

Experiment 04

Aim: To understand continuous integration, install and configure Jenkins with maven/ant/gradle to set up a build job.

Theory:

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

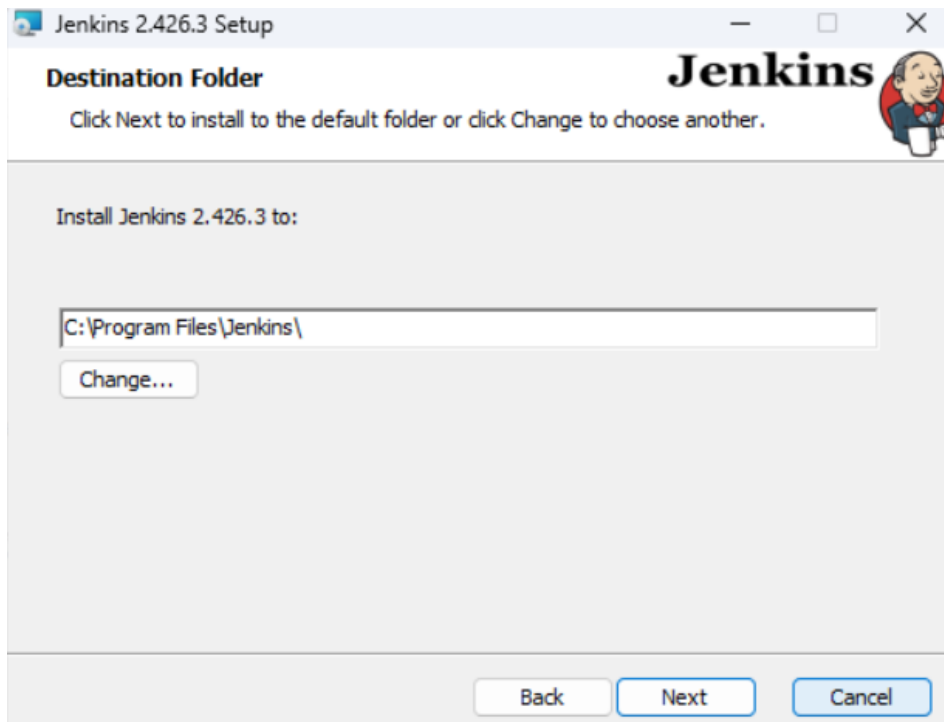
Jenkins is an open source automation tool written in Java with plugins built for Continuous Integration purpose. Jenkins is used to build and test your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh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

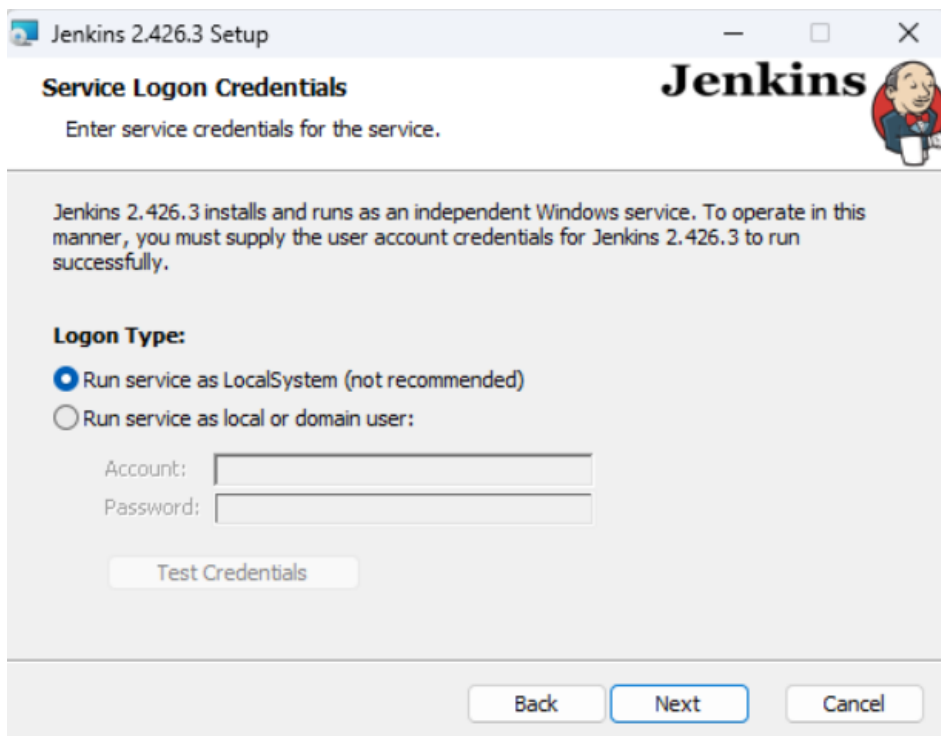
Open the installed .exe setup



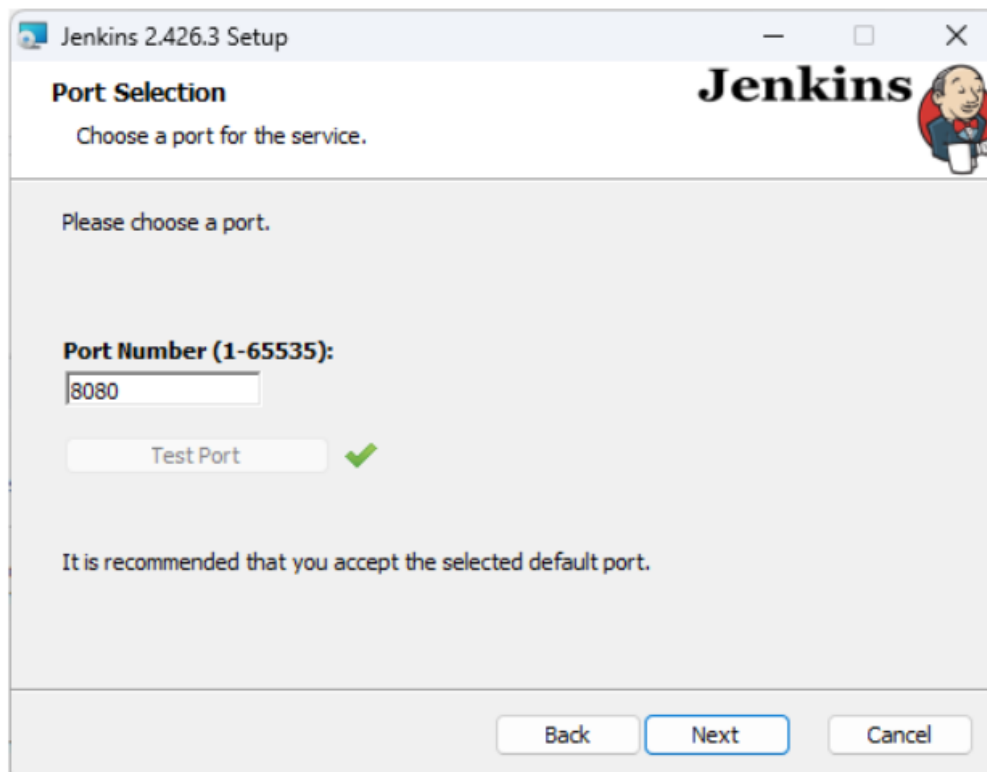
Step 2: Locate the folder where you want to install Jenkins in the location path:



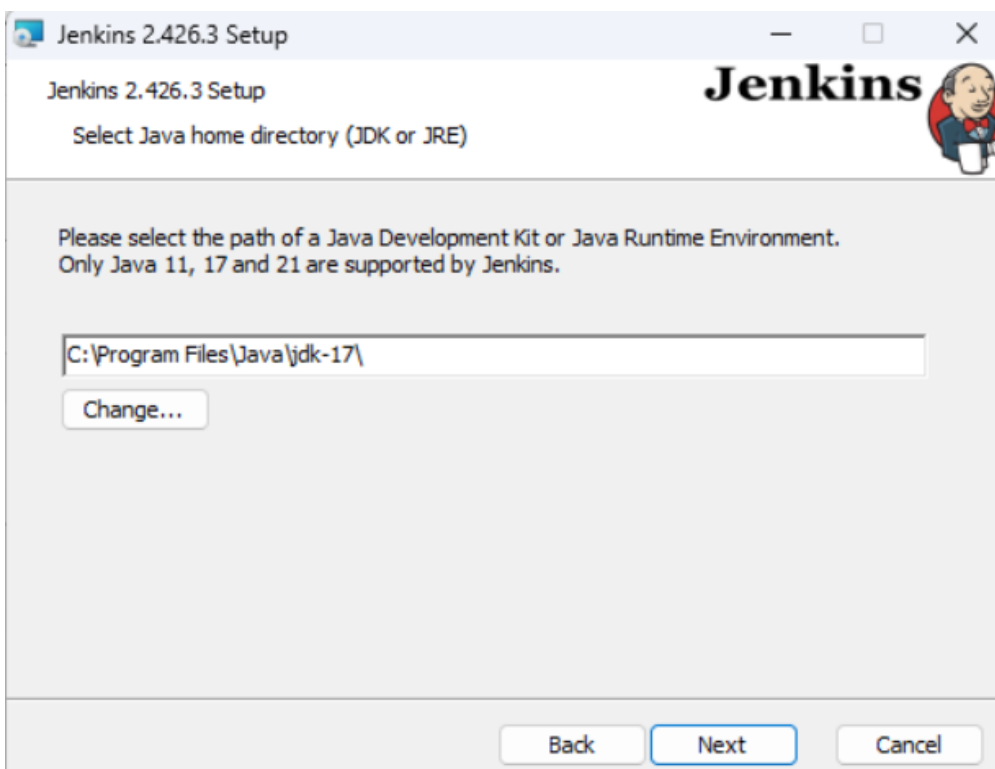
Step 3: Select service as Local System and proceed to Next.



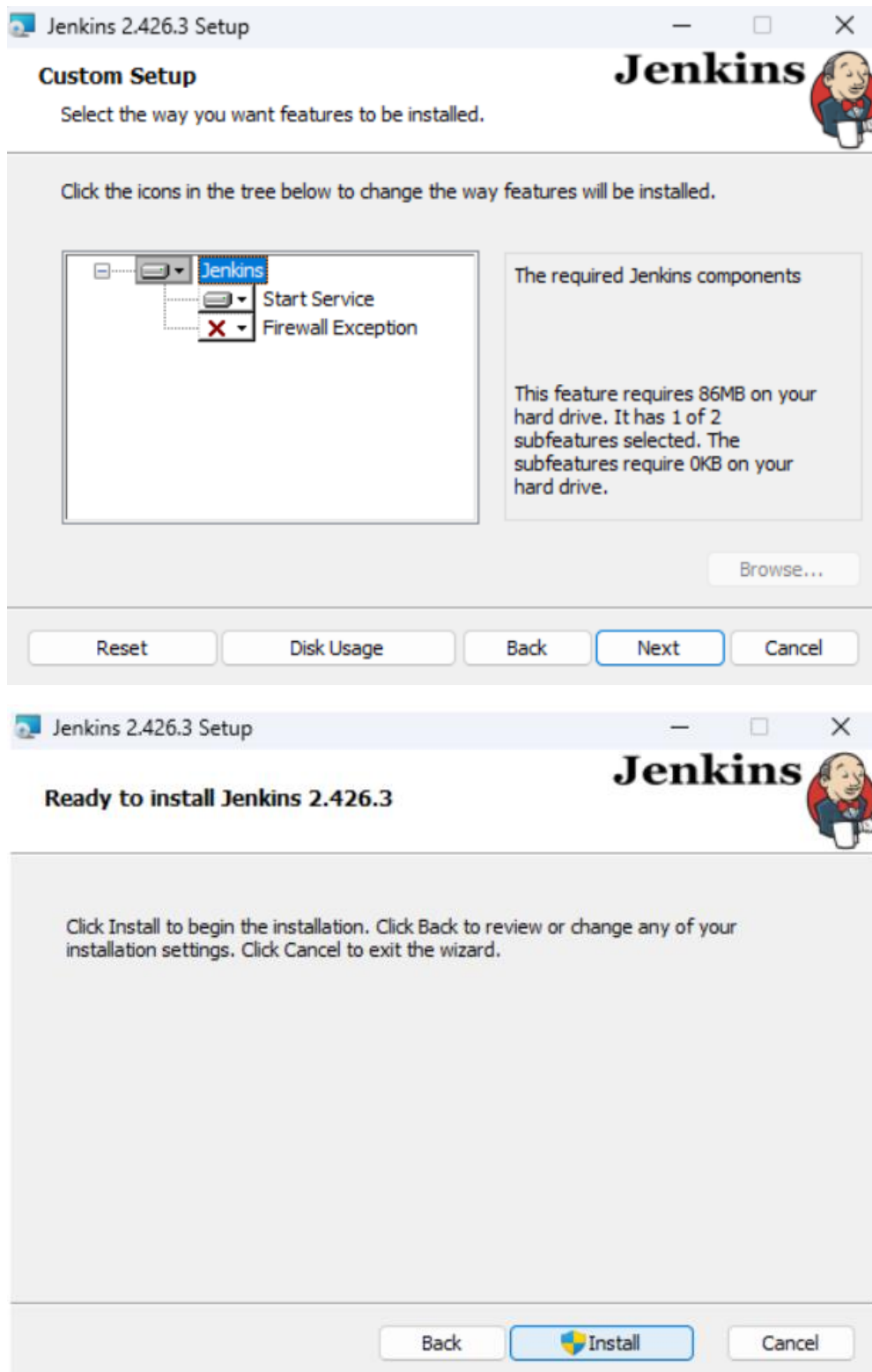
Step 4: Select the port 8080 and click Test Port button. The green tick will appear after which you can proceed to Next.



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



Proceed to Next

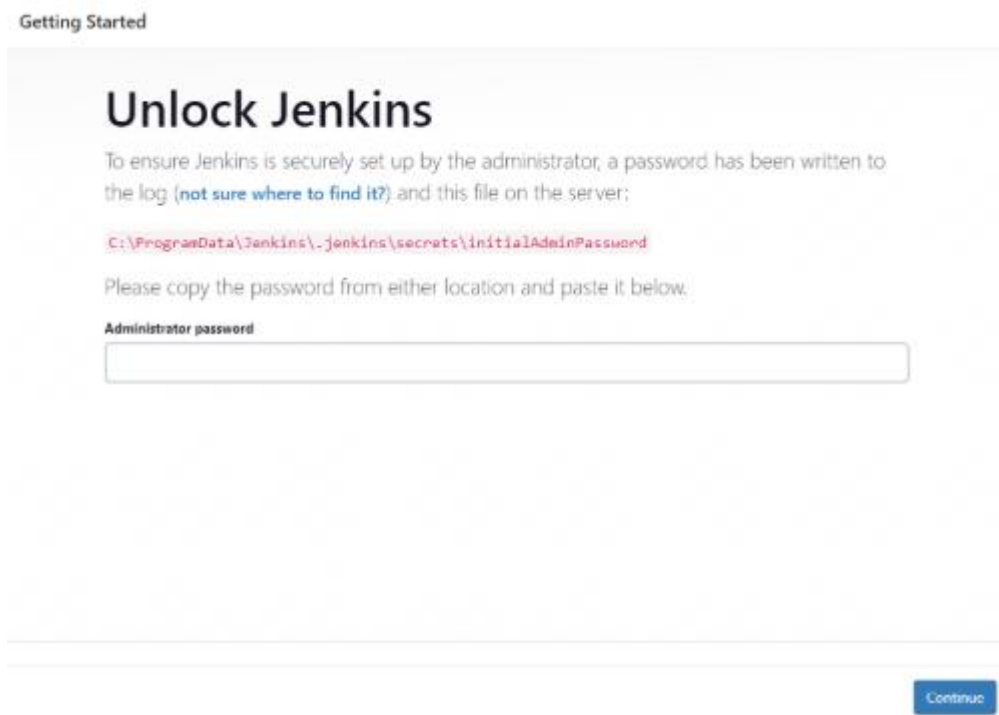


On clicking 'Install', installation is finished.

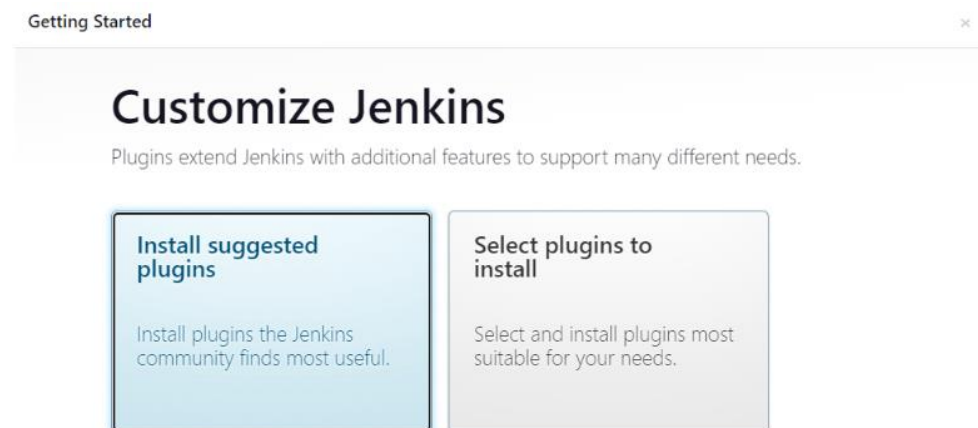


Step 6: Once Installation is done, you can test the Jenkins on <http://localhost:8080> on the browser.

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.

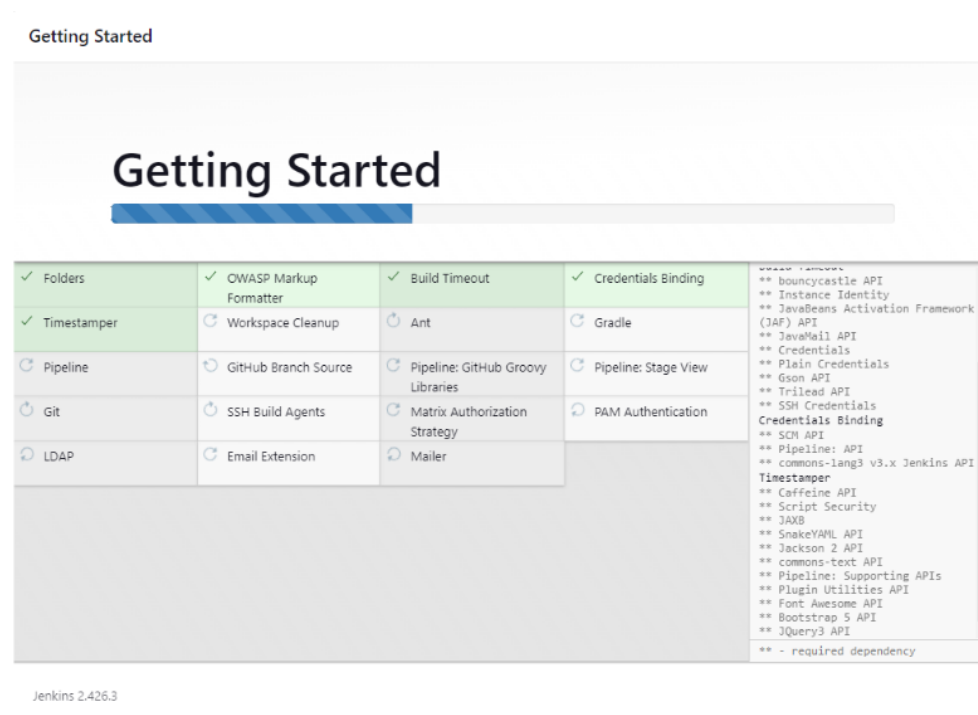


Step 7: On entering the password, you can continue to choose “Install Suggested Plugins”



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.



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

Getting Started

Create First Admin User

Username

musk

Password

Confirm password

Full name

Muskan Tolani

E-mail address

muskantolani7@gmail.com

Jenkins 2.426.3


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


Dashboard >


Enter an item name


example 1


+ Required field


**Freestyle project**
This is the control feature of Jenkins. Items will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Pipeline**
On-demand, 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**
A set of multibranch project subfolders by scanning for repositories.


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


Dashboard > example 1 > #11


 Status

 Changes

 Console Output

 Edit Build Information

 Delete build 'FIT'

 Previous Build

 **#11 (Jan 31, 2024, 9:07:13 AM)**

 Add description

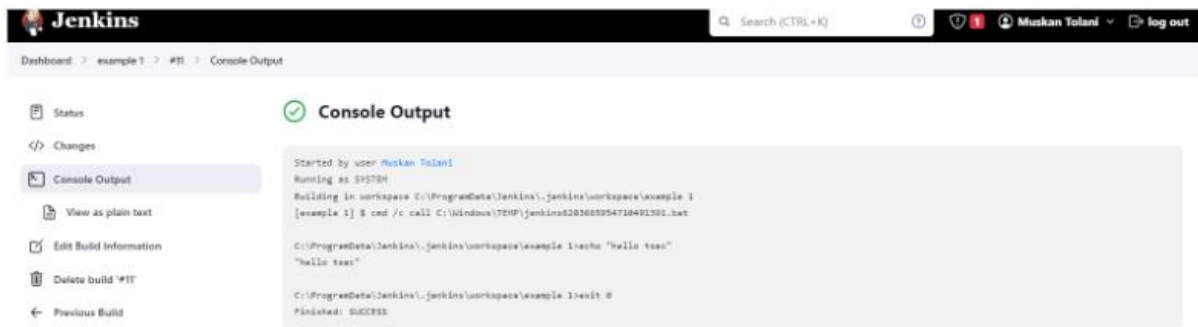
 No changes.

 Started by user [Muskan Tolani](#)

Keep this build forever

Started 34 sec ago

Task 0:11 sec



Conclusion: 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.