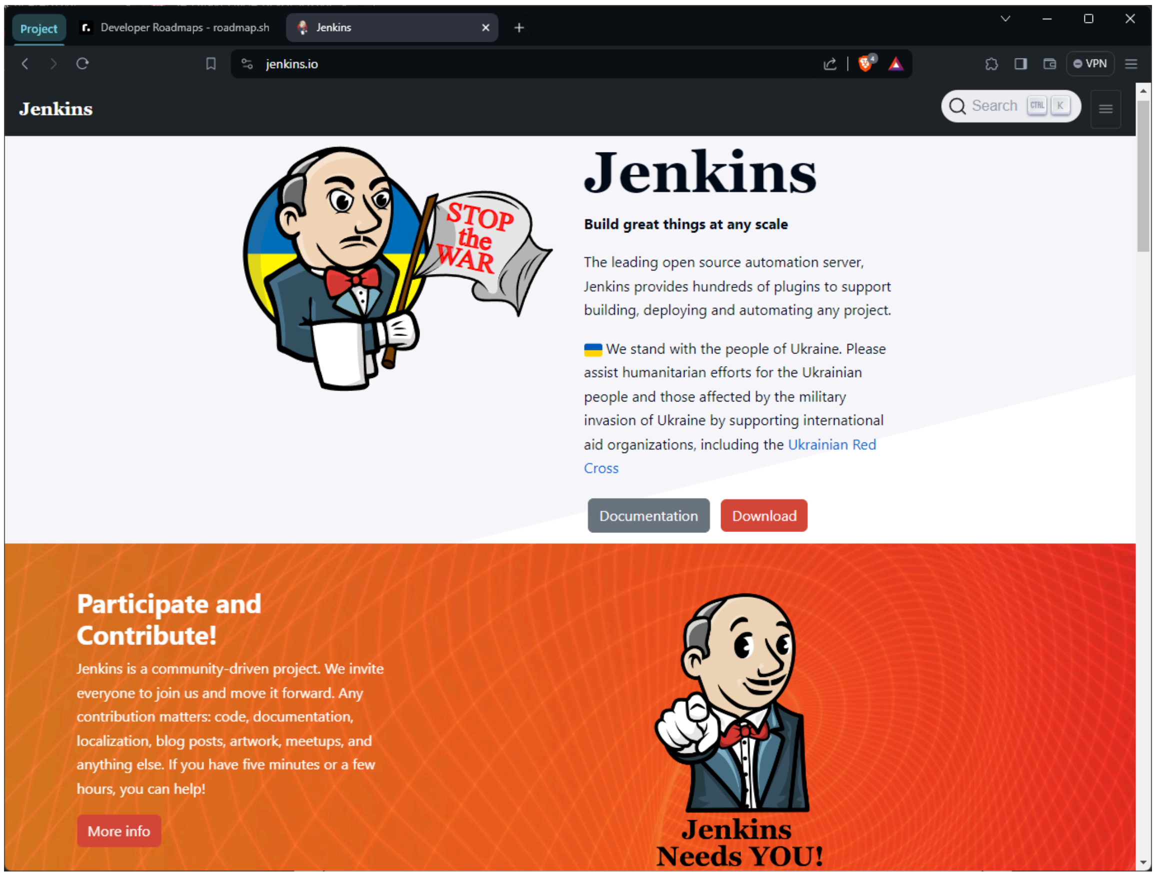
Name of Student: Pushkar Sane		
Roll Number: 45	**Lab Assignment Number: 4**	
Title of Lab Assignment: To deploy and test Java/web/Python application on Jenkins server.		
DOP: 06-02-2024	**DOS: 09-02-2024**	
CO Mapped: **CO3**	**PO Mapped:** **PO2, PO3, PO5, PSO1, PSO2**	**Signature:**

Practical No. 4

Aim: To deploy and test Java maven application on Jenkins server.

Steps to deploy and test the maven application on the Jenkins:

1. **Install and Set Up Jenkins:**
    - Install Jenkins on your server by downloading the Jenkins WAR file from the official website or using a package manager.
    - Start Jenkins by running the command `java -jar jenkins.war` in the terminal or by launching it as a service.
    - Access the Jenkins dashboard by opening a web browser and navigating to `http://localhost:8080` (replace `localhost` with your server's IP address if accessing remotely).



Jenkins

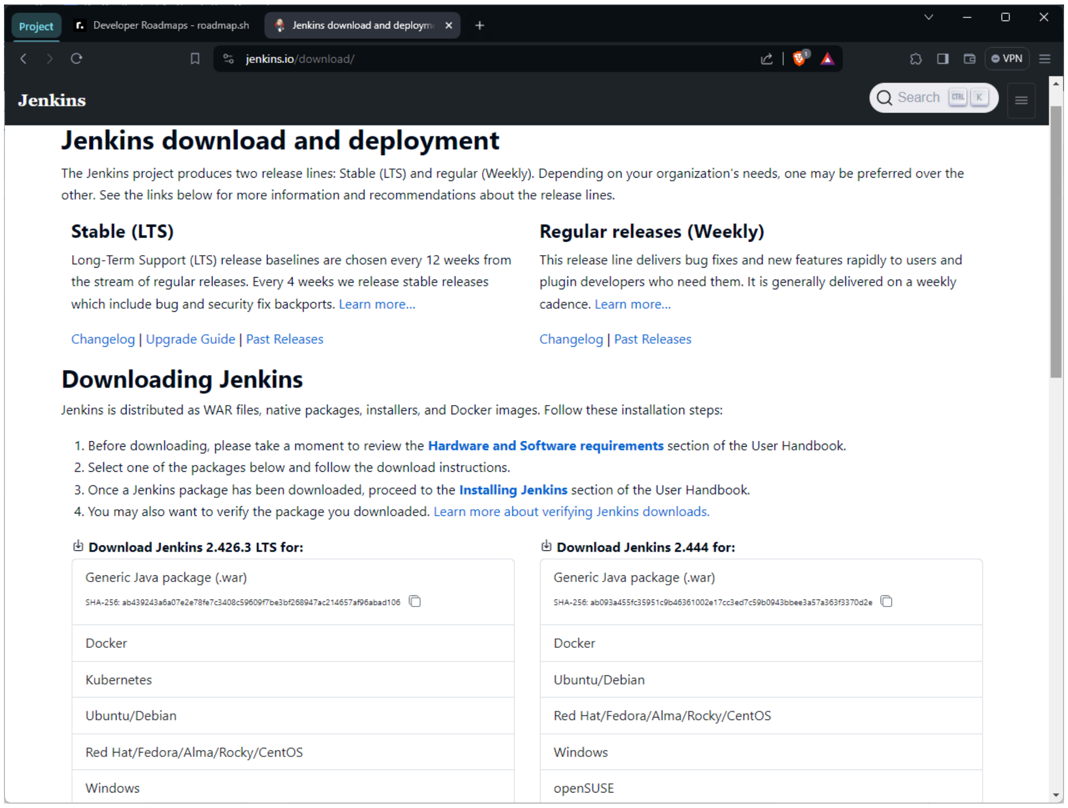
Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

We stand with the people of Ukraine. Please assist humanitarian efforts for the Ukrainian people and those affected by the military invasion of Ukraine by supporting international aid organizations, including the Ukrainian Red Cross

Participate and Contribute!

Jenkins is a community-driven project. We invite everyone to join us and move it forward. Any contribution matters: code, documentation, localization, blog posts, artwork, meetups, and anything else. If you have five minutes or a few hours, you can help!


Project
Developer Roadmaps - roadmap.sh
Jenkins download and deployment
jenkins.io/download/
Jenkins
Search

Jenkins download and deployment

The Jenkins project produces two release lines: Stable (LTS) and regular (Weekly). Depending on your organization's needs, one may be preferred over the other. See the links below for more information and recommendations about the release lines.

Stable (LTS)

Long-Term Support (LTS) release baselines are chosen every 12 weeks from the stream of regular releases. Every 4 weeks we release stable releases which include bug and security fix backports. Learn more...

Changelog | Upgrade Guide | Past Releases

Regular releases (Weekly)

This release line delivers bug fixes and new features rapidly to users and plugin developers who need them. It is generally delivered on a weekly cadence. Learn more...

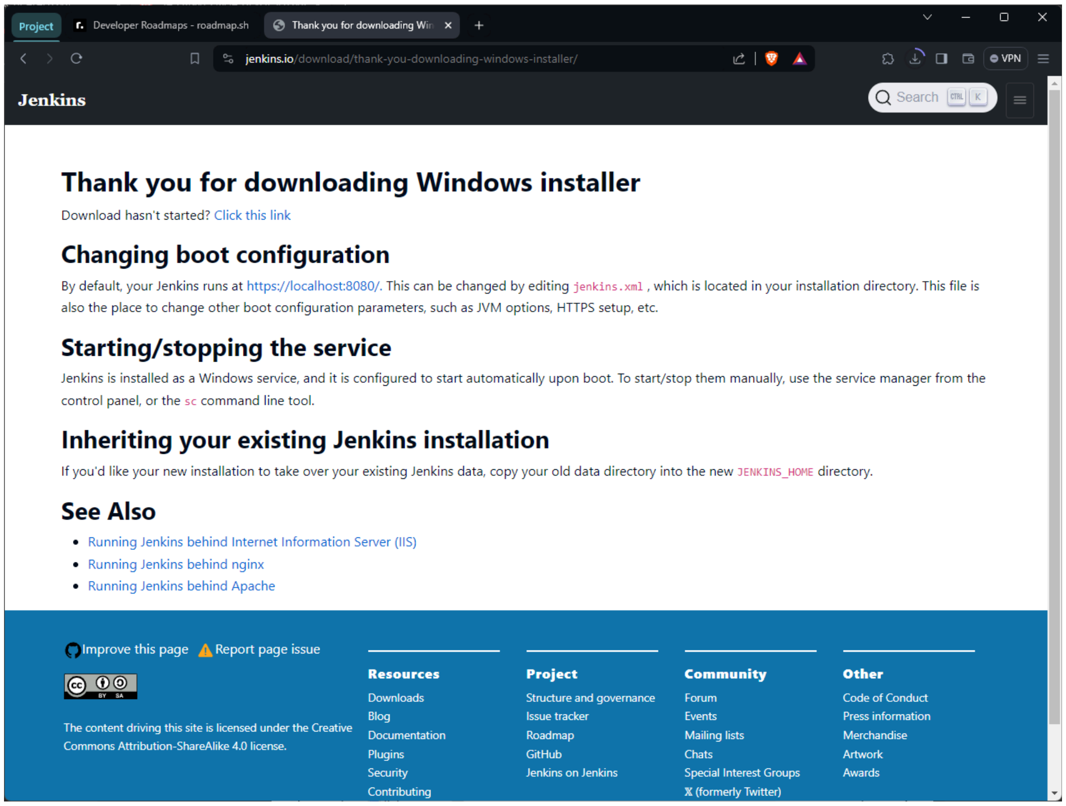
Changelog | Past Releases

Downloading Jenkins

Jenkins is distributed as WAR files, native packages, installers, and Docker images. Follow these installation steps:

1. Before downloading, please take a moment to review the **Hardware and Software requirements** section of the User Handbook.
2. Select one of the packages below and follow the download instructions.
3. Once a Jenkins package has been downloaded, proceed to the **Installing Jenkins** section of the User Handbook.
4. You may also want to verify the package you downloaded. Learn more about verifying Jenkins downloads.

Download Jenkins 2.426.3 LTS for:	Download Jenkins 2.444 for:
Generic Java package (.war) SHA-256: ab439243a6a07e2e78fe7c3408c59609f7be3bf268947ac214657af96abad106	Generic Java package (.war) SHA-256: ab093a455fc35951c9b46361002e17cc3ed7c59b0943bbee3a57a363f3370d2e
Docker	Docker
Kubernetes	Ubuntu/Debian
Ubuntu/Debian	Red Hat/Fedora/Alma/Rocky/CentOS
Red Hat/Fedora/Alma/Rocky/CentOS	Windows
Windows	openSUSE


Project
Developer Roadmaps - roadmap.sh
Thank you for downloading Win
jenkins.io/download/thank-you-downloading-windows-installer/
Jenkins
Search

Thank you for downloading Windows installer

Download hasn't started? Click this link

Changing boot configuration

By default, your Jenkins runs at https://localhost:8080/. This can be changed by editing `jenkins.xml`, which is located in your installation directory. This file is also the place to change other boot configuration parameters, such as JVM options, HTTPS setup, etc.

Starting/stopping the service

Jenkins is installed as a Windows service, and it is configured to start automatically upon boot. To start/stop them manually, use the service manager from the control panel, or the `sc` command line tool.

Inheriting your existing Jenkins installation

If you'd like your new installation to take over your existing Jenkins data, copy your old data directory into the new `JENKINS_HOME` directory.

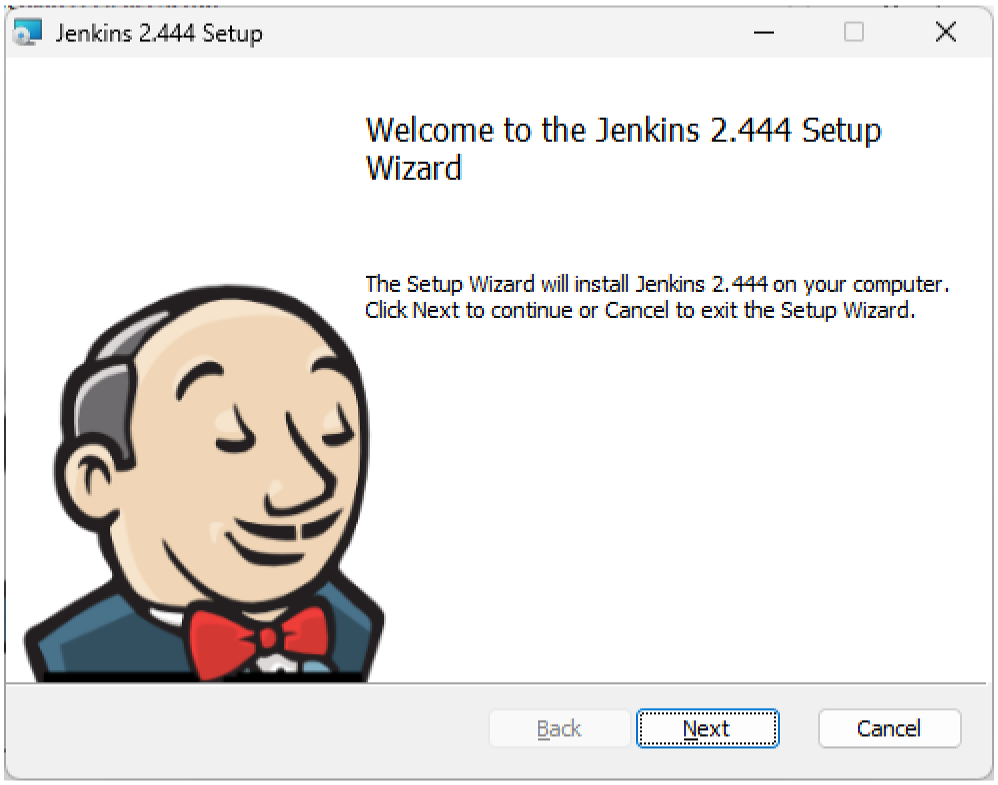
See Also

- Running Jenkins behind Internet Information Server (IIS)
- Running Jenkins behind nginx
- Running Jenkins behind Apache

Improve this page Report page issue

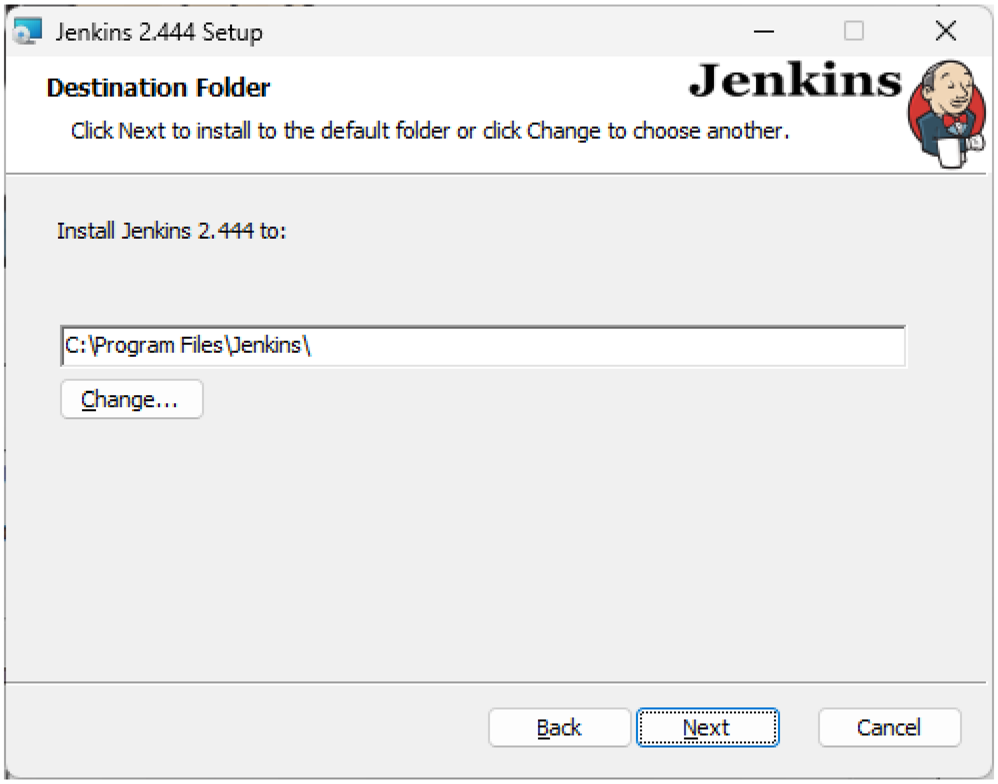
The content driving this site is licensed under the Creative Commons Attribution-ShareAlike 4.0 license.

Resources	Project	Community	Other
Downloads	Structure and governance	Forum	Code of Conduct
Blog	Issue tracker	Events	Press information
Documentation	Roadmap	Mailing lists	Merchandise
Plugins	GitHub	Chats	Artwork
Security	Jenkins on Jenkins	Special Interest Groups	Awards
Contributing		X (formerly Twitter)	



Welcome to the Jenkins 2.444 Setup Wizard

The Setup Wizard will install Jenkins 2.444 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

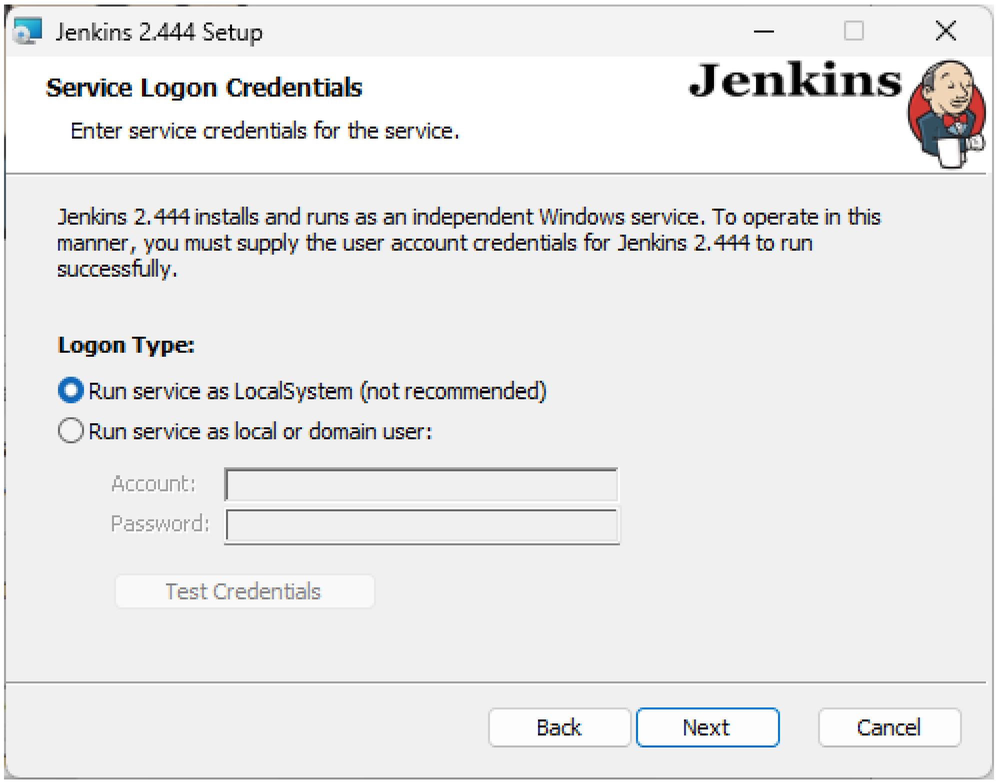


Destination Folder

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

Install Jenkins 2.444 to:

C:\Program Files\Jenkins\


Jenkins 2.444 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.444 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.444 to run successfully.

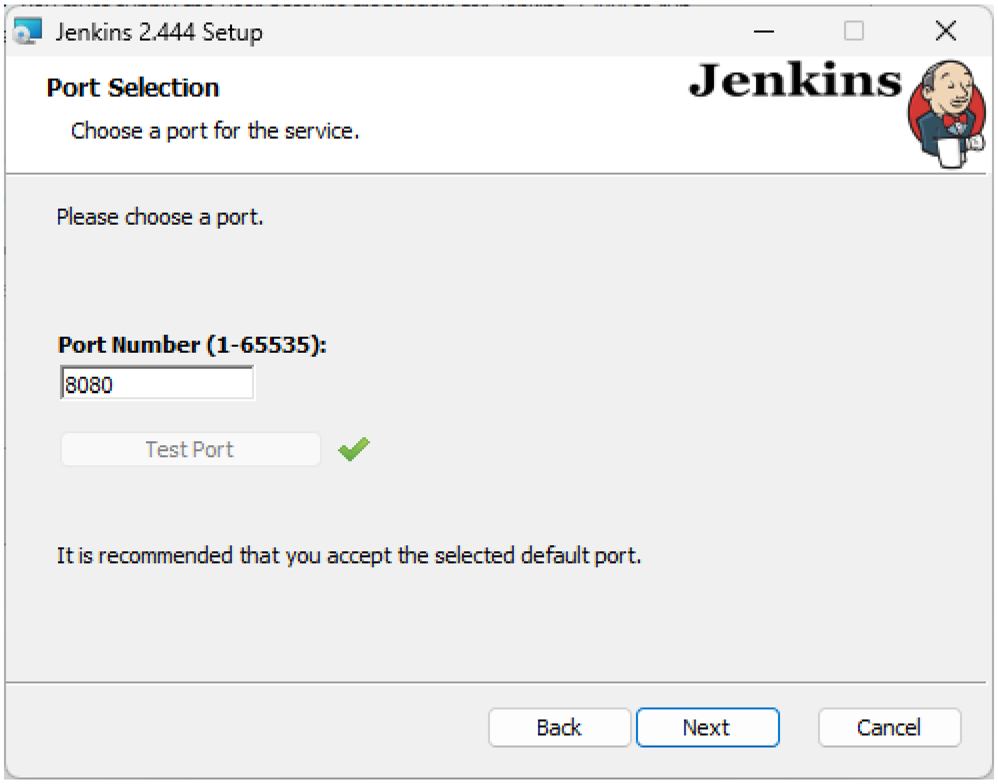
Logon Type:

[✓ Run service as LocalSystem (not recommended)]

Run service as local or domain user:

Account:

Password:


Jenkins 2.444 Setup

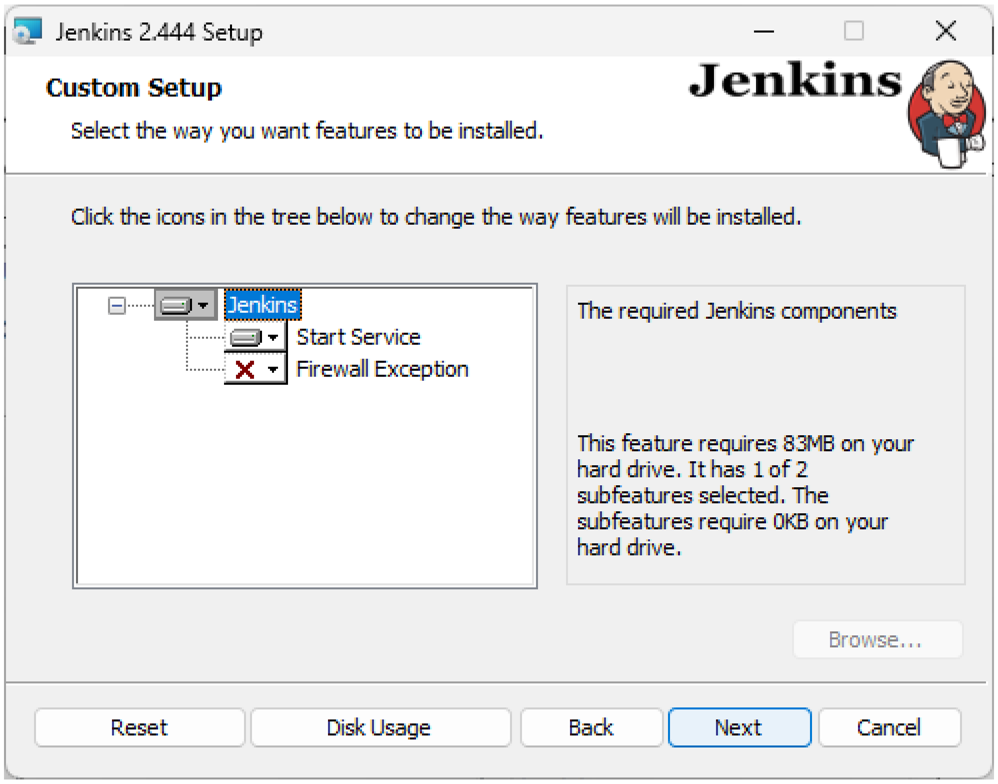
Port Selection

Choose a port for the service.

Please choose a port.

Port Number (1-65535): 8080

It is recommended that you accept the selected default port.


Jenkins 2.444 Setup

Jenkins

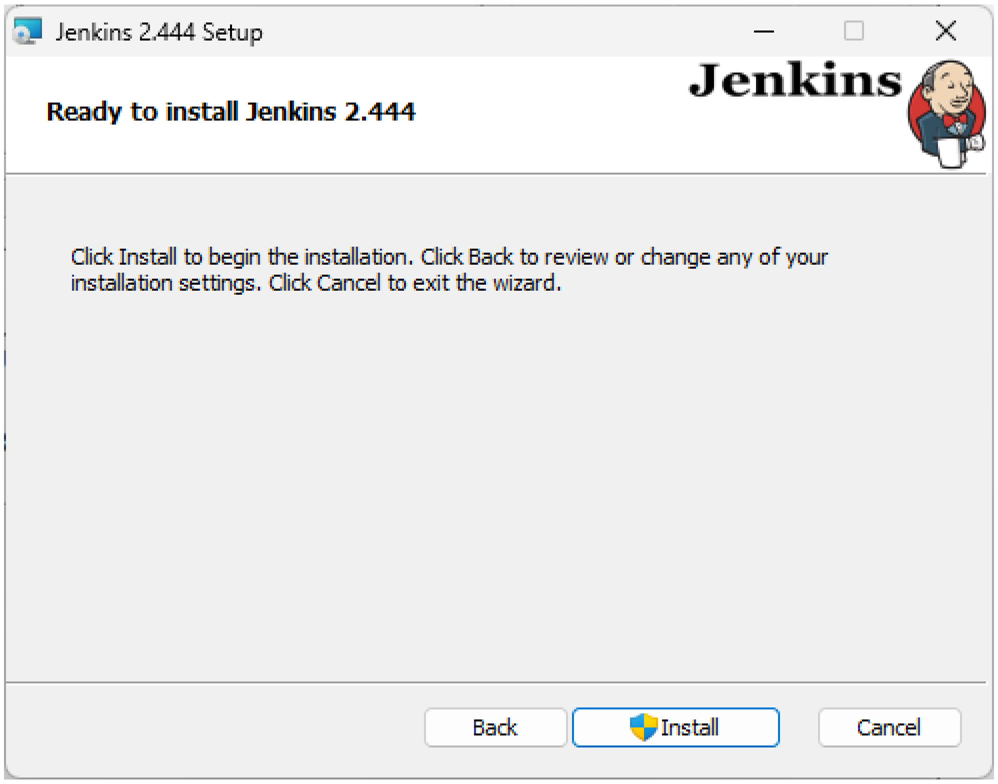
Custom Setup

Select the way you want features to be installed.

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

The required Jenkins components

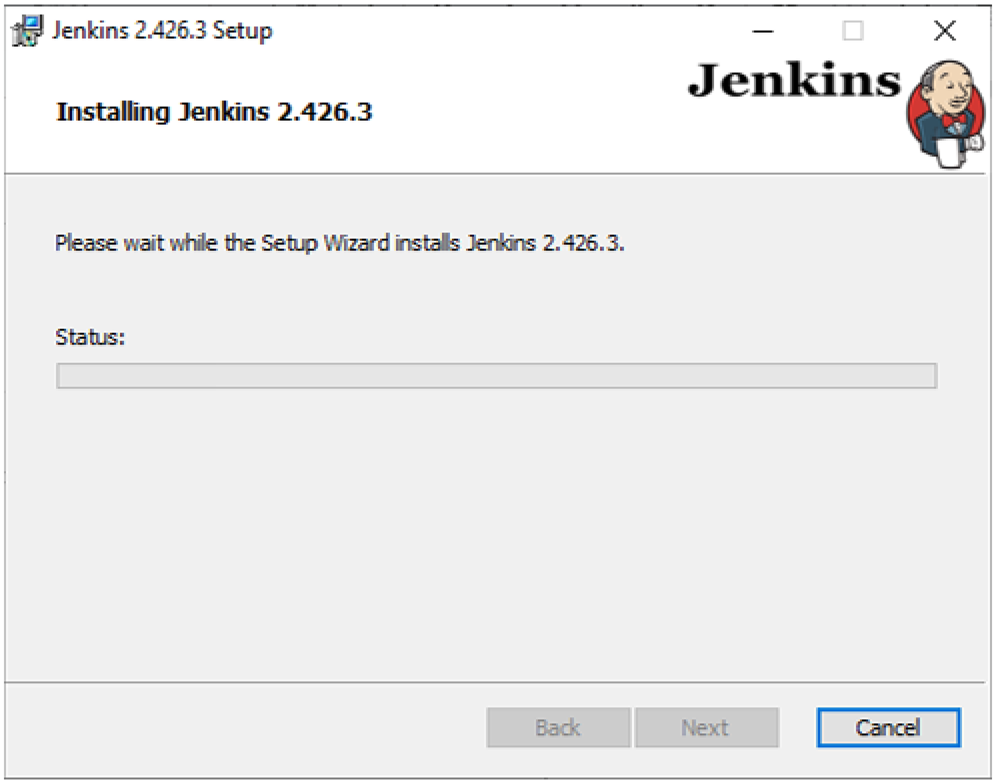
This feature requires 83MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.


Jenkins 2.444 Setup

Jenkins

Ready to install Jenkins 2.444

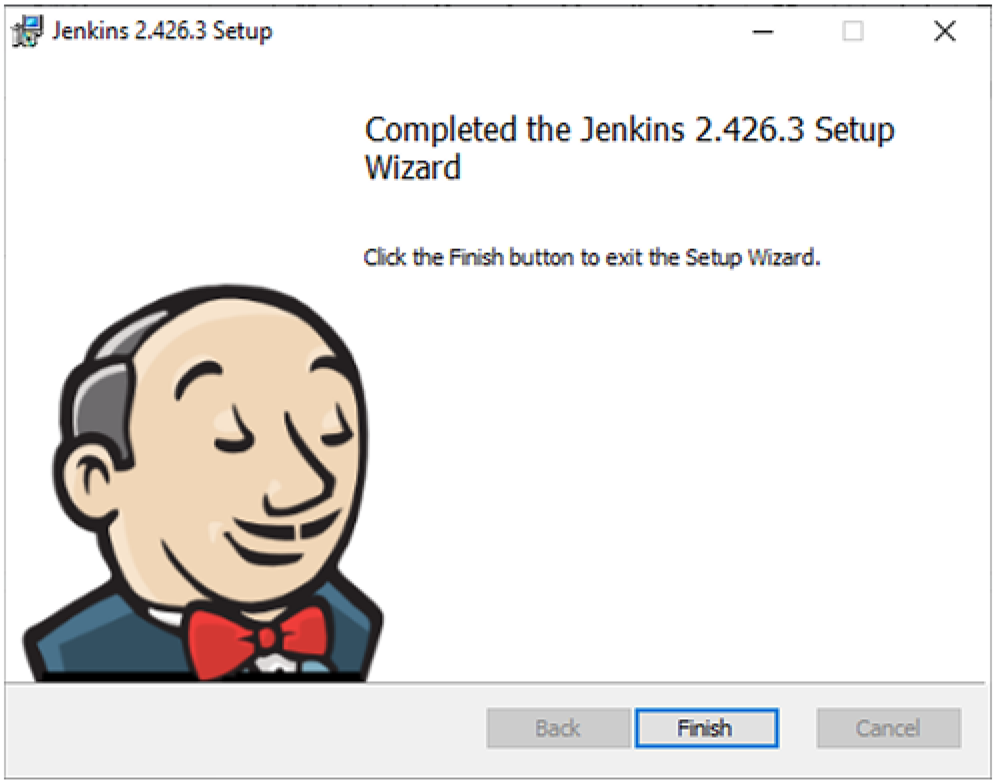
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.


Jenkins 2.426.3 Setup

Installing Jenkins 2.426.3

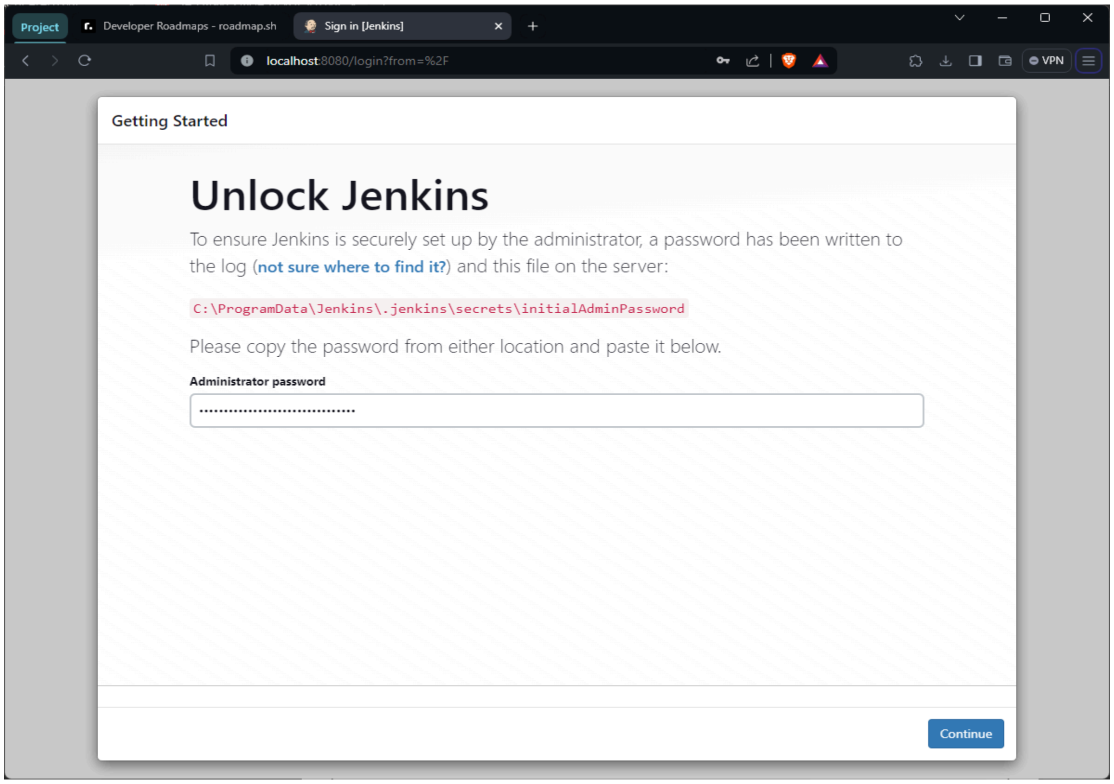
Please wait while the Setup Wizard installs Jenkins 2.426.3.

Status:


Jenkins 2.426.3 Setup

Completed the Jenkins 2.426.3 Setup Wizard

Click the Finish button to exit the Setup Wizard.


Project
Developer Roadmaps - roadmap.sh
Sign in [Jenkins]
localhost:8080/login?from=%2F
VPN

Getting Started

Unlock Jenkins

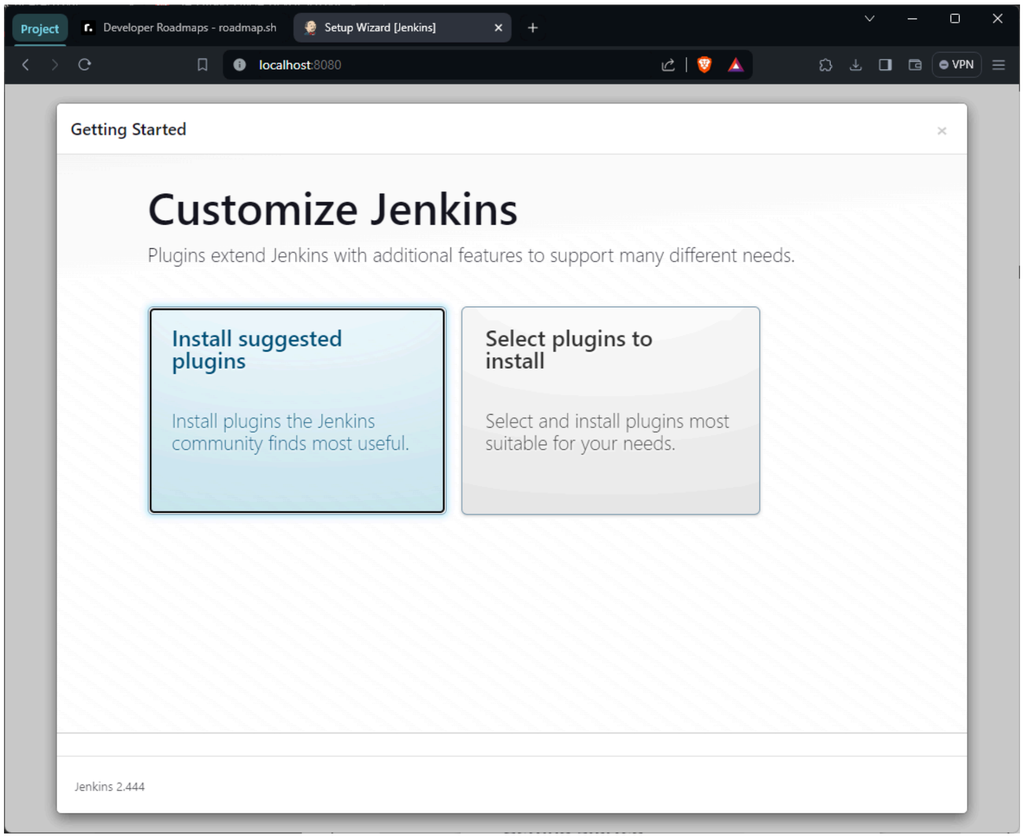
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

```
C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

································



Getting Started

Customize Jenkins

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

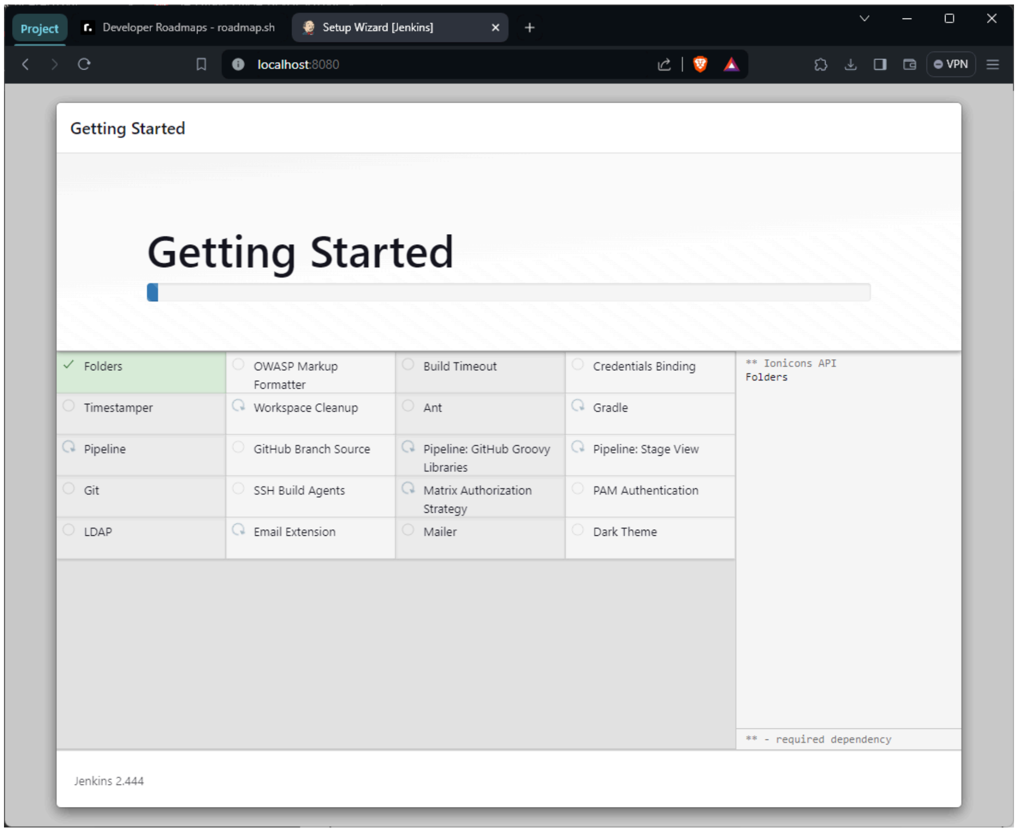
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.444



Getting Started

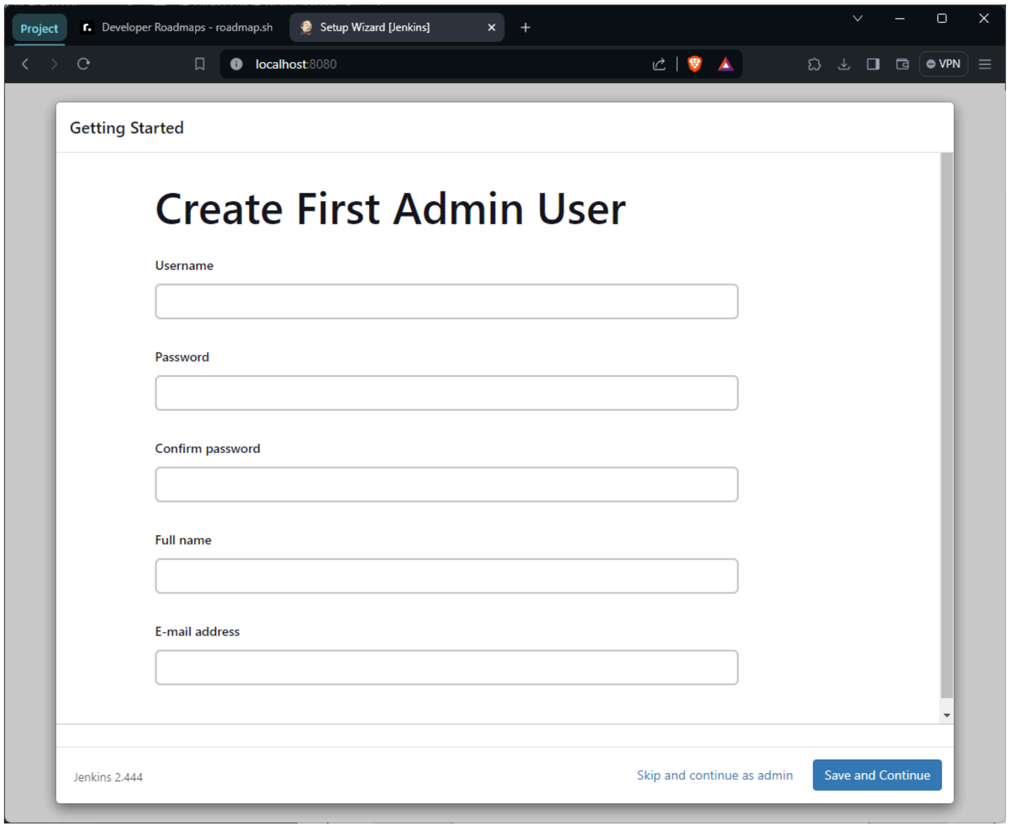
Getting Started

✓ 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	Dark Theme

```
** Ionicons API
Folders
```

```
** - required dependency
```

Jenkins 2.444



Getting Started

Create First Admin User

Username

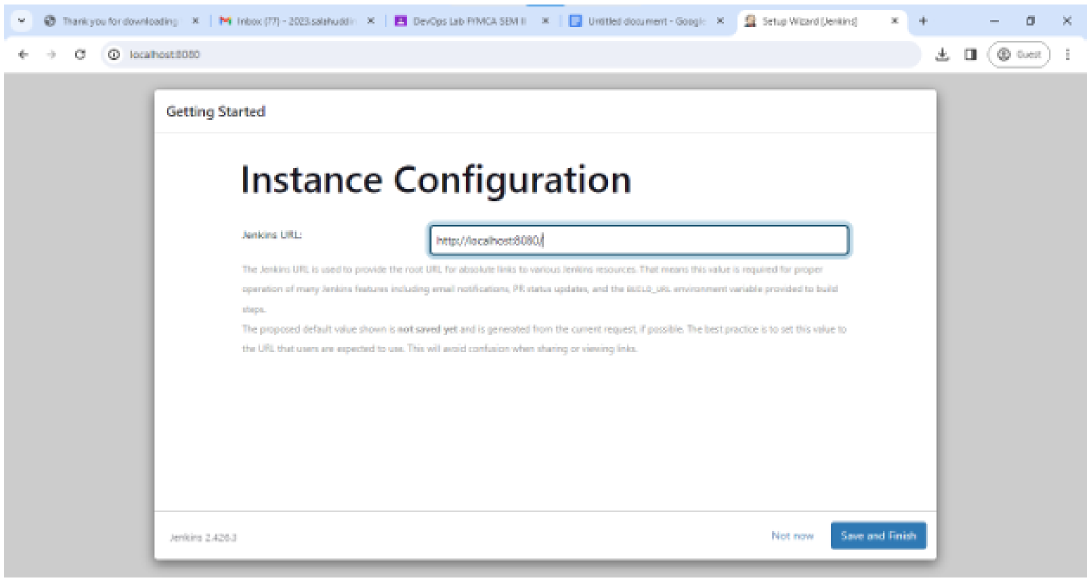
Password

Confirm password

Full name

E-mail address

Jenkins 2.444


Thank you for downloading
Inbox (77) - 2023.salahuddi
DevOps Lab F/MCA SEM II
Untitled document - Google
Setup Wizard [Jenkins]
localhost:8080
Guest

Getting Started

Instance Configuration

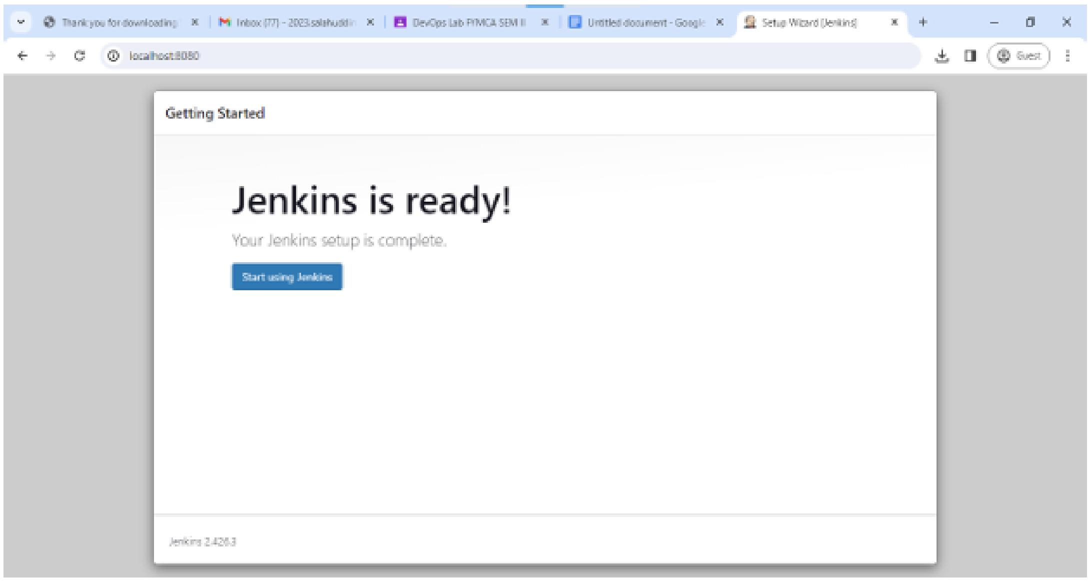
Jenkins URL: http://localhost:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.426.3

Not now Save and Finish


Thank you for downloading
Inbox (77) - 2023.salahuddi
DevOps Lab F/MCA SEM II
Untitled document - Google
Setup Wizard [Jenkins]
localhost:8080
Guest

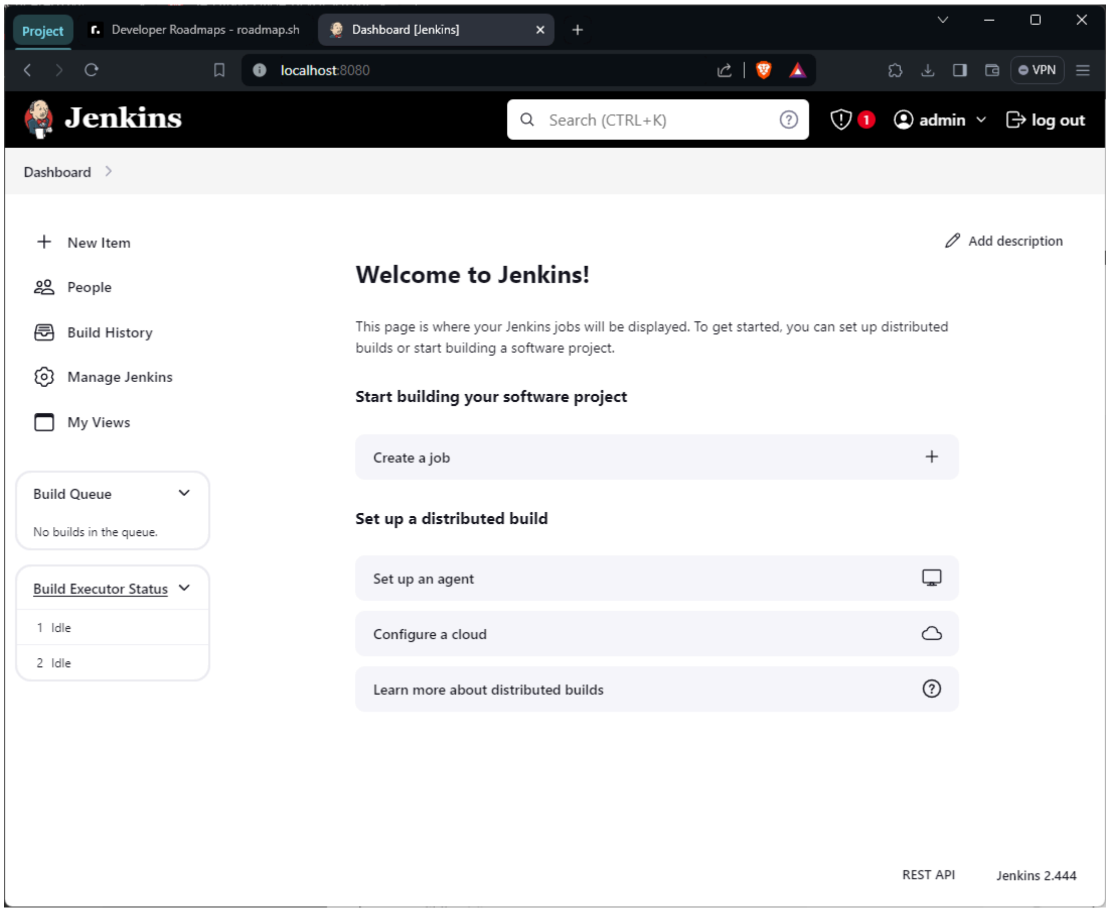
Getting Started

Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.426.3


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

Build Queue

No builds in the queue.

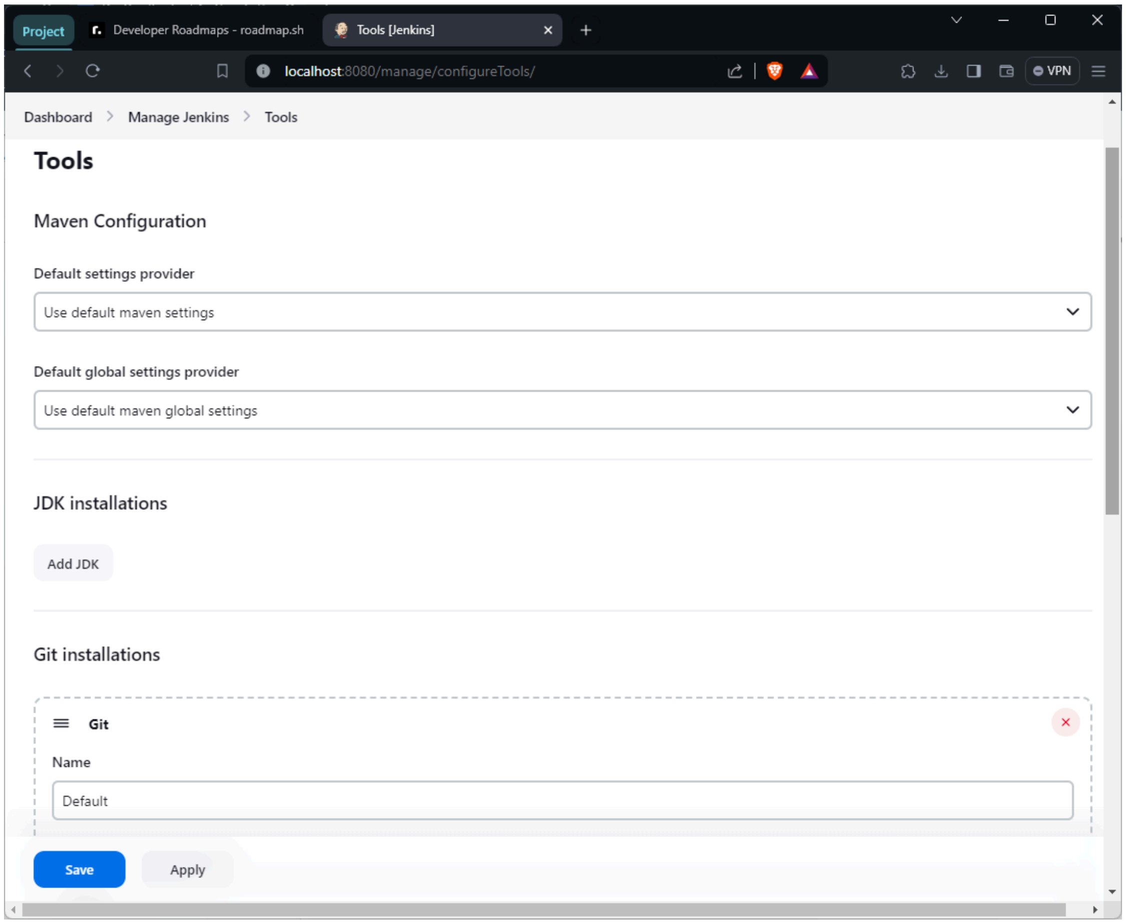
Build Executor Status

1 Idle

2 Idle

2. **Install Required Plugins: (If not installed)**
   - Once logged into Jenkins, click on "Manage Jenkins" from the left sidebar.
   - Select "Manage Plugins" and navigate to the "Available" tab.
   - Search for and install the "Maven Integration" plugin.

3. **Configure Global Tool Installations(Dashboard>Manage Jenkins> Tools):**
   - Scroll down to the "Maven" section and click on "Add Maven."
   - Configure jdk section and provide the JDK path (latest version)
   - Enter a name for the Maven installation and specify the Maven version to be installed.
   - Optionally, you can choose to install automatically from Apache or provide a custom Maven home directory.



localhost:8080/manage/configureTools/

Dashboard > Manage Jenkins > Tools

Tools

Maven Configuration

Default settings provider: Use default maven settings

Default global settings provider: Use default maven global settings

JDK installations

Add JDK

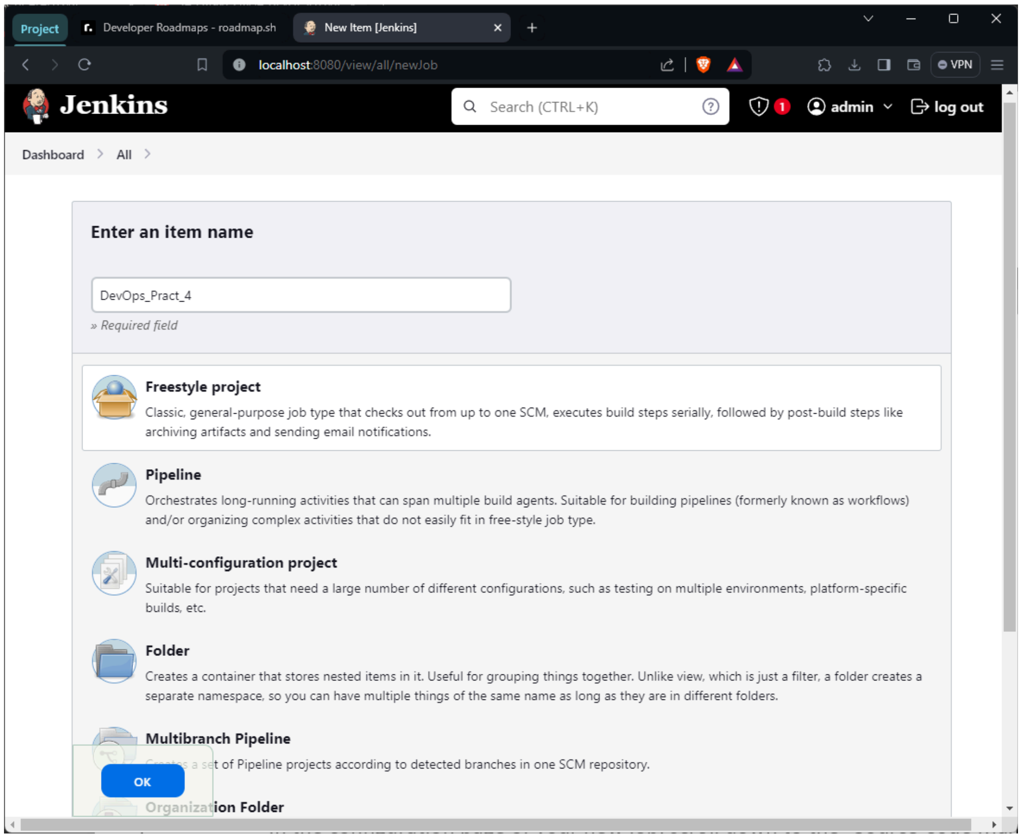
Git installations

Git

Name: Default

Save Apply

4. **Create a New Jenkins Job:**
   - Click on "New Item" from the Jenkins dashboard.
   - Enter a name for your project and select "Maven" as the project type.
   - Click "OK" to create the job.



Enter an item name

DevOps_Pract_4

» Required field

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.

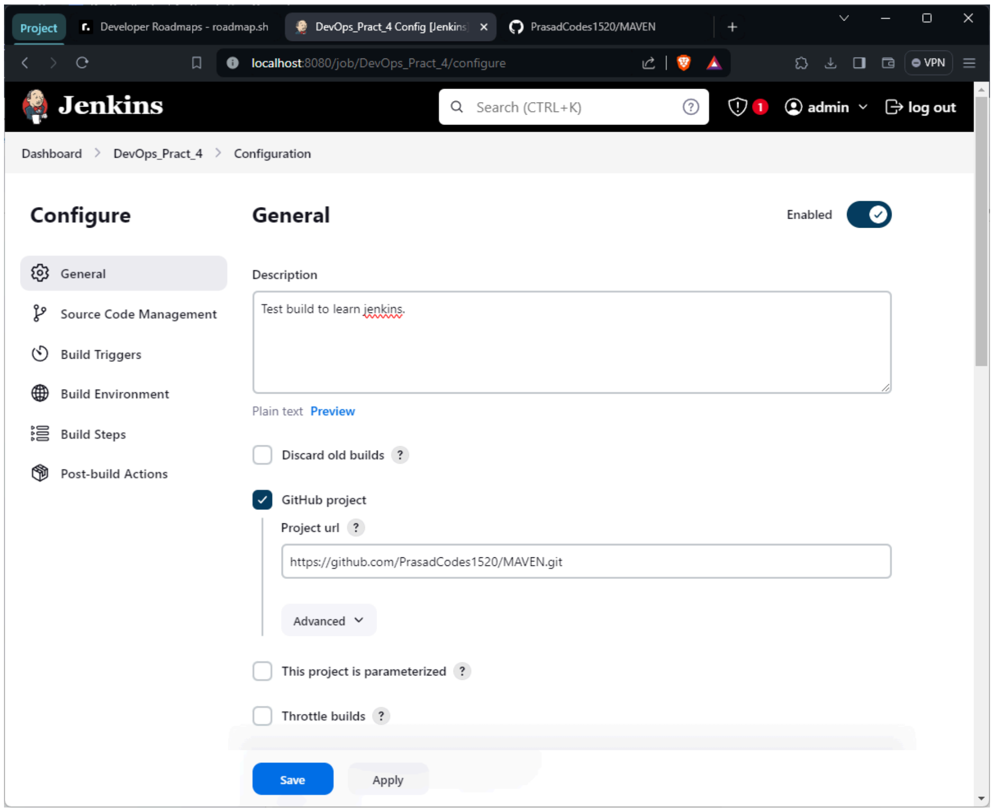
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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

5. **Configure Source Code Management:**
   - In the configuration page of your new job, scroll down to the "Source Code Management" section.
   - Select your version control system (e.g., Git) and provide the repository URL.



Dashboard > DevOps_Pract_4 > Configuration

Configure

General — Enabled

Description: Test build to learn jenkins.

Plain text Preview

[ ] Discard old builds

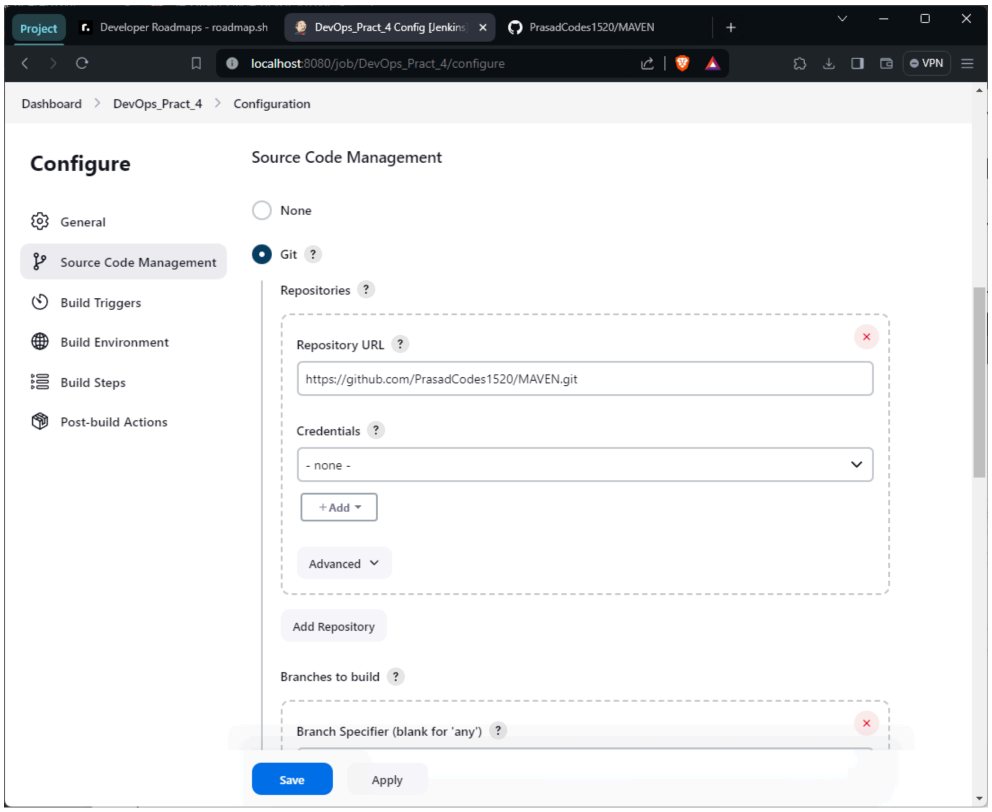
[✓] GitHub project

Project url: https://github.com/PrasadCodes1520/MAVEN.git

Advanced

[ ] This project is parameterized

[ ] Throttle builds



Dashboard > DevOps_Pract_4 > Configuration

Configure

Source Code Management

( ) None

(●) Git

Repositories

Repository URL: https://github.com/PrasadCodes1520/MAVEN.git

Credentials: - none -

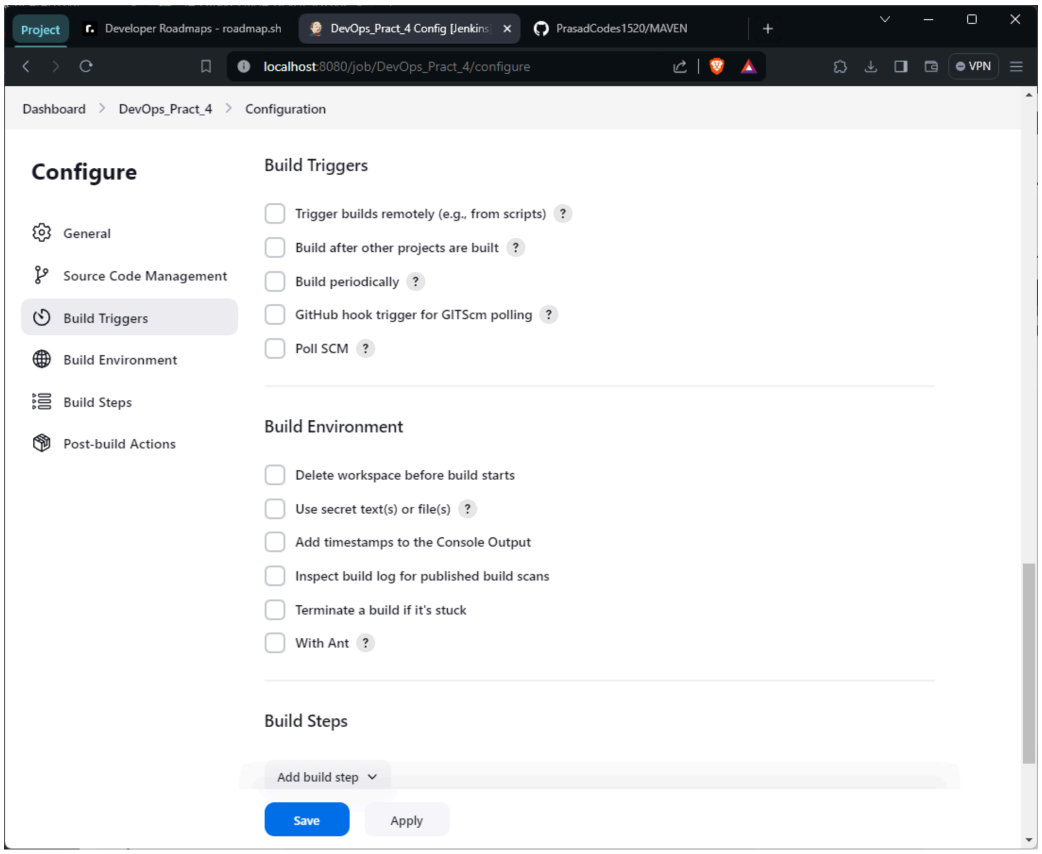
Advanced

Add Repository

Branches to build

Branch Specifier (blank for 'any')

6. **Configure Build Steps:**
   - Scroll down to the "Build" section and click on "Add build step."
   - Select "Invoke top-level Maven targets."
   - In the "Goals" field, enter the Maven goals to execute (e.g., `clean install`).
   - Optionally, specify any additional Maven options or parameters.


Project
Developer Roadmaps - roadmap.sh
DevOps_Pract_4 Config [Jenkins]
PrasadCodes1520/MAVEN
localhost:8080/job/DevOps_Pract_4/configure
VPN
Dashboard > DevOps_Pract_4 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions

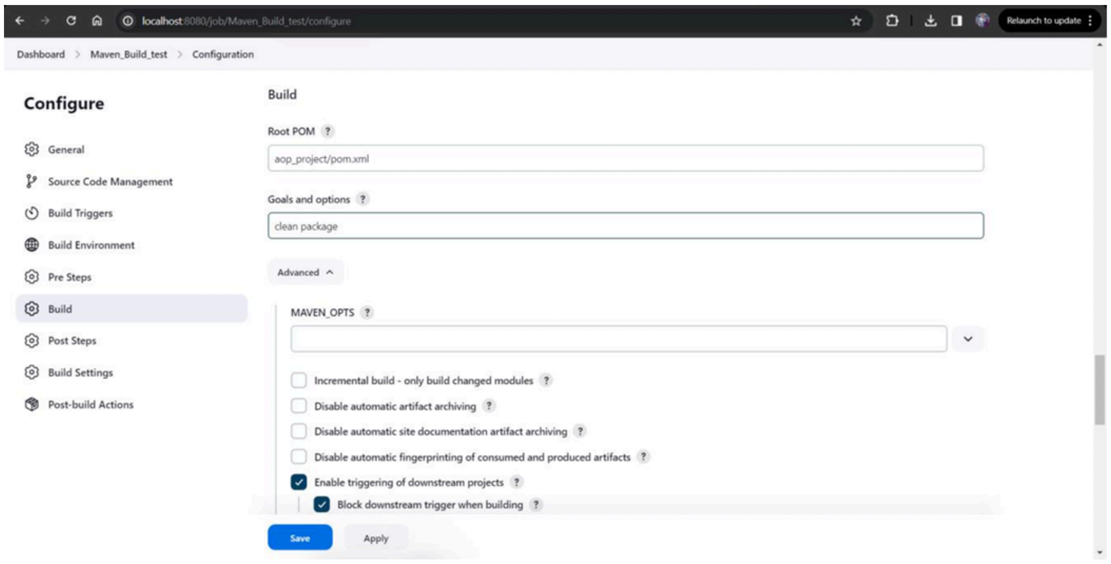
Build Triggers

- [ ] 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 build scans
- [ ] Terminate a build if it's stuck
- [ ] With Ant

Build Steps



Build

Root POM: aop_project/pom.xml

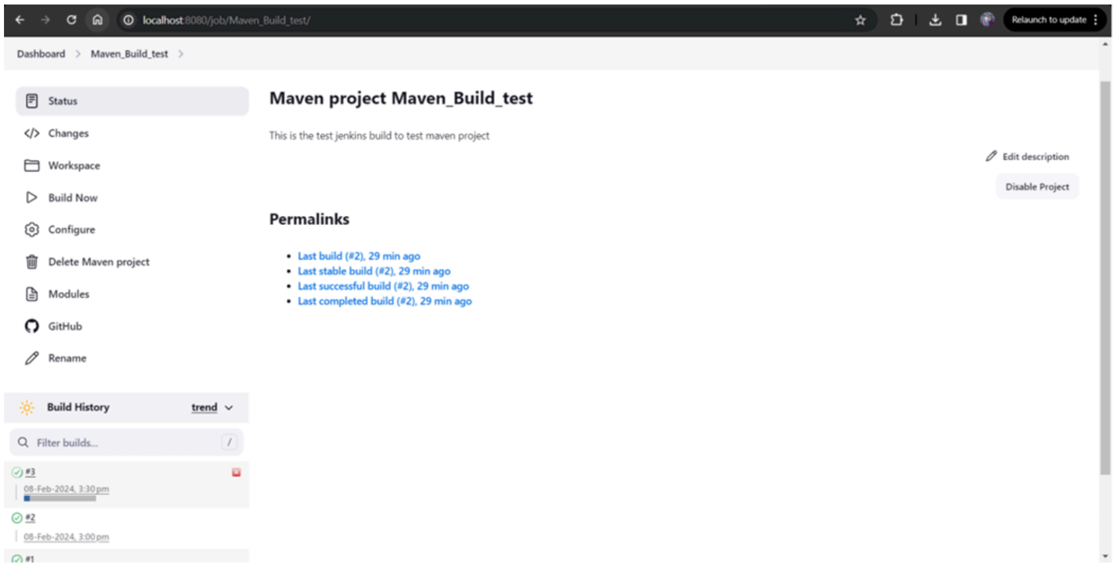
Goals and options: clean package

Advanced

MAVEN_OPTS:

- [ ] Incremental build - only build changed modules
- [ ] Disable automatic artifact archiving
- [ ] Disable automatic site documentation artifact archiving
- [ ] Disable automatic fingerprinting of consumed and produced artifacts
- [x] Enable triggering of downstream projects
  - [x] Block downstream trigger when building

7. **Save and Run the Job:**
   - Click "Save" to apply the configuration changes.
   - Trigger a build by clicking on "Build Now" from the job dashboard.
   - Jenkins will start the build process, pulling the source code from the repository, building the Maven project, and executing the specified goals.



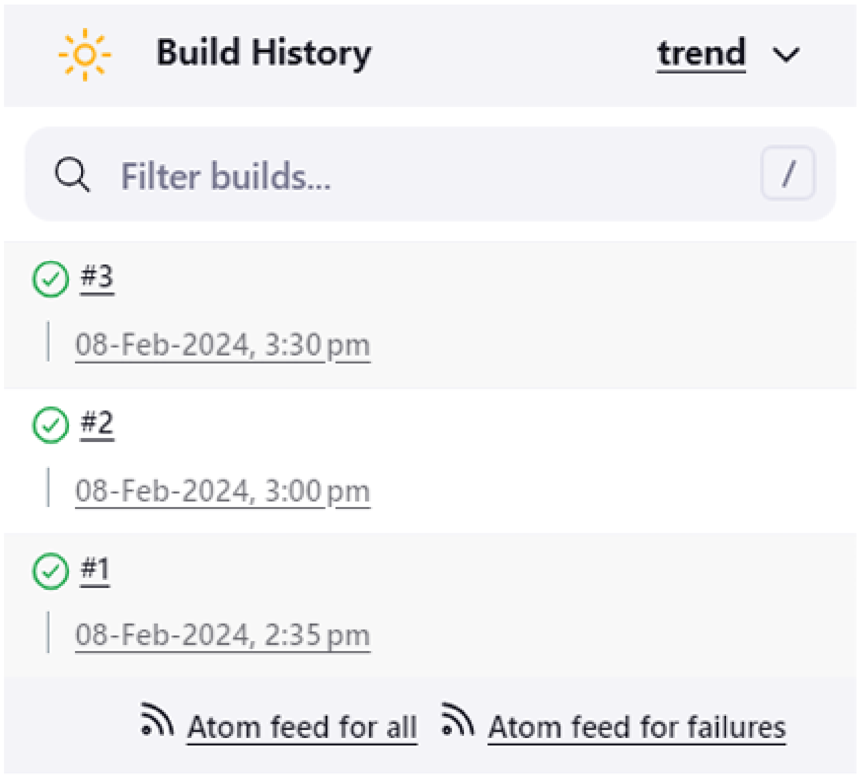
Maven project Maven_Build_test

This is the test jenkins build to test maven project

Permalinks

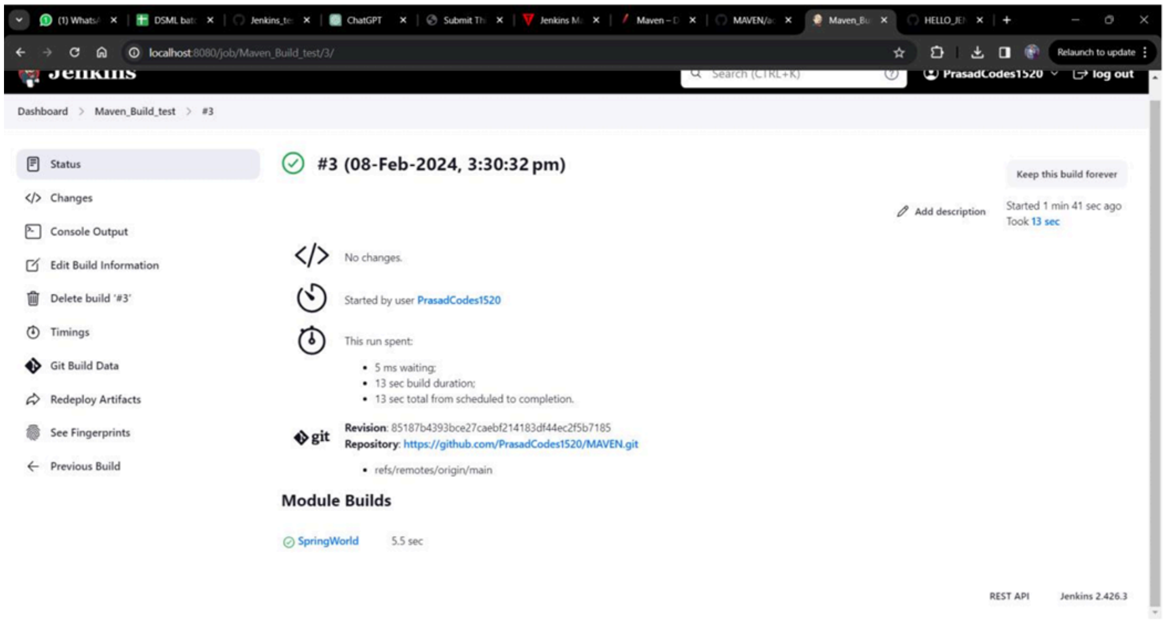
- Last build (#2), 29 min ago
- Last stable build (#2), 29 min ago
- Last successful build (#2), 29 min ago
- Last completed build (#2), 29 min ago

#3 08-Feb-2024, 3:30 pm
#2 08-Feb-2024, 3:00 pm
#1



#3
08-Feb-2024, 3:30 pm
#2
08-Feb-2024, 3:00 pm
#1
08-Feb-2024, 2:35 pm

8. **View Build Results:**
    - Once the build completes, navigate to the job dashboard to view the build history and results.
    - Click on the build number to access detailed build logs, test reports, and any generated artifacts.
    - Analyze the build output to ensure the Maven project was successfully built without errors.



Dashboard > Maven_Build_test > #3

#3 (08-Feb-2024, 3:30:32 pm)

Keep this build forever

Add description

Started 1 min 41 sec ago
Took 13 sec

No changes.

Started by user PrasadCodes1520

This run spent:

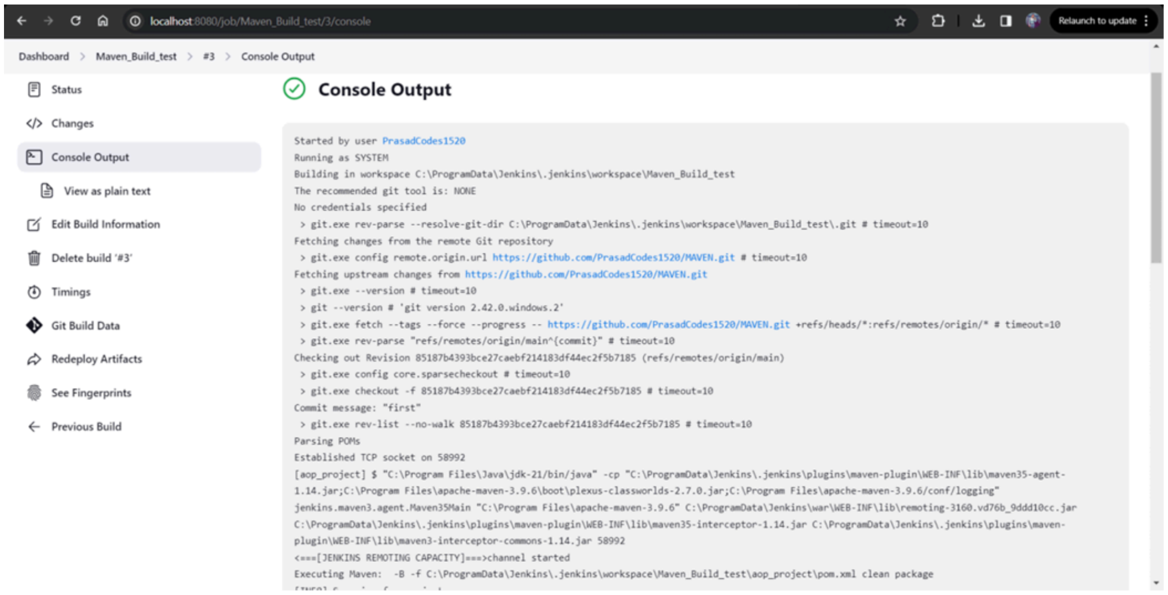
- 5 ms waiting;
- 13 sec build duration;
- 13 sec total from scheduled to completion.

Revision: 85187b4393bce27caebf214183df44ec2f5b7185
Repository: https://github.com/PrasadCodes1520/MAVEN.git

- refs/remotes/origin/main

Module Builds

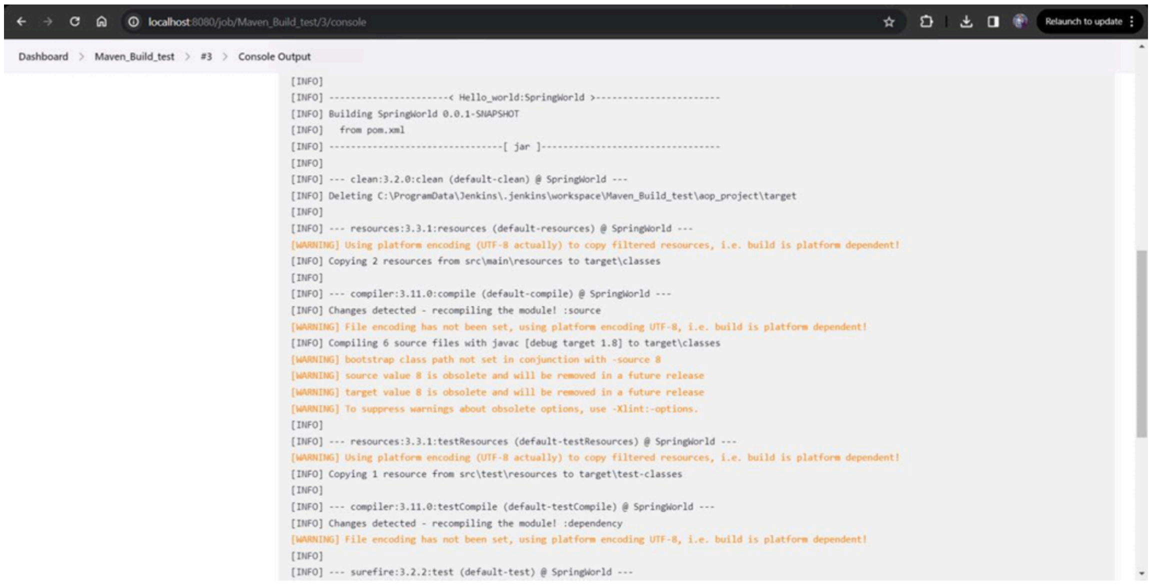
SpringWorld 5.5 sec



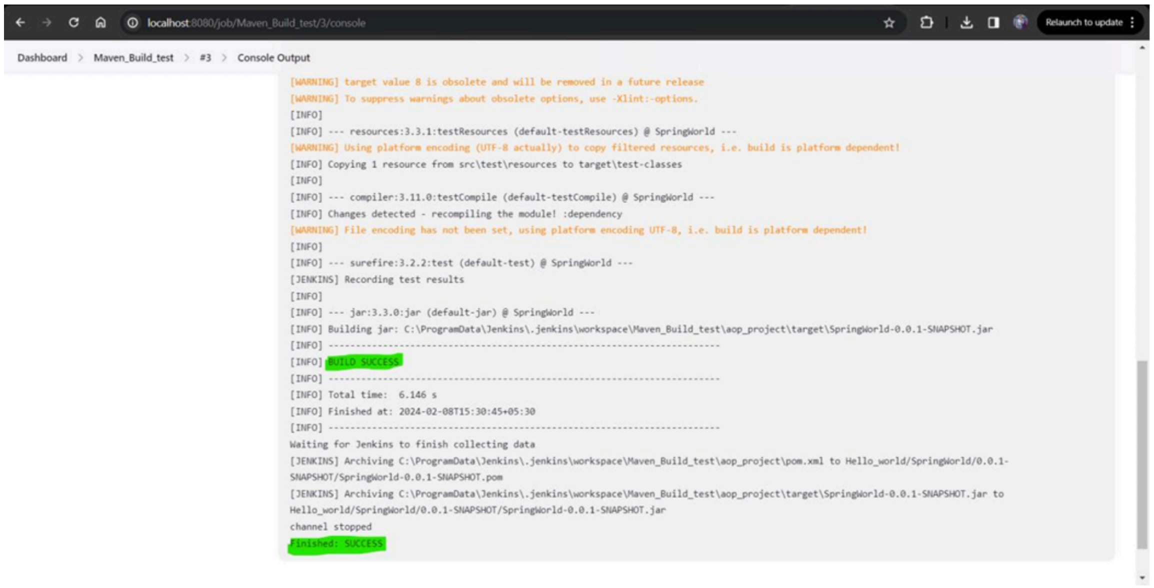
Dashboard > Maven_Build_test > #3 > Console Output

Console Output

```
Started by user PrasadCodes1520
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/PrasadCodes1520/MAVEN.git # timeout=10
Fetching upstream changes from https://github.com/PrasadCodes1520/MAVEN.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.42.0.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/PrasadCodes1520/MAVEN.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 85187b4393bce27caebf214183df44ec2f5b7185 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 85187b4393bce27caebf214183df44ec2f5b7185 # timeout=10
Commit message: "first"
 > git.exe rev-list --no-walk 85187b4393bce27caebf214183df44ec2f5b7185 # timeout=10
Parsing POMs
Established TCP socket on 58992
[aop_project] $ "C:\Program Files\Java\jdk-21/bin/java" -cp "C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven35-agent-
1.14.jar;C:\Program Files\apache-maven-3.9.6\boot\plexus-classworlds-2.7.0.jar;C:\Program Files\apache-maven-3.9.6/conf/logging"
jenkins.maven3.agent.Maven35Main "C:\Program Files\apache-maven-3.9.6" C:\ProgramData\Jenkins\war\WEB-INF\lib\remoting-3160.vd76b_9ddd10cc.jar
C:\ProgramData\Jenkins\.jenkins\plugins\maven-plugin\WEB-INF\lib\maven35-interceptor-1.14.jar C:\ProgramData\Jenkins\.jenkins\plugins\maven-
plugin\WEB-INF\lib\maven3-interceptor-commons-1.14.jar 58992
<===[JENKINS REMOTING CAPACITY]===>channel started
Executing Maven:  -B -f C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\aop_project\pom.xml clean package
```


localhost:8080/job/Maven_Build_test/3/console
Relaunch to update
Dashboard > Maven_Build_test > #3 > Console Output

```
[INFO]
[INFO] --------------------< Hello_world:SpringWorld >---------------------
[INFO] Building SpringWorld 0.0.1-SNAPSHOT
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]--------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ SpringWorld ---
[INFO] Deleting C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\aop_project\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ SpringWorld ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 2 resources from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.11.0:compile (default-compile) @ SpringWorld ---
[INFO] Changes detected - recompiling the module! :source
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 6 source files with javac [debug target 1.8] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 8
[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] --- resources:3.3.1:testResources (default-testResources) @ SpringWorld ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource from src\test\resources to target\test-classes
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ SpringWorld ---
[INFO] Changes detected - recompiling the module! :dependency
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO]
[INFO] --- surefire:3.2.2:test (default-test) @ SpringWorld ---
```


localhost:8080/job/Maven_Build_test/3/console
Relaunch to update
Dashboard > Maven_Build_test > #3 > Console Output

```
[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] --- resources:3.3.1:testResources (default-testResources) @ SpringWorld ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource from src\test\resources to target\test-classes
[INFO]
[INFO] --- compiler:3.11.0:testCompile (default-testCompile) @ SpringWorld ---
[INFO] Changes detected - recompiling the module! :dependency
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO]
[INFO] --- surefire:3.2.2:test (default-test) @ SpringWorld ---
[JENKINS] Recording test results
[INFO]
[INFO] --- jar:3.3.0:jar (default-jar) @ SpringWorld ---
[INFO] Building jar: C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\aop_project\target\SpringWorld-0.0.1-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.146 s
[INFO] Finished at: 2024-02-08T15:30:45+05:30
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\aop_project\pom.xml to Hello_world/SpringWorld/0.0.1-
SNAPSHOT/SpringWorld-0.0.1-SNAPSHOT.pom
[JENKINS] Archiving C:\ProgramData\Jenkins\.jenkins\workspace\Maven_Build_test\aop_project\target\SpringWorld-0.0.1-SNAPSHOT.jar to
Hello_world/SpringWorld/0.0.1-SNAPSHOT/SpringWorld-0.0.1-SNAPSHOT.jar
channel stopped
Finished: SUCCESS
```

9. **Configure Post-Build Actions (Optional):**
   - To automate additional tasks after the build completes, configure post-build actions.
   - Examples include archiving artifacts, triggering downstream jobs, sending notifications, or publishing reports.

10. **Set Up Continuous Integration (Optional):**
    - To enable continuous integration, configure Jenkins to poll your version control system for changes and trigger builds automatically.

- Navigate to the job configuration page and configure the "Build Triggers" section to specify how often Jenkins should check for changes.

Conclusion: In conclusion, this practical guide provides a streamlined approach to building Maven projects with Jenkins. By following these steps, users can seamlessly integrate Jenkins into their development workflow, automate the build process, and ensure the reliability and consistency of their Maven projects. Through the combination of Jenkins' powerful automation capabilities and Maven's dependency management, developers can efficiently manage and deploy their Java applications with confidence.