ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY

CI/CD Using DevOps

Jenkins Pipeline

By
B Manikyala Rao M.Tech(Ph.d)
Assistant Professor
Dept of Computer Science & Engineering
Aditya College of Engineering & Technology
Surampalem

Jenkins Pipeline

- Jenkins Pipeline is a feature of the Jenkins build server, deployed as a plugin, that lets you implement continuous delivery (CD) pipelines on the Jenkins automation server.
- Continuous delivery pipelines are automated sequences of processes to deliver software from version control to customers and end-users. Each software change a developer commits in source control passes through a set of automated processes before being released to production. The pipeline involves building software using repeatable, reliable steps and pushing the build through various testing and deployment stages.
- Traditional Jenkins pipelines are scripted, meaning they prescribe the exact order of steps that need to happen in the pipeline.

Jenkins Pipeline Concepts

- **Pipeline** :The pipeline is a set of instructions given in the form of code for continuous delivery and consists of instructions needed for the entire build process. With pipeline, you can build, test, and deliver the application.
- **Node**:The machine on which Jenkins runs is called a node. A node block is mainly used in scripted pipeline syntax.
- **Stage**: A stage block contains a series of steps in a pipeline. That is, the build, test, and deploy processes all come together in a stage. Generally, a stage block is used to visualize the Jenkins pipeline process.
- **Step**:A step is nothing but a single task that executes a specific process at a defined time. A pipeline involves a series of steps.

Jenkins Scripted Pipeline

- Here is a simple example of a scripted Jenkinsfile. To begin with, we use the node statement, which says that this pipeline and any of its stages should be run on any available Jenkins agent.

Jenkinsfile (Scripted Pipeline)

node {

- Now we define several stage blocks. These are optional, but highly recommended because they make it easy to understand the tasks or steps occurring in each stage. Stage block descriptions are displayed in the Jenkins UI.
- Here we are defining three stages – Build, Test and Deploy
- Each of them can contain specific scripted code that performs the required operations.

Syntax

```
stage('Build') {
//
}
stage('Test') {
//
}
stage('Deploy') {
//
 }
```

Jenkins Declarative Pipeline

- Declarative pipelines require predefined constructs, and so are less flexible than scripted pipelines. On the other hand, they are easier to work with and do not require knowledge of Groovy code. Another plus is that Jenkins can automatically validate the syntax of a declarative pipeline.
- The code looks like this. We start with the pipeline statement that specifies this is a declarative pipeline.

Jenkinsfile (Declarative Pipeline)

pipeline {

- The agent any statement is a declarative syntax that tells Jenkins to allocate an executor on a node and create a workspace for the pipeline.

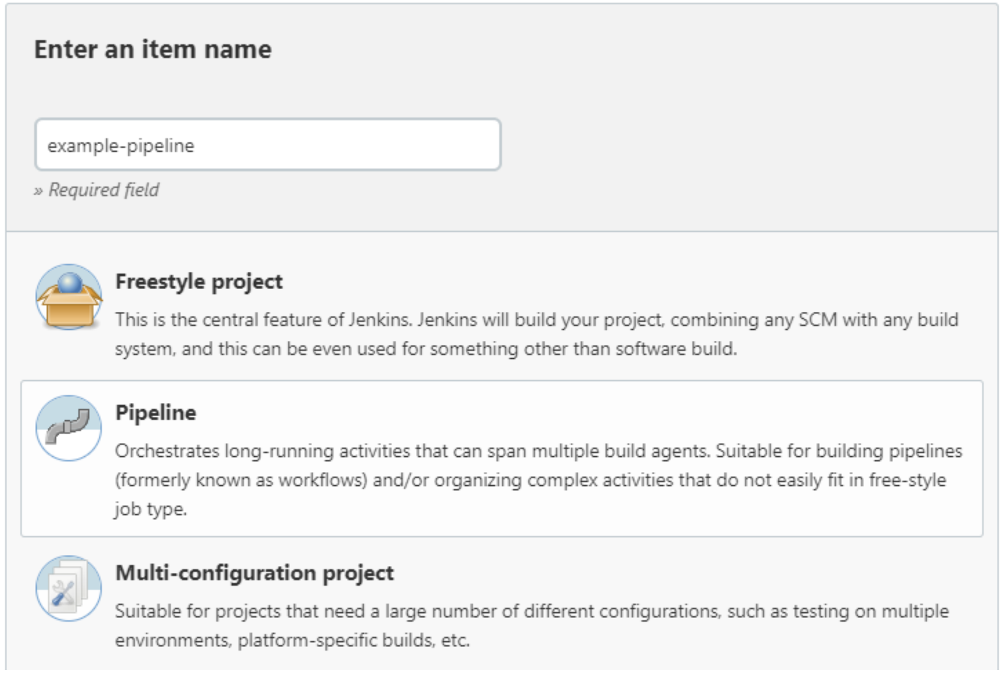
agent any

Example Script

```
pipeline {
    agent any
    stages {
            stage ('Build') {
                steps {
                        echo 'Running build phase...'
                }
            }
    }
}
```

Create a Jenkins Pipeline

1. Log into Jenkins.
2. In the Dashboard, select **New Item**.
3. Type an item name and select **Pipeline** from the list of item types. Click **OK**.



Enter an item name

example-pipeline

» *Required field*

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

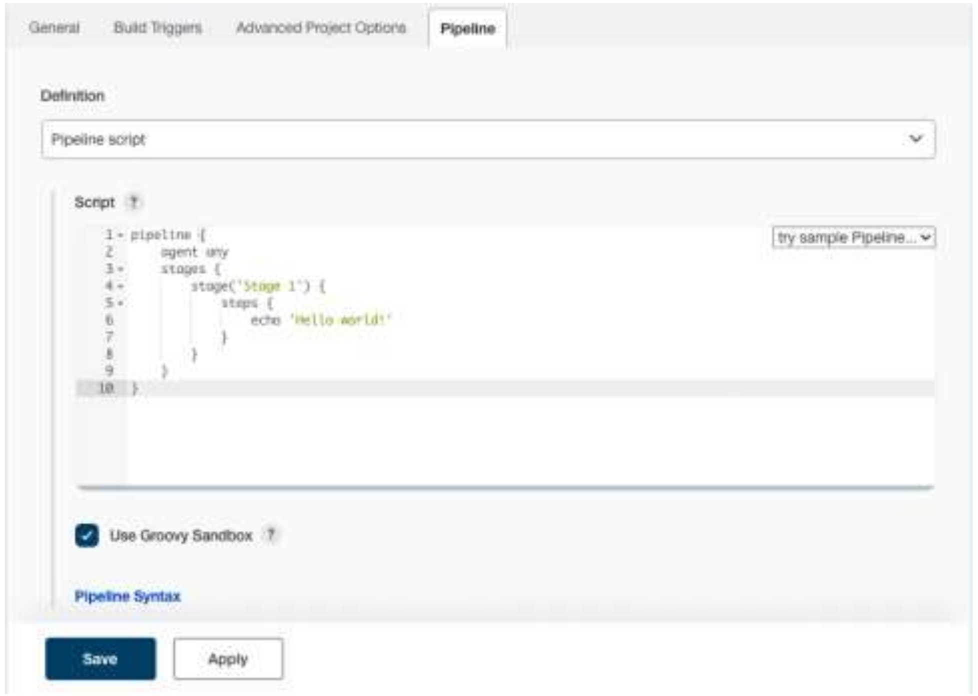
Multi-configuration project

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

Jenkins Pipeline

4.In the **Pipeline configuration** page, click the **Pipeline** tab. Under **Definition**, select the option **Pipeline script**.

5.Type your Pipeline code in the text area, as shown below.



Definition

Pipeline script

Script

```
pipeline {
    agent any
    stages {
        stage('Stage 1') {
            steps {
                echo 'Hello world!'
            }
        }
    }
}
```

✓ Use Groovy Sandbox

Jenkins Pipeline

6. Click **Save**. The **Pipeline project/item view** page appears. Click **Build Now**.
7. See full output from the pipeline run by clicking **Console Output**.

ANY QUERIES