

Module 6: Jenkins Assignment - 3

1. Create a pipeline in Jenkins
2. Once push is made to “develop” a branch in Git, trigger job “test”. This will copy Git files to test node
3. If test job is successful, then prod job should be triggered
4. Prod jobs should copy files to prod node


Solution:-

```
$ mkdir assignment3
$ cd assignment3/
$ git init
$ ls -al
```

```
ubuntu@Jenkins-Master:~$ mkdir assignment3
ubuntu@Jenkins-Master:~$ cd assignment3/
ubuntu@Jenkins-Master:~/assignment3$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ubuntu/assignment3/.git/
ubuntu@Jenkins-Master:~/assignment3$ ls -al
total 12
drwxrwxr-x 3 ubuntu ubuntu 4096 Aug 27 12:10 .
drwxr-x--- 7 ubuntu ubuntu 4096 Aug 27 12:09 ..
drwxrwxr-x 7 ubuntu ubuntu 4096 Aug 27 12:10 .git
```


* Create two nodes:-


```
* Test-node:- dashboard -> nodes -> test-node -> configure ->
name (test-node) -> remote root directory ->
/home/ubuntu/assignment-3
```





Jenkins


Dashboard > Nodes > test-node > Configure


 Status


 Delete Agent


 Configure


 Build History

 Load Statistics

 Script Console

 Log

 System Information

 Disconnect

Name ?

test-node

Description ?

Plain text [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/ubuntu/assignment-3

Build Executor Status ▼

1 Idle

* Configure host and credential for that:- Labels -> dev-node -> launch method -> launch agents via SSH -> host -> 172.31.45.149 -> Give credentials -> add -> (username and password)

Dashboard > Nodes > test-node > Configure

Build Executor Status ▼

1 Idle

Remote root directory ?

/home/ubuntu/assignment-3

Labels ?

dev-node

Usage ?

Use this node as much as possible

Launch method ?

Launch agents via SSH

Host ?

172.31.45.149

Credentials ?

ubuntu/*****

Add ▼

* Create key verification:- host key verification strategy -> non verifying verification strategy -> apply -> save

Dashboard > Nodes > test-node > Configure

172.31.45.149

Credentials ?

ubuntu/*****

Add ▾

Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced ▾ Edited

Availability ?


Keep this agent online as much as possible

Node Properties


☐ Disable deferred wipeout on this node ?


☐ Environment variables


* Prod node:- dashboard -> nodes -> test-node -> configure -> name (test-node) -> remote root directory -> /home/ubuntu/assignment-3


 **Jenkins**


Dashboard > Nodes > prod-node > Configure


 Status


 Delete Agent


 Configure


 Build History

 Load Statistics

 Script Console

 Log

 System Information

 Disconnect

Build Executor Status ▾

1 Idle

Name ?

prod-node

Description ?

Plain text [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/ubuntu/assignment-3

* Configure host and credentials:- Labels -> prod-node -> launch method -> launch agents via SSH -> host -> 172.31.39.81 -> Give credentials -> add -> (username and password)

Dashboard > Nodes > prod-node > Configure

Build Executor Status ▾

1 Idle

Remote root directory ?
/home/ubuntu/assignment-3

Labels ?
prod-node

Usage ?
Use this node as much as possible

Launch method ?
Launch agents via SSH

Host ?
172.31.39.81

Credentials ?
ubuntu/*****

Add ▾

* Configure host key verification:- host key verification strategy -> non verifying verification strategy -> apply -> save

Dashboard > Nodes > prod-node > Configure

ubuntu/*****

Add ▾

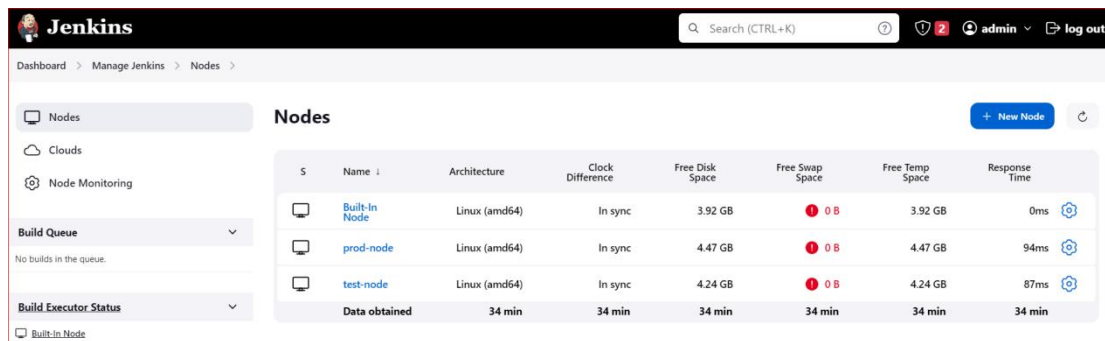
Host Key Verification Strategy ?
Non verifying Verification Strategy

Advanced ▾ Edited

Availability ?
Keep this agent online as much as possible

Node Properties

* Here we have prod node and test node:-



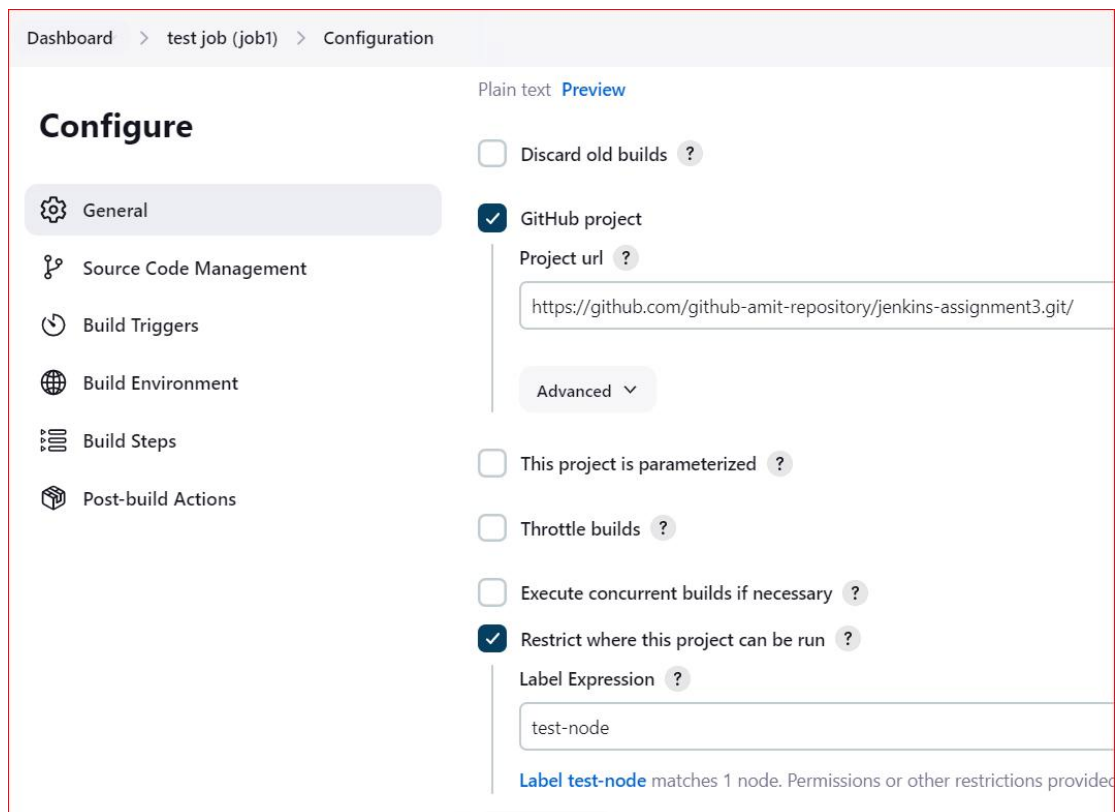
The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with 'Nodes', 'Clouds', and 'Node Monitoring'. Below that, a 'Build Queue' section shows 'No builds in the queue.' and a 'Build Executor Status' section shows 'Built-In Node'. The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. There are three nodes listed: 'Built-In Node', 'prod-node', and 'test-node'. All three are Linux (amd64) and 'In sync'. The 'prod-node' and 'test-node' have 4.47 GB and 4.24 GB of free disk space respectively, with 0 B of free swap space and 94ms and 87ms response times. A 'Data obtained' row at the bottom shows 34 min for all metrics.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	3.92 GB	0 B	3.92 GB	0ms
	prod-node	Linux (amd64)	In sync	4.47 GB	0 B	4.47 GB	94ms
	test-node	Linux (amd64)	In sync	4.24 GB	0 B	4.24 GB	87ms
Data obtained		34 min	34 min	34 min	34 min	34 min	34 min

* Now create two jobs:-

* Create job-1:-

* Configure test job give github url and labels:- github project -> project url -> <https://github.com/github-amit-repository/jenkins-assignment3.git> -> check on restrict where this project can be run -> label expression -> test-node



The screenshot shows the Jenkins 'Configuration' page for a job named 'test job (job1)'. The left sidebar has 'Configure' at the top, followed by 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'General' tab is selected. On the right, there are several checkboxes: 'Discard old builds' (unchecked), 'GitHub project' (checked), 'This project is parameterized' (unchecked), 'Throttle builds' (unchecked), 'Execute concurrent builds if necessary' (unchecked), and 'Restrict where this project can be run' (checked). The 'GitHub project' section shows the 'Project url' as 'https://github.com/github-amit-repository/jenkins-assignment3.git/'. The 'Restrict where this project can be run' section shows the 'Label Expression' as 'test-node'. A note at the bottom says 'Label test-node matches 1 node. Permissions or other restrictions provided'.

Dashboard > test job (job1) > Configuration

Plain text [Preview](#)

Configure

- ☐ Discard old builds ?
- ☒ GitHub project
 - Project url ?
 - Advanced ▾
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Execute concurrent builds if necessary ?
- ☒ Restrict where this project can be run ?
 - Label Expression ?
 - Label **test-node** matches 1 node. Permissions or other restrictions provided

* Configure repo:- source code management -> git -> repo url -> <https://github.com/github-amit-repository/jenkins-assignment3.git> -> Give credentials -> add -> (username and password)

The screenshot shows the Jenkins Configuration page for a job named 'test job (job1)'. The 'Source Code Management' section is active, with 'Git' selected as the source code management system. The 'Repository URL' is set to 'https://github.com/github-amit-repository/jenkins-assignment3.git'. The 'Credentials' field is set to 'ubuntu/*****'. There is an 'Add' button and an 'Advanced' dropdown menu. At the bottom, there is an 'Add Repository' button.

Dashboard > test job (job1) > Configuration

Configure

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

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

* Configure branch:- branch to build -> branch specifier -> */develop

The screenshot shows the Jenkins Configuration page for a job named 'test job (job1)'. The 'Source Code Management' section is active, and the 'Branches to build' section is expanded. The 'Branch Specifier (blank for 'any')' is set to '*/develop'. There is an 'Add Branch' button. Below this, the 'Repository browser' is set to '(Auto)'. At the bottom, there is an 'Add' button for 'Additional Behaviours'.

Dashboard > test job (job1) > Configuration

Configure

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

Advanced

Branches to build ?

Branch Specifier (blank for 'any') ?

Repository browser ?

Additional Behaviours

* Configure GITScm:- build triggers -> check on GitHub hook trigger for GITScm polling

Dashboard > test job (job1) > Configuration

Configure

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

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) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

* Configure build steps:- build other projects -> projects to build -> prod job (job2) -> select trigger only if build is stable -> apply -> save

Dashboard > test job (job1) > Configuration

Configure

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

Post-build Actions

Build other projects ?

Projects to build

prod job (job2)

- ☒ Trigger only if build is stable
- ☐ Trigger even if the build is unstable
- ☐ Trigger even if the build fails

Save **Apply**

Dashboard > test job (job1) >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Project test job (job1)

Downstream Projects

✓ prod job (job2)

Permalinks

* Create job-2:-

* Configure prod job give github url and labels:- [github project -> project url -> https://github.com/github-amit-repository/jenkins-assignment3.git -> check on restrict where this project can be run -> label expression -> prod-node](#)

Dashboard > prod job (job2) >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

GitHub

Project prod job (job2)

For develop branch do not publish it.

Upstream Projects

✓ test job (job1)

Permalinks

Dashboard > prod job (job2) > Configuration

Configure

Plain text [Preview](#)

- ☐ Discard old builds ?
- ☒ GitHub project
 - Project url ?
 - Advanced ▾
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Execute concurrent builds if necessary ?
- ☒ Restrict where this project can be run ?
 - Label Expression ?

* Configure repo:- source code management -> git -> repo url -> https://github.com/github-amit-repository/jenkins-assignment3.git -> Give credentials -> add -> (username and password)

Dashboard > prod job (job2) > Configuration

Configure

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

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

Add ▾

Advanced ▾

* Configure branch:- branch to build -> branch specifier -> */develop

Dashboard > prod job (job2) > Configuration

Configure

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

Branches to build ?

Branch Specifier (blank for 'any') ?

*/develop

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ▾

* In this job we does not onfigure GITScm:- build triggers -> do not check on GitHub hook trigger for GITScm polling

Dashboard > prod job (job2) > Configuration

Configure

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

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) ?
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

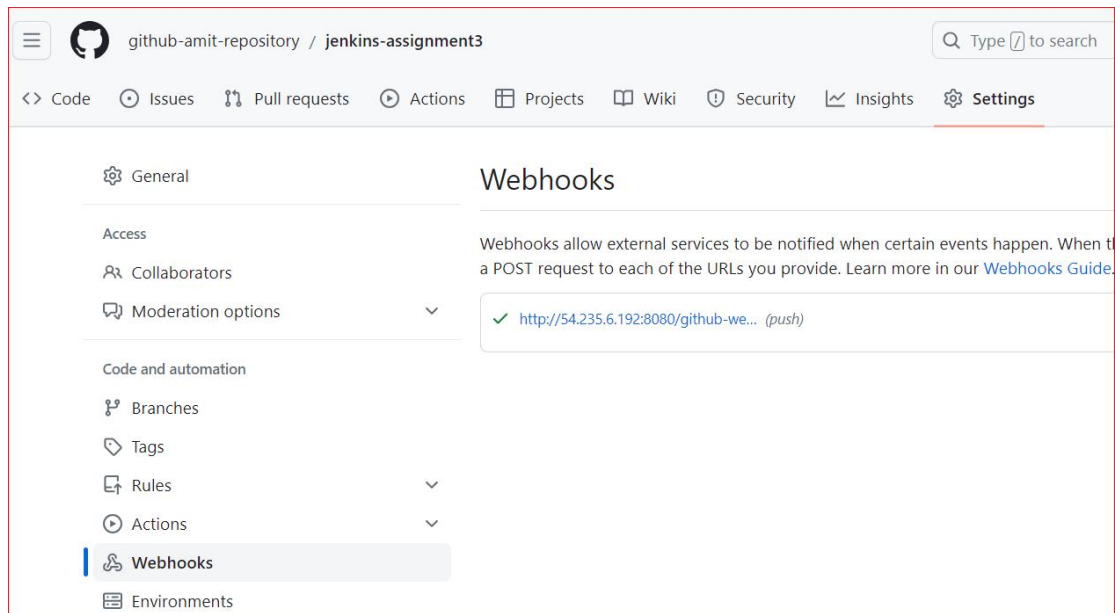
* Now we have two jobs:- Job1:- test job and Job2:- prod job

The screenshot shows the Jenkins Dashboard with a sidebar on the left containing links like 'New Item', 'People', 'Build History', etc. The main area displays a table of jobs under the 'assignment-3' filter. The table has columns for status (S), warnings (W), name, last success, last failure, and last duration. Two jobs are listed: 'prod job (job2)' and 'test job (job1)'. Below the table, there are filters for 'Icon: S M L' and 'Atom feed' options.

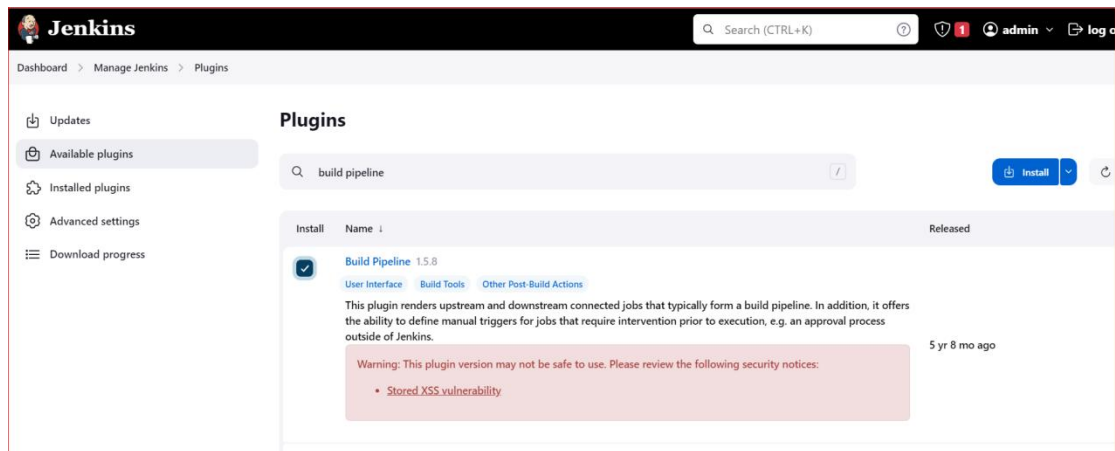
S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	prod job (job2)	20 min #6	N/A	0.46 sec
✓	☁	test job (job1)	20 min #20	1 hr 44 min #17	0.48 sec

* Create webhooks:- Github -> github repo -> settings -> webhooks -> add webhook -> payload url -> copy jenkins url and paste in (http://54.235.6.192:8080/github-webhook/) -> http://jenkins url till port/github-webhook/ -> add webhook -> webhooks/manage webhook/ recent deliveries -> save

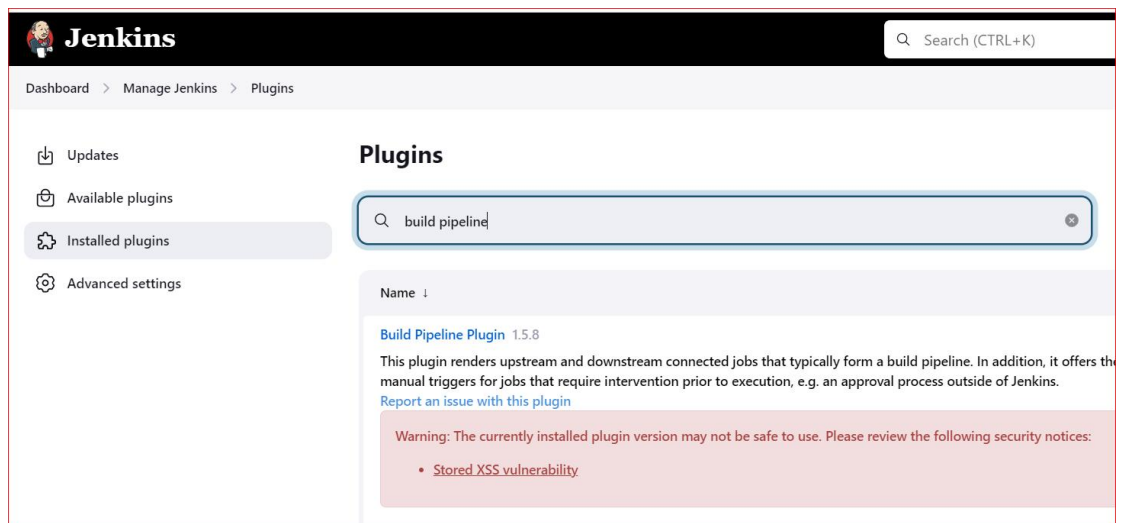
The screenshot shows the 'Add webhook' configuration page in GitHub. On the left is a sidebar with navigation options like 'Moderation options', 'Code and automation', 'Webhooks', etc. The main form area includes fields for 'Payload URL' (set to 'http://54.235.6.192:8080/github-webhook/'), 'Content type' (set to 'application/x-www-form-urlencoded'), and a 'Secret' field. Below these, there are radio buttons for 'Which events would you like to trigger this webhook?' with options: 'Just the push event.' (selected), 'Send me everything.', and 'Let me select individual events.'. At the bottom, there is a checkbox for 'Active' which is checked, and a green 'Add webhook' button.



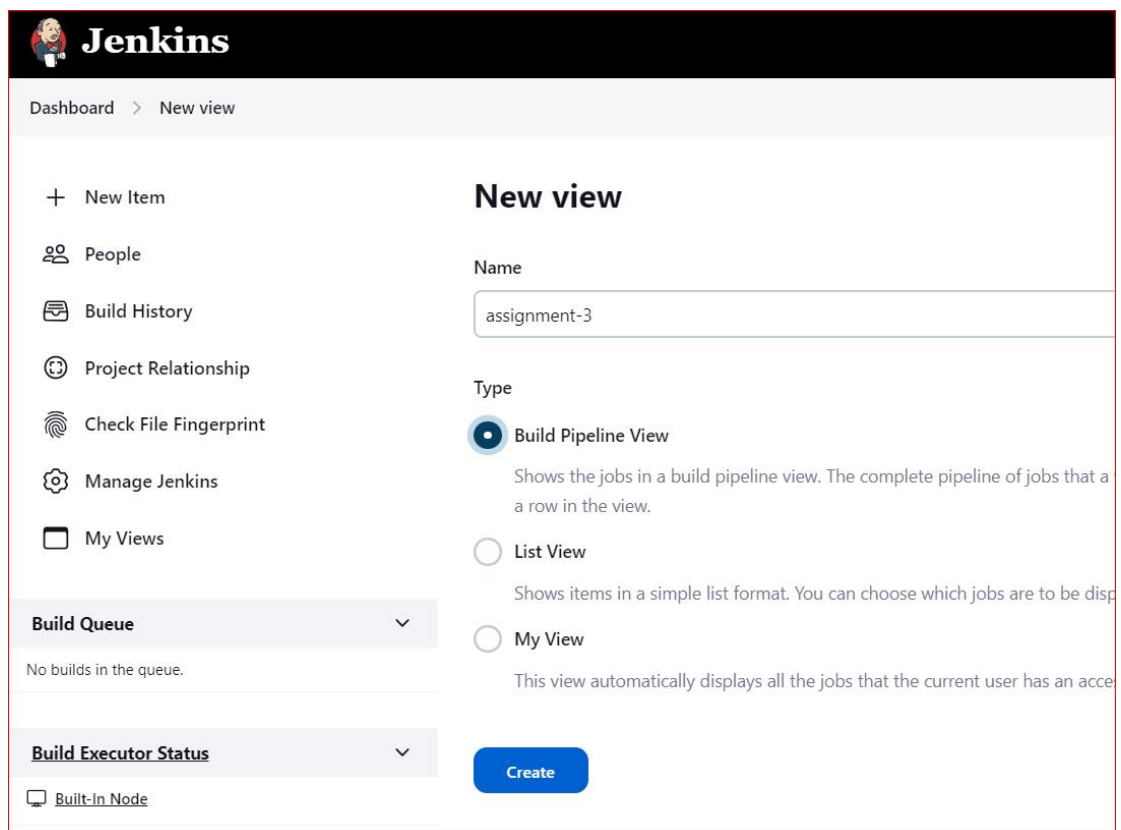
* Install plugins:- dashboard->manage jenkins->plugins->available plungins->search build pipeline-> install without restart



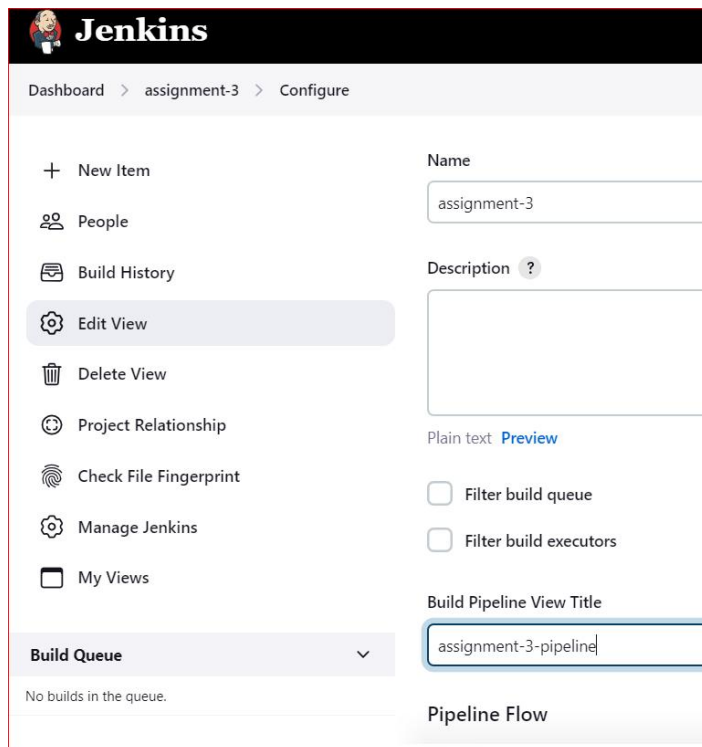
* Verify plugins are installed or not:- installed plugins->search plugins-> build pipeline



* Now create view:- dashboard->new view->name -> assignment-3 -> type -> select build pipeline view -> create

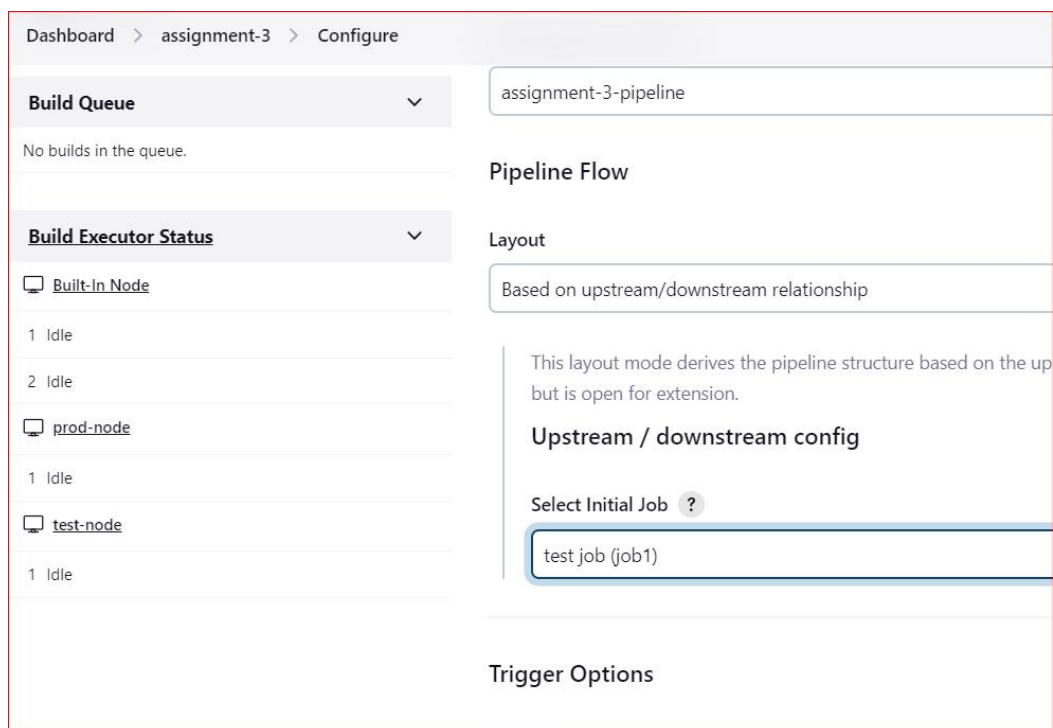


* Now configure view:- edit view->build pipeline view title->assignment-3-pipeline



The screenshot shows the Jenkins 'Configure' page for 'assignment-3'. On the left is a sidebar with navigation links: New Item, People, Build History, Edit View (highlighted), Delete View, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area contains a 'Name' field with 'assignment-3', a 'Description' field, and checkboxes for 'Filter build queue' and 'Filter build executors'. The 'Build Pipeline View Title' field is set to 'assignment-3-pipeline'. Below this is a 'Pipeline Flow' section.

* Give initial job:- pipeline flow->upstream / downstream config-> select initial job->test job (job1)

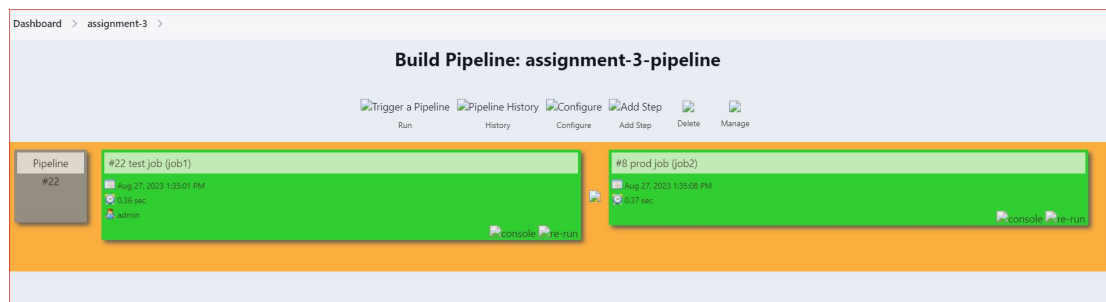


This screenshot shows the 'Pipeline Flow' configuration section of the Jenkins 'assignment-3' configuration page. The 'Layout' is set to 'Based on upstream/downstream relationship'. A text box explains: 'This layout mode derives the pipeline structure based on the upstream/downstream relationship but is open for extension.' Under the 'Upstream / downstream config' section, the 'Select Initial Job' dropdown is set to 'test job (job1)'. The left sidebar shows the 'Build Queue' (empty) and 'Build Executor Status' with four executors: 'Built-In Node' (1 Idle), 'prod-node' (1 Idle), and 'test-node' (1 Idle).

```
$ sudo touch dev_to_test_to_prod.html
$ git add .
$ git commit -m "Commit to trigger html to auto"
```

```
ubuntu@Jenkins-Master:~/assignment3$ sudo touch dev_to_test_to_prod.html
ubuntu@Jenkins-Master:~/assignment3$ git add .
ubuntu@Jenkins-Master:~/assignment3$ git commit -m "Commit to trigger html to auto"
[develop 2572622] Commit to trigger html to auto
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 dev_to_test_to_prod.html
ubuntu@Jenkins-Master:~/assignment3$ git push --all
Username for 'https://github.com': github-amit-repository
Password for 'https://github.com:github-amit-repository@github.com':
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 246 bytes | 246.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/github-amit-repository/jenkins-assignment3.git
a14f788..2572622 develop -> develop
ubuntu@Jenkins-Master:~/assignment3$
```

* Here is view of build pipeline in jenkins:-



```
$ ls
$ cd test\ job\ \ (job1\ )
$ ls
```

```
ubuntu@test-server:~/assignment-3/workspace$ ls
'test job (job1)' 'test job (job1)@tmp'
ubuntu@test-server:~/assignment-3/workspace$ cd test\ job\ \ (job1\ )
ubuntu@test-server:~/assignment-3/workspace/test job (job1)$ ls
dev_to test_to_prod.html dev_to test_to_prod.txt deve_branch to_test_to_prod.txt master.txt
ubuntu@test-server:~/assignment-3/workspace/test job (job1)$
```

```
$ ls
```

```
ubuntu@prod-server:~/assignment-3/workspace/prod job (job2)$ ls
dev_to test_to_prod.html dev_to test_to_prod.txt deve_branch to_test_to_prod.txt master.txt
ubuntu@prod-server:~/assignment-3/workspace/prod job (job2)$
```

* We can see in github repo dev_to_test_to_prod.html is present.

