_________________________________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>>>>>>>>>

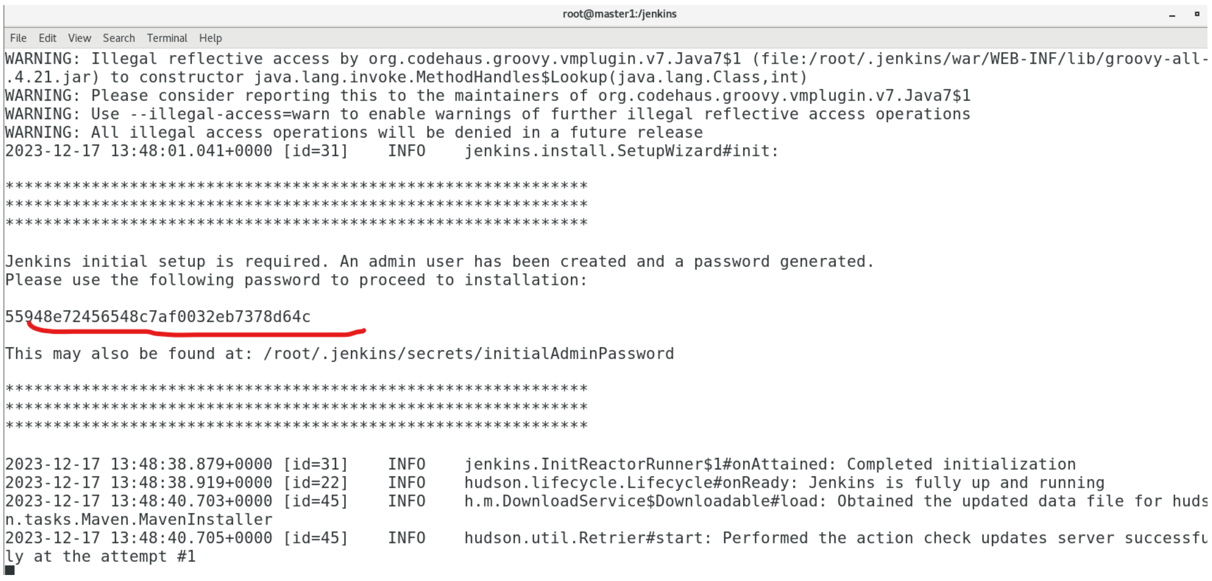
Download jenkins war file..........

```
 mkdir /jenkins
```

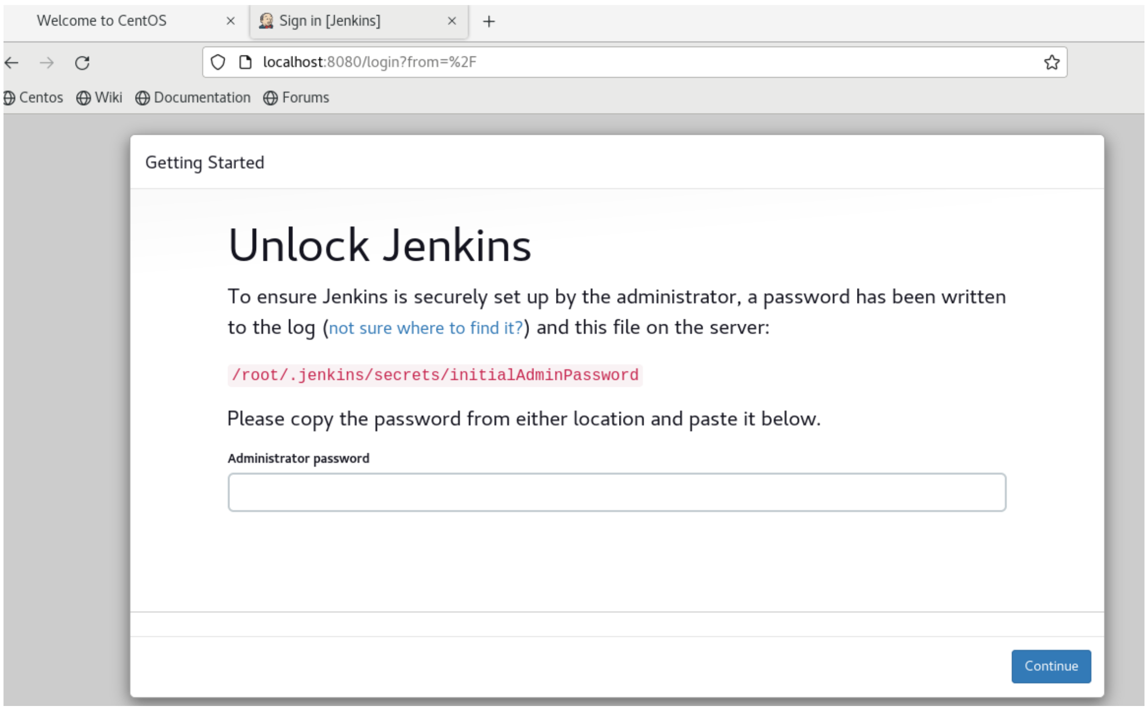
```
[root@master1 Downloads]# cp /home/binay/Downloads/jenkins.war /jenkins/

[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
```



```
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (file:/root/.jenkins/war/WEB-INF/lib/groovy-all-
.4.21.jar) to constructor java.lang.invoke.MethodHandles$Lookup(java.lang.Class,int)
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.vmplugin.v7.Java7$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
2023-12-17 13:48:01.041+0000 [id=31]    INFO    jenkins.install.SetupWizard#init:

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

55948e72456548c7af0032eb7378d64c

This may also be found at: /root/.jenkins/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************

2023-12-17 13:48:38.879+0000 [id=31]    INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2023-12-17 13:48:38.919+0000 [id=22]    INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2023-12-17 13:48:40.703+0000 [id=45]    INFO    h.m.DownloadService$Downloadable#load: Obtained the updated data file for huds
n.tasks.Maven.MavenInstaller
2023-12-17 13:48:40.705+0000 [id=45]    INFO    hudson.util.Retrier#start: Performed the action check updates server successfu
ly at the attempt #1
```


Welcome to CentOS

Sign in [Jenkins]

localhost:8080/login?from=%2F

Centos Wiki Documentation Forums

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
/root/.jenkins/secrets/initialAdminPassword
```

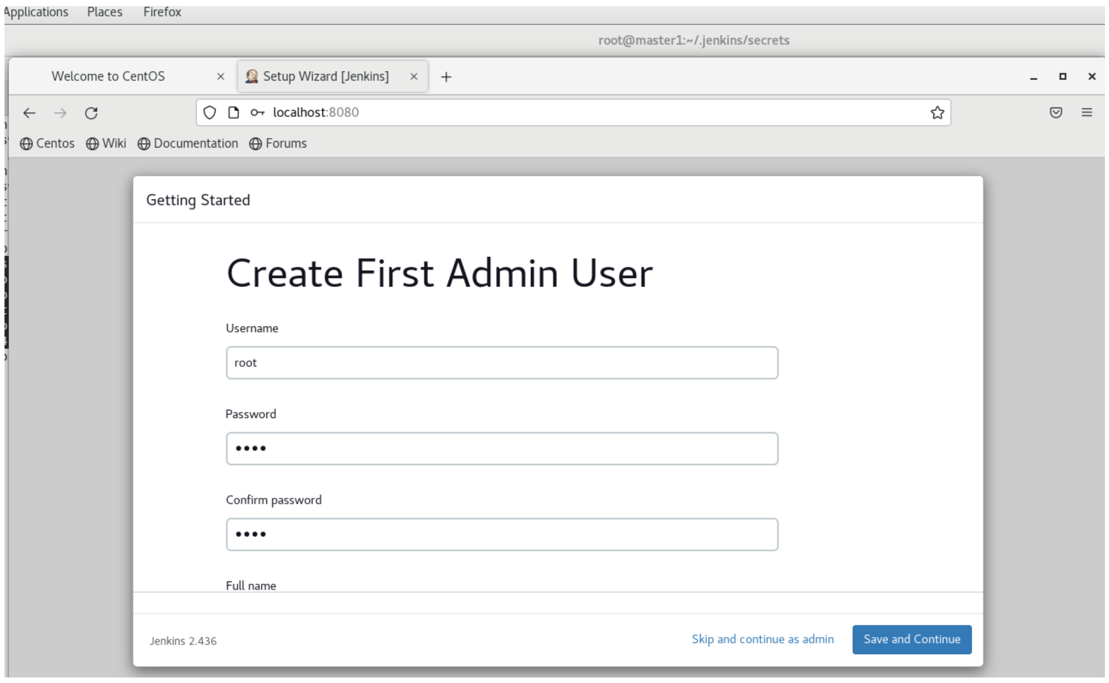
Please copy the password from either location and paste it below.

Administrator password

Continue

```
# 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:--------



Create First Admin User

Username: root

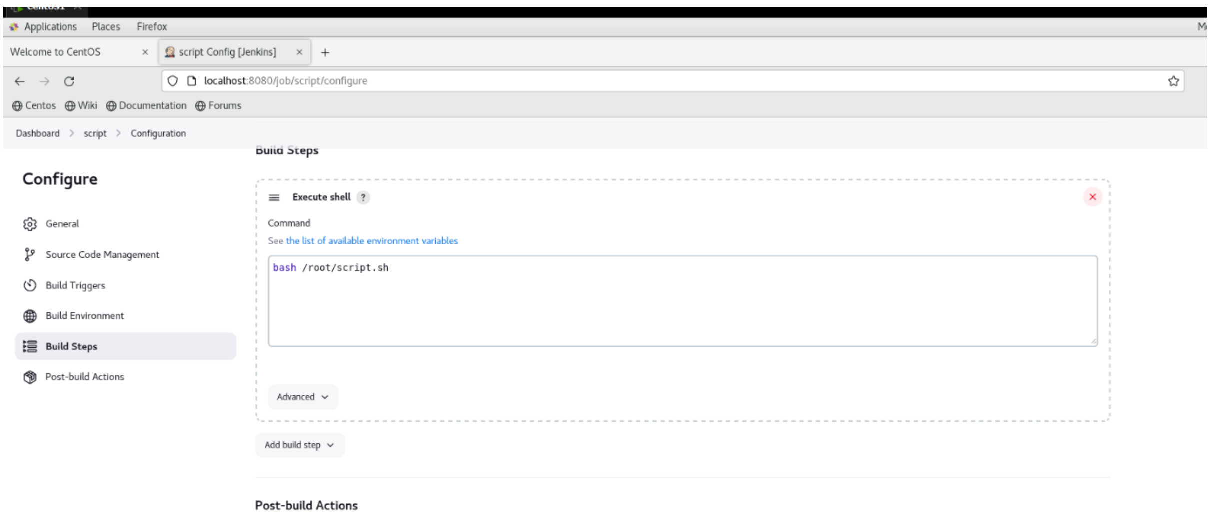
Password: ••••

Confirm password: ••••

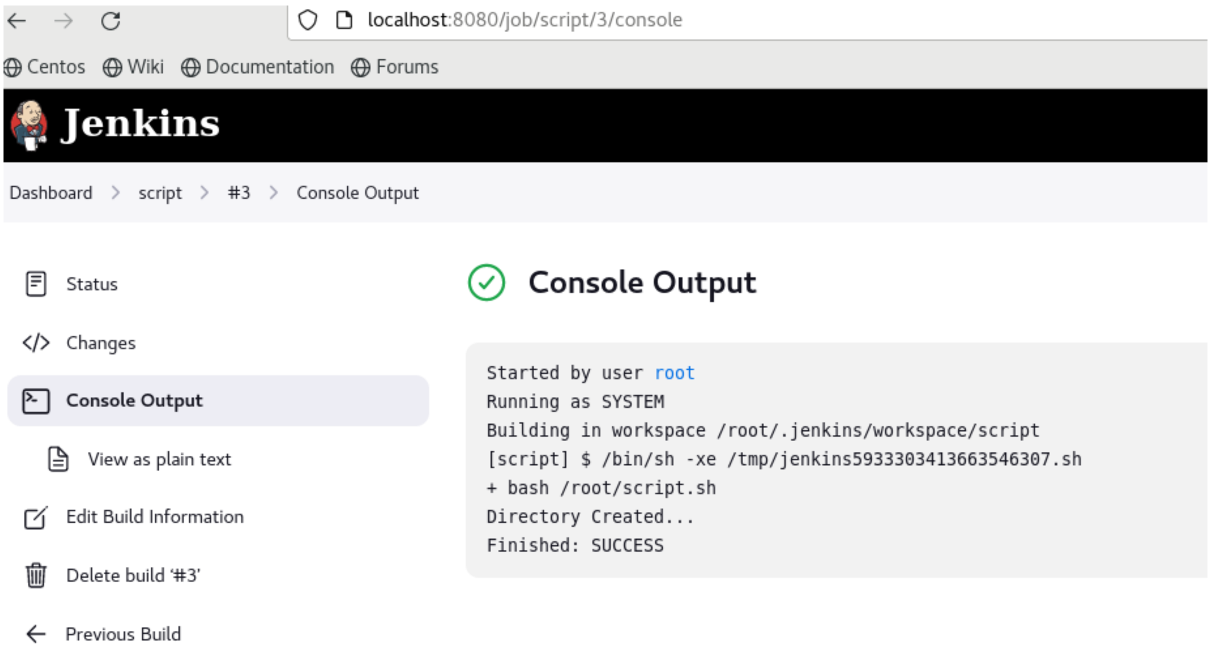
Full name

1) Create a shell script that will first check if a directory exists, if directory exists it will echo message "directory already exists", or if it does not exists it will create the directory.

2) If the script is successful create a job in jenkins to execute this script.



```
bash /root/script.sh
```

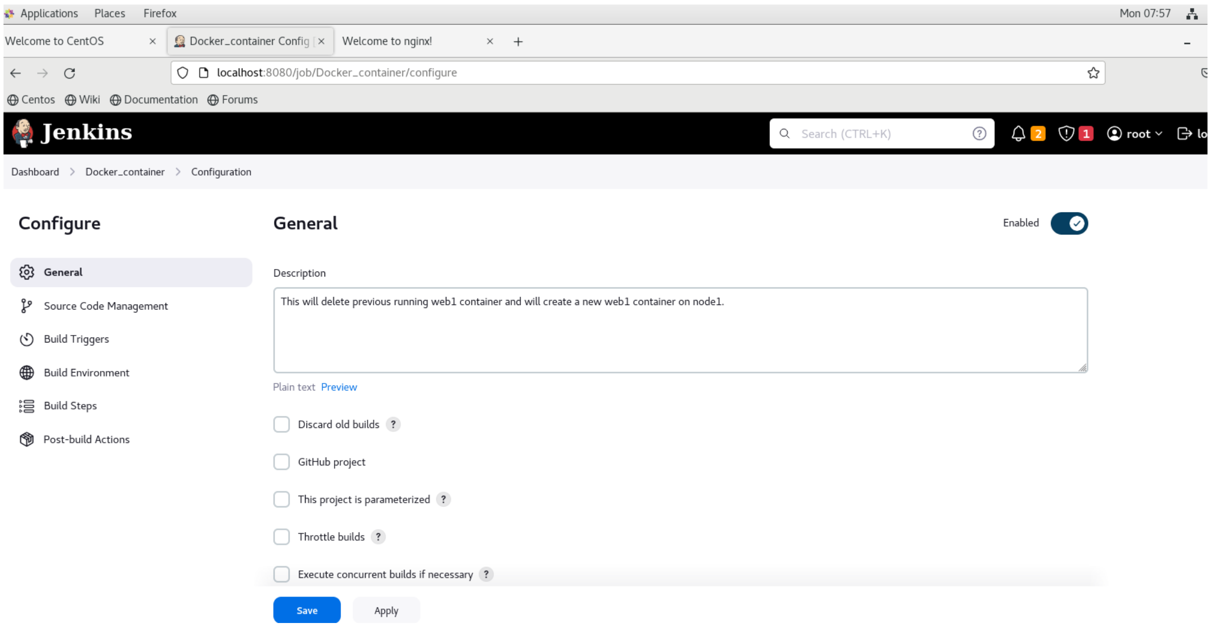


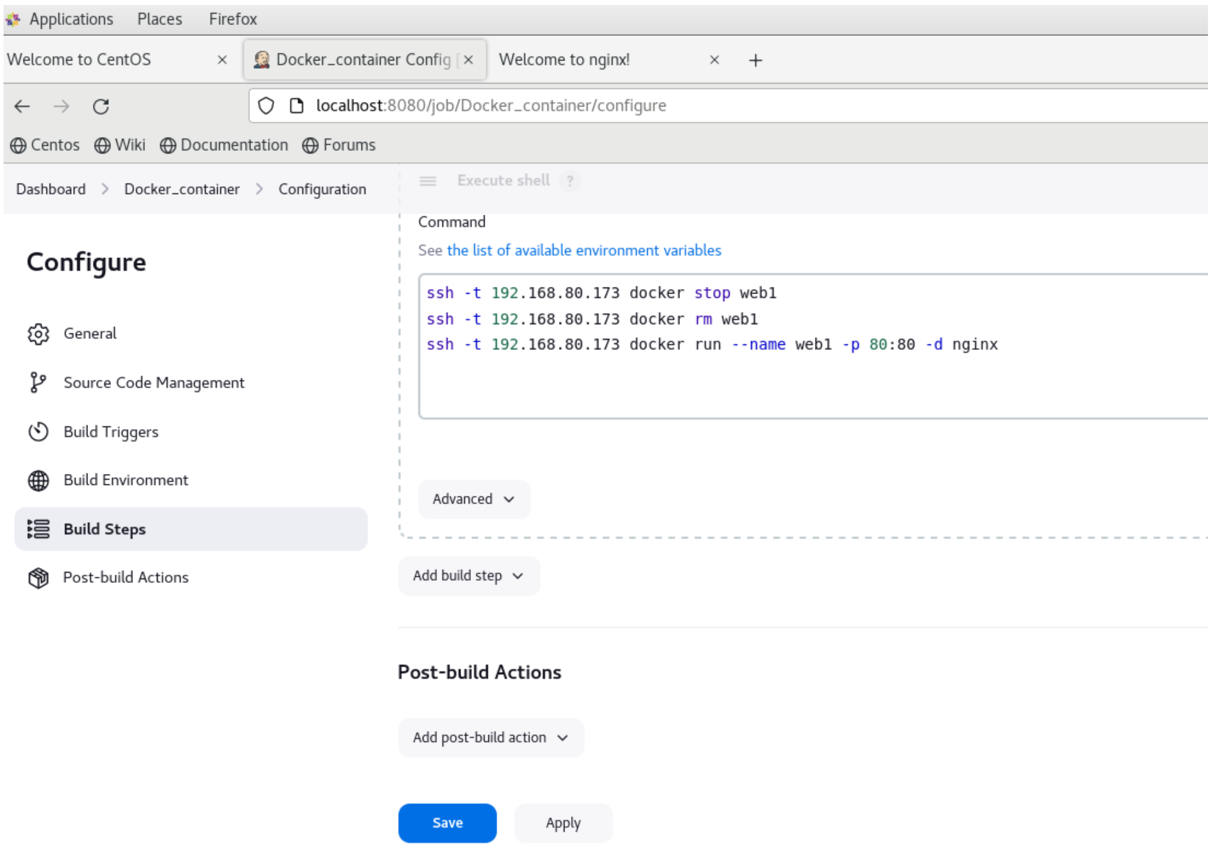
```
Started by user root
Running as SYSTEM
Building in workspace /root/.jenkins/workspace/script
[script] $ /bin/sh -xe /tmp/jenkins5933303413663546307.sh
+ bash /root/script.sh
Directory Created...
Finished: SUCCESS
```

```
# cat script.sh
if [[ -d "test" ]]
then
        echo "Directory exists..."
else
        mkdir test
        echo "Directory Created..."
fi
```

2nd Assignment:

Configure Passwordless ssh

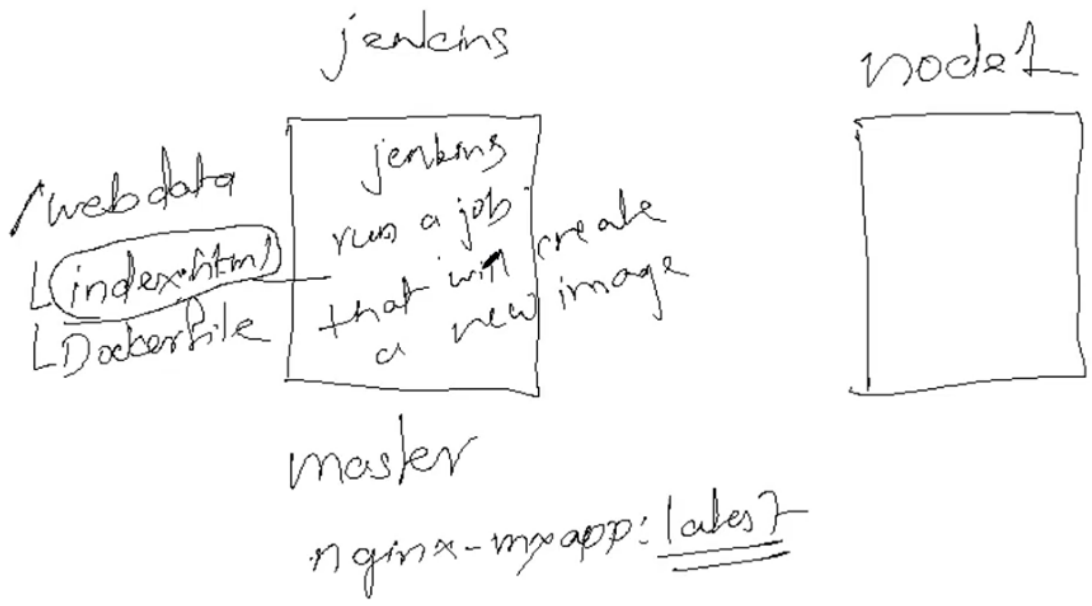

Description: This will delete previous running web1 container and will create a new web1 container on node1.


Execute shell

Command

```
ssh -t 192.168.80.173 docker stop web1
ssh -t 192.168.80.173 docker rm web1
ssh -t 192.168.80.173 docker run --name web1 -p 80:80 -d nginx
```

3rd assignment:-

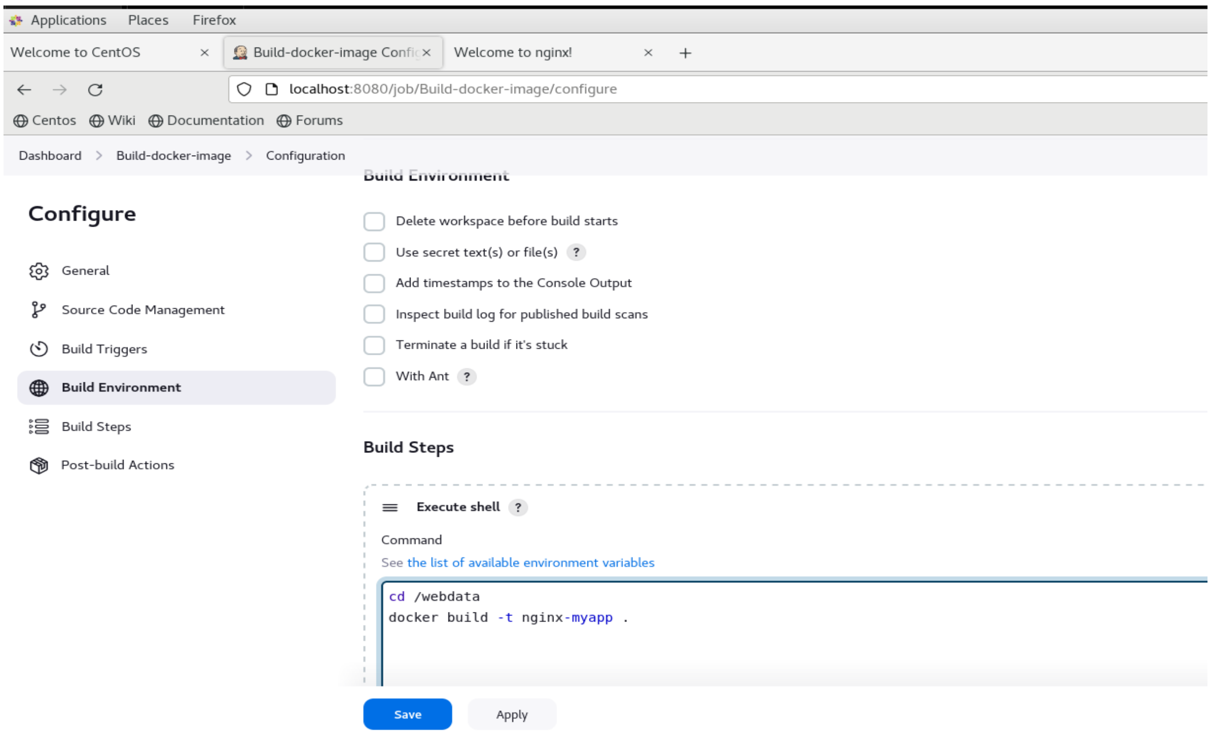


In this question, we create a docker image. And we will convert it into tar and will copy to node1.

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/
```



Build Environment

- [ ] Delete workspace before build starts
- [ ] Use secret text(s) or file(s)
- [ ] Add timestamps to the Console Output
- [ ] Inspect build log for published build scans
- [ ] Terminate a build if it's stuck
- [ ] With Ant

Build Steps

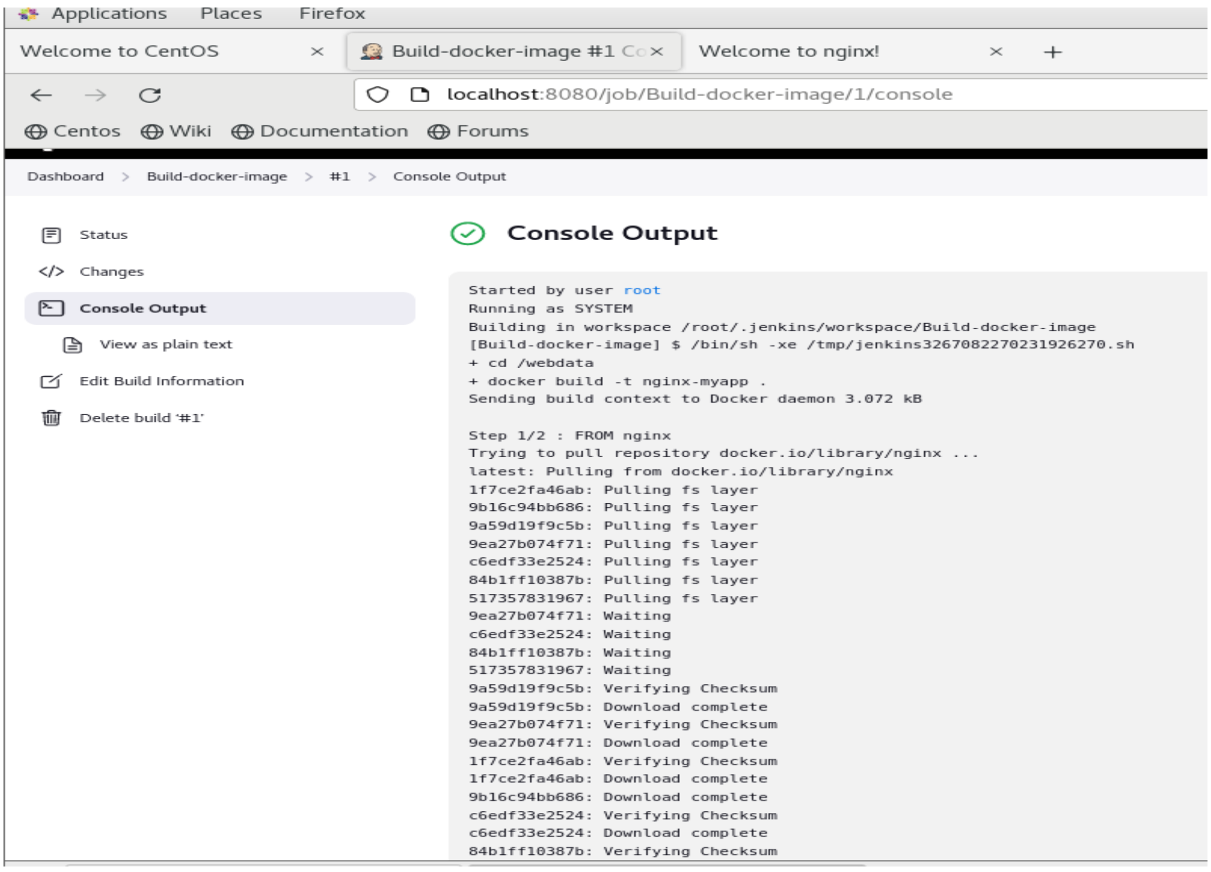
Execute shell

Command

See the list of available environment variables

```
cd /webdata
docker build -t nginx-myapp .
```

```
]# docker images
REPOSITORY      TAG         IMAGE ID       CREATED        SIZE
```



Console Output

```
Started by user root
Running as SYSTEM
Building in workspace /root/.jenkins/workspace/Build-docker-image
[Build-docker-image] $ /bin/sh -xe /tmp/jenkins3267082270231926270.sh
+ cd /webdata
+ docker build -t nginx-myapp .
Sending build context to Docker daemon 3.072 kB

Step 1/2 : FROM nginx
Trying to pull repository docker.io/library/nginx ...
latest: Pulling from docker.io/library/nginx
1f7ce2fa46ab: Pulling fs layer
9b16c94bb686: Pulling fs layer
9a59d19f9c5b: Pulling fs layer
9ea27b074f71: Pulling fs layer
c6edf33e2524: Pulling fs layer
84b1ff10387b: Pulling fs layer
517357831967: Pulling fs layer
9ea27b074f71: Waiting
c6edf33e2524: Waiting
84b1ff10387b: Waiting
517357831967: Waiting
9a59d19f9c5b: Verifying Checksum
9a59d19f9c5b: Download complete
9ea27b074f71: Verifying Checksum
9ea27b074f71: Download complete
1f7ce2fa46ab: Verifying Checksum
1f7ce2fa46ab: Download complete
9b16c94bb686: Download complete
c6edf33e2524: Verifying Checksum
c6edf33e2524: Download complete
84b1ff10387b: Verifying Checksum
```

```
# 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
```

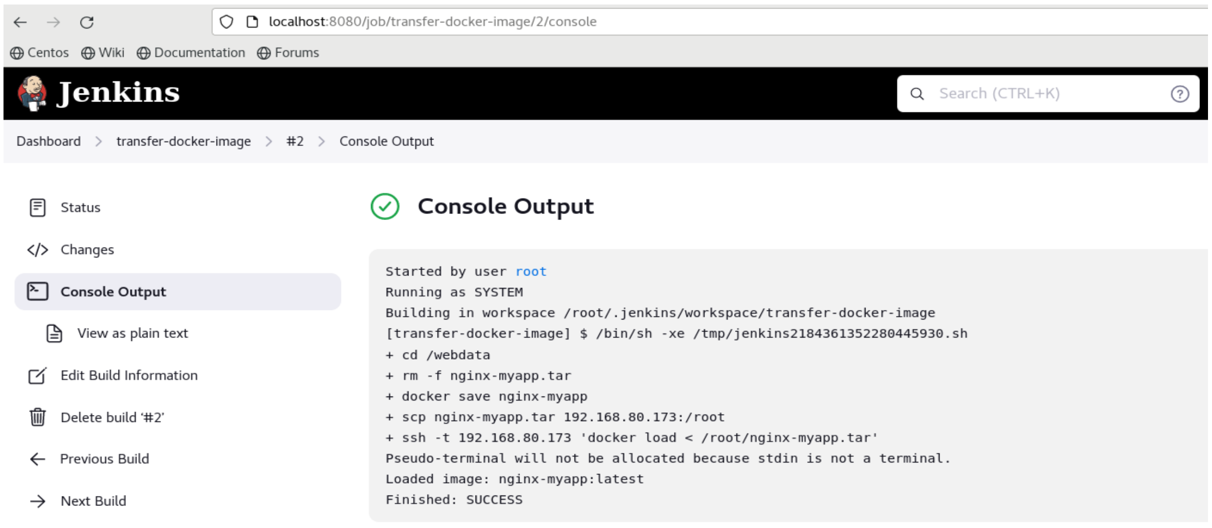
```
# scp nginx-myapp.tar 192.168.80.173:/root
nginx-myapp.tar
```

Now on nod1:

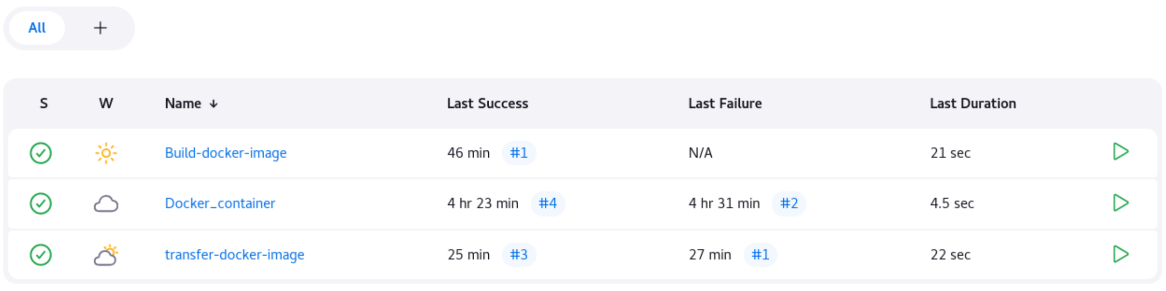
```
[root@node1 ~]# docker images
REPOSITORY        TAG             IMAGE ID          CREATED            SIZE
docker.io/nginx   latest          a6bd71f48f68      3 weeks ago        187 MB
[root@node1 ~]# ls
```

```
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 .



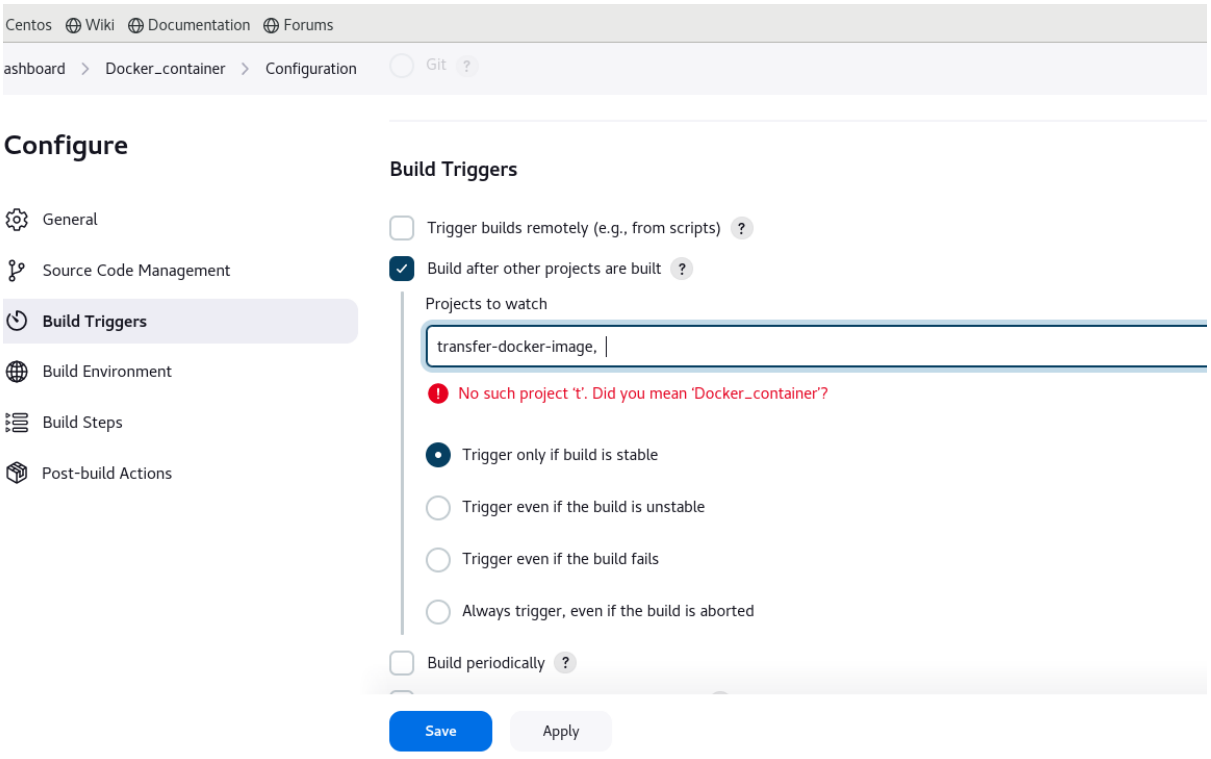
```
Started by user root
Running as SYSTEM
Building in workspace /root/.jenkins/workspace/transfer-docker-image
[transfer-docker-image] $ /bin/sh -xe /tmp/jenkins2184361352280445930.sh
+ cd /webdata
+ rm -f nginx-myapp.tar
+ docker save nginx-myapp
+ scp nginx-myapp.tar 192.168.80.173:/root
+ ssh -t 192.168.80.173 'docker load < /root/nginx-myapp.tar'
Pseudo-terminal will not be allocated because stdin is not a terminal.
Loaded image: nginx-myapp:latest
Finished: SUCCESS
```



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	Build-docker-image	46 min #1	N/A	21 sec
✓	☁	Docker_container	4 hr 23 min #4	4 hr 31 min #2	4.5 sec
✓	⛅	transfer-docker-image	25 min #3	27 min #1	22 sec

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



Centos Wiki Documentation Forums

ashboard > Docker_container > Configuration

Configure

- General
- Source Code Management
- **Build Triggers**
- Build Environment
- Build Steps
- Post-build Actions

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

Projects to watch

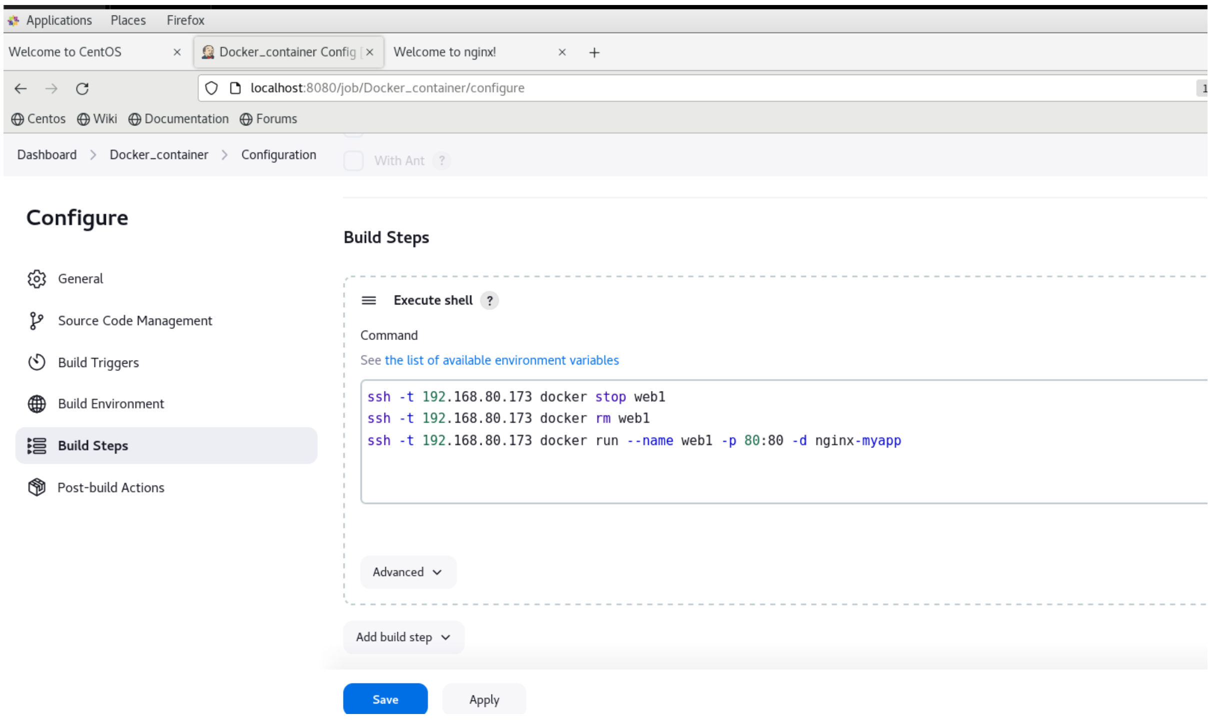
transfer-docker-image,

No such project 't'. Did you mean 'Docker_container'?

- [● Trigger only if build is stable]
- Trigger even if the build is unstable
- Trigger even if the build fails
- Always trigger, even if the build is aborted

- [ ] Build periodically

Save Apply



Applications Places Firefox

Welcome to CentOS | Docker_container Config | Welcome to nginx!

localhost:8080/job/Docker_container/configure

Centos Wiki Documentation Forums

Dashboard > Docker_container > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- **Build Steps**
- Post-build Actions

Build Steps

Execute shell

Command

See the list of available environment variables

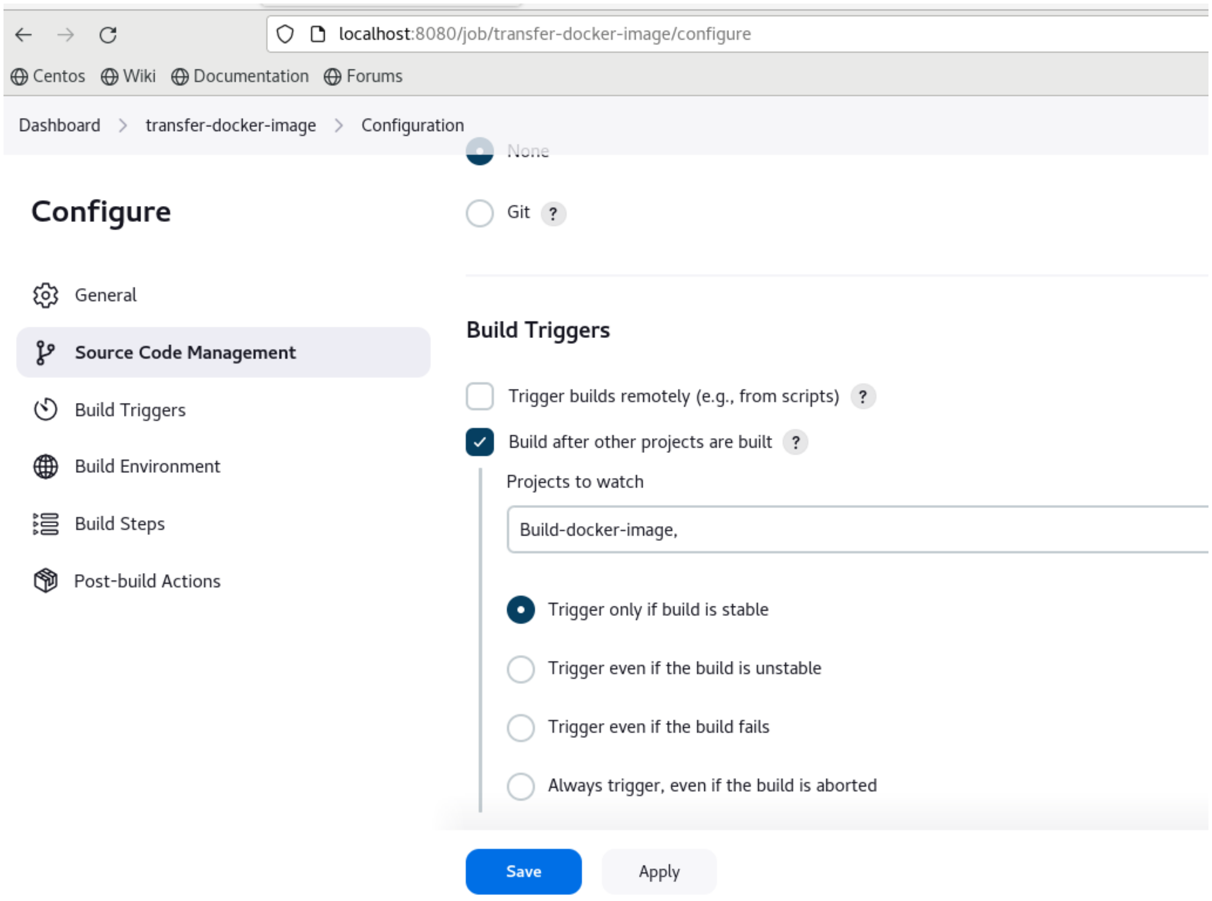
```
ssh -t 192.168.80.173 docker stop web1
ssh -t 192.168.80.173 docker rm web1
ssh -t 192.168.80.173 docker run --name web1 -p 80:80 -d nginx-myapp
```

Advanced

Add build step

Save Apply

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


localhost:8080/job/transfer-docker-image/configure

Dashboard > transfer-docker-image > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

[✓ None] Git

Build Triggers

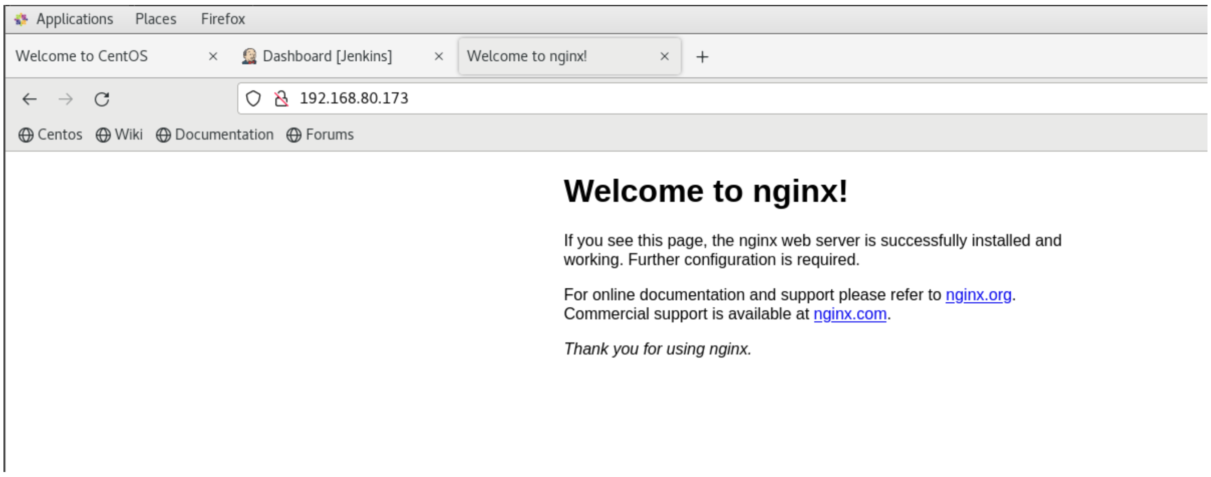
- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

Projects to watch: Build-docker-image,

[✓ Trigger only if build is stable] Trigger even if the build is unstable / Trigger even if the build fails / Always trigger, even if the build is aborted

Save Apply

On Node1 , old web1 container is running…


192.168.80.173

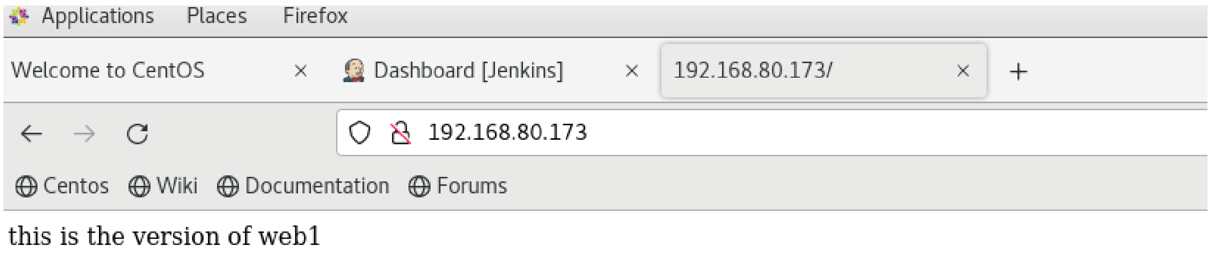
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

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



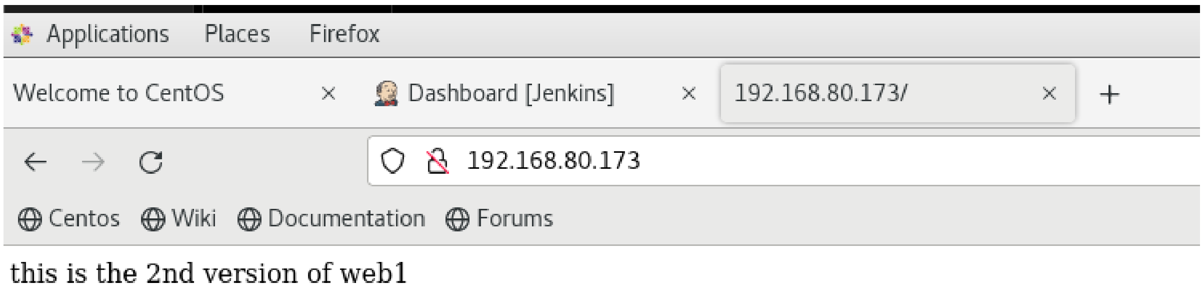
this is the version of web1

Now we will modify index.html and

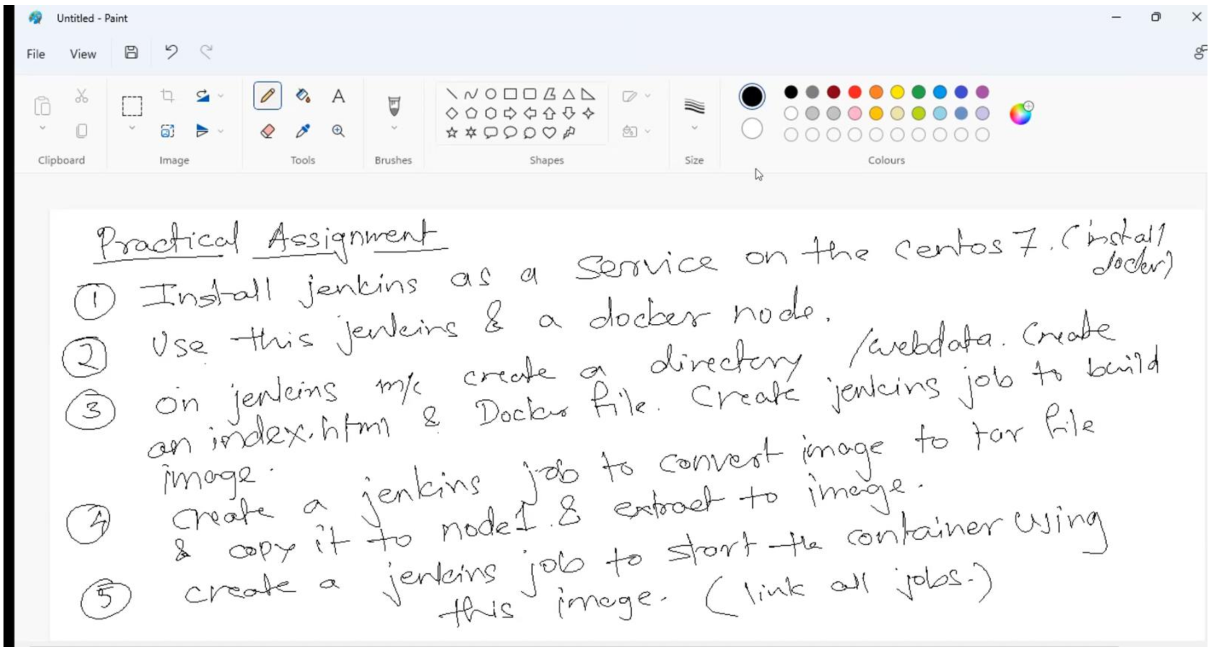
```
# cat index.html
this is the 2nd  version of web1
```

Again bulid Build-docker-image job.

Output:



this is the 2nd version of web1



Practical Assignment

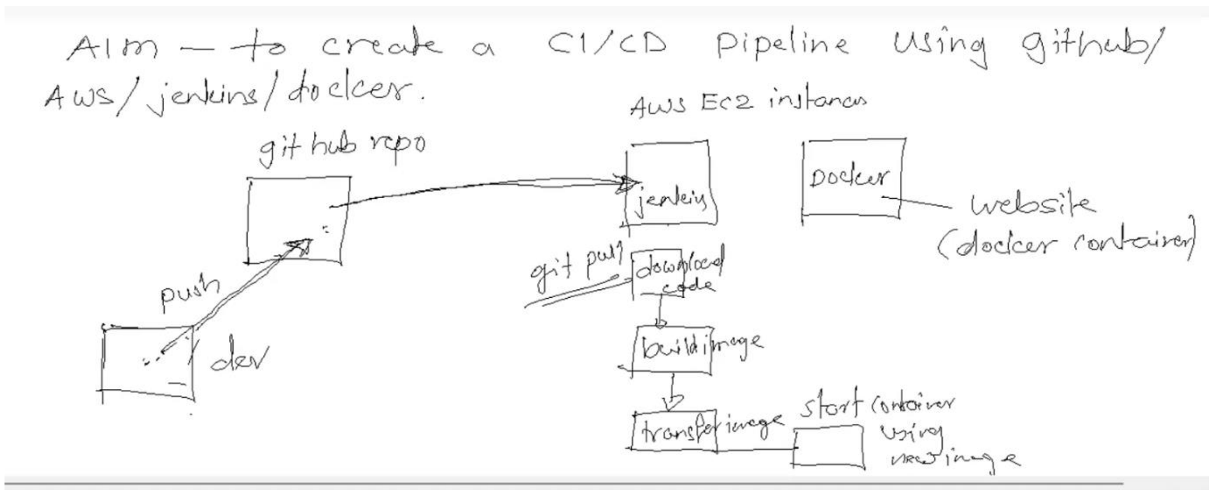
① Install jenkins as a Service on the centos 7. (install docker)

② Use this jenkins & a docker node.

③ on jenkins m/c create a directory /webdata. Create an index.html & Docker file. Create jenkins job to build image.

④ create a jenkins job to convert image to tar file & copy it to node1 & extract to image.

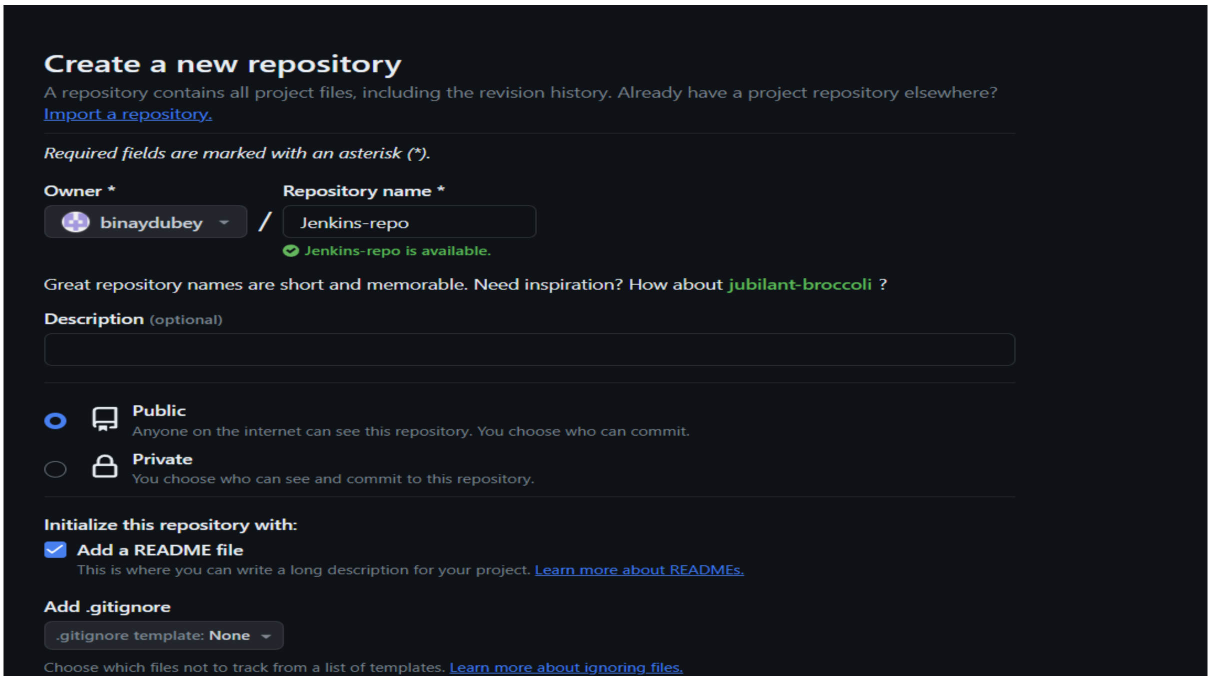
⑤ create a jenkins job to start the container using this image. (link all jobs.)



Aim — to create a CI/CD pipeline using github/ AWS/ jenkins/ docker.

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

Creating new repo :



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Required fields are marked with an asterisk (*).

Owner * binaydubey / Repository name * Jenkins-repo

Jenkins-repo is available.

Great repository names are short and memorable. Need inspiration? How about jubilant-broccoli ?

Description (optional)

[✓ Public] Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

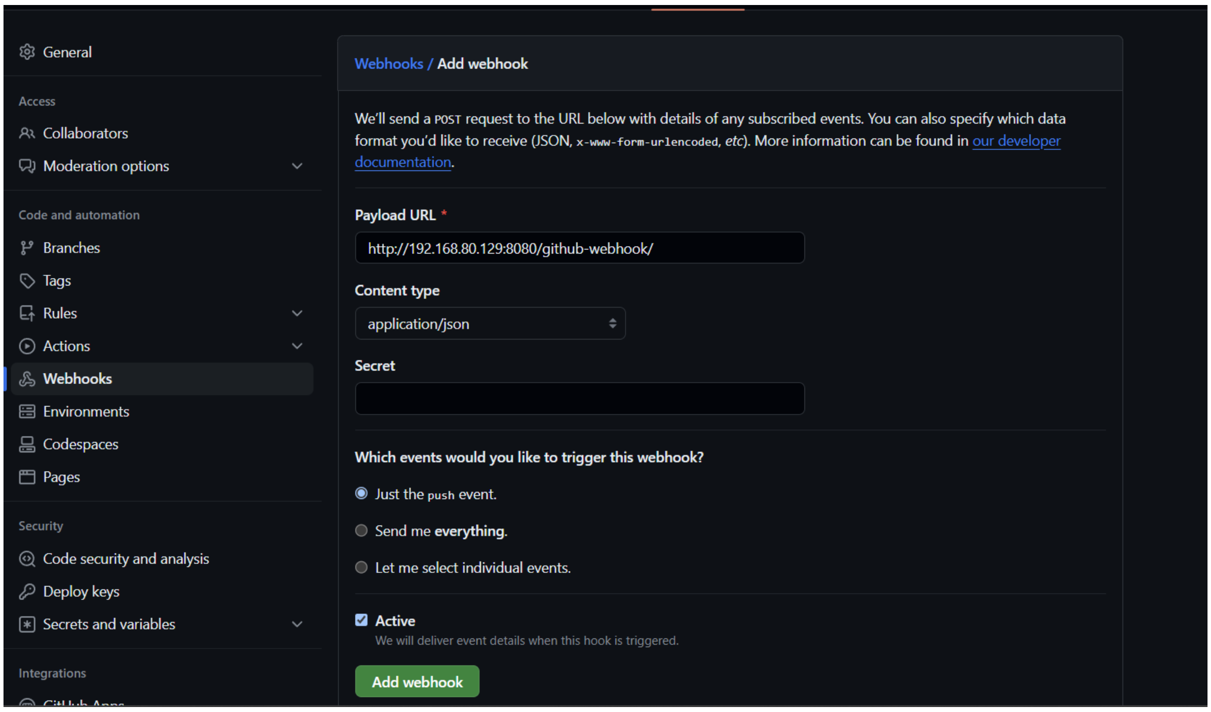
[✓] Add a README file
This is where you can write a long description for your project. Learn more about READMEs.

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. Learn more about ignoring files.

Go to Setting and add webhook:



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, *etc*). More information can be found in our developer documentation.

Payload URL *
http://192.168.80.129:8080/github-webhook/

Content type
application/json

Secret

Which events would you like to trigger this webhook?

[✓ Just the push event.]

Send me **everything**.

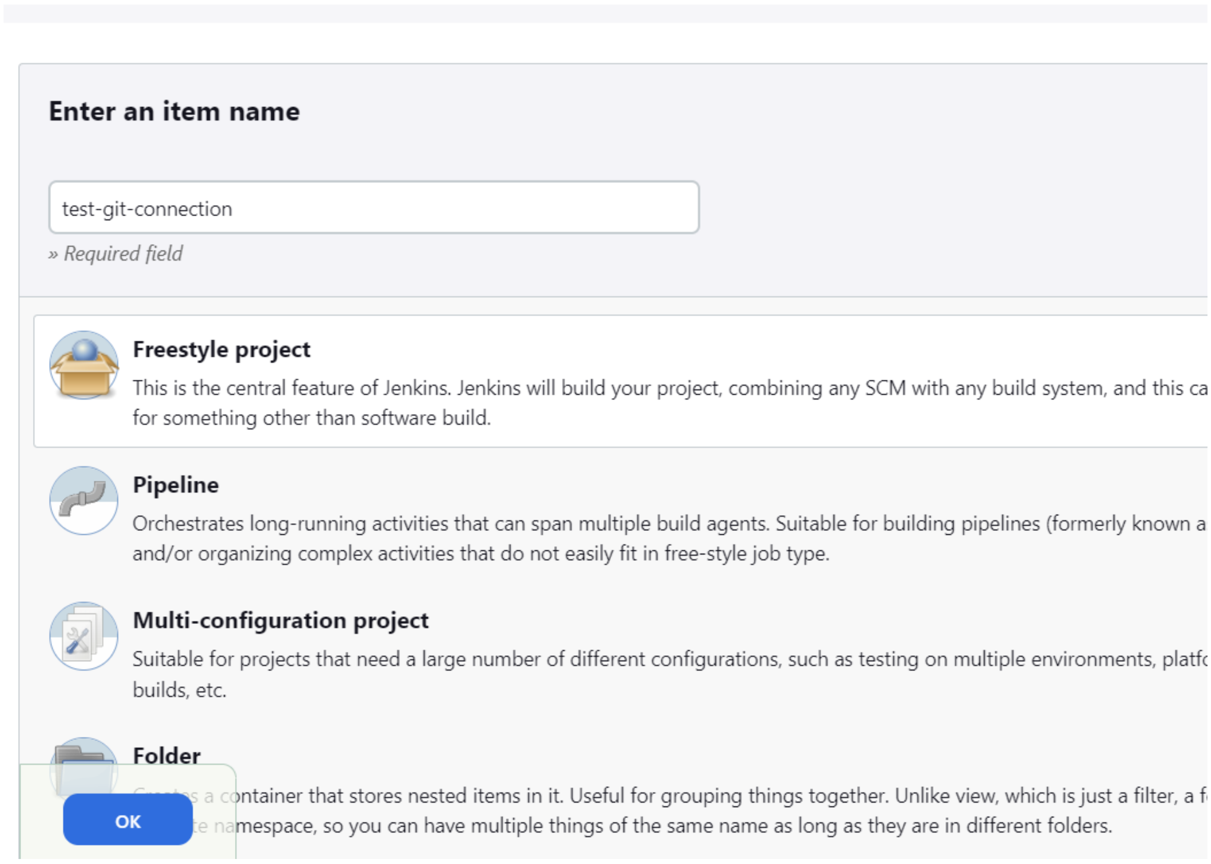
Let me select individual events.

[✓] Active
We will deliver event details when this hook is triggered.

Add webhook

Install git on jenkin:-

Create a freestyle project



Enter an item name

test-git-connection

» *Required field*

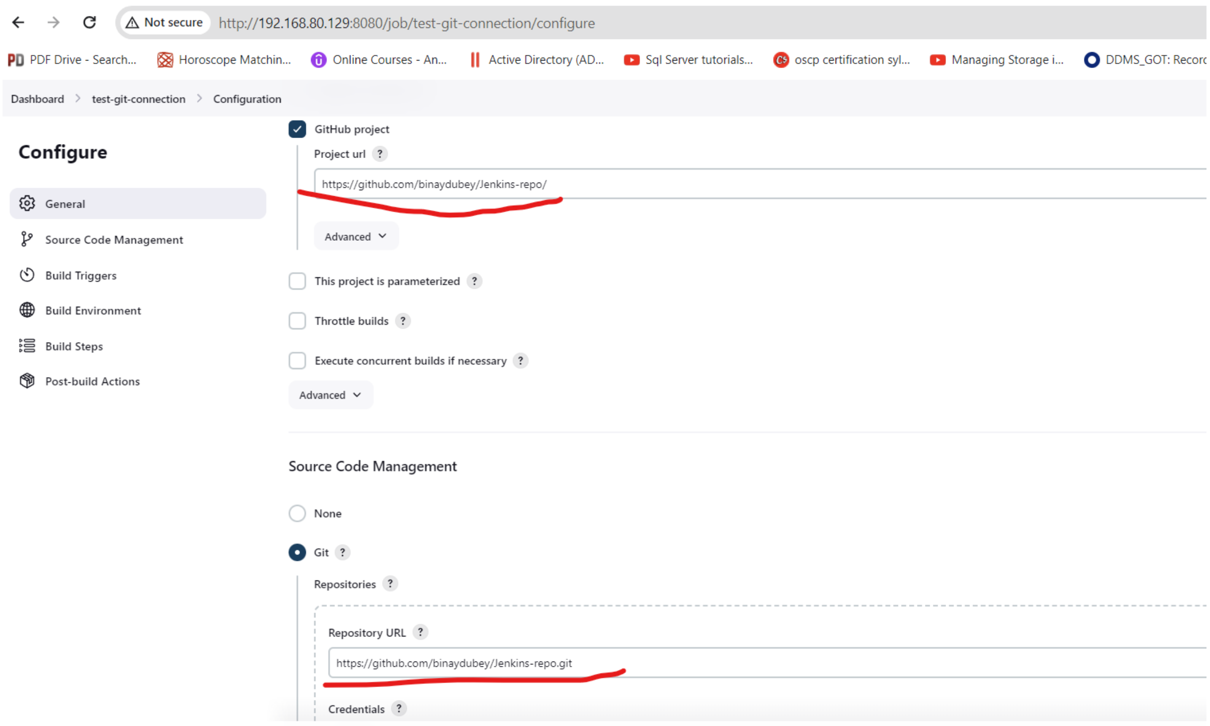
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this ca
for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known a
and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platfo
builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a f
te namespace, so you can have multiple things of the same name as long as they are in different folders.

Add your repo :-


http://192.168.80.129:8080/job/test-git-connection/configure

Dashboard > test-git-connection > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

[✓] GitHub project

Project url: https://github.com/binaydubey/Jenkins-repo/

[ ] This project is parameterized

[ ] Throttle builds

[ ] Execute concurrent builds if necessary

Source Code Management

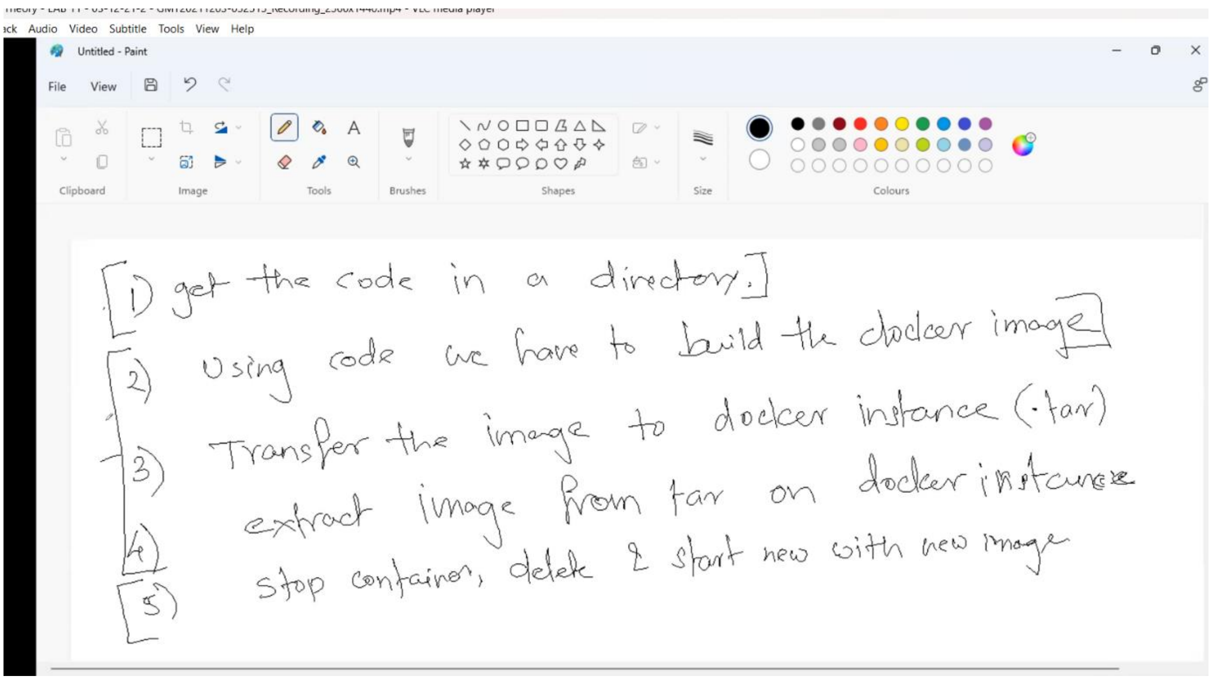
( ) None

(●) Git

Repositories

Repository URL: https://github.com/binaydubey/Jenkins-repo.git

Credentials


[1) get the code in a directory.]

2) Using code we have to build the docker image

3) Transfer the image to docker instance (.tar)

4) extract image from tar on docker instance

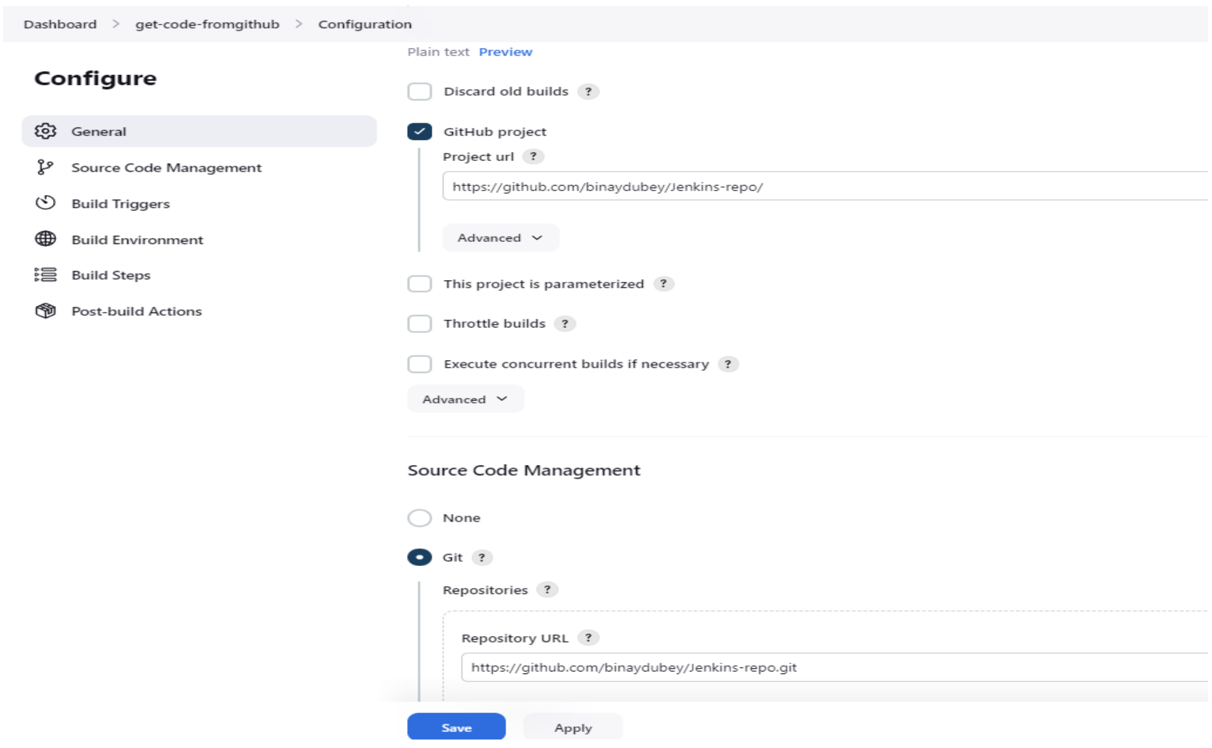
5) stop container, delete & start new with new image

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



Dashboard > get-code-fromgithub > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Plain text Preview

☐ Discard old builds

✓ GitHub project

Project url: https://github.com/binaydubey/Jenkins-repo/

Advanced

☐ This project is parameterized

☐ Throttle builds

☐ Execute concurrent builds if necessary

Advanced

Source Code Management

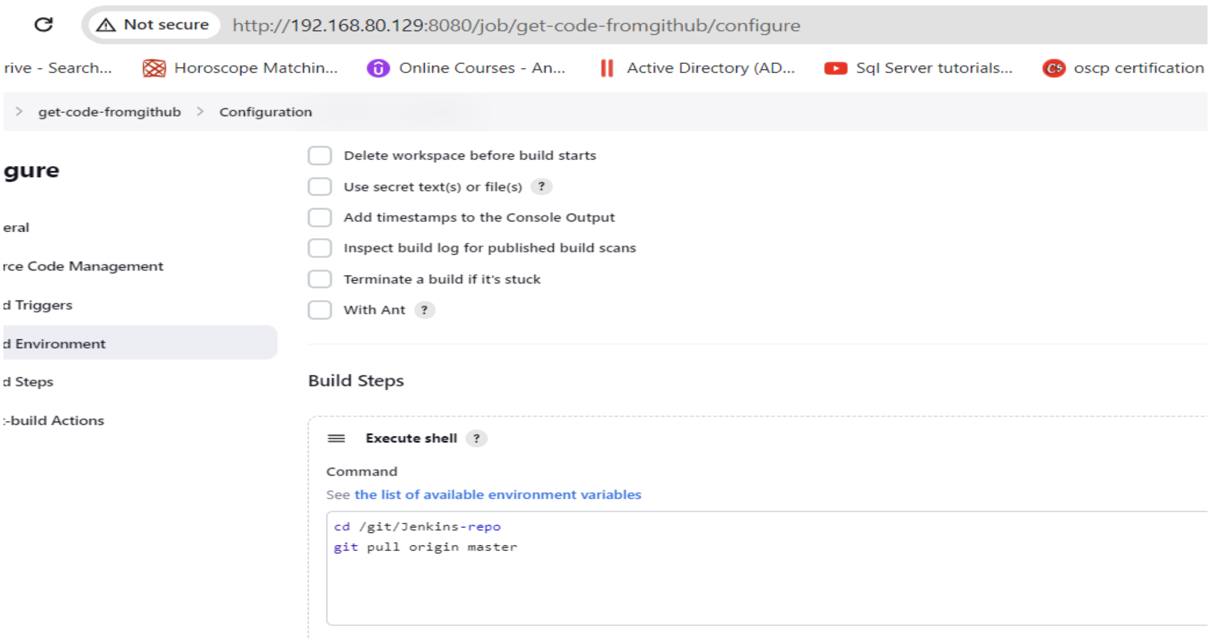
○ None

● Git

Repositories

Repository URL: https://github.com/binaydubey/Jenkins-repo.git

Save Apply


Not secure
http://192.168.80.129:8080/job/get-code-fromgithub/configure
rive - Search...
Horoscope Matchin...
Online Courses - An...
Active Directory (AD...
Sql Server tutorials...
oscp certification
get-code-fromgithub
Configuration
gure
eral
rce Code Management
d Triggers
d Environment
d Steps
:-build Actions
Delete workspace before build starts
Use secret text(s) or file(s)
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Execute shell
Command
See the list of available environment variables

```
cd /git/Jenkins-repo
git pull origin master
```

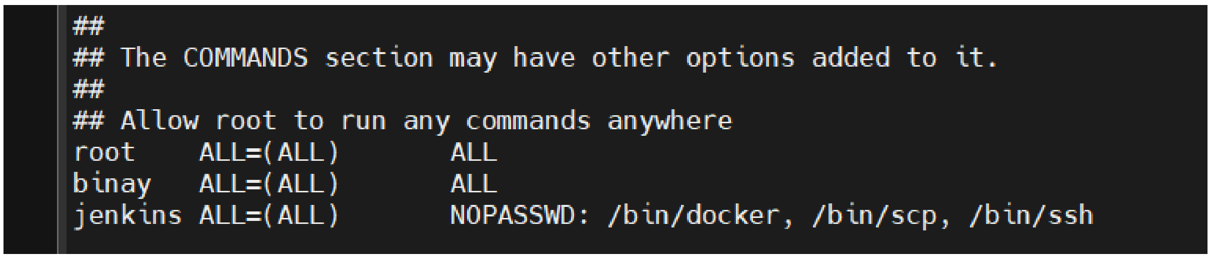
2: we have to build docker image

root@localhost Jenkins-repo]# cat Dockerfile

FROM nginx

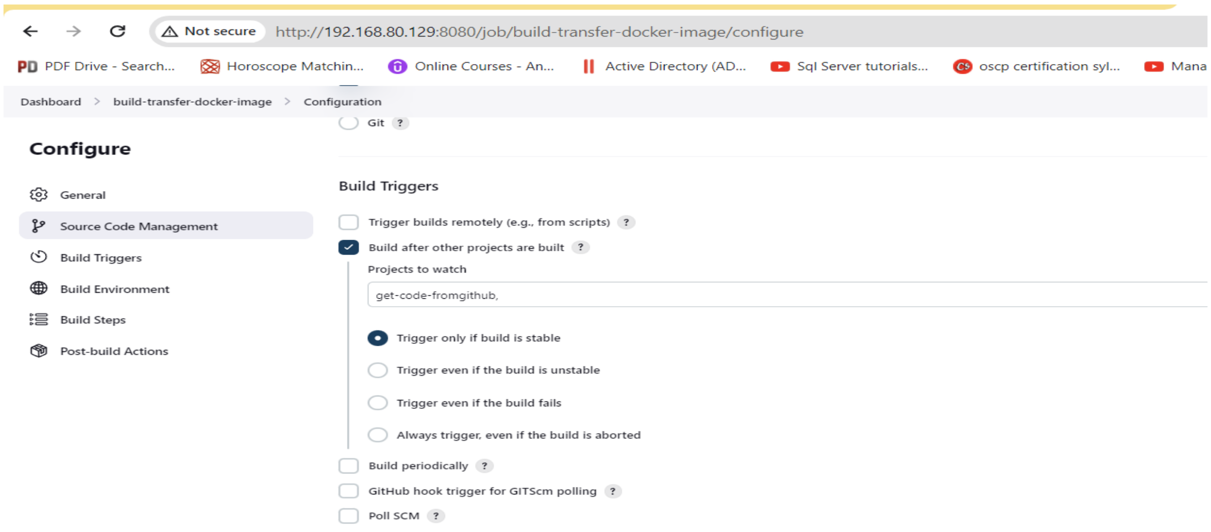
COPY index.html /usr/share/nginx/html

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



```
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
binay   ALL=(ALL)       ALL
jenkins ALL=(ALL)       NOPASSWD: /bin/docker, /bin/scp, /bin/ssh
```

Creating new job to create and transfer image



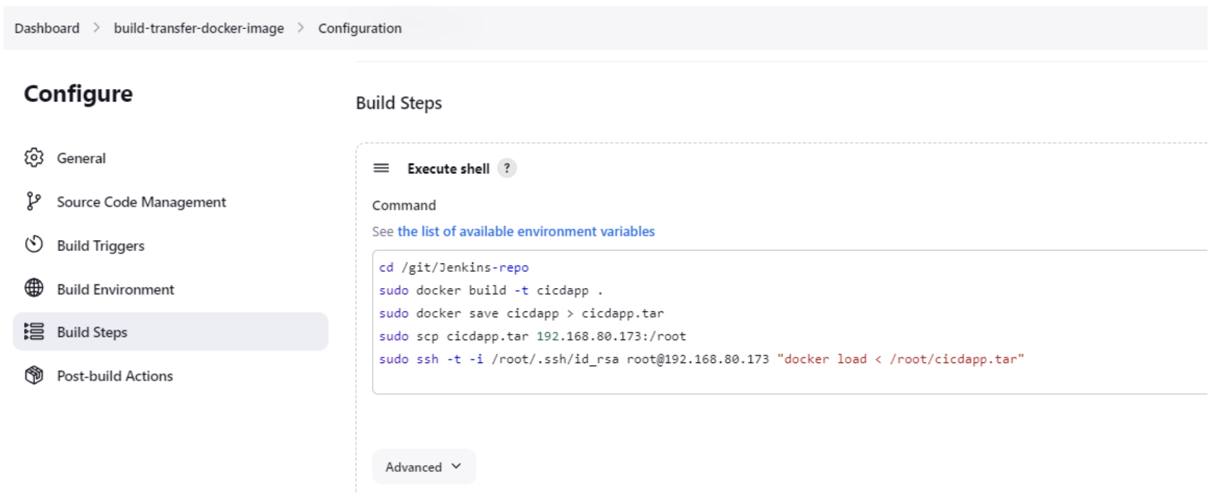
Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

Projects to watch: get-code-fromgithub,

- [● Trigger only if build is stable]
- Trigger even if the build is unstable
- Trigger even if the build fails
- Always trigger, even if the build is aborted

- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM



Build Steps

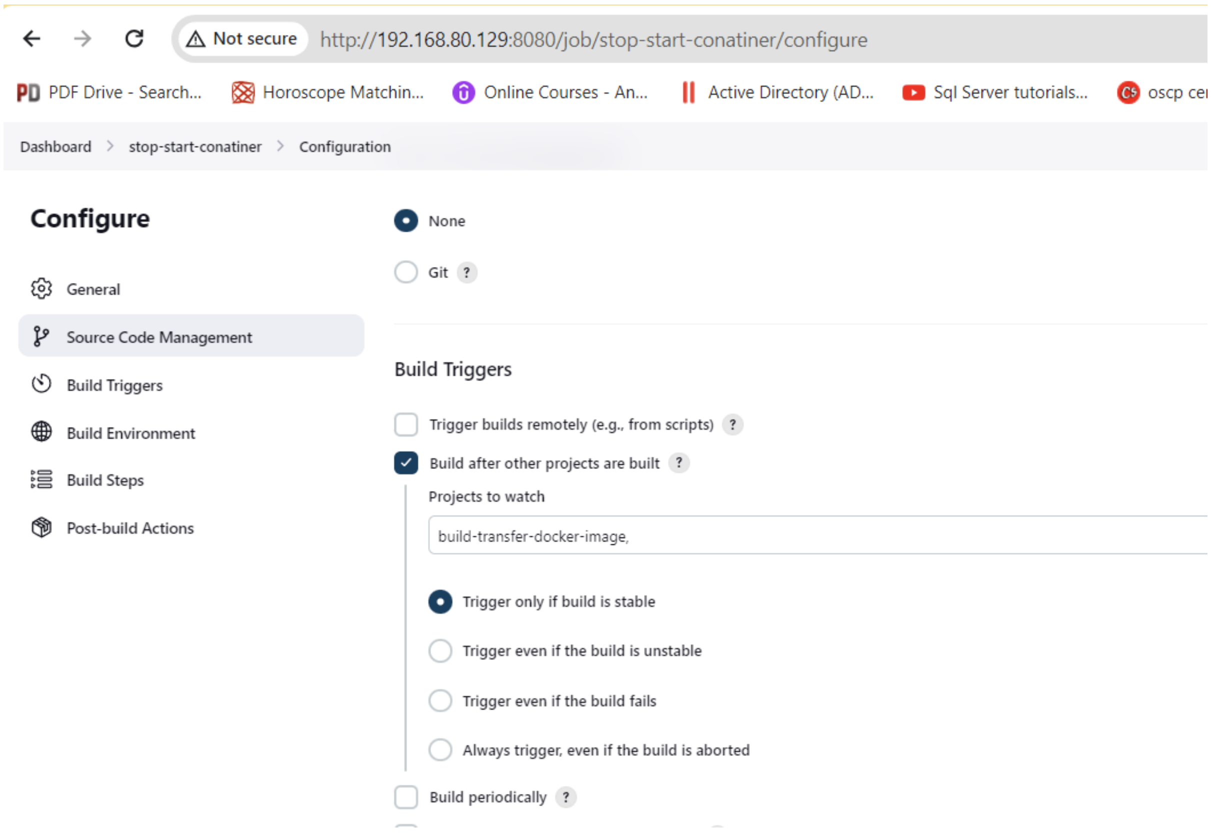
Execute shell

Command

See the list of available environment variables

```
cd /git/Jenkins-repo
sudo docker build -t cicdapp .
sudo docker save cicdapp > cicdapp.tar
sudo scp cicdapp.tar 192.168.80.173:/root
sudo ssh -t -i /root/.ssh/id_rsa root@192.168.80.173 "docker load < /root/cicdapp.tar"
```

Create 3rd job to start and stop container


http://192.168.80.129:8080/job/stop-start-conatiner/configure

Dashboard > stop-start-conatiner > Configuration

Configure

Source Code Management: [✓ None] Git

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [✓] Build after other projects are built

Projects to watch: build-transfer-docker-image,

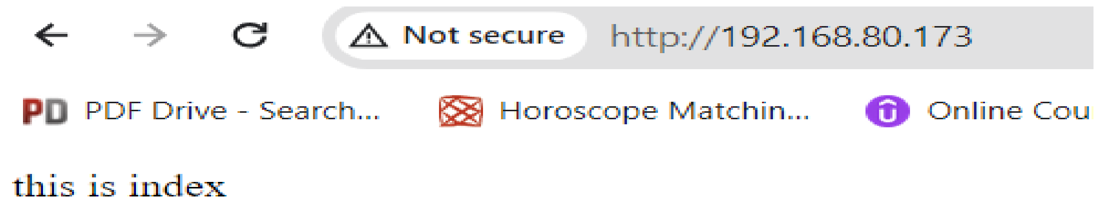
[✓ Trigger only if build is stable] Trigger even if the build is unstable / Trigger even if the build fails / Always trigger, even if the build is aborted

- [ ] Build periodically

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
```

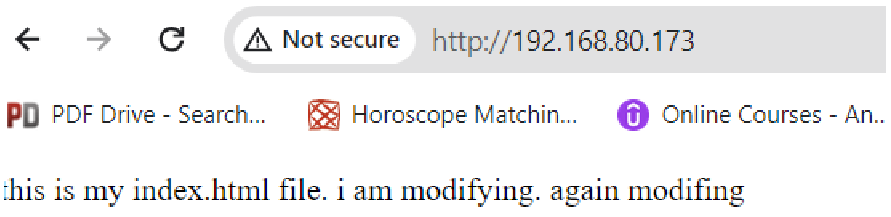
success


Not secure http://192.168.80.173

PDF Drive - Search... Horoscope Matchin... Online Cou

this is index

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


Not secure http://192.168.80.173

PDF Drive - Search... Horoscope Matchin... Online Courses - An..

this is my index.html file. i am modifying. again modifing