

## Jenkins Task 2

- 1 - What is Jenkins pipeline?
- 2 - What scripting language is Jenkins pipeline syntax based on?
- 3 - what are the different ways to trigger pipeline ?
- 4 - what is different between parameter and jenkins env variable?
- 5 - what is organization folder job and what is used for ?
- 6 - Create jenkins pipeline for your repo and use script file (jenkinsfile) to write pipeline syntax, include parameter functions and env variable in it ?
- 7 - Create another multibranch pipeline and filter branches to contain only (master , dev , test ) ?

## The Solution

- 1 - What is Jenkins pipeline?

Jenkins Pipeline is a suite of plugins that supports the implementation and integration of continuous delivery pipelines into Jenkins. It allows you to define and manage your build, test, and deployment processes as code. Jenkins Pipeline provides a way to create workflows that can be versioned, shared, and executed in a robust and scalable manner.

- 2 - What scripting language is Jenkins pipeline syntax based on?

The scripting language used in Jenkins Pipeline syntax is called Groovy. Groovy is a powerful, dynamic language that runs on the Java Virtual Machine (JVM) and provides a concise and expressive syntax for writing Jenkins Pipelines.

- 3 - What are the different ways to trigger pipeline ?

There are several ways to trigger a Jenkins Pipeline:

A - Triggering the pipeline manually by clicking on a button or link in the Jenkins web interface.

B - Triggering the pipeline automatically on a schedule using cron syntax.

C - Triggering the pipeline when changes are pushed to a version control system (e.g., Git) by configuring webhooks or polling.

D - Triggering the pipeline when a specific event occurs, such as a new artifact being published or an external system triggering a webhook.

#### 4 - What is different between parameter and jenkins env variable?

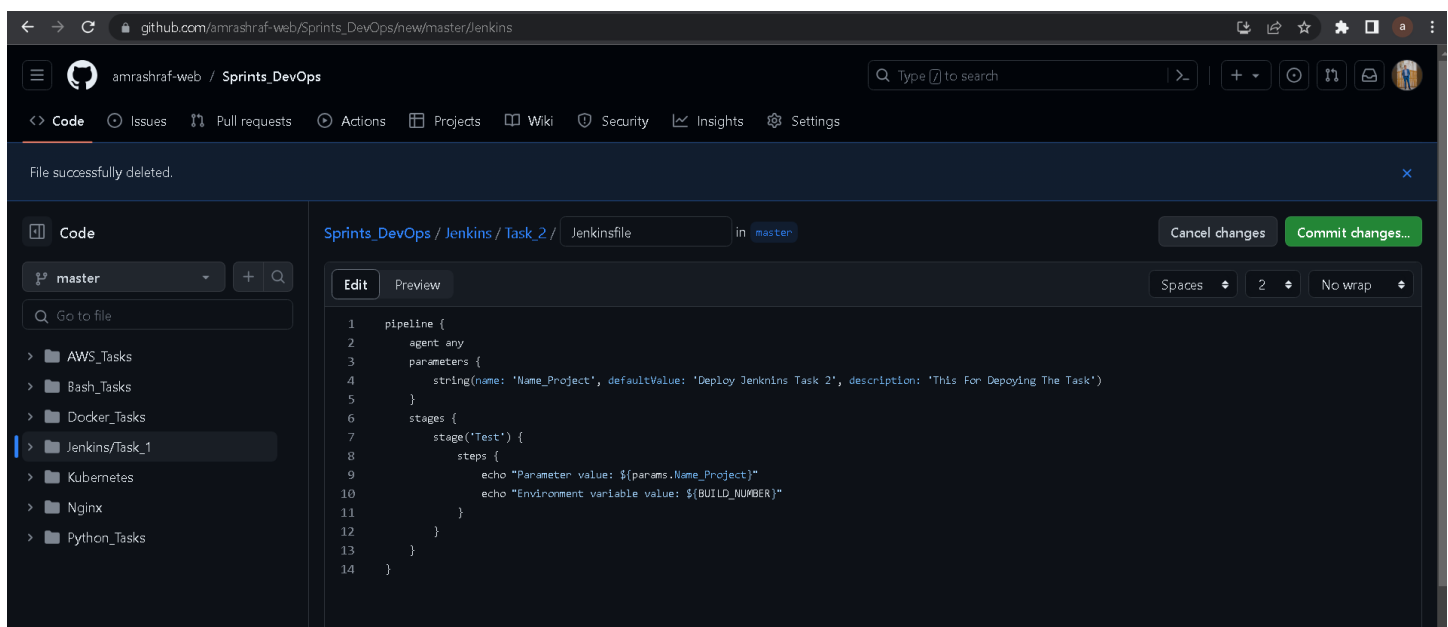
A - Parameters in Jenkins are values that can be passed into a pipeline at runtime. They allow users to customize the behavior of the pipeline by providing input values.

B - Jenkins environment variables, on the other hand, are predefined variables that store information about the Jenkins environment. They can be accessed and used within a pipeline to provide contextual information or configure specific behavior

#### 5 - What is organization folder job and what is used for ?

An organization folder job in Jenkins is a special type of job that allows you to organize and manage pipelines for multiple repositories or projects within a single Jenkins job. It provides a way to create and manage pipelines dynamically based on the repositories or projects present in a specified source code management system (e.g., Git).

#### 6 - Create jenkins pipeline for your repo and use script file (Jenkinsfile) to write pipeline syntax, include parameter functions and env variable in it ?



The screenshot shows a GitHub repository named 'amrashraf-web' with a branch 'Sprints\_DevOps'. The file explorer on the left shows a directory structure with folders like 'AWS\_Tasks', 'Bash\_Tasks', 'Docker\_Tasks', 'Jenkins/Task\_1', 'Kubernetes', 'Nginx', and 'Python\_Tasks'. The 'Jenkins/Task\_2' directory is selected, showing a file named 'Jenkinsfile'. The code editor displays the following Jenkinsfile syntax:

```
1 pipeline {
2   agent any
3   parameters {
4     string(name: 'Name_Project', defaultValue: 'Deploy Jenkins Task 2', description: 'This For Deploying The Task')
5   }
6   stages {
7     stage('Test') {
8       steps {
9         echo "Parameter value: ${params.Name_Project}"
10        echo "Environment variable value: ${BUILD_NUMBER}"
11      }
12    }
13  }
14 }
```

The interface includes a search bar at the top, navigation tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A notification bar at the top indicates 'File successfully deleted.' The code editor has tabs for 'Edit' and 'Preview', and a settings bar for 'Spaces' (set to 2) and 'No wrap'.

←

→

↻

localhost:8080/job/First\_Pipeline/configure

🔍

🔖

☆

⚙️

🗖

🔴

⋮

Dashboard > First\_Pipeline > Configuration

Configure

⚙️ General

🔧 Advanced Project Options

📄 Pipeline

Repository URL <sup>?</sup>

https://github.com/amrashraf-web/Sprints\_DevOps

Credentials <sup>?</sup>

amrashraf-web/\*\*\*\*\*

Add +

Advanced ▾

Add Repository

Branches to build <sup>?</sup>

Branch Specifier (blank for 'any') <sup>?</sup>

\*/master

Add Branch

Repository browser <sup>?</sup>

(Auto)

Additional Behaviours

Add +

Script Path <sup>?</sup>

Jenkins/Task\_2/Jenkinsfile

Save

Apply

Dashboard > First\_Pipeline > #1

```
> git fetch --tags --force --progress -- https://github.com/amrashraf-web/Sprints_DevOps +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/amrashraf-web/Sprints_DevOps # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f815cd7604a424a9aacc86ba0c5489e3ad0771fd (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f f815cd7604a424a9aacc86ba0c5489e3ad0771fd # timeout=10
Commit message: "Create Jenkinsfile"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Parameter value: Deploy Jenkins Task 2
[Pipeline] echo
Environment variable value: 1
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Dashboard > First\_Pipeline >

Status

Changes

Build with Parameters

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

## Pipeline First\_Pipeline

Add description

Disable Project

### Stage View

Jun 23 15:06

No Changes

27s

239ms

Declarative: Checkout SCM

Test

Average stage times: (Average full run time: ~58s)

27s

239ms

### Permalinks

- Last build (#1), 2 min 28 sec ago
- Last stable build (#1), 2 min 28 sec ago
- Last successful build (#1), 2 min 28 sec ago
- Last completed build (#1), 2 min 28 sec ago

Build History

trend

Filter builds...

Jun 23, 2023, 3:06 PM

Atom feed for all

Atom feed for failures

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

A – create two branch ( dev , test )

github.com/amrashraf-web/Sprints\_DevOps/branches/all

amrashraf-web / Sprints\_DevOps

Type / to search

Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Search branches...

Overview

Yours

Active

Stale

All branches

New branch

All branches

master

Updated 16 minutes ago by amrashraf-web

Default

test

Updated 16 minutes ago by amrashraf-web

0 | 0

New pull request

dev

Updated 16 minutes ago by amrashraf-web

0 | 0

New pull request

## Create multi branch pipeline on jenkins with filter branch by name

localhost:8080/job/multi-pipeline-branch/configure

Dashboard > multi-pipeline-branch > Configuration

### Configuration

- General
- Branch Sources
- Build Configuration
- Scan Multibranch Pipeline Triggers
- Orphaned Item Strategy
- Appearance
- Health metrics
- Properties

Filter by name (with regular expression)

Regular expression

Add

Property strategy

All branches get the same properties

Add property

Add source

### Build Configuration

Mode

by Jenkinsfile

Script Path

Jenkins/Task\_2/Jenkinsfile

Save Apply

localhost:8080/job/multi-pipeline-branch/indexing/console

Dashboard > multi-pipeline-branch > Scan Multibranch Pipeline

View as plain text

Multibranch Pipeline Events

Delete Multibranch Pipeline

People

Build History

Project Relationship

Check File Fingerprint

Rename

Pipeline Syntax

Credentials

Build Queue (3)

- multi-pipeline-branch » master
- multi-pipeline-branch » test
- multi-pipeline-branch » dev

Build Executor Status

- 1 Idle
- 2 Idle

```
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git ls-remote --symref -- https://github.com/amrashraf-web/Sprints_DevOps # timeout=10
> git rev-parse --resolve-git-dir /var/lib/jenkins/caches/git-f610194b705a57e966d91b78647ed285/.git # timeout=10
Setting origin to https://github.com/amrashraf-web/Sprints_DevOps
> git config remote.origin.url https://github.com/amrashraf-web/Sprints_DevOps # timeout=10
Fetching and pruning origin...
Listing remote references...
> git config --get remote.origin.url # timeout=10
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
> git ls-remote -h -- https://github.com/amrashraf-web/Sprints_DevOps # timeout=10
Fetching upstream changes from origin
> git config --get remote.origin.url # timeout=10
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress --prune -- origin +refs/heads/*:refs/remotes/origin/* # timeout=10
Checking branches...
Checking branch master
'Jenkins/Task_2/Jenkinsfile' found
Met criteria
Scheduled build for branch: master
Checking branch test
'Jenkins/Task_2/Jenkinsfile' found
Met criteria
Scheduled build for branch: test
Checking branch dev
'Jenkins/Task_2/Jenkinsfile' found
Met criteria
Scheduled build for branch: dev
Processed 3 branches
[Fri Jun 23 14:24:03 EET 2023] Finished branch indexing. Indexing took 3.3 sec
Finished: SUCCESS
```

←

→

↻

localhost:8080/job/multi-pipeline-branch/

🔍

🔖


☆

⚙️

🖨️

👤 a

⋮

 **Jenkins**

Search (CTRL+K) ⓘ

🛡️ 1

👤 Amr Ashraf Mohamed ▾

🚪 log out

Dashboard > multi-pipeline-branch >

Status

Configure

▶ Scan Multibranch Pipeline Now

📄 Scan Multibranch Pipeline Log

📅 Multibranch Pipeline Events

🗑️ Delete Multibranch Pipeline

👥 People

📧 Build History

🔗 Project Relationship


🔍 Check File Fingerprint

✏️ Rename

❓ Pipeline Syntax

🔑 Credentials

Build Queue ▾

 **multi-pipeline-branch**

Disable Multibranch Pipeline

Branches (3)

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✔️	☀️	dev	1 min 50 sec #1	N/A	1 min 6 sec	▶
✔️	☀️	master	1 min 50 sec #1	N/A	1 min 4 sec	▶
✔️	☀️	test	1 min 50 sec #1	N/A	1 min 35 sec	▶

Icon: S M L

Icon legend

📡 Atom feed for all

📡 Atom feed for failures

📡 Atom feed for just latest builds