Software Engineering and Project Management Lab Experiment No. 4

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** o Install Java (JDK) as a prerequisite. o 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** o Install **Maven**, **Ant**, or **Gradle** depending on project requirements.
   - Configure Jenkins to recognize the installed build tool.
3. **Creating a Build Job in Jenkins** o 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:

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

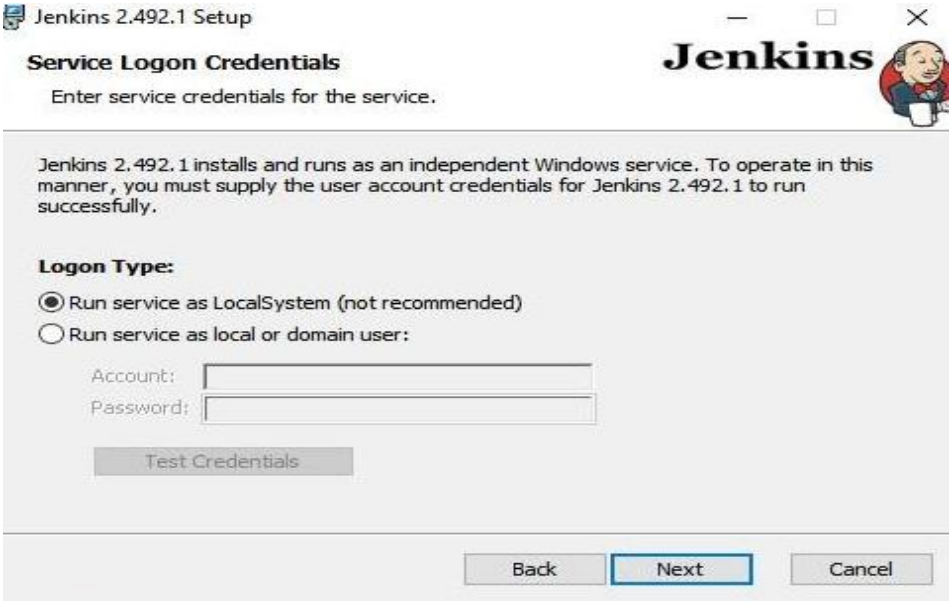
Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh 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.

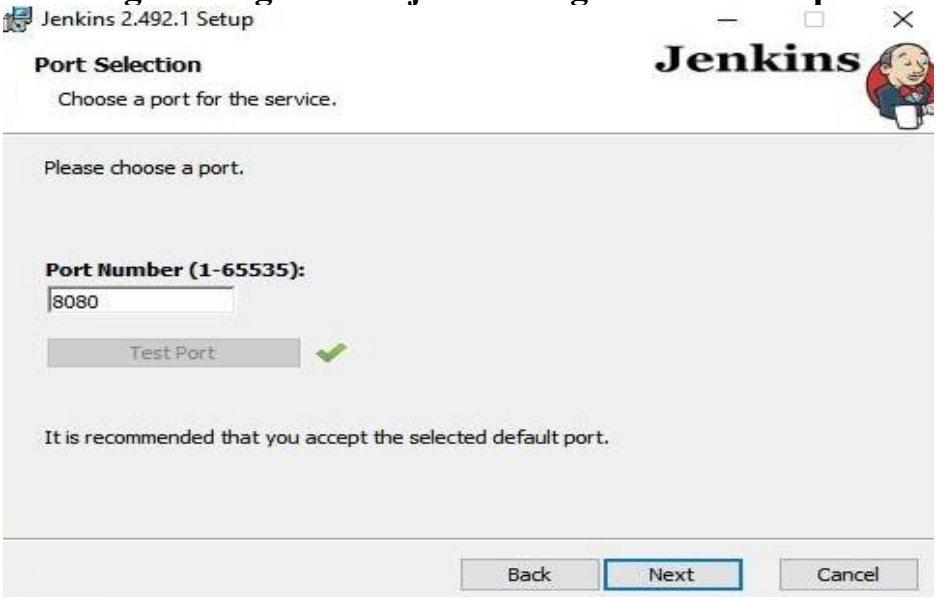
Open the installed .exe setup

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

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

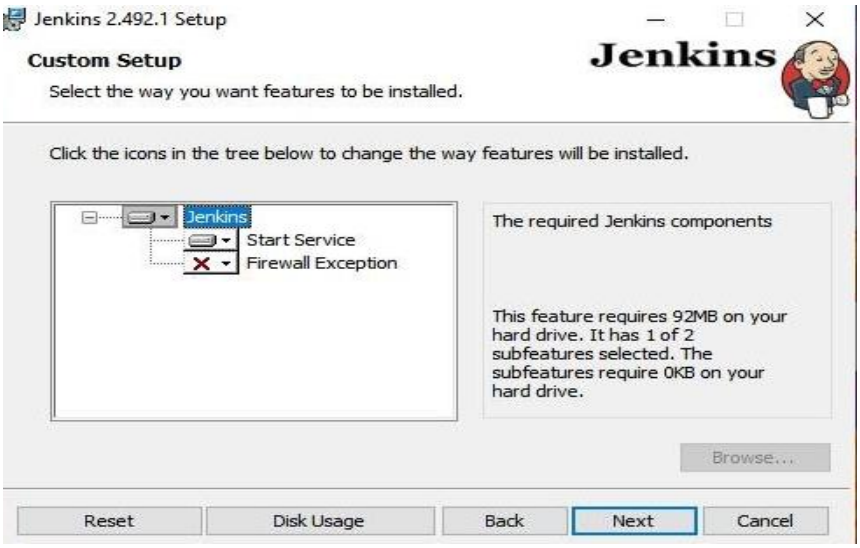


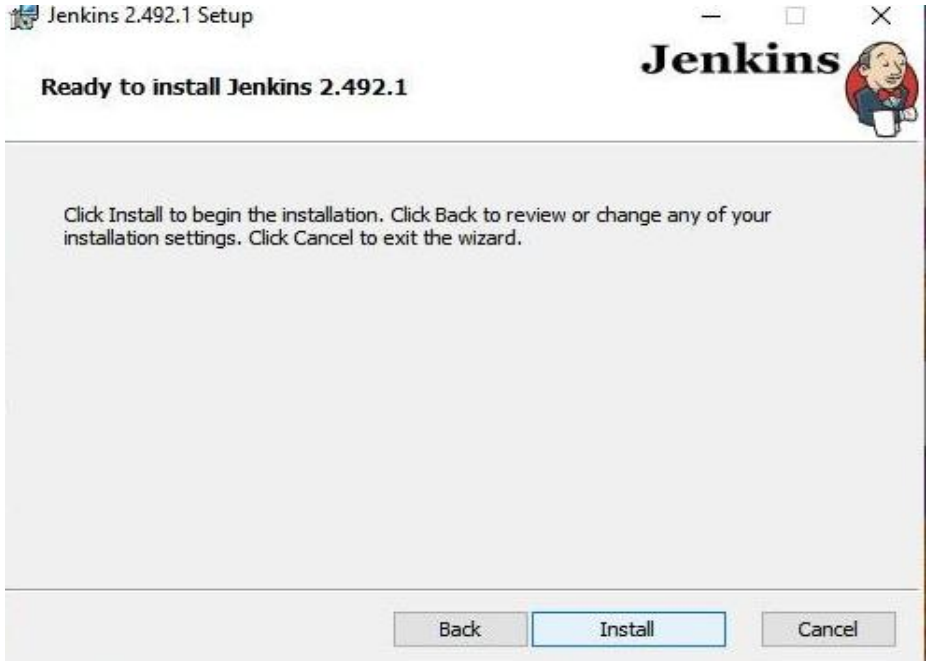
Step 4: Select the port 8080 and click Test Port button. The green tick will appear after which you can proceed to Next.



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

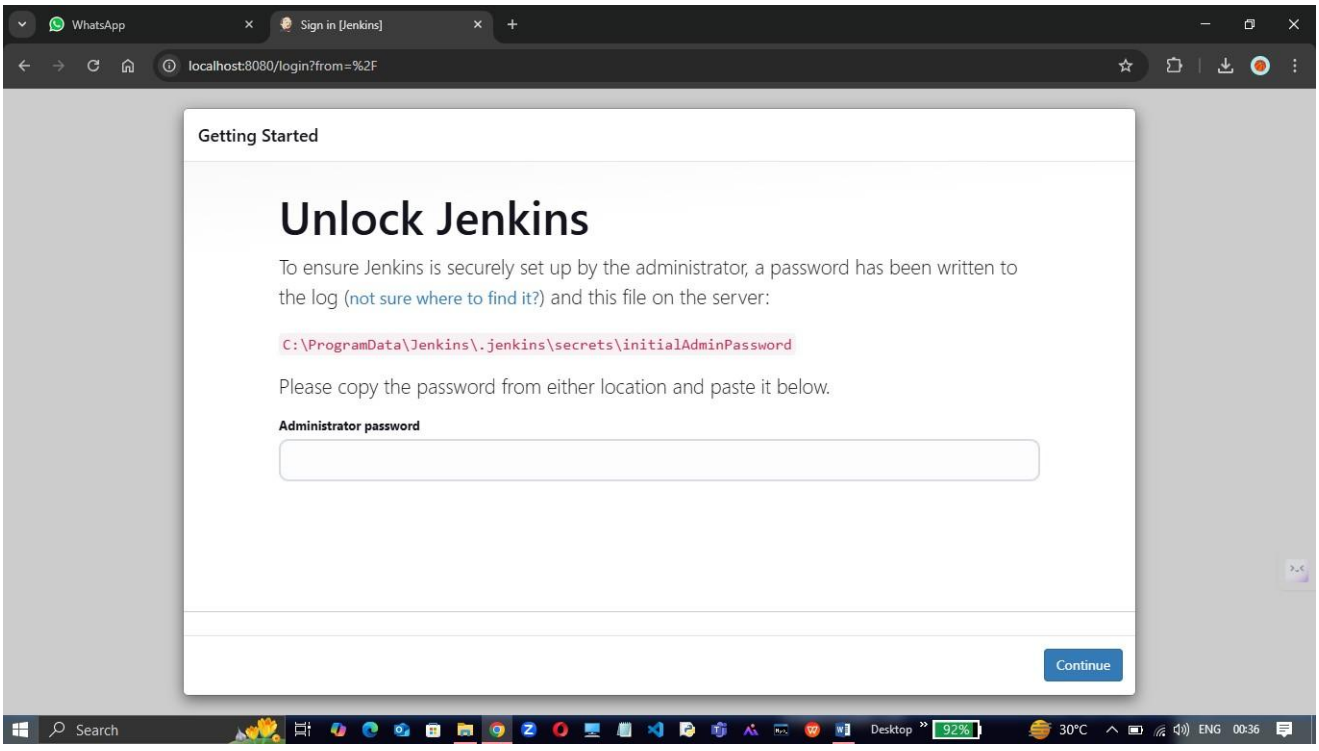
Proceed to Next


Jenkins 2.492.1 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
The required Jenkins components
This feature requires 92MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.
Browse...
Reset
Disk Usage
Back
Next
Cancel

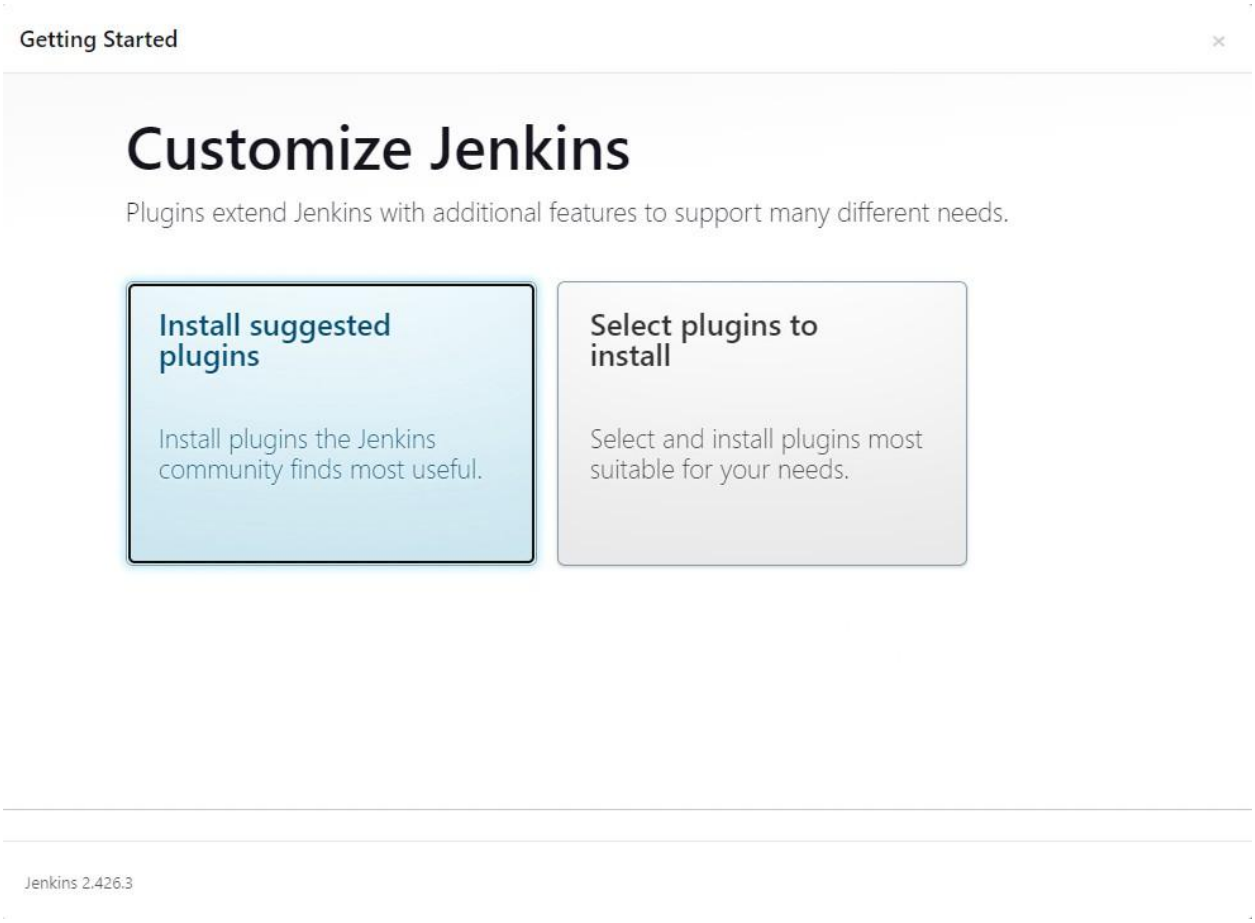

Jenkins 2.492.1 Setup
Ready to install Jenkins 2.492.1
Jenkins
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

On clicking 'Install', installation is finished.

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.

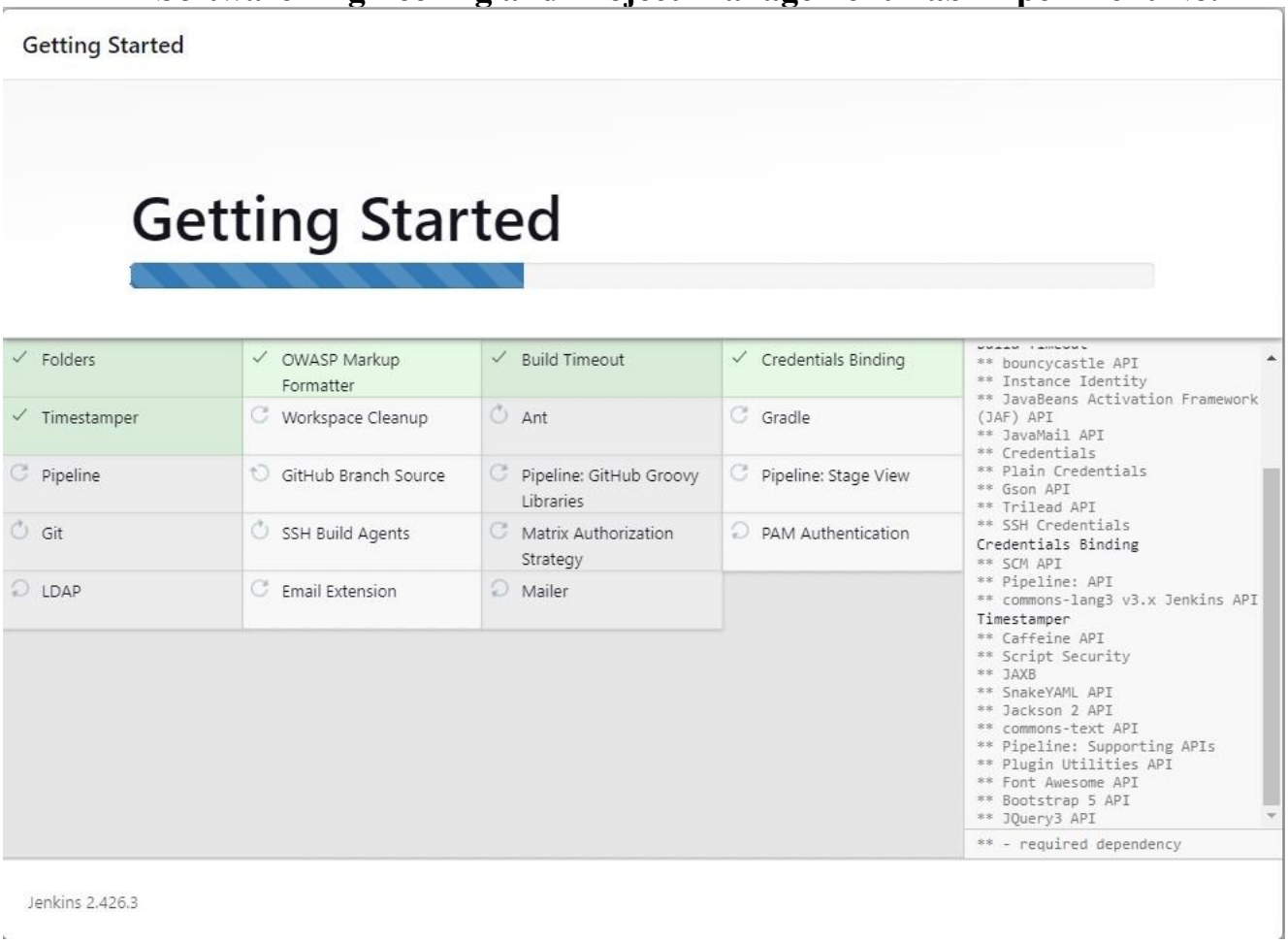


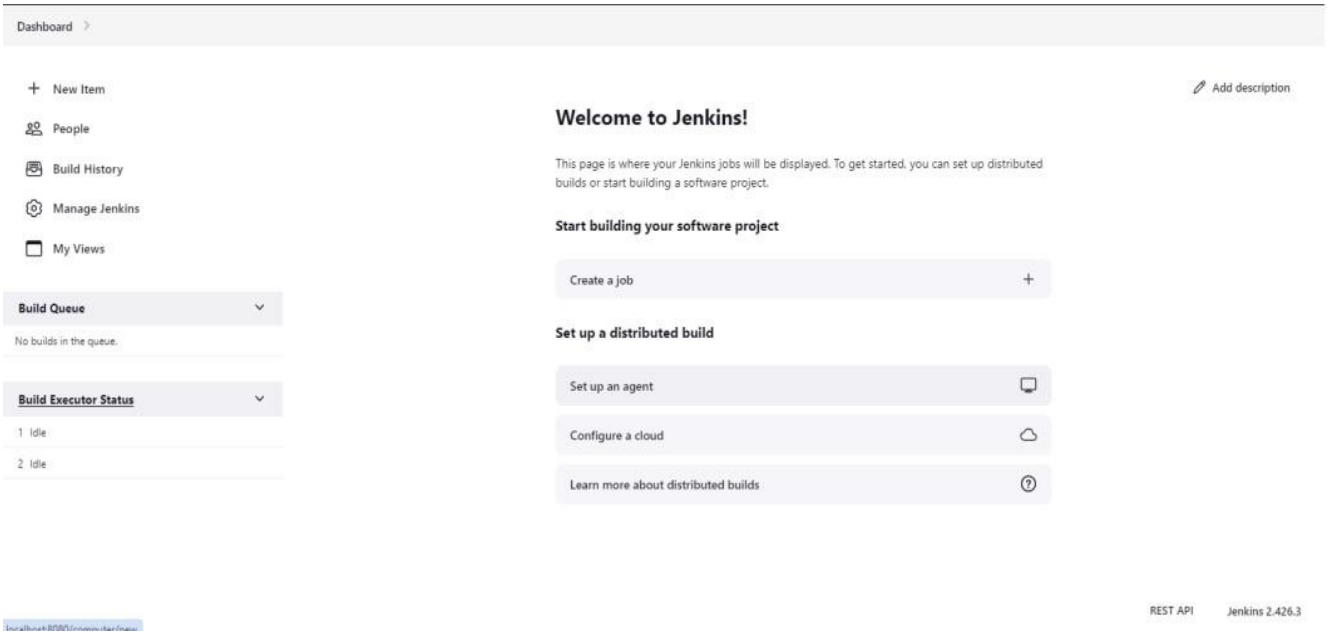
Step 7: On entering the password, you can continue to choose "Install Suggested Plugins"

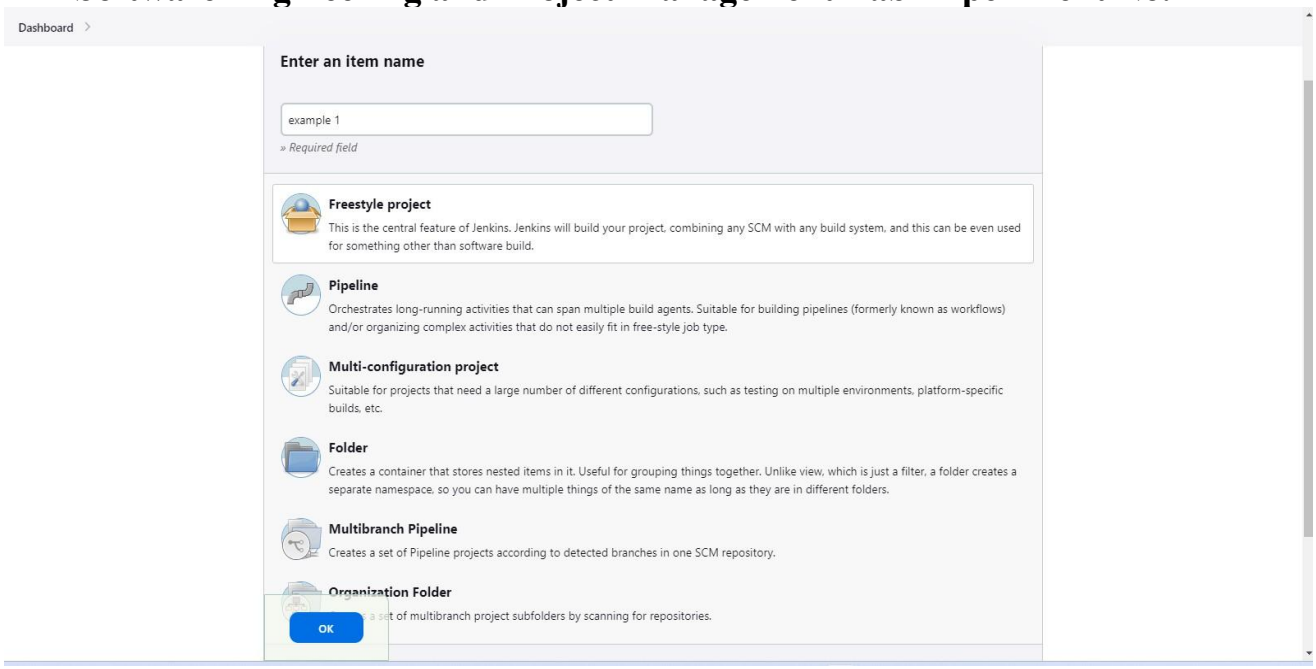


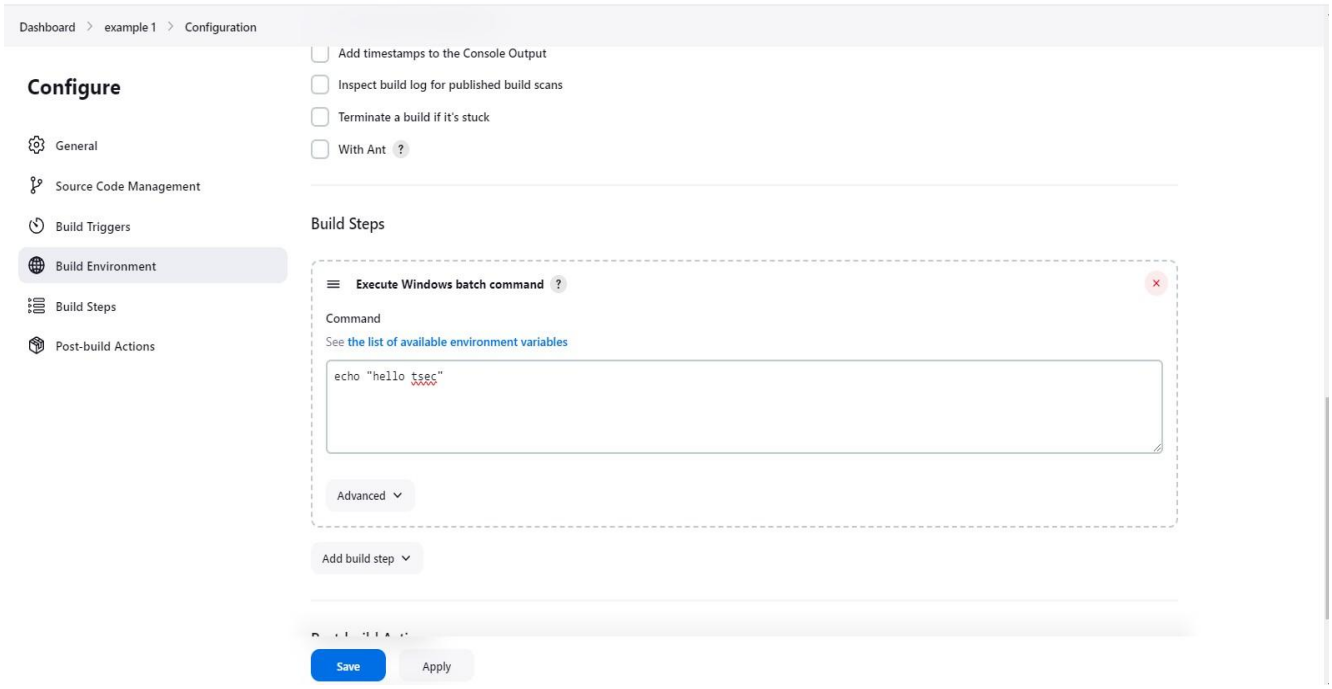
Once plugins are installed, click on next and specify the admin details along with the new password for Jenkins admin and click on finish to complete the installation.

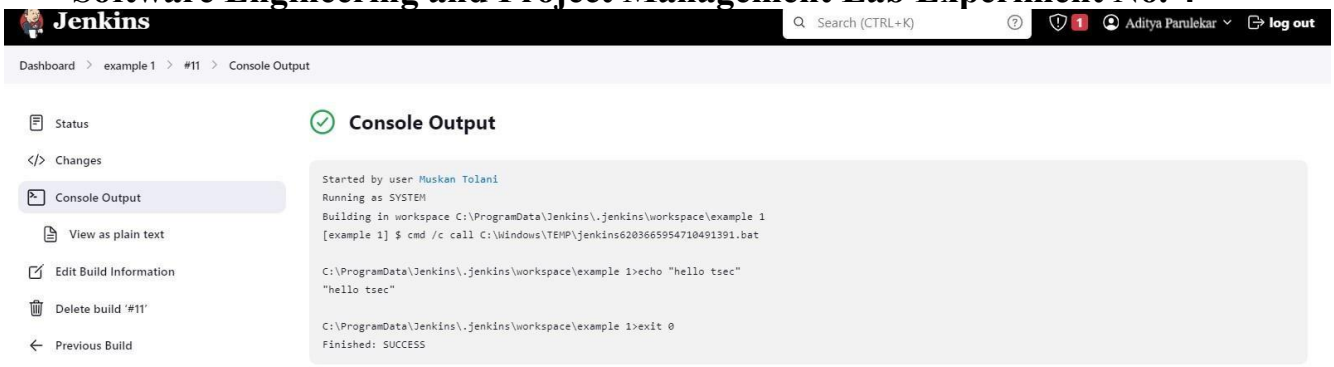
After filling the details, click on Save & Continue, you will be redirected to the dashboard.

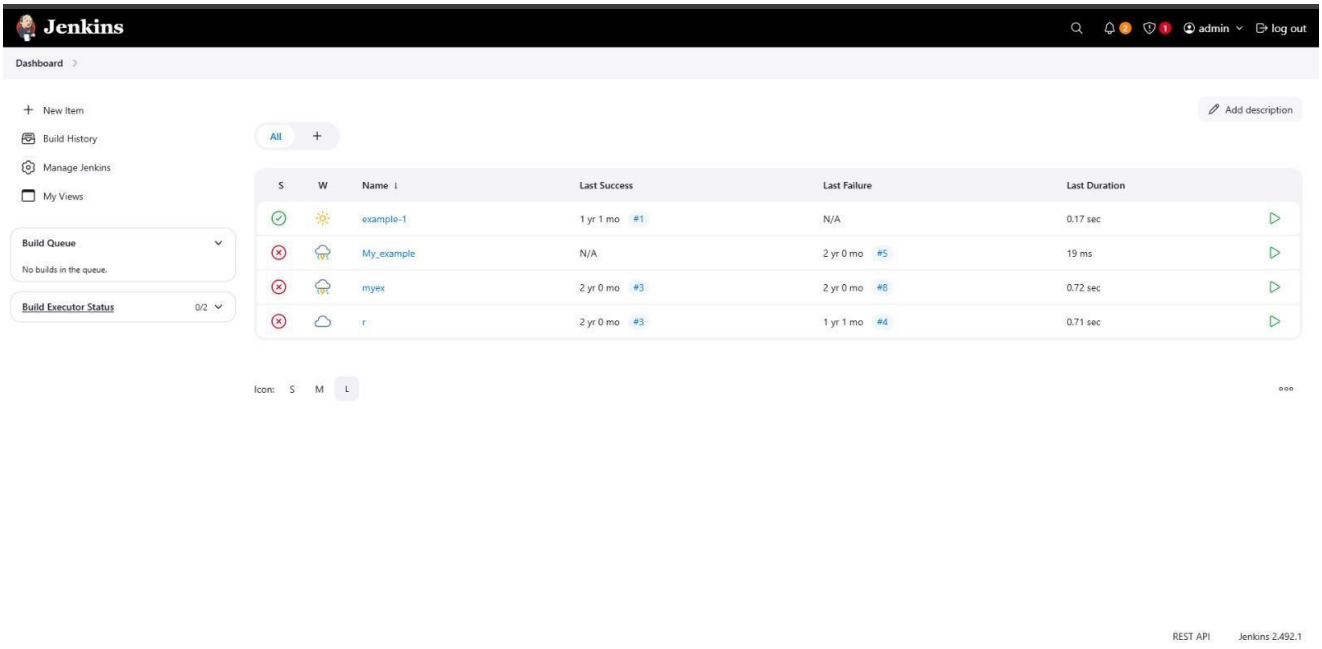

Getting Started
Getting Started
Folders
OWASP Markup Formatter
Build Timeout
Credentials Binding
Timestamper
Workspace Cleanup
Ant
Gradle
Pipeline
GitHub Branch Source
Pipeline: GitHub Groovy Libraries
Pipeline: Stage View
Git
SSH Build Agents
Matrix Authorization Strategy
PAM Authentication
LDAP
Email Extension
Mailer
** bouncycastle API
** Instance Identity
** JavaBeans Activation Framework (JAF) API
** JavaMail API
** Credentials
** Plain Credentials
** Gson API
** Trilead API
** SSH Credentials
Credentials Binding
** SCM API
** Pipeline: API
** commons-lang3 v3.x Jenkins API
Timestamper
** Caffeine API
** Script Security
** JAXB
** SnakeYAML API
** Jackson 2 API
** commons-text API
** Pipeline: Supporting APIs
** Plugin Utilities API
** Font Awesome API
** Bootstrap 5 API
** JQuery3 API
** - required dependency
Jenkins 2.426.3

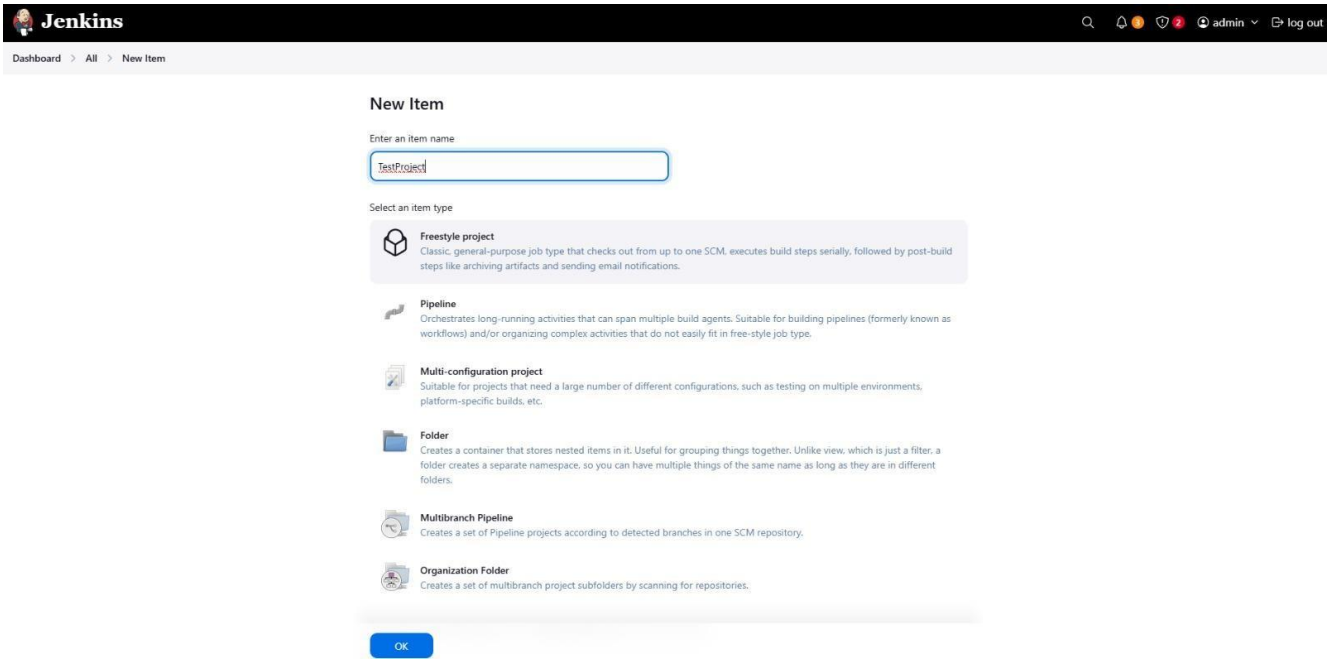

Dashboard
New Item
People
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
1 Idle
2 Idle
Add description
Welcome to Jenkins!
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software 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

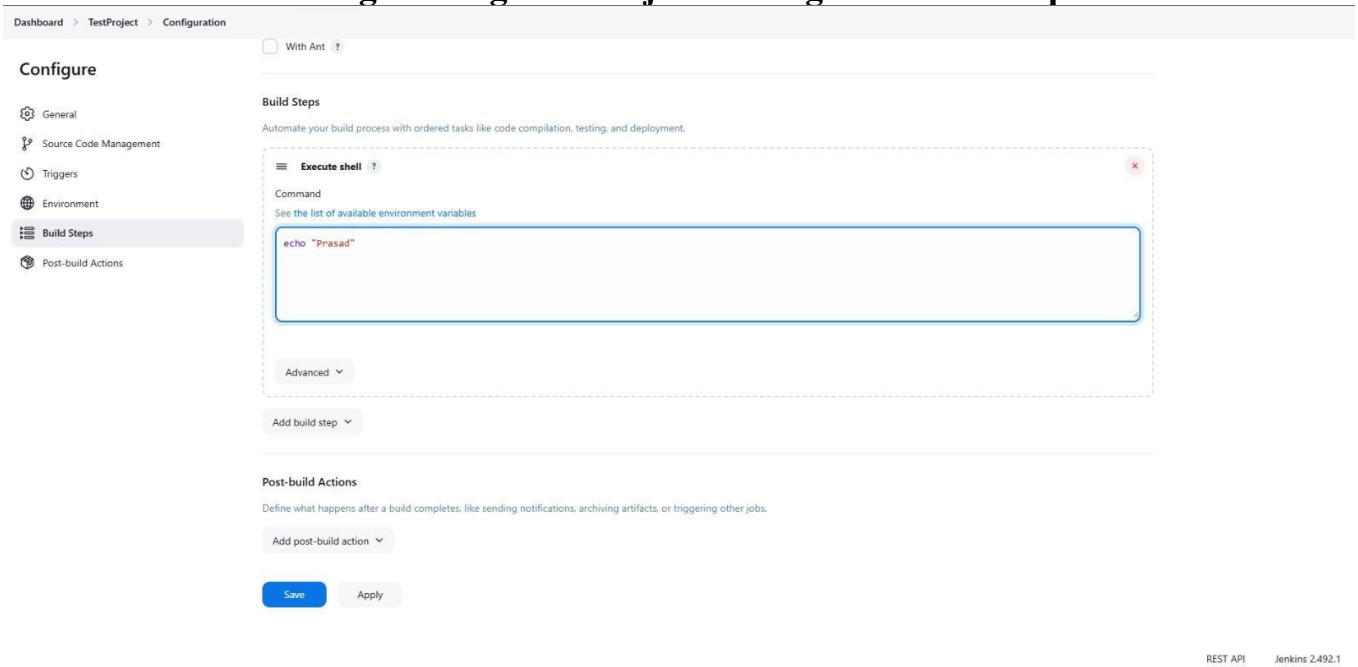

Dashboard
Enter an item name
example 1
» Required field
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
Pipeline
Orchestrates long-running activities 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
OK

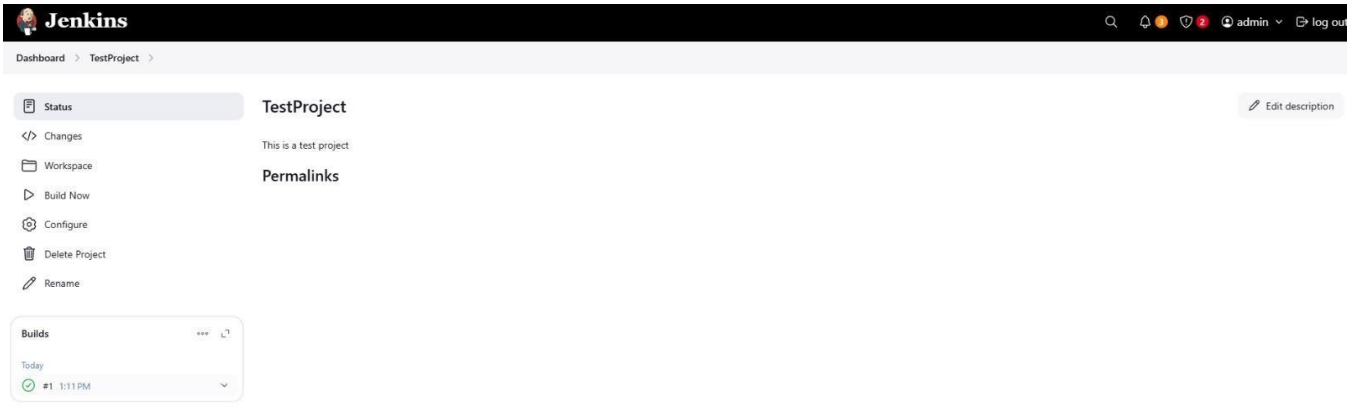

Dashboard
example 1
Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Add timestamps to the Console Output
Inspect build log for published build scans
Terminate a build if it's stuck
With Ant
Build Steps
Execute Windows batch command
Command
See the list of available environment variables
echo "hello tsec"
Advanced
Add build step
Save
Apply


Jenkins
Search (CTRL+K)
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


Jenkins
admin
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status 0/2
Add description
All
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 #6 0.72 sec
r 2 yr 0 mo #3 1 yr 1 mo #4 0.71 sec
Icon: S M L
REST API
Jenkins 2.492.1

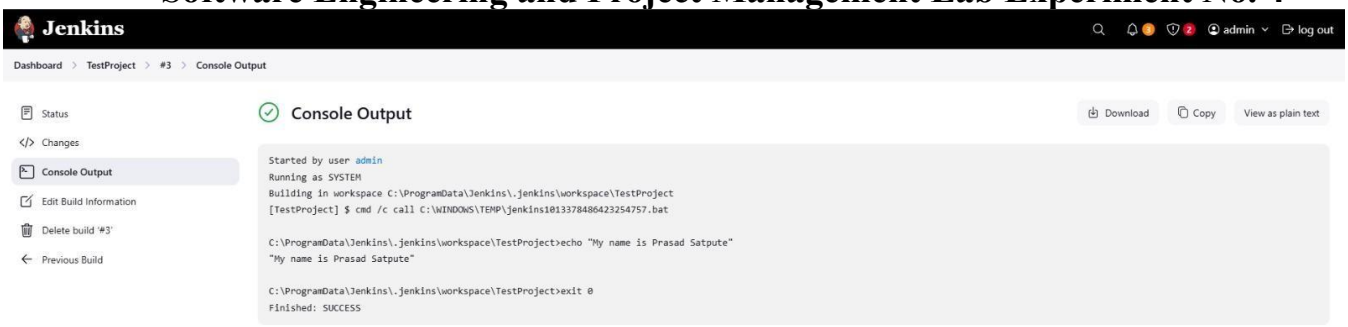

Jenkins
admin
log out
Dashboard > All > New Item
New Item
Enter an item name
TestProject
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
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

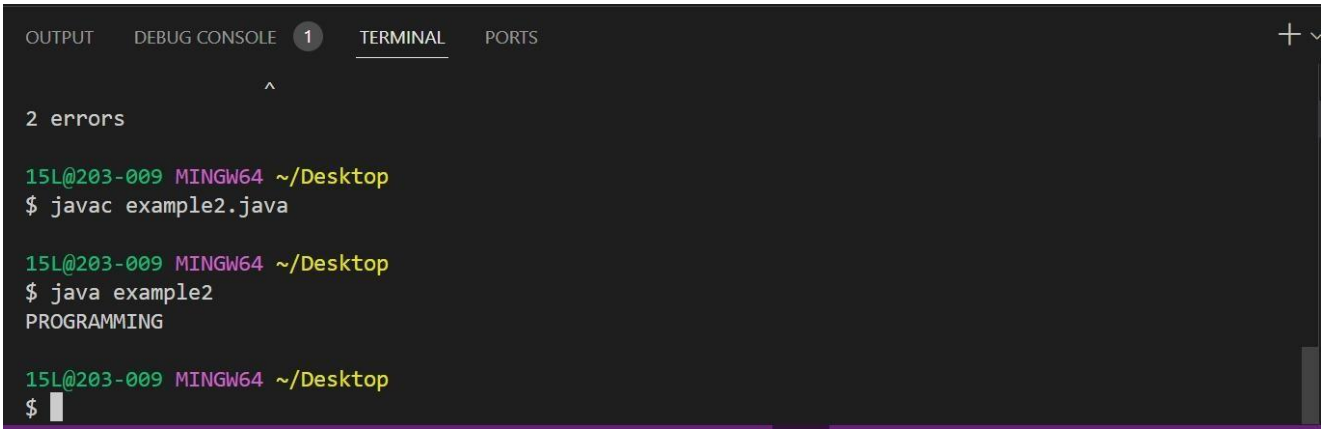

Dashboard > TestProject > Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
With Ant
Build Steps
Automate your build process with ordered tasks like code compilation, testing, and deployment.
Execute shell
Command
See the list of available environment 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

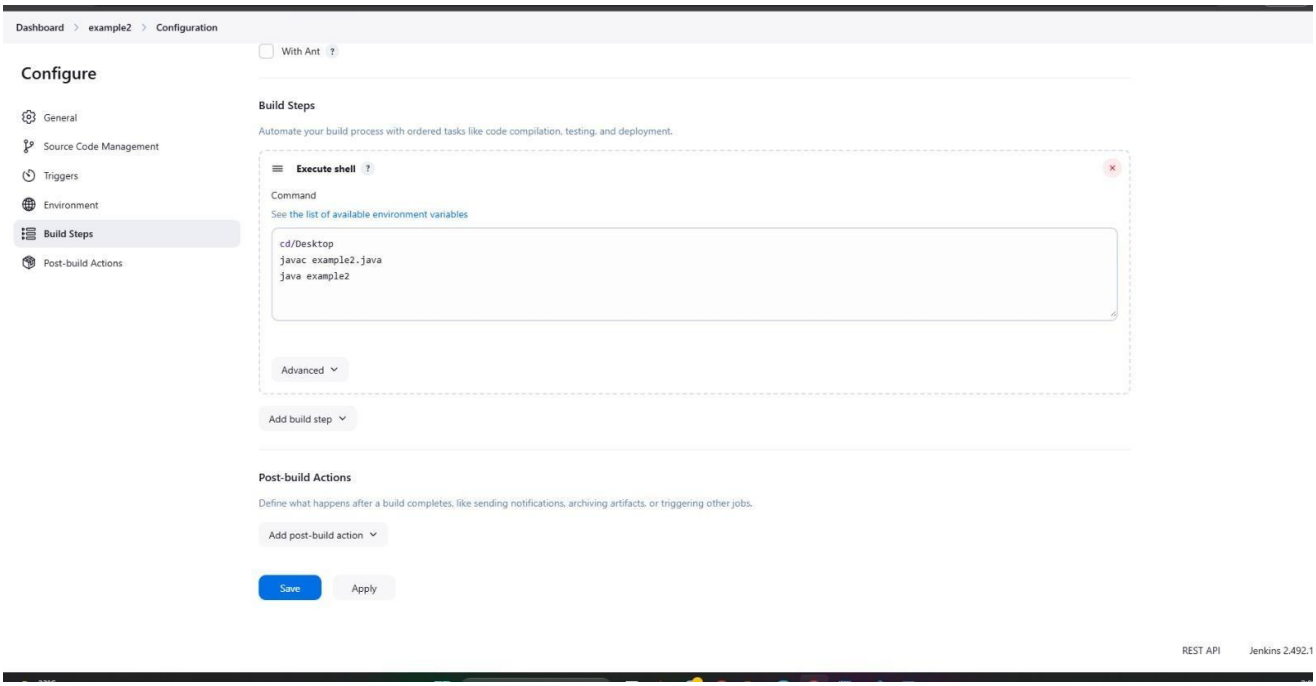

Jenkins
admin
log out
Dashboard > TestProject >
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Builds
Today
#1 1:11 PM
TestProject
Edit description
This is a test project
Permalinks

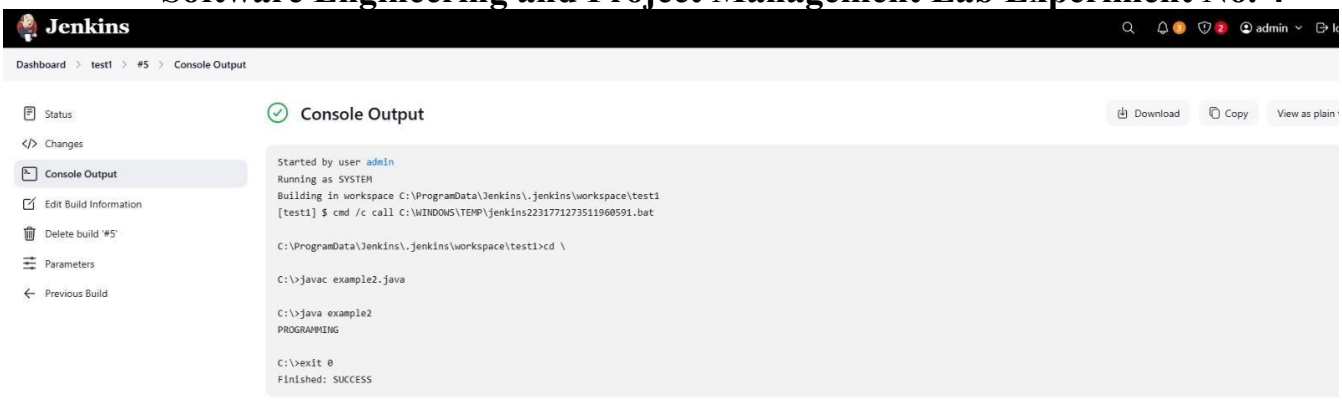

Jenkins
admin
log out
Dashboard > TestProject > #1 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Console Output
Download
Copy
View as plain text
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\TestProject
[TestProject] $ C:\Windows\system32\cmd.exe -xe C:\WINDOWS\TEMP\jenkins10724866366536973722.sh
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.
C:\ProgramData\Jenkins\.jenkins\workspace\TestProject>Finished: SUCCESS


Code
bash
15L@203-009 MINGW64 ~
$ cat > example1.sh
#!/bin/bash
name=$1
Address=$2
echo "Hello $name ..your address is $Address"
[1]+ Stopped cat > example1.sh
15L@203-009 MINGW64 ~
$


Jenkins
admin
log out
Dashboard > TestProject > #3 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#3'
Previous Build
Console Output
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\jenkins10133784864232547 57.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

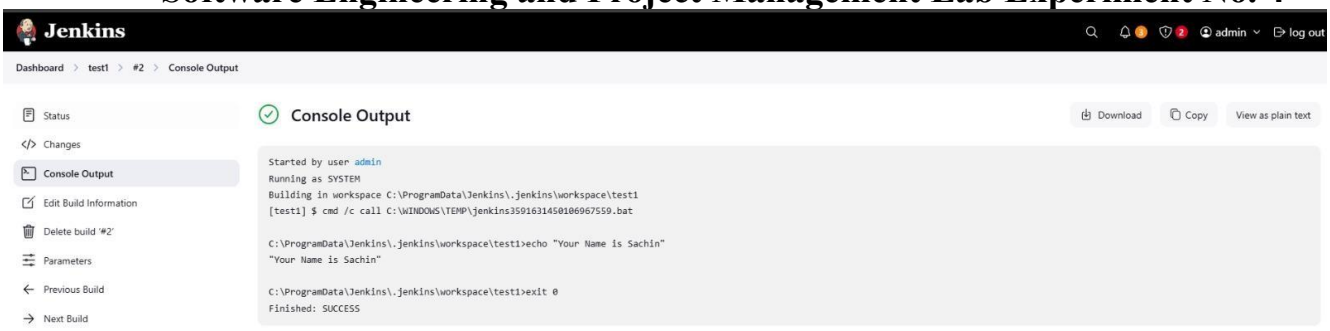

OUTPUT
DEBUG CONSOLE 1
TERMINAL
PORTS
^
2 errors
15L@203-009 MINGW64 ~/Desktop
$ javac example2.java
15L@203-009 MINGW64 ~/Desktop
$ java example2
PROGRAMMING
15L@203-009 MINGW64 ~/Desktop
$

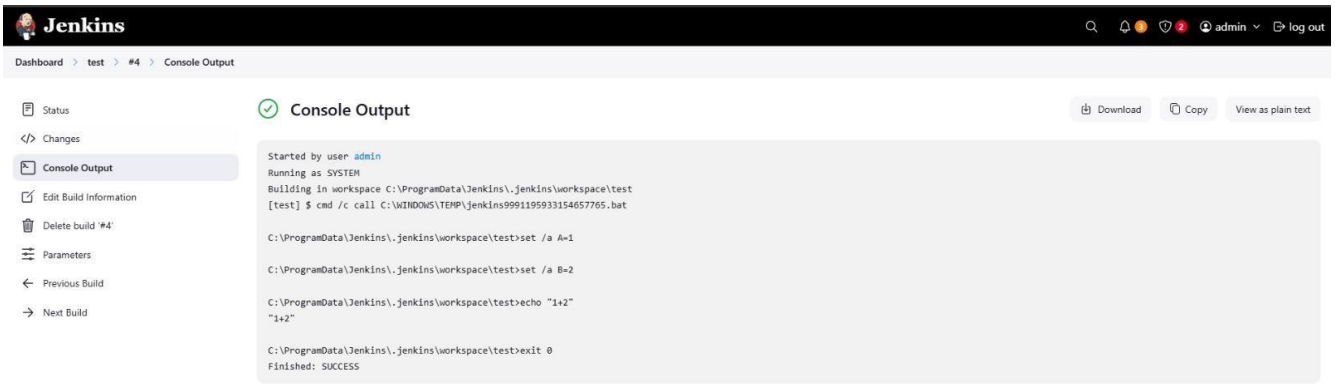

Dashboard > example2 > Configuration
Configure
General
Source Code Management
Triggers
Environment
Build Steps
Post-build Actions
With Ant
Build Steps
Automate your build process with ordered tasks like code compilation, testing, and deployment.
Execute shell
Command
See the list of available environment 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 2.492.1


Jenkins
admin
Dashboard > test1 > #5 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#5'
Parameters
Previous Build
Console Output
Download
Copy
View as plain
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

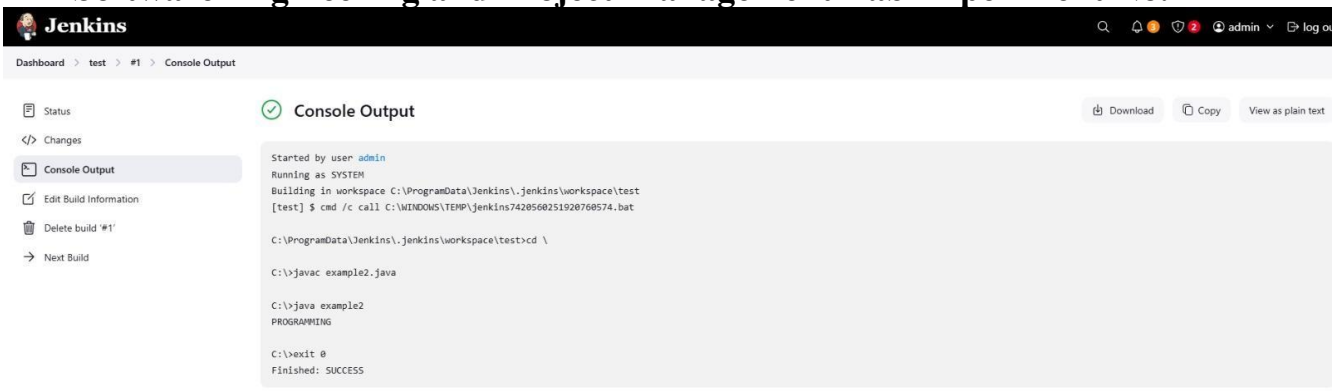

Jenkins
admin
log out
Dashboard > test1 > #4 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#4'
Parameters
Previous Build
Next Build
Console Output
Download
Copy
View as plain text
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test1
[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


Jenkins
admin
log out
Dashboard > test1 > #3 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#3'
Parameters
Previous Build
Next Build
Console Output
Download
Copy
View as plain text
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins9536516207865739292.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


Jenkins
admin
log out
Dashboard > test1 > #2 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#2'
Parameters
Previous Build
Next Build
Console Output
Download
Copy
View as plain text
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test1
[test1] $ cmd /c call C:\WINDOWS\TEMP\jenkins3591631450106967559.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


Jenkins
admin
log out
Dashboard > test > #4 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#4'
Parameters
Previous Build
Next Build
Console Output
Download
Copy
View as plain text
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


Jenkins
admin
log out
Dashboard > test > #3 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#3'
Parameters
Previous Build
Next Build
Console Output
Download
Copy
View as plain text
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test
[test] $ cmd /c call C:\WINDOWS\TEMP\jenkins2360247137534955462.bat
C:\ProgramData\Jenkins\.jenkins\workspace\test>echo "ABC and DEF"
"ABC and DEF"
C:\ProgramData\Jenkins\.jenkins\workspace\test>exit 0
Finished: SUCCESS



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.