**Experiment no. 4 : To understand Continuous Integration, Install and Configure Jenkins with Maven/Ant/Gradle to setup a build job**

**1. Introduction to Continuous Integration (CI)**

Continuous Integration (CI) is a software development practice where developers frequently integrate code into a shared repository. Each integration is automatically tested to detect errors early. The key goals of CI are:

* Detect and fix bugs early.
* Improve software quality.
* Speed up the development process.
* Ensure code consistency and automation.

CI relies on automated build tools and testing frameworks to verify code changes before merging them into the main codebase.

**2. Role of Jenkins in CI**

Jenkins is an open-source automation server widely used for implementing CI/CD pipelines. It automates the process of building, testing, and deploying applications. Jenkins can be integrated with various build tools such as Maven, Ant, and Gradle to streamline software development.

* Free and open-source.
* Supports multiple plugins for extensibility.
* Allows automated builds and tests.
* Provides an easy-to-use web-based interface.
* Can integrate with various DevOps tools like Git, Docker, Kubernetes, etc.

**3. Installing Jenkins**

**a. Prerequisites**

Before installing Jenkins, ensure that the following software is installed:

* Java (JDK 8 or later)
* Apache Maven/Ant/Gradle (for build automation)
* Git (for version control)

**b. Installation Steps**

1. **Download Jenkins**:
   * Download the latest stable Jenkins version from [Jenkins official website](https://www.jenkins.io/).
   * Install Jenkins using the appropriate installer (Windows, Linux, macOS).
2. **Start Jenkins**:
   * On Windows: Run java -jar jenkins.war from the command line.
   * On Linux/macOS: Use system services or run Jenkins as a Docker container.
3. **Access Jenkins**:
   * Open a web browser and navigate to http://localhost:8080/.
   * Unlock Jenkins using the initial administrator password found in the installation logs.
4. **Install Plugins**:
   * During the initial setup, install recommended plugins.
   * Additional plugins like "Maven Integration", "Gradle Plugin", or "Ant Plugin" can be installed from **Manage Jenkins > Manage Plugins**.

**4. Configuring Jenkins with Maven/Ant/Gradle**

Jenkins can be configured to use any of the popular build tools:

**a. Configuring Maven in Jenkins**

1. **Install Maven Plugin**:
   * Navigate to Manage Jenkins > Manage Plugins.
   * Search for **Maven Integration Plugin** and install it.
2. **Set Up Maven in Jenkins**:
   * Go to Manage Jenkins > Global Tool Configuration.
   * Add a new Maven installation by specifying the path or letting Jenkins install it automatically.
3. **Create a Maven Build Job**:
   * Click on New Item > Freestyle Project.
   * In **Build Environment**, select **Invoke top-level Maven targets**.
   * Provide the build goals (e.g., clean install).
   * Save and trigger the build.

**b. Configuring Ant in Jenkins**

1. **Install Ant Plugin**:
   * Go to Manage Plugins and install the **Ant Plugin**.
2. **Set Up Ant**:
   * Go to Manage Jenkins > Global Tool Configuration.
   * Add Ant by specifying its installation path.
3. **Create an Ant Build Job**:
   * Create a new **Freestyle Project**.
   * In **Build Steps**, choose **Invoke Ant**.
   * Provide the target (e.g., compile or build.xml location).
   * Save and build the job.

**c. Configuring Gradle in Jenkins**

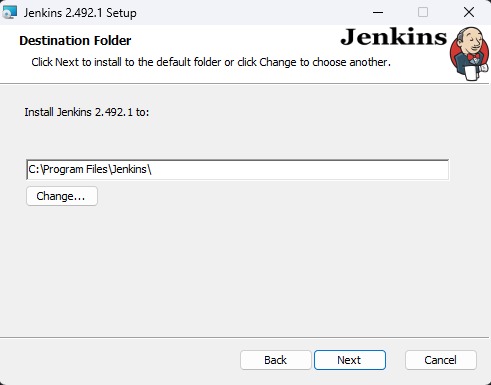
1. **Install Gradle Plugin**:
   * Go to Manage Plugins and install the **Gradle Plugin**.
2. **Set Up Gradle**:
   * Navigate to Manage Jenkins > Global Tool Configuration.
   * Add Gradle by specifying its installation path.
3. **Create a Gradle Build Job**:
   * Create a new **Freestyle Project**.
   * In **Build Steps**, select **Invoke Gradle Script**.
   * Specify tasks (e.g., clean build).
   * Save and run the build.

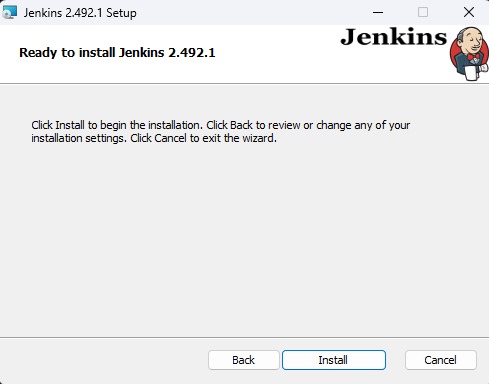
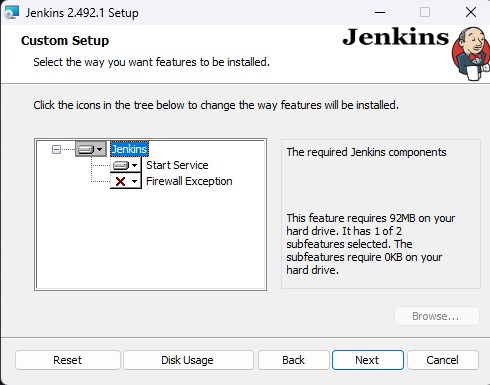
**5. Running and Monitoring the Build Job**

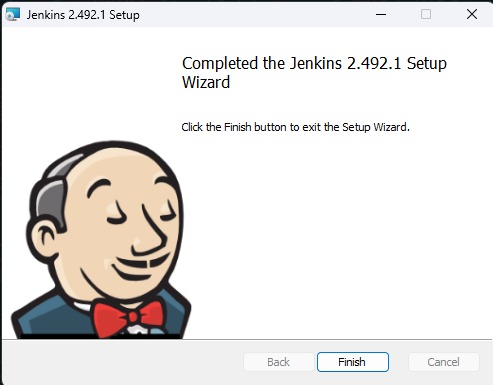
After setting up a build job:

* Click **Build Now** to trigger the job.
* Monitor progress in the **Build Console Output**.
* Review logs and test results.
* Set up notifications (email, Slack, etc.) for build failures.









**Conclusion**

Thus, we have successfully installed Jenkins and configure with maven/gradle/ant.