

Jenkins Installation

Jenkins is one of the most popular tools for continuous integration and continuous delivery on any platform. A java application, Jenkins has many plugins for automating almost everything at the infrastructure level. The use of Jenkins has widely increased rapidly due to a rich set of functionalities, which it provides in the form of plugins. In this document we will show a step by step guide of how to install Jenkins on a Windows platform.

Let's get started.

The installation is a 2 Step Process:

- 1) Install JDK
- 2) Install Jenkins

- **Step 1 Install JDK**

1. Go to [Java SE Downloads on Oracle.](#)
2. Currently Gatling only supports JDK8 so select the "**JDK Download**" of JDK8.

Java SE 8u171/ 8u172
Java SE 8u171 includes important bug fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u172 is a patch-set update, including all of 8u171 plus additional bug fixes (described in the release notes).
[Learn more](#) ▶

<ul style="list-style-type: none">▪ Installation Instructions▪ Release Notes▪ Oracle License▪ Java SE Licensing Information User Manual<ul style="list-style-type: none">▪ Includes Third Party Licenses▪ Certified System Configurations▪ Readme Files<ul style="list-style-type: none">▪ JDK ReadMe▪ JRE ReadMe	JDK DOWNLOAD ↓
	Server JRE DOWNLOAD ↓
	JRE DOWNLOAD ↓

3. Select the "**Accept License Agreement**" radio button.

Java SE Development Kit 8u171

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ **Accept License Agreement**
☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.97 MB	jdk-8u171-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.89 MB	jdk-8u171-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.05 MB	jdk-8u171-linux-i586.rpm
Linux x86	184.88 MB	jdk-8u171-linux-i586.tar.gz
Linux x64	167.14 MB	jdk-8u171-linux-x64.rpm
Linux x64	182.05 MB	jdk-8u171-linux-x64.tar.gz
Mac OS X x64	247.84 MB	jdk-8u171-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.83 MB	jdk-8u171-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.19 MB	jdk-8u171-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.6 MB	jdk-8u171-solaris-x64.tar.Z
Solaris x64	97.05 MB	jdk-8u171-solaris-x64.tar.gz
Windows x86	199.1 MB	jdk-8u171-windows-i586.exe
Windows x64	207.27 MB	jdk-8u171-windows-x64.exe

4. Select the appropriate installation. In this case, it is the **'Windows-64'** bit. Click the appropriate link and save the .exe file to your disk.

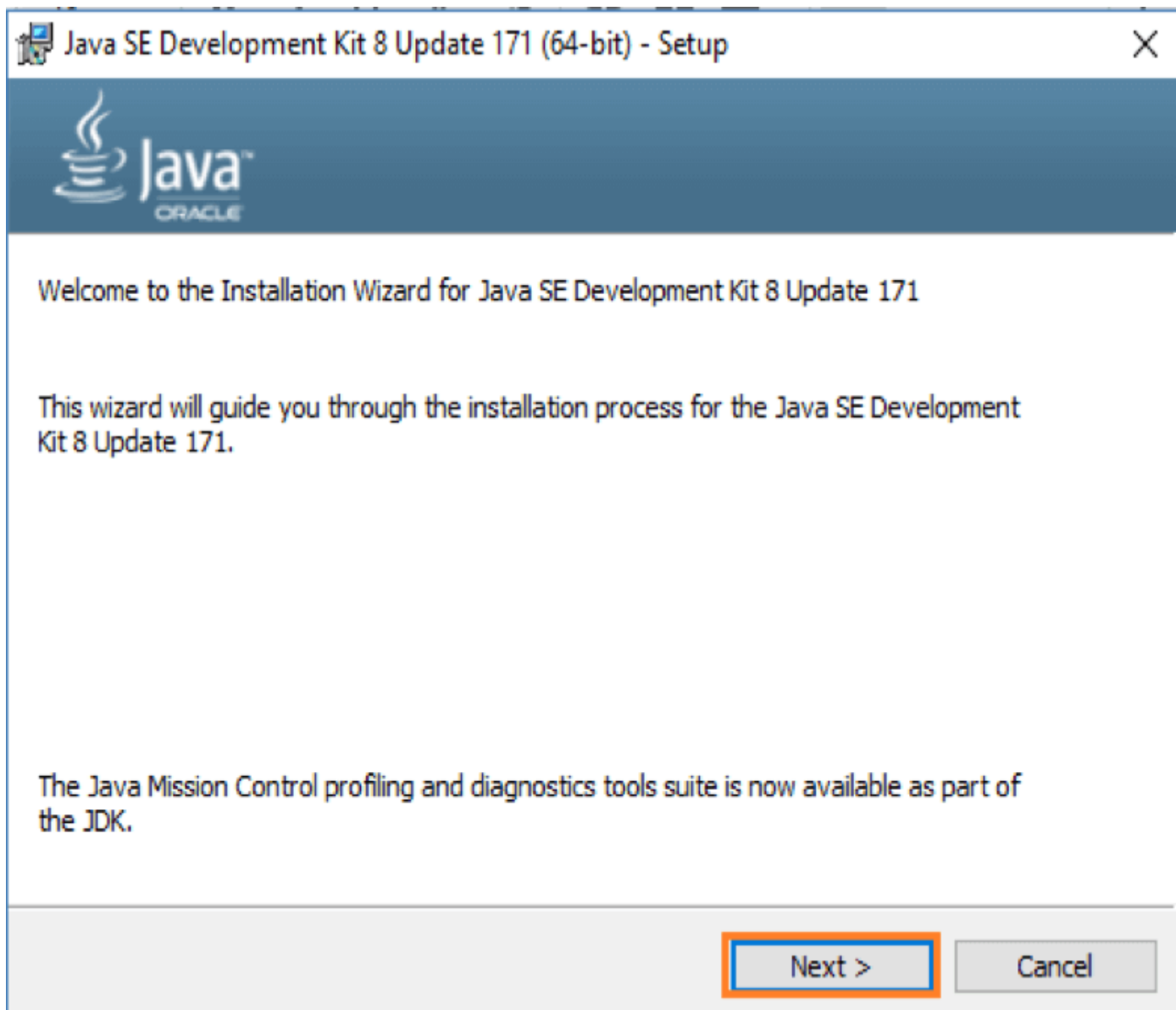
Java SE Development Kit 8u171

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

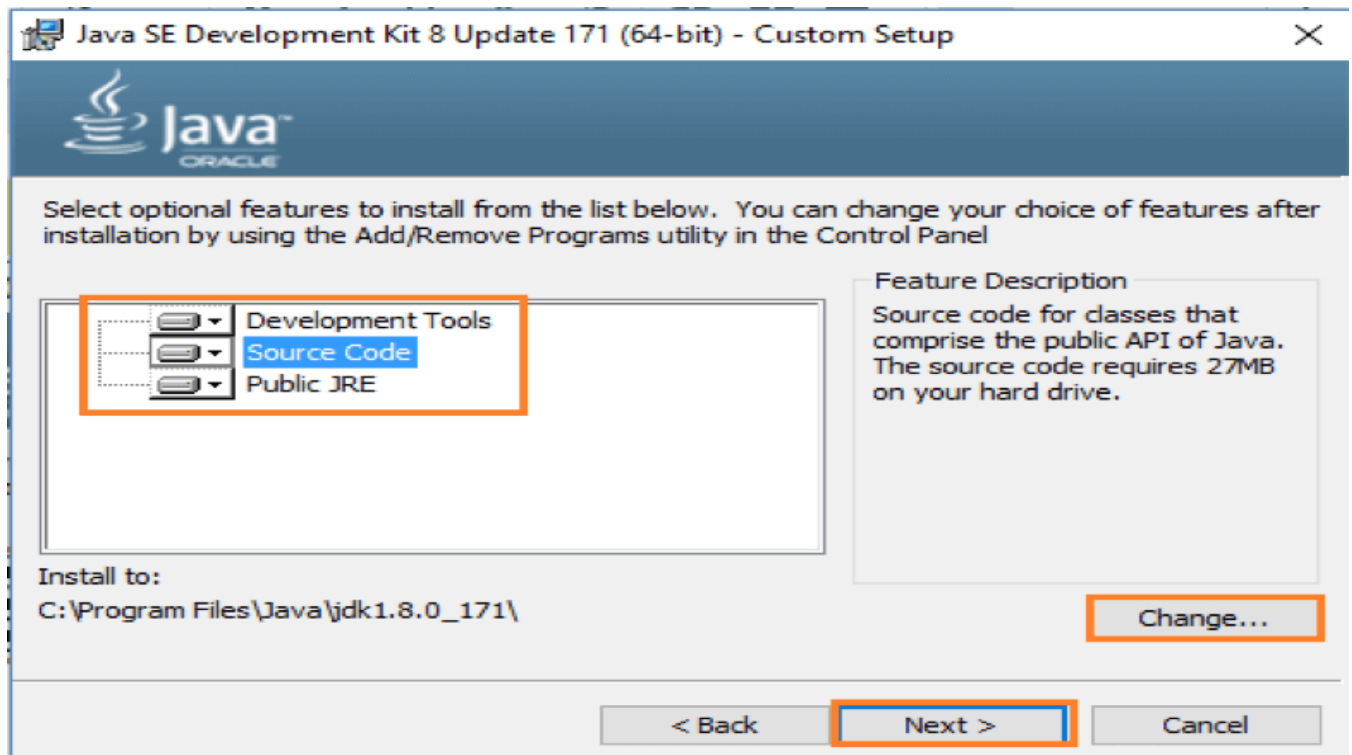
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	77.97 MB	jdk-8u171-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	74.89 MB	jdk-8u171-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.05 MB	jdk-8u171-linux-i586.rpm
Linux x86	184.88 MB	jdk-8u171-linux-i586.tar.gz
Linux x64	167.14 MB	jdk-8u171-linux-x64.rpm
Linux x64	182.05 MB	jdk-8u171-linux-x64.tar.gz
Mac OS X x64	247.84 MB	jdk-8u171-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	139.83 MB	jdk-8u171-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.19 MB	jdk-8u171-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	140.6 MB	jdk-8u171-solaris-x64.tar.Z
Solaris x64	97.05 MB	jdk-8u171-solaris-x64.tar.gz
Windows x86	199.1 MB	jdk-8u171-windows-i586.exe
Windows x64	207.27 MB	jdk-8u171-windows-x64.exe

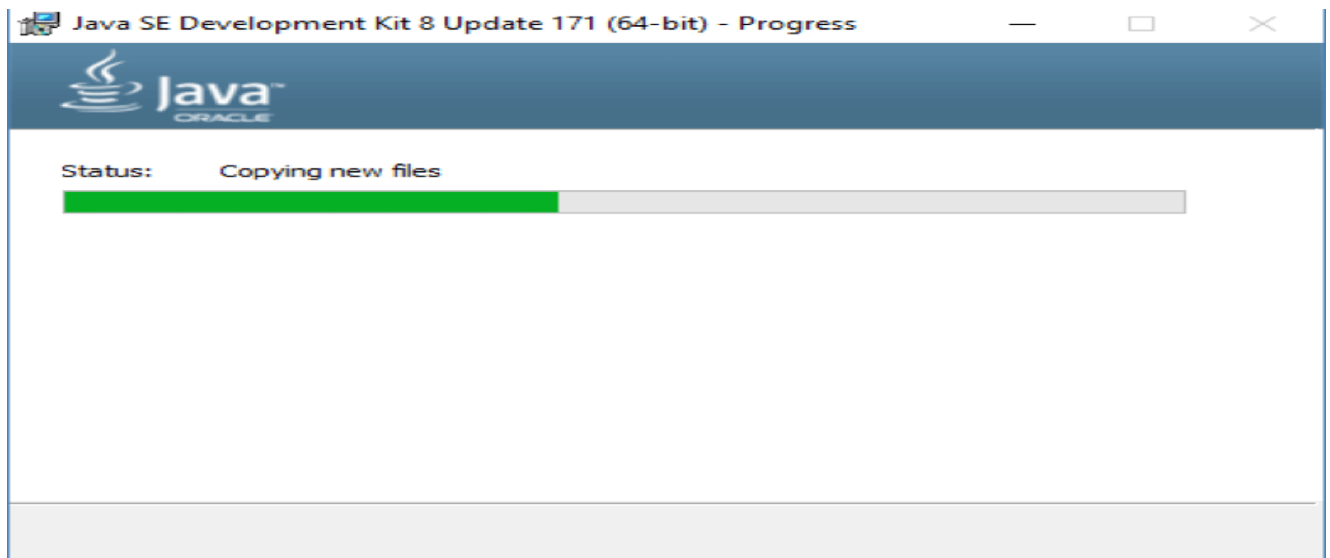
5. Run the downloaded exe file to launch the Installer wizard. Click **'Next'** to continue.



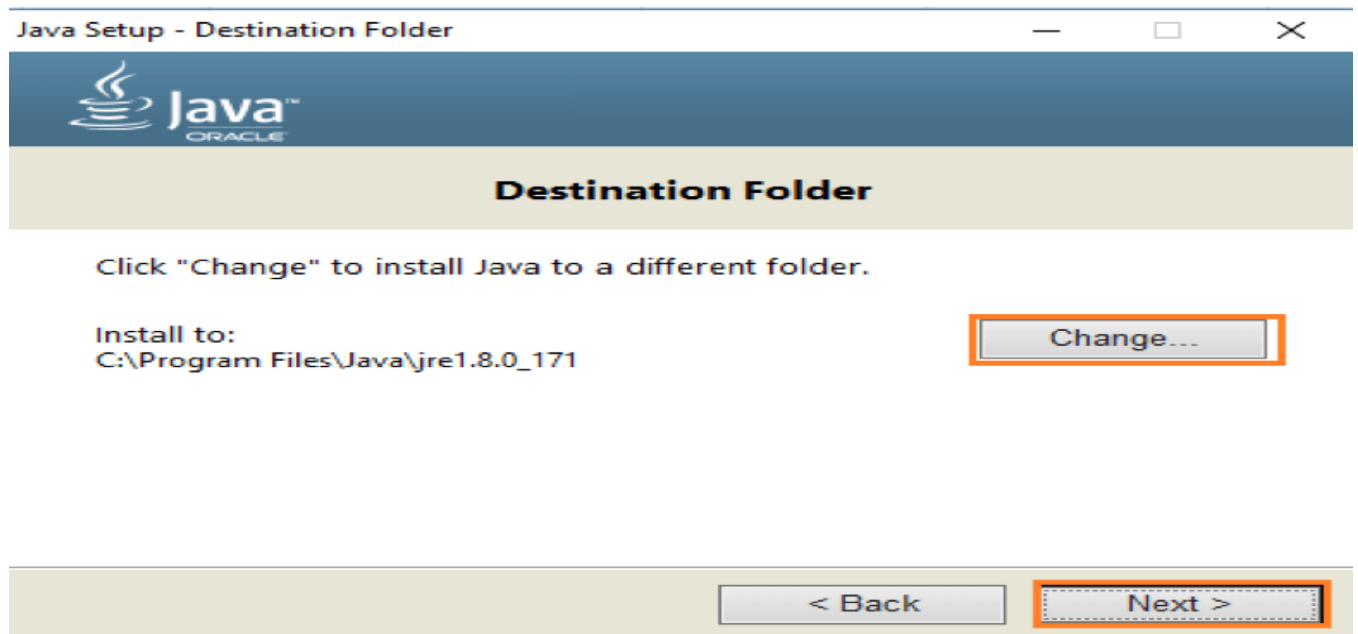
6. At this step you can select optional features to install, such as: install/don't install "Source Code", "Public JRE" or change the default installation folder of jdk8. But we will leave the default option that installs development tools+source code+public jre, and click '**Next**'.



The installer will be extracted and the progress will be shown in the wizard.



7. Change the default installation folder of jre8 if you want to, but we recommend leaving the default and click '**Next**'.



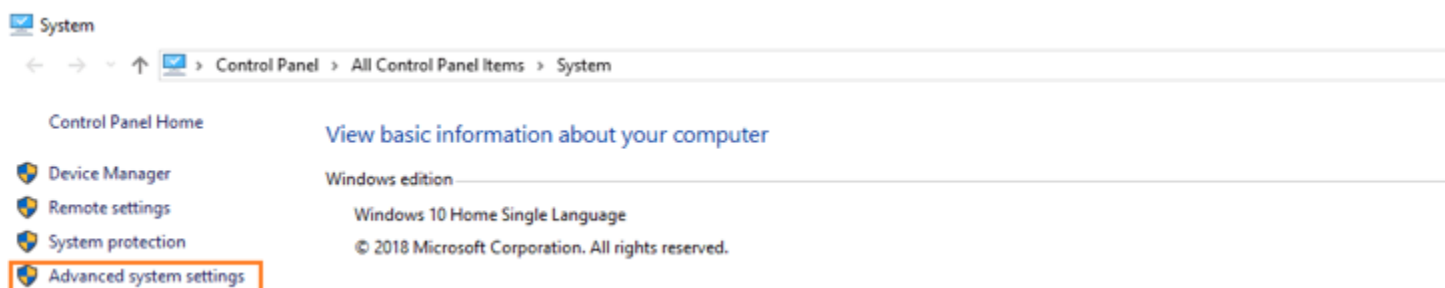
The installer will now install jre8 and the new files will be copied to the installation folder chosen above.



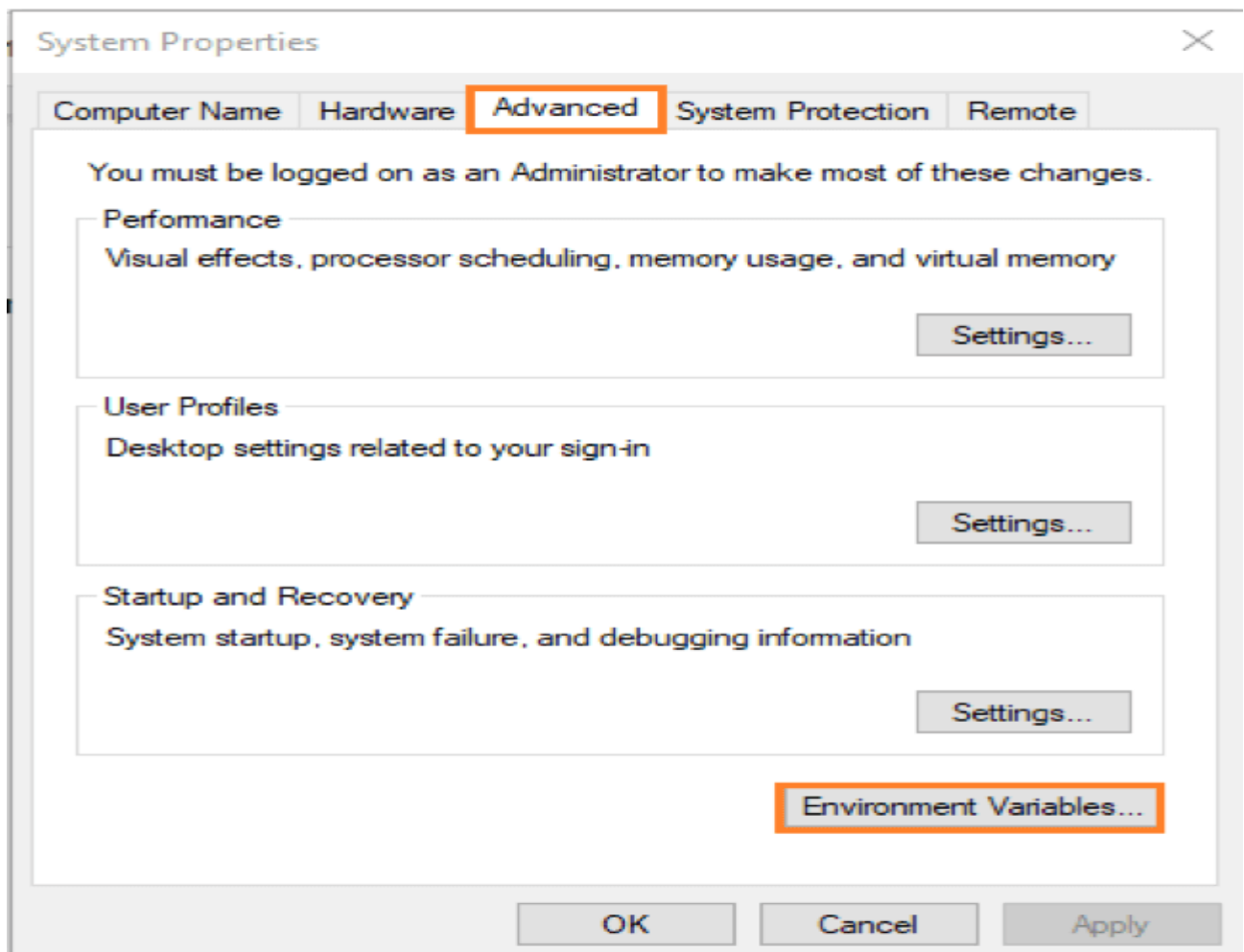
A successful installation looks like this:



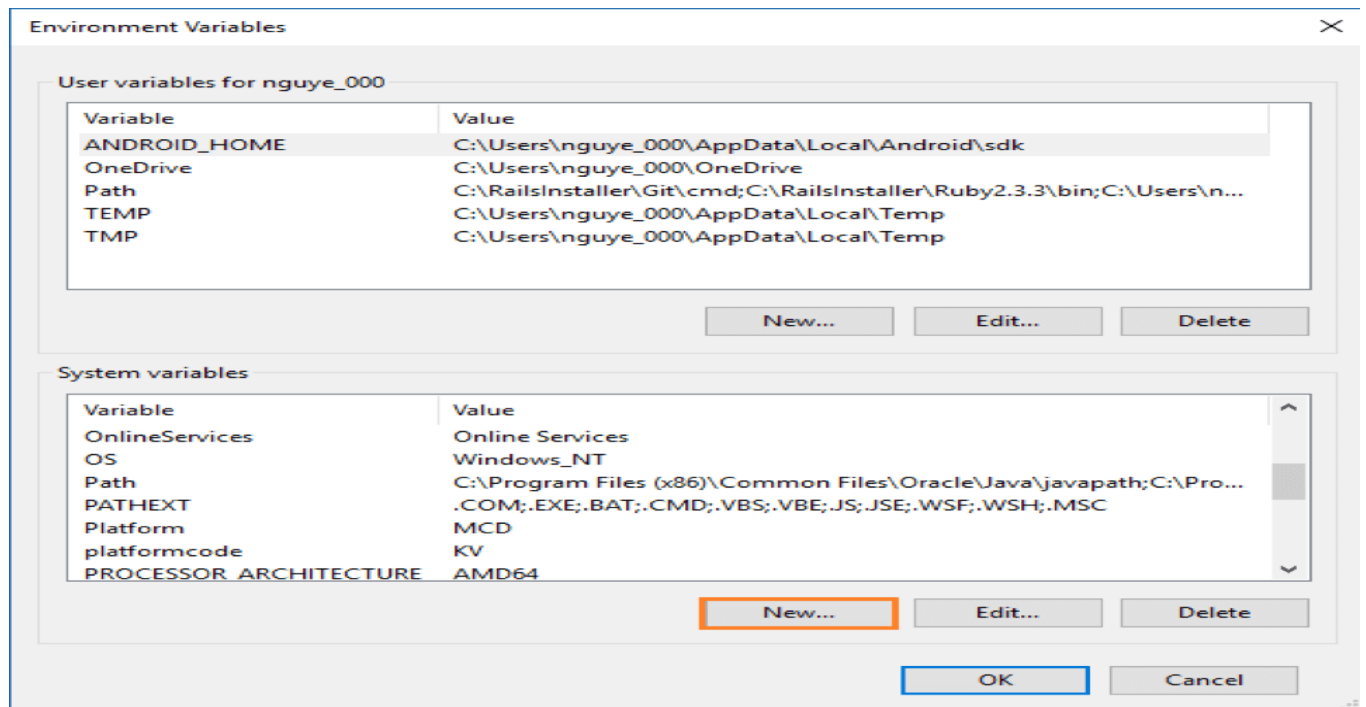
8. This step we will set the environment variable for java. Right-click on “**My Computer**” and select “**Properties**”. Click the “**Advanced system settings**” button like the screenshot below:



9. Under the 'Advanced' tab, choose the “**Environment Variables...**” button.

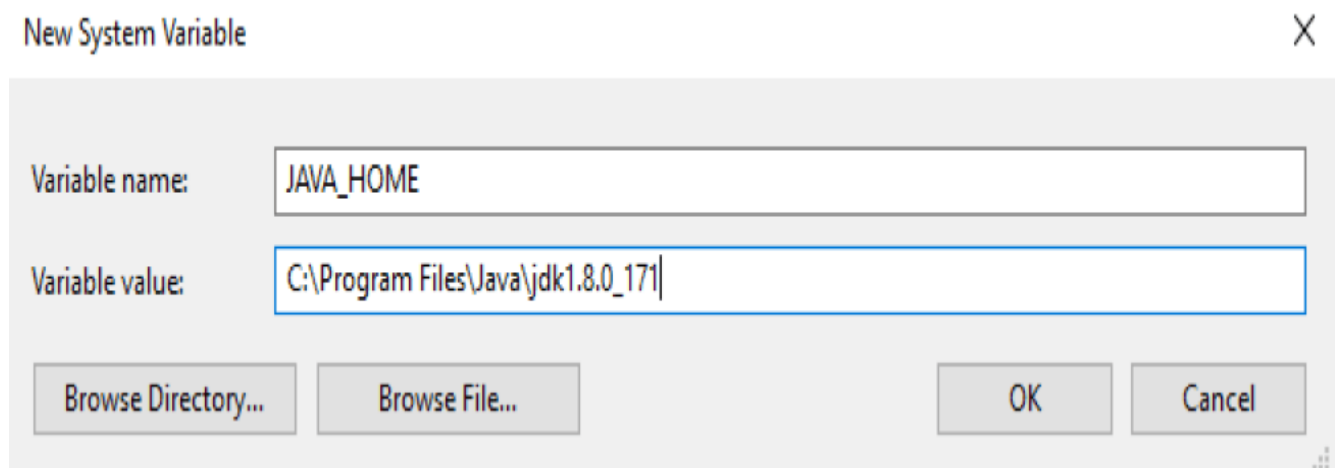


10. If we want the environment variable for java to be used only by the **current windows user**, go to the **User Variables for <user>** section. If we want the environment variable for java to be used by all **windows users**, go to the **System variables** section. In this example we will choose the **System variables** and click on the “**New...**” button, as shown below:

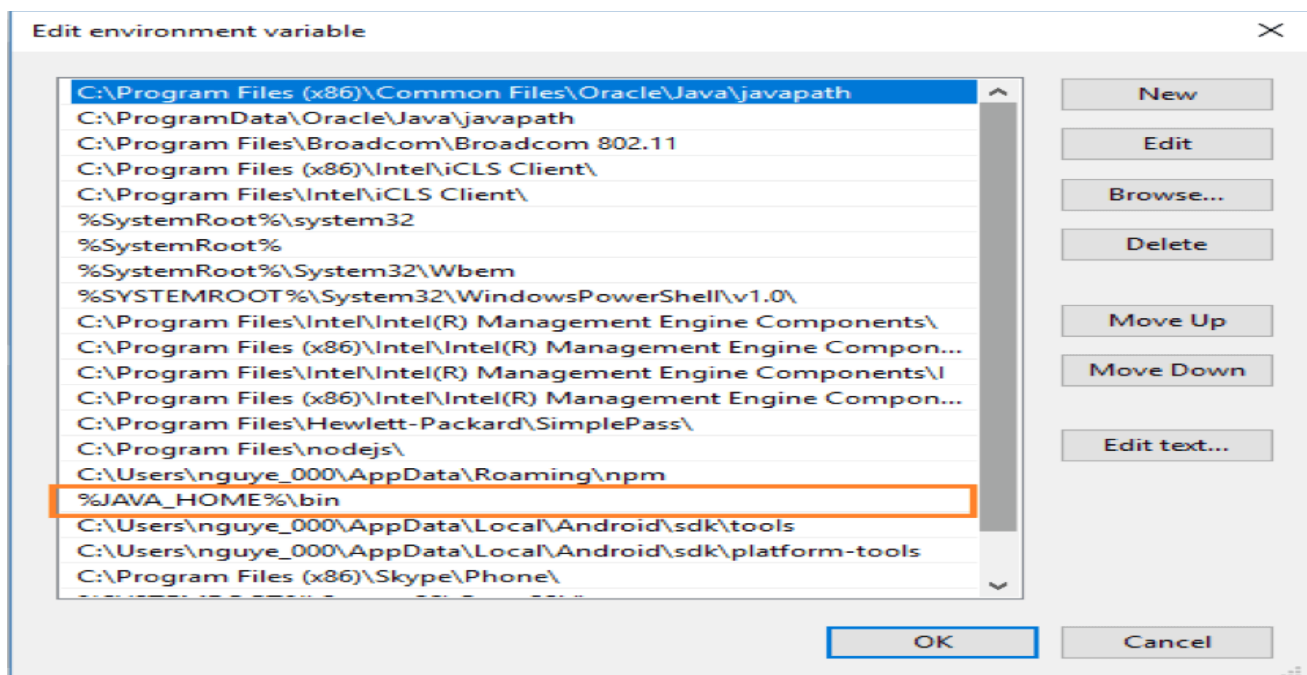


11. Type in “**Variable name**” and “**Variable value**” with the variable name as the name of the environment variable for java and the variable value as the path directory of jdk8.

- **Variable name:** JAVA_HOME
- **Variable value:** C:\Program Files\Java\jdk1.8.0_171



Then we have:



14. To verify the success of the installation, go to the command prompt and type 'java' as a command. The output of the command should look like below. If the Java installation is unsuccessful or if it had NOT been installed, it will throw an "unknown command" error.

```

Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\nguye_000>java
Usage: java [-options] class [args...]
           (to execute a class)
or java [-options] -jar jarfile [args...]
           (to execute a jar file)
where options include:
-d32          use a 32-bit data model if available
-d64          use a 64-bit data model if available
-server      to select the "server" VM
              The default VM is server.
-cp <class search path of directories and zip/jar files>
  -classpath <class search path of directories and zip/jar files>
              A ; separated list of directories, JAR archives,
              and ZIP archives to search for class files.
-D<name>=<value> set a system property
-verbose[:<class>[:<gc>[:<n1>]]
  enable verbose output
-version      print product version and exit
-version:<value>
  Warning: this feature is deprecated and will be removed
  in a future release.
  require the specified version to run
-showversion print product version and continue
-jre-restrict-search | -no-jre-restrict-search
  Warning: this feature is deprecated and will be removed
  in a future release.
  include/exclude user private JREs in the version search
-? -help     print this help message
-X           print help on non-standard options
-esa[:<packagename>...[:<classname>]]
  -enableassertions[:<packagename>...[:<classname>]]
  enable assertions with specified granularity
-da[:<packagename>...[:<classname>]]
  -disableassertions[:<packagename>...[:<classname>]]
  disable assertions with specified granularity
-ea | -enableSystemAssertions
  -disableSystemAssertions
  -dsa | -disableSystemAssertions
  disable system assertions
-agentlib:<libname>[:<options>]
  load native agent library <libname>, e.g. -agentlib:hprof
  see also, -agentlib:jdwp=help and -agentlib:hprof=help
-agentpath:<pathname>[:<options>]
  load native agent library by full pathname
-javaagent:<jarpath>[:<options>]
  load Java programming language agent, see java.lang.instrument
-splash:<imagepath>
  show splash screen with specified image
See http://www.oracle.com/technetwork/java/javase/documentation/index.html for more details.

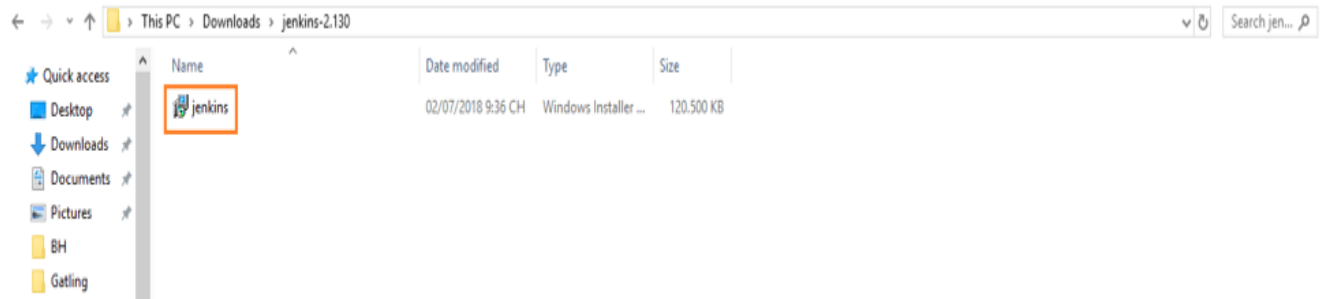
C:\Users\nguye_000>

```

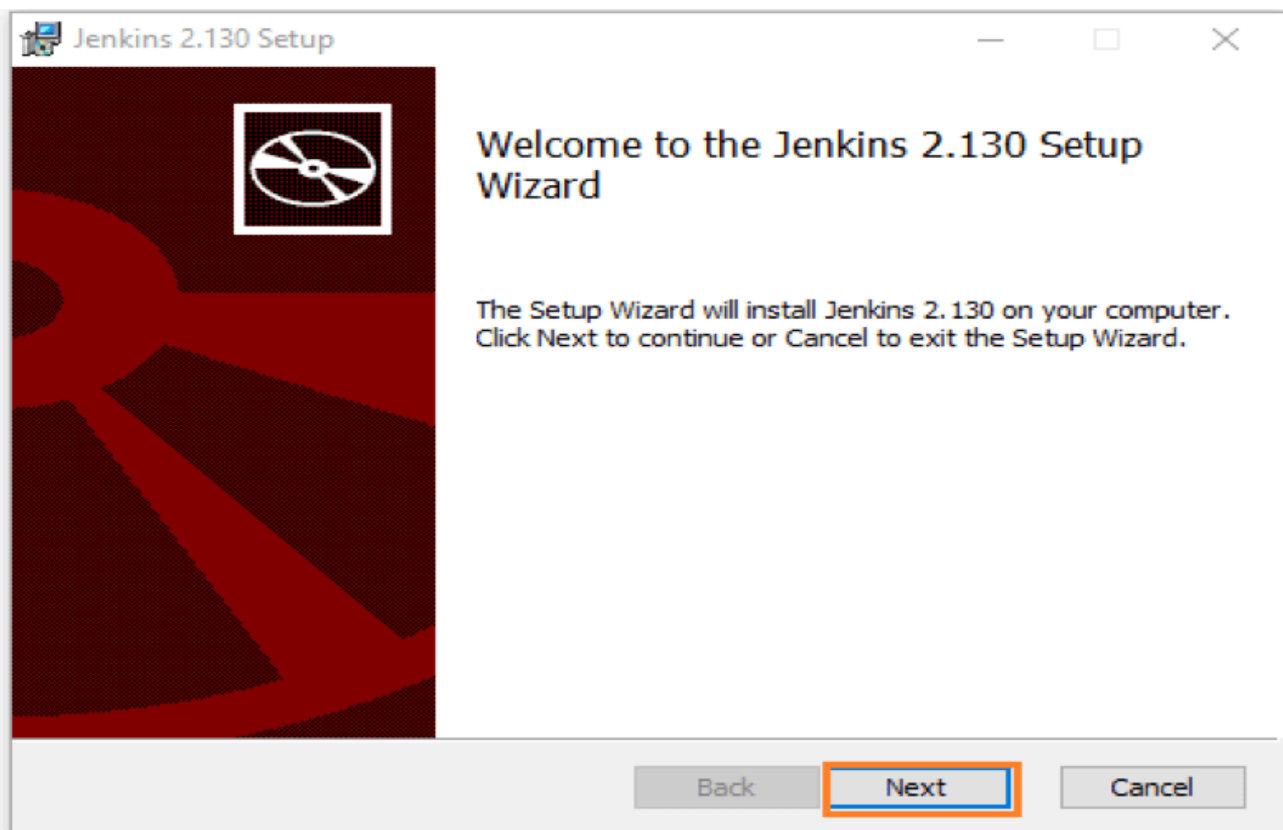
Step 2 Install Jenkins

First, you need to install JDK. Jenkins currently only supports JDK8. If you need help with installing Java [please look at the steps here](#). Once Java is running, you can install Jenkins.

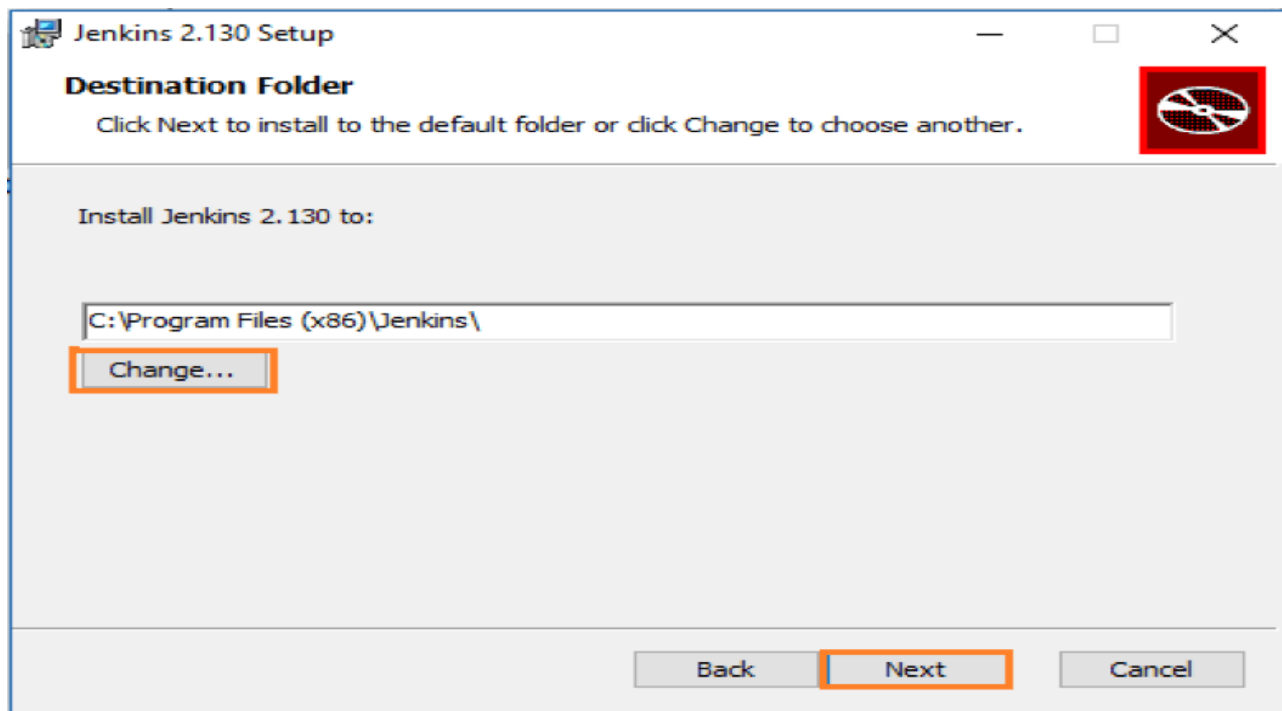
1. Click [here](#) to download the latest Jenkins package for Windows (currently it is version 2.130).
2. Unzip the file to a folder and click on the Jenkins **exe** file.



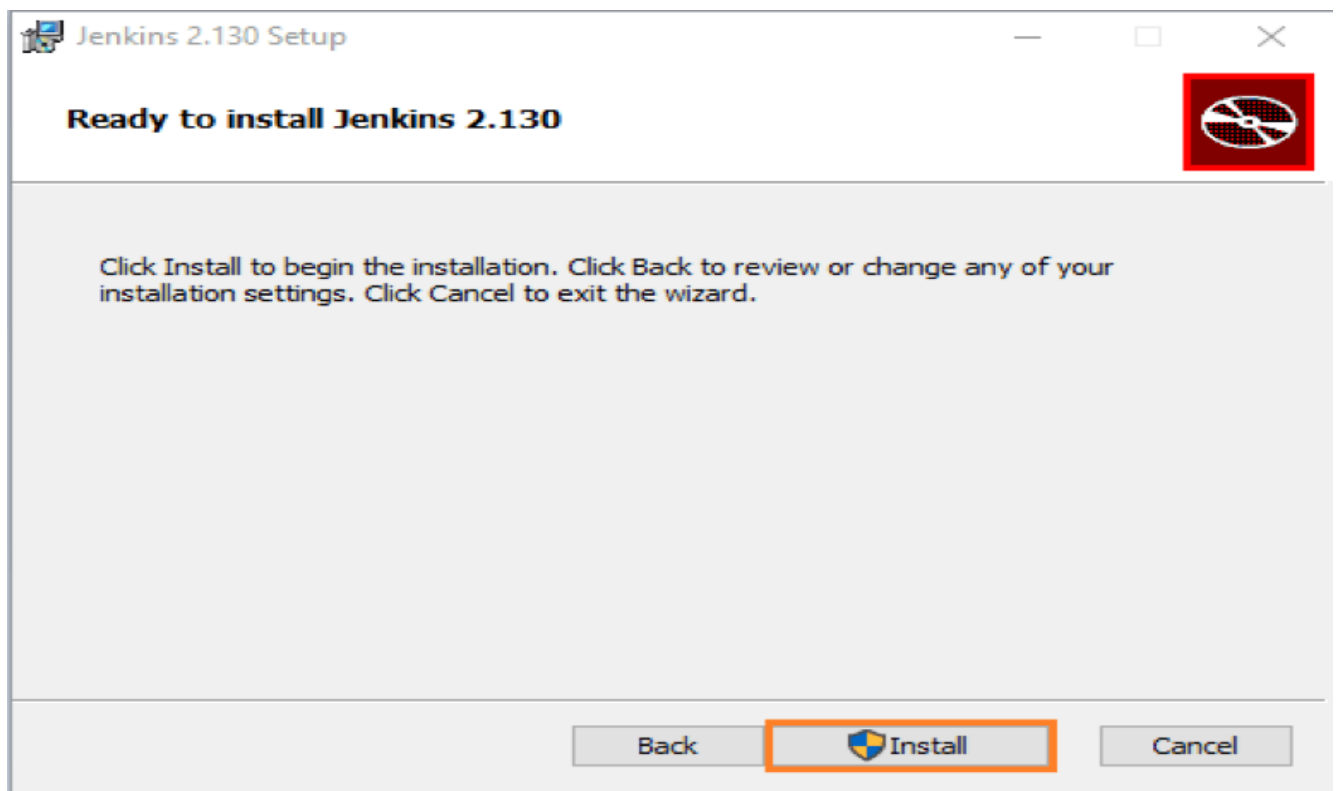
3. Click “**Next**” to start the installation.



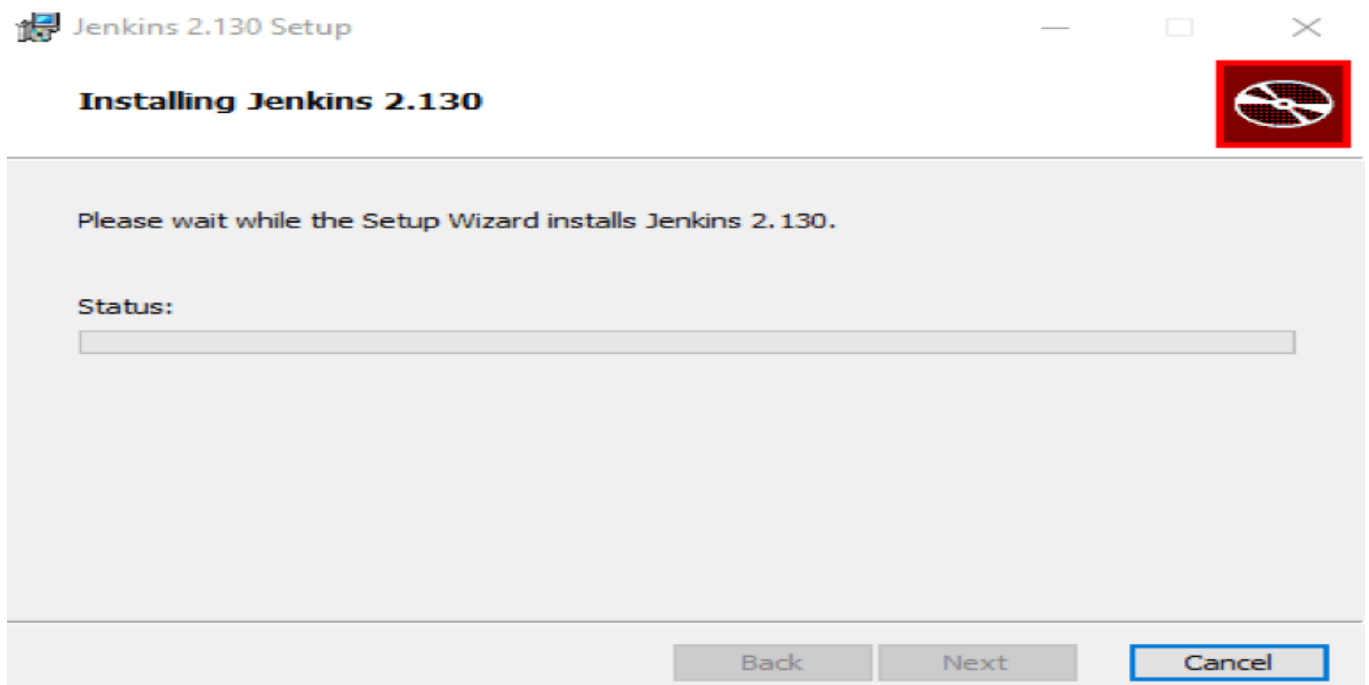
4. Click the “**Change...**” button if you want to install Jenkins in another folder. In this example I will keep the default option and click on the “**Next**” button.



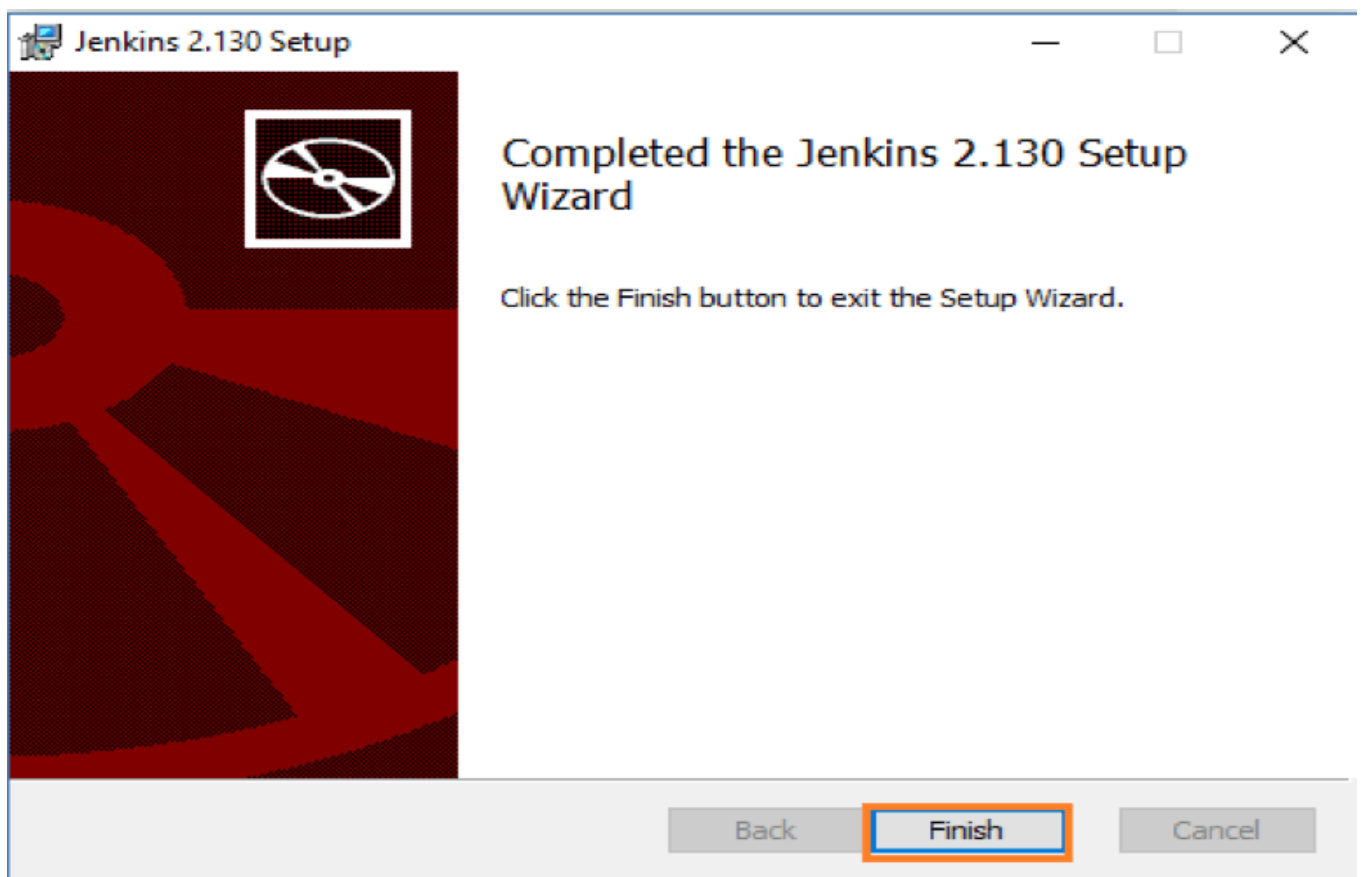
5. Click the “Install” button to start the installation process.



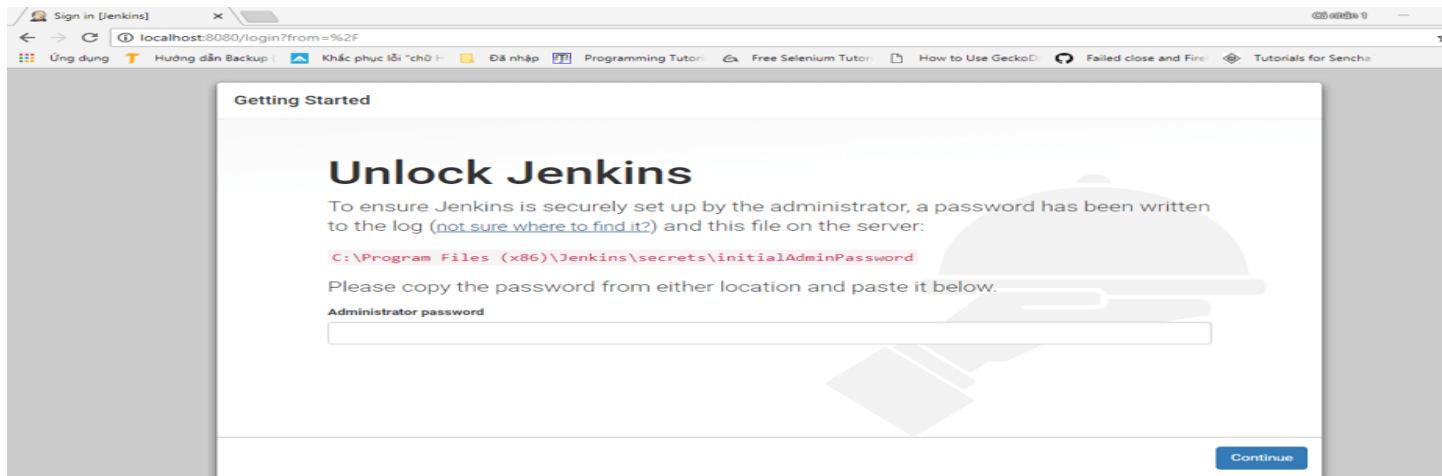
6. The installation is processing.



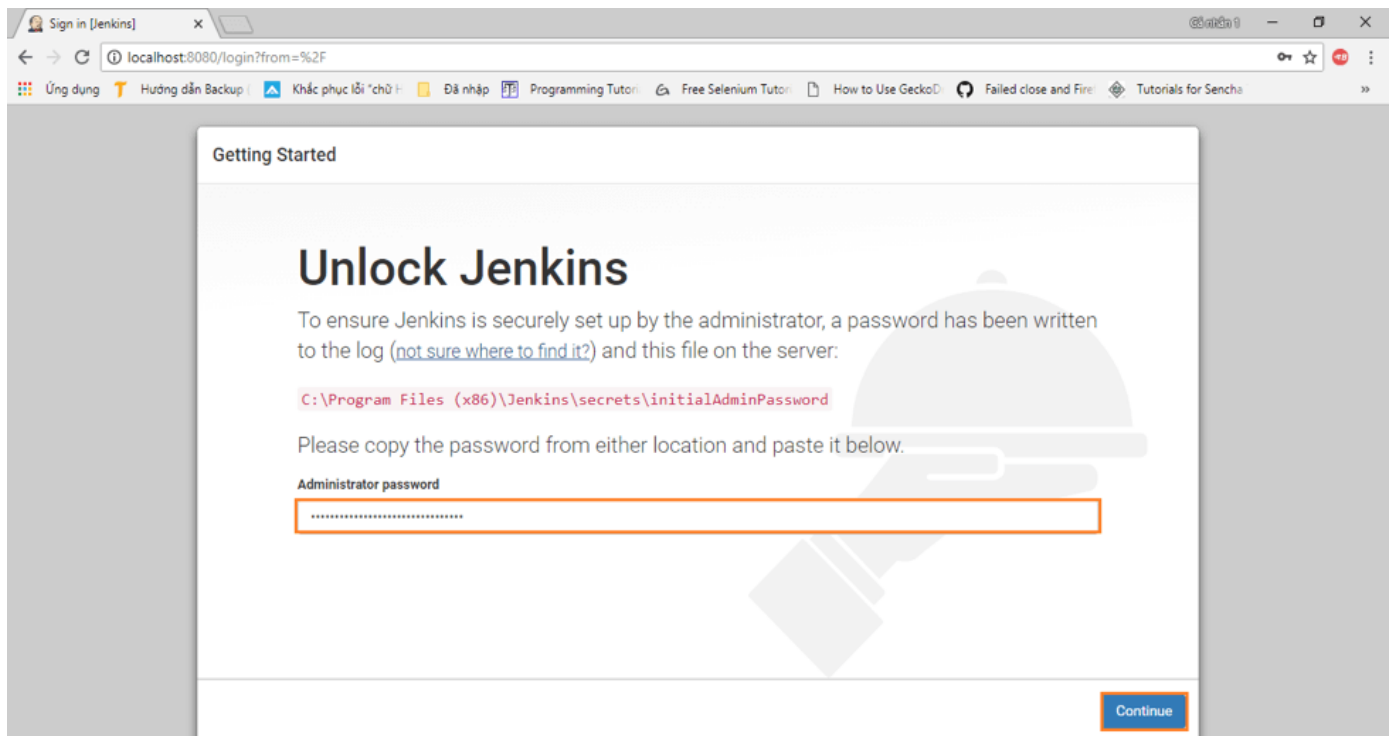
7. When done, click the “**Finish**” button to complete the installation process.



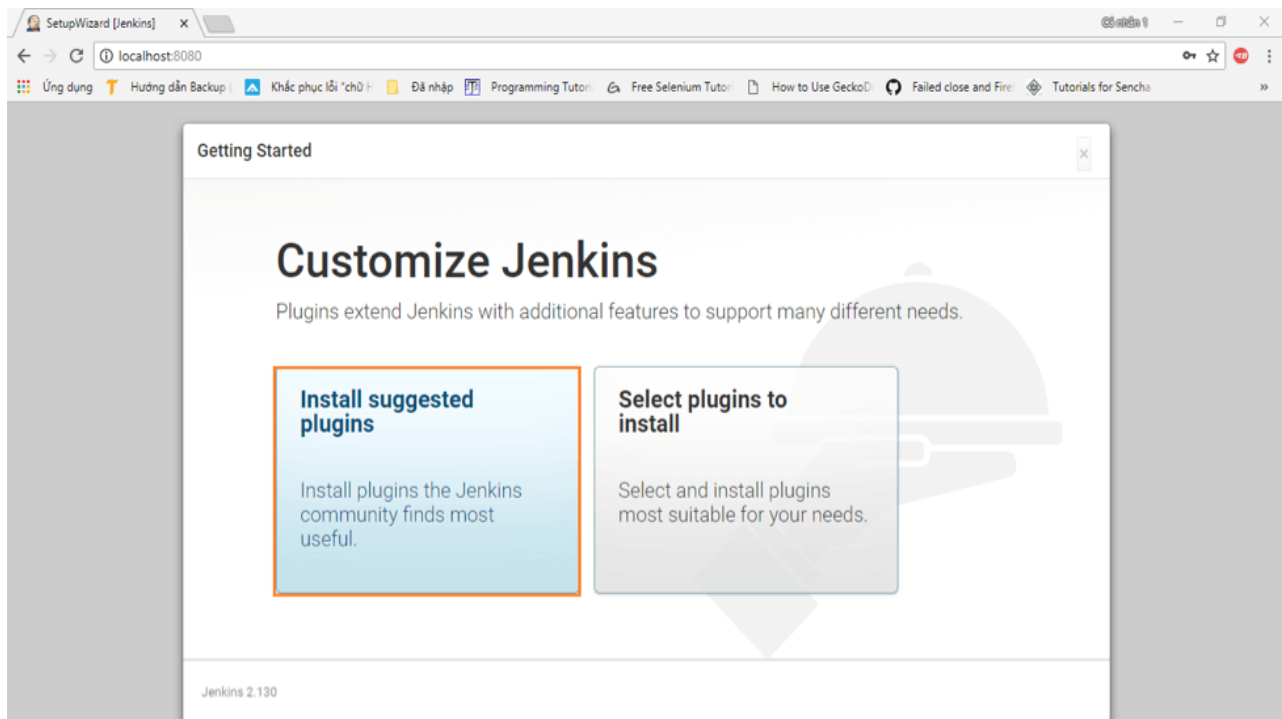
8. You will automatically be redirected to a local Jenkins page, or you can paste the URL <http://localhost:8080> in a browser.



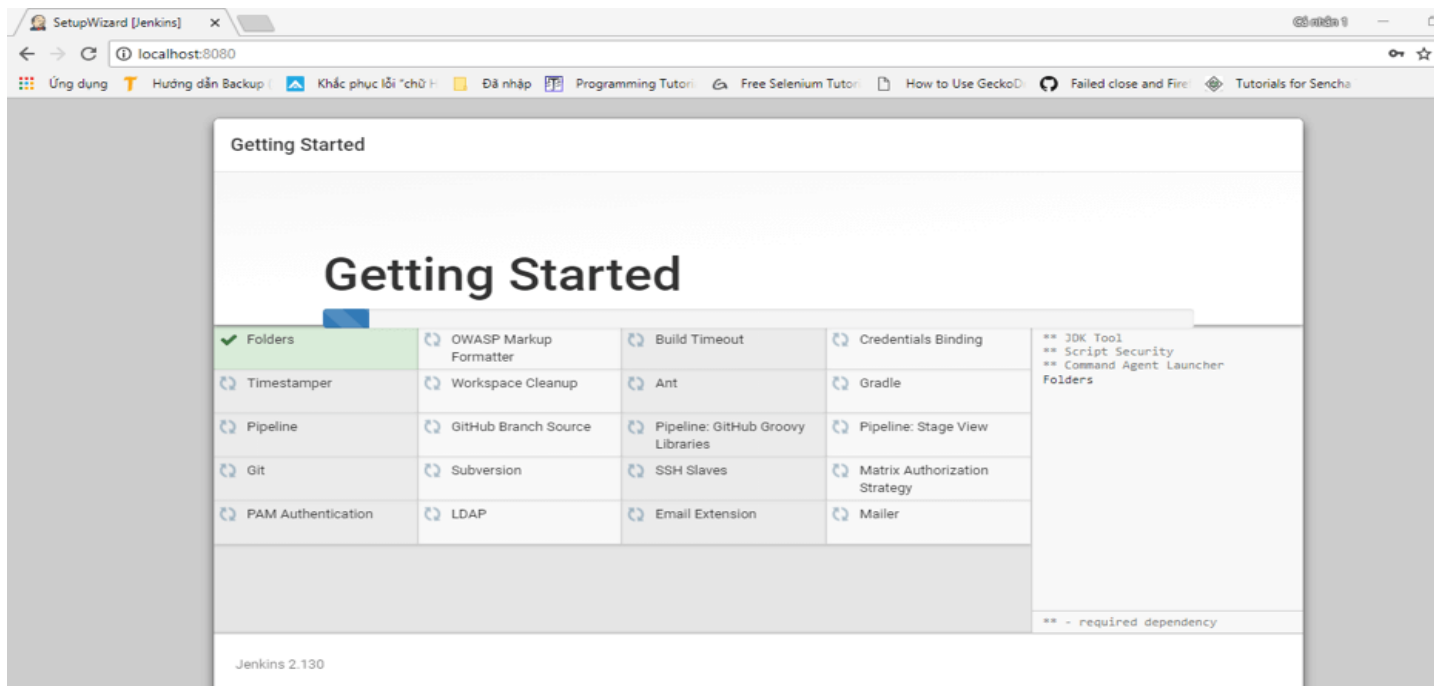
9. To unlock Jenkins, copy the password from the file at **C:\Program Files (x86)\Jenkins\secrets\initialAdminPassword** and paste it in the **Administrator password** field. Then, click the “**Continue**” button.



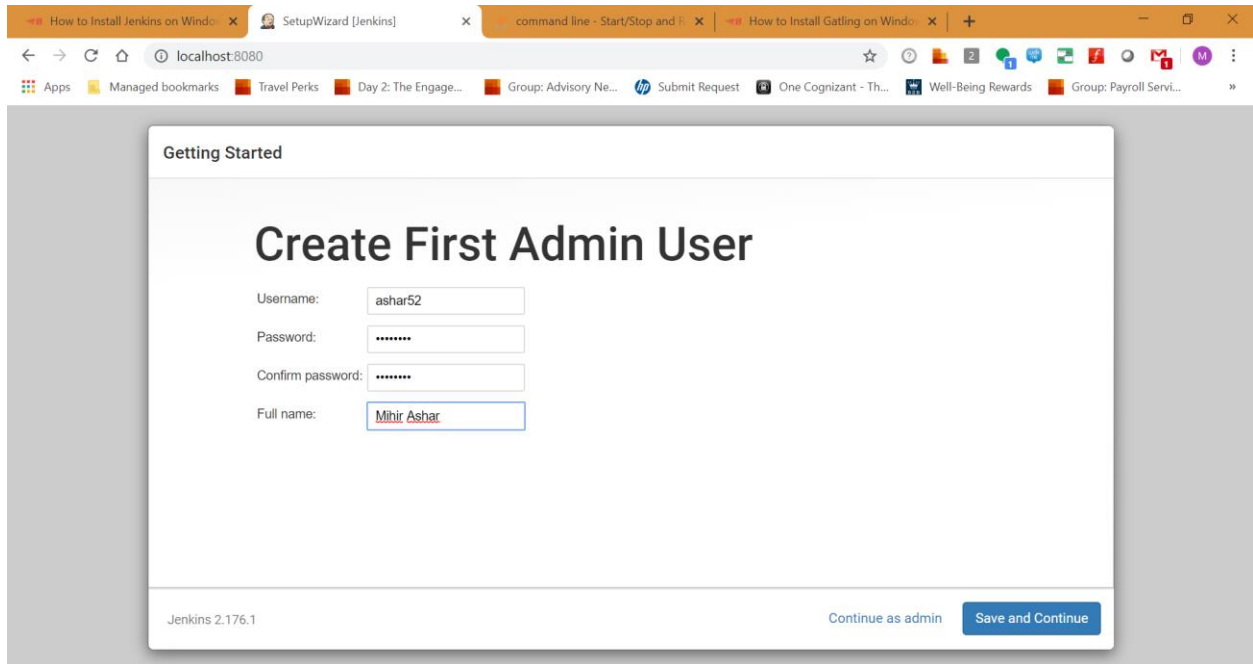
10. You can install either the suggested plugins or selected plugins you choose. To keep it simple, we will install the suggested plugins.



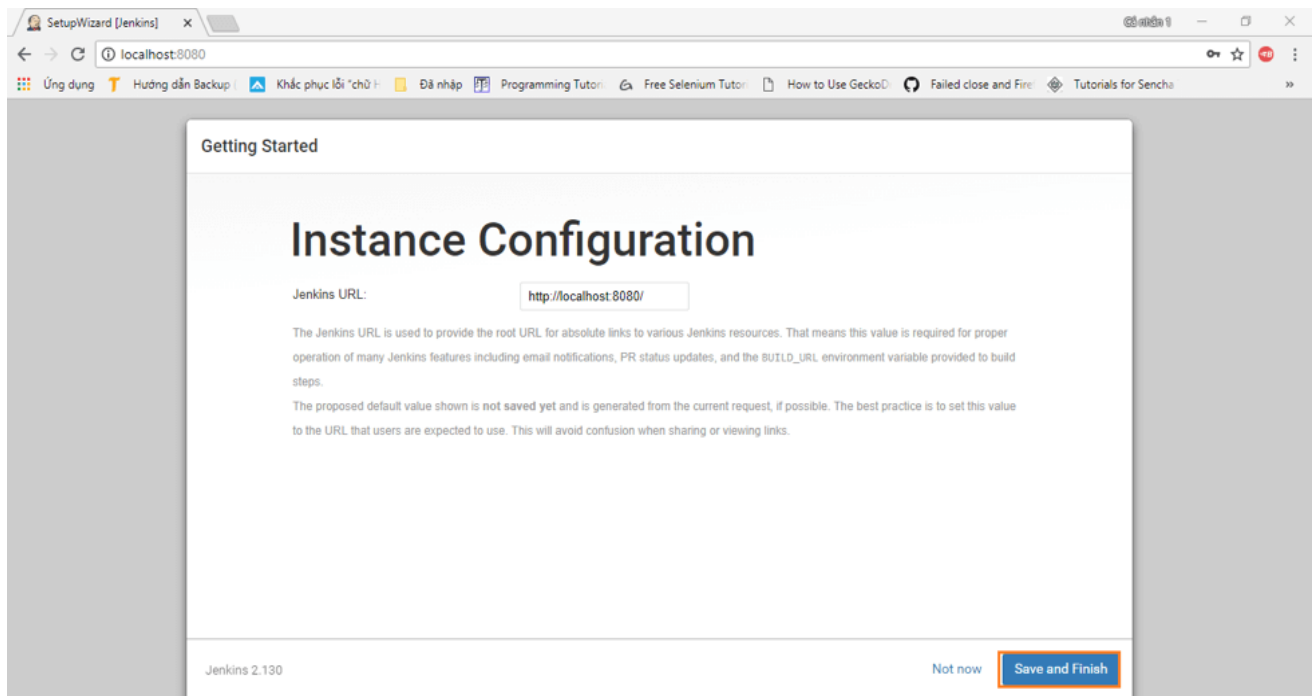
11. Wait until the plugins are completely installed.



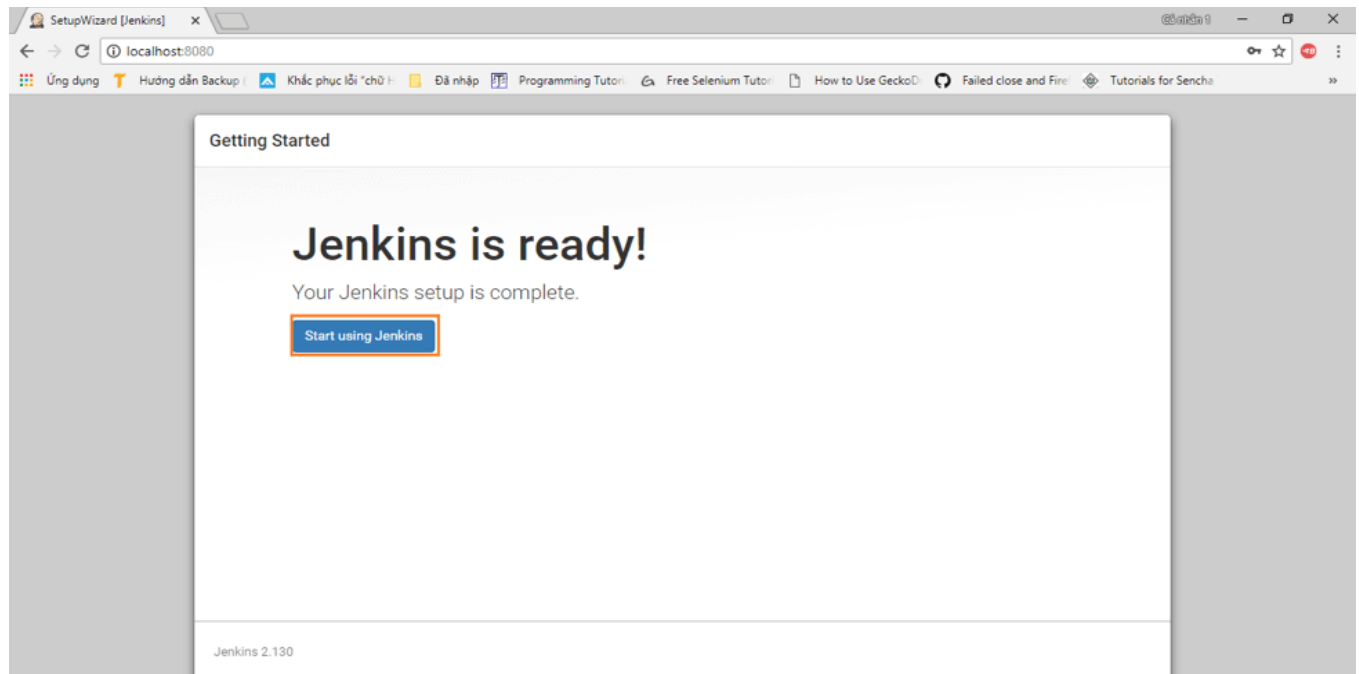
12. The next thing we should do is create an admin user for Jenkins. Put in your details and click **“Save and Continue”**.



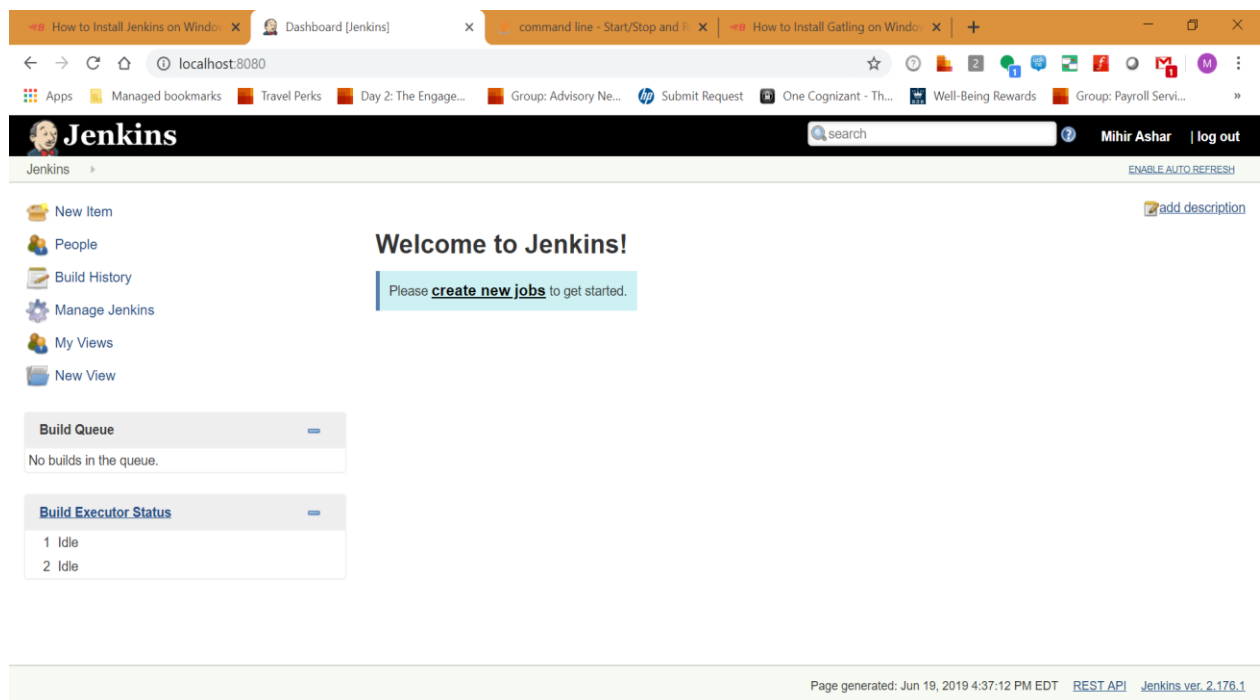
13. Click **“Save and Finish”** to complete the Jenkins installation.



14. Now, click **“Start using Jenkins”** to start Jenkins.



15. Finally, here is the default Jenkins page.



You can now start creating your continuous integration pipeline!

Below tutorial links will guide the developers and administrators for step-by-step walkthrough for the below activities:

- Install and configure Git Repo and Github:
 - https://kbroman.org/github_tutorial/pages/init.html (Creating a new Git repo from scratch)
 - https://www.youtube.com/watch?v=J_Clau1bYco (Install Github on Windows)

- <http://www.codebind.com/linux-tutorials/basic-git-commands-list/> (Popular Git commands)
- Integrate Github with Jenkins:
 - <https://www.youtube.com/watch?v=Z3S2gMBUkBo>
- Connect Mulesoft Anypoint Studio to Github Repo via Git Integration Plugin
 - <https://www.youtube.com/watch?v=9-lnHz5-cVQ>
 - <https://support.mulesoft.com/s/article/How-to-configure-egit-in-Studio>
- Deploy Mule Project on CloudHub via JenkinsFile
 - <https://blogs.mulesoft.com/dev/mule-dev/how-to-deploy-mule-application-maven-jenkins-pipeline/>
 - <https://javastreet.com/blog/mule-maven-deployment-and-jenkins-pipeline.html>

Automated CI/CD Flow (Mulesoft CloudHub, Github and Jenkins)

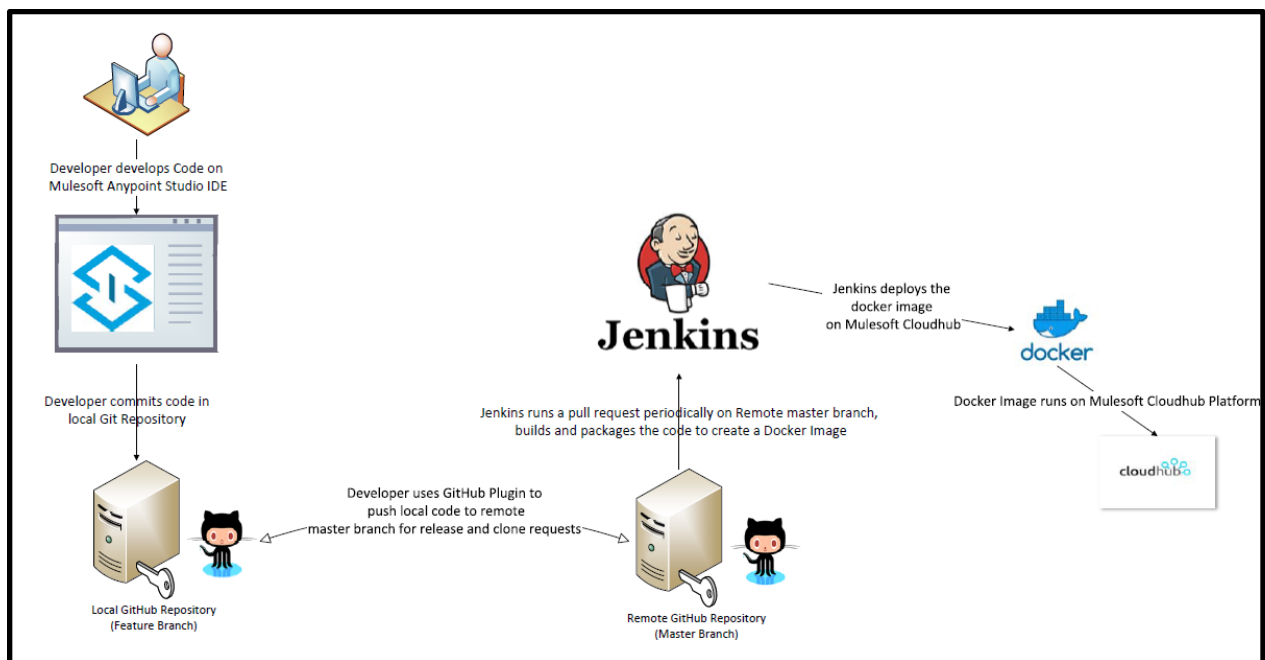


Diagram: CI/CD Flow for deploying Mule Code on CloudHub through Jenkins and Github
Following plugins can be installed in Jenkins for integration of following technologies:

Plugins	Description
Salesforce	
OSF Builder Suite For Salesforce Commerce Cloud :: Deploy	Deploy your build to a Salesforce CommerceCloud instance
Quality Clouds Scan	Run Quality Clouds Scan against your ServiceNow and Salesforce instances
CodeScan	Jenkins plugins to run hosted CodeScan for Salesforce builds
Github	
Github	Integrates Github to Jenkins
Github Integration	Github Integration Plugin for Jenkins
Github Authentication	Authentication plugin using Github OAuth to provide authentication and authorization capabilities for Github and Github Enterprise
Code Quality and Unit test coverage	
SonarQube Scanner	Easy integration of SonarQube, the open source platform for Continuous Integration of code quality
Cobertura	Integrates Cobertura coverage reports to Jenkins for unit test coverage
Maven	
Maven Integration	Provides deep integration of Maven with Jenkins: Automatic triggers between projects depending on Snapshots, automated configurations of various Jenkins publishers (JUnit etc.)
Continuous Integration game	The continuous integration build game
Maven Release	Enables you to perform releases using the Maven release-plugin from Jenkins
JIRA	

JIRA	Integrates Jenkins to Atlassian JIRA
JIRA Pipeline steps	JIRA Pipeline steps
JIRA Trigger	JIRA Trigger
Docker and Kubernetes	
Docker	This plugin integrates Jenkins with Docker
Docker Swarm	Docker swarm cloud plugin
Docker Pipeline	Build and use Docker containers from pipelines
Docker Compose Build step	Docker Compose plugin for Jenkins
Kubernetes :: Pipelines :: Kubernetes Steps	Build and uses Docker images and Kubernetes Pods
Docker Slaves	Uses Docker containers to run Jenkins build agents
Kubernetes	Integrates Jenkins with Kubernetes
Kubernetes Credentials	Common classes for Kubernetes credentials
AWS	
AWS CloudWatch Logs Publisher	Publishes builds logs to Amazon CloudWatch Logs
AWS CodePipeline	AWS CodePipeline integration
AWS CodeBuild	Build your project to AWS CodeBuild
AWS Lambda	Allows to upload a zip file or folder to AWS Lambda
Pipeline: AWS steps	Adds Jenkins pipeline steps to AWS API
AWS Global configuration	Jenkins plugin to configure AWS related settings
Jenkins-cloudformation-plugin	Adds a build wrapper that can spawn an AWS Cloud formation recipe at the start of the build

	and take it down at the end.
Miscellaneous	
HTTP Request	Sends HTTP requests to an url with some parameters
Batch task	Adds the 'task' action to the project for performing batch tasks on the server workspace
Artifactory	Allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them.
Publish over FTP	Send build artifacts over FTP
FTP Rename	For renaming a file in FTP server
FTP Publisher	Used to upload project artifacts and whole directories to an FTP server
Email Extension	Replacement for Jenkin's email publisher allowing to configure every aspect of email notifications: when an email sent, who should receive it and what email says.
LDAP Email	Resolves user email addresses from an LDAP directory. Not needed is Jenkins uses LDAP as its authentication and user records have standard mail attributes.
Performance	Allows tracking performance KPIs, based on results read from popular testing tools (Jmeter, JUnit, Taurus)
ATDD	
Selenium	Integrate Selenium with Jenkins
Selenium Build	Jenkins plugin which invokes scripts using the Selenium Builder Interpreter
Selenium HTML reports	Visualize the results of Selenium tests
Cucumber	Run Cucumber tests under Jenkins CI

Cucumber-perf	Collects and displays information on the performance of Cucumber tests, particularly the duration of each step
Cucumber reports	Provides pretty html reports for Cucumber. Can be used anywhere json report is generated.