



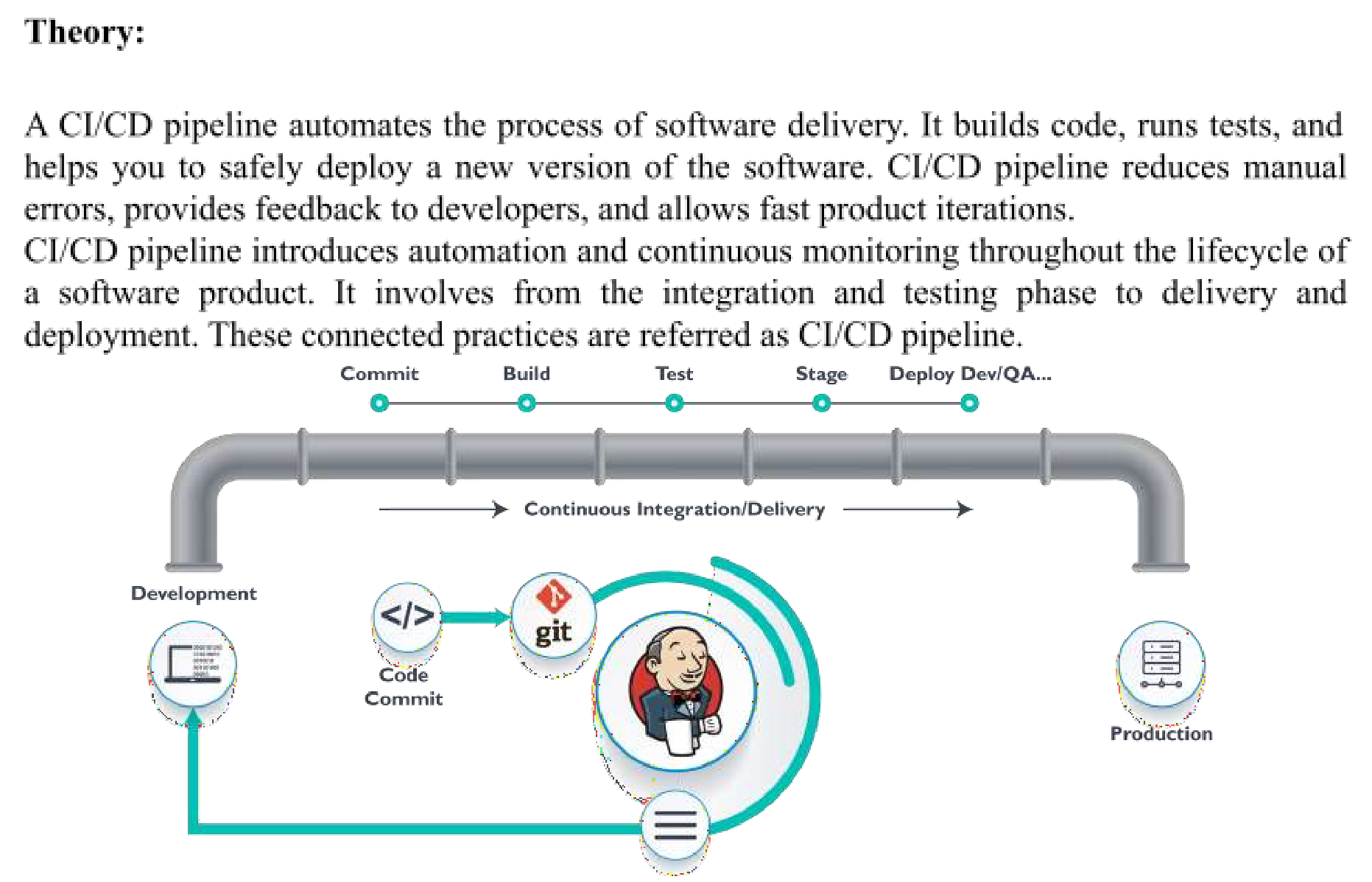
|  |  |  |
| --- | --- | --- |
| **Batch:B1** | **Roll No.:16010421119** | **Experiment No.:** 05 |

**Aim:** To create CI/CD pipeline to deliver a web Application on DockerHub using Git and

GitHub, Maven and Jenkins

**Resources needed:** Maven, GitHub and DockerHub Account

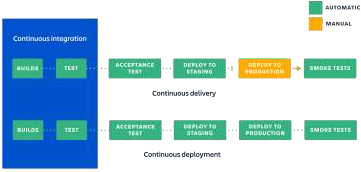




**Continuous integration** is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.

**Continuous delivery** is a software engineering method in which a team develops software products in a short cycle. It ensures that software can be easily released at any time.

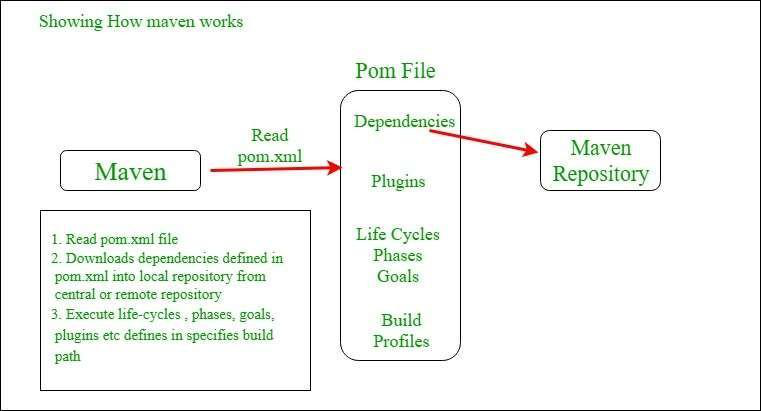
**Continuous deployment** is a software engineering process in which product functionalities are delivered using automatic deployment. It helps testers to validate whether the codebase changes are correct, and it is stable or not.





**What is Maven?**

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. But it is too much advanced than ANT. In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to- day work of Java developers easier and generally help with the comprehension of any Java- based project. **How maven works?**





**Procedure:**   
 1. Clone any Meven project available on github. [Hello world] or write a Maven project.

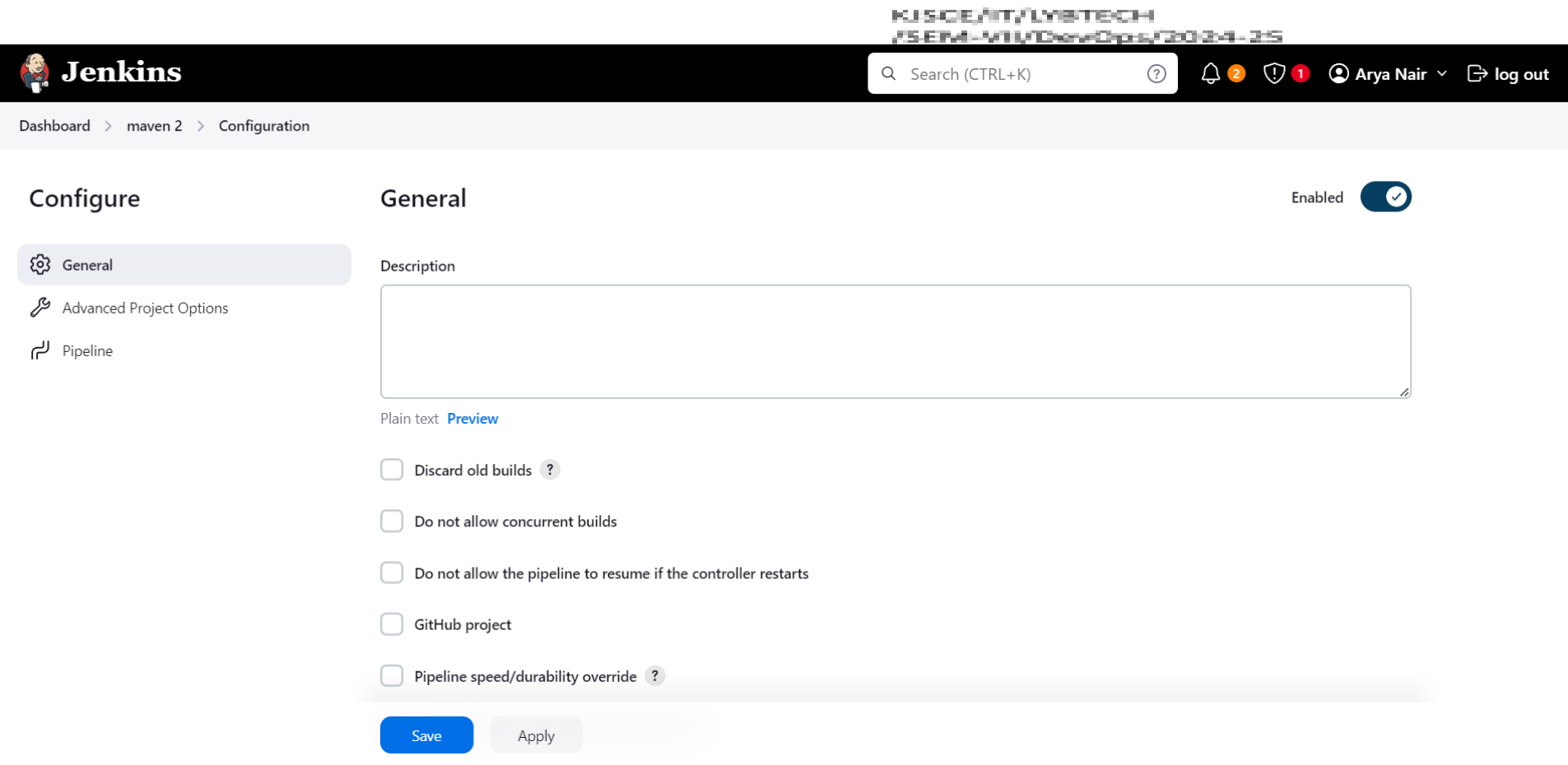
2. Explore pom.xml file.

3. Integrate your project with GitHub   
4. Integrate GitHub with Jenkins   
5. Integrate DockerHub with Jenkins   
6. Modify the project code, push it on GitHub and check the modified image on DockerHub

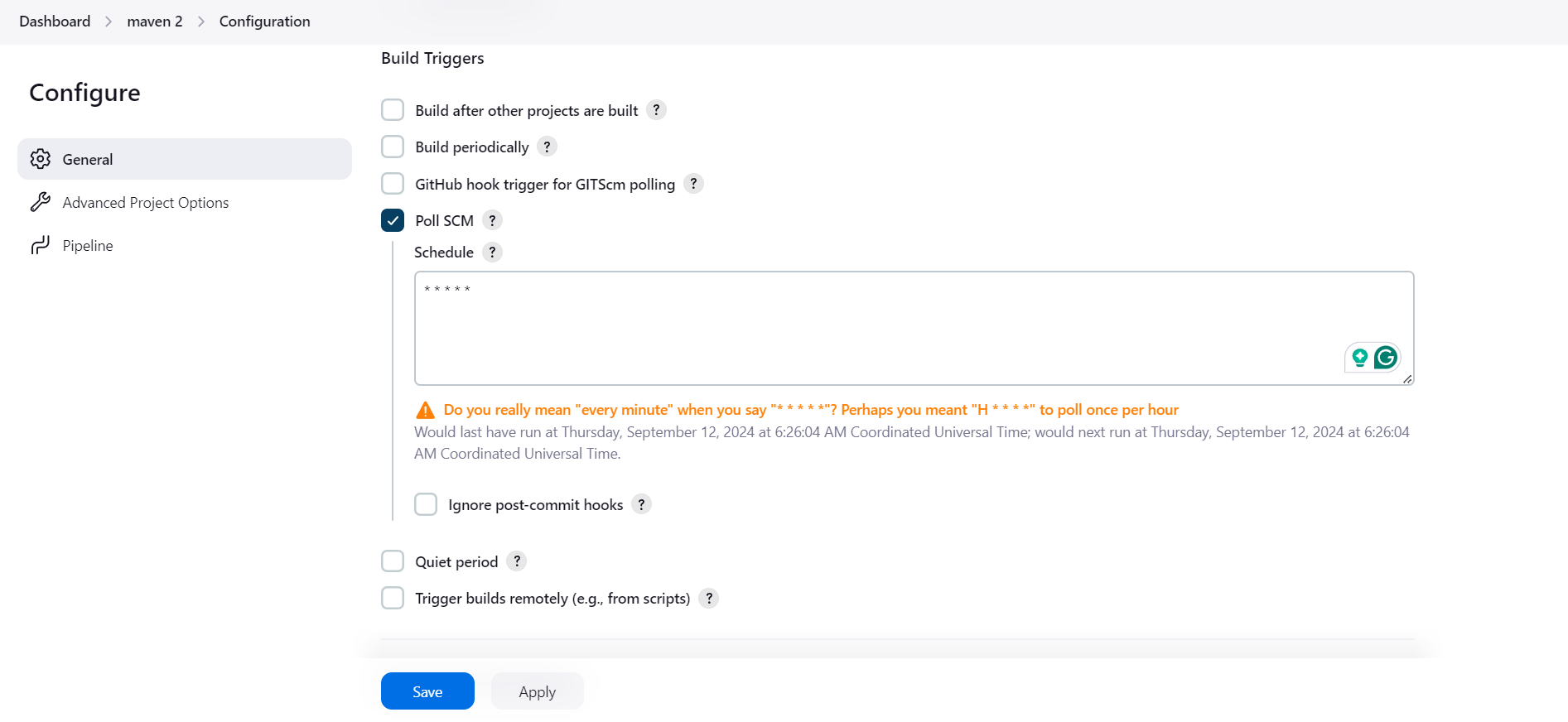
**Results: (Document with screenshots)**

**1.** All steps to create Continuous integration and delivery pipeline of a web Application

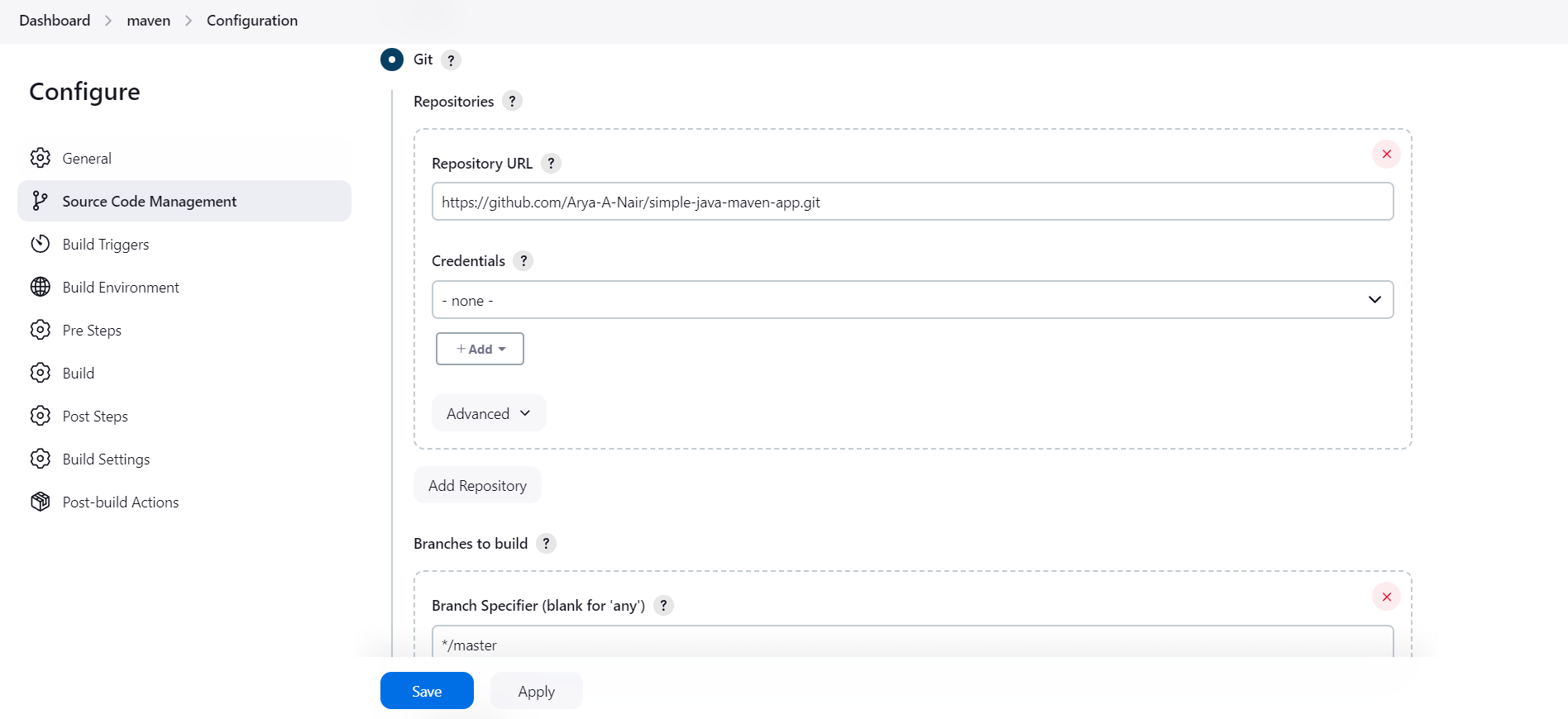
after selecting pipeline option in new item

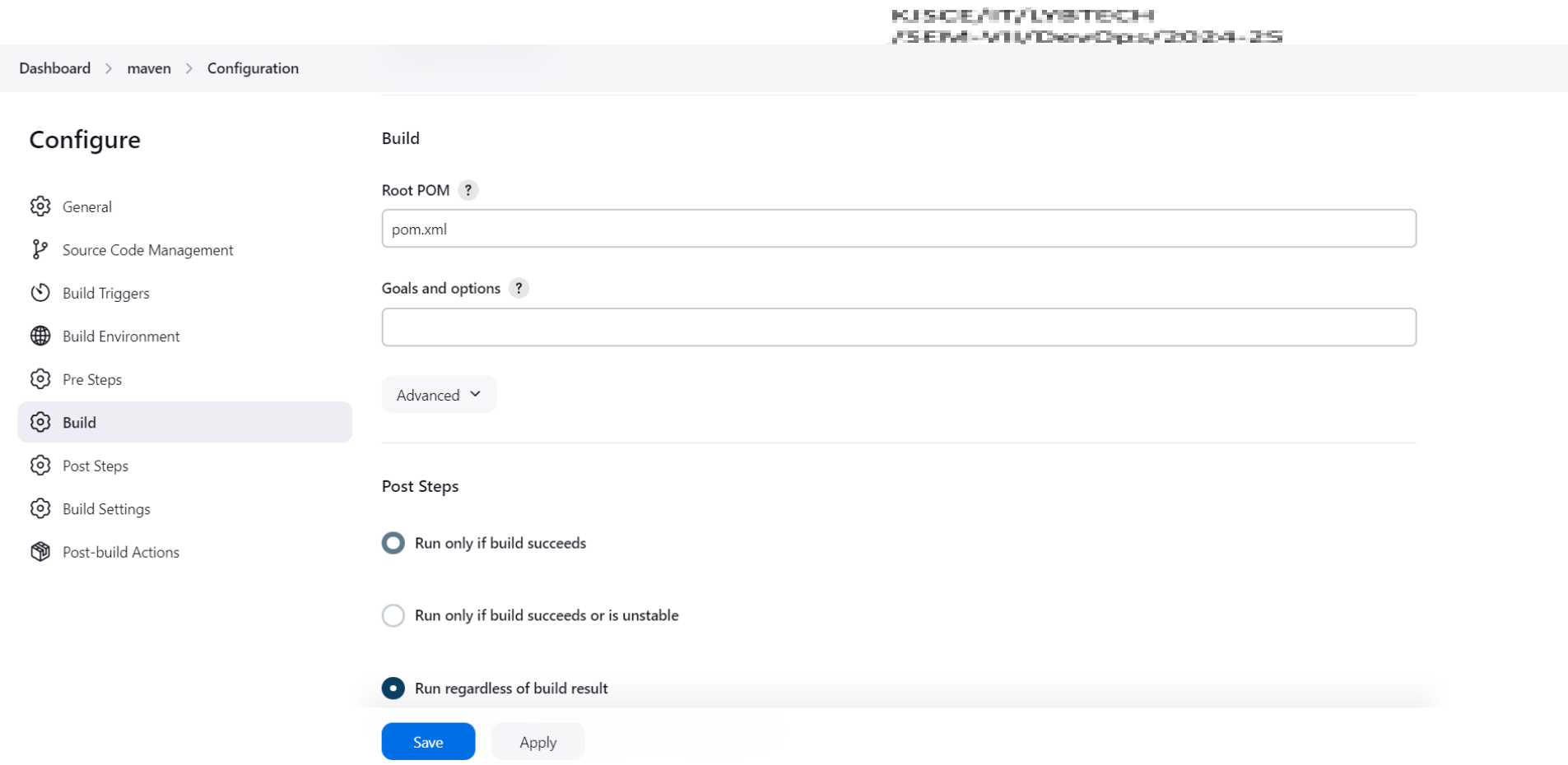


Create a new maven project in jenkins

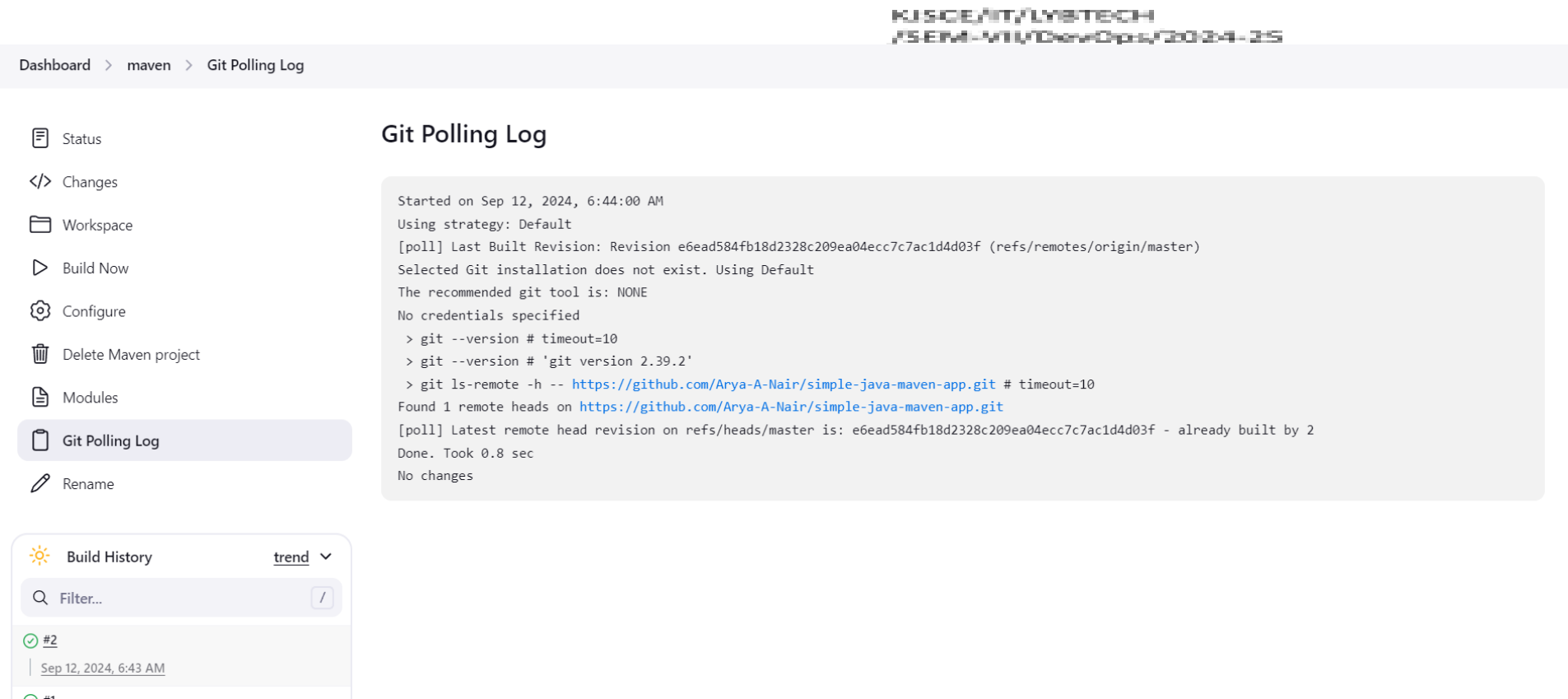


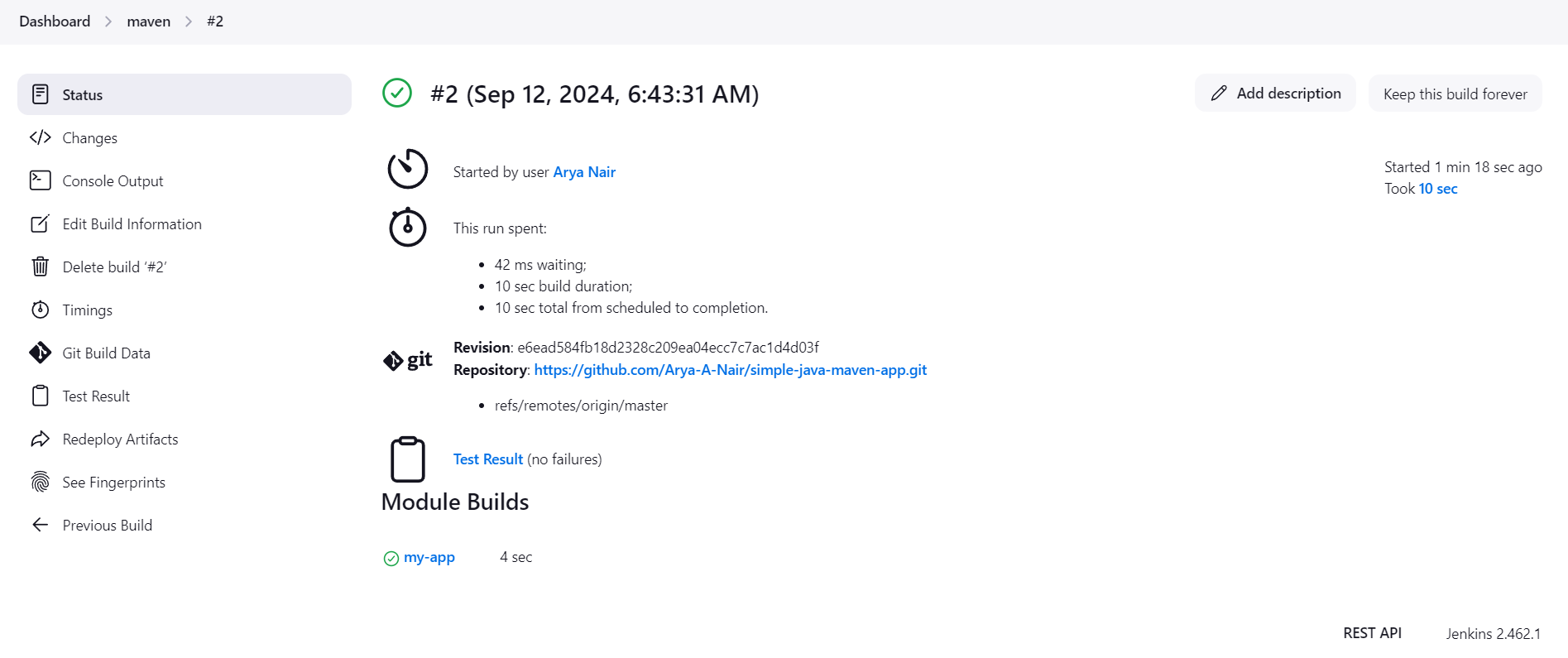
add Poll SCM to check github for updates

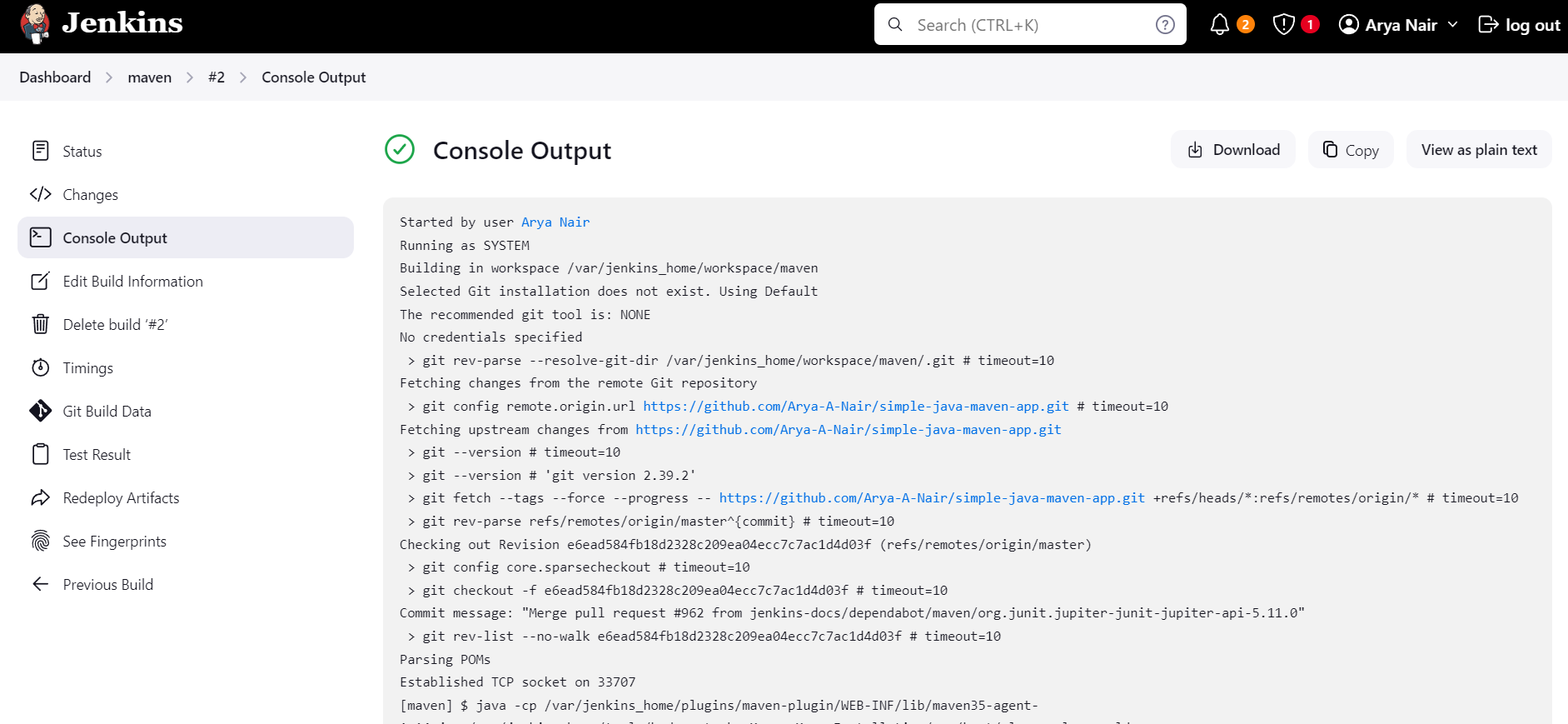




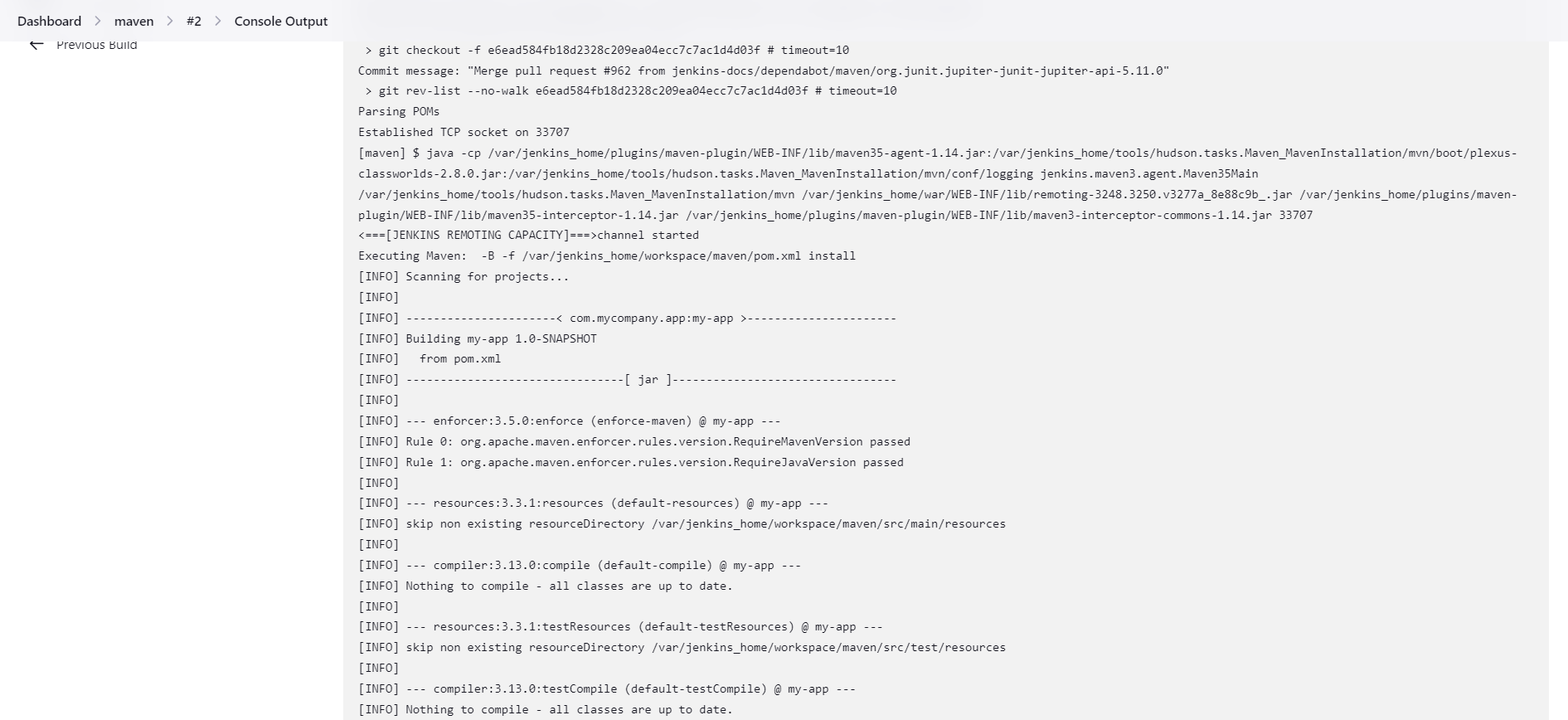
Add the path to pom file











Output of the maven project

**Questions:**

1. What is difference between Continuous integration, Continuous Delivery and Continuous Deployment?

Continuous Integration (CI) involves frequently merging code changes into a shared repository and automatically testing them to catch issues early. Continuous Delivery (CD) extends CI by ensuring that the code is always in a deployable state, allowing for manual releases at any time. Continuous Deployment (CD) goes further by automating the entire release process, pushing every validated change directly to production without manual intervention.

2. Compare build and compile.

Build is the overall process of converting source code into a runnable program, which may include compiling, linking libraries, packaging, and other tasks like running tests. Compile, on the other hand, is a specific step in the build process where source code is translated from a high-level language (e.g., C++, Java) into machine code or intermediate code. Essentially, compilation is one part of the broader build process.

3. Write other two build tools

|  |  |
| --- | --- |
| Two other popular build tools are: |  |

Maven: Primarily used for Java projects, Maven manages dependencies, compiles code, runs tests, and packages the project into a deployable format.

Gradle: A flexible build tool that supports Java, Groovy, and other languages. Gradle automates building, testing, and deployment, offering faster performance through incremental builds. .

4. What is artifact?

An artifact in software development refers to a file or set of files generated during the build



process that is necessary for deployment or further stages of development. Common examples include compiled binaries, libraries (e.g., JAR files), executables, and documentation. Artifacts are typically stored in a repository and used for testing, deployment, or distribution.

**Outcomes**: CO4 Explain code deployment and monitoring systems and their tool support.





**Conclusion: (Conclusion to be based on the Results and outcomes achieved)**

**Run maven project on jenkins**





**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of faculty in-charge with date**



**References:**   
**Books/ Journals/ Websites:**

1. https:/[/www.g](http://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-)e[eksforgeeks.org/introduction-apache-maven-build-automation-tool-](http://www.geeksforgeeks.org/introduction-apache-maven-build-automation-tool-) java-projects/   
2. Maven And Jenkins Integration | How To Integrate Maven With Jenkins | Jenkins Pipeline | Simplilearn <https://www.youtube.com/watch?v=-5tA3hZTVfA>  
3. How To Configure Jenkins With Github | Build Maven Github Project In Jenkins https:/[/www](http://www.youtube.com/watch?v=GlQHS7FdVGM).[youtube.com/watch?v=GlQHS7FdVGM](http://www.youtube.com/watch?v=GlQHS7FdVGM)



7. DevOps Experiment 2 -Setup a Continous Integration Continuous Deployment (CICD) Pipeline.

[https://medium.com/@kalpananand.sharma/devops-experiment-2-setup-a-](https://medium.com/%40kalpananand.sharma/devops-experiment-2-setup-a-continous-integration-continuous-deployment-cicd-pipeline-ff1f48f0f1e8) [continous-integration-continuous-deployment-cicd-pipeline-ff1f48f0f1e8](https://medium.com/%40kalpananand.sharma/devops-experiment-2-setup-a-continous-integration-continuous-deployment-cicd-pipeline-ff1f48f0f1e8) 8. Simple DevOps Project-3 | DevOps project with Git, Jenkins and Docker on





AWS | CICD on containers: <https://www.youtube.com/watch?v=nMLQgXf8tZ0>10. Practical DevOps: Implement DevOps in your organization by effectively building, deploying, testing, and monitoring code,Joakim Verona, Packt Publishing, 2nd Edition,2016