

## LAB 2

01) What is Jenkins pipeline?

Jenkins Pipeline is a suite of plugins which supports implementing and integrating *continuous delivery pipelines* into Jenkins.

*A continuous delivery pipeline* is an automated expression of your process for getting software from version control right through to your users and customers.

02) What scripting language is Jenkins pipeline syntax based on?

Groovy

03) What are the ways you can write pipeline in Jenkins?

A Jenkinsfile can be written using two types of syntax - **Declarative** and **Scripted**.

Declarative and Scripted Pipelines are constructed fundamentally differently.

Declarative Pipeline is a more recent feature of Jenkins Pipeline which:

- provides richer syntactical features over Scripted Pipeline syntax, and
- is designed to make writing and reading Pipeline code easier.

04) Create Jenkins pipeline for your repo and use script file (jenkinsfile) to write pipeline syntax?

The image shows two screenshots of the Jenkins web interface. The top screenshot displays the configuration page for a pipeline named 'lab2-pipeline'. The left sidebar contains links for Status, Changes, Build with Parameters, Configure, Delete Pipeline, Full Stage View, Rename, and Pipeline Syntax. The main area shows parameters: 'project' (checked) with a help link, and 'namespace' (dropdown menu set to 'prod'). A 'Build' button is at the bottom. The bottom screenshot shows the 'Stage View' for the same pipeline. It includes a table of stage times and a build history section.

**Pipeline lab2-pipeline**

This build requires parameters:

- ☒ project  
this parameter help you to know project name
- namespace  
**prod**

**Build**

**Build History** trend

Filter builds...

#7

**Pipeline lab2-pipeline**

Stage View

Average stage times:  
(Average full run time: ~12s)

	Declarative: Checkout SCM	check	test	deployment
Average	3s	564ms	542ms	389ms
Feb 08 06:58	2s	275ms	150ms	252ms

Build History trend

Filter builds...

#8

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build #8
- Parameters
- Git Build Data
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces
- Previous Build

## Console Output

```
Started by user Marwan Ibrahim Khamis ElSayed Aly
Obtained Jenkinsfile from git https://github.com/swelans/jenkins-demo
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/lab2-pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential github-token
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/lab2-pipeline/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/swelans/jenkins-demo # timeout=10
Fetching upstream changes from https://github.com/swelans/jenkins-demo
> git --version # timeout=10
> git --version # 'git version 2.38.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/swelans/jenkins-demo +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 96f34093ac3a9987e6c61cc1ba48ee7edc5ae0f3 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 96f34093ac3a9987e6c61cc1ba48ee7edc5ae0f3 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 96f34093ac3a9987e6c61cc1ba48ee7edc5ae0f3 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
```

Dashboard > lab2-pipeline > #8

```
> git config core.sparsecheckout # timeout=10
> git checkout -f 96f34093ac3a9987e6c61cc1ba48ee7edc5ae0f3 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 96f34093ac3a9987e6c61cc1ba48ee7edc5ae0f3 # timeout=10
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (check)
[Pipeline] echo
checking your code
[Pipeline] echo
prod
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (test)
[Pipeline] echo
testing your app
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deployment)
[Pipeline] echo
your code is deployed right now
[Pipeline] echo
this build number 8
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

05) Create another multibranch pipeline and filter branches to contain only (master, dev, test)?

The image shows two screenshots. The top screenshot is from the Jenkins web interface, displaying the configuration for a multibranch pipeline named 'lab2-Multibranch-pipeline'. The left sidebar contains various options like Status, Configure, Scan Multibranch Pipeline Now, Scan Multibranch Pipeline Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, People, Build History, Project Relationship, and Check File Fingerprint. The main area shows a table of branches with columns for S, W, Name, Last Success, Last Failure, and Last Duration. The table lists three branches: dev, master, and test, all with a status of 'Success' and a duration of approximately 30 seconds. Below the table, there are icons for S, M, and L, and a legend. The bottom screenshot is from the GitHub web interface, showing the repository 'Marwan Ibrahim Khamis / Marwan\_Sprints\_Bootcamp'. The 'Code' tab is selected, and a 'Switch branches/tags' dropdown menu is open, showing a list of branches: main (selected), dev, master, test\_remote\_repo, and test. The 'About' section on the right shows '0 stars', '1 watching', and '0 forks'. The 'Releases' section shows 'No releases published' and a link to 'Create a new release'.

Jenkins interface showing the configuration for a multibranch pipeline named 'lab2-Multibranch-pipeline'. The pipeline is configured to scan for branches containing 'dev', 'master', and 'test'. The table below shows the status of these branches.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	dev	9 min 31 sec #1	N/A	36 sec
✓	☀	master	3 min 36 sec #1	N/A	33 sec
✓	☀	test	9 min 31 sec #1	N/A	1 min 26 sec

GitHub interface showing the repository 'Marwan Ibrahim Khamis / Marwan\_Sprints\_Bootcamp'. The 'Code' tab is selected, and a dropdown menu is open showing the list of branches: main, dev, master, test\_remote\_repo, and test.