Project : Application Deployment Using CI/CD Tools

Problem Statement:

Company ABC want to implement DevOps Lifecycle in their company for the application.

Following are the specifications of the lifecycle:

1. Install the necessary software on the machines using a configuration management tool
2. Git workflow has to be implemented
3. CodeBuild should automatically be triggered once a commit is made to master branch or develop branch.
a. If a commit is made to master branch, test and push to prod
b. If a commit is made to develop branch, just test the product, do not push to prod
4. The code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to GitHub. Use the following pre-built container for your application: hshar/webapp
The code should reside in '/var/www/html'
5. The above tasks should be defined in a Jenkins Pipeline with the following jobs:
a. Job1 : build
b. Job2 : test
c. Job3 : prod

Tools:

Configuration Management Tool: Ansible
Source Code Management Tool: Git
Containerization Tool: Docker
CI/CD Tool: Jenkins

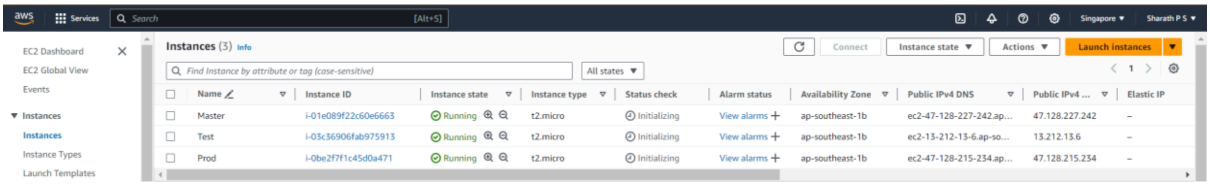
Solution:

1. Launch three EC2 machines and install the required services and tools
   Master: Ansible, Docker, Java, Jenkins
   Test: Docker, Java

Prod: Docker, Java

2. Connecting Master and Slaves
3. Configuring Master and Salves using ansible
4. Launch jenkins in master and connect the slaves to the master.
5. Create the dockerfile and index.html file in the github repo
6. Create the jobs and build as per the specifications
7. Install plugin and add the jobs for the pipeline view

1. EC2 instances using AMI as ubuntu



Instances (3) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
□	Master	i-01e089f22c60e6663	Running	t2.micro	Initializing	View alarms +	ap-southeast-1b	ec2-47-128-227-242.ap...	47.128.227.242	–
□	Test	i-03c36906fab975913	Running	t2.micro	Initializing	View alarms +	ap-southeast-1b	ec2-13-212-13-6.ap-so...	13.212.13.6	–
□	Prod	i-0be2f7f1c45d0a471	Running	t2.micro	Initializing	View alarms +	ap-southeast-1b	ec2-47-128-215-234.ap...	47.128.215.234	–

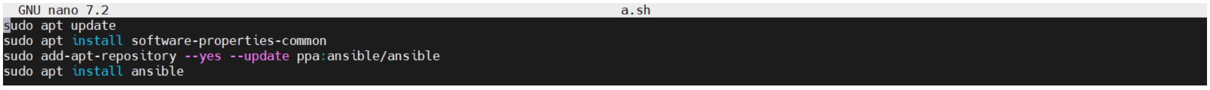
2. Connected to three EC2 instance and install required tools and services

Master:

Connecting to EC2 using the public IP. Then processing the below linux commands to install the services.

- sudo apt-get update
- sudo nano a.sh (Installation commands of ansible)

$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible



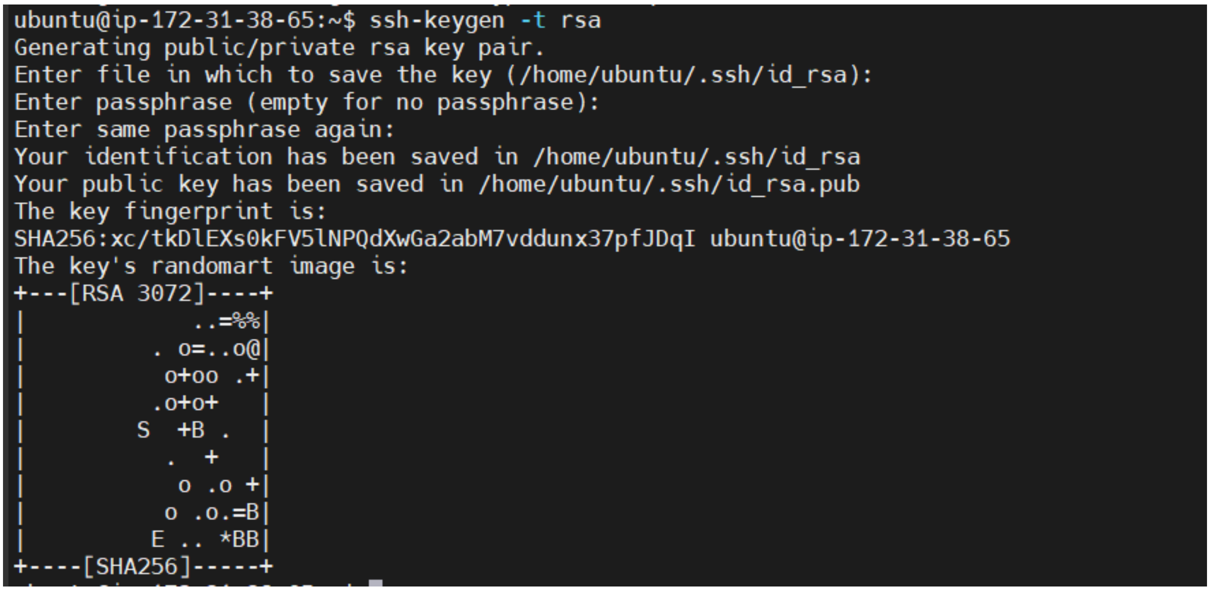
```
  GNU nano 7.2                                   a.sh
sudo apt update
sudo apt install software-properties-common
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

- bash a.sh

3. ssh connection is made to connect between master and slaves:

on Master:

- ssh-keygen -t rsa



```
ubuntu@ip-172-31-38-65:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:xc/tkDlEXs0kFV5lNPQdXwGa2abM7vddunx37pfJDqI ubuntu@ip-172-31-38-65
The key's randomart image is:
+---[RSA 3072]----+
|            ..=%%|
|         . o=..o@|
|          o+oo .+|
|         .o+o+   |
|        S  +B .  |
|          .  +   |
|           o .o +|
|          o .o.=B|
|         E .. *BB|
+----[SHA256]-----+
```

Copy the public key of master and paste it into the slave

- cd .ssh
- ls
- cat id_rsa.pub

go to slaves

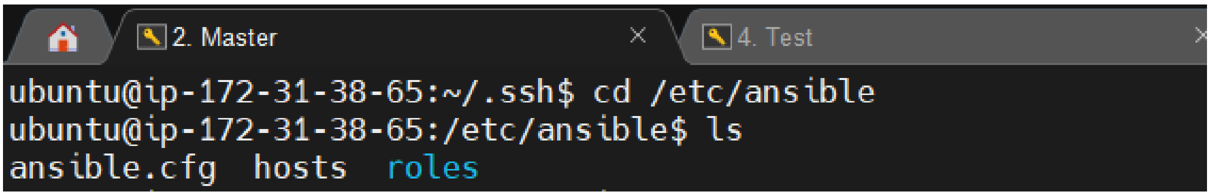
- cd .ssh
- sudo nano authorized_keys

Paste the public key of the master to slaves

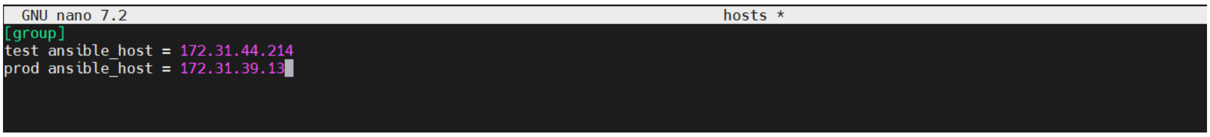
4. Slaves connections to master

Giving the private ip of the slaves to hosts

- cd /etc/ansible/
- sudo nano hosts



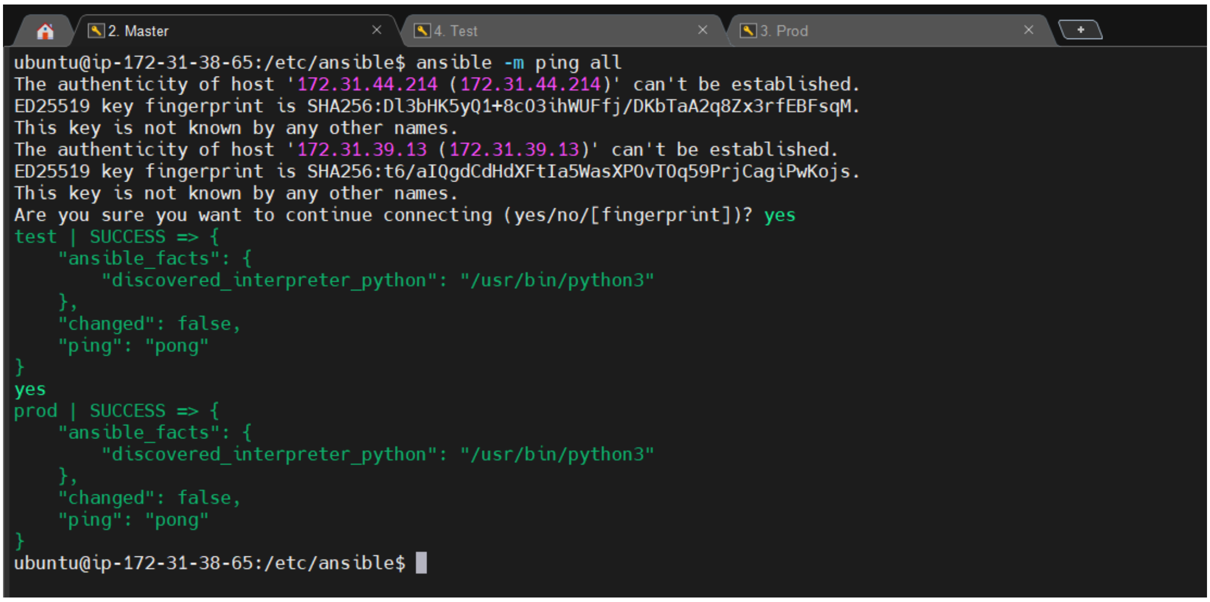
```
ubuntu@ip-172-31-38-65:~/.ssh$ cd /etc/ansible
ubuntu@ip-172-31-38-65:/etc/ansible$ ls
ansible.cfg  hosts  roles
```



```
  GNU nano 7.2                                    hosts *
[group]
test ansible_host = 172.31.44.214
prod ansible_host = 172.31.39.13
```

To establish the connection

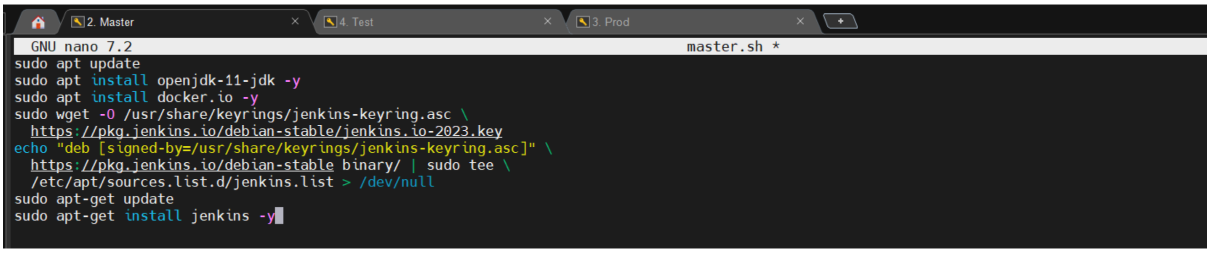
- ansible -m ping all



```
ubuntu@ip-172-31-38-65:/etc/ansible$ ansible -m ping all
The authenticity of host '172.31.44.214 (172.31.44.214)' can't be established.
ED25519 key fingerprint is SHA256:Dl3bHK5yQ1+8c03ihWUFfj/DKbTaA2q8Zx3rfEBFsqM.
This key is not known by any other names.
The authenticity of host '172.31.39.13 (172.31.39.13)' can't be established.
ED25519 key fingerprint is SHA256:t6/aIQgdCdHdXFtIa5WasXPOvT0q59PrjCagiPwKojs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
test | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
prod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-38-65:/etc/ansible$
```

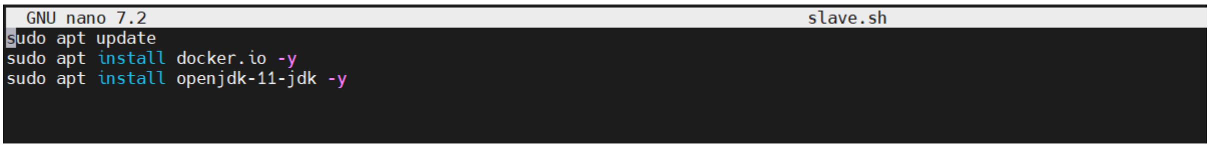
5. Creating a sh file to install the required tools and services on master and slaves

- sudo nano master.sh



```
  GNU nano 7.2                                                  master.sh *
sudo apt update
sudo apt install openjdk-11-jdk -y
sudo apt install docker.io -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
```

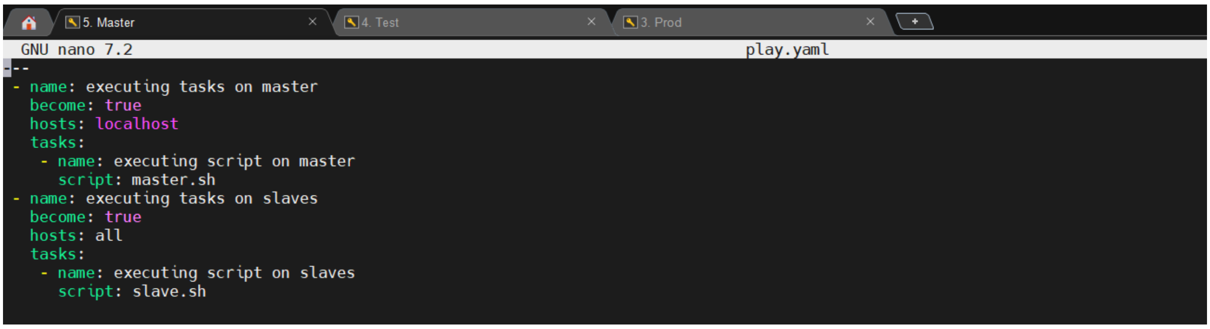
- sudo nano slaves.sh



```
  GNU nano 7.2                                                  slave.sh
sudo apt update
sudo apt install docker.io -y
sudo apt install openjdk-11-jdk -y
```

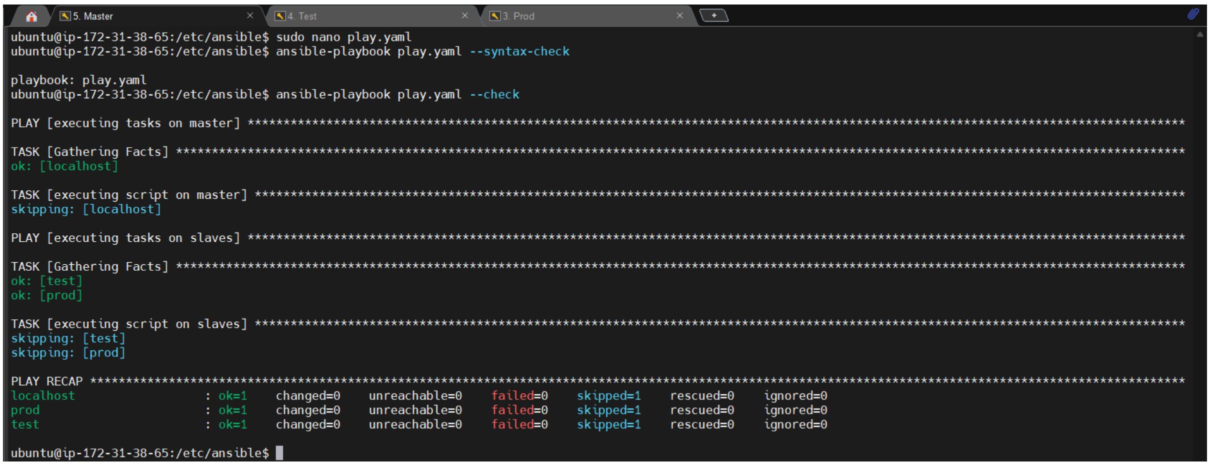
Creating playbook for the configuration

- sudo nano play.yaml



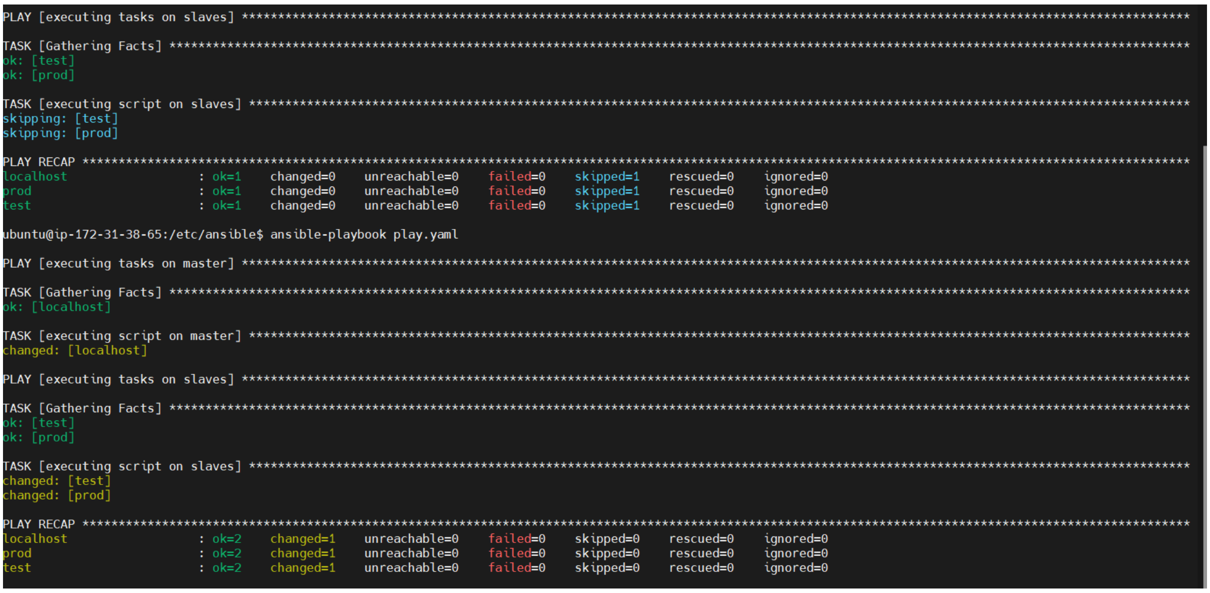
```
 GNU nano 7.2                                                      play.yaml
---
- name: executing tasks on master
  become: true
  hosts: localhost
  tasks:
   - name: executing script on master
     script: master.sh
- name: executing tasks on slaves
  become: true
  hosts: all
  tasks:
   - name: executing script on slaves
     script: slave.sh
```

- ansible-playbook play.yaml –syntax-check
- ansible-playbook play.yaml –check



```
ubuntu@ip-172-31-38-65:/etc/ansible$ sudo nano play.yaml
ubuntu@ip-172-31-38-65:/etc/ansible$ ansible-playbook play.yaml --syntax-check

playbook: play.yaml
ubuntu@ip-172-31-38-65:/etc/ansible$ ansible-playbook play.yaml --check

PLAY [executing tasks on master] ************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [localhost]

TASK [executing script on master] ***********************************************************************************************
skipping: [localhost]

PLAY [executing tasks on slaves] ************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [test]
ok: [prod]

TASK [executing script on slaves] ***********************************************************************************************
skipping: [test]
skipping: [prod]

PLAY RECAP **********************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
prod                       : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
test                       : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ubuntu@ip-172-31-38-65:/etc/ansible$
```

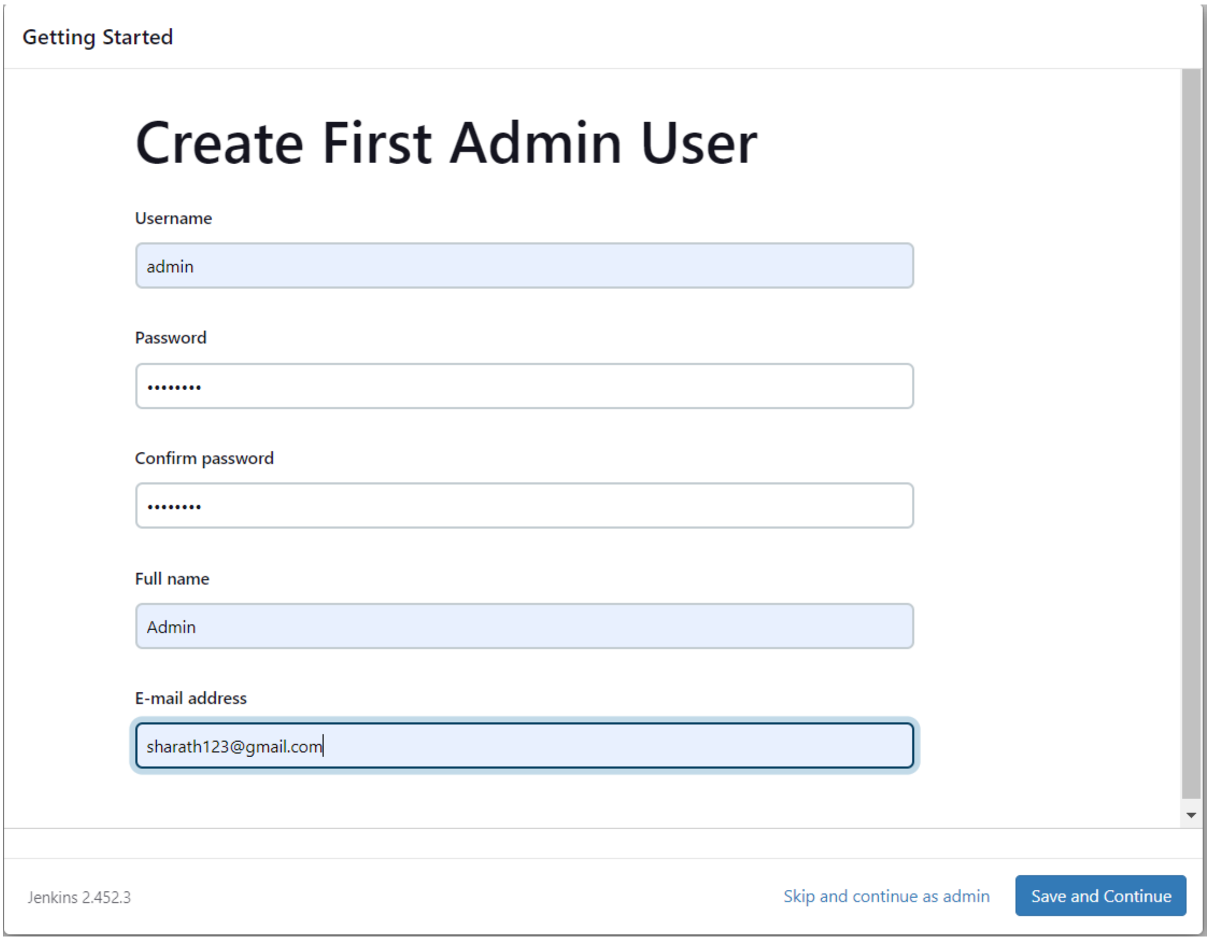
- ansible-playbook play.yaml



```
PLAY [executing tasks on slaves] ************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [test]
ok: [prod]

TASK [executing script on slaves] ***********************************************************************************************
skipping: [test]
skipping: [prod]

PLAY RECAP **********************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
prod                       : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
test                       : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0

ubuntu@ip-172-31-38-65:/etc/ansible$ ansible-playbook play.yaml

PLAY [executing tasks on master] ************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [localhost]

TASK [executing script on master] ***********************************************************************************************
changed: [localhost]

PLAY [executing tasks on slaves] ************************************************************************************************

TASK [Gathering Facts] **********************************************************************************************************
ok: [test]
ok: [prod]

TASK [executing script on slaves] ***********************************************************************************************
changed: [test]
changed: [prod]

PLAY RECAP **********************************************************************************************************************
localhost                  : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
prod                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
test                       : ok=2    changed=1    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

6. Accessing Jenkins that installed on Master:

http://Public_IP:8080/



Getting Started

Create First Admin User

Username: admin

Password: ••••••••

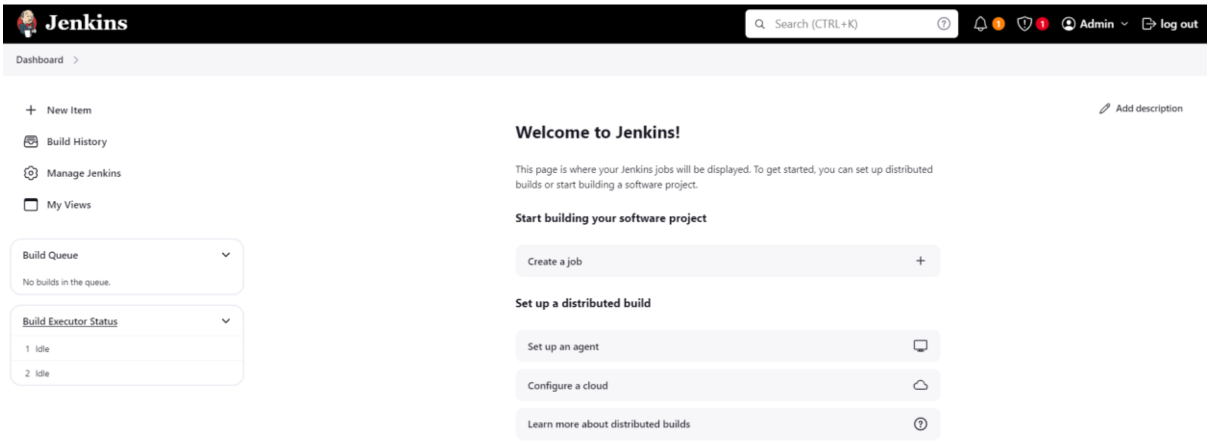
Confirm password: ••••••••

Full name: Admin

E-mail address: sharath123@gmail.com

Jenkins 2.452.3

username: admin
password: admin123



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Set up a distributed build

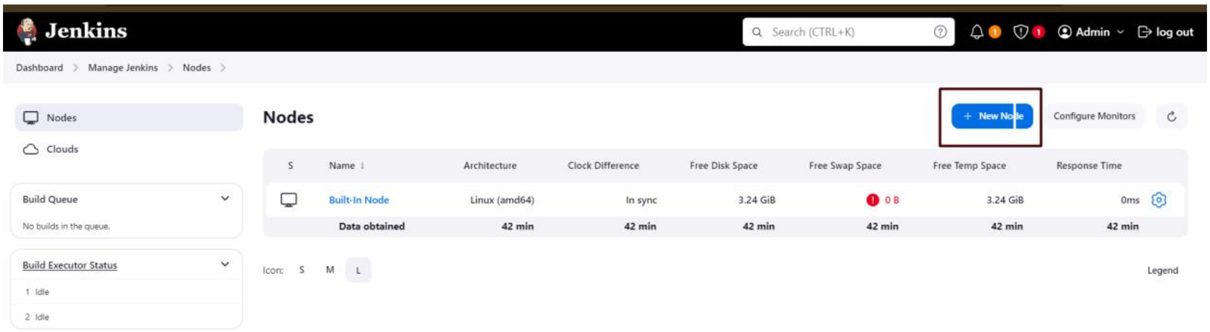
Build Queue

No builds in the queue.

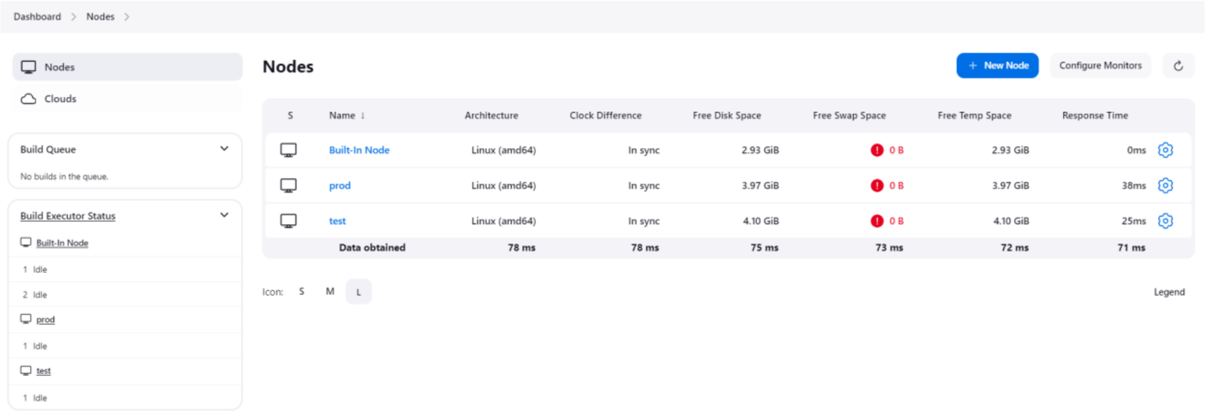
Build Executor Status

1 Idle
2 Idle

7. Connecting the Test and Prod nodes to Jenkins



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.24 GiB	0 B	3.24 GiB	0ms
	Data obtained	42 min	42 min	42 min	42 min	42 min	42 min



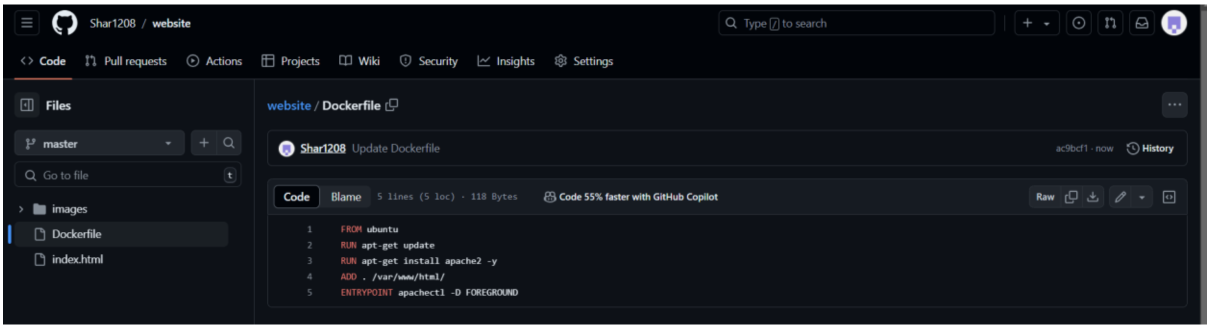
S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	2.93 GiB	0 B	2.93 GiB	0ms
	prod	Linux (amd64)	In sync	3.97 GiB	0 B	3.97 GiB	38ms
	test	Linux (amd64)	In sync	4.10 GiB	0 B	4.10 GiB	25ms
	Data obtained	78 ms	78 ms	75 ms	73 ms	72 ms	71 ms

8. Creating Jobs

Slave1 – Test
Slave2 – Prod

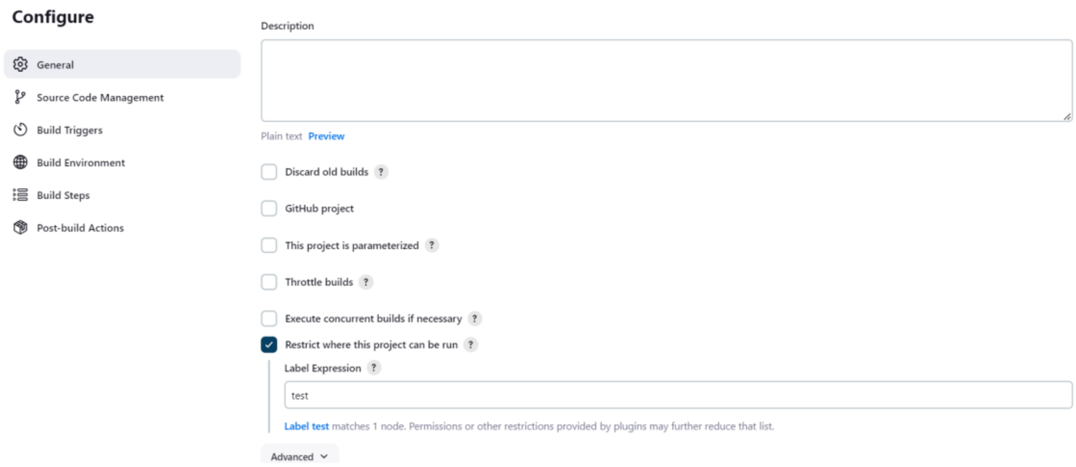
job1 – connected to master (test) – Git from master branch
job2 – connected to slave1 (test) – Git from develop branch
job3 – connected to slave2 (prod) – Git from master branch

Dockerfile in the Github



```
FROM ubuntu
RUN apt-get update
RUN apt-get install apache2 -y
ADD . /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND
```

Configuration of job1



Description

Plain text Preview

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

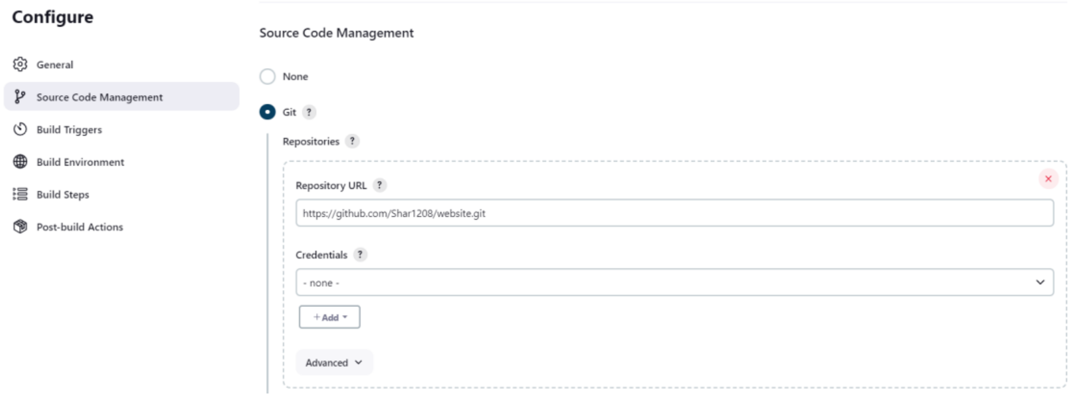
☐ Execute concurrent builds if necessary ?

☑ Restrict where this project can be run ?

Label Expression ?

test

Label test matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.



Source Code Management

○ None

● Git ?

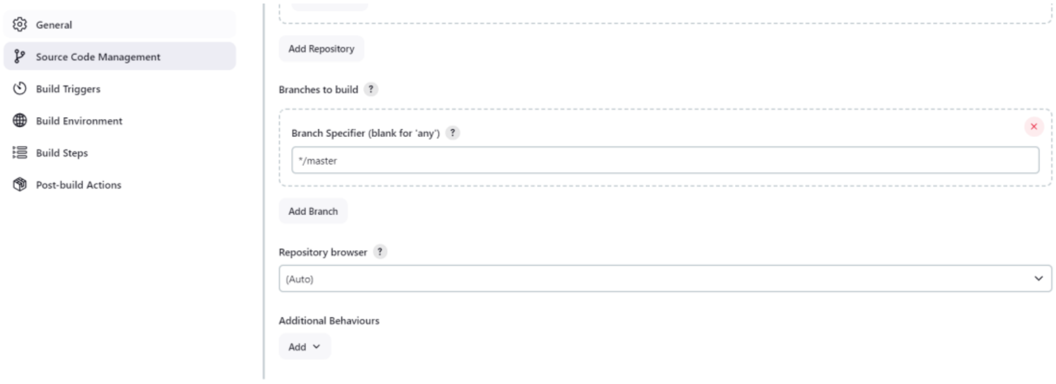
Repositories ?

Repository URL ?

https://github.com/Shar1208/website.git

Credentials ?

- none -



Branches to build ?

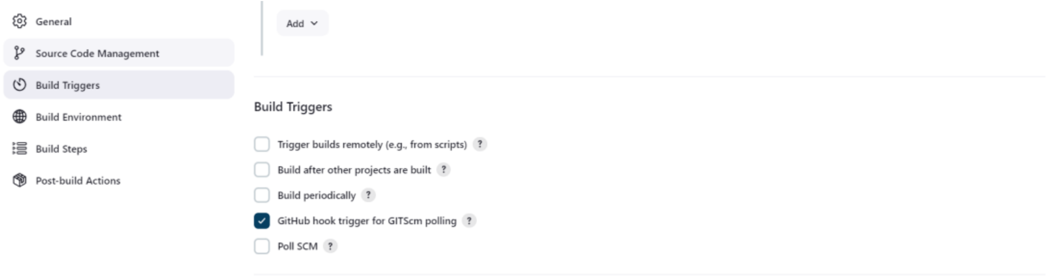
Branch Specifier (blank for 'any') ?

*/master

Repository browser ?

(Auto)

Additional Behaviours



Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

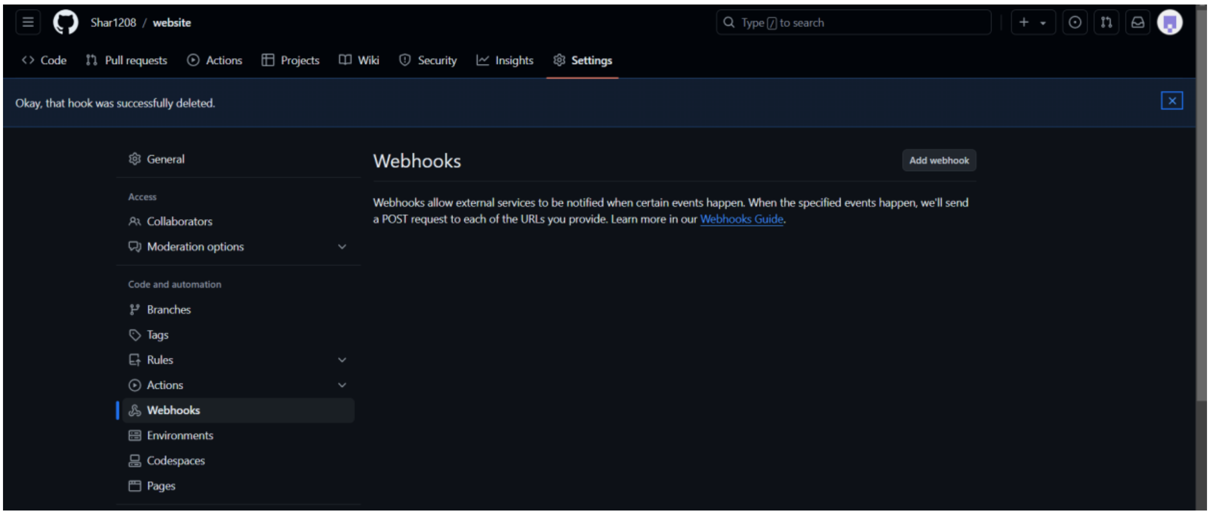
☑ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Creating github webhook:

Add this url:

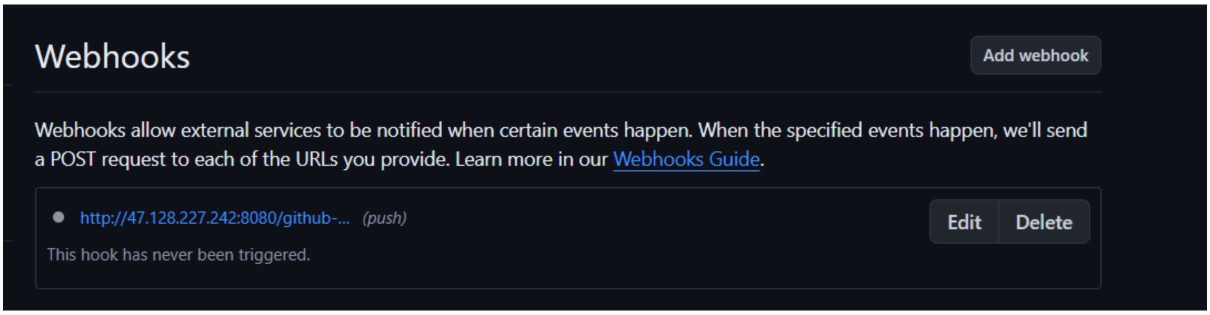
http://47.128.227.242:8080/github-webhook/



Okay, that hook was successfully deleted.

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

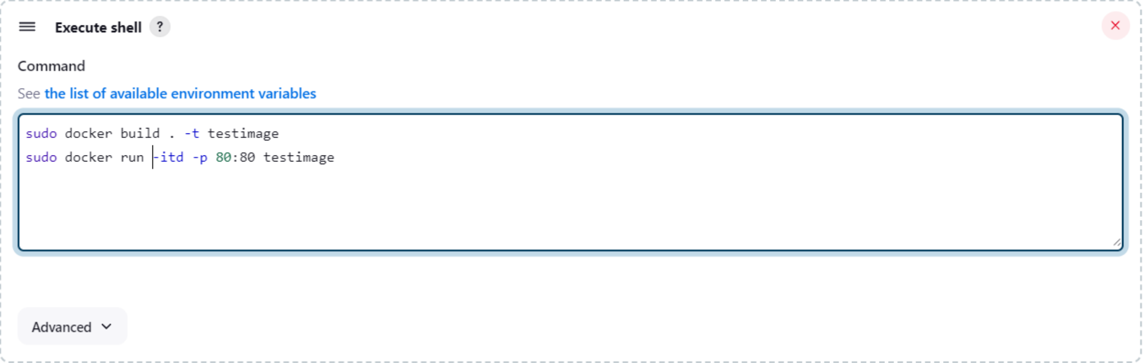


Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

● http://47.128.227.242:8080/github-... (push)

This hook has never been triggered.

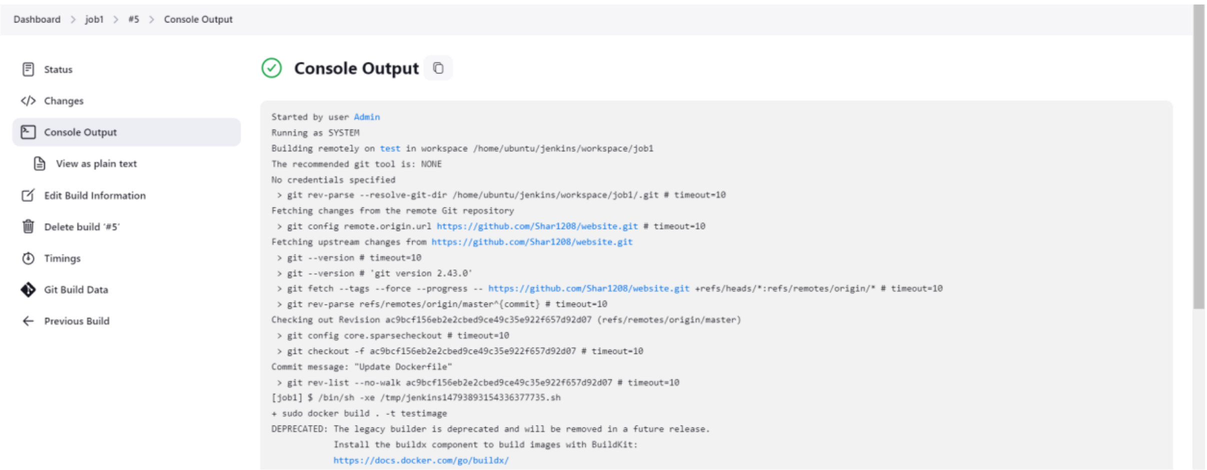


Command

See the list of available environment variables

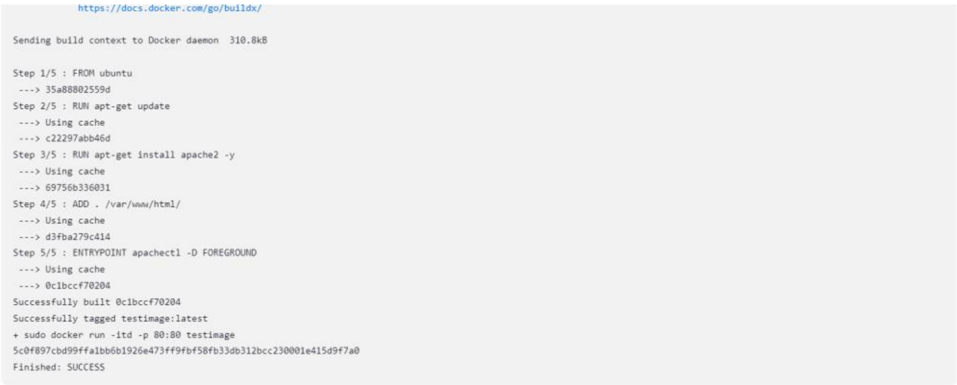
```
sudo docker build . -t testimage
sudo docker run -itd -p 80:80 testimage
```

After build the job



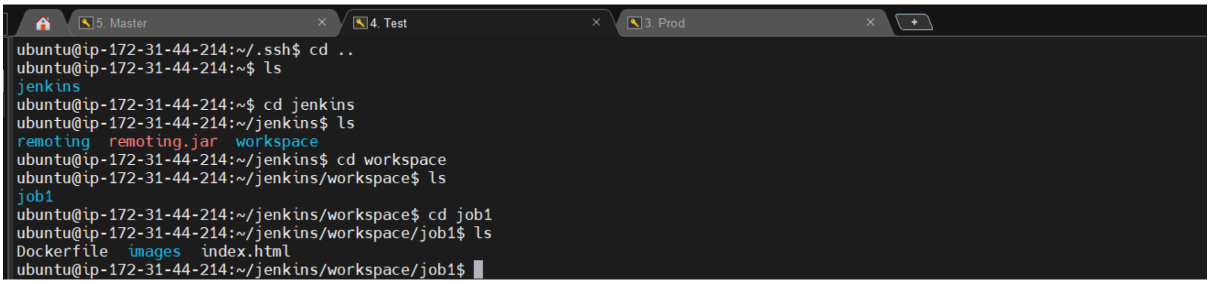
Console Output

```
Started by user Admin
Running as SYSTEM
Building remotely on test in workspace /home/ubuntu/jenkins/workspace/job1
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/job1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Shar1208/website.git # timeout=10
Fetching upstream changes from https://github.com/Shar1208/website.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/Shar1208/website.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision ac9bcf156eb2e2cbed9ce49c35e922f657d92d07 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f ac9bcf156eb2e2cbed9ce49c35e922f657d92d07 # timeout=10
Commit message: "Update Dockerfile"
 > git rev-list --no-walk ac9bcf156eb2e2cbed9ce49c35e922f657d92d07 # timeout=10
[job1] $ /bin/sh -xe /tmp/jenkins14793893154336377735.sh
+ sudo docker build . -t testimage
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/
```

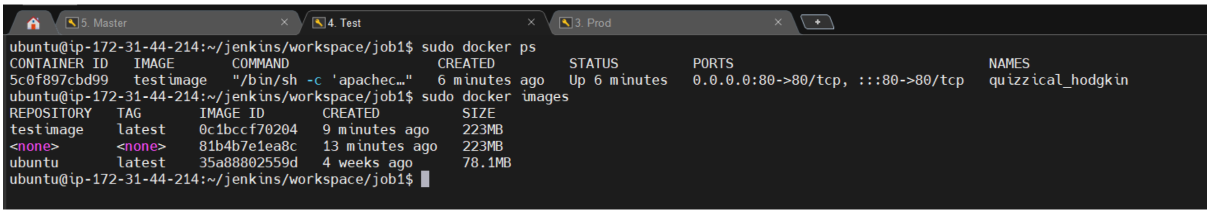


```
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  310.8kB

Step 1/5 : FROM ubuntu
 ---> 35a88802559d
Step 2/5 : RUN apt-get update
 ---> Using cache
 ---> c22297abb46d
Step 3/5 : RUN apt-get install apache2 -y
 ---> Using cache
 ---> 69756b336031
Step 4/5 : ADD . /var/www/html/
 ---> Using cache
 ---> d3fba279c414
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
 ---> Using cache
 ---> 0c1bccf70204
Successfully built 0c1bccf70204
Successfully tagged testimage:latest
+ sudo docker run -itd -p 80:80 testimage
5c0f897cbd99ffa1bb6b1926e473ff9fbf58fb33db312bcc230001e415d9f7a0
Finished: SUCCESS
```

Going to test machine to find the path and the build

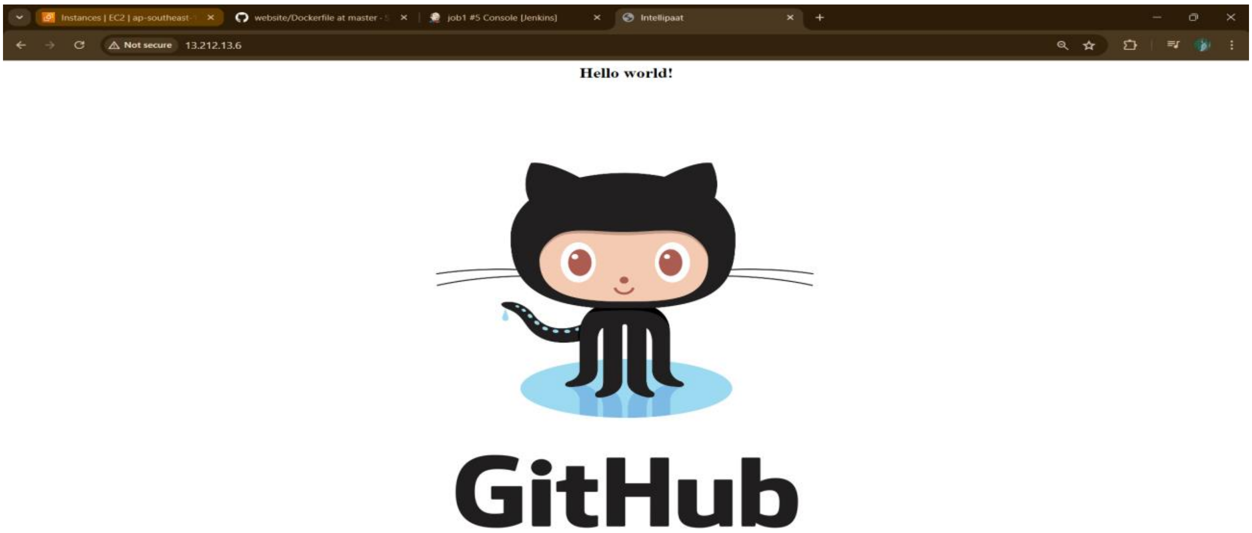


```
ubuntu@ip-172-31-44-214:~/.ssh$ cd ..
ubuntu@ip-172-31-44-214:~$ ls
jenkins
ubuntu@ip-172-31-44-214:~$ cd jenkins
ubuntu@ip-172-31-44-214:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-44-214:~/jenkins$ cd workspace
ubuntu@ip-172-31-44-214:~/jenkins/workspace$ ls
job1
ubuntu@ip-172-31-44-214:~/jenkins/workspace$ cd job1
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$
```



```
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$ sudo docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS                               NAMES
5c0f897cbd99   testimage   "/bin/sh -c 'apachec…"   6 minutes ago   Up 6 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   quizzical_hodgkin
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
testimage    latest    0c1bccf70204   9 minutes ago    223MB
<none>       <none>    81b4b7e1ea8c   13 minutes ago   223MB
ubuntu       latest    35a88802559d   4 weeks ago      78.1MB
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$
```

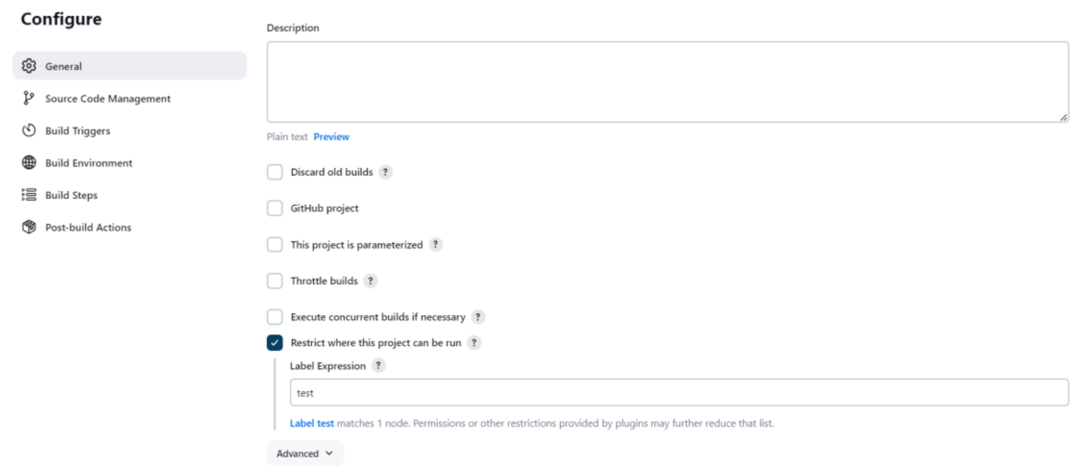
Accessing the port 80:


Instances | EC2 | ap-southeast
website/Dockerfile at master
job1 #5 Console [Jenkins]
Intellipaat
Not secure

13.212.13.6

Hello world!

Configuration of job2



Description

□ Discard old builds

□ GitHub project

□ This project is parameterized

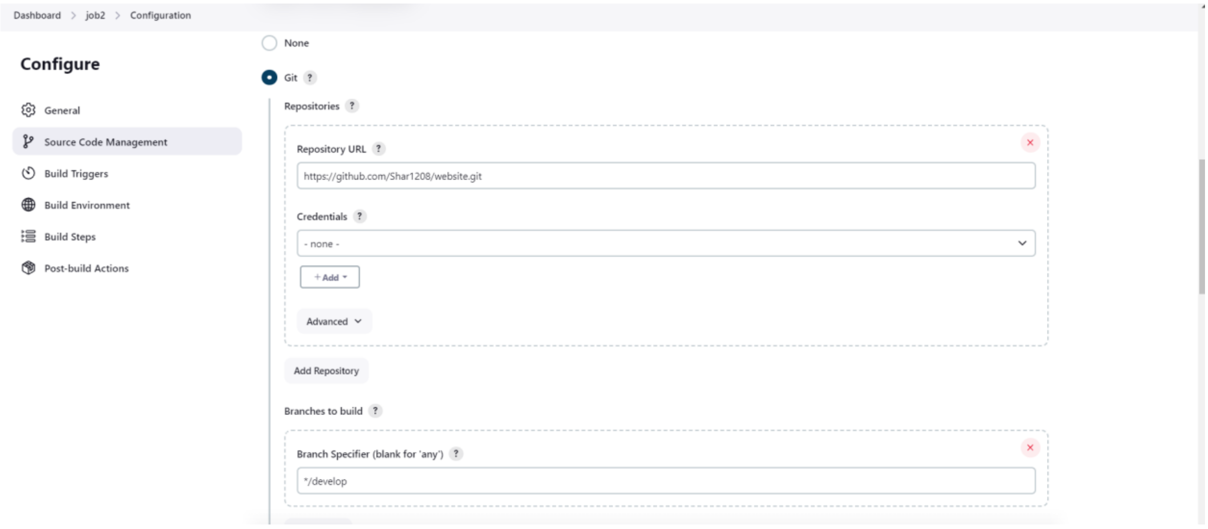
□ Throttle builds

□ Execute concurrent builds if necessary

☑ Restrict where this project can be run

Label Expression: test

Label test matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.



○ None

● Git

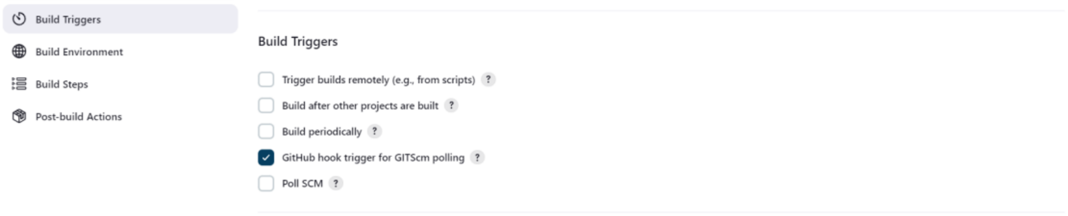
Repositories

Repository URL: https://github.com/Shar1208/website.git

Credentials: - none -

Branches to build

Branch Specifier (blank for 'any'): */develop



Build Triggers

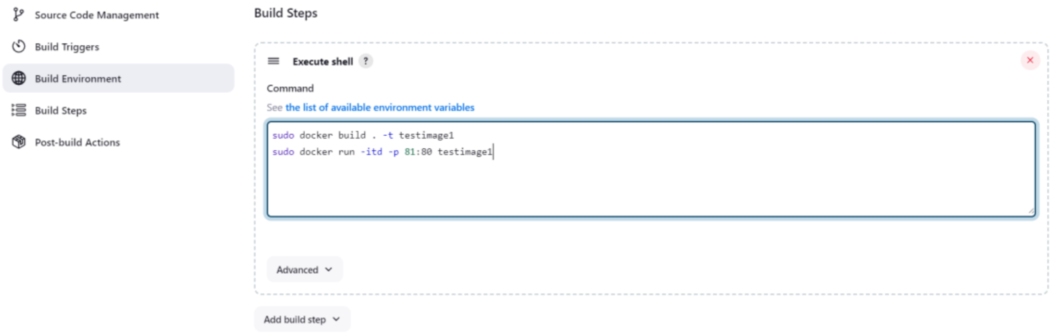
□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?

□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?



Build Steps

Execute shell ?

Command

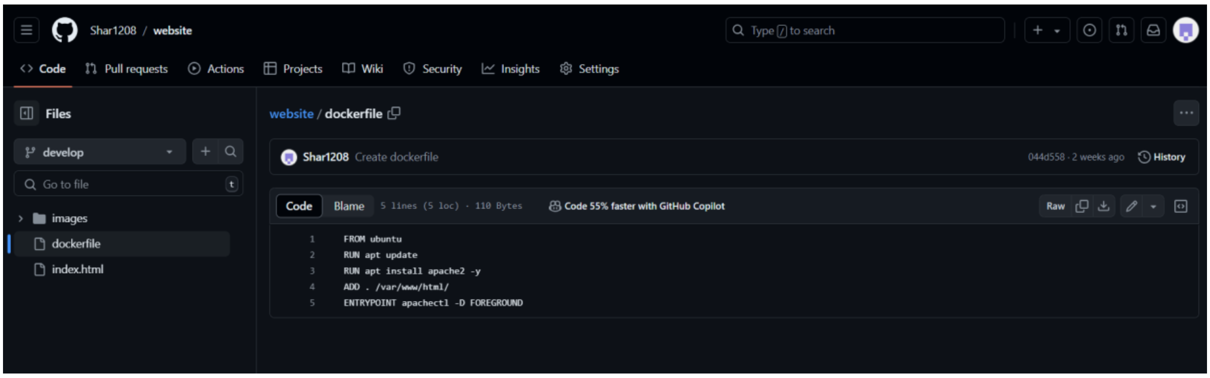
See the list of available environment variables

```
sudo docker build . -t testimage1
sudo docker run -itd -p 81:80 testimage1
```

After build the job2:

From develop branch

On Test machine

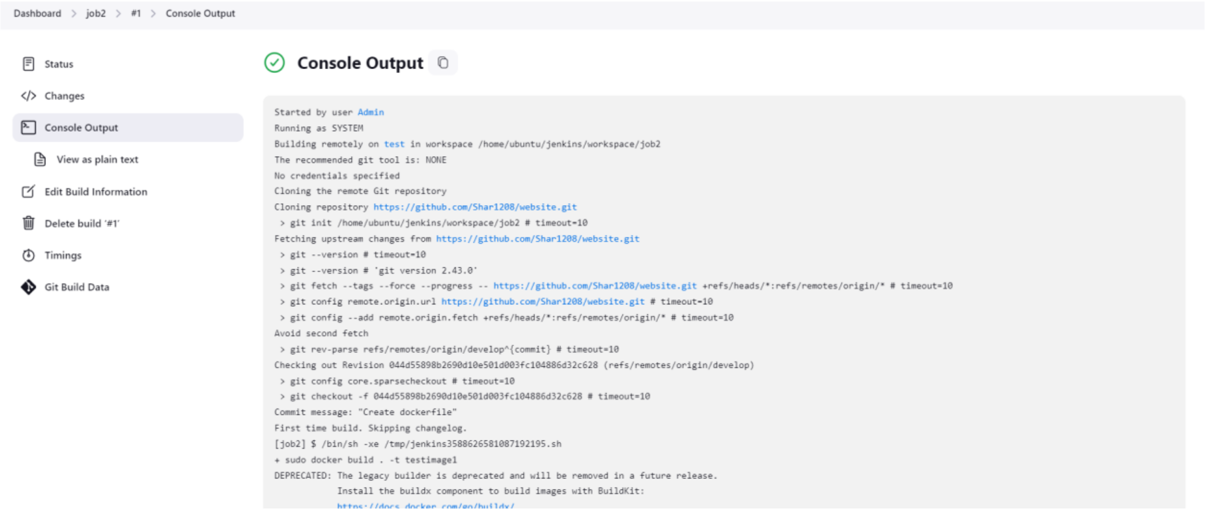


website / dockerfile

Shar1208 Create dockerfile 044d558 · 2 weeks ago History

5 lines (5 loc) · 110 Bytes

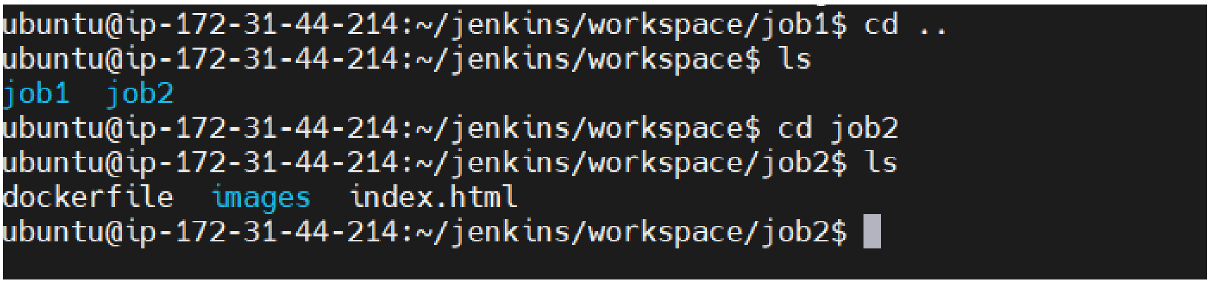
```
FROM ubuntu
RUN apt update
RUN apt install apache2 -y
ADD . /var/www/html/
ENTRYPOINT apachectl -D FOREGROUND
```



Console Output

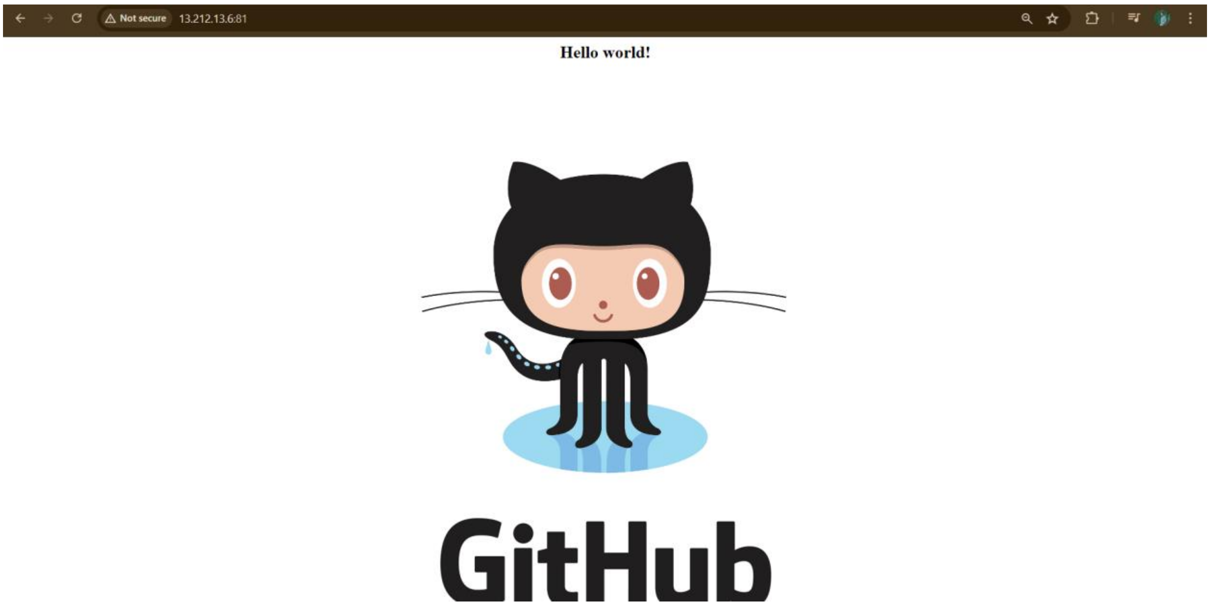
```
Started by user Admin
Running as SYSTEM
Building remotely on test in workspace /home/ubuntu/jenkins/workspace/job2
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Shar1208/website.git
 > git init /home/ubuntu/jenkins/workspace/job2 # timeout=10
Fetching upstream changes from https://github.com/Shar1208/website.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/Shar1208/website.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Shar1208/website.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/develop^{commit} # timeout=10
Checking out Revision 044d55898b2690d10e501d003fc104886d32c628 (refs/remotes/origin/develop)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 044d55898b2690d10e501d003fc104886d32c628 # timeout=10
Commit message: "Create dockerfile"
First time build. Skipping changelog.
[job2] $ /bin/sh -xe /tmp/jenkins3588626581087192195.sh
+ sudo docker build . -t testimage1
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/
```

Go to the test machine:



```
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job1$ cd ..
ubuntu@ip-172-31-44-214:~/jenkins/workspace$ ls
job1  job2
ubuntu@ip-172-31-44-214:~/jenkins/workspace$ cd job2
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job2$ ls
dockerfile  images  index.html
ubuntu@ip-172-31-44-214:~/jenkins/workspace/job2$
```

Accessing the port 81:

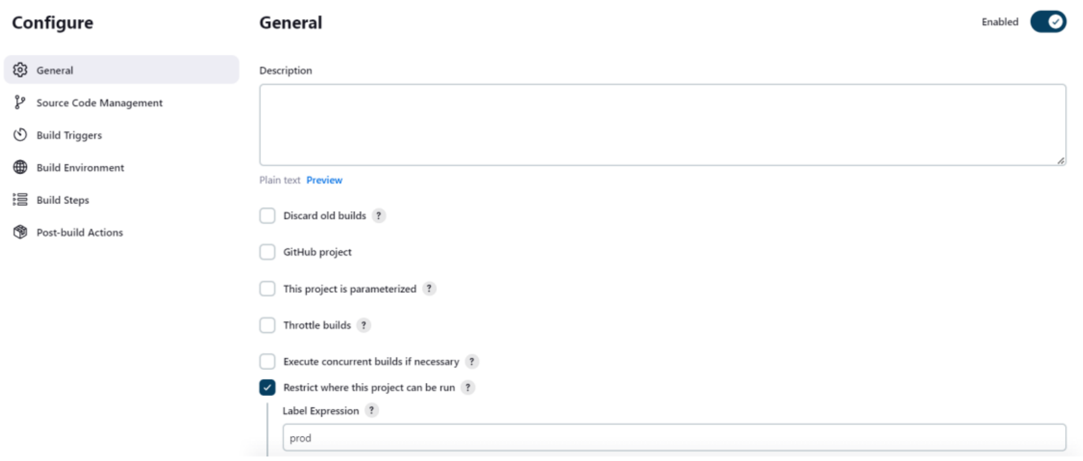


Hello world!

Configuration of job3

From master branch

On prod machine



General

Description

Plain text Preview

□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?

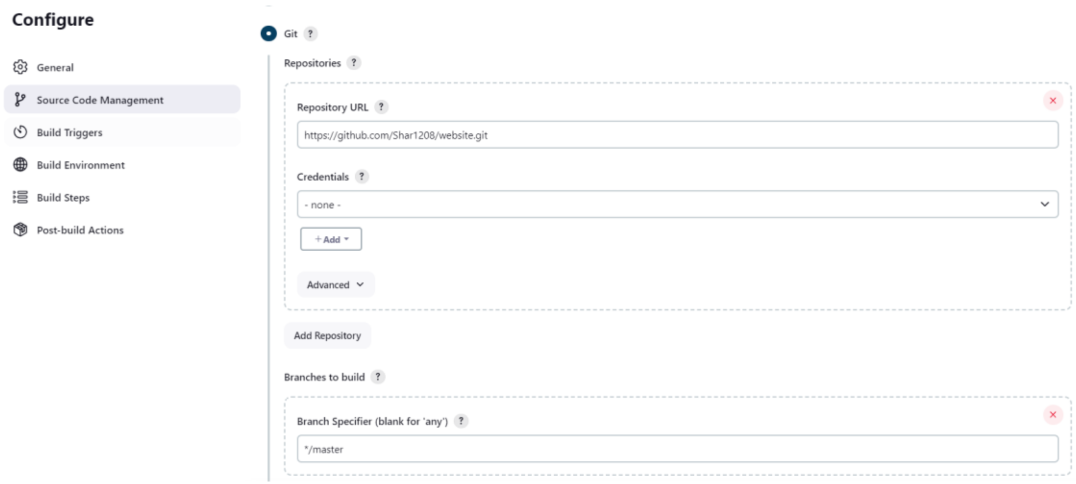
□ Throttle builds ?

□ Execute concurrent builds if necessary ?

☑ Restrict where this project can be run ?

Label Expression ?

prod



● Git ?

Repositories ?

Repository URL ?

https://github.com/Shar1208/website.git

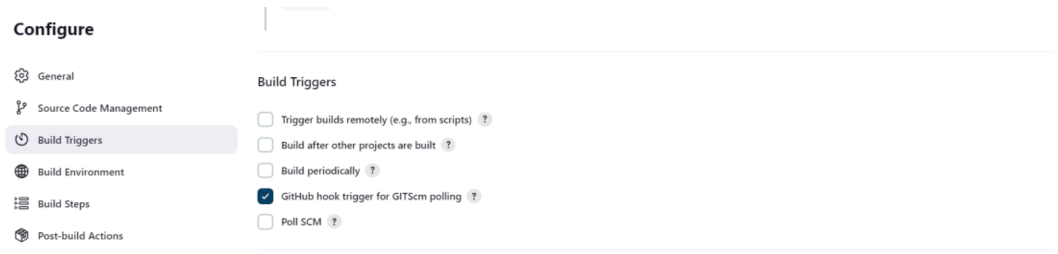
Credentials ?

- none -

Branches to build ?

Branch Specifier (blank for 'any') ?

*/master



Build Triggers

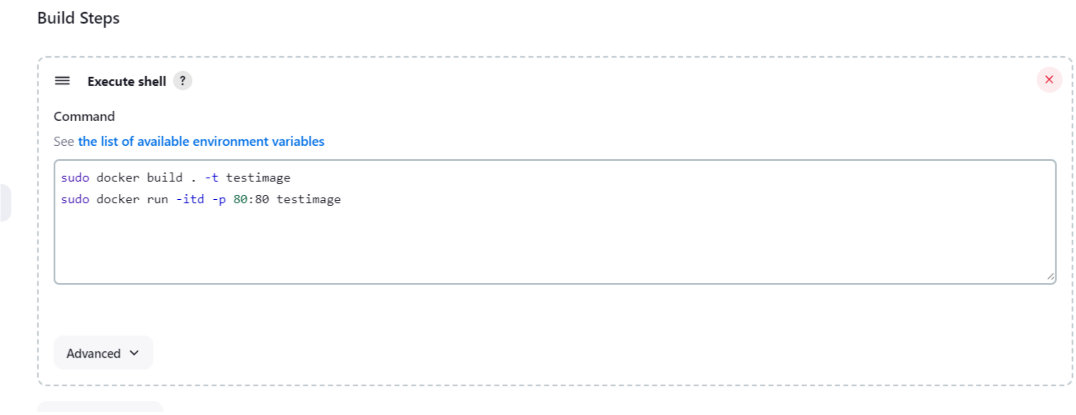
□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?

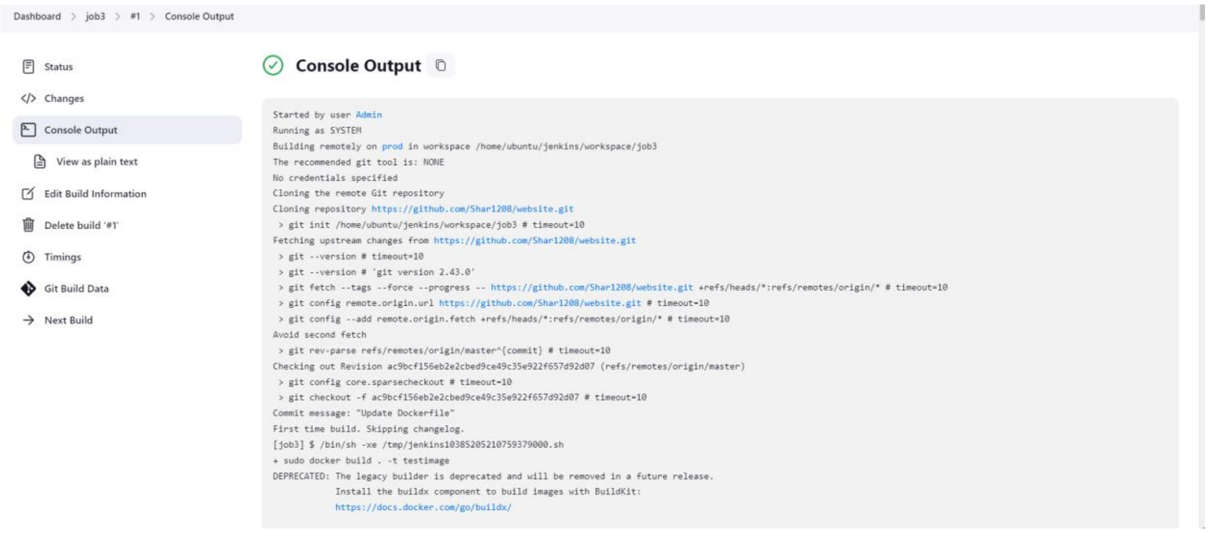
□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

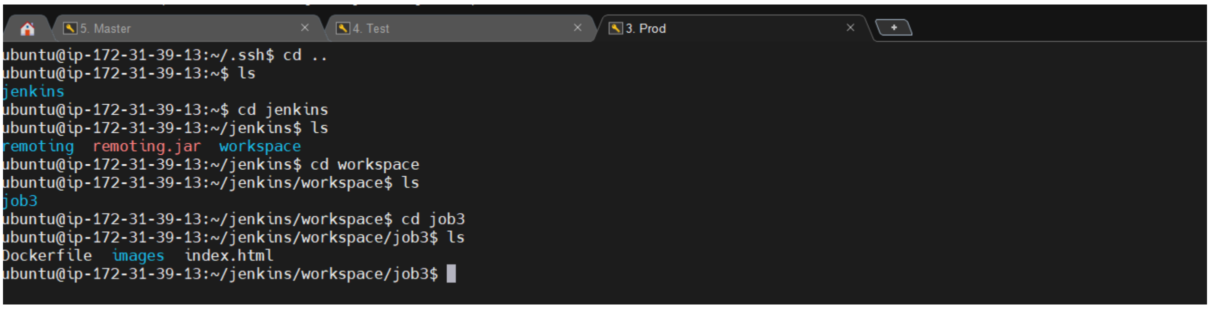


```
sudo docker build . -t testimage
sudo docker run -itd -p 80:80 testimage
```

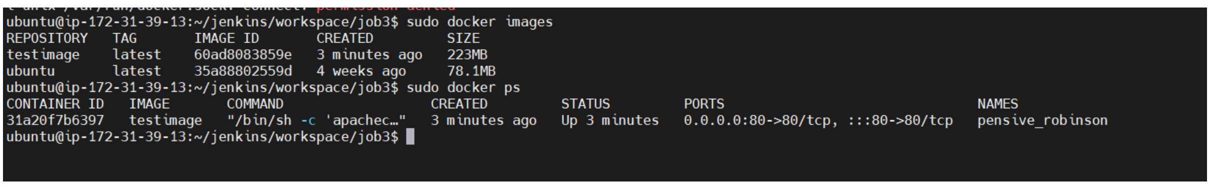


```
Started by user Admin
Running as SYSTEM
Building remotely on prod in workspace /home/ubuntu/jenkins/workspace/job3
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Shar1208/website.git
 > git init /home/ubuntu/jenkins/workspace/job3 # timeout=10
Fetching upstream changes from https://github.com/Shar1208/website.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
 > git fetch --tags --force --progress -- https://github.com/Shar1208/website.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Shar1208/website.git # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision ac9bcf156eb2e2cbed9ce49c35e922f657d92d07 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f ac9bcf156eb2e2cbed9ce49c35e922f657d92d07 # timeout=10
Commit message: "Update Dockerfile"
First time build. Skipping changelog.
[job3] $ /bin/sh -xe /tmp/jenkins10385205210759379000.sh
+ sudo docker build . -t testimage
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/
```

Go to the prod machine:

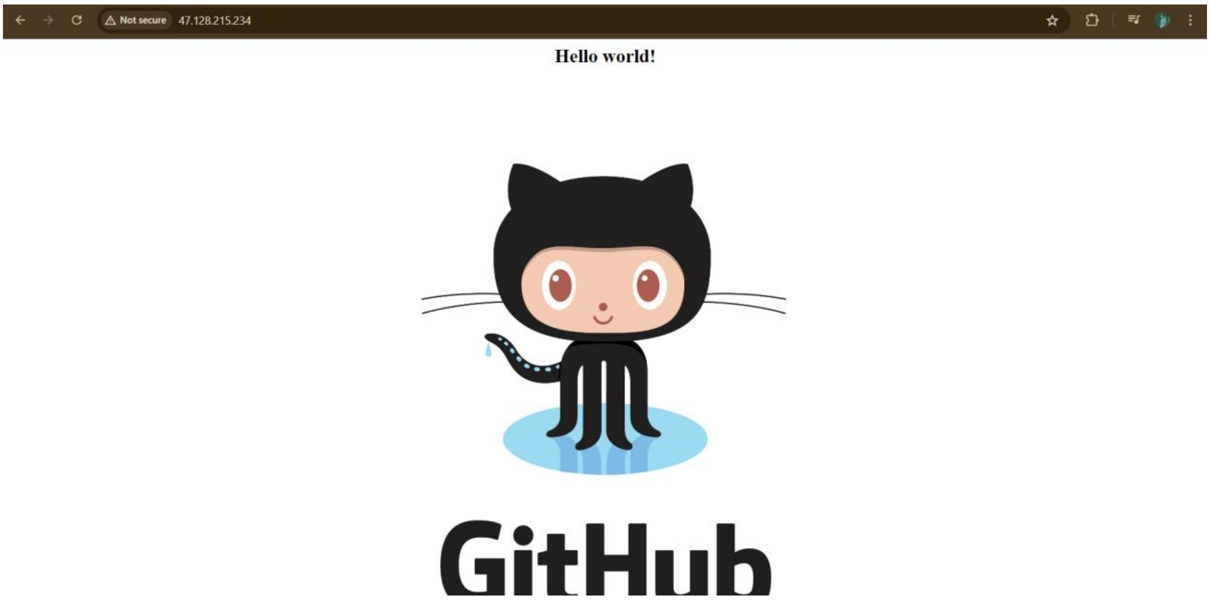


```
ubuntu@ip-172-31-39-13:~/.ssh$ cd ..
ubuntu@ip-172-31-39-13:~$ ls
jenkins
ubuntu@ip-172-31-39-13:~$ cd jenkins
ubuntu@ip-172-31-39-13:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-39-13:~/jenkins$ cd workspace
ubuntu@ip-172-31-39-13:~/jenkins/workspace$ ls
job3
ubuntu@ip-172-31-39-13:~/jenkins/workspace$ cd job3
ubuntu@ip-172-31-39-13:~/jenkins/workspace/job3$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-39-13:~/jenkins/workspace/job3$
```



```
ubuntu@ip-172-31-39-13:~/jenkins/workspace/job3$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
testimage    latest    60ad8083859e   3 minutes ago   223MB
ubuntu       latest    35a88802559d   4 weeks ago     78.1MB
ubuntu@ip-172-31-39-13:~/jenkins/workspace/job3$ sudo docker ps
CONTAINER ID   IMAGE       COMMAND                  CREATED         STATUS         PORTS                               NAMES
31a20f7b6397   testimage   "/bin/sh -c 'apachec…"   3 minutes ago   Up 3 minutes   0.0.0.0:80->80/tcp, :::80->80/tcp   pensive_robinson
ubuntu@ip-172-31-39-13:~/jenkins/workspace/job3$
```

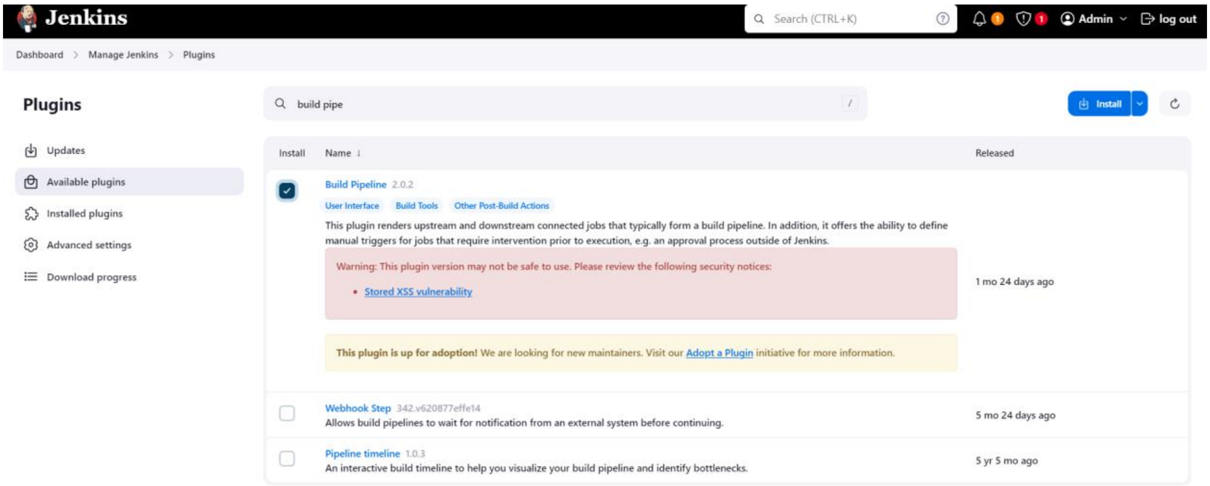
accessing on port 80 of the prod:



Hello world!

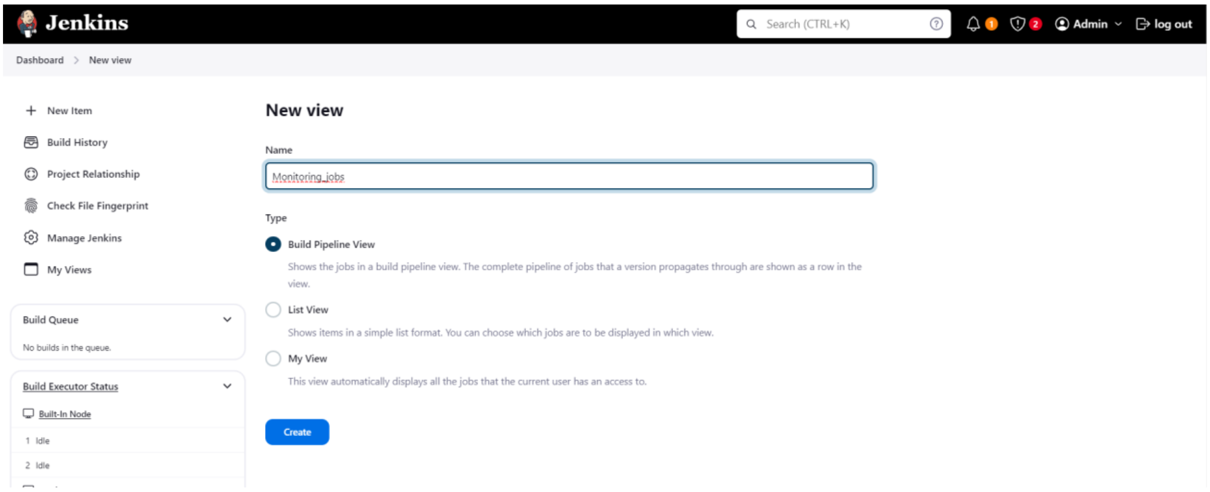
9. Build Pipeline:

For this we have to install the plugin build pipeline



build pipe

Install	Name ↓	Released
☑	Build Pipeline 2.0.2 User Interface Build Tools Other Post-Build Actions This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins. Warning: This plugin version may not be safe to use. Please review the following security notices: • Stored XSS vulnerability **This plugin is up for adoption!** We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	1 mo 24 days ago
□	Webhook Step 342.v620877effe14 Allows build pipelines to wait for notification from an external system before continuing.	5 mo 24 days ago
□	Pipeline timeline 1.0.3 An interactive build timeline to help you visualize your build pipeline and identify bottlenecks.	5 yr 5 mo ago



New view

Name: Monitoring_jobs

Type

● Build Pipeline View

Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

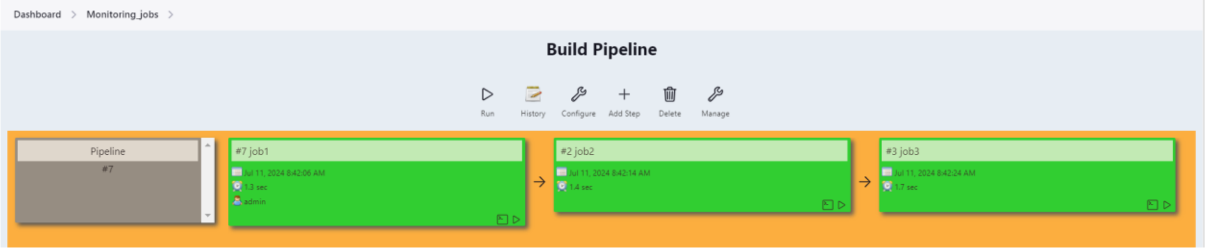
○ List View

Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

○ My View

This view automatically displays all the jobs that the current user has an access to.

Create



#7 job1
Jul 11, 2024 8:42:06 AM
1.3 sec
admin

#2 job2
Jul 11, 2024 8:42:14 AM
1.4 sec

#3 job3
Jul 11, 2024 8:42:24 AM
1.7 sec