

Jenkins

Lab_2

01) What is Jenkins pipeline?

- Jenkins pipeline is a collection 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 (java-language)

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

Scripted Pipeline	Declarative Pipeline
<p>A scripted pipeline uses the Groovy (JVM-based) language to create a pipeline as code</p> <p>-Problems with scripted pipeline:</p> <ul style="list-style-type: none">• The imbalance between developers' experience in Groovy and Jenkins creates an impedance mismatch that is not easy to remove.• Although scripted syntax is powerful and expedites the creation of complex pipelines, it concurrently makes the pipeline hard to read and hard to manage	<p>The declarative pipeline follows a relatively newer syntax, which narrows the scope of the pipeline structure through a pre-defined "minimum necessary structure." On the technical front, the declarative syntax follows a conditional statement logic that generates a series of stages, executed sequentially throughout the pipeline's lifecycle. declarative pipelines don't make use of Groovy code, the declarative syntax doesn't access valuable APIs.</p>
<pre>node { stage('Build') { } stage('Test') { } stage('Deploy') { } }</pre>	<pre>pipeline { agent any stages { stage('Build') { steps { } } stage('Test') { steps { } } stage('Deploy') { steps { } } } }</pre>

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

The screenshot shows the Jenkins Pipeline configuration page. On the left, the 'Pipeline' tab is selected under 'Advanced project options'. The 'Repository URL' is set to 'https://github.com/klafawagdy/jenkins_Test/'. The 'Script Path' is set to 'Jenkinsfile'. The 'Lightweight checkout' checkbox is checked. Below the configuration, the 'Pipeline Syntax' section shows the following Jenkinsfile content:

```
[Pipeline] echo
testing your app
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (deployment)
[Pipeline] echo
your code is deployed right now
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

On the right, the 'stage view' shows a table of stage times:

Declarative: Checkout SCM	check	build	test	deployment
7s	881ms	784ms	674ms	637ms

Below the table, the 'Permalinks' section shows a table of build history:

Build	Run time
#1	33 Jan 2023, 00:43

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

Note: I named the main branch "main" not "master"

The screenshot shows the Jenkins Multibranch Pipeline configuration page. The 'Filter by name (with wildcards)' section is expanded, showing the 'Include' field with the value 'main dev test'. The 'Exclude' field is empty. The 'Switch branches/tags' section is also expanded, showing a list of branches: 'main' (checked), 'dev', 'extera', and 'test'. The 'main' branch is selected as the default.

Below the configuration, the 'Build History' table shows the following data:

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	dev	42 sec #1	N/A	25 sec
✓	☀	main	5 min 32 sec #1	N/A	20 sec
✓	☀	test	42 sec #1	N/A	23 sec