# RBG technologies

KNOWLEDGE CREATES UNLIMITED TECHNOLOGIES - LIKE PRIMARY COLORS

OUR ESTEEMED COURSES :

**JAVA**
**SELENIUM**
**API TESTING**
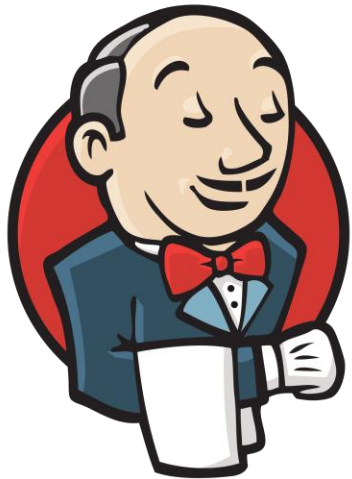**MANUAL TESTING**

100% JOB ASSISTANCE 100% JOB

SCAN ME

CONTACT US :
**+91 93986 94373**

# Getting Started with Jenkins

## A Beginner's Guide

Jenkins

# INDEX

# WELCOME TO THE JENKINS

## What is Jenkins?

- Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

- Jenkins can be installed through native system packages, Docker, or even run standalone by any machine with a Java Runtime Environment (JRE) installed.

- It's a core tool for continuous integration and continuous delivery (CI/CD) pipelines, allowing developers to automate the software delivery lifecycle, from code changes to deployment

# Installation :

## Prerequisites :

For this, you will require:

A machine with:
256 MB of RAM, although more than 2 GB is recommended
10 GB of drive space (for Jenkins and your Docker image)
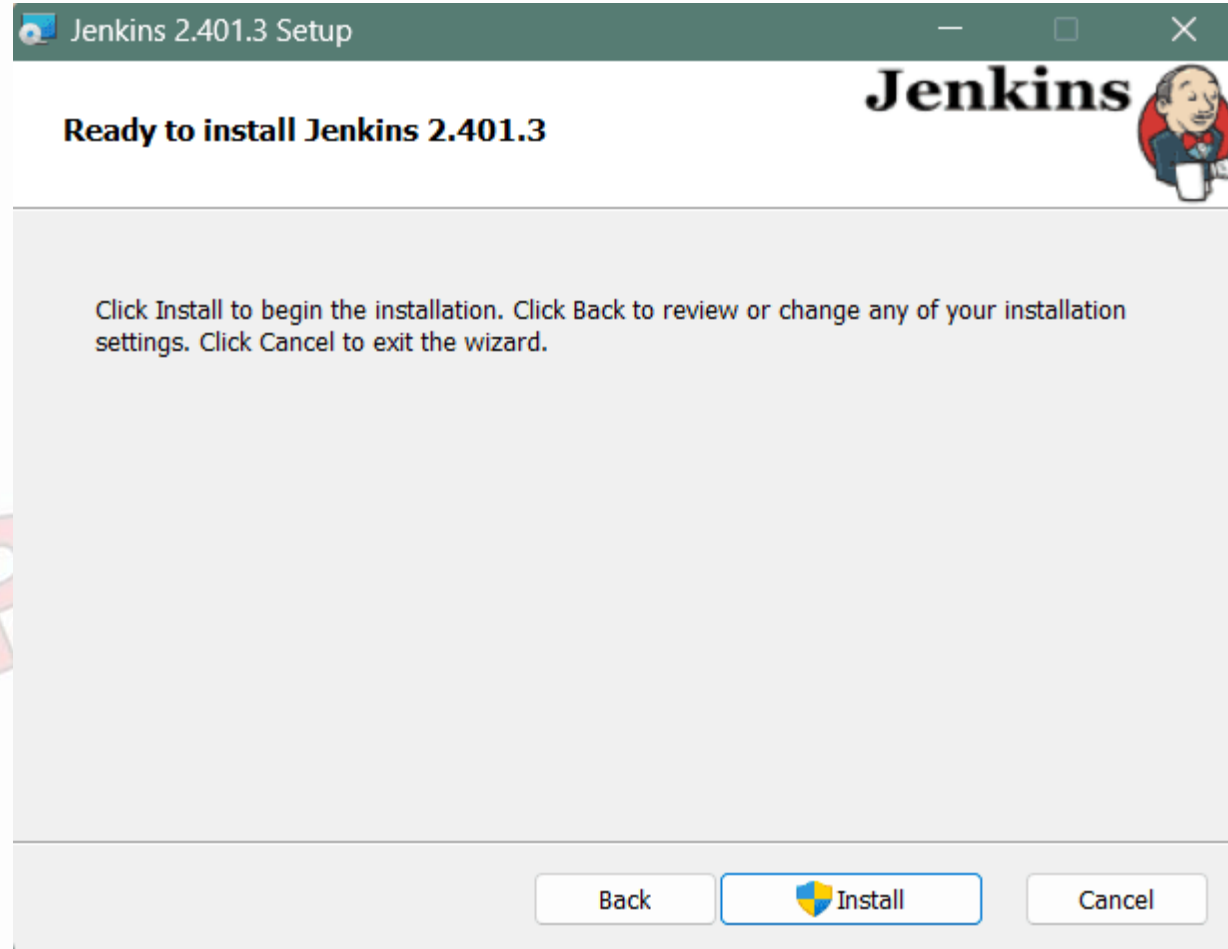
The following software installed:
Java 17 or 21

Jenkins Download Link:
https://www.jenkins.io/download/thank-you-downloading-windows-installer-stable/

## Step 1: Setup wizard

On opening the Windows Installer, an **Installation Setup Wizard** appears, Click **Next** on the Setup Wizard to start your installation.
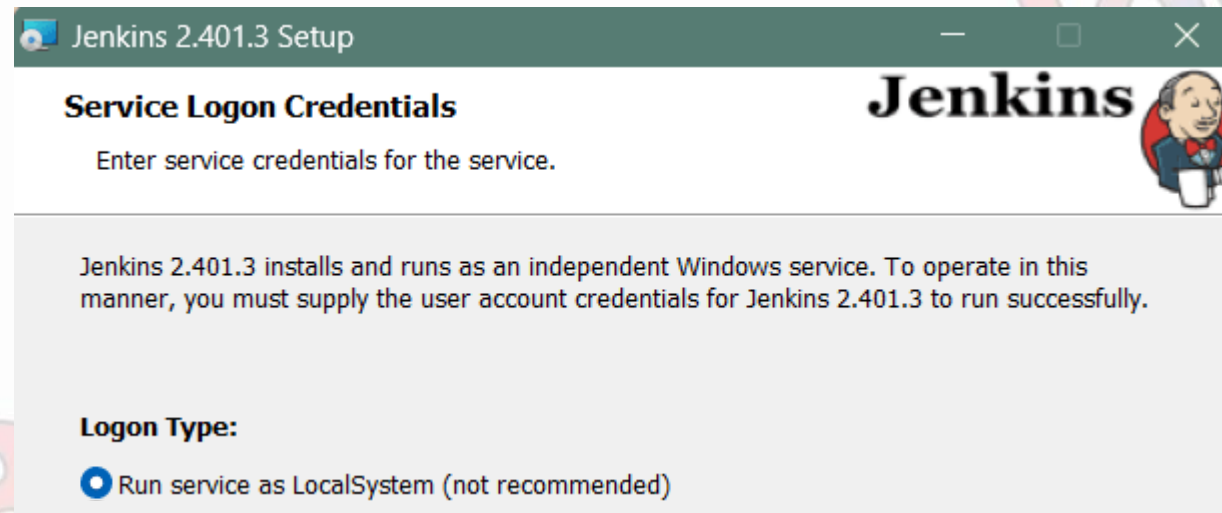
# Step 2: Select destination folder

Select the destination folder to store your Jenkins Installation and click **Next** to continue.
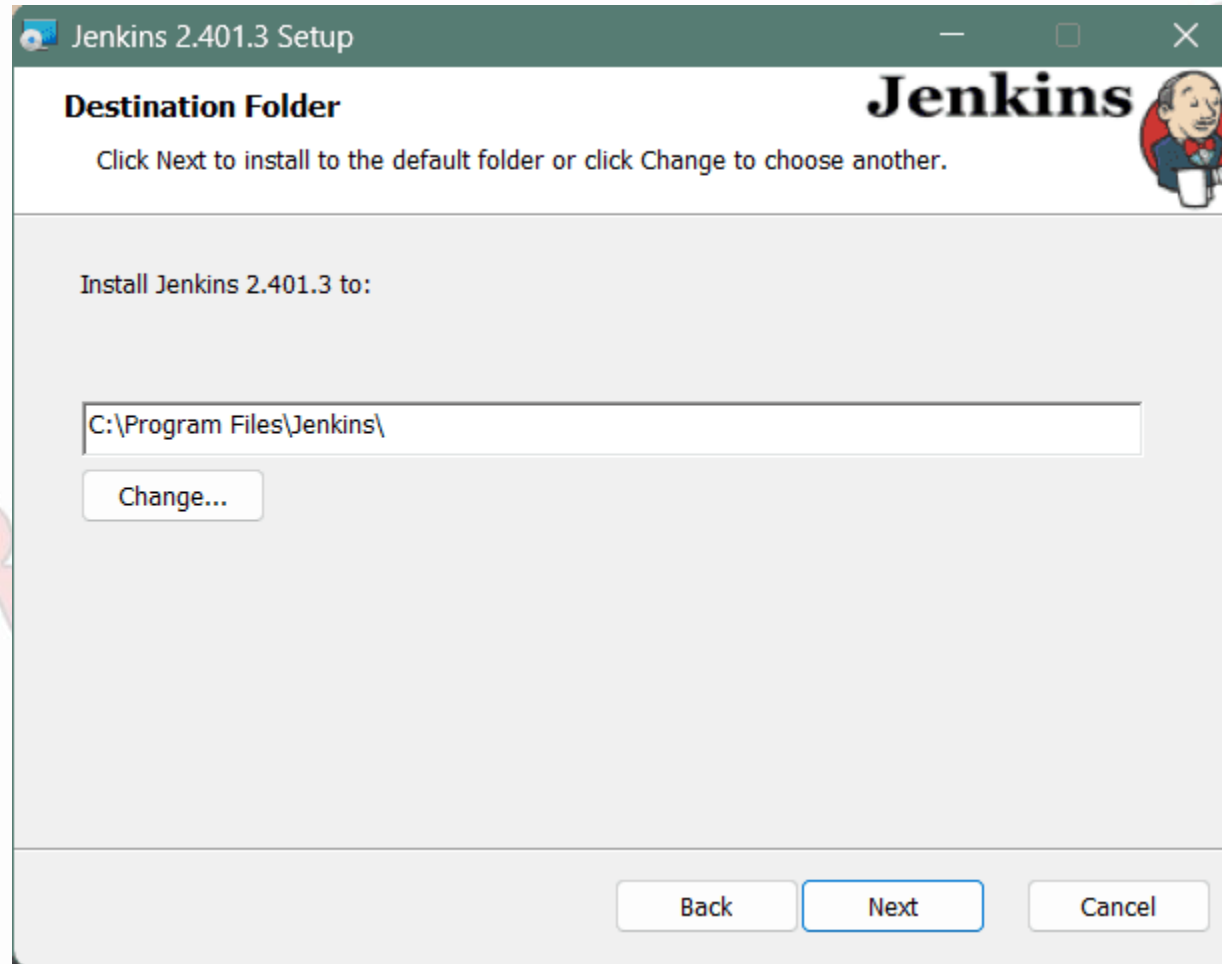
# Step 3: Service logon credentials

When Installing Jenkins, Click On **Run Service as Local System(Windows equivalent of root)** which will grant Jenkins full access to your machine and services.



After Selecting Local System, Click on Next

## Step 4: Port selection

Specify the port on which Jenkins will be running, **Test Port** button to validate whether the specified port if free on your machine or not. Consequently, if the port is free, it will show a green tick mark as shown below, then click on **Next**.
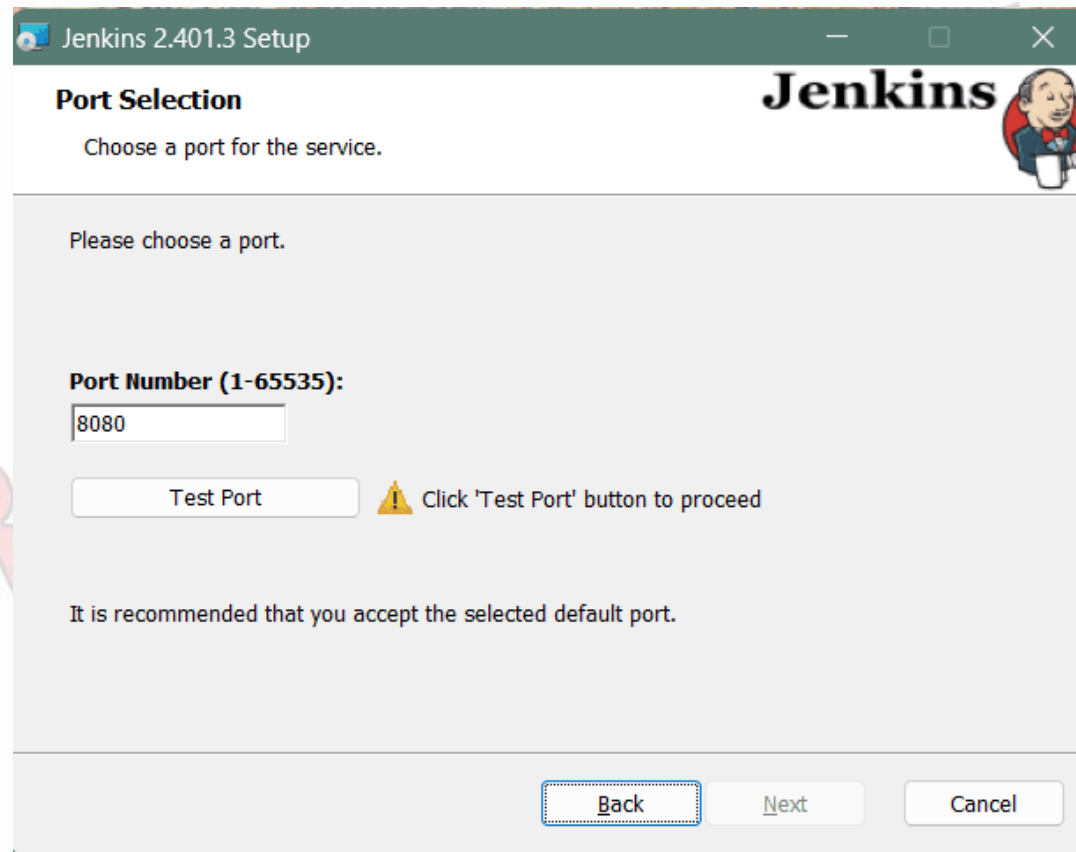


In Port Number we have to Use these ports example : 8001,8002,8081

## Step 5: Select Java home directory

The installation process checks for Java on your machine and prefills the dialog with the Java home directory. If the needed Java version is not installed on your machine, you will be prompted to install it.

Once your Java home directory has been selected, click on **Next** to continue.
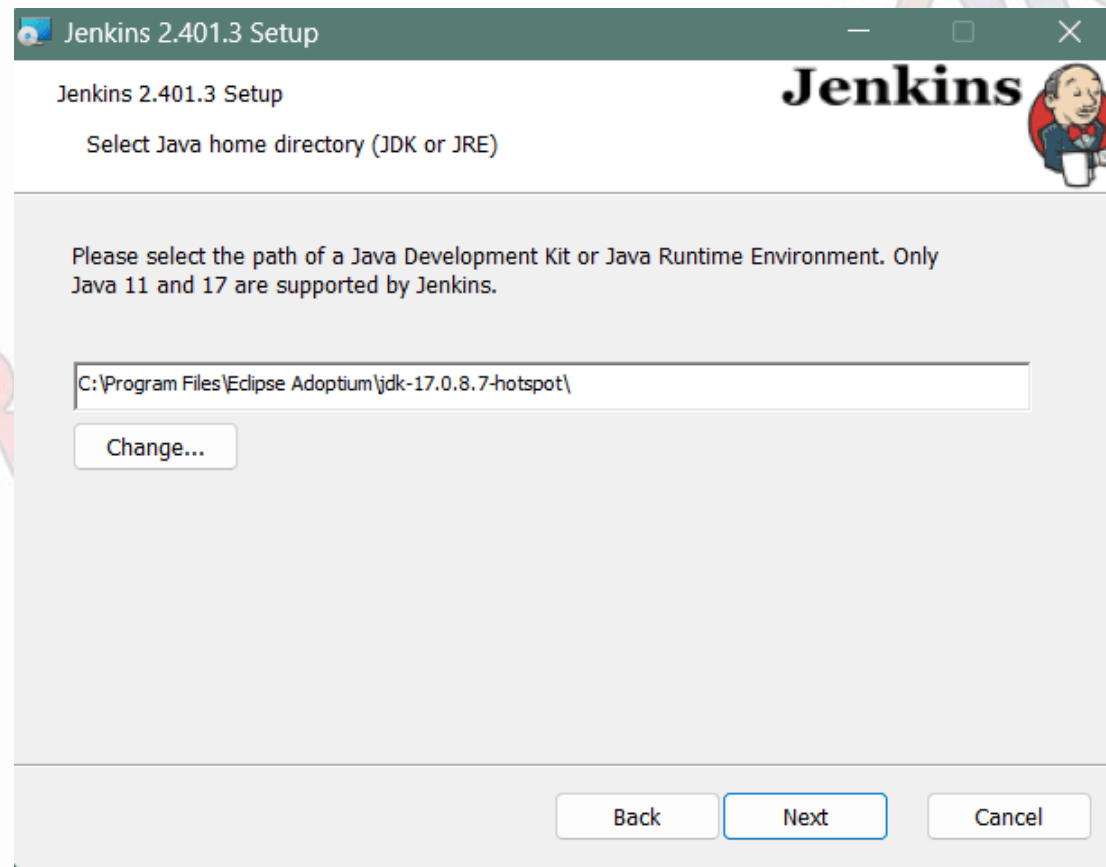
# Step 6: Custom setup
Select other services that need to be installed with Jenkins and click on **Next**
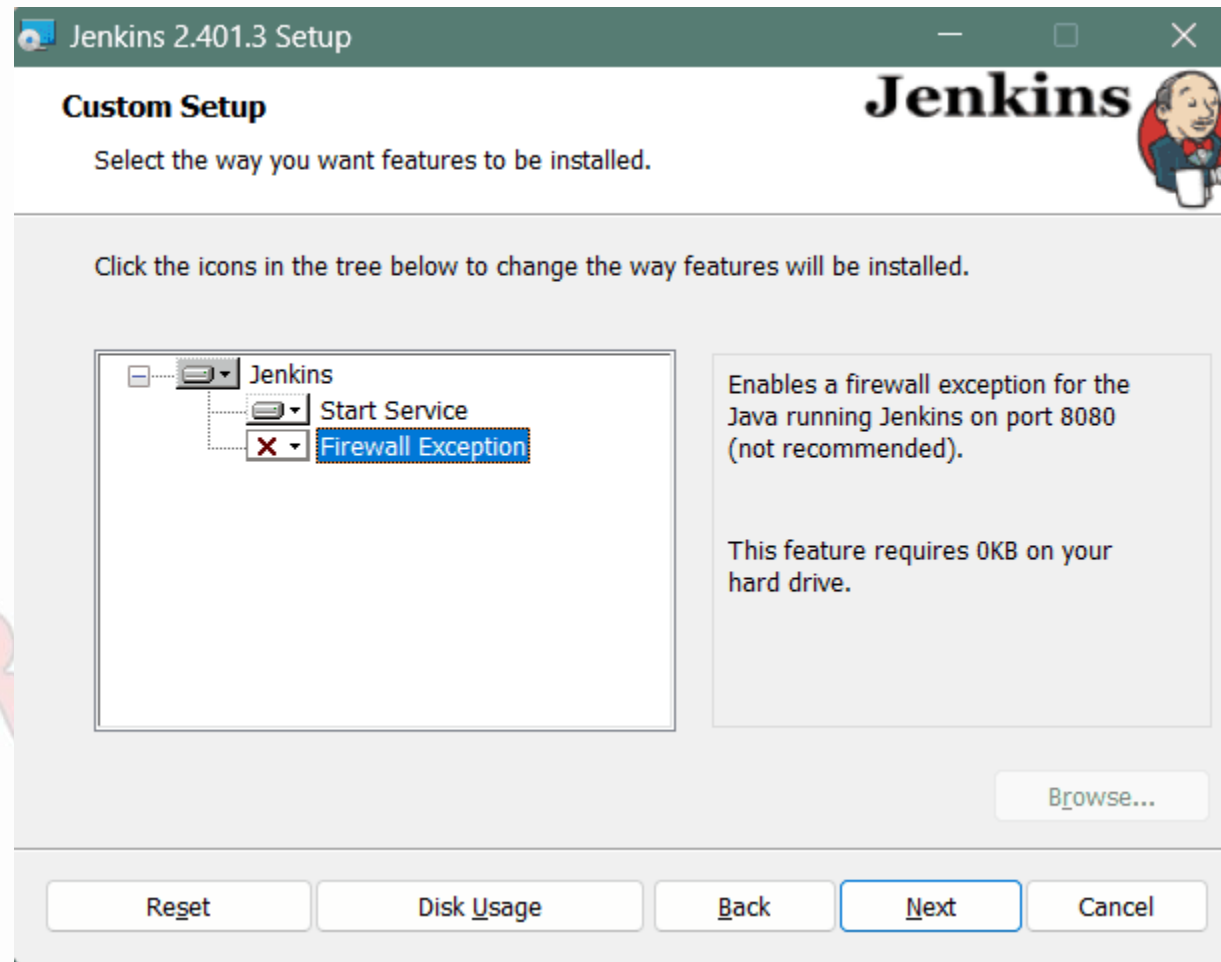


If we need any Firewall Exception we need to Enable.

# Step 7: Install Jenkins

Click on the **Install** button to start the installation of Jenkins.

Additionally, clicking on the **Install** button will show the progress bar of installation, as shown below:
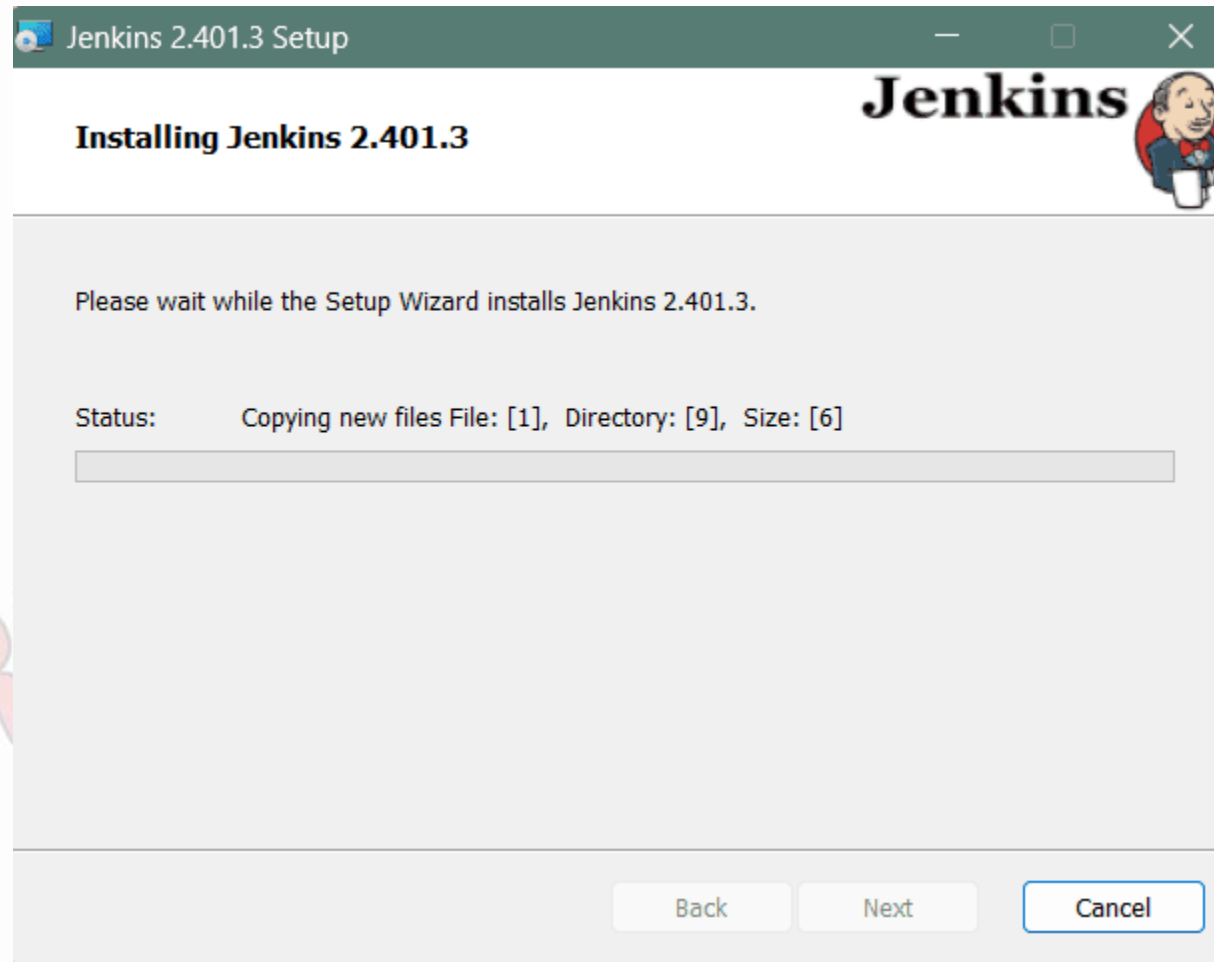
## Step 8: Finish Jenkins installation
Once the installation completes, click on **Finish** to complete the installation.

# Post-installation setup wizard
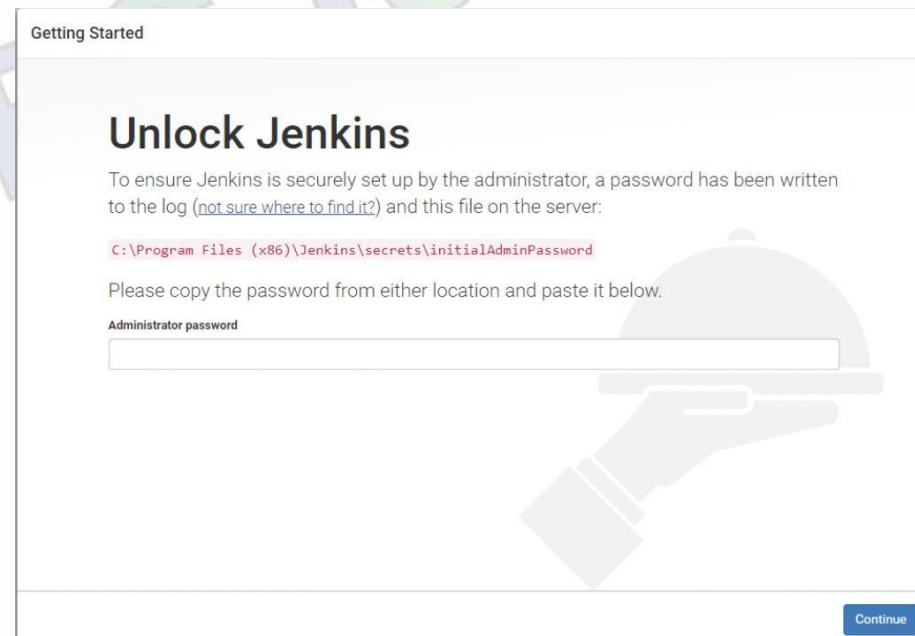
After downloading, installing and running Jenkins, the post-installation setup wizard begins.

# Unlocking Jenkins

When you first access a new Jenkins controller, you are asked to unlock it using an automatically-generated password.

# Step 1

Browse to http://localhost:8080 (or whichever port you configured for Jenkins when installing it) and wait until the **Unlock Jenkins** page appears.

## Step 2

The initial Administrator password should be found under the Jenkins installation path (set at Step 2 in Jenkins Installation).
For default installation location to C:\Program Files\Jenkins, a file called **initialAdminPassword** can be found under C:\Program Files\Jenkins\secrets.
However, If a custom path for Jenkins installation was selected, then you should check that location for **initialAdminPassword** file.
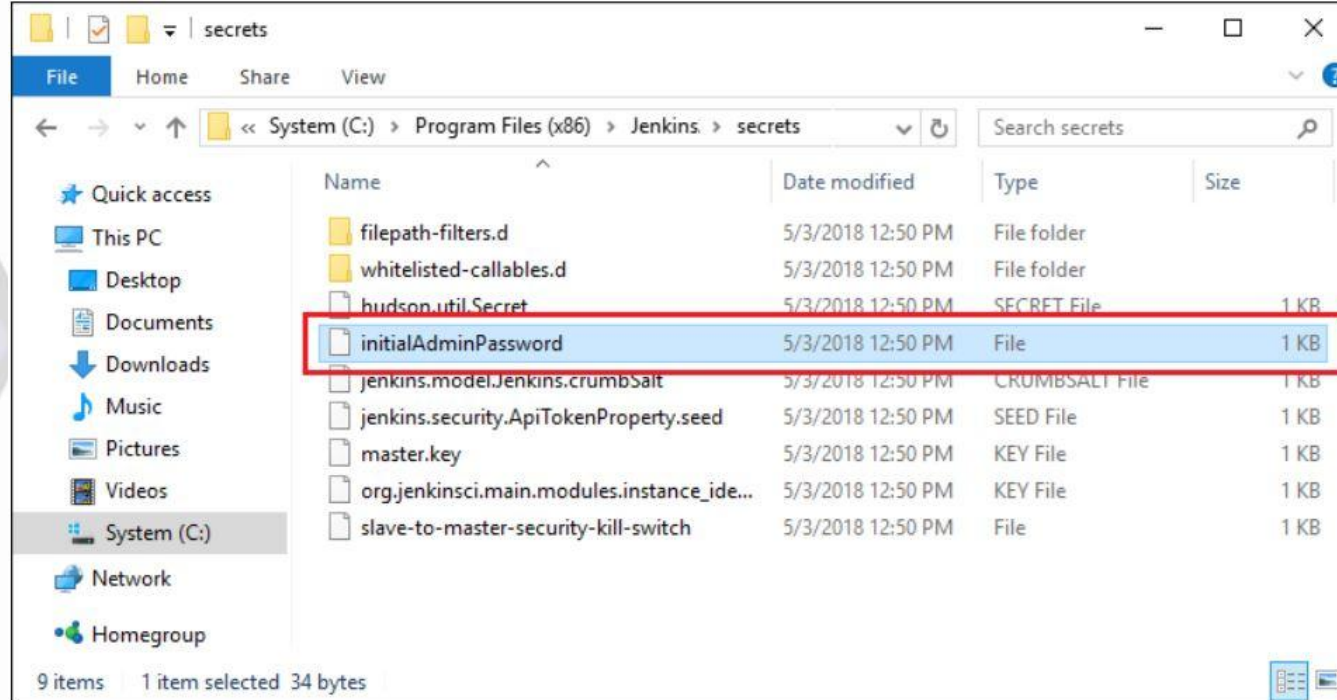
## Step 3

Open the highlighted file and copy the content of the **initialAdminPassword** file.
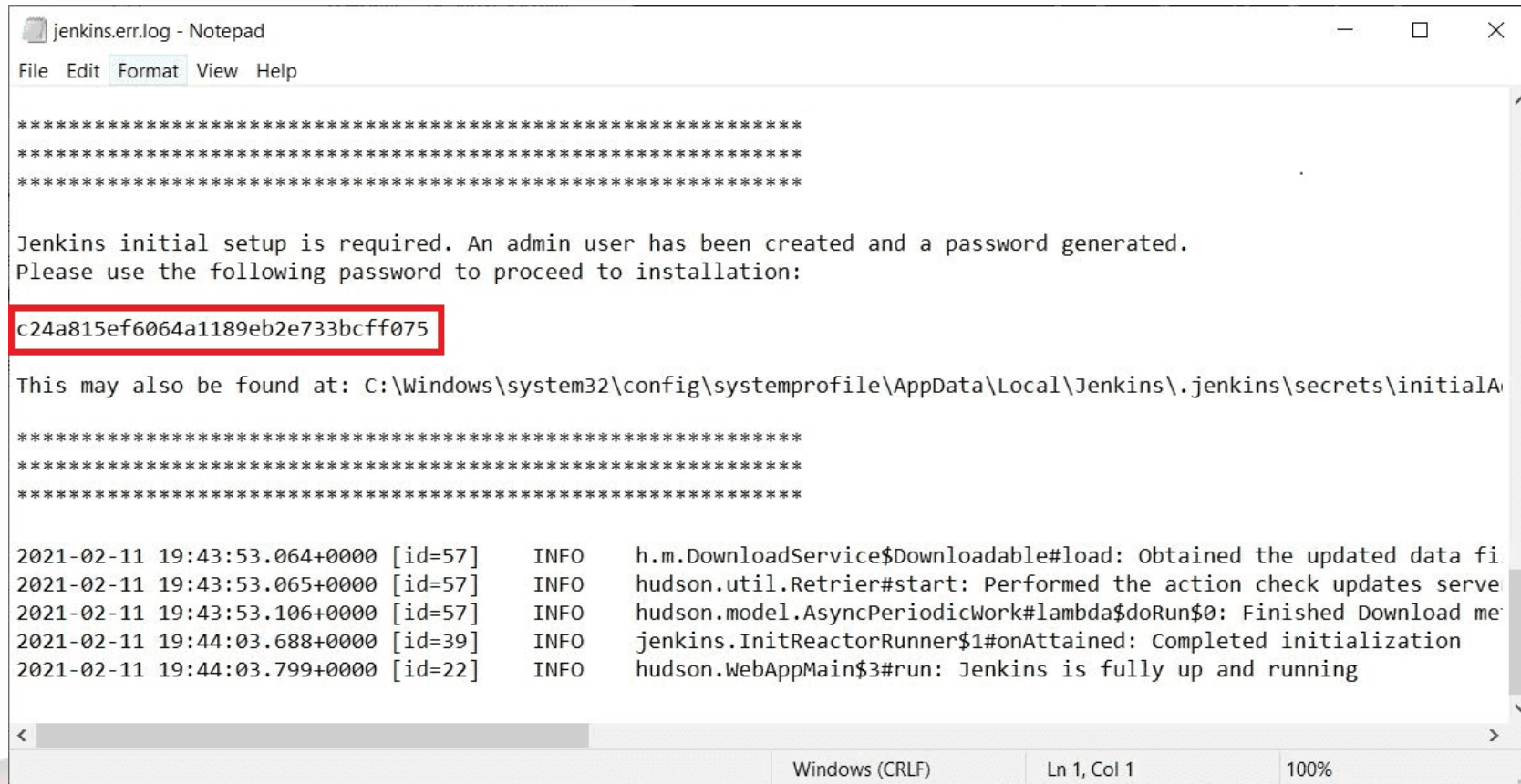


## Step 4

On the **Unlock Jenkins** page, paste this password into the **Administrator password** field and click **Continue**.

**Notes:**

You can also access Jenkins logs in the **jenkins.err.log** file in your Jenkins directory specified during the installation.

The Jenkins log file is another location (in the Jenkins home directory) where the initial password can also be obtained.

This password must be entered in the setup wizard on new Jenkins installations before you can access Jenkins's main UI. This password also serves as the default administrator account's password (with username "admin") if you happen to skip the subsequent user-creation step in the setup wizard.

We need to enter Administrator Password, copied in local files, then click on continue.
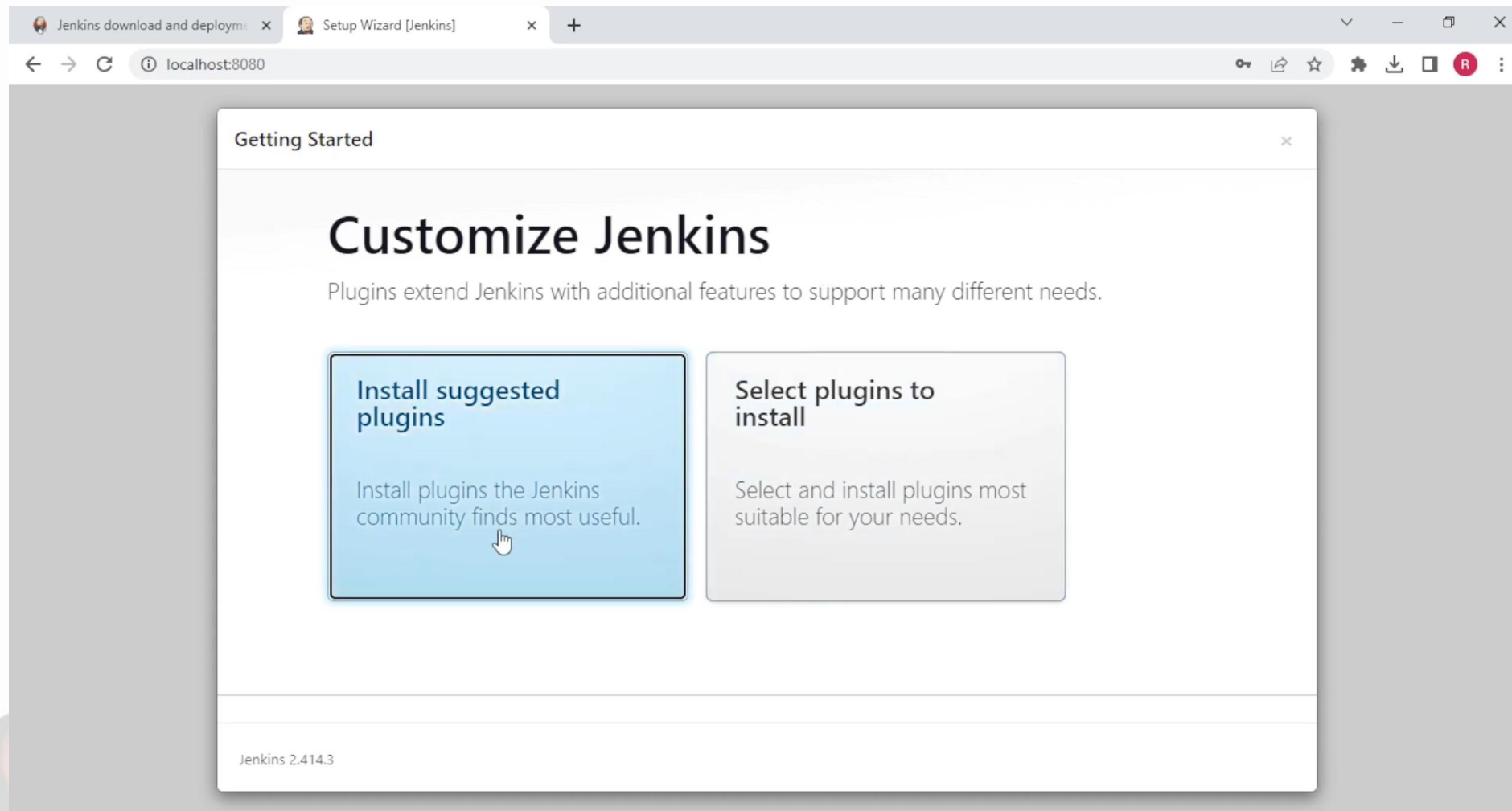
After Successful install of Jenkins, we have to navigate to http://localhost:8888/
And you will see this Getting Started window, then after click on
Install Suggested Plugins

in this window all required Plugins for Jenkins will be downloaded here.

Getting Started

# Create First Admin User

Username

Password
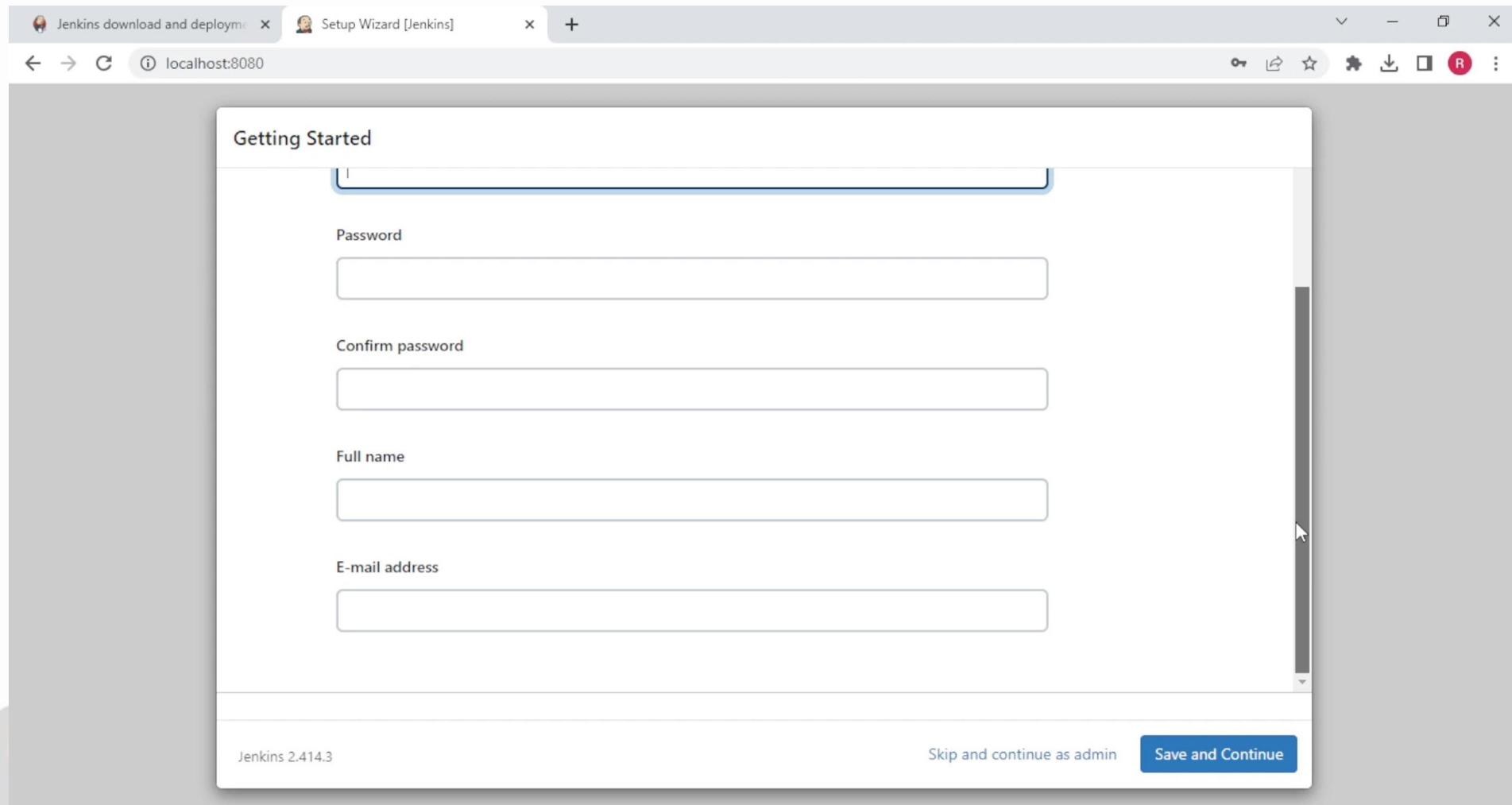
Confirm password

Full name
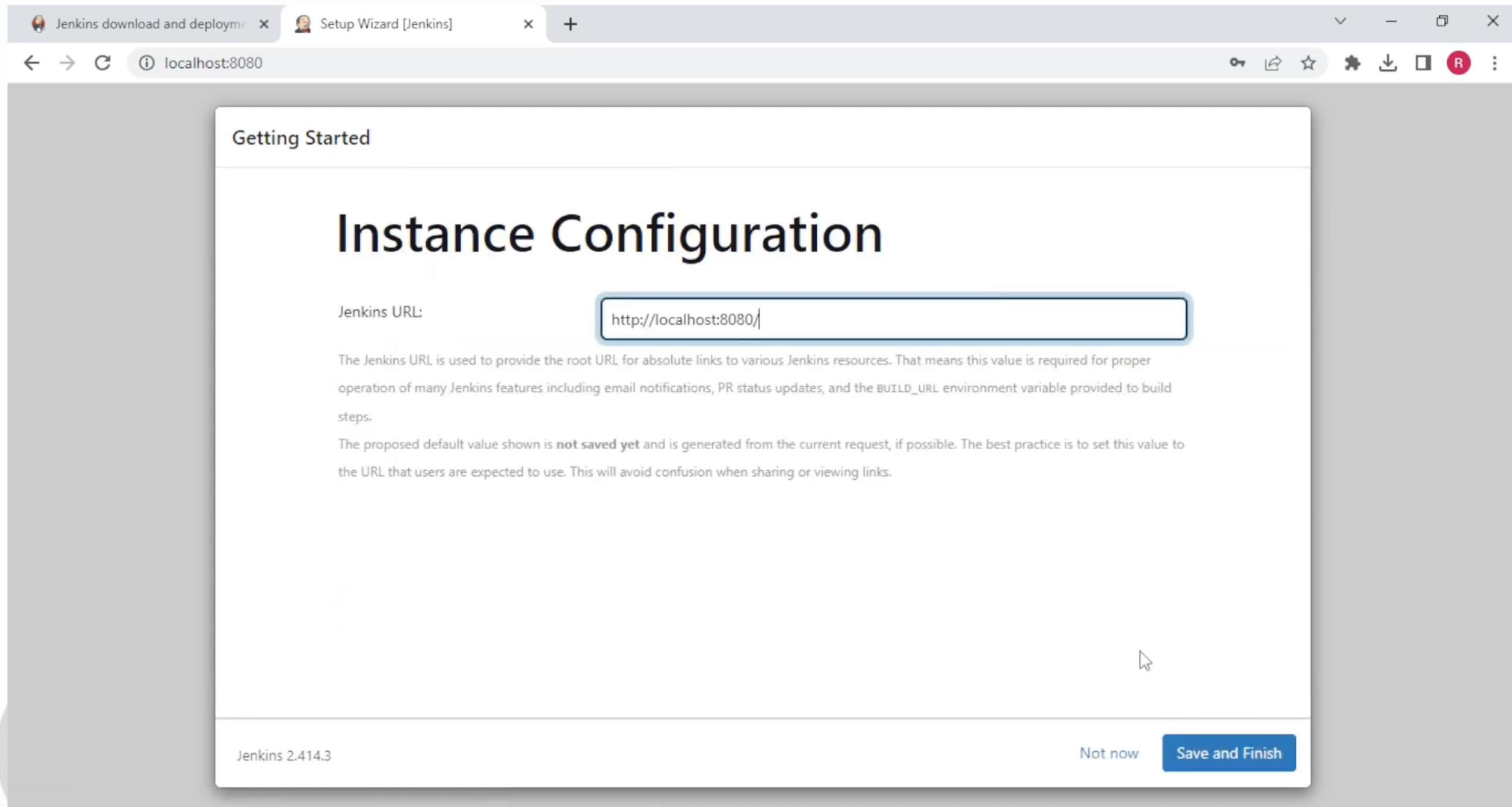
Jenkins 2.414.3

Skip and continue as admin    Save and Continue

We need to enter Username, password to create Jenkins account.

We have to enter Username, Password, Full name, Email to Login to Jenkins Account. Then Click on Save and Continue. *remember Credentials.

This window will be displayed Instance Configuration, here we need to set our URL, then click on Save and Finish.

**Jenkins**

+ New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status          0/2

## 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

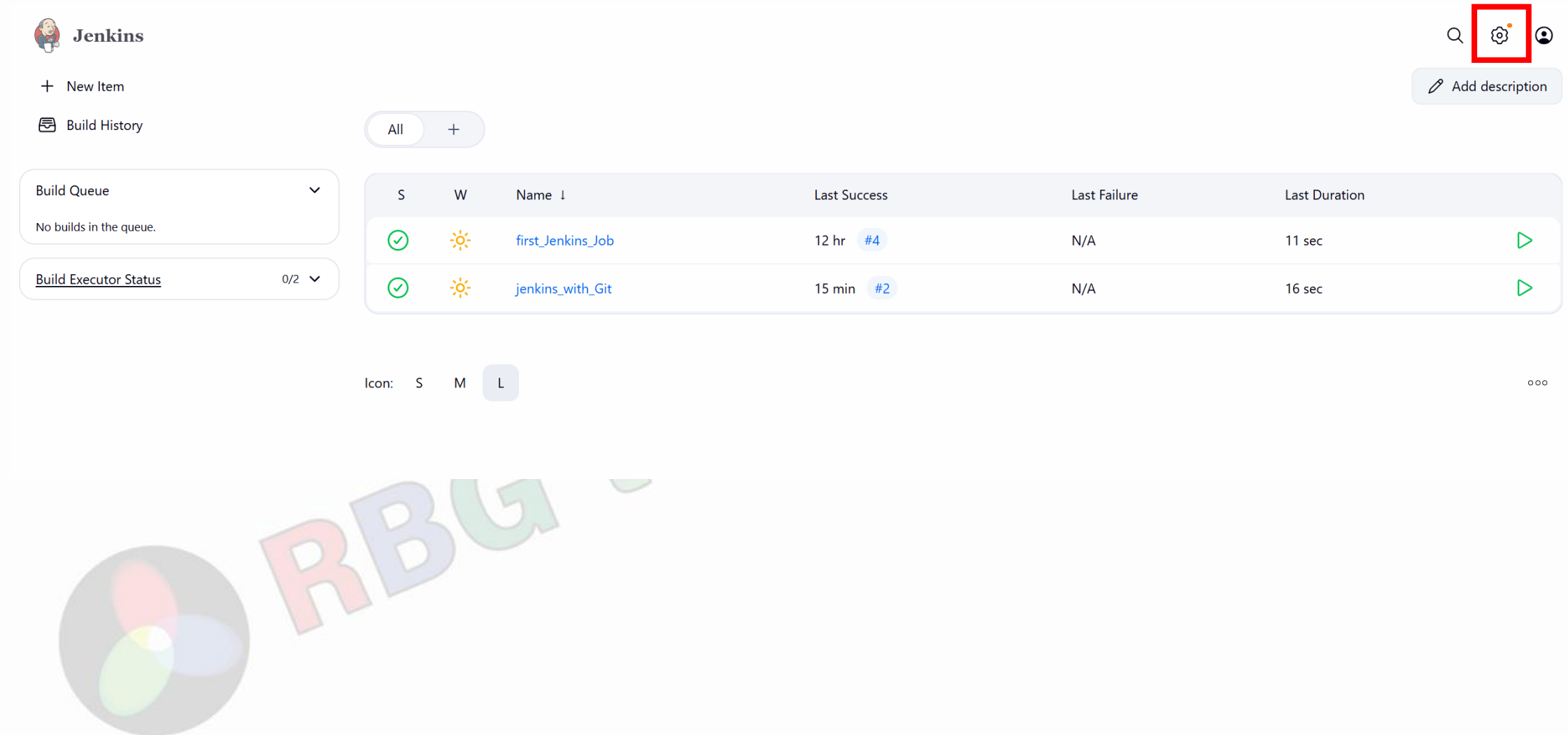Create a job                                        +

### Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds                 ?

REST API          Jenkins 2.528.1

After Account creation & Setup, we will be Re-Directed to Jenkins Homepage.

# How to Create

# Jenkins Job

*Using Local Maven Project*

**Jenkins**

+ New Item

📁 Build History

All  +

| | | | | | |
|---|---|---|---|---|---|
| **Build Queue** ⌄ | | | | | |

No builds in the queue.

| **Build Executor Status** | 0/2 ⌄ |
|---|---|

Add description

| S | W | Name ↓ | Last Success | Last Failure | Last Duration | |
|---|---|---|---|---|---|---|
| ✓ | ☀ | first_Jenkins_Job | 12 hr  #4 | N/A | 11 sec | ▷ |
| ✓ | ☀ | jenkins_with_Git | 15 min  #2 | N/A | 16 sec | ▷ |

Icon:  S  M  L

Before going to create a job, Firstly we have to add Plugin for Maven for that we have to click on settings,

**Jenkins** / Manage Jenkins ∨ / Plugins

## Plugins

🔍 maven integration                                                           /          ⬇ Update          ↻

| ⊟ | Name ↓ | Released | Installed | Health |
|---|---|---|---|---|

☑ Maven Integration plugin  3.27

This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit.                    100

Report an issue with this plugin

Maven Release Plug-in Plug-in  0.16.4

A plug-in that enables you to perform releases using the maven-release-plugin from Jenkins.

Report an issue with this plugin

- Updates    26
- Available plugins
- Installed plugins
- Advanced settings

Search for Maven Integration & select that and click on install

**Jenkins**

+ New Item

Build History

Build Queue

No builds in the queue.

Build Executor Status                0/2

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                                                    +
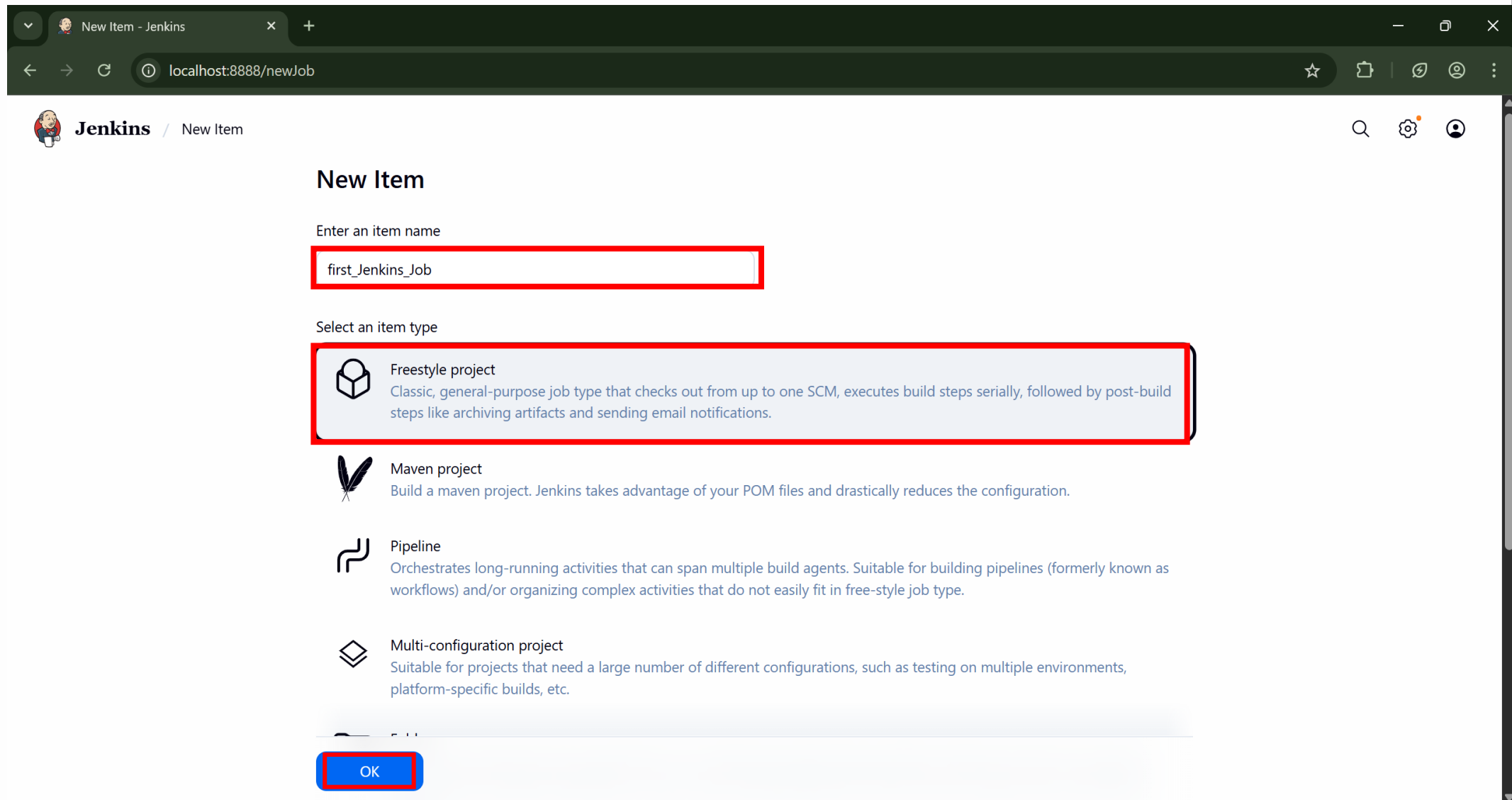
**Set up a distributed build**

Set up an agent

Configure a cloud

Learn more about distributed builds                    ?

REST API          Jenkins 2.528.1

Click on Create a job.

Firstly, we have to give a name for Jenkins Job.
Then select Freestyle project.
Click on Ok.

We will see this after clicking on Ok in before slide.

Select "use custom workspace" checkbox,
Then paste your maven project file location in directory. That should contain pom.xml in the folder

In Build step, we have to click on "+Add build Step", & Then select "Invoke top-level Maven targets".

**Jenkins** / first_Jenkins_Job / Configuration

**Configure**

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

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

☰ **Invoke top-level Maven targets** ?

Maven Version

Maven_3.9.9 ▾

Goals

clean test ▾

Advanced ⌃    ✎ Edited

POM ?

pom.xml

Properties ?

JVM Options ?

☐ Inject build variables ?

☐ Use private Maven repository

**Save**    Apply

Here in Goals, we have to enter "clean test" and then after mention POM file > pom.xml then click on Save.

After clicking on save, will redirect to this page, In this we have option Build Now, click Build now to run your First Jenkins Job.

After clicking on Build Now button, we will see the Builds down there, that's Running our Job.
After the job has run Successfully, that we can see in second image as Green tick mark

Jenkins / first_Jenkins_Job ⌄ / #4

Status

Changes

Console Output

Edit Build Information

Delete build '#4'

Timings

✓ #4 (Dec 10, 2025, 10:35:18 AM)

✎ Add description    Keep this build forever

Started by user admin

Started 58 sec ago
Took 11 sec

This run spent:

- 5 ms waiting;
- 11 sec build duration;
- 11 sec total from scheduled to completion.

No changes.

REST API          Jenkins 2.528.1

Here we can check status for the build, for output click on Console output.

Jenkins / first_Jenkins_Job ⌄ / #4 ⌄ / Console Output

```
[WARNING] source value 8 is obsolete and will be removed in a future release
[WARNING] target value 8 is obsolete and will be removed in a future release
[WARNING] To suppress warnings about obsolete options, use -Xlint:-options.
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ TestCaseDemo ---
[INFO] Using auto detected provider org.apache.maven.surefire.testng.TestNGProvider
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running jenkins.TestCaseDemo.jenkinsTest
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
Dec 10, 2025 10:35:27 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 143
Dec 10, 2025 10:35:27 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 143.0.7499.41. You may need to include a dependency on a specific version of the CDP using something
similar to `org.seleniumhq.selenium:selenium-devtools-v86:4.13.0` where the version ("v86") matches the version of the chromium-based browser you're
using and the version number of the artifact is the same as Selenium's.
Browser launched successfully       ◄─────────
Facebook login page opened successfully  ◄─────────
Browser closed successfully   ◄─────────
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.817 s -- in jenkins.TestCaseDemo.jenkinsTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.172 s
[INFO] Finished at: 2025-12-10T10:35:29+05:30
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS      ◄─────────
```

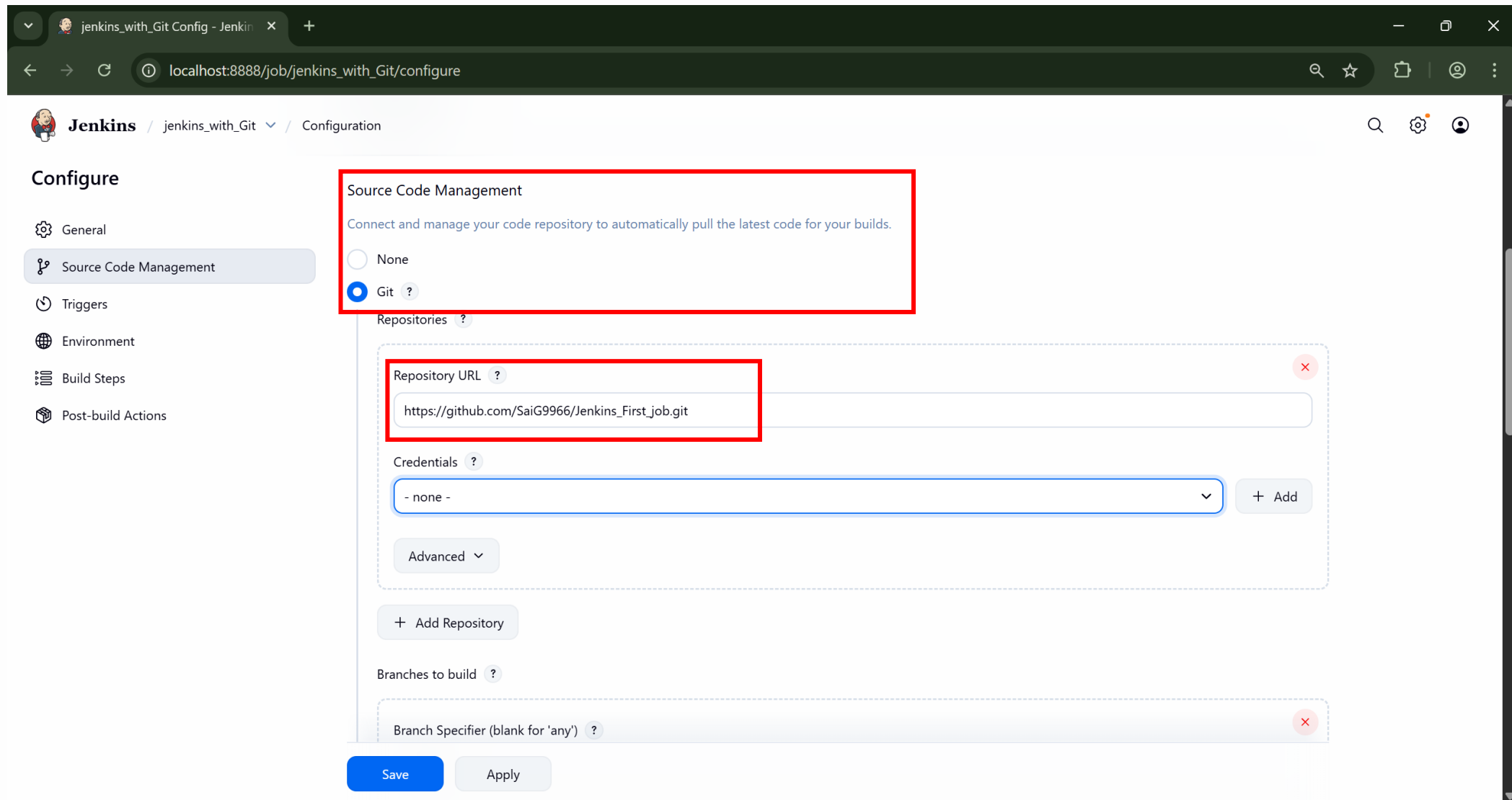www.slf4j.org/codes.html#StaticLoggerBinder

Here we can see the Output for Jenkins Job. This is the process to create and run job in Jenkins using local maven project.

# How to Create
# Jenkins Job
## *Using Git Maven Project*

first_Jenkins_Job Config - Jenki...

localhost:8888/job/first_Jenkins_Job/configure

**Jenkins** / first_Jenkins_Job ∨ / Configuration

## Configure

### General

Enabled ●

| ⚙ General |
| 🔀 Source Code Management |
| ⏱ Triggers |
| 🌐 Environment |
| ☰ Build Steps |
| 📦 Post-build Actions |

Description

Plain text  Preview

☐ Discard old builds  ?

☐ GitHub project

☐ This project is parameterized  ?
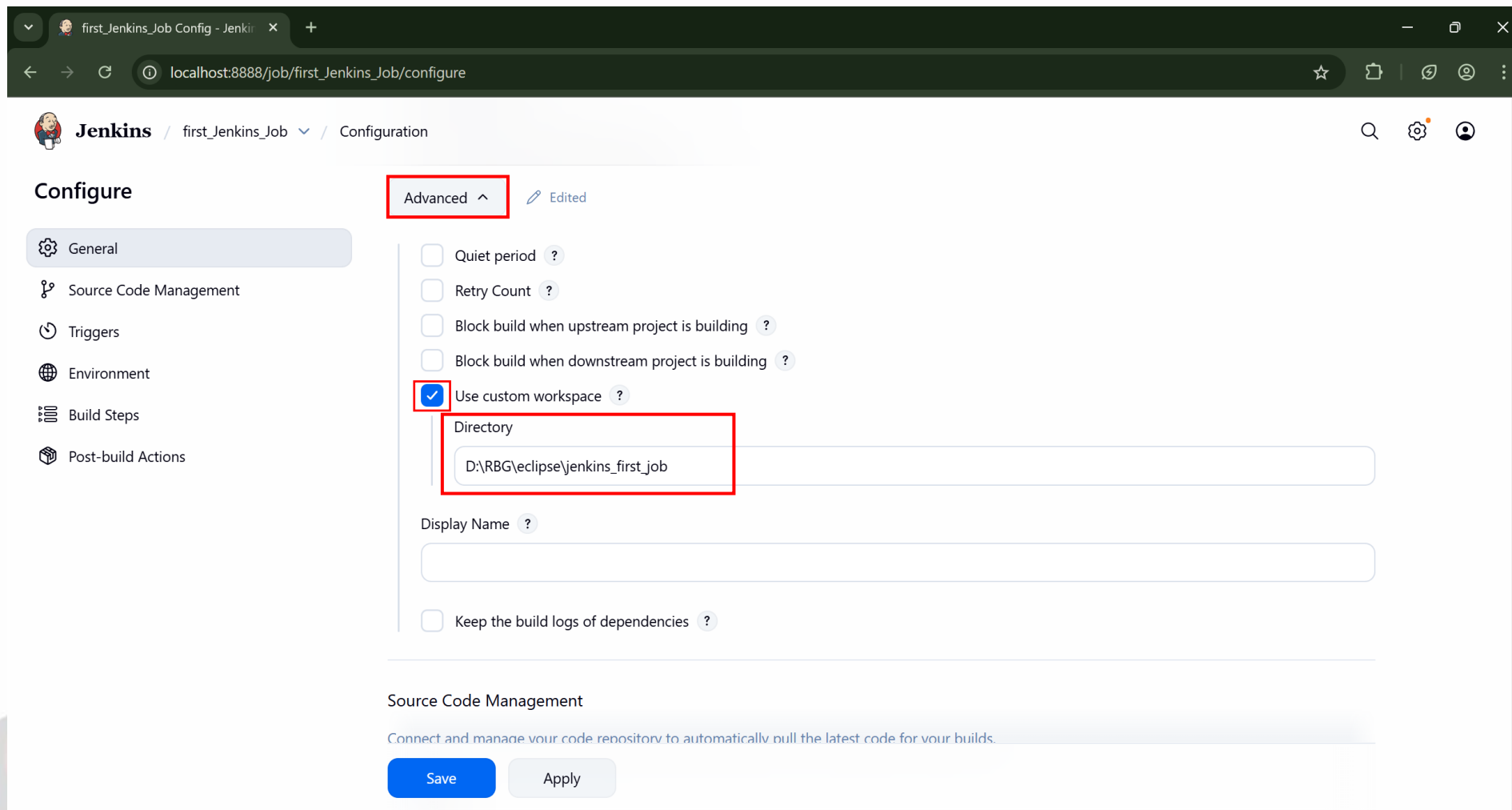
☐ Throttle builds  ?

☐ Execute concurrent builds if necessary  ?

Advanced ∨

**Save**    Apply

Here we can see the Output for Jenkins Job. This is the process to create and run job in Jenkins using git maven project. Scroll down to Source code management

In Source Code Management, select Git to run Job, after in Repository we have to enter our Git maven project URL,

**Jenkins** / jenkins_with_Git ⌄ / Configuration

**Configure**

⚙ General

⑂ Source Code Management

⏱ Triggers

🌐 Environment

≣ Build Steps

📦 Post-build Actions

☐ Provide Configuration files  ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☐ Terminate a build if it's stuck

☐ With Ant  ?

**Build Steps**

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

**+ Add build step**

▽ Filter

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Provide Configuration files

Run with timeout

Set build status to "pending" on GitHub commit

sending notifications, archiving artifacts, or triggering other jobs.

REST API          Jenkins 2.528.1

Here in Build Steps, we have to select Invoke top-level Maven Targets

jenkins_with_Git Config - Jenkin ×

localhost:8888/job/jenkins_with_Git/configure

**Jenkins** / jenkins_with_Git ⌄ / Configuration

## Configure

- ⚙ General
- 🔀 Source Code Management
- ⏱ Triggers
- 🌐 Environment
- ☰ Build Steps
- 📦 Post-build Actions

☐ Terminate a build if it's stuck

☐ With Ant  ?

### Build Steps

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

☰  **Invoke top-level Maven targets**  ?          ✕

Maven Version

Maven_3.9.9  ⌄

Goals

clean test          ⌄

Advanced ⌄

＋ Add build step

### Post-build Actions

**Save**   Apply

Here we have to click on Build steps, and in Goals we have enter a command "clean test" and then click on Save.

After clicking on Save, we have to click on Build Now, the Job will run, and u can see the number of jobs run in Builds, click on the job and we can see console output click on that we going to see the Console output.

# Tear Points

*In case of Eclipse Error*

After installing Java 17 or 21 we get an issue with eclipse, if you get this type of issue then follow these steps to follow
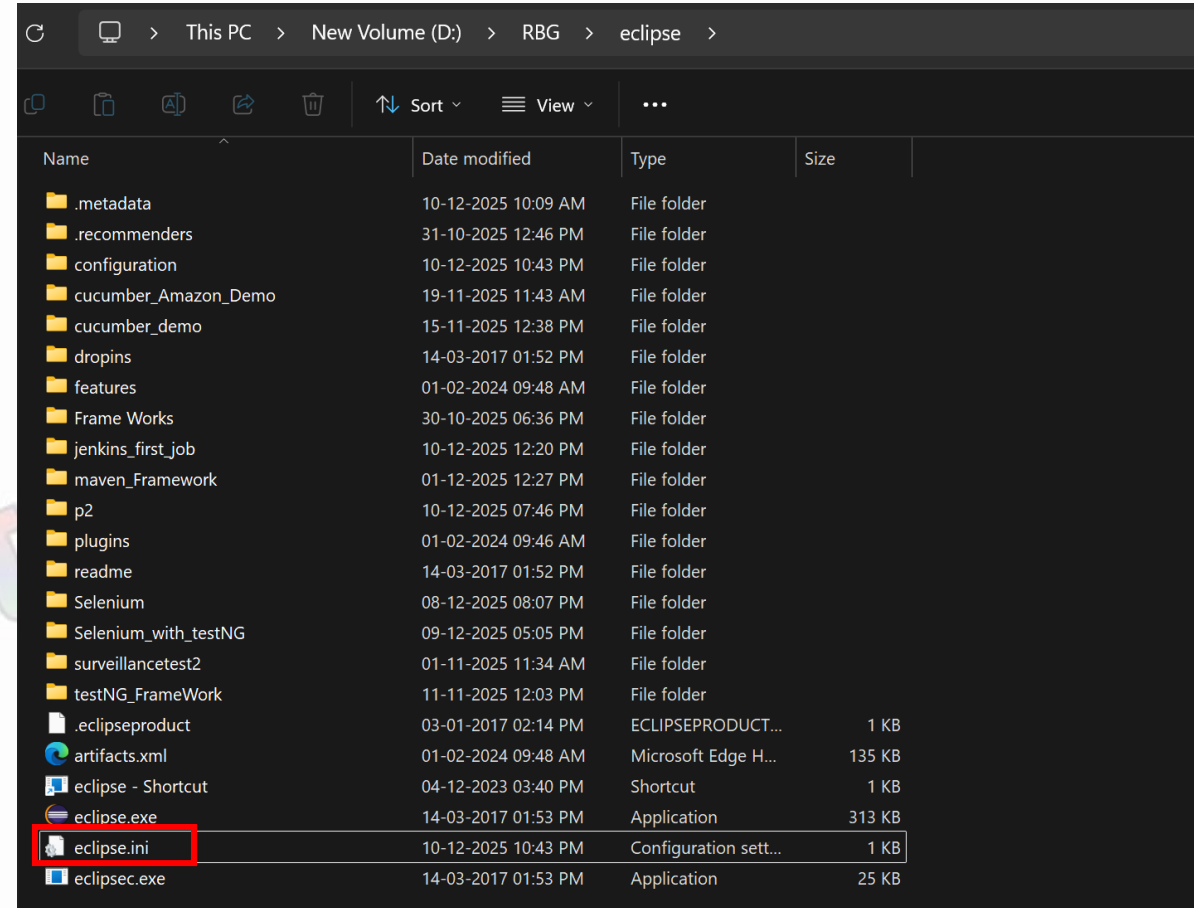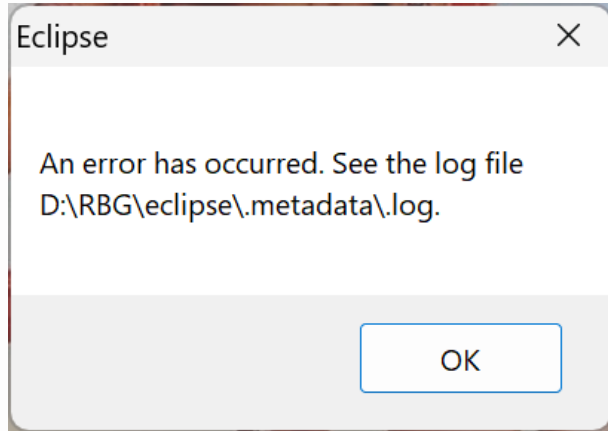


Navigate to your eclipse directory, then search for eclipse.ini file, open with note pad & type "–vm C:\Program Files\Java\jdk-1.8\jre\bin\server"
We have to mention JVM path here in the note pad above –vm args.

```
-startup
plugins/org.eclipse.equinox.launcher_1.3.201.v20161025-1711.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.401.v20161122-1740
-product
org.eclipse.epp.package.java.product
--launcher.defaultAction
openFile
-showsplash
org.eclipse.platform
--launcher.defaultAction
openFile
--launcher.appendVmargs
-vm
C:\Program Files\Java\jdk-1.8\jre\bin\server
-vmargs
-Dosgi.requiredJavaVersion=1.8
-XX:+UseG1GC
-XX:+UseStringDeduplication
-Dosgi.requiredJavaVersion=1.8
-Xms256m
-Xmx1024m
```

Above -vmargs, we have to paste the JVM path then save the notepad file, then your Eclipse will be Works Fine.
*locate JVM path from your local directory

*Thank You...*

## REGISTRATIONS NOW OPEN

THE CAREER COURSE FOR YOUR FUTURE FOR YOUR SUCCESS

# AUTOMATION & MANUAL

CORE JAVA WITH SELENIUM

**Plot no: 623, Vivekananda Nagar, Kukatpally, Hyderabad-500072**
**LAND MARK NEAR UDUPI GRAND HOTEL.**

**Contact Us**
**+91 93986 94373**