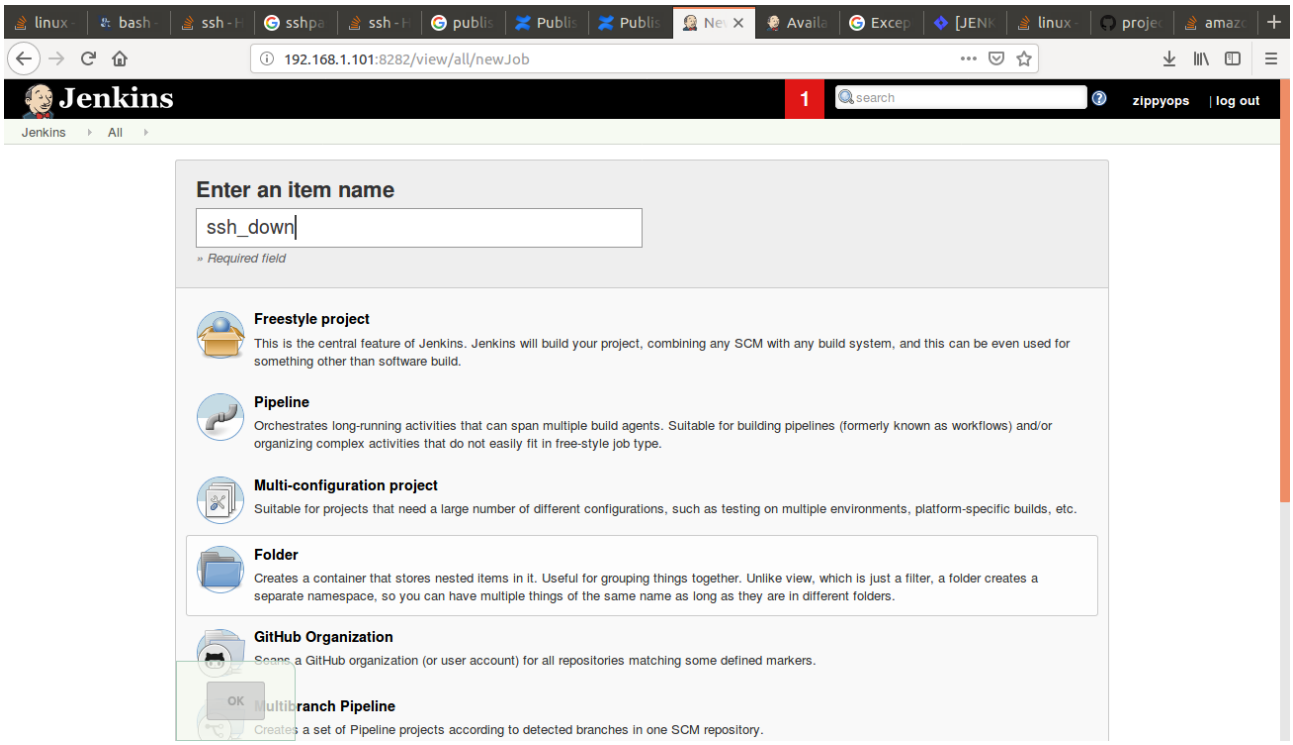


Step2:

Manage jenkins >>> configure management

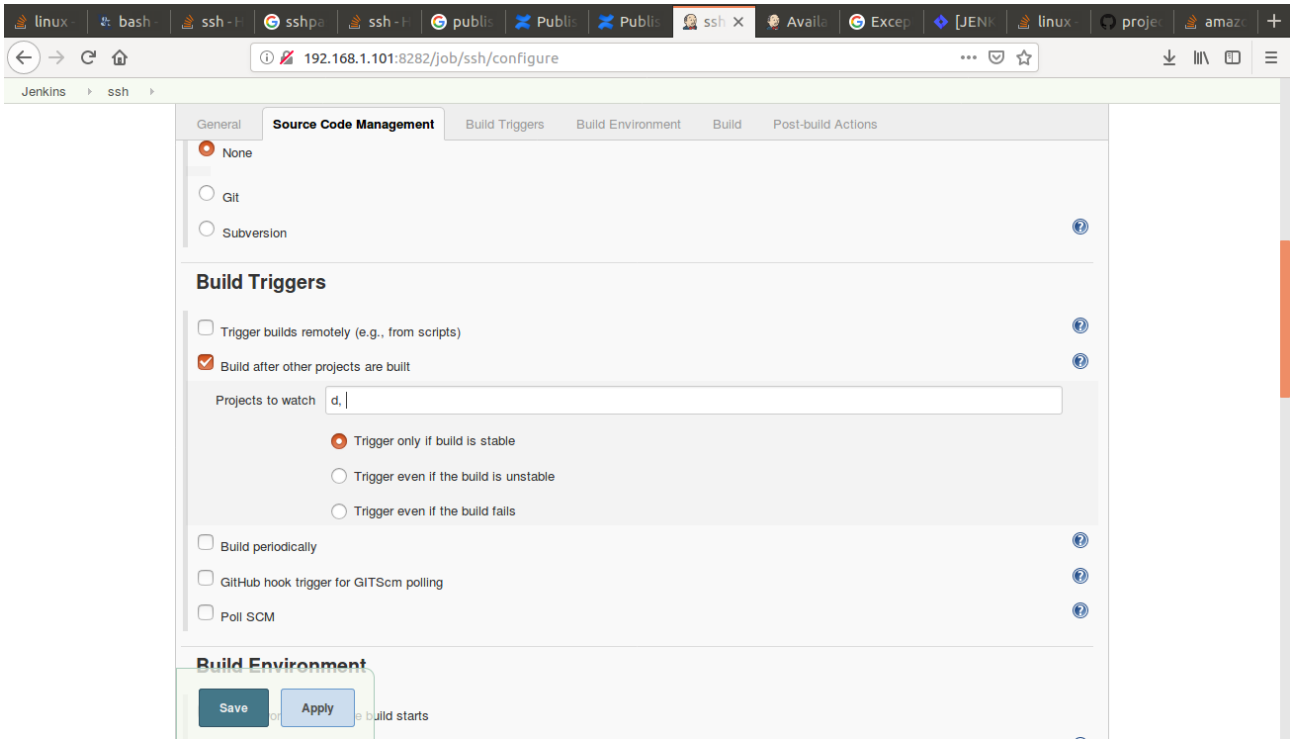


Create one downstream project in jenkins



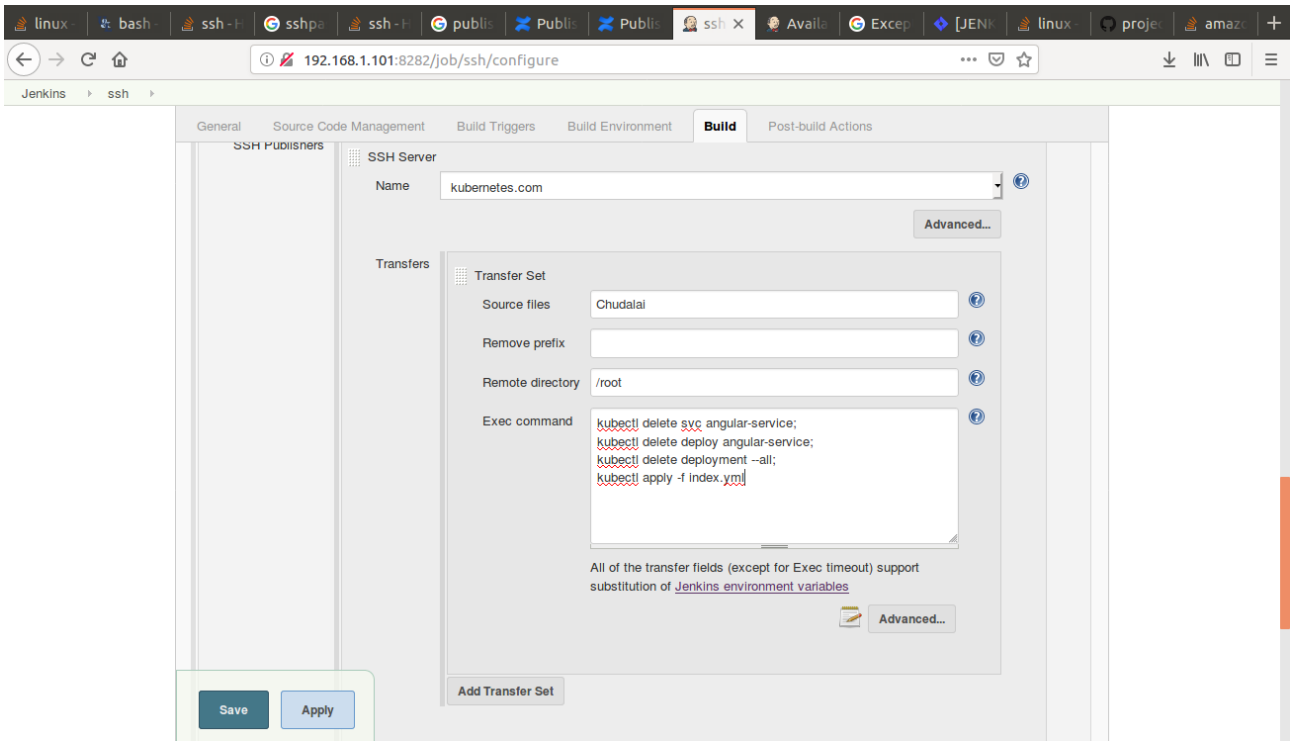
Step2:

And in add build triggers i specified run this project after run the old project

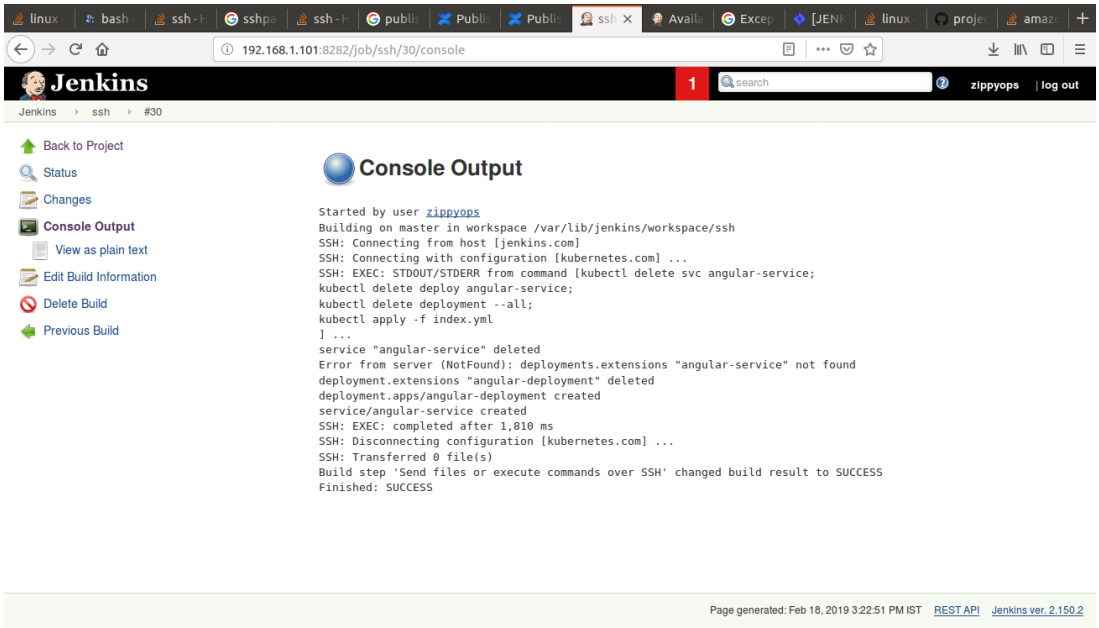


And Exec Command via jenkins

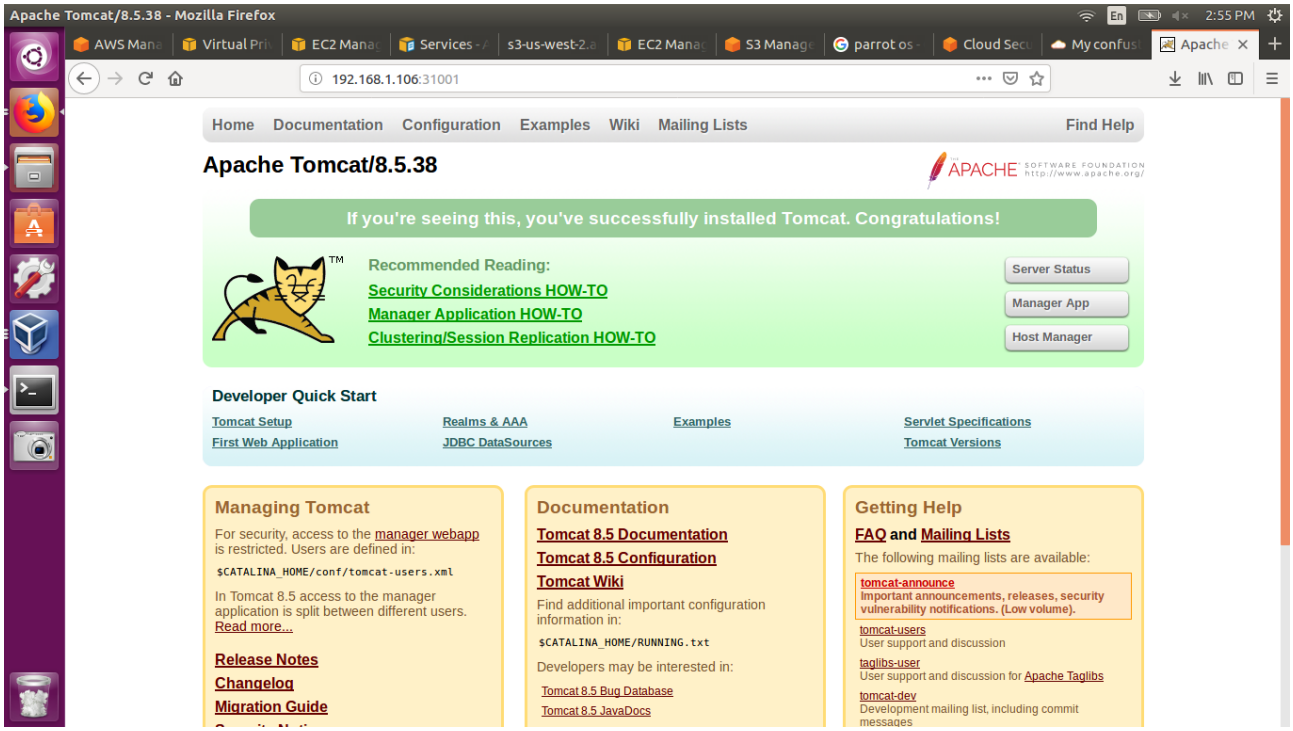
kubectl delete svc angular-service;
kubectl delete deploy angular-service;
kubectl delete deployment --all;
kubectl apply -f index.yml



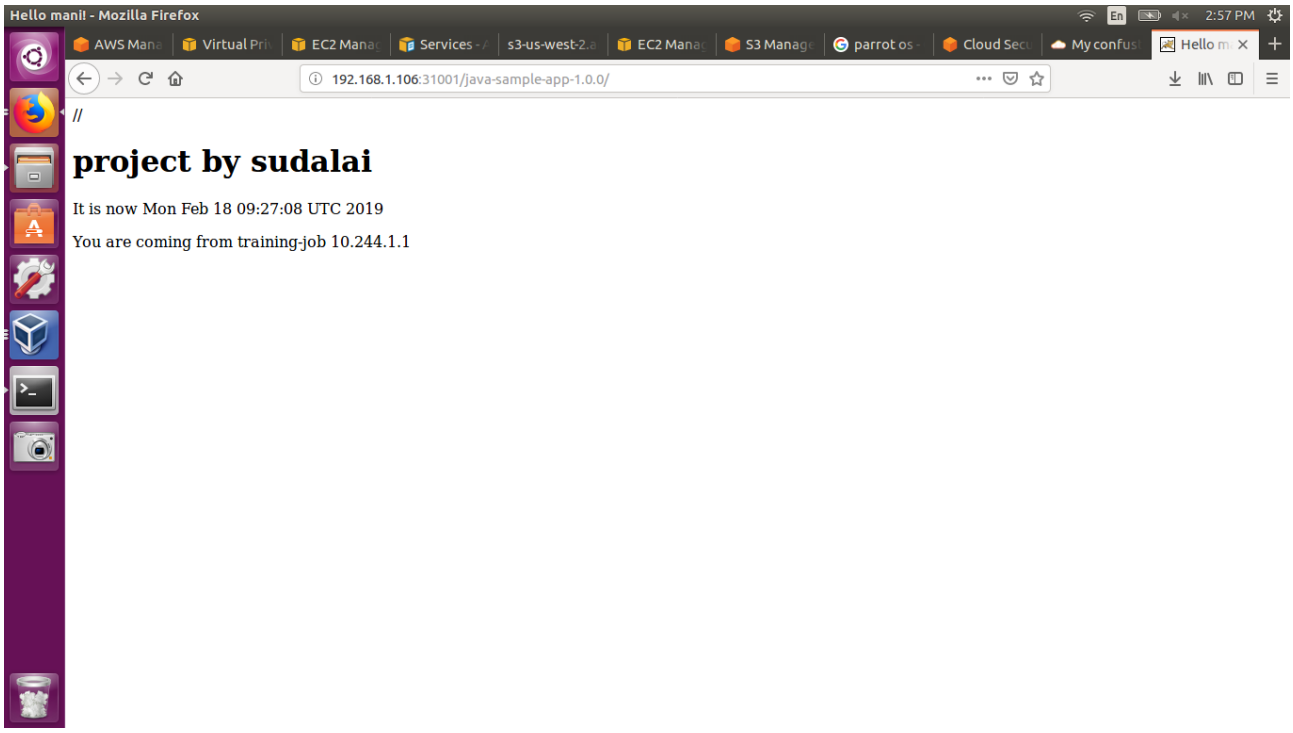
Run the above jenkins projec:
You get those success build output



Save and Build the project in local machine by using kubernetes node IP

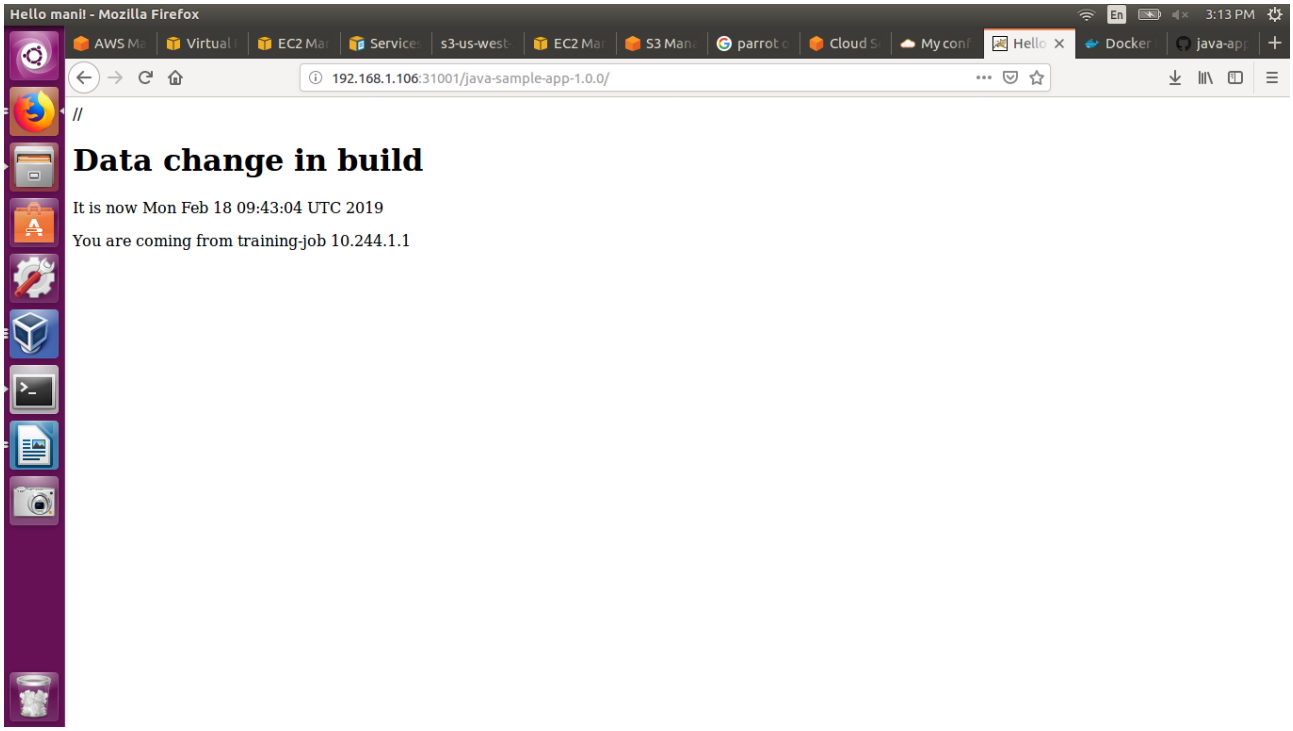


click manage apps:
It is asked for password and username
these details are in tomcat_user.xml file



Hereafter i changed littlebit in java application and now build the D project in one moretime. The output will be like

just refresh the page



Hence you got the output of the kubernetes and all files deploy automatic in every kubenodes.