JENKINS ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Multi-Node Jenkins Setup for Test & Prod Branch-Based Deployments

---

Problem Statement

Set up a Jenkins Master with two remote nodes (Test and Prod).

Configure Jenkins such that:

- When code is pushed to the test branch → Deploy to Test Server
- When code is pushed to the master branch → Deploy to Prod Server

The setup includes EC2 creation, node configuration, job creation, deployment logic, and GitHub webhook integration.

Environment Used

Three Ubuntu EC2 instances launched using the same key pair

Jenkins-Master

Test-Node

Prod-Node

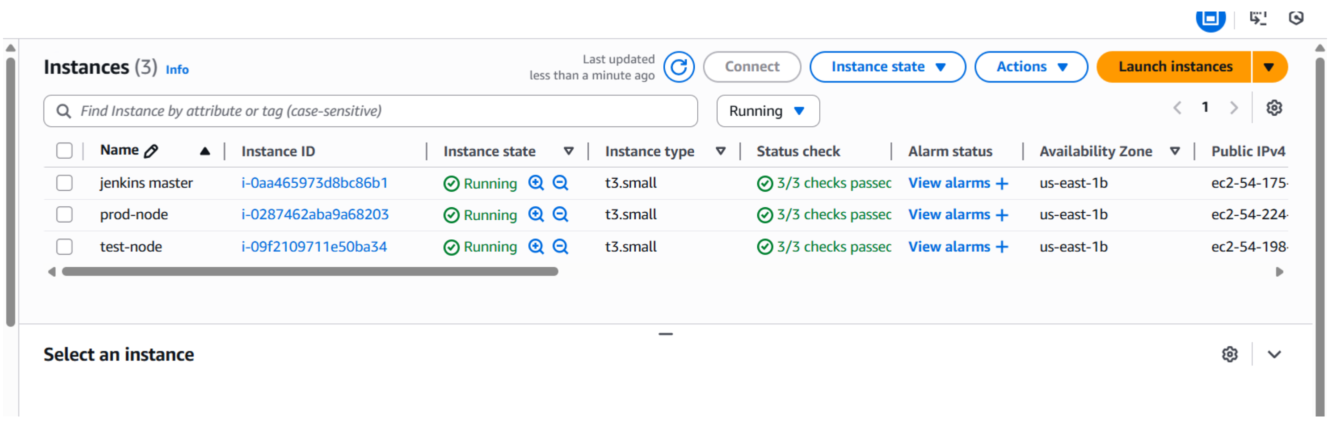
All three were connected through EC2 Instance Connect**,** and Java 17 + Git were installed on all instances.

Tasks Performed

□ TASK 1: Launch EC2 Instances

Steps Taken

- Launched 3 Ubuntu EC2 instances:
    - Jenkins-Master
    - Test-Node
    - Prod-Node
- All three instances were launched using the same key pair
- Connected to each instance using EC2 Instance Connect



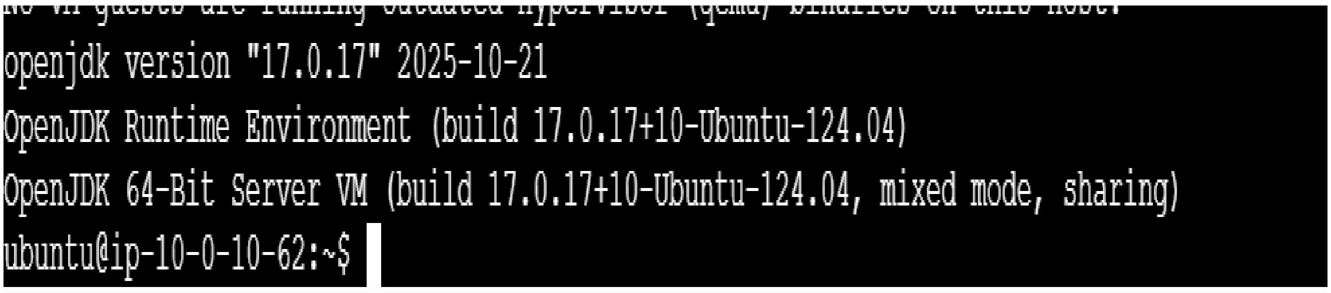
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
jenkins master	i-0aa465973d8bc86b1	Running	t3.small	3/3 checks passec	View alarms +	us-east-1b	ec2-54-175-
prod-node	i-0287462aba9a68203	Running	t3.small	3/3 checks passec	View alarms +	us-east-1b	ec2-54-224-
test-node	i-09f2109711e50ba34	Running	t3.small	3/3 checks passec	View alarms +	us-east-1b	ec2-54-198-

---

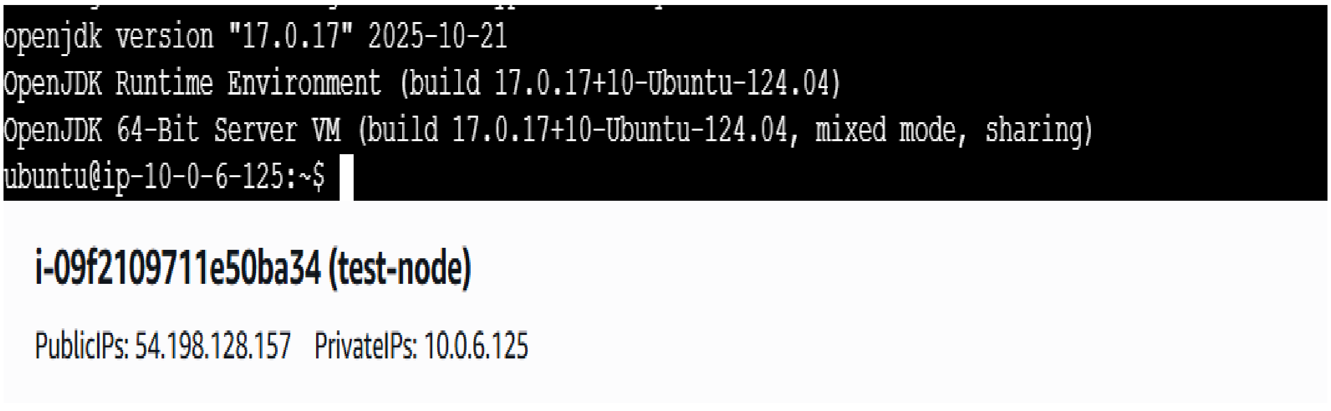
☐ TASK 2: Install Java 17 & Git on All Servers

Executed on all 3 instances:

```
sudo apt update -y
sudo apt install openjdk-17-jre git -y
java -version
```



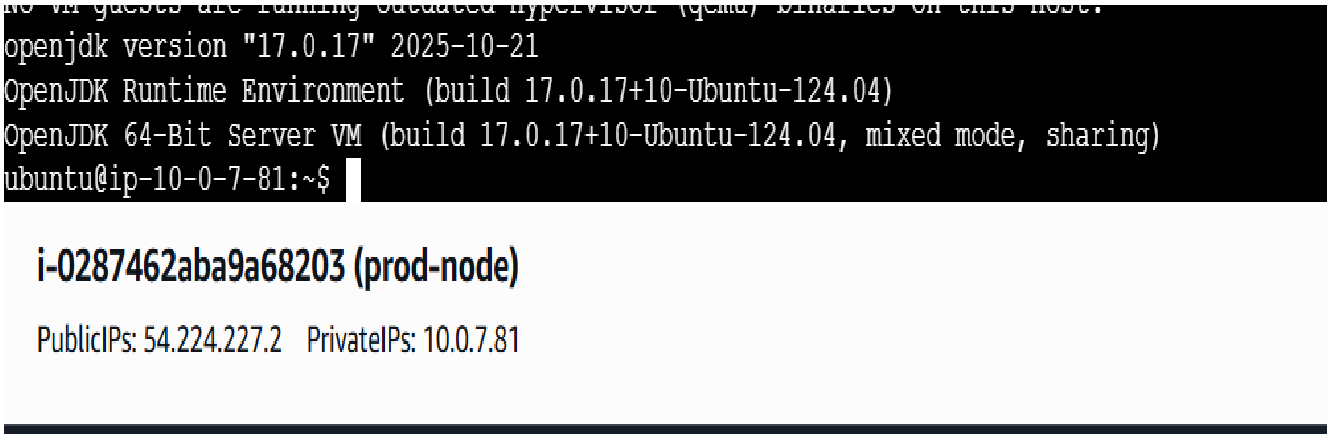
```
openjdk version "17.0.17" 2025-10-21
OpenJDK Runtime Environment (build 17.0.17+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.17+10-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-10-0-10-62:~$
```



```
openjdk version "17.0.17" 2025-10-21
OpenJDK Runtime Environment (build 17.0.17+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.17+10-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-10-0-6-125:~$
```

i-09f2109711e50ba34 (test-node)

PublicIPs: 54.198.128.157 PrivateIPs: 10.0.6.125



```
openjdk version "17.0.17" 2025-10-21
OpenJDK Runtime Environment (build 17.0.17+10-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.17+10-Ubuntu-124.04, mixed mode, sharing)
ubuntu@ip-10-0-7-81:~$
```

i-0287462aba9a68203 (prod-node)

PublicIPs: 54.224.227.2 PrivateIPs: 10.0.7.81

□ TASK 3: Install Jenkins on Jenkins-Master

Performed on the **Jenkins-Master** instance:

```
curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null


echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null


sudo apt update
sudo apt install jenkins -y
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

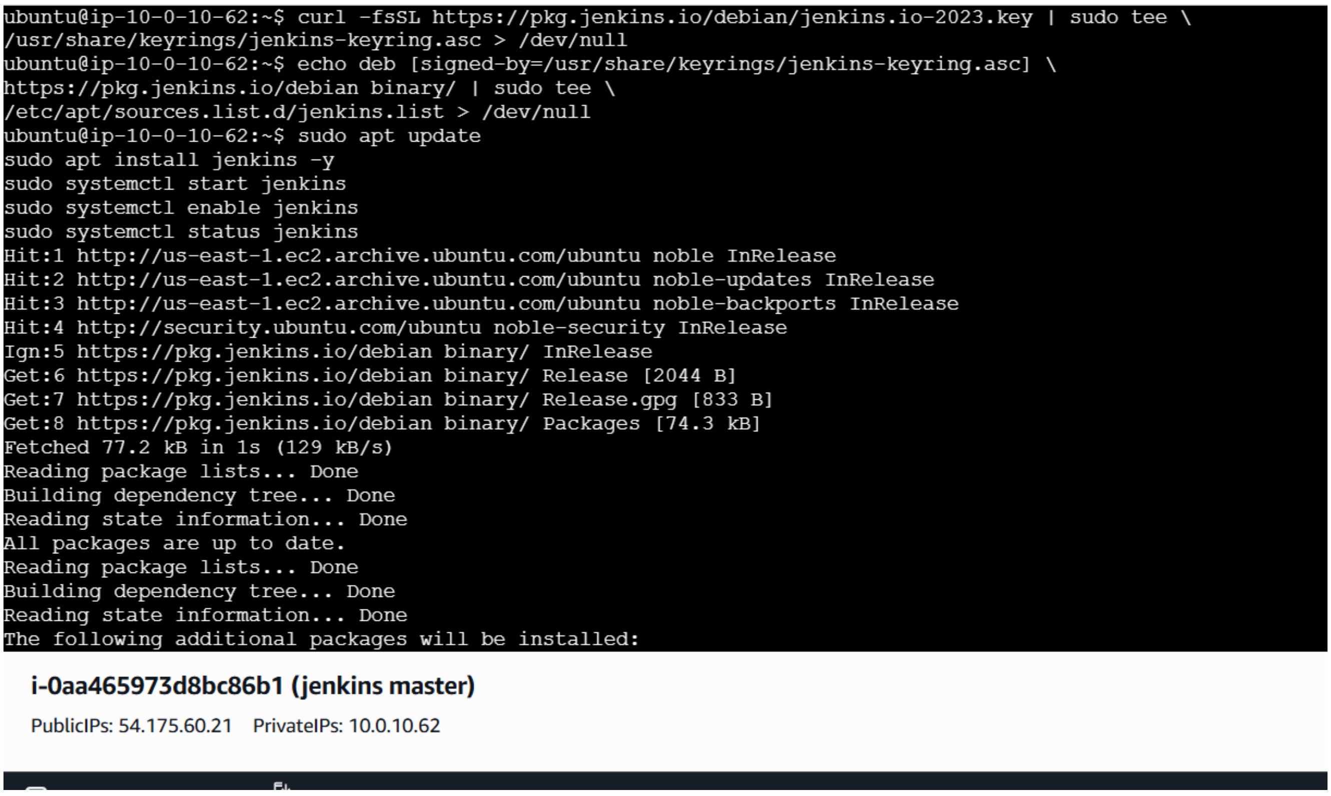
Access Jenkins:

```
http://<JENKINS_MASTER_PUBLIC_IP>:8080
```

Retrieve password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

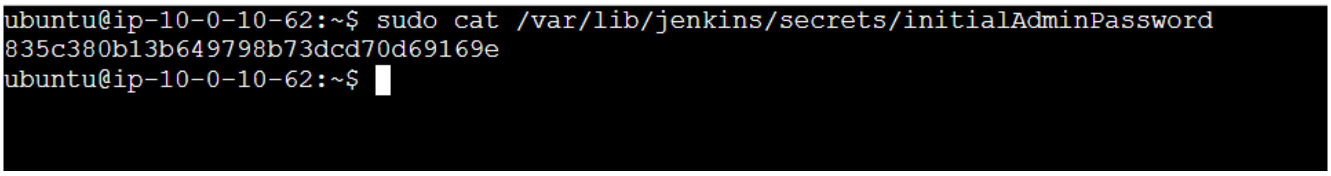
Installed suggested plugins.



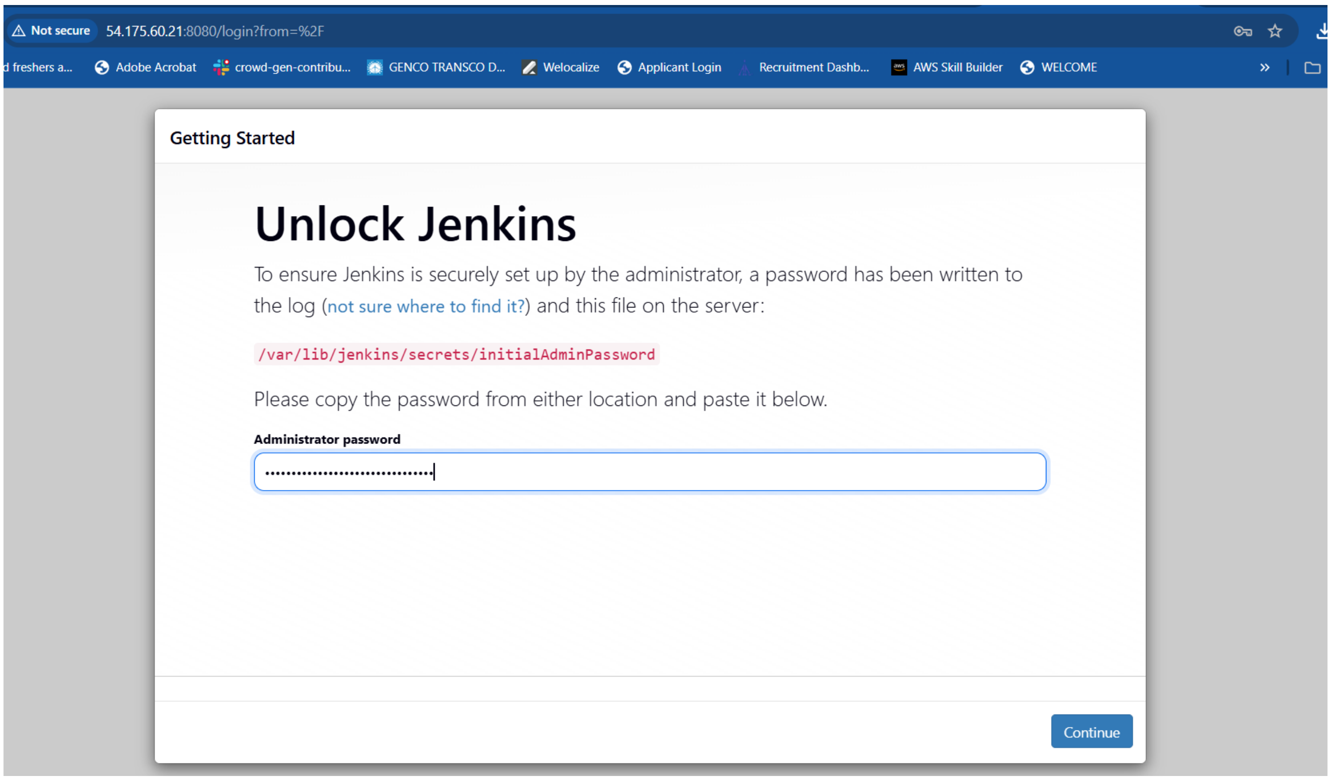
```
ubuntu@ip-10-0-10-62:~$ curl -fsSL https://pkg.jenkins.io/debian/jenkins.io-2023.key | sudo tee \
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
ubuntu@ip-10-0-10-62:~$ echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
https://pkg.jenkins.io/debian binary/ | sudo tee \
/etc/apt/sources.list.d/jenkins.list > /dev/null
ubuntu@ip-10-0-10-62:~$ sudo apt update
sudo apt install jenkins -y
sudo systemctl start jenkins
sudo systemctl enable jenkins
sudo systemctl status jenkins
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:5 https://pkg.jenkins.io/debian binary/ InRelease
Get:6 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian binary/ Packages [74.3 kB]
Fetched 77.2 kB in 1s (129 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

i-0aa465973d8bc86b1 (jenkins master)

PublicIPs: 54.175.60.21   PrivateIPs: 10.0.10.62



```
ubuntu@ip-10-0-10-62:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
835c380b13b649798b73dcd70d69169e
ubuntu@ip-10-0-10-62:~$
```



Getting Started

Unlock Jenkins

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

`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

....

□ TASK 4: Configure SSH Access for Nodes

Since all 3 instances use the SAME keypair

we use that key in Jenkins.

□Copy your keypair contents into Jenkins:

Open your key pair pem file → copy full text:

-----BEGIN RSA PRIVATE KEY-----

...

-----END RSA PRIVATE KEY-----

□Add Credentials in Jenkins:

Jenkins Dashboard →

Manage Jenkins →

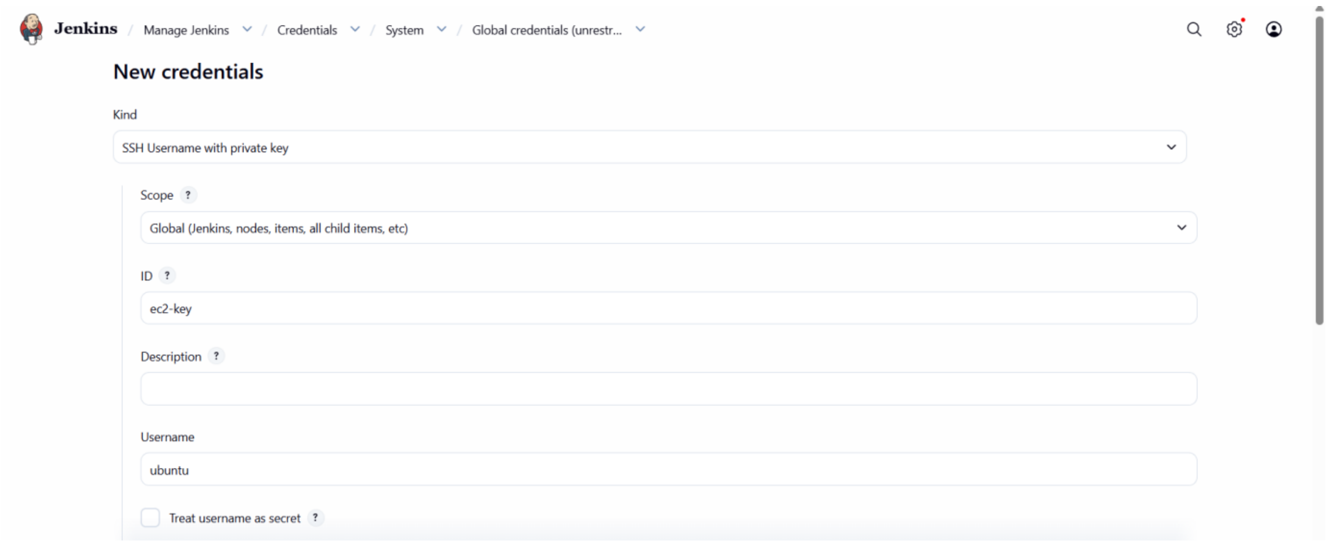
Credentials →

Global → Add Credentials

- Kind: SSH Username with private key
- Username: ubuntu

- Private Key: Enter directly → paste key
- ID: ec2-key
- Save

This lets Jenkins SSH into Test & Prod nodes.


Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

New credentials

Kind: SSH Username with private key

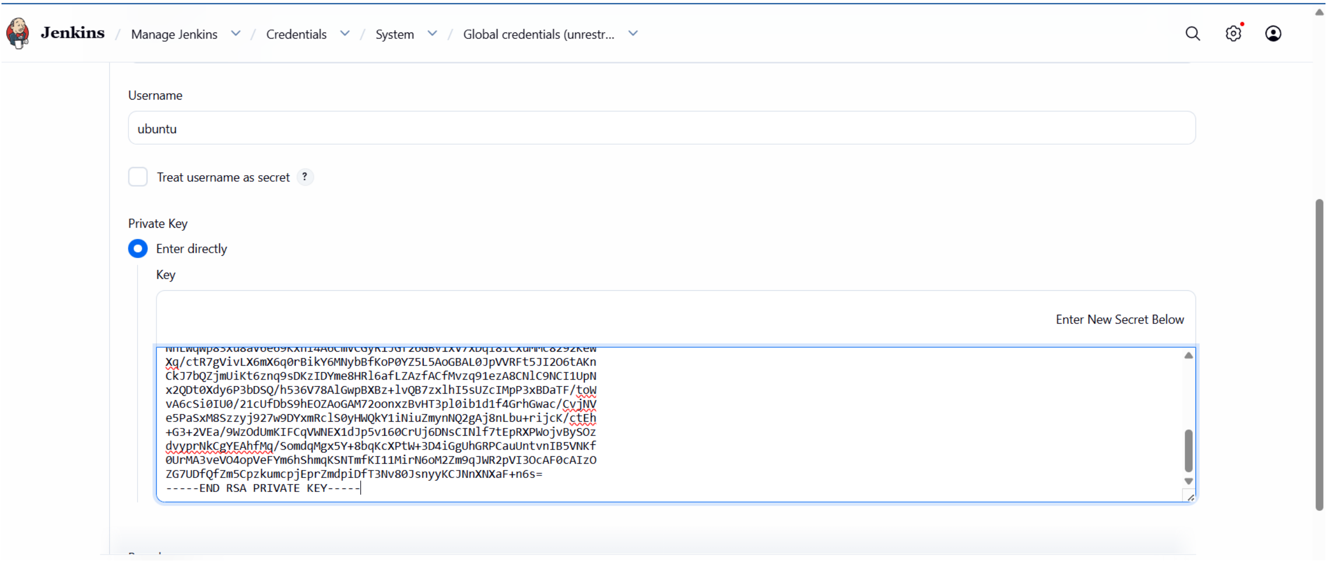
Scope: Global (Jenkins, nodes, items, all child items, etc)

ID: ec2-key

Description:

Username: ubuntu

Treat username as secret


Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...

Username: ubuntu

Treat username as secret

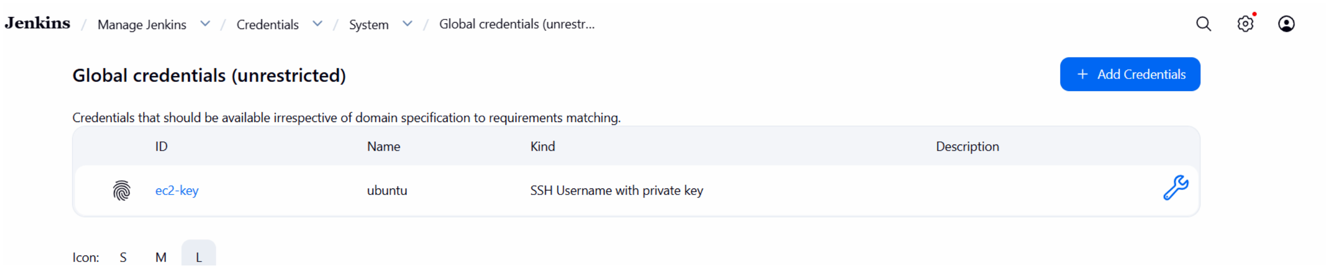
Private Key

● Enter directly

Key

Enter New Secret Below

```
Xq/ctR7gVivLX6mX6q0rBikY6MNybBfKoP0YZ5L5AoGBAL0JpVVRFt5JI2O6tAKn
CkJ7bQZjmUiKt6znq9sDKzIDYme8HRl6afLZAzfACfMvzq91ezA8CNlC9NCI1UpN
x2QDt0Xdy6P3bDSQ/h536V78AlGwpBXBz+lvQB7zxlhI5sUZcIMpP3xBDaTF/toW
vA6cSi0IU0/21cUfDbS9hEOZAoGAM72oonxzBvHT3pl0ib1d1f4GrhGwac/CvjNV
e5PaSxM8Szzyj927w9DYxmRclS0yHWQkY1iNiuZmynNQ2gAj8nLbu+rijcK/ctEh
+G3+2VEa/9WzOdUmKIFCqVWNEX1dJp5v160CrUj6DNsCINlf7tEpRXPWojvBySOz
dvyprNkCgYEAhfMq/SomdqMgx5Y+8bqKcXPtW+3D4iGgUhGRPCauUntvnIB5VNKf
0UrMA3veVO4opVeFYm6hShmqKSNTmfKI11MirN6oM2Zm9qJWR2pVI3OcAF0cAIzO
ZG7UDfQfZm5CpzkumcpjEprZmdpiDfT3Nv80JsnyyKCJNnXNXaF+n6s=
-----END RSA PRIVATE KEY-----
```


Jenkins / Manage Jenkins / Credentials / System / Global credentials (unrestr...
+ Add Credentials

Global credentials (unrestricted)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
ec2-key	ubuntu	SSH Username with private key	

Icon: S M L

□ TASK 6: Add Nodes to Jenkins Master

Go to:

Manage Jenkins → Nodes → New Node

✓ Node 1: test-node

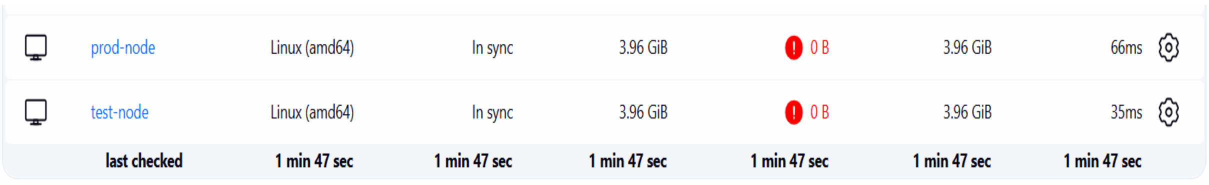
- Name: test-node
- Type: Permanent Agent
- Remote Root Directory: /home/ubuntu/jenkins
- Launch Method: SSH
- Host: Private IP of Test-Node
- Credentials: ubuntu
- Host Key Verification → Non-verifying

Save → It should show online

✓ Node 2: prod-node

- Same settings
- Host: Private IP of Prod-Node
- Credentials: ubuntu

Save → It should show online



	prod-node	Linux (amd64)	In sync	3.96 GiB	❗ 0 B	3.96 GiB	66ms	
	test-node	Linux (amd64)	In sync	3.96 GiB	❗ 0 B	3.96 GiB	35ms	
	last checked	1 min 47 sec	1 min 47 sec	1 min 47 sec	1 min 47 sec	1 min 47 sec	1 min 47 sec	

□ TASK 7: Create Jenkins Job – push-to-test (Triggered by test branch)

Jenkins Dashboard → New Item → Freestyle Project → Name:

push-to-test

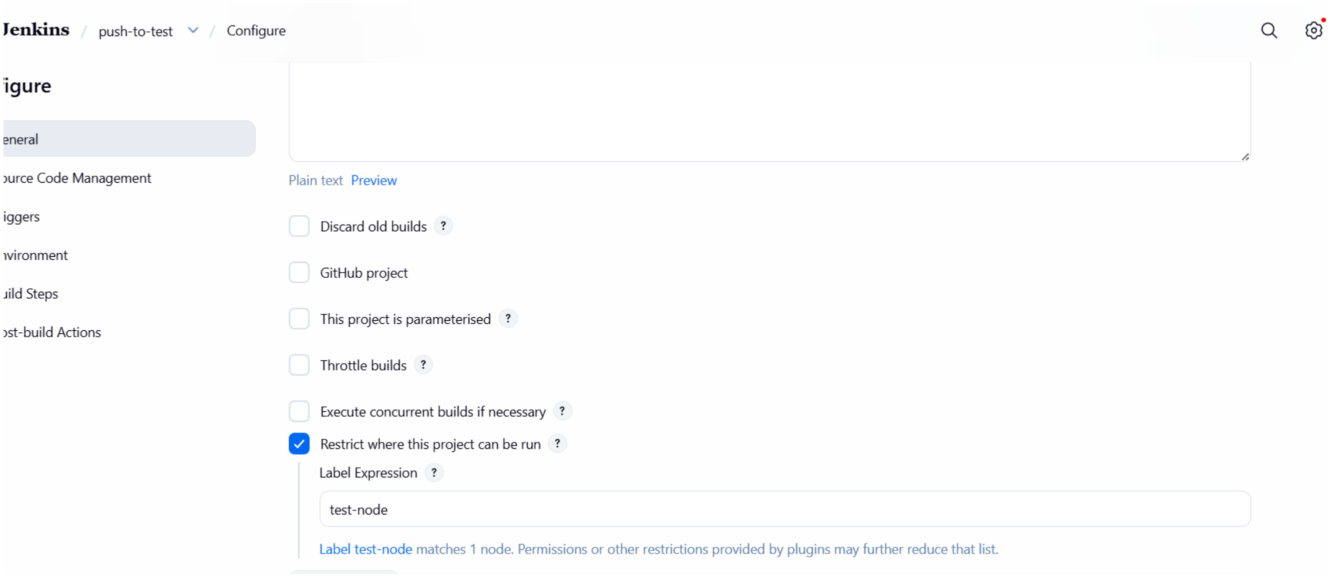
Settings:

- Restrict job to run on: test-node
- SCM → Git
  - Repo URL: your GitHub repo
  - Branch:
  - */test
- Build Trigger:

- GitHub hook trigger for GITScm polling
- Build Step → Execute Shell:

```
echo "Deploying to Test Server"
mkdir -p /home/ubuntu/test-deploy
rm -rf /home/ubuntu/test-deploy/*
cp -r * /home/ubuntu/test-deploy/
chown -R ubuntu:ubuntu /home/ubuntu/test-deploy
```

Save.



Jenkins / push-to-test / Configure

igure

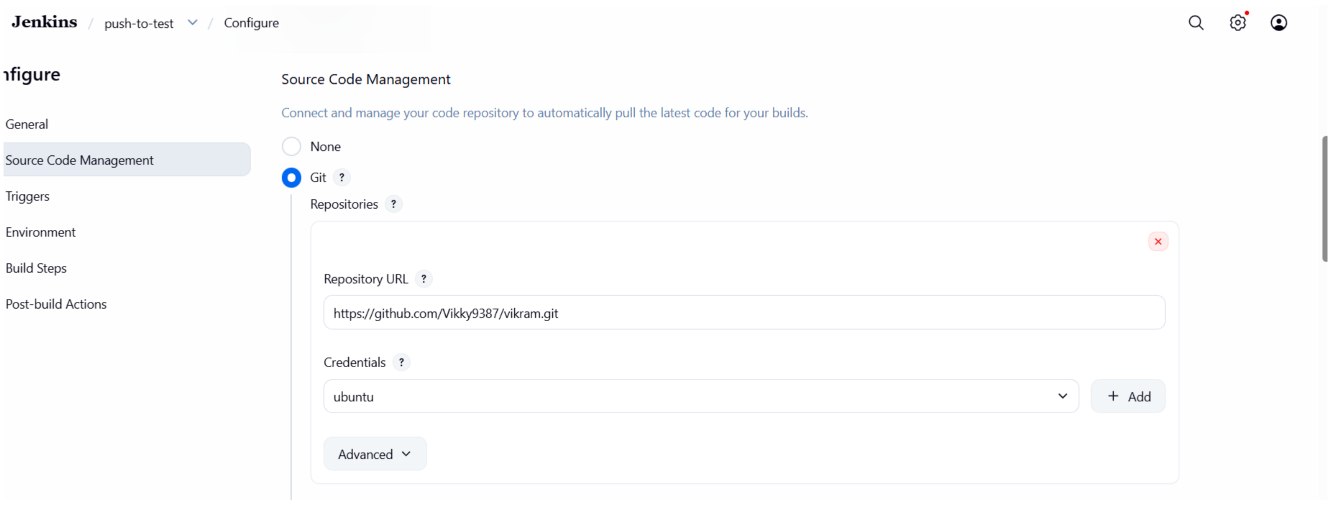
eneral
ource Code Management
iggers
nvironment
uild Steps
ost-build Actions

Plain text Preview

[ ] Discard old builds
[ ] GitHub project
[ ] This project is parameterised
[ ] Throttle builds
[ ] Execute concurrent builds if necessary
[✓] Restrict where this project can be run

Label Expression: test-node

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



Jenkins / push-to-test / Configure

nfigure

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

Source Code Management

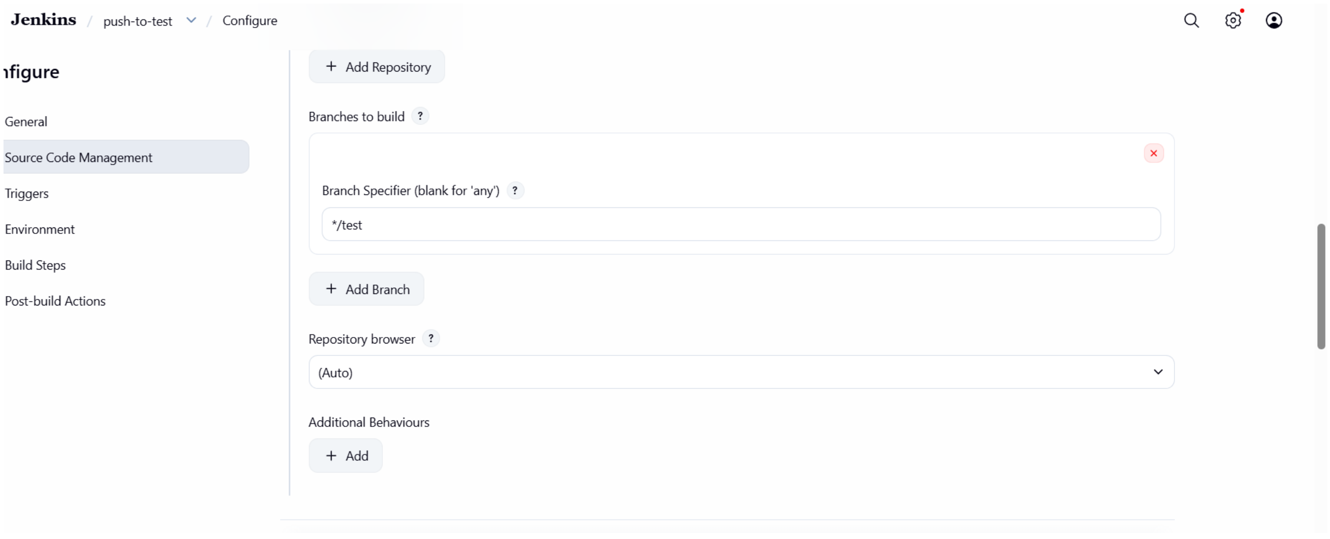
Connect and manage your code repository to automatically pull the latest code for your builds.

None [✓ Git]

Repositories

Repository URL: https://github.com/Vikky9387/vikram.git

Credentials: ubuntu


Jenkins / push-to-test / Configure

nfigure

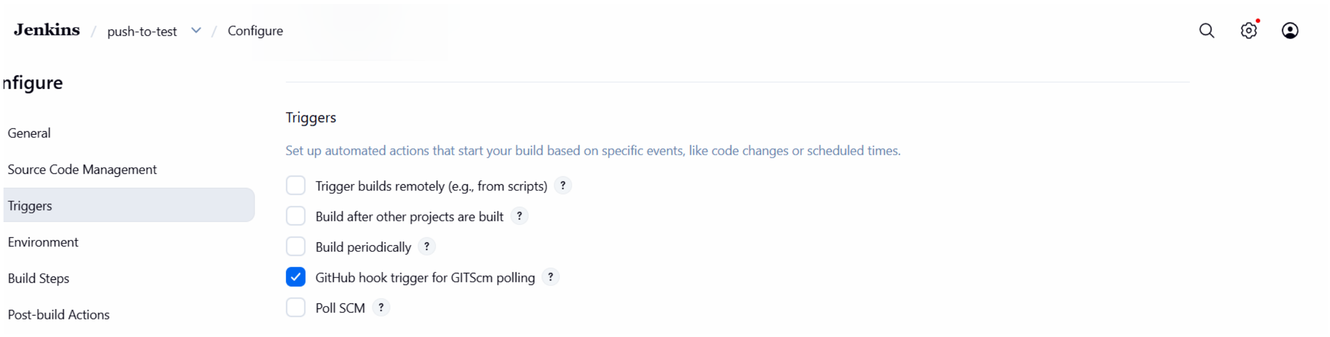
- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Branches to build

Branch Specifier (blank for 'any'): `*/test`

Repository browser: (Auto)

Additional Behaviours


Jenkins / push-to-test / Configure

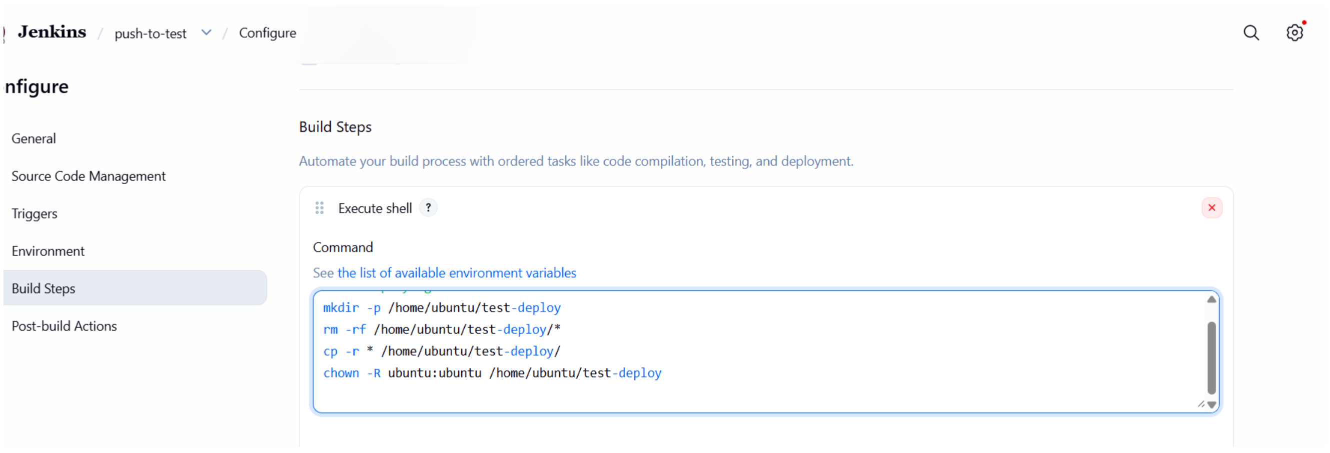
nfigure

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

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [x] GitHub hook trigger for GITScm polling
- [ ] Poll SCM


Jenkins / push-to-test / Configure

nfigure

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

Build Steps

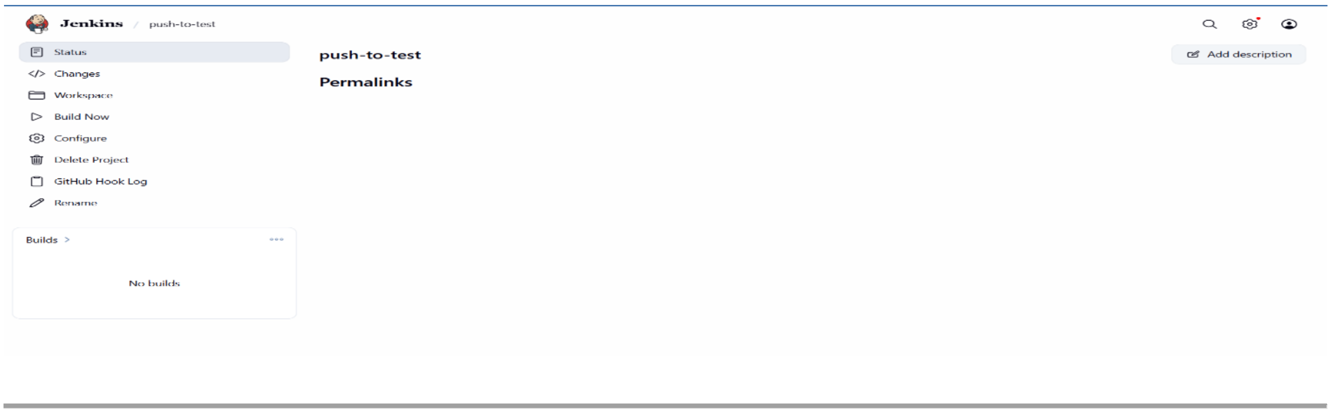
Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

Command

See the list of available environment variables

```
mkdir -p /home/ubuntu/test-deploy
rm -rf /home/ubuntu/test-deploy/*
cp -r * /home/ubuntu/test-deploy/
chown -R ubuntu:ubuntu /home/ubuntu/test-deploy
```


Jenkins / push-to-test

push-to-test

Permalinks

Builds

No builds

□ TASK 8: Create Jenkins Job – push-to-prod (Triggered by master branch)

New Item → Freestyle → Name:

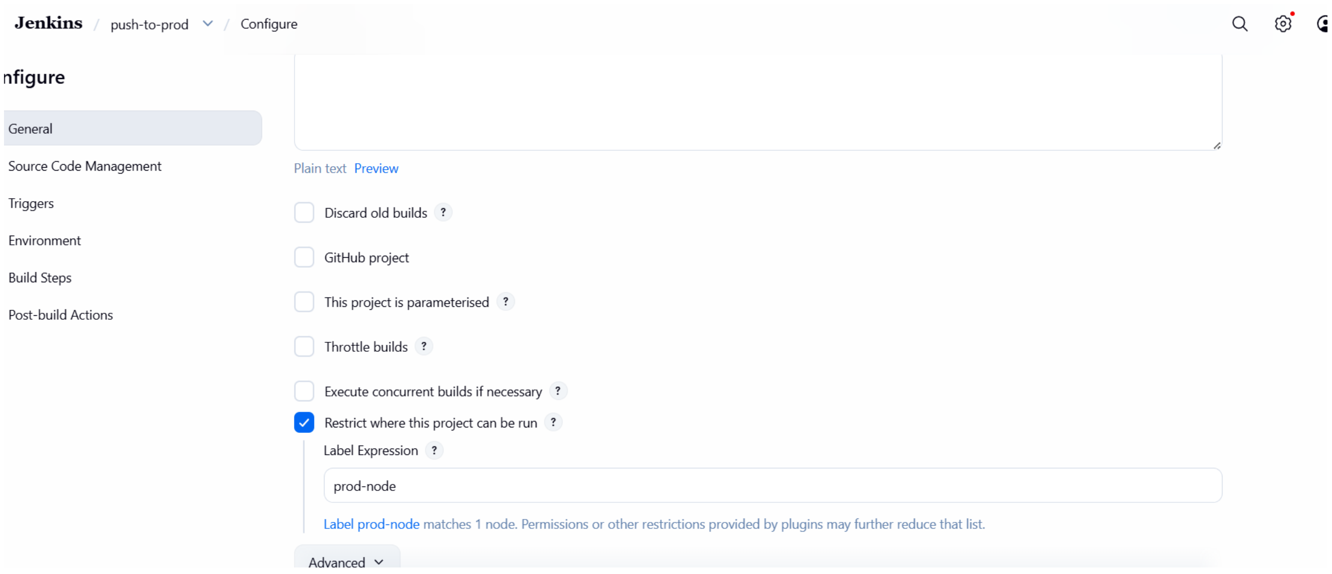
push-to-prod

Settings:

- Restrict job to: prod-node
- SCM → Git
  - Branch:
  - */master
- Build Trigger:
- GitHub hook trigger for GITScm polling
- Build Step → Execute Shell:

```
echo "Deploying to Prod Server"
mkdir -p /home/ubuntu/prod-deploy
rm -rf /home/ubuntu/prod-deploy/*
cp -r * /home/ubuntu/prod-deploy/
chown -R ubuntu:ubuntu /home/ubuntu/prod-deploy
```

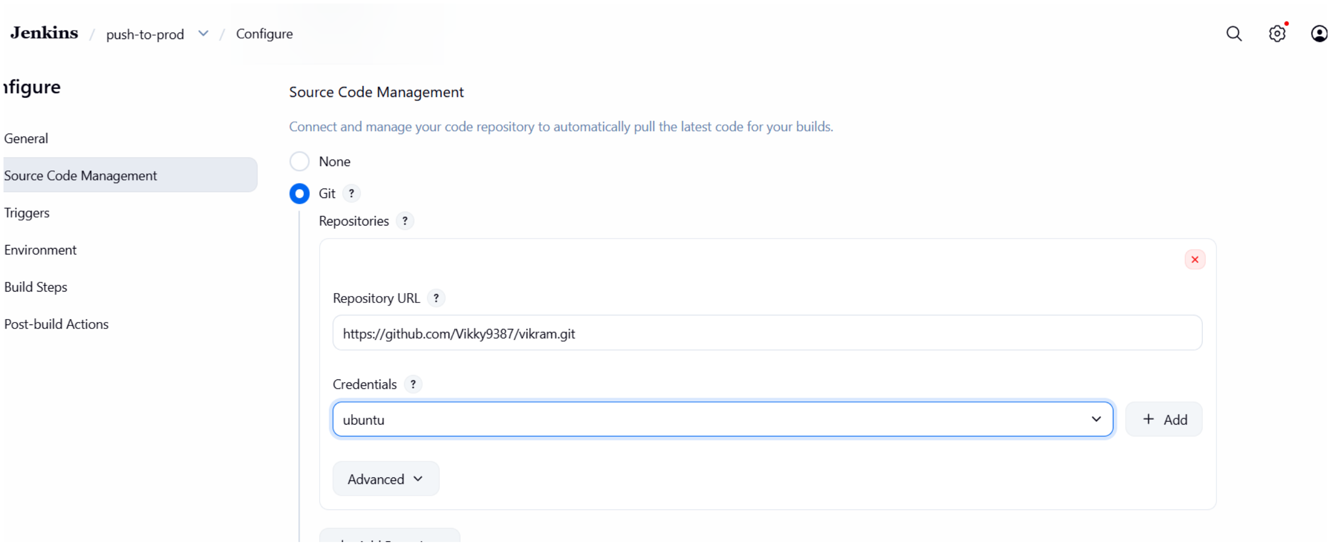
Save.



- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterised
- [ ] Throttle builds
- [ ] Execute concurrent builds if necessary
- [✓] Restrict where this project can be run

Label Expression: prod-node

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


Jenkins / push-to-prod / Configure

ıfigure

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

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

- None
- [✓ Git] ?

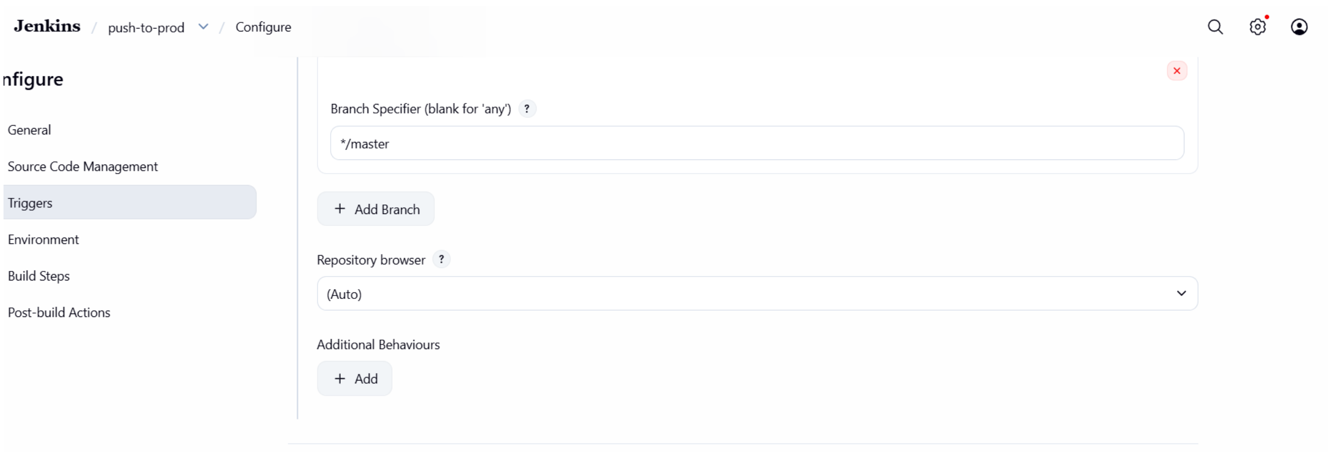
Repositories ?

Repository URL ?
https://github.com/Vikky9387/vikram.git

Credentials ?
ubuntu

Add

Advanced


Jenkins / push-to-prod / Configure

nfigure

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

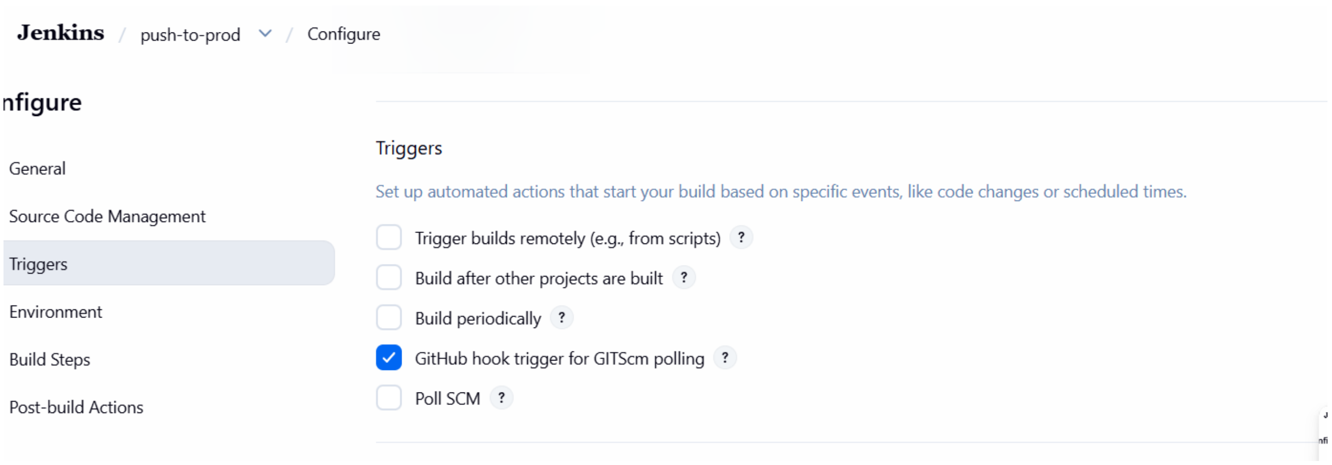
Branch Specifier (blank for 'any') ?
*/master

Add Branch

Repository browser ?
(Auto)

Additional Behaviours

Add


Jenkins / push-to-prod / Configure

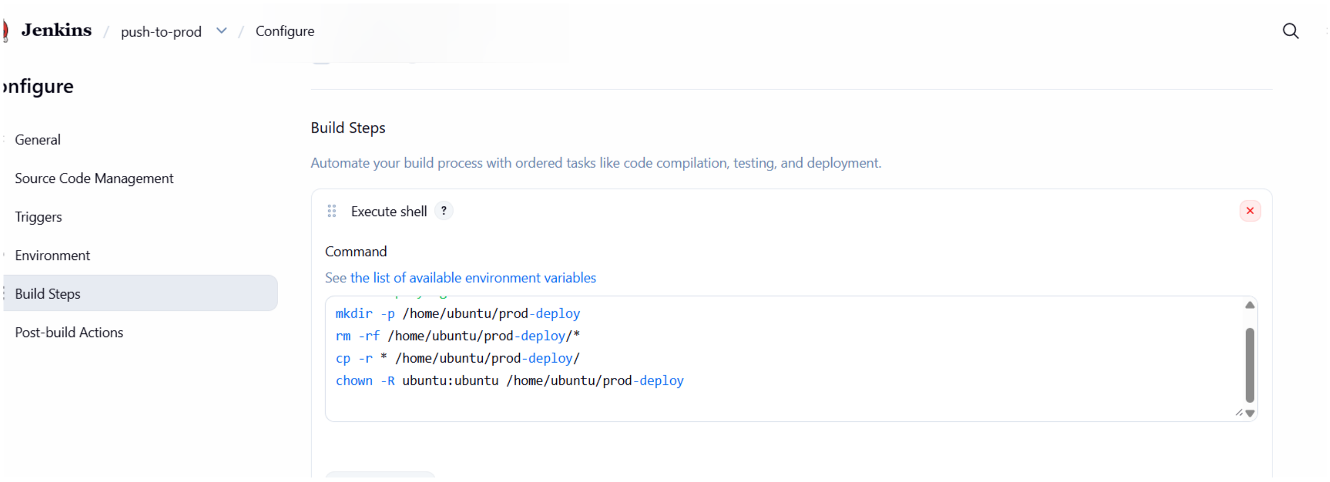
nfigure

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

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- [ ] Trigger builds remotely (e.g., from scripts) ?
- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [✓] GitHub hook trigger for GITScm polling ?
- [ ] Poll SCM ?


Jenkins / push-to-prod / Configure

Configure

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

Build Steps

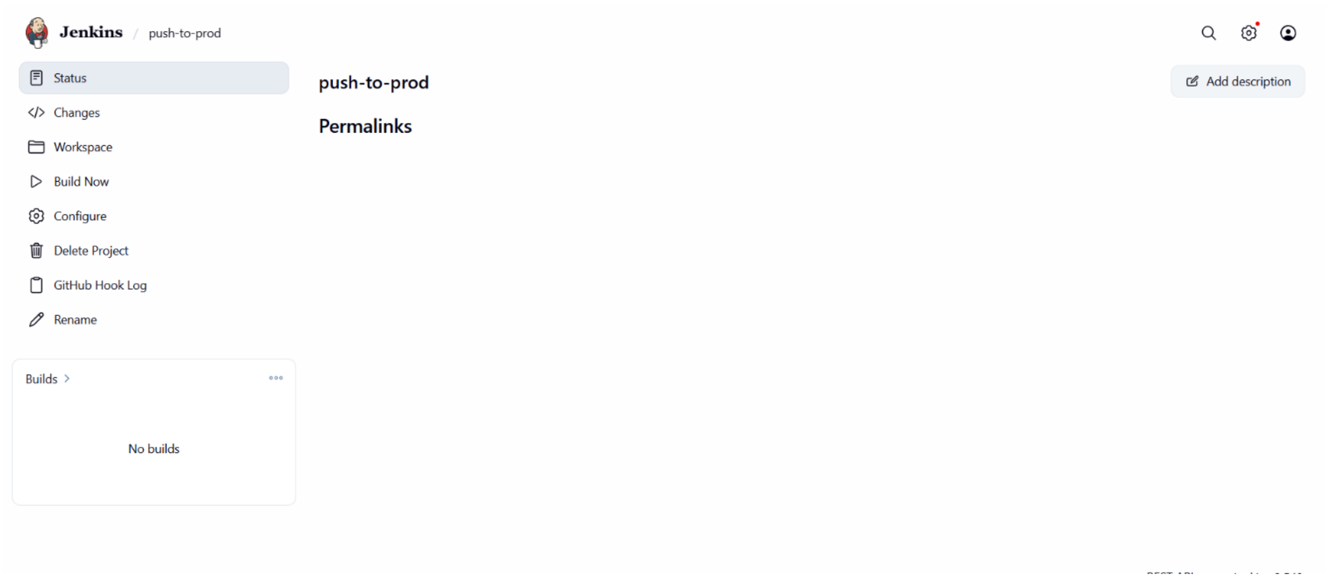
Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

Command

See the list of available environment variables

```
mkdir -p /home/ubuntu/prod-deploy
rm -rf /home/ubuntu/prod-deploy/*
cp -r * /home/ubuntu/prod-deploy/
chown -R ubuntu:ubuntu /home/ubuntu/prod-deploy
```


Jenkins / push-to-prod

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

push-to-prod

Add description

Permalinks

Builds

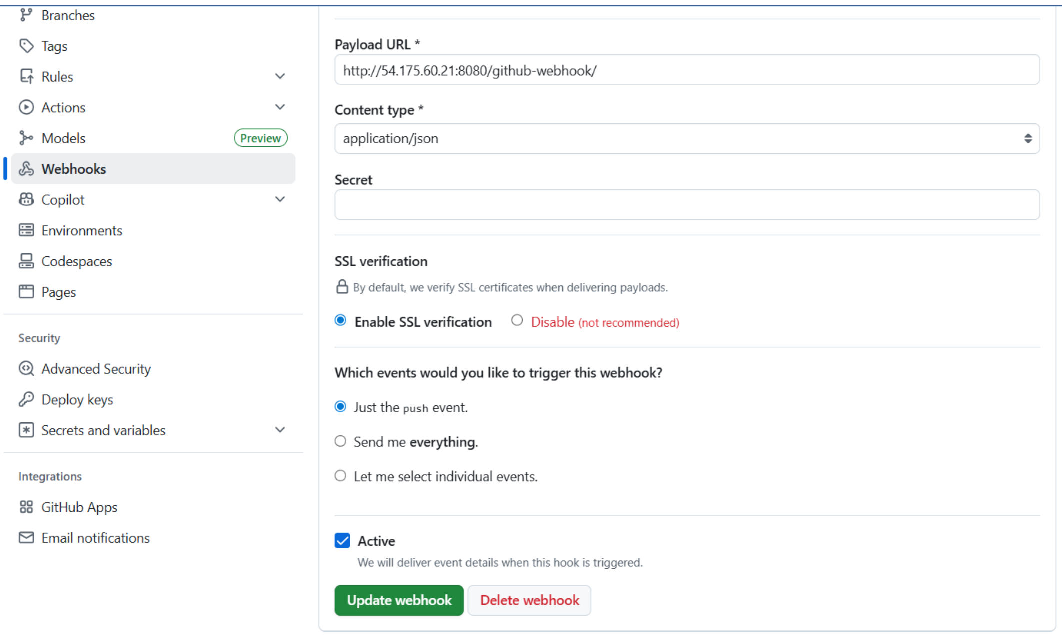
No builds

□ TASK 9: Configure GitHub Webhook

On GitHub → Repository → Settings → Webhooks → Add Webhook

- Payload URL:
- http://<JENKINS_MASTER_PUBLIC_IP>:8080/github-webhook/
- Content type: application/json
- Event: Just the push event

Add webhook.



Payload URL *
http://54.175.60.21:8080/github-webhook/

Content type *
application/json

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
[✓ Enable SSL verification] Disable (not recommended)

Which events would you like to trigger this webhook?
[✓ Just the push event.]
Send me everything.
Let me select individual events.

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

□ TASK 10: Test the Pipeline

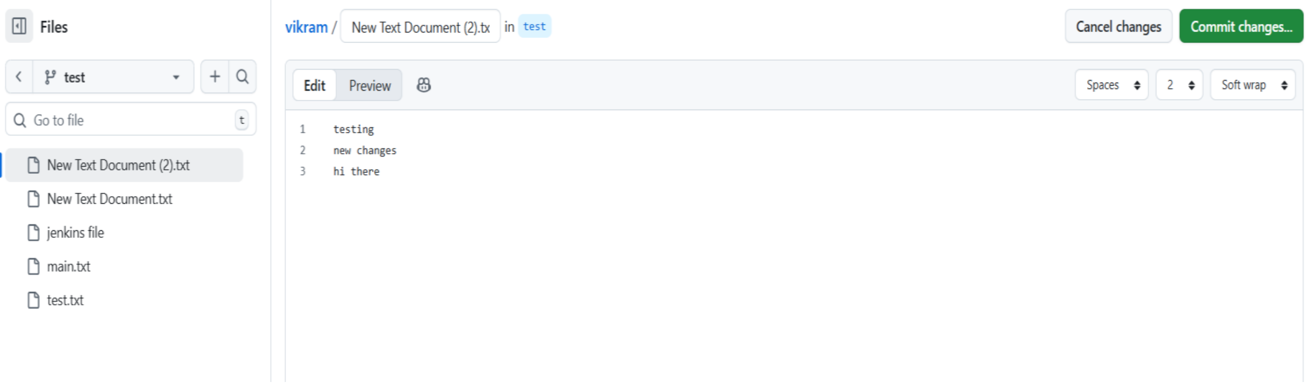
✓ Test Test-Node Deployment

On GitHub:

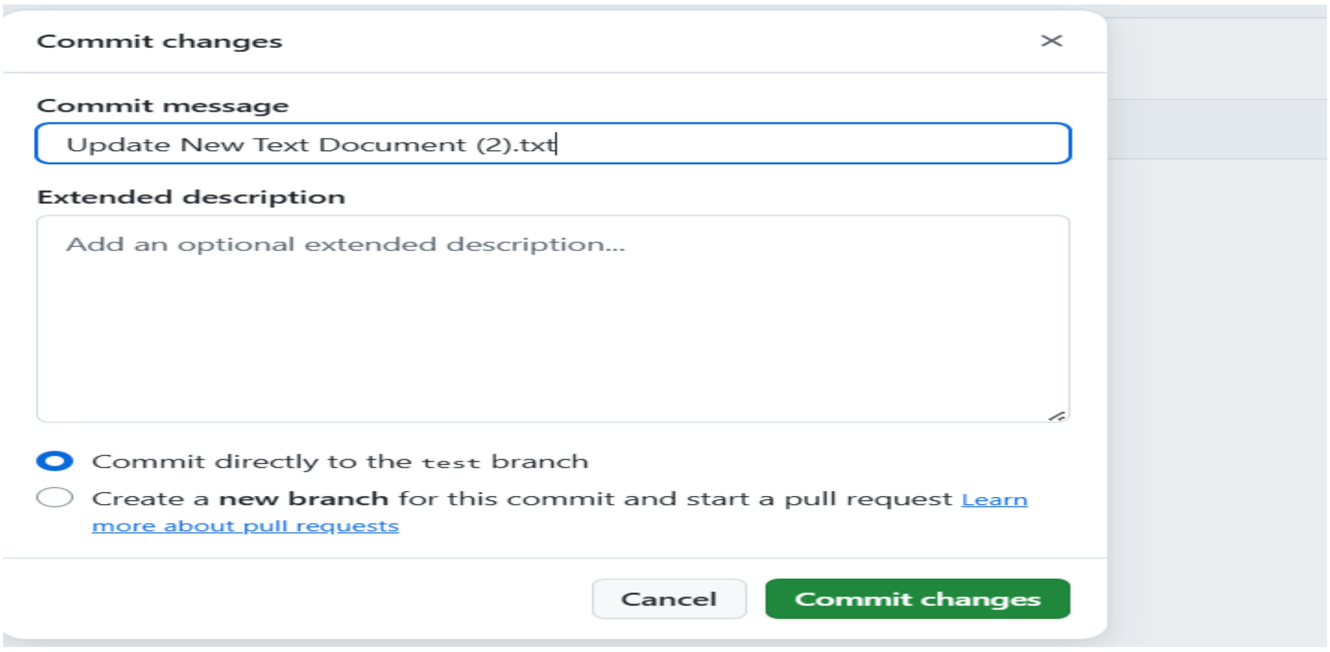
- Edit/create file
- Commit to test branch

Expected:

- Jenkins triggers push-to-test
- Files copied to:
- /home/ubuntu/test-deploy



```
testing
new changes
hi there
```


Commit changes

Commit message

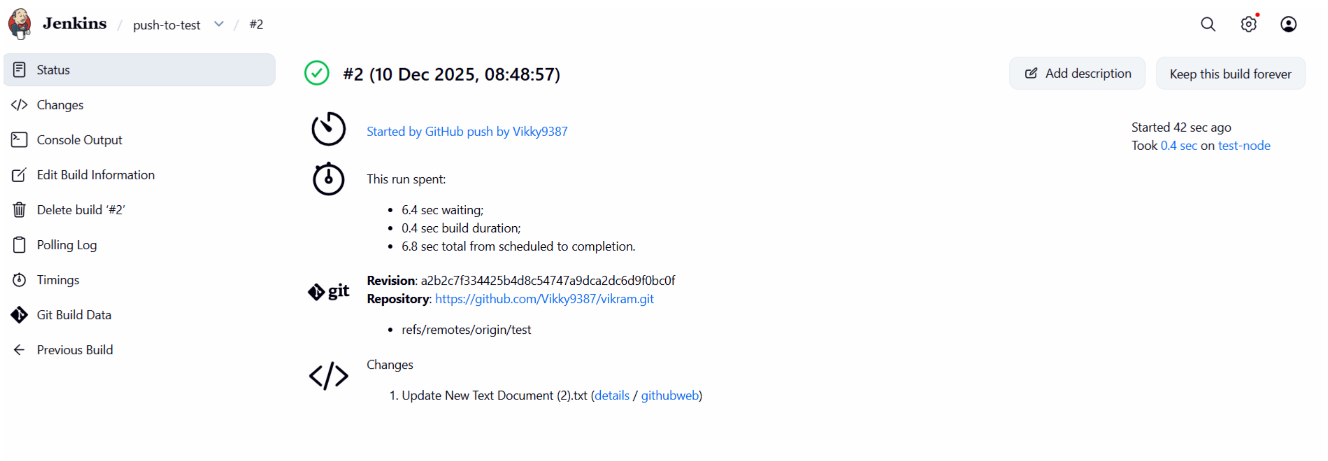
Update New Text Document (2).txt

Extended description

Add an optional extended description...

[✓ Commit directly to the test branch]

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



#2 (10 Dec 2025, 08:48:57)

Started by GitHub push by Vikky9387

Started 42 sec ago
Took 0.4 sec on test-node

This run spent:

- 6.4 sec waiting;
- 0.4 sec build duration;
- 6.8 sec total from scheduled to completion.

Revision: a2b2c7f334425b4d8c54747a9dca2dc6d9f0bc0f
Repository: https://github.com/Vikky9387/vikram.git

- refs/remotes/origin/test

Changes

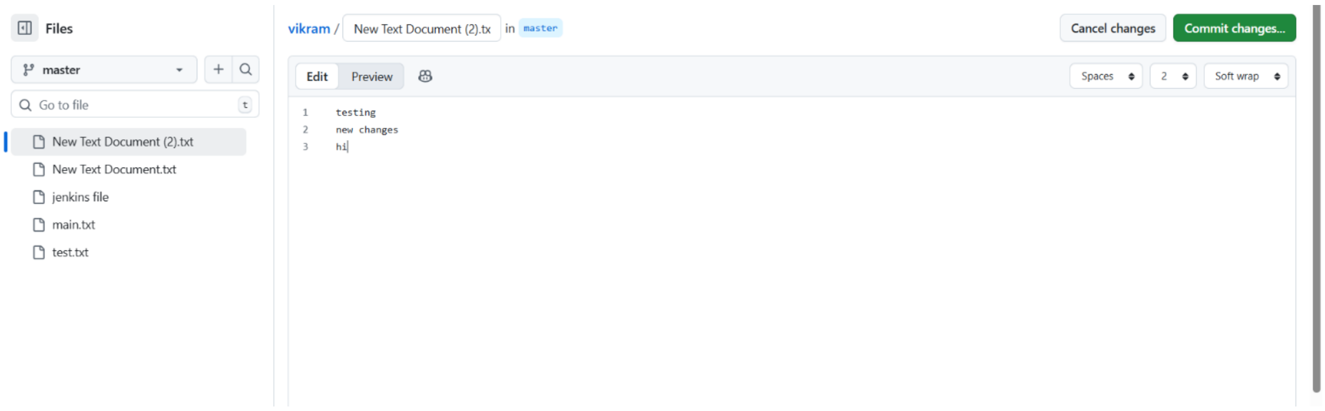
1. Update New Text Document (2).txt (details / githubweb)

✓ Test Prod-Node Deployment

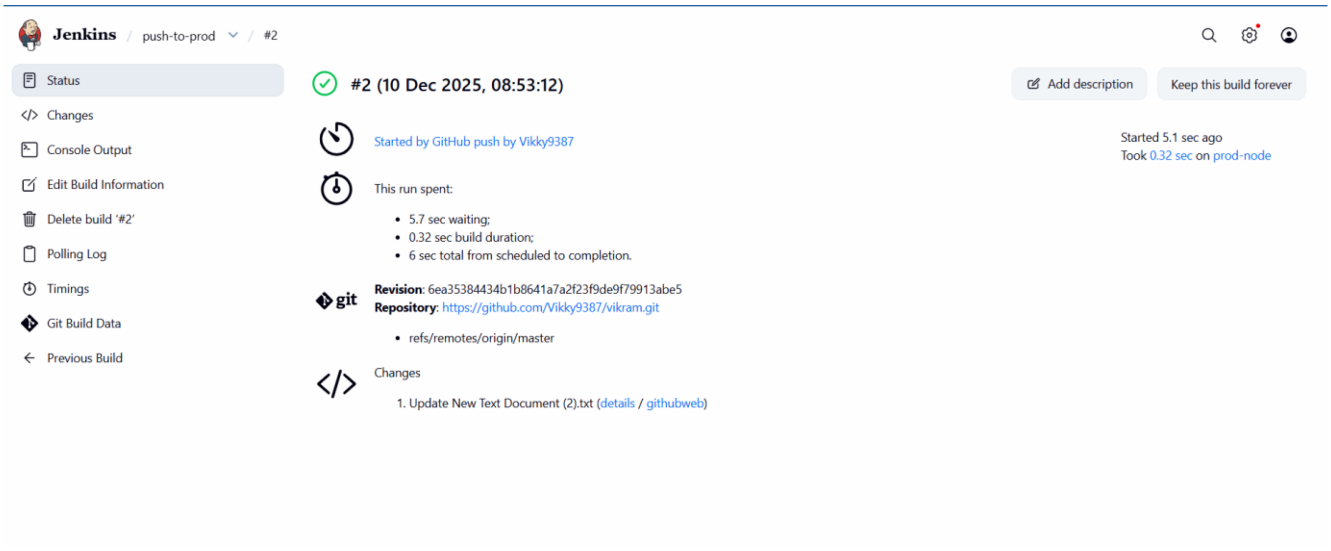
Commit to master branch

Expected:

- Jenkins triggers push-to-prod
- Files copied to:
- /home/ubuntu/prod-deploy



```
testing
new changes
hi
```


Jenkins / push-to-prod / #2

- Status
- Changes
- Console Output
- Edit Build Information
- Delete build '#2'
- Polling Log
- Timings
- Git Build Data
- Previous Build

#2 (10 Dec 2025, 08:53:12)

Add description | Keep this build forever

Started by GitHub push by Vikky9387

Started 5.1 sec ago
Took 0.32 sec on prod-node

This run spent:

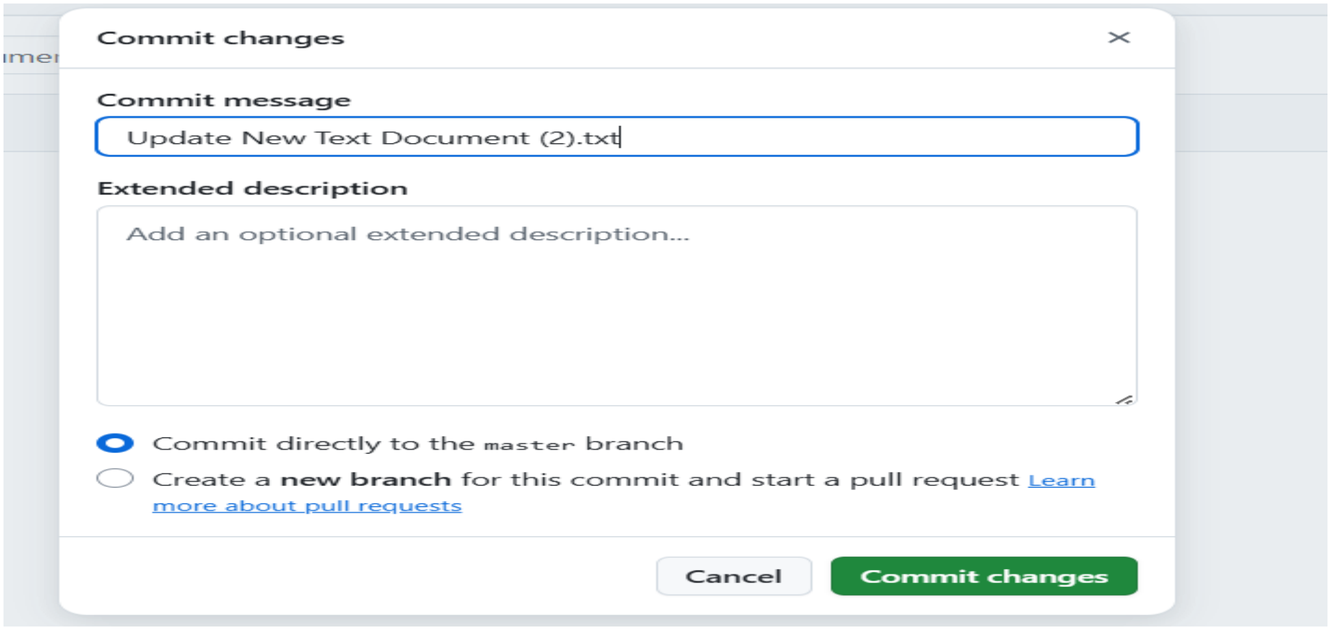
- 5.7 sec waiting;
- 0.32 sec build duration;
- 6 sec total from scheduled to completion.

Revision: 6ea35384434b1b8641a7a2f23f9de9f79913abe5
Repository: https://github.com/Vikky9387/vikram.git

- refs/remotes/origin/master

Changes

1. Update New Text Document (2).txt (details / githubweb)


Commit changes

Commit message

Update New Text Document (2).txt

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

□ Conclusion

Successfully launched 3 EC2 instances, installed Jenkins, added test & prod nodes, configured SSH using the same keypair, created two branch-based jobs, and implemented automatic deployments using GitHub Webhooks.

Test branch pushes deploy to Test Server; master branch pushes deploy to Prod Server.