

## SEPM Exp 4

Rahul Kumhar

T13 58

Aim:-

To understand Continuous integration, install and configure Jenkins with Maven/Ant/Gradle to setup a build Job.

Theory:-

Continuous Integration (CI) :

Continuous Integration is a key practice in modern software development methodologies like Agile and DevOps. Its core principle is to integrate code changes into a shared repository frequently, ideally multiple times a day. With each integration, an automated build and test process is triggered to verify the changes. CI aims to detect integration errors as early as possible in the development cycle, enabling teams to address them swiftly.

**Extensibility:** Jenkins offers a vast ecosystem of plugins that extend its functionality to support various build tools, version control systems, and deployment targets.

**Pipeline as Code:** Jenkins allows defining build pipelines as code using Jenkinsfile, enabling teams to version control and automate the entire CI/CD process.

**Distributed Builds:** Jenkins supports distributed builds, allowing teams to distribute build tasks across multiple machines or agents for faster build execution.

**Rich Integration:** Jenkins integrates seamlessly with popular version control systems like Git, SVN, and Mercurial, as well as with build tools like Maven, Ant, Gradle, and more.

**Robust Notifications:** Jenkins provides extensive notification features, including email alerts, chat notifications (Slack, Microsoft Teams), and integration with issue tracking systems, to keep team members informed about build statuses and failures.

## Continuous Integration with Jenkins:


Setting up CI with Jenkins typically involves configuring Jenkins to monitor version control repositories for code changes, triggering automated builds on each commit, and executing build scripts or tasks to build, test, and package the software. Jenkins provides a user-friendly web interface for configuring jobs, defining build steps, and monitoring build results, making it accessible to both developers and DevOps engineers.

By leveraging Jenkins for Continuous Integration, teams can streamline their development workflows, accelerate delivery cycles, and deliver high-quality software with confidence.



Jenkins 2.440.1 Setup

Jenkins



Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.440.1 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.440.1 to run successfully.

**Logon Type:**

☒ Run service as LocalSystem (not recommended)

☐ Run service as local or domain user:

Account:

Password:

Test Credentials


Back

Next

Cancel

Jenkins 2.440.1 Setup

Jenkins



Port Selection


Choose a port for the service.

Please choose a port.

**Port Number (1-65535):**

8086

Test Port

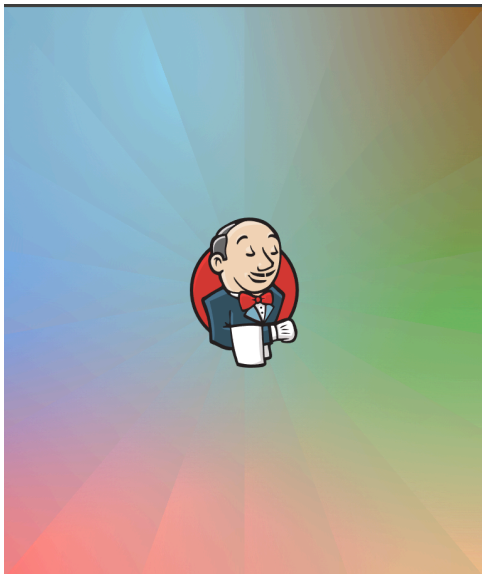


It is recommended that you accept the selected default port.

Back

Next

Cancel



### Sign in to Jenkins

Username

Password

☐ Keep me signed in

Sign in

### Conclusion:-

Continuous Integration with Jenkins empowers teams to embrace Agile practices, foster collaboration, and deliver high-quality software with speed and confidence. By adopting CI principles and leveraging automation tools like Jenkins, organizations can respond more effectively to changing requirements, mitigate risks, and ultimately, deliver value to their customers more efficiently.