

Experiment No : 04

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

Theory :

Jenkins is an open-source automation server that automates the repetitive technical tasks involved in the continuous integration and delivery of software. Jenkins is Java-based and can be installed from Ubuntu packages or by downloading and running its web application archive (WAR) file — a collection of files that make up a complete web application to run on a server.

You need to consider continuous integration (CI) and continuous delivery (CD) to understand Jenkins:

- **Continuous integration** – the practice of continuous production combined with the main industry.
- **Continuous delivery** – the code is constantly delivered to an area after the code is ready for delivery. It could be for production or staging. The commodity is supplied to a consumer base that can provide QA or inspection by customers.

Developers update the code regularly in the shared repository (such as GitHub or TFS). Improvements made in the source code are made at the end of the day, making it difficult to identify the errors. So, Jenkins is used here.

Once a developer changes the repository, Jenkins will automatically enable the build and immediately warn you in the event of an error (Continuous Integration CI).

Jenkins is loved by teams of all sizes, for different language projects like Java, Ruby, Dot Net, PHP etc. Jenkins is a platform that is autonomous, and can be used on Windows, Linux or any other operating system.

Advantages of Jenkins include:

- It is an open-source tool with great community support.
- It is easy to install.
- It has 1000+ plugins to ease your work. If a plugin does not exist, you can code it and share it with the community.
- It is free of cost.
- It is built with Java and hence, it is portable to all the major platforms.

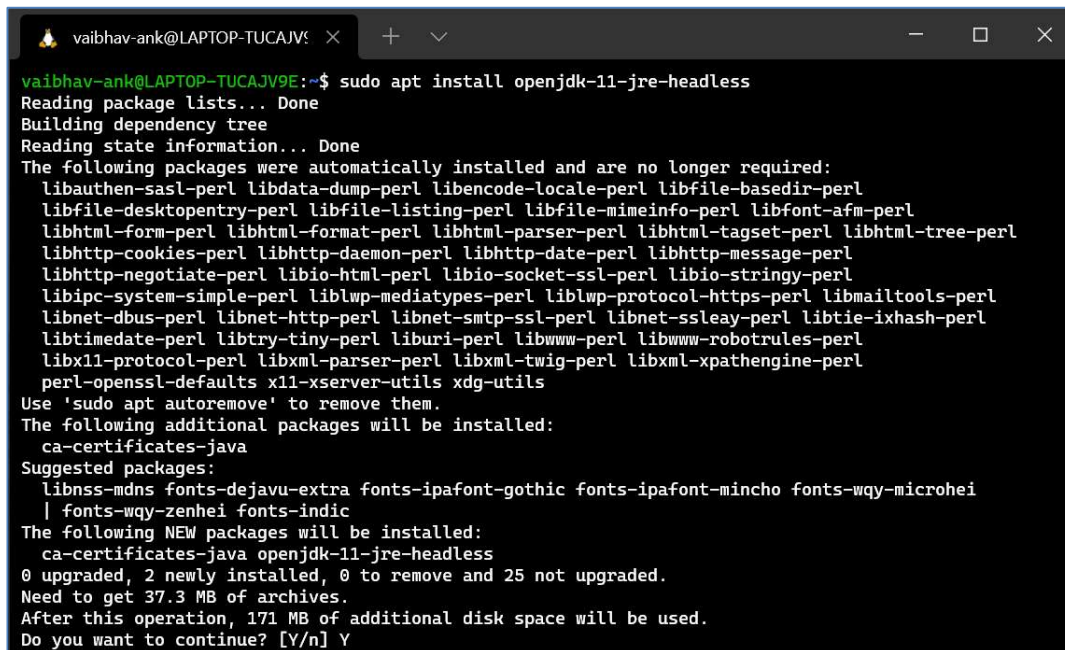
PART 1 : Install Java

To install Java run any one command from below :

```
$ sudo apt install openjdk-8-jdk
```

OR

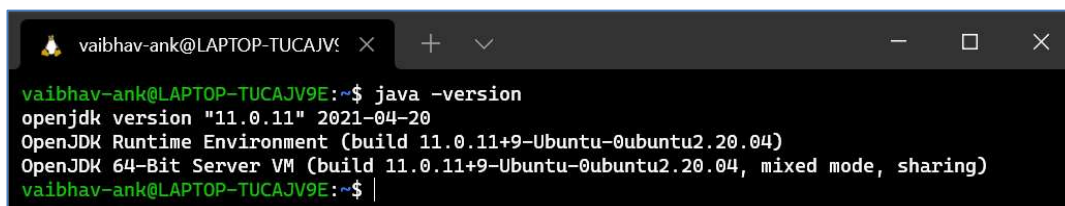
```
$ sudo apt install openjdk-11-jre-headless
```

A terminal window titled 'vaibhav-ank@LAPTOP-TUCAJV:' shows the command 'sudo apt install openjdk-11-jre-headless' being executed. The output indicates that several packages were automatically installed and are no longer required, including libauthen-sasl-perl, libdata-dump-perl, libencode-locale-perl, libfile-basedir-perl, libfile-desktopentry-perl, libfile-listing-perl, libfile-mimeinfo-perl, libfont-afm-perl, libhtml-form-perl, libhtml-format-perl, libhtml-parser-perl, libhtml-tagset-perl, libhtml-tree-perl, libhttp-cookies-perl, libhttp-daemon-perl, libhttp-date-perl, libhttp-message-perl, libhttp-negotiate-perl, libio-html-perl, libio-socket-ssl-perl, libio-stringy-perl, libipc-system-simple-perl, liblwp-mediatypes-perl, liblwp-protocol-https-perl, libmailtools-perl, libnet-dbus-perl, libnet-http-perl, libnet-smtp-ssl-perl, libnet-ssleay-perl, libtie-ixhash-perl, libtimedate-perl, libtry-tiny-perl, liburi-perl, libwww-perl, libwww-robotrules-perl, libxml-parser-perl, libxml-twig-perl, libxml-xpathengine-perl, perl-openssl-defaults, x11-xserver-utils, and xdg-utils. It also suggests installing ca-certificates-java, libnss-mdns, fonts-dejavu-extra, fonts-ipafont-gothic, fonts-ipafont-mincho, fonts-wqy-microhei, and fonts-wqy-zenhei. The final output shows that 0 packages were upgraded, 2 were newly installed, 0 were to be removed, and 25 were not upgraded. The total size of the archives is 37.3 MB, and 171 MB of additional disk space will be used. The user is prompted to continue, and they respond with 'Y'.

```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ sudo apt install openjdk-11-jre-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libauthen-sasl-perl libdata-dump-perl libencode-locale-perl libfile-basedir-perl
  libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl libfont-afm-perl
  libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl
  libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl
  libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl libio-stringy-perl
  libipc-system-simple-perl liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl
  libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libtie-ixhash-perl
  libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl
  libxml-parser-perl libxml-twig-perl libxml-xpathengine-perl
  perl-openssl-defaults x11-xserver-utils xdg-utils
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  ca-certificates-java
Suggested packages:
  libnss-mdns fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei fonts-indic
The following NEW packages will be installed:
  ca-certificates-java openjdk-11-jre-headless
0 upgraded, 2 newly installed, 0 to remove and 25 not upgraded.
Need to get 37.3 MB of archives.
After this operation, 171 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Check if java is installed using following command.

```
$ java -version
```

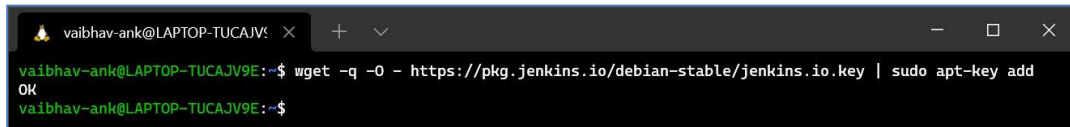
A terminal window titled 'vaibhav-ank@LAPTOP-TUCAJV:' shows the command 'java -version' being executed. The output displays the OpenJDK version '11.0.11' (2021-04-20), the OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04), and the OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing).

```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ java -version
openjdk version "11.0.11" 2021-04-20
OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
OpenJDK 64-Bit Server VM (build 11.0.11+9-Ubuntu-0ubuntu2.20.04, mixed mode, sharing)
vaibhav-ank@LAPTOP-TUCAJV9E:~$
```

PART 2 : Install Jenkins

- 1) First, add the repository key to the system:

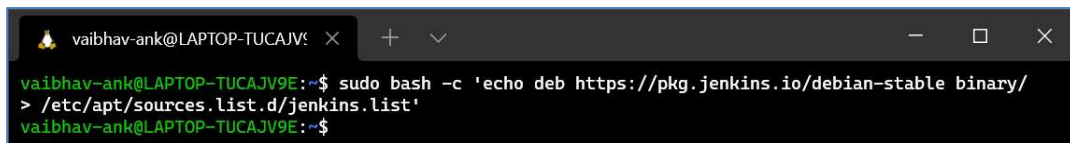
```
$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |  
sudo apt-key add
```



```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add  
OK  
vaibhav-ank@LAPTOP-TUCAJV9E:~$
```

- 2) When the key is added, the system will return OK. Next, append the Debian package repository address to the server's sources.list:

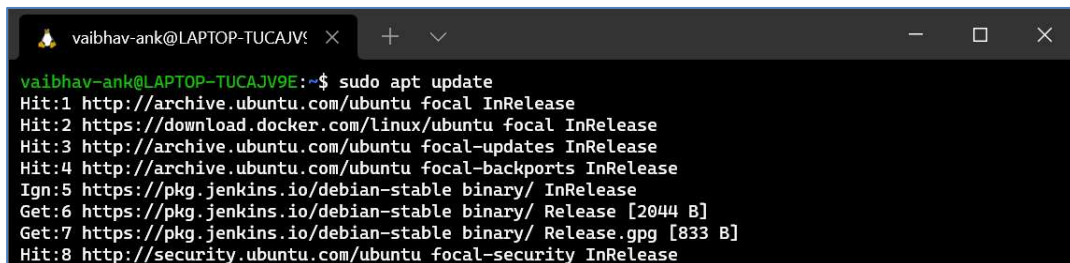
```
$ sudo bash -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ >  
/etc/apt/sources.list.d/jenkins.list'
```



```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ sudo bash -c 'echo deb https://pkg.jenkins.io/debian-stable binary/  
> /etc/apt/sources.list.d/jenkins.list'  
vaibhav-ank@LAPTOP-TUCAJV9E:~$
```

- 3) When both of these are in place, run update so that apt will use the new repository:

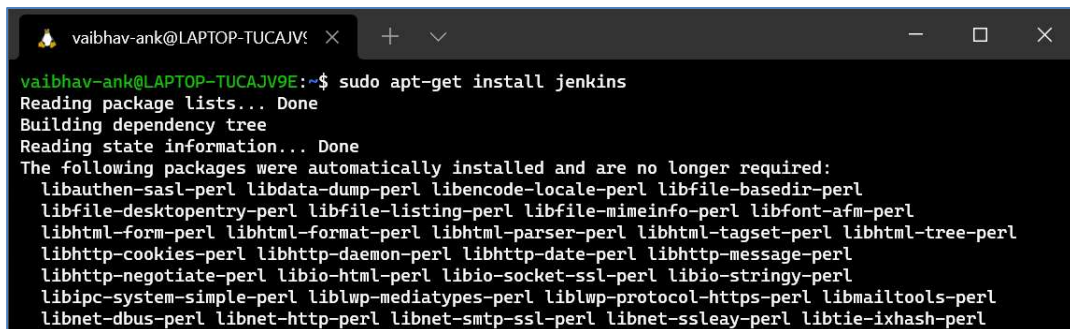
```
$ sudo apt update
```



```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ sudo apt update  
Hit:1 http://archive.ubuntu.com/ubuntu focal InRelease  
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease  
Hit:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease  
Hit:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease  
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]  
Get:7 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]  
Hit:8 http://security.ubuntu.com/ubuntu focal-security InRelease
```

- 4) Finally, install Jenkins and its dependencies:

```
$ sudo apt-get install jenkins
```

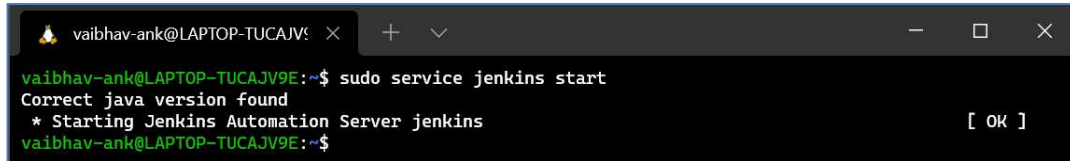


```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ sudo apt-get install jenkins  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
libauthen-sasl-perl libdata-dump-perl libencode-locale-perl libfile-basedir-perl  
libfile-desktopentry-perl libfile-listing-perl libfile-mimeinfo-perl libfont-afm-perl  
libhtml-form-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl  
libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl libhttp-message-perl  
libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl libio-stringy-perl  
libipc-system-simple-perl liblwp-mediatypes-perl liblwp-protocol-https-perl libmailtools-perl  
libnet-dbus-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libtie-ixhash-perl
```

PART 3 : Configure Jenkins

Run Jenkins server

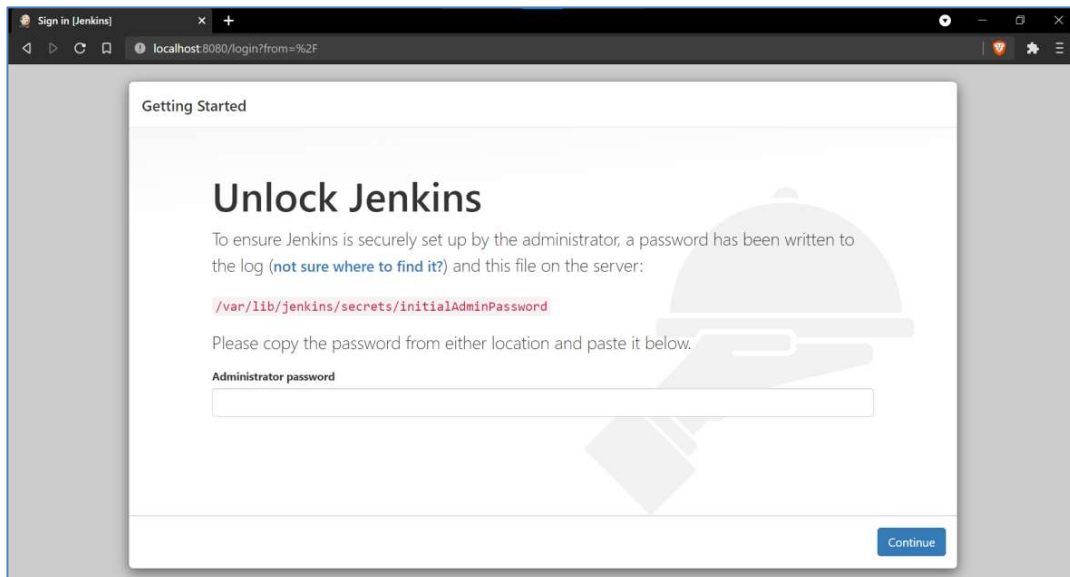
```
$ sudo systemctl start jenkins    OR    $ sudo service jenkins start
```

A terminal window with the title 'vaibhav-ank@LAPTOP-TUCAJV9E'. The command 'sudo service jenkins start' has been executed. The output shows 'Correct java version found' and '* Starting Jenkins Automation Server jenkins'. A '[OK]' status is visible on the right side of the terminal.

```
vaibhav-ank@LAPTOP-TUCAJV9E:~$ sudo service jenkins start
Correct java version found
* Starting Jenkins Automation Server jenkins
vaibhav-ank@LAPTOP-TUCAJV9E:~$
```

Navigate to your host browser, and type:

`http://localhost:8080/`



If nothing shows up, open the port 8080 with ufw:

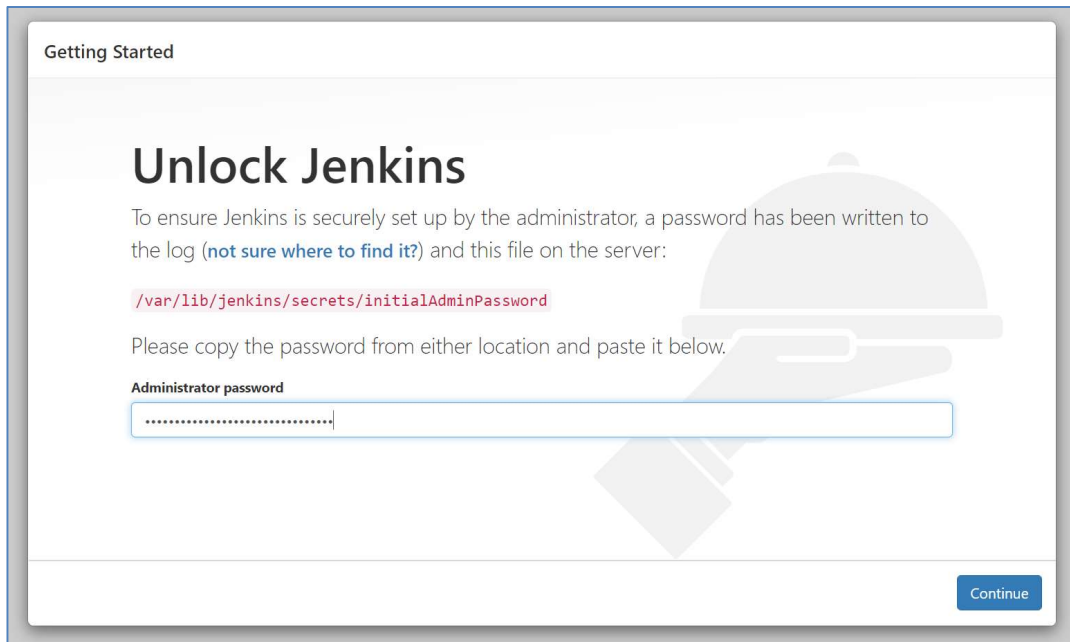
```
$ sudo ufw allow OpenSSH
$ sudo ufw enable
$ sudo ufw allow 8080
```

where you will be prompted to provide an initial password to unlock Jenkins. Follow up the screen instructions and in the terminal, type below:

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



Copy the password you retrieved to clipboard. Return to the host browser window, and paste the password there

The image shows the 'Unlock Jenkins' screen in the Jenkins web interface. At the top, it says 'Getting Started'. The main heading is 'Unlock Jenkins'. Below it, a paragraph explains that a password was written to the log (not sure where to find it?) and this file on the server: `/var/lib/jenkins/secrets/initialAdminPassword`. It then asks the user to please copy the password from either location and paste it below. There is a text input field labeled 'Administrator password' with a masked password '.....'. A 'Continue' button is at the bottom right.

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:

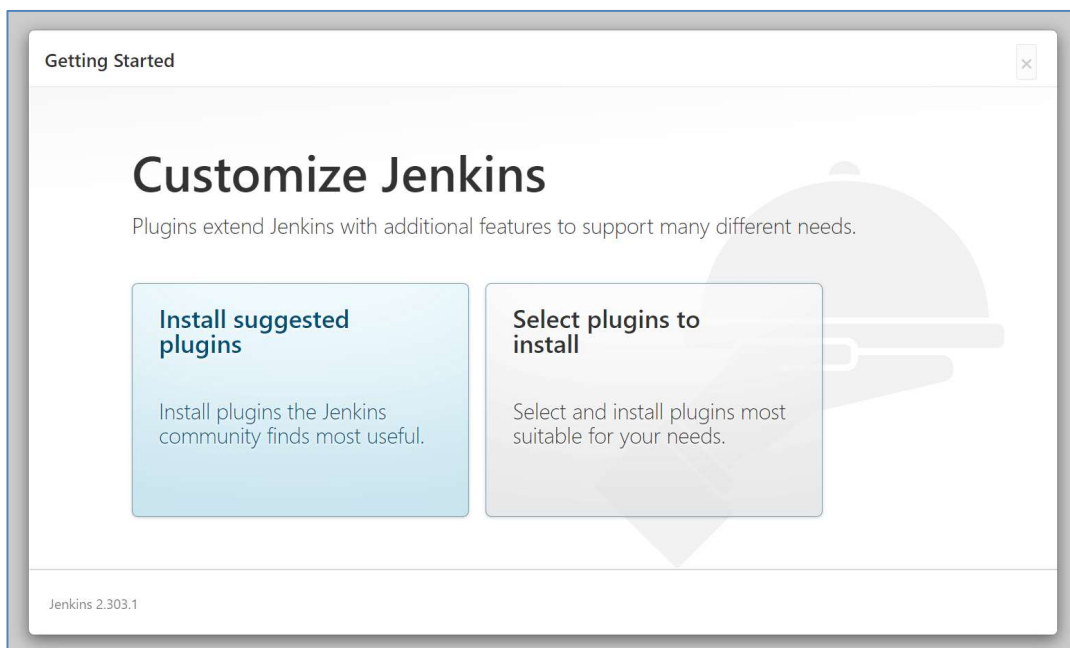
```
/var/lib/jenkins/secrets/initialAdminPassword
```

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

Administrator password

Continue

Click "Continue" button!

The image shows the 'Customize Jenkins' screen in the Jenkins web interface. At the top, it says 'Getting Started'. The main heading is 'Customize Jenkins'. Below it, a paragraph explains that plugins extend Jenkins with additional features to support many different needs. There are two main options: 'Install suggested plugins' (highlighted in blue) and 'Select plugins to install' (highlighted in grey). The 'Install suggested plugins' option says 'Install plugins the Jenkins community finds most useful.' The 'Select plugins to install' option says 'Select and install plugins most suitable for your needs.' The Jenkins version 'Jenkins 2.303.1' is shown at the bottom left.

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.

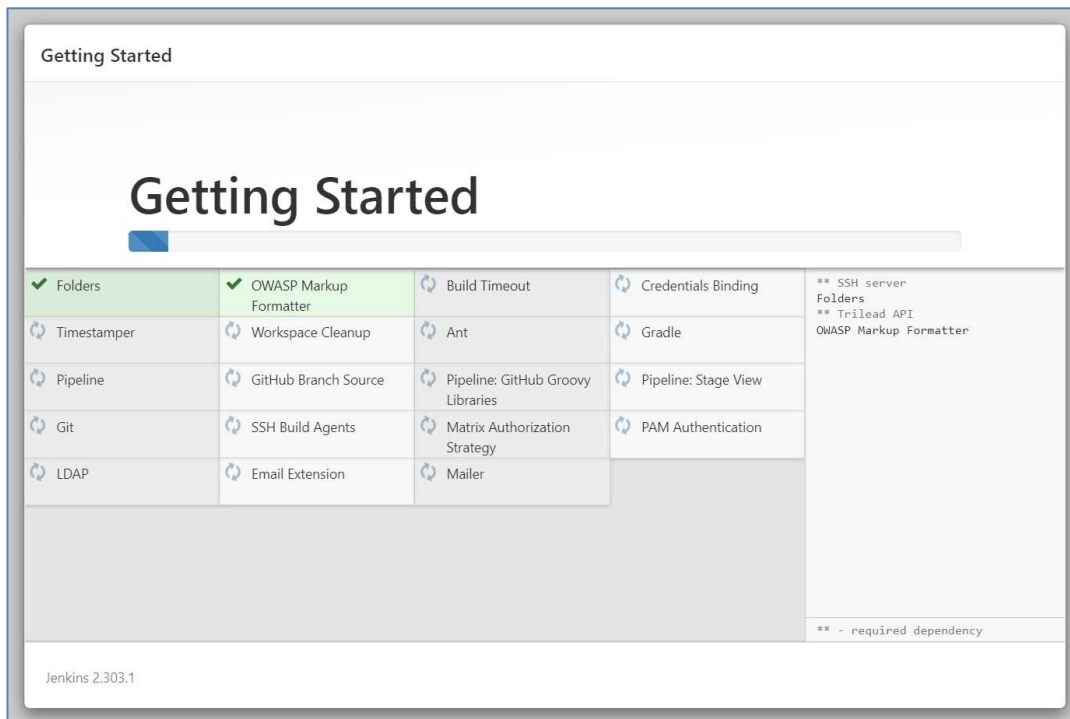
Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.303.1

Choose ***Install suggested plugins*** (or *Select plugins to install* if you want to install custom plugins).

All plugins will be installed in Jenkins now.



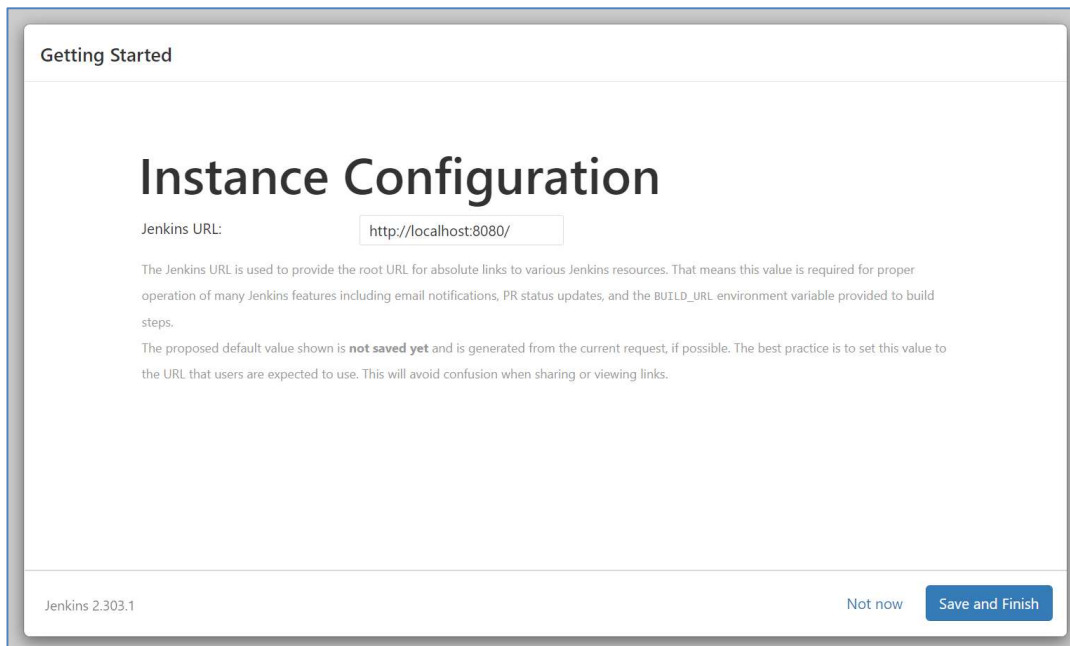
It will ask you to create an Admin User, fill the details and click *Continue*.

The screenshot shows the 'Getting Started' screen in Jenkins 2.303.1, specifically the 'Create First Admin User' section. The form contains the following fields:

- Username: vaibhav-ank
- Password:
- Confirm password:
- Full name: Vaibhav Ankolekar
- E-mail address: vaibhavank18@gmail.com

At the bottom, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The version 'Jenkins 2.303.1' is displayed in the bottom left corner.

Leave the Jenkins URL default or change the port number. Then click *Save and Finish*.



The screenshot shows the 'Getting Started' section of the Jenkins Instance Configuration page. The title 'Instance Configuration' is prominently displayed. Below it, the 'Jenkins URL' field is pre-filled with 'http://localhost:8080/'. A detailed explanation of the Jenkins URL's purpose is provided, noting its role in providing absolute links and its use in email notifications and build steps. A warning indicates that the default value is 'not saved yet'. At the bottom right, there are two buttons: 'Not now' and 'Save and Finish'.

Getting Started

Instance Configuration

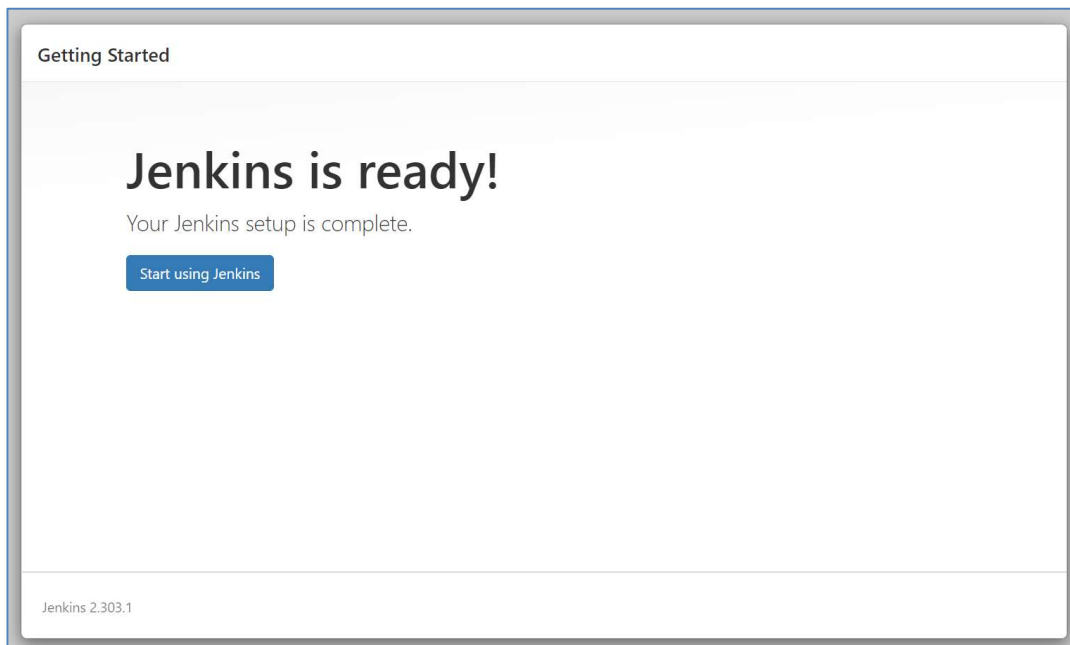
Jenkins URL:

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 2.303.1 Not now Save and Finish

Jenkin's setup is completed. Click on *Start using Jenkins*.



The screenshot shows the 'Getting Started' section of the Jenkins 'Jenkins is ready!' page. The title 'Jenkins is ready!' is prominently displayed. Below it, a message states 'Your Jenkins setup is complete.' A single button labeled 'Start using Jenkins' is centered on the page. The Jenkins version '2.303.1' is shown in the bottom left corner.

Getting Started

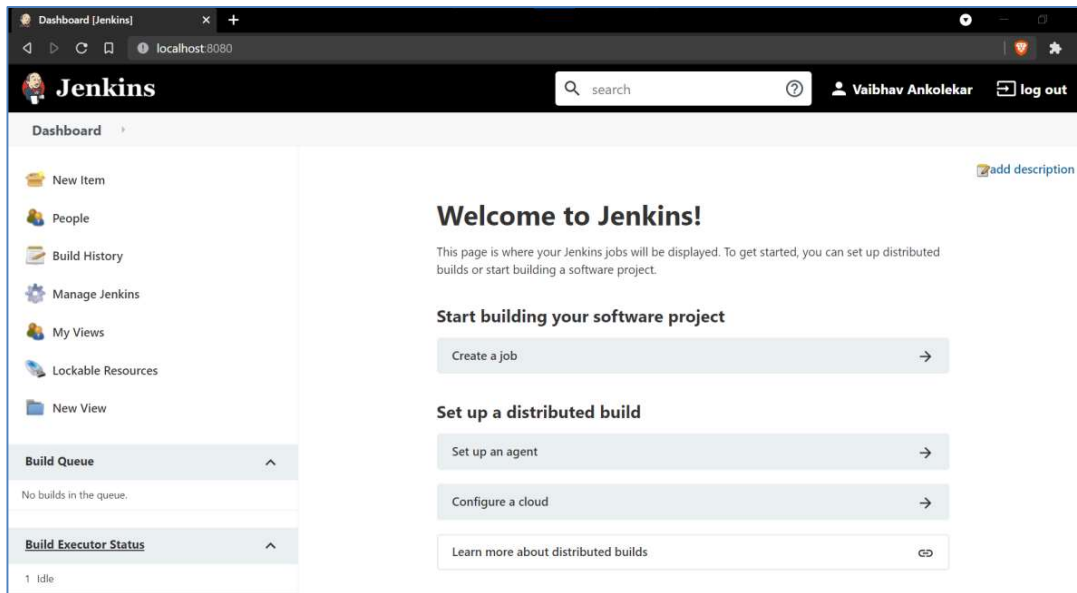
Jenkins is ready!

Your Jenkins setup is complete.

Start using Jenkins

Jenkins 2.303.1

Now the Jenkins Dashboard will be displayed. You can now use Jenkins to continuously trigger build and test for every change made in the source code



Conclusion :

Jenkins is used to build and test your product continuously, so developers can continuously integrate changes into the build. Jenkins is the most popular open-source CI/CD tool on the market today and is used in support of DevOps, alongside other cloud native tools. Thus, we have successfully installed Jenkins.