

120A3051

Shreya Idate

Batch: E3

Experiment No: 4

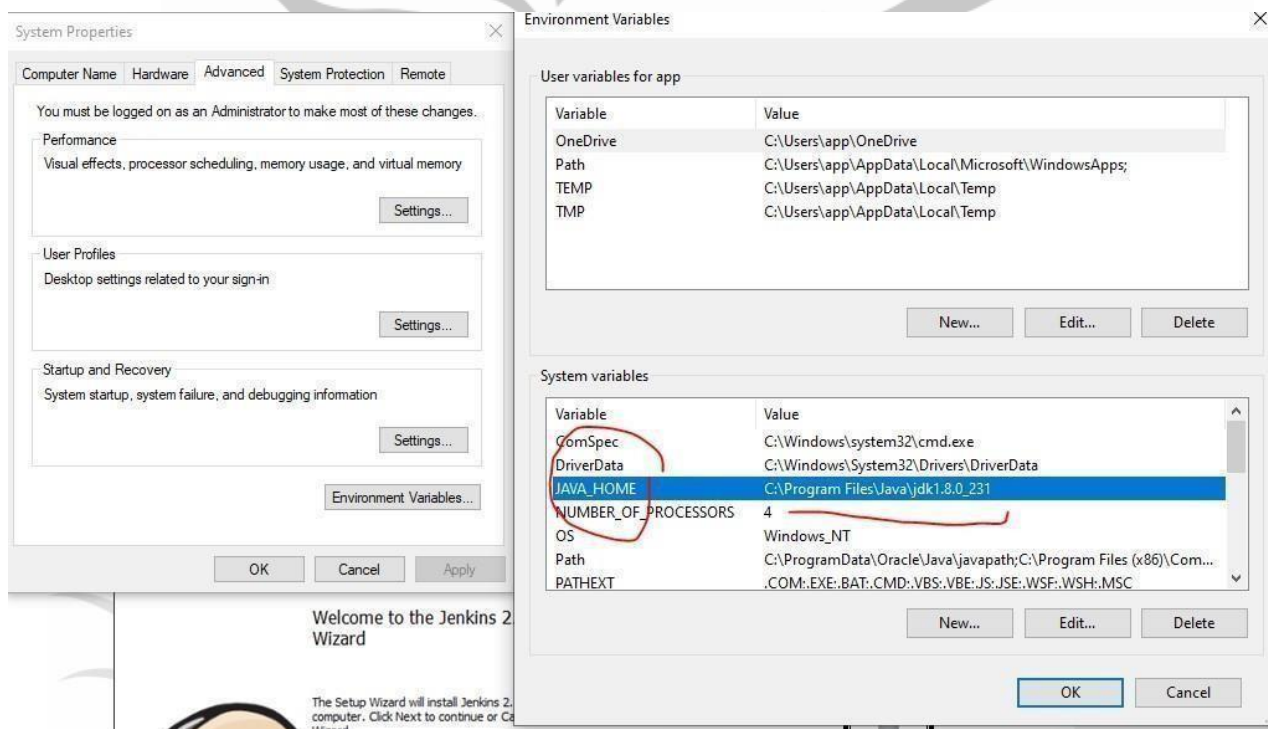
AIM: Installation of Jenkins with Maven/Ant/Gradle, GitHub and Python Plugins to setup a build Job.

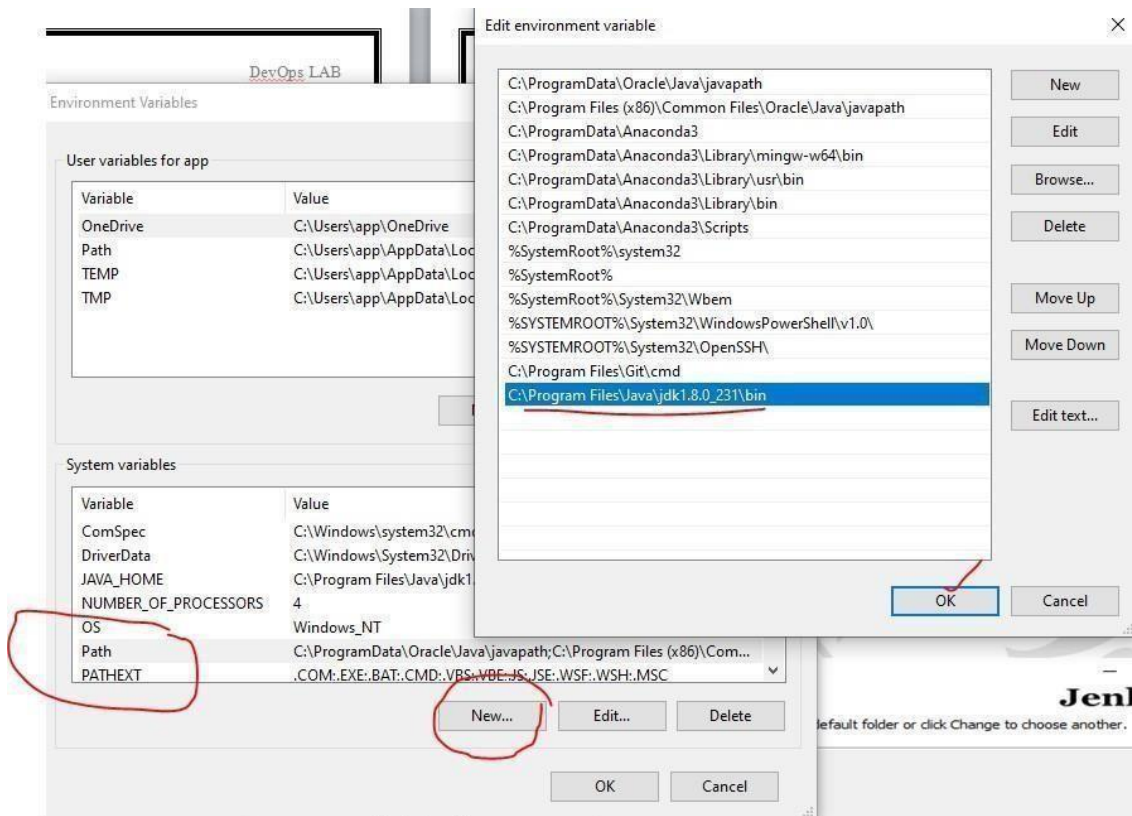
THEORY:

Jenkins is a free and open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat

Installation process of Jenkins Install Java: Jdk 1.8 or Java 11.

Set environment variable : path and JAVA_HOME





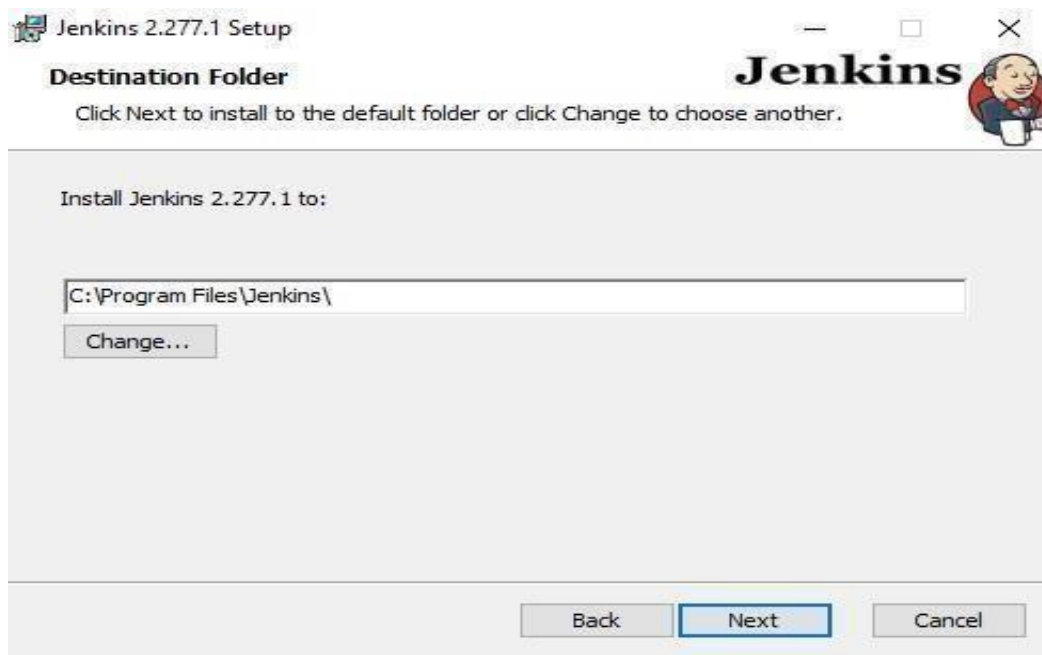
Step 1) Got to <https://www.jenkins.io/download/> and select the platform. In our case Windows

Step 2) Go to download location from local computer and unzip the downloaded package. Doubleclick on unzipped jenkins.msi. You can also Jenkins using a WAR (Web application ARchive) but that is not recommended.

Step 3) In the Jenkins Setup screen, click Next.



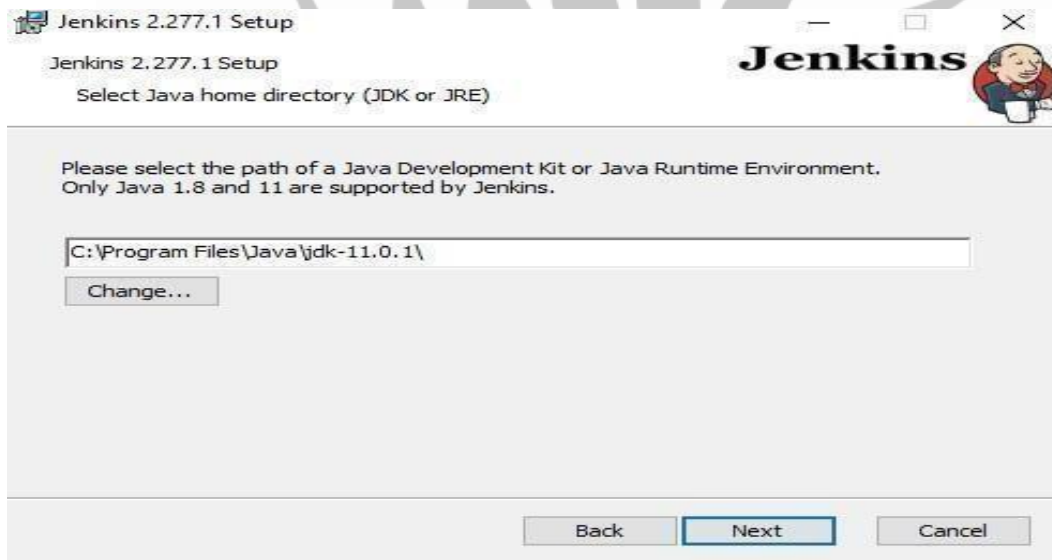
Step 4) Choose the location where you want to have the Jenkins instance installed (default location is C:\Program Files (x86)\Jenkins), then click on Next button.

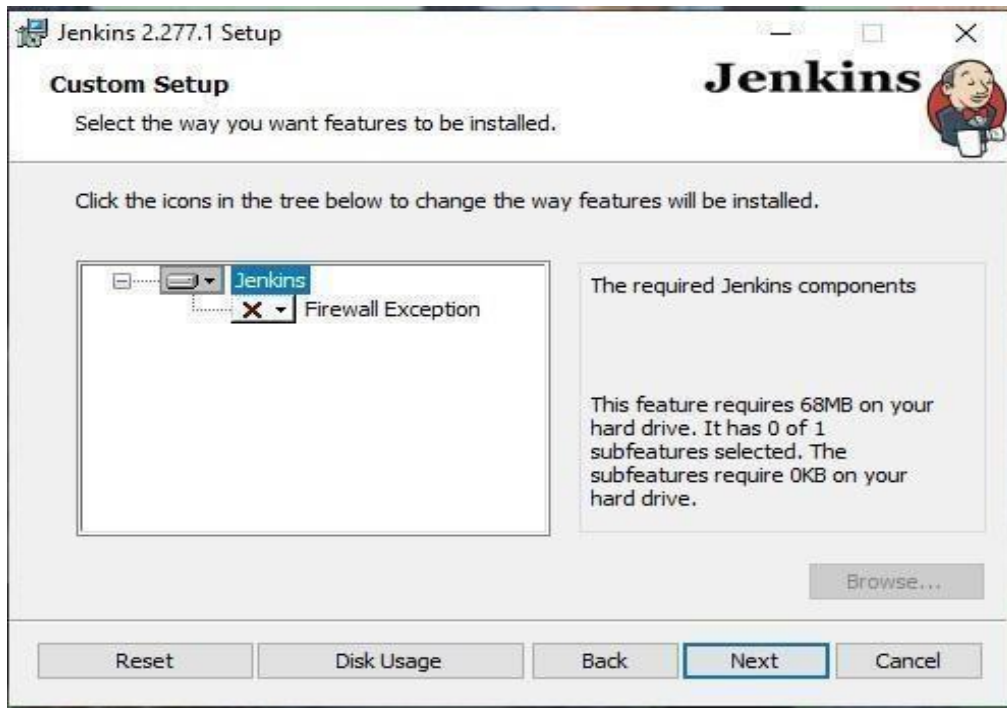


Here, Select Logon type as _ Run service as LocalSystem(Windows equivalent root)



Step 5) Click on the Install button.





Step 6) Once install is complete, click Finish.

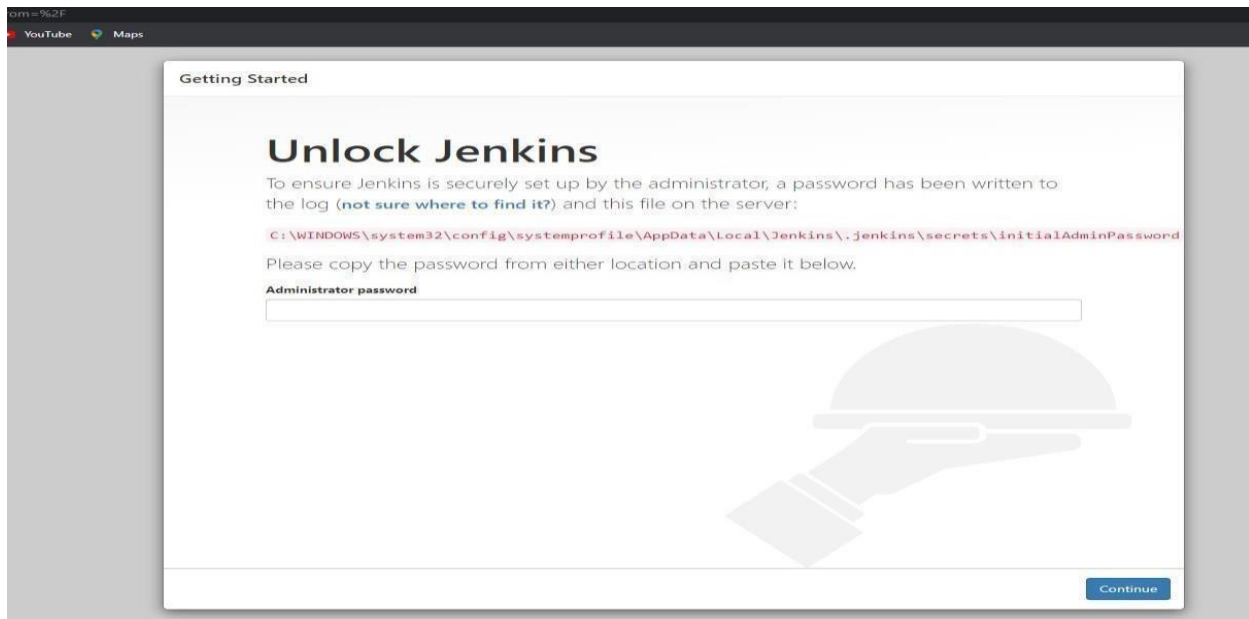


Step 7) During the installation process an info panel may pop-up to inform the user that for a complete setup, the system should be rebooted at the end of the current installation. Click on OK button when the Info panel is popping-up.

How to Unblock Jenkins?

After completing the Jenkins installation phase, you should proceed further and start its configuration. Next steps will guide you how you can unblock Jenkins application:

Step 1) After completing the Jenkins installation process, a browser tab will pop-up asking for the initial Administrator password. To access Jenkins, you need to go to browse the following path in your web browser. <http://localhost:8080>

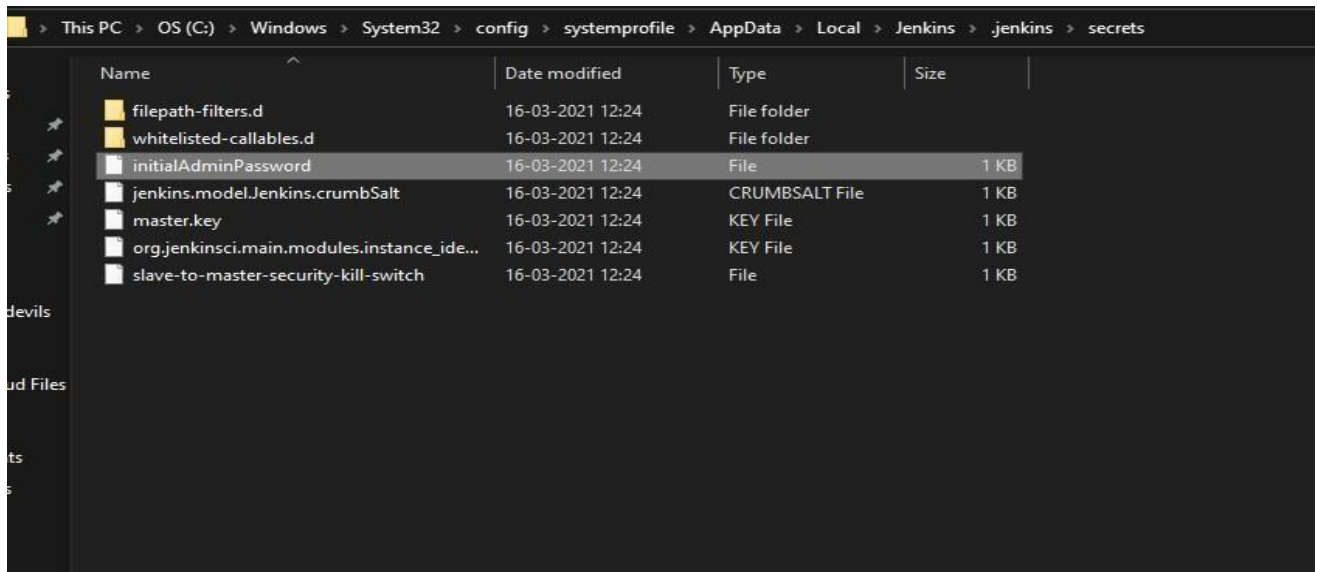


If you can access the above URL, then it confirms that Jenkins is successfully installed in your system.

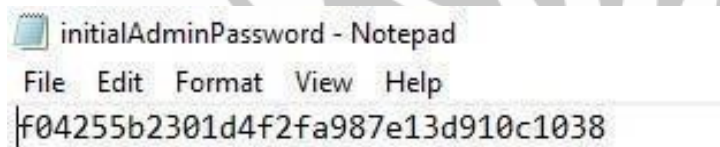
Step 2) The initial Administrator password should be found under the Jenkins installation path (set at Step 4 in Jenkins Installation).

For default installation location to C:\Program Files (x86)\Jenkins, a file called initialAdminPassword can be found under C:\Program Files (x86)\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.

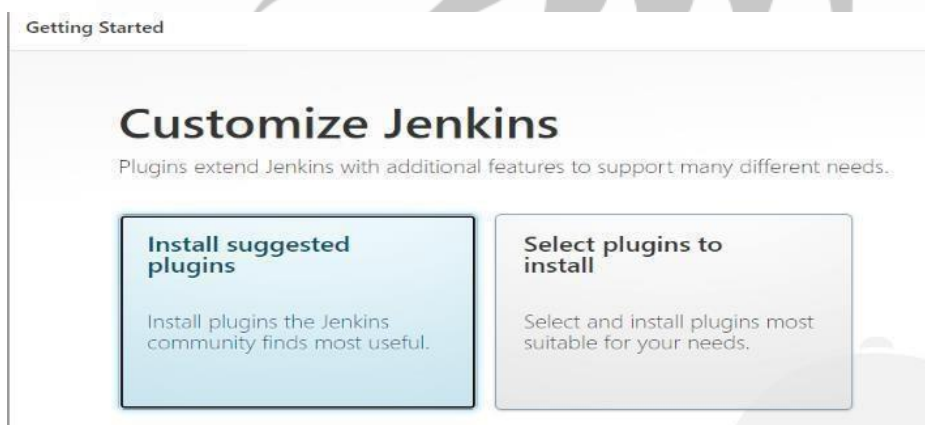


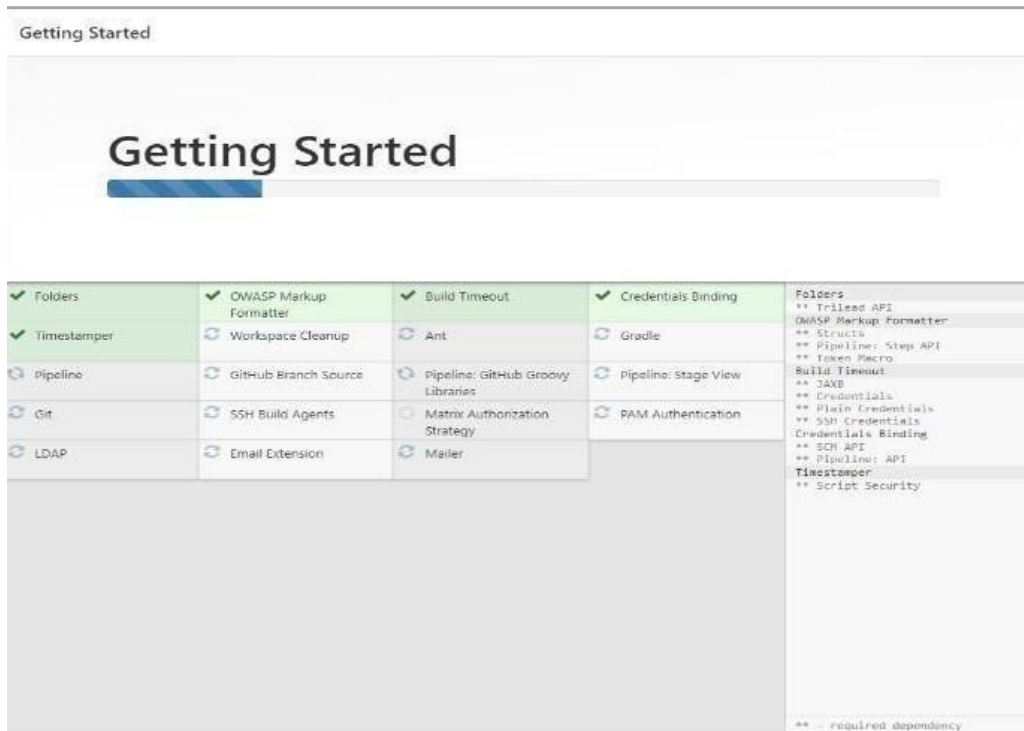
Step 4) Paste the password it into browser's pop-up tab (<http://localhost:8080/login?form=%2F>) and click on Continue button.

Customize Jenkins

You can also customize your Jenkins environment by below-given steps:

Step 1) Click on the "Install suggested plugins button" so Jenkins will retrieve and install the essential plugins





Jenkins will start to download and install all the necessary plugins needed to create new Jenkins Jobs.

Note: You can choose the Option "Select Plugins to Install" and select the plugins you want to install

Step 2) After all suggested plugins were installed, the "Create First Admin User" panel will show up. Fill all the fields with desired account details and hit the "Save and Finish" button.

The screenshot shows the 'Create First Admin User' form in Jenkins. The form has fields for Username, Password, Confirm password, Full name, and E-mail address. At the bottom, there's a 'Skip and continue as admin' link and a 'Save and Continue' button. The Jenkins version 2.277.1 is displayed at the bottom left.

Getting Started

Create First Admin User

Username:

Password:

Confirm password:

Full name:

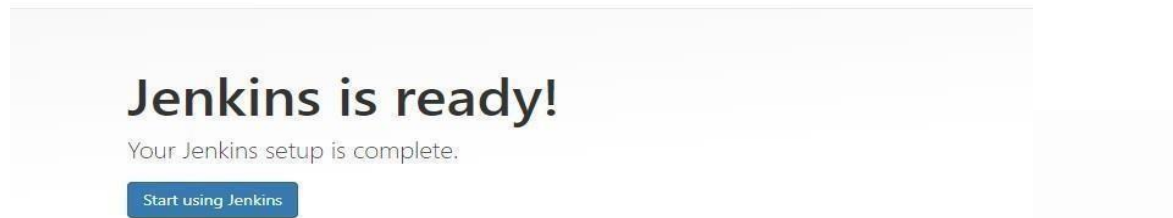
E-mail address:

Jenkins 2.277.1

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

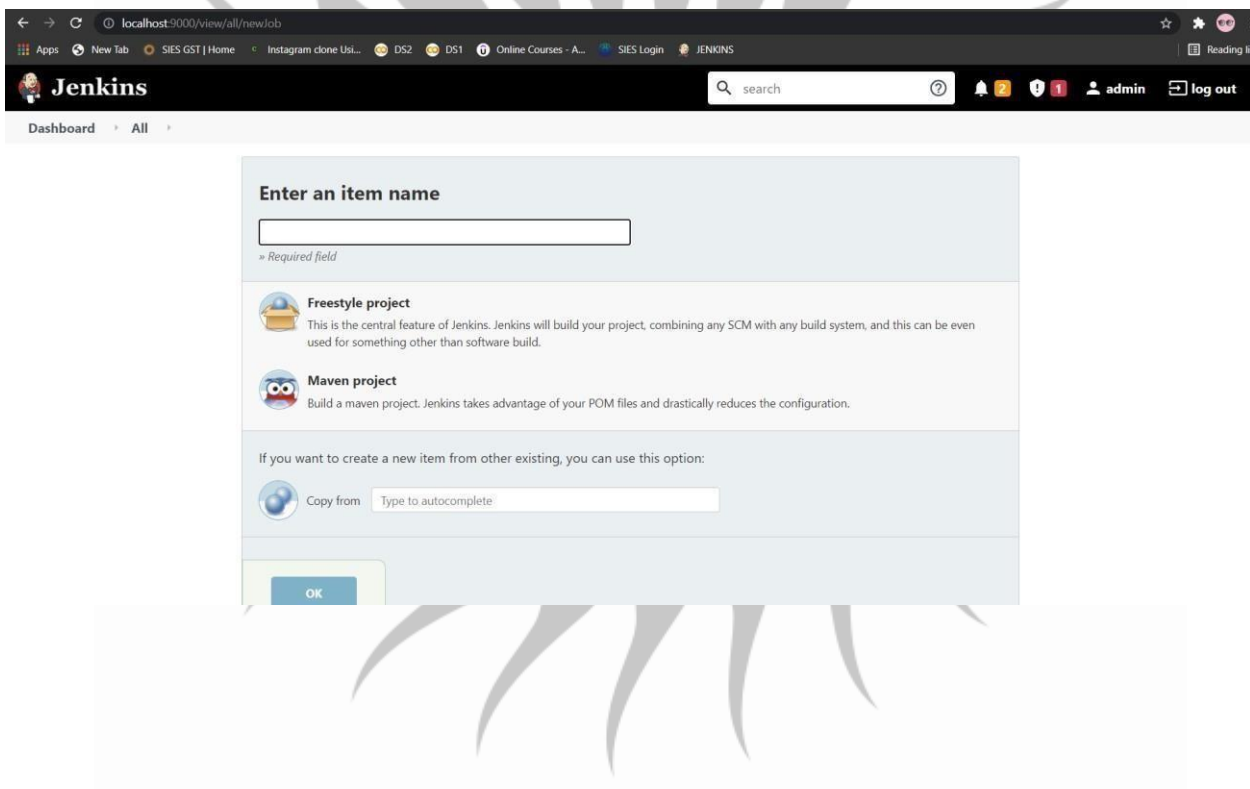
Step 3) Once you have filled the above data, finally it will ask for URL information where you can configure the default instance path for Jenkins. Leave it as it is to avoid any confusion later. However, if another application is already using 8080 port, you can use another port for Jenkins and finally save the settings, and you are done with installation of Jenkins. Hit the "Save and Continue" button:

Getting Started

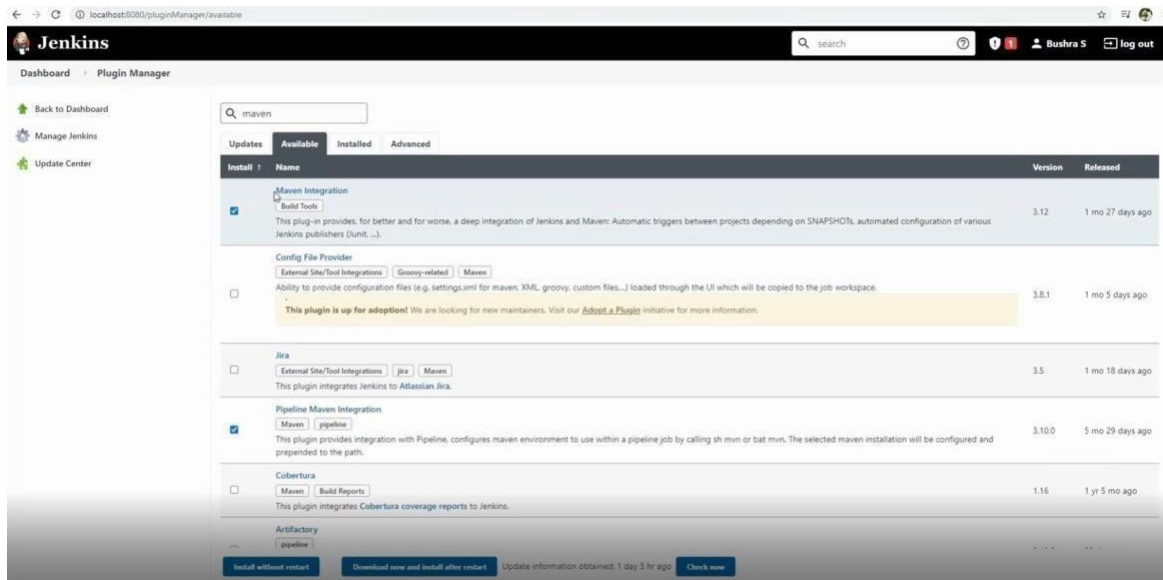


We have successfully installed a new Jenkins Server. Hit the "Start using Jenkins" button.

Below you can find the Jenkins instance up and run, ready to create first Jenkins jobs:



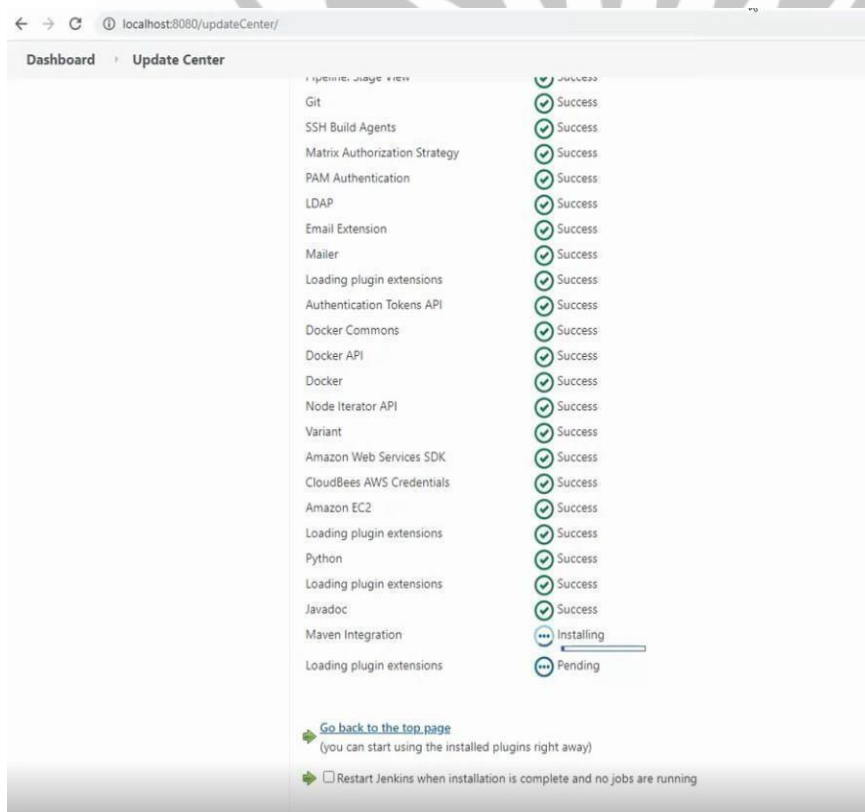
Installation of Maven Integration Plugins:



The screenshot shows the Jenkins Plugin Manager interface. The 'Available' tab is selected, and a search for 'maven' is performed. The following table lists the available plugins:

Install	Name	Version	Released
<input checked="" type="checkbox"/>	Maven Integration Build Tools This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (Junit, ...).	3.12	1 mo 27 days ago
<input type="checkbox"/>	Config File Provider External Site/Tool Integrations Groovy-related Maven Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files...) loaded through the UI which will be copied to the job workspace. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	3.8.1	1 mo 5 days ago
<input type="checkbox"/>	Jira External Site/Tool Integrations jira Maven This plugin integrates Jenkins to Atlassian Jira.	3.5	1 mo 18 days ago
<input checked="" type="checkbox"/>	Pipeline Maven Integration Maven pipeline This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	3.10.0	5 mo 29 days ago
<input type="checkbox"/>	Cobertura Maven Build Reports This plugin integrates Cobertura coverage reports to Jenkins.	1.16	1 yr 5 mo ago
<input type="checkbox"/>	Artifactory pipeline		

Buttons at the bottom: [Install without restart](#), [Download now and install after restart](#), [Update information obtained: 1 day 3 hr ago](#), [Check now](#).



The screenshot shows the Jenkins Update Center. The 'Maven Integration' plugin is listed as 'Installing'. The following table shows the status of other plugins:

Plugin	Status
Git	Success
SSH Build Agents	Success
Matrix Authorization Strategy	Success
PAM Authentication	Success
LDAP	Success
Email Extension	Success
Mailer	Success
Loading plugin extensions	Success
Authentication Tokens API	Success
Docker Commons	Success
Docker API	Success
Docker	Success
Node Iterator API	Success
Variant	Success
Amazon Web Services SDK	Success
CloudBees AWS Credentials	Success
Amazon EC2	Success
Loading plugin extensions	Success
Python	Success
Loading plugin extensions	Success
Javadoc	Success
Maven Integration	Installing
Loading plugin extensions	Pending

Buttons at the bottom: [Go back to the top page](#) (you can start using the installed plugins right away), ☐ Restart Jenkins when installation is complete and no jobs are running.



This close-up screenshot shows the 'Maven Integration' plugin status as 'Success'. The list of plugins and their statuses is as follows:

Plugin	Status
Python	Success
Loading plugin extensions	Success
Javadoc	Success
Maven Integration	Success
Loading plugin extensions	Success

Q maven

Updates Available **Installed** Advanced

Enabled	Name ↓
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
<input checked="" type="checkbox"/>	Javadoc Plugin
<input checked="" type="checkbox"/>	JSch dependency plugin Jenkins plugin that brings the JSch library as a plugin dependency, and provides an SSHAuthenticatorFactory for using JSch wi
<input checked="" type="checkbox"/>	JUnit Allows JUnit-format test results to be published.
<input checked="" type="checkbox"/>	Mailer Plugin This plugin allows you to configure email notifications for build results
<input checked="" type="checkbox"/>	Maven Integration plugin This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects, automated configuration of various Jenkins publishers (JUnit, ...).

Q python plugin

Updates Available **Installed** Advanced

Enabled	Name ↓	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	bouncycastle API This plugin provides a stable API to Bouncy Castle related tasks.	2.23		Uninstall
<input checked="" type="checkbox"/>	Command Agent Launcher Plugin Allows agents to be launched using a specified command.	1.6		Uninstall
<input checked="" type="checkbox"/>	JUnit Plugin Allows JUnit-format test results to be published.	1.52		Uninstall
<input checked="" type="checkbox"/>	Oracle Java SE Development Kit Installer Plugin Allows the Oracle Java SE Development Kit (JDK) to be installed via download from Oracle's website.	1.5		Uninstall
<input checked="" type="checkbox"/>	Python Plugin Adds the ability to execute python scripts as build steps.	1.3		Uninstall
<input checked="" type="checkbox"/>	Trilead API Plugin Trilead API Plugin provides the Trilead library to any dependent plugins in an easily update-able manner.	1.0.13		Uninstall

Installation of GitHub Plugin:

Updates

Available

Installed

Advanced

Install	Name	Version	Released
<input type="checkbox"/>	<div>GitHub</div> <div>External Site/Tool Integrationsgithub</div> <div>This plugin integrates GitHub to Jenkins.</div>	1.34.1	3 days 8 hr ago
<input type="checkbox"/>	<div>GitHub API</div> <div>api-plugingithubLibrary plugins (for use by other plugins)</div> <div>This plugin provides GitHub API for other plugins.</div>	1.123	6 mo 7 days ago
<input type="checkbox"/>	<div>GitHub Branch Source</div> <div>githubpipelineSource Code Management</div> <div>Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc.</div>	2.11.2	1 mo 13 days ago
<input type="checkbox"/>	<div>Pipeline: GitHub Groovy Libraries</div> <div>githubpipeline</div> <div>Allows Pipeline Groovy libraries to be loaded on the fly from GitHub.</div>	1.0	4 yr 7 mo ago

Install without restart

Download now and install after restart

Update information obtained: 5 min 43 sec ago

Check now

```
at sun.net.www.protocol.https.HttpsURLConnectionImpl.getResponseCode(Unknown Source)
at hudson.model.UpdateCenter$updateCenterConfiguration.testConnection(UpdateCenter.java:1406)
at hudson.model.UpdateCenter$updateCenterConfiguration.checkUpdateCenter(UpdateCenter.java:1189)
at hudson.model.UpdateCenter$ConnectionCheckJob.run(UpdateCenter.java:1640)
at java.util.concurrent.Executors$RunnableAdapter.call(Unknown Source)
at java.util.concurrent.FutureTask.run(Unknown Source)
at hudson.remoting.AtMostOneThreadExecutor$Worker.run(AtMostOneThreadExecutor.java:121)
at java.lang.Thread.run(Unknown Source)
```

OkHttp

Success

GitHub API

Success

Plain Credentials

Success

Credentials Binding

Success

Pipeline: SCM Step

Success

Git client

Success

Git

Success

Token Macro

Success

GitHub

Success

Loading plugin extensions

Success

Go back to the top page

(you can start using the installed plugins right away)

Restart Jenkins when installation is complete and no jobs are running

<input checked="" type="checkbox"/>	<div>GitHub API Plugin</div> <div>This plugin provides GitHub API for other plugins.</div>	1.123	Uninstall
<input checked="" type="checkbox"/>	<div>GitHub plugin</div> <div>This plugin integrates GitHub to Jenkins.</div>	1.34.1	Uninstall

OUTPUT:

Dashboard > Plugin Manager

[Back to Dashboard](#)
[Manage Jenkins](#)

Plugin Manager

UpdatesAvailableInstalledAdvanced

Q maven

Name	Enabled
Maven Integration plugin 3.19 This plug-in provides a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...). This plugin is deprecated and it's recommended to avoid using it. Report an issue with this plugin	<input checked="" type="checkbox"/> <input type="checkbox"/>

Jenkins

Jenkins > Update Center

[Back to Dashboard](#)
[Manage Jenkins](#)
[Manage Plugins](#)

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Python Downloaded Successfully. Will be activated during the next bootRestarting Jenkins Running

→ [Go back to the top page](#)
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Dashboard > Plugin Manager

[Back to Dashboard](#)
[Manage Jenkins](#)
[Update Center](#)

Plugin Manager

UpdatesAvailableInstalledAdvanced

Q github

Name	Enabled
GitHub API Plugin 1.303-400.v35c2d8258028 This plugin provides GitHub API for other plugins. Report an issue with this plugin	<input checked="" type="checkbox"/> <input type="checkbox"/>
GitHub Branch Source Plugin 1687.v7618247e672d Multibranch projects and organization folders from GitHub. Maintained by CloudBees, Inc. Report an issue with this plugin	<input checked="" type="checkbox"/> <input type="checkbox"/>
GitHub plugin 1.34.5 This plugin integrates GitHub to Jenkins. Report an issue with this plugin	<input checked="" type="checkbox"/> <input type="checkbox"/>
Pipeline: GitHub Groovy Libraries 38.v445716ea_edda_ Allows Pipeline Groovy libraries to be loaded on the fly from GitHub. Report an issue with this plugin	<input checked="" type="checkbox"/> <input type="checkbox"/>

[↑ Back to Dashboard](#)[Manage Jenkins](#)[Update Center](#)

Plugin Manager

[Updates](#) [Available](#) [Installed](#) [Advanced](#)

Name ↓	Enabled
--------	---------

Maven Integration plugin 3.19

This plug-in provides a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...). This plugin is deprecated and it's recommended to avoid using it.
[Report an issue with this plugin](#)

[↑ Back to Dashboard](#)[Manage Jenkins](#)[Update Center](#)

Plugin Manager

[Updates](#) [Available](#) [Installed](#) [Advanced](#)

Name ↓	Enabled
--------	---------

Python Plugin 1.3

Adds the ability to execute python scripts as build steps.
[Report an issue with this plugin](#)



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.

If you want to create a new item from other existing, you can use this option:



Copy from

OK

autocomplete

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

Description

Hello World program in Python

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Disable this project ?

☐ Execute concurrent builds if necessary ?

Advanced...

Source Code Management

☒ None

☐ Git ?

Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published Gradle build scans

☐ Terminate a build if it's stuck

☐ With Ant ?

hello.py ×

hello.py

1 print("Hello World!")

Build

Execute Windows batch command ?

Command

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

```
cd ..  
cd ..  
d:  
cd 120A3051  
python hello.py
```

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save

Apply



Console Output

Started by user [Shreya Idate](#)

Running as SYSTEM

Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\HelloPy

[HelloPy] \$ cmd /c call C:\Windows\TEMP\jenkins14028921767187002157.bat

C:\ProgramData\Jenkins\.jenkins\workspace\HelloPy>d:

D:\>cd 120A3051

D:\120A3051>python hello.py

Hello World!

D:\120A3051>exit 0

Finished: SUCCESS

CONCLUSION: Successfully installed Jenkins and configured Jenkins with Maven/Ant/Gradle, GitHub and Python Plugins to setup a build Job.