JENKINS ASSIGNMENT SUBMISSION

Name: Vikram

Assignment: Jenkins Pipeline – Trigger Test Job → Trigger Prod Job → Deploy to Nodes

---

Problem Statement

Create a Jenkins Pipeline that:

1. Triggers automatically when code is pushed to the develop branch in GitHub
2. Runs the test job, which copies Git files to the Test Node
3. If the test job succeeds, automatically triggers the prod job
4. Prod job copies Git files to the Prod Node

This implements a real CI/CD pipeline with build → test → production stages.

Environment Used

Three Ubuntu EC2 instances:

- **Jenkins-Master** (Pipeline runs here)
- **Test-Node** (Deployment target for test job)
- **Prod-Node** (Deployment target for prod job)

All launched using the same keypair

All connected using EC2 Instance Connect

Jenkins has two freestyle jobs:

- test-job → deploys to Test Node
- prod-job → deploys to Prod Node

Tasks Performed

▯ TASK 1: Create Two Freestyle Jobs (test-job & prod-job)

✓ 1. test-job (deploy to test node)

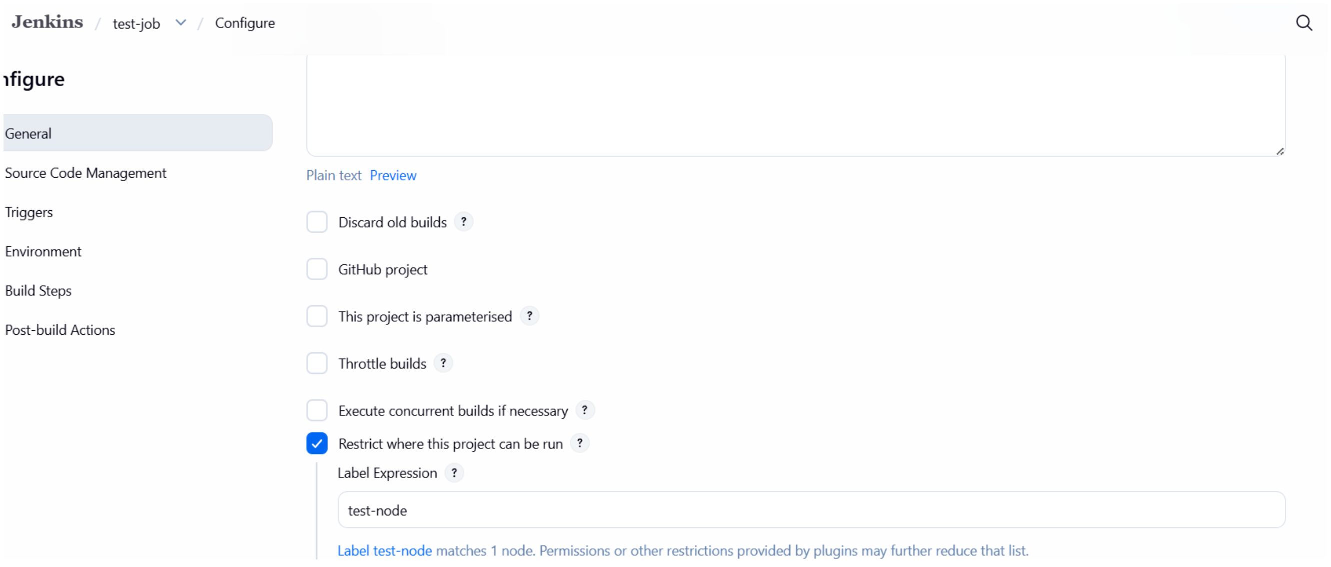
Jenkins → New Item → Freestyle Project → Name: test-job

Configuration:

- Restrict where to run: test-node
- SCM → Git → Branch:

- */develop
- Build Step → Execute Shell:

```
echo "Deploying to Test Node"
mkdir -p /home/ubuntu/test-deploy
rm -rf /home/ubuntu/test-deploy/*
cp -r * /home/ubuntu/test-deploy/
```



Jenkins / test-job / Configure

ıfigure

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

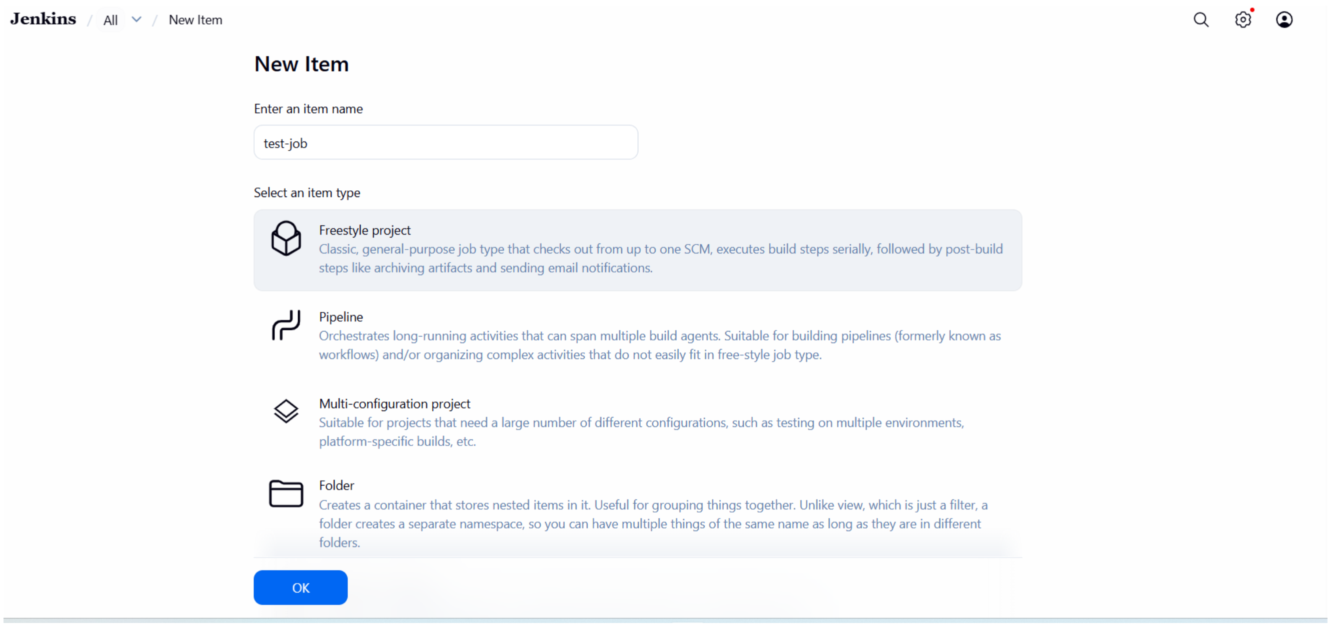
Plain text Preview

- [ ] 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 / All / New Item

New Item

Enter an item name

test-job

Select an item type

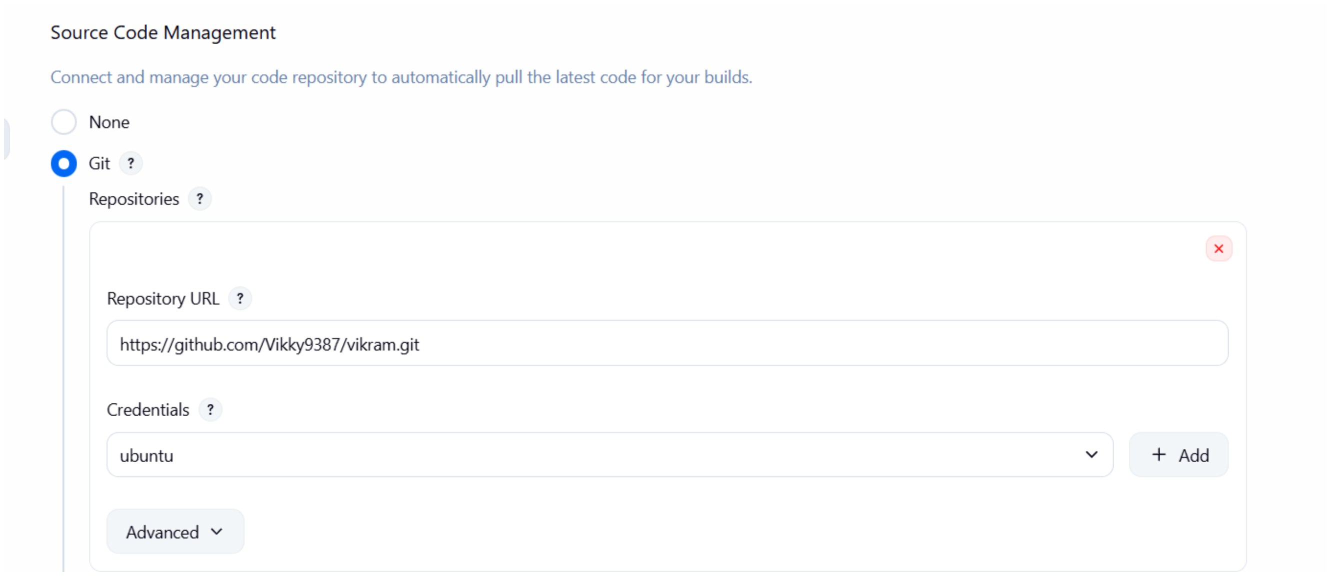
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

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

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

OK



Source Code Management

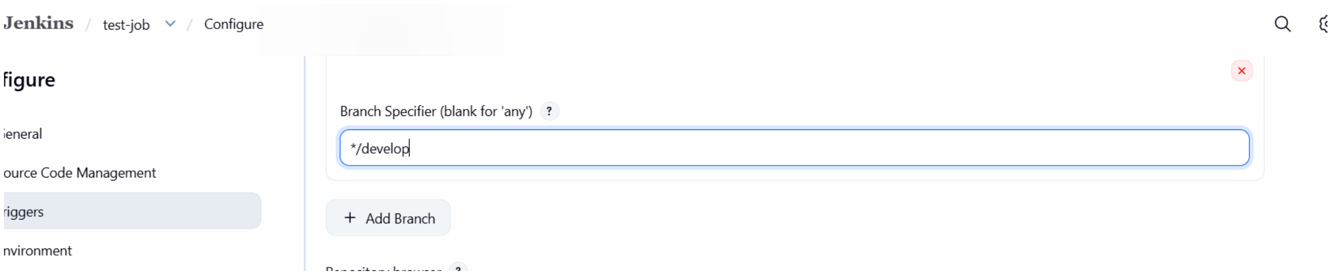
Connect and manage your code repository to automatically pull the latest code for your builds.

None [✓ Git]

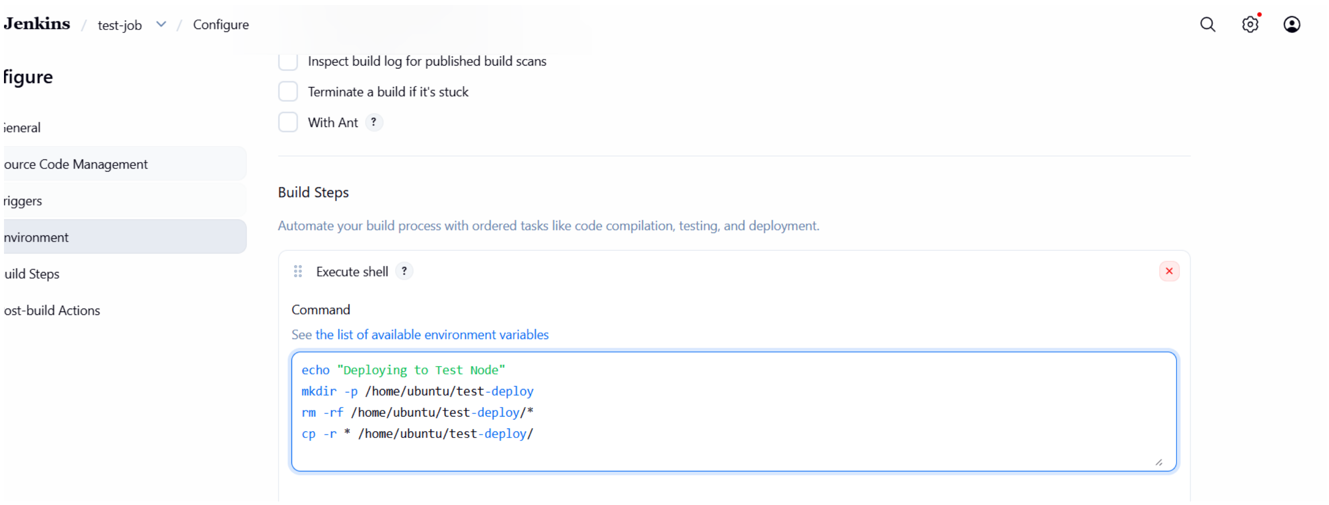
Repositories

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

Credentials: ubuntu



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



Inspect build log for published build scans

Terminate a build if it's stuck

With Ant

Build Steps

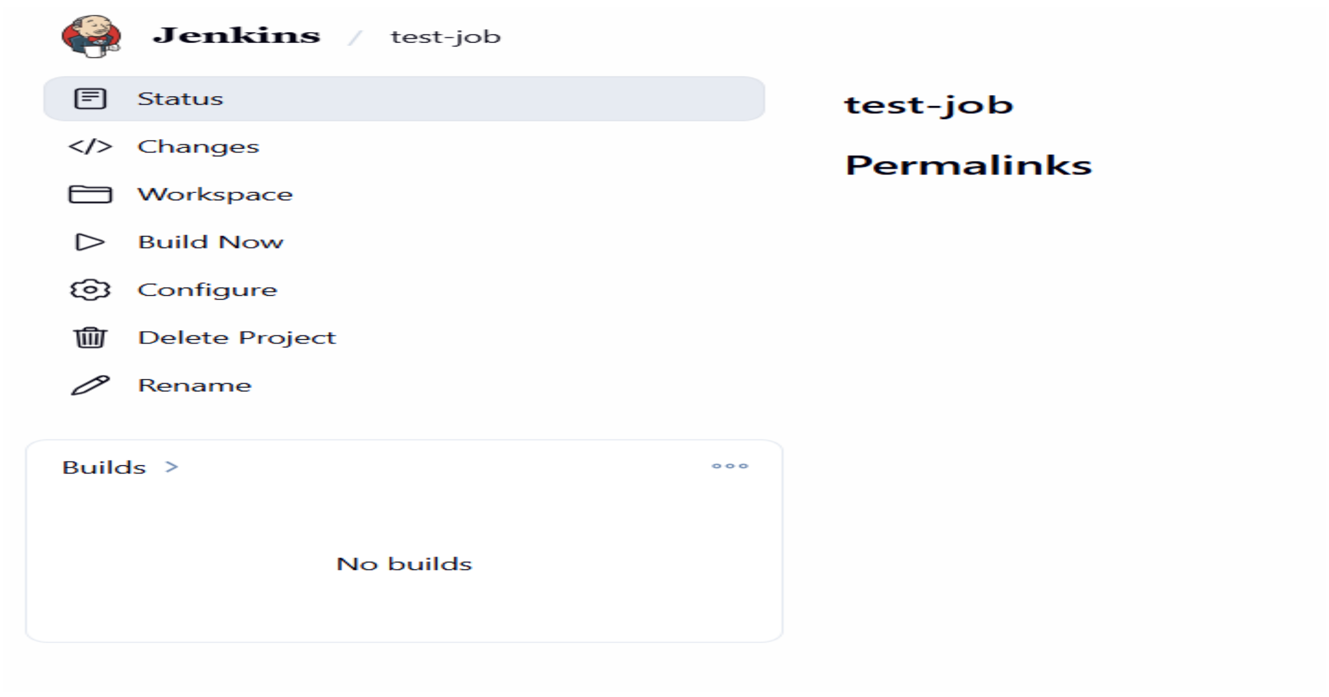
Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

Command

See the list of available environment variables

```
echo "Deploying to Test Node"
mkdir -p /home/ubuntu/test-deploy
rm -rf /home/ubuntu/test-deploy/*
cp -r * /home/ubuntu/test-deploy/
```


Jenkins / test-job

test-job

Permalinks

Builds >

No builds

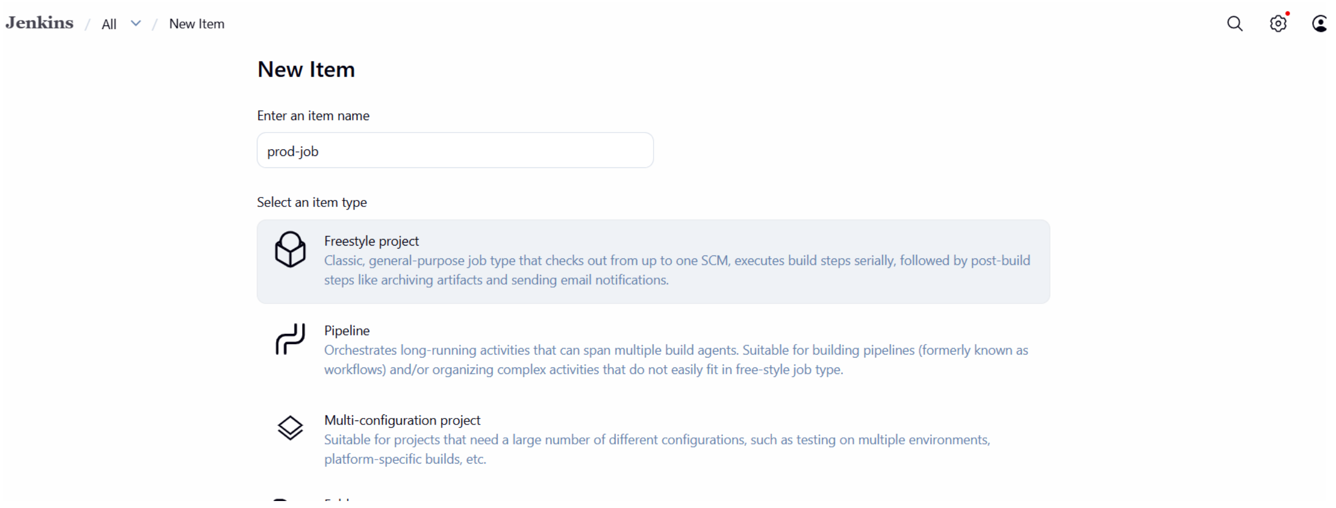
✓ 2. prod-job (deploy to prod node)

Jenkins → New Item → prod-job

Configuration:

- Restrict where to run: prod-node
- SCM → Git → Branch:
- */develop
- Build Step → Execute Shell:

```
echo "Deploying to Prod Node"
mkdir -p /home/ubuntu/prod-deploy
rm -rf /home/ubuntu/prod-deploy/*
cp -r * /home/ubuntu/prod-deploy/
```



New Item

Enter an item name

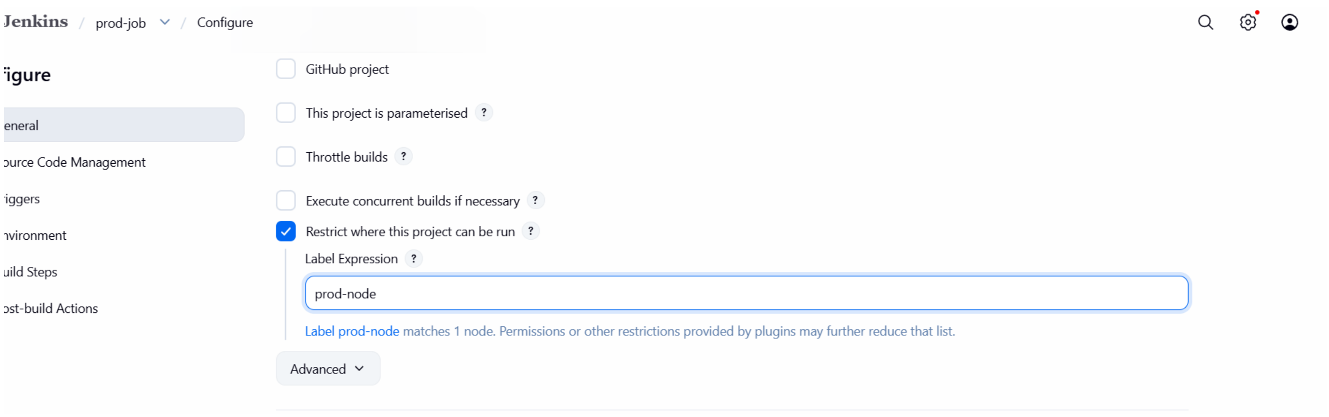
prod-job

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

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



Jenkins / prod-job / Configure

igure

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

☐ GitHub project

☐ This project is parameterised ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

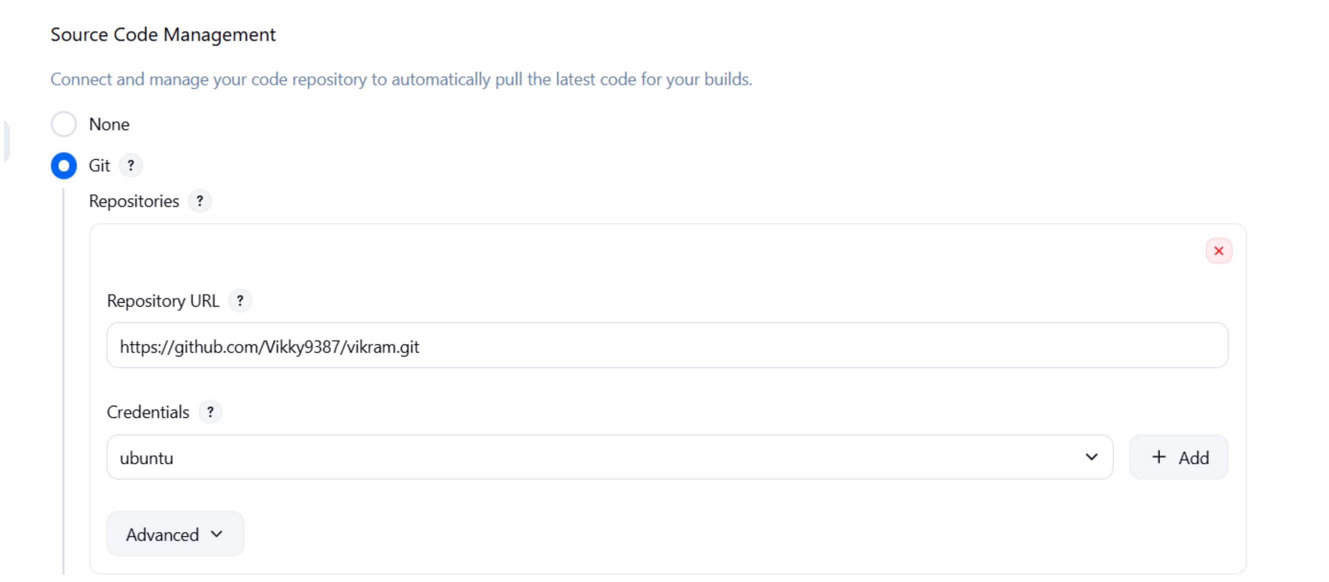
☑ Restrict where this project can be run ?

Label Expression ?

prod-node

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

Advanced



Source Code Management

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

○ None

● Git ?

Repositories ?

Repository URL ?

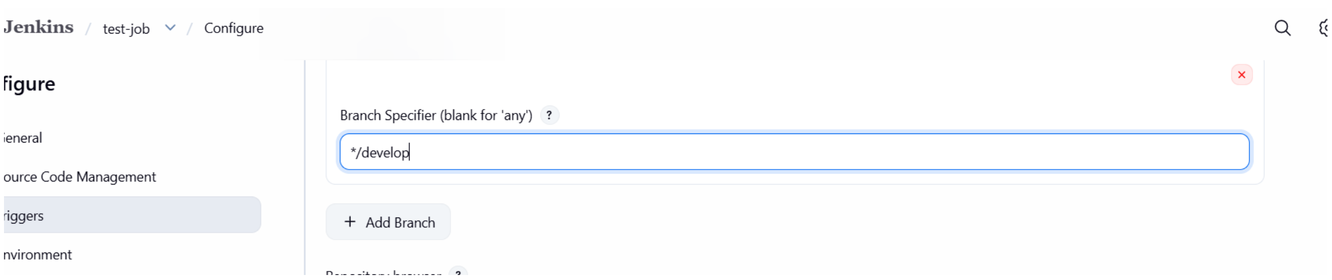
https://github.com/Vikky9387/vikram.git

Credentials ?

ubuntu

+ Add

Advanced



Jenkins / test-job / Configure

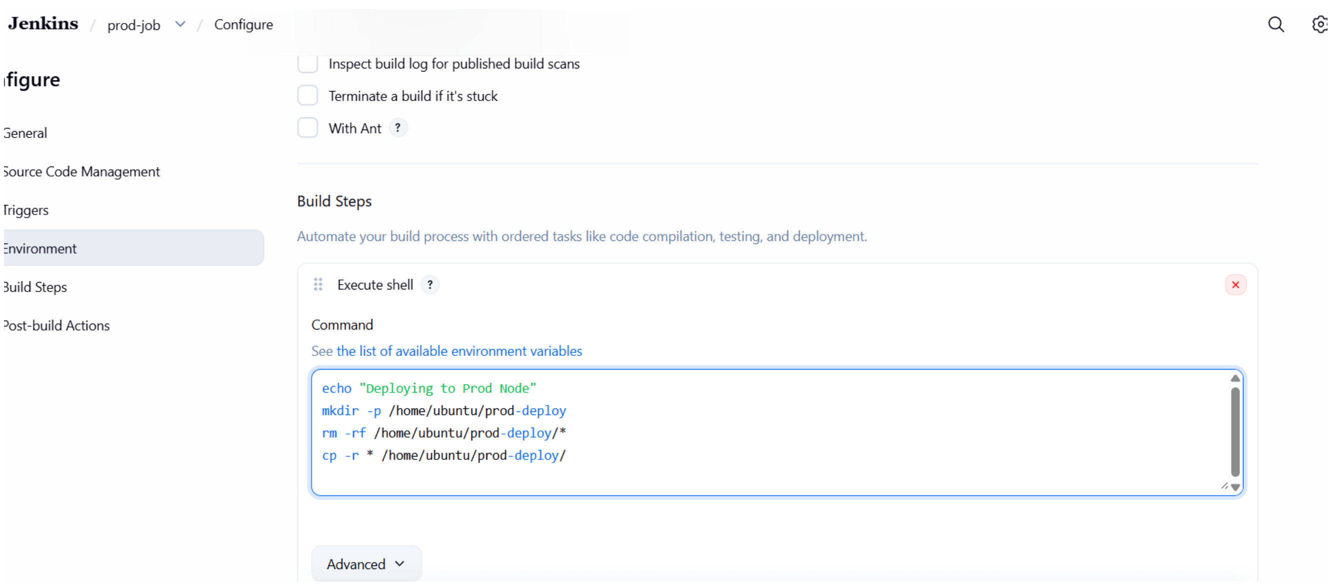
figure

eneral
ource Code Management
riggers
nvironment

Branch Specifier (blank for 'any') ?

*/develop

+ Add Branch



Jenkins / prod-job / Configure

figure

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

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

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

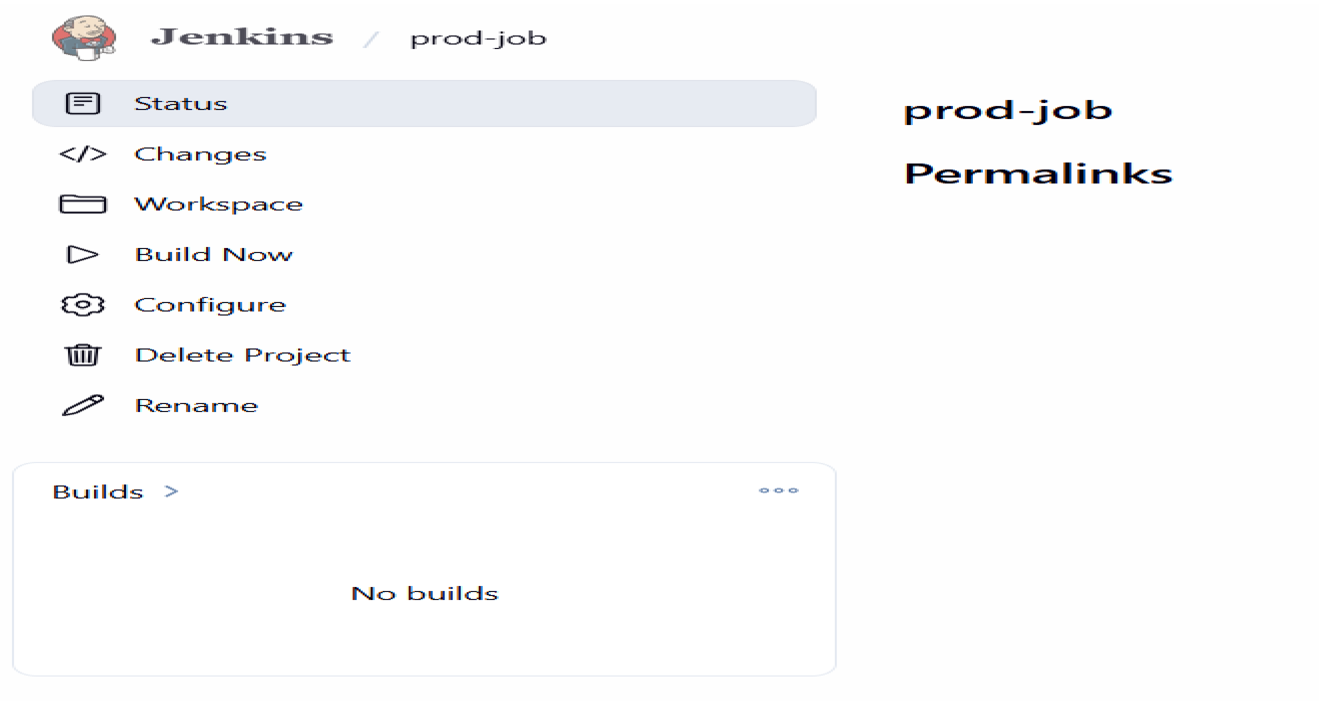
Execute shell ?

Command

See the list of available environment variables

```
echo "Deploying to Prod Node"
mkdir -p /home/ubuntu/prod-deploy
rm -rf /home/ubuntu/prod-deploy/*
cp -r * /home/ubuntu/prod-deploy/
```

Advanced



No builds

▯ TASK 2: Create Jenkins Pipeline

Jenkins → New Item → Pipeline → Name: deploy-pipeline

Enable:

✔ GitHub hook trigger for GITScm polling

▯ Pipeline Script Used

```
pipeline {
  agent any

  triggers {
    githubPush()
  }

  stages {

    stage('Checkout Develop Branch') {
      steps {
        git branch: 'develop',
          url: 'https://github.com/<your-username>/<your-repo>.git'
      }
    }
```

```
        stage('Run Test Job') {
            steps {
                build job: 'test-job'
            }
        }

        stage('Run Prod Job') {
            steps {
                build job: 'prod-job'
            }
        }
    }
}
```

☆ Explanation of Pipeline Stages

✔ Stage 1: Checkout develop branch

Triggered when code is pushed to develop branch.

✔ Stage 2: Run Test Job

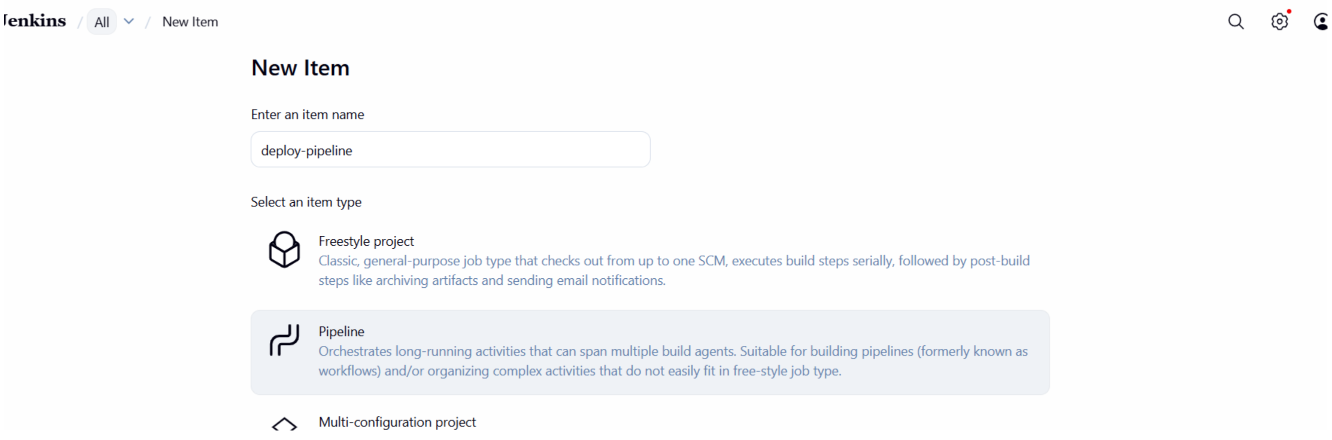
test-job copies files into:

/home/ubuntu/test-deploy

✔ Stage 3: Run Prod Job (only if test succeeds)

prod-job deploys files to:

/home/ubuntu/prod-deploy



New Item

Enter an item name

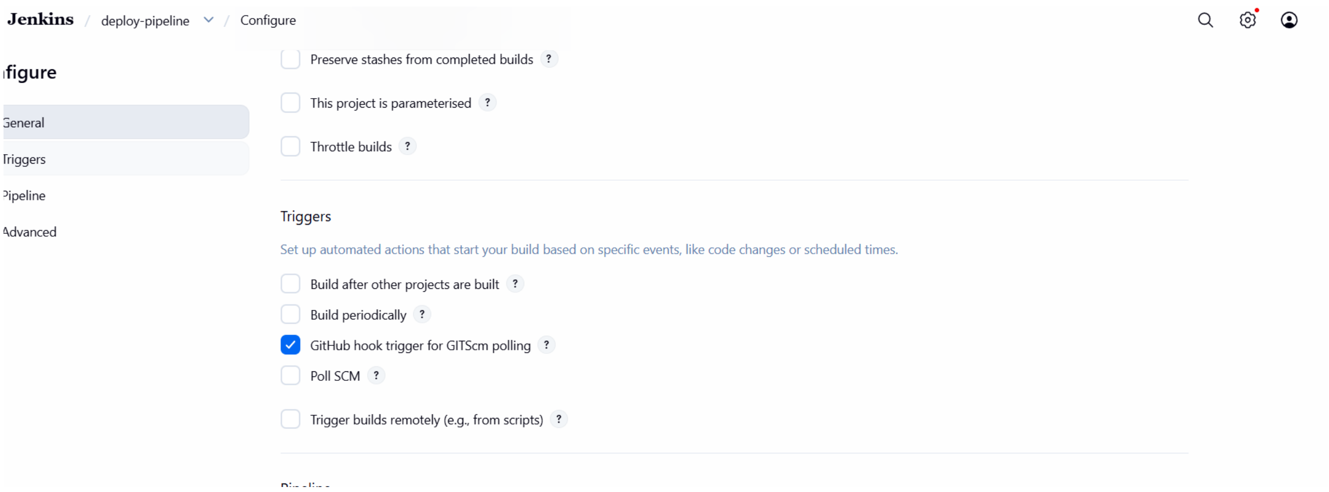
deploy-pipeline

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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

Multi-configuration project



Jenkins / deploy-pipeline / Configure

figure

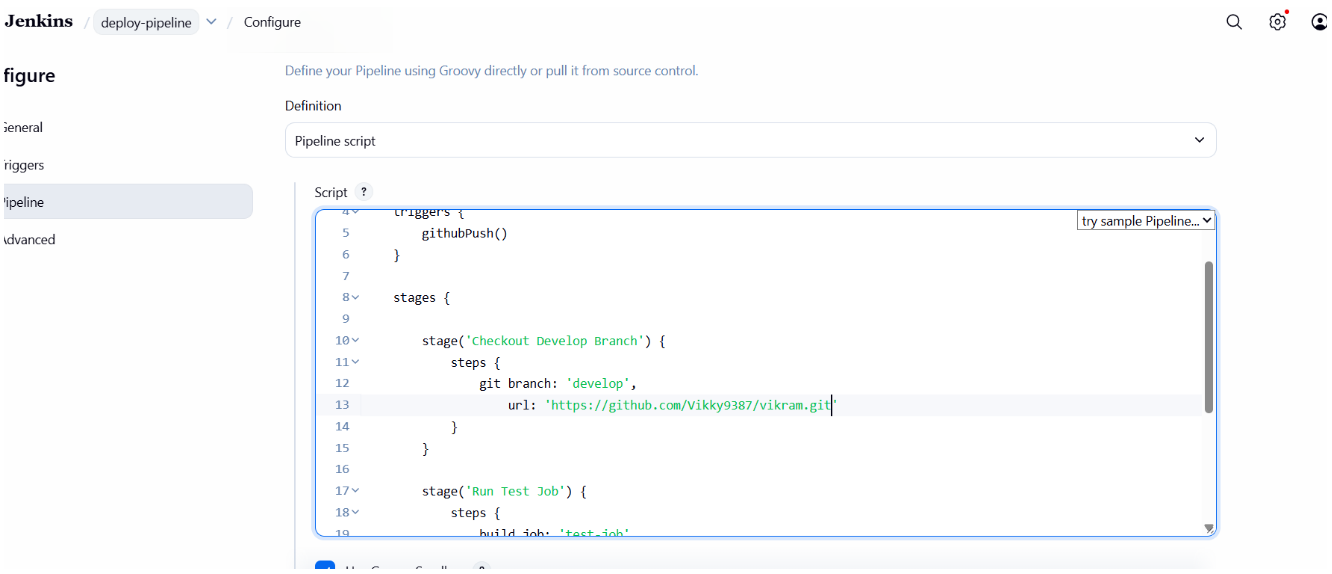
- eneral
- riggers
- ipeline
- dvanced

- [ ] Preserve stashes from completed builds ?
- [ ] This project is parameterised ?
- [ ] Throttle builds ?

Triggers

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

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



Jenkins / deploy-pipeline / Configure

figure

- eneral
- riggers
- ipeline
- dvanced

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script ?

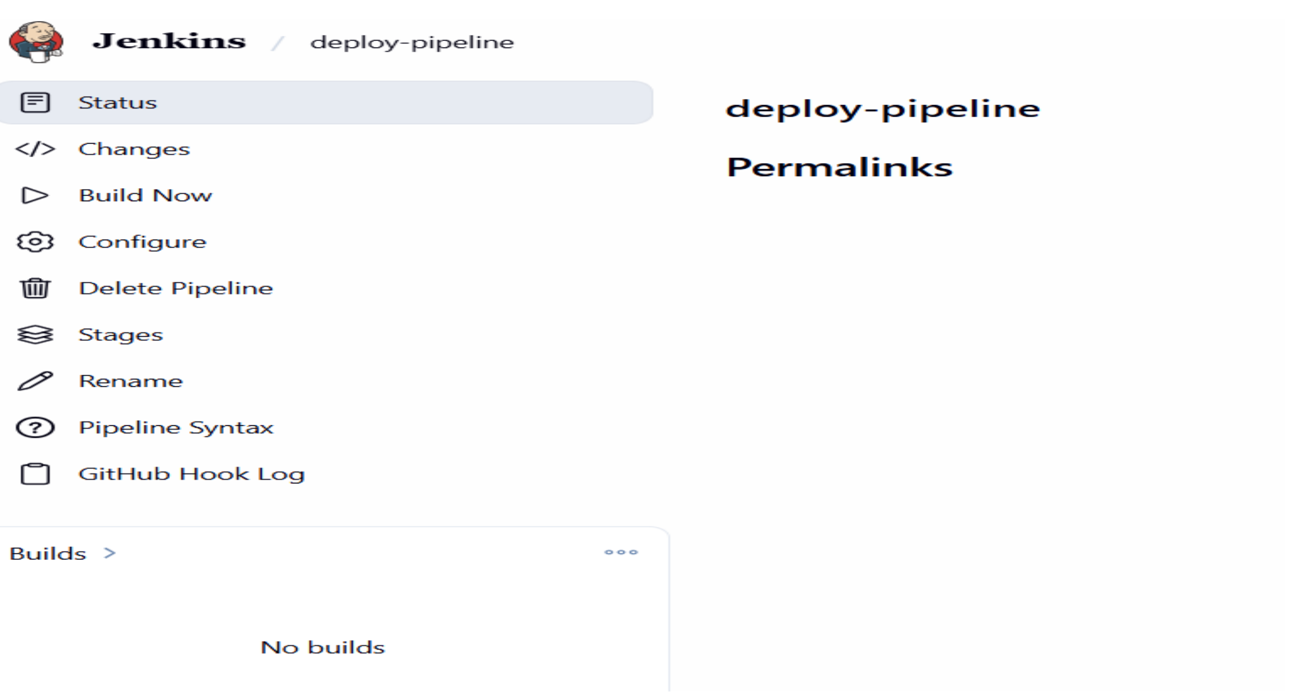
try sample Pipeline...

```
    triggers {
        githubPush()
    }

    stages {

        stage('Checkout Develop Branch') {
            steps {
                git branch: 'develop',
                    url: 'https://github.com/Vikky9387/vikram.git'
            }
        }

        stage('Run Test Job') {
            steps {
                build job: 'test-job'
```



Jenkins / deploy-pipeline

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Stages
- Rename
- Pipeline Syntax
- GitHub Hook Log

deploy-pipeline

Permalinks

Builds >

No builds

▯ TASK 3: Configure GitHub Webhook

GitHub → Repo → Settings → Webhooks → Add Webhook

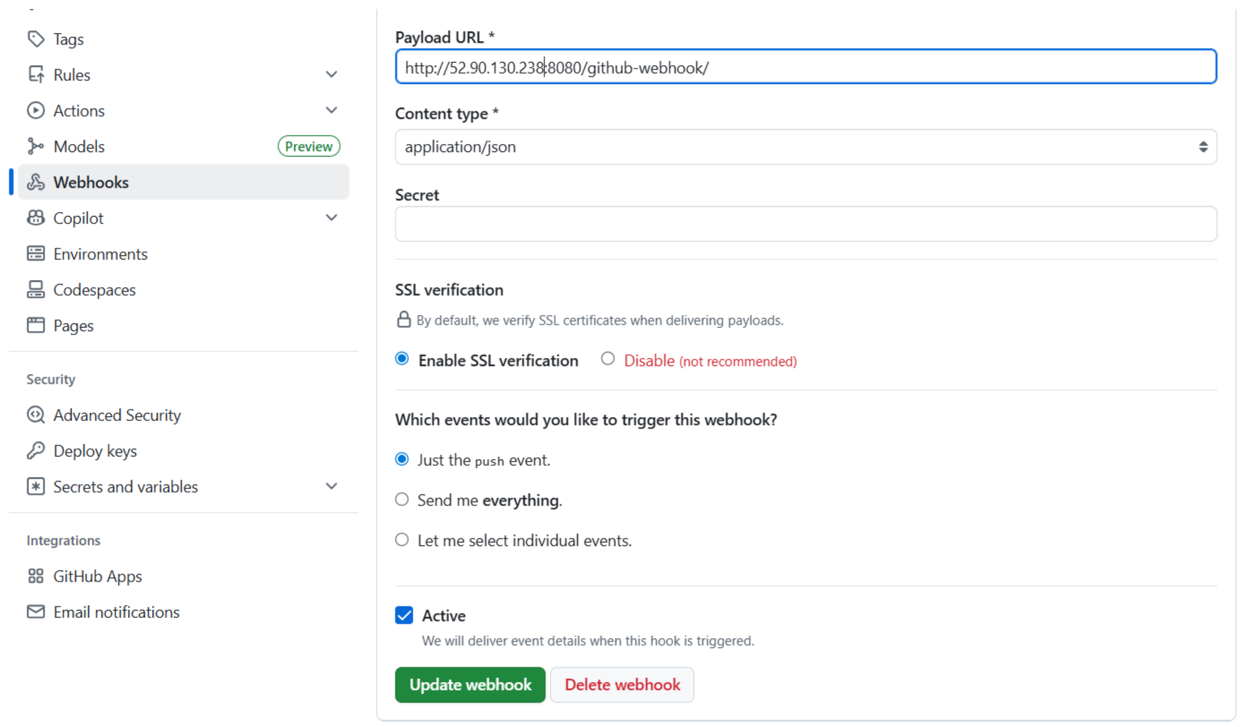
Payload URL:

http://<JENKINS_MASTER_PUBLIC_IP>:8080/github-webhook/

Content type: application/json

Event: Just the push event

Webhook created successfully.



Payload URL *
http://52.90.130.238:8080/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 4: Test the Pipeline Execution

✔ Step 1

Edit/create a file in GitHub and commit to:

develop branch

✔ Step 2

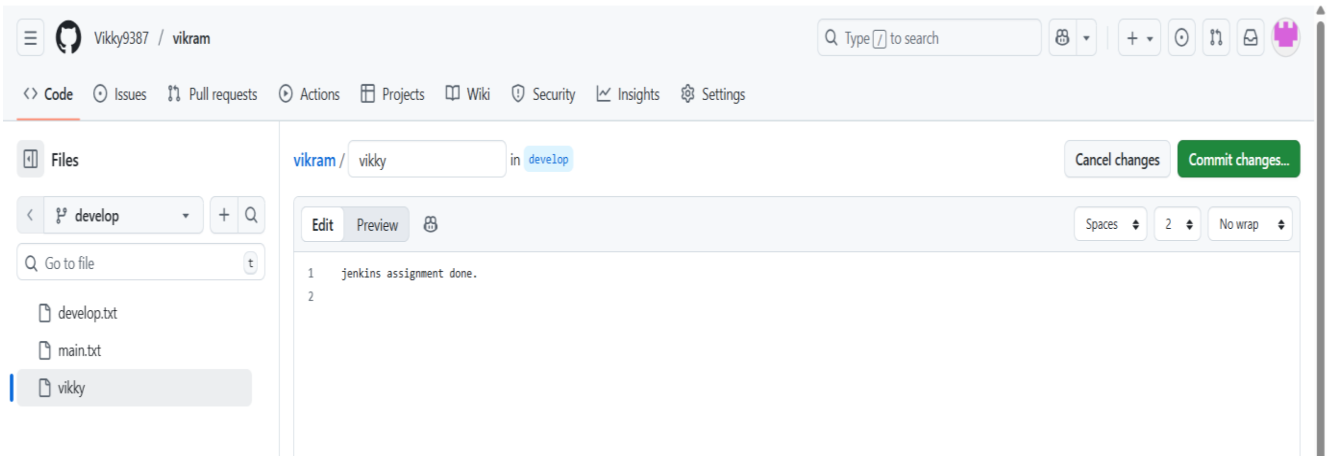
Webhook triggers deploy-pipeline.

✔ Step 3

Pipeline runs:

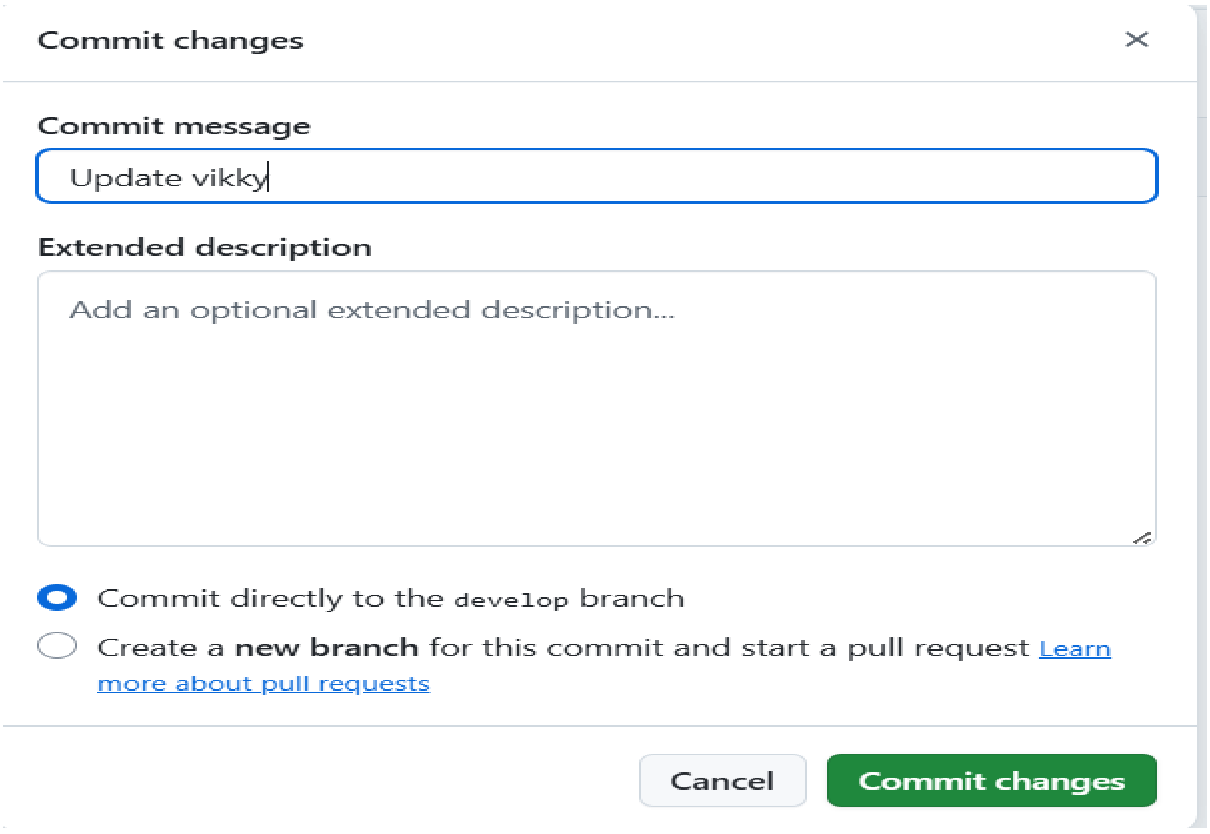
1. Checkout develop branch → SUCCESS
2. Run test-job → Copies files to Test Node
3. Run prod-job → Copies files to Prod Node

Both nodes received deployments correctly.



vikram / vikky in develop

```
jenkins assignment done.
```



Commit changes

Commit message

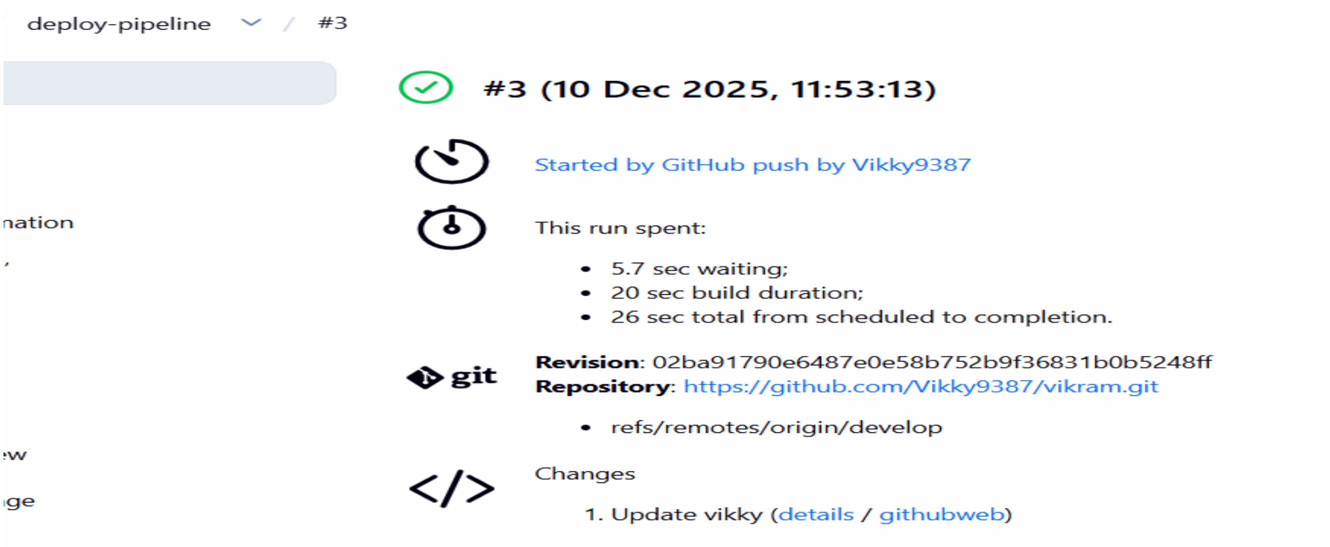
Update vikky

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



#3 (10 Dec 2025, 11:53:13)

Started by GitHub push by Vikky9387

This run spent:

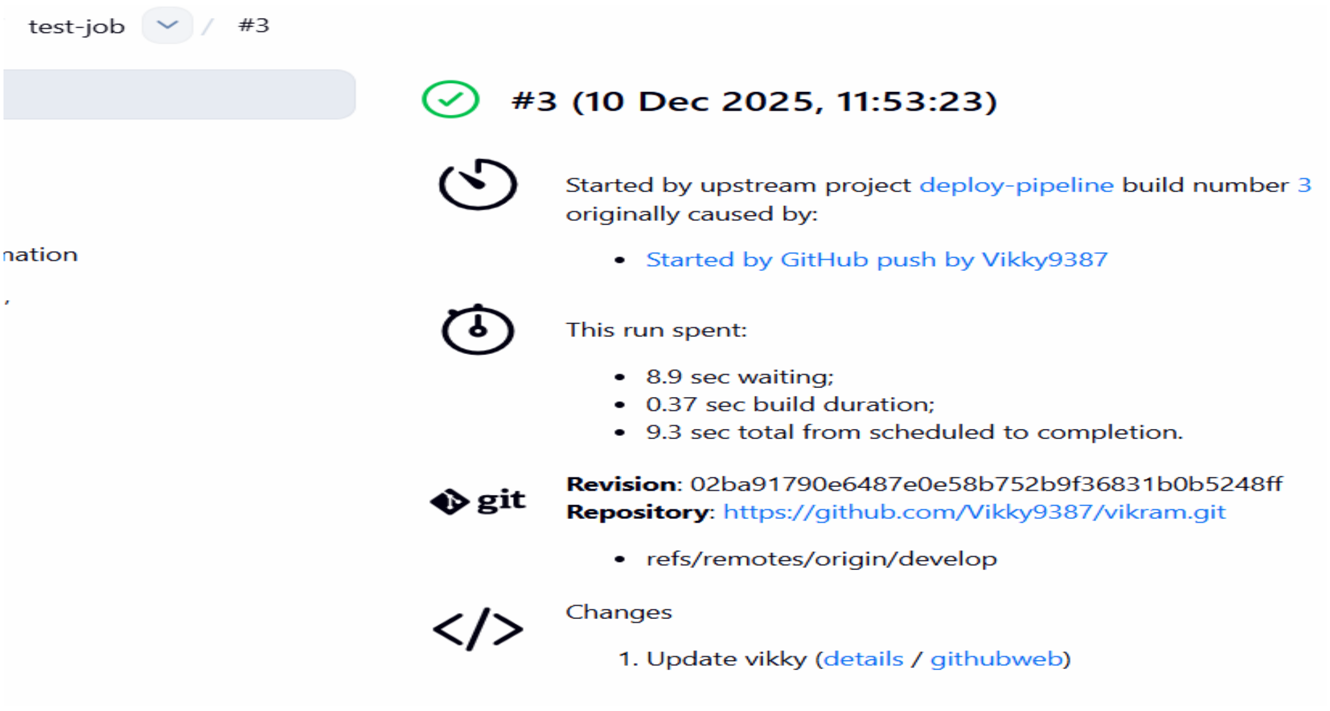
- 5.7 sec waiting;
- 20 sec build duration;
- 26 sec total from scheduled to completion.

Revision: 02ba91790e6487e0e58b752b9f36831b0b5248ff
Repository: https://github.com/Vikky9387/vikram.git

- refs/remotes/origin/develop

Changes

1. Update vikky (details / githubweb)



test-job / #3

nation

#3 (10 Dec 2025, 11:53:23)

Started by upstream project deploy-pipeline build number 3
originally caused by:

- Started by GitHub push by Vikky9387

This run spent:

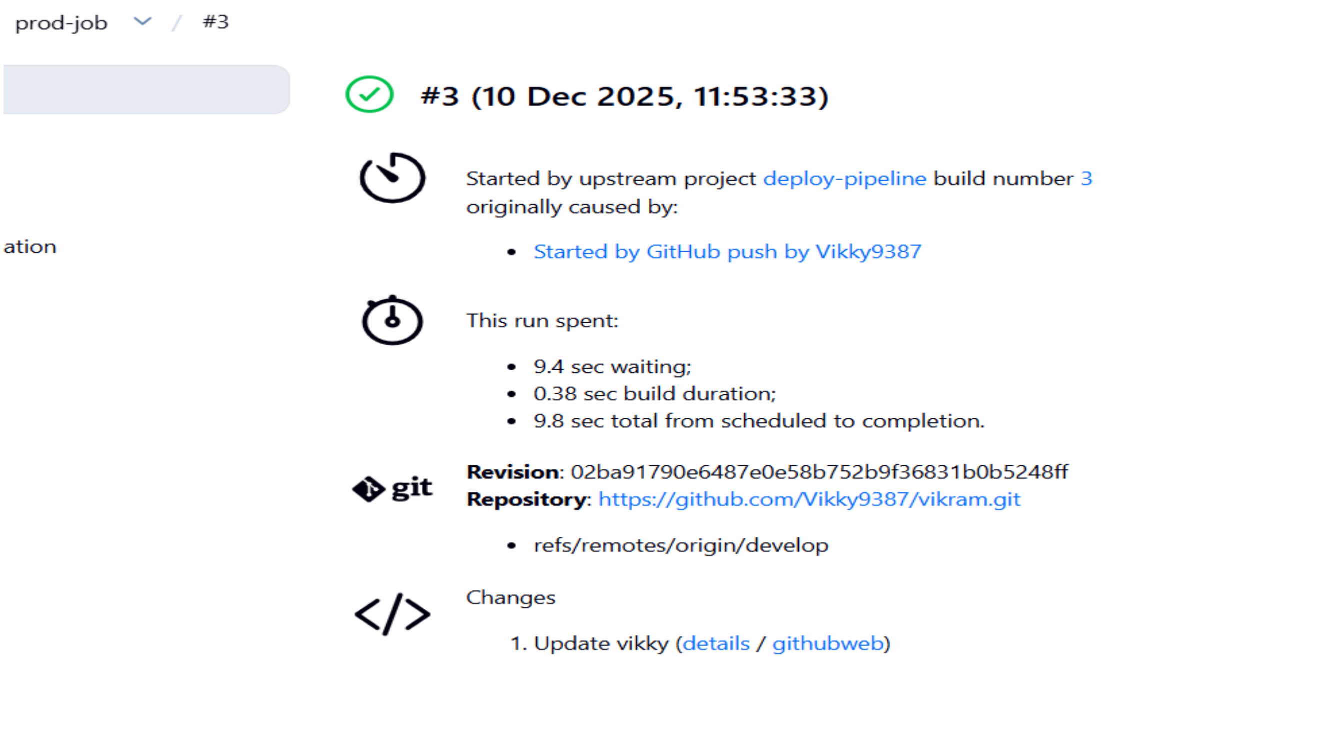
- 8.9 sec waiting;
- 0.37 sec build duration;
- 9.3 sec total from scheduled to completion.

Revision: 02ba91790e6487e0e58b752b9f36831b0b5248ff
Repository: https://github.com/Vikky9387/vikram.git

- refs/remotes/origin/develop

Changes

1. Update vikky (details / githubweb)



prod-job / #3

ation

#3 (10 Dec 2025, 11:53:33)

Started by upstream project deploy-pipeline build number 3
originally caused by:

- Started by GitHub push by Vikky9387

This run spent:

- 9.4 sec waiting;
- 0.38 sec build duration;
- 9.8 sec total from scheduled to completion.

Revision: 02ba91790e6487e0e58b752b9f36831b0b5248ff
Repository: https://github.com/Vikky9387/vikram.git

- refs/remotes/origin/develop

Changes

1. Update vikky (details / githubweb)

Conclusion

Successfully created a Jenkins Pipeline that automates a full CI/CD flow.

A push to the develop branch triggers:

1. Test job → deploys to Test Node
2. If successful, Prod job → deploys to Prod Node

This demonstrates:

- Pipeline scripting
- Build chaining
- Automated deployments
- Multi-node Jenkins configuration

GitHub webhook integration