Lab Experiment-6
Create pipeline using Jenkins File

NAME -Divyaansh Jain

ROLL NO – R171218040

SAP ID – 500067134

COURSE – B. Tech CSE- DevOps Batch 1

SUBJECT – Continuous Integration and Deployment

SEMESTER – 5th semester

Submitted To:

Hitesh Kumar Sir

Experiment for demonstrating how to create pipeline using Jenkins File

- Create a Jenkins File and push this project to GitHub. The Jenkins File is created such that the pipeline code will configure pipeline in two phases, first is build phase where "build stage" is the output and second is test phase where the command "mvn clean" is executed.

```
pipeline{
  agent any
  stages{
   stage('Build'){
      steps{
      echo 'Build Stage'
      }
   }
   stage('Test'){
      steps{
      shell('mvn clean')
      }
    }
   }
}
```


Divyaansh313 / maven-jenkins
forked from ArjunRekhi/maven-jenkins

<> Code   Pull requests   Actions   Projects   Wiki   Security   Insights   Settings

master   1 branch   0 tags   Go to file   Add file   Code

This branch is even with ArjunRekhi:master.   Pull request   Compare

ArjunRekhi Update pom.xml   6327a4c 15 days ago   9 commits

my-app   Update pom.xml   15 days ago
Jenkinsfile   Jenkinsfile added   2 months ago

Help people interested in this repository understand your project by adding a README.   Add a README

- Create a new Pipeline Project.

Enter an item name

Jenkins_File

» 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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

Bitbucket Team/Project
Scans a Bitbucket Cloud Team (or Bitbucket Server Project) for all repositories matching some defined markers.

Folder

OK container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a namespace, so you can have multiple things of the same name as long as they are in different folders.

- In the Pipeline Options, configure the job to fetch the Jenkins file from GitHub and execute the jobs as written in the file. Note that the name of file should be "JenkinsFile" without any extension.

General Build Triggers Advanced Project Options **Pipeline**

Pipeline

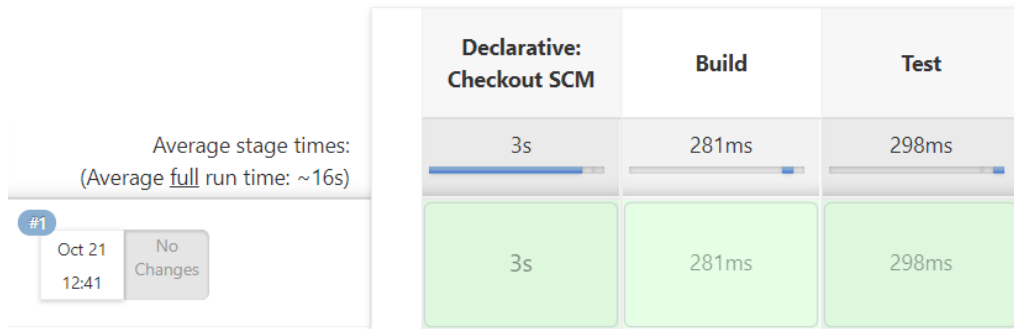|	
Definition	Pipeline script from SCM
SCM	Git
Repositories	
Repository URL	https://github.com/Divyaansh313/maven-jenkins
Credentials	- none - Add
	Advanced...
	Add Repository
Branches to build	
Branch Specifier (blank for 'any')	*/master
	Add Branch
Repository browser	(Auto)
Additional Behaviours	Add
Script Path	Jenkinsfile
Lightweight checkout	☑

Save **Apply**

- Saving this project and running/building this project. A pipeline will be shown at the screen giving the build time for both the phases.

Stage View

	Declarative: Checkout SCM	Build	Test
Average stage times: (Average full run time: ~16s)	3s	281ms	298ms
#1 Oct 21 12:41 — No Changes	3s	281ms	298ms

- The console output for the project is given below.

Console Output

```
Started by user Divyaansh Jain
Obtained Jenkinsfile from git https://github.com/Divyaansh313/maven-jenkins
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\Users\ASUS\.jenkins\workspace\Jenkins_File
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git.exe
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Divyaansh313/maven-jenkins
 > git.exe init C:\Users\ASUS\.jenkins\workspace\Jenkins_File # timeout=10
Fetching upstream changes from https://github.com/Divyaansh313/maven-jenkins
 > git.exe --version # timeout=10
 > git --version # 'git version 2.24.0.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/Divyaansh313/maven-jenkins +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe config remote.origin.url https://github.com/Divyaansh313/maven-jenkins # timeout=10
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
Checking out Revision 6327a4cabca080403526a2d20d2da4e4d2e4f438 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 6327a4cabca080403526a2d20d2da4e4d2e4f438 # timeout=10
Commit message: "Update pom.xml"
First time build. Skipping changelog.
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Build Stage
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```