Experiment No. 3

1. Aim: To install and configure Jenkins to setup a build Job.

2. **Objectives:** From this experiment, the student will be able
   9. To install and build job in Jenkins for deploying application DevOps environment.

3. Outcomes: The learner will be able to
   10. To understand the importance of Jenkins to Build and deploy Software Applications on server environment.

4. Hardware / Software Required: Linux / Windows Operating System

5. Theory:

Jenkins is a Java-based open-source automation platform with built-in continuous integration plugins. Jenkins is used to continuously build and test your software projects, making it simpler for developers to incorporate changes to the project and for users to get a new build. By interacting with a wide range of testing and deployment tools, it also enables you to release your software continually.

Organizations can use Jenkins to automate and speed up the software development process. Jenkins combines all stages of the development lifecycle, including build, document, test, package, stage, deploy, static analysis, and many others.

Jenkins achieves Continuous Integration with the help of plugins. Plugins allow the integration of Various DevOps stages. If you want to integrate a particular tool, you need to install the plugins for that tool. For example: Git, Maven 2 project, Amazon EC2, HTML publisher etc.

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.

Jenkins Features:

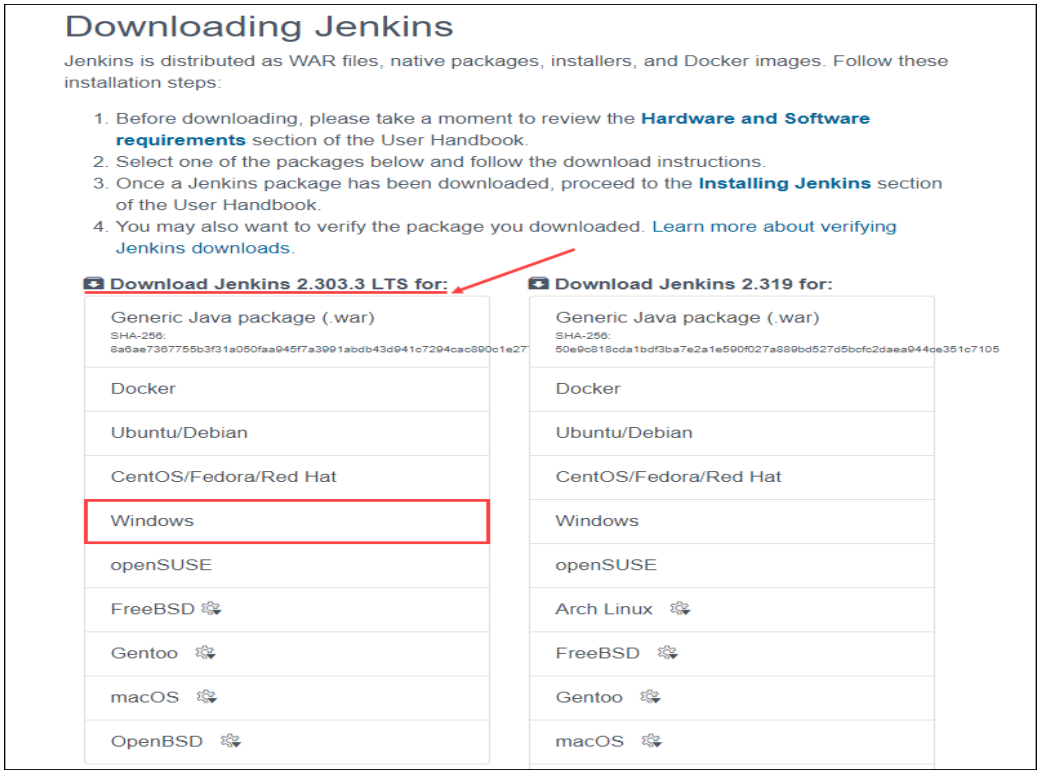
11. Continuous Integration and Continuous Delivery: As an extensible automation server, Jenkins can be used as a simple CI server or turned into the continuous delivery hub for any project.

12. **Easy configuration**
    Jenkins can be easily set up and configured via its web interface, which includes on-the-fly error checks and built-in help.
13. **Easy installation**
    Jenkins is a self-contained Java-based program, ready to run out-of-the-box, with packages for Windows, Linux, macOS and other Unix-like operating systems.
14. **Plugins**
    With hundreds of plugins in the Update Center, Jenkins integrates with practically every tool in the continuous integration and continuous delivery toolchain.
15. **Extensible**
    Jenkins can be extended via its plugin architecture, providing nearly infinite possibilities for what Jenkins can do.
16. **Distributed**
    Jenkins can easily distribute work across multiple machines, helping drive builds, tests and deployments across multiple platforms faster.

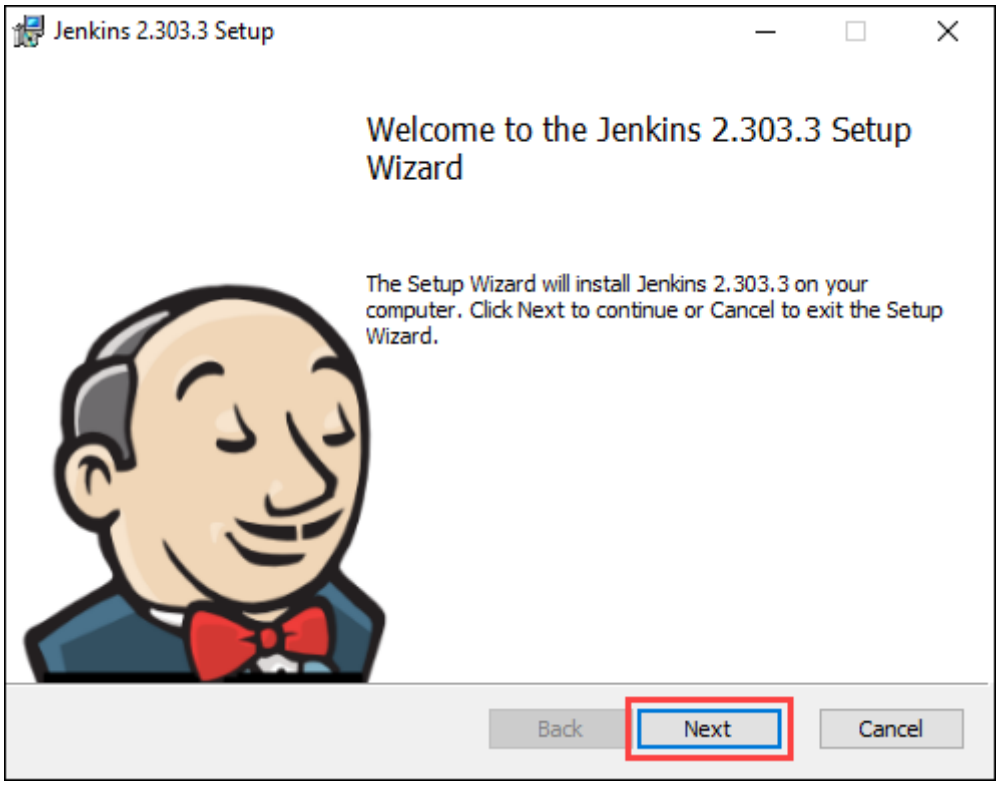
Steps for Installation of Jenkins:

1. **Installation of JAVA:** Download JDK 11/17 and choose windows 32-bit or 64-bit according to your system configuration. Click on "accept the license agreement." Set the Path for the Environmental Variable for JDK:
   - Go to System Properties. Under the "Advanced" tab, select "Environment Variables."
   - Under system variables, select "new." Then copy the path of the JDK folder and paste it in the corresponding value field. Similarly, do this for JRE.
   - Under system variables, set up a bin folder for JDK in PATH variables.
   - Go to command prompt and type the following to check if Java has been successfully installed:

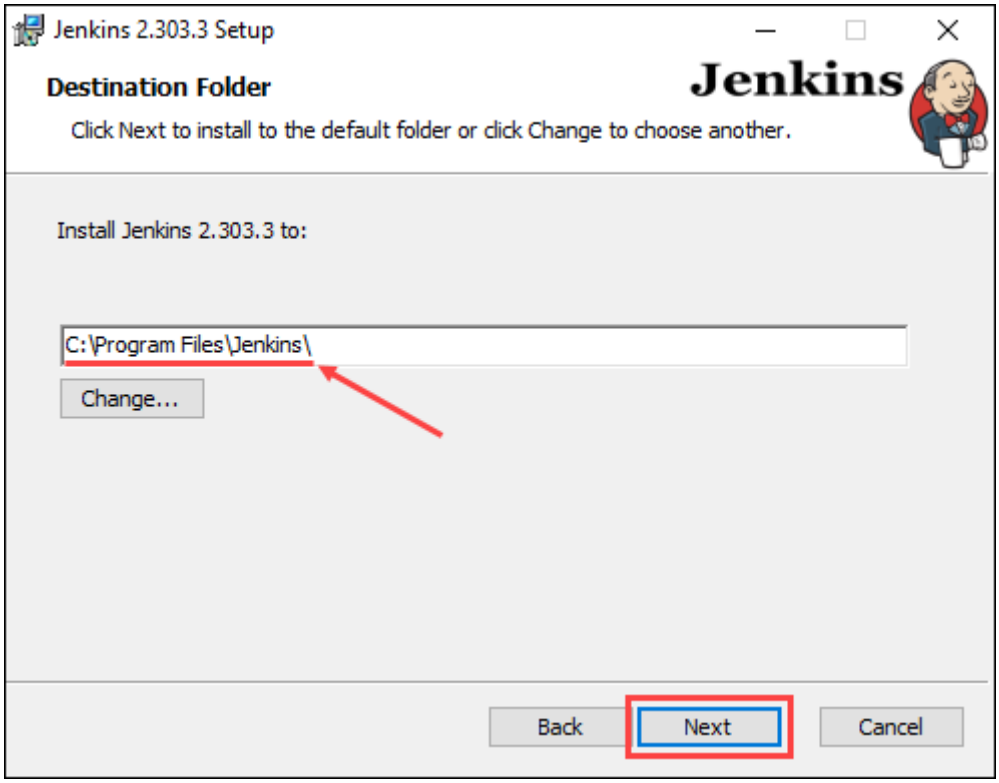
     C:\Users\Aditi>java -version
2. Browse to the official Jenkins download page. Under the Downloading Jenkins section is a list of installers for the long-term support (LTS) version of Jenkins. Click the Windows link to begin the download.
   Once the download is complete, run the **jenkins.msi** installation file.



3. The setup wizard starts. Click **Next** to proceed.



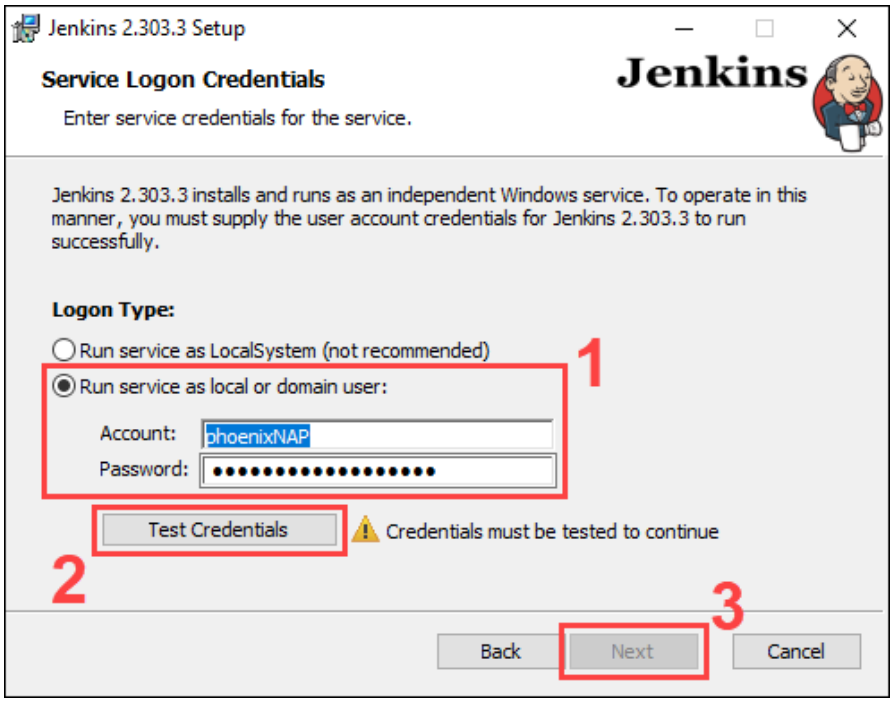
4. Select the install destination folder and click **Next** to continue.



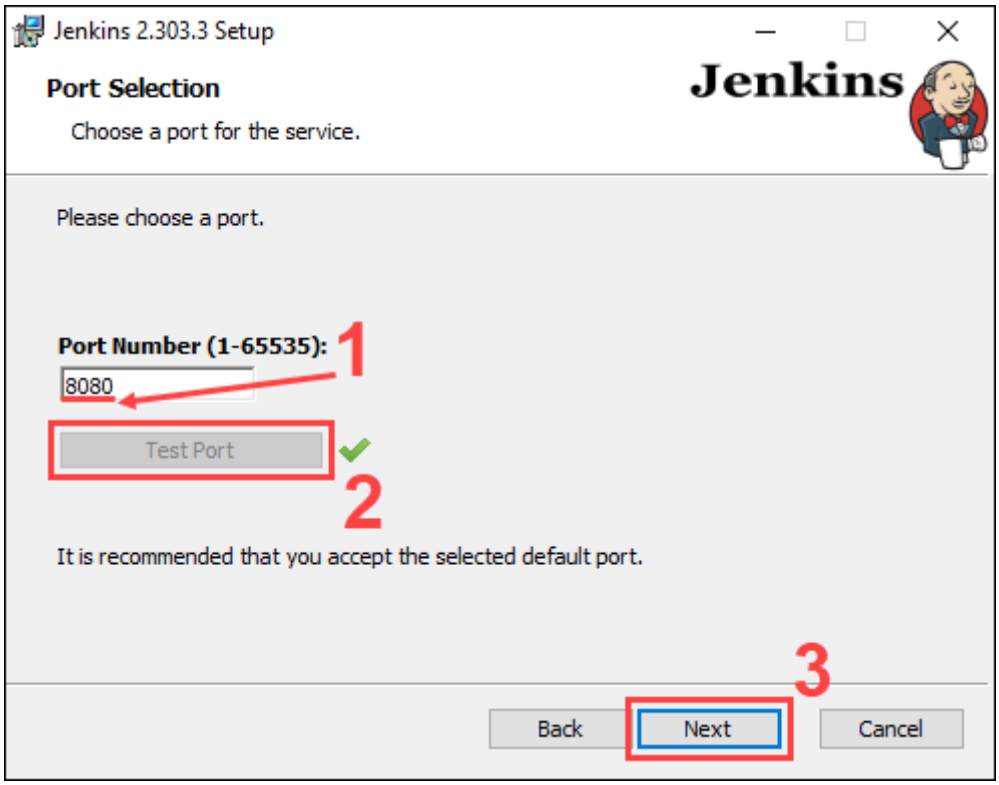
5. Under the **Run service as a local or domain user** option, enter the domain username and password for the user account you want to run Jenkins with. Click **Test Credentials** to verify the login data, then click **Next** to proceed.

Or

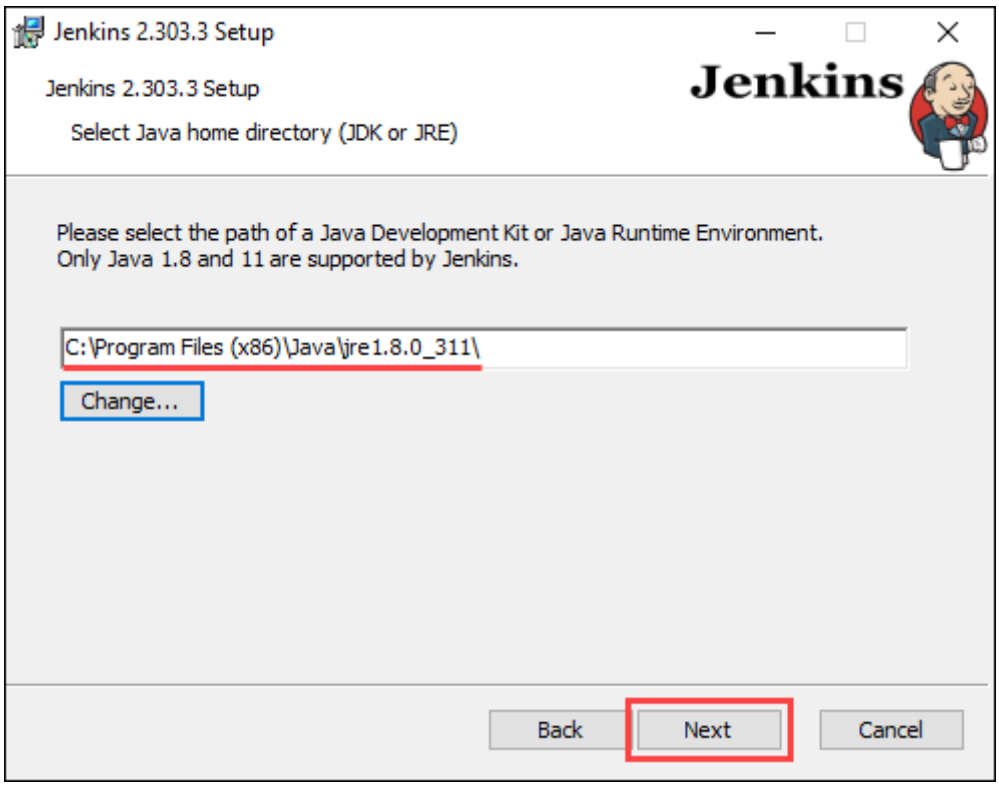
Using the **Run service as LocalSystem** option doesn't require you to enter user login data. Instead, it grants Jenkins full access to your system and services. Note that this is a less secure option and is thus not recommended.



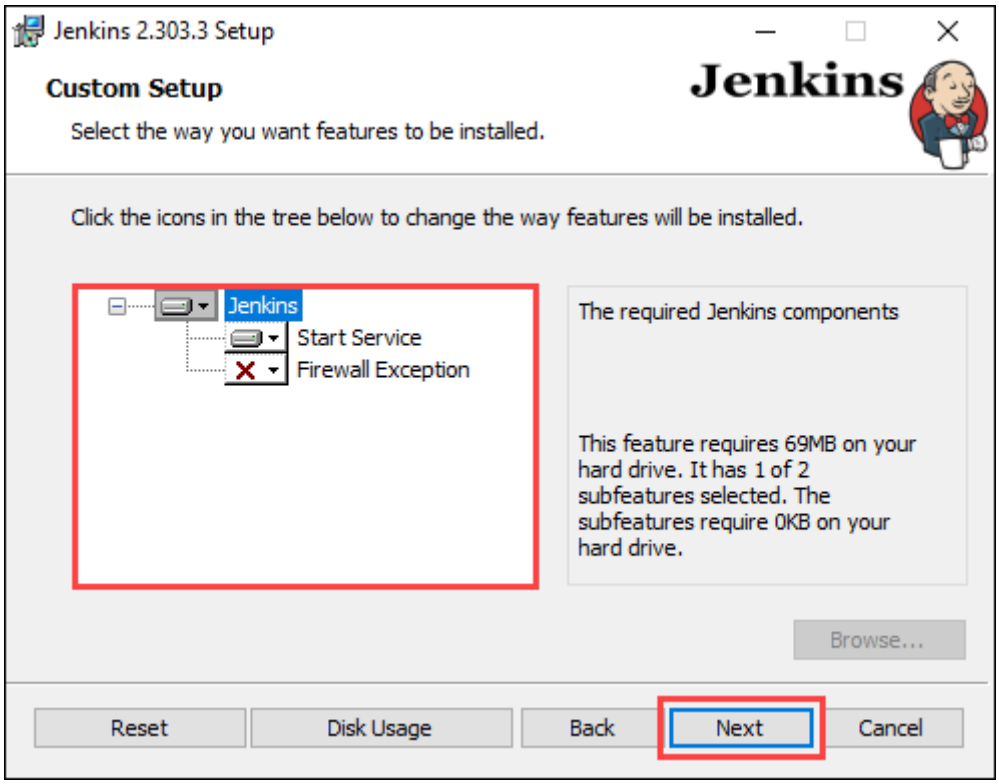
6.Enter the port number you want Jenkins to run on. Click **Test Port** to check if the selected port is available, then click **Next** to continue.



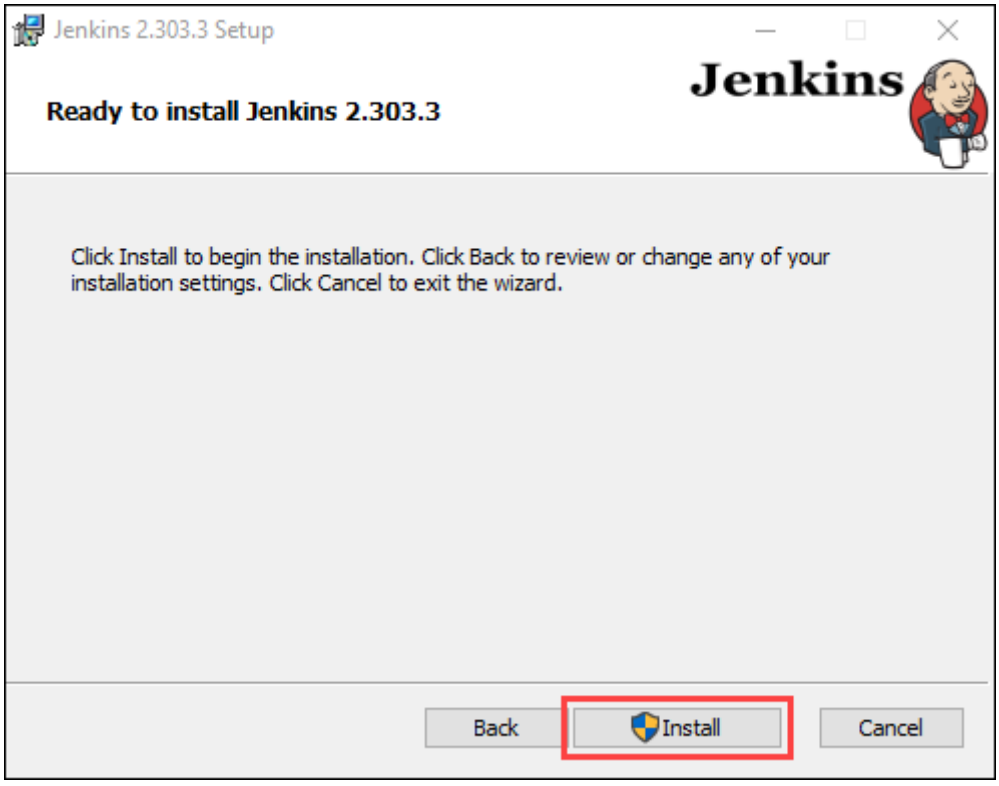
7. Select the directory where Java is installed on your system and click **Next** to proceed.



8. Select the features you want to install with Jenkins and click **Next** to continue.



9. Click **Install** to start the installation process.



10. Once the installation is complete, click **Finish** to exit the install wizard.

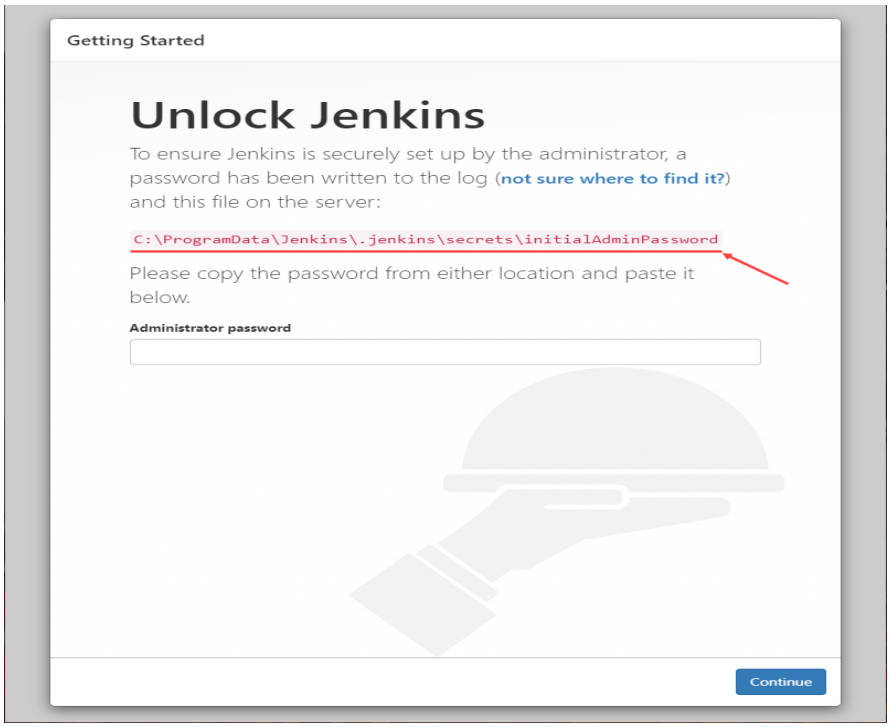
Unblock Jenkins

After completing the installation process, you have to unblock Jenkins before you can customize and start using it.

1. In your web browser, navigate to the port number you selected during the installation using the following address:

http://localhost:[port number]

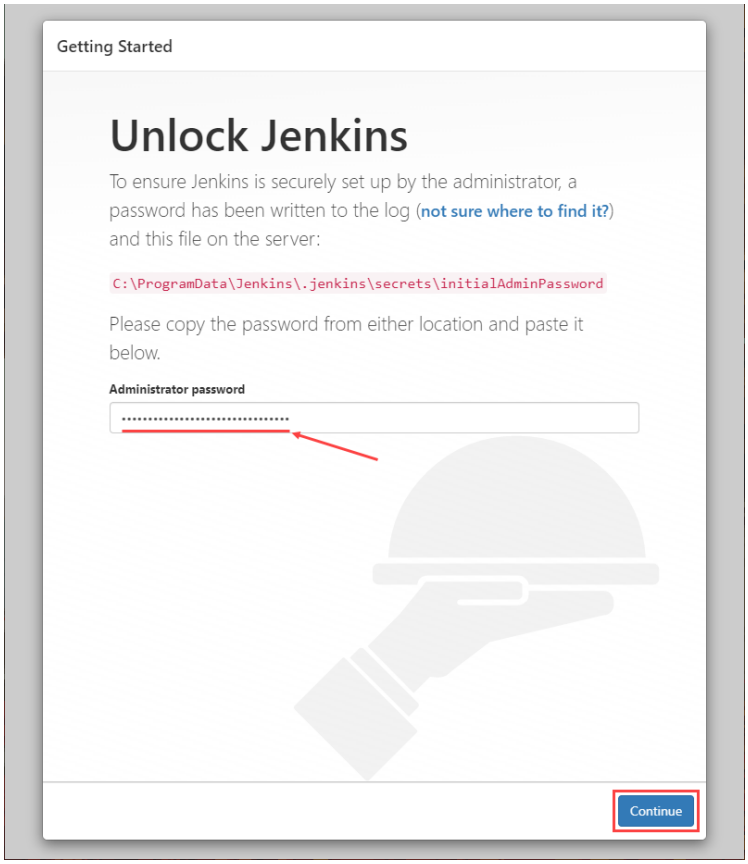
2. Navigate to the location on your system specified by the Unblock Jenkins page.



3. Open the **initialAdminPassword** file using a text editor such as Notepad.

4. Copy the password from the **initialAdminPassword** file.

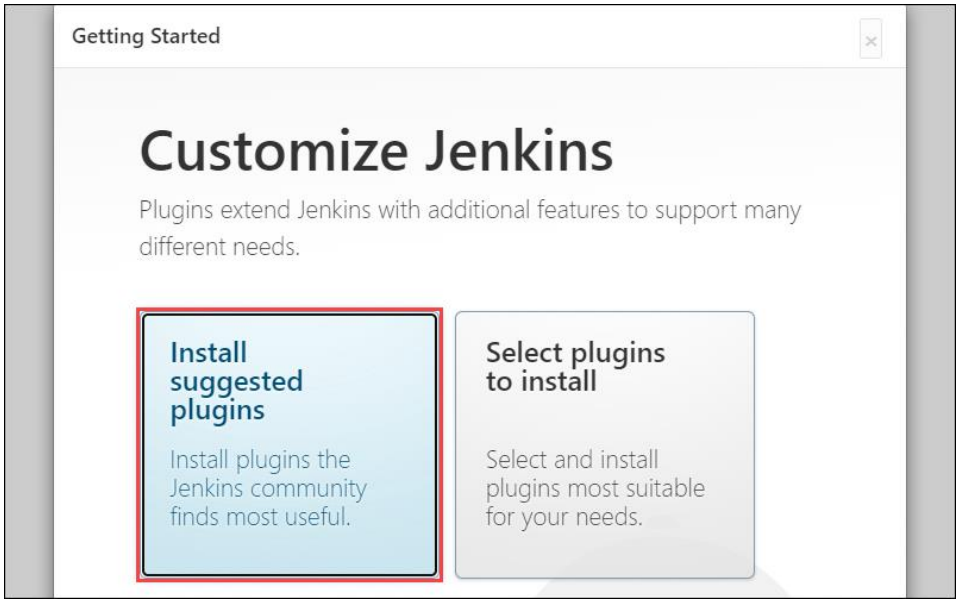
5. Paste the password in the **Administrator password** field on the Unblock Jenkins page and click **Continue** to proceed.



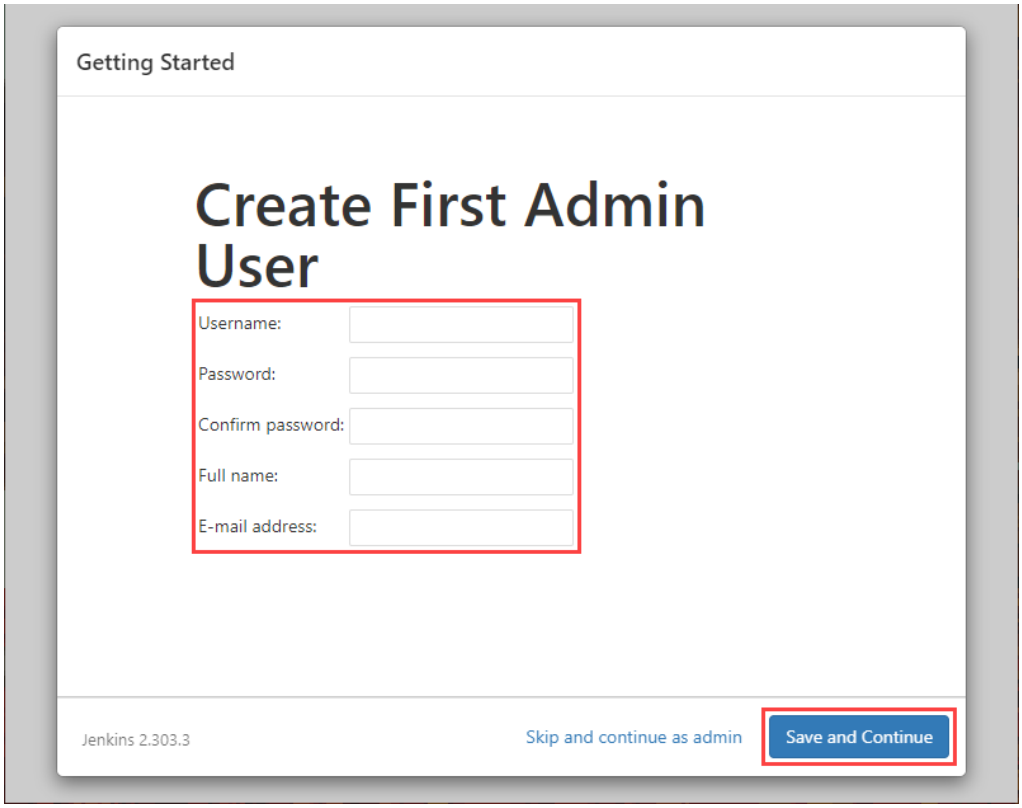
Customize Jenkins

Once you unlock Jenkins, customize and prepare the Jenkins environment.

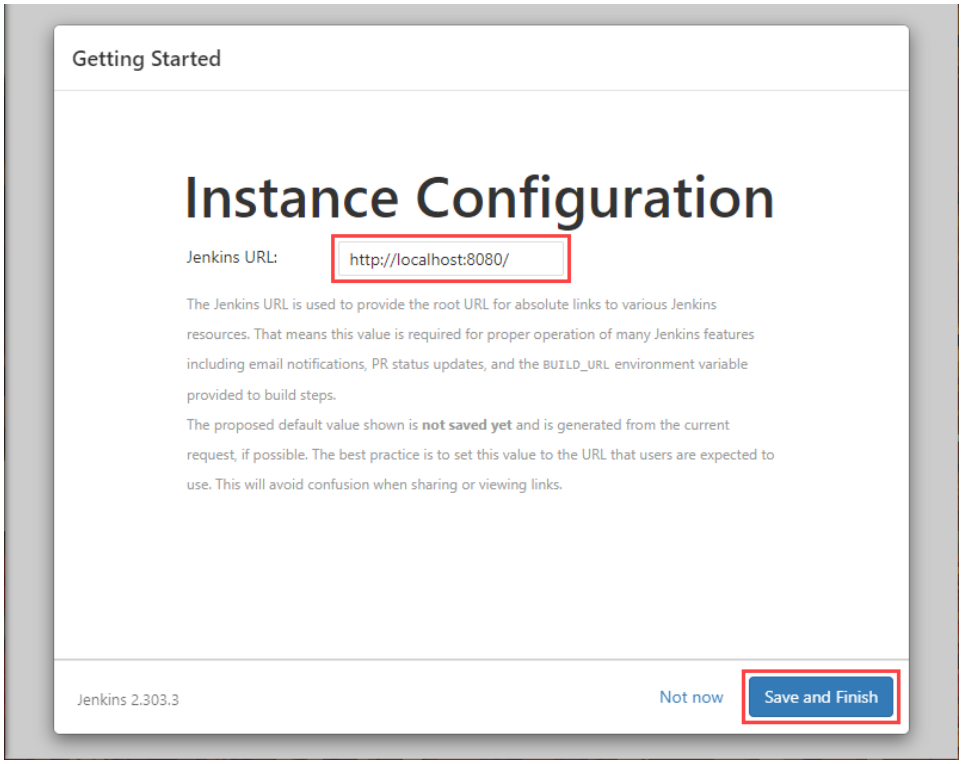
1. Click the **Install suggested plugins** button to have Jenkins automatically install the most frequently used plugins.



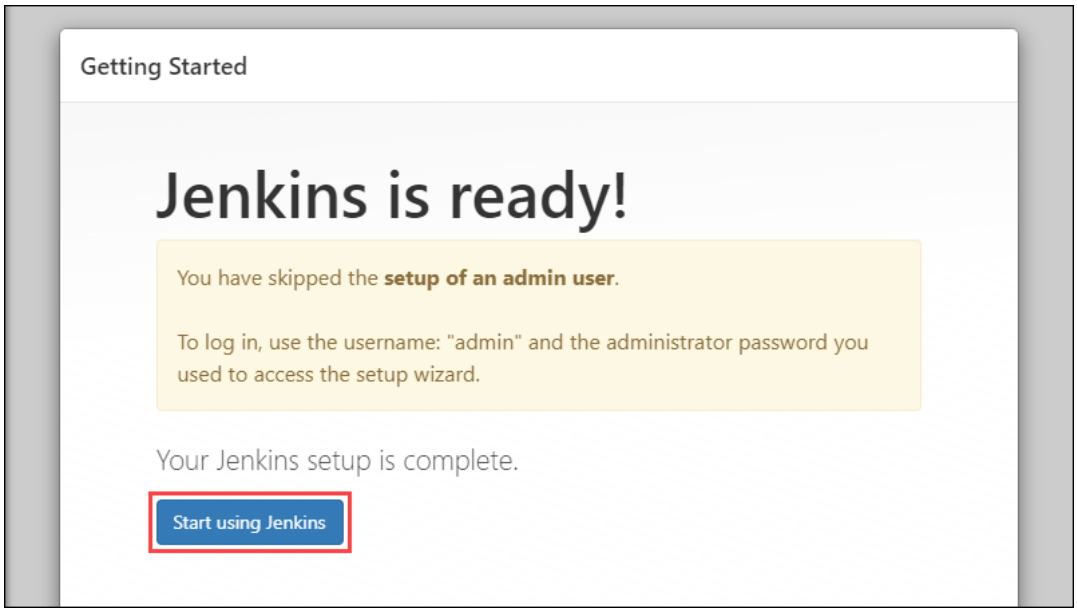
2. After Jenkins finishes installing the plugins, enter the required information on the **Create First Admin User** page. Click **Save and Continue** to proceed.



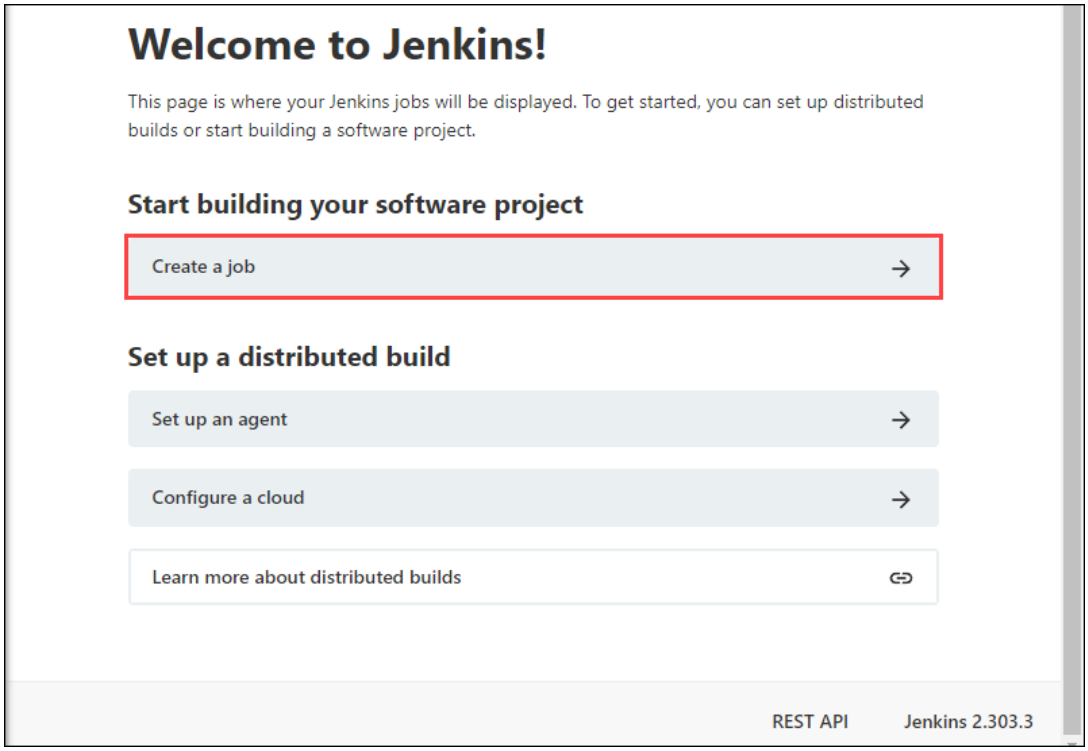
3. On the **Instance Configuration** page, confirm the port number you want Jenkins to use and click **Save and Finish** to finish the initial customization.



4. Click the **Start using Jenkins** button to move to the Jenkins dashboard.



5. Using the Jenkins dashboard, click **Create a job** to build your first Jenkins software project.



Students should build a job and execute in Jenkins environment.

6. **Conclusion and Discussion:**

Students are supposed to write your own conclusion

7. **Viva Questions:**
   17. What is Continuous Integration?
   18. What is CI/CD?
   19. Which tools can be plugged with Jenkins?

8. **References:**

1. Jenkins: The Definitive Guide: Continuous Integration for the Masses, by John Ferguson Smart Published by O'Reilly Media

2. Jenkins 2: Up and Running Authored by Brent Laster Published by O'Reilly Media