**Name: Rohit Harawade**

**Roll No: A09**

**ID: TU4F2122009**

**Experiment No. 4**

**Aim:** To install and configure Jenkins to build Java or Web Applications.

**Theory:**

*What is Jenkins?*

Jenkins is an open-source automation server that is used to automate various aspects of the software development process. It helps developers streamline the building, testing, and deployment of applications by providing a platform for automating repetitive tasks, integrating different tools, and managing the continuous integration and continuous delivery (CI/CD) pipeline.

**Lab Objective:**

To Integrate and deploy tools like Jenkins and Maven, which is used to build, test and deploy applications in DevOps environment

**Key Features:**

* Continuous Integration: Jenkins supports the practice of continuous integration, where code changes are frequently integrated into a shared repository. It automatically builds and tests the code whenever changes are committed, helping to identify and resolve integration issues early.
* Automated Builds: Jenkins can be configured to automatically build software projects from source code repositories. This includes compiling code, generating executable files, and creating deployable artifacts.
* Testing and Quality Assurance: Jenkins can trigger automated tests, such as unit tests, integration tests, and regression tests, to ensure that code changes don't introduce new bugs or break existing functionality.
* Continuous Delivery/Deployment: Jenkins can be used to automate the deployment process, enabling the continuous delivery or deployment of applications to various environments, such as staging and production, with minimal manual intervention.
* Extensibility: Jenkins has a rich ecosystem of plugins that extend its functionality. These plugins allow integration with various version control systems, build tools, testing frameworks, deployment platforms, and more.
* Pipeline Automation: Jenkins supports the creation of complex build and deployment pipelines using a domain-specific language called "Jenkins Pipeline." Pipelines define the entire software delivery process, including building, testing, deploying, and even manual approval steps.
* Distributed Builds: Jenkins can distribute build and test tasks across multiple machines to speed up the process and efficiently use available resources.

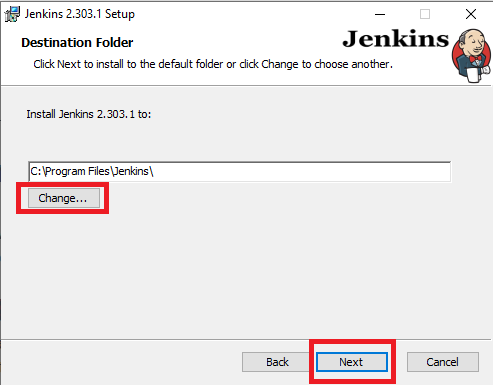
**Setting up Jenkins:**

1. Download the latest Jenkins package for Windows.

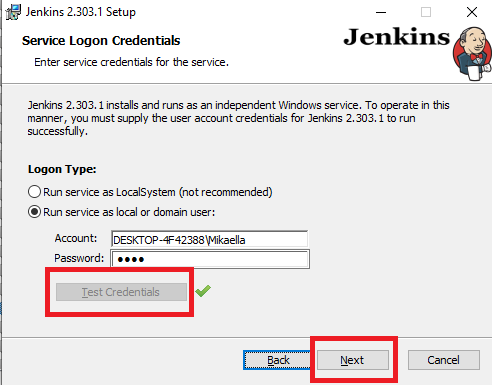
2. Once it is downloaded, it will open a wizard on your screen. Click “Next” to start the Jenkins installation.



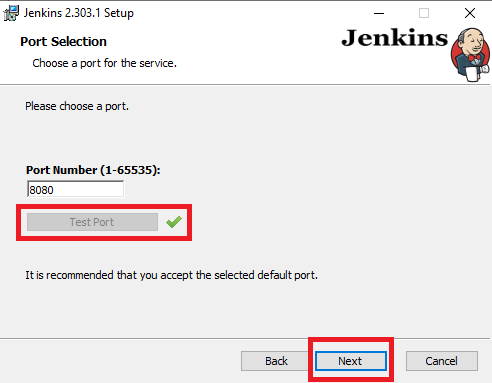
3. Click the “Change…” button if you want to install your Jenkins in another folder. In this example, I will keep it simple and use the default option by clicking on the “Next” button.



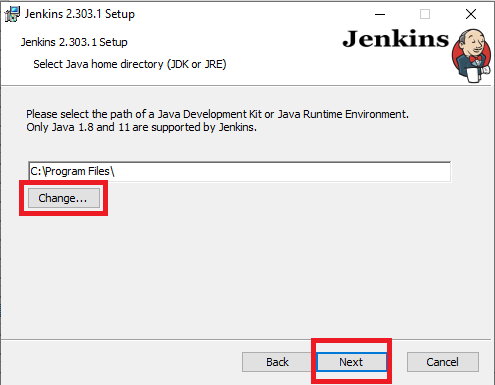
4. Enter the service logon credentials. It is recommended to select the second option “Run service as local or domain user” as it is more secure. To run the Jenkins service using this option, you have to enter the domain username and password. Click on the “Test Credentials” button to test your domain credentials, and you will enable the “Next” button. Then, click on the “Next” button after it appears.



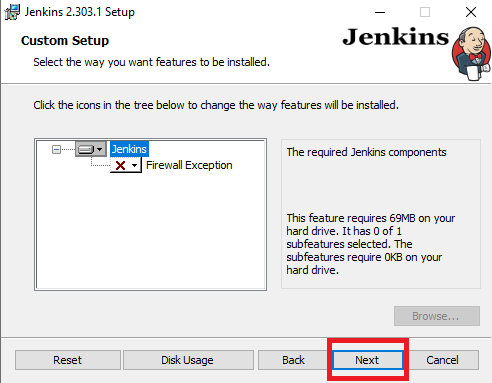
5. Enter the port on which Jenkins will be running. Click on the “Test Port” button to validate if it is free.



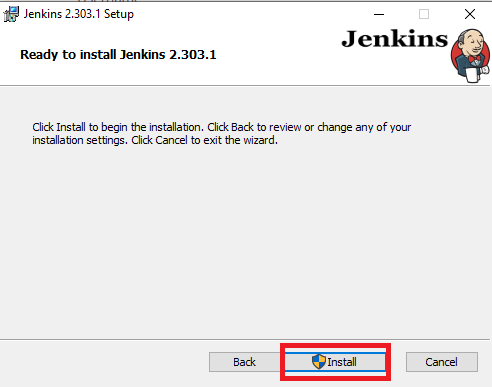
6. Select the Java home directory, and then click on the “Next” button.



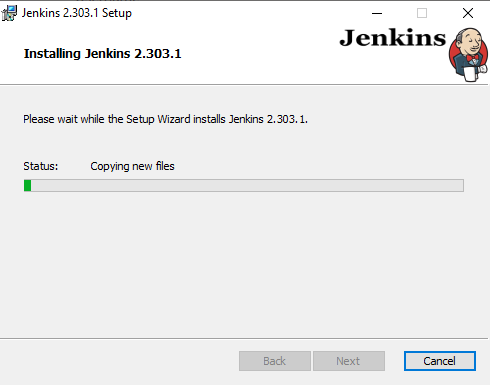
7. Select any other services that need to be installed with Jenkins and click on the “Next” button.



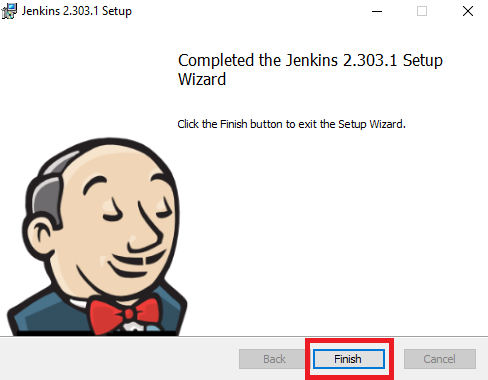
8. Click the “Install” button to start the installation process.



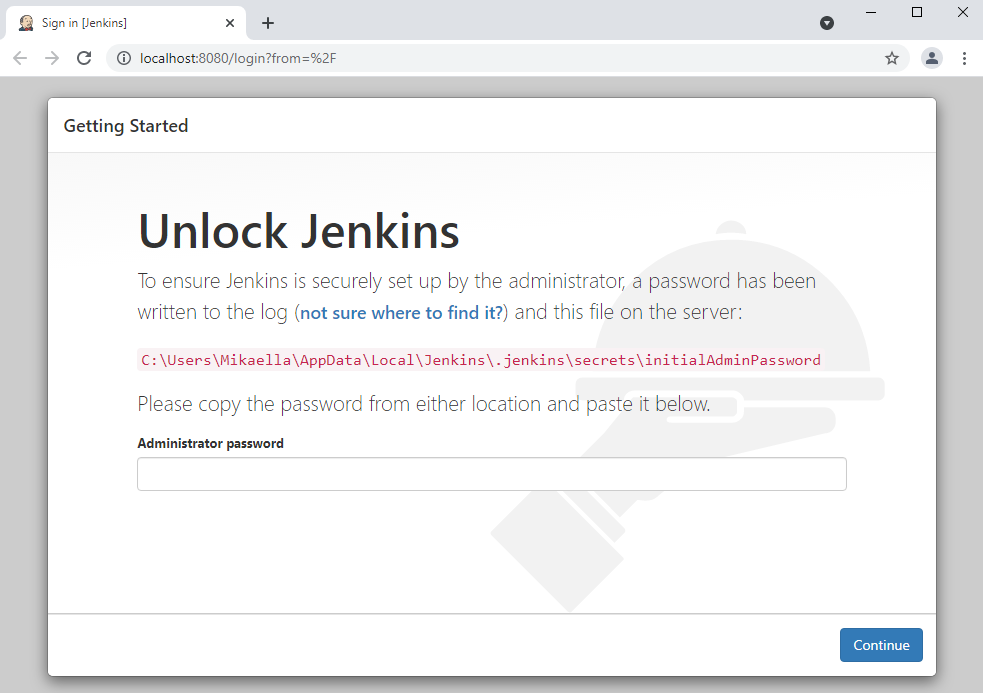
9. The installation will proceed accordingly.



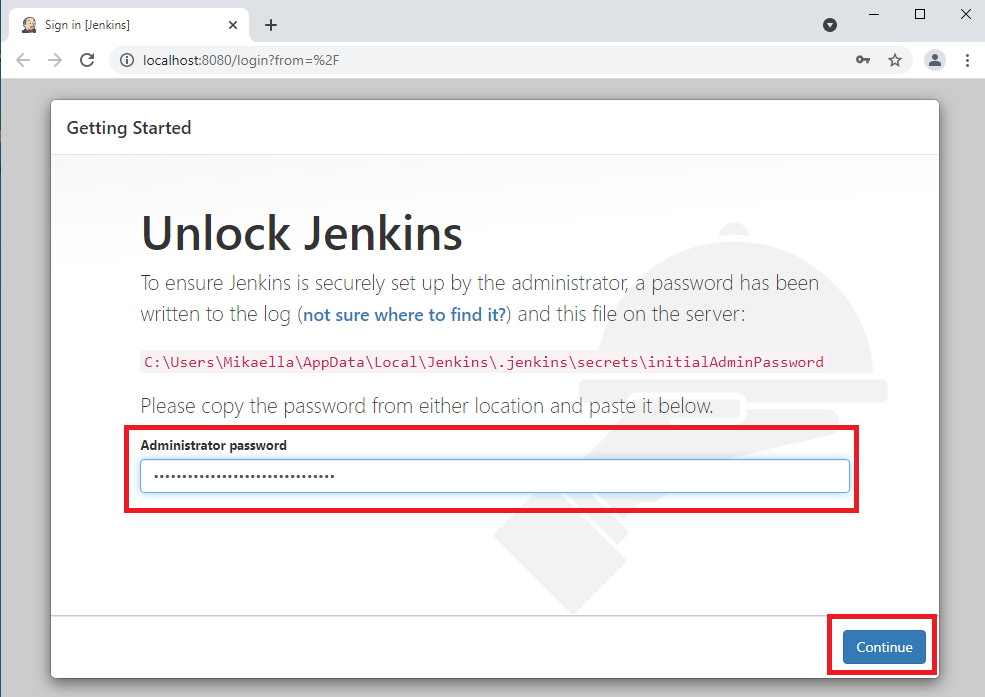
10. When done, click on “Finish” to complete the installation process. Now that the installation process is out of the way, you need to follow a few more quick steps before you start using Jenkins on Windows.



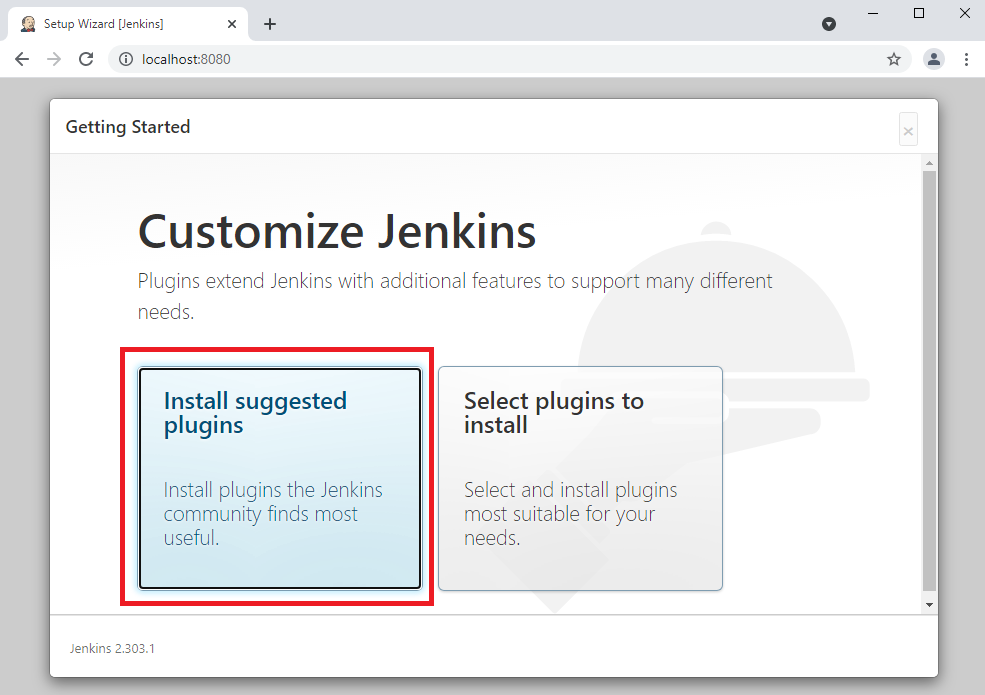
11. You will automatically be redirected to a local Jenkins page, or you can paste the URL http://localhost:8080 in a browser.



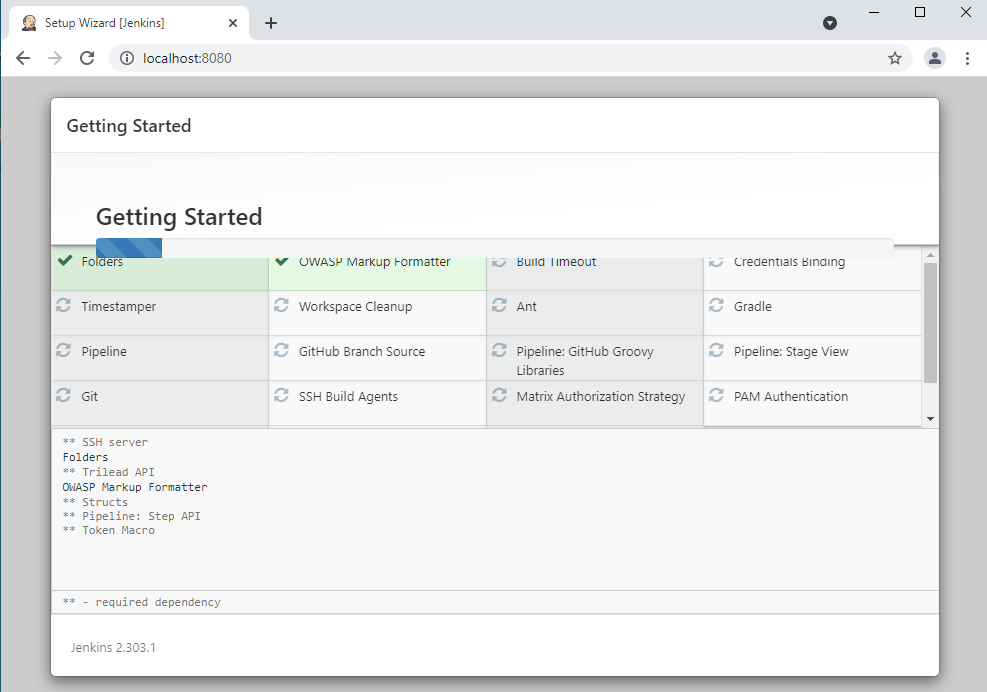
12. To unlock Jenkins, copy the password from the file intialAdminPassword. This file should be found under the Jenkins installation path (set at step 3 in Jenkins installation). If a custom path was entered, then you should check that location. Copy the content of the initialAdminPassword file and paste it into the Administrator password field. Then, click the Continue button.



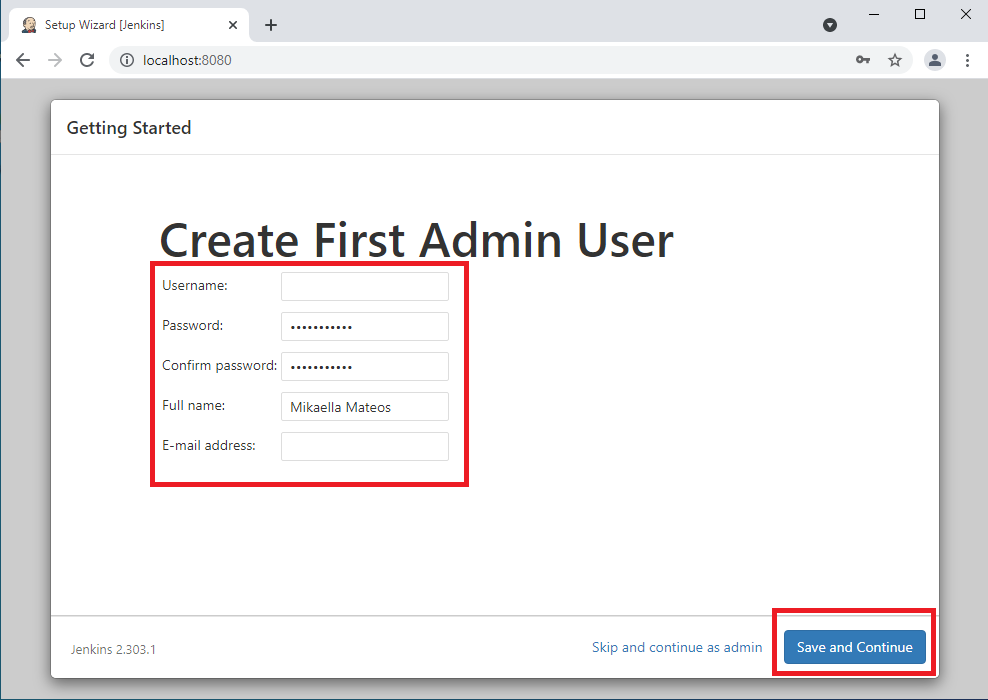
13. You can install either the suggested plugins or selected plugins you choose based on your needs. To keep it simple, we will install the suggested plugins that the Jenkins community finds most useful.



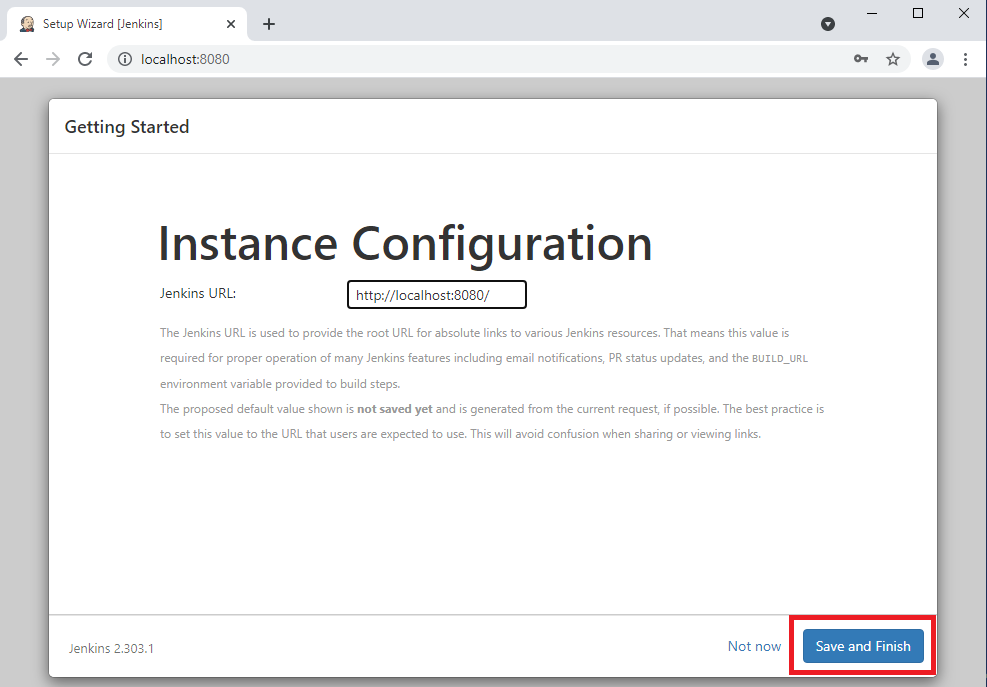
14. Wait until the plugins are completely installed.



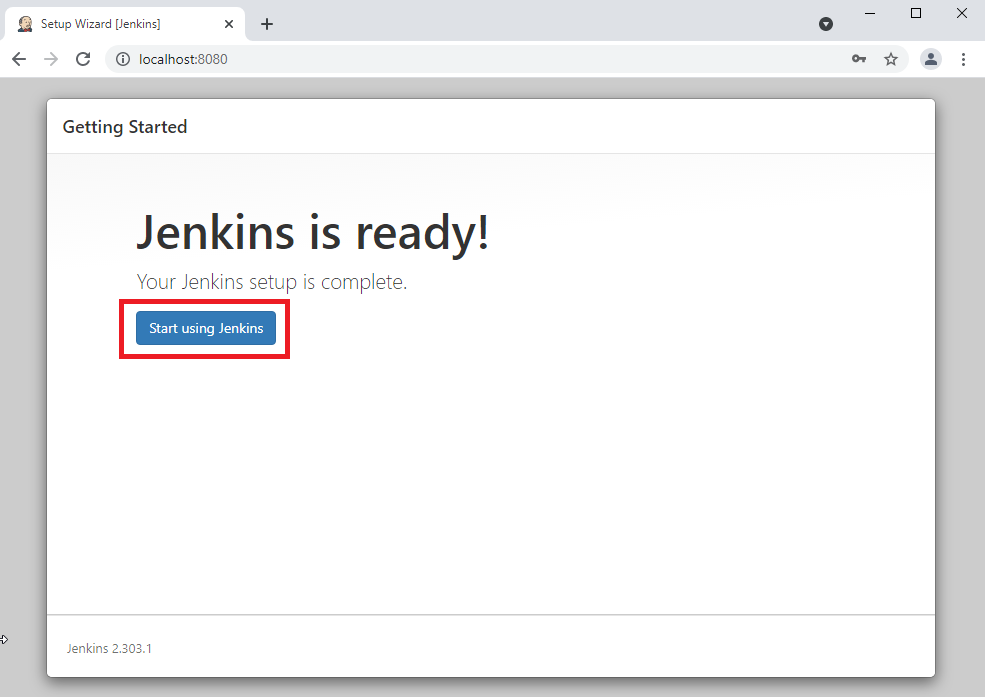
15. The next thing that you should do is create an Admin user for Jenkins. Then, enter your details and click Save and Continue.



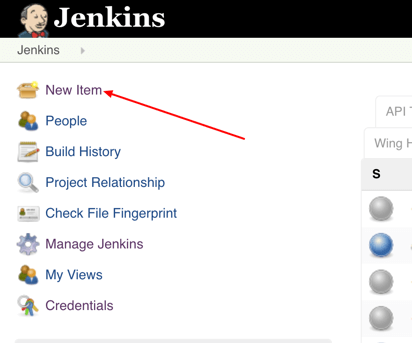
16. Click on Save and Finish to complete the Jenkins installation.



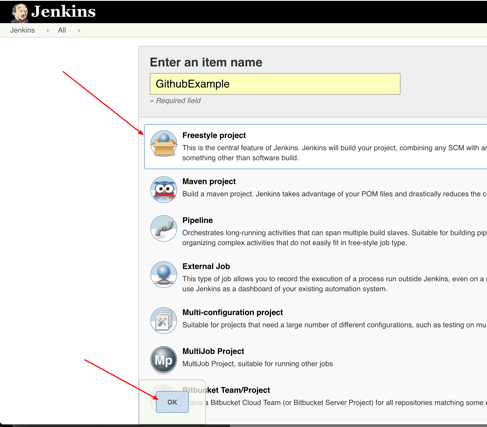
17. Now, click on the "Start using Jenkins" button to start Jenkins.



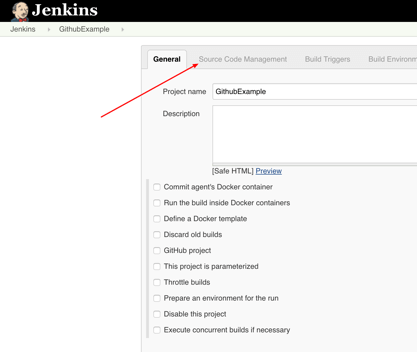
18. In Jenkins, click on ‘New Item’ to create a new project.



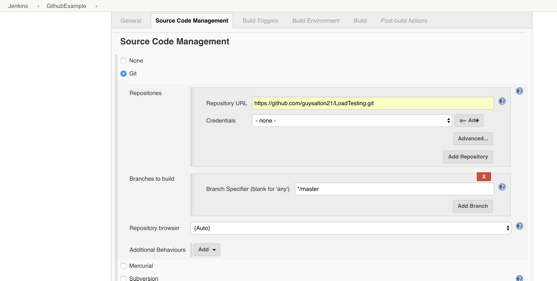
19. Give your project a name, then choose ‘Freestyle project’ and finally, click on ‘OK’.



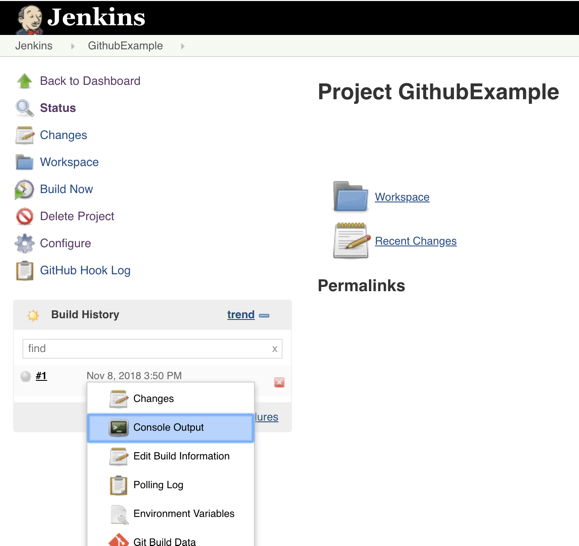
20. Click on the ‘Source Code Management’ tab.



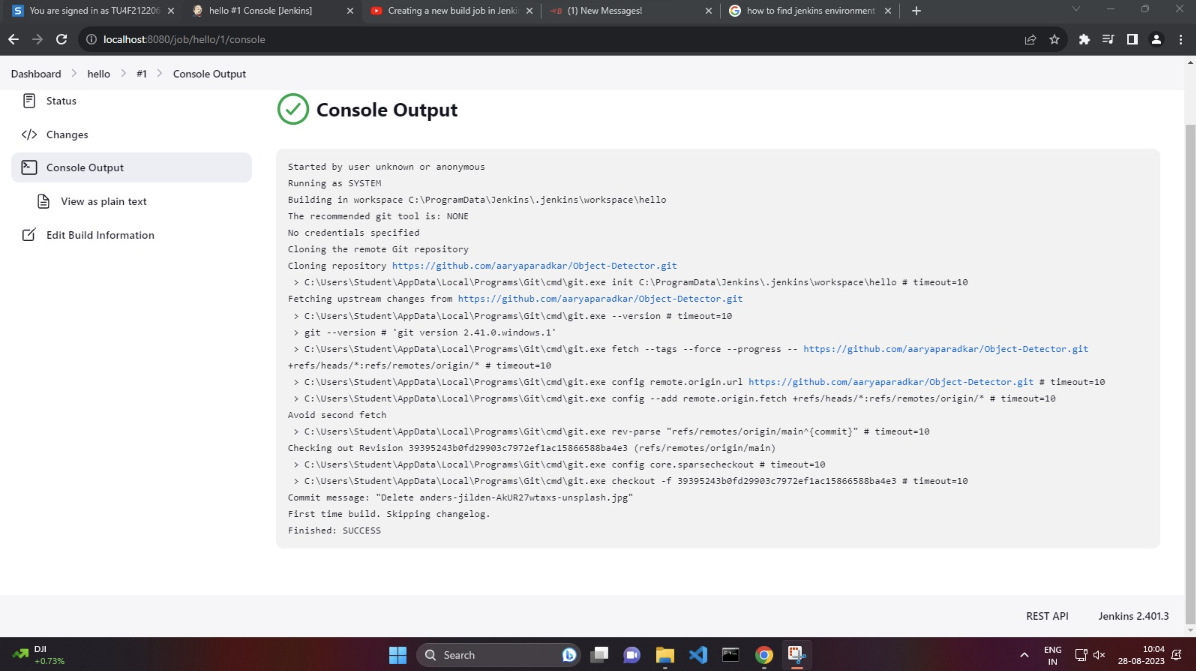
21. Click on Git and paste your GitHub repository URL in the ‘Repository URL’ field.



22. Now, keep all the other settings as default and then save the build. After this click on the “Build Now” option.



22. The console output will be shown.



**Lab Outcome:**

To understand the importance of Jenkins to Build and deploy Software Applications on server environment.

**Conclusion:**

Thus, we have successfully completed the experiment, and understood about the concepts of Jenkins in depth.