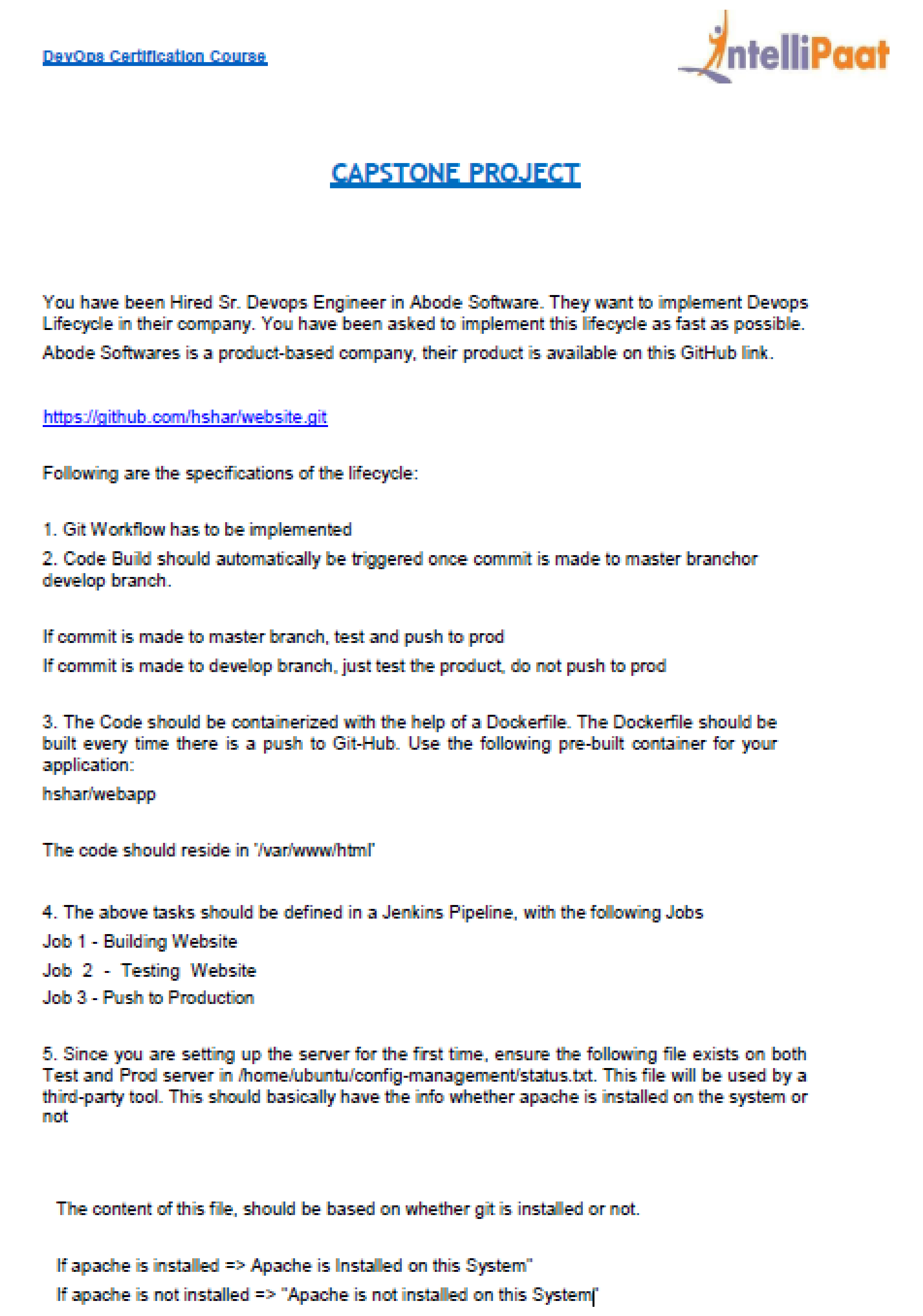
DevOps Project - I

Capstone Project- I



DevOps Certification Course

IntelliPaat

CAPSTONE PROJECT

You have been Hired 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 Softwares is a product-based company, their product is available on this GitHub link.

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

Following are the specifications of the lifecycle:

1. Git Workflow has to be implemented
2. Code Build should automatically be triggered once commit is made to master branchor develop branch.

If commit is made to master branch, test and push to prod

If commit is made to develop branch, just test the product, do not push to prod

3. The Code should be containerized with the help of a Dockerfile. The Dockerfile should be built every time there is a push to Git-Hub. Use the following pre-built container for your application:

hshar/webapp

The code should reside in '/var/www/html'

4. The above tasks should be defined in a Jenkins Pipeline, with the following Jobs

Job 1 - Building Website

Job 2 - Testing Website

Job 3 - Push to Production

5. Since you are setting up the server for the first time, ensure the following file exists on both Test and Prod server in /home/ubuntu/config-management/status.txt. This file will be used by a third-party tool. This should basically have the info whether apache is installed on the system or not

The content of this file, should be based on whether git is installed or not.

If apache is installed => Apache is Installed on this System"

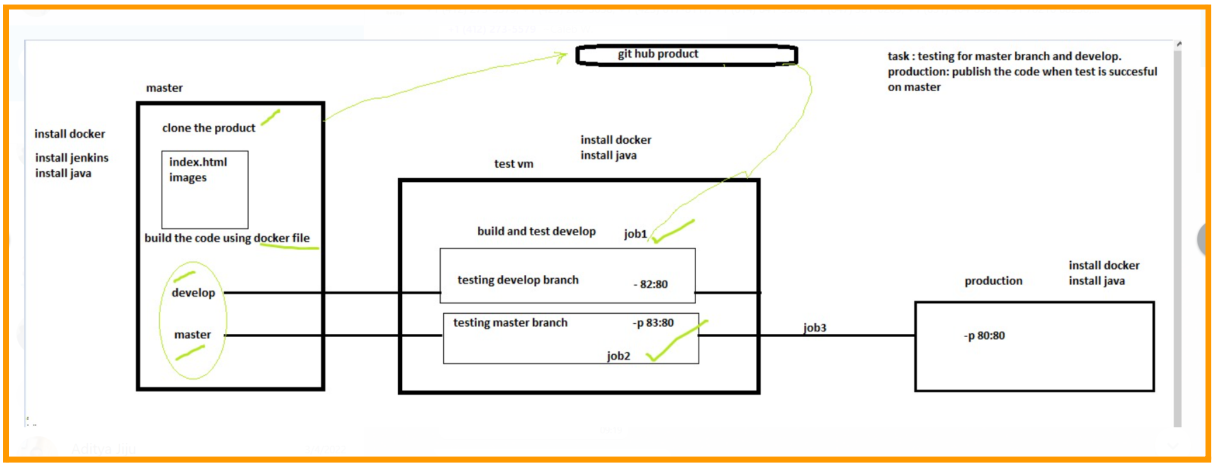
If apache is not installed => "Apache is not installed on this System"

<u>Project Workflow</u> --

Create 3 servers on AWS "t2.micro"

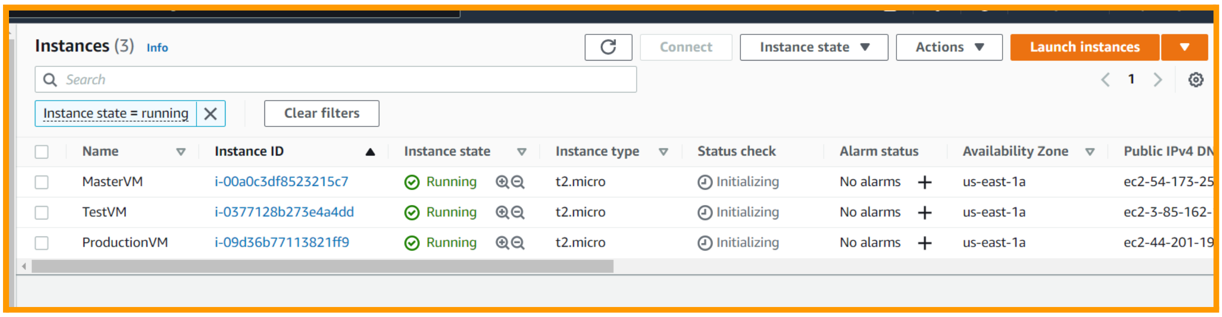
Server 1 - should have Jenkins Master, Puppet Master and Nagios Installed
Server 2 - Testing Server, Jenkins Slave

Server 3 - Prod Server, Jenkins Slave



task : testing for master branch and develop.
production: publish the code when test is succesful on master

Now 3 VMs are required – Master, Test and Production using t2.micro



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DN
□	MasterVM	i-00a0c3df8523215c7	Running	t2.micro	Initializing	No alarms +	us-east-1a	ec2-54-173-25
□	TestVM	i-0377128b273e4a4dd	Running	t2.micro	Initializing	No alarms +	us-east-1a	ec2-3-85-162-
□	ProductionVM	i-09d36b77113821ff9	Running	t2.micro	Initializing	No alarms +	us-east-1a	ec2-44-201-19

On Master – install Jenkins

Install docker

Install java

On Test and Production Vm –

Install docker

Install java

Commands-

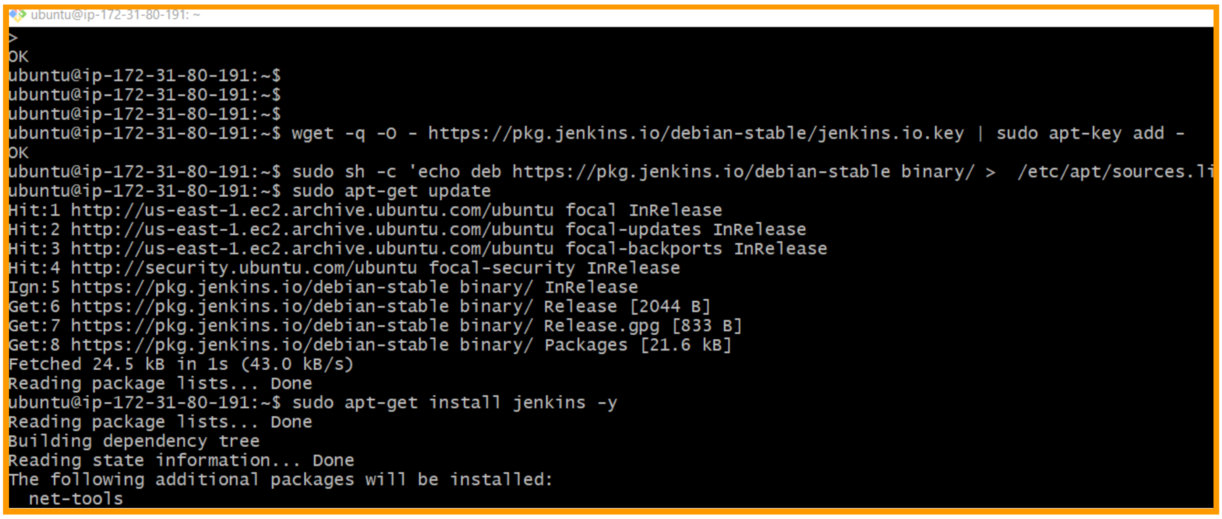
On master-

```
sudo apt-get update
sudo apt install ca-certificates
sudo apt-get install openjdk-11-jdk -y
wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins -y
sudo apt-get install docker.io
```

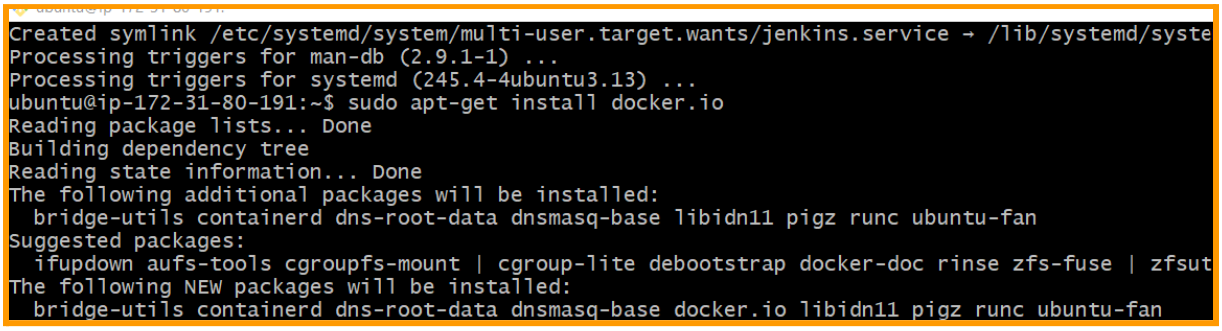
On Test & Production-

```
sudo apt-get update
sudo apt-get install openjdk-11-jdk -y
sudo apt-get install docker.io
```

On master

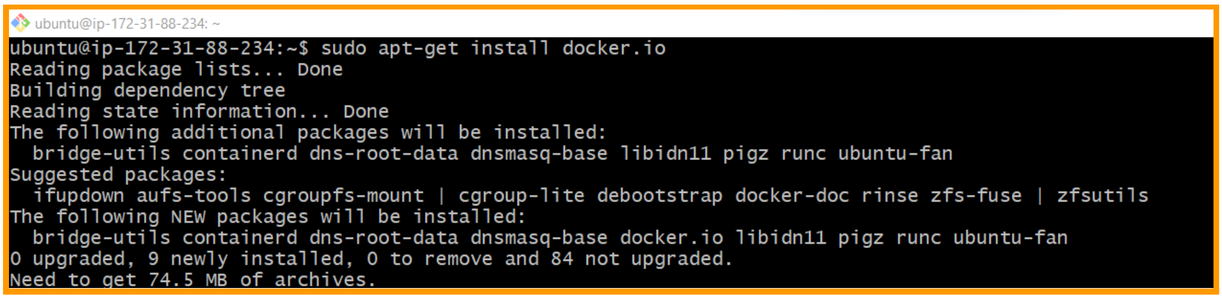


```
>
OK
ubuntu@ip-172-31-80-191:~$
ubuntu@ip-172-31-80-191:~$
ubuntu@ip-172-31-80-191:~$
ubuntu@ip-172-31-80-191:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -
OK
ubuntu@ip-172-31-80-191:~$ sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >  /etc/apt/sources.li
ubuntu@ip-172-31-80-191:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [21.6 kB]
Fetched 24.5 kB in 1s (43.0 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-80-191:~$ sudo apt-get install jenkins -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  net-tools
```



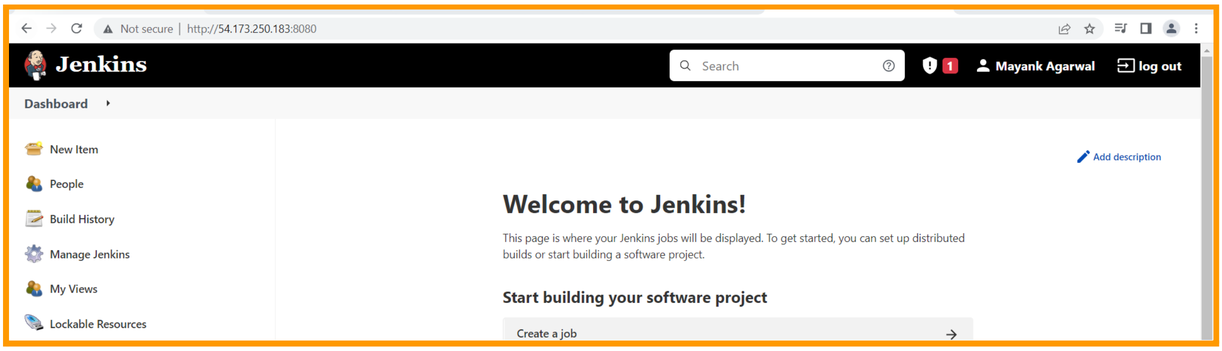
```
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/syste
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for systemd (245.4-4ubuntu3.13) ...
ubuntu@ip-172-31-80-191:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsut
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
```

On Test & Production-



```
ubuntu@ip-172-31-88-234:~$ sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 84 not upgraded.
Need to get 74.5 MB of archives.
```

Now Configure Jenkins using 8080 and adding my 2 nodes Test & Production in Master –



Welcome to Jenkins!

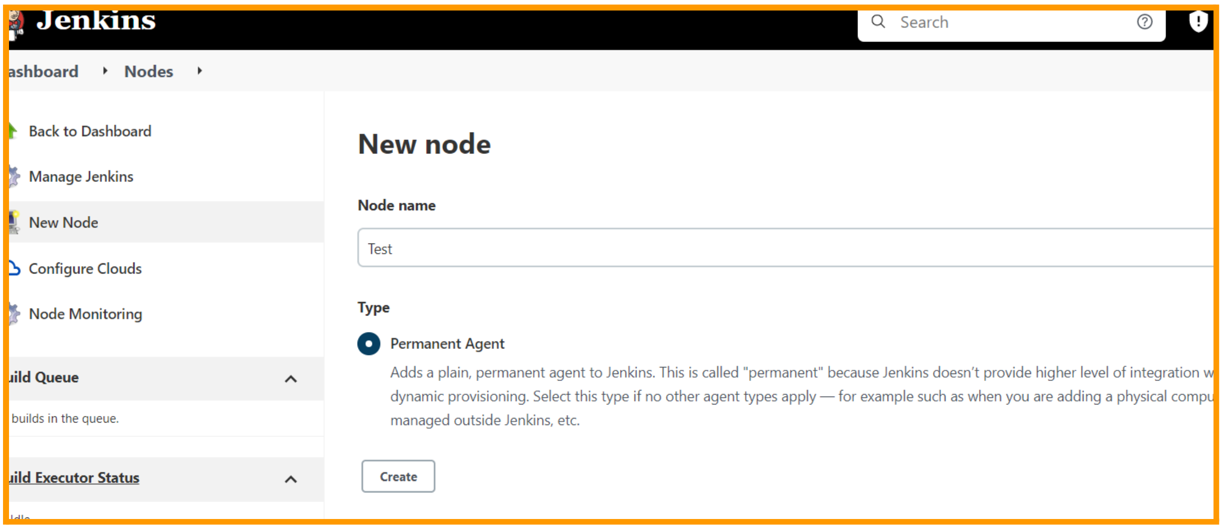
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Before add node we have to do changes in Jenkins dashboard –

GoTo Manage Jenkins --- > Configure Global Security - Scroll down Agent - Select Random Apply & Save.

Now Add node GoTo Manage Node -- . New Node



New node

Node name: Test

Type

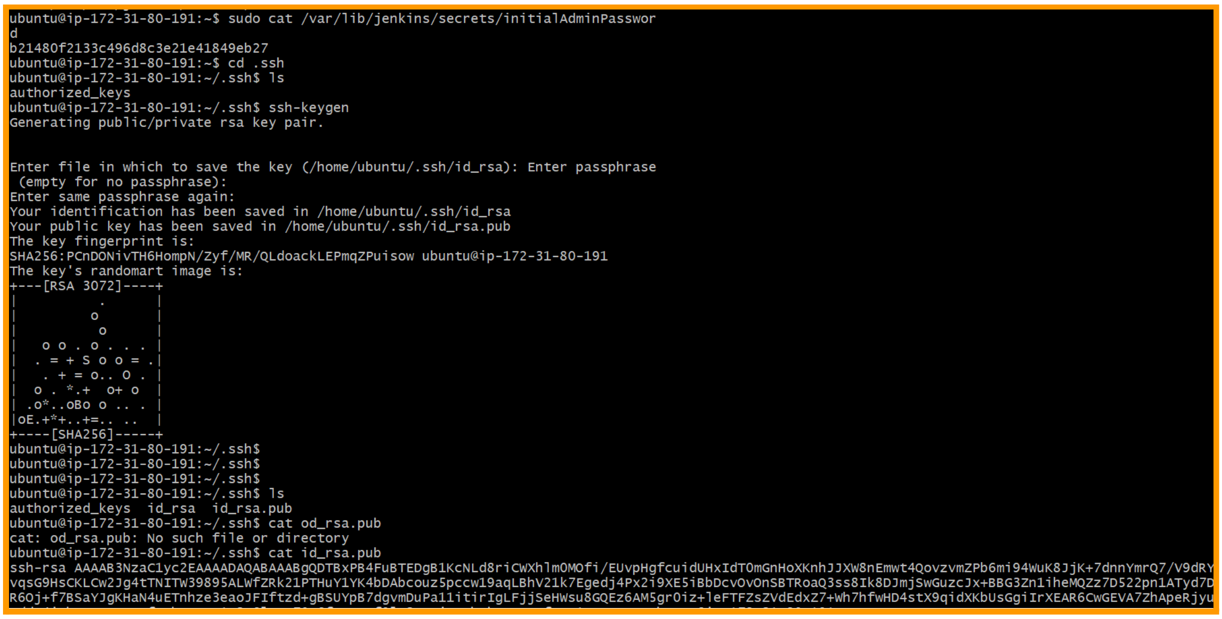
● Permanent Agent

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

Create private_key for Jenkins

Go to master

cd .ssh ls - ssh-keygen -- enter enter enter- ls - cat id_rsa.pub



```
ubuntu@ip-172-31-80-191:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPasswor
d
b21480f2133c496d8c3e21e41849eb27
ubuntu@ip-172-31-80-191:~$ cd .ssh
ubuntu@ip-172-31-80-191:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-80-191:~/.ssh$ ssh-keygen
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:PCnDONivTH6HompN/Zyf/MR/QLdoackLEPmqZPuisow ubuntu@ip-172-31-80-191
The key's randomart image is:
+---[RSA 3072]----+
|        .        |
|       o         |
|        o        |
|   o o . o . . . |
|  . = + S o o = .|
|   . + = o.. O . |
|  o . *.+  o+ o  |
| .o*..oBo o .. . |
|oE.+*+..+=.. ..  |
+----[SHA256]-----+
ubuntu@ip-172-31-80-191:~/.ssh$
ubuntu@ip-172-31-80-191:~/.ssh$
ubuntu@ip-172-31-80-191:~/.ssh$
ubuntu@ip-172-31-80-191:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub
ubuntu@ip-172-31-80-191:~/.ssh$ cat od_rsa.pub
cat: od_rsa.pub: No such file or directory
ubuntu@ip-172-31-80-191:~/.ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDTBxPB4FuBTEDgB1KcNLd8riCWXhlm0MOfi/EUvpHgfcuidUHxIdTOmGnHoXKnhJJXW8nEmwt4QovzvmZPb6mi94WuK8JjK+7dnnYmrQ7/V9dRY
vqsG9HsCKLCw2Jg4tTNITW39895ALWfZRk21PTHuY1YK4bDAbcouz5pccw19aqLBhV21k7Egedj4Px2i9XE5iBbDcvOvOnSBTRoaQ3ss8Ik8DJmjSwGuzcJx+BBG3Zn1iheMQZz7D522pn1ATyd7D
R6Oj+f7BSaYJgKHaN4uETnhze3eaoJFIftzd+gBSUYpB7dgvmDuPa11itirIgLFjjSeHWsu8GQEz6AM5gr0iz+leFTFZsZVdEdxZ7+Wh7hfwHD4stX9qidXKbUsGgiIrXEAR6CwGEVA7ZhApeRjyu
```

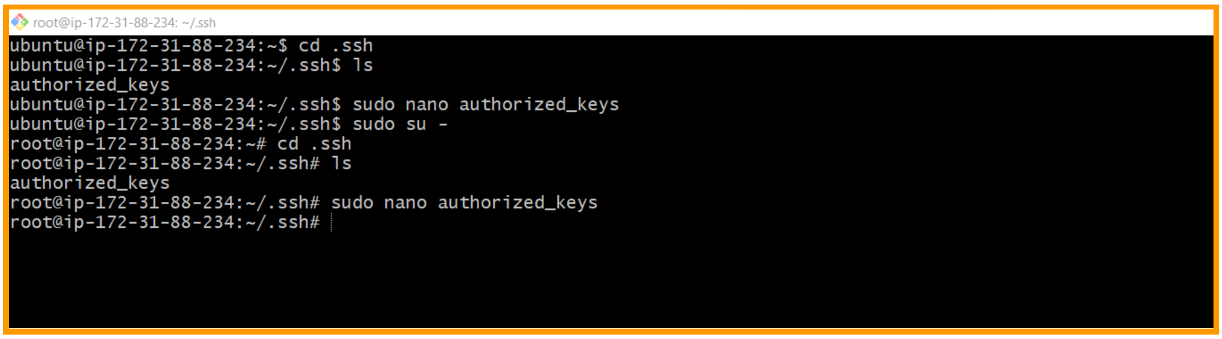
GoTo Node paste this key

cd .ssh -- > ls -- sudo nano authorized_keys

now

sudo su –

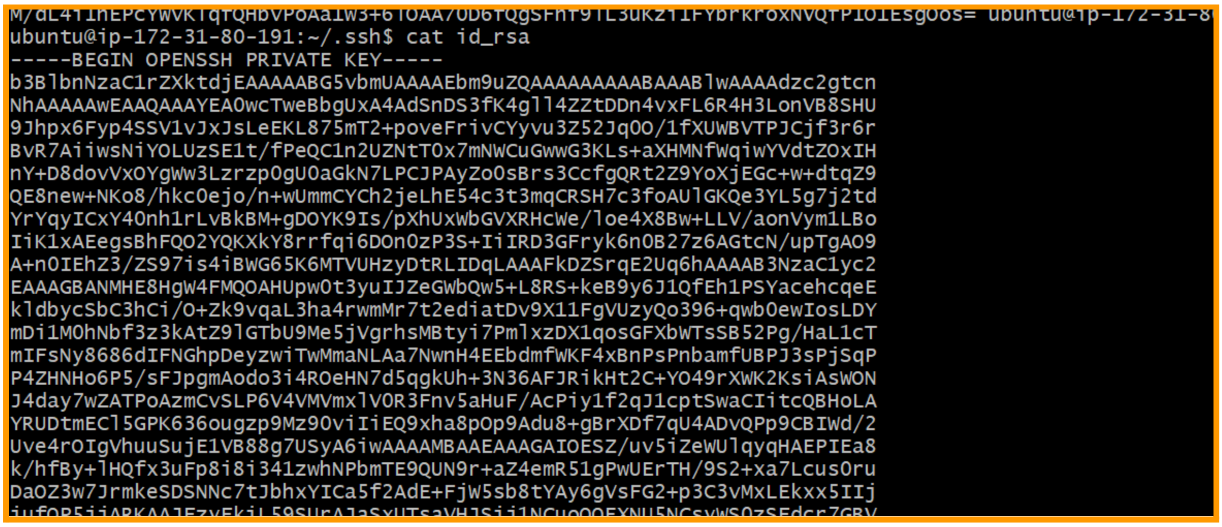
cd .ssh -- > ls -- sudo nano authorized_keys



```
ubuntu@ip-172-31-88-234:~$ cd .ssh
ubuntu@ip-172-31-88-234:~/.ssh$ ls
authorized_keys
ubuntu@ip-172-31-88-234:~/.ssh$ sudo nano authorized_keys
ubuntu@ip-172-31-88-234:~/.ssh$ sudo su -
root@ip-172-31-88-234:~# cd .ssh
root@ip-172-31-88-234:~/.ssh# ls
authorized_keys
root@ip-172-31-88-234:~/.ssh# sudo nano authorized_keys
root@ip-172-31-88-234:~/.ssh#
```

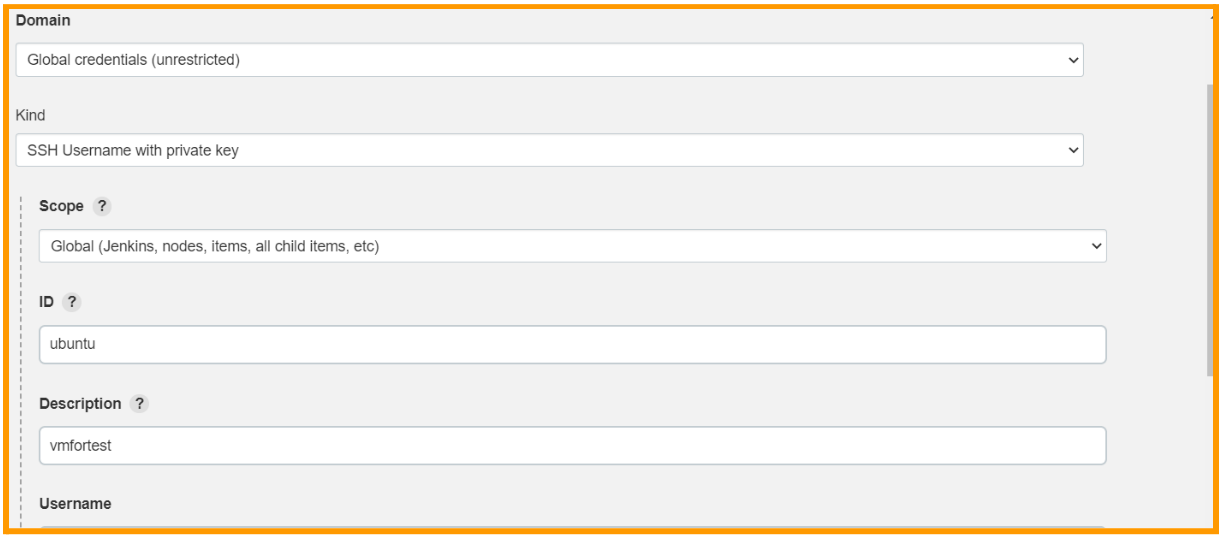
Now for add credential in Jenkins node

Cat id_rsa



```
M/dL4i1nEPcYWvKIqfQHbVPoAa1W3+6TOAA/OD6fQgSFnf9TL3uKZ1IFYbrKroxNVQfPIOIEsgoos= ubuntu@ip-172-31-8
ubuntu@ip-172-31-80-191:~/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEA0wcTweBbgUxA4AdSnDS3fK4gll4ZZtDDn4vxFL6R4H3LonVB8SHU
9Jhpx6Fyp4SSV1vJxJsLeEKL875mT2+poveFrivCYyvu3Z52JqOO/1fXUWBVTPJCjf3r6r
BvR7AiiwsNiYOLUzSE1t/fPeQC1n2UZNtTOx7mNWCuGwwG3KLs+aXHMNfWqiwYVdtZOxIH
nY+D8dovVxOYgWw3LzrzpOgU0aGkN7LPCJPAyZoOsBrs3CcfgQRt2Z9YoXjEGc+w+dtqZ9
QE8new+NKo8/hkc0ejo/n+wUmmCYCh2jeLhE54c3t3mqCRSH7c3foAUlGKQe3YL5g7j2td
YrYqyICxY40nh1rLvBkBM+gDOYK9Is/pXhUxWbGVXRHcWe/loe4X8Bw+LLV/aonVym1LBo
IiK1xAEegsBhFQO2YQKXkY8rrfqi6DOn0zP3S+IiIRD3GFryk6n0B27z6AGtcN/upTgAO9
A+nOIEhZ3/ZS97is4iBWG65K6MTVUHzyDtRLIDqLAAAFkDZSrqE2Uq6hAAAAB3NzaC1yc2
EAAAGBANMHE8HgW4FMQOAHUpw0t3yuIJZeGWbQw5+L8RS+keB9y6J1QfEh1PSYacehcqeE
kldbycSbC3hCi/O+Zk9vqaL3ha4rwmMr7t2ediatDv9X11FgVUzyQo396+qwb0ewIosLDY
mDi1MOhNbf3z3kAtZ9lGTbU9Me5jVgrhsMBtyi7PmlxzDX1qosGFXbWTsSB52Pg/HaL1cT
mIFsNy8686dIFNGhpDeyzwiTwMmaNLAa7NwnH4EEbdmfWKF4xBnPsPnbamfUBPJ3sPjSqP
P4ZHNHo6P5/sFJpgmAodo3i4ROeHN7d5qgkUh+3N36AFJRikHt2C+YO49rXWK2KsiAsWON
J4day7wZATPoAzmCvSLP6V4VMVmxlVOR3Fnv5aHuF/AcPiy1f2qJ1cptSwaCIitcQBHoLA
YRUDtmECl5GPK636ougzp9Mz90viIiEQ9xha8pOp9Adu8+gBrXDf7qU4ADvQPp9CBIWd/2
Uve4rOIgVhuuSujE1VB88g7USyA6iwAAAAMBAAEAAAGAIOESZ/uv5iZeWUlqyqHAEPIEa8
k/hfBy+lHQfx3uFp8i8i341zwhNPbmTE9QUN9r+aZ4emR51gPwUErTH/9S2+xa7LcusOru
DaOZ3w7JrmkeSDSNNc7tJbhxYICa5f2AdE+FjW5sb8tYAy6gVsFG2+p3C3vMxLEkxx5IIj
```

Copy and paste it



Domain

Global credentials (unrestricted)

Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

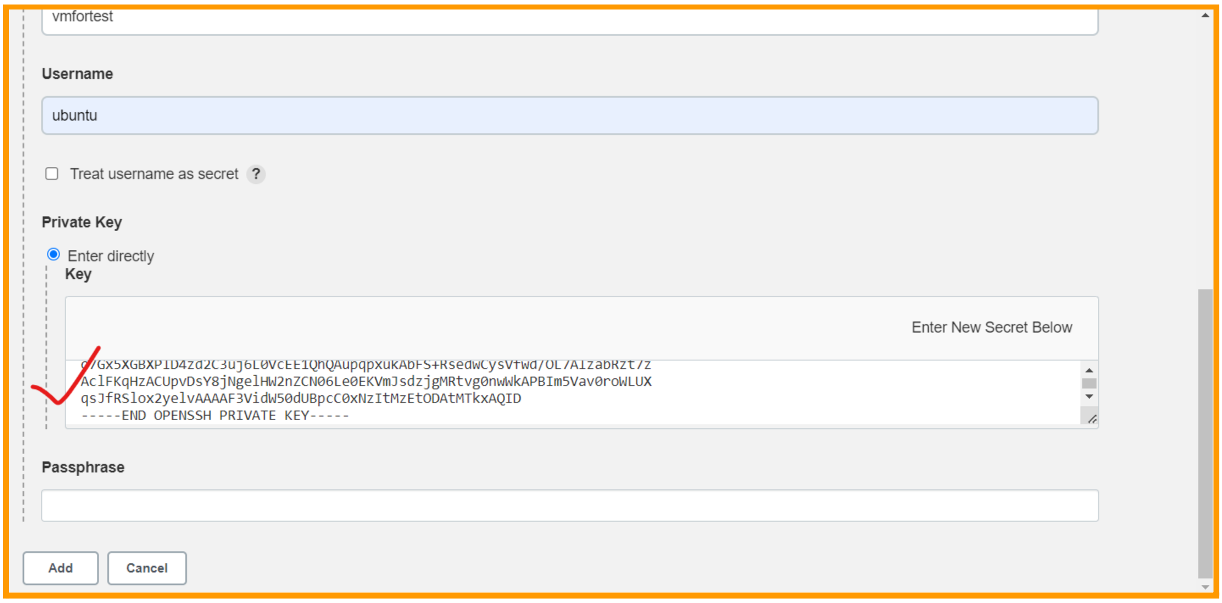
ID ?

ubuntu

Description ?

vmfortest

Username



vmfortest

Username

ubuntu

□ Treat username as secret ?

Private Key

● Enter directly

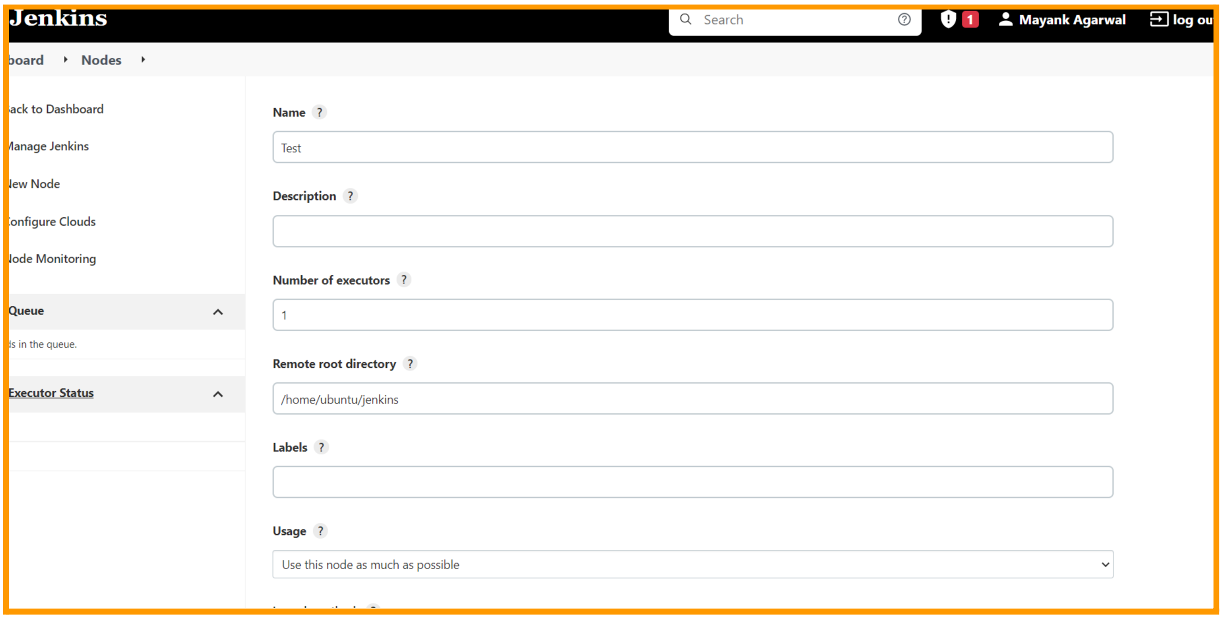
Key

Enter New Secret Below

```
d/Gx5XGBXPlD4zd2C3uj6L0VcEE1QhQAupqpxukAbFS+RsedwCysVfwd/OL7AlzabRzt7z
AclFKqHzACUpvDsY8jNgelHW2nZCN06Le0EKVmJsdzjgMRtvg0nwWkAPBIm5Vav0roWLUX
qsJfRSlox2yelvAAAAF3VidW50dUBpcC0xNzItMzEtODAtMTkxAQID
-----END OPENSSH PRIVATE KEY-----
```

Passphrase

Test Node



ls in the queue.

Name ?

Test

Description ?

Number of executors ?

1

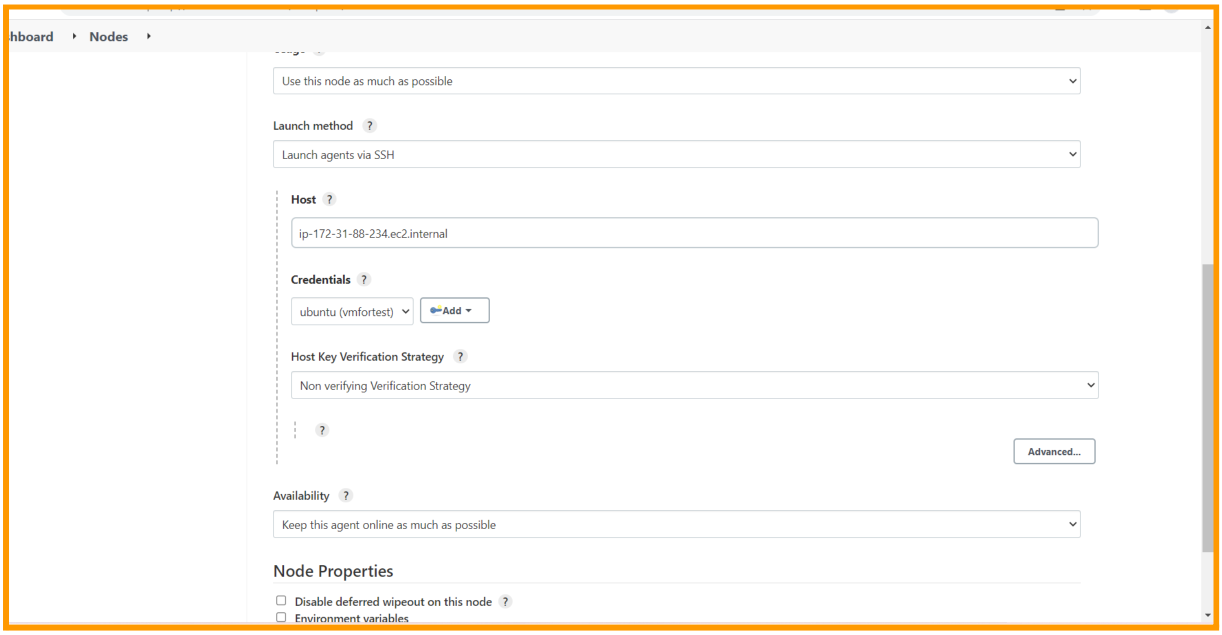
Remote root directory ?

/home/ubuntu/jenkins

Labels ?

Usage ?

Use this node as much as possible



Use this node as much as possible

Launch method ?
Launch agents via SSH

Host ?
ip-172-31-88-234.ec2.internal

Credentials ?
ubuntu (vmfortest)

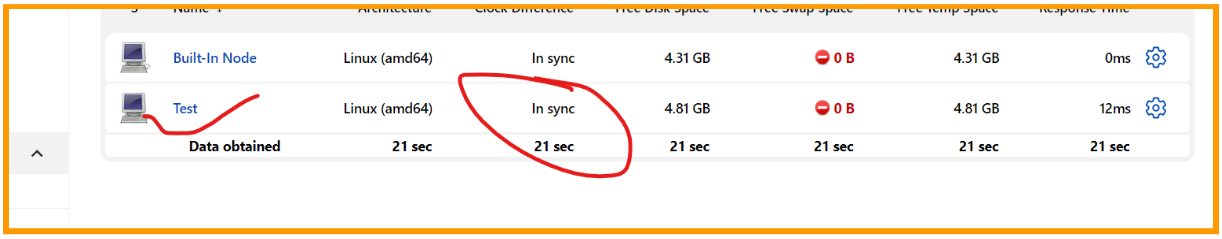
Host Key Verification Strategy ?
Non verifying Verification Strategy

Availability ?
Keep this agent online as much as possible

Node Properties

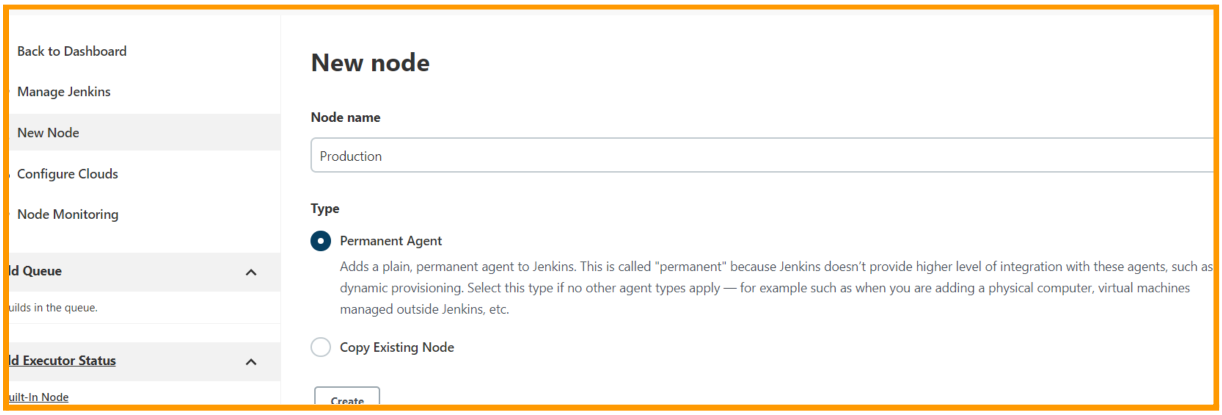
□ Disable deferred wipeout on this node ?
□ Environment variables

Save



S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	Built-In Node	Linux (amd64)	In sync	4.31 GB	⛔ 0 B	4.31 GB	0ms	⚙
	Test	Linux (amd64)	In sync	4.81 GB	⛔ 0 B	4.81 GB	12ms	⚙
	Data obtained	21 sec	21 sec	21 sec	21 sec	21 sec	21 sec	

Now Add New node for Production with same step



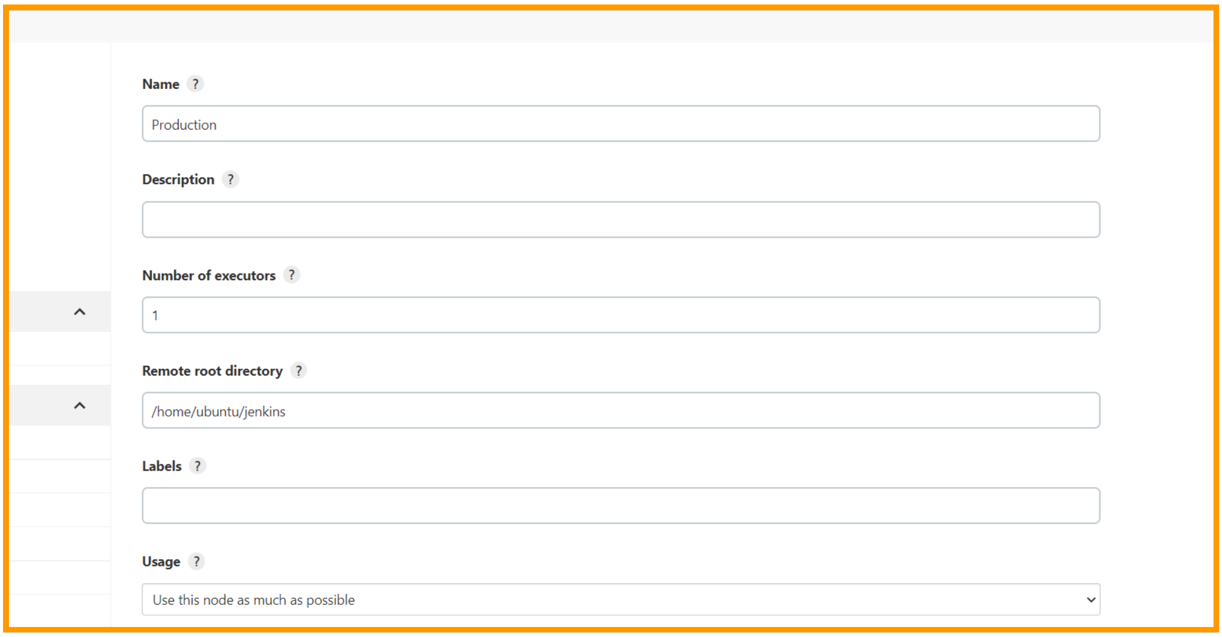
New node

Node name
Production

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



Name ?

Production

Description ?

Number of executors ?

1

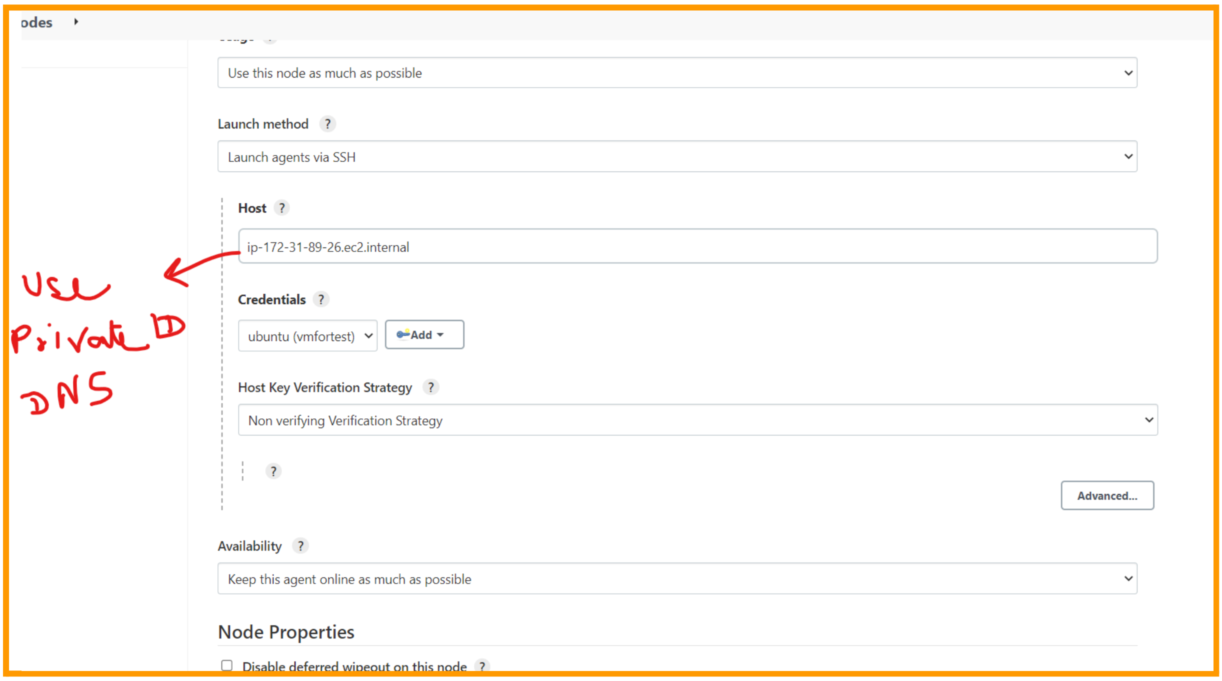
Remote root directory ?

/home/ubuntu/jenkins

Labels ?

Usage ?

Use this node as much as possible



Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

ip-172-31-89-26.ec2.internal

Use Private ID DNS

Credentials ?

ubuntu (vmfortest)

Host Key Verification Strategy ?

Non verifying Verification Strategy

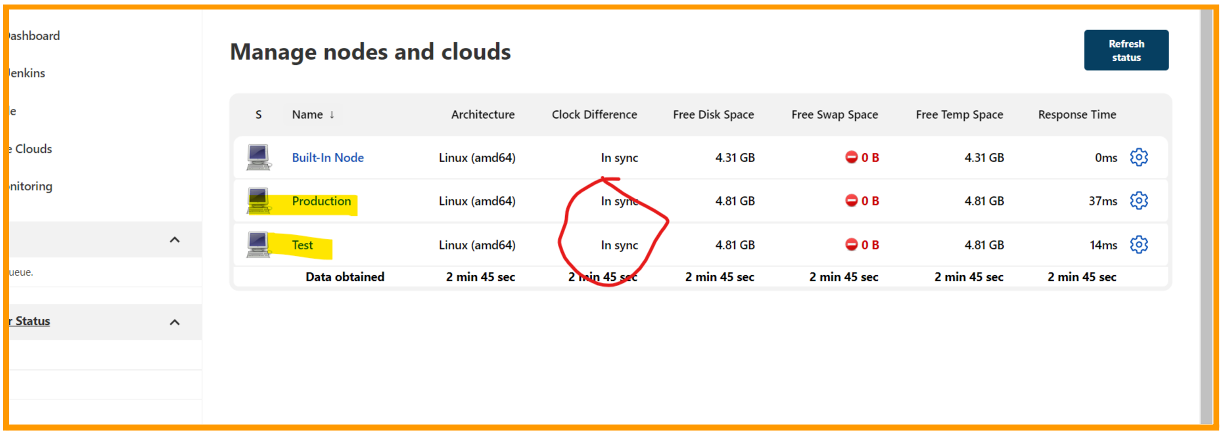
Availability ?

Keep this agent online as much as possible

Node Properties

□ Disable deferred wipeout on this node ?

Save



Manage nodes and clouds

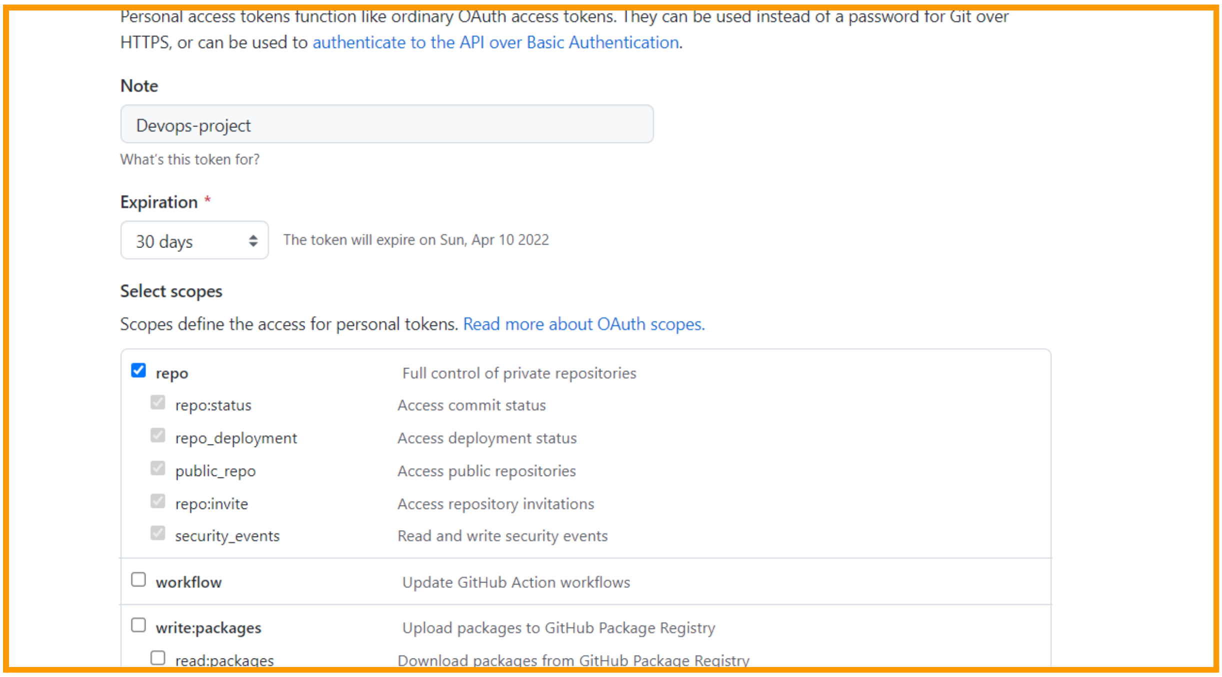
S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	Built-In Node	Linux (amd64)	In sync	4.31 GB	0 B	4.31 GB	0ms	
	Production	Linux (amd64)	In sync	4.81 GB	0 B	4.81 GB	37ms	
	Test	Linux (amd64)	In sync	4.81 GB	0 B	4.81 GB	14ms	
	Data obtained	2 min 45 sec	2 min 45 sec	2 min 45 sec	2 min 45 sec	2 min 45 sec	2 min 45 sec	

Now goTo github link –

https://github.com/Aayushinam/website --- > fork this website to our github

Configure GitHub with our Master so we can directly push to our GitHub--

GoTo Setting -- > Developer Setting --- > Personal Access Token --- > Generate New Token enter name and check on repo Generate Token - copy password


Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

Devops-project

What's this token for?

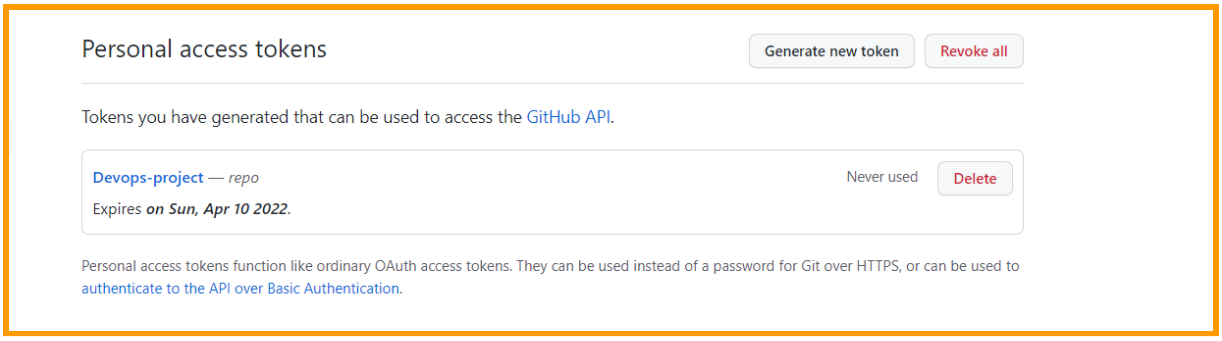
Expiration *

30 days The token will expire on Sun, Apr 10 2022

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

☑ repo	Full control of private repositories
☑ repo:status	Access commit status
☑ repo_deployment	Access deployment status
☑ public_repo	Access public repositories
☑ repo:invite	Access repository invitations
☑ security_events	Read and write security events
□ workflow	Update GitHub Action workflows
□ write:packages	Upload packages to GitHub Package Registry
□ read:packages	Download packages from GitHub Package Registry



Personal access tokens

Tokens you have generated that can be used to access the GitHub API.

Devops-project — *repo*

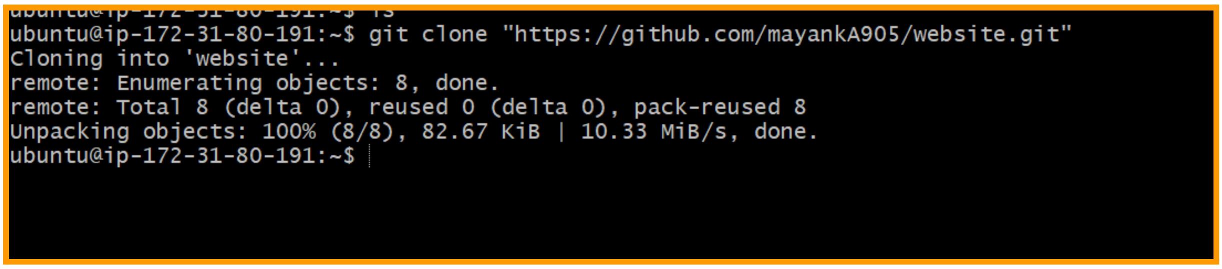
Never used

Expires *on Sun, Apr 10 2022*.

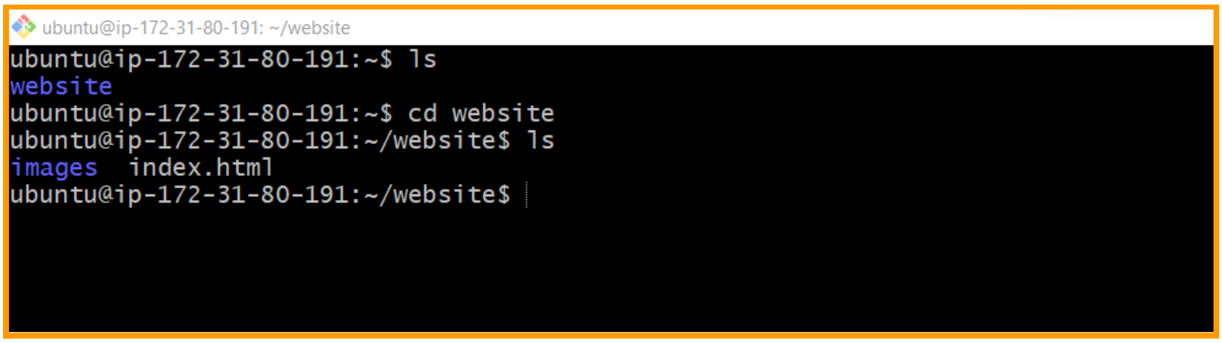
Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Now clone our repository in our master

Git clone " "



```
ubuntu@ip-172-31-80-191:~$ ls
ubuntu@ip-172-31-80-191:~$ git clone "https://github.com/mayankA905/website.git"
Cloning into 'website'...
remote: Enumerating objects: 8, done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 8
Unpacking objects: 100% (8/8), 82.67 KiB | 10.33 MiB/s, done.
ubuntu@ip-172-31-80-191:~$
```

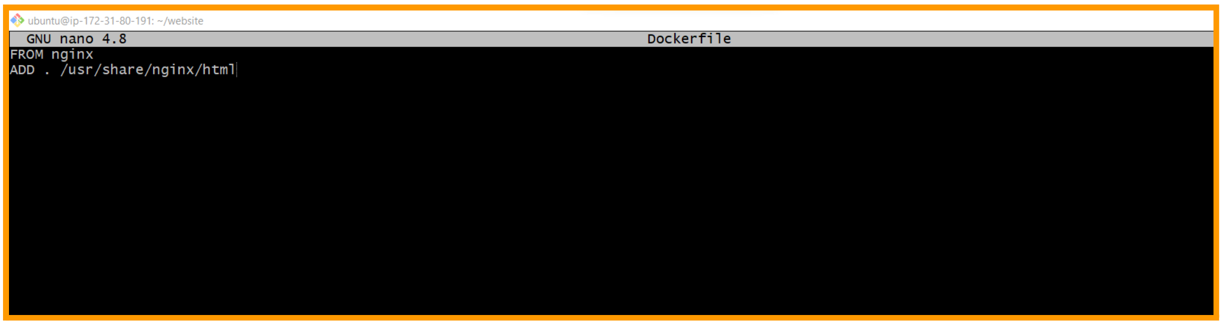


```
ubuntu@ip-172-31-80-191: ~/website
ubuntu@ip-172-31-80-191:~$ ls
website
ubuntu@ip-172-31-80-191:~$ cd website
ubuntu@ip-172-31-80-191:~/website$ ls
images  index.html
ubuntu@ip-172-31-80-191:~/website$
```

Now create and Build DockerFile

sudo nano Dockerfile

FROM nginx
ADD . /usr/share/nginx/html

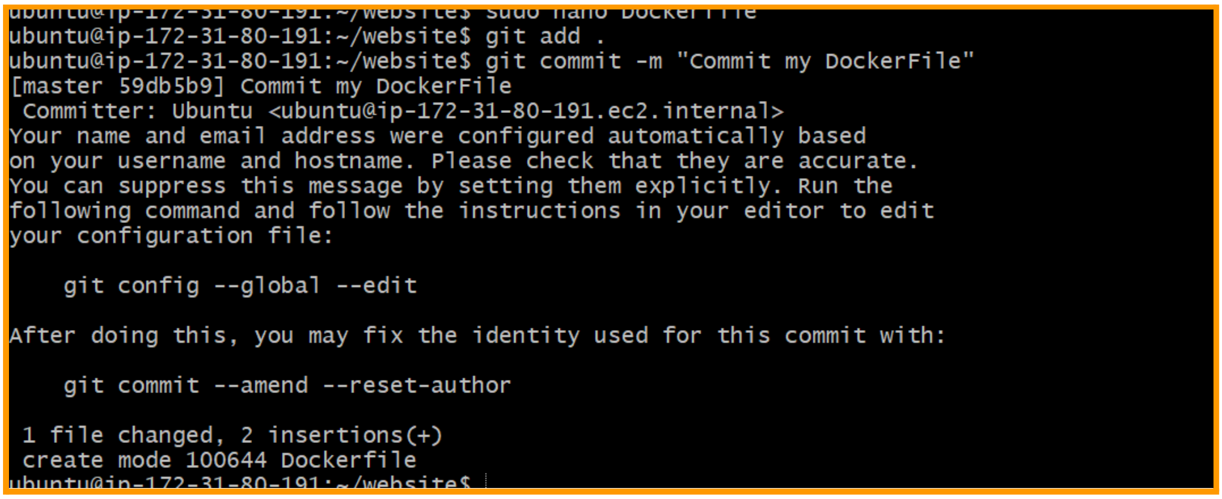


```
ubuntu@ip-172-31-80-191: ~/website
  GNU nano 4.8                                   Dockerfile
FROM nginx
ADD . /usr/share/nginx/html
```

Now

git add .

git commit -m “commit my Dockerfile”

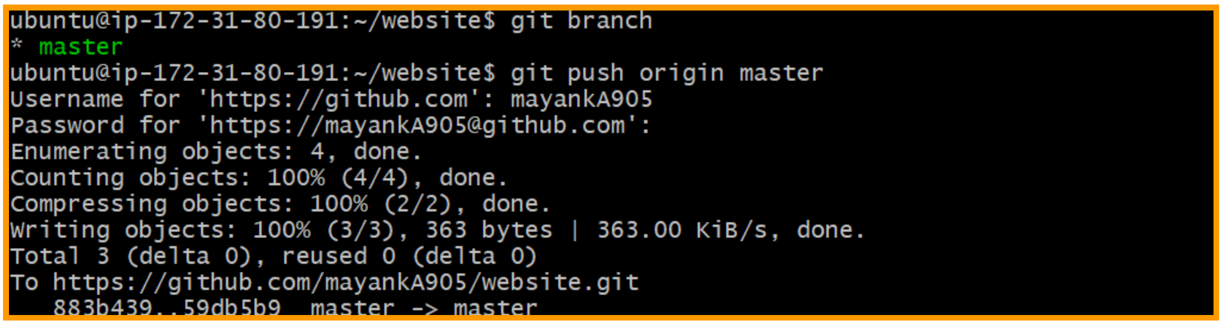


```
ubuntu@ip-172-31-80-191:~/website$ sudo nano Dockerfile
ubuntu@ip-172-31-80-191:~/website$ git add .
ubuntu@ip-172-31-80-191:~/website$ git commit -m "Commit my DockerFile"
[master 59db5b9] Commit my DockerFile
 Committer: Ubuntu <ubuntu@ip-172-31-80-191.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 2 insertions(+)
 create mode 100644 Dockerfile
ubuntu@ip-172-31-80-191:~/website$
```

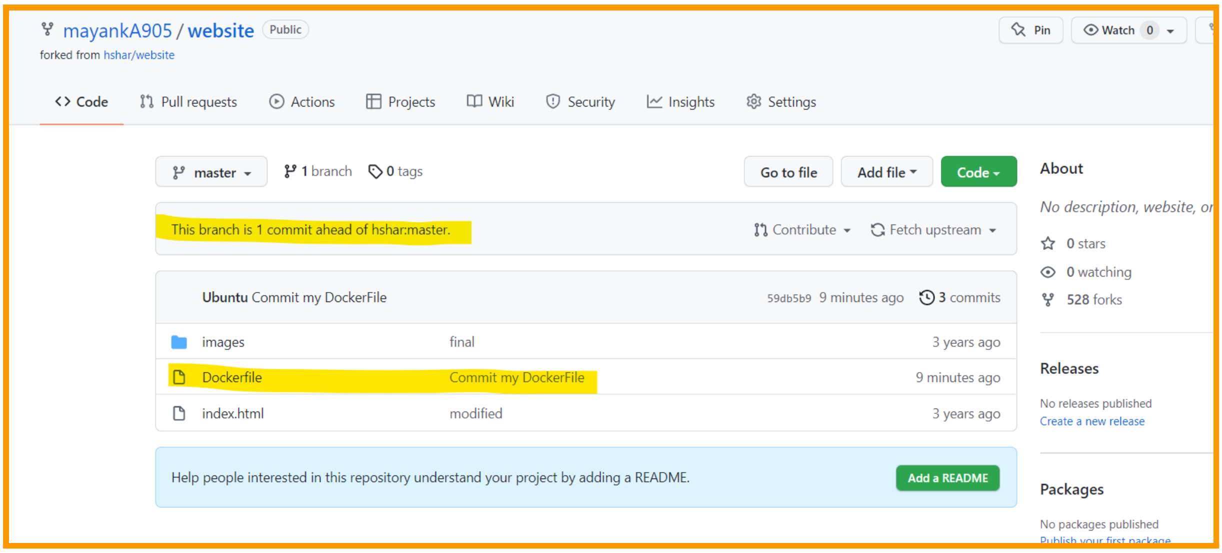
Push this to remote repo

git branch

git push origin master - enter Github username - enter personal access token password



```
ubuntu@ip-172-31-80-191:~/website$ git branch
* master
ubuntu@ip-172-31-80-191:~/website$ git push origin master
Username for 'https://github.com': mayankA905
Password for 'https://mayankA905@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 363 bytes | 363.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/mayankA905/website.git
   883b439..59db5b9  master -> master
```



mayankA905 / website Public

forked from hshar/website

images	final	3 years ago
Dockerfile	Commit my DockerFile	9 minutes ago
index.html	modified	3 years ago

This branch is 1 commit ahead of hshar:master.

Ubuntu Commit my DockerFile 59db5b9 9 minutes ago 3 commits

Help people interested in this repository understand your project by adding a README.

About

No description, website, or

0 stars

0 watching

528 forks

Releases

No releases published

Create a new release

Packages

No packages published

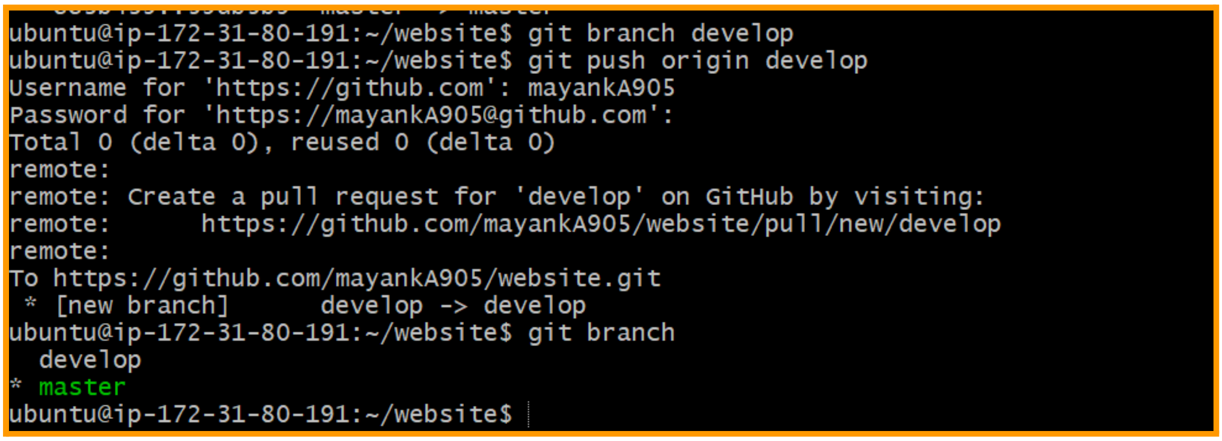
Publish your first package

Now create develop branch

git branch develop

git push origin develop

git branch



```
   883b439..59db5b9  master -> master
ubuntu@ip-172-31-80-191:~/website$ git branch develop
ubuntu@ip-172-31-80-191:~/website$ git push origin develop
Username for 'https://github.com': mayankA905
Password for 'https://mayankA905@github.com':
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'develop' on GitHub by visiting:
remote:      https://github.com/mayankA905/website/pull/new/develop
remote:
To https://github.com/mayankA905/website.git
 * [new branch]      develop -> develop
ubuntu@ip-172-31-80-191:~/website$ git branch
  develop
* master
ubuntu@ip-172-31-80-191:~/website$
```

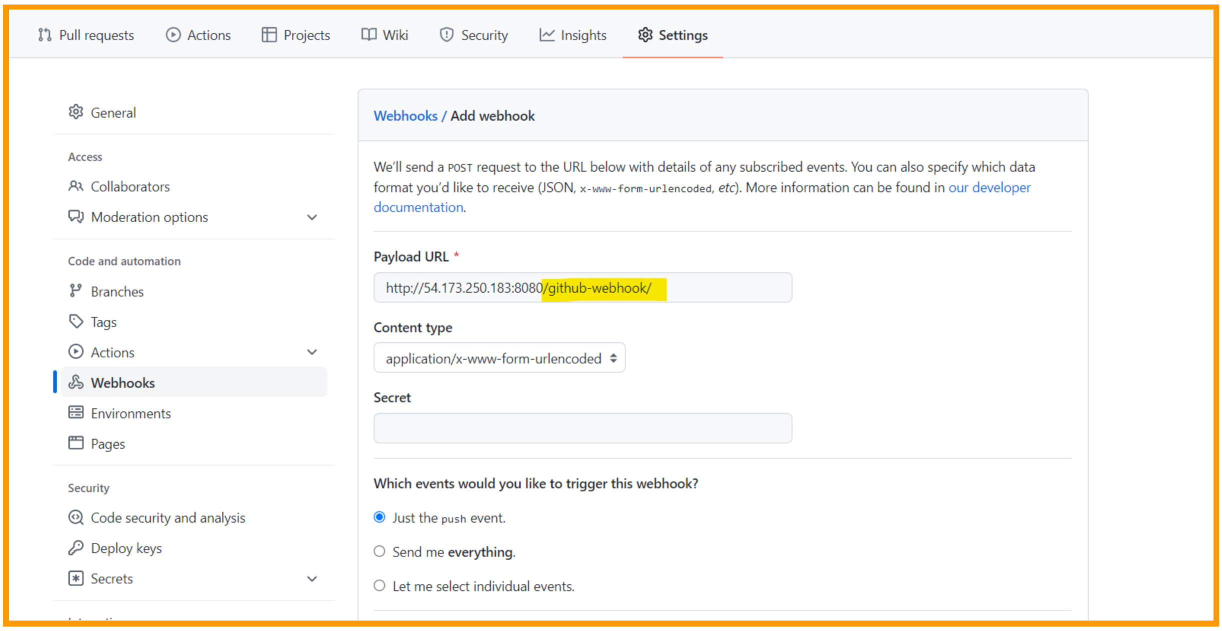
Now we need to create our jobs in Jenkins

Dashboard- new item job1 freestyle project - save

Create WebHook first

Copy Jenkins URL upto 8080

Goto setting WebHooks Add Webhook Copy URL and type /github-webhook/ 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://54.173.250.183:8080/github-webhook/

Content type

application/x-www-form-urlencoded

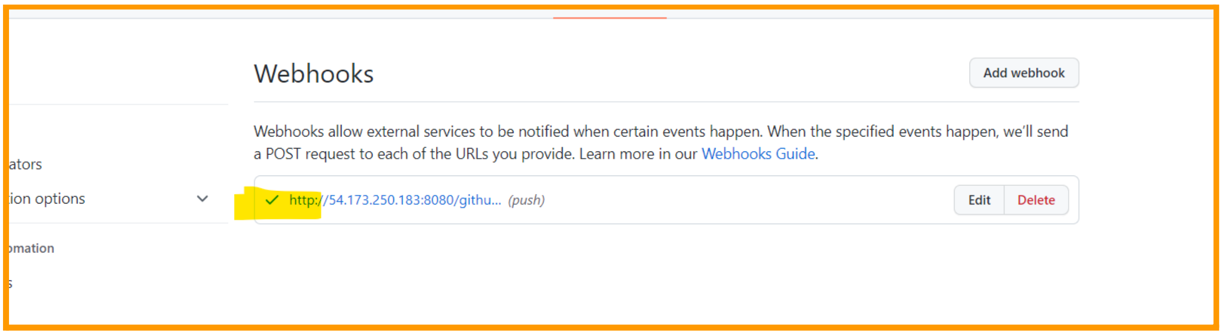
Secret

Which events would you like to trigger this webhook?

● Just the push event.

○ Send me **everything**.

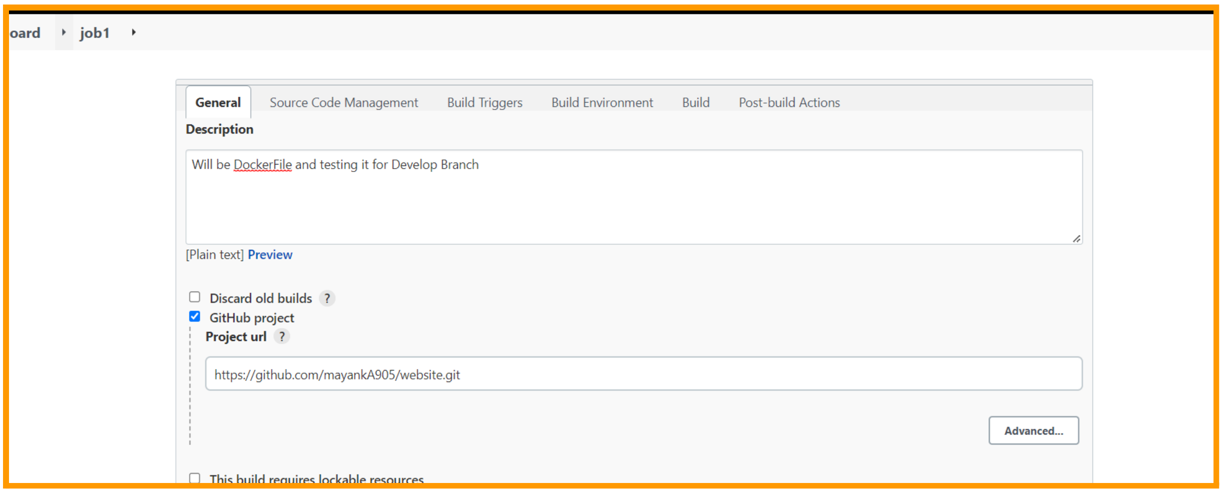
○ Let me select individual events.



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://54.173.250.183:8080/githu... *(push)*



Description

Will be DockerFile and testing it for Develop Branch

[Plain text] Preview

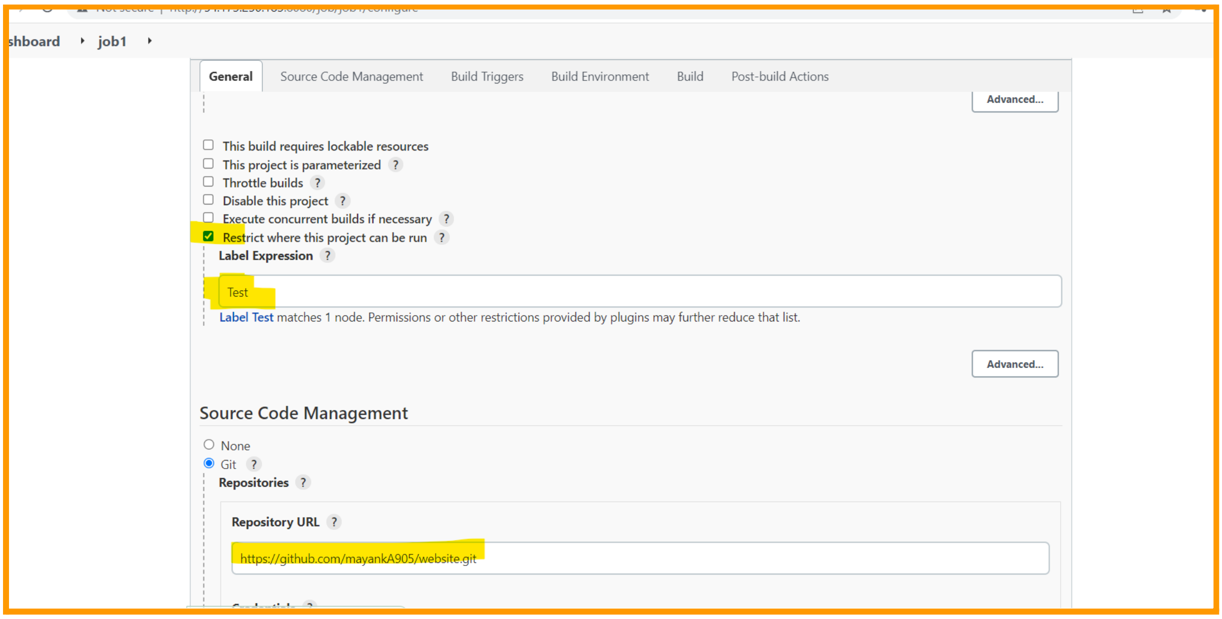
□ Discard old builds ?

☑ GitHub project

Project url ?

https://github.com/mayankA905/website.git

□ This build requires lockable resources



□ This build requires lockable resources

□ This project is parameterized ?

□ Throttle builds ?

□ Disable this project ?

□ 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

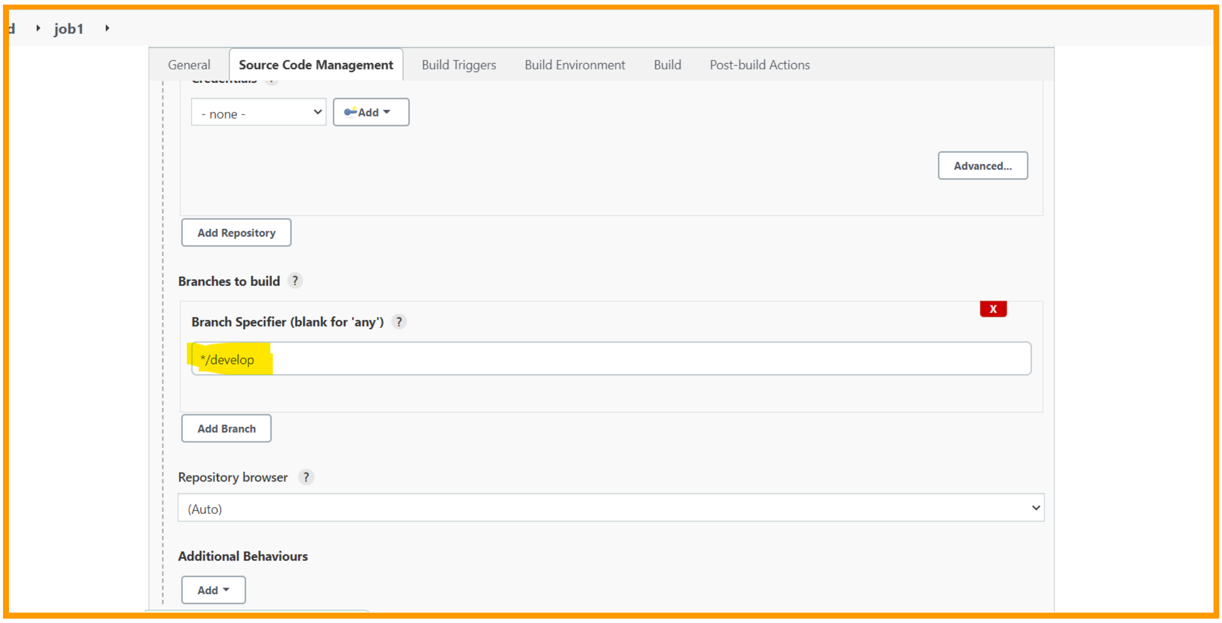
○ None

● Git ?

Repositories ?

Repository URL ?

https://github.com/mayankA905/website.git



- none -

Branches to build ?

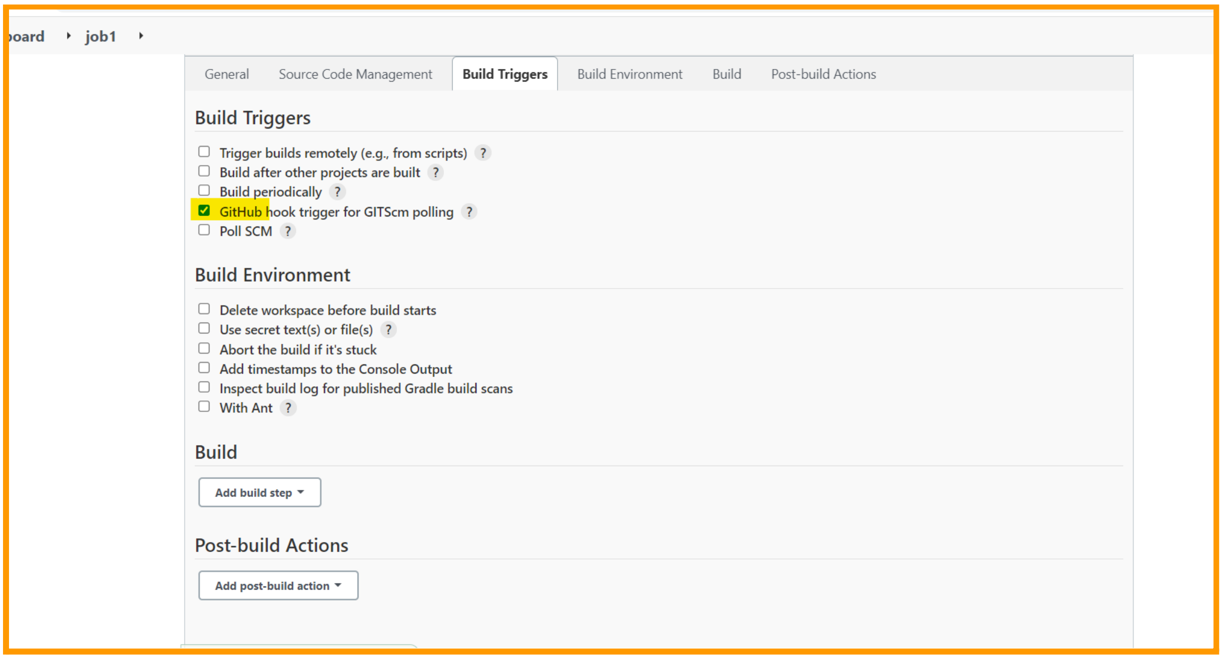
Branch Specifier (blank for 'any') ?

*/develop

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 ?

Build Environment

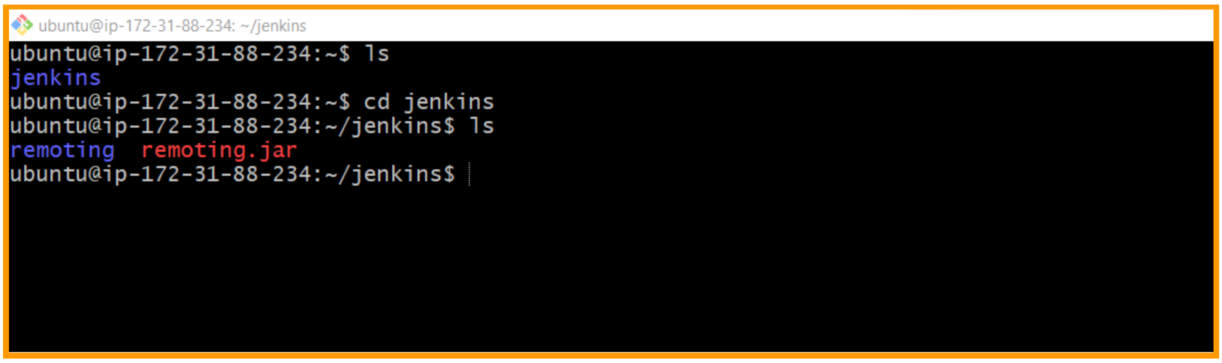
□ Delete workspace before build starts
□ Use secret text(s) or file(s) ?
□ Abort the build if it's stuck
□ Add timestamps to the Console Output
□ Inspect build log for published Gradle build scans
□ With Ant ?

Build

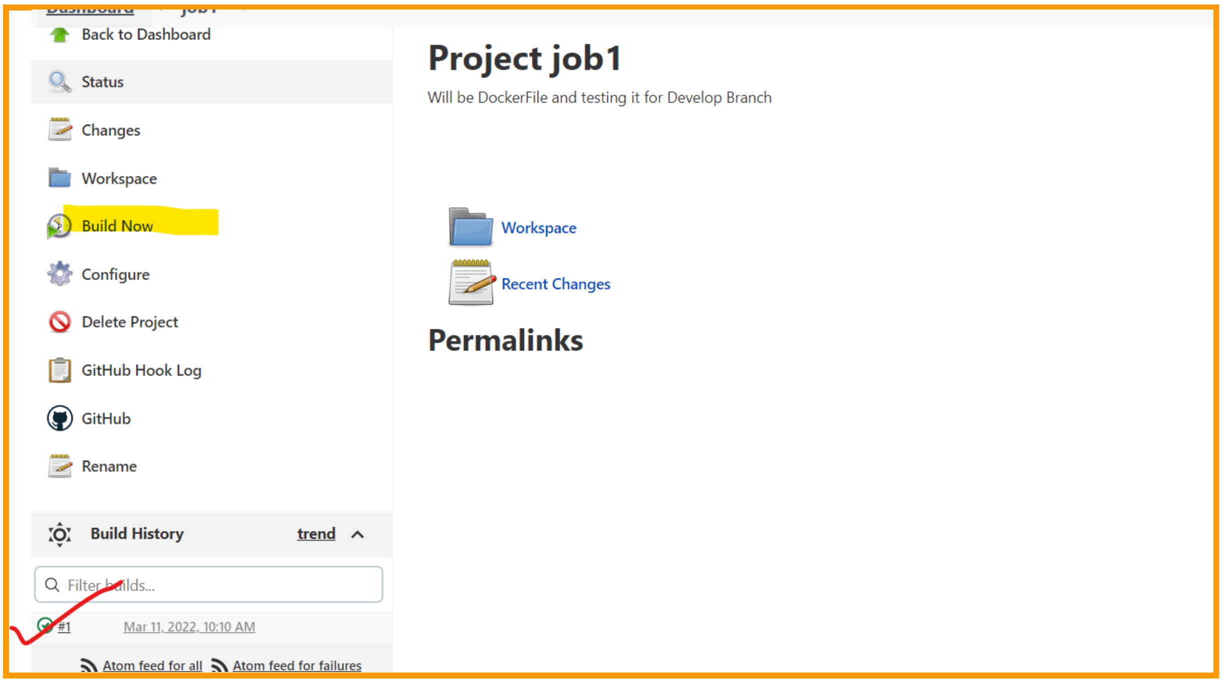
Post-build Actions

Apply and Save

In Test Vm—create workspace by run the job



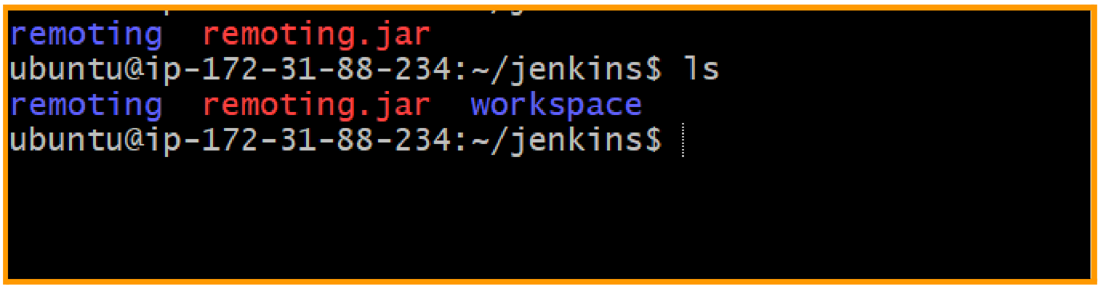
```
ubuntu@ip-172-31-88-234: ~/jenkins
ubuntu@ip-172-31-88-234:~$ ls
jenkins
ubuntu@ip-172-31-88-234:~$ cd jenkins
ubuntu@ip-172-31-88-234:~/jenkins$ ls
remoting  remoting.jar
ubuntu@ip-172-31-88-234:~/jenkins$
```



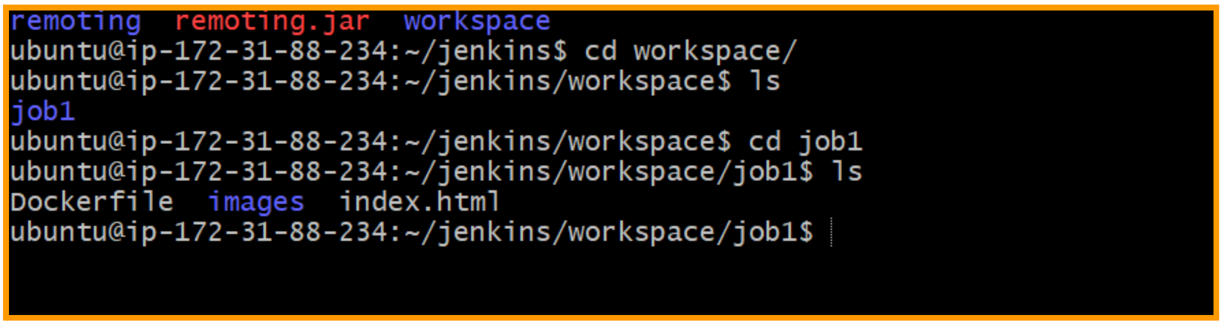
Project job1

Will be DockerFile and testing it for Develop Branch

Now check for workspace in Test Vm



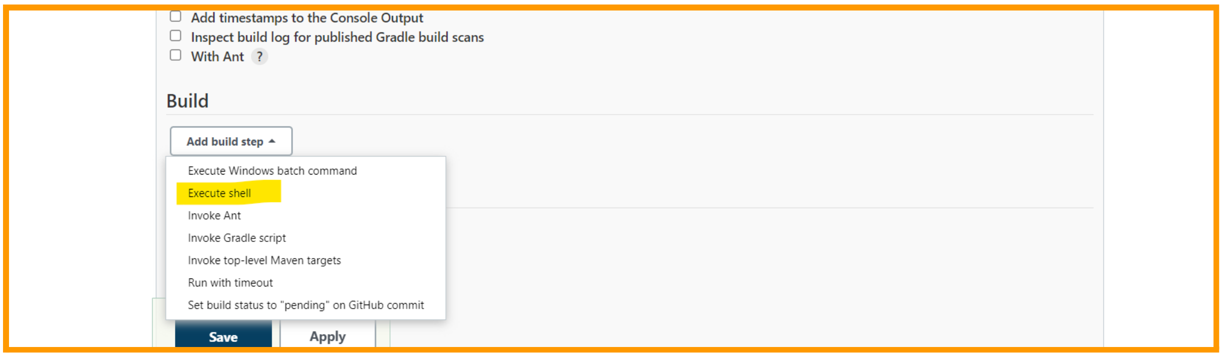
```
remoting  remoting.jar
ubuntu@ip-172-31-88-234:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-88-234:~/jenkins$
```



```
remoting  remoting.jar  workspace
ubuntu@ip-172-31-88-234:~/jenkins$ cd workspace/
ubuntu@ip-172-31-88-234:~/jenkins/workspace$ ls
job1
ubuntu@ip-172-31-88-234:~/jenkins/workspace$ cd job1
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job1$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job1$
```

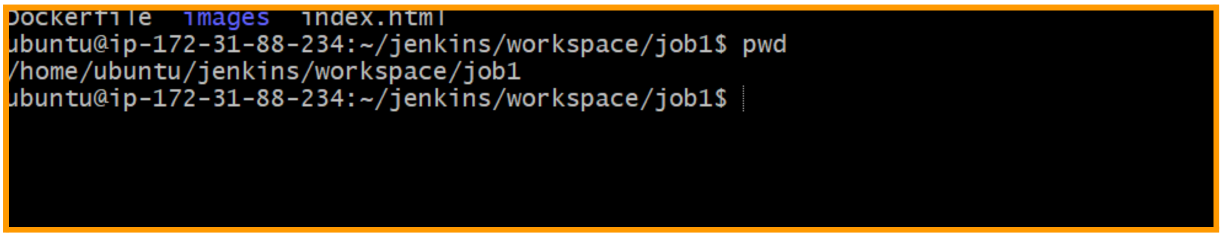
Now let us build this code with some port

GoTo Configure in Jenkins Job1   add build step   select Execute shell



□ Add timestamps to the Console Output
□ Inspect build log for published Gradle build scans
□ With Ant ?
Build
Add build step
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
Save
Apply

pwd on Test Vm   copy path

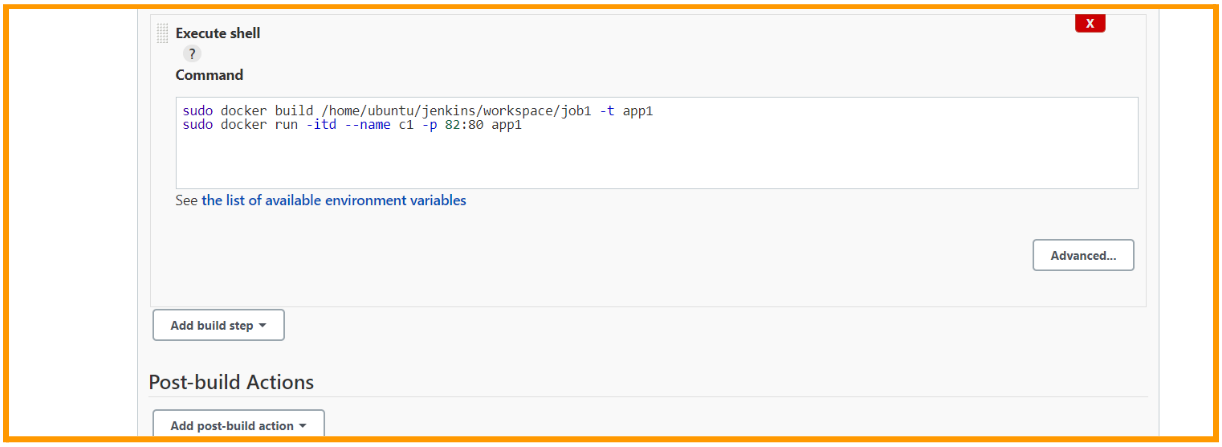


```
Dockerfile  images  index.html
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job1$ pwd
/home/ubuntu/jenkins/workspace/job1
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job1$
```

Now in execute shell,

sudo docker build /home/ubuntu/jenkins/workspace/job1 -t app1

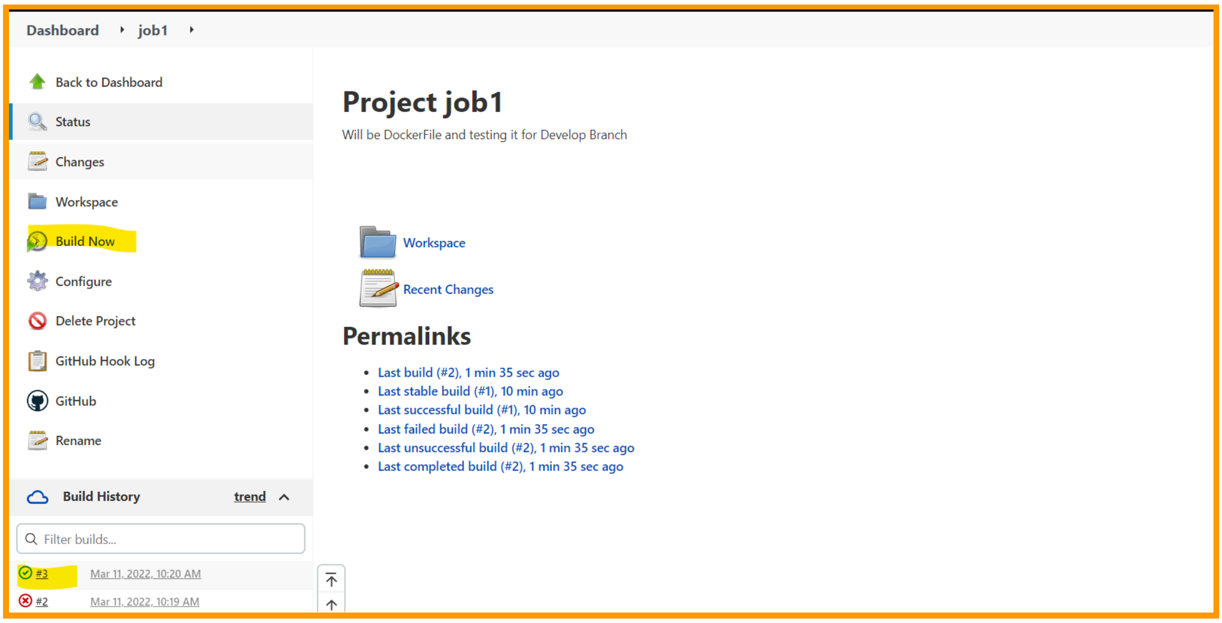
sudo docker run -itd --name c1 -p 82:80 app1



```
sudo docker build /home/ubuntu/jenkins/workspace/job1 -t app1
sudo docker run -itd --name c1 -p 82:80 app1
```

See the list of available environment variables

Apply   Save   Build Now Job



Dashboard ▸ job1 ▸

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- GitHub Hook Log
- GitHub
- Rename

Build History trend

Filter builds...

✓ #3	Mar 11, 2022, 10:20 AM
✗ #2	Mar 11, 2022, 10:19 AM

Project job1

Will be DockerFile and testing it for Develop Branch

Workspace

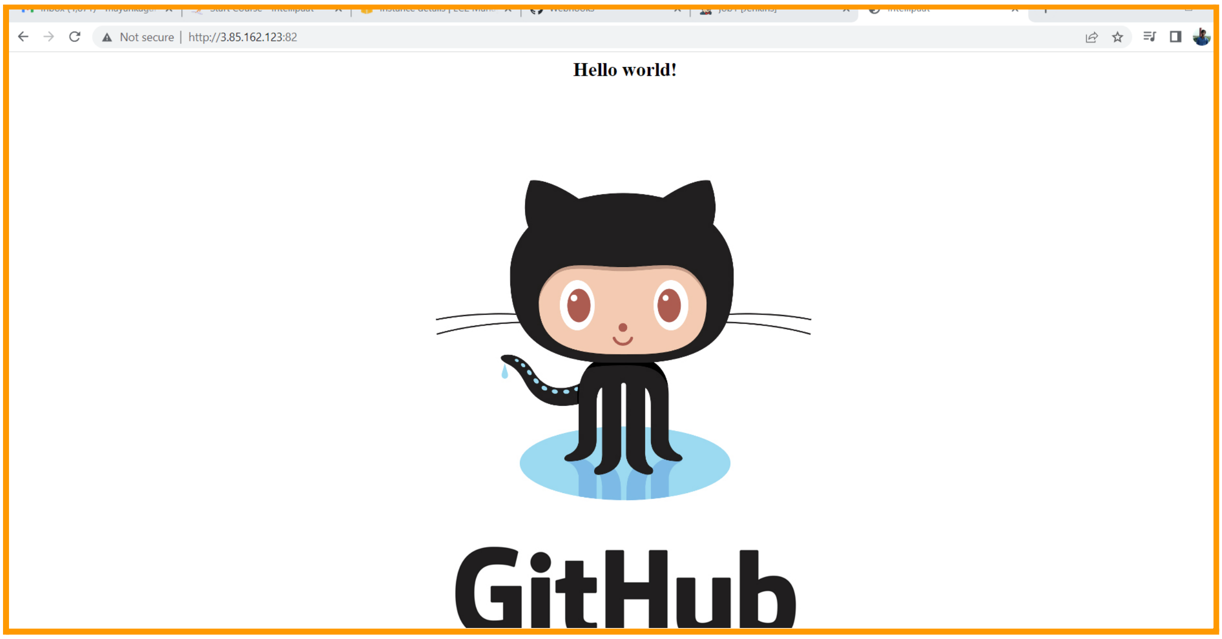
Recent Changes

Permalinks

- Last build (#2), 1 min 35 sec ago
- Last stable build (#1), 10 min ago
- Last successful build (#1), 10 min ago
- Last failed build (#2), 1 min 35 sec ago
- Last unsuccessful build (#2), 1 min 35 sec ago
- Last completed build (#2), 1 min 35 sec ago

Now check in browser –

Copy Ip address with 82 port

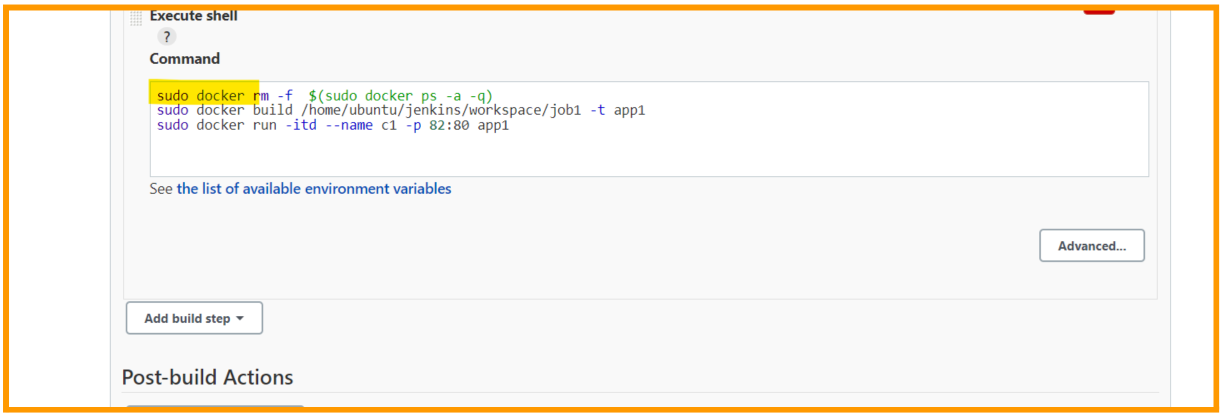


Not secure | http://3.85.162.123:82

Hello world!

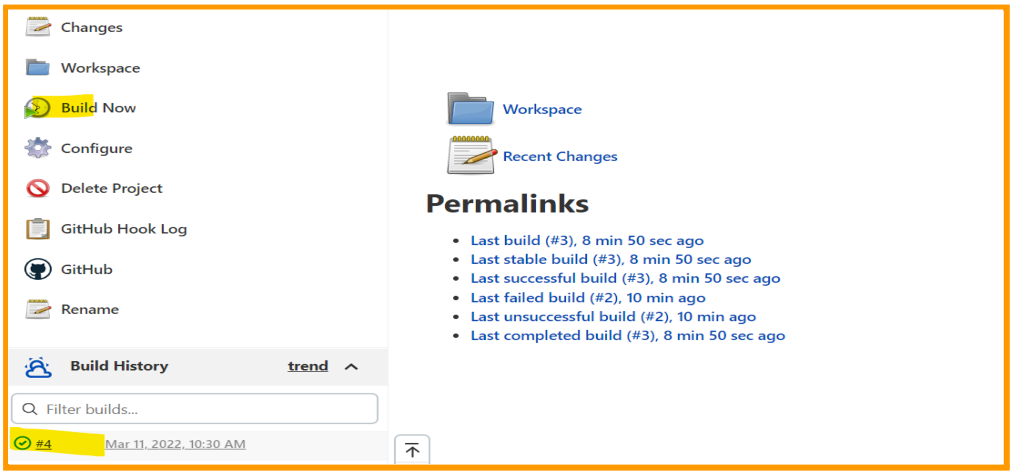
Now to remove conflict of port issue after build we need to add one more command in execute shell

sudo docker rm -f $(sudo docker ps -a -q)



```
sudo docker rm -f  $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/jenkins/workspace/job1 -t app1
sudo docker run -itd --name c1 -p 82:80 app1
```

See the list of available environment variables



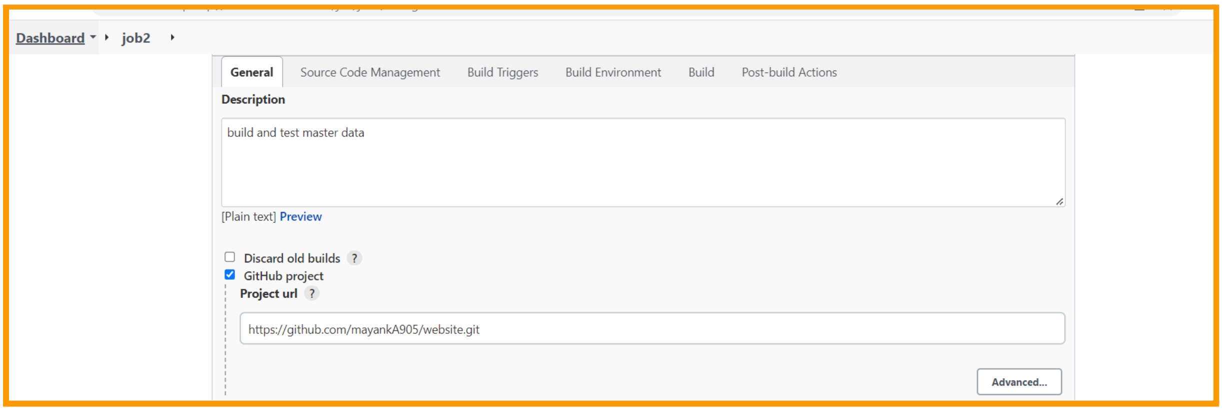
#4 Mar 11, 2022, 10:30 AM

Permalinks

- Last build (#3), 8 min 50 sec ago
- Last stable build (#3), 8 min 50 sec ago
- Last successful build (#3), 8 min 50 sec ago
- Last failed build (#2), 10 min ago
- Last unsuccessful build (#2), 10 min ago
- Last completed build (#3), 8 min 50 sec ago

Now create Job2 with same method same configuration but now for testing master branch

Jenkin Dashboard - new item - Job2 - Freestyle project



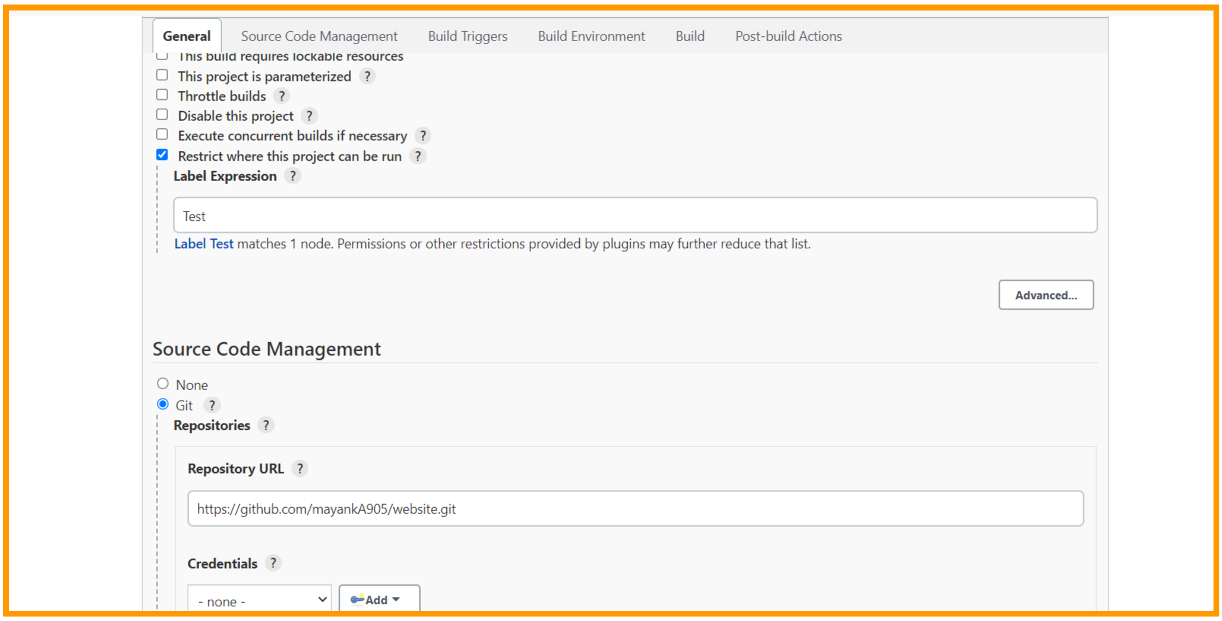
Description

build and test master data

[Plain text] Preview

□ Discard old builds ?
☑ GitHub project
Project url ?

https://github.com/mayankA905/website.git



□ This build requires lockable resources

□ This project is parameterized ?

□ Throttle builds ?

□ Disable this project ?

□ 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 ?

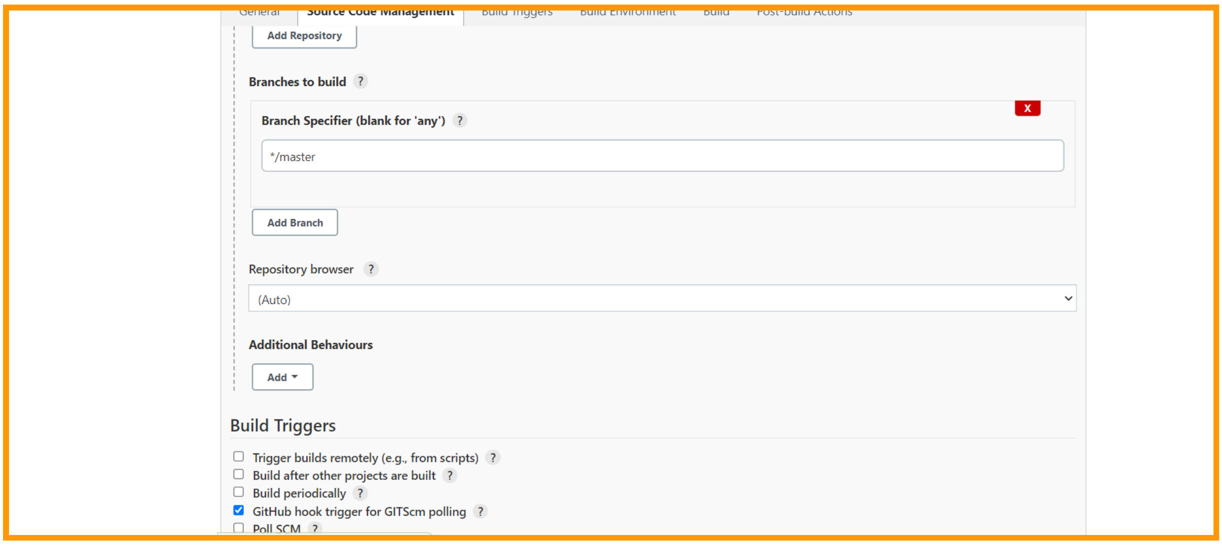
Repositories ?

Repository URL ?

https://github.com/mayankA905/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 ?

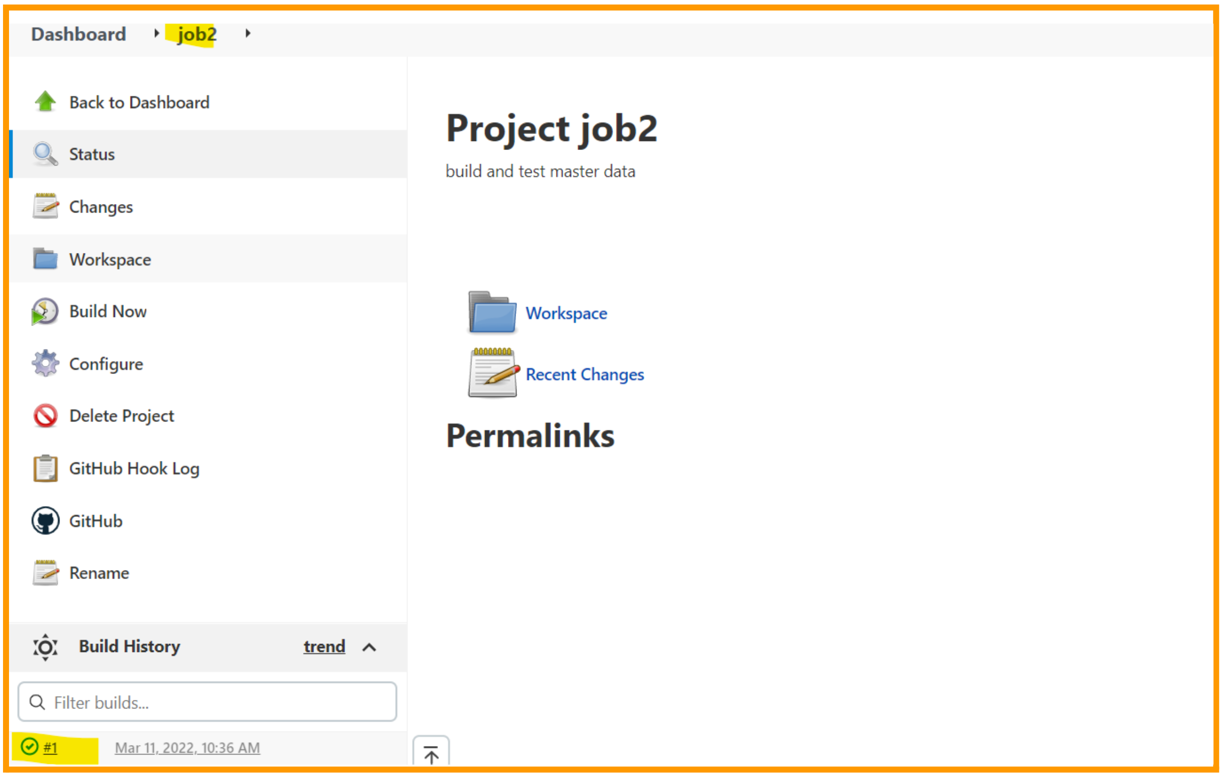
□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

□ Poll SCM ?

Apply and Save

Build Now

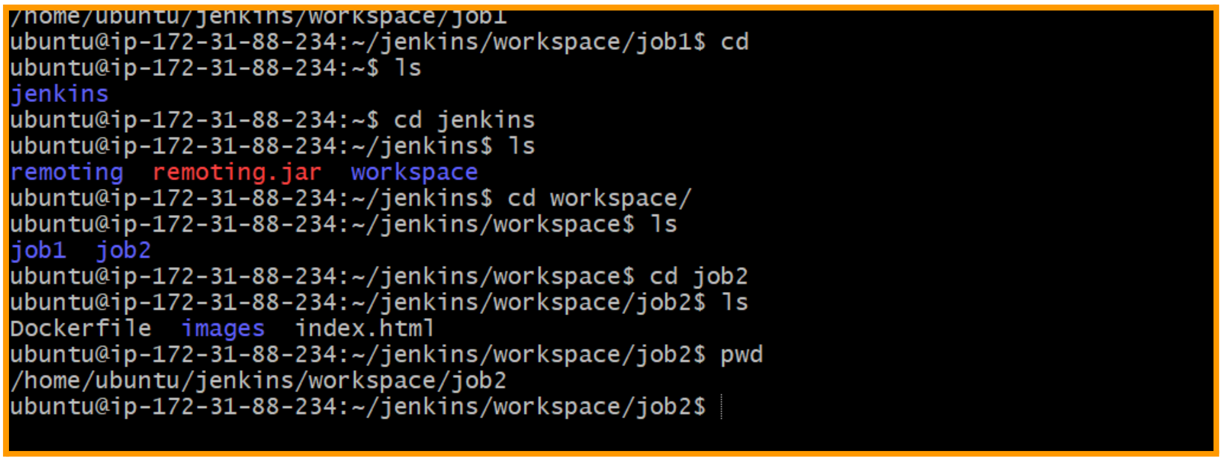


Project job2

build and test master data

Permalinks

Check for job2 in workspace in test Vm and fetch pwd



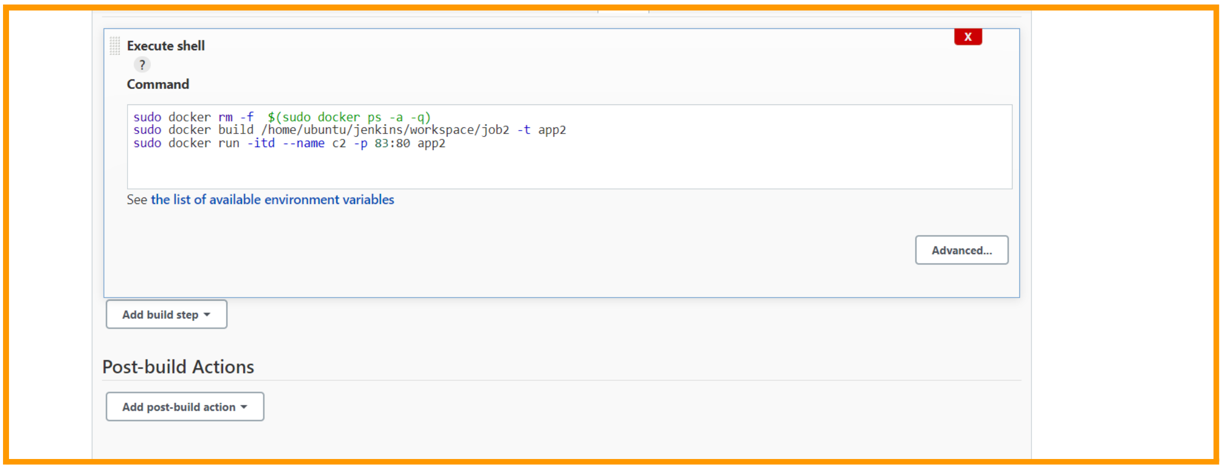
```
/home/ubuntu/jenkins/workspace/job1
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job1$ cd
ubuntu@ip-172-31-88-234:~$ ls
jenkins
ubuntu@ip-172-31-88-234:~$ cd jenkins
ubuntu@ip-172-31-88-234:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-88-234:~/jenkins$ cd workspace/
ubuntu@ip-172-31-88-234:~/jenkins/workspace$ ls
job1  job2
ubuntu@ip-172-31-88-234:~/jenkins/workspace$ cd job2
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job2$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job2$ pwd
/home/ubuntu/jenkins/workspace/job2
ubuntu@ip-172-31-88-234:~/jenkins/workspace/job2$
```

Configure Job2 -   Build   Execute Shell

sudo docker rm -f $(sudo docker ps -a -q)

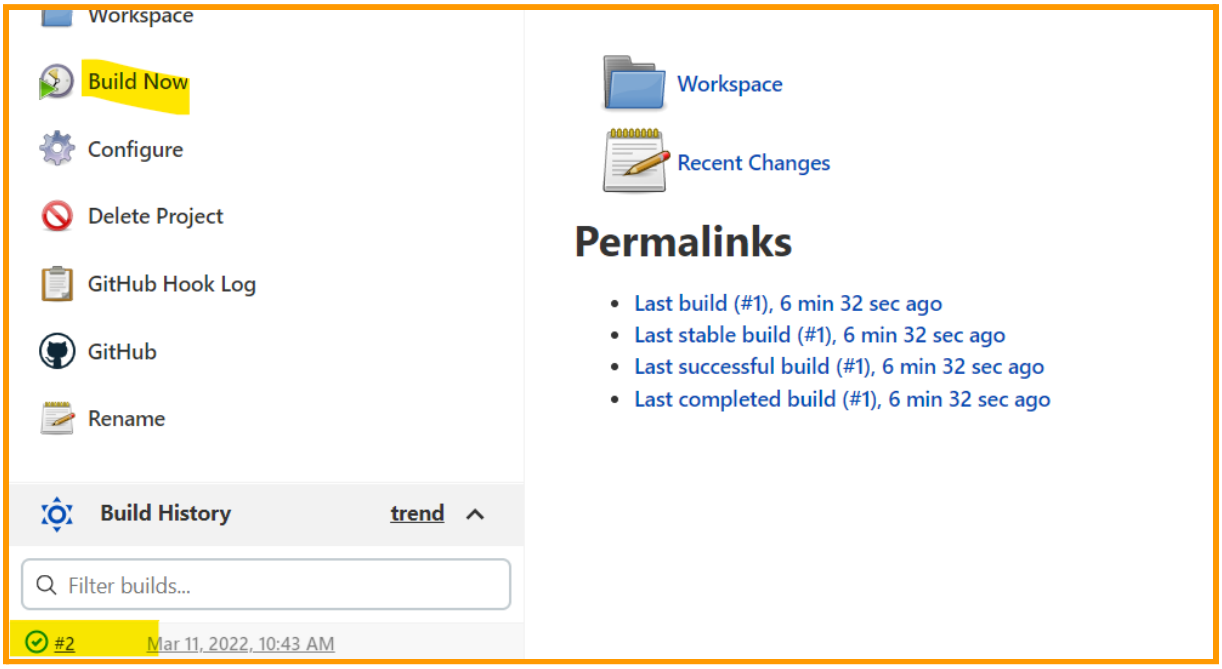
sudo docker build /home/ubuntu/jenkins/workspace/job2 -t app2

sudo docker run -itd --name c2 -p 83:80 app2



```
sudo docker rm -f  $(sudo docker ps -a -q)
sudo docker build /home/ubuntu/jenkins/workspace/job2 -t app2
sudo docker run -itd --name c2 -p 83:80 app2
```

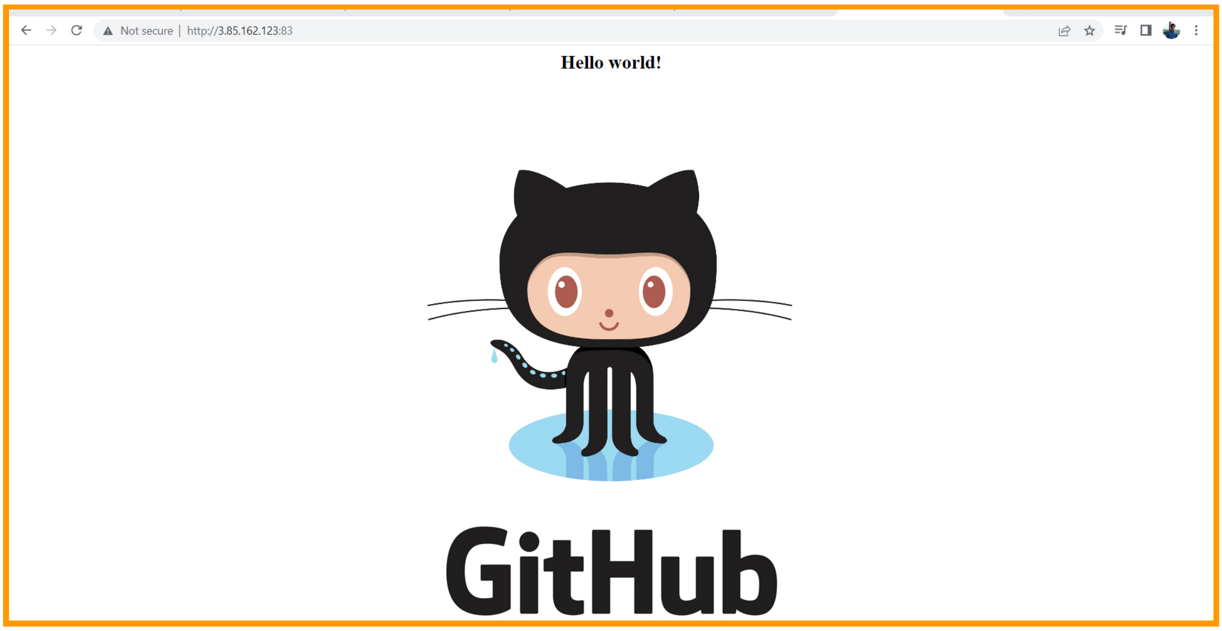
Apply and Save   Build Now



Permalinks

- Last build (#1), 6 min 32 sec ago
- Last stable build (#1), 6 min 32 sec ago
- Last successful build (#1), 6 min 32 sec ago
- Last completed build (#1), 6 min 32 sec ago

Check with port 83 on browser



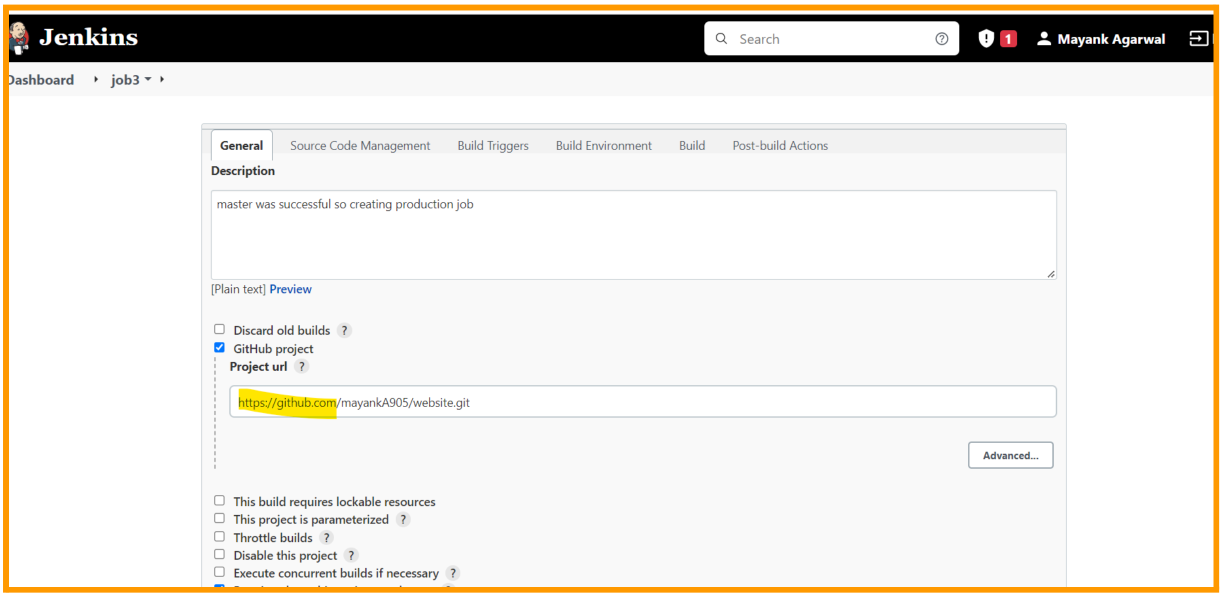
Not secure | http://3.85.162.123:83

Hello world!

Now Create Job3 for Production VM which is linked with Job2

Dashboard-> new Item Job3 FreeStyle

In Job3 job we don't need to enable Webhook trigger because job3 will just work if Job2 if fine.



Description

master was successful so creating production job

[Plain text] Preview

☐ Discard old builds ?

☑ GitHub project

Project url ?

https://github.com/mayankA905/website.git

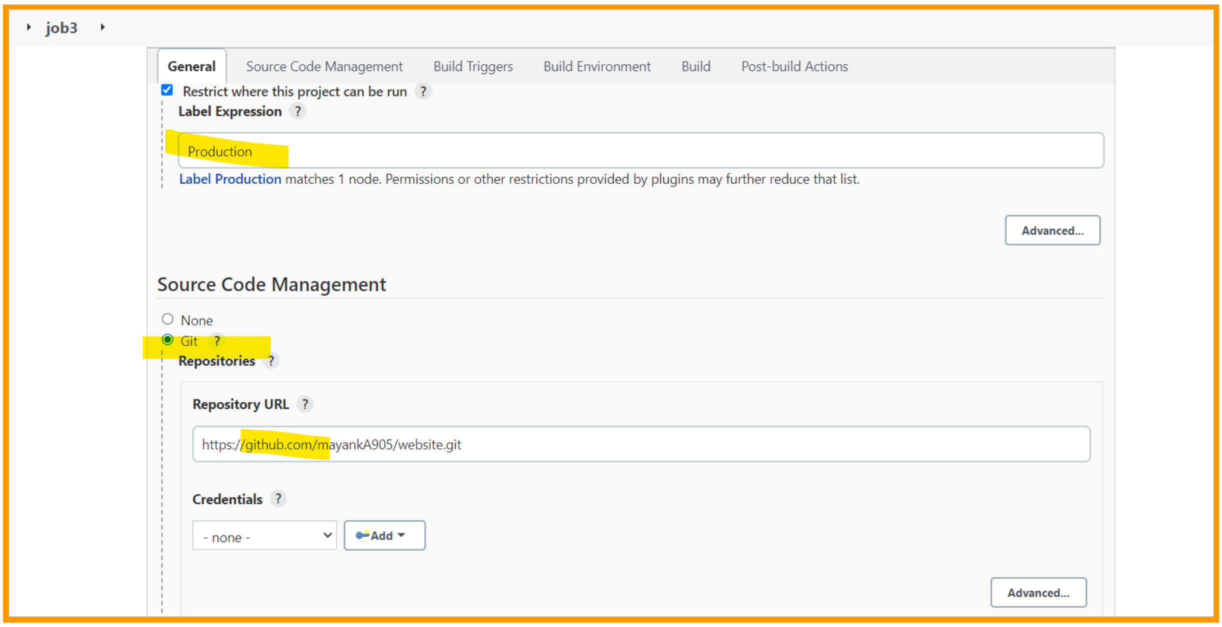
☐ This build requires lockable resources

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Disable this project ?

☐ Execute concurrent builds if necessary ?



☑ Restrict where this project can be run ?

Label Expression ?

Production

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

Source Code Management

○ None

● Git ?

Repositories ?

Repository URL ?

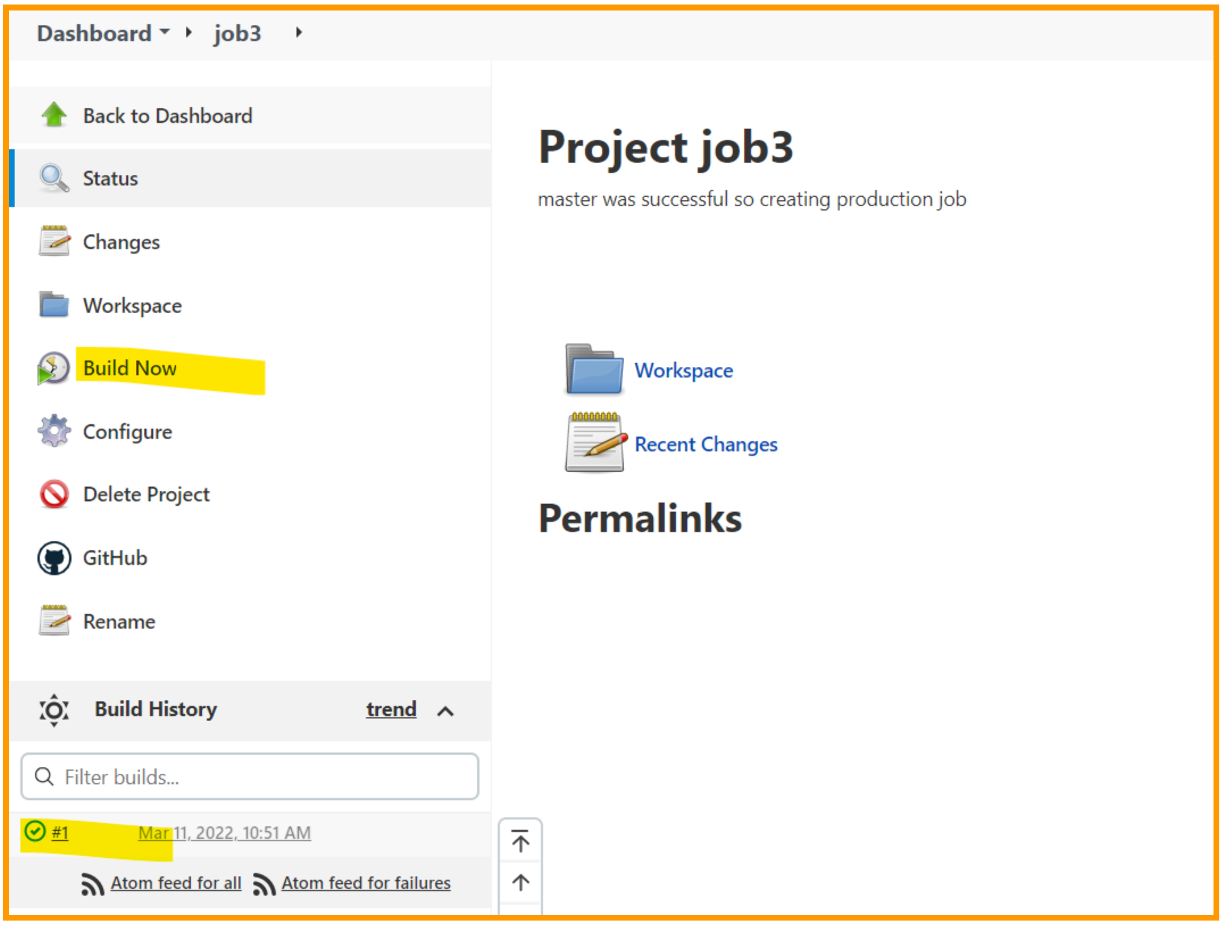
https://github.com/mayankA905/website.git

Credentials ?

- none -

Apply and Save

Build now

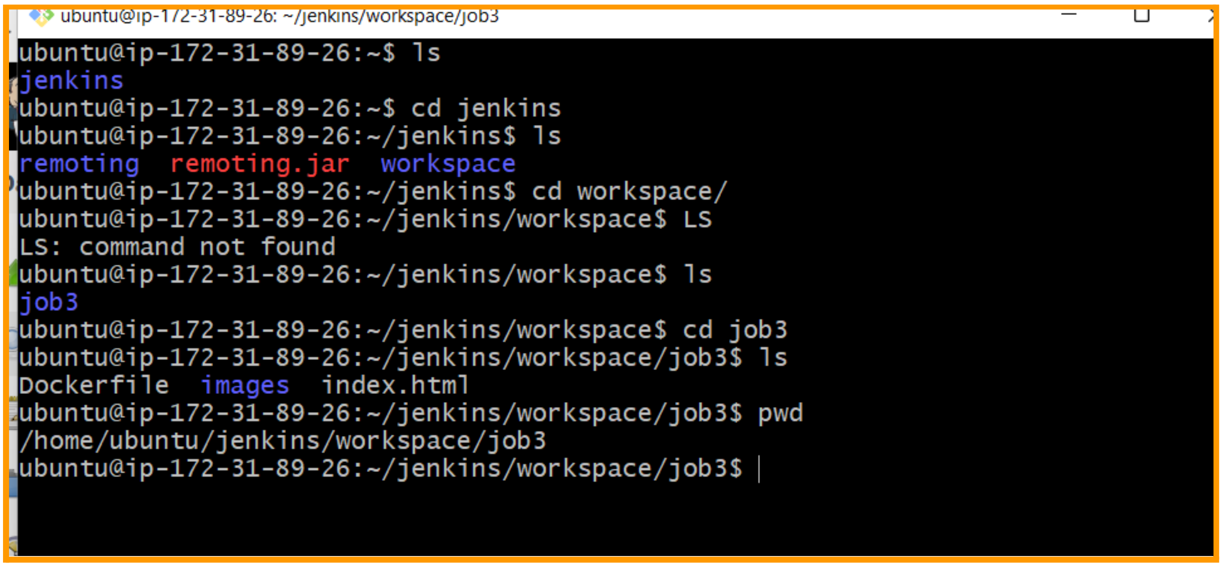


Project job3

master was successful so creating production job

#1 Mar 11, 2022, 10:51 AM

Go To Production Vm and check for workspace and copy pwd path

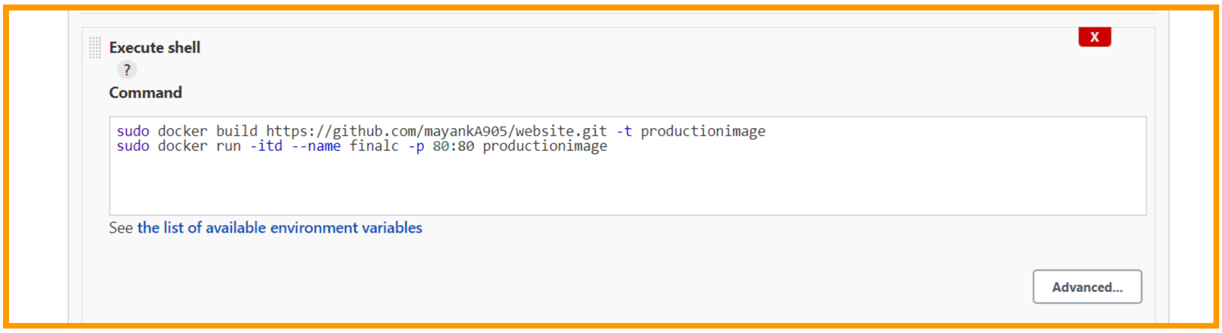


```
ubuntu@ip-172-31-89-26:~$ ls
jenkins
ubuntu@ip-172-31-89-26:~$ cd jenkins
ubuntu@ip-172-31-89-26:~/jenkins$ ls
remoting  remoting.jar  workspace
ubuntu@ip-172-31-89-26:~/jenkins$ cd workspace/
ubuntu@ip-172-31-89-26:~/jenkins/workspace$ LS
LS: command not found
ubuntu@ip-172-31-89-26:~/jenkins/workspace$ ls
job3
ubuntu@ip-172-31-89-26:~/jenkins/workspace$ cd job3
ubuntu@ip-172-31-89-26:~/jenkins/workspace/job3$ ls
Dockerfile  images  index.html
ubuntu@ip-172-31-89-26:~/jenkins/workspace/job3$ pwd
/home/ubuntu/jenkins/workspace/job3
ubuntu@ip-172-31-89-26:~/jenkins/workspace/job3$ 
```

Configure Job3 - Build Execute Shell

sudo docker build https://github.com/mayankA905/website.git -t productionimage

sudo docker run -itd --name finalc -p 80:80 productionimage



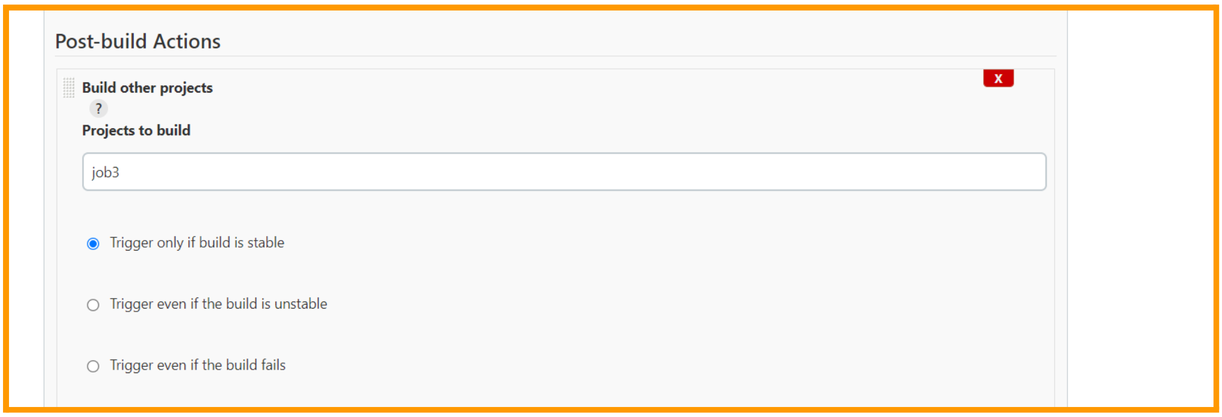
Command

```
sudo docker build https://github.com/mayankA905/website.git -t productionimage
sudo docker run -itd --name finalc -p 80:80 productionimage
```

See the list of available environment variables

Now enable Job3 configuration in Job2

Goto Job2 configure Post Build Action Select Build other project apply and Save



Projects to build: job3

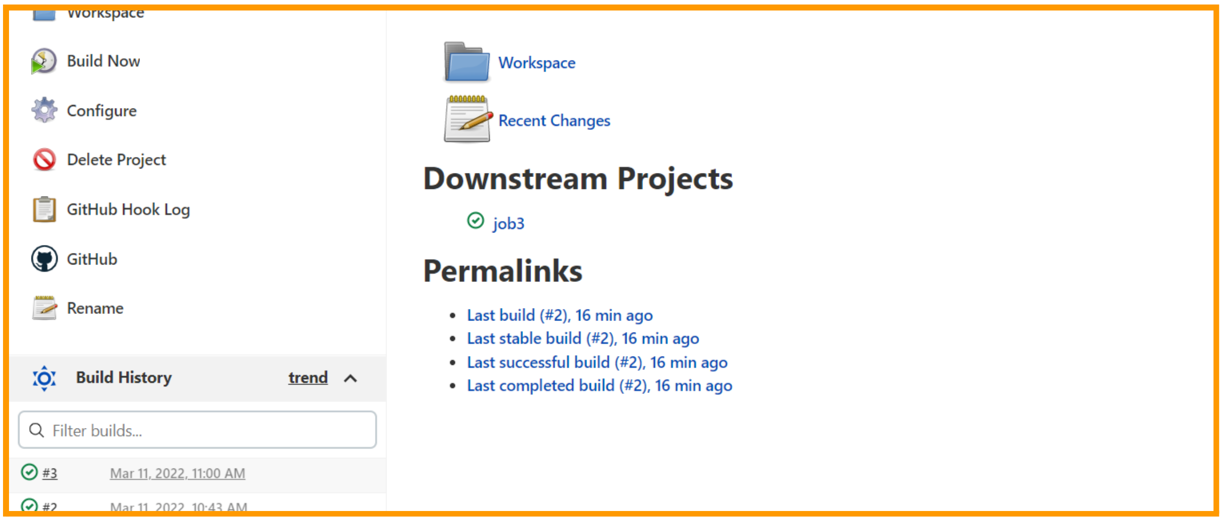
● Trigger only if build is stable

○ Trigger even if the build is unstable

○ Trigger even if the build fails

Now Run Job2 and then see will our Job3 is automatic run or not

Build now



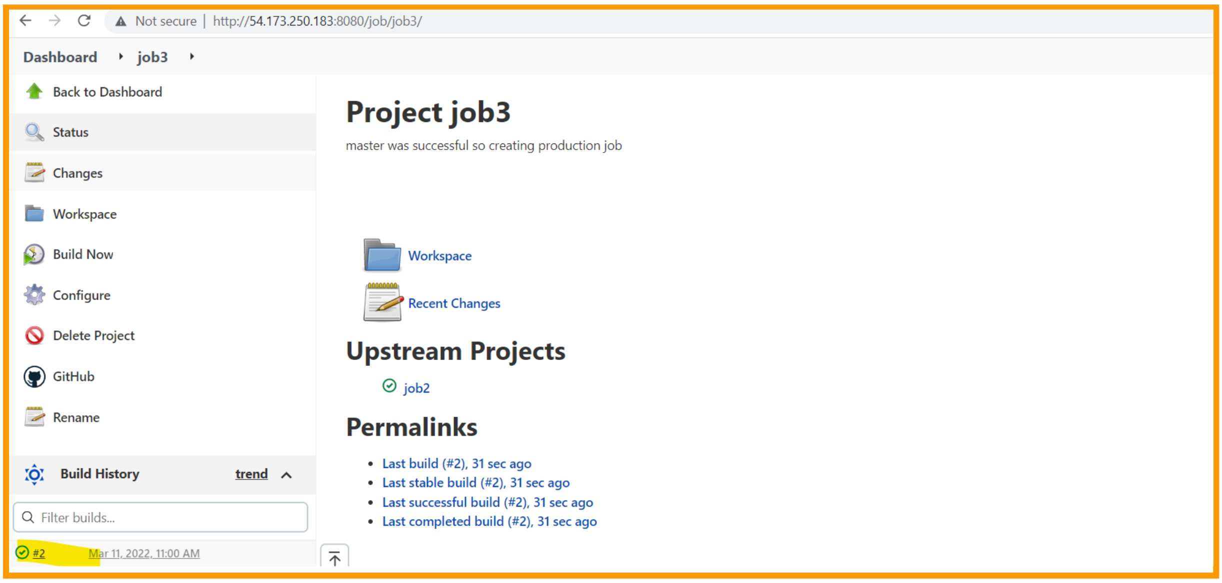
Downstream Projects

✓ job3

Permalinks

- Last build (#2), 16 min ago
- Last stable build (#2), 16 min ago
- Last successful build (#2), 16 min ago
- Last completed build (#2), 16 min ago

Now check in Job3



Project job3

master was successful so creating production job

Upstream Projects

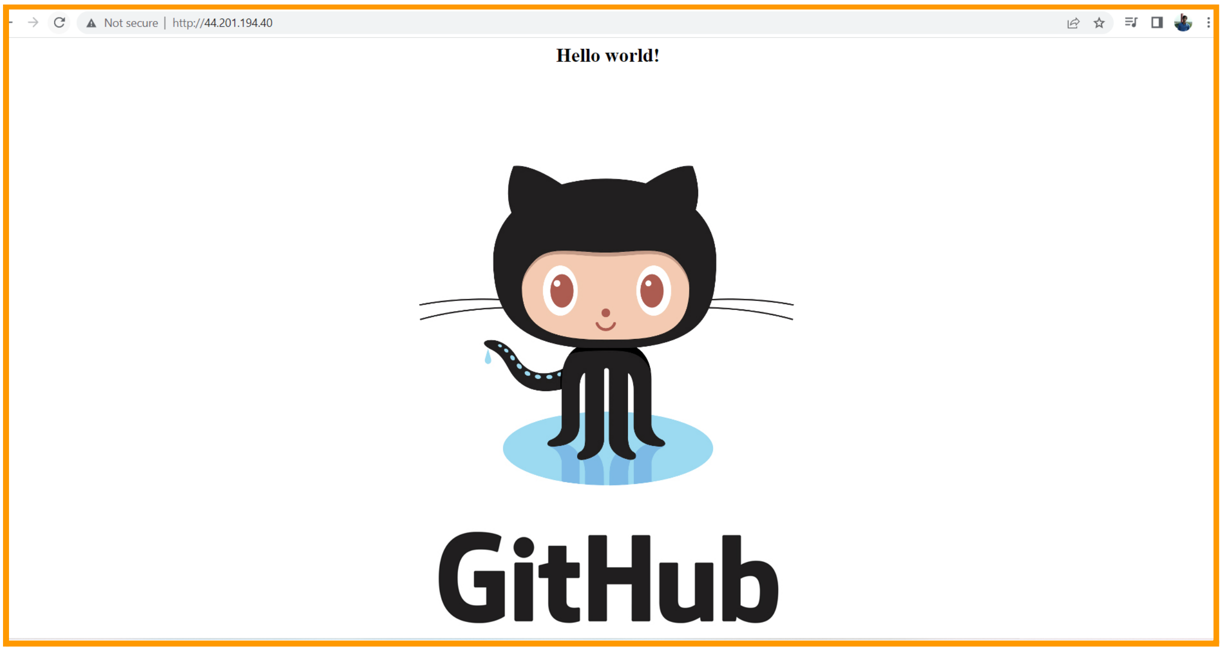
✓ job2

Permalinks

- Last build (#2), 31 sec ago
- Last stable build (#2), 31 sec ago
- Last successful build (#2), 31 sec ago
- Last completed build (#2), 31 sec ago

Build completed automatically

Now check with production IP address with port 80 on browser

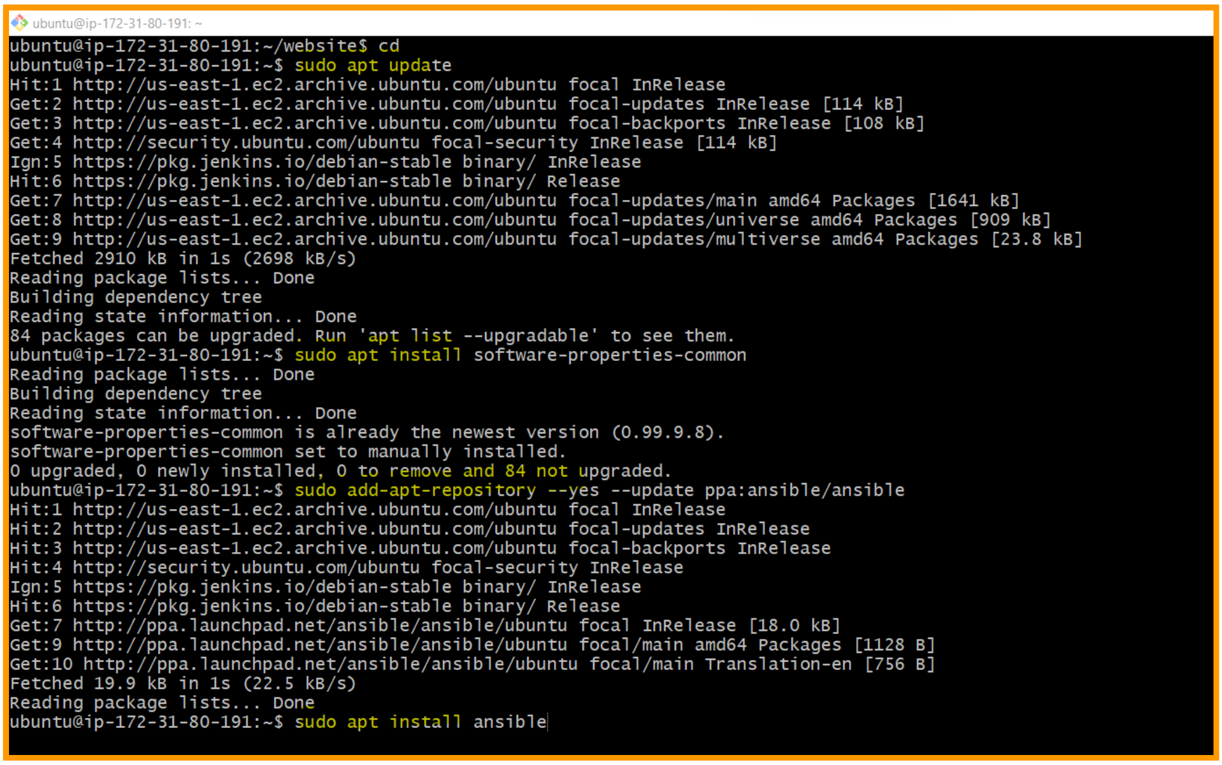


Not secure | http://44.201.194.40

Hello world!

Now the last part is to install ansible in master for nginx configuration

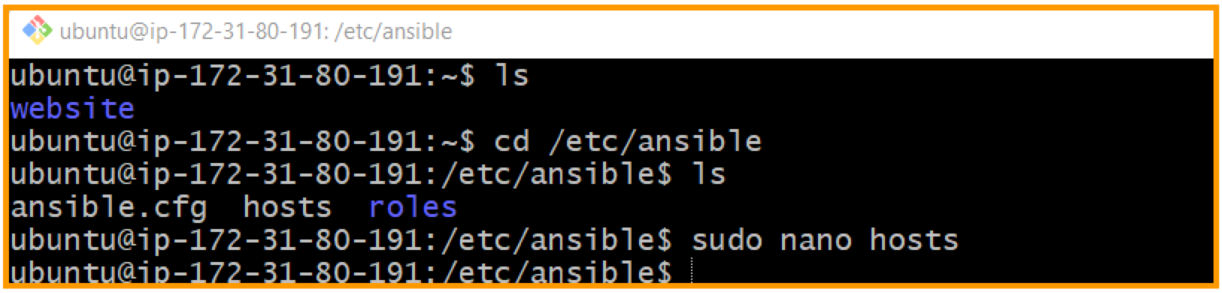
```
$ sudo apt update
$ sudo apt install software-properties-common
$ sudo add-apt-repository --yes --update ppa:ansible/ansible
$ sudo apt install ansible
```



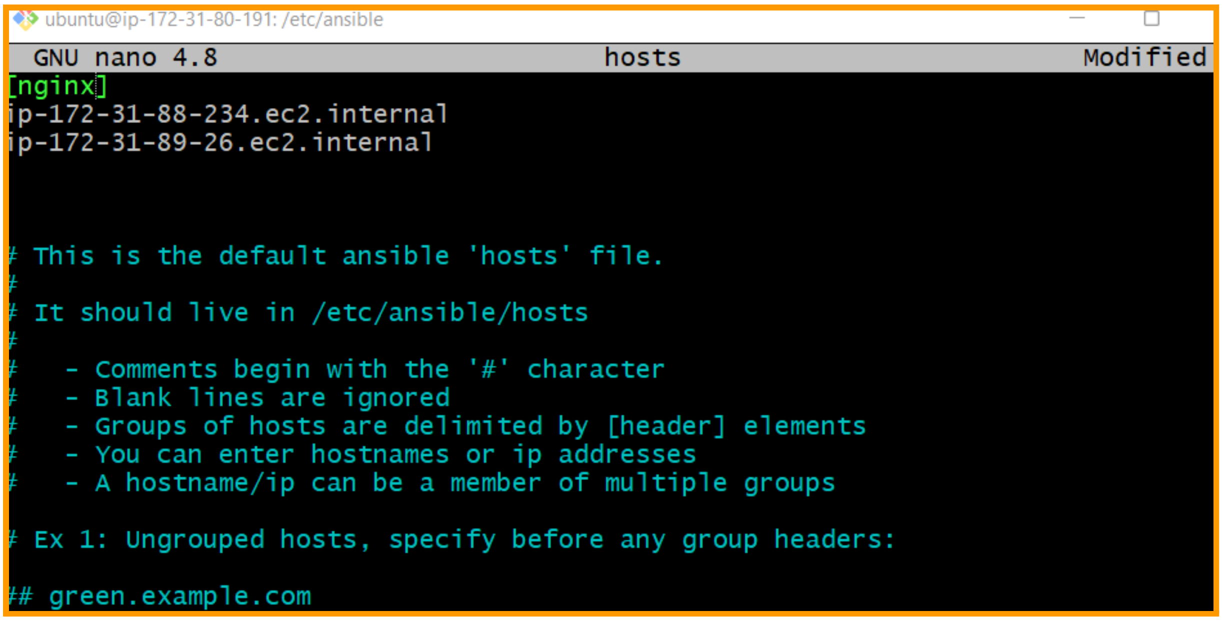
```
ubuntu@ip-172-31-80-191:~/website$ cd
ubuntu@ip-172-31-80-191:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [1641 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [909 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [23.8 kB]
Fetched 2910 kB in 1s (2698 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
84 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-80-191:~$ sudo apt install software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
software-properties-common is already the newest version (0.99.9.8).
software-properties-common set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 84 not upgraded.
ubuntu@ip-172-31-80-191:~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release
Get:7 http://ppa.launchpad.net/ansible/ansible/ubuntu focal InRelease [18.0 kB]
Get:9 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main amd64 Packages [1128 B]
Get:10 http://ppa.launchpad.net/ansible/ansible/ubuntu focal/main Translation-en [756 B]
Fetched 19.9 kB in 1s (22.5 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-80-191:~$ sudo apt install ansible
```

Now

ls-> cd /etc/ansible   ls   sudo nano hosts

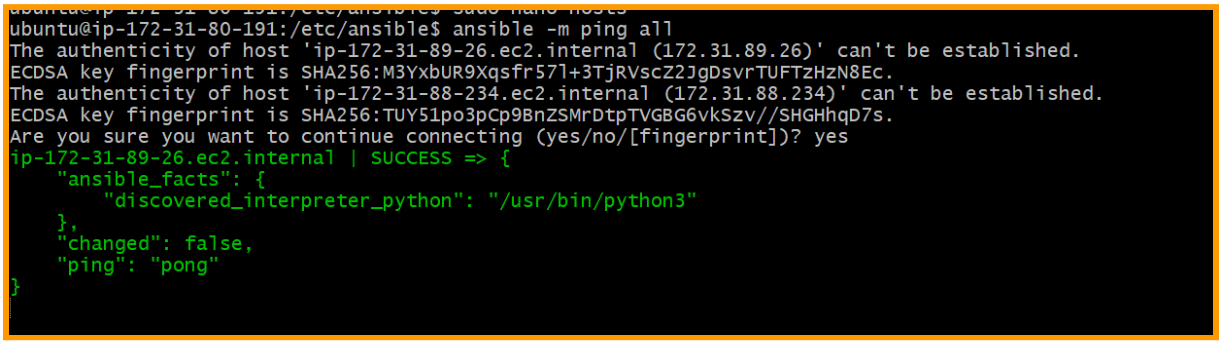


```
ubuntu@ip-172-31-80-191:~$ ls
website
ubuntu@ip-172-31-80-191:~$ cd /etc/ansible
ubuntu@ip-172-31-80-191:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-80-191:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-80-191:/etc/ansible$
```

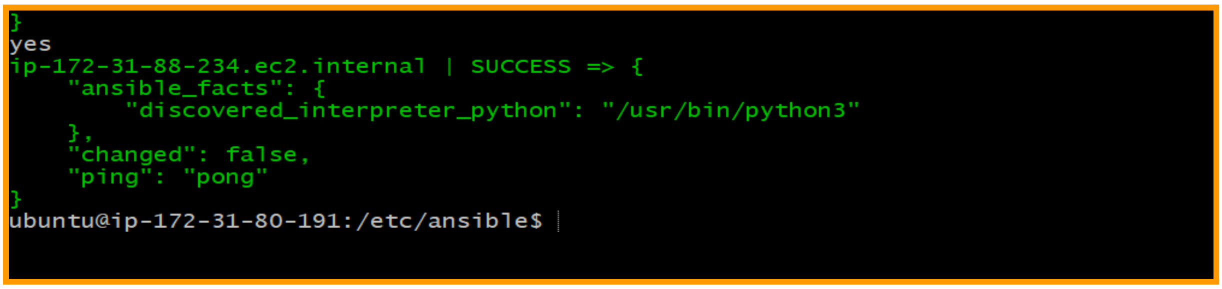


```
ubuntu@ip-172-31-80-191: /etc/ansible
  GNU nano 4.8                         hosts                          Modified
[nginx]
ip-172-31-88-234.ec2.internal
ip-172-31-89-26.ec2.internal


# This is the default ansible 'hosts' file.
#
# It should live in /etc/ansible/hosts
#
#   - Comments begin with the '#' character
#   - Blank lines are ignored
#   - Groups of hosts are delimited by [header] elements
#   - You can enter hostnames or ip addresses
#   - A hostname/ip can be a member of multiple groups

# Ex 1: Ungrouped hosts, specify before any group headers:

## green.example.com
```

ansible -m ping all



```
ubuntu@ip-172-31-80-191:/etc/ansible$ sudo nano hosts
ubuntu@ip-172-31-80-191:/etc/ansible$ ansible -m ping all
The authenticity of host 'ip-172-31-89-26.ec2.internal (172.31.89.26)' can't be established.
ECDSA key fingerprint is SHA256:M3YxbUR9Xqsfr57l+3TjRVscZ2JgDsvrTUFTzHzN8Ec.
The authenticity of host 'ip-172-31-88-234.ec2.internal (172.31.88.234)' can't be established.
ECDSA key fingerprint is SHA256:TUY51po3pCp9BnZSMrDtpTVGBG6vkSzv//SHGHhqD7s.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ip-172-31-89-26.ec2.internal | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```



```
}
yes
ip-172-31-88-234.ec2.internal | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
ubuntu@ip-172-31-80-191:/etc/ansible$
```

Now

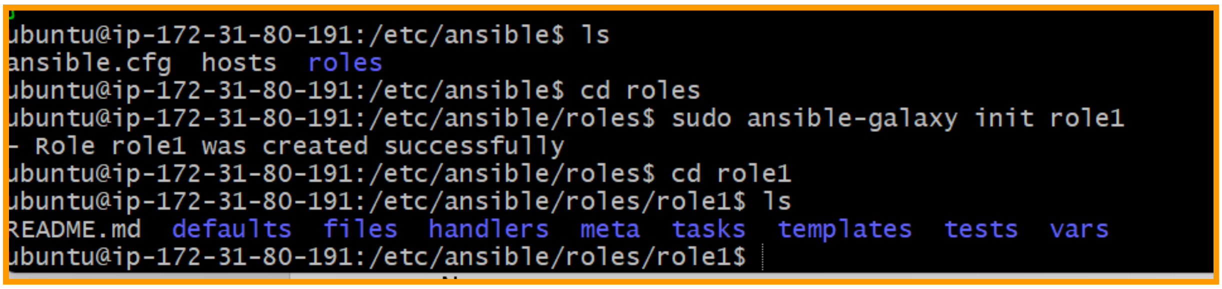
ls

cd roles

sudo ansible-galaxy init role1

cd role1

ls

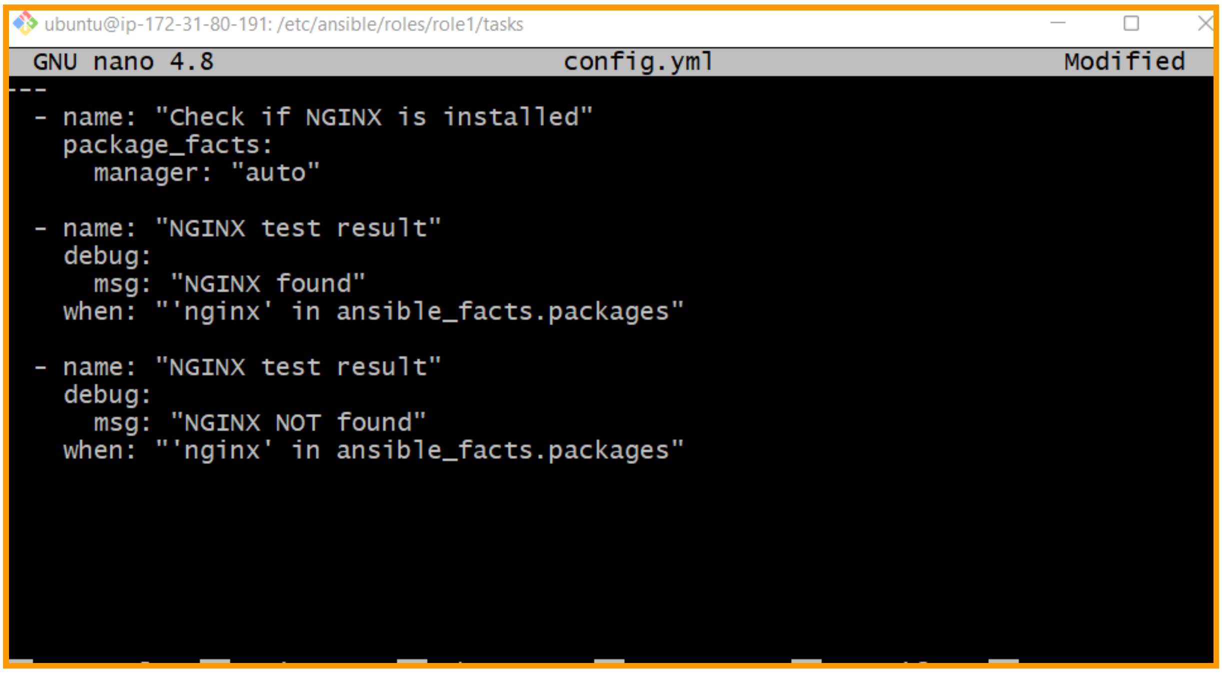


```
ubuntu@ip-172-31-80-191:/etc/ansible$ ls
ansible.cfg  hosts  roles
ubuntu@ip-172-31-80-191:/etc/ansible$ cd roles
ubuntu@ip-172-31-80-191:/etc/ansible/roles$ sudo ansible-galaxy init role1
- Role role1 was created successfully
ubuntu@ip-172-31-80-191:/etc/ansible/roles$ cd role1
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1$ ls
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1$
```

cd task

ls   sudo nano config.yml

```
---
 - name: "Check if NGINX is installed"
   package_facts:
     manager: "auto"

 - name: "NGINX test result"
   debug:
     msg: "NGINX found"
   when: "'nginx' in ansible_facts.packages"

 - name: "NGINX test result"
   debug:
     msg: "NGINX NOT found"
   when: "'nginx' in ansible_facts.packages"
```



ubuntu@ip-172-31-80-191: /etc/ansible/roles/role1/tasks

```
  GNU nano 4.8                         config.yml                          Modified
---
 - name: "Check if NGINX is installed"
   package_facts:
     manager: "auto"

 - name: "NGINX test result"
   debug:
     msg: "NGINX found"
   when: "'nginx' in ansible_facts.packages"

 - name: "NGINX test result"
   debug:
     msg: "NGINX NOT found"
   when: "'nginx' in ansible_facts.packages"
```

Now

sudo main.yml

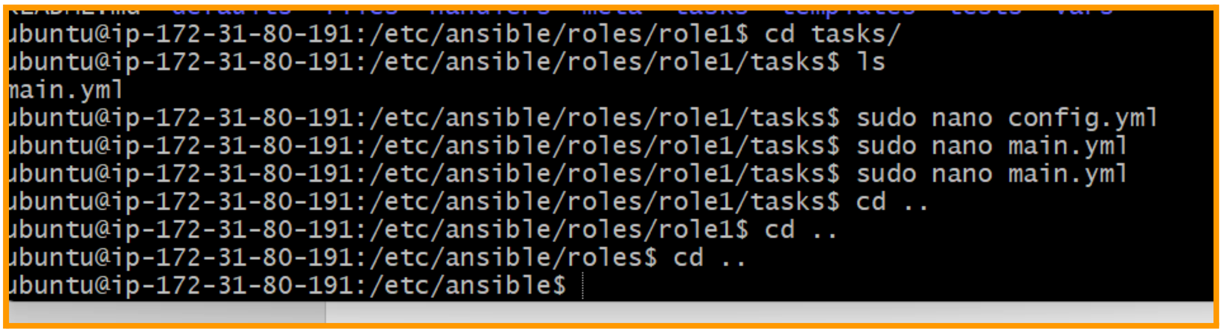
 - include : config.yml

Create Playbook now in ansible directory

Cd ..

Cd ..

Cd ..

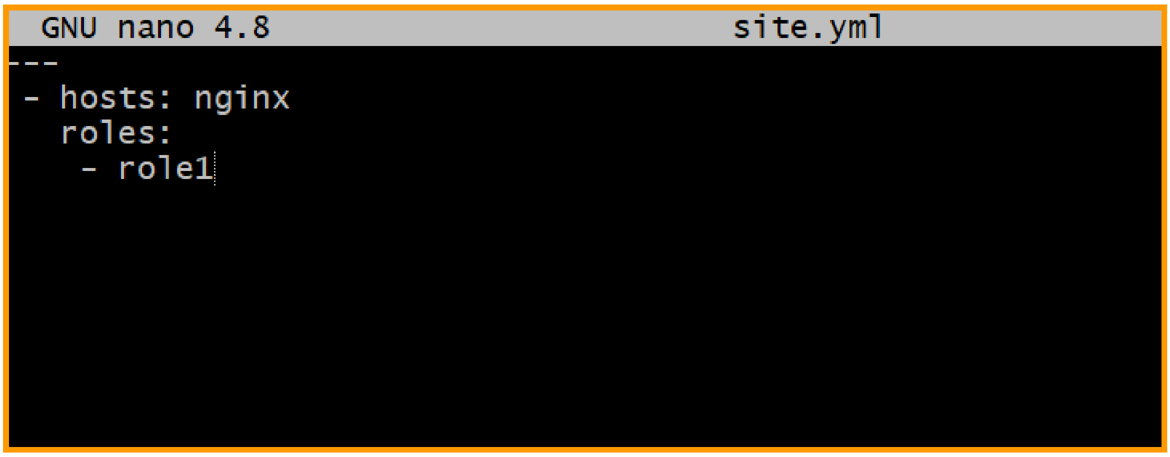


```
README.md  defaults  files  handlers  meta  tasks  templates  tests  vars
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1$ cd tasks/
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1/tasks$ ls
main.yml
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1/tasks$ sudo nano config.yml
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1/tasks$ sudo nano main.yml
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1/tasks$ sudo nano main.yml
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1/tasks$ cd ..
ubuntu@ip-172-31-80-191:/etc/ansible/roles/role1$ cd ..
ubuntu@ip-172-31-80-191:/etc/ansible/roles$ cd ..
ubuntu@ip-172-31-80-191:/etc/ansible$
```

ls

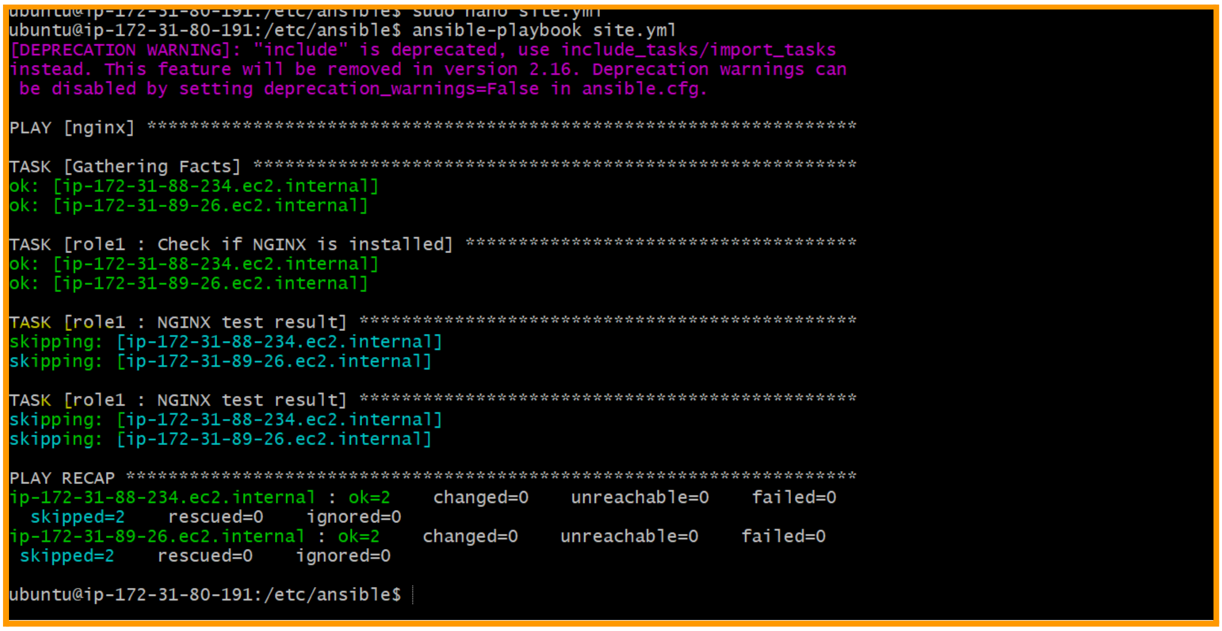
sudo nano site.yml

```
- hosts: nginx
  roles:
   - role1
```



```
  GNU nano 4.8                                  site.yml
---
 - hosts: nginx
   roles:
     - role1
```

Ansible-playbook site.yml

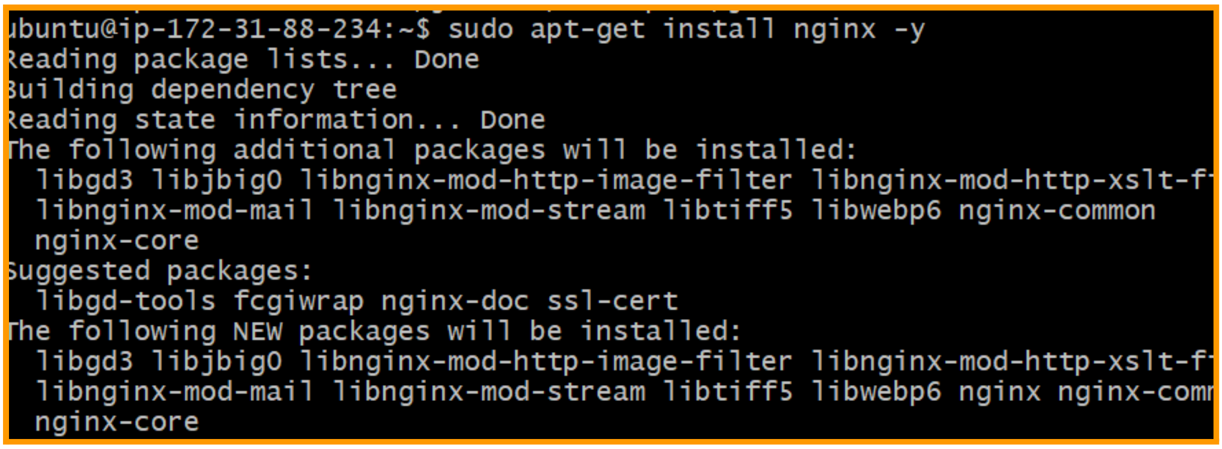


```
ubuntu@ip-172-31-80-191:/etc/ansible$ sudo nano site.yml
ubuntu@ip-172-31-80-191:/etc/ansible$ ansible-playbook site.yml
[DEPRECATION WARNING]: "include" is deprecated, use include_tasks/import_tasks
instead. This feature will be removed in version 2.16. Deprecation warnings can
 be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [nginx] ******************************************************************

TASK [Gathering Facts] ********************************************************
ok: [ip-172-31-88-234.ec2.internal]
ok: [ip-172-31-89-26.ec2.internal]

TASK [role1 : Check if NGINX is installed] ************************************
ok: [ip-172-31-88-234.ec2.internal]
ok: [ip-172-31-89-26.ec2.internal]

TASK [role1 : NGINX test result] **********************************************
skipping: [ip-172-31-88-234.ec2.internal]
skipping: [ip-172-31-89-26.ec2.internal]

TASK [role1 : NGINX test result] **********************************************
skipping: [ip-172-31-88-234.ec2.internal]
skipping: [ip-172-31-89-26.ec2.internal]

PLAY RECAP ********************************************************************
ip-172-31-88-234.ec2.internal : ok=2    changed=0    unreachable=0    failed=0
  skipped=2    rescued=0    ignored=0
ip-172-31-89-26.ec2.internal : ok=2    changed=0    unreachable=0    failed=0
 skipped=2    rescued=0    ignored=0

ubuntu@ip-172-31-80-191:/etc/ansible$
```

As we saw it is skipping the part so if we have install Nginx in one of the Salve server

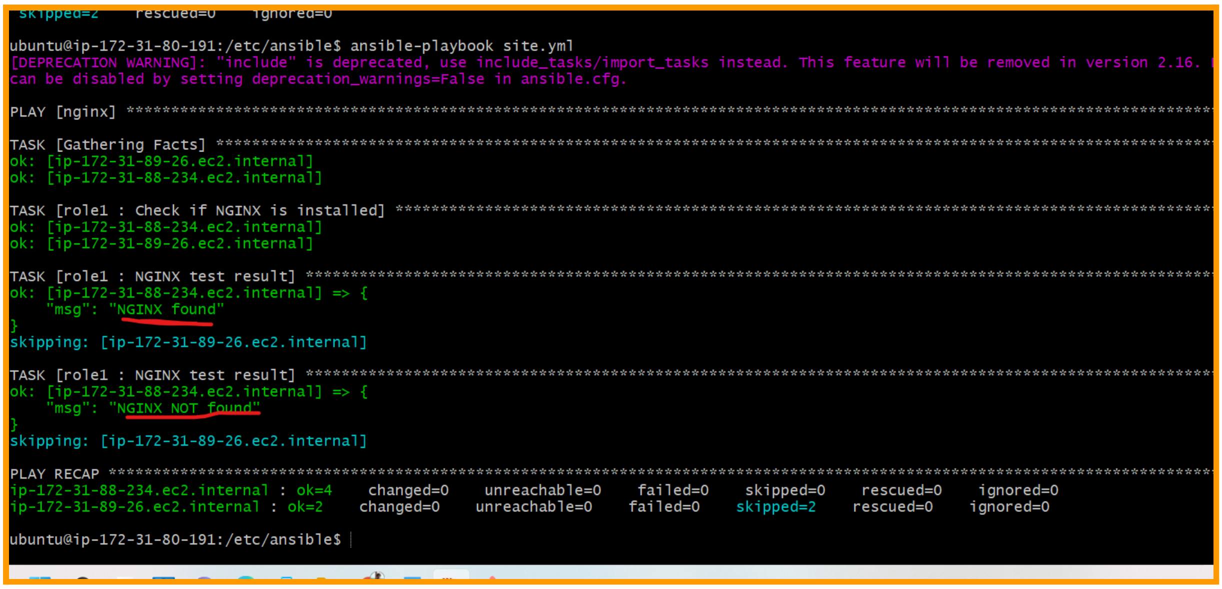
Goto test Vm

sudo apt-get install nginx -y



```
ubuntu@ip-172-31-88-234:~$ sudo apt-get install nginx -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libgd3 libjbig0 libnginx-mod-http-image-filter libnginx-mod-http-xslt-f
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 nginx-common
  nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  libgd3 libjbig0 libnginx-mod-http-image-filter libnginx-mod-http-xslt-f
  libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 nginx nginx-comm
  nginx-core
```

Run the ansible playbook again



```
 skipped=2    rescued=0    ignored=0

ubuntu@ip-172-31-80-191:/etc/ansible$ ansible-playbook site.yml
[DEPRECATION WARNING]: "include" is deprecated, use include_tasks/import_tasks instead. This feature will be removed in version 2.16.
can be disabled by setting deprecation_warnings=False in ansible.cfg.

PLAY [nginx] ******************************************************************************************************************

TASK [Gathering Facts] ********************************************************************************************************
ok: [ip-172-31-89-26.ec2.internal]
ok: [ip-172-31-88-234.ec2.internal]

TASK [role1 : Check if NGINX is installed] ************************************************************************************
ok: [ip-172-31-88-234.ec2.internal]
ok: [ip-172-31-89-26.ec2.internal]

TASK [role1 : NGINX test result] **********************************************************************************************
ok: [ip-172-31-88-234.ec2.internal] => {
    "msg": "NGINX found"
}
skipping: [ip-172-31-89-26.ec2.internal]

TASK [role1 : NGINX test result] **********************************************************************************************
ok: [ip-172-31-88-234.ec2.internal] => {
    "msg": "NGINX NOT found"
}
skipping: [ip-172-31-89-26.ec2.internal]

PLAY RECAP ********************************************************************************************************************
ip-172-31-88-234.ec2.internal : ok=4    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
ip-172-31-89-26.ec2.internal : ok=2    changed=0    unreachable=0    failed=0    skipped=2    rescued=0    ignored=0

ubuntu@ip-172-31-80-191:/etc/ansible$
```