Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.

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

Theory:

Continuous Integration (CI):

Continuous Integration is a software development practice where code changes are automatically built, tested, and integrated into a shared repository on a frequent basis. The primary goal of CI is to detect and address integration issues early in the development process, ensuring that the codebase remains stable and reliable. This approach allows teams to deliver high-quality software more efficiently and with greater confidence.

Key Concepts of Continuous Integration:

- Version Control System (VCS):

CI relies on a VCS (e.g., Git, SVN) to manage and track changes in the codebase. Developers commit their changes to the repository, triggering the CI process.

- Automated Build:

CI systems automate the process of compiling source code into executable artifacts. This ensures consistency and eliminates manual errors in the build process.

- Automated Testing:

CI includes automated testing to validate that code changes do not introduce new bugs or break existing functionality. Common types of tests include unit tests, integration tests, and acceptance tests.

- Build Server:

A CI server, like Jenkins, is responsible for orchestrating the CI process. It monitors the VCS for changes, triggers builds, runs tests, and provides feedback to developers.

Installing and Configuring Jenkins with Maven/Ant/Gradle:

1. Install Jenkins:
   Download and install Jenkins from the official website.
   Start the Jenkins service.
2. Configure Jenkins:
   Open Jenkins in a web browser and follow the initial setup wizard.
   Install necessary plugins, including the ones for Maven, Ant, or Gradle integration.
3. Create a Jenkins Job:
   Click on "New Item" to create a new job.
   Choose the type of project (e.g., Freestyle project or Pipeline).
   Configure the job settings, such as source code repository, build triggers, and post-build actions.
4. Configure Build Tool:
   If using Maven, specify the path to the Maven executable and configure Maven goals (e.g., clean install).
   For Ant or Gradle, configure the respective build tool settings.

5. Save and Build:
   Save the job configuration and manually trigger the build to test the setup.
   Observe the build console output for any errors or issues.

Continuous Integration Benefits:

- ❖ Early Detection of Bugs:

CI ensures that code changes are continuously tested, allowing for the early detection and resolution of bugs.

- ❖ Consistent Builds:

Automated builds eliminate variations caused by manual processes, ensuring that each build is consistent and reproducible.

- ❖ Integration Testing:

CI helps in integrating code changes frequently, reducing the likelihood of integration issues that arise when merging changes from multiple developers.

- ❖ Rapid Feedback:

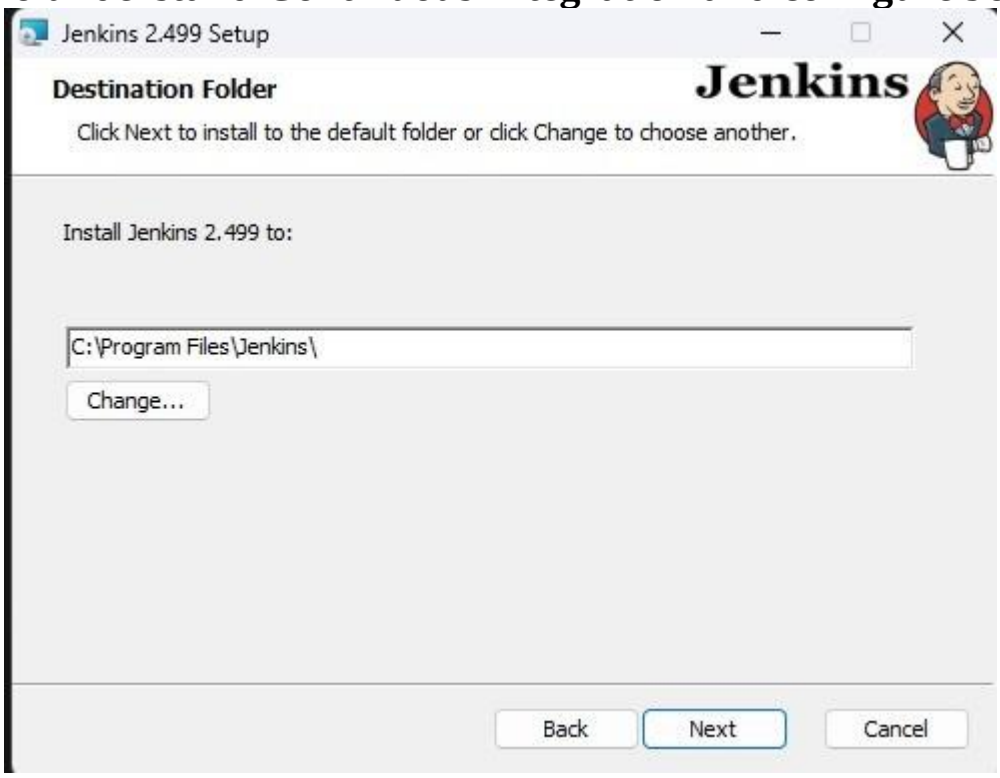
Developers receive rapid feedback on the quality of their code through automated tests and build reports, enabling them to address issues promptly.

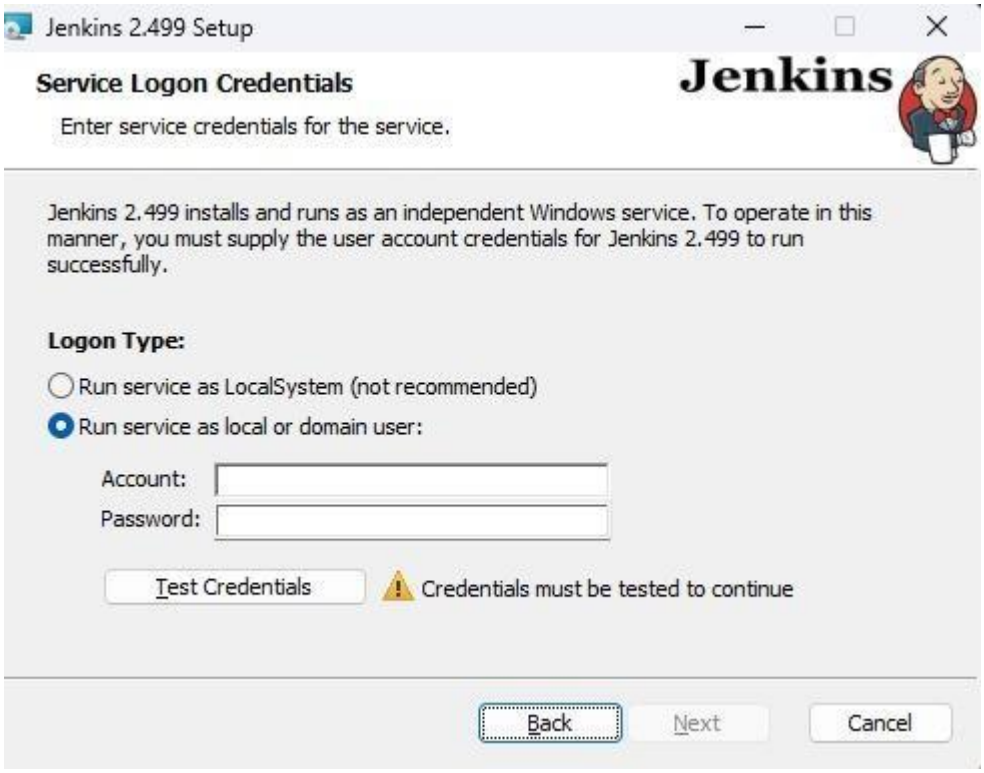
- ❖ Improved Collaboration:

CI promotes collaboration by providing a central platform where developers can share and integrate their work regularly.

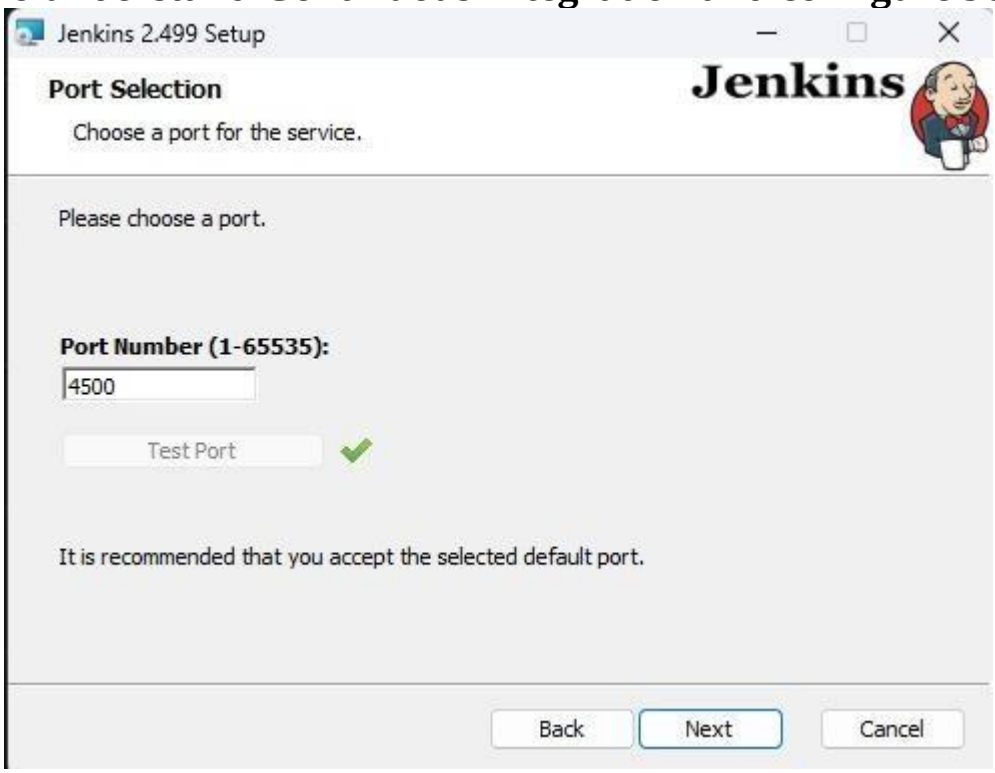
Implementation:

Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.

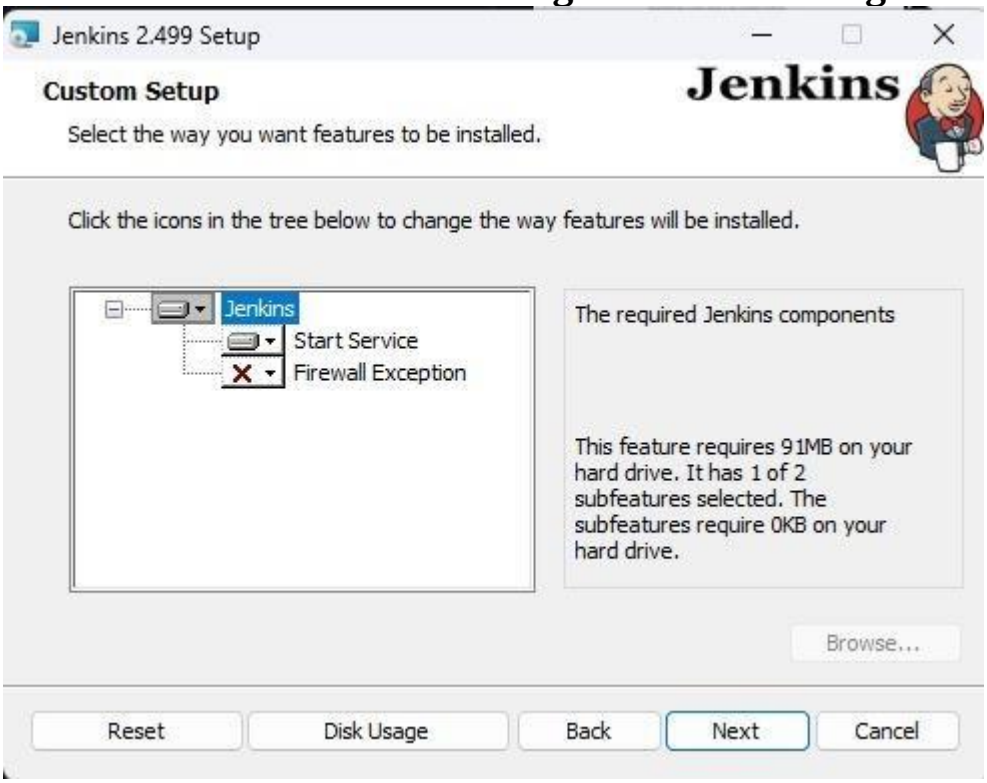


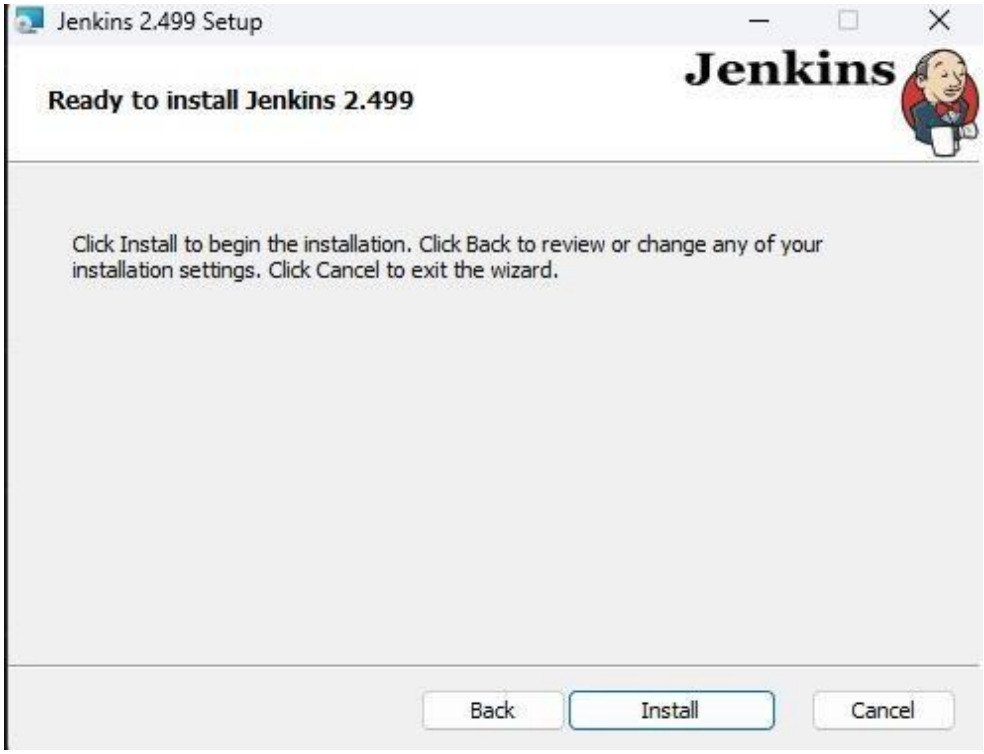


Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.

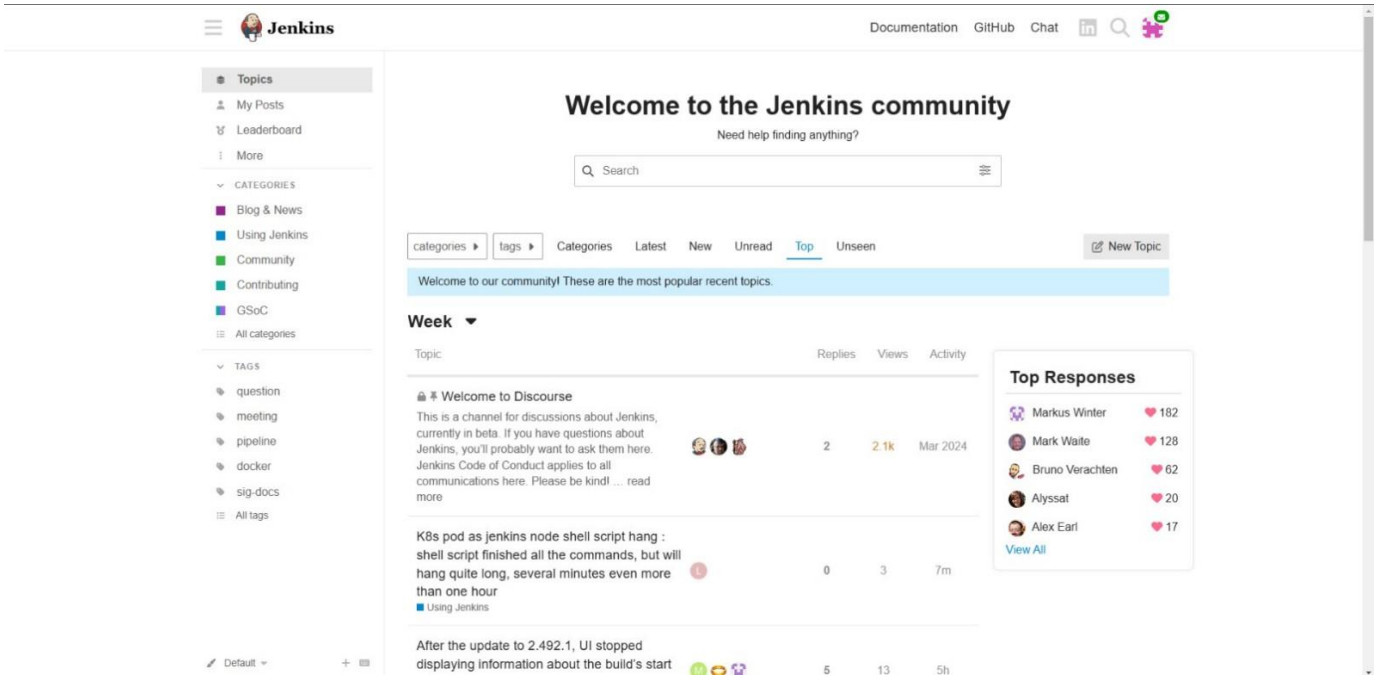


Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.

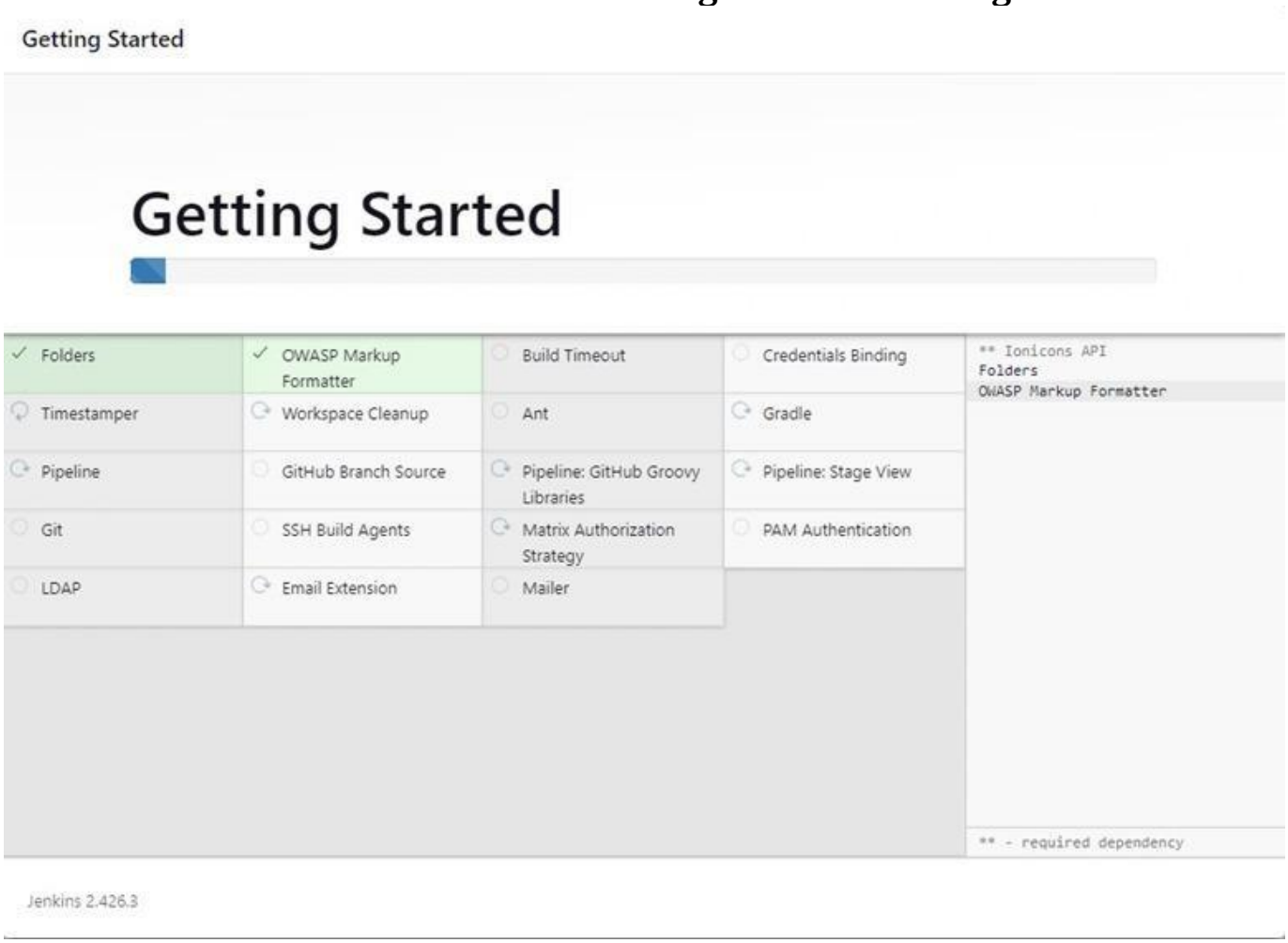


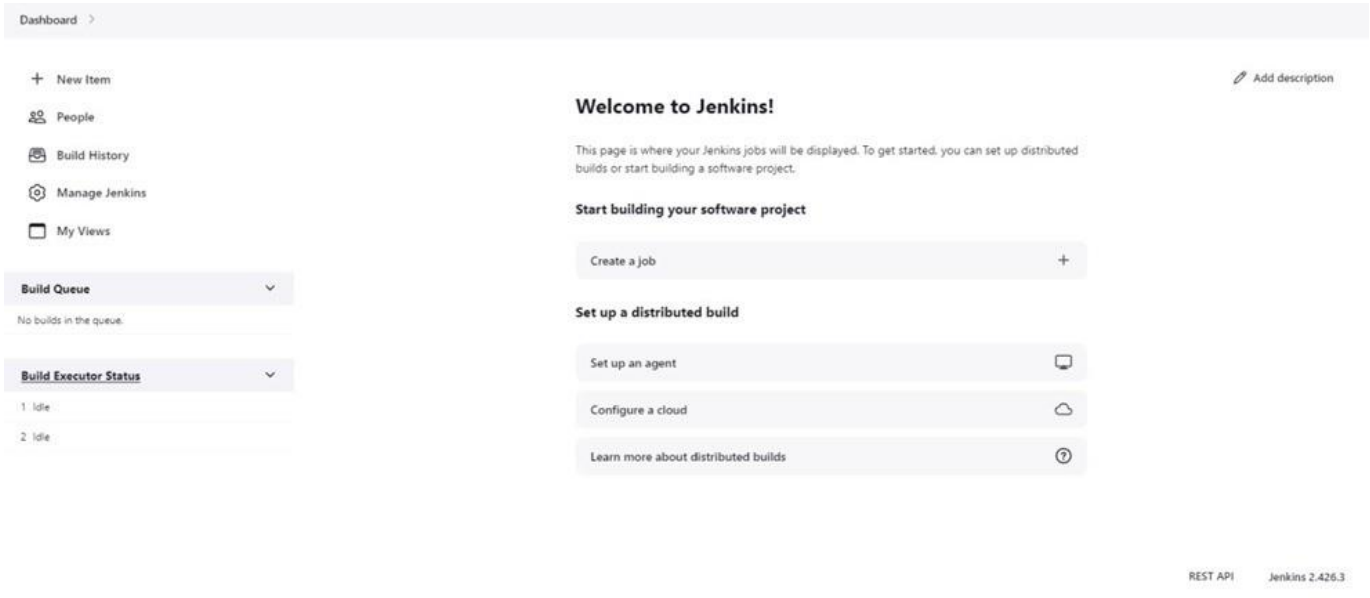


Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.



Software Engineering & Project Management Lab Experiment No: - 04
Aim: To understand Continuous Integration and configure Jenkins.





Conclusion: Thus, we have successfully understood Continuous Integration, installation, and configuration of Jenkins with Maven/Ant/Gradle to set up a build job.

LO Mapping: *LO3 is mapped.*

TSEC Batch:-T12 Name & Roll No:- Sarthak Hinge - 35