\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Jenkins\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Initially in centos, java's old version will be installed. either you can uninstall and install latest versio or you can usealtenative options,

# yum search jdk

java-11-openjdk.x86\_64 : OpenJDK 11 Runtime Environment ----installed this version

# yum install java-11-openjdk.x86\_64

alternatives --config java

There are 3 programs which provide 'java'.

Selection Command

-----------------------------------------------

1 java-1.7.0-openjdk.x86\_64 (/usr/lib/jvm/java-1.7.0-openjdk-1.7.0.261-2.6.22.2.el7\_8.x86\_64/jre/bin/java)

\*+ 2 java-1.8.0-openjdk.x86\_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.392.b08-2.el7\_9.x86\_64/jre/bin/java)

3 java-11-openjdk.x86\_64 (/usr/lib/jvm/java-11-openjdk-11.0.21.0.9-1.el7\_9.x86\_64/bin/java)

Enter to keep the current selection[+], or type selection number: 3

[root@master1 ~]# java -version

openjdk version "11.0.21" 2023-10-17 LTS

OpenJDK Runtime Environment (Red\_Hat-11.0.21.0.9-1.el7\_9) (build 11.0.21+9-LTS)

OpenJDK 64-Bit Server VM (Red\_Hat-11.0.21.0.9-1.el7\_9) (build 11.0.21+9-LTS, mixed mode, sharing)

=========================================================================

Step 1>>>>>>>>>

Download jenkins war file..........

mkdir /jenkins

[root@master1 Downloads]# cp /home/binay/Downloads/jenkins.war /jenkins/

[root@master1 Downloads]# cd /jenkins/

[root@master1 jenkins]# ls

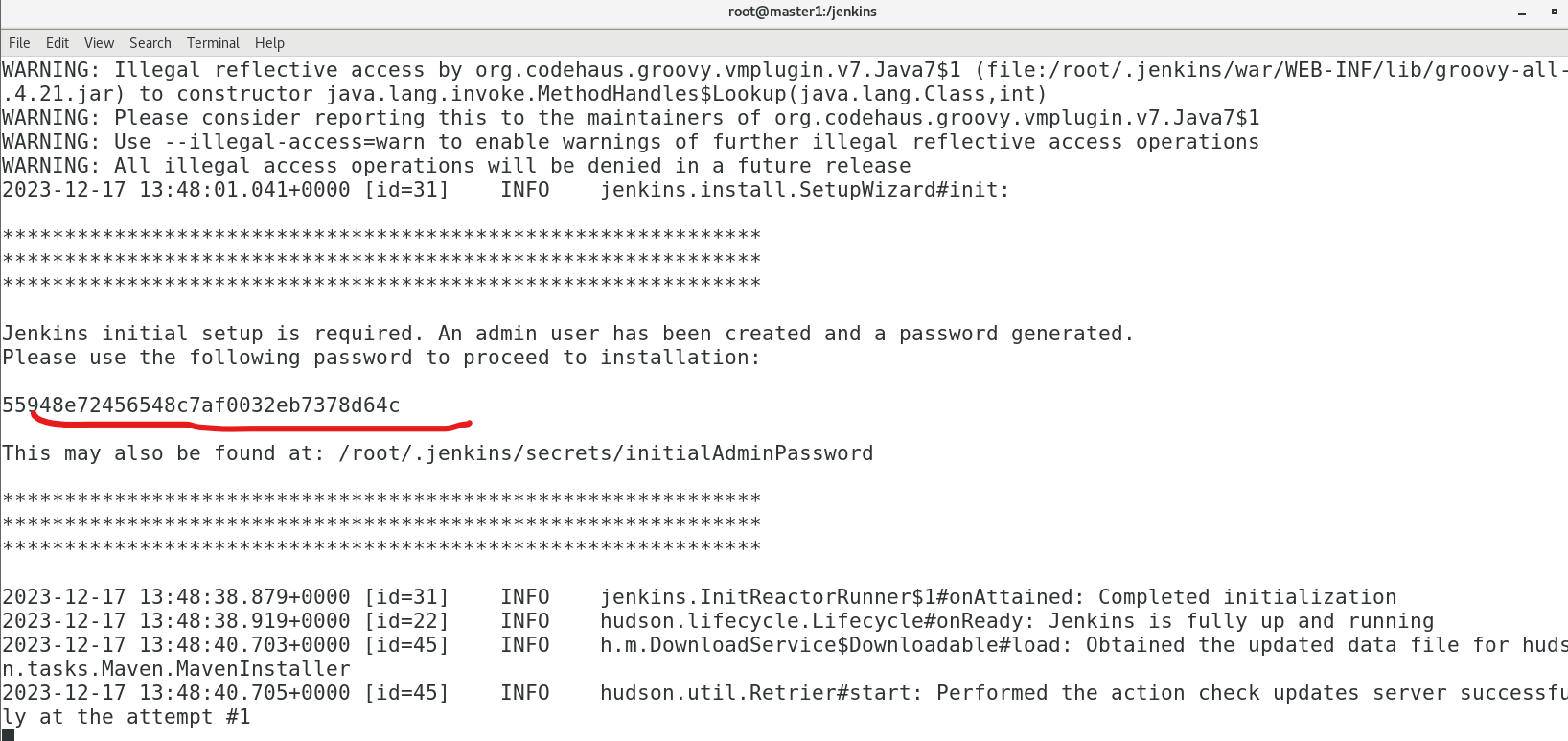
jenkins.war

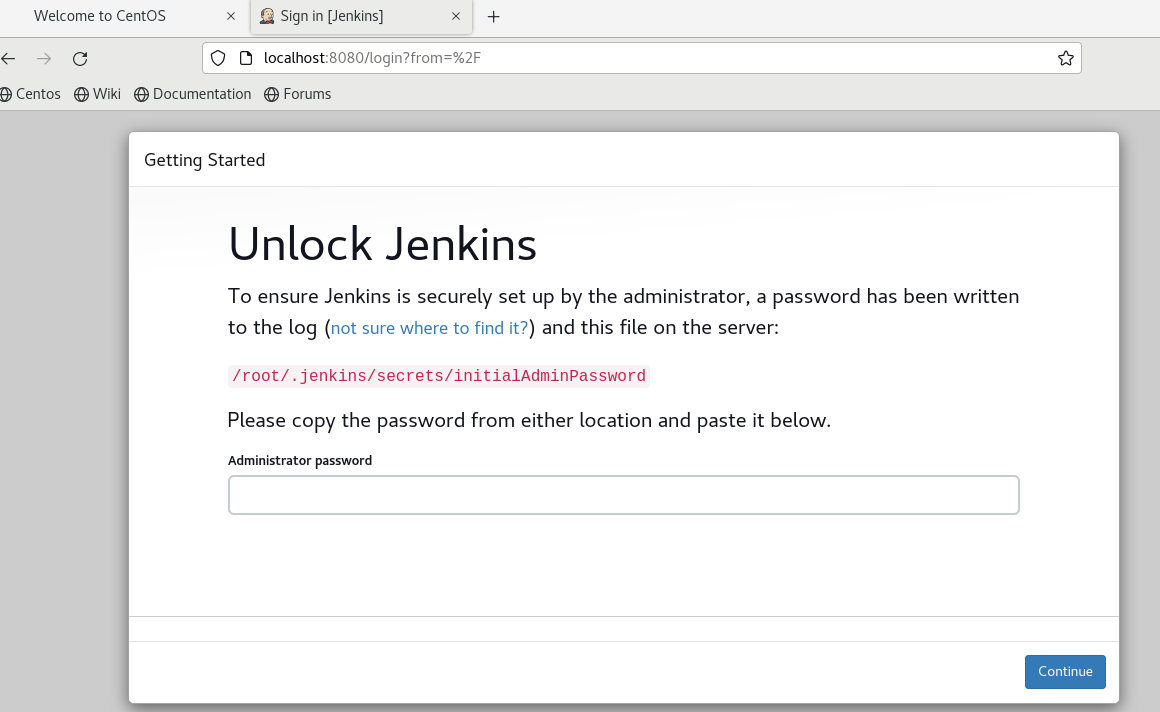
Step2>>>>>>>>>>>>>>>>>>>>>

jenkins]# java -jar jenkins.war

Running from: /jenkins/jenkins.war

webroot: /root/.jenkins/war





# cd /root/.jenkins/

jobs/ nodes/ plugins/ secrets/ updates/ userContent/ users/ war/

[root@master1 ~]# cd /root/.jenkins/secrets/

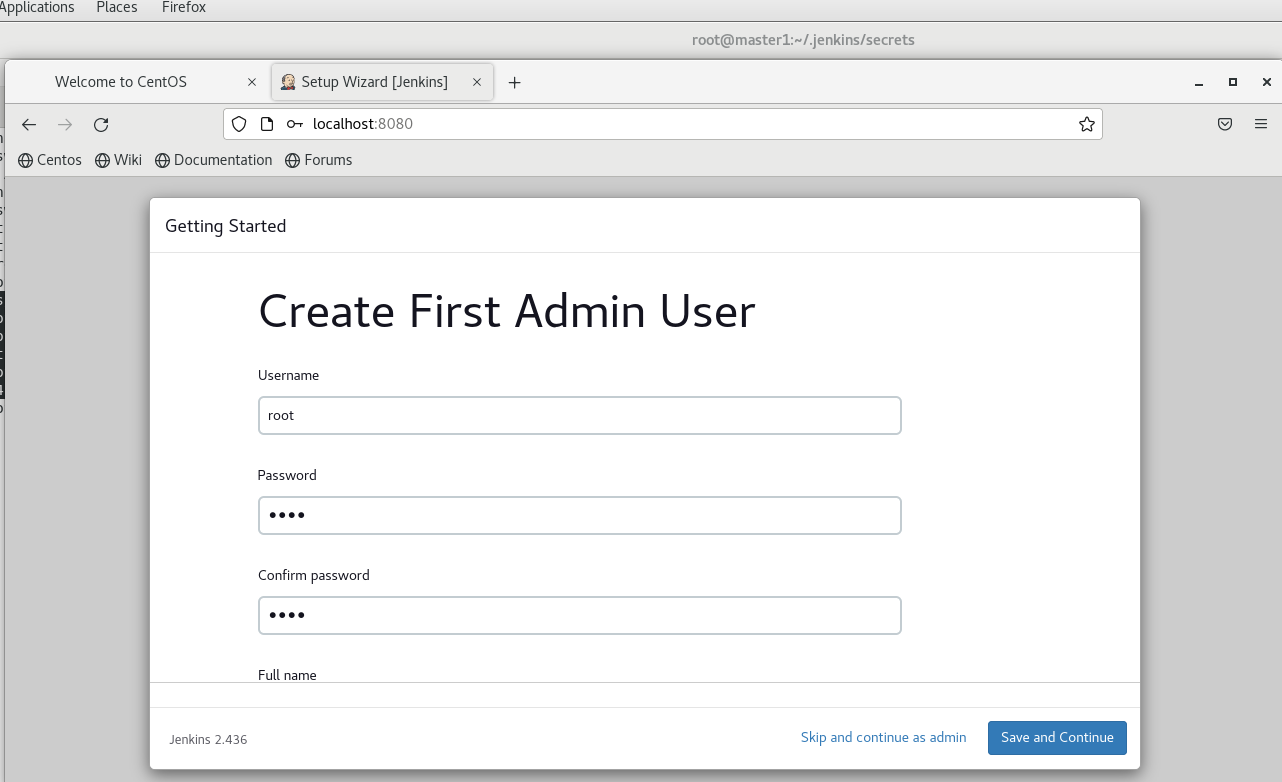
[root@master1 secrets]# ls

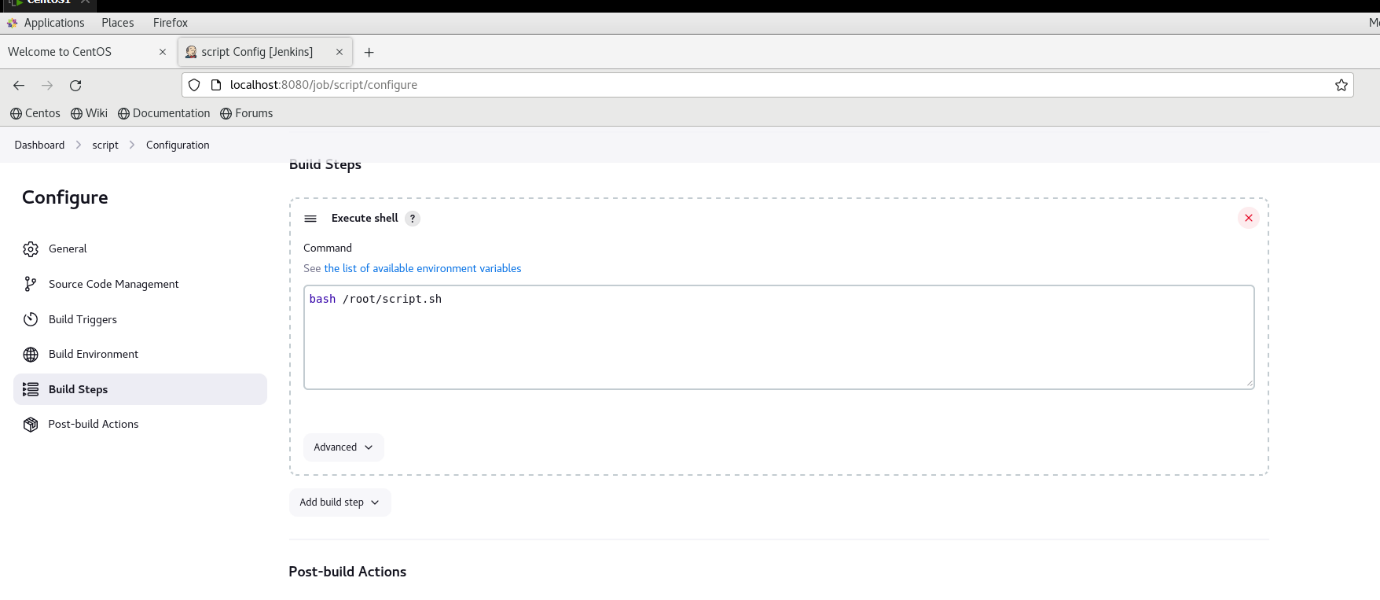
initialAdminPassword jenkins.model.Jenkins.crumbSalt master.key

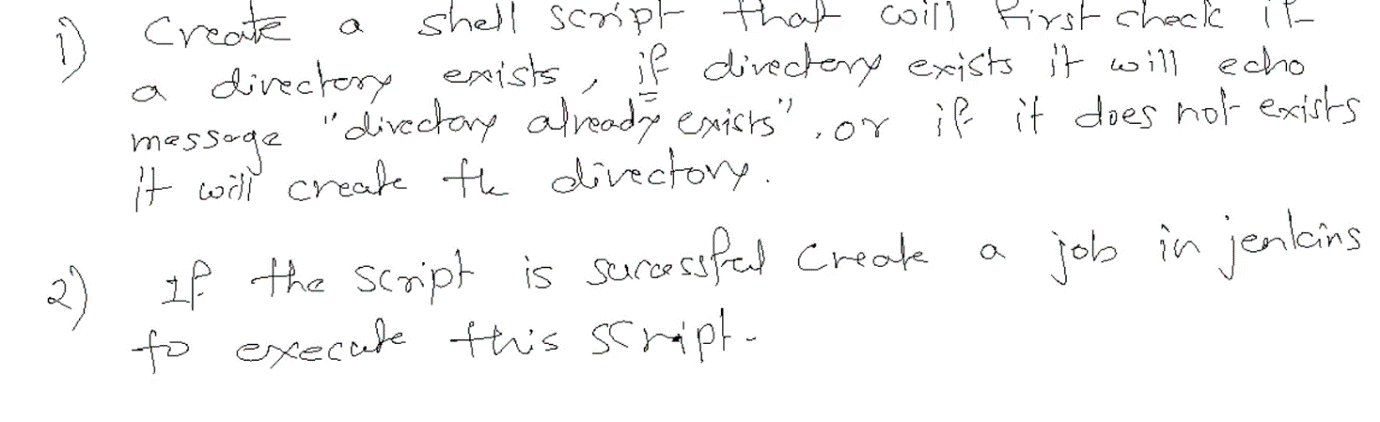
[root@master1 secrets]# cat initialAdminPassword

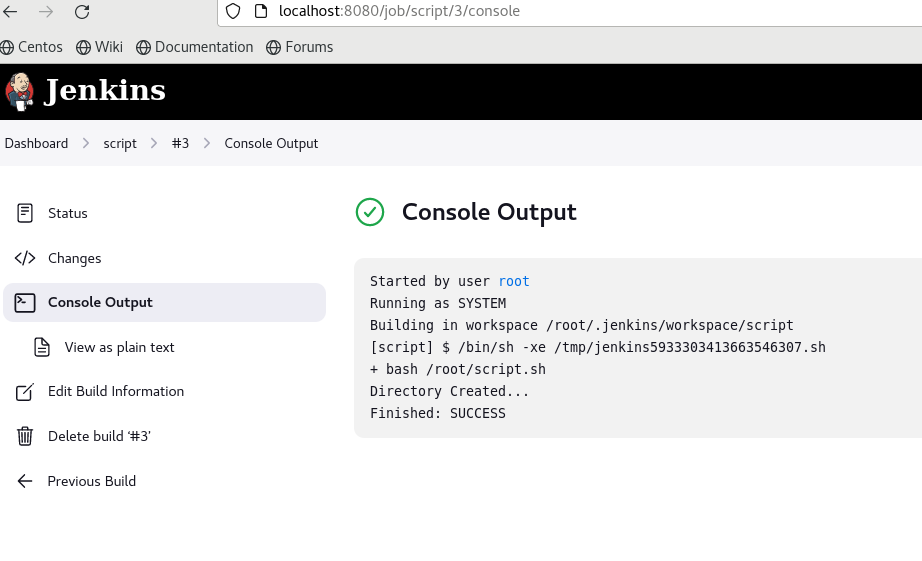
55948e72456548c7af0032eb7378d64c

Set username and Password:--------









# cat script.sh

if [[ -d "test" ]]

then

echo "Directory exists..."

else

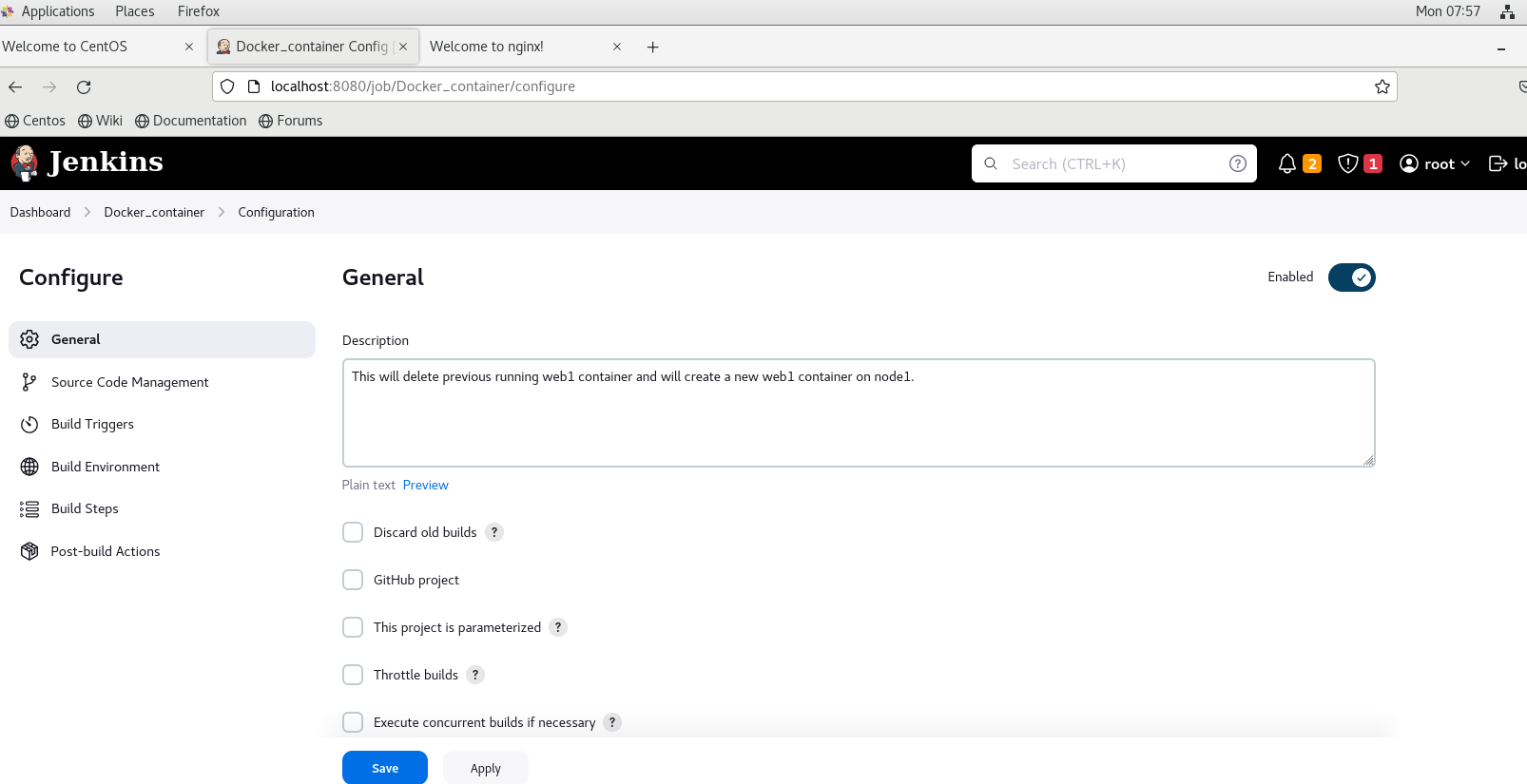
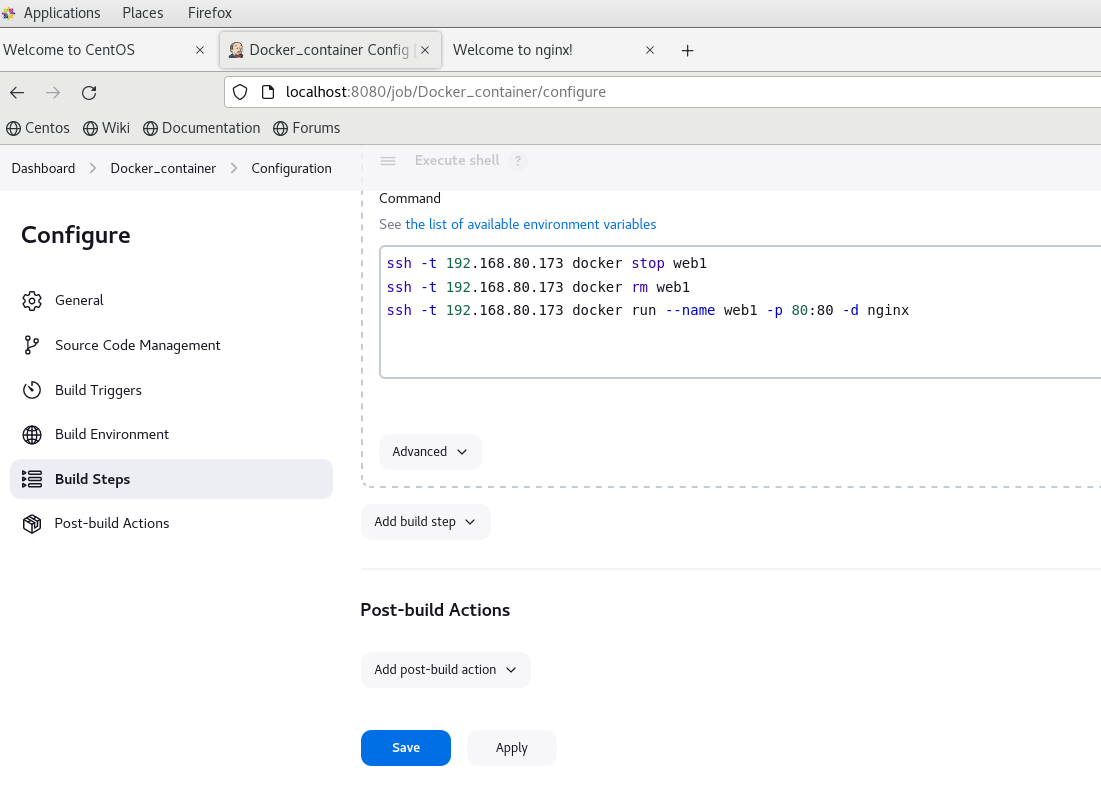
mkdir test

echo "Directory Created..."

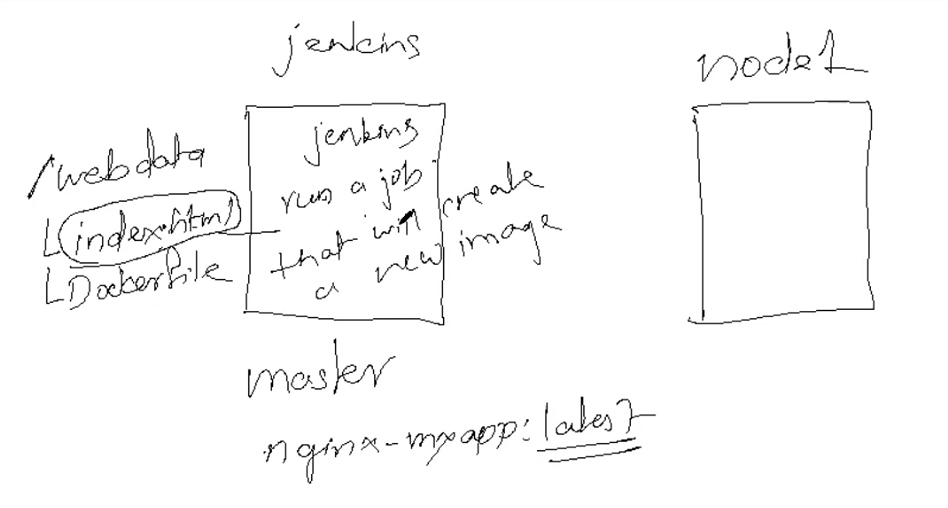
fi

2nd Assignment:

Configure Passwordless ssh



3rd assignment:-

In this question, we create a docker image. And we will convert it into tar and will copy to node1.

On node 1, we will untar the image and create container.

This all operations will be done through Jenkins..

]# mkdir /webdata

[root@master1 ~]# cd /webdata/

[root@master1 webdata]# vim index.html

[root@master1 webdata]# cat index.html

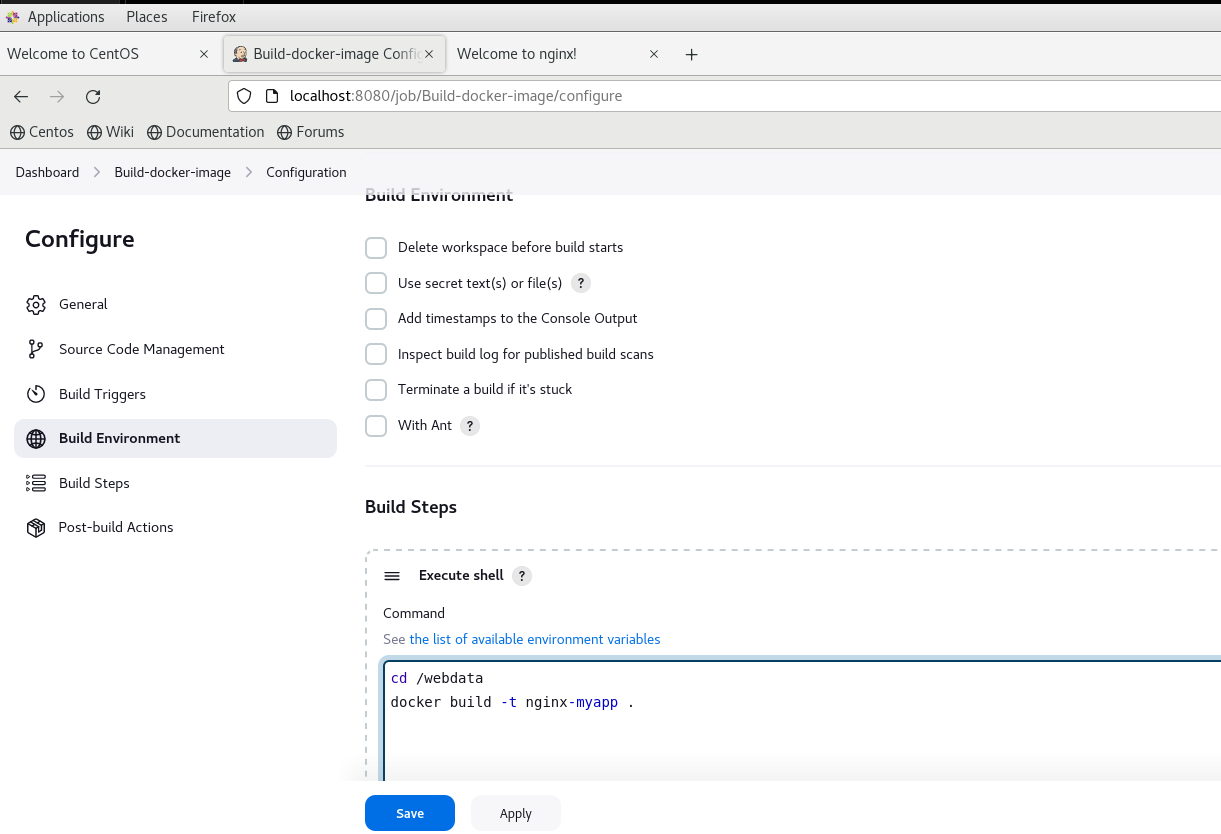
this is the version of web1

[root@master1 webdata]# vi Dockerfile

# cat Dockerfile

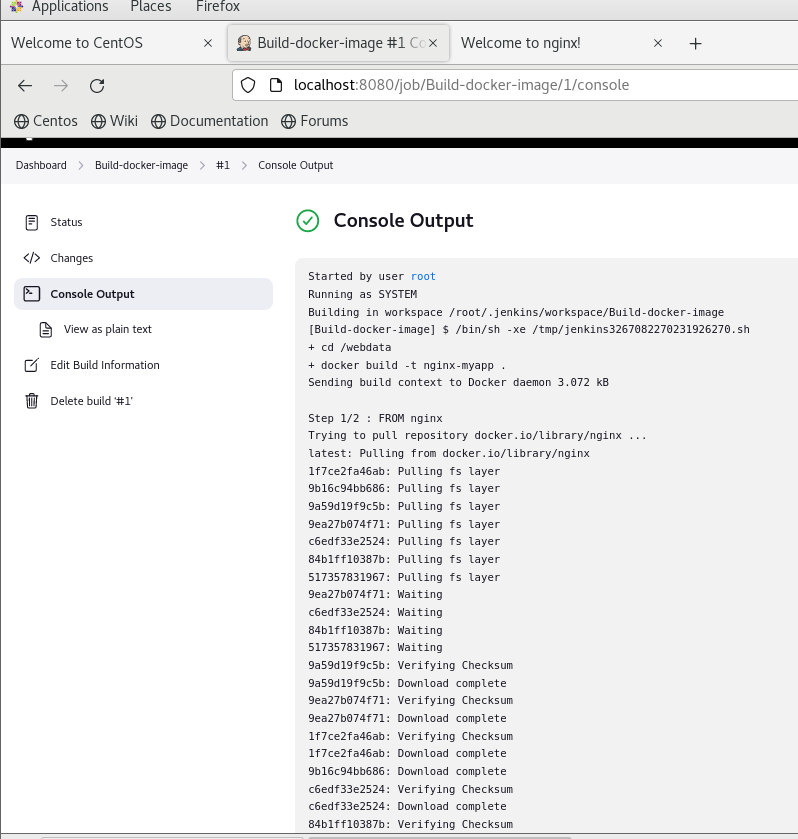
FROM nginx

COPY index.html /usr/share/nginx/html/



]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE



# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

nginx-myapp latest ef1ab686b851 About a minute ago 187 MB

Now we manually save docker image as tar .

# docker save nginx-myapp > nginx-myapp.tar

[root@master1 webdata]# ls

Dockerfile index.html nginx-myapp.tar

# scp nginx-myapp.tar 192.168.80.173:/root

nginx-myapp.tar

Now on nod1:

[root@node1 ~]# docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

docker.io/nginx latest a6bd71f48f68 3 weeks ago 187 MB

[root@node1 ~]# ls

anaconda-ks.cfg initial-setup-ks.cfg join nginx-myapp.tar

[root@node1 ~]# docker load < nginx-myapp.tar

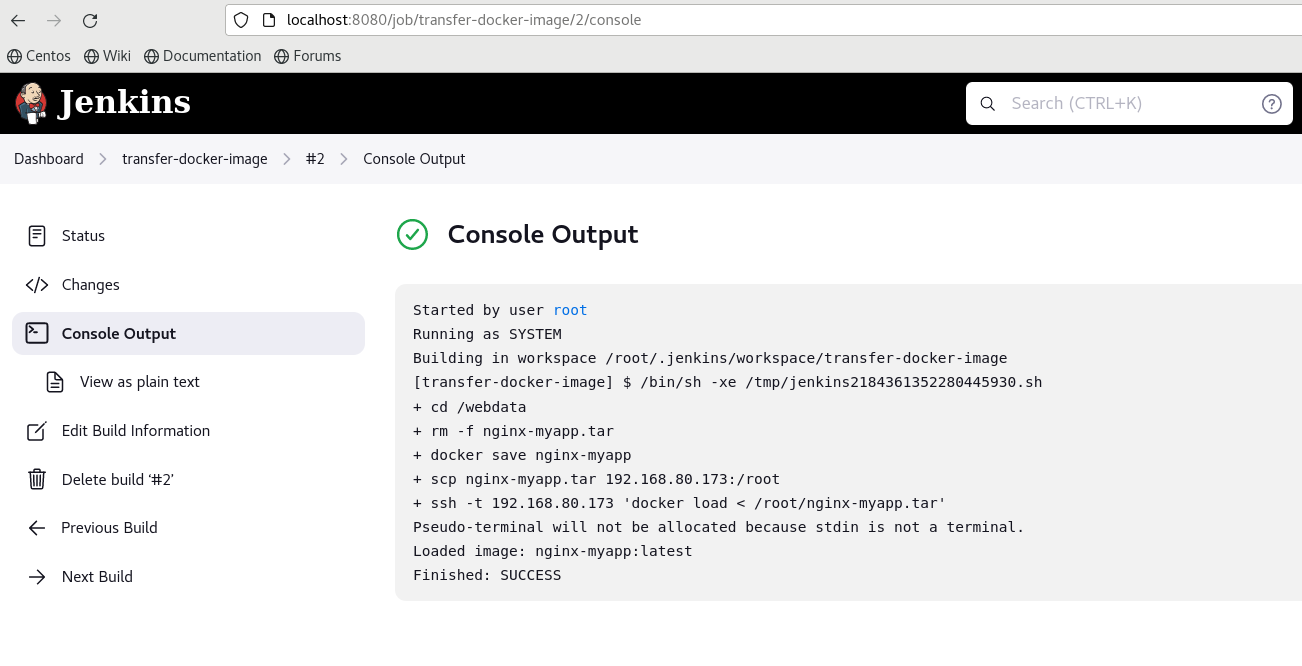
c6060ef9741a: Loading layer [==================================================>] 4.096 kB/4.096 kB

Loaded image: nginx-myapp:latest

[root@node1 ~]# docker images

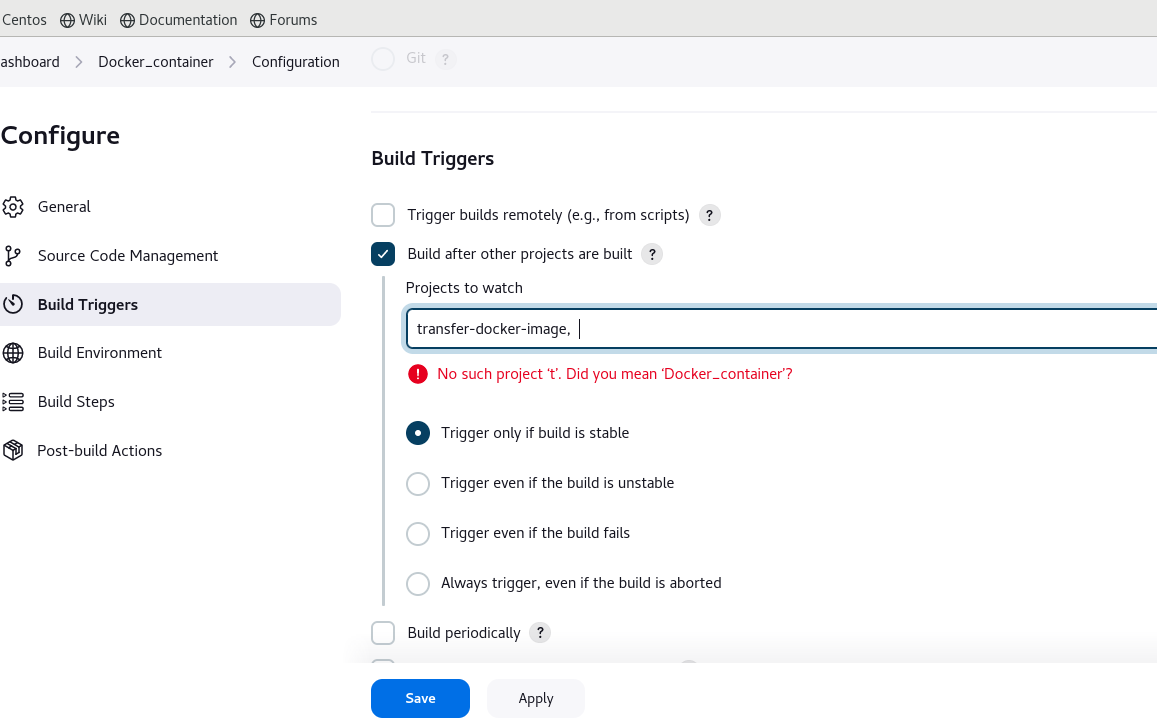
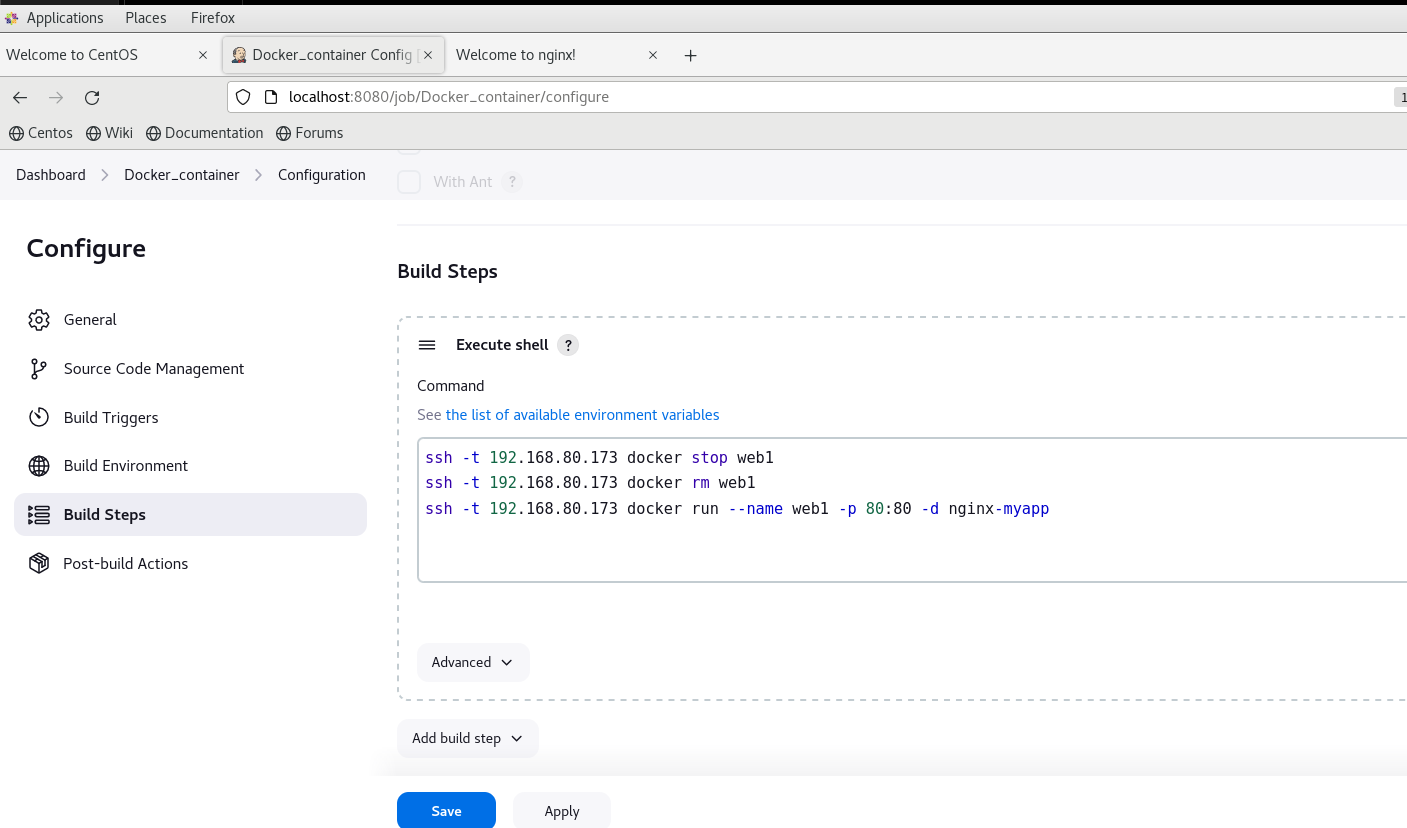
REPOSITORY TAG IMAGE ID CREATED SIZE

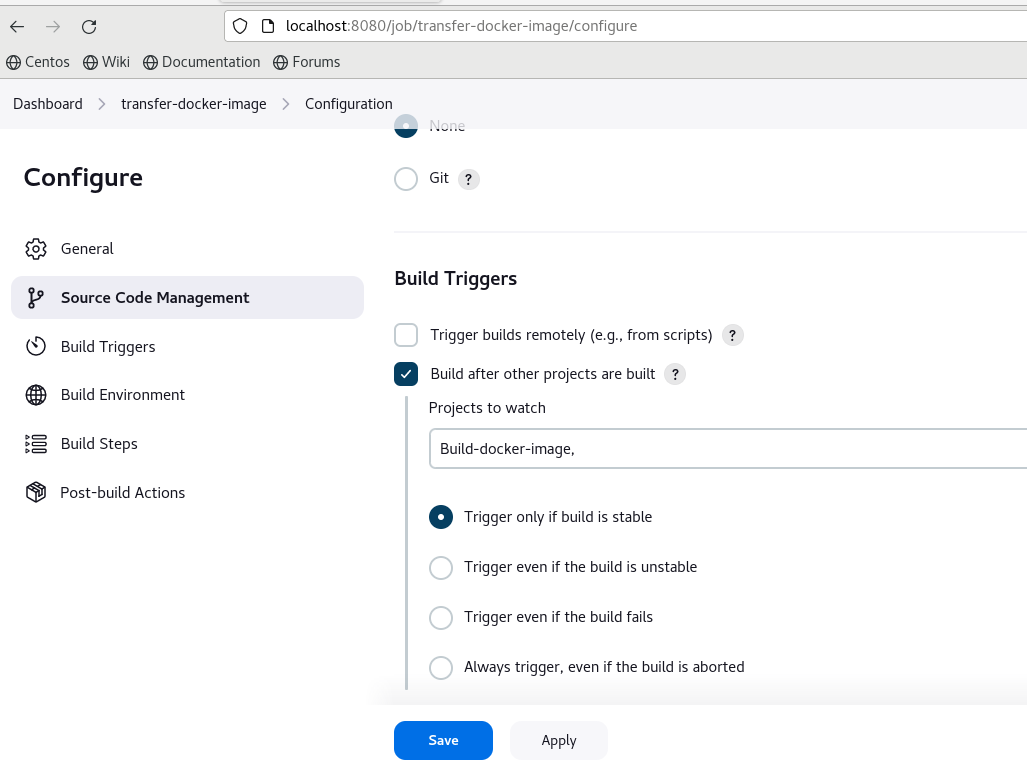
nginx-myapp latest ef1ab686b851 9 minutes ago 187 MB

We did manually, Now Same thing we will do using Jenkins. We will create new job to transfer image on node1 .

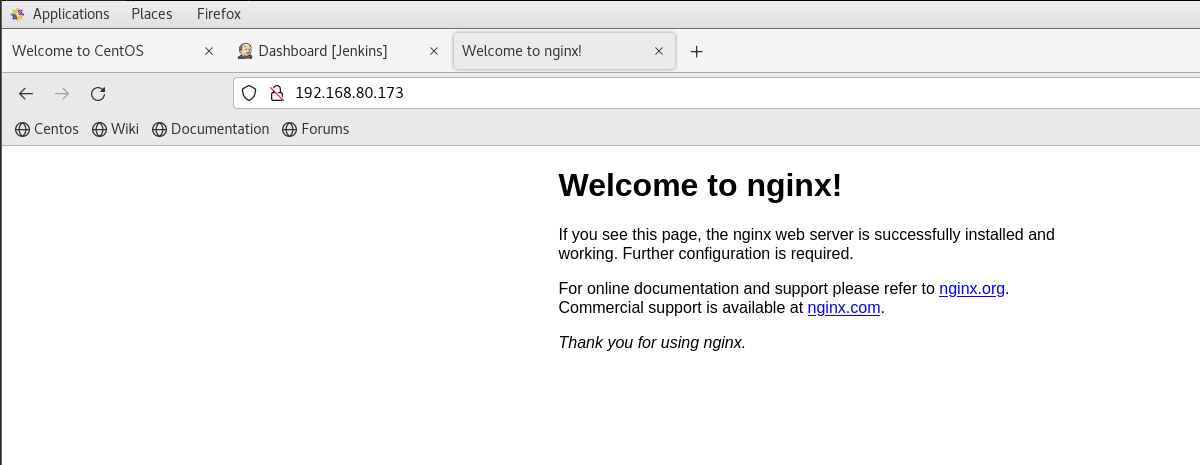
Now we have to link above 3 jobs to each other. So when we modify index.html, it will create new image and it will copy to node 1. One Node 1, will stop and delete running container and it will create new container.

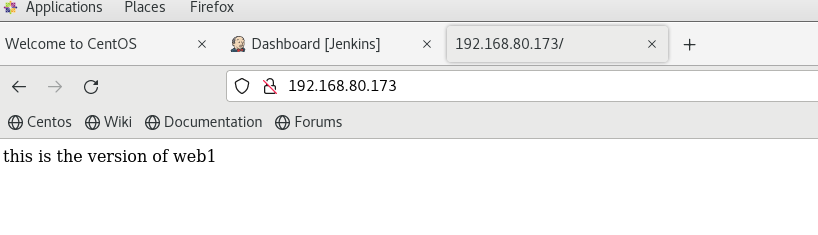
Now Go to Docker\_container job and link to transfer-docker-image



Now we will link transfer-docker-image job to Build-docker-image job.

On Node1 , old web1 container is running…



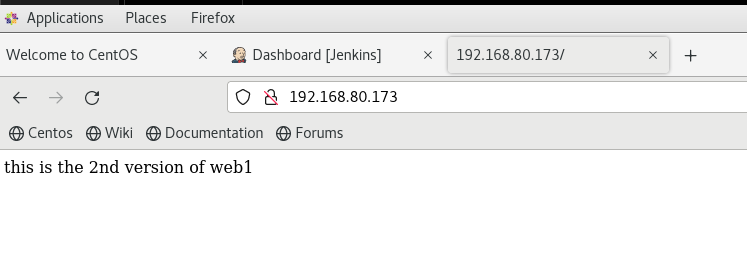
Now we will build Build-docker-image job and remaining job will automatically get triggered.

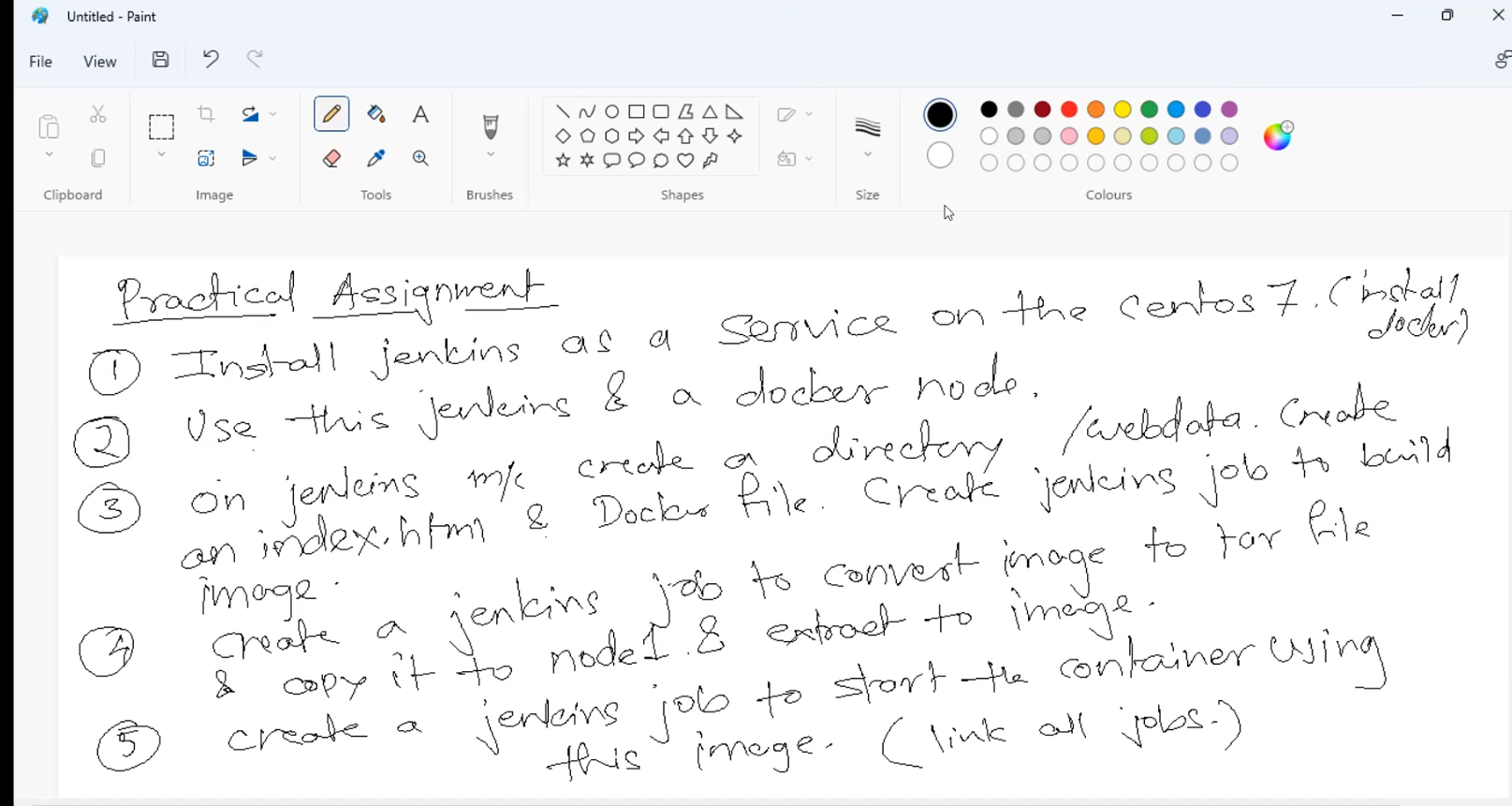
Now we will modify index.html and

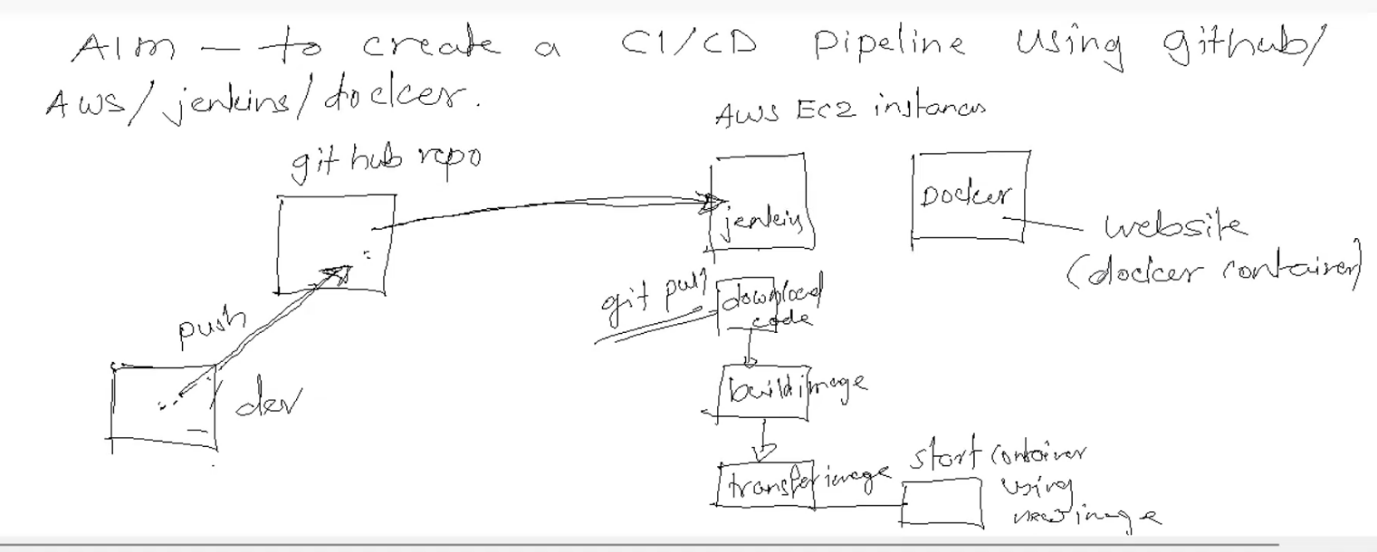
# cat index.html

this is the 2nd version of web1

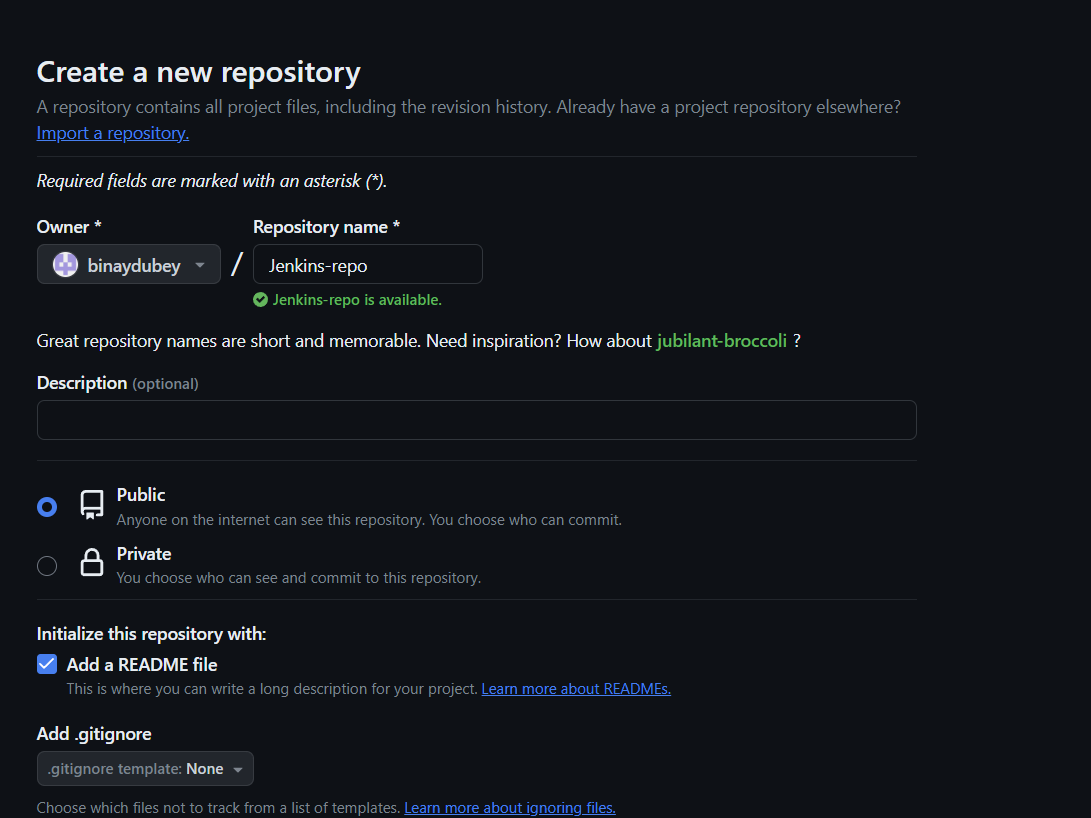
Again bulid Build-docker-image job.

Output:

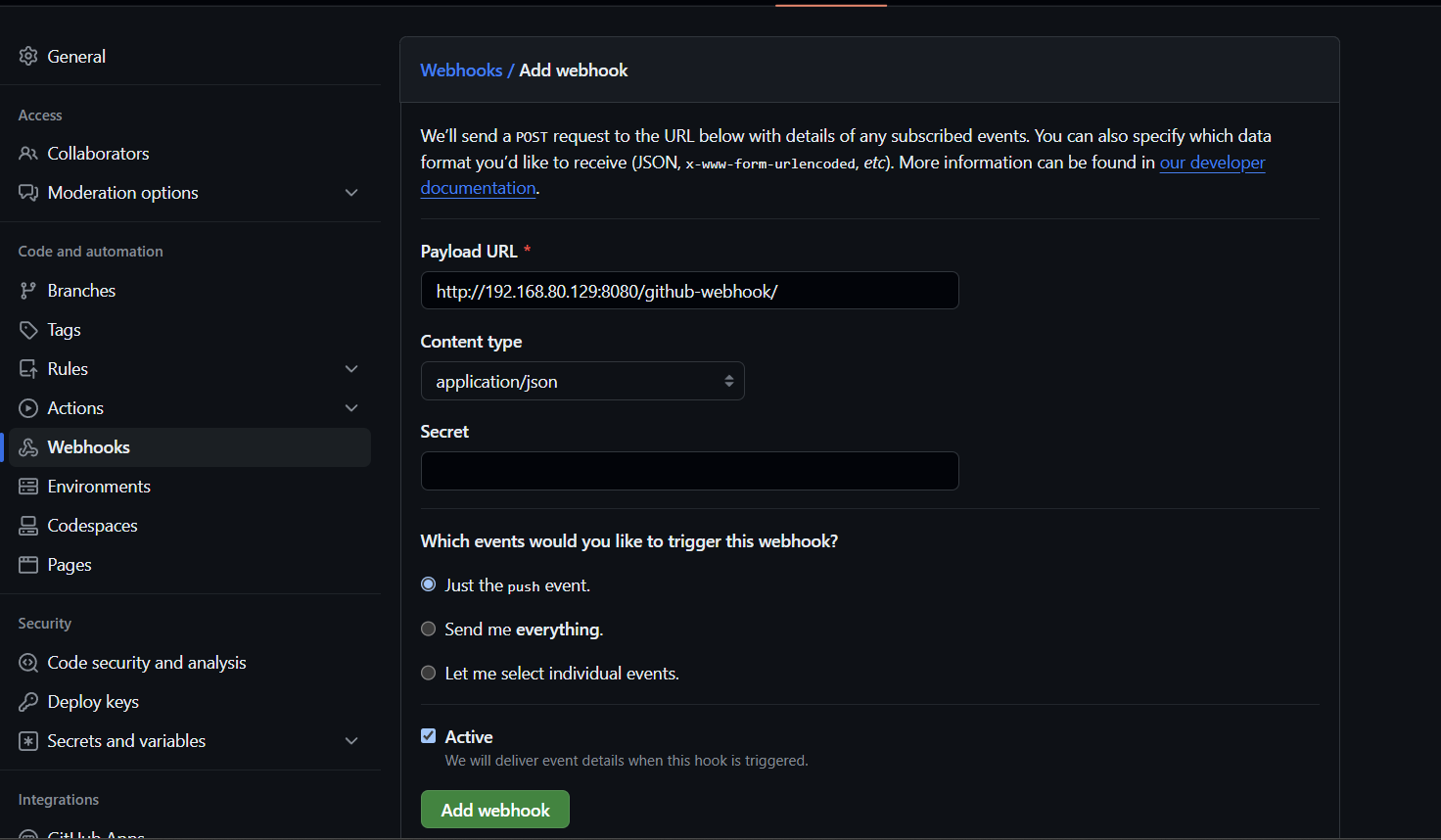




CI/CD Pipeline:------------

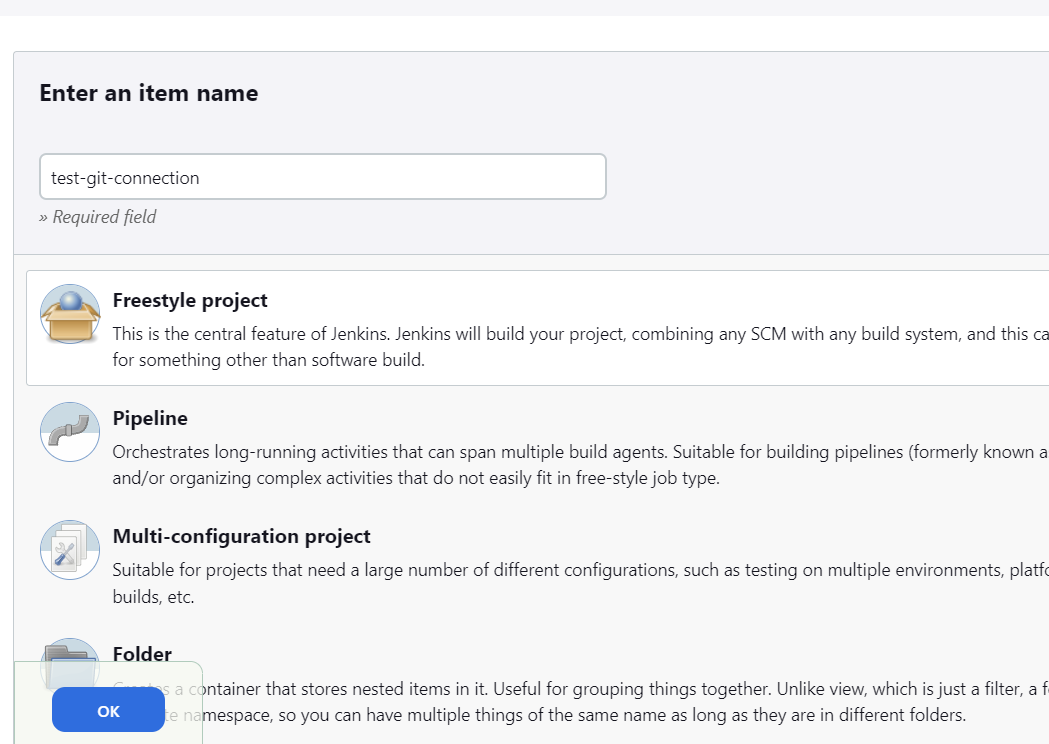
Creating new repo :

Go to Setting and add webhook:

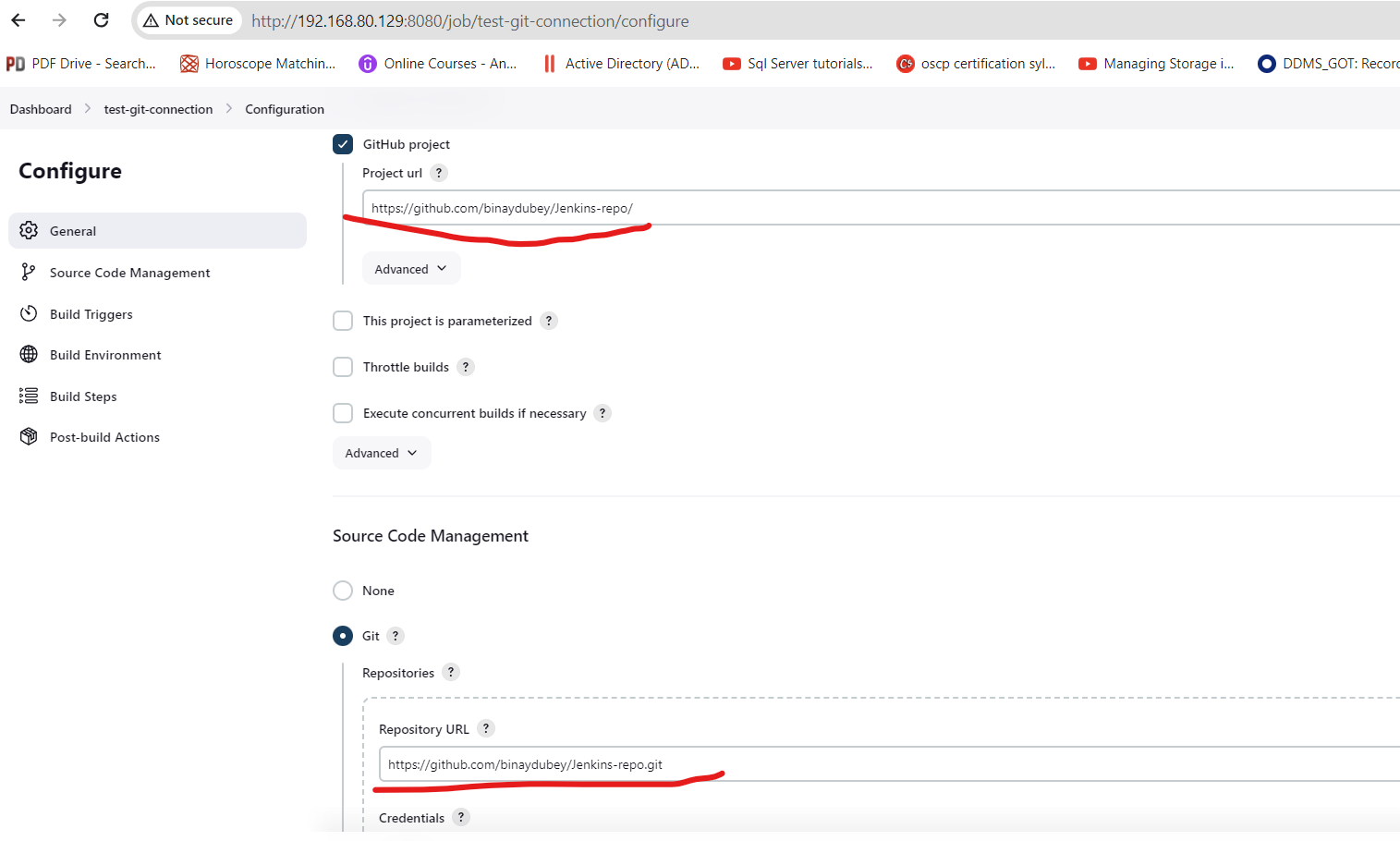
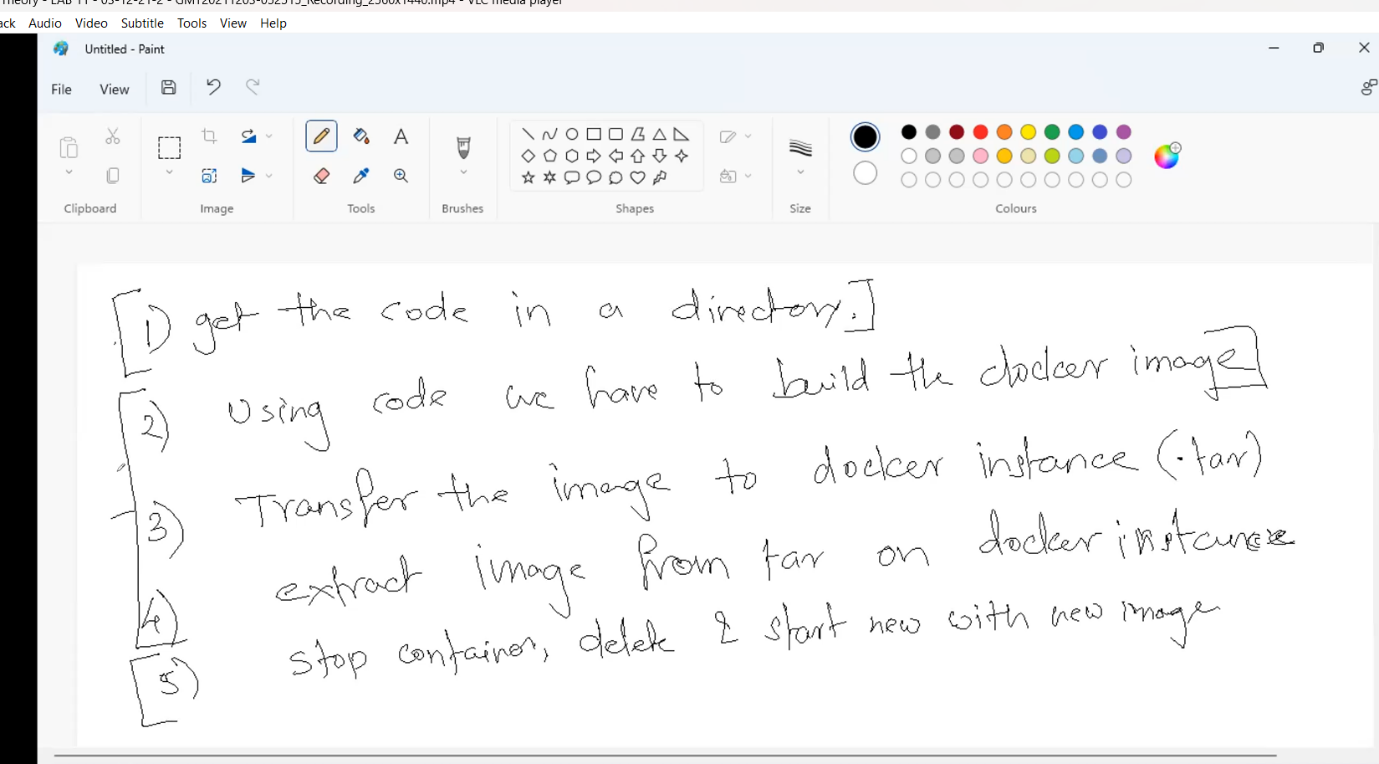


Install git on jenkin:-

Create a freestyle project



Add your repo :-



1: get the code

# mkdir /git/

[root@localhost ~]# cd /git/

[root@localhost git]# git clone https://github.com/binaydubey/Jenkins-repo

Cloning into 'Jenkins-repo'...

remote: Enumerating objects: 6, done.

remote: Counting objects: 100% (6/6), done.

remote: Compressing objects: 100% (3/3), done.

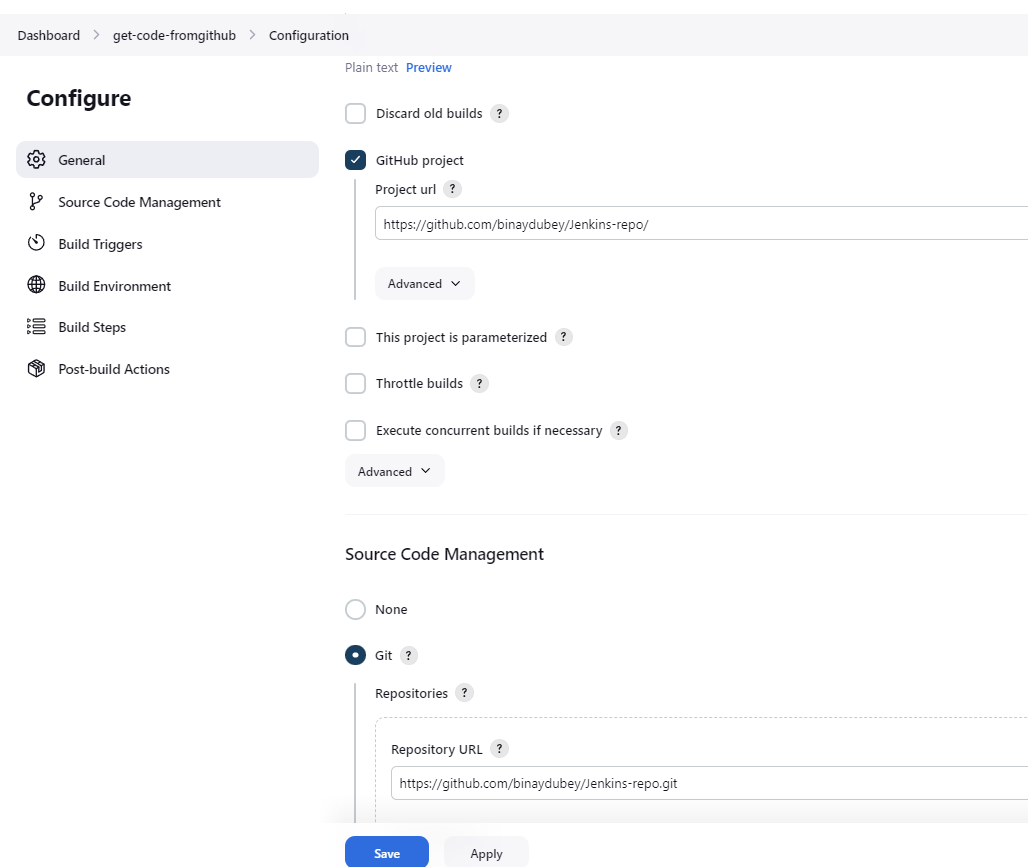
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0

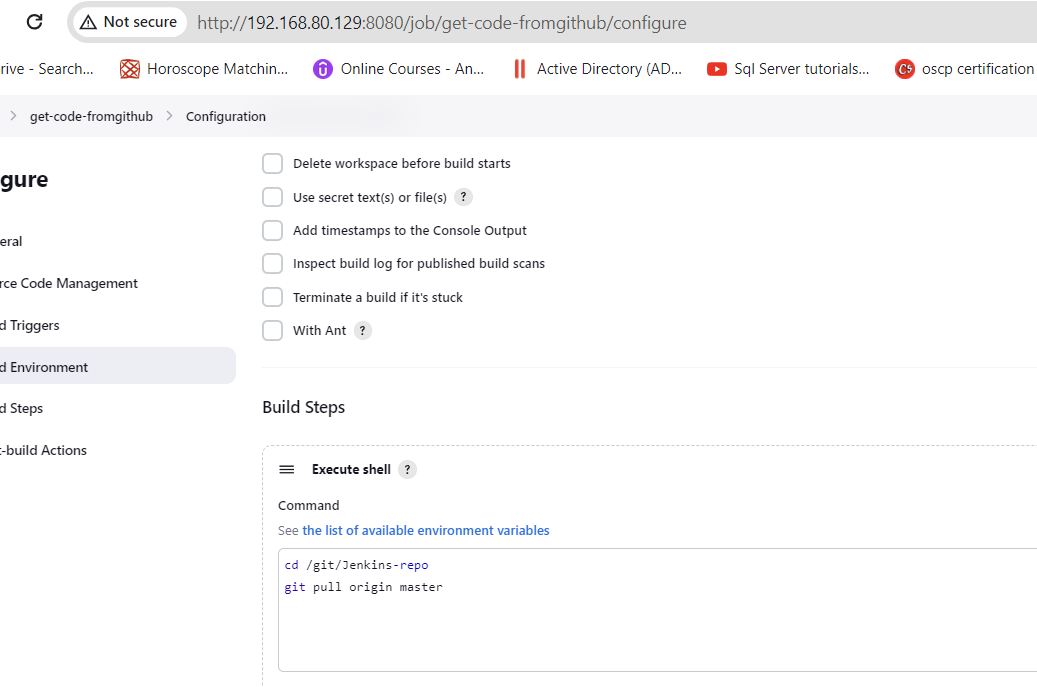
Unpacking objects: 100% (6/6), done.

[root@localhost git]# ls

Jenkins-repo

]# chown -R jenkins /git/Jenkins-repo/

Create a job to pull code from github



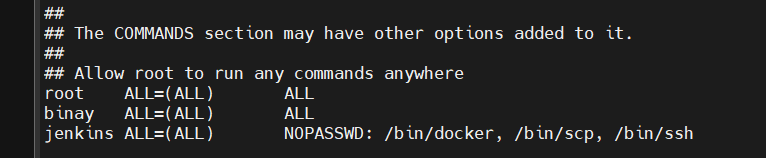
2: we have to build docker image

root@localhost Jenkins-repo]# cat Dockerfile

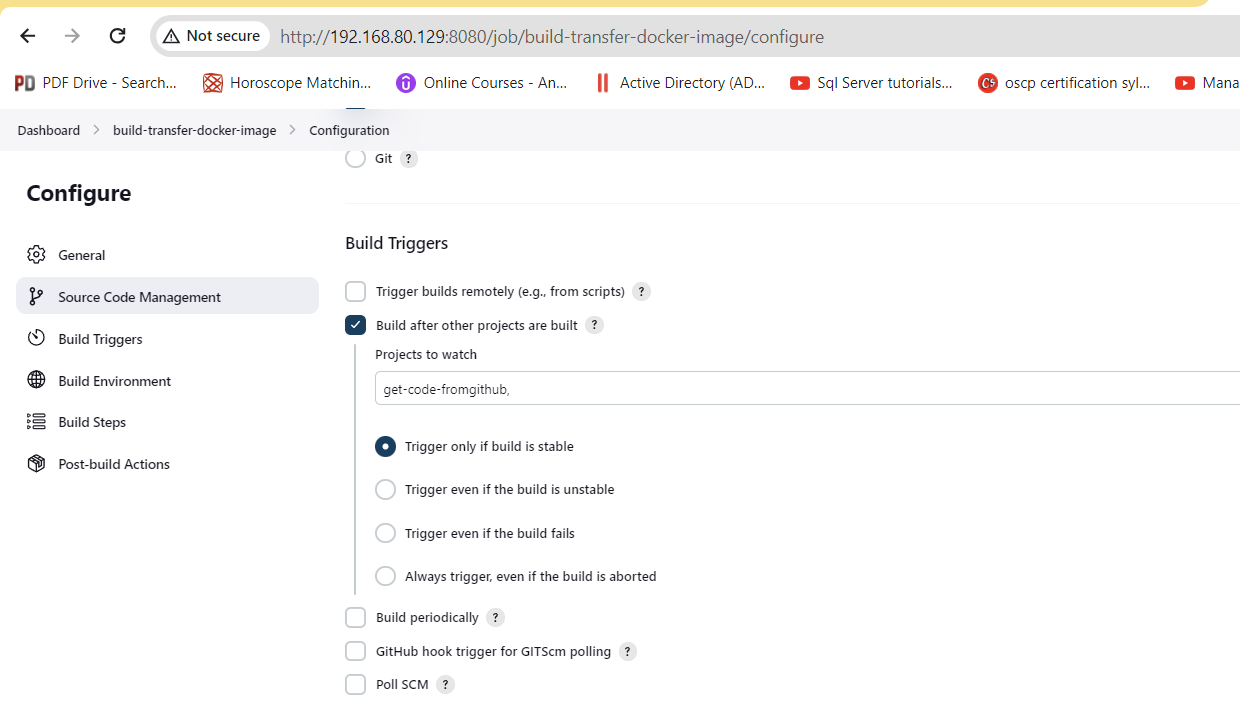
FROM nginx

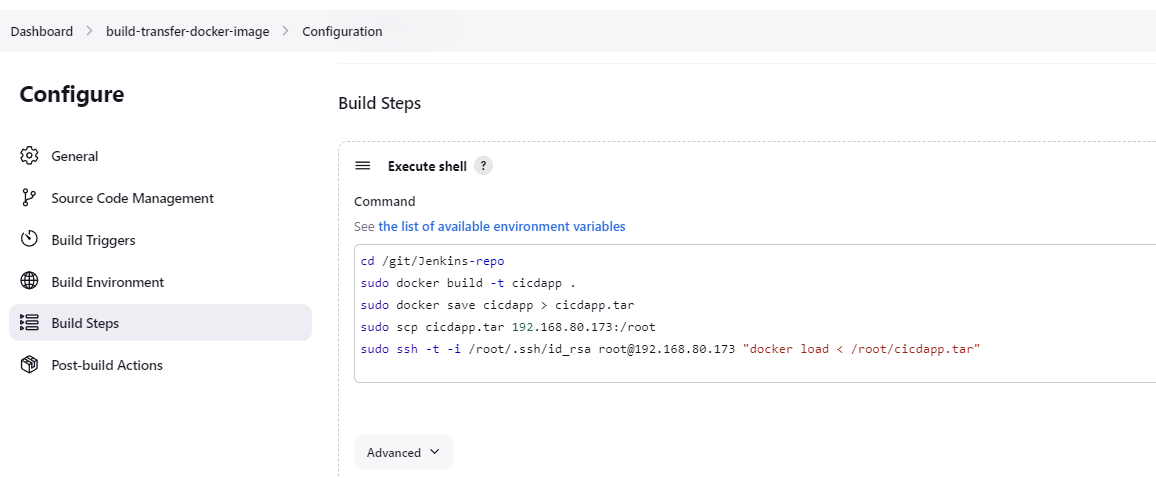
COPY index.html /usr/share/nginx/html

Give permission to run ssh,docker and scp command to Jenkins user without password

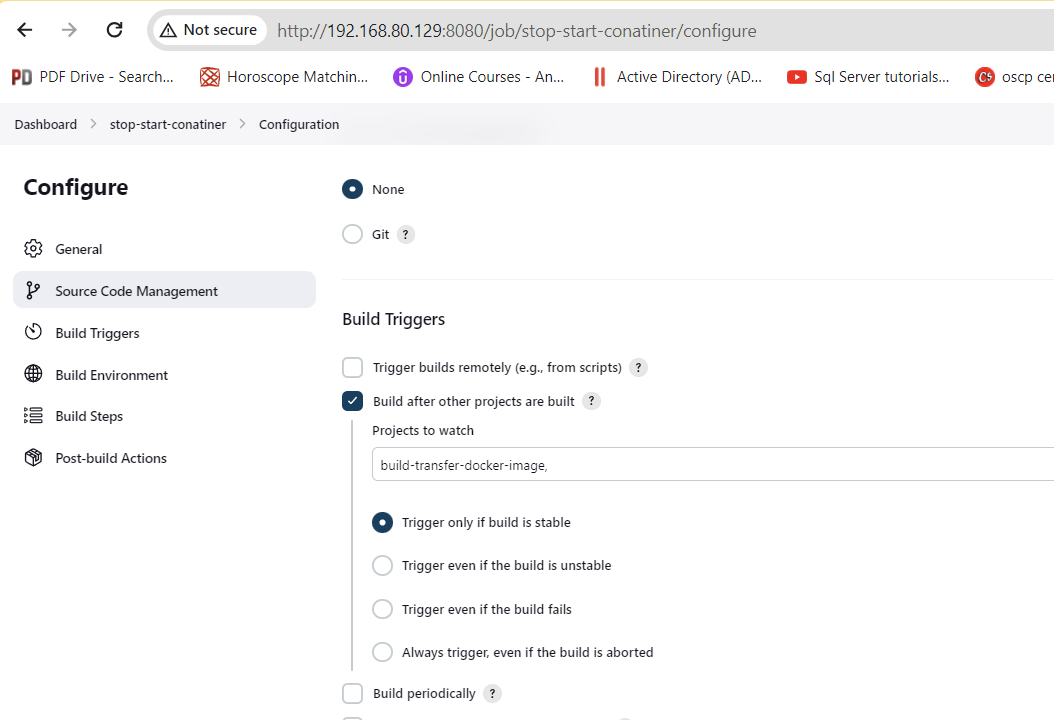


Creating new job to create and transfer image

****

****

Create 3rd job to start and stop container

****

On node you have to create a contaiber manually

# docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

[root@ditiss ~]# docker run --name web1 -p 80:80 -d cicdapp

7bb423f51fd4e90b3a5a46cdc6fbb43d34bd07191676d90a4558cb03cf49d21e

[root@ditiss ~]# curl loclahost

^C

[root@ditiss ~]# curl localhost

this is index

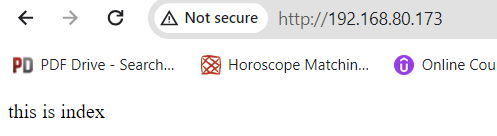
[root@ditiss ~]# curl 192.168.80.173

this is index

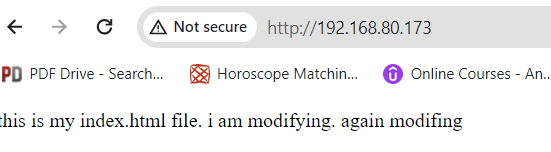
[root@ditiss ~]# firewall-cmd --add-port=80/tcp

success

[root@ditiss ~]# firewall-cmd --add-port=80/tcp --permanent

****success

**Now we will do changes in github’s index.html file**

****