THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Rohan Asnani - B12-28

Experiment No. 04

Aim :- To implement continous integration using Jenkins.

Theory :-
Jenkins is a popular open-source tool to perform continous integration and build automation. Jenkins allow to execute a predefined kids of steps, eg to compile Java source code and build a JAR from the resulting classes. The trigger for this execution can be time or event based. Jenkins monitors the execution of the steps and allows to the process if one of the steps fails. For example, you can install plugins to support building and testing Android applications.

Continuous integration is a process in which all development work is integrated as early as possible. The resulting artifacts are automatically created and tested. This process allows to identify errors in an early stage of the project. The Jenkins build server is a tool to provide this functionality.

A pipeline in a Jenkins is a collection of jobs that brings the software from version control into the hands of the end-users by using automation tools. It is a feature used to incorporate continous delivery in our software development workflow. Jenkins pipelines help you align the build process of a project.

P.T.O

This is done by specifying tasks and the order in which they are executed. When several different changes are made by several developers at once, so it is useful to know which change is getting tested or which change is sitting in the queue or which build is broken. This is where pipeline comes into picture. The Jenkins Pipeline gives you an overview of where tests are up to. In build pipeline the build as a whole is broken down into sections, such as the unit test, acceptance test, packaging, reporting and deployment phase. The pipeline phases can be executed in series or parallel, and if one phase is successful, it automatically moves on to the next phase (hence the relevance of the name "pipeline").

Setup of Pipelines.

Jenkins allows to specify pipelines using a Jenkinsfile. This is just a textfile that contains the necessary data for Jenkins to execute the pipeline. It is called Jenkinsfile (notice: No file extensions) and should be placed in the root of your project.

Jenkins support two different syntaxes,

1. Declarative (since Pipeline version 2.5)
2. Scripted.

T.Y.O

Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source a source control management system such as Git. Whereas the scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.

The groovy syntax for pipeline is shown below.

```
pipeline {
    agent any
    stages {
        stage ('Build') {
            ....
        }
        stage ('Test') {
            ...
        }
        stage ('QA') {
            ...
        }
        stage ('Deploy') {
            ...
        }
        stage ('Monitor') {
        }
    }
}
```

1.1.0.

The steps to create a pipeline are give as follows:

Step 1:- Log into Jenkins and select 'New item' from the dashboard.

Step 2:- Next, enter a name for your pipeline and select 'pipeline' project. Click on 'OK' to proceed.

Step 3:- Scroll down to the pipeline and choose if you want a declrative pipeline or a scripted one.

Step 4a:- If you want a scripted pipeline then choose 'pipeline script' and start typing your code.

Step 4b:- If you want a declarative pipeline then select 'pipeline script from SCM' and choose your SCM. In my case I'm going to use Git throughout this demo. Enter your repository URL.

Step 5:- Within the script path is the name of Jenkinsfile that is going to be accessed from your SCM to run. Finally click on "apply" and "save". You have successfully created your first Jenkins pipeline.

Step 6:- To run pipeline, click on Build option followed by Full stage view to see the different

THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

Conclusion:- Thus, we have successfully studied and implemented continuous integration using Jenkins.