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.

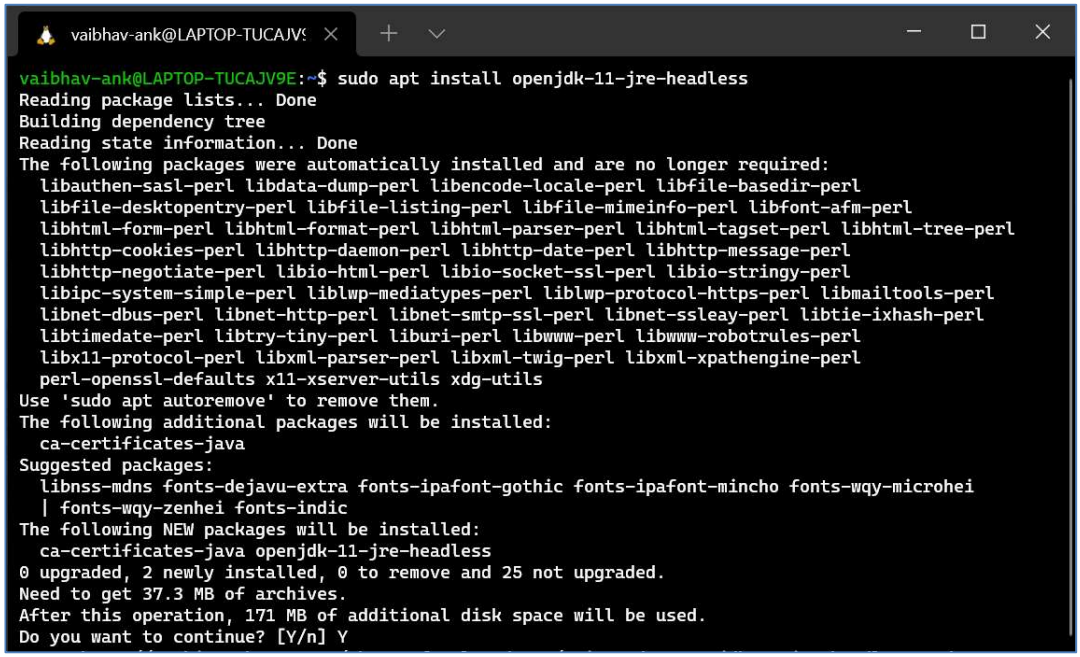
Advantages of Jenkins include:

- 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

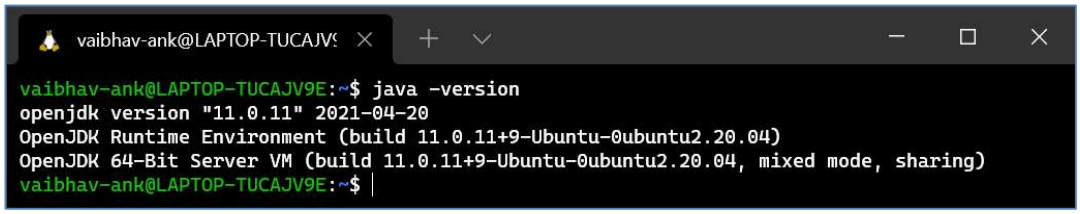
To install Java run any one command from below :

```
$ sudo apt install openjdk-8-jdk
        OR
$ sudo apt install openjdk-11-jre-headless
```



Check if java is installed using following command.

```
$ java -version
```



PART 2 : Install Jenkins

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

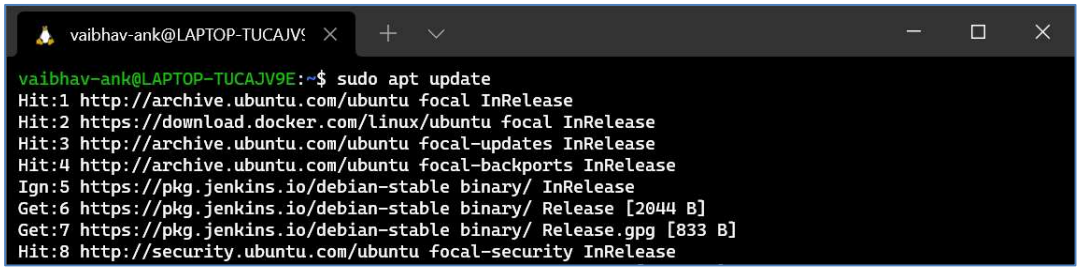
```
$ wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key |
  sudo apt-key add
```

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'
```

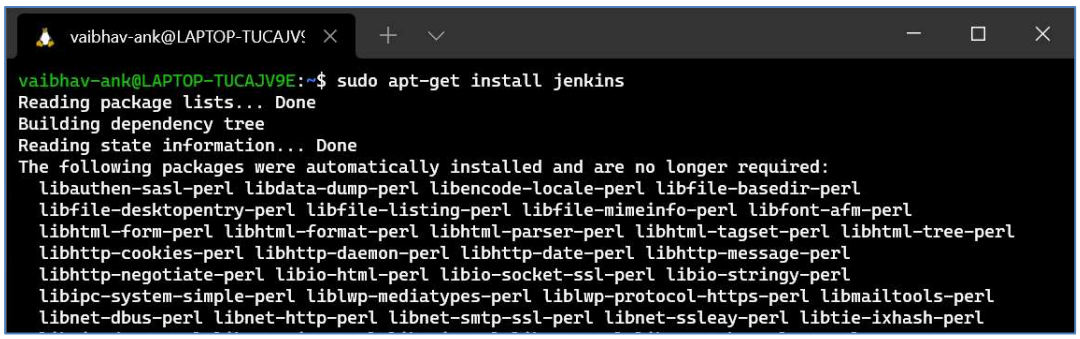
3) When both of these are in place, run update so that apt will use the new repository:

```
$ sudo apt update
```



4) Finally, install Jenkins and its dependencies:

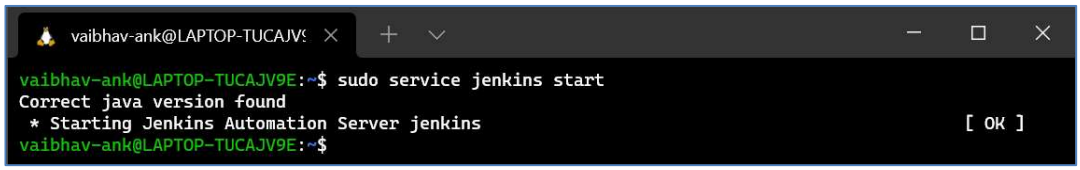
```
$ sudo apt-get install jenkins
```



PART 3 : Configure Jenkins

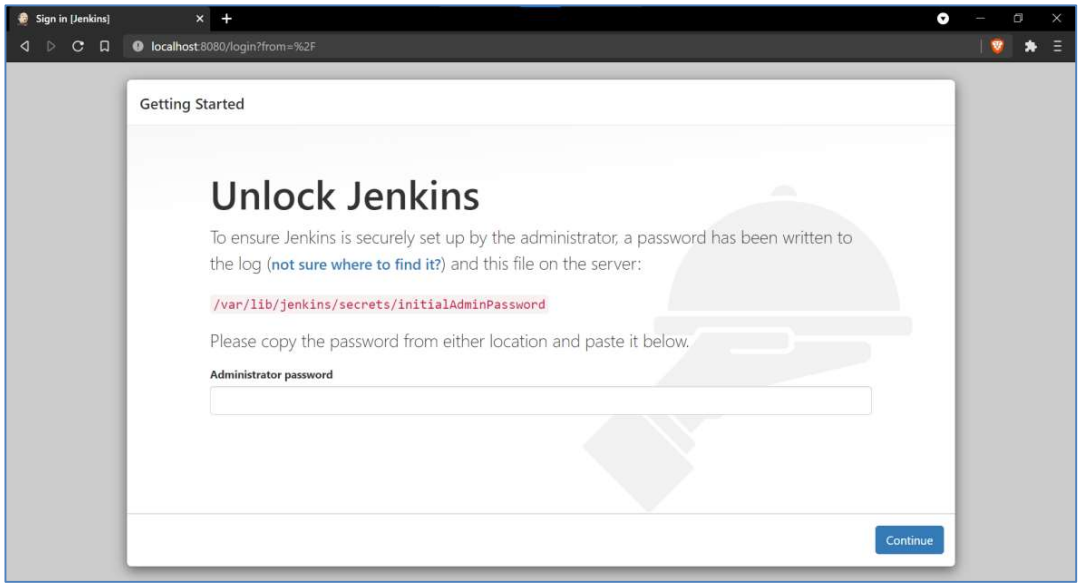
Run Jenkins server

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



Navigate to your host browser, and type:

```
http://localhost:8080/
```

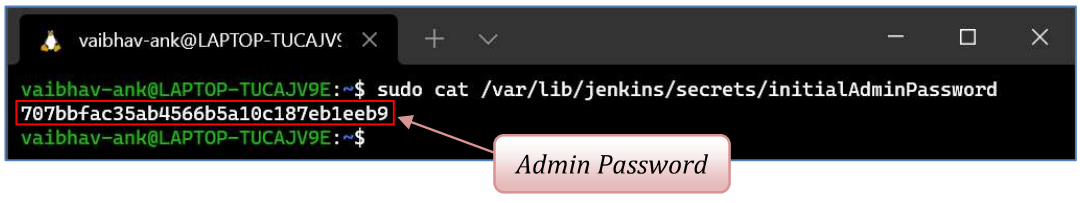


If nothing shows up, open the port 8080 with ufw:

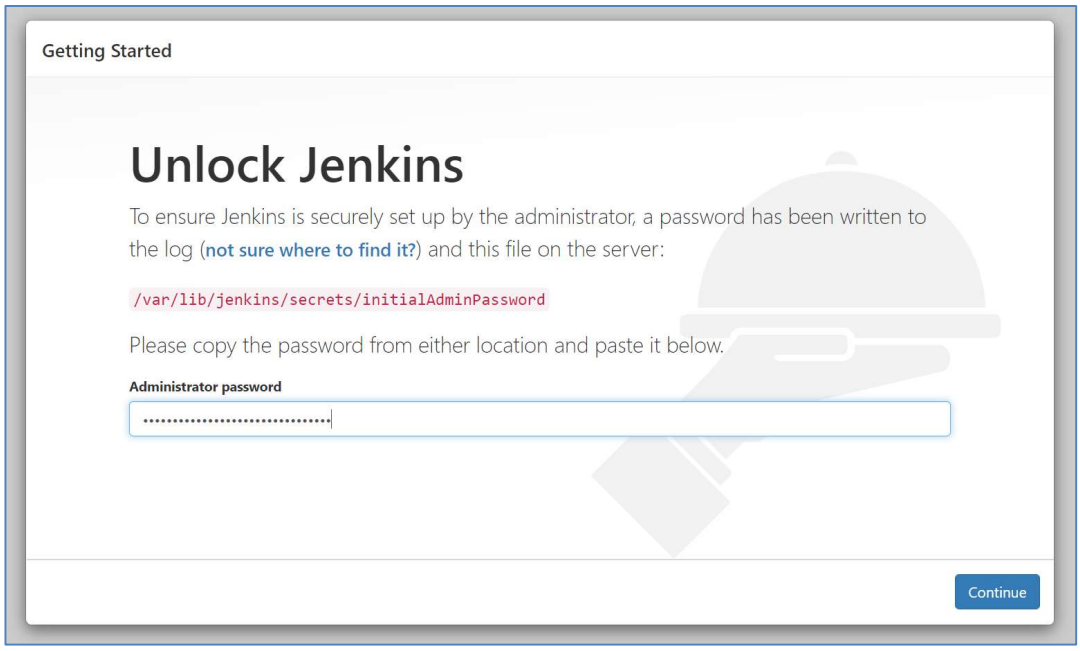
```
$ sudo ufw allow OpenSSH
$ sudo ufw enable
$ sudo ufw allow 8080
```

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

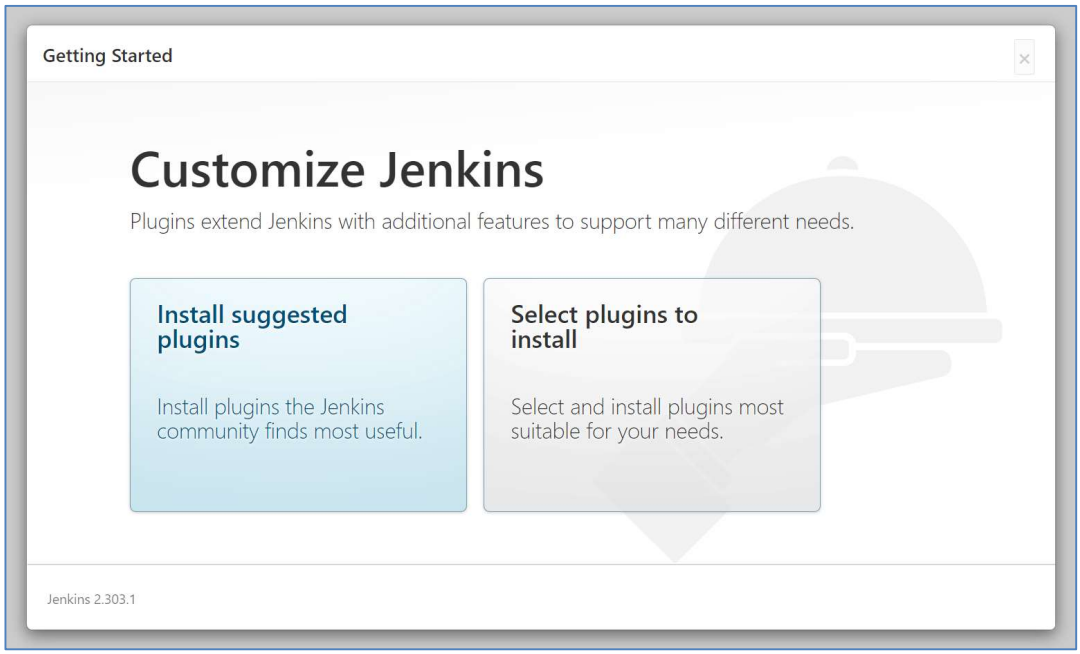
```
$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```



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

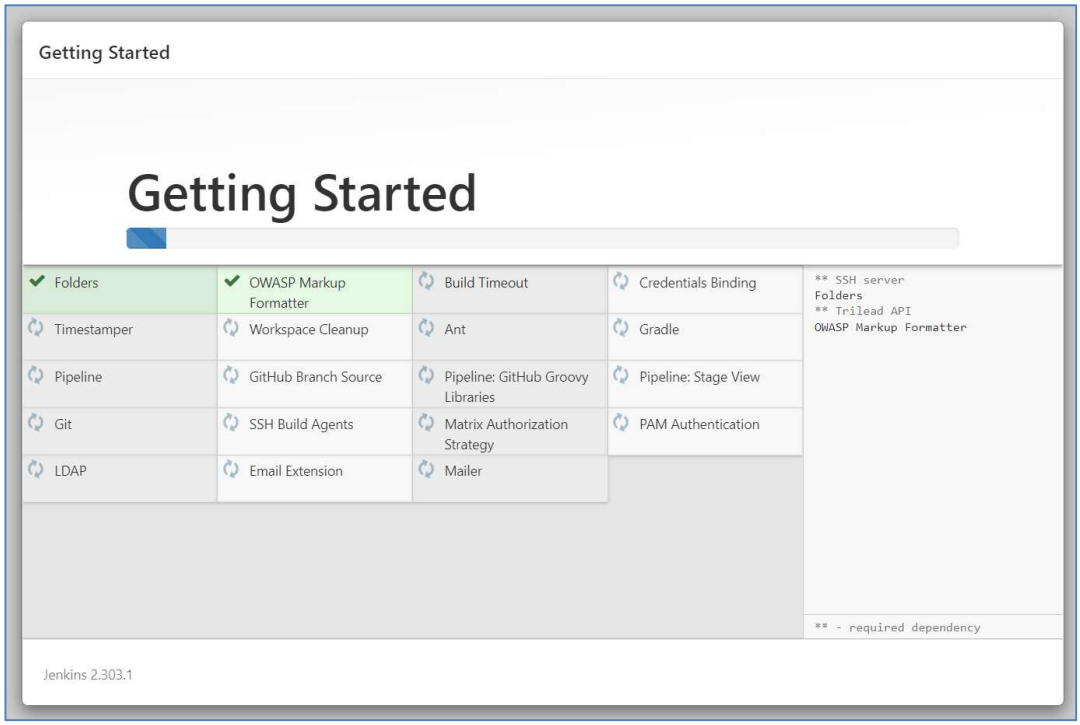


Click "*Continue*" button!

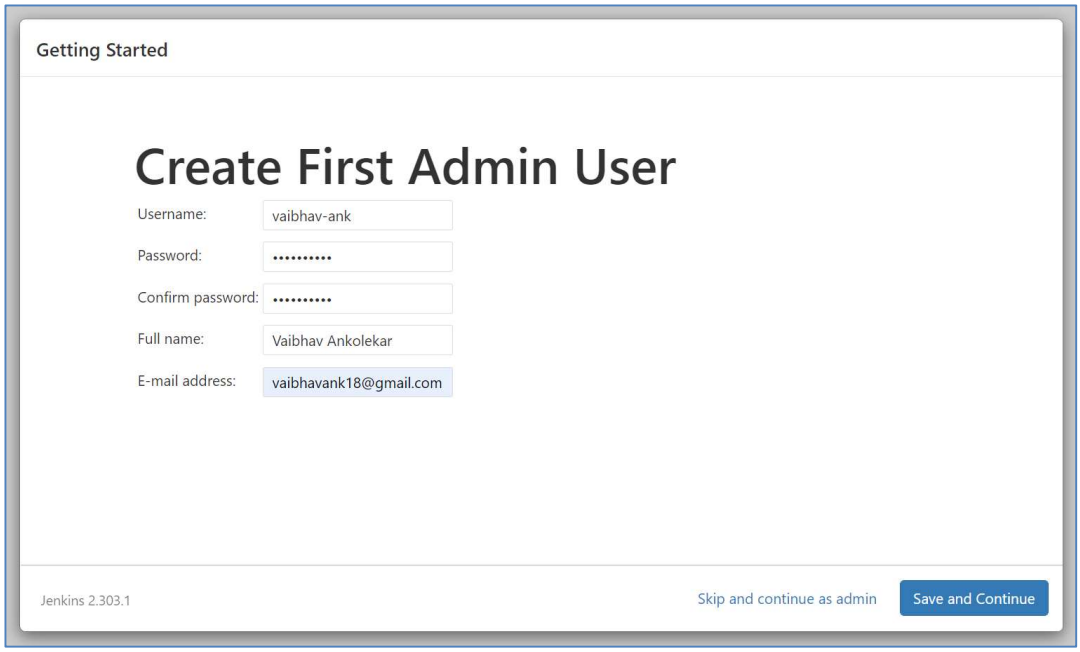


Choose ***Install suggested plugins*** (or *Select plugins to install* if you want to install custom plugins).

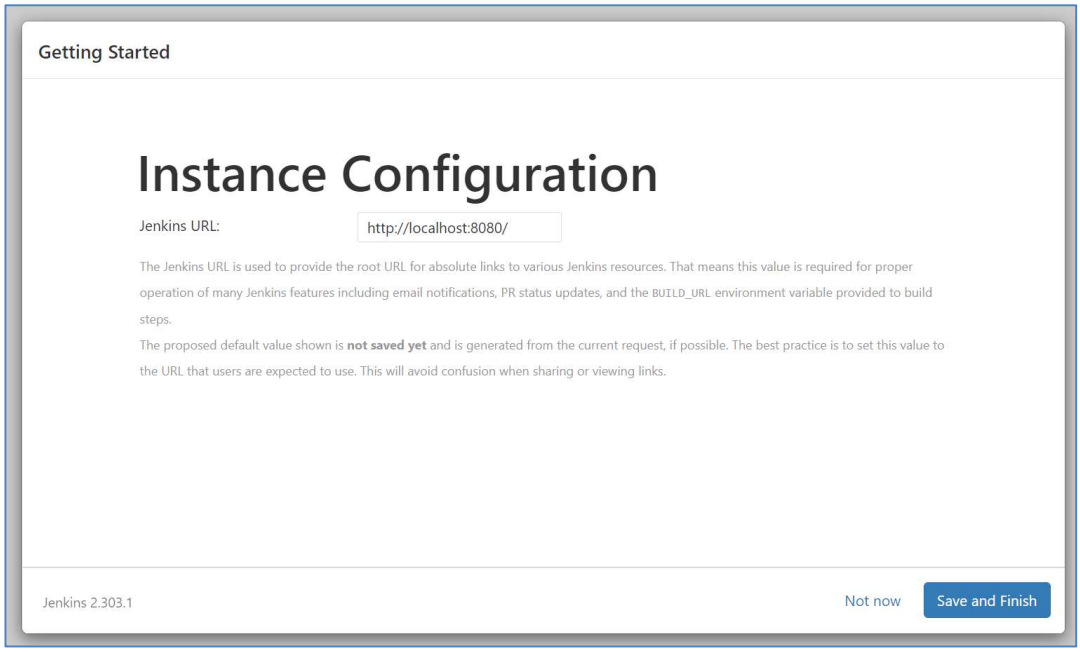
All plugins will be installed in Jenkins now.



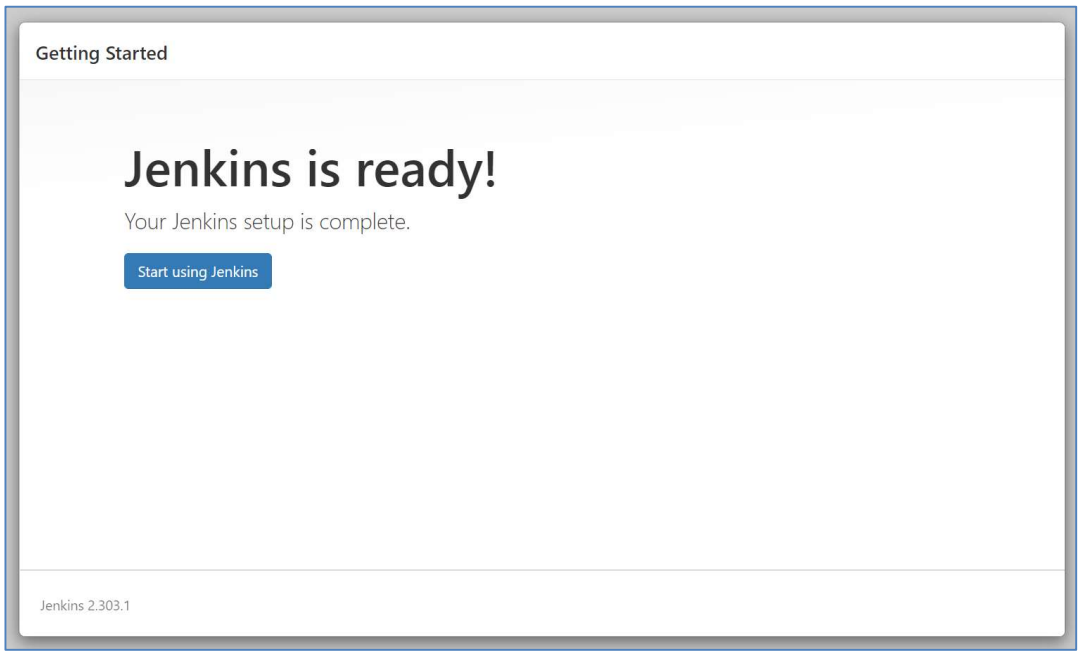
It will ask you to create an Admin User, fill the details and click *Continue*.



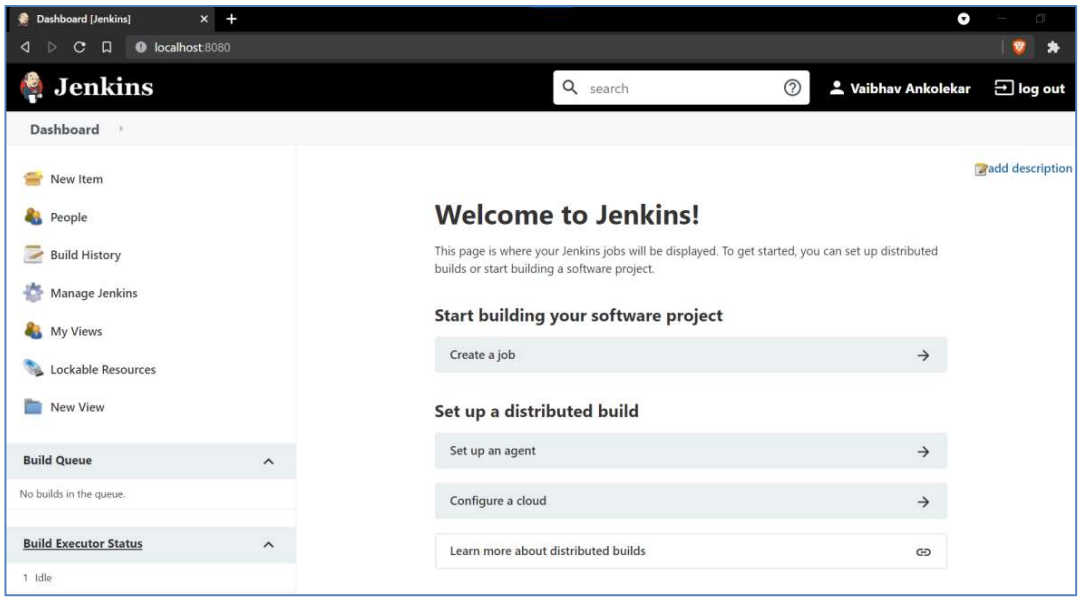
Leave the Jenkins URL default or change the port number. Then click *Save and Finish*.



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



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.