

# 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.

## Software Engineering & Project Management Lab Experiment No: - 04

**Aim: To understand Continuous Integration and configure Jenkins.**

### 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:

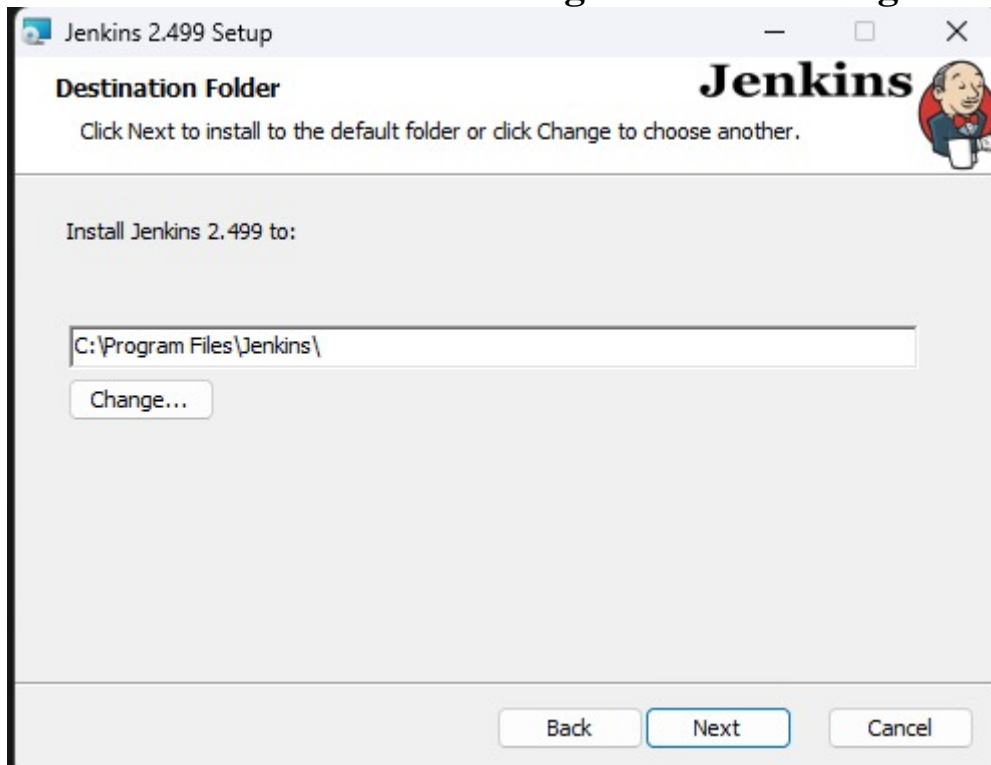
CI promotes collaboration by providing a central platform where developers can share and integrate their work regularly.

### Implementation:



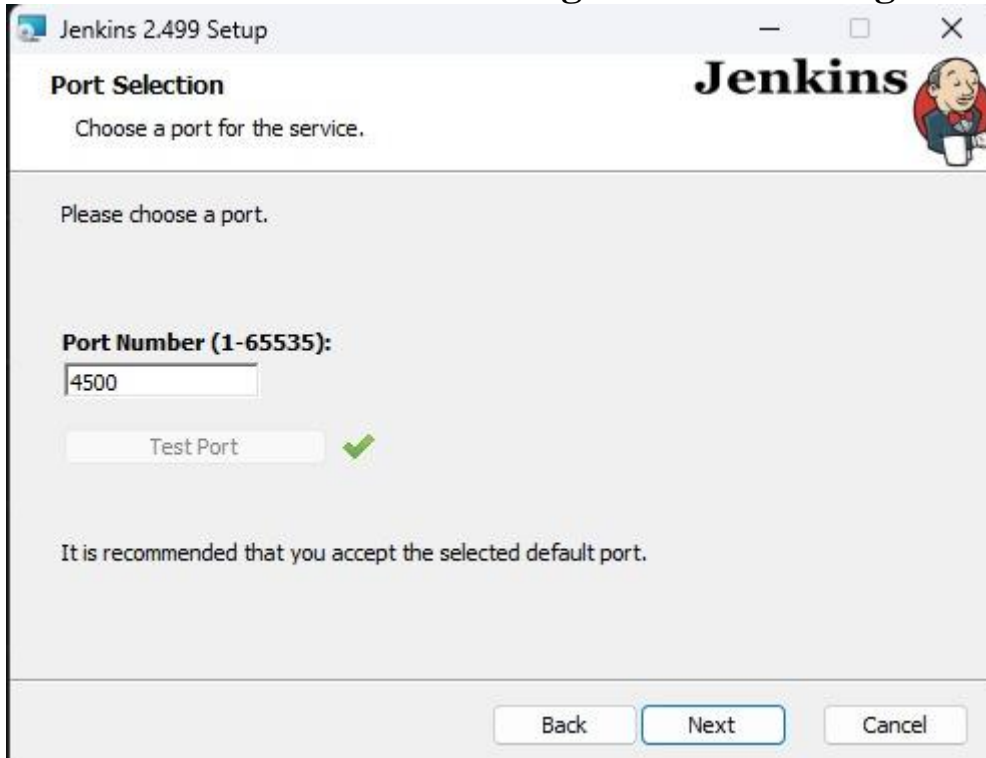
## Software Engineering & Project Management Lab Experiment No: - 04

**Aim: To understand Continuous Integration and configure Jenkins.**



## Software Engineering & Project Management Lab Experiment No: - 04

**Aim: To understand Continuous Integration and configure Jenkins.**



The screenshot shows the 'Jenkins 2.499 Setup' window with the 'Port Selection' tab. The window has a title bar with standard Windows controls and the Jenkins logo. The main content area is titled 'Port Selection' and includes the instruction 'Choose a port for the service.' Below this, it says 'Please choose a port.' A text input field labeled 'Port Number (1-65535):' contains the value '4500'. To the right of the input field is a green checkmark icon. Below the input field is a 'Test Port' button. At the bottom of the window are three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'. A message at the bottom states: 'It is recommended that you accept the selected default port.'

Jenkins 2.499 Setup

**Port Selection**

Choose a port for the service.

Please choose a port.

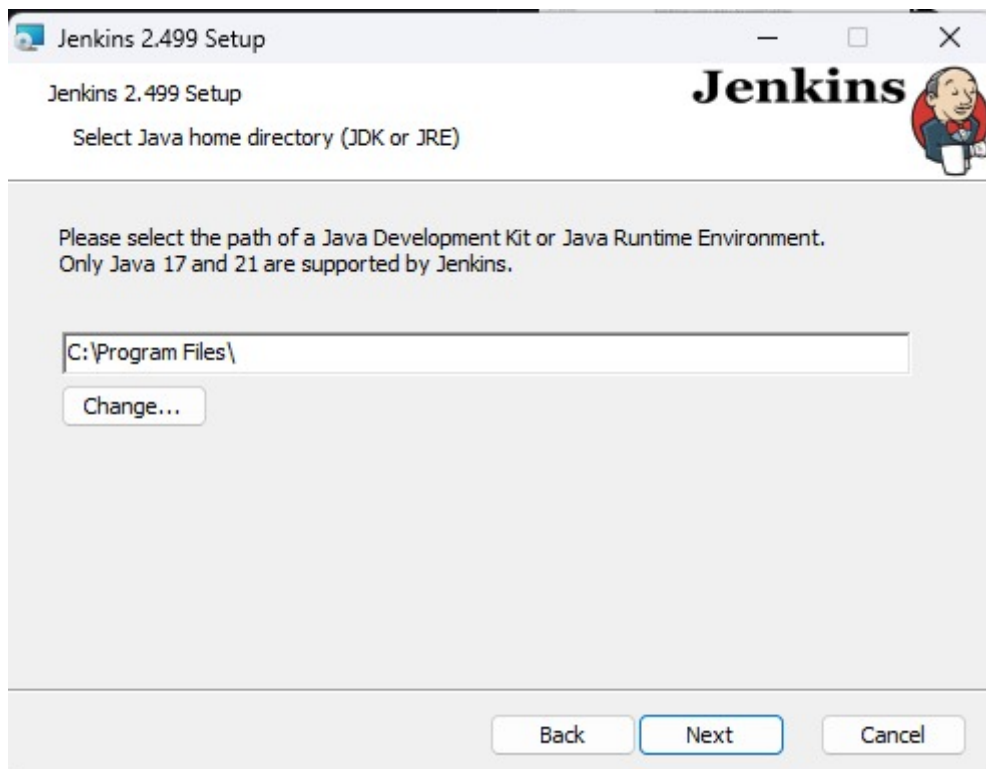
**Port Number (1-65535):**

4500

Test Port

It is recommended that you accept the selected default port.

Back Next Cancel



The screenshot shows the 'Jenkins 2.499 Setup' window with the 'Select Java home directory (JDK or JRE)' tab. The window has a title bar with standard Windows controls and the Jenkins logo. The main content area is titled 'Select Java home directory (JDK or JRE)' and includes the instruction 'Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 17 and 21 are supported by Jenkins.' Below this is a text input field containing the path 'C:\Program Files\'. To the right of the input field is a 'Change...' button. At the bottom of the window are three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.

Jenkins 2.499 Setup

**Select Java home directory (JDK or JRE)**

Please select the path of a Java Development Kit or Java Runtime Environment. Only Java 17 and 21 are supported by Jenkins.

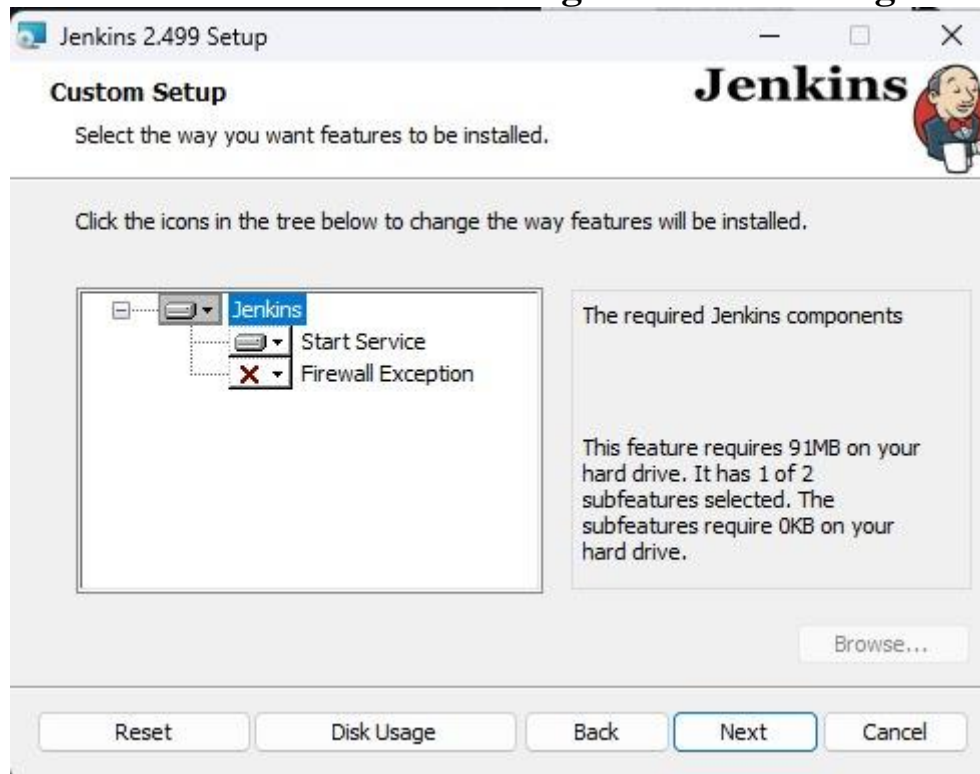
C:\Program Files\

Change...

Back Next Cancel

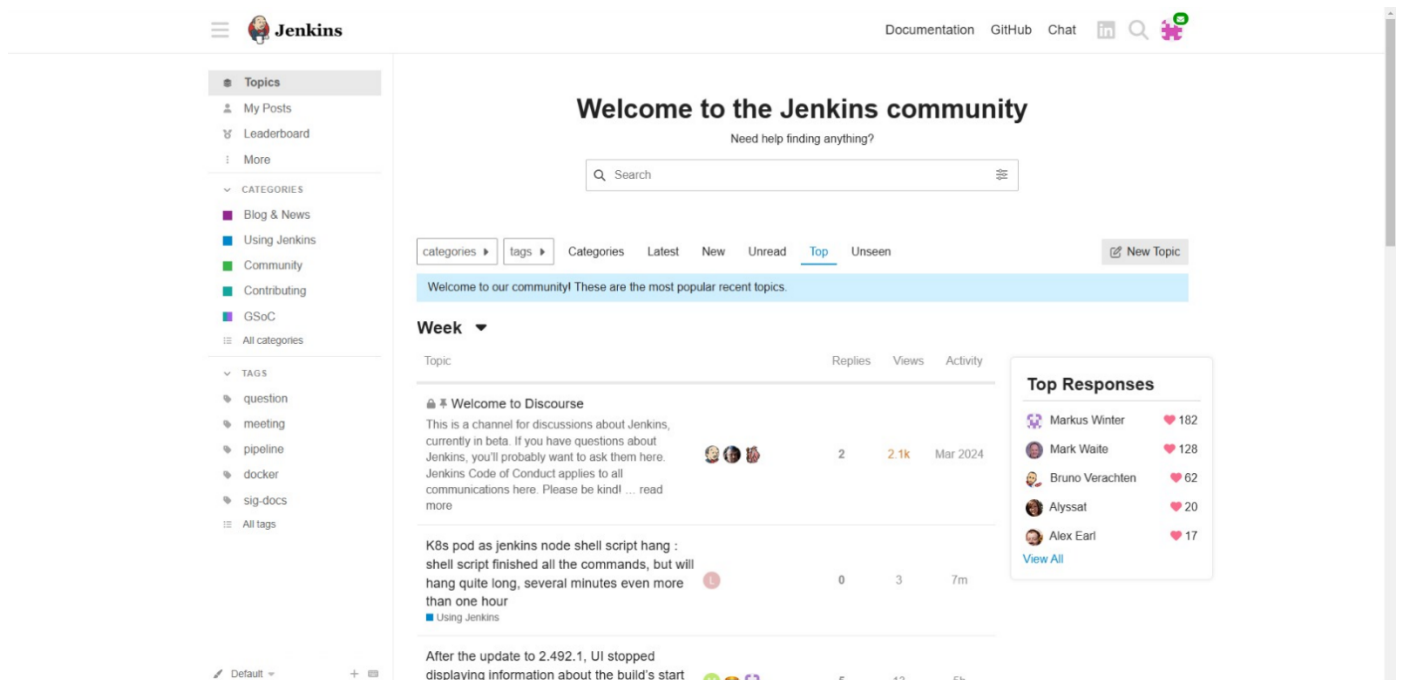
## Software Engineering & Project Management Lab Experiment No: - 04

**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.

### Getting Started

## Getting Started

<input checked="" type="checkbox"/> Folders	<input checked="" type="checkbox"/> OWASP Markup Formatter	<input type="checkbox"/> Build Timeout	<input type="checkbox"/> Credentials Binding	** Ionicons API
<input type="checkbox"/> Timestampers	<input type="checkbox"/> Workspace Cleanup	<input type="checkbox"/> Ant	<input type="checkbox"/> Gradle	Folders
<input type="checkbox"/> Pipeline	<input type="checkbox"/> GitHub Branch Source	<input type="checkbox"/> Pipeline: GitHub Groovy Libraries	<input type="checkbox"/> Pipeline: Stage View	OWASP Markup Formatter
<input type="checkbox"/> Git	<input type="checkbox"/> SSH Build Agents	<input type="checkbox"/> Matrix Authorization Strategy	<input type="checkbox"/> PAM Authentication	
<input type="checkbox"/> LDAP	<input type="checkbox"/> Email Extension	<input type="checkbox"/> Mailer		

Jenkins 2.426.3

Dashboard >

[Add description](#)

+ New Item

[People](#)

[Build History](#)

[Manage Jenkins](#)

[My Views](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

### Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

#### Start building your software project

Create a job



#### Set up a distributed build

Set up an agent



Configure a cloud



Learn more about distributed builds



REST API Jenkins 2.426.3

**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.