Experiment 05

Aim : To build the pipeline of jobs using Maven/Gradle/Ant in Jenkins , create a pipeline script to test and deploy an application

Theory : Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating *continuous delivery pipelines* into Jenkins.
A *continuous delivery (CD) pipeline* is an automated expression of your process for getting software from version control right through to your users and customers.

the definition of a Jenkins Pipeline is written into a text file (called a Jenkinsfile) which in turn can be committed to a project's source control repository.

Creating a Jenkinsfile and committing it to source control provides a number of immediate benefits:

- Automatically creates a Pipeline build process for all branches and pull requests.
- Code review/iteration on the Pipeline (along with the remaining source code).
- Audit trail for the Pipeline.

By modeling a series of related tasks, users can take advantage of the many features of Pipeline:

- **Code**: Pipelines are implemented in code and typically checked into source control, giving teams the ability to edit, review, and iterate upon their delivery pipeline.
- **Durable**: Pipelines can survive both planned and unplanned restarts of the Jenkins controller.
- **Pausable**: Pipelines can optionally stop and wait for human input or approval before continuing the Pipeline run.
- **Versatile**: Pipelines support complex real-world CD requirements, including the ability to fork/join, loop, and perform work in parallel.
- **Extensible**: The Pipeline plugin supports custom extensions to its DSL [1] and multiple options for integration with other plugins.

Pipeline concepts

Pipeline

A Pipeline is a user-defined model of a CD pipeline. Also, a pipeline block is a key part of Declarative Pipeline syntax.

Node

A node is a machine which is part of the Jenkins environment and is capable of executing a Pipeline.

Stage

A stage block defines a conceptually distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages).

Step

A single task. Fundamentally, a step tells Jenkins *what* to do at a particular point in time (or "step" in the process). For example, to execute the shell command make use the sh step: sh 'make'.

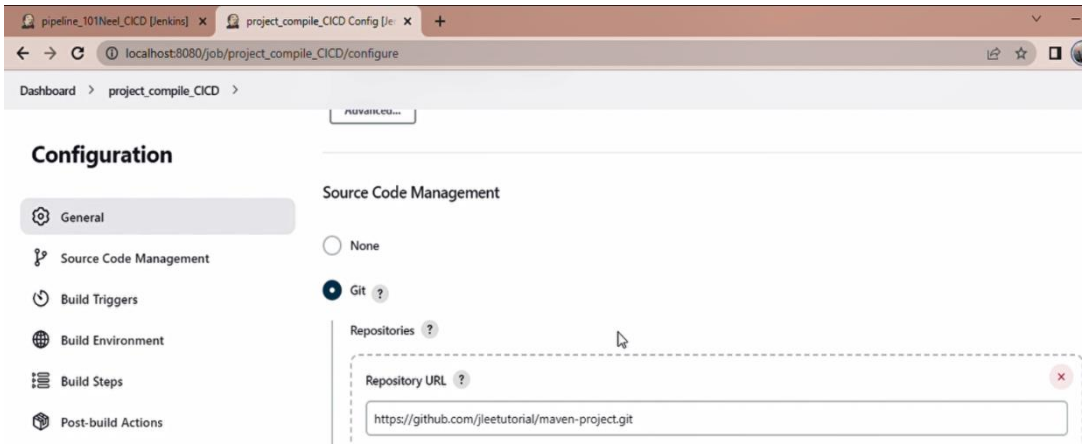
Procedure :

1. First Build a pipeline by naming it and choosing pipeline in the project type section. We will then create three projects to build a pipeline , mainly a test

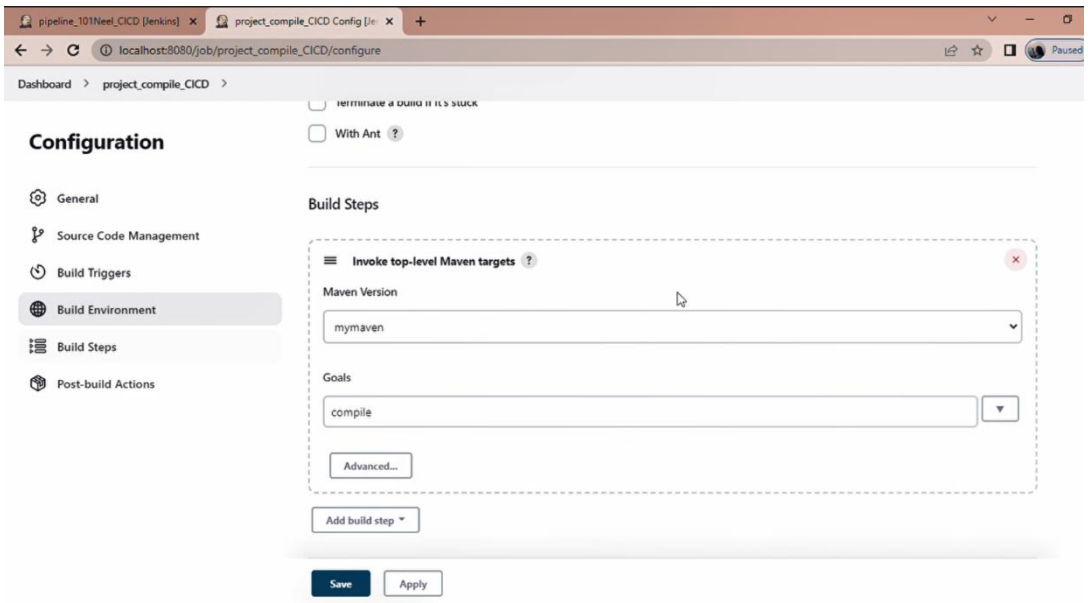
project , a compile project and a package project .

2. Now click on the compile project and perform the following actions , go to the source code management and paste the following Git repo link.

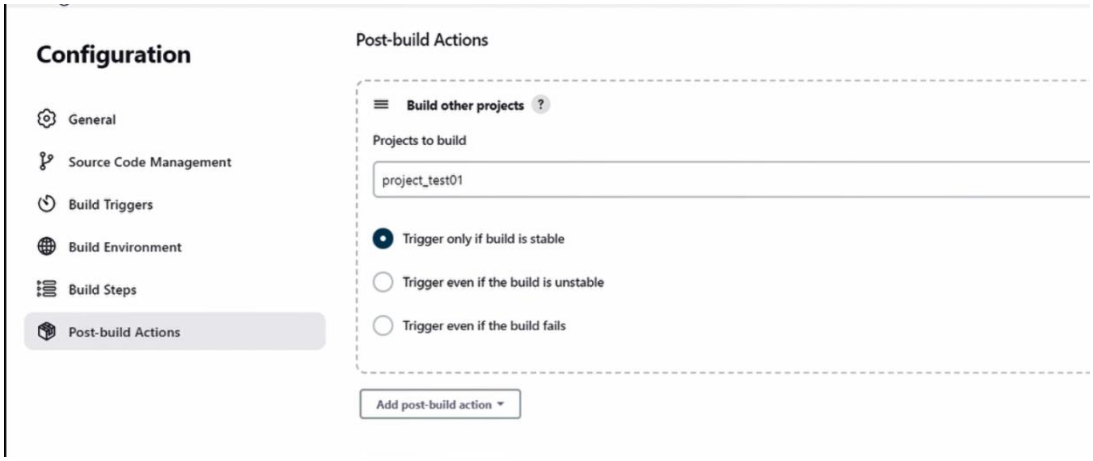
   https://github.com/jleetutorial/maven-project/



3. Go to the Add Build Step action and select invoke top-level maven targets and select the <u>mymaven</u> version and in goals write <u>compile</u>



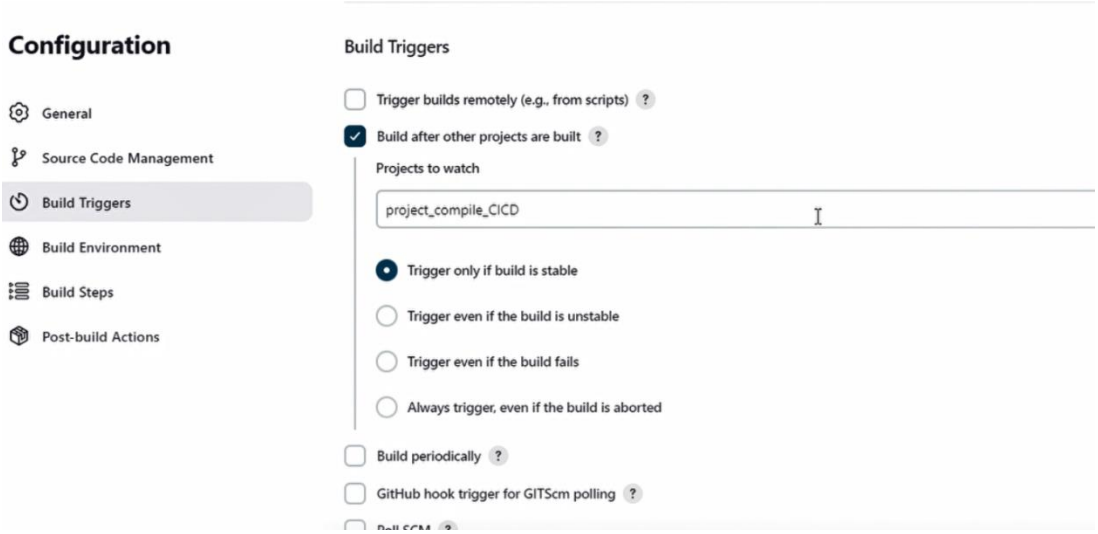
4. Go to post build section and choose build other projects and in that add your test project (desired name) and keep the default trigger setting . after that click on apply and save respectively .



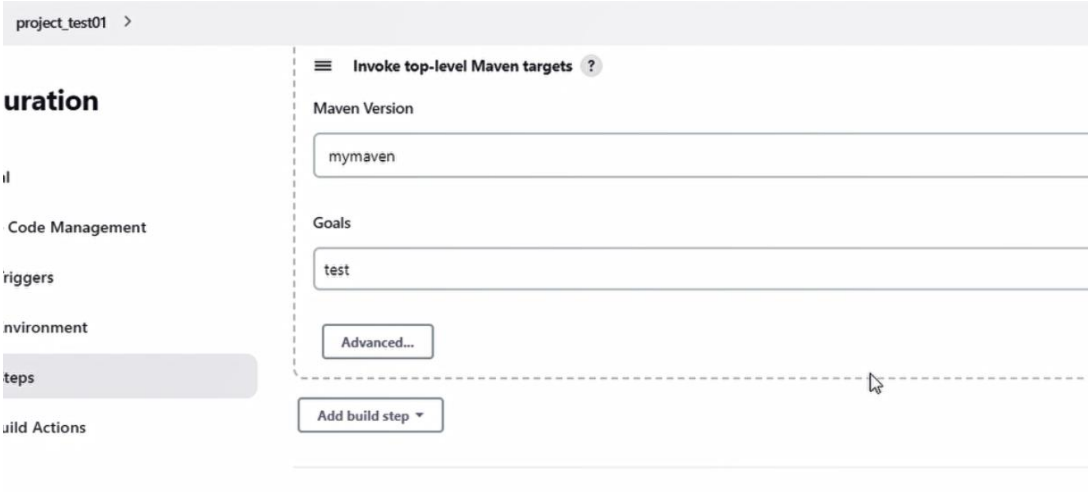
5. Don't run and build any project right now .

6. Now go to your test project , again go to the source code management and paste the following Git repo link.
https://github.com/jleetutorial/maven-project

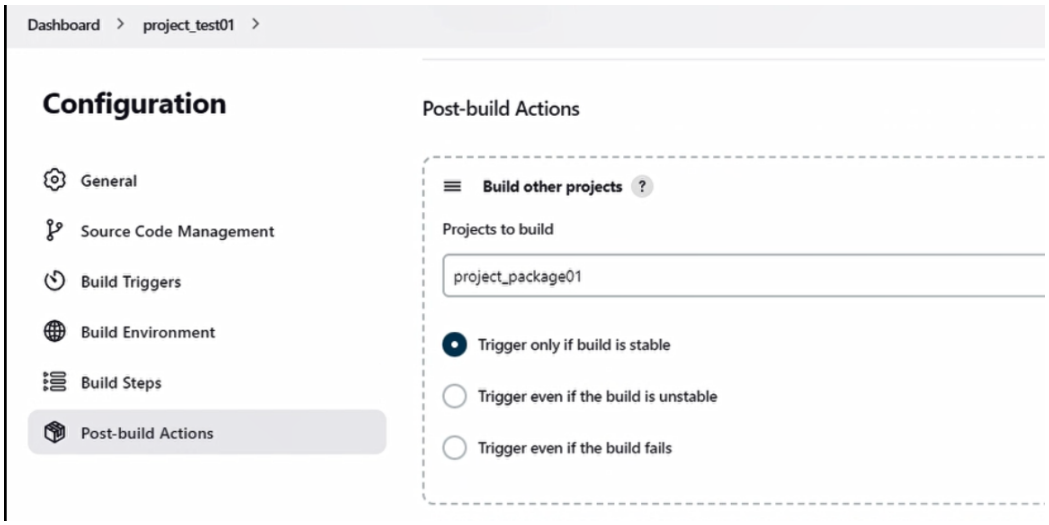
7. Go to build triggers section and choose build after project are build and add the name of the project compile in it.



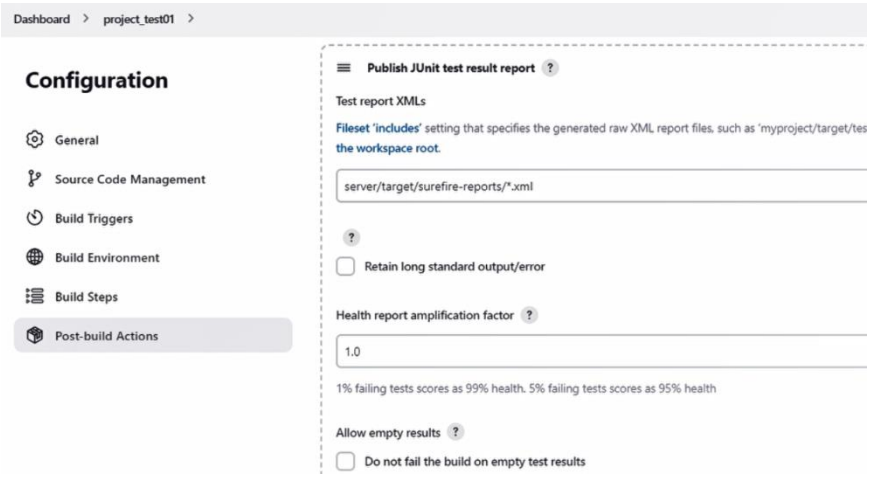
8. Go to the Add Build Step action and select invoke top-level maven targets and select the mymaven version and in goals write test



9. Go to post-build section and choose your project package name .

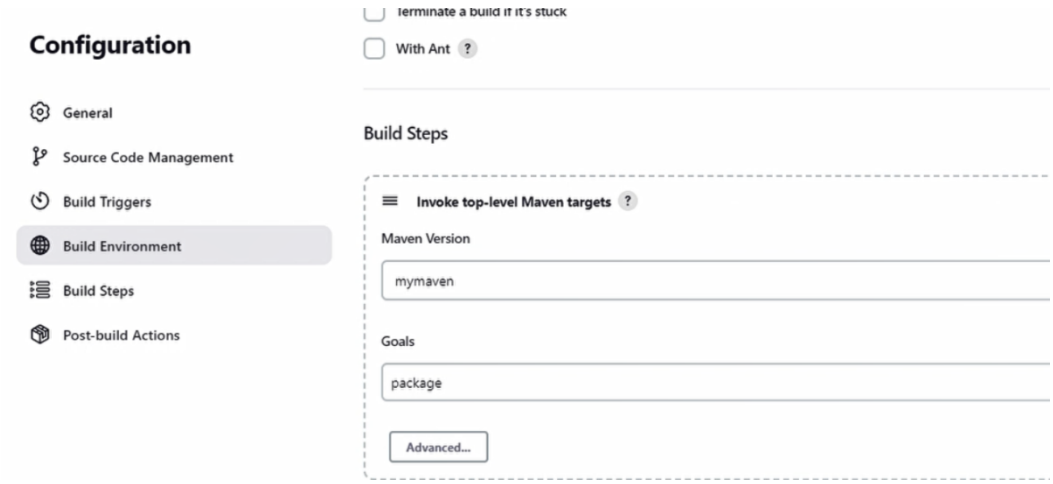


10. Now again add publish Junit test result report in post build actions paste the xml file link from the same git repository . keep the rest same and save



11. Now go to your project package , Now go to your test project , again go to the source code management and paste the following Git repo link. https://github.com/jleetutorial/maven-project

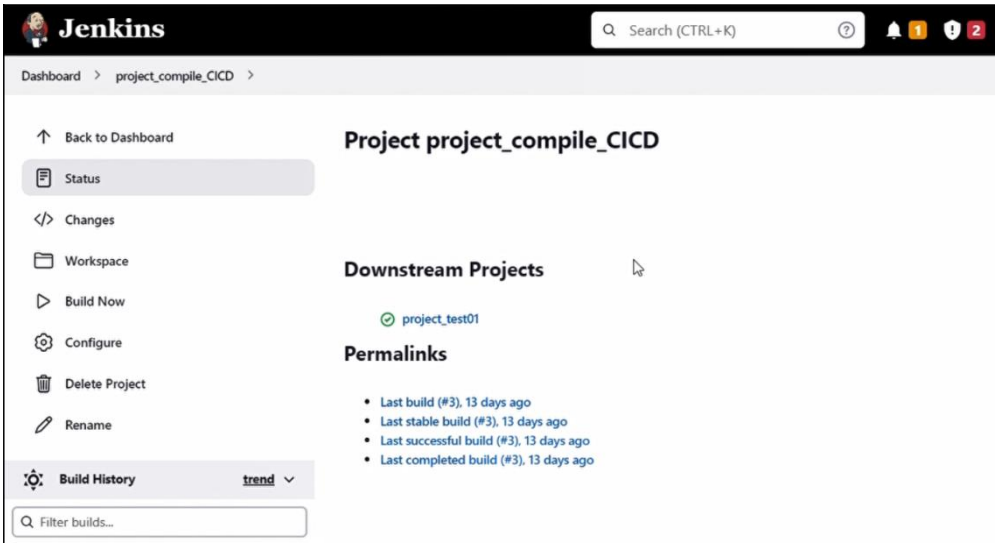
12. Go to build steps and choose top-level maven targets again choose the mymaven version (as per your pc) and in goals write package



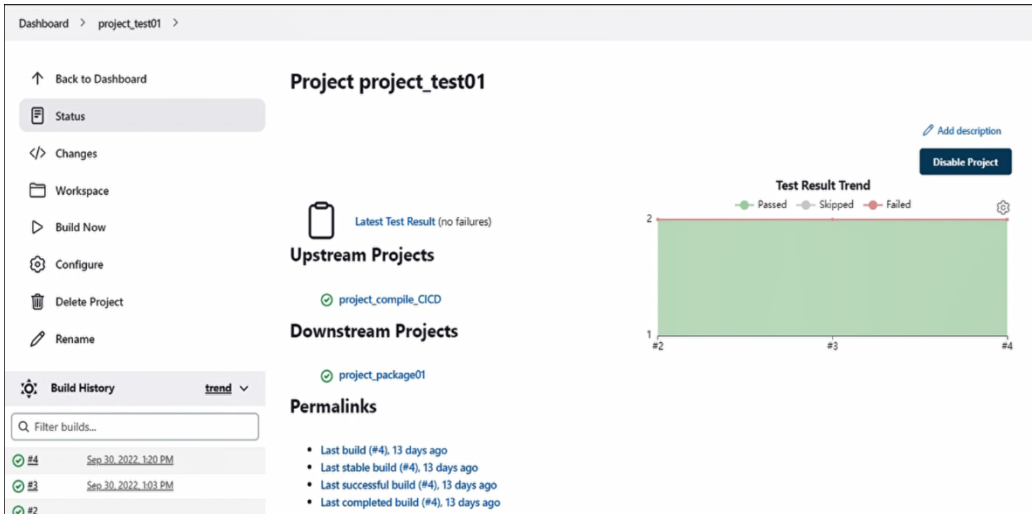
13. Keep the rest setting same and click on apply and save .

14. Now go the dashboard or the project page and build them . after building them all successfully , this should appear in status section .

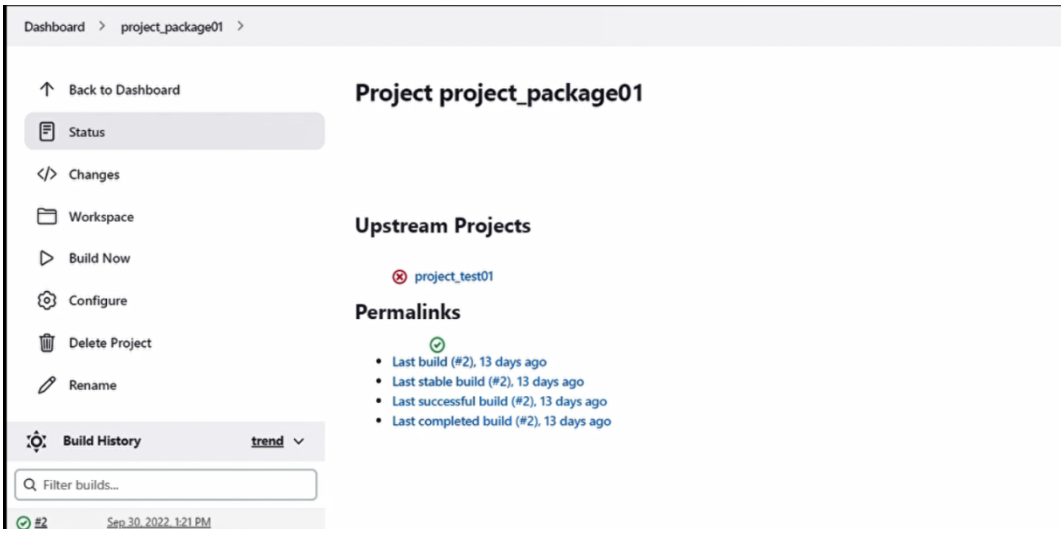
Project Compile -



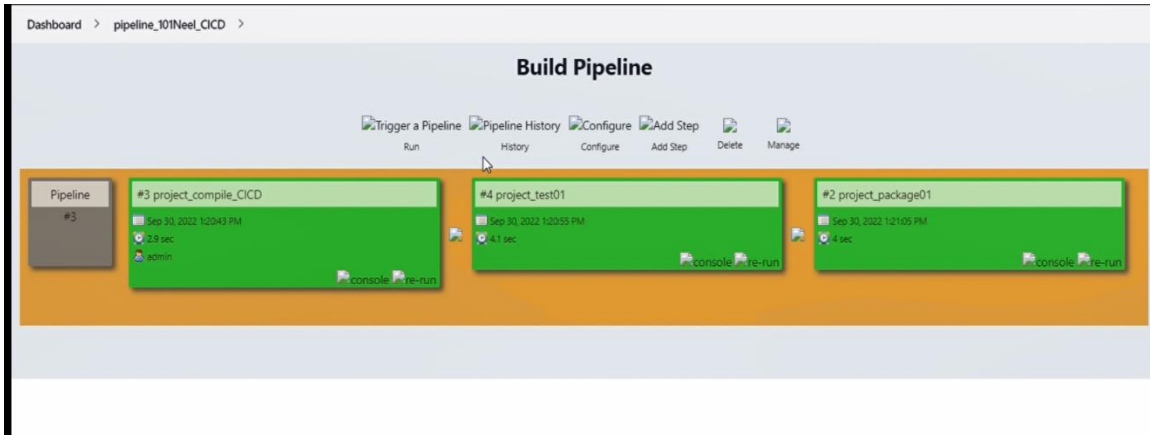
Project test –



Project Package –



15. So after all projects are successfully build go to your pipleline in the top views section and then reload the page until the pipeline is successfully created and try to analyse it and manage .



Conclusion : Therfore we studied pipelining in Jenkins , it's advantages to developers , successfully created a pipeline and imported pipeline from maven repository from git and analysed it.