Experiment 05 - Jenkins Setup

| Roll No. | 37 |
| --- | --- |
| Name | Mikil Lalwani |
| Class | D15-A |
| Subject | DevOps Lab |
| LO Mapped | LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements  LO3: To understand the importance of Jenkins to Build and deploy Software Applications on server environment |
|  |  |

**Aim**: To understand Continuous Integration, install and configure Jenkins with Maven/Ant/Gradle to set up a build Job.

**Introduction**:

Continuous integration (CI) is a software development practice in which developers merge their changes to the main branch many times per day. Each merge triggers an automated code build and test sequence, which ideally runs in less than 10 minutes. A successful CI build may lead to further stages of continuous delivery.

If a build fails, the CI system blocks it from progressing to further stages. The team receives a report and repairs the build quickly, typically within minutes.

All competitive technology companies today practice continuous integration. By working in small iterations, the software development process becomes predictable and reliable. Developers can iteratively build new features. Product managers can bring the right products to market, faster. Developers can fix bugs quickly and usually discover them before they even reach users.

**Jenkins**:

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

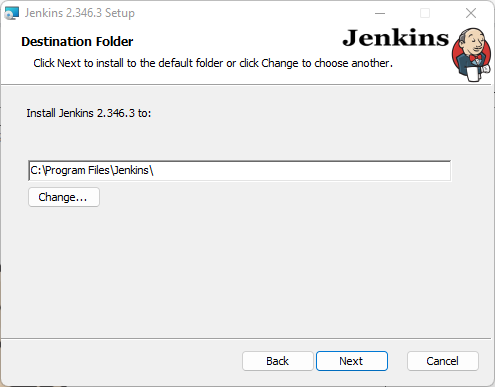
Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example Git, Maven 2 project, Amazon EC2, HTML publisher etc.

**Installation**:

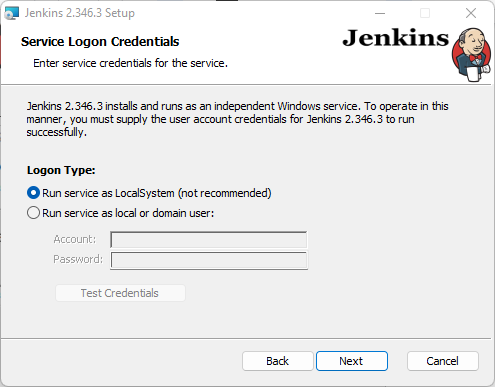
1. Download Jenkins setup from the official website and run it.



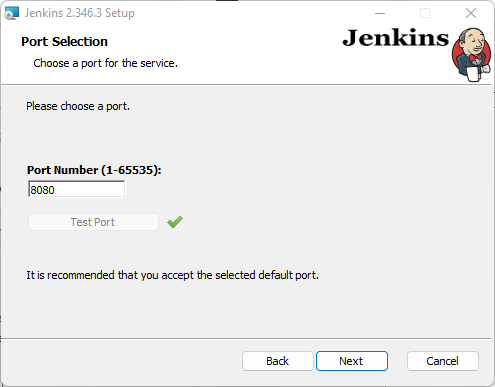
1. Click on next.



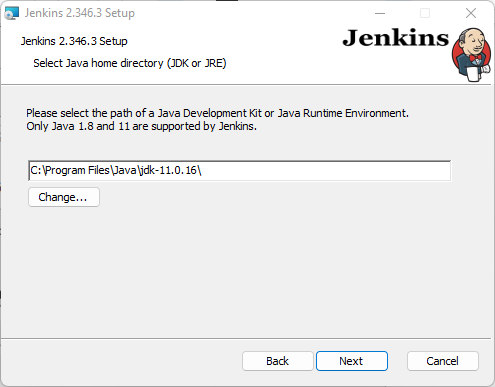
1. Select “Run service as LocalSystem” and click next.



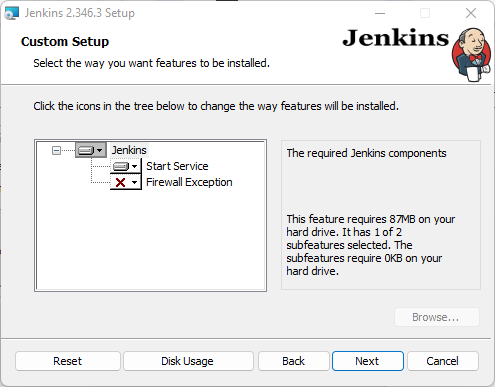
1. Click on “Test Port” and click on next.



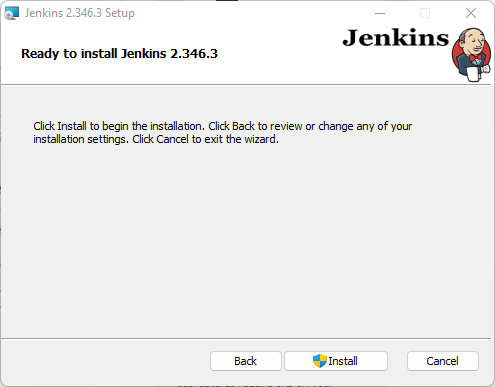
1. Locate the jdk 11 folder and click on next.

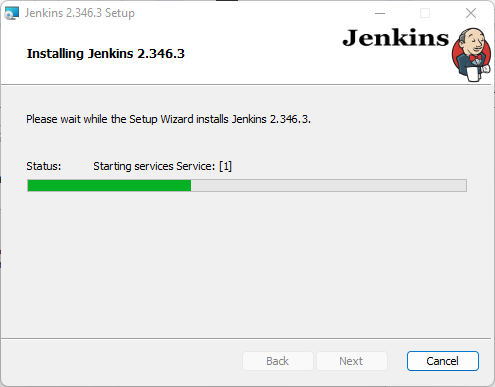


1. Click on next.



1. Click on install.



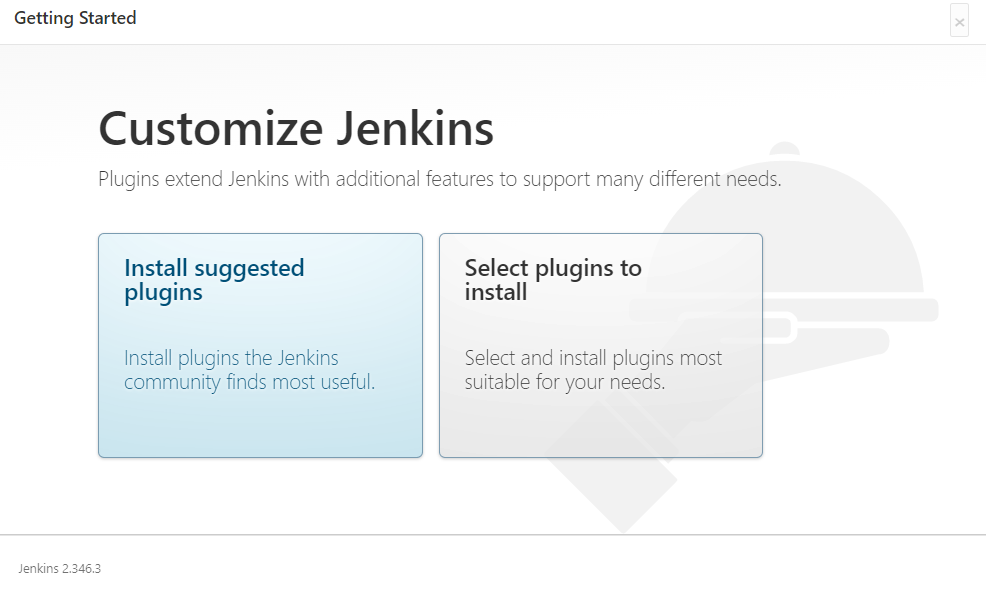


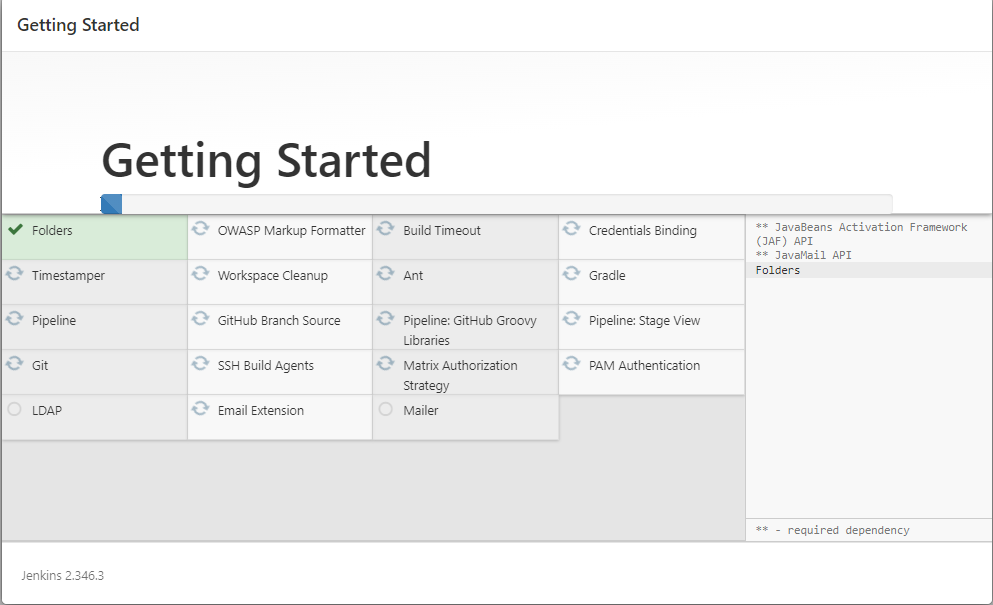


1. Jenkins is now installed on your pc.

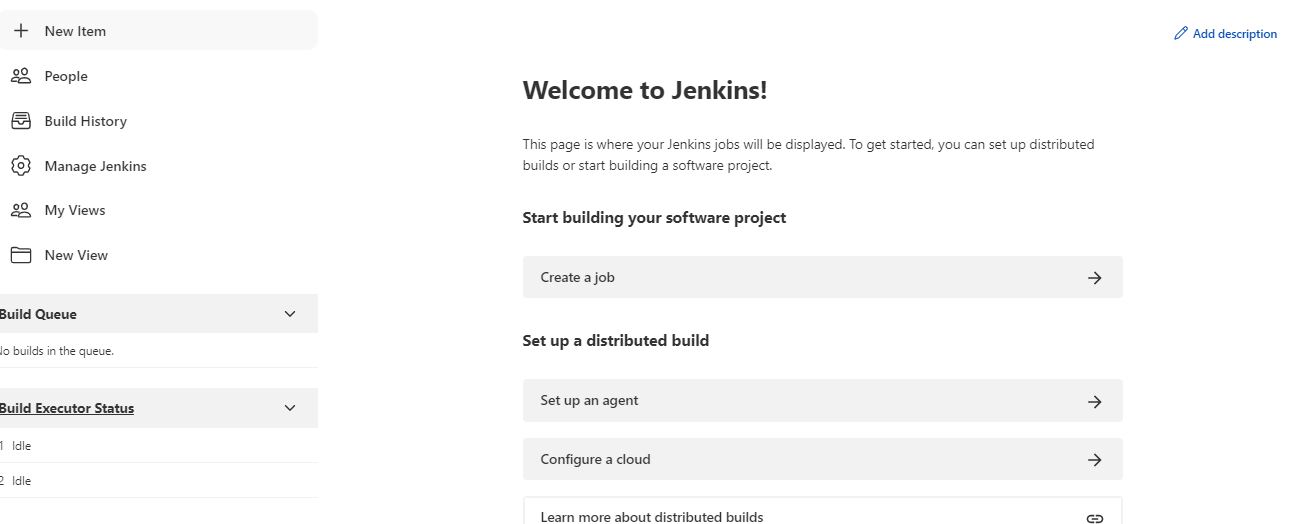
**Build Job**:

1. Start Jenkins by searching localhost:8080 on your search engine’s search bar. Select install suggested plugins.

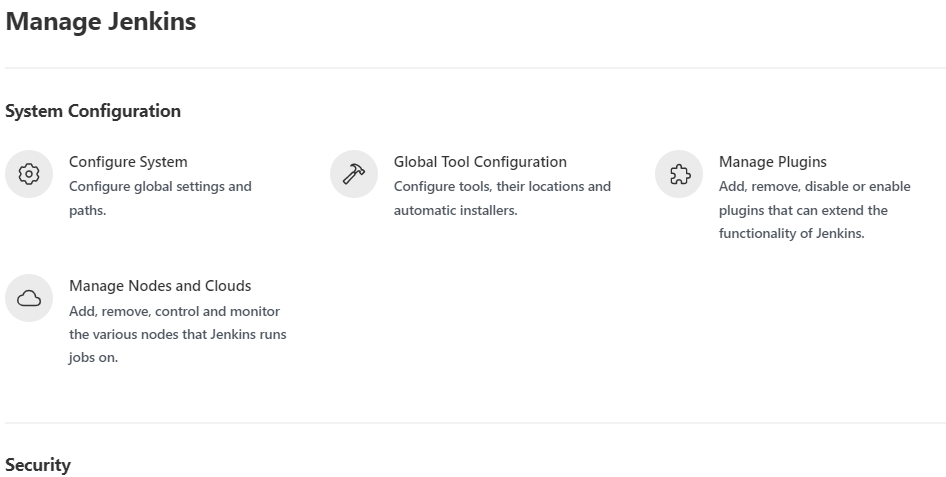




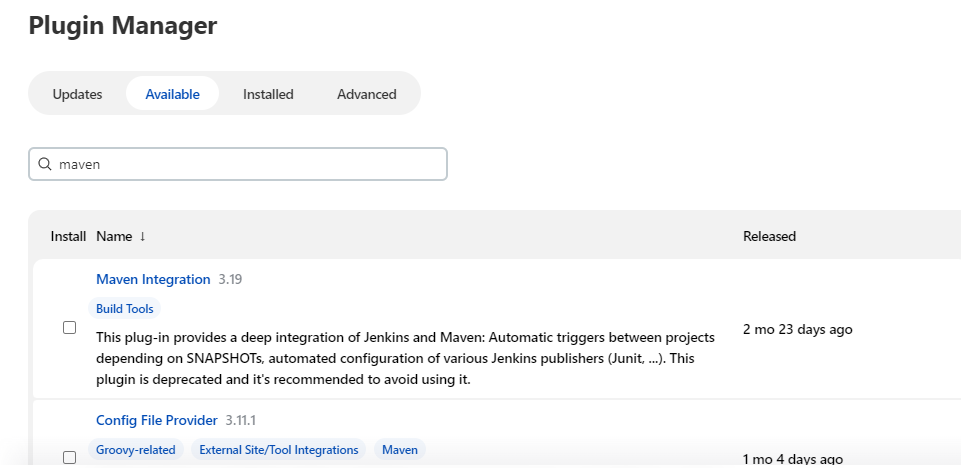
1. After all the plugins are installed we start using Jenkins.



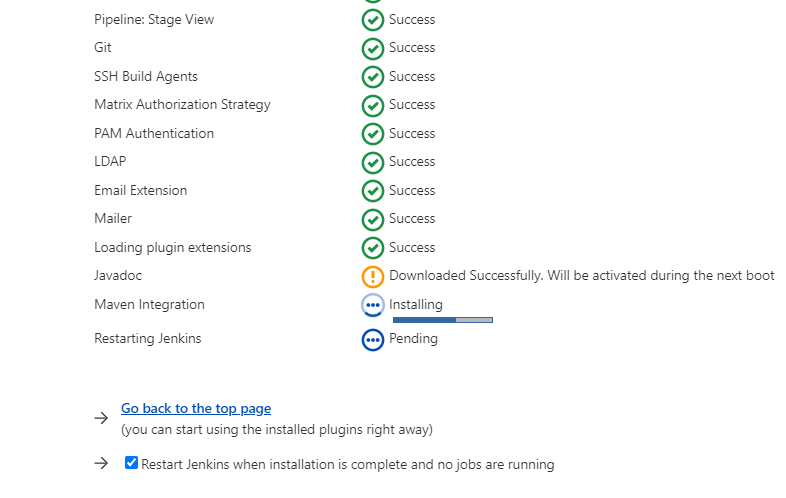
1. Click on Mange Jenkins.



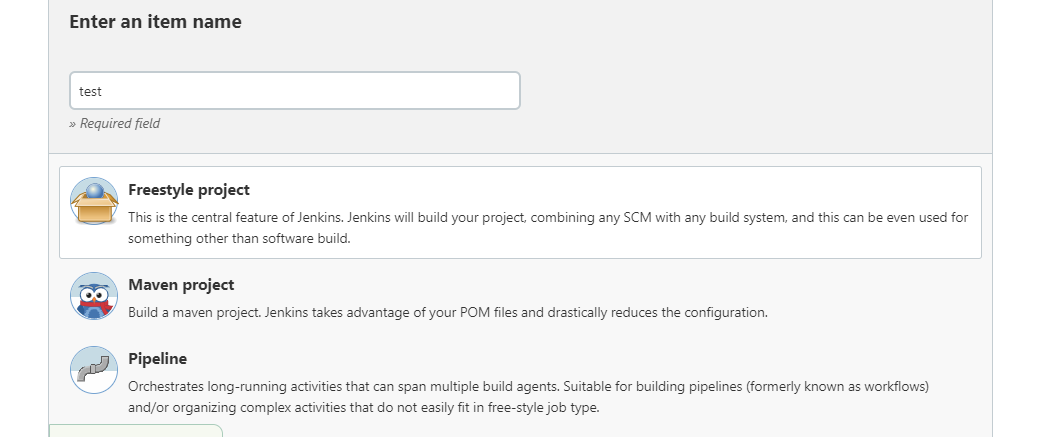
1. Click on Manage Plugins.



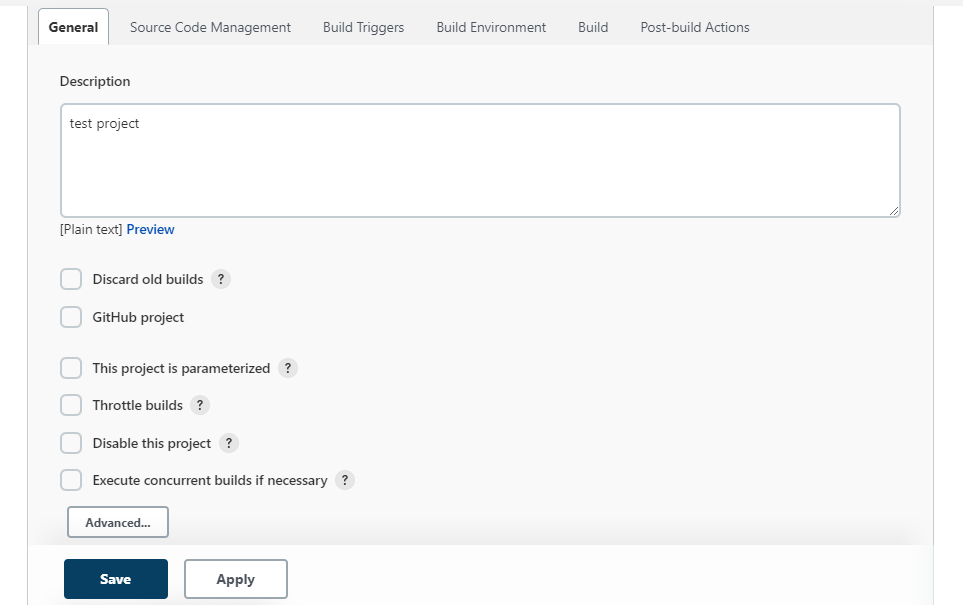
1. Download Maven integration.



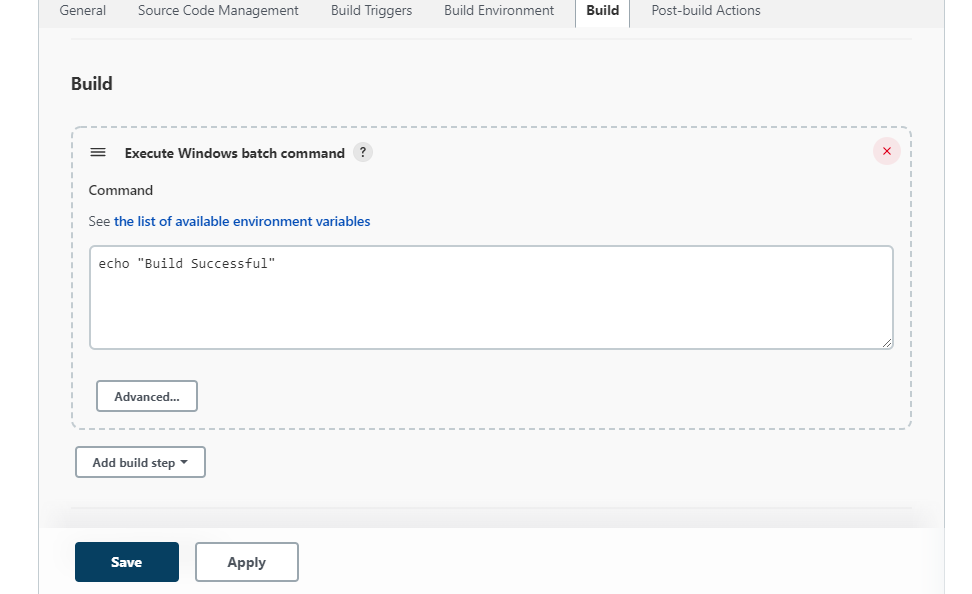
1. Start a new freestyle project.



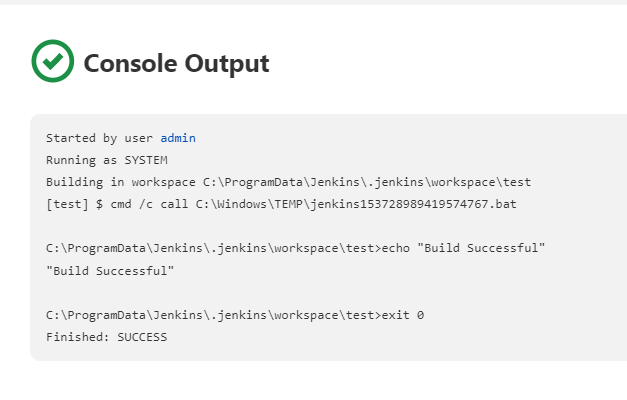
1. Give a name and description as you want.



1. Go to the build stage, select “Execute in Windows Batch” and write echo “Build successful”.



1. Now click on build and check the console output. If you see the Build Successful message then we are successful.



**Conclusion:**

Thus we installed and configured Jenkins with Maven/Ant/Gradle to set up a build Job.