Step 9: Resolving the conflict - editing the file containing the conflict markers

Step 10: Resolving the conflict - re-applying your changes to new upstream version

Step 11: Completing the merge procedure

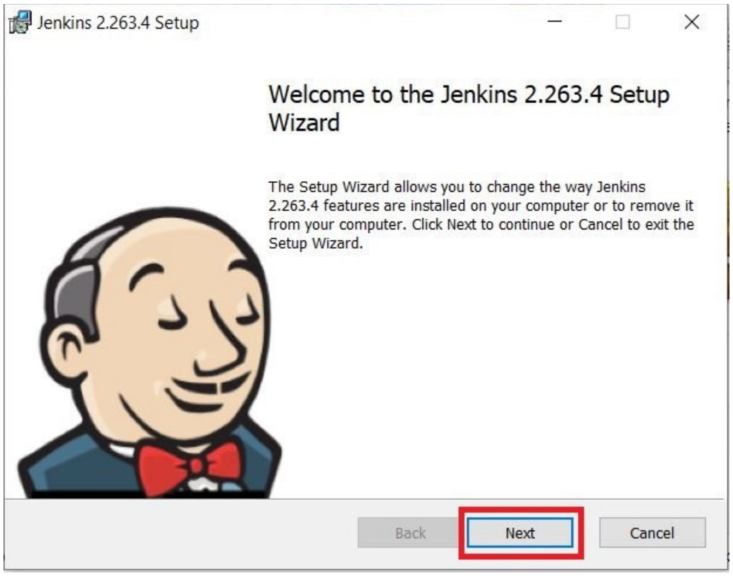
Step 12: Recovering from Errors

WEEK 10

1. Install and configure Jenkins

Step 1: Setup wizard

On opening the Windows Installer, an **Installation Setup Wizard** appears, Click **Next** on the Setup Wizard to start your installation.

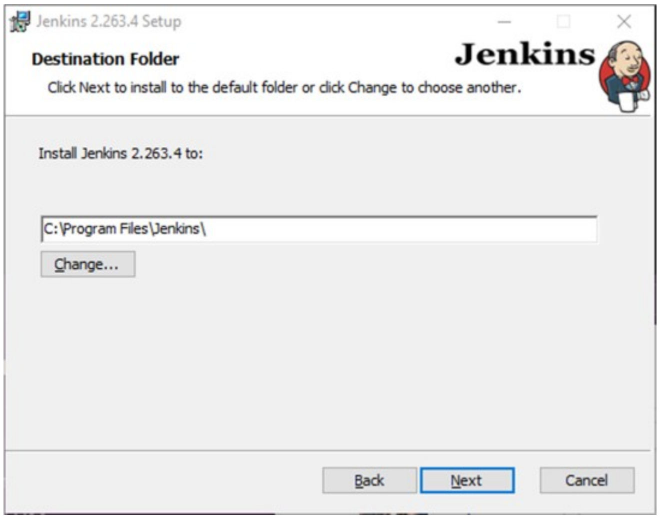


Welcome to the Jenkins 2.263.4 Setup Wizard

The Setup Wizard allows you to change the way Jenkins 2.263.4 features are installed on your computer or to remove it from your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Step 2: Select destination folder

Select the destination folder to store your Jenkins Installation and click **Next** to continue.



Jenkins 2.263.4 Setup

Destination Folder

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

Install Jenkins 2.263.4 to:

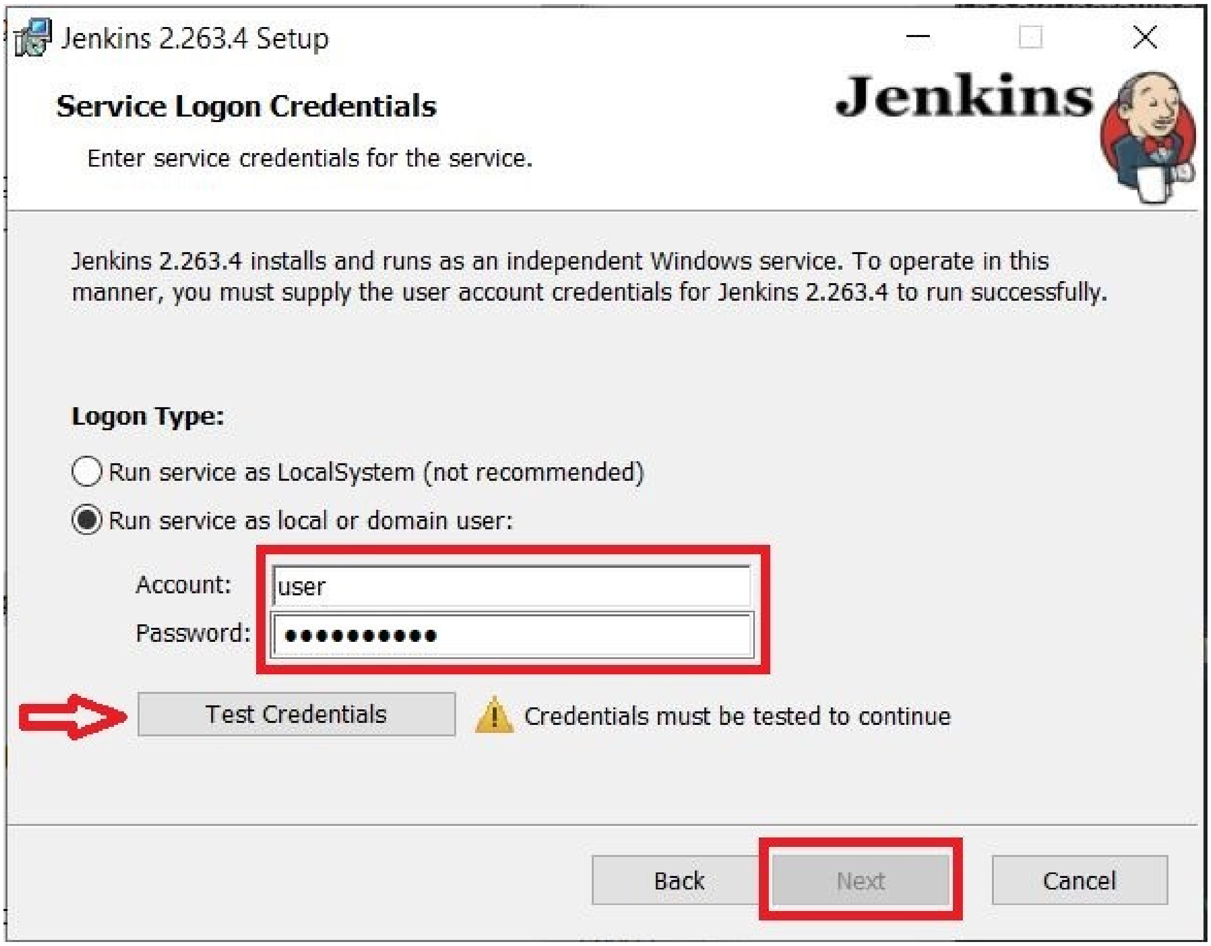
C:\Program Files\Jenkins\

Step 3: Service logon credentials

When Installing Jenkins, it is recommended to install and run Jenkins as an independent windows service using a **local or domain user** as it is much safer than running Jenkins using **LocalSystem(Windows equivalent of root)** which will grant Jenkins full access to your machine and services.

To run Jenkins service using a **local or domain user**, specify the domain user name and password with which you want to run Jenkins, click on **Test Credentials** to test your domain credentials and click on **Next**.

s



Jenkins 2.263.4 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.263.4 installs and runs as an independent Windows service. To operate in this manner, you must supply the user account credentials for Jenkins 2.263.4 to run successfully.

Logon Type:

○ Run service as LocalSystem (not recommended)

● Run service as local or domain user:

Account: user

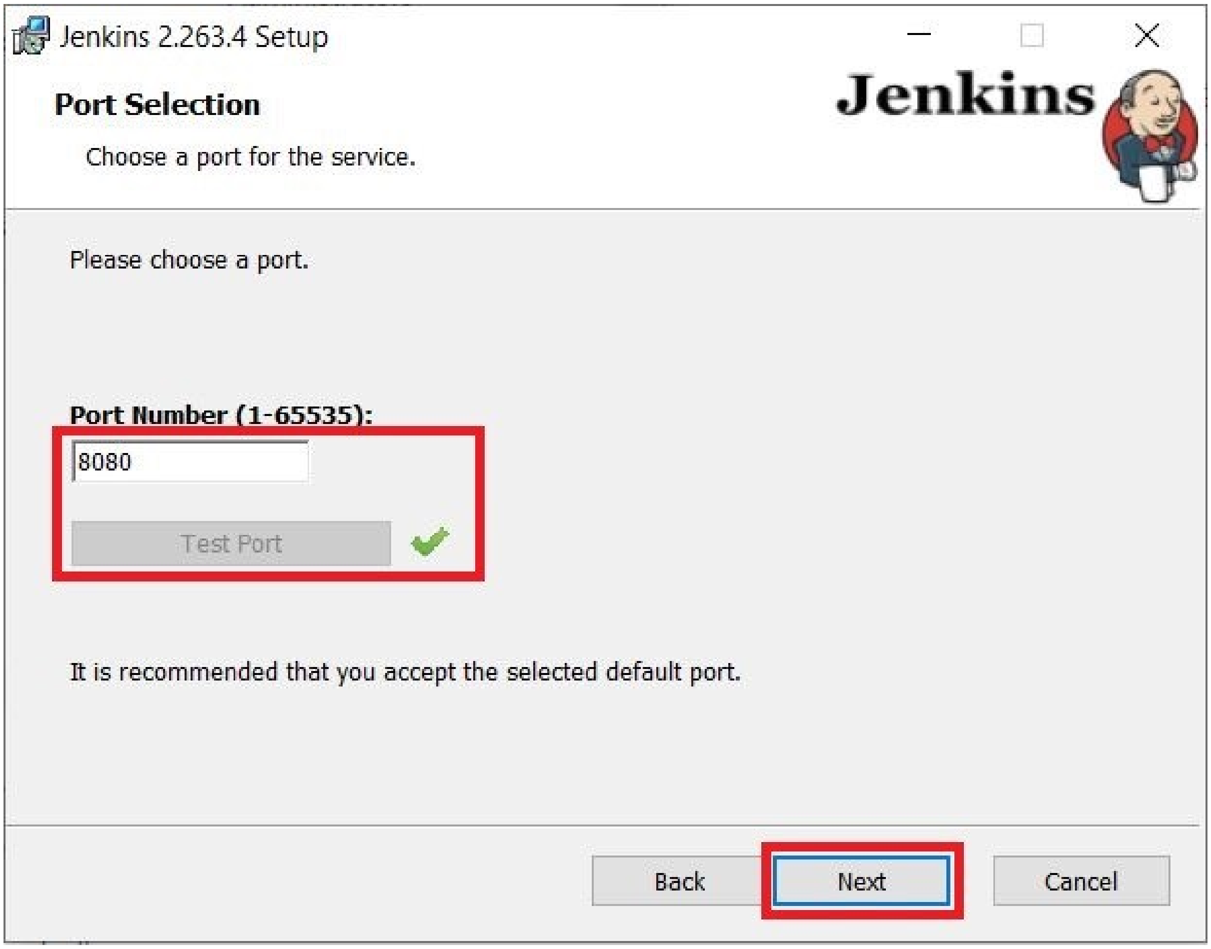
Password: ●●●●●●●●●●●

⚠ Credentials must be tested to continue

If you get **Invalid Logon** Error pop-up while trying to test your credentials, follow the steps explained here to resolve it.

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**.



Jenkins 2.263.4 Setup

Port Selection

Choose a port for the service.

Please choose a port.

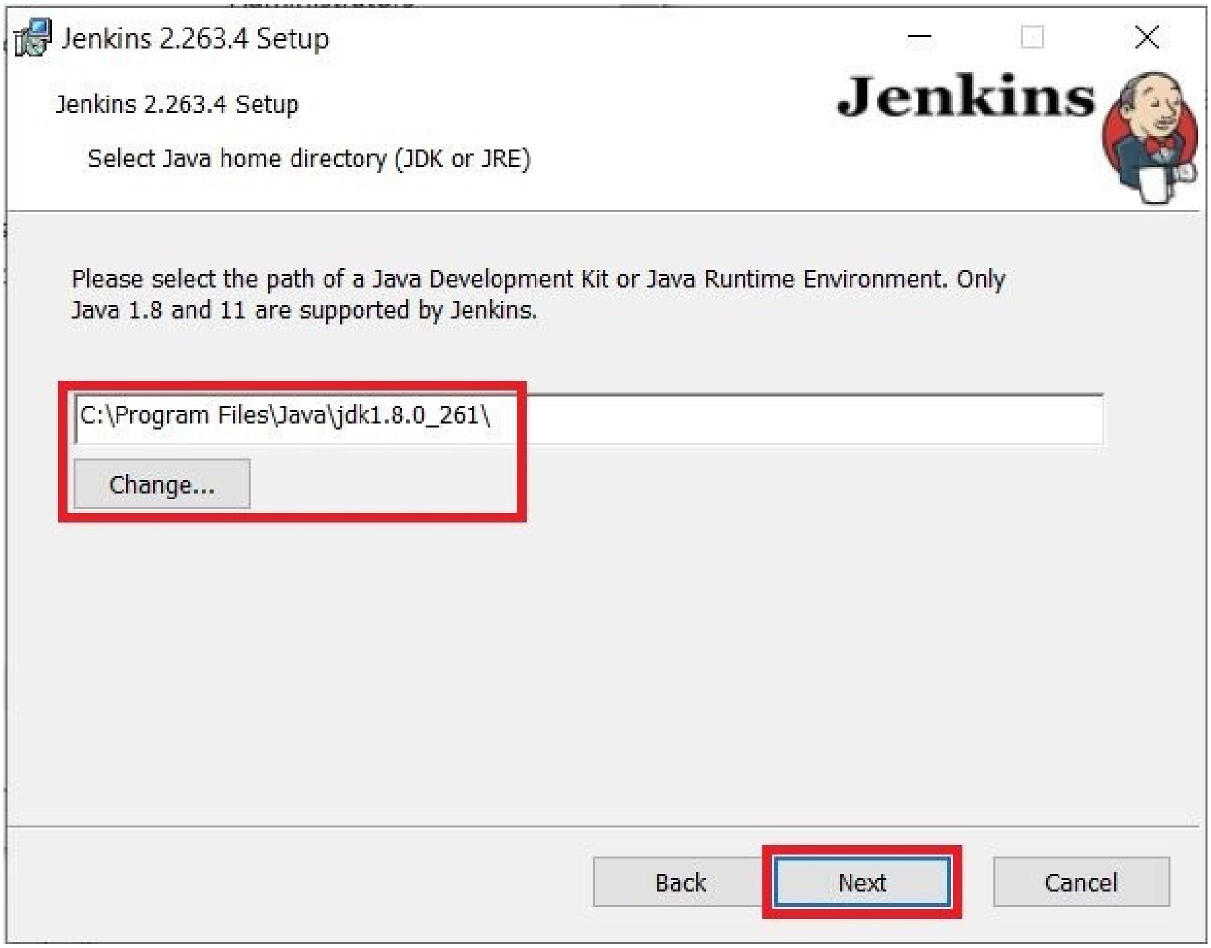
Port Number (1-65535): 8080

It is recommended that you accept the selected default port.

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.



Jenkins 2.263.4 Setup

Jenkins 2.263.4 Setup

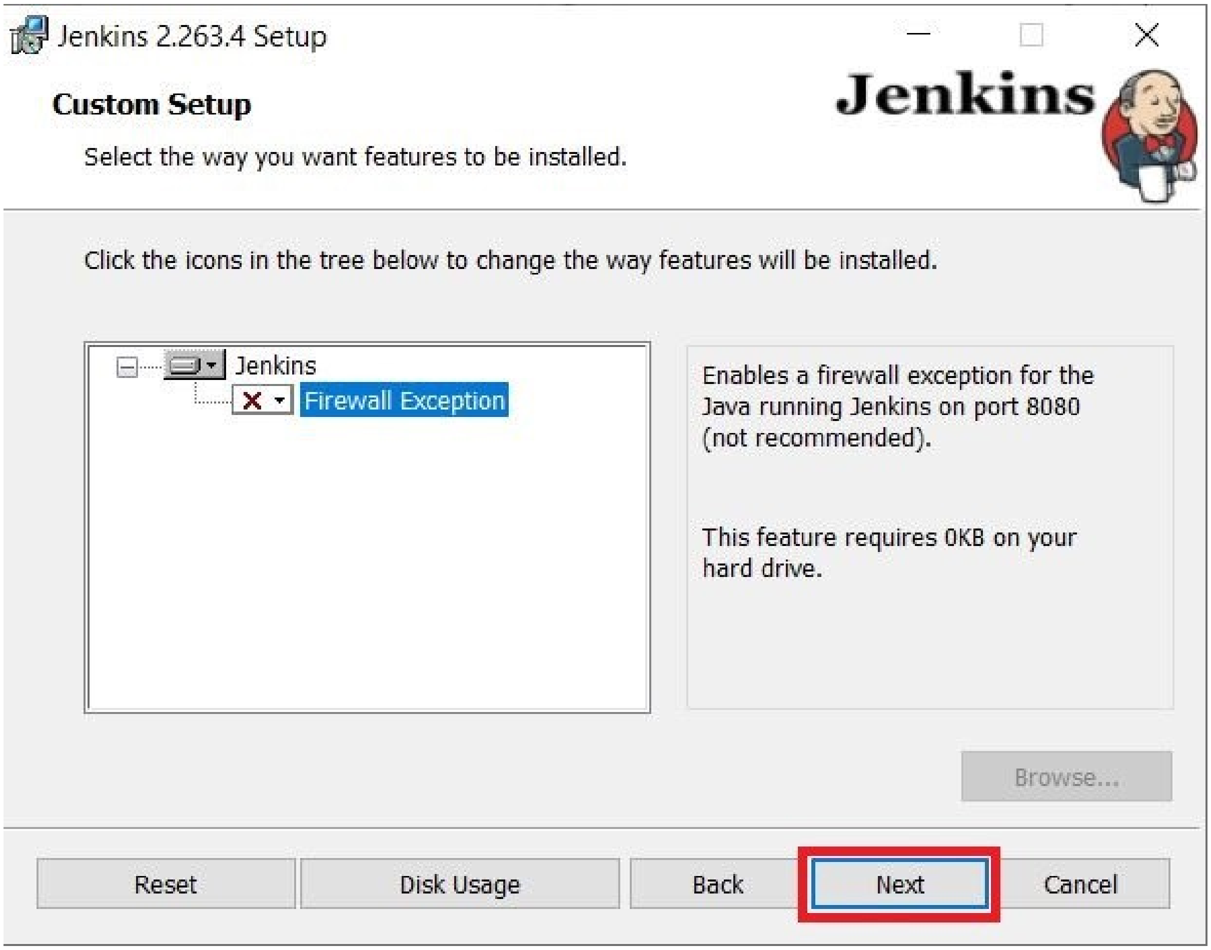
Select Java home directory (JDK or JRE)

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\jdk1.8.0_261\

Step 6: Custom setup

Select other services that need to be installed with Jenkins and click on **Next**.



Jenkins 2.263.4 Setup

Custom Setup

Select the way you want features to be installed.

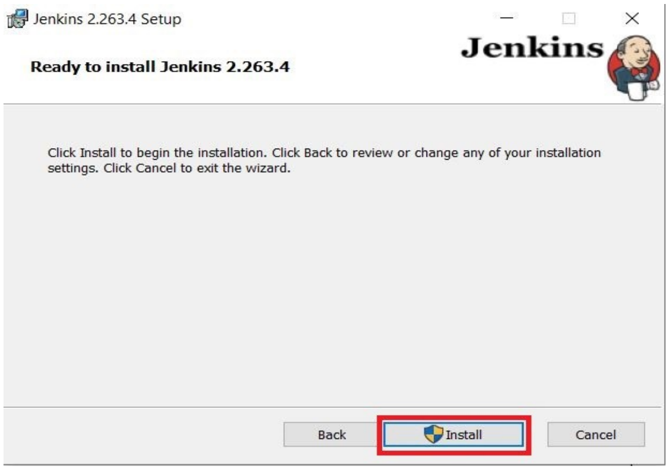
Click the icons in the tree below to change the way features will be installed.

Enables a firewall exception for the Java running Jenkins on port 8080 (not recommended).

This feature requires 0KB on your hard drive.

Step 7: Install Jenkins

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

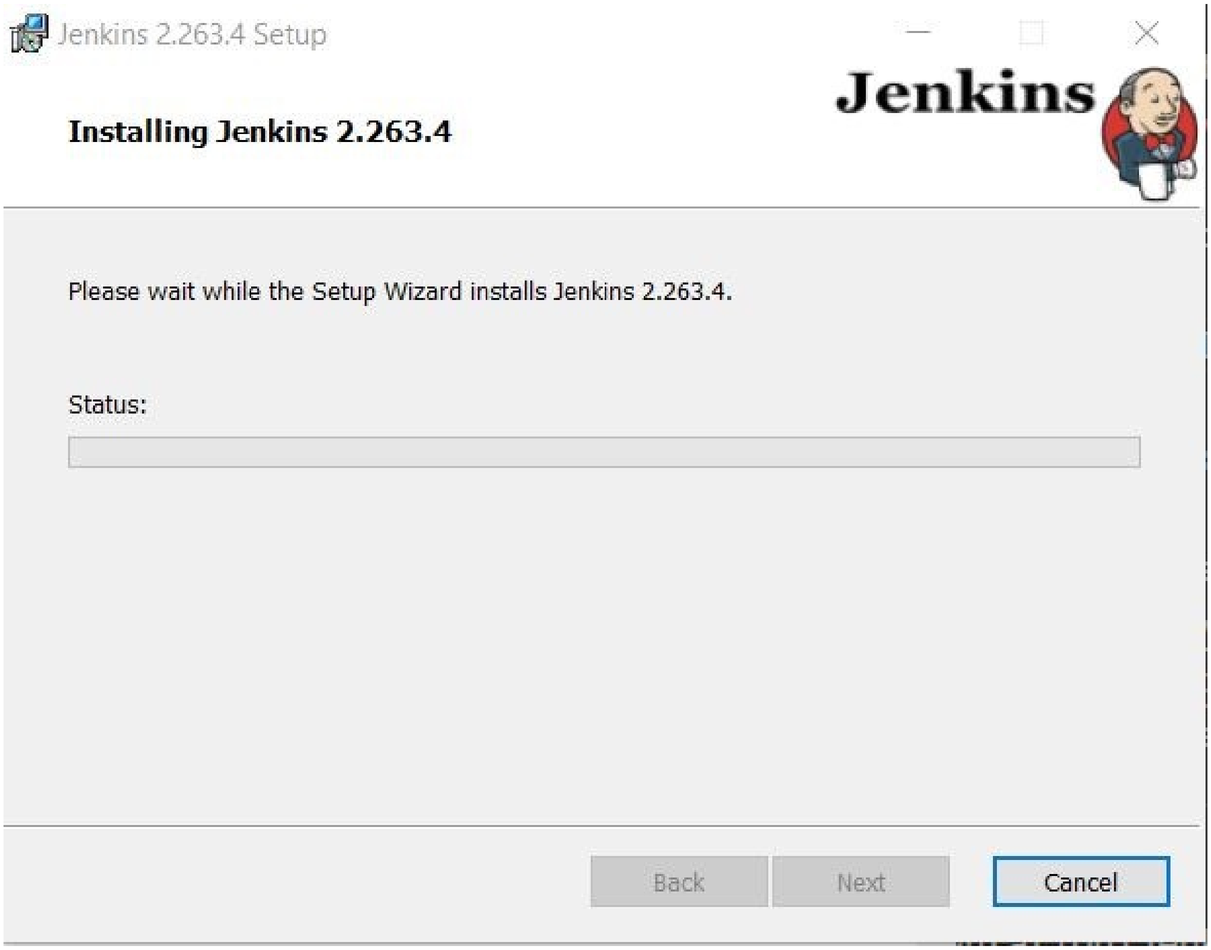


Jenkins 2.263.4 Setup

Ready to install Jenkins 2.263.4

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

Additionally, clicking on the **Install** button will show the progress bar of installation, as shown below:



Jenkins 2.263.4 Setup

Installing Jenkins 2.263.4

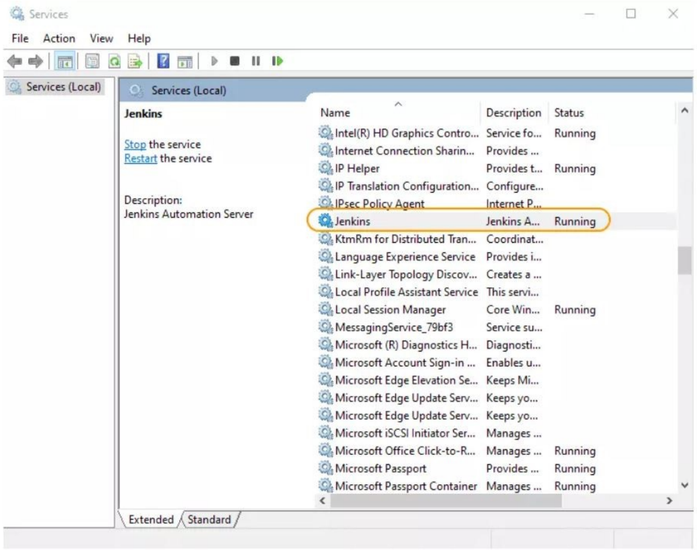
Please wait while the Setup Wizard installs Jenkins 2.263.4.

Status:

Step 8: Finish Jenkins installation

Once the installation completes, click on **Finish** to complete the installation.

Jenkins will be installed as a **Windows Service**. You can validate this by browsing the **services** section, as shown below:



Jenkins

Stop the service
Restart the service

Description:
Jenkins Automation Server

Name	Description	Status
Intel(R) HD Graphics Contro...	Service fo...	Running
Internet Connection Sharin...	Provides ...	
IP Helper	Provides t...	Running
IP Translation Configuration...	Configure...	
IPsec Policy Agent	Internet P...	
Jenkins	Jenkins A...	Running
KtmRm for Distributed Tran...	Coordinat...	
Language Experience Service	Provides i...	
Link-Layer Topology Discov...	Creates a ...	
Local Profile Assistant Service	This servi...	
Local Session Manager	Core Win...	Running
MessagingService_79bf3	Service su...	
Microsoft (R) Diagnostics H...	Diagnosti...	
Microsoft Account Sign-in ...	Enables u...	
Microsoft Edge Elevation Se...	Keeps Mi...	
Microsoft Edge Update Serv...	Keeps yo...	
Microsoft Edge Update Serv...	Keeps yo...	
Microsoft iSCSI Initiator Ser...	Manages ...	
Microsoft Office Click-to-R...	Manages ...	Running
Microsoft Passport	Provides ...	Running
Microsoft Passport Container	Manages ...	Running

See the upgrade steps when you upgrade to a new release.

Post-installation setup wizard

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

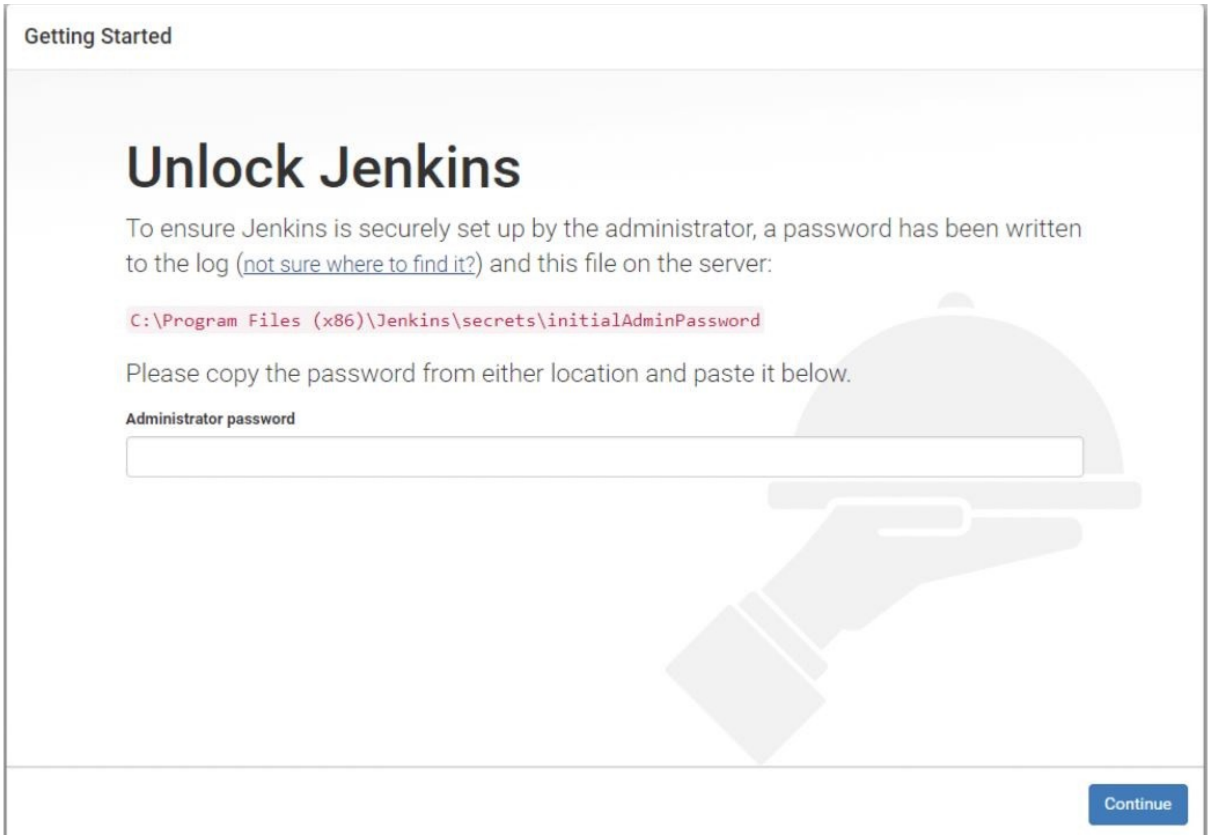
This setup wizard takes you through a few quick "one-off" steps to unlock Jenkins, customize it with plugins and create the first administrator user through which you can continue accessing Jenkins.

Unlocking Jenkins

When you first access a new Jenkins instance, 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.



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:\Program Files (x86)\Jenkins\secrets\initialAdminPassword
```

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

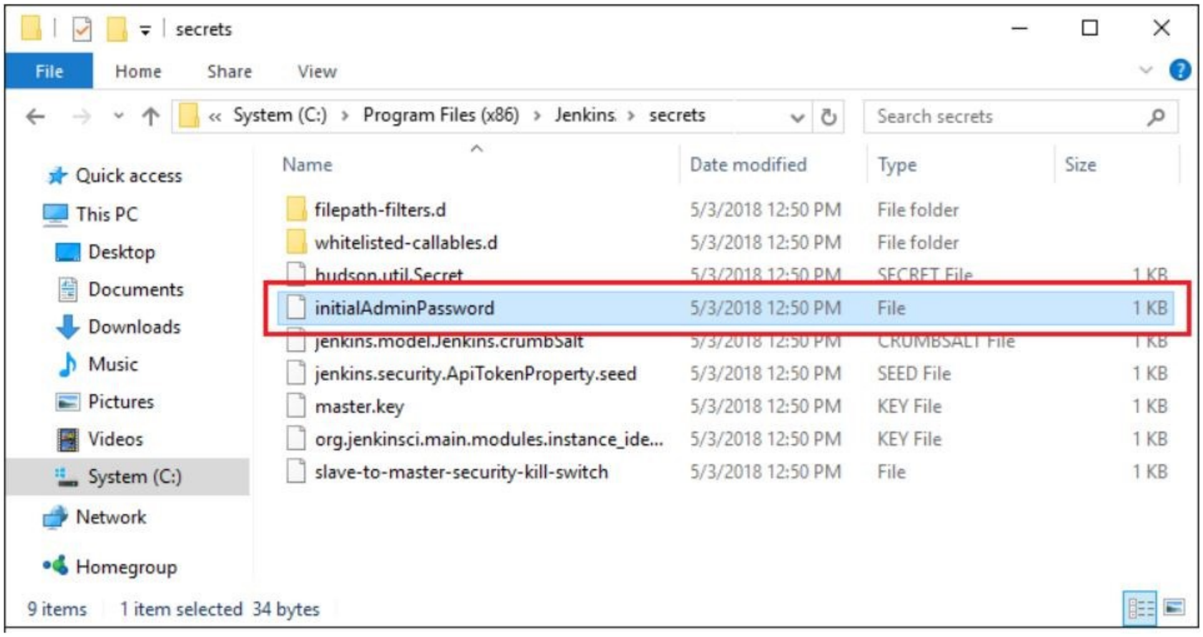
Administrator password

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.

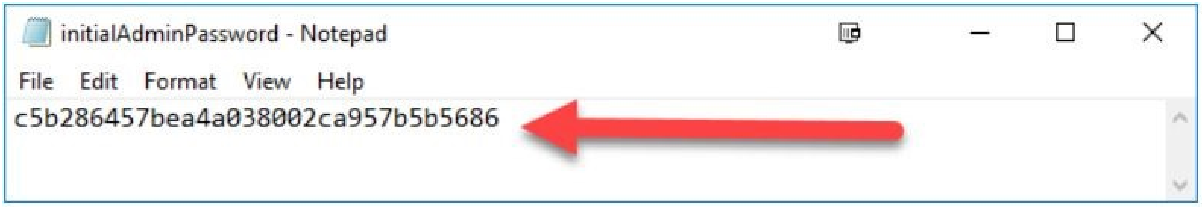


Name	Date modified	Type	Size
filepath-filters.d	5/3/2018 12:50 PM	File folder	
whitelisted-callables.d	5/3/2018 12:50 PM	File folder	
hudson.util.Secret	5/3/2018 12:50 PM	SECRET File	1 KB
initialAdminPassword	5/3/2018 12:50 PM	File	1 KB
jenkins.model.Jenkins.crumbSalt	5/3/2018 12:50 PM	CRUMBSALT File	1 KB
jenkins.security.ApiTokenProperty.seed	5/3/2018 12:50 PM	SEED File	1 KB
master.key	5/3/2018 12:50 PM	KEY File	1 KB
org.jenkinsci.main.modules.instance_ide...	5/3/2018 12:50 PM	KEY File	1 KB
slave-to-master-security-kill-switch	5/3/2018 12:50 PM	File	1 KB

9 items 1 item selected 34 bytes

Step 3

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



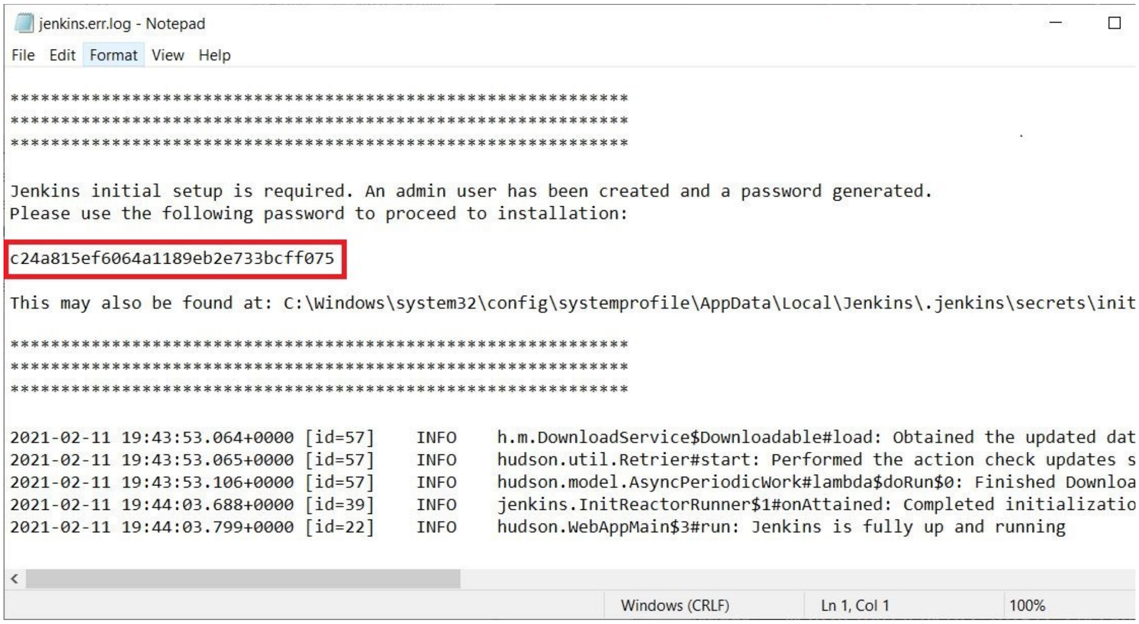
```
c5b286457bea4a038002ca957b5b5686
```

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.



```
*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

c24a815ef6064a1189eb2e733bcff075

This may also be found at: C:\Windows\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\secrets\init

*************************************************************
*************************************************************
*************************************************************

2021-02-11 19:43:53.064+0000 [id=57]    INFO    h.m.DownloadService$Downloadable#load: Obtained the updated dat
2021-02-11 19:43:53.065+0000 [id=57]    INFO    hudson.util.Retrier#start: Performed the action check updates s
2021-02-11 19:43:53.106+0000 [id=57]    INFO    hudson.model.AsyncPeriodicWork#lambda$doRun$0: Finished Downloa
2021-02-11 19:44:03.688+0000 [id=39]    INFO    jenkins.InitReactorRunner$1#onAttained: Completed initializatio
2021-02-11 19:44:03.799+0000 [id=22]    INFO    hudson.WebAppMain$3#run: Jenkins is fully up and running
```

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.

Customizing Jenkins with plugins

After unlocking Jenkins, the **Customize Jenkins** page appears. Here you can install any number of useful plugins as part of your initial setup.

Click one of the two options shown:

- **Install suggested plugins** - to install the recommended set of plugins, which are based on most common use cases.
- **Select plugins to install** - to choose which set of plugins to initially install. When you first access the plugin selection page, the suggested plugins are selected by default.

> If you are not sure what plugins you need, choose **Install suggested plugins**. You can install (or remove) additional Jenkins plugins at a later point in time via the **Manage Jenkins > Manage Plugins** page in Jenkins.

The setup wizard shows the progression of Jenkins being configured and your chosen set of Jenkins plugins being installed. This process may take a few minutes.

Creating the first administrator user

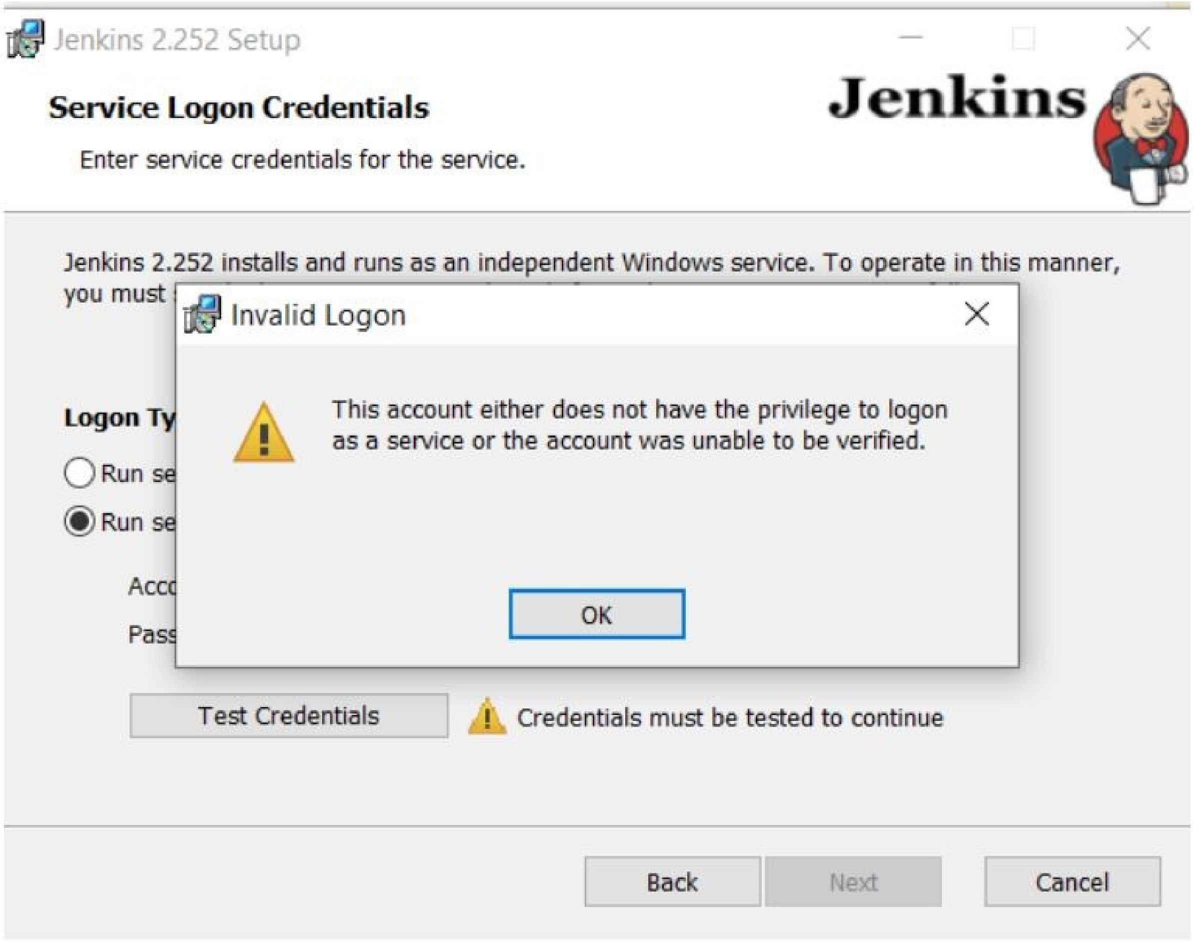
Finally, after customizing Jenkins with plugins, Jenkins asks you to create your first administrator user.

1. When the **Create First Admin User** page appears, specify the details for your administrator user in the respective fields and click **Save and Finish**.
2. When the **Jenkins is ready** page appears, click **Start using Jenkins**.
   Notes:

  - This page may indicate **Jenkins is almost ready!** instead and if so, click **Restart**.
  - If the page does not automatically refresh after a minute, use your web browser to refresh the page manually.
3. If required, log in to Jenkins with the credentials of the user you just created and you are ready to start using Jenkins!

Troubleshooting Windows installation

Invalid service logon credentials



Jenkins 2.252 Setup

Service Logon Credentials

Enter service credentials for the service.

Jenkins 2.252 installs and runs as an independent Windows service. To operate in this manner, you must

Invalid Logon

This account either does not have the privilege to logon as a service or the account was unable to be verified.

Logon Ty

○ Run se

● Run se

Credentials must be tested to continue

When installing a service to run under a domain user account, the account must have the right to logon as a service. This logon permission applies strictly to the local computer and must be granted in the Local Security Policy.

Perform the following steps below to edit the Local Security Policy of the computer you want to define the ‘logon as a service’ permission:

1. Logon to the computer with administrative privileges.
2. Open the **Administrative Tools** and open the **Local Security Policy**
3. Expand **Local Policy** and click on **User Rights Assignment**
4. In the right pane, right-click **Log on as a service** and select properties.

5. Click on the **Add User or Group…** button to add the new user.
6. In the **Select Users or Groups** dialogue, find the user you wish to enter and click **OK**
7. Click **OK** in the **Log on as a service Properties** to save changes.

After completing the steps above, try logging in again with the added user.

2. Create a container image for Hello world project

Create a container image for Hello world project And Setup build for container image using Jenkins (Hello world application)

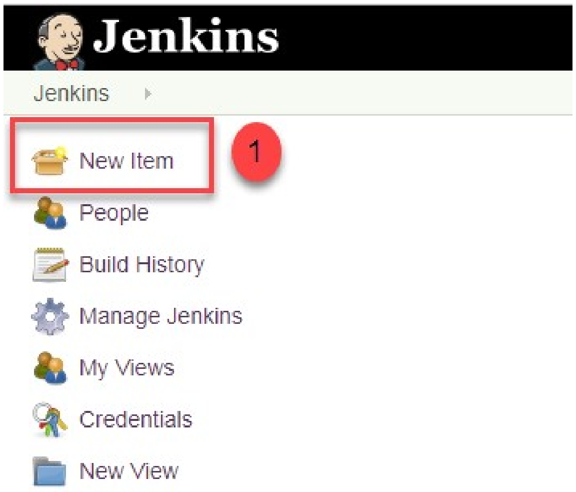
How to Create a New Build Job in Jenkins

The freestyle build job is a highly flexible and easy-to-use option. You can use it for any type of project; it is easy to set up, and many of its options appear in other build jobs. Below is a step by step process to create job in Jenkin.

Step 1) Login to Jenkins

To create a Jenkins freestyle job, log on to your Jenkins dashboard by visiting your Jenkins installation path. Usually, it will be hosted on localhost at http://localhost:8080 If you have installed Jenkins in another path, use the appropriate URL to access your dashboard as shown in the below Jenkins job creation example.

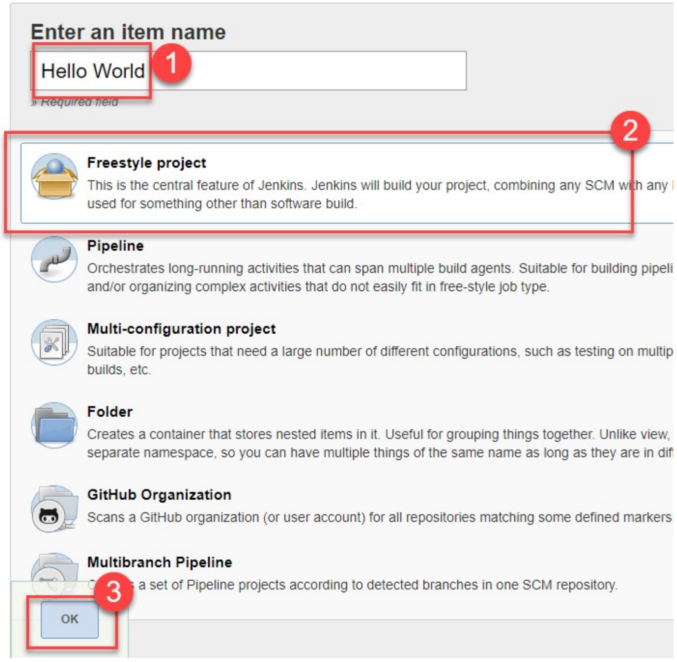
Step 2) Create New Item
Click on "**New Item**" at the top left-hand side of your dashboard.



Step 3) Enter Item details

In the next screen,

1. Enter the name of the item you want to create. We shall use the “Hello world” for this demo.
2. Select Freestyle project
3. Click Okay



Enter an item name

Hello World

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any
used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipeli
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 multip
builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view,
separate namespace, so you can have multiple things of the same name as long as they are in dif

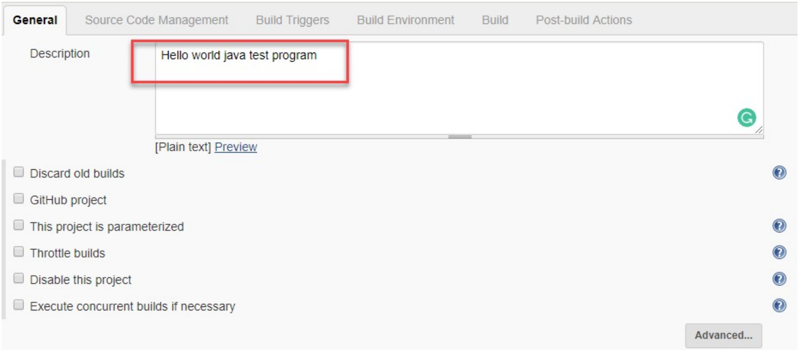
GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers

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

OK

Step 4) Enter Project details

Enter the details of the project you want to test.

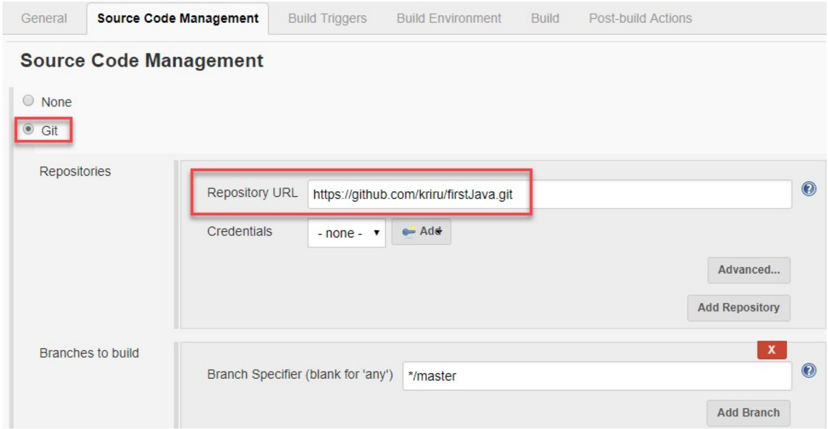


Description: Hello world java test program

☐ Discard old builds
☐ GitHub project
☐ This project is parameterized
☐ Throttle builds
☐ Disable this project
☐ Execute concurrent builds if necessary

Step 5) Enter repository URL

Under Source Code Management, Enter your repository URL. We have a test repository located at https://github.com/kriru/firstJava.git



Source Code Management

○ None

● Git

Repositories

Repository URL https://github.com/kriru/firstJava.git

Credentials - none -

Branches to build

Branch Specifier (blank for 'any') */master

It is also possible for you to use a local repository.

If your GitHub repository is private, Jenkins will first validate your login credentials with GitHub and only then pull the source code from your GitHub repository.

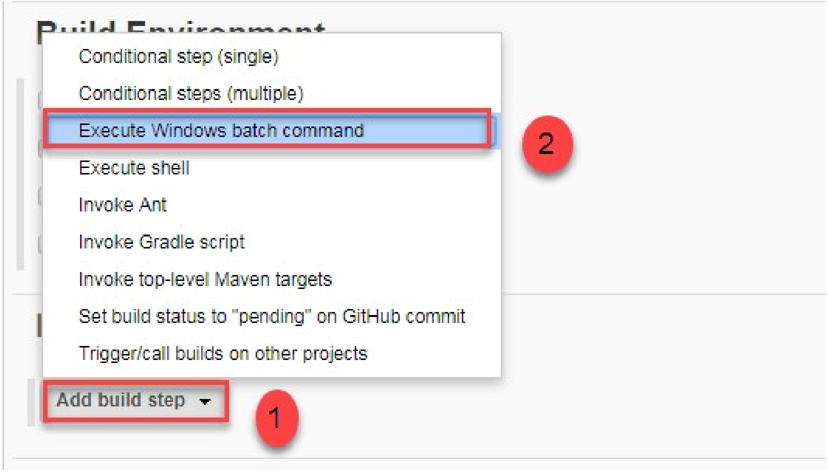
Step 6) Tweak the settings

Now that you have provided all the details, it’s time to build the code. Tweak the settings under the **build** section to build the code at the time you want. You can even schedule the build to happen periodically, at set times.

Under **build**,

1. Click on “**Add build step**”

2. Click on “**Execute Windows batch command**” and add the commands you want to execute during the build process.



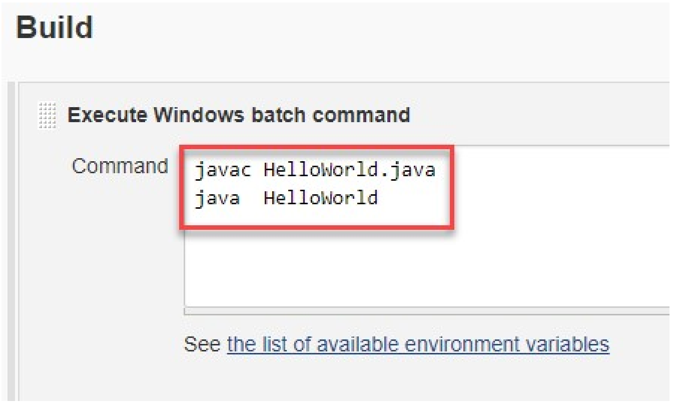
Build Environment
Conditional step (single)
Conditional steps (multiple)
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Set build status to "pending" on GitHub commit
Trigger/call builds on other projects
Add build step
2
1

Here, I have added the java commands to compile the java code.

I have added the following windows commands:

javac HelloWorld.java

java HelloWorld


Build

Execute Windows batch command

Command

```
javac HelloWorld.java
java  HelloWorld
```

See the list of available environment variables

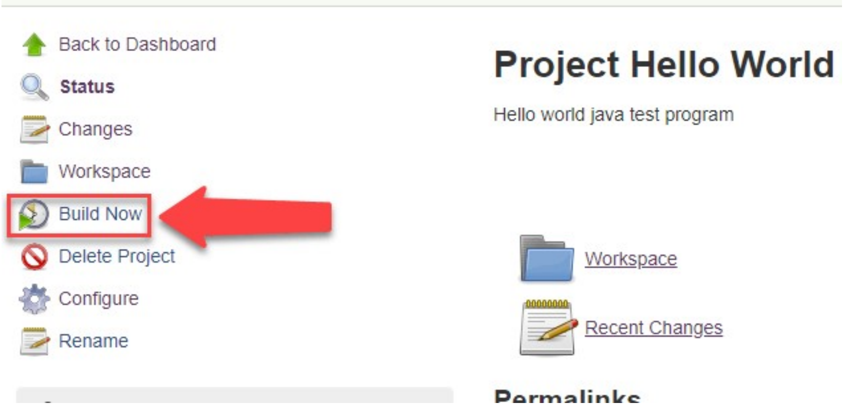
Step 7) Save the project

When you have entered all the data,

1. Click **Apply**
2. **Save** the project.

Step 8) Build Source code

Now, in the main screen, Click the **Build Now** button on the left-hand side to build the source code.

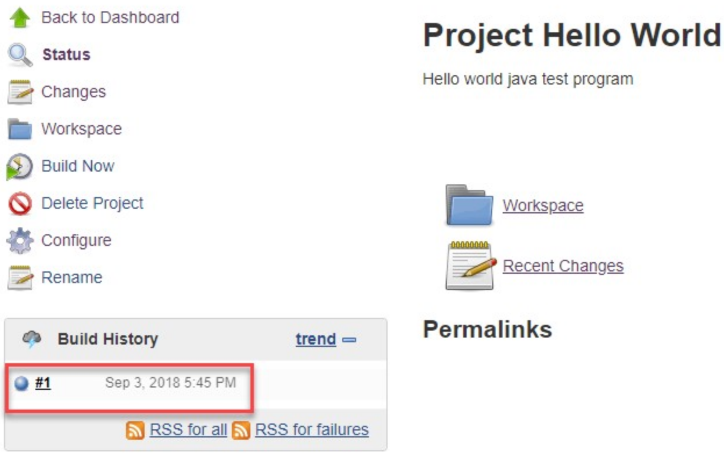


Project Hello World

Hello world java test program

Step 9) Check the status

After clicking on **Build now,** you can see the status of the build you run under **Build History**.



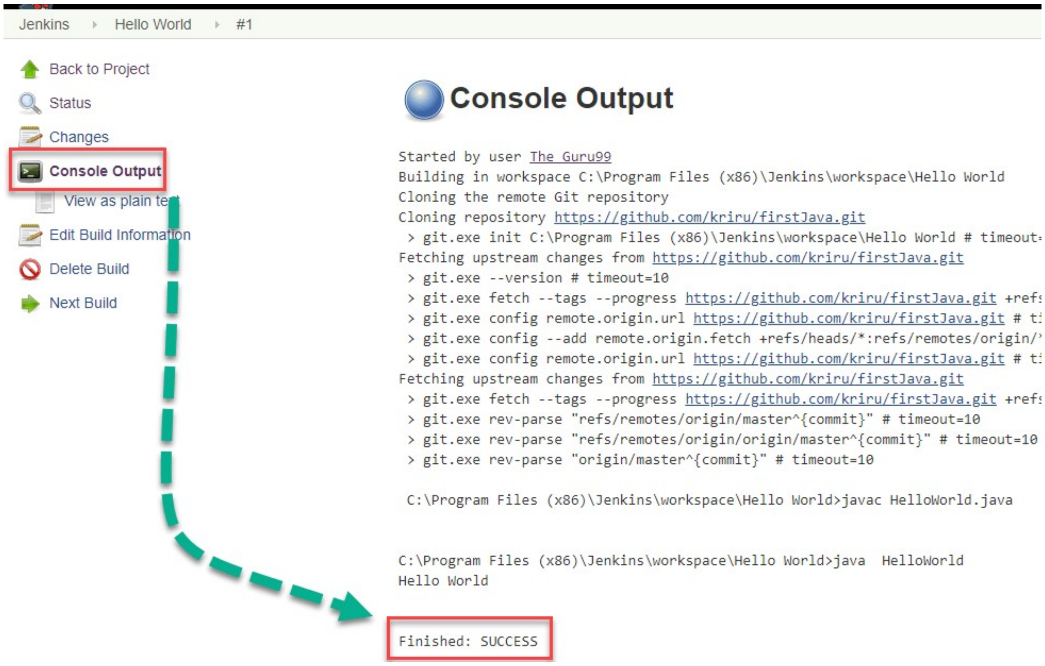
Project Hello World

Hello world java test program

Permalinks

Step 10) See the console output

Click on the **build number** and then Click on **console output** to see the status of the build you run. It should show you a success message, provided you have followed the setup properly as shown in the below Jenkins create new job example.



Console Output

```
Started by user The Guru99
Building in workspace C:\Program Files (x86)\Jenkins\workspace\Hello World
Cloning the remote Git repository
Cloning repository https://github.com/kriru/firstJava.git
 > git.exe init C:\Program Files (x86)\Jenkins\workspace\Hello World # timeout=
Fetching upstream changes from https://github.com/kriru/firstJava.git
 > git.exe --version # timeout=10
 > git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +refs
 > git.exe config remote.origin.url https://github.com/kriru/firstJava.git # ti
 > git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/
 > git.exe config remote.origin.url https://github.com/kriru/firstJava.git # ti
Fetching upstream changes from https://github.com/kriru/firstJava.git
 > git.exe fetch --tags --progress https://github.com/kriru/firstJava.git +refs
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "refs/remotes/origin/origin/master^{commit}" # timeout=10
 > git.exe rev-parse "origin/master^{commit}" # timeout=10

 C:\Program Files (x86)\Jenkins\workspace\Hello World>javac HelloWorld.java


C:\Program Files (x86)\Jenkins\workspace\Hello World>java  HelloWorld
Hello World

Finished: SUCCESS
```

In sum, we have executed a HelloWorld program hosted on GitHub. Jenkin pulls the code from the remote repository and builds continuously at a frequency you define.