

Software Engineering & Project Management Lab Experiment No: - 04

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

Theory:

Continuous Integration (CI) is a DevOps practice where code changes are automatically built, tested, and integrated into a shared repository multiple times a day. It helps in early detection of errors, reduces integration problems, and improves software quality.

Jenkins: An Overview

Jenkins is an open-source CI/CD automation tool used for building, testing, and deploying applications. It allows developers to automate software development workflows and ensures a seamless integration process. Jenkins supports various build tools like **Maven**, **Ant**, and **Gradle** to compile and package applications.

Installing and Configuring Jenkins

1. **Download and Install Jenkins** ○ Install Java (JDK) as a prerequisite. ○ Download Jenkins from the official website and install it on the server. ○ Start Jenkins and configure initial setup using an administrator password.
2. **Installing Build Tools** ○ Install **Maven**, **Ant**, or **Gradle** depending on project requirements. ○ Configure Jenkins to recognize the installed build tool.
3. **Creating a Build Job in Jenkins** ○ Navigate to **Jenkins Dashboard** → **New Item** → **Freestyle Project/Pipeline**.
 - Configure the **Git repository URL** to fetch the source code.
 - Select the **Build Tool (Maven/Ant/Gradle)** and define the build command.
 - Set up triggers (e.g., Git webhooks) for automatic build execution.
 - Save and trigger the build job to verify the setup.

To install Jenkins following software packages are required:

Software Engineering & Project Management Lab Experiment No: - 04

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

- 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

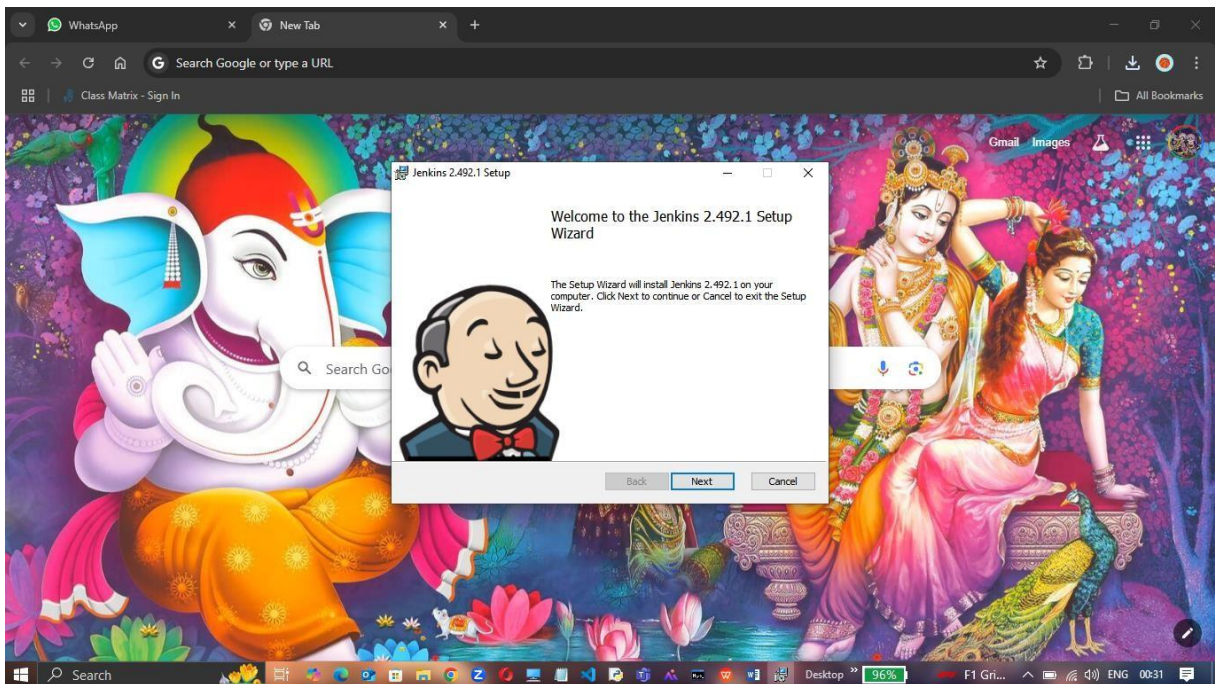
Step 5 -: Install Maven

Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.

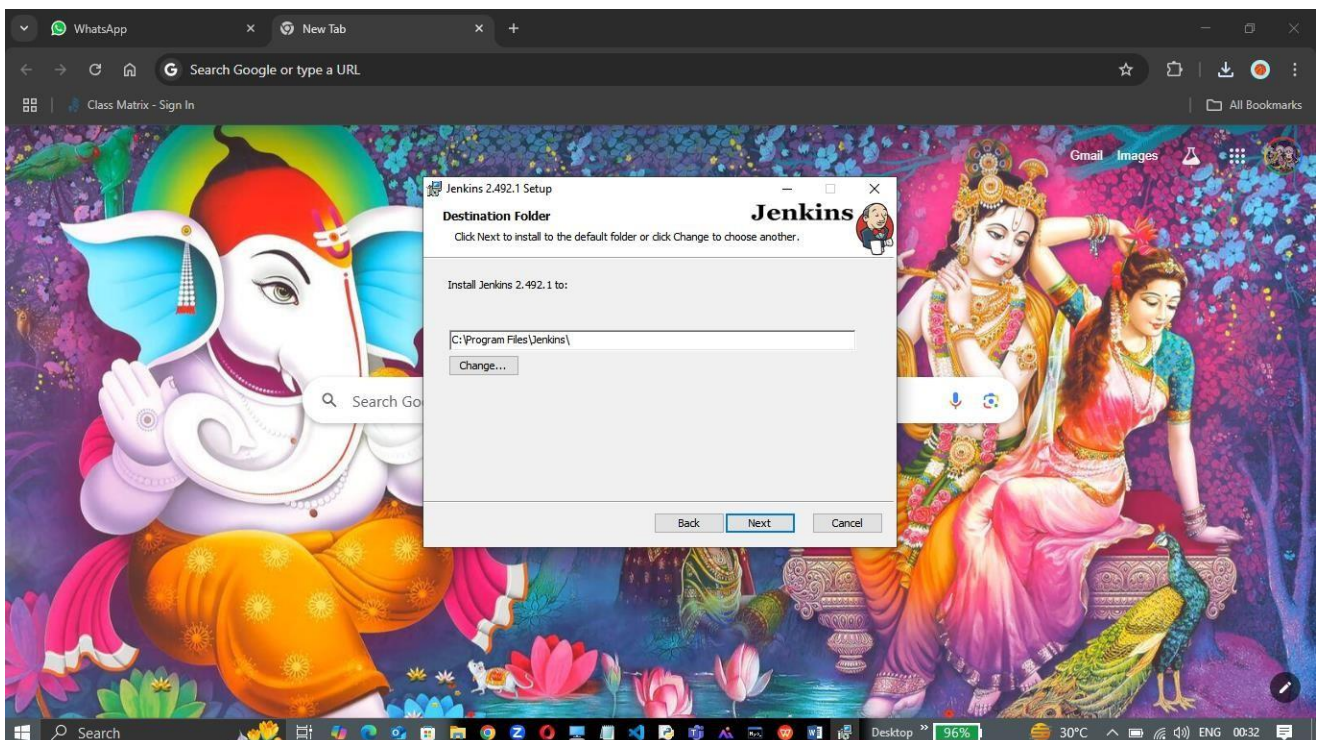
Step 1-: Open <https://www.jenkins.io/doc/book/installing/windows/> and install Jenkins.
Open the installed .exe setup

Software Engineering & Project Management Lab Experiment No: - 04

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



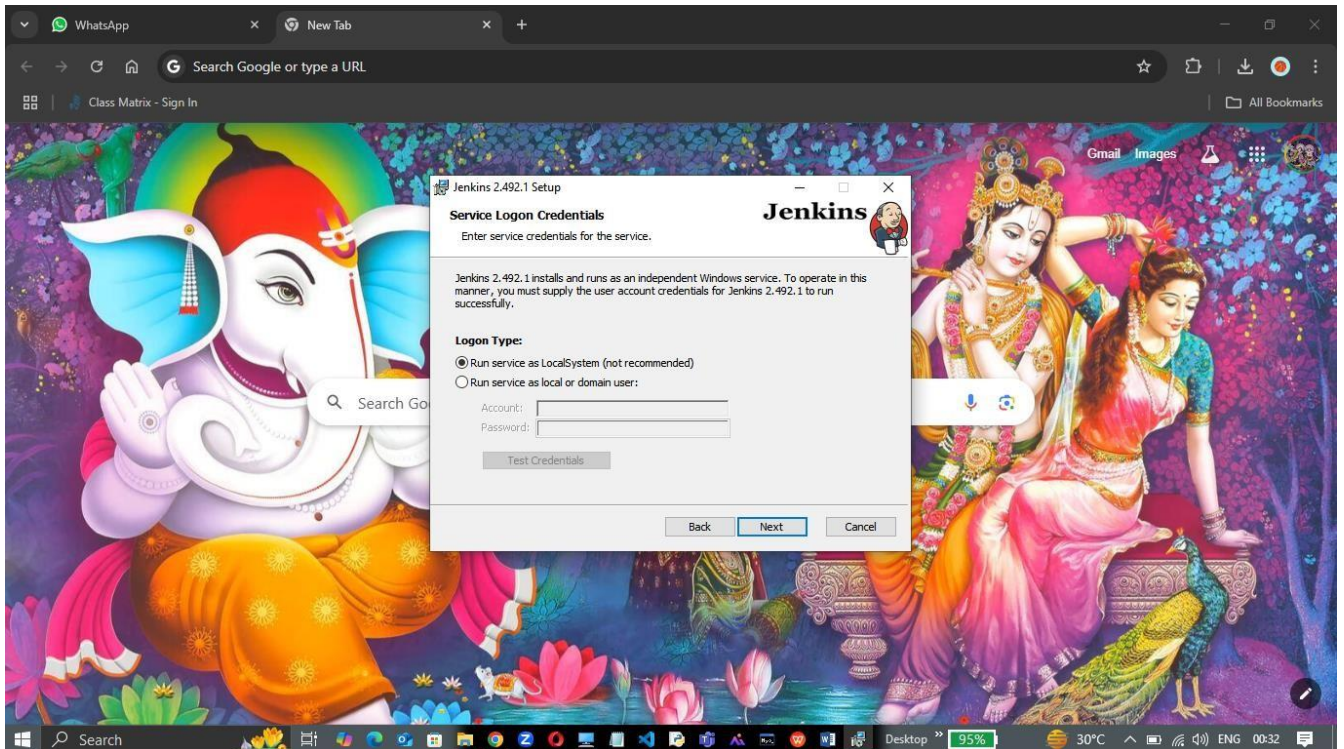
Step 2: Locate the folder where you want to install Jenkins in the location path:



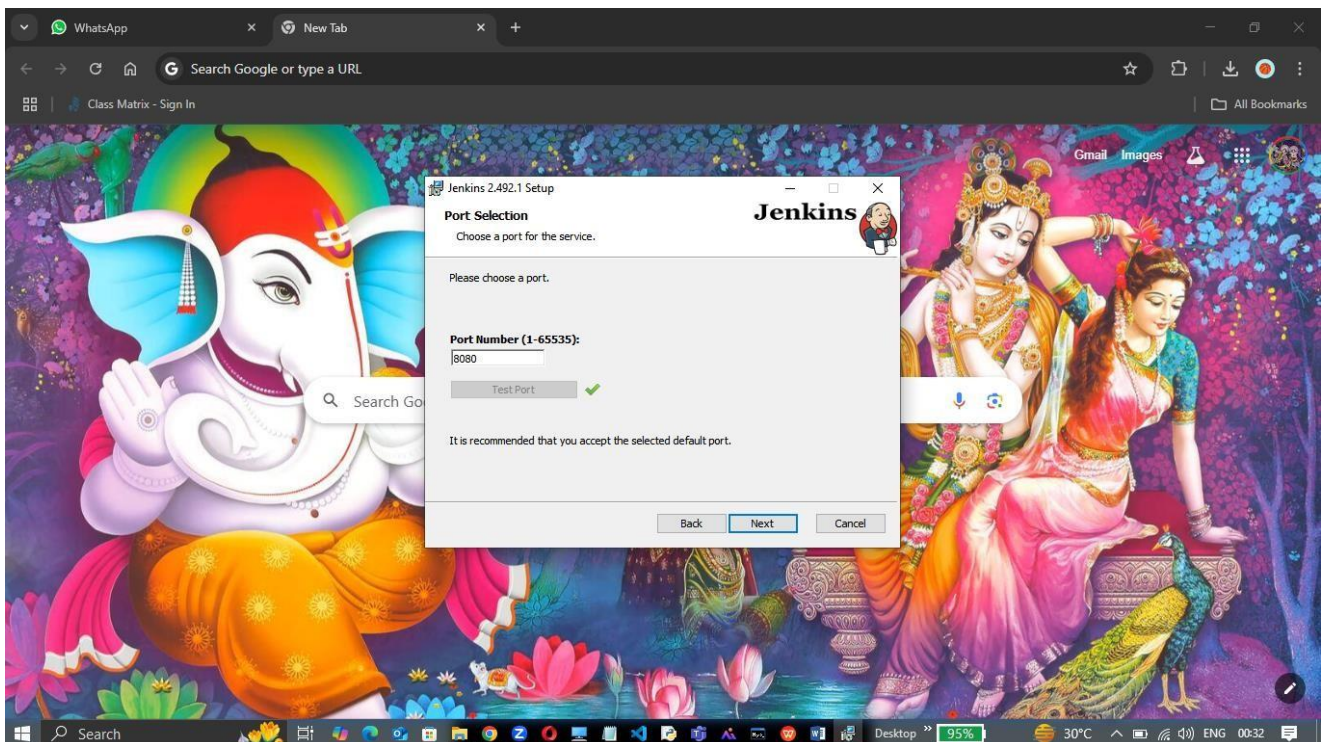
Software Engineering & Project Management Lab Experiment No: - 04

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

Step 3: Select service as Local System and proceed to Next.



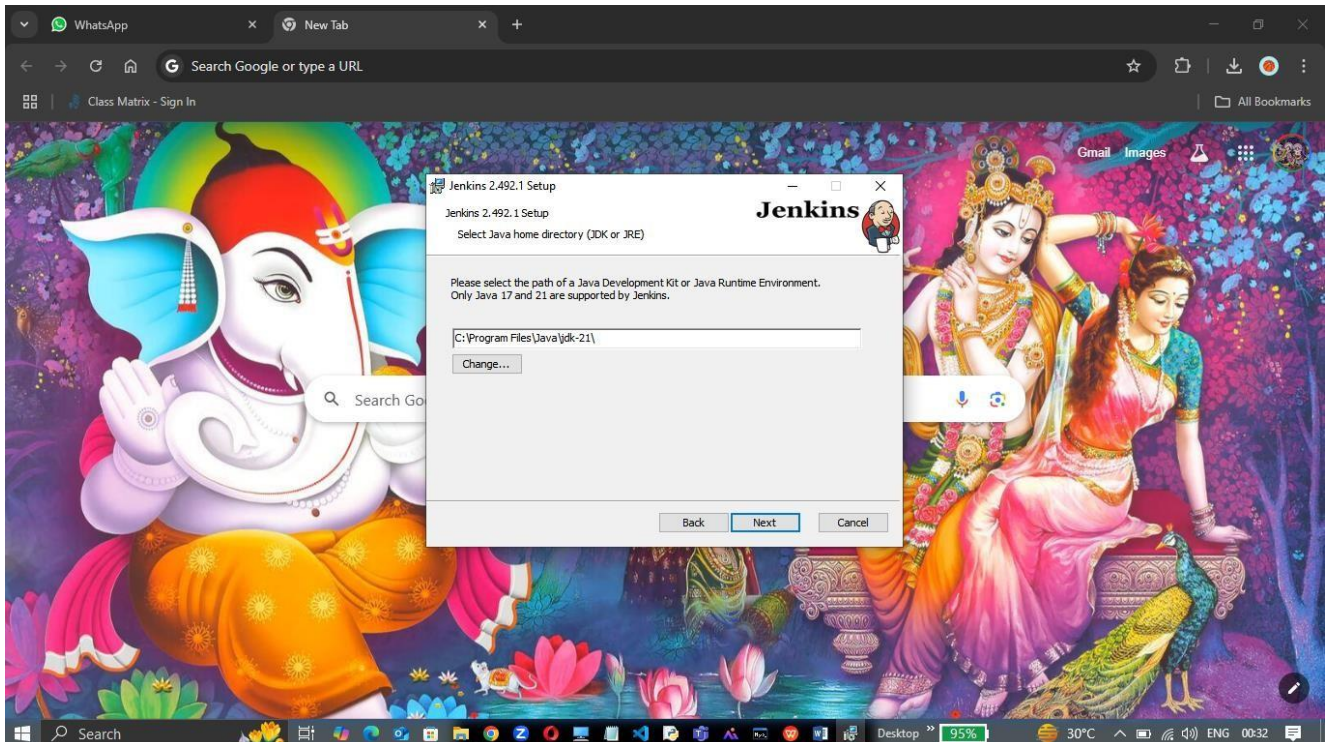
Step 4: Select the port 8080 and click Test Port button. The green tick will appear after which you can proceed to Next.



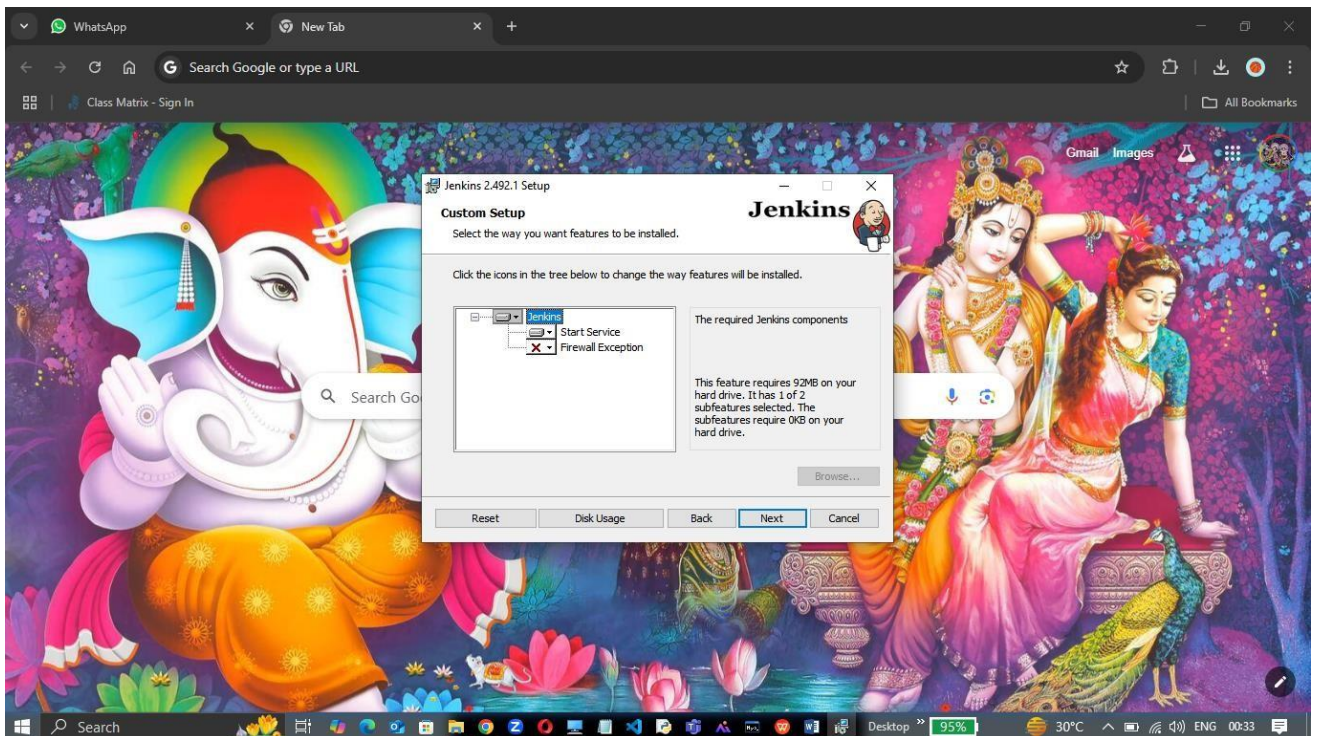
Software Engineering & Project Management Lab Experiment No: - 04

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

Step 5: Locate the folder where you have installed JDK in the location path:

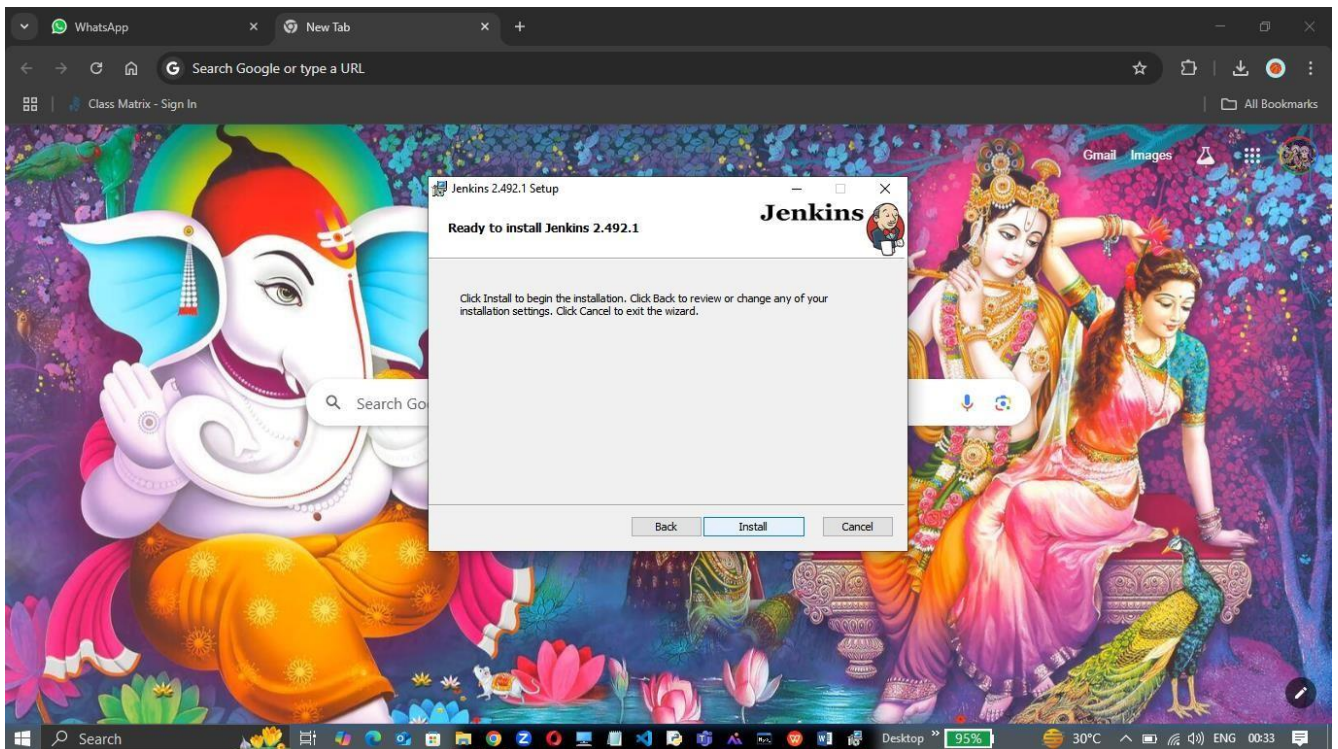


Proceed to Next

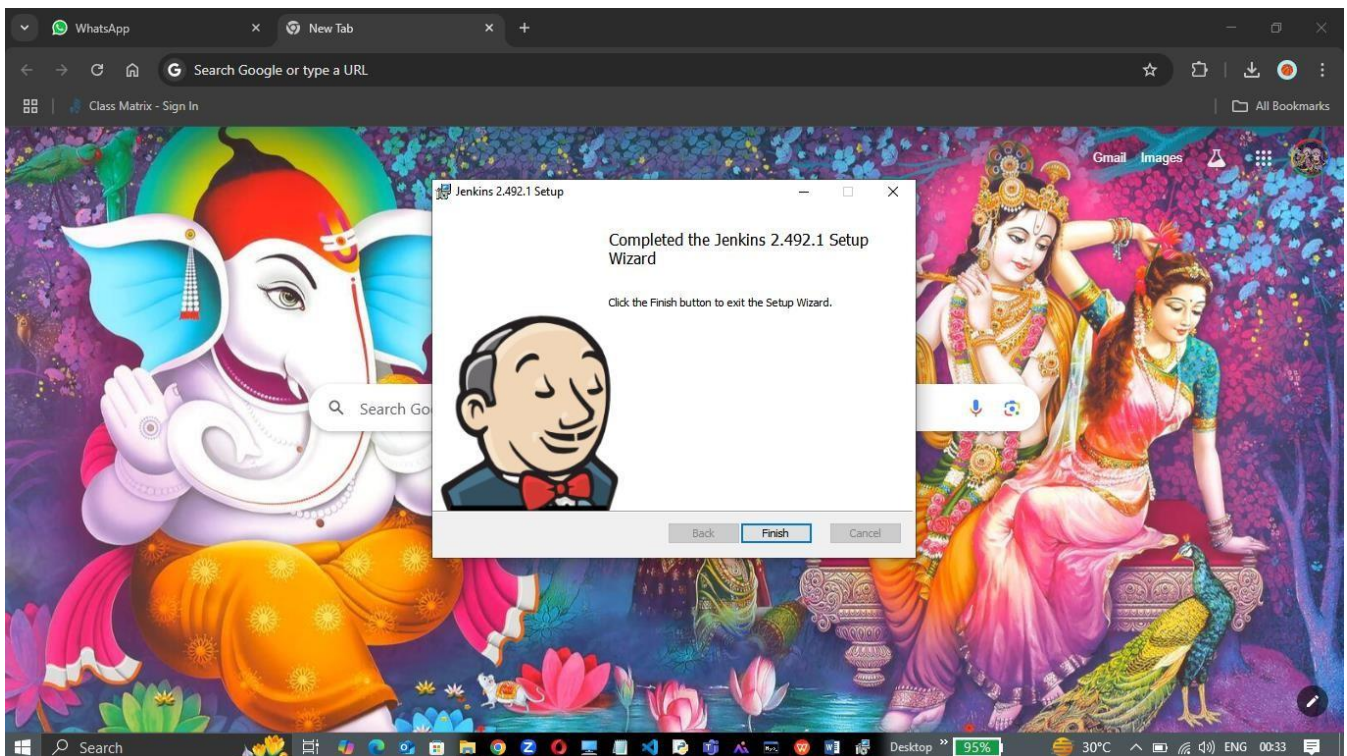


Software Engineering & Project Management Lab Experiment No: - 04

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



On clicking 'Install', installation is finished.

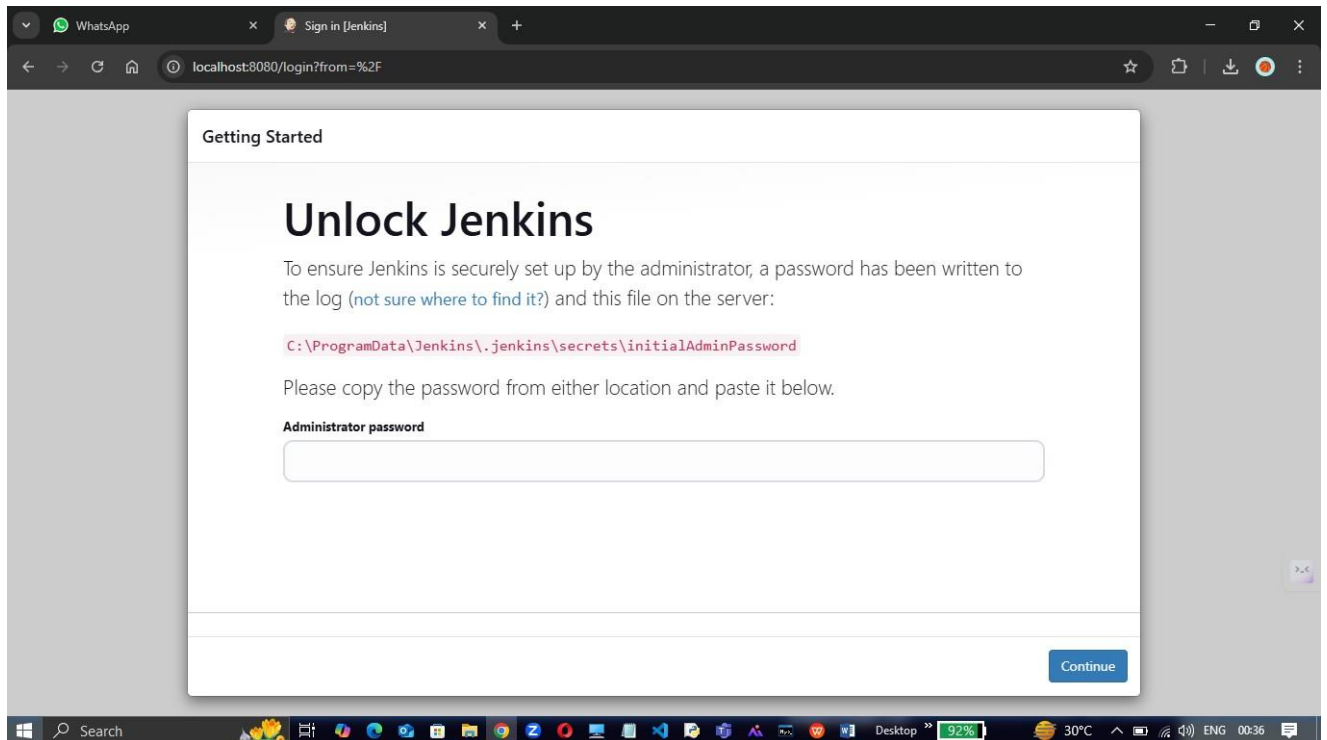


Software Engineering & Project Management Lab Experiment No: - 04

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

Step 6: Once Installation is done, you can test the Jenkins on <http://localhost:8080> on the browser.

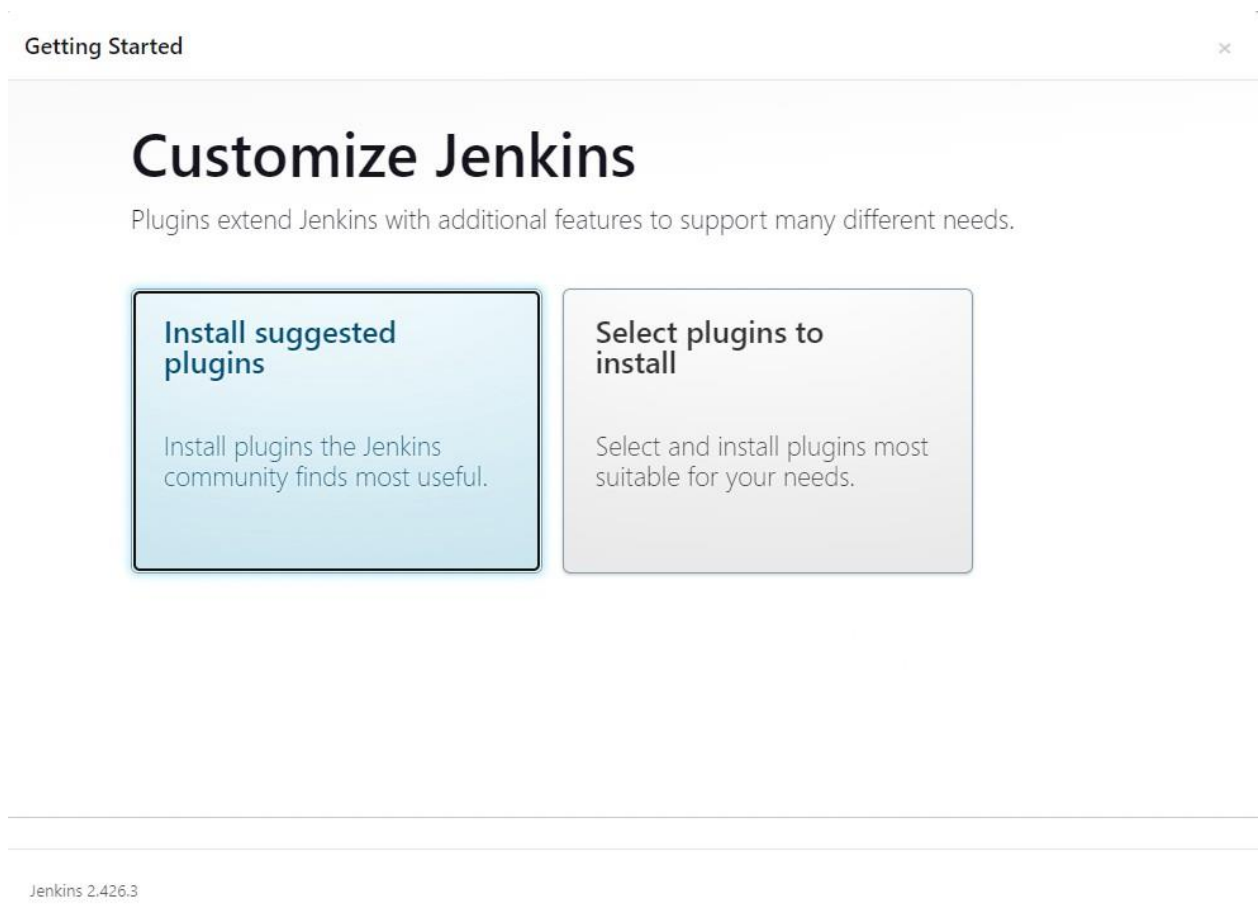
First time, when you open Jenkins portal it will ask to put admin default password which is stored in `/var/lib/jenkins/secrets/initialAdminPassword` file.



Step 7: On entering the password, you can continue to choose “Install Suggested Plugins”

Software Engineering & Project Management Lab Experiment No: - 04

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



Once plugins are installed, click on next and specify the admin details along with the new password for Jenkins admin and click on finish to complete the installation.

After filling the details, click on Save & Continue, you will be redirected to the dashboard.

Software Engineering & Project Management Lab Experiment No: - 04

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

Getting Started

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

Jenkins 2.426.3

Dashboard >

+ 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

Software Engineering & Project Management Lab Experiment No: - 04

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

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address







Jenkins 2.426.3

[Skip and continue as admin](#) [Save and Continue](#)

Dashboard >

Enter an item name

» Required field

-  **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
-  **Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
-  **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
-  **Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
-  **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.
-  **Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

[OK](#)

Software Engineering & Project Management Lab Experiment No: - 04

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

Dashboard > example 1 > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps
- Post-build Actions

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

☐ With Ant ?

Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

echo "hello tsec"

Advanced ▾

Add build step ▾

Save

Apply

Jenkins Search (CTRL+K) ? 1 Aditya Parulekar ▾ log out

Dashboard > example 1 > #11 > Console Output

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#11'

Previous Build

✓ Console Output

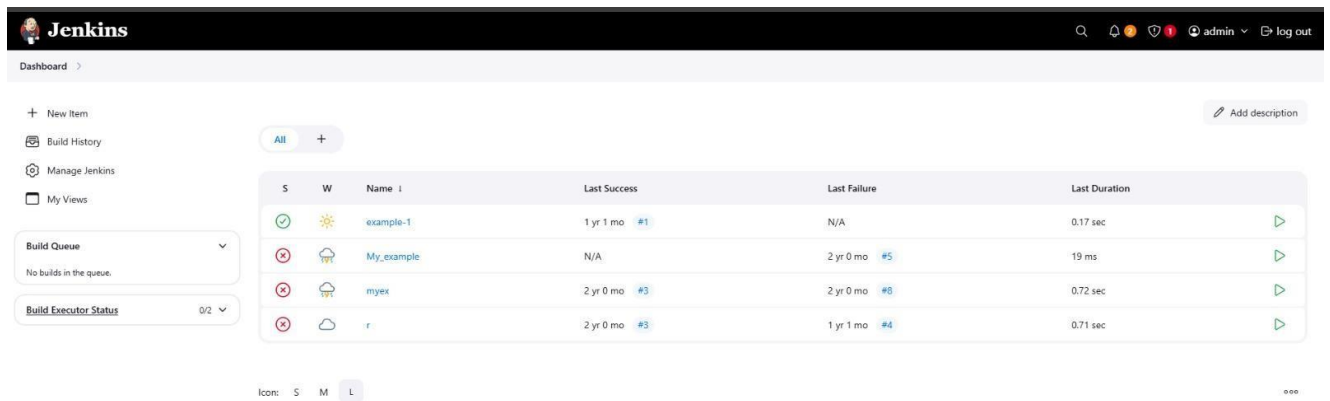
Started by user Muskan Tolani
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\example 1
[example 1] \$ cmd /c call C:\Windows\TEMP\jenkins6203665954710491391.bat

C:\ProgramData\Jenkins\jenkins\workspace\example 1>echo "hello tsec"
"hello tsec"

C:\ProgramData\Jenkins\jenkins\workspace\example 1>exit 0
Finished: SUCCESS

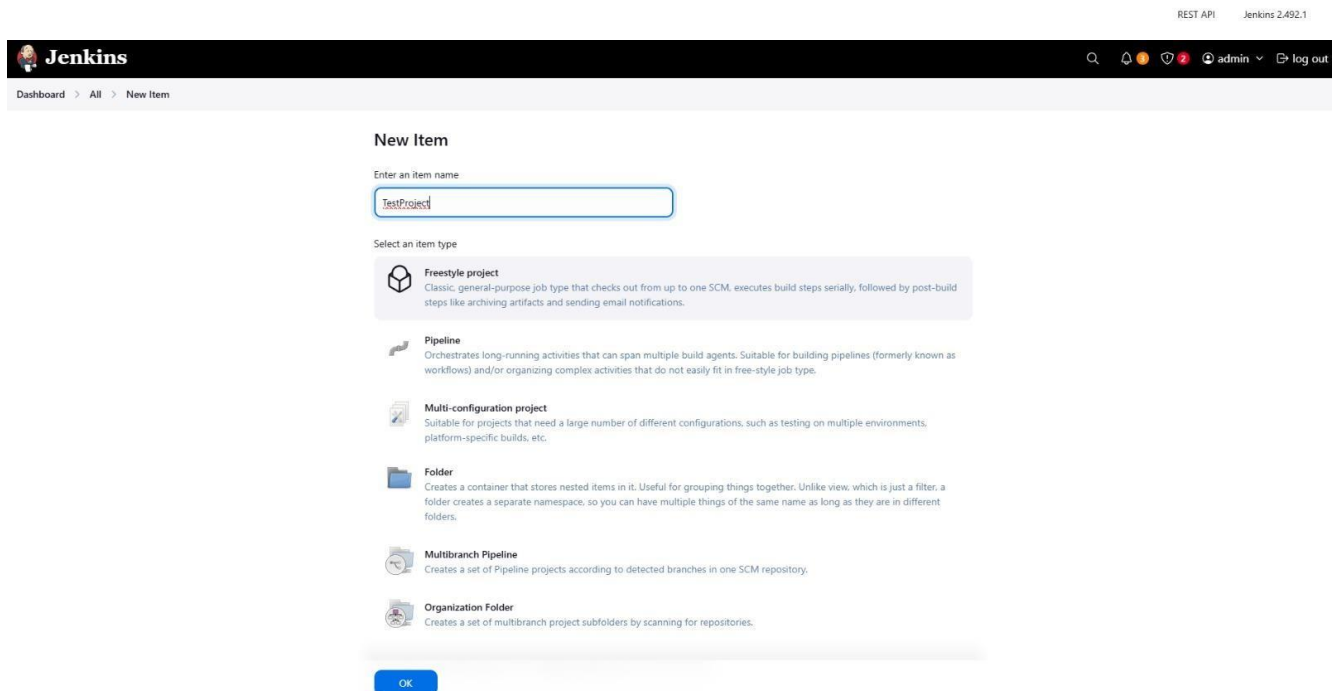
Software Engineering & Project Management Lab Experiment No: - 04

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



The screenshot shows the Jenkins Dashboard. The top navigation bar includes the Jenkins logo, a search icon, notification icons, and user information (admin, log out). The left sidebar contains links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. The main area displays a table of build jobs with columns for status, icon, name, last success, last failure, and last duration. The table lists four jobs: 'example-1' (success), 'My_example' (failure), 'myex' (failure), and 'r' (failure). Below the table, there are filters for 'Icons', 'S', 'M', and 'L'. The bottom right corner shows 'REST API' and 'Jenkins 2.492.1'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	example-1	1 yr 1 mo #1	N/A	0.17 sec
✗	☁	My_example	N/A	2 yr 0 mo #5	19 ms
✗	☁	myex	2 yr 0 mo #3	2 yr 0 mo #8	0.72 sec
✗	☁	r	2 yr 0 mo #3	1 yr 1 mo #4	0.71 sec



The screenshot shows the 'New Item' form in Jenkins. The top navigation bar is the same as the dashboard. The left sidebar shows 'Dashboard > All > New Item'. The main area is titled 'New Item' and contains a form to create a new job. The 'Enter an item name' field is filled with 'TestProject'. Below this, there is a 'Select an item type' section with several options: 'Freestyle project' (Classic, general-purpose job type), 'Pipeline' (Orchestrates long-running activities), 'Multi-configuration project' (Suitable for projects that need a large number of different configurations), 'Folder' (Creates a container that stores nested items), 'Multibranch Pipeline' (Creates a set of Pipeline projects according to detected branches), and 'Organization Folder' (Creates a set of multibranch project subfolders by scanning for repositories). An 'OK' button is at the bottom.

Software Engineering & Project Management Lab Experiment No: - 04

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

The screenshot displays the Jenkins Configuration page for a project named 'TestProject'. The breadcrumb navigation at the top shows 'Dashboard > TestProject > Configuration'. On the left, a sidebar lists configuration categories: General, Source Code Management, Triggers, Environment, Build Steps (selected), and Post-build Actions. The main content area is titled 'Configure' and includes a checkbox for 'With Ant'. The 'Build Steps' section, described as 'Automate your build process with ordered tasks like code compilation, testing, and deployment', contains a single step named 'Execute shell'. The command field for this step contains the text 'echo "Prasad"'. Below the command field is an 'Advanced' dropdown menu and an 'Add build step' button. The 'Post-build Actions' section, described as 'Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs', includes an 'Add post-build action' button. At the bottom of the configuration area are 'Save' and 'Apply' buttons. The top right corner of the page shows 'REST API' and 'Jenkins 2.492.1'. Below the configuration page, a second screenshot shows the Jenkins 'TestProject' overview page. This page has a dark header with the Jenkins logo and a top navigation bar with search, notifications, and user information (admin). The left sidebar lists actions: Status (selected), Changes, Workspace, Build Now, Configure, Delete Project, and Rename. The main area shows 'TestProject' with the description 'This is a test project.' and a 'Permalinks' section. A 'Builds' section at the bottom shows a list of builds for 'Today', with the first build being '#1' at '1:11 PM'.

Dashboard > TestProject > Configuration

☐ With Ant

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps**
- Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell

Command

See the list of available environment variables

```
echo "Prasad"
```

Advanced

Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

Add post-build action

Save Apply

REST API Jenkins 2.492.1

Jenkins

Dashboard > TestProject

- Status
- Changes
- Workspace
- Build Now
- Configure
- Delete Project
- Rename

TestProject

This is a test project.

Permalinks

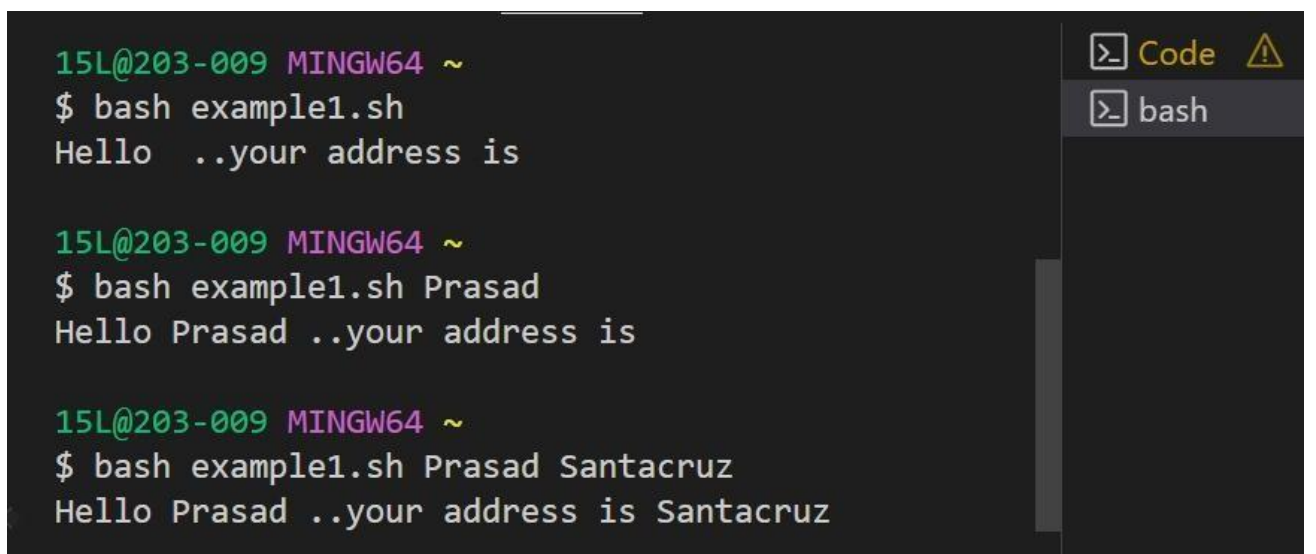
Builds

Today

#1 1:11 PM

Software Engineering & Project Management Lab Experiment No: - 04

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



Software Engineering & Project Management Lab Experiment No: - 04

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

The image displays two screenshots from the Jenkins web interface. The top screenshot shows the 'Console Output' for a build named 'TestProject' (#3). The output indicates a successful build initiated by 'admin', running as 'SYSTEM'. It shows the build directory, the execution of a batch file, and the output of a command that prints 'My name is Prasad Satpute'. The build finished successfully.

The bottom screenshot shows the 'Configuration' page for a build named 'example2'. The 'Build Steps' section is active, showing a single step 'Execute shell' with the command: `cd/Desktop
javac example2.java
java example2`. The 'Post-build Actions' section is empty. The 'Save' button is highlighted.

Dashboard > TestProject > #3 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build #3

Previous Build

Download Copy View as plain text

Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\TestProject
[TestProject] \$ cmd /c call C:\WINDOWS\TEMP\jenkins1013378486423254757.bat

C:\ProgramData\Jenkins\jenkins\workspace\TestProject>echo "My name is Prasad Satpute"
"My name is Prasad Satpute"

C:\ProgramData\Jenkins\jenkins\workspace\TestProject>exit 0
Finished: SUCCESS

OUTPUT DEBUG CONSOLE 1 TERMINAL PORTS

2 errors

15L@203-009 MINGW64 ~/Desktop
\$ javac example2.java

15L@203-009 MINGW64 ~/Desktop
\$ java example2
PROGRAMMING

15L@203-009 MINGW64 ~/Desktop
\$

Dashboard > example2 > Configuration

With Ant ?

Configure

General

Source Code Management

Triggers

Environment

Build Steps

Post-build Actions

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

Execute shell ?

Command

See the list of available environment variables

cd/Desktop
javac example2.java
java example2

Advanced

Add build step

Post-build Actions

Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.

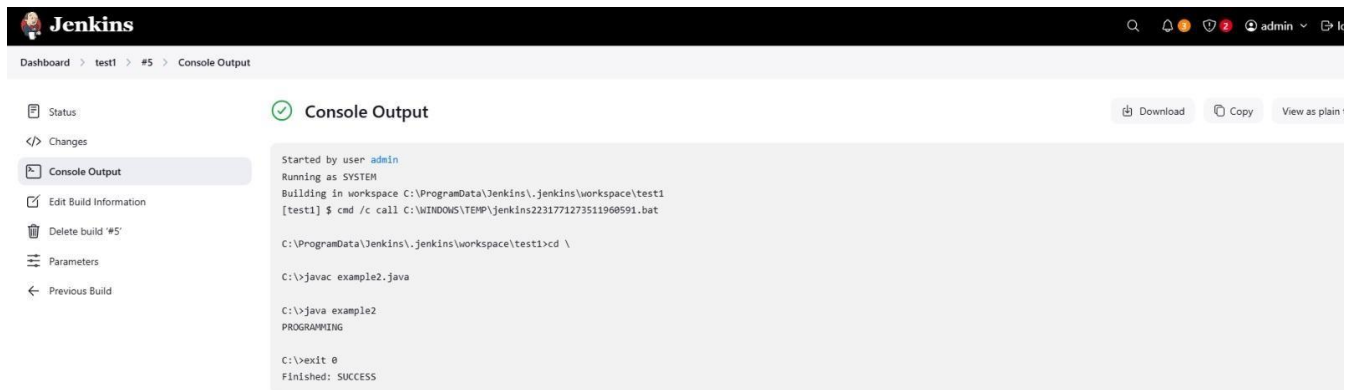
Add post-build action

Save Apply

REST API Jenkins 2.432.1

Software Engineering & Project Management Lab Experiment No: - 04

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



The screenshot shows the Jenkins web interface for build #5 of job 'test1'. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #5, Parameters, and Previous Build. The main area displays the console output for build #5, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, the compilation of 'example2.java', the execution of 'java example2', and the successful completion of the build.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins2231771273511960591.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>cd \

C:\>javac example2.java

C:\>java example2
PROGRAMMING

C:\>exit 0
Finished: SUCCESS
```



The screenshot shows the Jenkins web interface for build #4 of job 'test1'. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #4, Parameters, Previous Build, and Next Build. The main area displays the console output for build #4, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, the setting of the environment variable 'c=1+2', the execution of 'echo "Your Name is 3"', and the successful completion of the build.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins11493019808206271570.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>set /a c=1+2

C:\ProgramData\Jenkins\jenkins\workspace\test1>echo "Your Name is 3"
"Your Name is 3"

C:\ProgramData\Jenkins\jenkins\workspace\test1>exit 0
Finished: SUCCESS
```



The screenshot shows the Jenkins web interface for build #3 of job 'test1'. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build #3, Parameters, Previous Build, and Next Build. The main area displays the console output for build #3, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, the setting of the environment variable 'c=12+34', the execution of 'echo "Your Name is 12+34"', and the successful completion of the build.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins9536516287865739292.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>set c=12+34

C:\ProgramData\Jenkins\jenkins\workspace\test1>echo "Your Name is 12+34"
"Your Name is 12+34"

C:\ProgramData\Jenkins\jenkins\workspace\test1>exit 0
Finished: SUCCESS
```

Software Engineering & Project Management Lab Experiment No: - 04

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

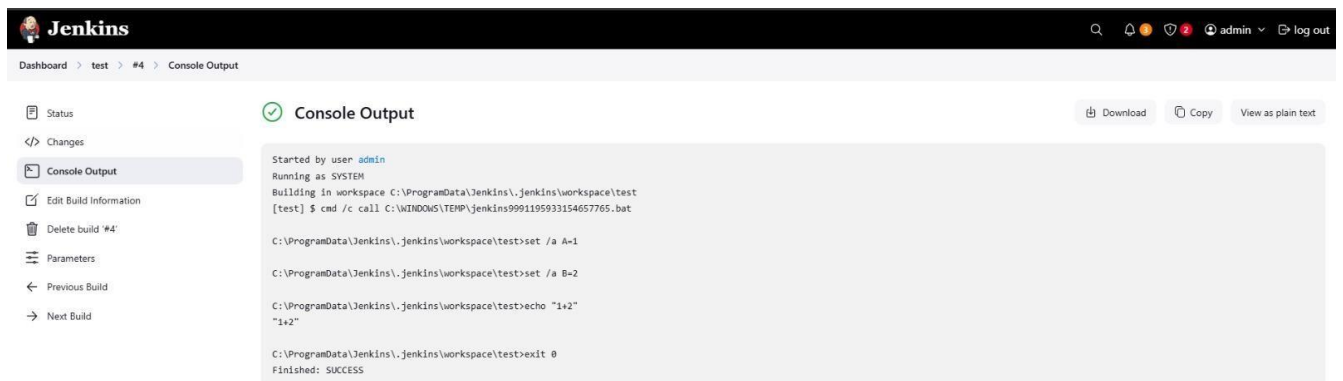


The screenshot shows the Jenkins web interface for build #2 of the 'test1' job. The left sidebar contains a list of actions: Status, Changes, Console Output (selected), Edit Build Information, Delete build #2, Parameters, Previous Build, and Next Build. The main area displays the console output for build #2, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, an echo command printing 'Your Name Is Sachin', and a successful exit.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins3591631458186967559.bat

C:\ProgramData\Jenkins\jenkins\workspace\test1>echo "Your Name Is Sachin"
"Your Name Is Sachin"

C:\ProgramData\Jenkins\jenkins\workspace\test1>exit 0
Finished: SUCCESS
```



The screenshot shows the Jenkins web interface for build #4 of the 'test' job. The left sidebar contains a list of actions: Status, Changes, Console Output (selected), Edit Build Information, Delete build #4, Parameters, Previous Build, and Next Build. The main area displays the console output for build #4, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, two 'set' commands for variables A and B, an echo command printing '1+2', and a successful exit.

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins9991195933154657765.bat

C:\ProgramData\Jenkins\jenkins\workspace\test>set /a A=1
C:\ProgramData\Jenkins\jenkins\workspace\test>set /a B=2

C:\ProgramData\Jenkins\jenkins\workspace\test>echo "1+2"
"1+2"

C:\ProgramData\Jenkins\jenkins\workspace\test>exit 0
Finished: SUCCESS
```



The screenshot shows the Jenkins web interface for build #3 of the 'test' job. The left sidebar contains a list of actions: Status, Changes, Console Output (selected), Edit Build Information, Delete build #3, Parameters, Previous Build, and Next Build. The main area displays the console output for build #3, which was started by user 'admin' and ran as 'SYSTEM'. The output shows the build directory, the execution of a batch file, an echo command printing 'ABC and DEF', and a successful exit.

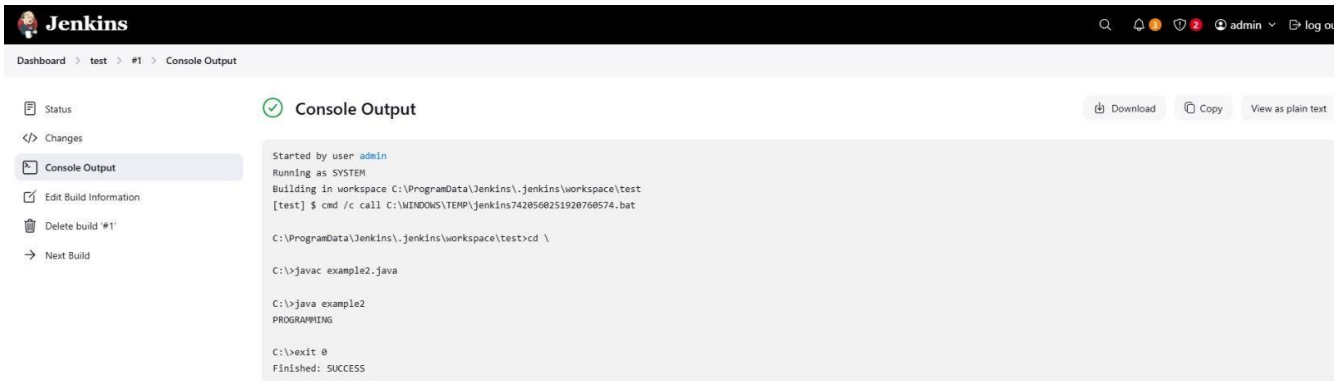
```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2368247137534955462.bat

C:\ProgramData\Jenkins\jenkins\workspace\test>echo "ABC and DEF"
"ABC and DEF"

C:\ProgramData\Jenkins\jenkins\workspace\test>exit 0
Finished: SUCCESS
```


Software Engineering & Project Management Lab Experiment No: - 04

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



Conclusion: Thus, we have successfully installed and configured Jenkins with Maven/Ant/Gradle to setup a build Job and learnt about the implementation of Jenkins in open source continuous integration.