**EXPERIMENT-5**

**DEVOPS**

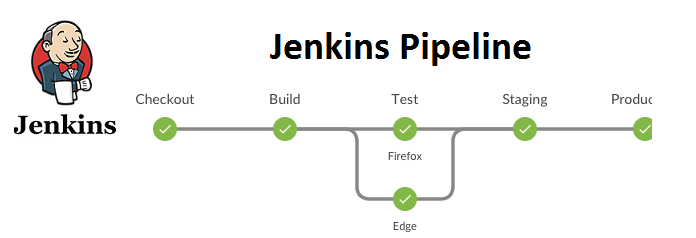
**Aim:** To build the pipeline of jobs using Maven / Gradle/ Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.

**LO:** **3 -** Illustrate the importance of Jenkins to Build and Deploy Software Applications on a Server Environment.

**THEORY:**

**What is Jenkins Pipeline?**

* In Jenkins, a pipeline is a collection of events or jobs which are interlinked with one another in a sequence.
* It is a combination of plugins that support the integration and implementation of **continuous delivery pipelines** using Jenkins.
* In other words, a Jenkins Pipeline is a collection of jobs or events that brings the software from version control into the hands of the end users by using automation tools. It is used to incorporate continuous delivery in our software development workflow.
* A pipeline has an extensible automation server for creating simple or even complex delivery pipelines "as code", via DSL (Domain-specific language).



## JenkinsFile

Jenkins Pipeline can be defined by a text file called JenkinsFile. You can implement pipeline as code using JenkinsFile, and this can be defined by using a DSL (Domain Specific Language). With the help of JenkinsFile, you can write the steps required for running a Jenkins Pipeline.

The benefits of using JenkinsFile are:

* You can make pipelines automatically for all branches and can execute pull requests with just one JenkinsFile.
* You can review your code on the pipeline.
* You can review your Jenkins pipeline.
* This is the singular source for your pipeline and can be customized by multiple users.

JenkinsFile can be defined by using either Web UI or with a JenkinsFile.

**Pipeline syntax**

Two types of syntax are used for defining your JenkinsFile.

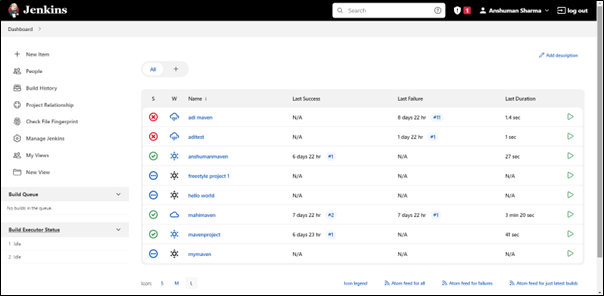
* Declarative
* Scripted

**Here are the reasons why you should use Jenkins pipeline:**

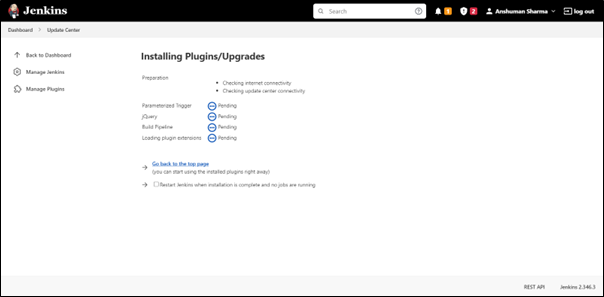
* Jenkins pipeline is implemented as a code which allows several users to edit and execute the pipeline process.
* Pipelines are robust. So if your server undergoes an unpredicted restart, the pipeline will be automatically resumed.
* You can pause the pipeline process and make it wait to continue until there is an input from the user.
* Jenkins Pipelines support big projects. You can run many jobs, and even use pipelines in a loop.

**OUTPUT:**

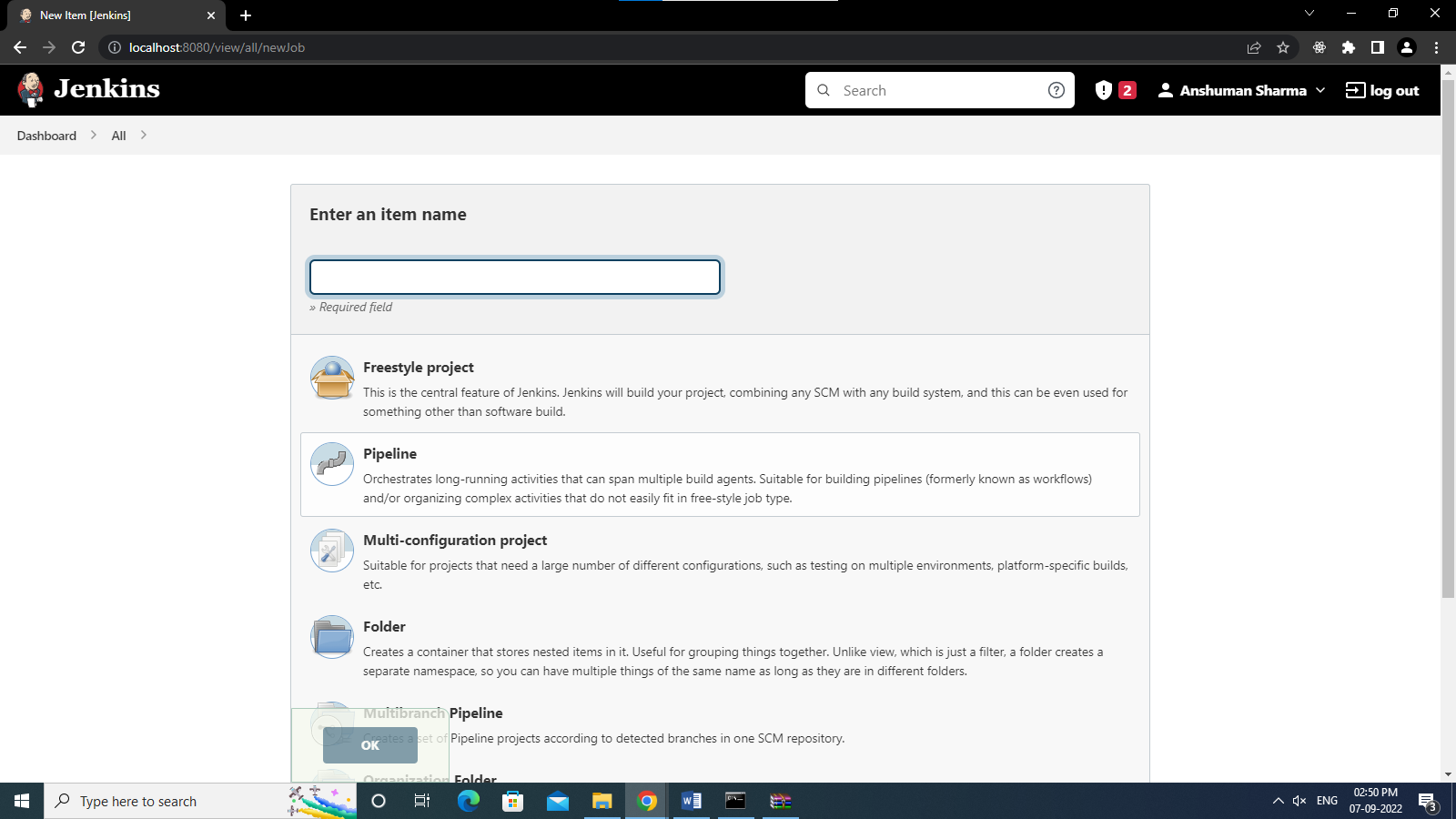
Step 1: Login into the Jenkins Dashboard



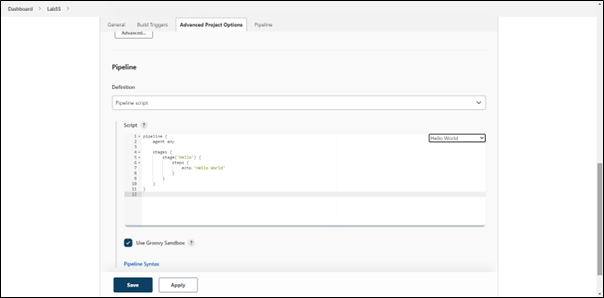
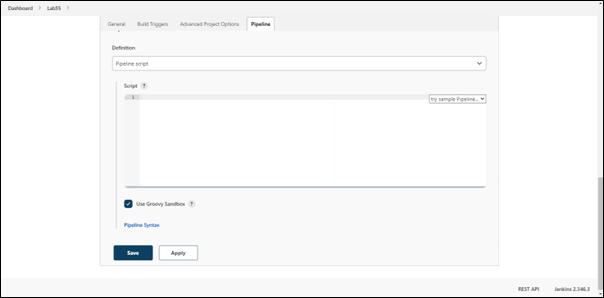
Step 2: Installing the necessary plugins



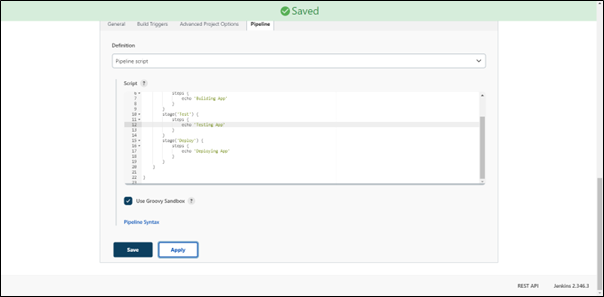
Step 3: Build an item



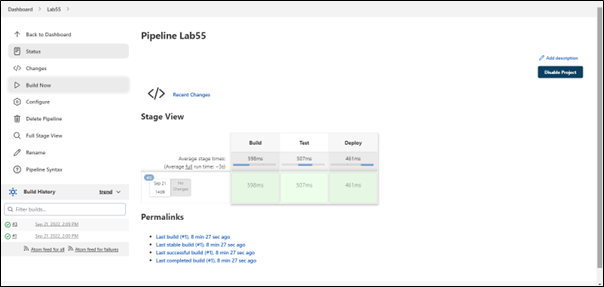
Step 4: Go through with the pipeline script



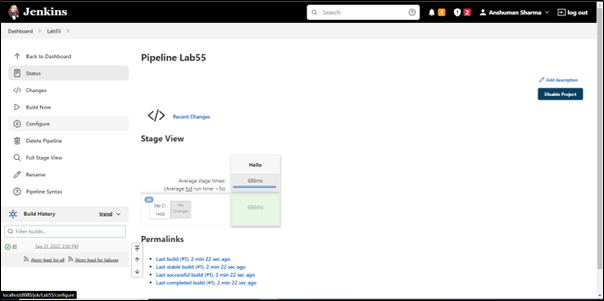
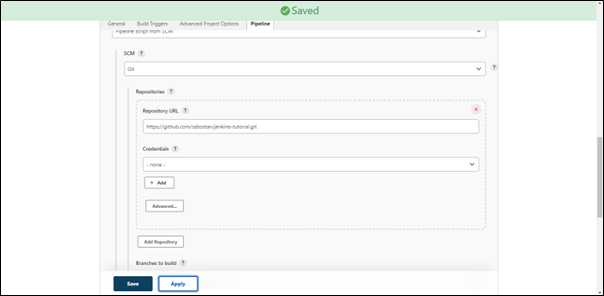
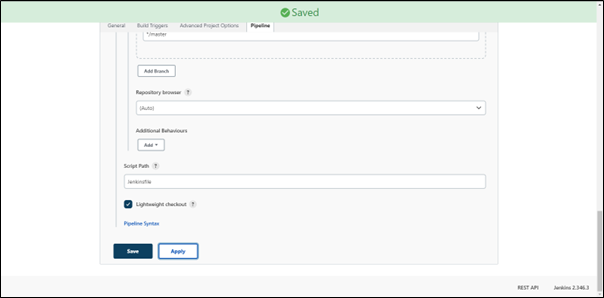
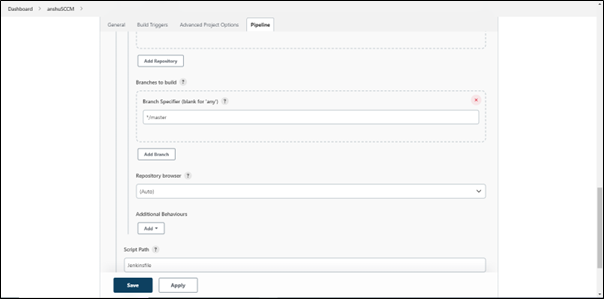
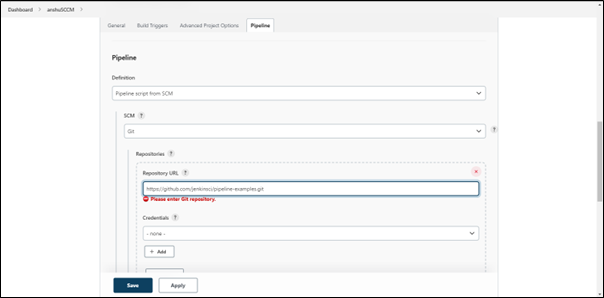
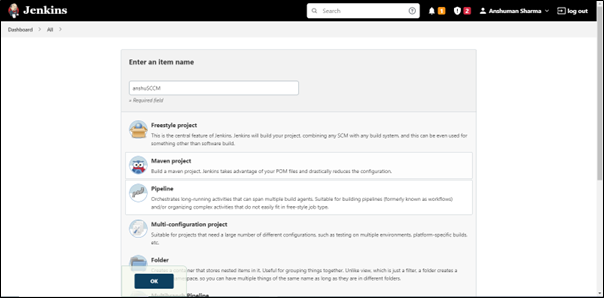
Step 5: Apply and save



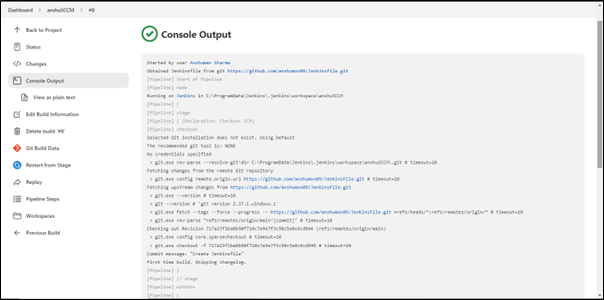
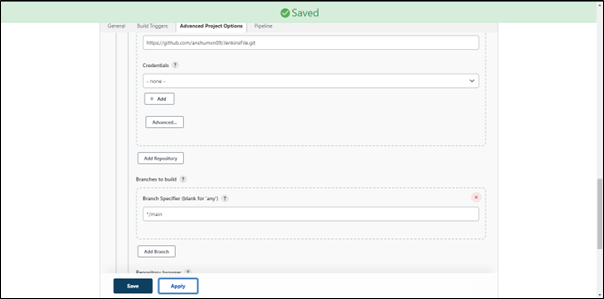
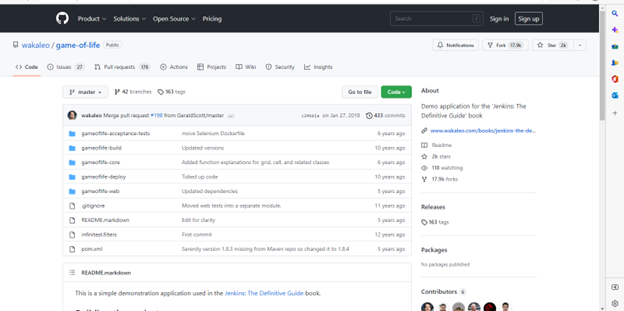
Step 6 : Click on the build now



Step 7 : Performing Pipeline script SCM



Step 8: Performing the pipeline script through GIT Repository.



CONCLUSION:

LO3 and PO1-PO5, PO12.