**EXPERIMENT NO. 5**

**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.

**THEORY :**

**Jenkins Pipeline**

Jenkins Pipeline is a combination of plugins that supports integration and implementation of continuous delivery pipelines. It has an extensible automation server to create simple and complex delivery pipelines as code via pipeline DSL. A Pipeline is a group of events interlinked with each other in a sequence.

In a Jenkins pipeline, every job or event has some sort of dependency on at least one or more events.

The benefits of using Jenkins are:

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

**Maven Jenkins**

Maven is used to define project structure, dependencies, build, and test management.

Using pom. xml(Maven) you can configure dependencies needed for building testing and running code.

Maven automatically downloads the necessary files from the repository while building the project.

**Purpose of Maven :**

* A maven is a build tool designed to manage dependencies and the software lifecycle. It is also designed to work with plugins that allow users to add other tasks to the standard compile, test, package, install, deploy tasks.
* Jenkins is designed for the purpose of implementing Continuous Integration (CI). It checks code out of a repository, builds and packages it, and sends it out to a server for testing – automatically. Jenkins can use Maven as its build tool.

**Gradle Jenkins :**

Gradle is managed as another tool inside Jenkins (the same way as Ant or Maven), including support for automatic installation and a new build step is provided to execute Gradle tasks.

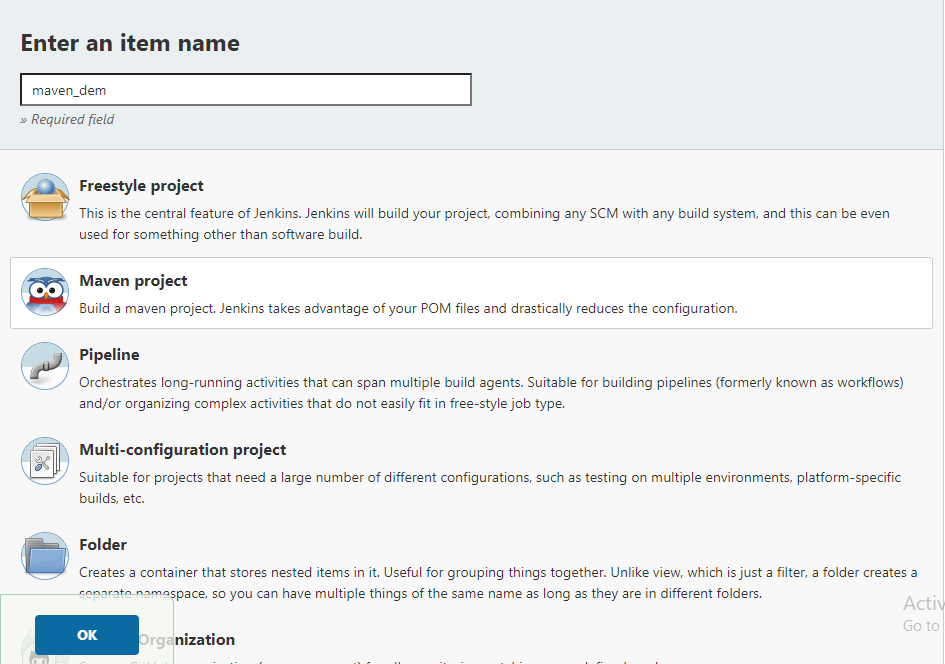
It also allows detecting Build Scans in arbitrary console logs, for Maven and Gradle builds and display them in the Jenkins UI. It is a powerful build tool for the JVM.

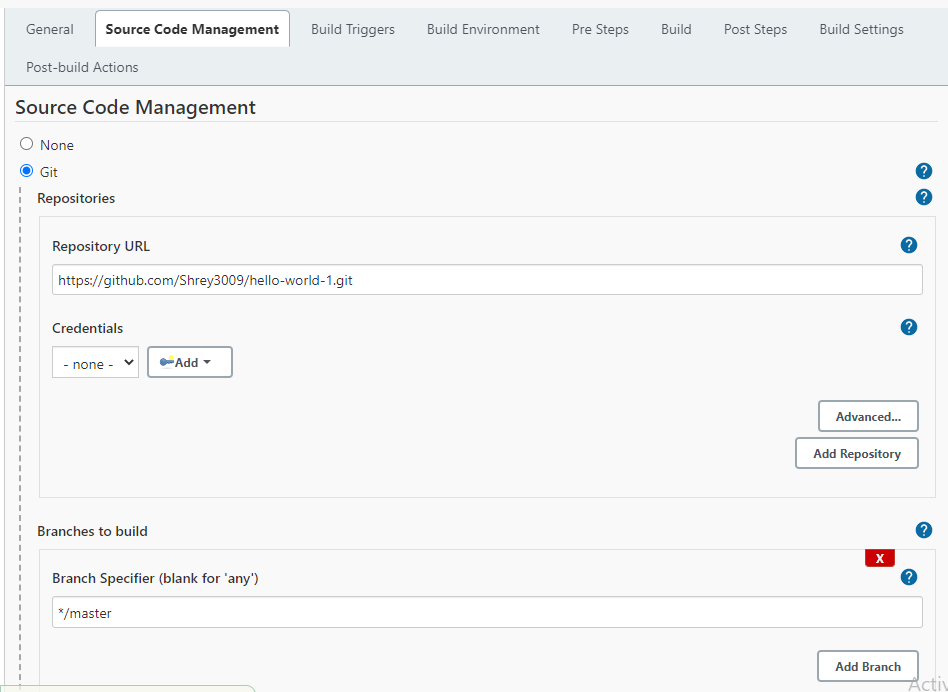
It primarily focuses on build automation and supports multi-language development.

If we are building, testing, publishing, and deploying software on any platform, Gradle provides a flexible model to support the entire development lifecycle from compiling and deploying the project.

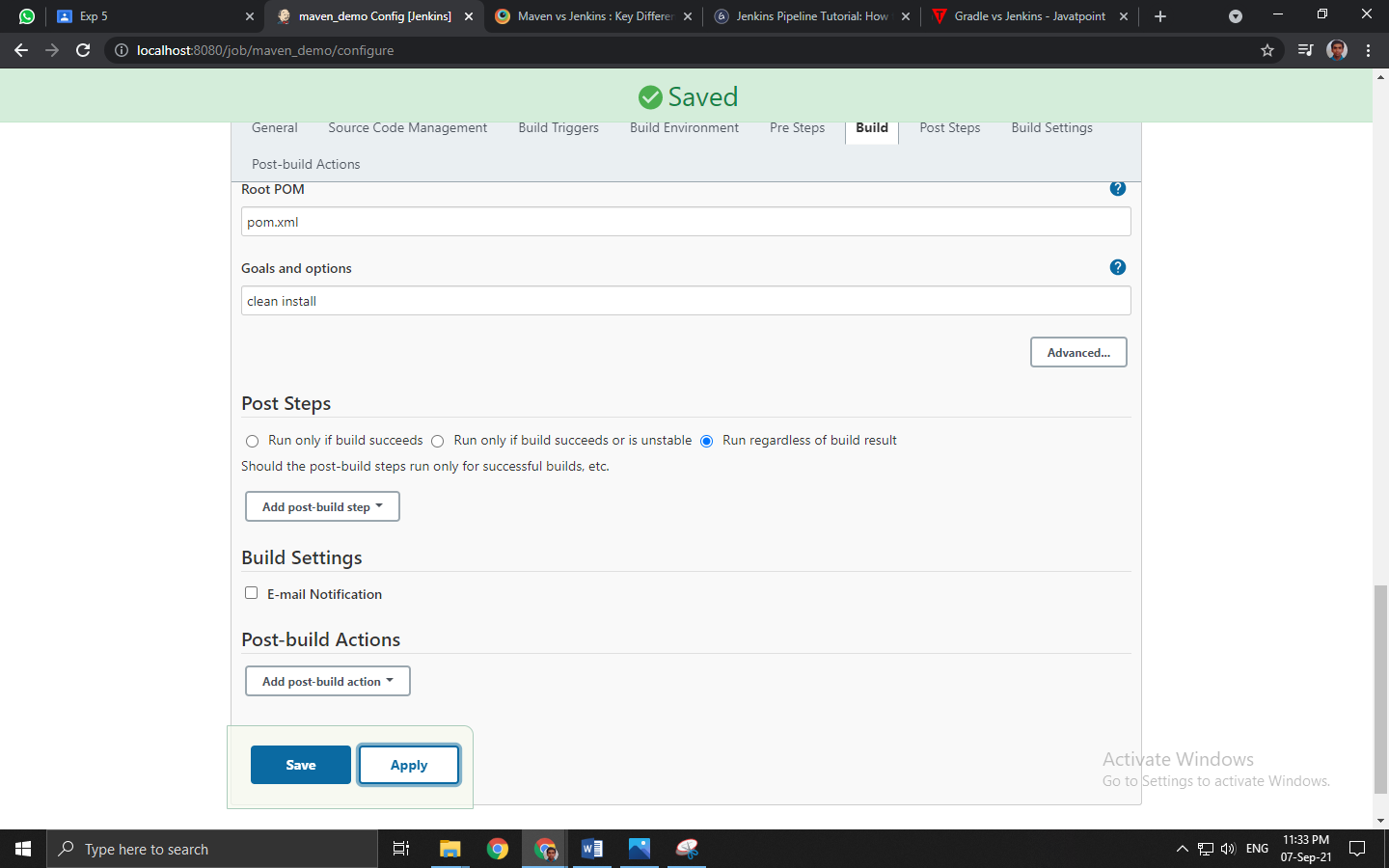
**OUTPUT :**

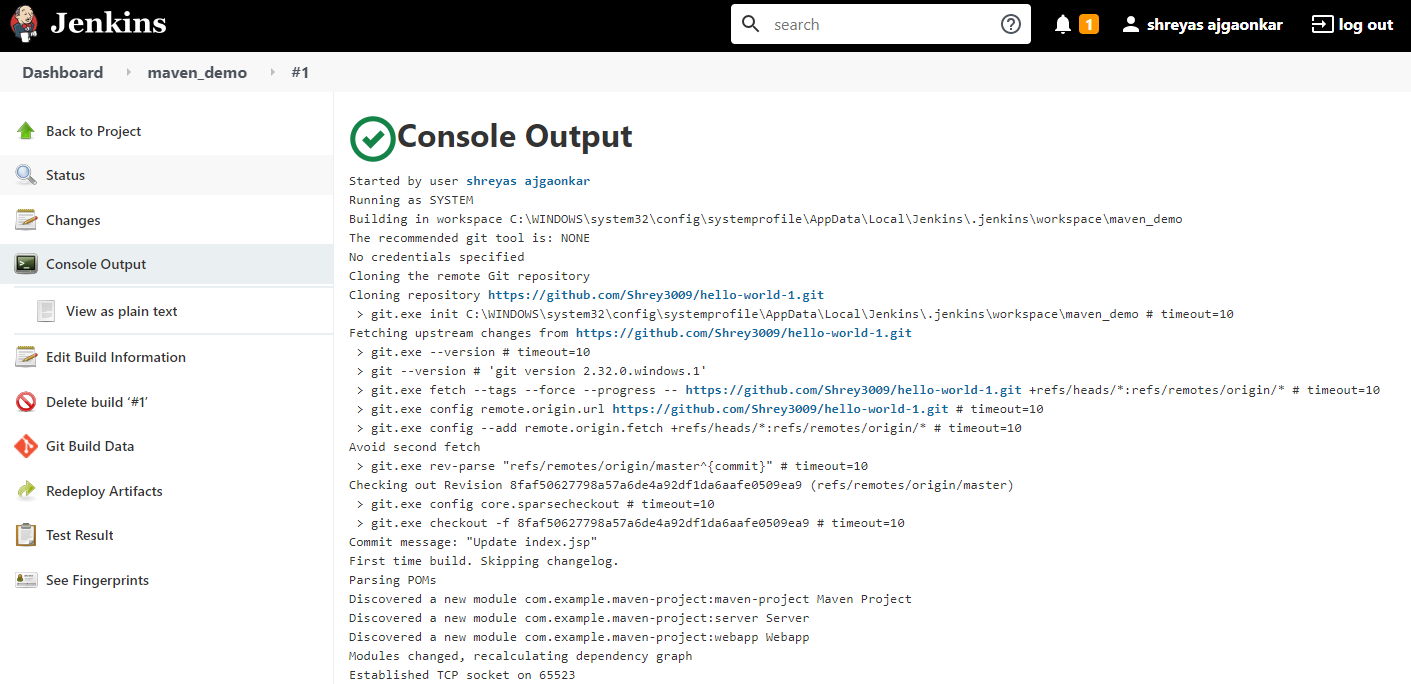
**Installing Maven in Jenkins :**

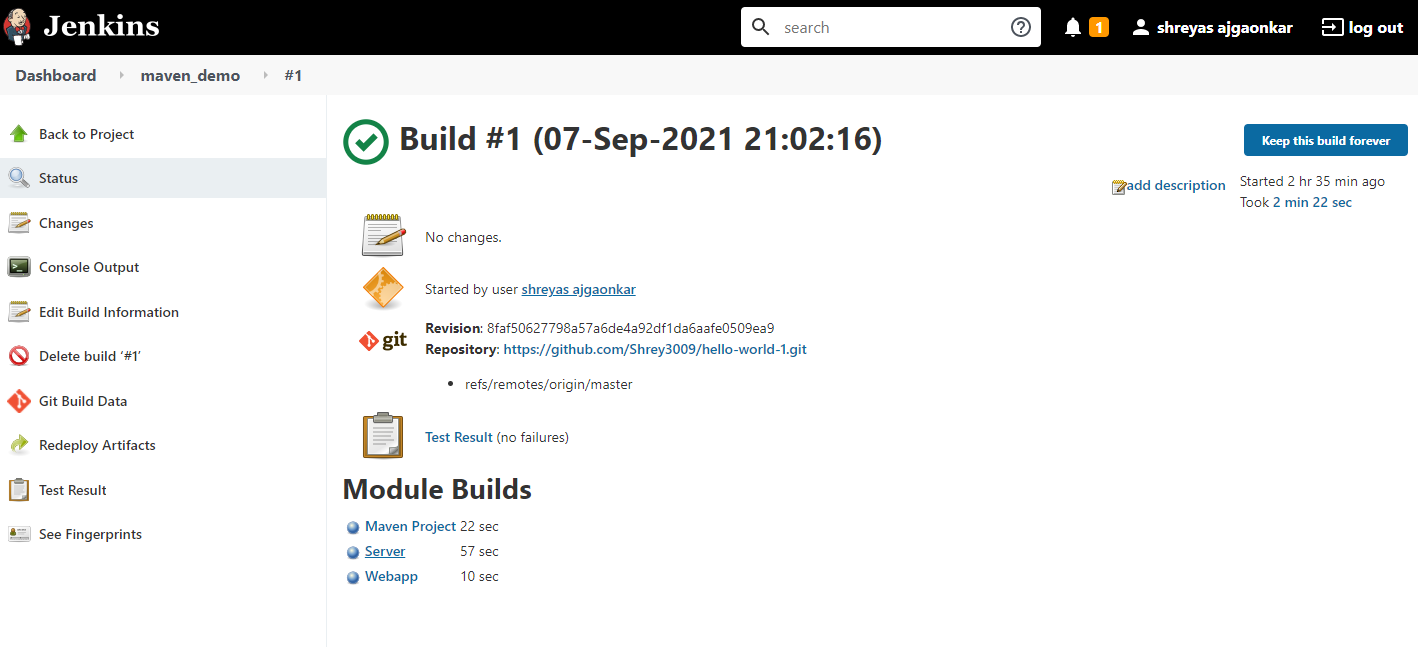




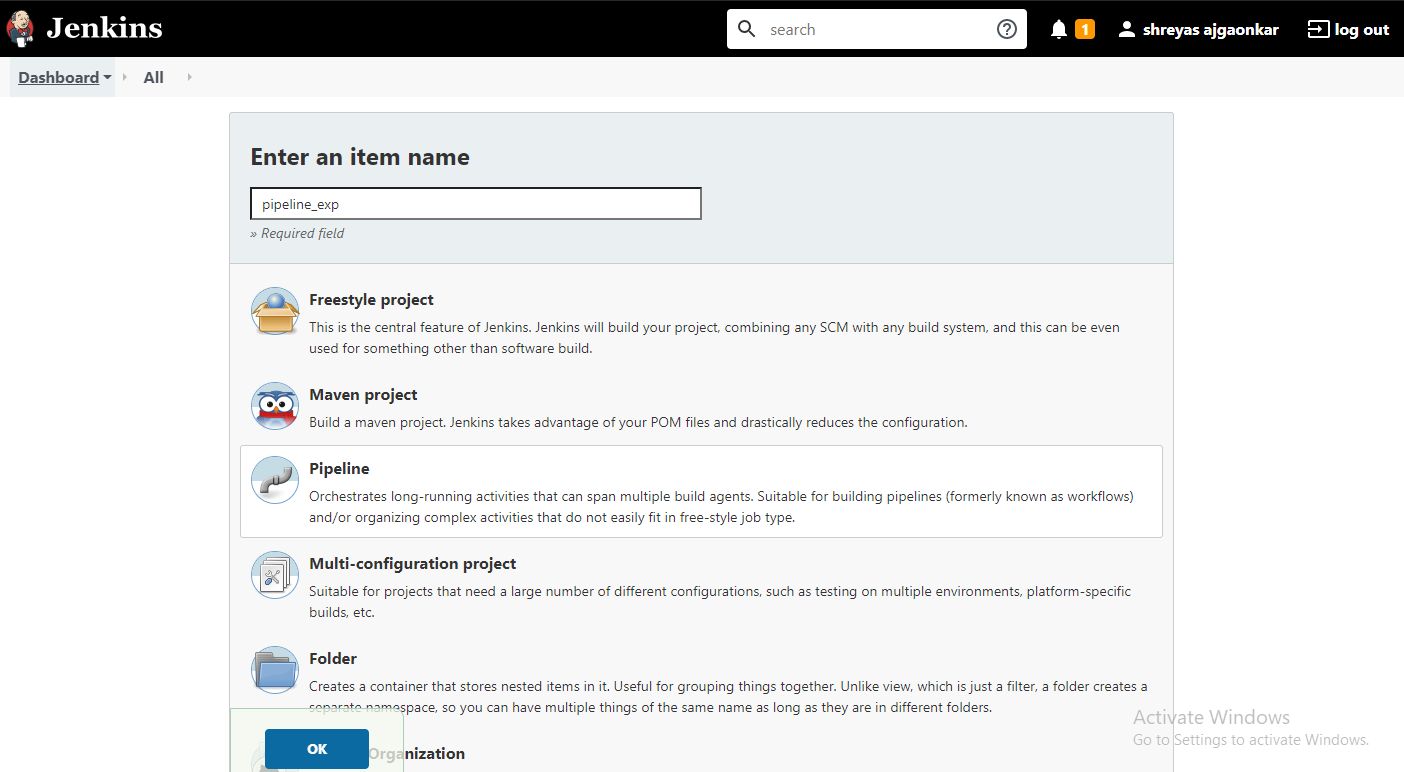
https://github.com/jay-2000/test

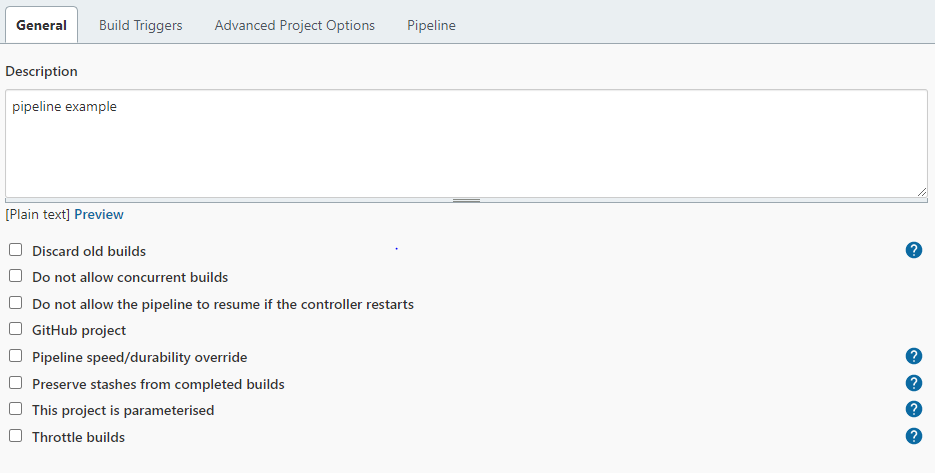
****

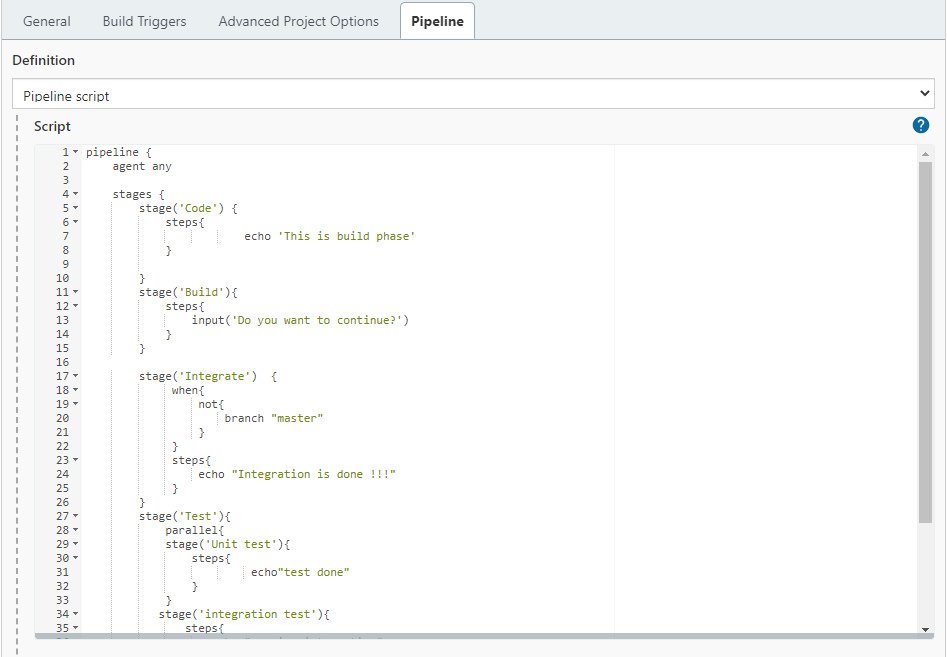


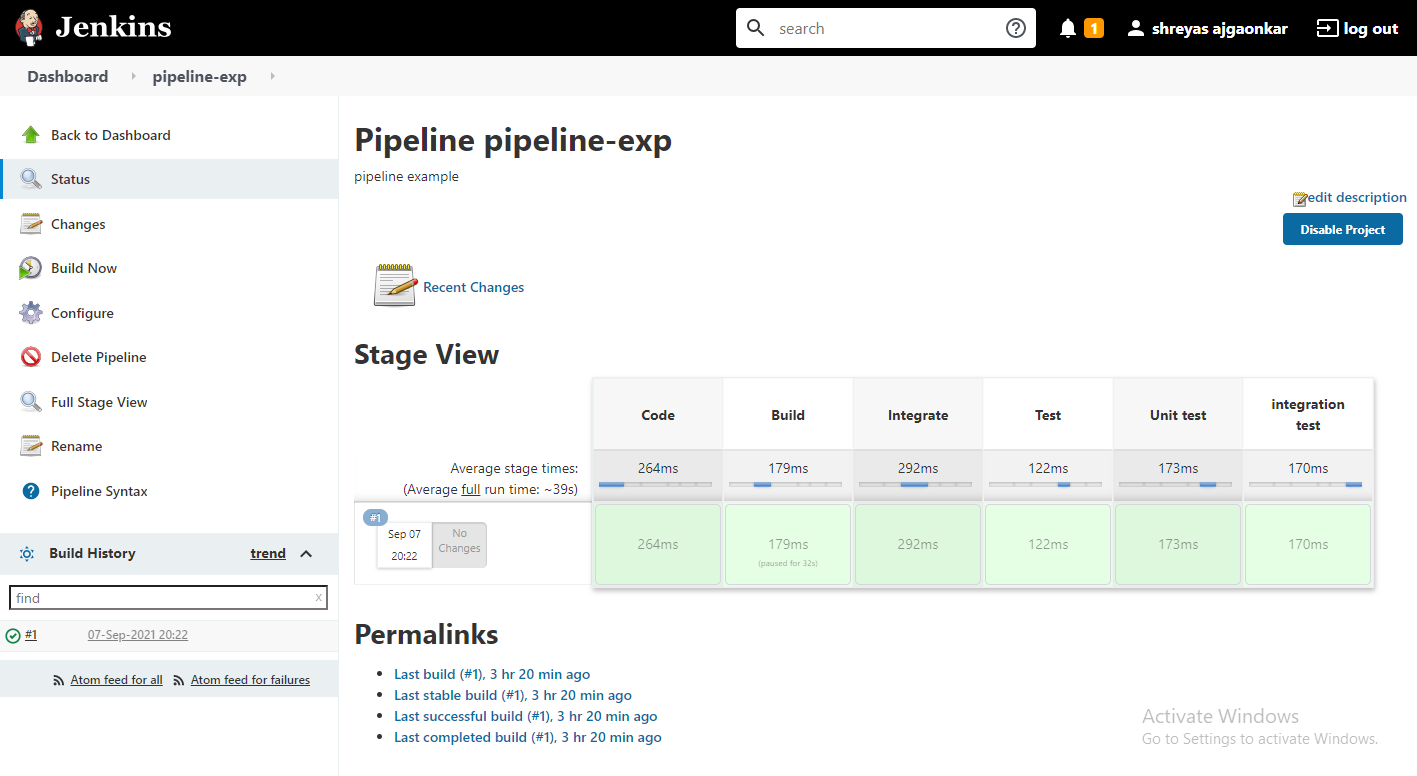


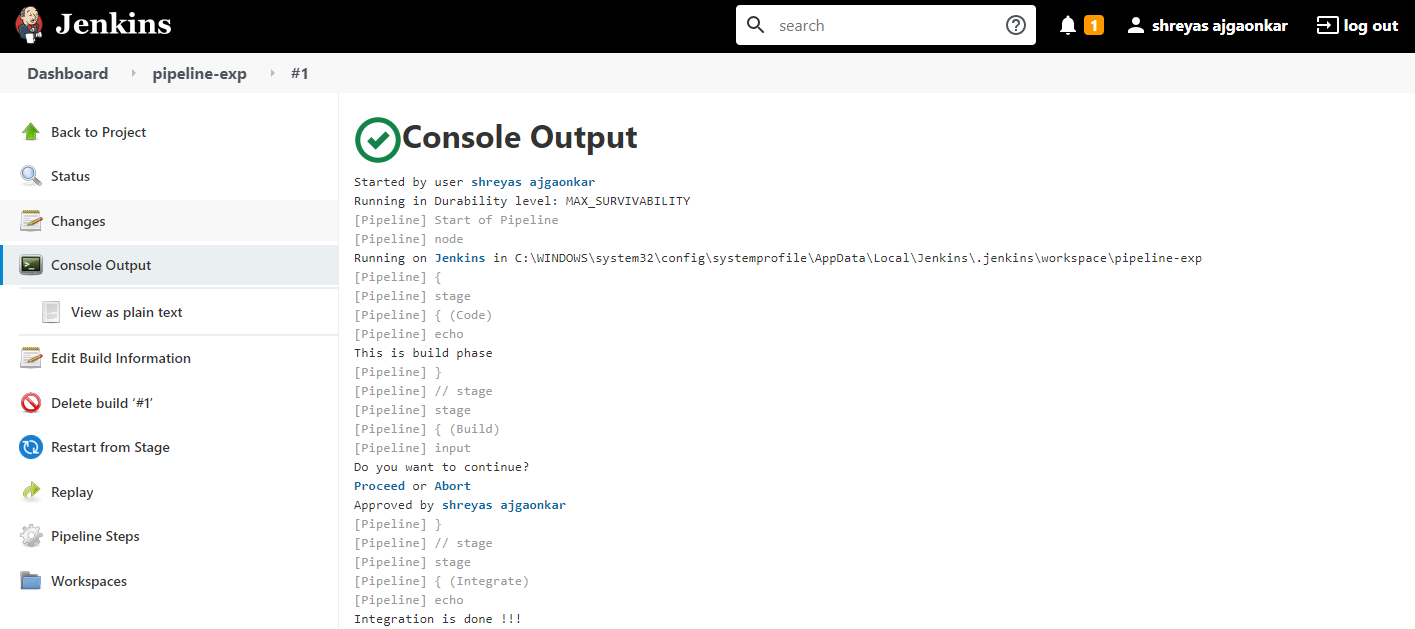
**Pipeline of jobs in Jenkins :**











admin

admin



**CONCLUSION :** Thus we successfully studied about pipeline of jobs using Maven / Gradle / Ant in Jenkins, created a pipeline script to Test and deploy an application over the tomcat server.