Name : Taha Alotwala
Roll No. : 03
Batch : T11

Experiment 4 : Continuous Integration - Jenkins

Aim :

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

Theory :

1. Continuous Integration (CI)

Continuous Integration (CI) is a software development practice where code changes are automatically integrated, tested, and deployed frequently, often multiple times a day. The goal of CI is to improve code quality, reduce integration issues, and speed up the development lifecycle.

Key Concepts of CI:

- **Automated Testing:** Code is automatically tested every time it is committed to the repository to catch bugs early.
- **Frequent Integration:** Developers integrate their code changes into a shared repository regularly to avoid conflicts.
- **Continuous Feedback:** Developers receive quick feedback about the health of their codebase, enabling them to fix issues promptly.
- **Build Automation:** The process of compiling code, running tests, and deploying applications is automated.

Benefits of Continuous Integration:

1. **Early Detection of Errors:** Automated tests catch bugs soon after they are introduced, reducing the cost of fixing them.

2. **Improved Code Quality:** Continuous testing ensures that only high-quality code is merged into the main codebase.
3. **Faster Development Cycles:** Developers can work in parallel without waiting for integration, enabling faster feature releases.
4. **Reduced Manual Effort:** Automating builds and tests reduces manual errors and saves time.

Key CI Practices:

- **Version Control System (VCS) Integration:** CI tools integrate with Git, SVN, etc., to monitor code changes.
- **Automated Builds:** Code is automatically compiled and built into deployable artifacts.
- **Automated Testing:** Unit tests, integration tests, and UI tests run automatically after every code change.
- **Deployment Automation:** Code can be automatically deployed to staging or production environments.

---

2. Jenkins: A Leading CI/CD Tool

Jenkins is an open-source automation server commonly used to implement CI/CD pipelines. It automates parts of software development related to building, testing, and deploying code.

Key Features of Jenkins:

- **Extensibility:** Jenkins has a rich ecosystem of plugins to support building, deploying, and automating projects for multiple languages and technologies.
- **Distributed Builds:** Jenkins can distribute tasks across multiple machines for faster processing.
- **Pipeline as Code:** Using Jenkins Pipeline DSL (Domain-Specific Language), you can define CI/CD workflows as code in a `Jenkinsfile`.

- **Integration with VCS:** Jenkins integrates seamlessly with Git, SVN, and other version control systems.

---

3. Jenkins Architecture

Jenkins operates on a master-slave (or controller-agent) architecture:

- **Master (Controller):** The central server that manages the CI/CD environment, schedules jobs, and monitors their execution.
- **Agents (Slaves):** Machines that perform the actual build and test tasks. Jenkins can distribute workloads to multiple agents for scalability.

---

4. Jenkins Components

- **Jobs:** Individual tasks that Jenkins can execute, such as building code, running tests, or deploying applications.
- **Builds:** The process of compiling and packaging the code. Each build can be triggered manually or automatically.
- **Pipelines:** A series of automated steps defined in a `Jenkinsfile` that describe the CI/CD process.
- **Plugins:** Extend Jenkins' functionality, allowing integration with various tools like Git, Docker, Maven, etc.

---

5. Setting Up Jenkins

Step 1: Install Jenkins
On Ubuntu:

```
sudo apt update
sudo apt install openjdk-11-jdk
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo
apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/
> /etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins
sudo systemctl start jenkins
sudo systemctl status jenkins
```

- **On Windows:**
  Download the Jenkins installer from the Jenkins official website and follow the installation wizard.

Step 2: Access Jenkins Web Interface

- Open your browser and go to: `http://localhost:8080`

Retrieve the initial admin password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

- Unlock Jenkins and install the suggested plugins.

---

6. Configuring Jenkins for CI

Step 1: Create a New Job

1. Click **“New Item”** in the Jenkins dashboard.

2. Enter a name for the job.

3. Choose the job type:
    - **Freestyle Project:** Basic job configuration.
    - **Pipeline:** Advanced, code-defined CI/CD workflow.

Step 2: Configure Source Code Repository

- **For GitHub Integration:**
    - In the job configuration, go to **“Source Code Management”**.
    - Select **Git** and enter the repository URL.
    - Add credentials if necessary.

Step 3: Add Build Triggers

- Configure triggers to automate builds:
    - **Poll SCM:** Periodically check for changes in the repository.
    - **GitHub Hook Trigger:** Trigger builds automatically when changes are pushed to GitHub.

Step 4: Add Build Steps

- **Execute Shell:** Run shell commands or scripts.
- **Invoke Ant/Maven/Gradle:** Run build tools for Java projects.
- **Execute Windows Batch Command:** Run batch files on Windows systems.

Step 5: Add Post-Build Actions

- **Archive Artifacts:** Save build outputs for later use.
- **Send Notifications:** Email alerts or Slack messages after builds.
- **Deploy Applications:** Automate deployment to servers.

7. Jenkins Pipeline (Declarative & Scripted)

Declarative Pipeline (Recommended for Most Cases):

A declarative pipeline is defined in a `Jenkinsfile` using a simplified syntax.

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'Building the project...'
                sh 'mvn clean install'
            }
        }
        stage('Test') {
            steps {
                echo 'Running tests...'
                sh 'mvn test'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying application...'
                sh 'scp target/app.jar
user@server:/path/to/deploy'
            }
        }
    }
}
```

Scripted Pipeline (More Flexible):

Scripted pipelines offer more control but require a deeper understanding of Groovy.

```
node {
    stage('Build') {
        sh 'mvn clean install'
    }
    stage('Test') {
        sh 'mvn test'
    }
    stage('Deploy') {
        sh 'scp target/app.jar user@server:/path/to/deploy'
    }
}
```

---

8. Common Jenkins Plugins

- **Git Plugin:** Integrates Jenkins with Git repositories.
- **Pipeline Plugin:** Enables the use of Jenkins Pipelines.
- **Docker Pipeline:** Manages Docker containers in CI/CD workflows.
- **Blue Ocean:** A modern UI for visualizing Jenkins pipelines.
- **Slack Notification Plugin:** Sends notifications to Slack channels.

---

9. Advanced Jenkins Concepts

9.1. Jenkinsfile as Code

Defining your pipeline in a `Jenkinsfile` allows you to version control your CI/CD process alongside your application code.

9.2. Parallel Execution

You can run multiple stages in parallel to speed up your pipeline:

```
pipeline {
    agent any
    stages {
        stage('Parallel Stages') {
            parallel {
                stage('Unit Test') {
                    steps {
                        sh 'mvn test'
                    }
                }
                stage('Integration Test') {
                    steps {
                        sh 'mvn verify'
                    }
                }
            }
        }
    }
}
```

9.3. Parameterized Builds

Allow users to pass parameters to Jenkins jobs:

```
pipeline {
    agent any
    parameters {
        string(name: 'ENV', defaultValue: 'staging',
description: 'Deployment Environment')
    }
    stages {
        stage('Deploy') {
            steps {
                sh "deploy.sh ${params.ENV}"
            }
        }
    }
}
```

---

10. Jenkins Security Best Practices

- **Use Role-Based Access Control:** Limit permissions based on user roles.
- **Secure Credentials:** Store sensitive data (like API keys) in Jenkins Credentials Manager.

- **Keep Jenkins Updated:** Regularly update Jenkins and plugins to fix security vulnerabilities.

---

11. Troubleshooting Jenkins Issues

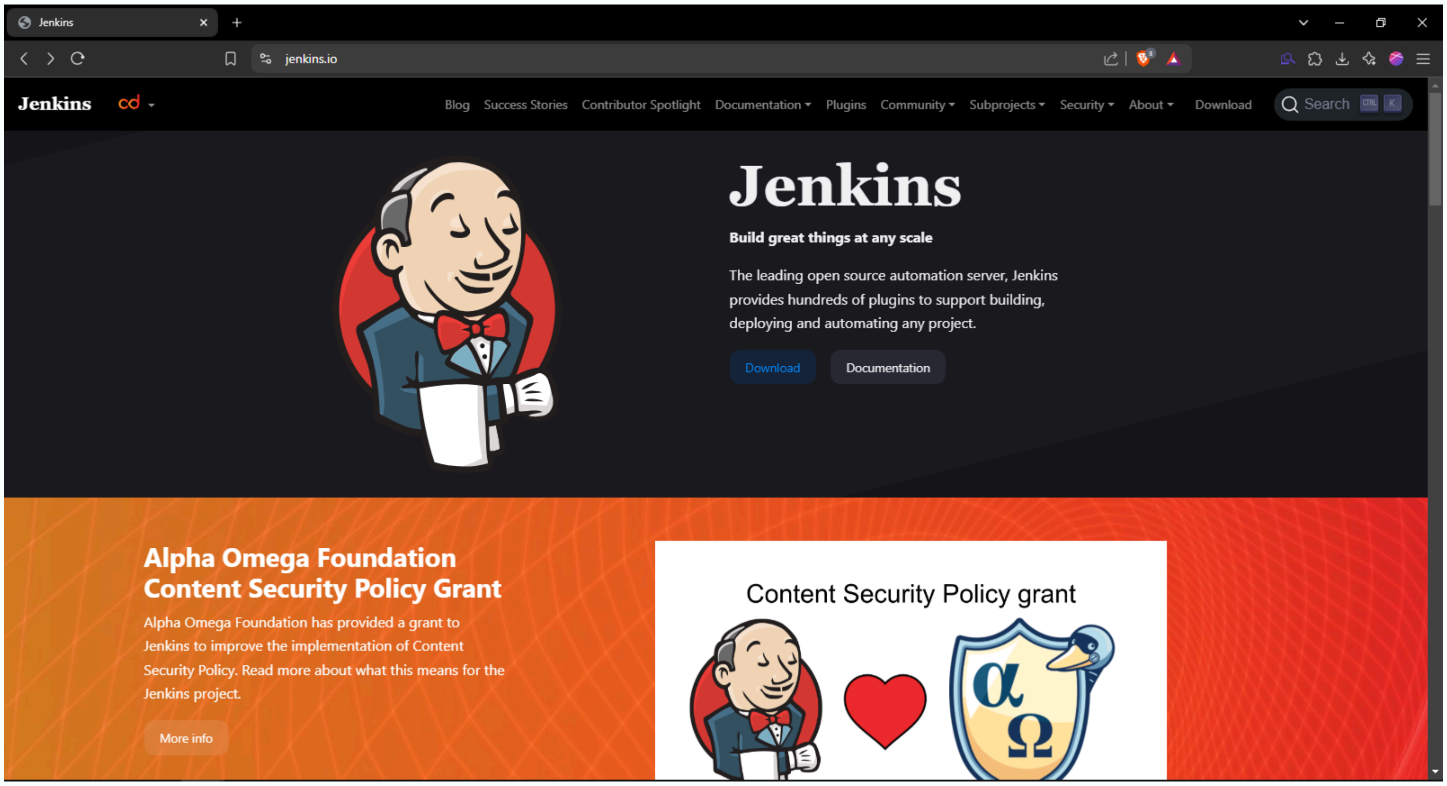
- **Build Fails:** Check the console output for error logs.
- **Permissions Issues:** Verify user permissions in Jenkins settings.
- **Plugin Conflicts:** Update or reinstall problematic plugins.

---

12. Summary of Key Jenkins Commands

Operation	Command/Action
Start Jenkins	`sudo systemctl start jenkins`
Stop Jenkins	`sudo systemctl stop jenkins`
Restart Jenkins	`sudo systemctl restart jenkins`
Check Jenkins Status	`sudo systemctl status jenkins`
Access Jenkins Dashboard	`http://localhost:8080`
Install Plugins	Manage Jenkins → Manage Plugins
Run Pipeline Manually	Click **“Build Now”**
View Build Logs	Click on a build → **Console Output**

Output :



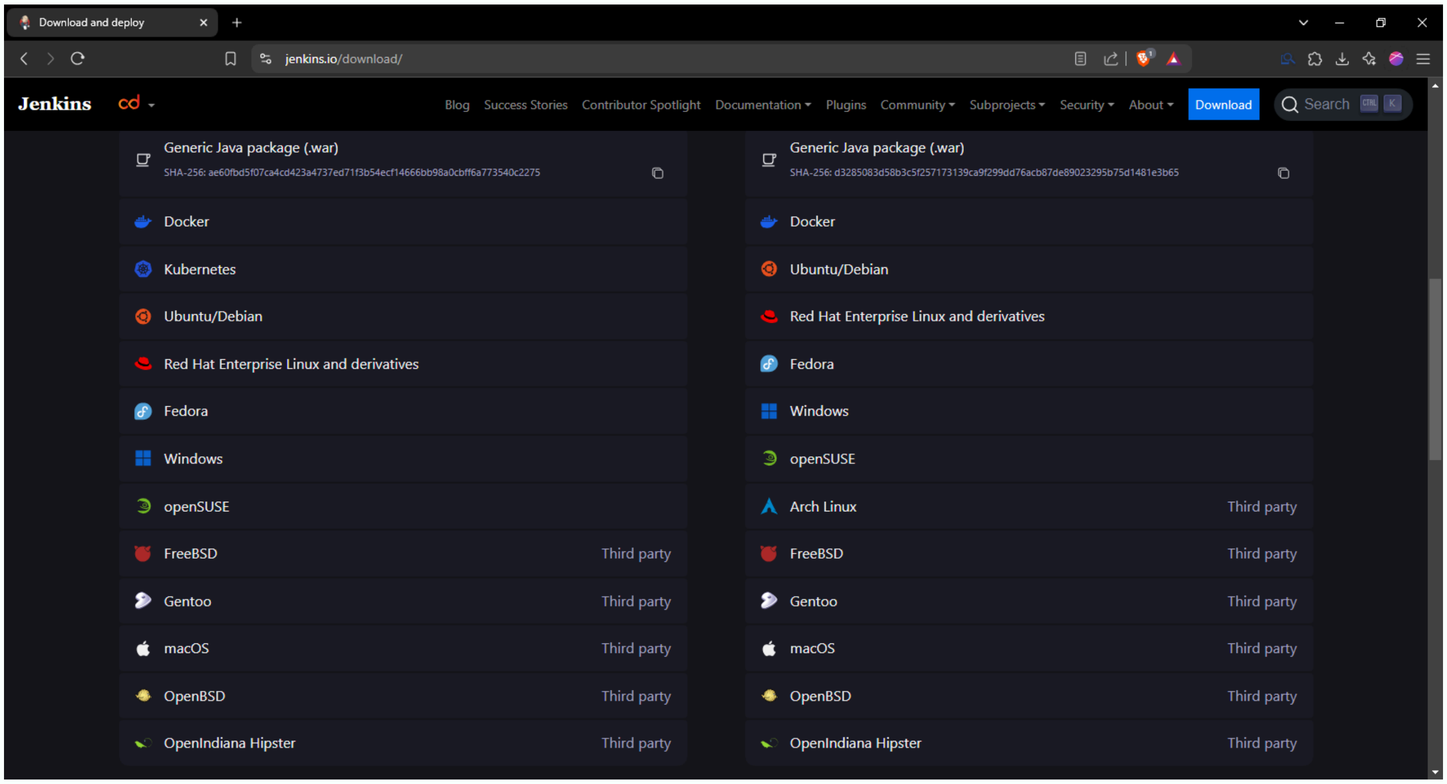
Jenkins

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.

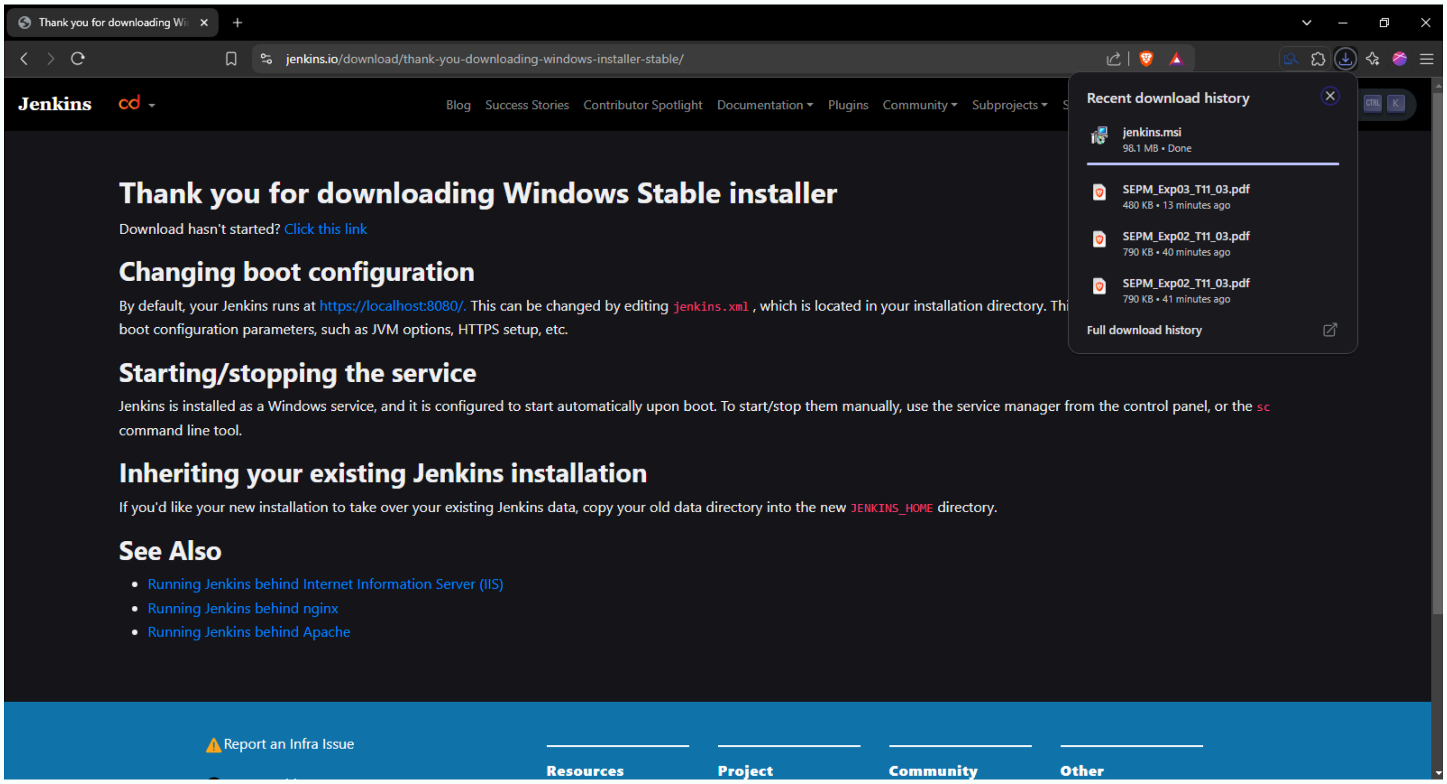
Alpha Omega Foundation Content Security Policy Grant

Alpha Omega Foundation has provided a grant to Jenkins to improve the implementation of Content Security Policy. Read more about what this means for the Jenkins project.



Generic Java package (.war) SHA-256: ae60fbd5f07ca4cd423a4737ed71f3b54ecf14666bb98a0cbff6a773540c2275	
Docker	
Kubernetes	
Ubuntu/Debian	
Red Hat Enterprise Linux and derivatives	
Fedora	
Windows	
openSUSE	
FreeBSD	Third party
Gentoo	Third party
macOS	Third party
OpenBSD	Third party
OpenIndiana Hipster	Third party

Generic Java package (.war) SHA-256: d3285083d58b3c5f257173139ca9f299dd76acb87de89023295b75d1481e3b65	
Docker	
Ubuntu/Debian	
Red Hat Enterprise Linux and derivatives	
Fedora	
Windows	
openSUSE	
Arch Linux	Third party
FreeBSD	Third party
Gentoo	Third party
macOS	Third party
OpenBSD	Third party
OpenIndiana Hipster	Third party


Thank you for downloading W

jenkins.io/download/thank-you-downloading-windows-installer-stable/

Recent download history

jenkins.msi
98.1 MB • Done

SEPM_Exp03_T11_03.pdf
480 KB • 13 minutes ago

SEPM_Exp02_T11_03.pdf
790 KB • 40 minutes ago

SEPM_Exp02_T11_03.pdf
790 KB • 41 minutes ago

Full download history

Thank you for downloading Windows Stable 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. Thi
boot configuration parameters, such as JVM options, HTTPS setup, etc.

Starting/stopping the service

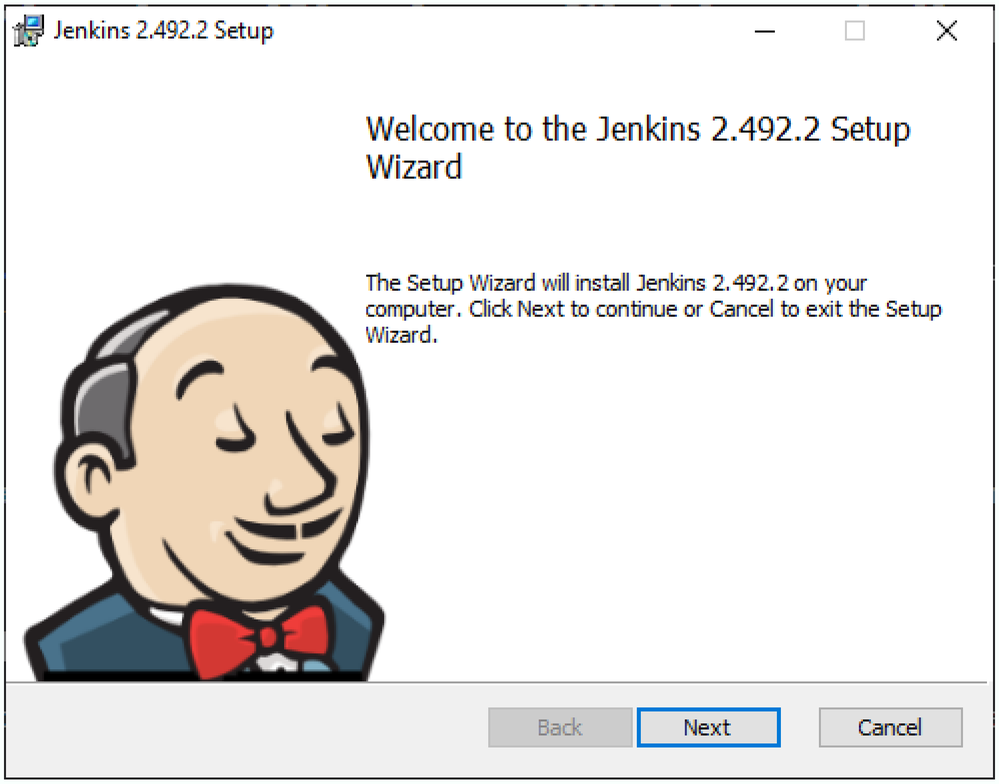
Jenkins is installed as a Windows service, and it is configured to start automatically upon boot. To start/stop them manually, use the service manager from the control panel, or the `sc` command line tool.

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.

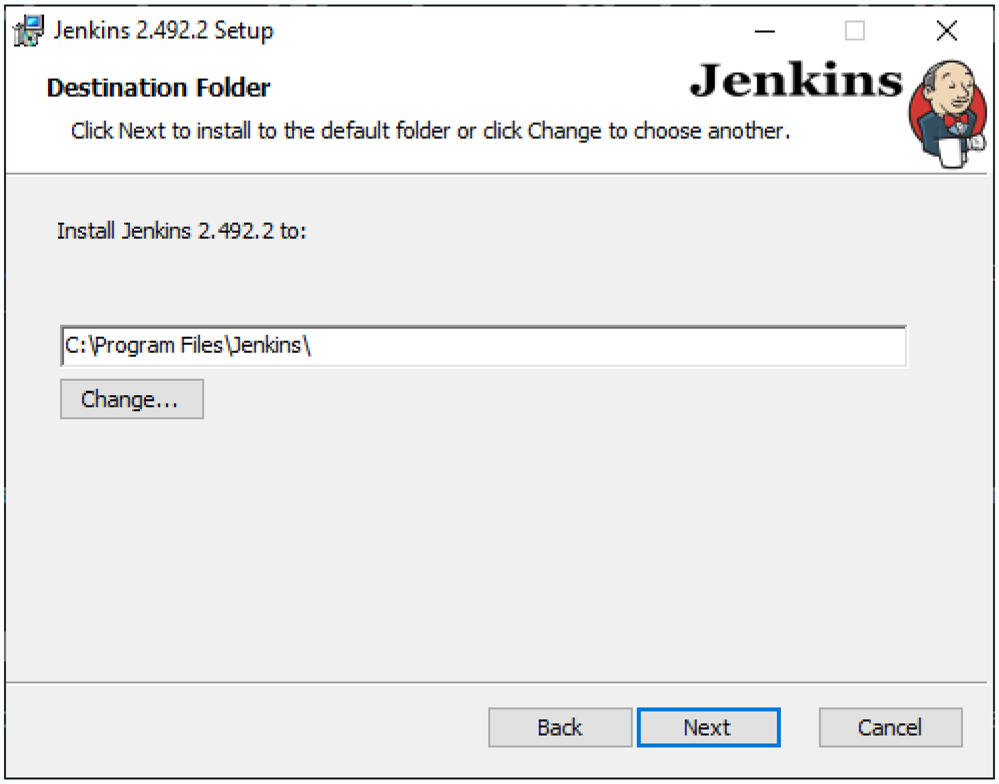
See Also

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


Jenkins 2.492.2 Setup

Welcome to the Jenkins 2.492.2 Setup Wizard

The Setup Wizard will install Jenkins 2.492.2 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.



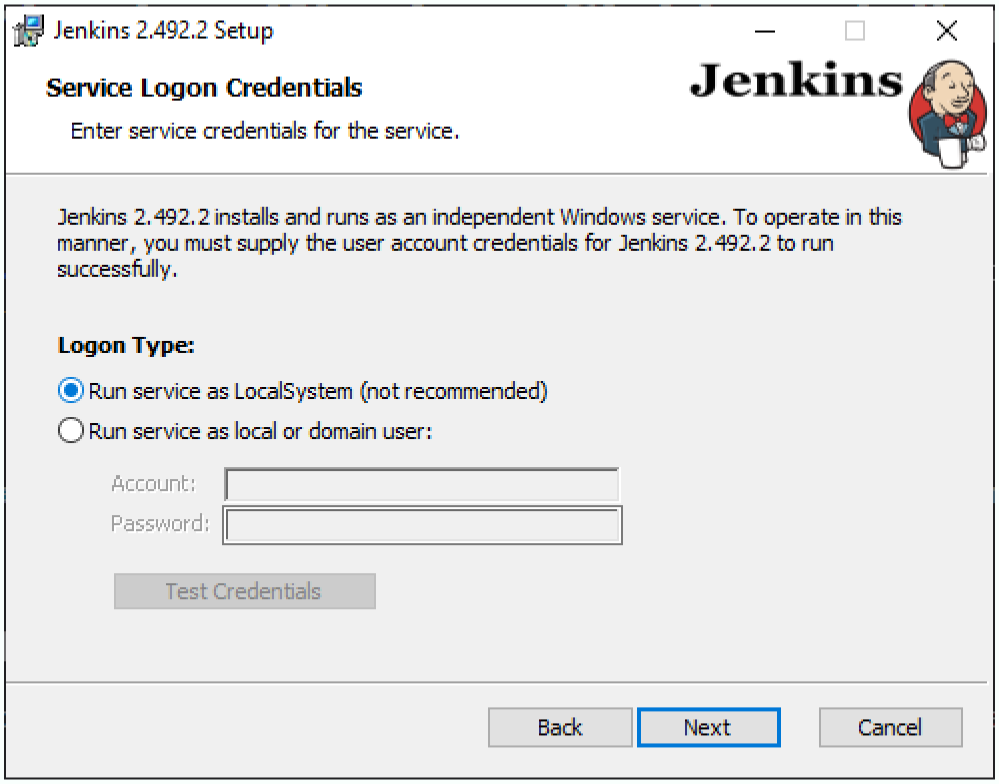
Jenkins 2.492.2 Setup

Destination Folder

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

Install Jenkins 2.492.2 to:

C:\Program Files\Jenkins\



Jenkins 2.492.2 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.492.2 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.492.2 to run successfully.

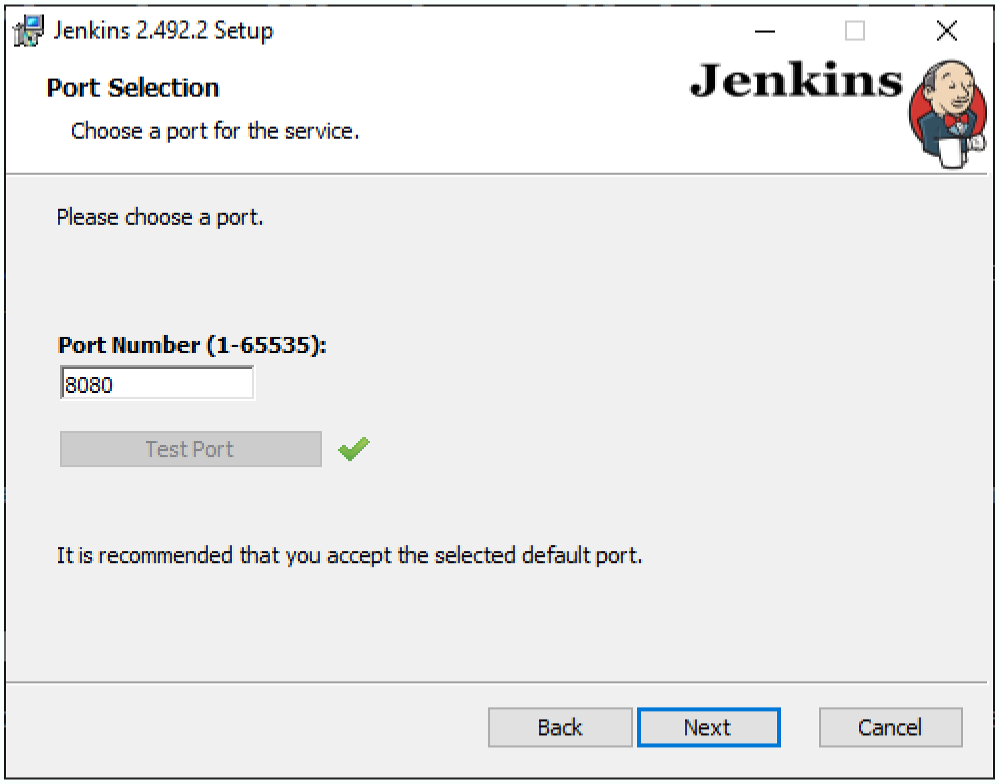
Logon Type:

● Run service as LocalSystem (not recommended)

○ Run service as local or domain user:

Account:

Password:


Jenkins 2.492.2 Setup

Port Selection

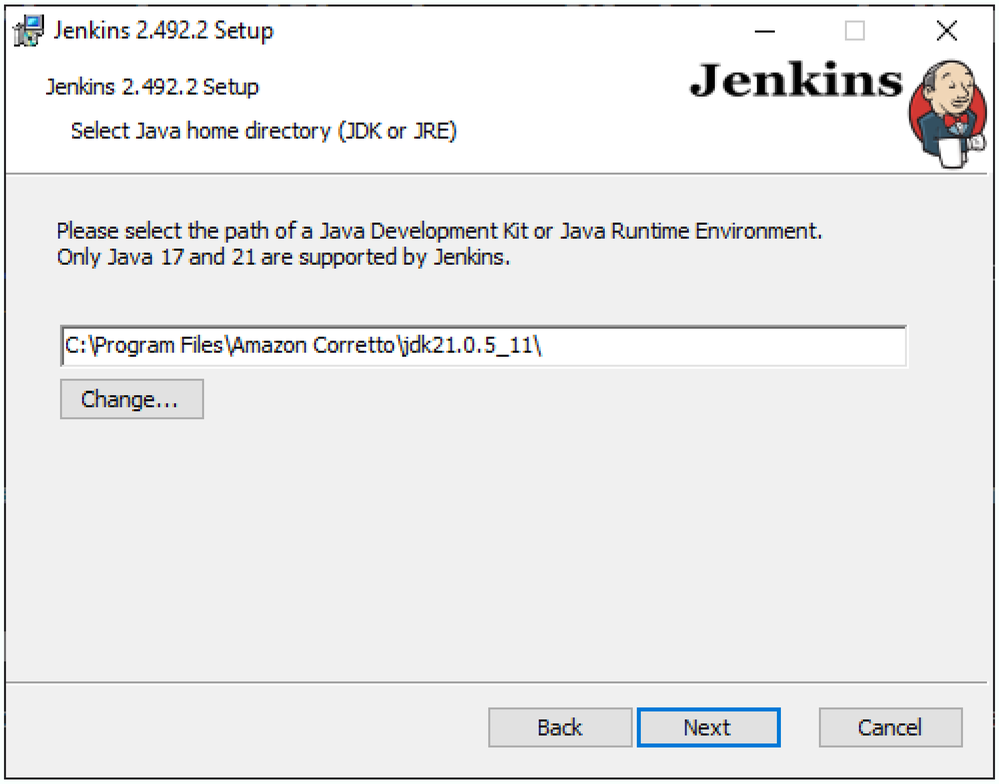
Choose a port for the service.

Please choose a port.

Port Number (1-65535):

8080

It is recommended that you accept the selected default port.

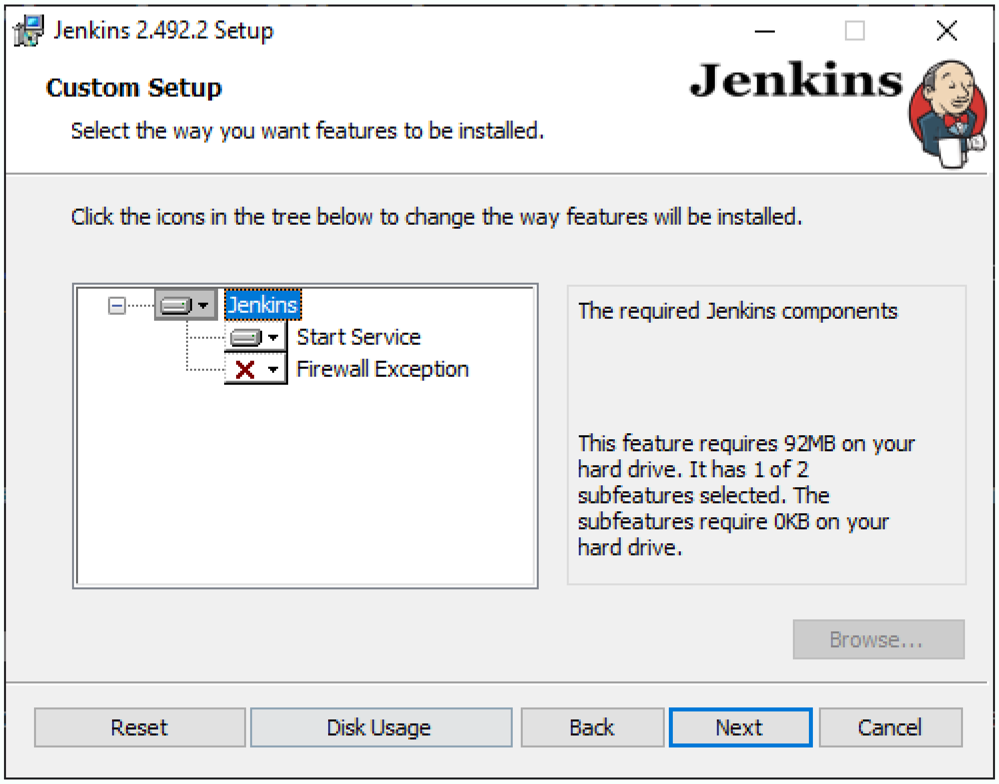

Jenkins 2.492.2 Setup

Jenkins 2.492.2 Setup

Select Java home directory (JDK or JRE)

Please select the path of a Java Development Kit or Java Runtime Environment.
Only Java 17 and 21 are supported by Jenkins.

C:\Program Files\Amazon Corretto\jdk21.0.5_11\


Jenkins 2.492.2 Setup

Custom Setup

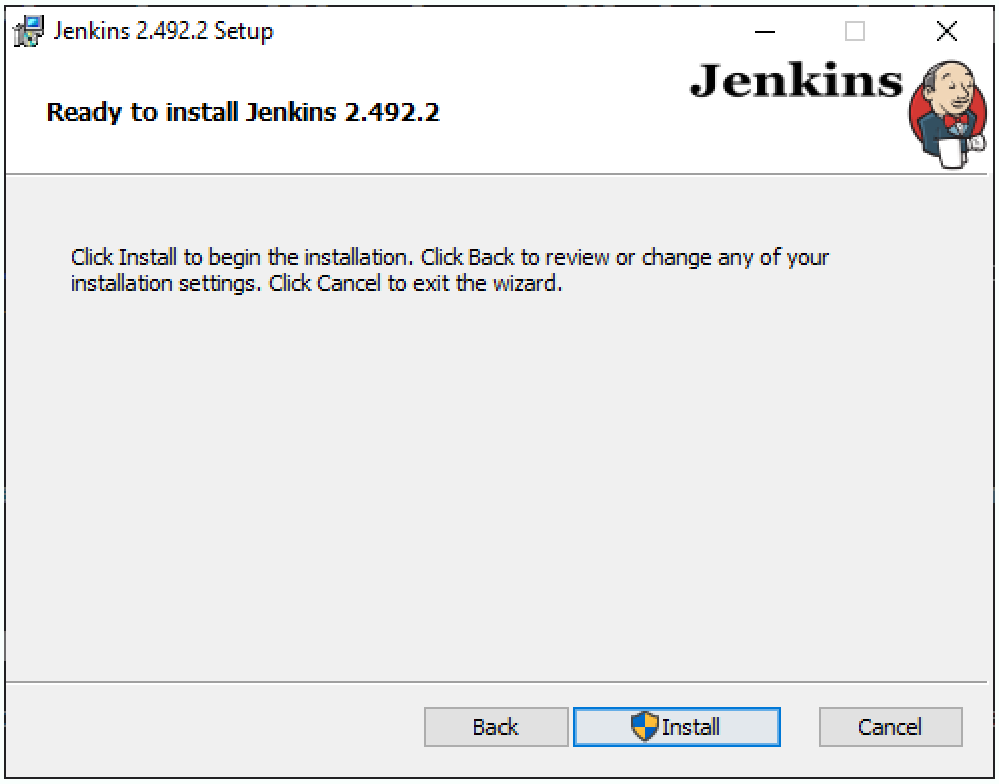
Select the way you want features to be installed.

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

- Jenkins
  - Start Service
  - Firewall Exception

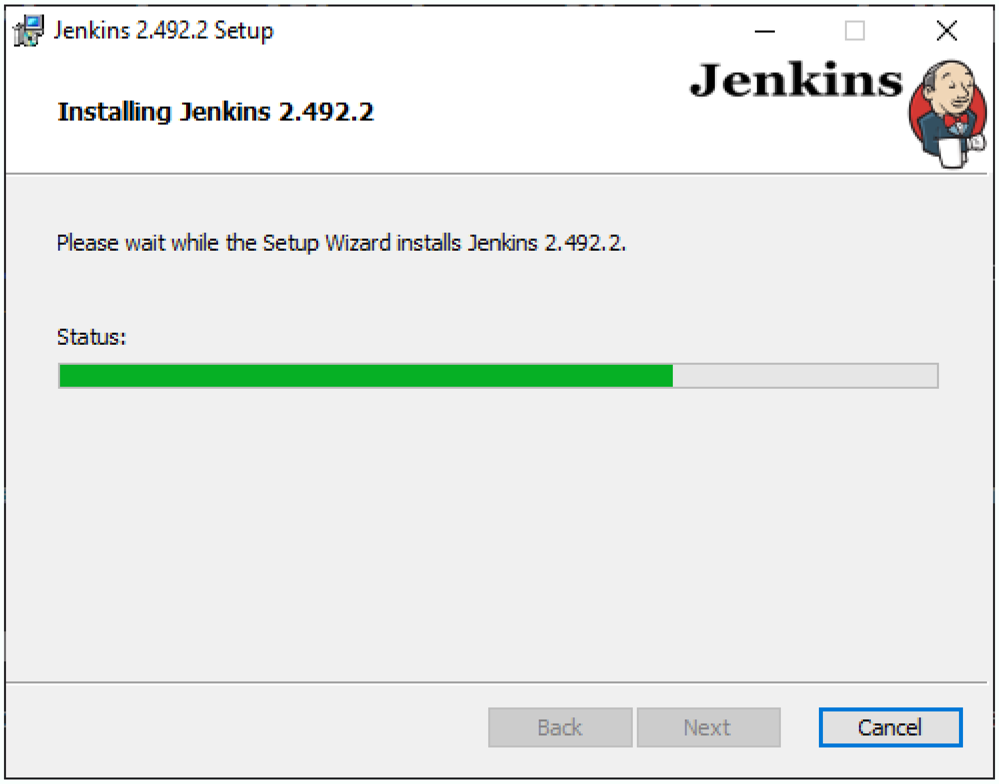
The required Jenkins components

This feature requires 92MB on your hard drive. It has 1 of 2 subfeatures selected. The subfeatures require 0KB on your hard drive.


Jenkins 2.492.2 Setup

Ready to install Jenkins 2.492.2

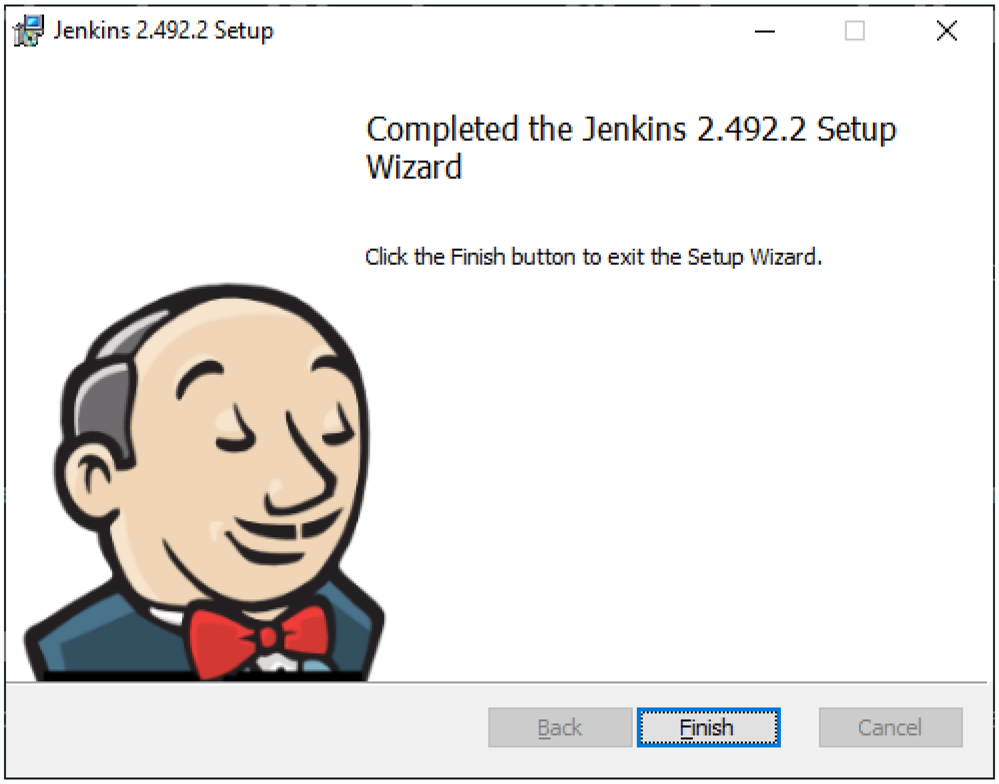
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.


Jenkins 2.492.2 Setup

Installing Jenkins 2.492.2

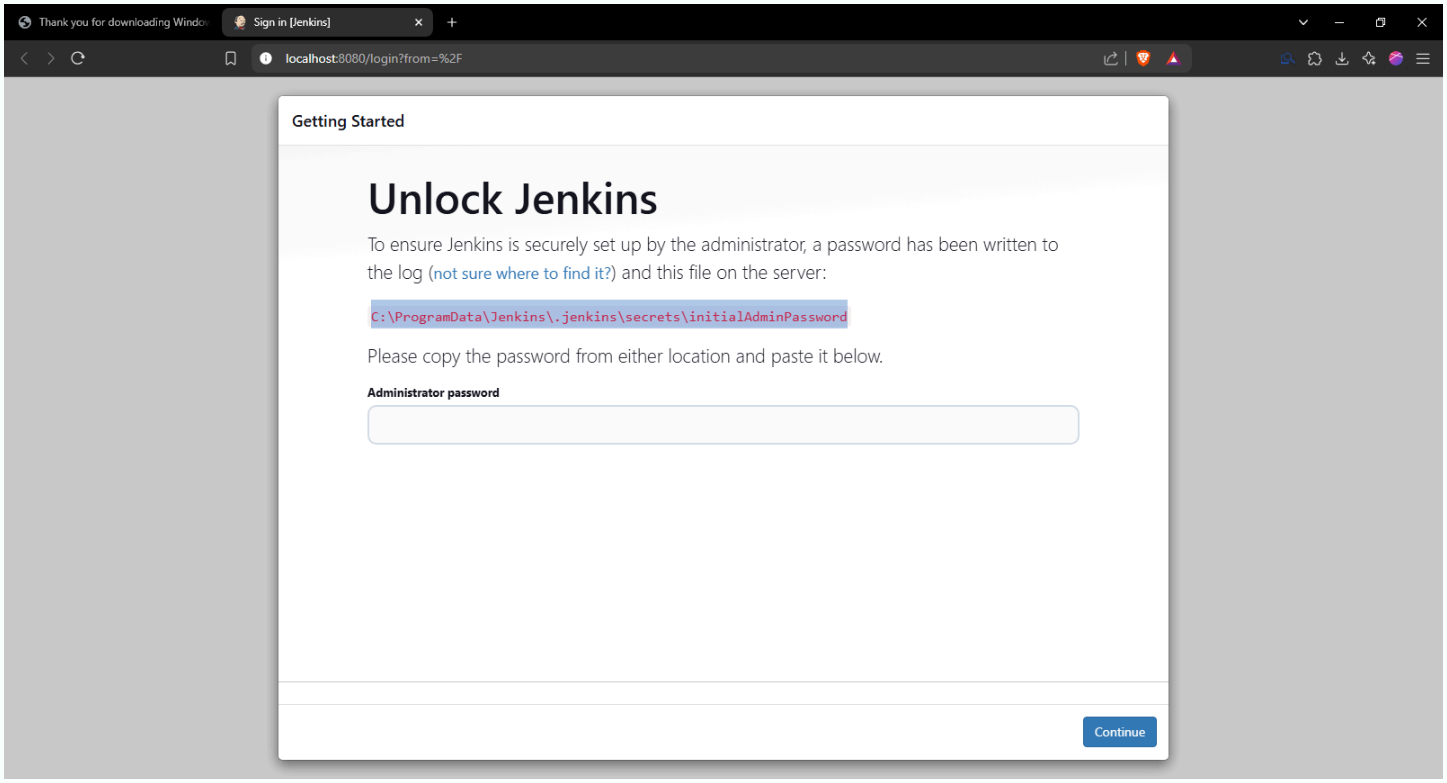
Please wait while the Setup Wizard installs Jenkins 2.492.2.

Status:


Jenkins 2.492.2 Setup

Completed the Jenkins 2.492.2 Setup Wizard

Click the Finish button to exit the Setup Wizard.


Thank you for downloading Windo
Sign in [Jenkins]
localhost:8080/login?from=%2F
Getting Started

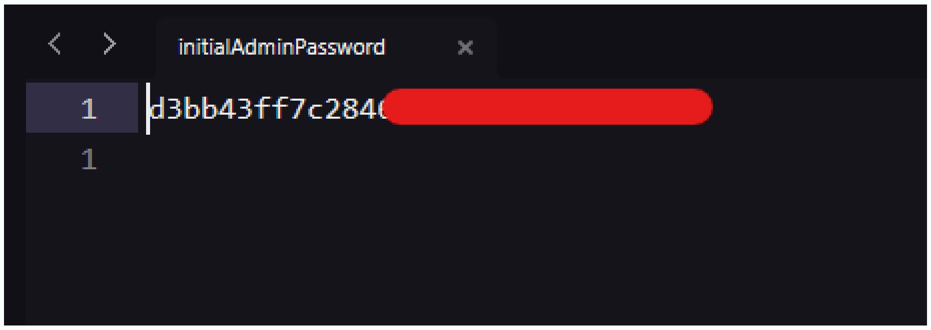
Unlock Jenkins

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:

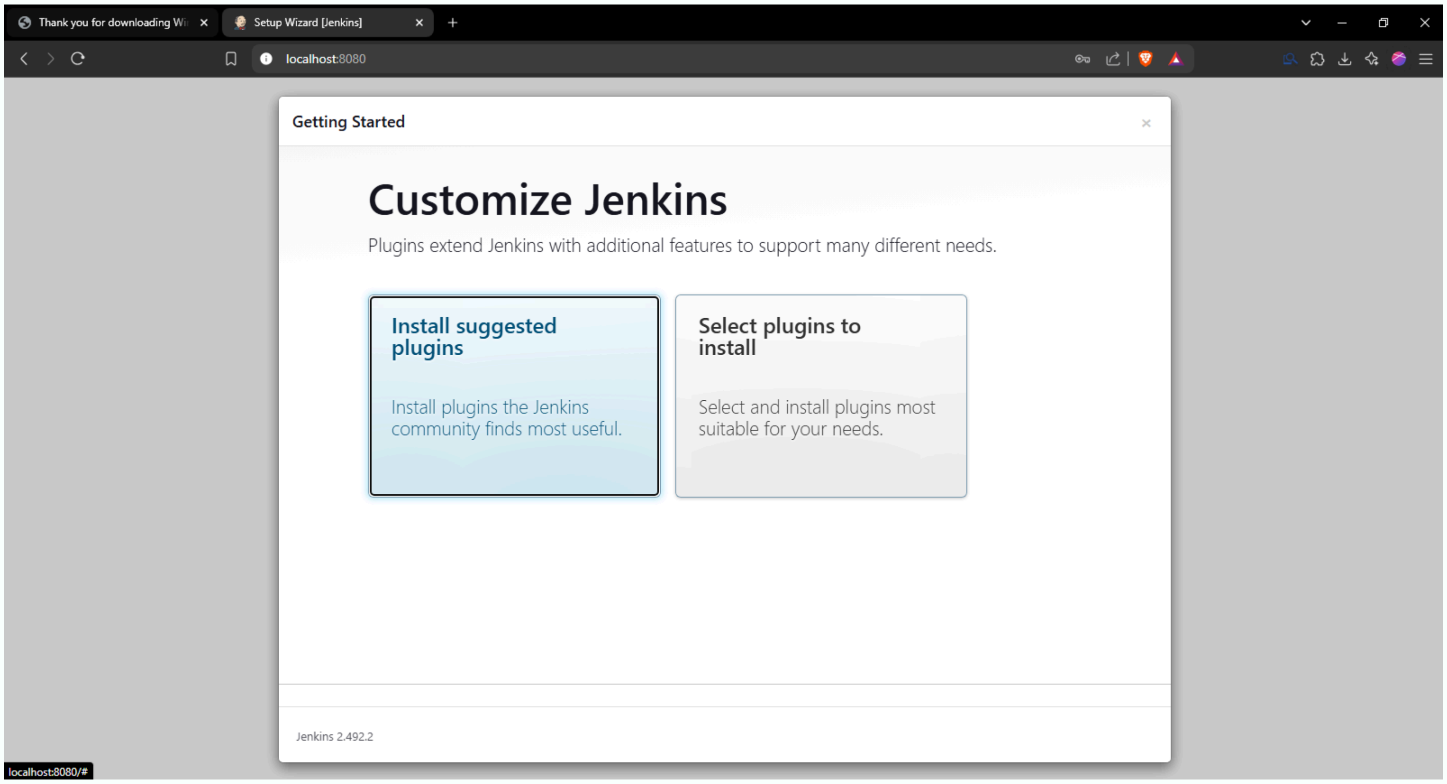
`C:\ProgramData\Jenkins\.jenkins\secrets\initialAdminPassword`

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

Administrator password



```
d3bb43ff7c284[redacted]
```



Customize Jenkins

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

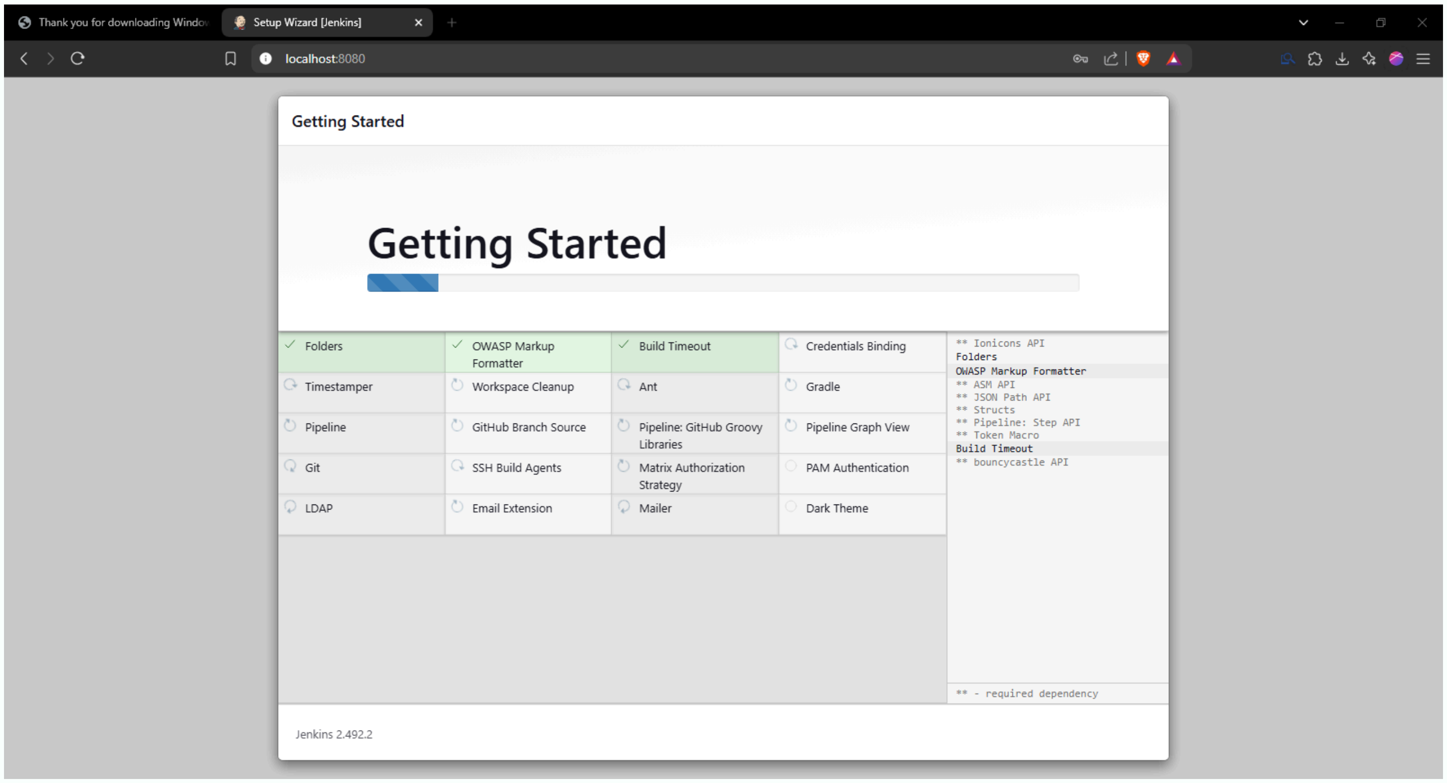
Install suggested plugins

Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.492.2



Getting Started

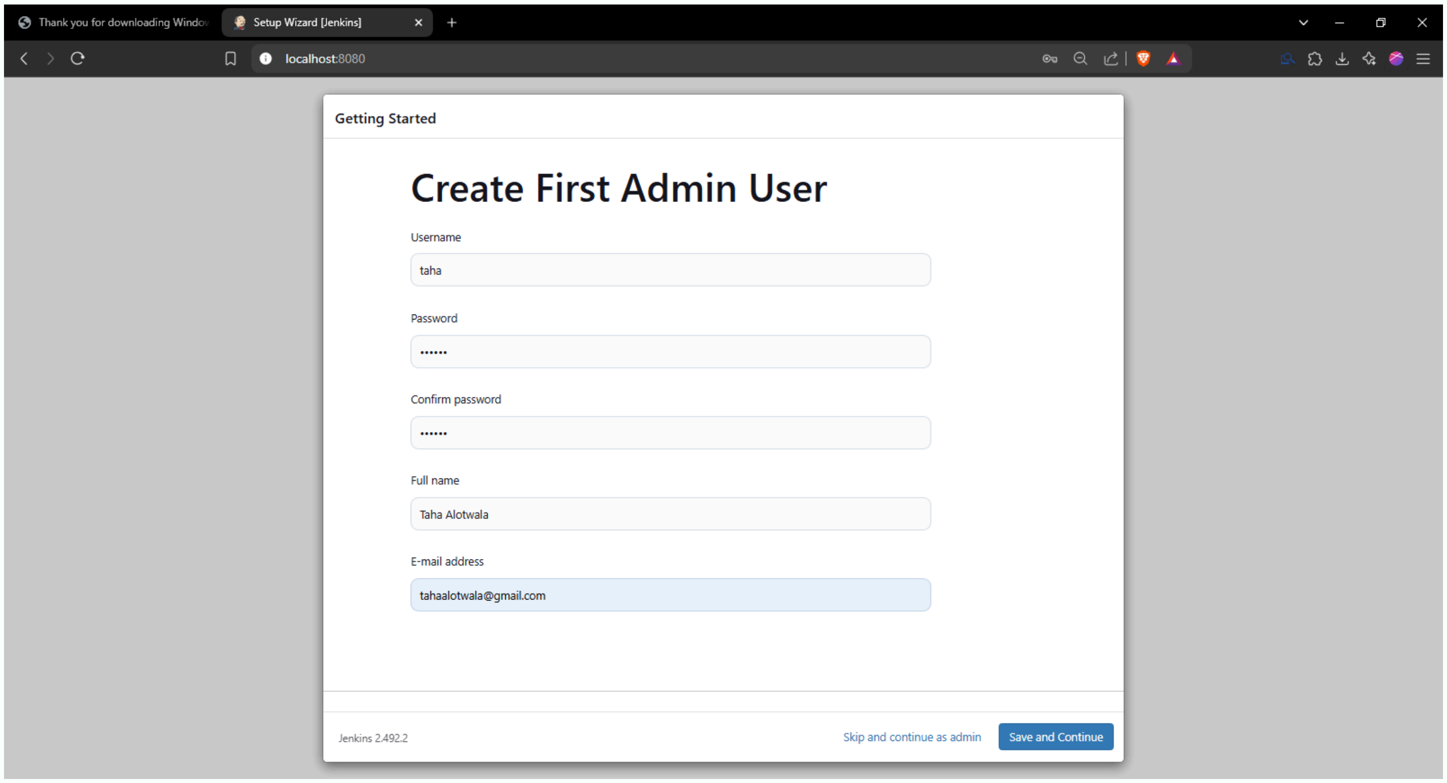
Getting Started

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

```
** Ionicons API
Folders
OWASP Markup Formatter
** ASM API
** JSON Path API
** Structs
** Pipeline: Step API
** Token Macro
Build Timeout
** bouncycastle API
```

```
** - required dependency
```

Jenkins 2.492.2



Getting Started

Create First Admin User

Username: taha

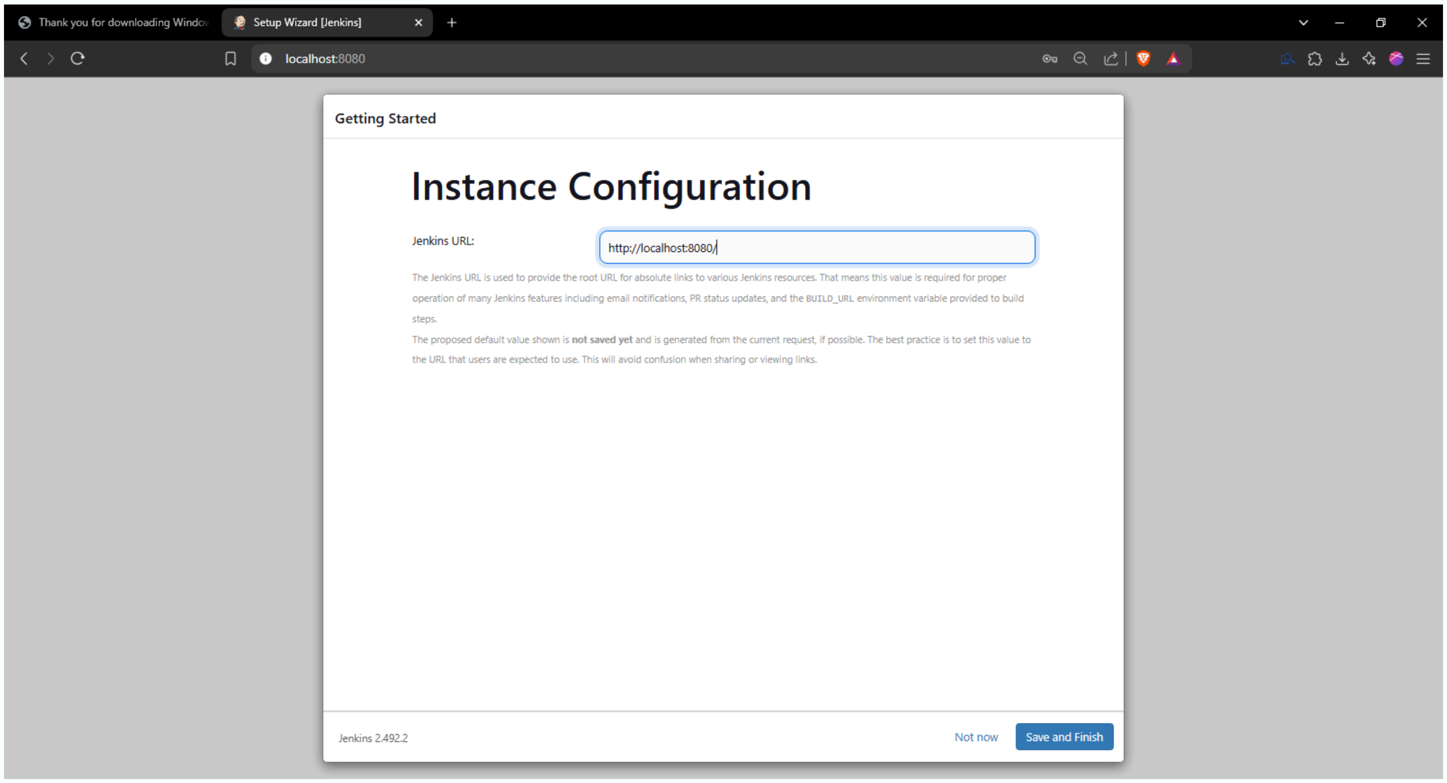
Password: ••••••

Confirm password: ••••••

Full name: Taha Alotwala

E-mail address: tahaalotwala@gmail.com

Jenkins 2.492.2



Getting Started

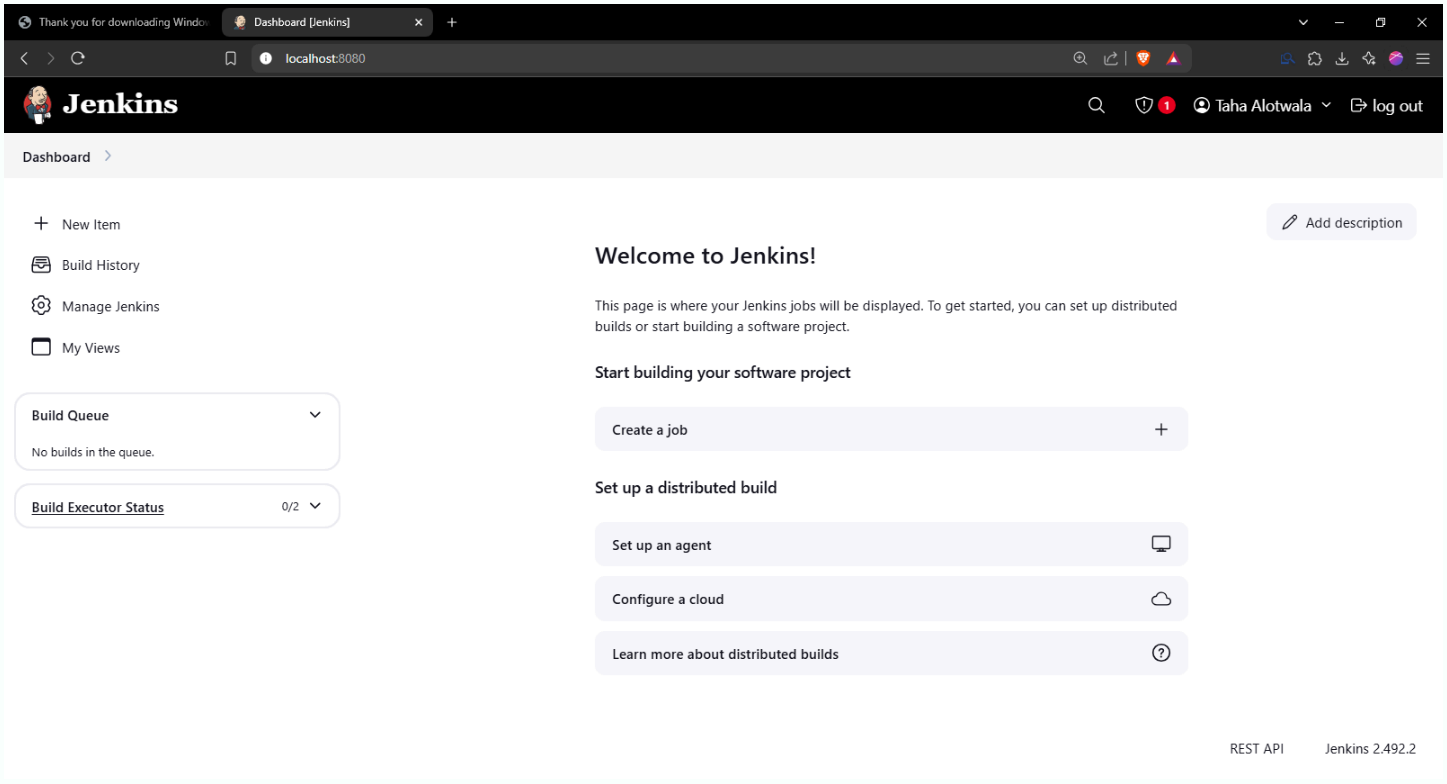
Instance Configuration

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 2.492.2


Thank you for downloading Windo | Dashboard [Jenkins]

localhost:8080

Jenkins — Taha Alotwala — log out

Dashboard

New Item
Build History
Manage Jenkins
My Views

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.

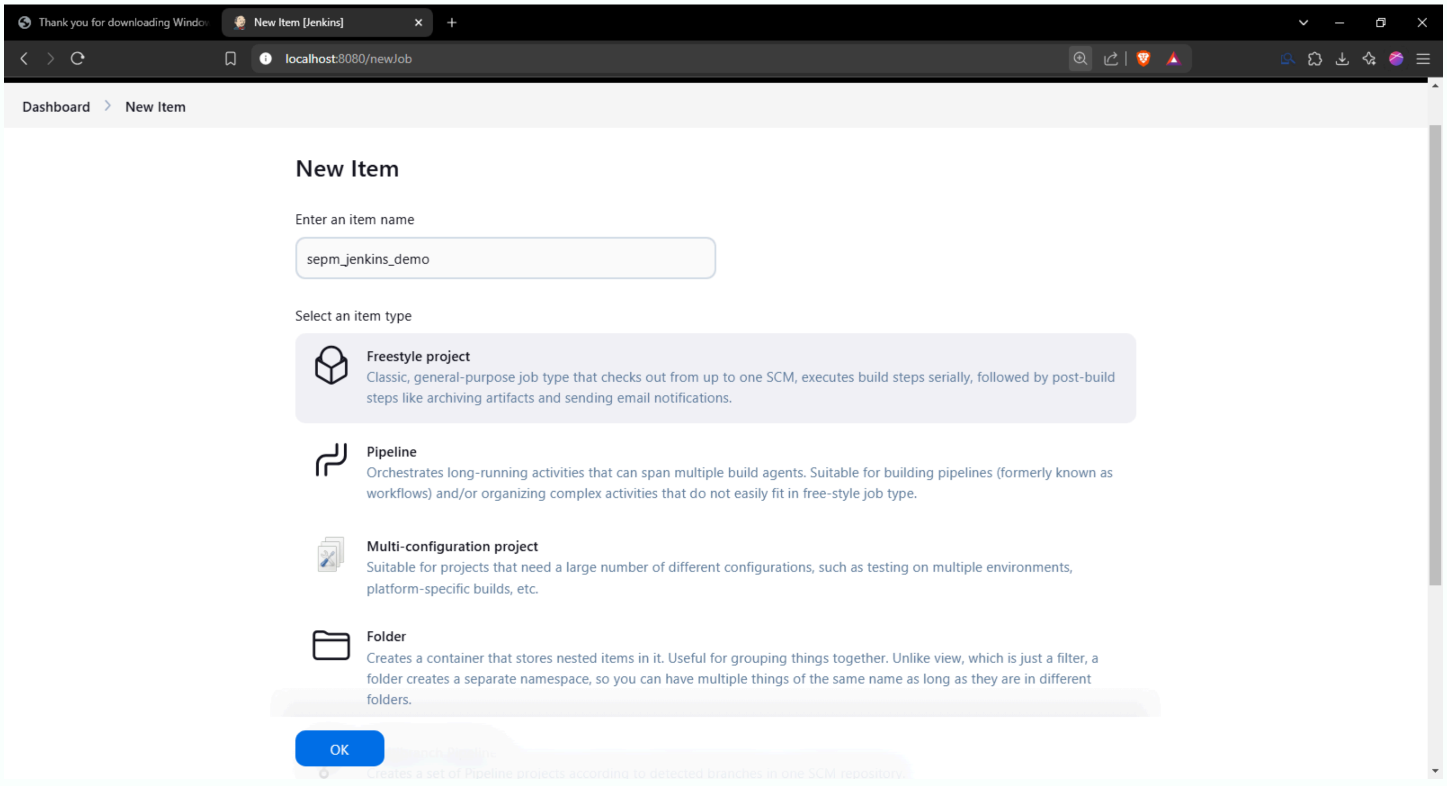
Start building your software project

Create a job

Set up a distributed build

Set up an agent
Configure a cloud
Learn more about distributed builds

REST API   Jenkins 2.492.2


Thank you for downloading Windo | New Item [Jenkins]

localhost:8080/newJob

Dashboard > New Item

New Item

Enter an item name

sepm_jenkins_demo

Select an item type

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.

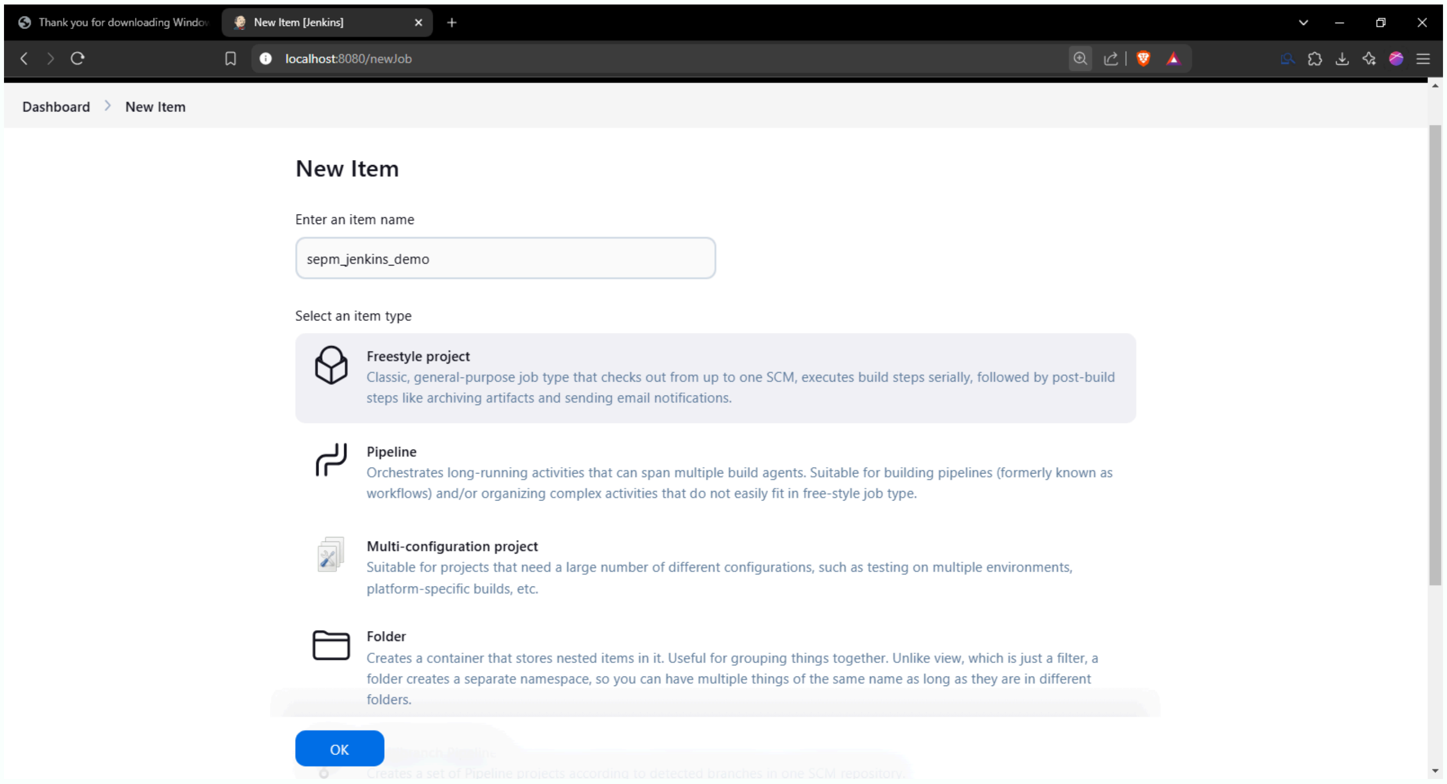
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

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.

OK

Creates a set of Pipeline projects according to detected branches in one SCM repository.


Dashboard > New Item

New Item

Enter an item name

sepm_jenkins_demo

Select an item type

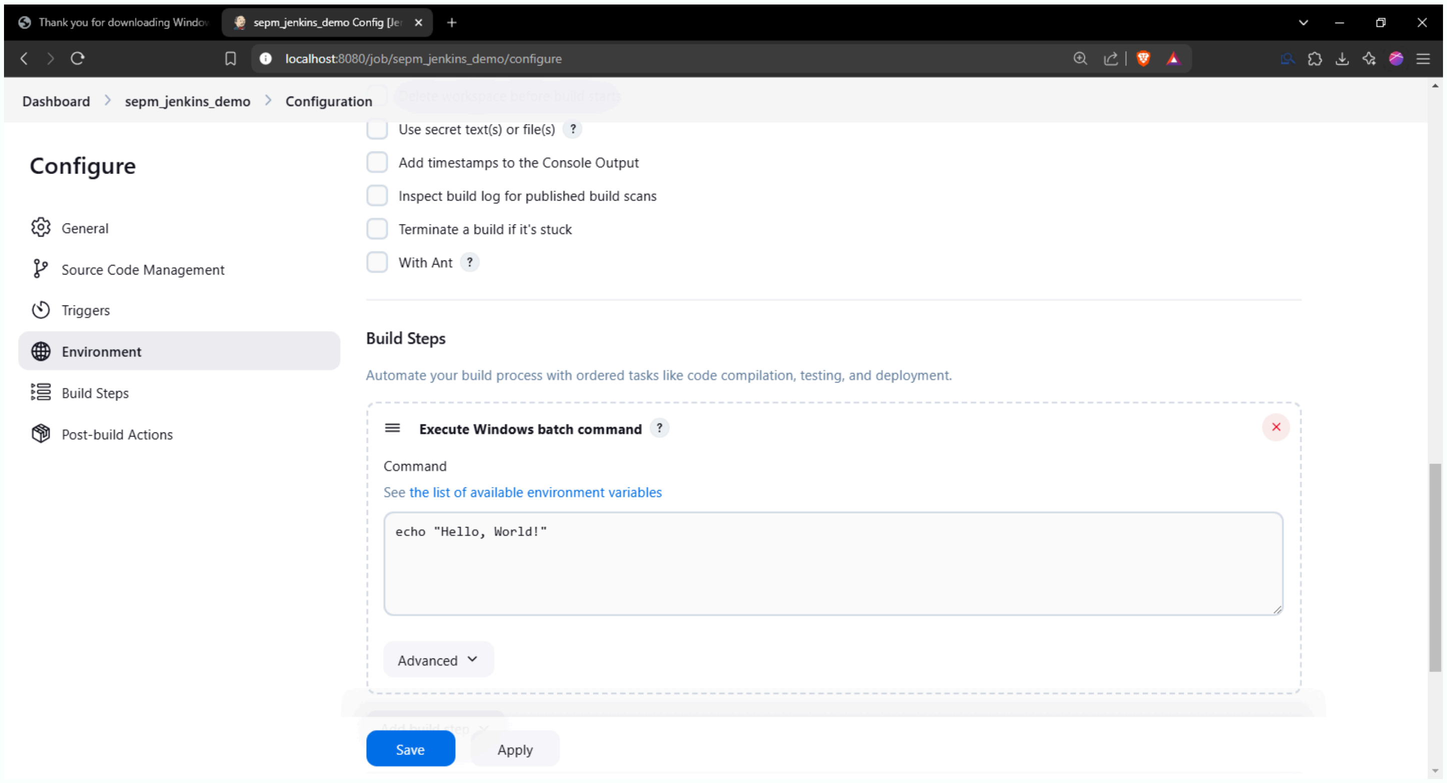
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

OK


Dashboard > sepm_jenkins_demo > Configuration

Configure

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

- [ ] 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

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

Execute Windows batch command

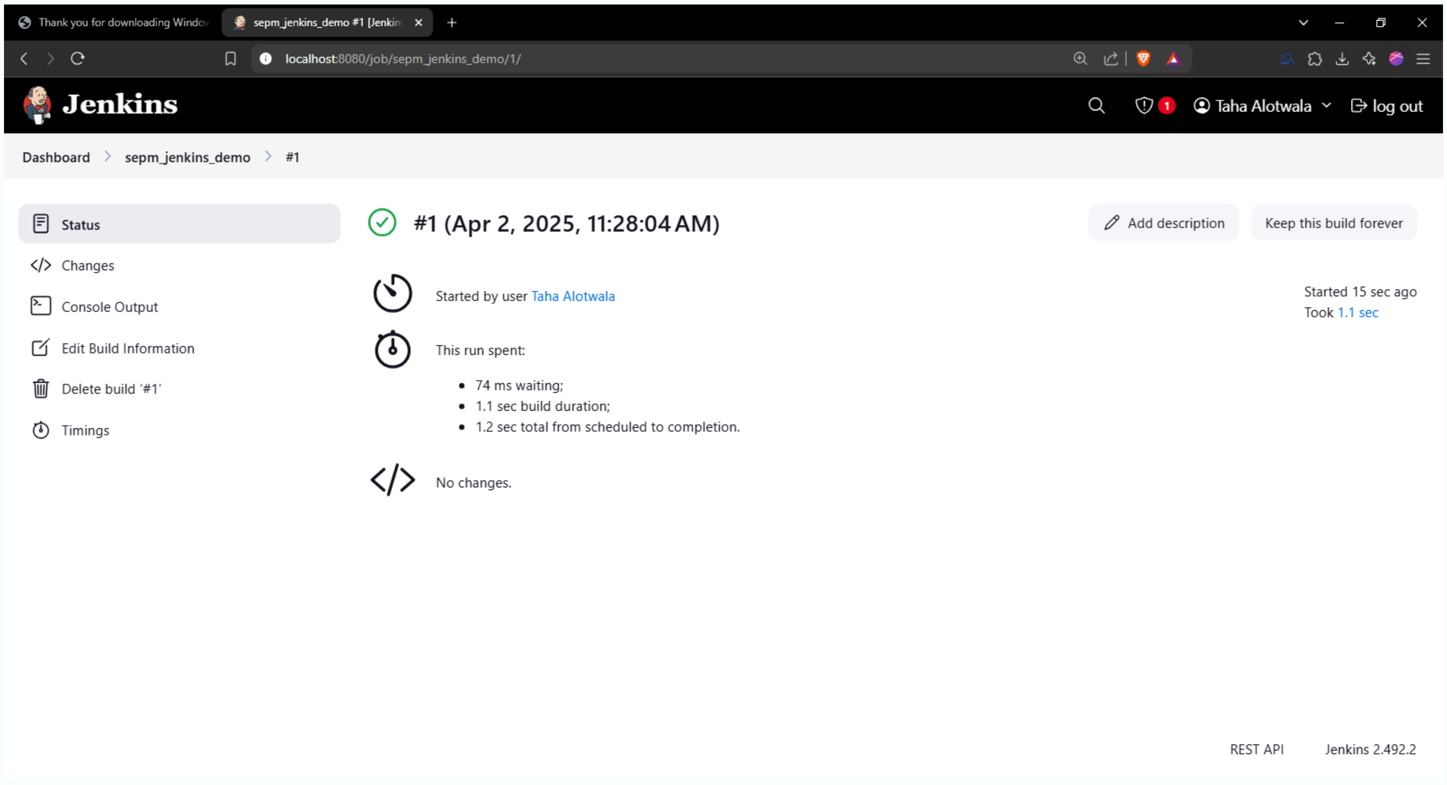
Command

See the list of available environment variables

```
echo "Hello, World!"
```

Advanced

Save Apply



#1 (Apr 2, 2025, 11:28:04 AM)

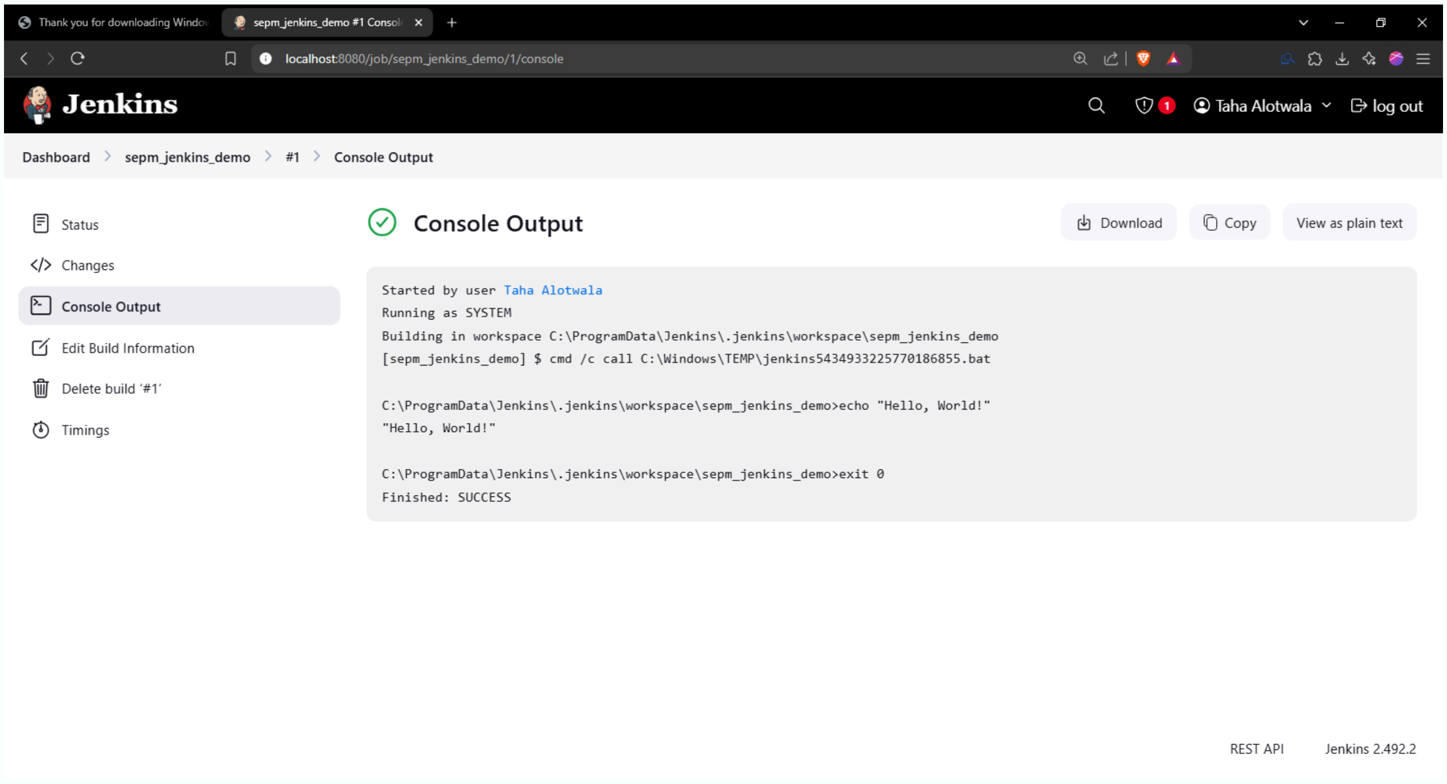
Started by user Taha Alotwala

Started 15 sec ago
Took 1.1 sec

This run spent:

- 74 ms waiting;
- 1.1 sec build duration;
- 1.2 sec total from scheduled to completion.

No changes.


Thank you for downloading Windo
sepm_jenkins_demo #1 Consol
localhost:8080/job/sepm_jenkins_demo/1/console
Jenkins
Taha Alotwala
log out
Dashboard > sepm_jenkins_demo > #1 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#1'
Timings
Download
Copy
View as plain text
REST API
Jenkins 2.492.2

Console Output

```
Started by user Taha Alotwala
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\sepm_jenkins_demo
[sepm_jenkins_demo] $ cmd /c call C:\Windows\TEMP\jenkins5434933225770186855.bat

C:\ProgramData\Jenkins\.jenkins\workspace\sepm_jenkins_demo>echo "Hello, World!"
"Hello, World!"

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

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.