Experiment 05 - Jenkins Setup

Roll No.	37
Name	Mikil Lalwani
Class	D15-A
Subject	DevOps Lab
LO Mapped	LO1: To understand the fundamentals of DevOps engineering and be fully proficient with DevOps terminologies, concepts, benefits, and deployment options to meet your business requirements LO3: To understand the importance of Jenkins to Build and deploy Software Applications on server environment

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

Introduction:

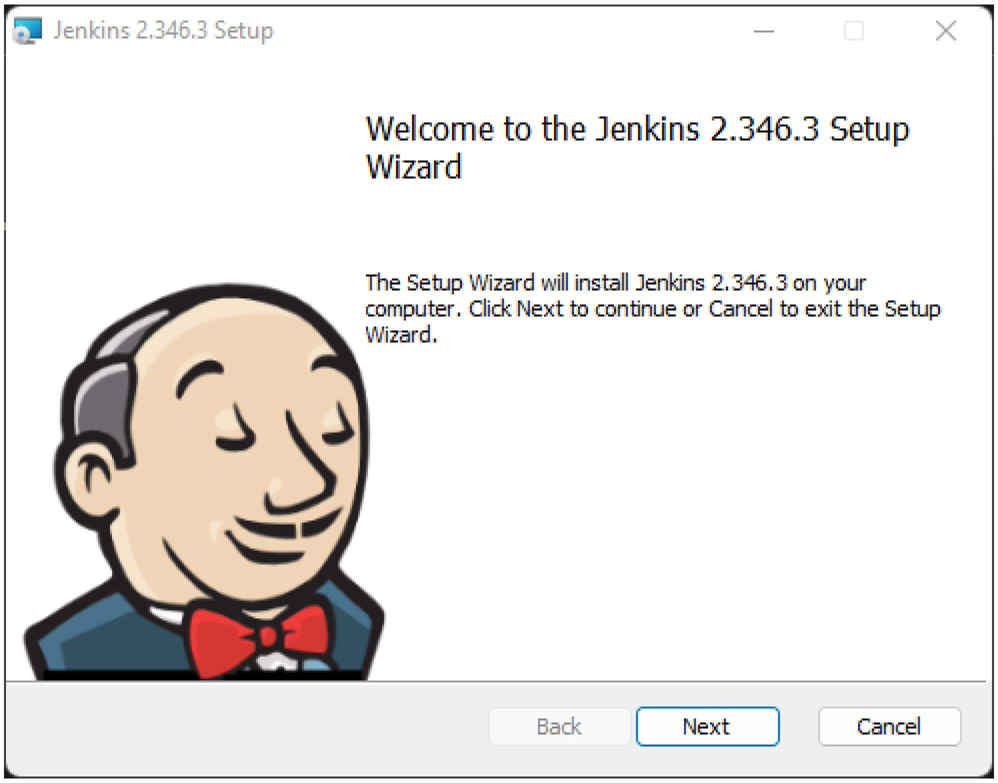
Continuous integration (CI) is a software development practice in which developers merge their changes to the main branch many times per day. Each merge triggers an automated code build and test sequence, which ideally runs in less than 10 minutes. A successful CI build may lead to further stages of continuous delivery.
If a build fails, the CI system blocks it from progressing to further stages. The team receives a report and repairs the build quickly, typically within minutes.
All competitive technology companies today practice continuous integration. By working in small iterations, the software development process becomes predictable and reliable. Developers can iteratively build new features. Product managers can bring the right products to market, faster. Developers can fix bugs quickly and usually discover them before they even reach users.

Jenkins:

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. 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.
With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.
Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example Git, Maven 2 project, Amazon EC2, HTML publisher etc.

Installation:

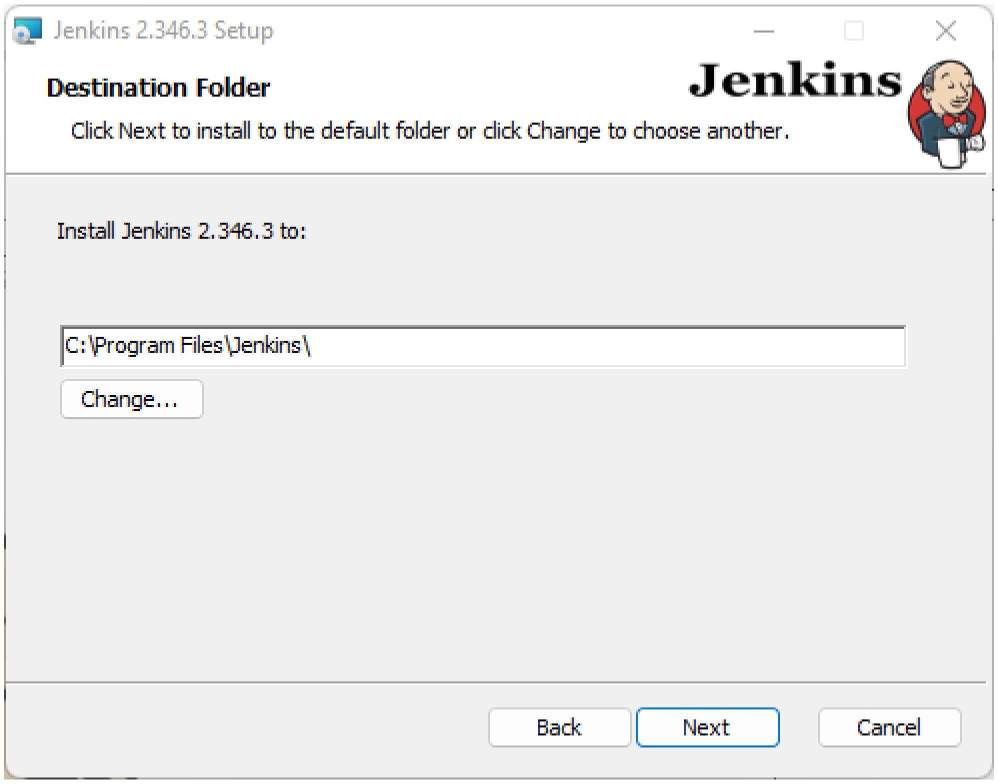
1. Download Jenkins setup from the official website and run it.


Jenkins 2.346.3 Setup

Welcome to the Jenkins 2.346.3 Setup Wizard

The Setup Wizard will install Jenkins 2.346.3 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

2. Click on next.


Jenkins 2.346.3 Setup

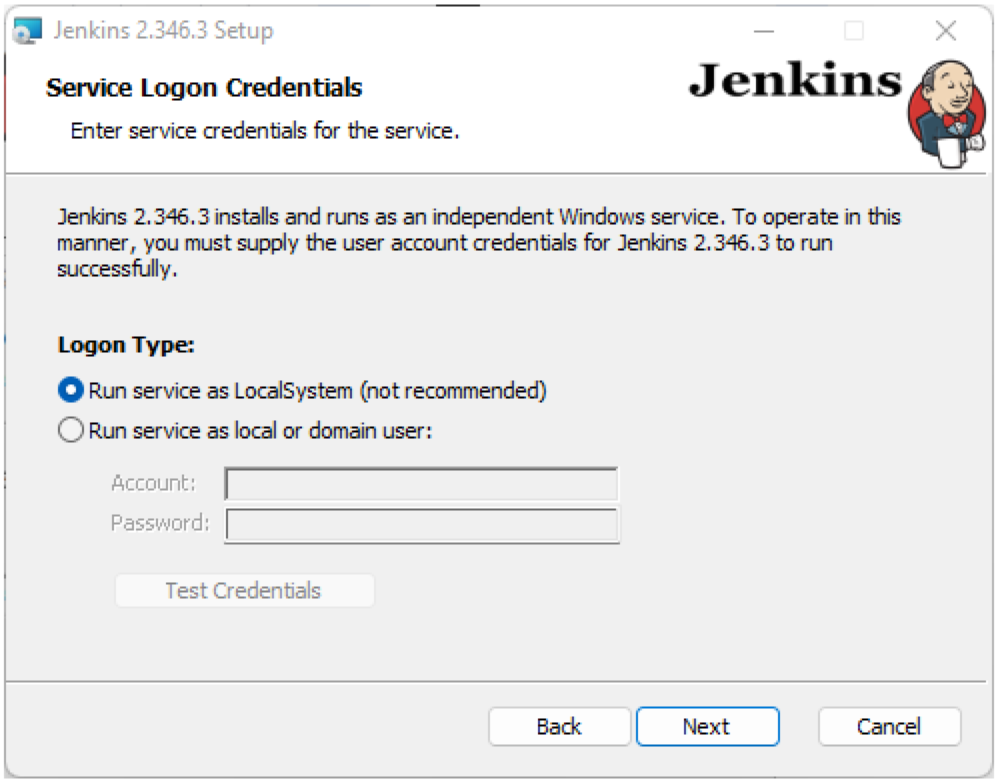
Destination Folder

Click Next to install to the default folder or click Change to choose another.

Install Jenkins 2.346.3 to:

C:\Program Files\Jenkins\

3. Select "Run service as LocalSystem" and click next.


Jenkins 2.346.3 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.346.3 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.346.3 to run successfully.

Logon Type:

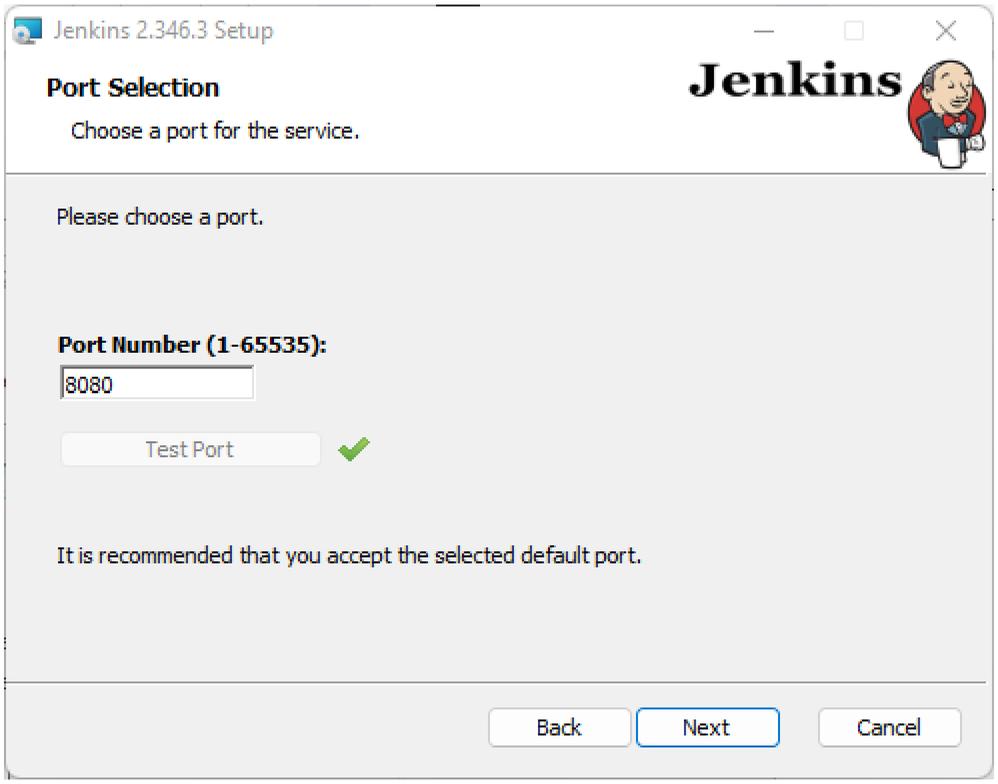
[✓ Run service as LocalSystem (not recommended)]

Run service as local or domain user:

Account:

Password:

4. Click on "Test Port" and click on next.


Jenkins 2.346.3 Setup

Port Selection

Choose a port for the service.

Please choose a port.

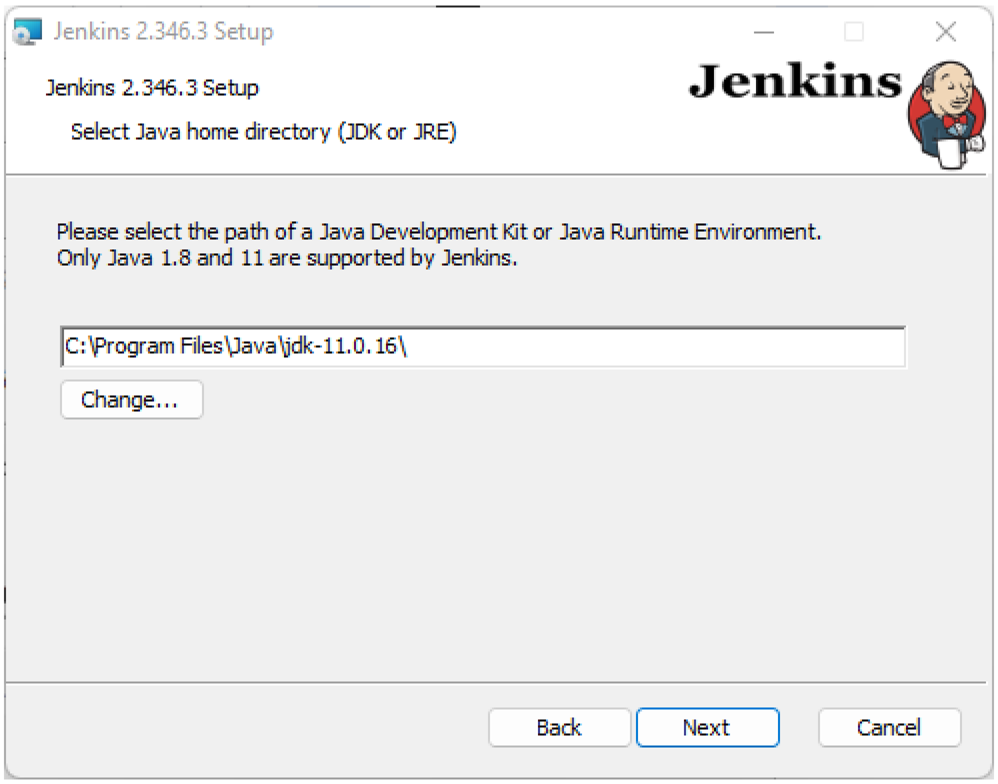
Port Number (1-65535):

8080

Test Port ✓

It is recommended that you accept the selected default port.

5. Locate the jdk 11 folder and click on next.


Jenkins 2.346.3 Setup

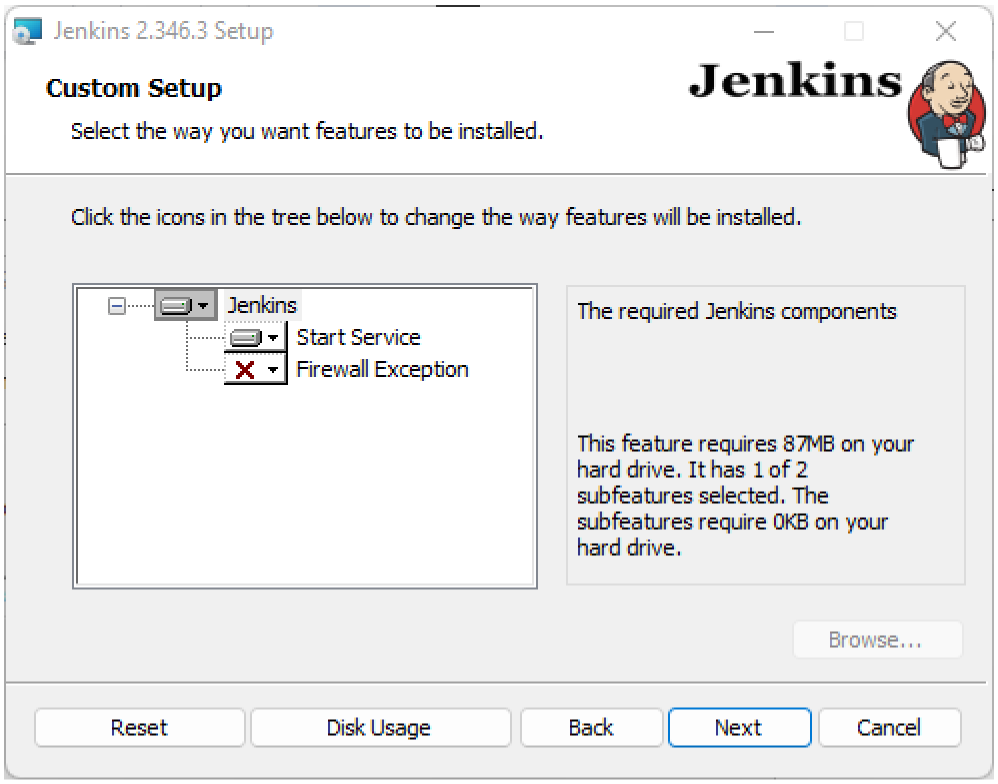
Jenkins 2.346.3 Setup

Select Java home directory (JDK or JRE)

Please select the path of a Java Development Kit or Java Runtime Environment.
Only Java 1.8 and 11 are supported by Jenkins.

C:\Program Files\Java\jdk-11.0.16\

6. Click on next.


Jenkins 2.346.3 Setup

Custom Setup

Select the way you want features to be installed.

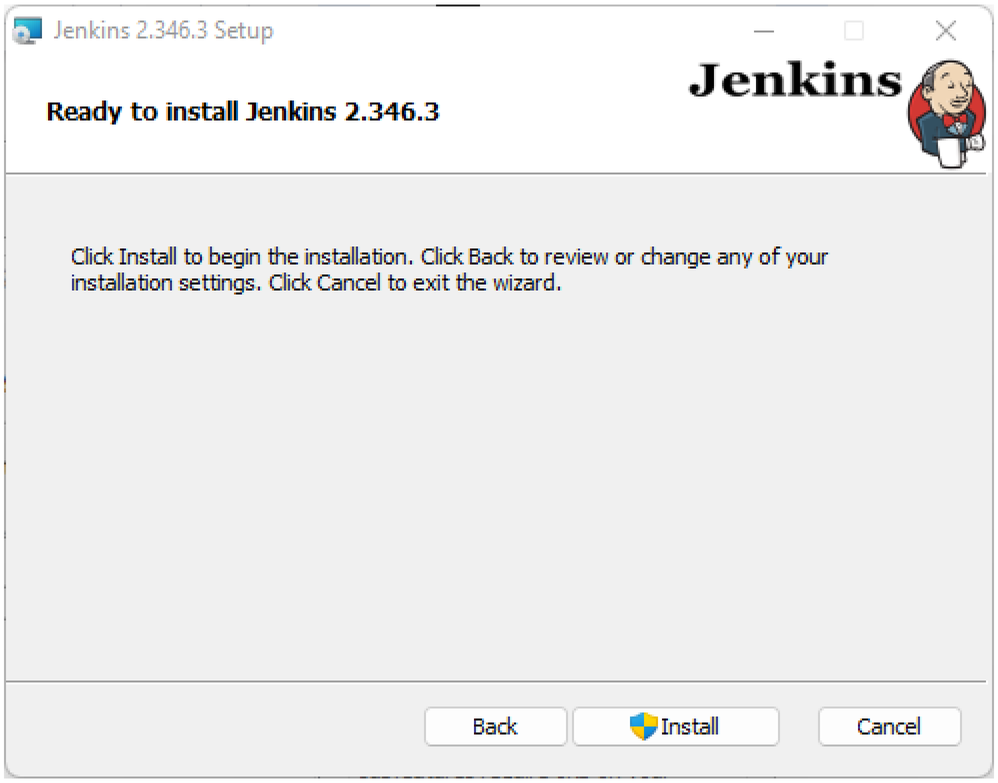
Click the icons in the tree below to change the way features will be installed.

Jenkins
Start Service
Firewall Exception

The required Jenkins components

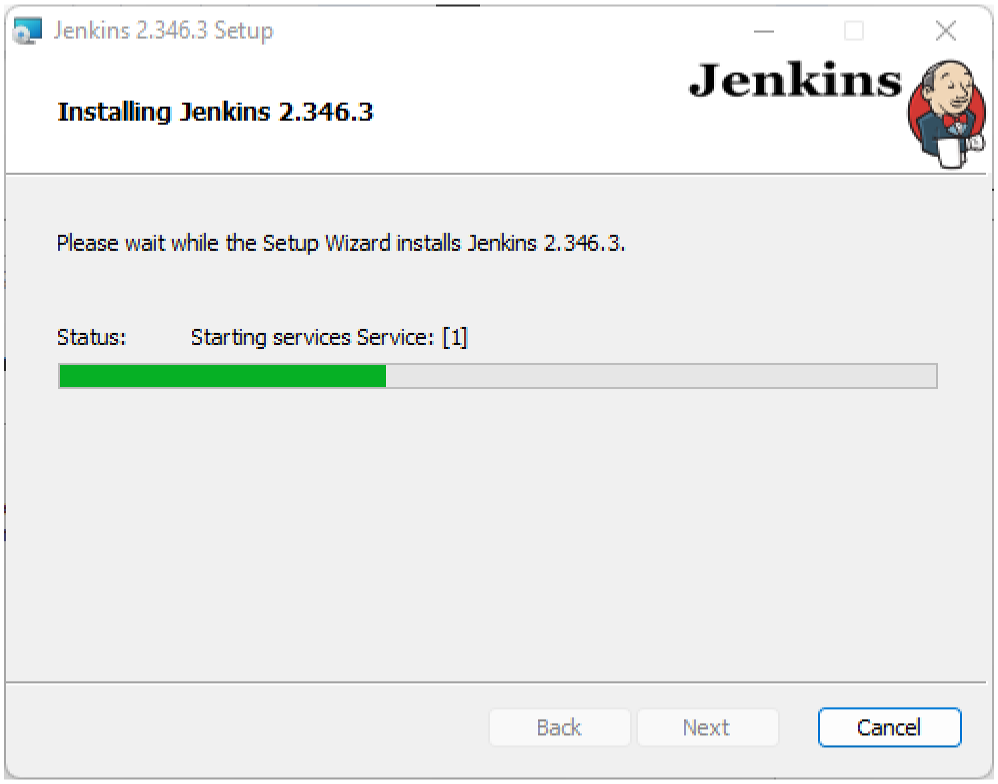
This feature requires 87MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.

7. Click on install.



Ready to install Jenkins 2.346.3

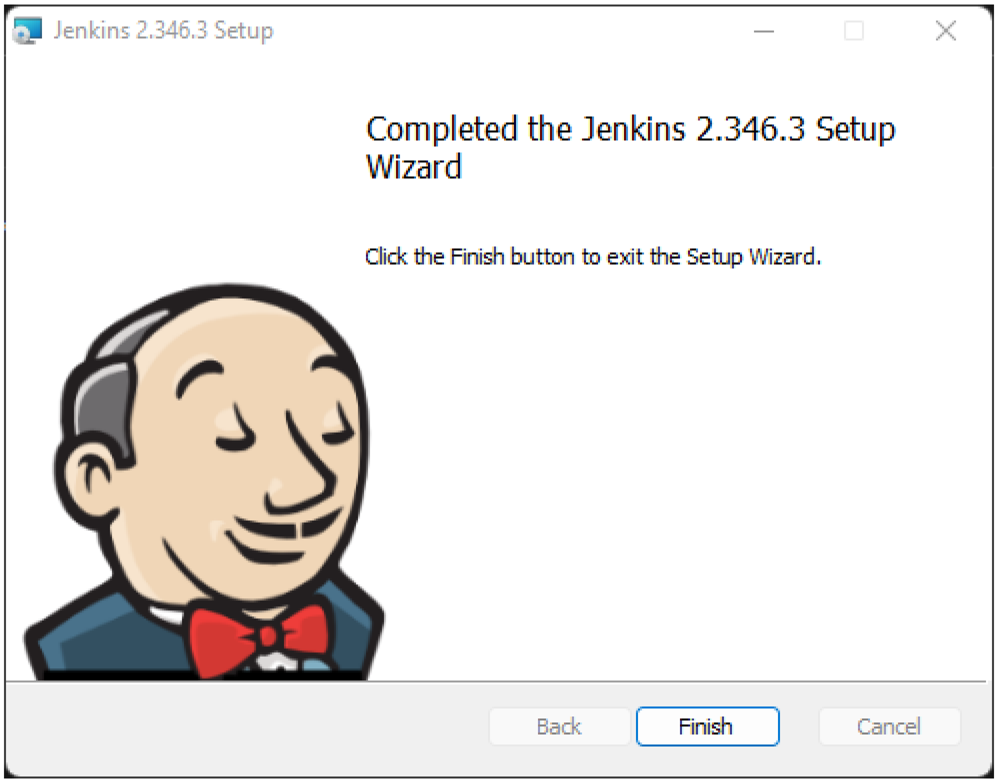
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.



Installing Jenkins 2.346.3

Please wait while the Setup Wizard installs Jenkins 2.346.3.

Status: Starting services Service: [1]

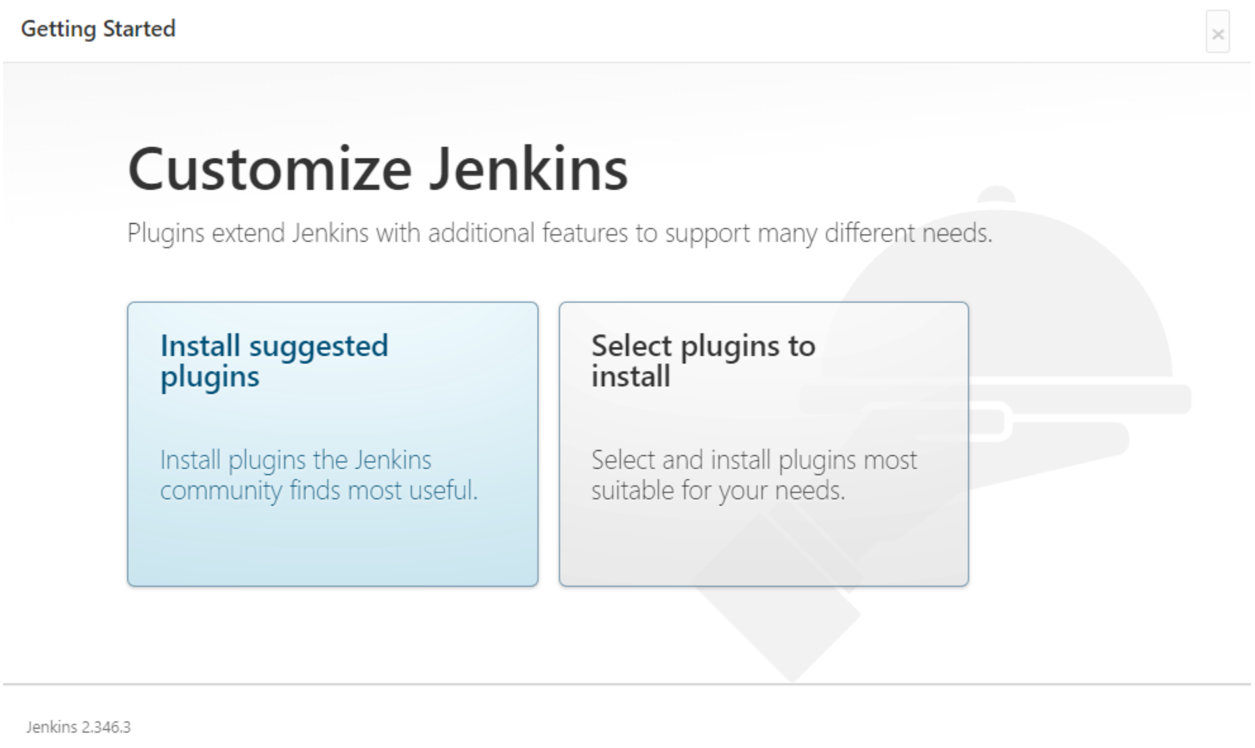

Completed the Jenkins 2.346.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.

8. Jenkins is now installed on your pc.

Build Job:

1. Start Jenkins by searching localhost:8080 on your search engine's search bar. Select install suggested plugins.


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

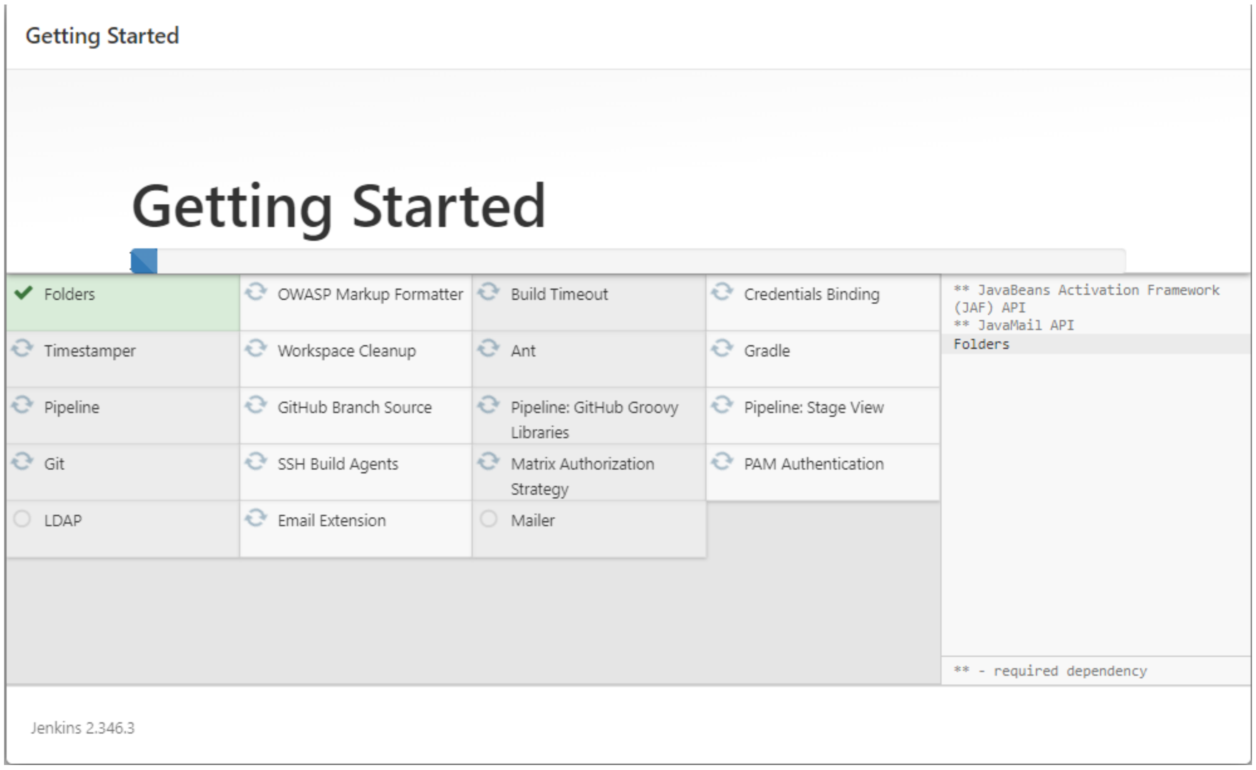
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.346.3


Getting Started

Getting Started

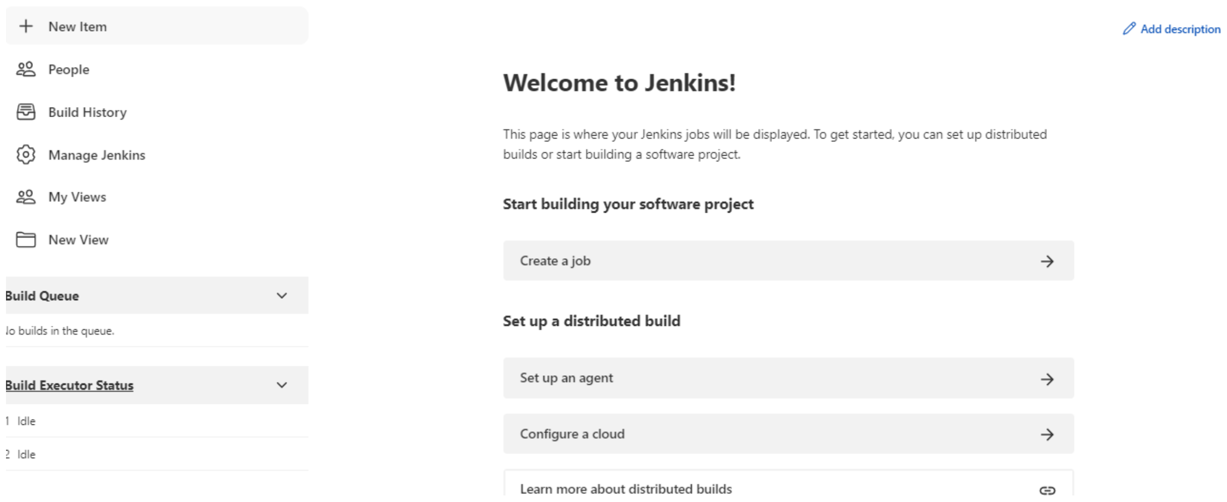
✓ Folders	OWASP Markup Formatter	Build Timeout	Credentials Binding
Timestamper	Workspace Cleanup	Ant	Gradle
Pipeline	GitHub Branch Source	Pipeline: GitHub Groovy Libraries	Pipeline: Stage View
Git	SSH Build Agents	Matrix Authorization Strategy	PAM Authentication
LDAP	Email Extension	Mailer	

```
** JavaBeans Activation Framework
(JAF) API
** JavaMail API
Folders

** - required dependency
```

Jenkins 2.346.3

2. After all the plugins are installed we start using Jenkins.



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

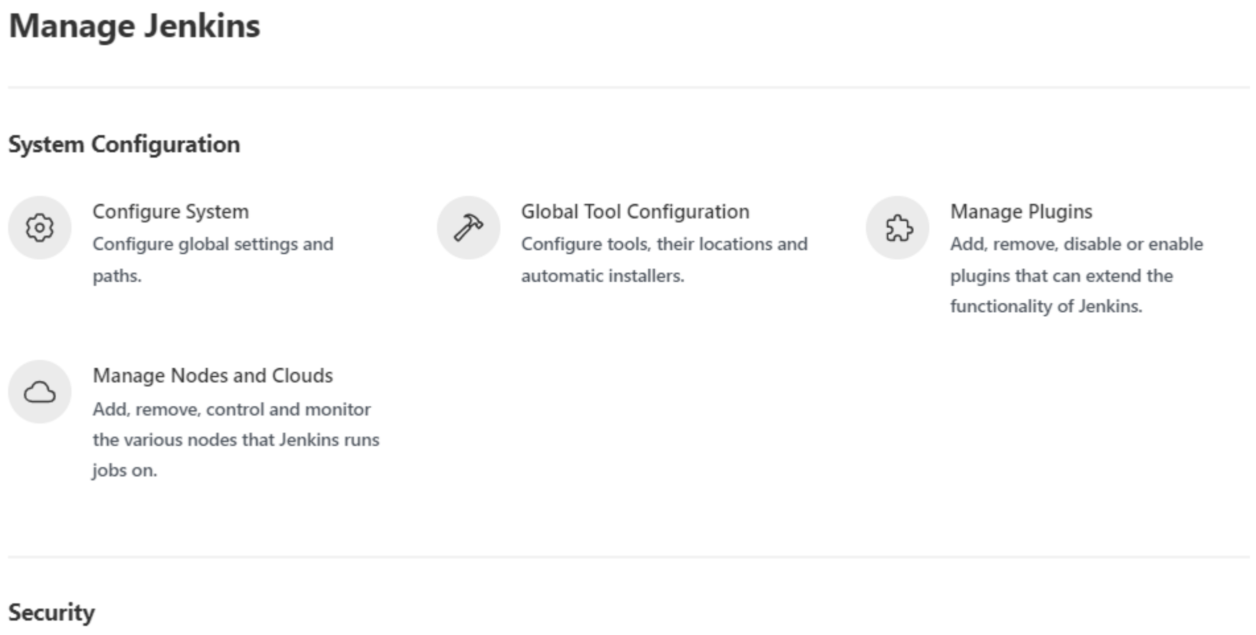
Set up a distributed build

lo builds in the queue.

1 Idle

2 Idle

3. Click on Mange Jenkins.



Manage Jenkins

System Configuration

Configure System
Configure global settings and paths.

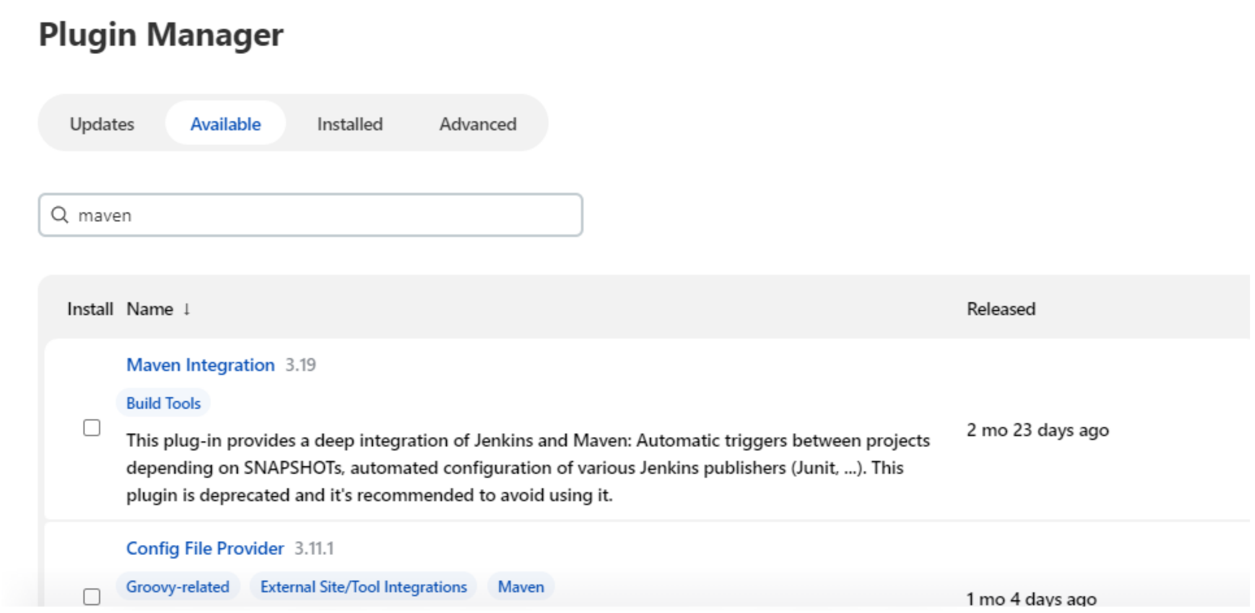
Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

4. Click on Manage Plugins.



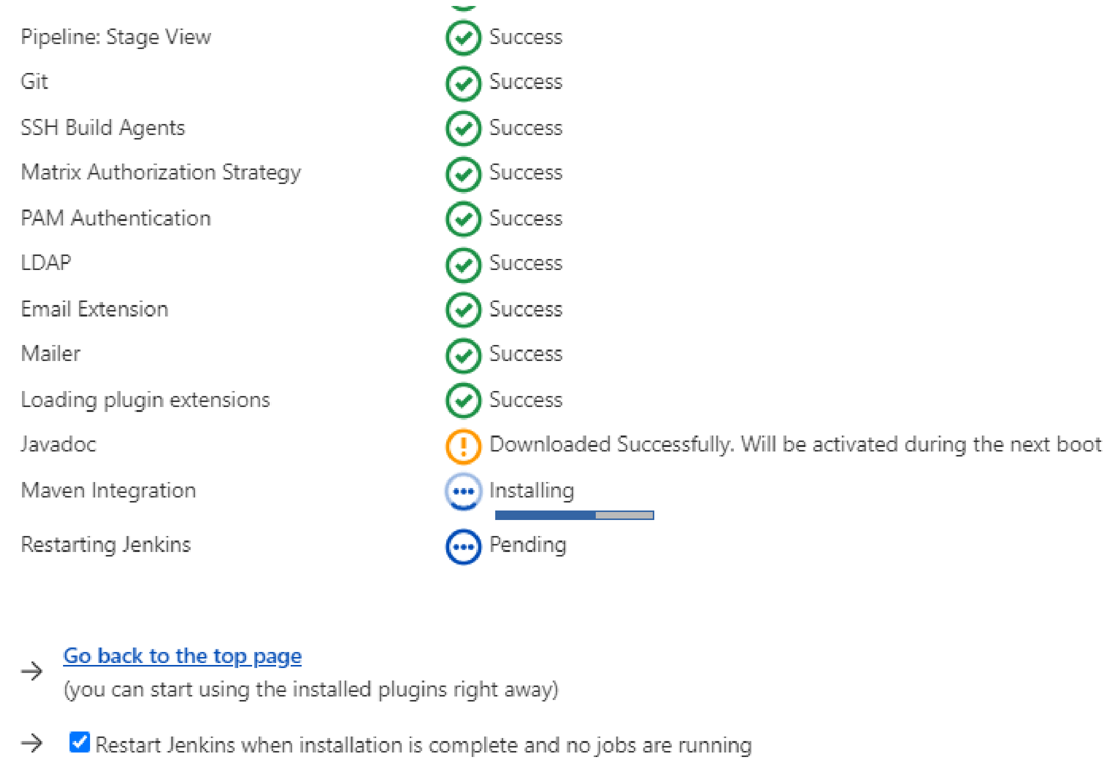
Plugin Manager

Updates | Available | Installed | Advanced

maven

Install	Name ↓	Released
☐	Maven Integration 3.19 Build Tools 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.	2 mo 23 days ago
☐	Config File Provider 3.11.1 Groovy-related External Site/Tool Integrations Maven	1 mo 4 days ago

5. Download Maven integration.

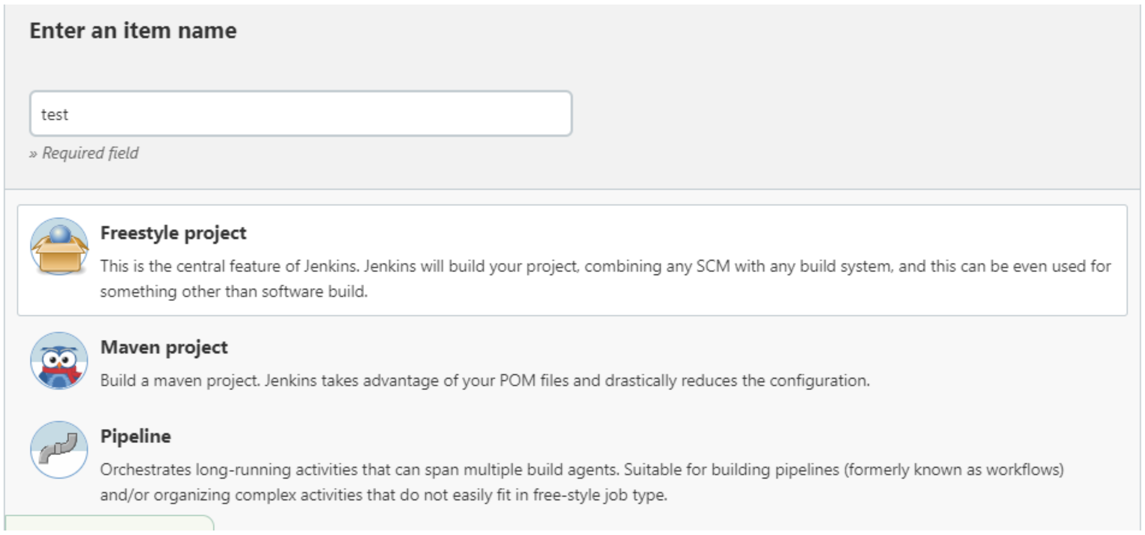


Pipeline: Stage View	✓ Success
Git	✓ Success
SSH Build Agents	✓ Success
Matrix Authorization Strategy	✓ Success
PAM Authentication	✓ Success
LDAP	✓ Success
Email Extension	✓ Success
Mailer	✓ Success
Loading plugin extensions	✓ Success
Javadoc	! Downloaded Successfully. Will be activated during the next boot
Maven Integration	Installing
Restarting Jenkins	Pending

→ 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

6. Start a new freestyle project.



Enter an item name

test

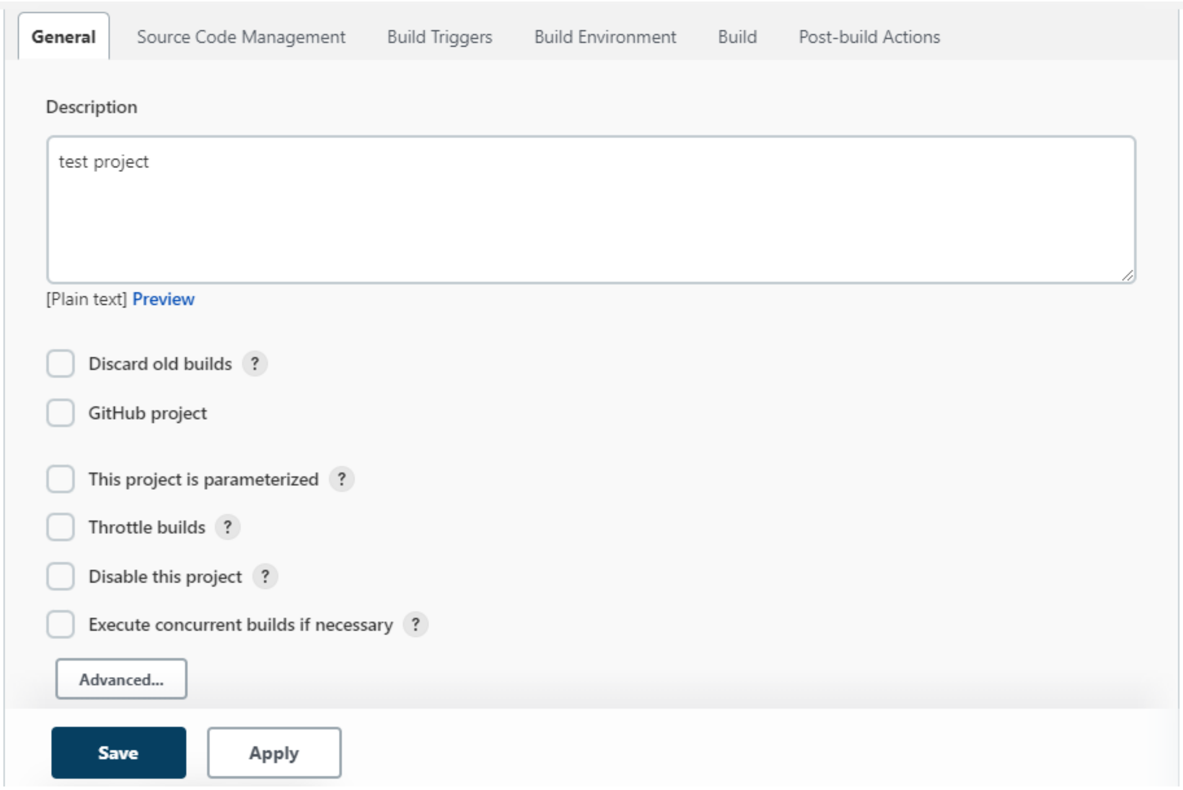
» Required field

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.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

7. Give a name and description as you want.



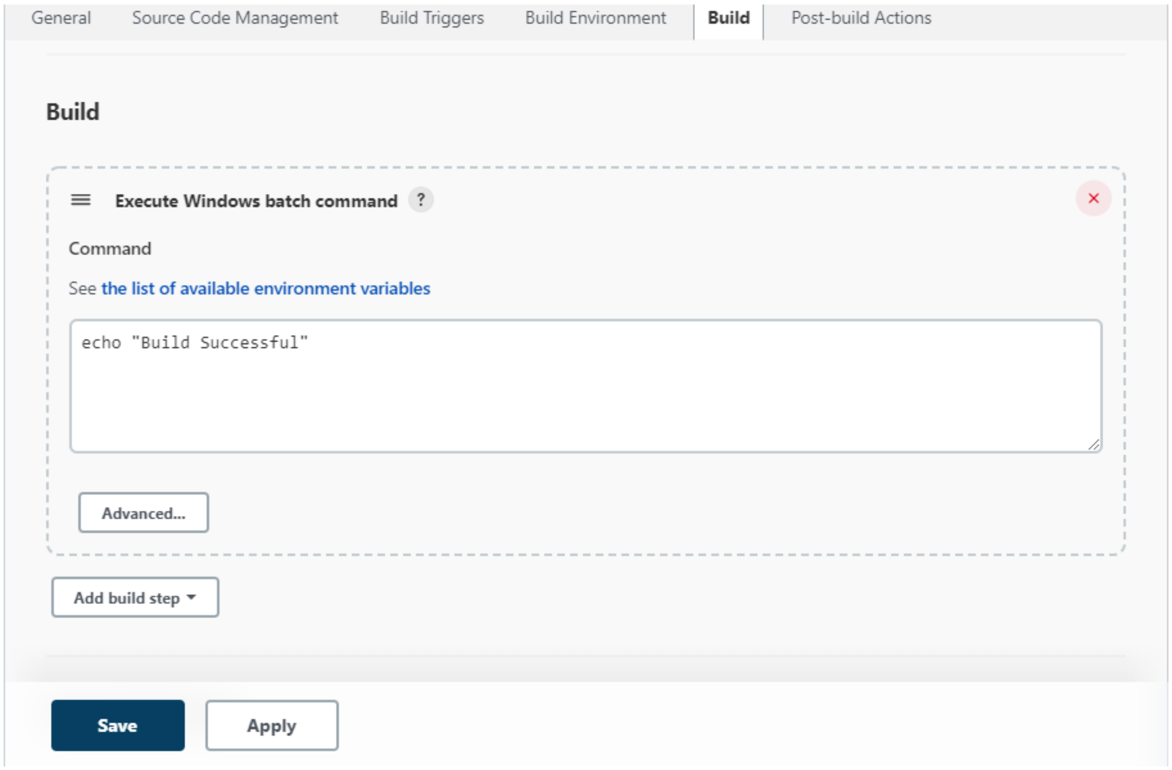
Description

test project

[Plain text] Preview

- [ ] Discard old builds
- [ ] GitHub project
- [ ] This project is parameterized
- [ ] Throttle builds
- [ ] Disable this project
- [ ] Execute concurrent builds if necessary

8. Go to the build stage, select “Execute in Windows Batch” and write echo “Build successful”.



Build

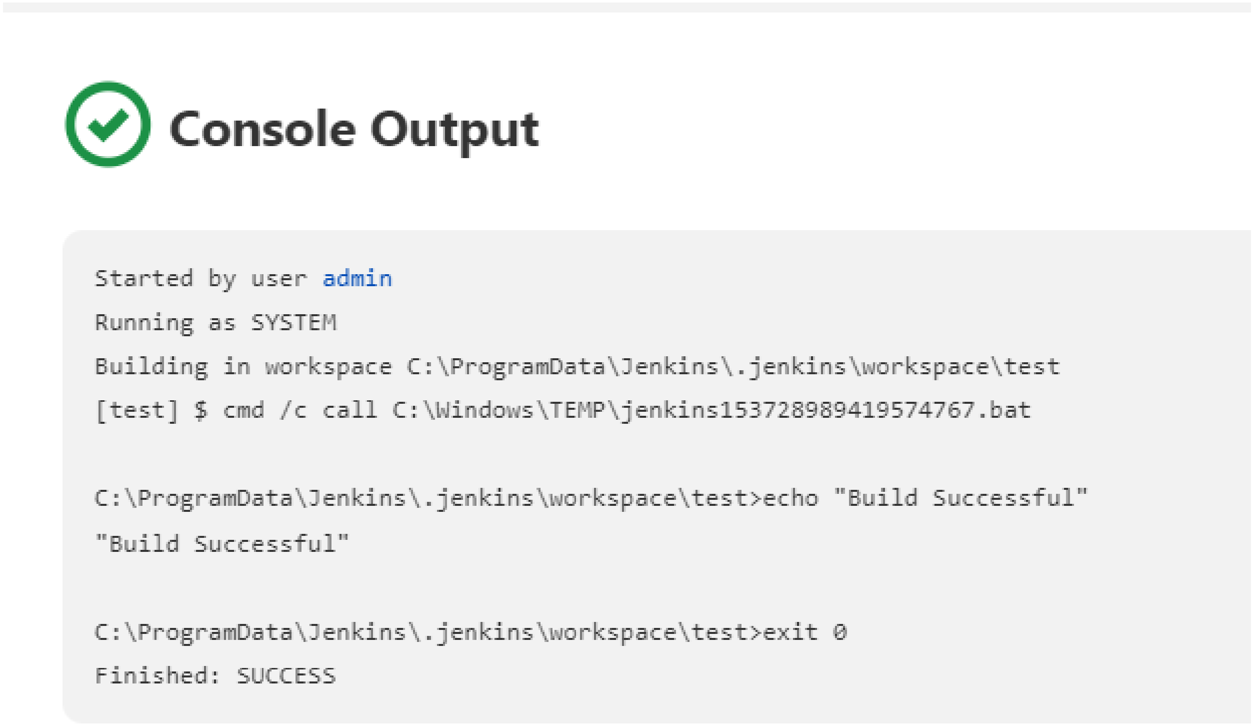
Execute Windows batch command

Command

See the list of available environment variables

```
echo "Build Successful"
```

9. Now click on build and check the console output. If you see the Build Successful message then we are successful.



Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\test
[test] $ cmd /c call C:\Windows\TEMP\jenkins153728989419574767.bat

C:\ProgramData\Jenkins\.jenkins\workspace\test>echo "Build Successful"
"Build Successful"

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

Conclusion:

Thus we installed and configured Jenkins with Maven/Ant/Gradle to set up a build Job.