Project : Capstone I

You have been hired as a Sr. DevOps Engineer in Abode Software. They want to implement DevOps Lifecycle in their company. You have been asked to implement this lifecycle as fast as possible. Abode Software is a product-based company and their product is available on this GitHub link.

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

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

Solution:

Launch 3 instances for master, test and prod

Project1-master	i-0c9933c7e2298e88f	Running	t2.micro	2/2 checks passed	View alarms +	ap-sc
Project1-test	i-093b896d0fb164098	Running	t2.micro	2/2 checks passed	View alarms +	ap-sc
Project1-prod	i-0e84565cd3964ba37	Running	t2.micro	2/2 checks passed	View alarms +	ap-sc

Connect to the instances and update all

```
Building dependency tree... Done
Reading state information... Done
77 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-23-82:~$
```

i-0c9933c7e2298e88f (Project1-master)

PublicIPs: 54.179.90.103 PrivateIPs: 172.31.23.82

```
Reading state information... Done
77 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-30-253:~$
```

i-093b896d0fb164098 (Project1-test)

PublicIPs: 13.215.141.128 PrivateIPs: 172.31.30.253

```
Building dependency tree... Done
Reading state information... Done
77 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-23-210:~$
```

i-0e84565cd3964ba37 (Project1-prod)

PublicIPs: 52.74.219.177 PrivateIPs: 172.31.23.210

In master branch create a.sh file and install ansible

```
ubuntu@ip-172-31-23-82:~$ sudo nano a.sh
ubuntu@ip-172-31-23-82:~$ bash a.sh
```

Create a keypair $ ssh-keygen

```
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_ed25519
Your public key has been saved in /home/ubuntu/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:ObKqF8NJfHDxHrD68x9gG+e4O7VdFV/EJp+kFIC4rK8 ubuntu@ip-172-31-23-82
The key's randomart image is:
+--[ED25519 256]--+
|      o.. .....o.|
|     . .= .   ..o+|
|    . oo +   . o+=|
|     o..+ o   . .+|
|    o.oo S .    . |
|     =o + O.   .  |
|      o= o.oo .   |
|     .. +.....    |
|   .o.E. ++.      |
+----[SHA256]-----+
```

Go inside ssh there you will find the keys – private key and public key

```
ubuntu@ip-172-31-23-82:~$ cd .ssh
ubuntu@ip-172-31-23-82:~/.ssh$ ls
authorized_keys  id_ed25519  id_ed25519.pub
ubuntu@ip-172-31-23-82:~/.ssh$ sudo cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDsq1GAdstaT0KD8QzVoRYCtMGQr0lBxyE6weOiOIkeV ubuntu@ip-172-31-23-82
```

Go inside authorized key

```
ubuntu@ip-172-31-30-253:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-30-253:~/.ssh$
```

and paste the public key in both test and prod servers

```
  GNU nano 7.2                                                    authorized_keys *
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCAha9lFDdaSWXxfAL/v61BpNUoZV6D8QmKbn1ipj7XTSnkI
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIDsq1GAdstaT0KDSQzVoRYCtMGQr0lBxyE6weOiOIkeV ubun
```

Come to master node go to the location /etc/ansible/hosts

```
ubuntu@ip-172-31-23-82:~/.ssh$ cd
ubuntu@ip-172-31-23-82:~$ cd /etc/ansible/
ubuntu@ip-172-31-23-82:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-23-82:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-23-82:/etc/ansible$
```

Create the hosts adding private ip of test and prod server

```
  GNU nano 7.2
[group]
Test ansible_host=172.31.30.253
Prod ansible_host=172.31.23.210
```

Ping the request for slave nodes $ ansible –m ping all

```
Prod | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
yes
Test | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Now install the tools over the playbook of ansible

```
ubuntu@ip-172-31-23-82:~$ sudo nano b.sh
ubuntu@ip-172-31-23-82:~$ sudo nano c.sh
```

$ sudo nano b.sh → for master node

```
  GNU nano 7.2
sudo apt update
sudo apt install openjdk-17-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 c.sh → for slaves

```
  GNU nano 7.2
sudo apt update
sudo apt install openjdk-17-jdk -y
sudo apt install docker.io -y
```

Create a playbook yaml file for execution of shell script

```
ubuntu@ip-172-31-23-82:~$ sudo cat play.yml
---
 - name: executing script on master
   hosts: localhost
   become: true
   tasks:
    - name: executing b.sh script on master
      script: b.sh
 - name: executing script on  slaves
   hosts: all
   become: true
   tasks:
    - name: executing c.sh script on slaves
      script: c.sh
```

Do the syntax check

```
ubuntu@ip-172-31-23-82:~$ ansible-playbook play.yml --syntax-check

playbook: play.yml
```

Run the playbook

```
TASK [executing c.sh script on slaves] ***********************************************************************
changed: [Test]
changed: [Prod]

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

Check the docker version and java version on test server

```
ubuntu@ip-172-31-30-253:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4
ubuntu@ip-172-31-30-253:~$ java --version
openjdk 17.0.11 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1,
ubuntu@ip-172-31-30-253:~$
```

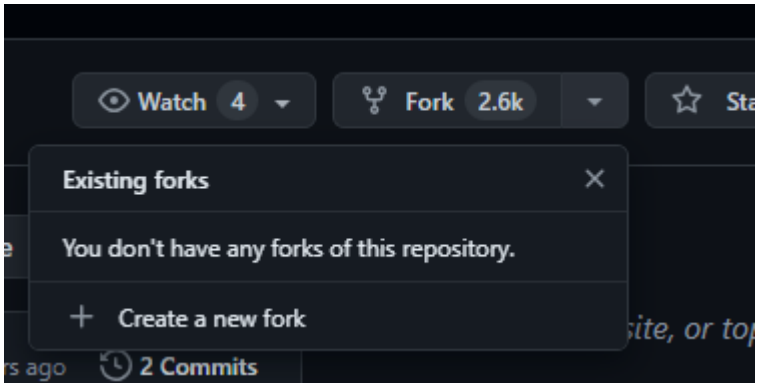
i-093b896d0fb164098 (Project1-test)

Check the docker version and java version on prod server as well

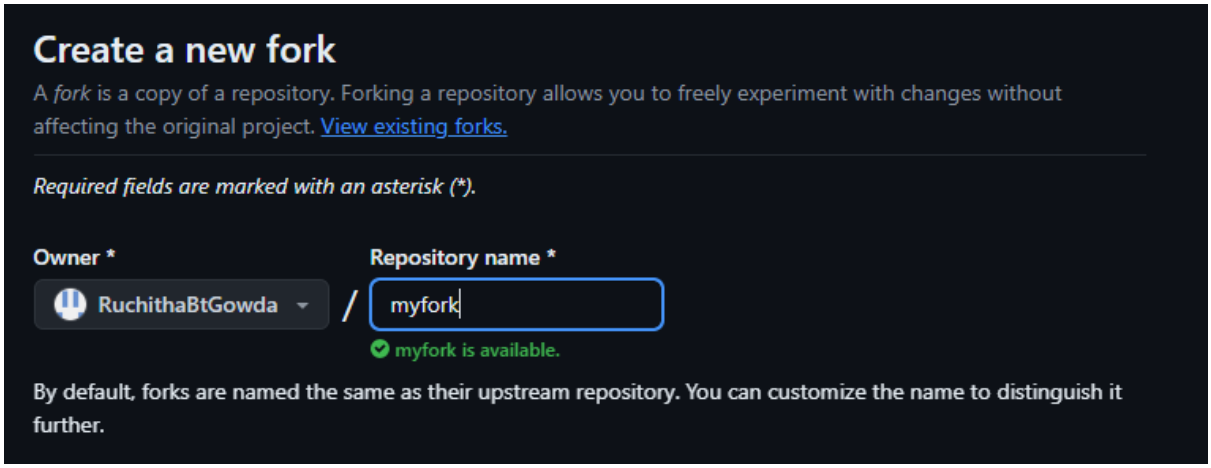
```
ubuntu@ip-172-31-23-210:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu4
ubuntu@ip-172-31-23-210:~$ java --version
openjdk 17.0.11 2024-04-16
OpenJDK Runtime Environment (build 17.0.11+9-Ubuntu-1)
OpenJDK 64-Bit Server VM (build 17.0.11+9-Ubuntu-1, mixed mode, sharing)
ubuntu@ip-172-31-23-210:~$
```

i-0e84565cd3964ba37 (Project1-prod)

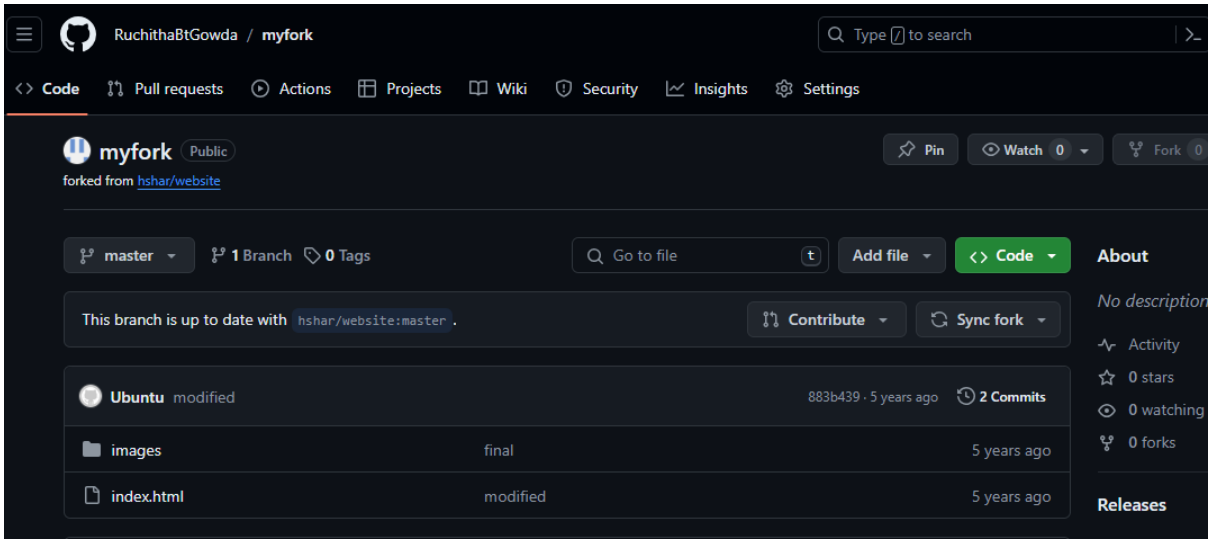
Go to the git hub location → https://github.com/hshar/website.git and create a new fork



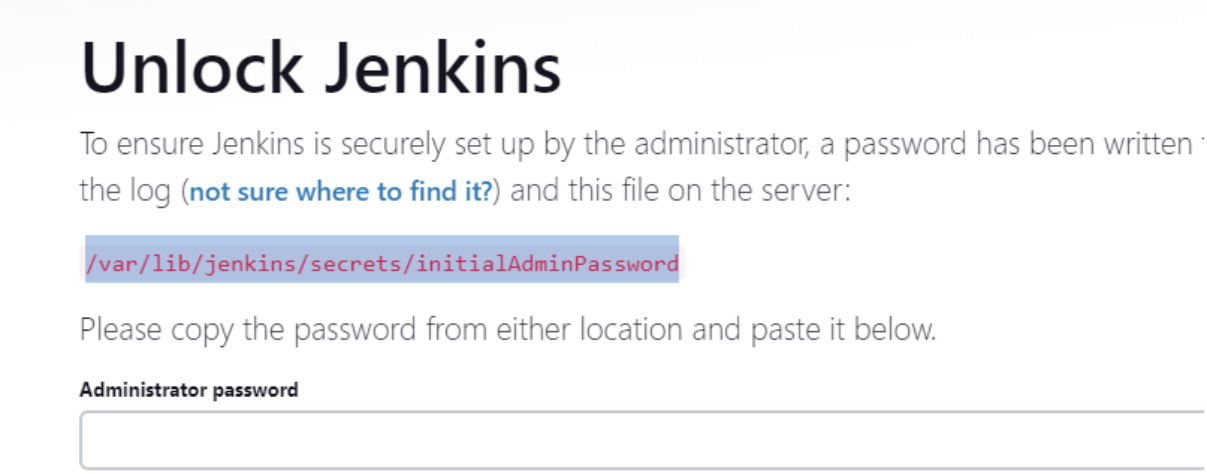
Give the repository name → myfork



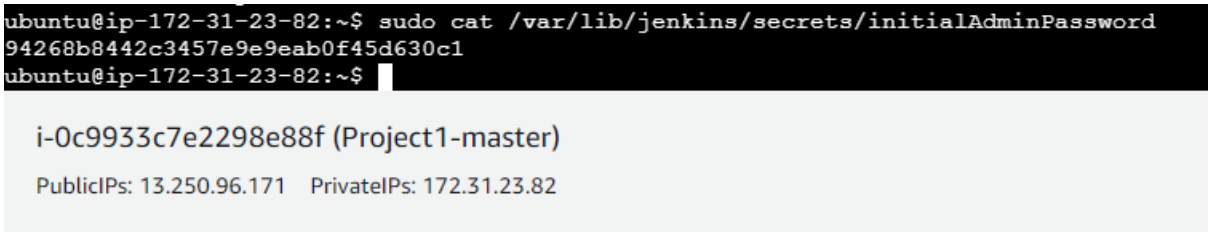
New copy is created



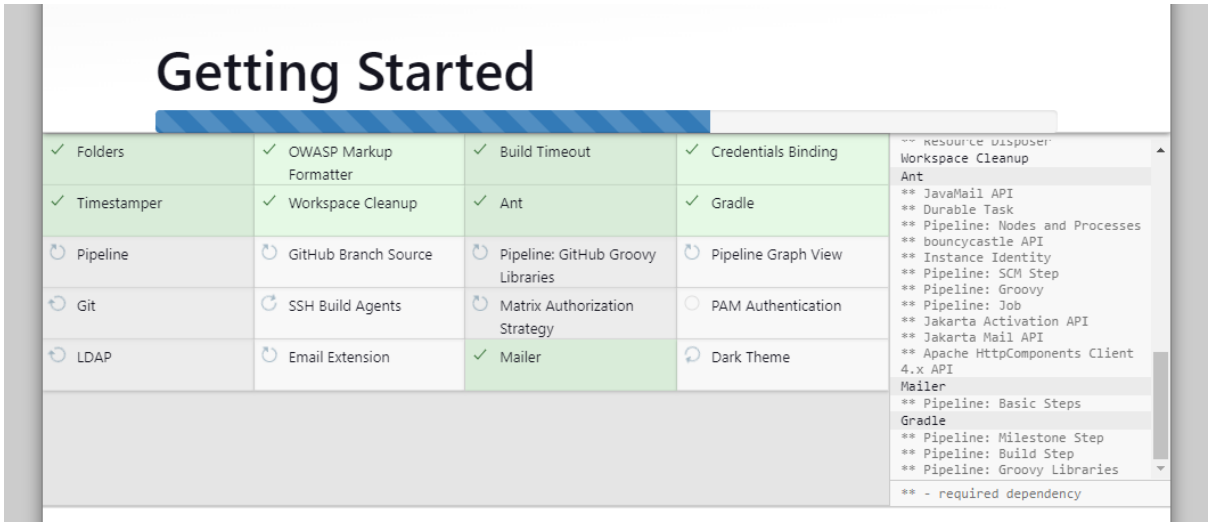
As the Jenkins is installed in the master node, copy the IP address of master node followed by port number 8080, Jenkins will be opened.



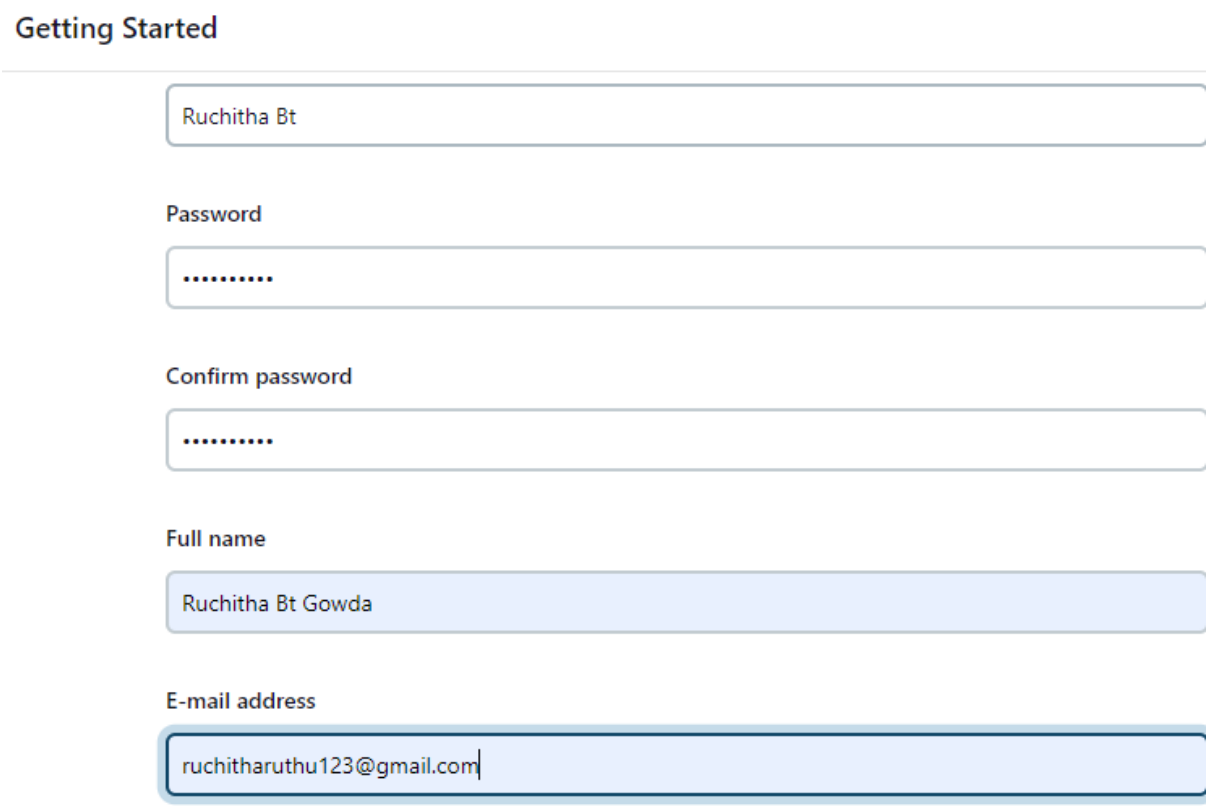
Go to location given and with the help of the password login into the Jenkins



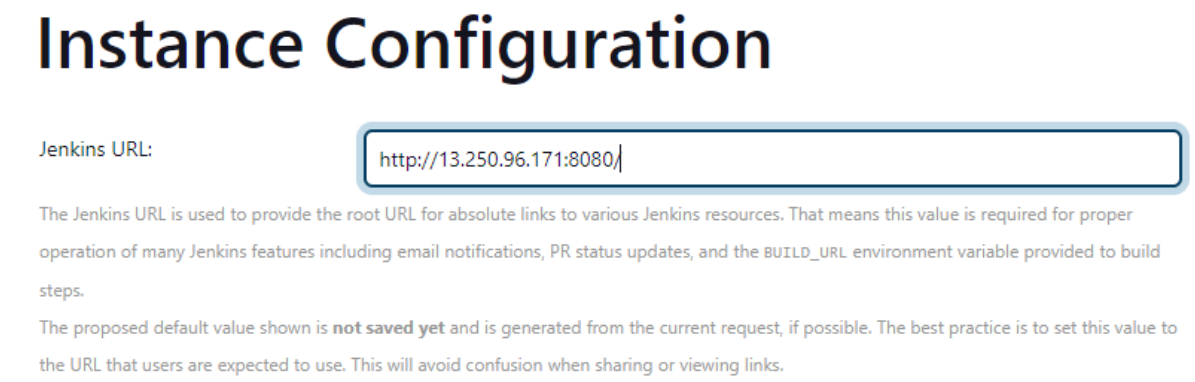
Go to the installation of available plugins and it get started installing.



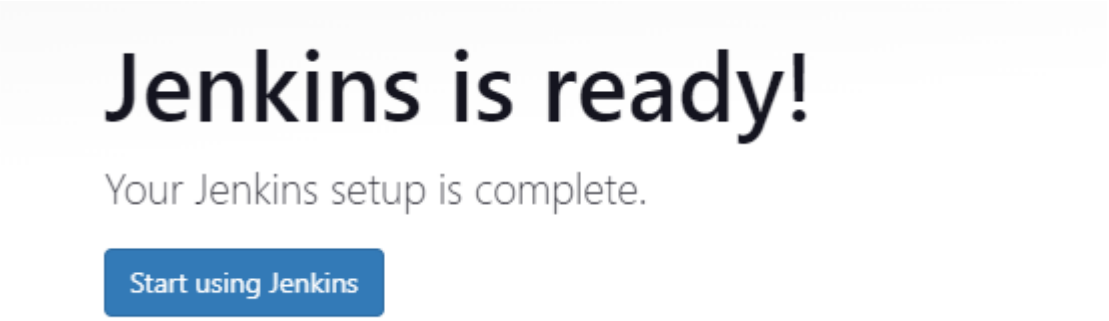
Give the necessary username, password, email address



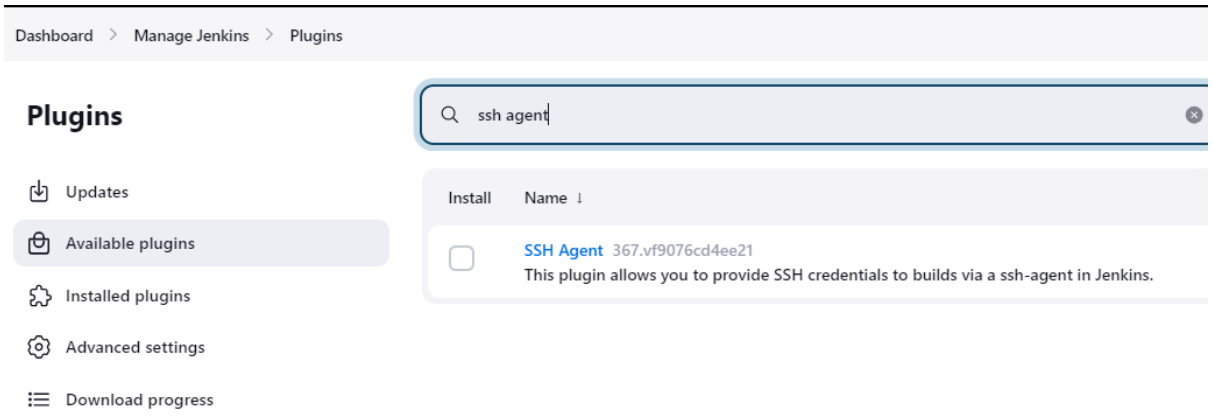
Jenkins URL



Setup is completed



Install the plugin – ssh agent



Install is success

Download progress

Preparation	• Checking internet connectivity • Checking update center connectivity • Success
Ionicons API	Success
Folders	Success
OWASP Markup Formatter	Success
ASM API	Success
JSON Path API	Success
Structs	Success
Pipeline: Step API	Success
Token Macro	Success
Build Timeout	Success
Credentials	Success
Plain Credentials	Success
Variant	Success

→ Go back to the top page
(you can start using the installed plugins right away)

Go back to the top page and setup an agent

Create a job

Set up a distributed build

Set up an agent

Configure a cloud

Create a nodes for test and prod

Dashboard > Nodes > New node

New node

Node name

test

Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

Remote root directory ?

/home/ubuntu/jenkins/

Labels ?

Usage ?

Use this node as much as possible

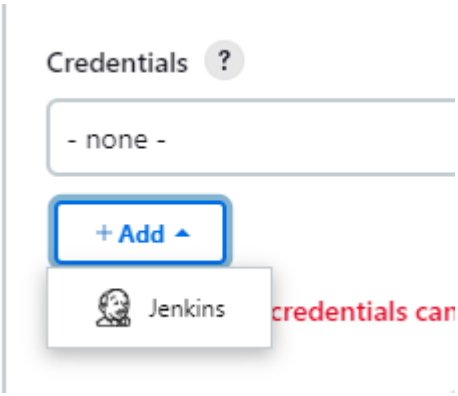
Launch method ?

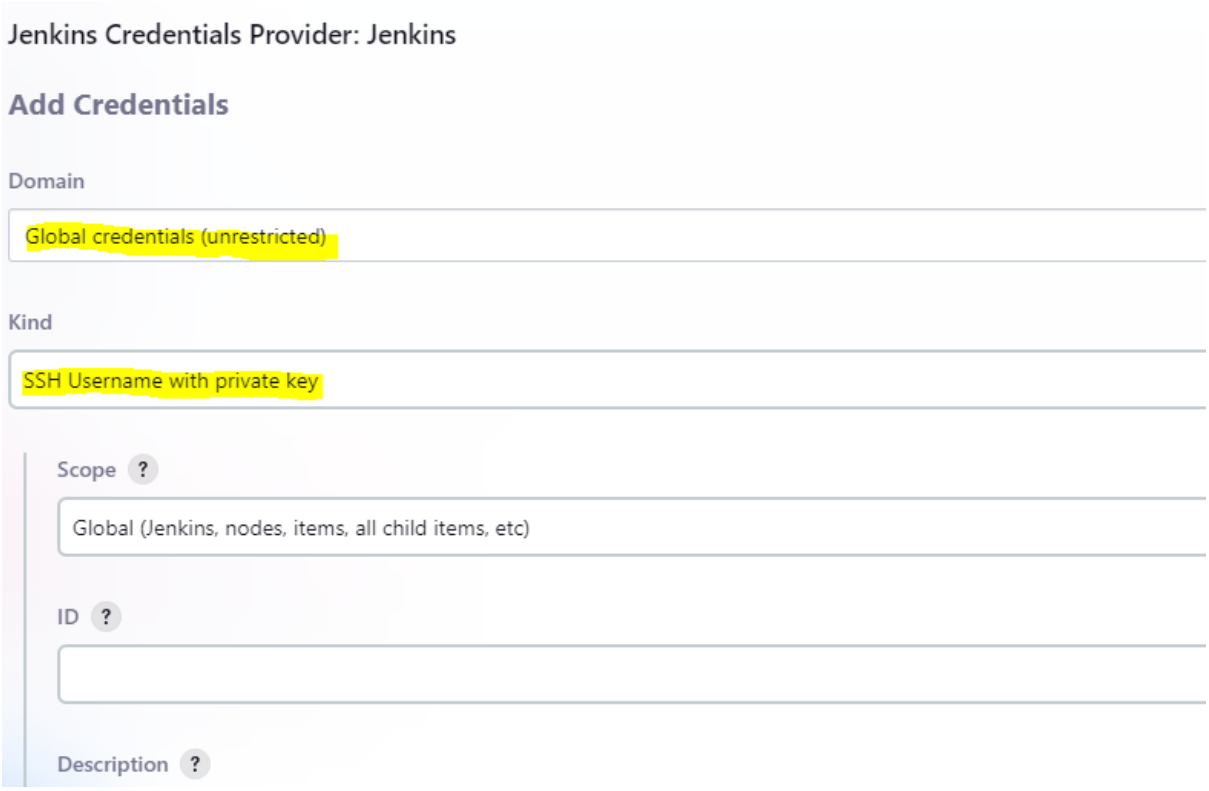
Launch agents via SSH

Host ?

172.31.30.253

Credentials ?


Credentials
- none -
+ Add
Jenkins
credentials can


Jenkins Credentials Provider: Jenkins
Add Credentials
Domain
Global credentials (unrestricted)
Kind
SSH Username with private key
Scope
Global (Jenkins, nodes, items, all child items, etc)
ID
Description

Jenkins Credentials Provider: Jenkins

Username

ubuntu

Treat username as secret ?

Private Key

Enter directly

Key

-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEAgIWvZRQ3Wkll8XwC/7+tQaTVKGVeg/EJim59YqY+100p5CPS
ITklyD5Xwu8Kv7nn7n01snVz2Gt2rIOdCMESnuFXge6gFCqv7jsJi/05IyajwoJT

Cancel

Add

Credentials ?

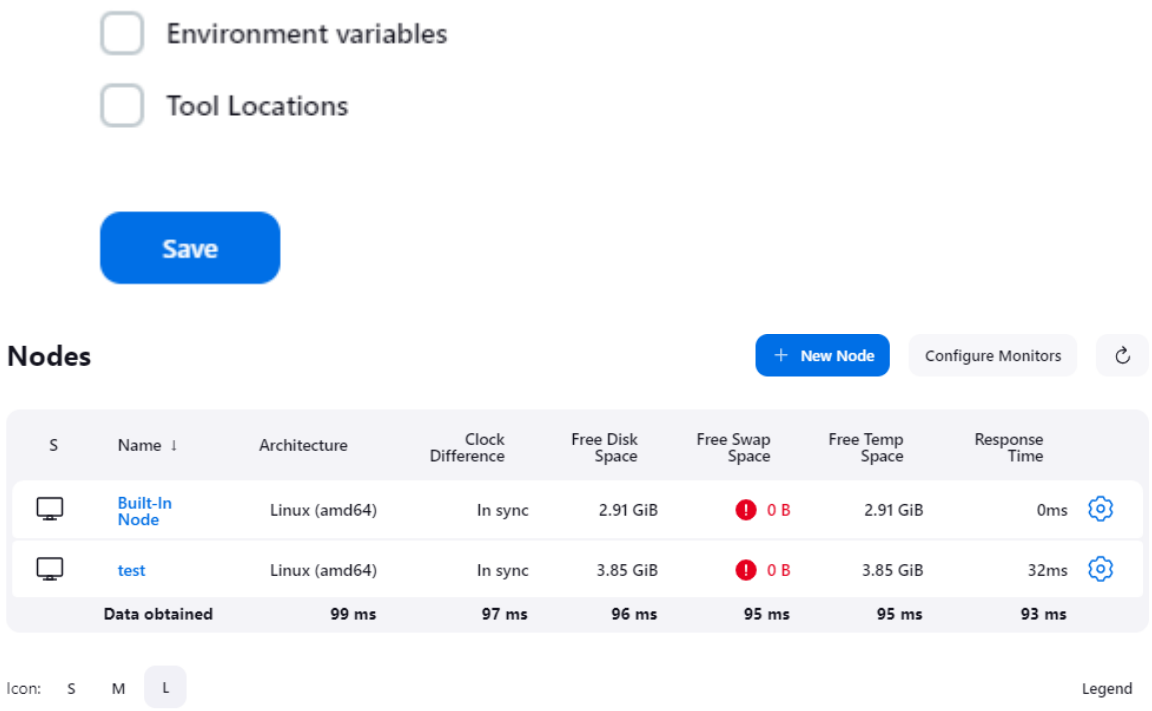
ubuntu

+ Add

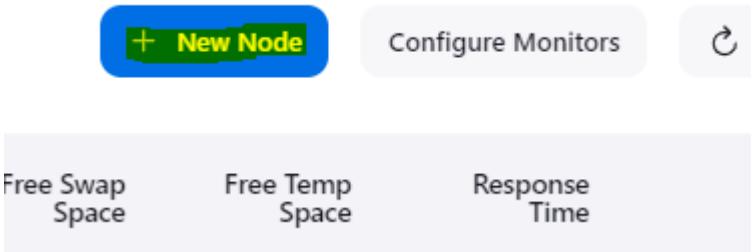
Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced


Environment variables
Tool Locations
Save
Nodes
New Node
Configure Monitors
S
Name
Architecture
Clock Difference
Free Disk Space
Free Swap Space
Free Temp Space
Response Time
Built-In Node
Linux (amd64)
In sync
2.91 GiB
0 B
2.91 GiB
0ms
test
Linux (amd64)
In sync
3.85 GiB
0 B
3.85 GiB
32ms
Data obtained
99 ms
97 ms
96 ms
95 ms
95 ms
93 ms
Icon:
S
M
L
Legend

For prod


New Node
Configure Monitors
Free Swap Space
Free Temp Space
Response Time

New node

Node name

prod

Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Copy Existing Node

test

Create

Launch method ?

Launch agents via SSH

Host ?

172.31.23.210

Credentials ?

ubuntu

Tool Locations

Save

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	2.91 GiB	0 B	2.91 GiB	0ms
	prod	Linux (amd64)	In sync	3.85 GiB	0 B	3.85 GiB	34ms
	test	Linux (amd64)	In sync	3.85 GiB	0 B	3.85 GiB	5ms
	Data obtained	3 sec	3 sec	3 sec	2.9 sec	3 sec	3 sec

Go to dashboard and create a new job

This page is where your Jenkins jobs will be displayed. To get started, y
builds or start building a software project.

Start building your software project

Create a job

Enter an item name

sample-job

» *Required field*

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM,
archiving artifacts and sending email notifications.

Pipeline

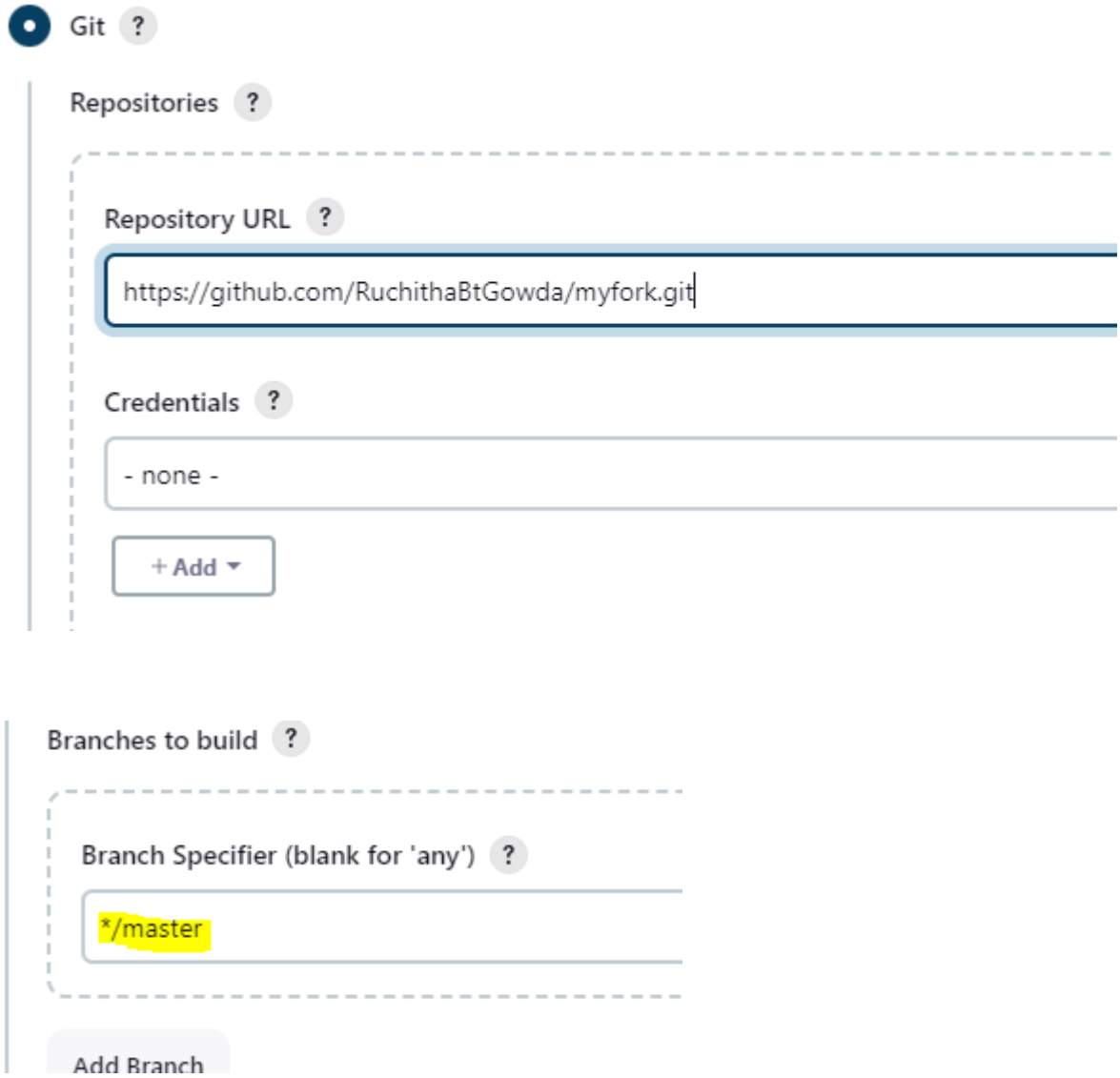
Orchestrates long-running activities that can span multiple build agents
and/or organizing complex activities that do not easily fit in free-style j

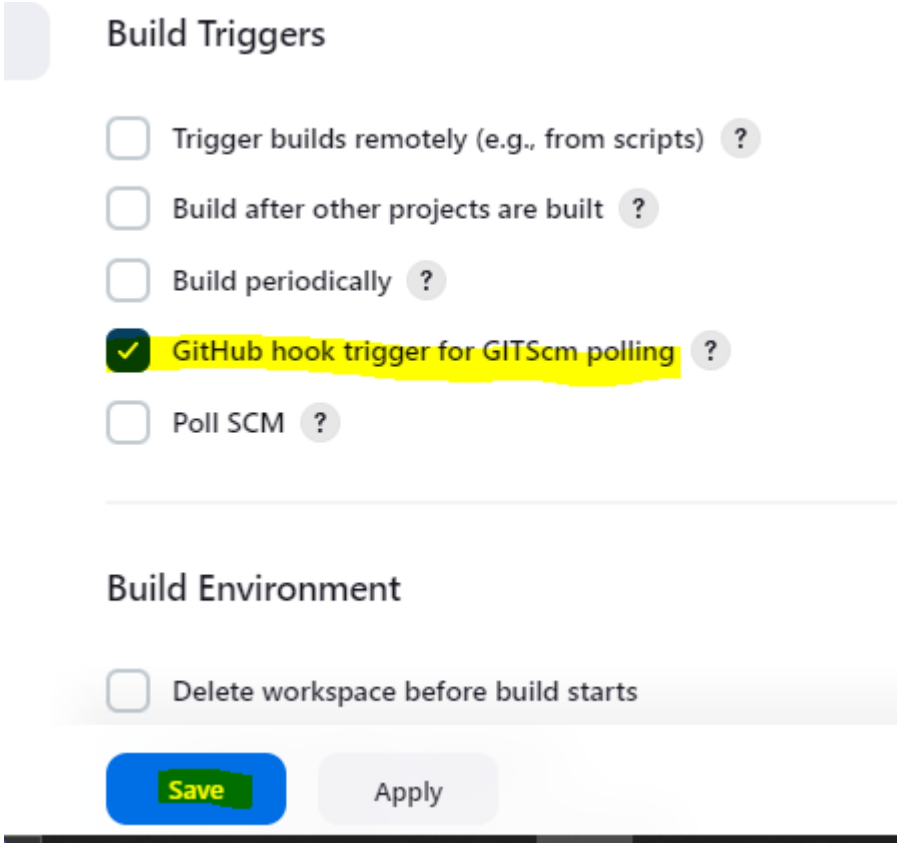
Multi-configuration project

Suitable for projects that need a large number of different configuratio
builds, etc.

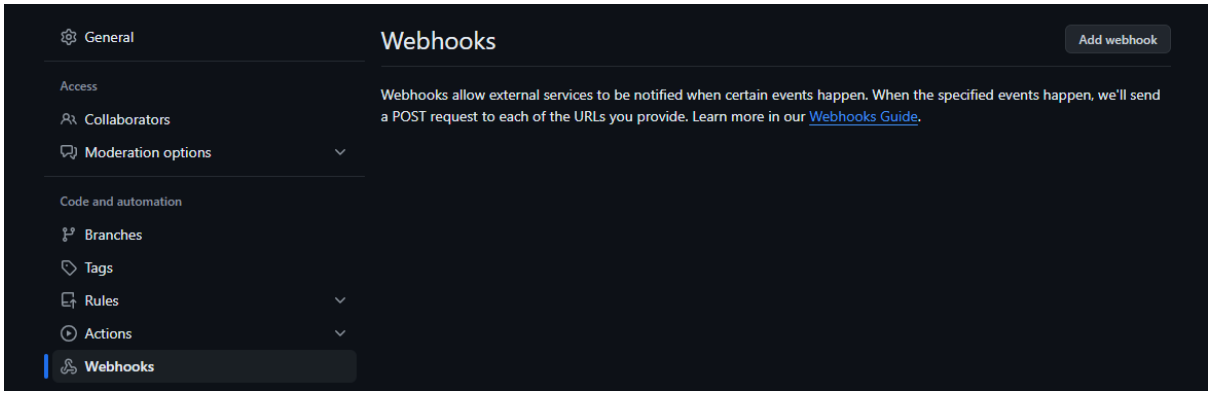
OK

As this public repo no need of giving credentials




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 Environment
Delete workspace before build starts
Save
Apply

Add webhook


General
Access
Collaborators
Moderation options
Code and automation
Branches
Tags
Rules
Actions
Webhooks
Webhooks
Add webhook
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.

We'll send a POST request to the URL below with details of any subscribed events. You c
format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can b
documentation.

Payload URL *

http://13.250.96.171:8080/github-webhook/

Content type

application/x-www-form-urlencoded

Secret

Active

We will deliver event details when this hook is triggered.

Add webhook

Webhooks

Add webhook

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://13.250.96.171:8080/github-w... *(push)*

Edit Delete

Build the job

sample-job > #1 > Console Output

s

Output

as plain text

ld Information

build '#1'

Console Output

```
Started by user Ruchitha Bt Gowda
Running as SYSTEM
Building on the built-in node in workspace /var/lib/jenkins/workspace/sample-job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/RuchithaBtGowda/myfork.git
 > git init /var/lib/jenkins/workspace/sample-job # timeout=10
Fetching upstream changes from https://github.com/RuchithaBtGowda/myfork.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
```

Check in master

```
ubuntu@ip-172-31-23-82:~$ cd /var/lib/jenkins/workspace/sample-job
ubuntu@ip-172-31-23-82:/var/lib/jenkins/workspace/sample-job$ ls
images  index.html
ubuntu@ip-172-31-23-82:/var/lib/jenkins/workspace/sample-job$
```

i-0c9933c7e2298e88f (Project1-master)

Create another job for test–test-job

Dashboard > test-job > Configuration

But as per question it should build in prod

Restrict where this project can be run ?

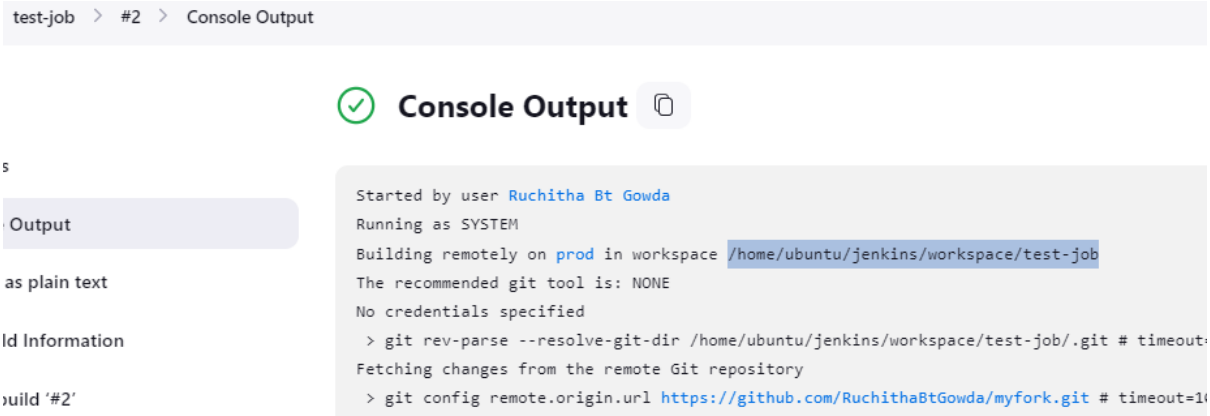
Label Expression ?

prod

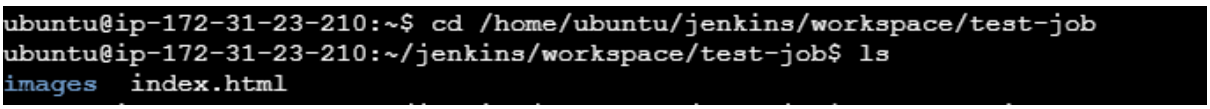
Label prod matches 1 node. Permissions or other restrictions provided by plugins may further reduce th

Advanced

You can see it is building remotely on prod



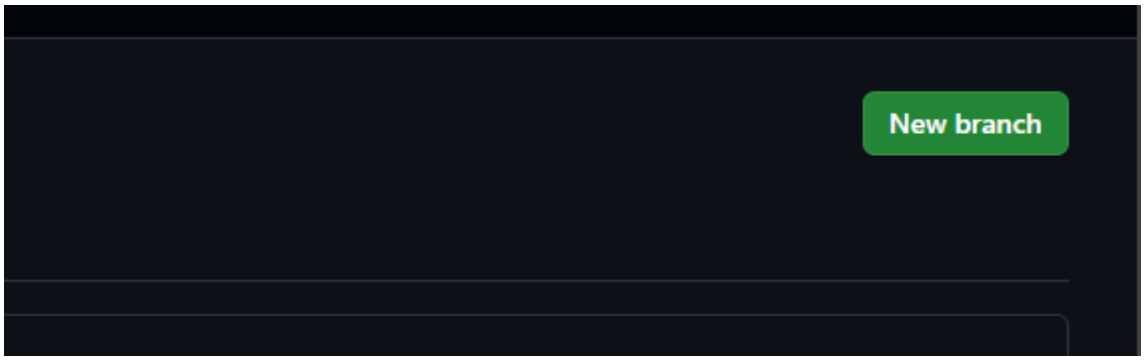
Go to prod server and check it



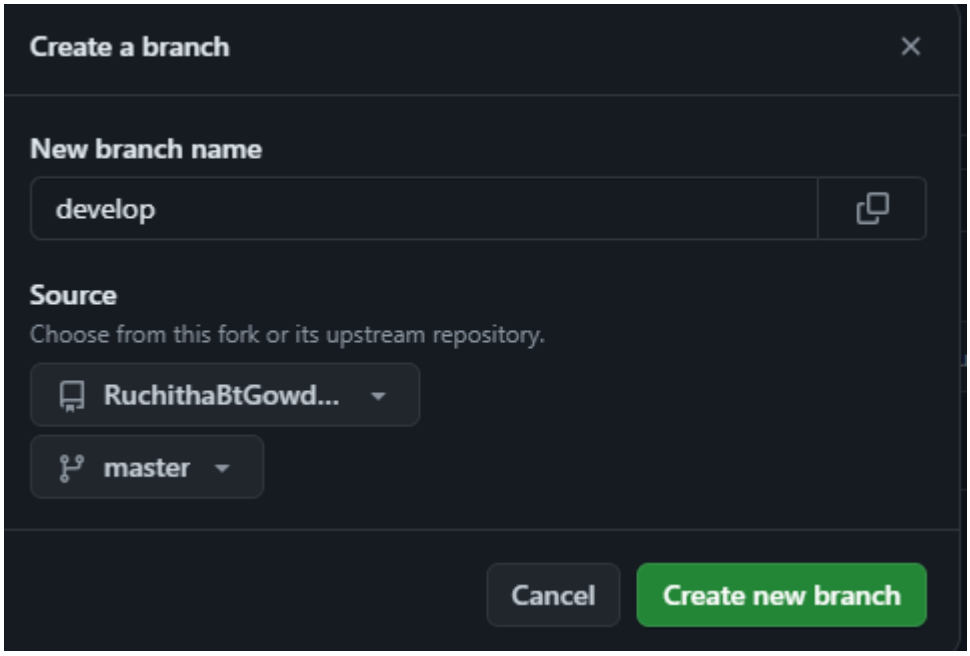
As of now two jobs are created where the commit is made to master branch

S	W	Name ↓	Last Success	Last Failure	Las
✓	☼	sample-job	1 day 18 hr #1	N/A	4.5
✓	☼	test-job	3 min 43 sec #4	N/A	0.8

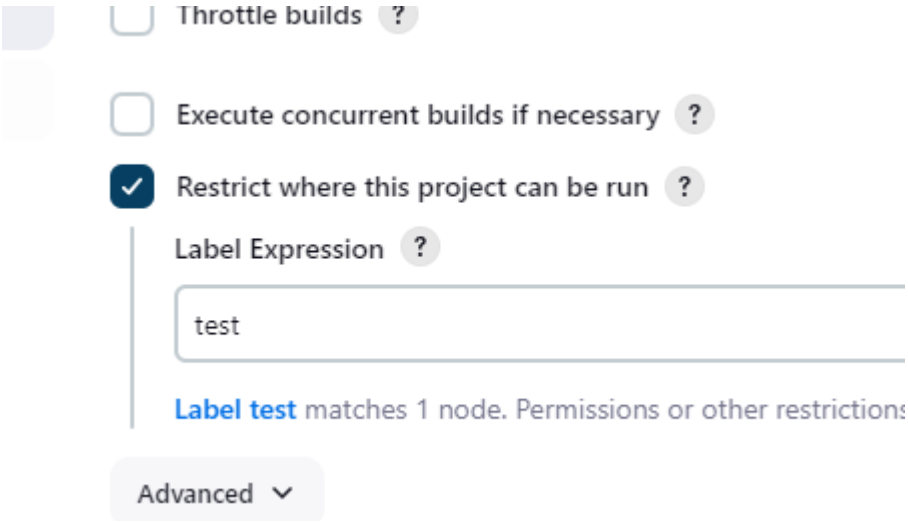
Now to create the prod job we need develop branch → go to new branch



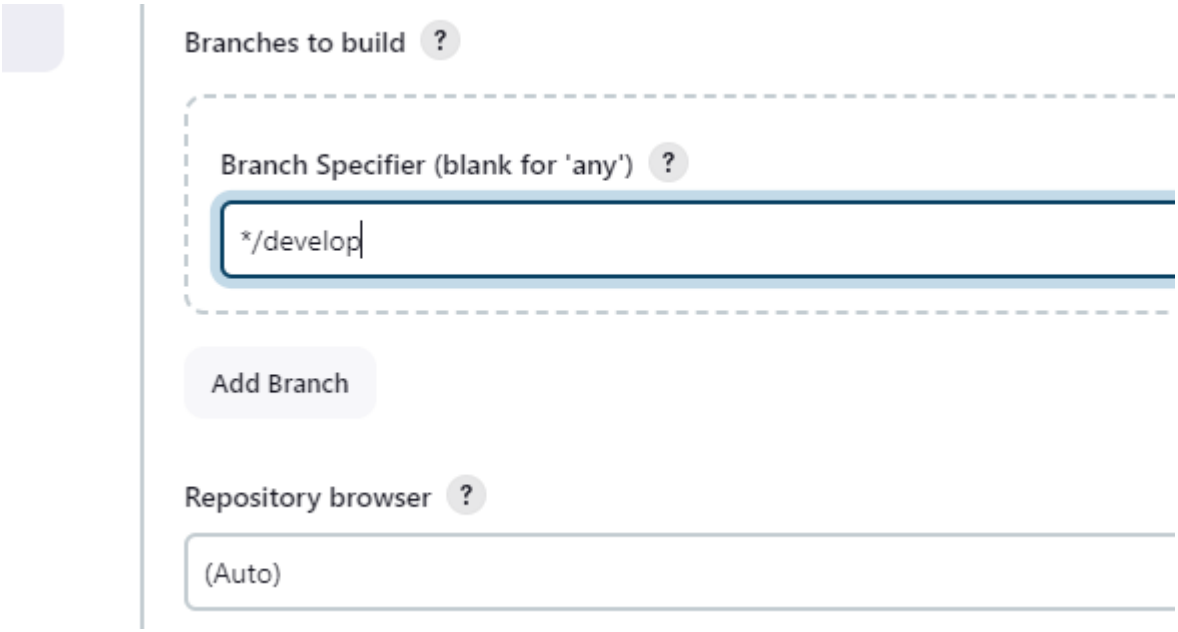
Give the name → create new branch



Give here test node



Branch to build is develop



Run the job, you can see prod job is building remotely on test server

Console Output

```
Started by user Ruchitha Bt Gowda
Running as SYSTEM
Building remotely on test in workspace /home/ubuntu/jenkins/workspace/prod-job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/RuchithaBtGowda/myfork.git
 > git init /home/ubuntu/jenkins/workspace/prod-job # timeout=10
Fetching upstream changes from https://github.com/RuchithaBtGowda/myfork.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.0'
```

Go to test server and check the location files

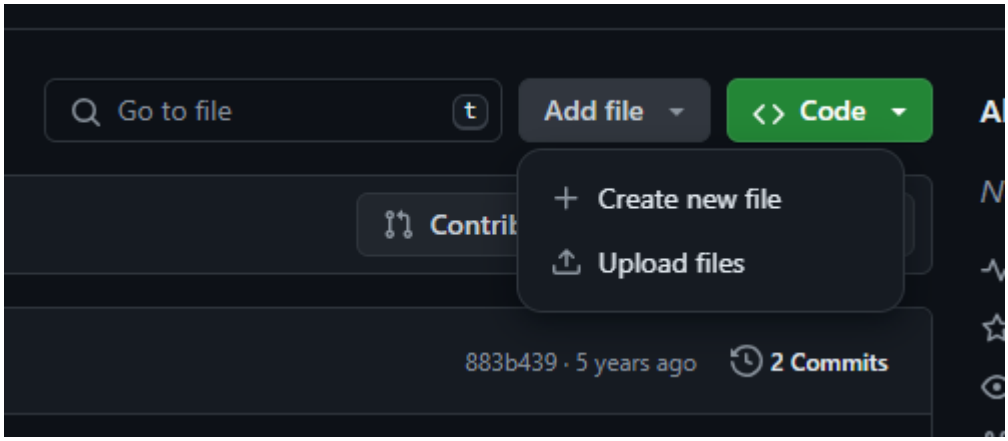
```
ubuntu@ip-172-31-30-253:~$ cd /home/ubuntu/jenkins/workspace/prod-job
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$ ls
images  index.html
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$
```

i-093b896d0fb164098 (Project1-test)

Now till 3rd task completed

Create docker file

In master branch



Write a Docker file which installs apache2 which is already present on Ubuntu

myfork / Dockerfile in master

Edit Preview Code 55% faster with GitHub Copilot

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

Commit the changes

Commit changes

Commit message

Create Dockerfile

Extended description

Add an optional extended description..

Commit directly to the master branch

Create a new branch for this commit and start a pull request Learn more about pull requests

Cancel Commit changes

Go to prod server and check

```
Last login: Sat Jun 22 12:04:05 2024 from 3.0.5.35
ubuntu@ip-172-31-23-210:~$ cd /home/ubuntu/jenkins/workspace/test-job
ubuntu@ip-172-31-23-210:~/jenkins/workspace/test-job$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-23-210:~/jenkins/workspace/test-job$
```

i-0e84565cd3964ba37 (Project1-prod)

PublicIPs: 54.255.144.252 PrivateIPs: 172.31.23.210

Similarly create docker file on develop branch

myfork / Dockerfile in develop

Edit Preview Code 55% faster with GitHub Copilot

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

Commit the changes

Cancel changes Commit changes...

Spaces 2 No wrap

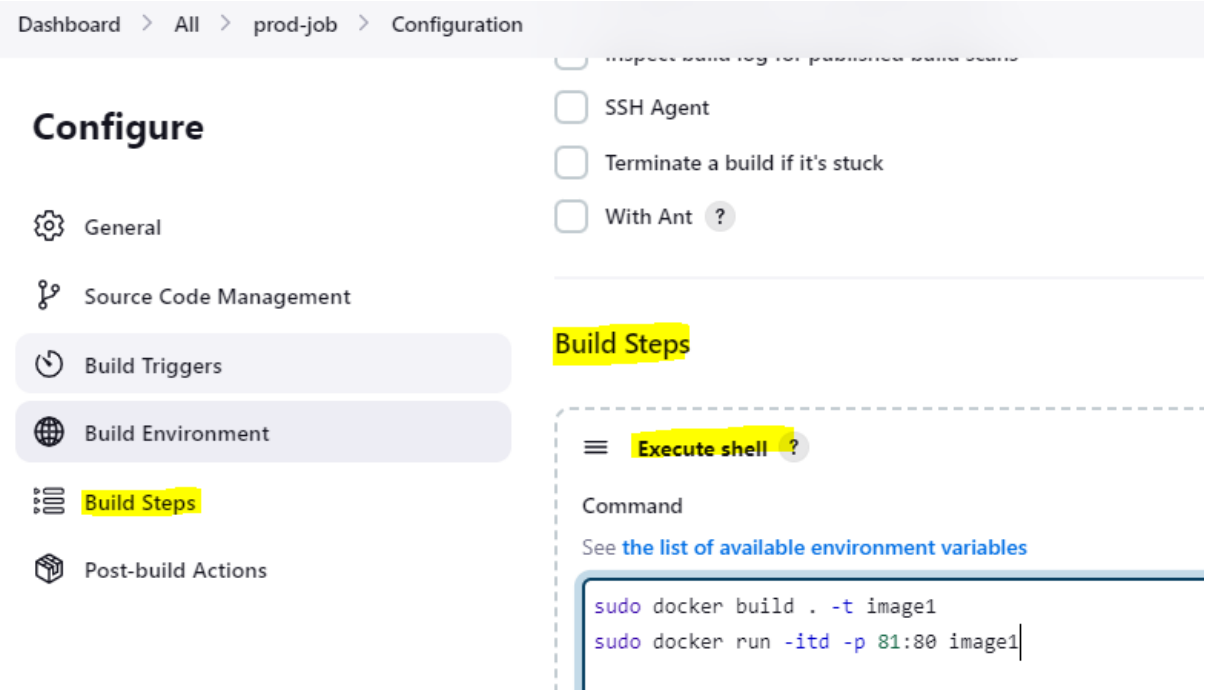
Check in the test server

```
ubuntu@ip-172-31-30-253:~$ cd /home/ubuntu/jenkins/workspace/prod-job
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$
```

i-093b896d0fb164098 (Project1-test)

PublicIPs: 54.255.89.166 PrivateIPs: 172.31.30.253

Now creating docker image



Run the prod job.

```
done.
Removing intermediate container fd3e12701599
 ---> 2c941af9bc8d
Step 4/5 : COPY . /var/www/html/
 ---> c96bf5a2533a
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
 ---> Running in 34459871ac34
Removing intermediate container 34459871ac34
 ---> c4c1fa8505db
Successfully built c4c1fa8505db
Successfully tagged image1:latest
+ sudo docker run -itd -p 81:80 image1
f8ce18dc21056587639c7578a02d6312dac6ac9ddc3552109d17a0725aeb
Finished: SUCCESS
```

Check in test server

$ sudo docker images

```
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED              SIZE
image1       latest    c4c1fa8505db   About a minute ago   222MB
ubuntu       latest    35a88802559d   2 weeks ago          78.1MB
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$
```

i-093b896d0fb164098 (Project1-test)

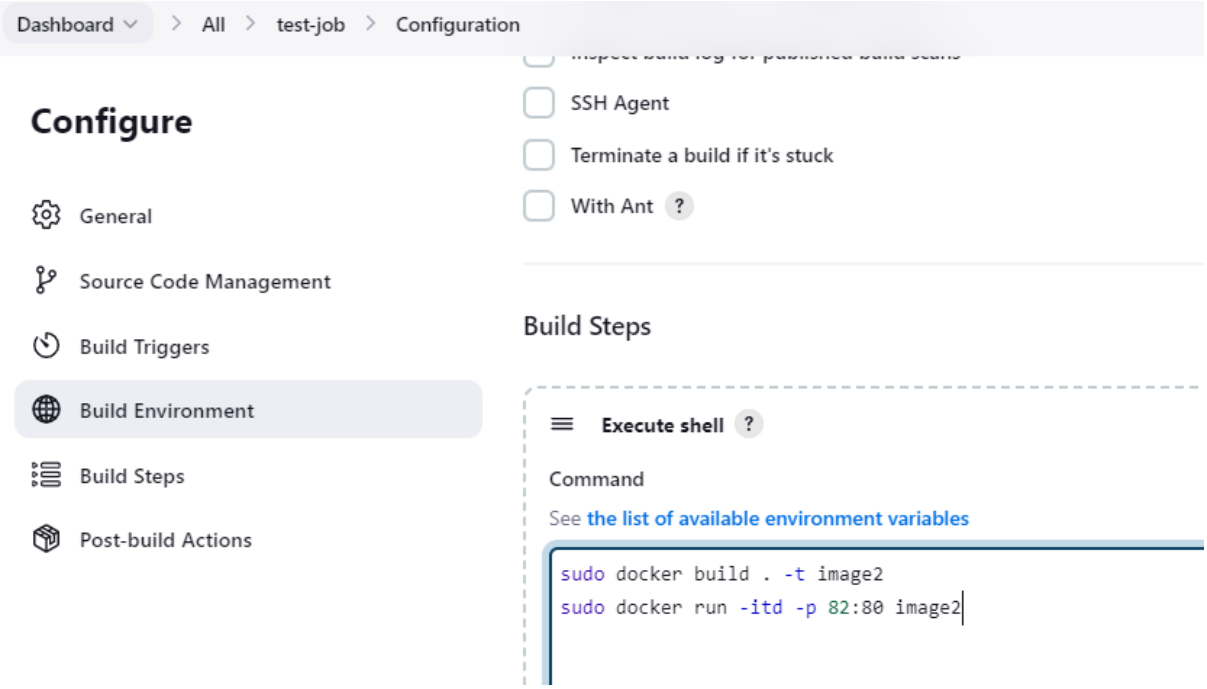
$ sudo docker ps

```
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS
f8ce18dc2105   image1    "/bin/sh -c 'apachec…"   2 minutes ago    Up 2 minutes    0.0.0.0:81->80/tcp, :::81->8
ubuntu@ip-172-31-30-253:~/jenkins/workspace/prod-job$
```

i-093b896d0fb164098 (Project1-test)

Public IP of test server with the given port

Similarly create a docker image in test job



Build the job, if you create webhooks it automatically builds

```
 ---> d19771942bbb
Step 4/5 : COPY . /var/www/html/
 ---> 7d0a140bd4e9
Step 5/5 : ENTRYPOINT apachectl -D FOREGROUND
 ---> Running in 8b8c93d12346
Removing intermediate container 8b8c93d12346
 ---> 9dd883f0ebe7
Successfully built 9dd883f0ebe7
Successfully tagged image2:latest
+ sudo docker run -itd -p 82:80 image2
8cb09b231f56b8e15360df4e85903ca501cc1b7500176c9726981ae3b3792403
Finished: SUCCESS
```

Go to prod server and check the images

$ sudo docker images

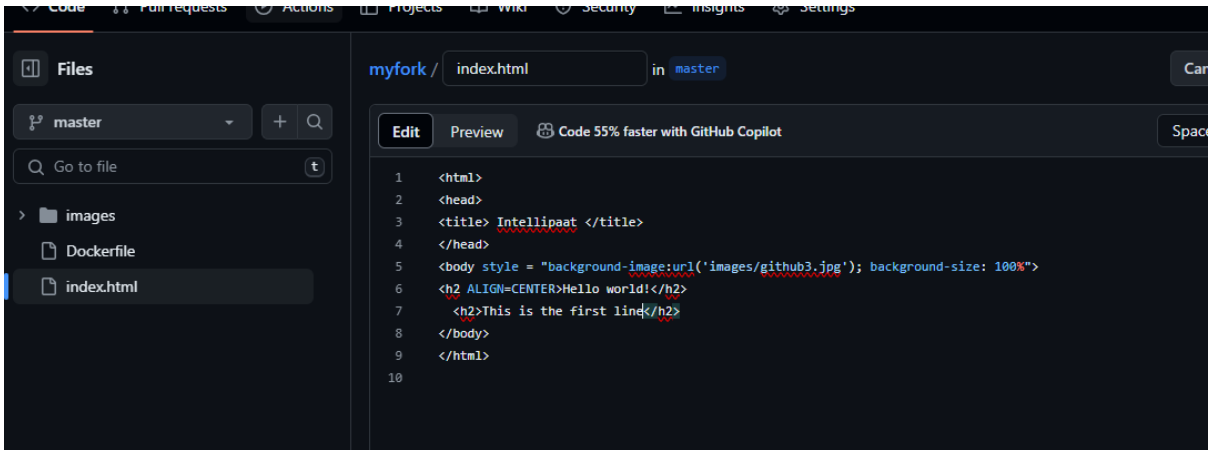
$ sudo docker ps

```
ubuntu@ip-172-31-23-210:~/jenkins/workspace/test-job$ sudo docker images
REPOSITORY   TAG       IMAGE ID       CREATED          SIZE
image2       latest    9dd883f0ebe7   8 minutes ago    222MB
ubuntu       latest    35a88802559d   2 weeks ago      78.1MB
ubuntu@ip-172-31-23-210:~/jenkins/workspace/test-job$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS
8cb09b231f56   image2    "/bin/sh -c 'apachec…"   8 minutes ago    Up 8 minutes
ubuntu@ip-172-31-23-210:~/jenkins/workspace/test-job$
```

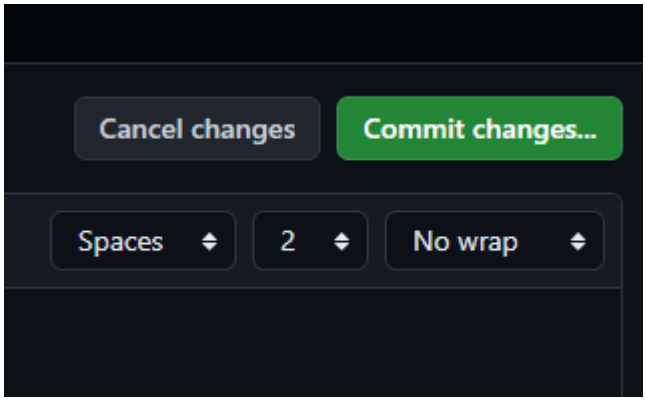
i-0e84565cd3964ba37 (Project1-prod)

Public IP of prod ec2 with given port number

You can perform the html change add any line



Commit the changes



And check them

Do the changes in shell script if you are further doing the html changes..

Build Steps

Execute shell

Command

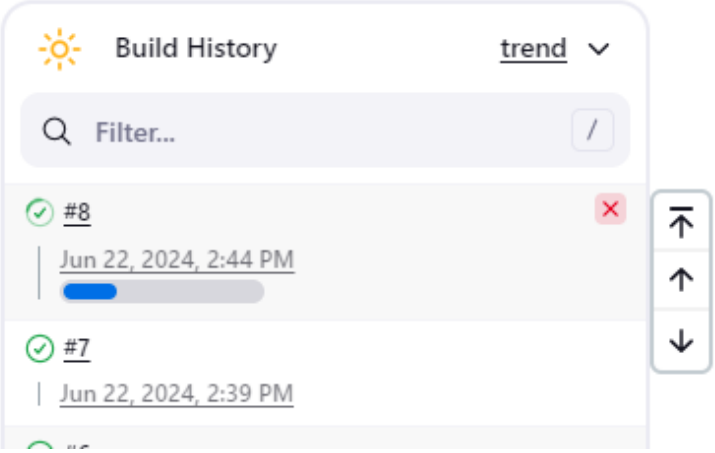
See the list of available environment variables

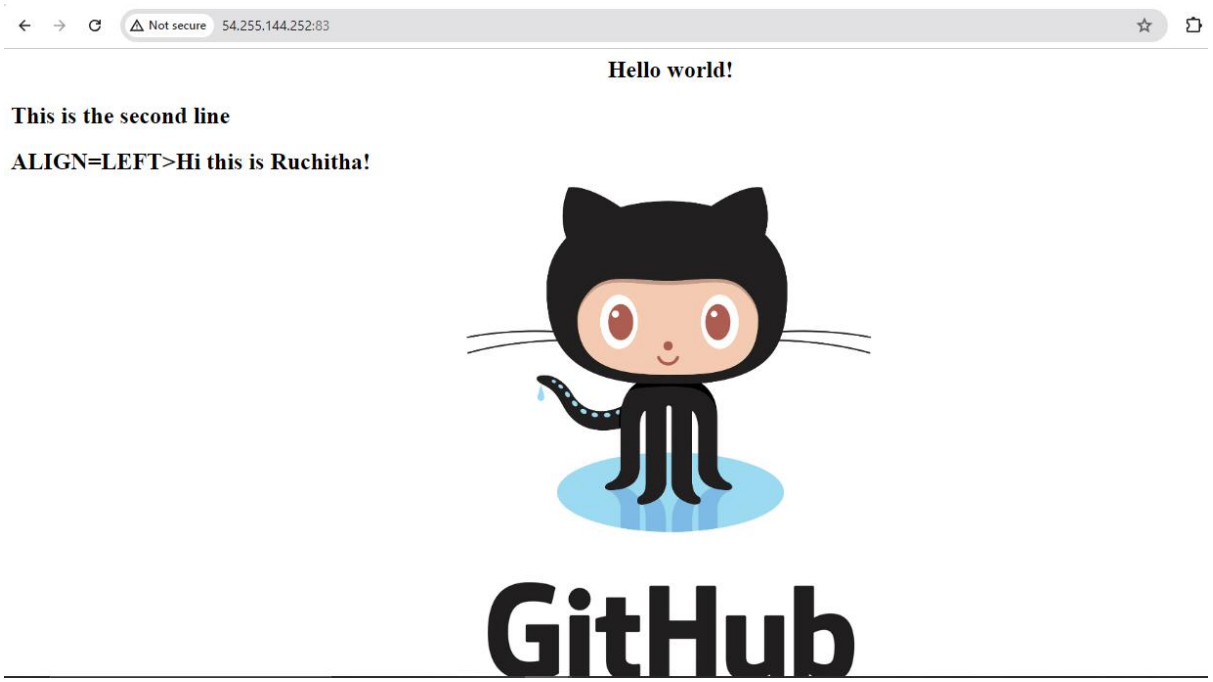
```
sudo docker rm -f c1
sudo docker build . -t image2
sudo docker run -itd -p 83:80 --name=c1 image2
```

Edit Preview Code 55% faster with GitHub Copilot

```
<html>
<head>
<title> Intellipaat </title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN=CENTER>Hello world!</h2>
  <h2>This is the second line</h2>
  <h2>ALIGN=LEFT>Hi this is Ruchitha!</h2>
</body>
</html>
```


Configure
Delete Project
GitHub Hook Log
Rename
Last build (#7), 2 min 27 sec ago
Last stable build (#7), 2 min 27 s
Last successful build (#7), 2 min
Last completed build (#7), 2 min


Build History
trend
Filter...
/
#8
Jun 22, 2024, 2:44 PM
#7
Jun 22, 2024, 2:39 PM


Not secure
54.255.144.252:83
Hello world!
This is the second line
ALIGN=LEFT>Hi this is Ruchitha!
GitHub

Build Steps

Execute shell

Command

See the list of available environment variables

```
#sudo docker rm -f Ruchitha
sudo docker build . -t image1
sudo docker run -itd -p 84:80 --name=Ruchitha image1
```

Advanced

Code | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

Files

develop

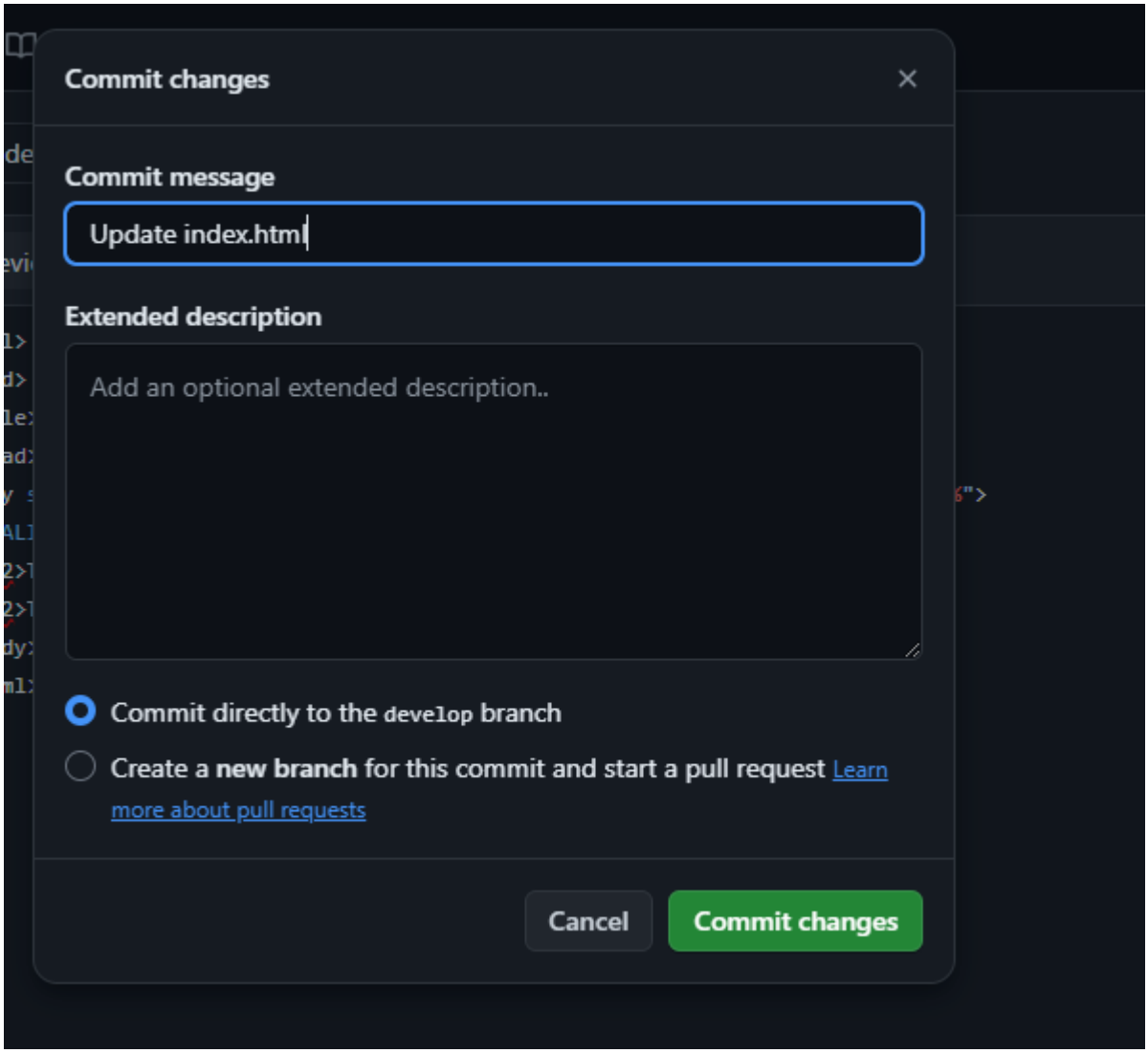
Go to file t

- images
- Dockerfile
- index.html

myfork / index.html in develop

Edit | Preview | Code 55% faster with GitHub Copilot

```
1  <html>
2  <head>
3  <title> Intellipaat </title>
4  </head>
5  <body style = "background-image:url('images/github3.jpg'); background-size: 100%">
6  <h2 ALIGN=CENTER>Hello world!</h2>
7    <h2>This my first line</h2>
8    <h2>This is my last line</h2>
9  </body>
10 </html>
11
```


Commit changes
Commit message
Update index.html
Extended description
Add an optional extended description..
Commit directly to the develop branch
Create a new branch for this commit and start a pull request Learn more about pull requests
Cancel
Commit changes

Not secure 54.255.89.166:84

Hello world!

This my first line

This is my last line

Execute shell ?

Command

See the list of available environment variables

```
sudo docker rm -f Ruchitha
sudo docker build . -t image1
sudo docker run -itd -p 84:80 --name=Ruchitha image1
```

Advanced

myfork / index.html in develop

Edit Preview Code 55% faster with GitHub Copilot

```html
<html>
<head>
<title> Intellipaat </title>
</head>
<body style = "background-image:url('images/github3.jpg'); background-size: 100%">
<h2 ALIGN=CENTER>Hello world!</h2>
  <h2>This my first line</h2>
  <h2>This is my last line</h2>
  <h2>Thankyou!</h2>
</body>
</html>
```

Commit changes

Commit message

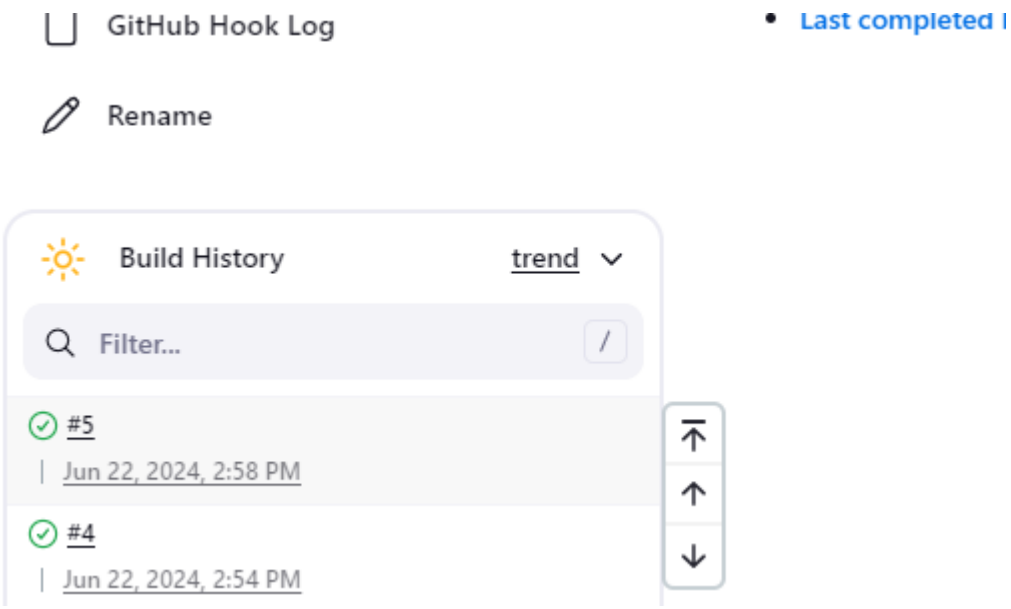
Update index.html

Extended description

Add an optional extended description..

- (●) Commit directly to the `develop` branch
- ( ) Create a **new branch** for this commit and start a pull request Learn more about pull requests

Cancel | Commit changes


GitHub Hook Log
Last completed
Rename
Build History
trend
Filter...
/
#5
Jun 22, 2024, 2:58 PM
#4
Jun 22, 2024, 2:54 PM


Not secure
54.255.89.166:84
Hello world!
This my first line
This is my last line
Thankyou!
GitHub