Experiment no. 4

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

Theory:

1. **Introduction to Continuous Integration (CI)**
Continuous Integration is a software development practice that encourages developers to integrate code into a shared repository frequently, ideally several times a day. Each integration is verified by an automated build, allowing teams to detect problems early.

- Why CI is needed:
Traditional development models often suffered from "integration hell" where developers would work in silos, only to face countless conflicts and bugs when merging code. CI minimizes these issues by ensuring frequent testing and integration.
CI Lifecycle Overview:

- Code is pushed to the version control system (Git, SVN, etc.)
- The CI tool pulls the latest code and builds it
- Automated tests are run
- Code is analyzed, packaged, and optionally deployed to a staging environment

2. Benefits of Continuous Integration

- CI isn't just about automation—it brings tangible, measurable advantages to the development process.
- Early bug detection and faster feedback loops
- Reduced integration issues and smoother merges
- Consistent build process and reproducible outcomes
- Higher code quality through automated testing
- Improved team collaboration and developer confidence

3. Introduction to Jenkins
Jenkins is a powerful open-source automation server widely used for implementing CI/CD pipelines. Originally developed as Hudson by Kohsuke Kawaguchi, it was later renamed Jenkins and gained community traction.

- Key Features of Jenkins:

- Supports hundreds of plugins
- Integrates with almost all major tools (Git, Docker, Maven, Gradle, Ant, etc.)
- Provides both freestyle and pipeline job configurations
- User-friendly web interface and CLI support
- Scalable with master-slave architecture
- Common Use Cases:
- Automated builds and tests
- Continuous deployment to test/production environments
- Monitoring of cron jobs and infrastructure automation

4. Installing Jenkins
Before configuring Jenkins, it must be installed either on a local server, cloud VM, or containerized via Docker.

System Requirements:
- Java 8 or 11 installed (Jenkins is Java-based)
- Minimum 1 GB RAM (more for plugins)
- Network access for downloading plugins and integrating with version control

Installation Methods:
- Using .war file: java -jar jenkins.war
- Using package manager: apt, yum, or brew
- Via Docker container for easy setup and portability

5. Configuring Jenkins After Installation
Once Jenkins is up and running, initial configuration is required before creating build jobs.

Initial Setup:
- Unlocking Jenkins using admin password
- Installing recommended plugins (Git, Maven, Gradle, etc.)
- Creating the first admin user

Setting up Global Tools:
- Navigate to Manage Jenkins → Global Tool Configuration

Configure:

- JDK (if not already present)
- Maven, Ant, and Gradle installation paths
- Git executable path

6. Understanding Build Tools: Maven, Ant, and Gradle
Jenkins doesn't compile code by itself; it integrates with build tools. Let's understand these tools briefly

- Apache Maven:

Based on Project Object Model (POM). It uses XML for configuration and follows a strict convention-over-configuration approach. It's best for large Java projects.

- Apache Ant:

Older than Maven, uses XML build files but gives more control and flexibility. It doesn't follow conventions like Maven.

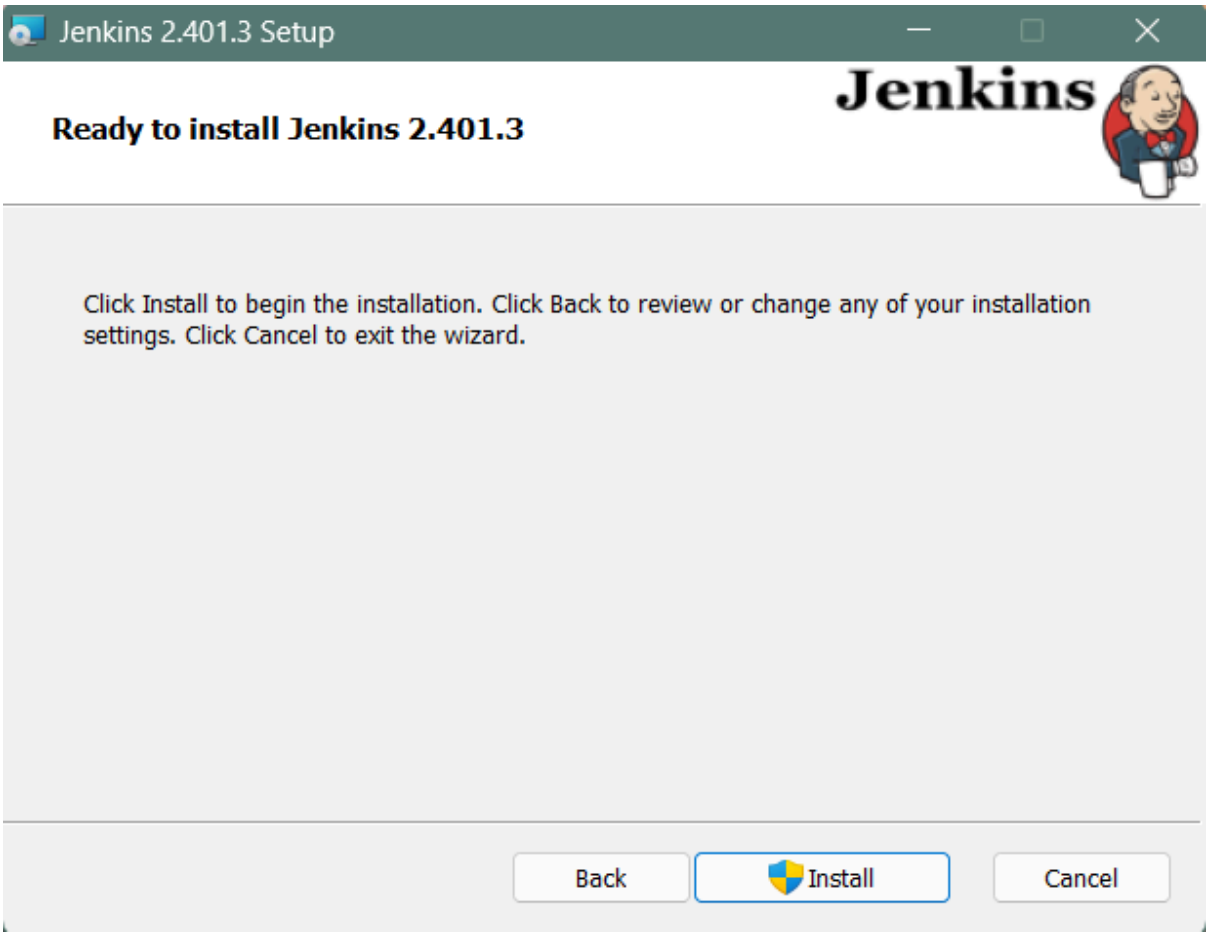
- Gradle:

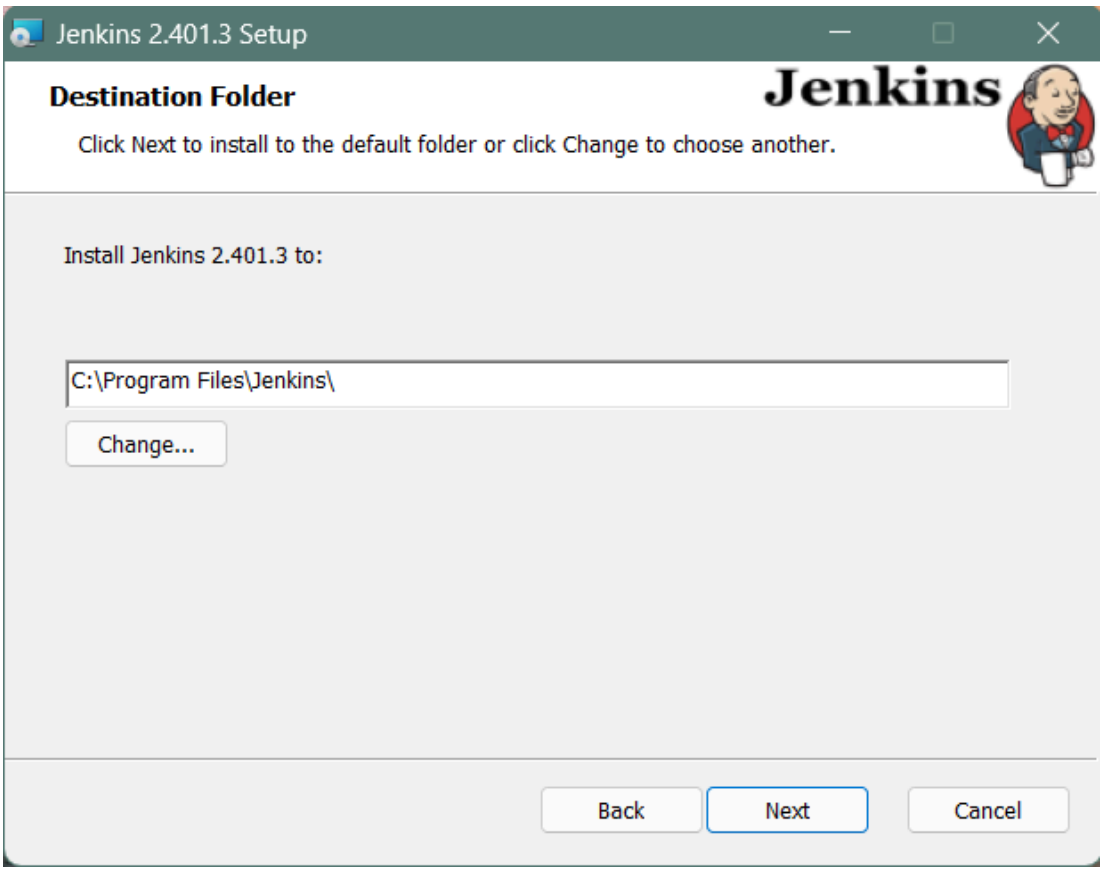
Combines the best of both Ant and Maven. Uses Groovy or Kotlin DSL instead of XML, supports incremental builds, and is highly customizable.

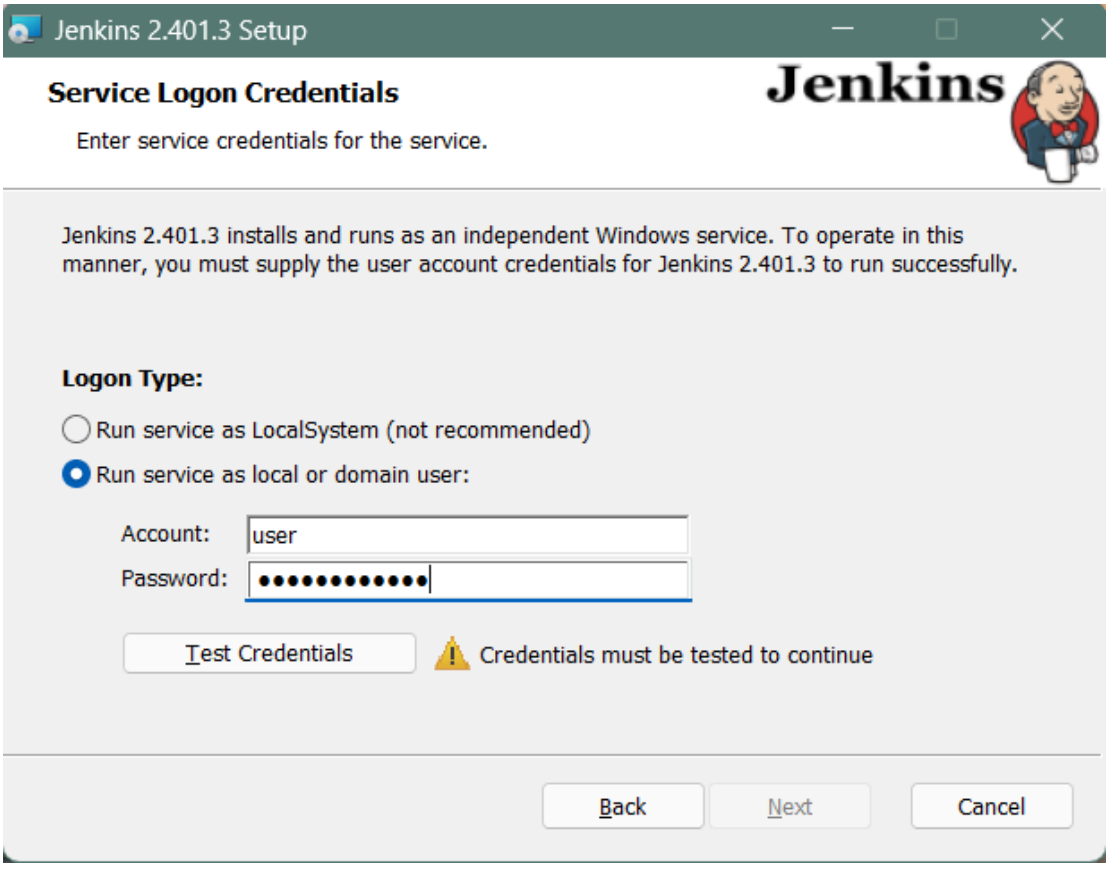
Implementation:
To install Jenkins following software packages are required
1) GIT (git-scm.com)
2) Notepad++ (https://notepad-plus-plus.org/downloads/)
3) Latest Java development kit (JDK)
4) Jenkins
5) Apache Maven (Optional)

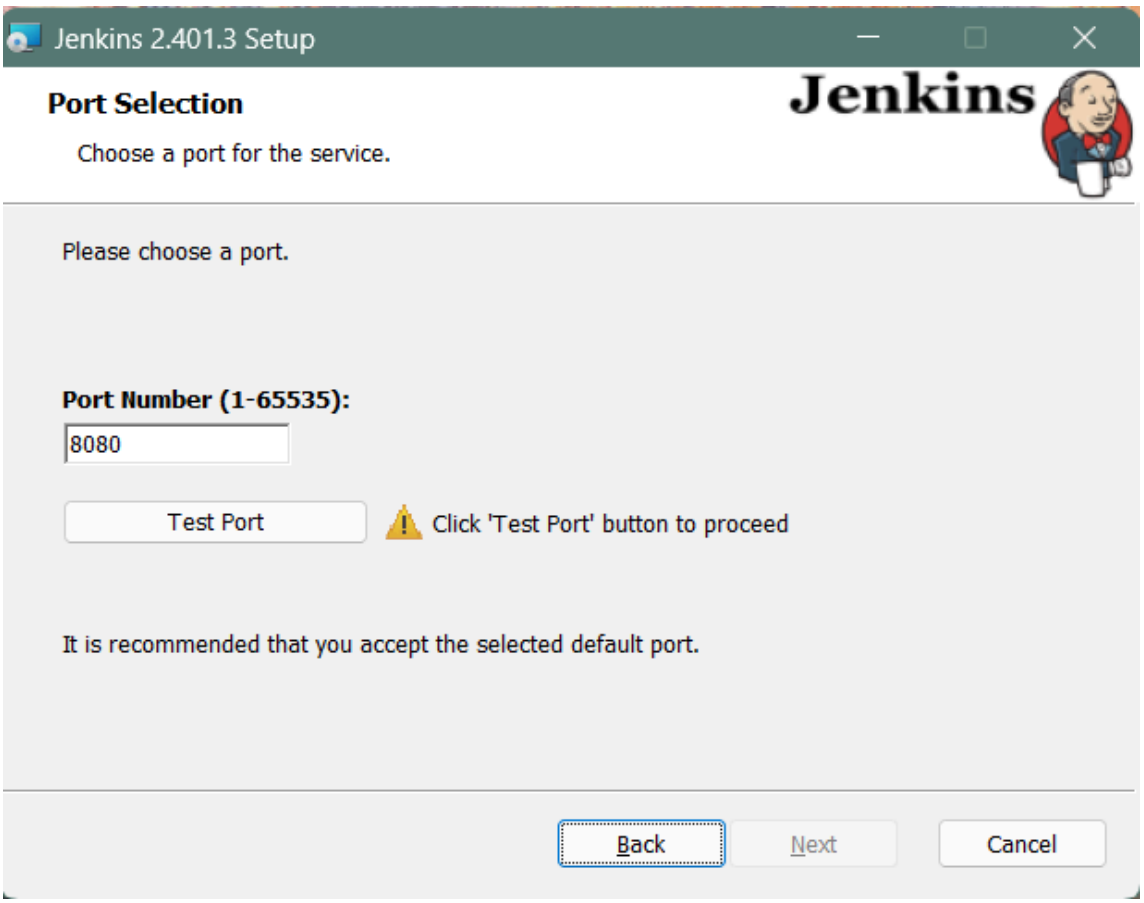
Step 1-: Install GIT
Step 2 -: Install Notepad++
Step 3 -: Install Java
Step 4 -: Install Jenkins
Step 5 -: Install Maven


jenkins.io
Jenkins
Blog
Success Stories
Contributor Spotlight
Documentation
Plugins
Community
Subprojects
Security
About
Download
Search
Jenkins
Build great things at any scale
The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.
Download
Documentation
Alpha Omega Foundation
Content Security Policy Grant
Alpha Omega Foundation has provided a grant to Jenkins to improve the implementation of Content Security Policy. Read more about what this means for the Jenkins project.
Content Security Policy grant

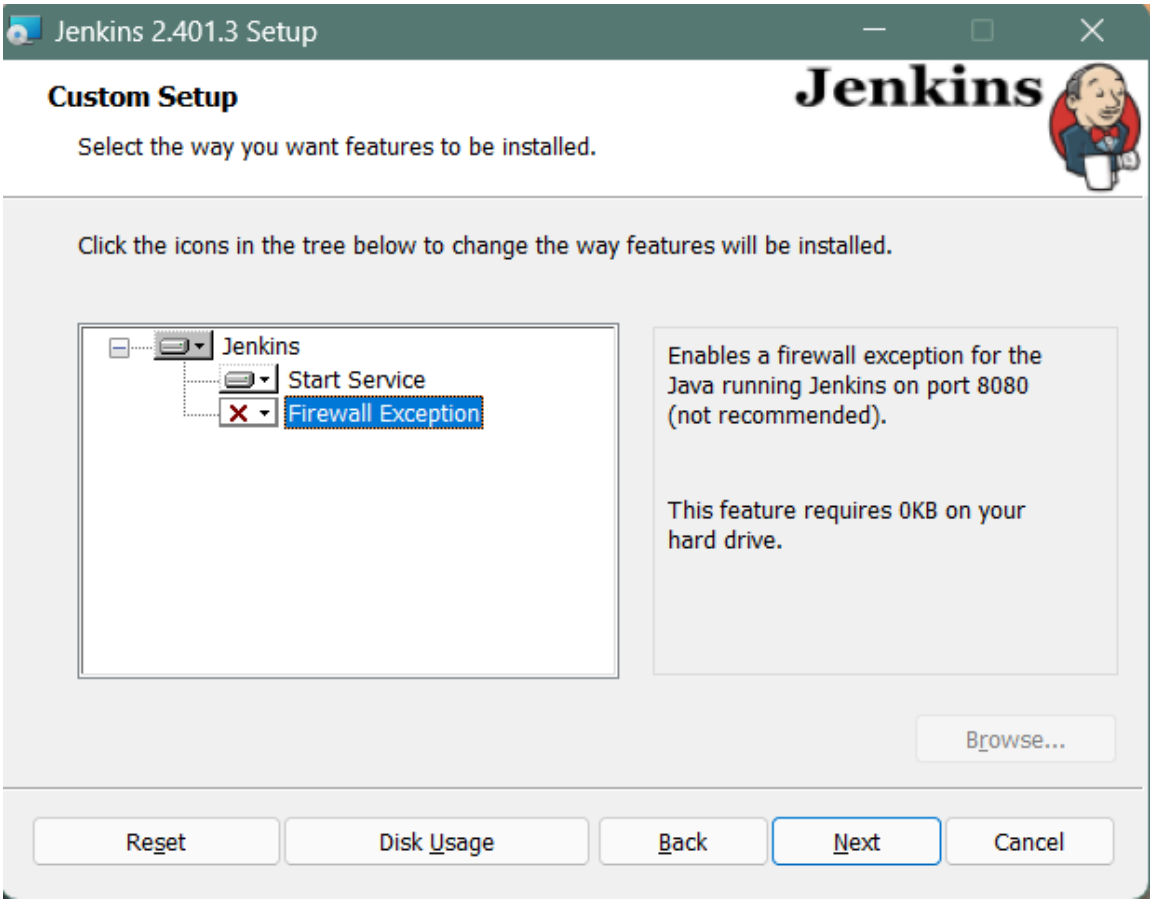

Jenkins 2.401.3 Setup
Ready to install Jenkins 2.401.3
Jenkins
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel


Jenkins 2.401.3 Setup
Destination Folder
Click Next to install to the default folder or click Change to choose another.
Jenkins
Install Jenkins 2.401.3 to:
C:\Program Files\Jenkins\
Change...
Back
Next
Cancel

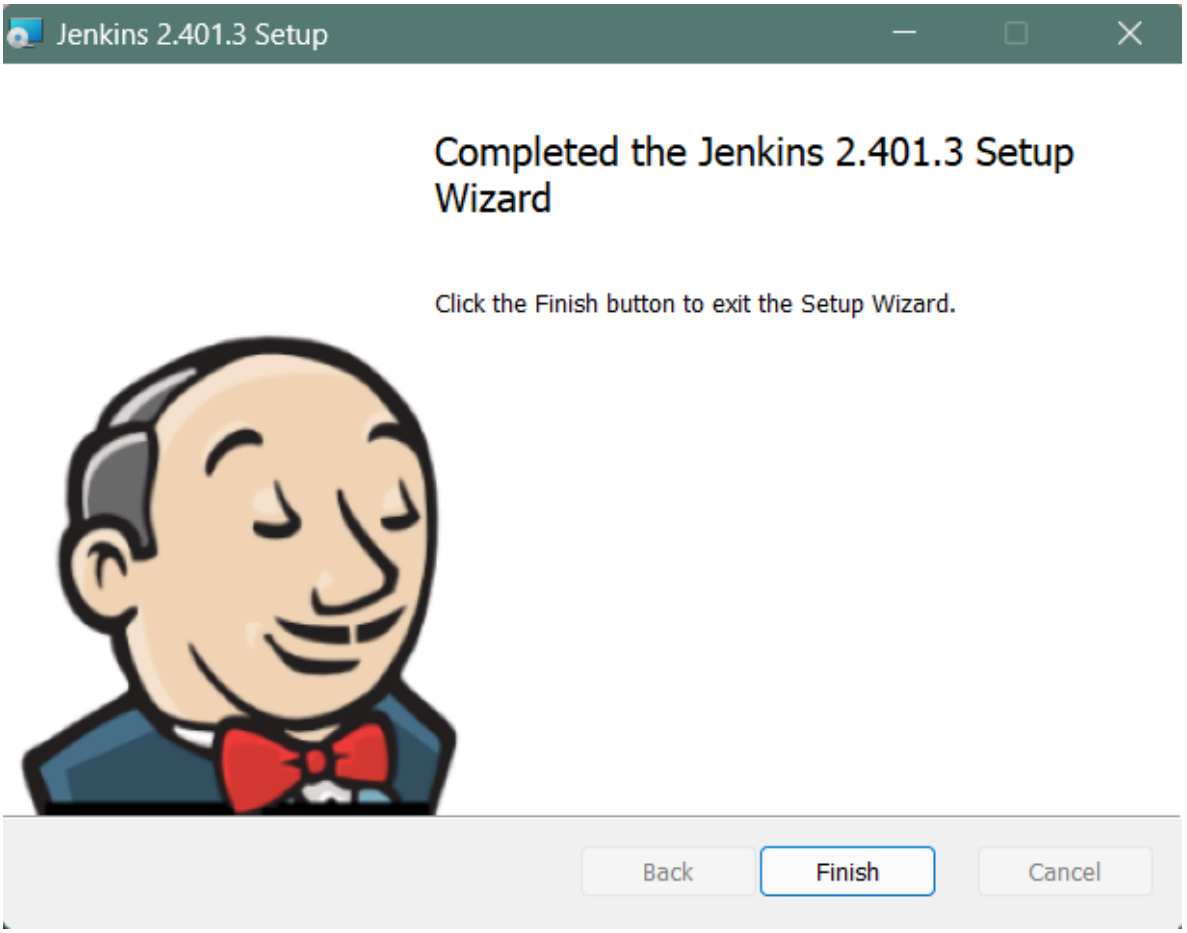

Jenkins 2.401.3 Setup
Service Logon Credentials
Enter service credentials for the service.
Jenkins
Jenkins 2.401.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.401.3 to run successfully.
Logon Type:
Run service as LocalSystem (not recommended)
Run service as local or domain user:
Account:
user
Password:
Test Credentials
Credentials must be tested to continue
Back
Next
Cancel


Jenkins 2.401.3 Setup
Port Selection
Choose a port for the service.
Jenkins
Please choose a port.
Port Number (1-65535):
8080
Test Port
Click 'Test Port' button to proceed
It is recommended that you accept the selected default port.
Back
Next
Cancel

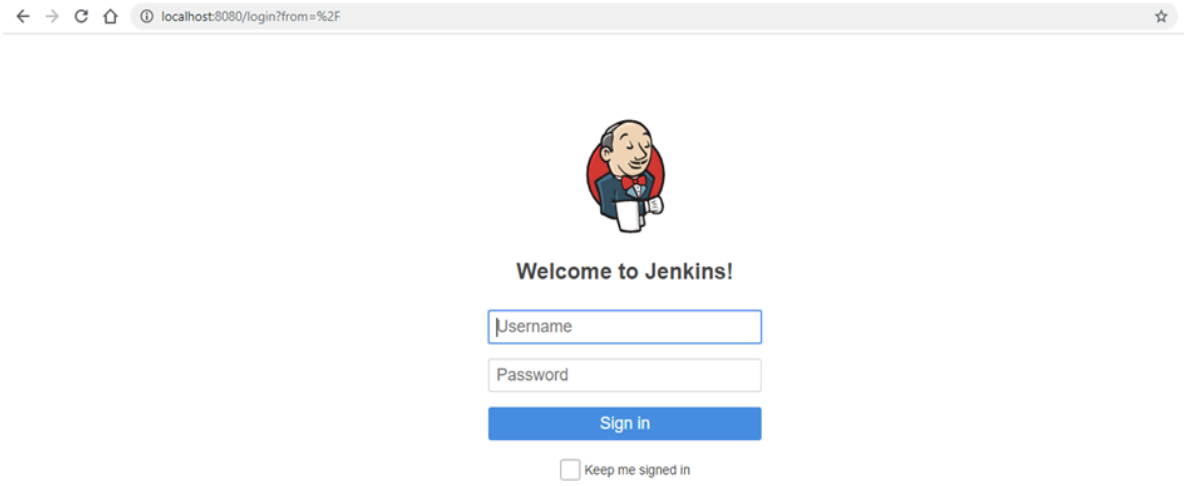

Jenkins 2.401.3 Setup
Jenkins 2.401.3 Setup
Select Java home directory (JDK or JRE)
Jenkins
Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 11 and 17 are supported by Jenkins.
C:\Program Files\Eclipse Adoptium\jdk-17.0.8.7-hotspot\
Change...
Back
Next
Cancel


Jenkins 2.401.3 Setup
Custom Setup
Select the way you want features to be installed.
Jenkins
Click the icons in the tree below to change the way features will be installed.
Jenkins
Start Service
Firewall Exception
Enables a firewall exception for the Java running Jenkins on port 8080 (not recommended).
This feature requires 0KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel


Jenkins 2.401.3 Setup
Welcome to the Jenkins 2.401.3 Setup Wizard
The Setup Wizard will install Jenkins 2.401.3 on your computer.
Click Next to continue or Cancel to exit the Setup Wizard.
Back
Next
Cancel



Once installation is done, open Jenkins dashboard using http://localhost:8080



Conclusion: Hence we have successfully understood Continuous Integration, installed and configured Jenkins with Maven/Ant/Gradle to setup a build Job