3.3 Continuous Integration with Jenkins



This section will guide you to:

* Setup Jenkins in Windows 10
* Setup a .NET Build Project in Jenkins for continuous integration

This guide has three subsections, namely:

3.3.1 Logging into Jenkins dashboard

3.3.2 Setting up a Sample Message in Jenkins for continuous integration

3.3.3 Pushing the code to your GitHub repositories

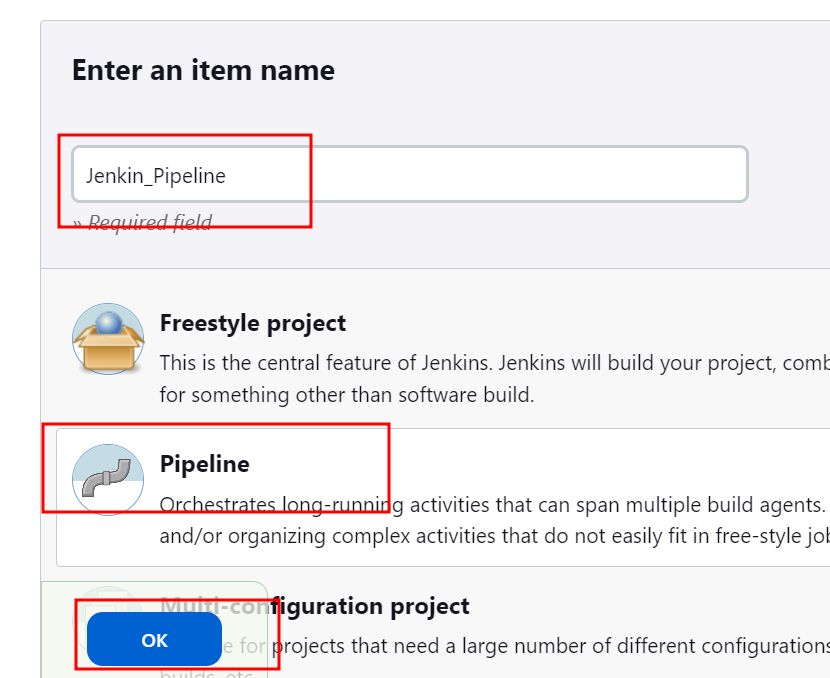
* Jenkins is installed in your practice lab. Refer to **DotNet Lab guide: Phase 4** for more information.

**Step 3.3.1:** Logging into Jenkins dashboard

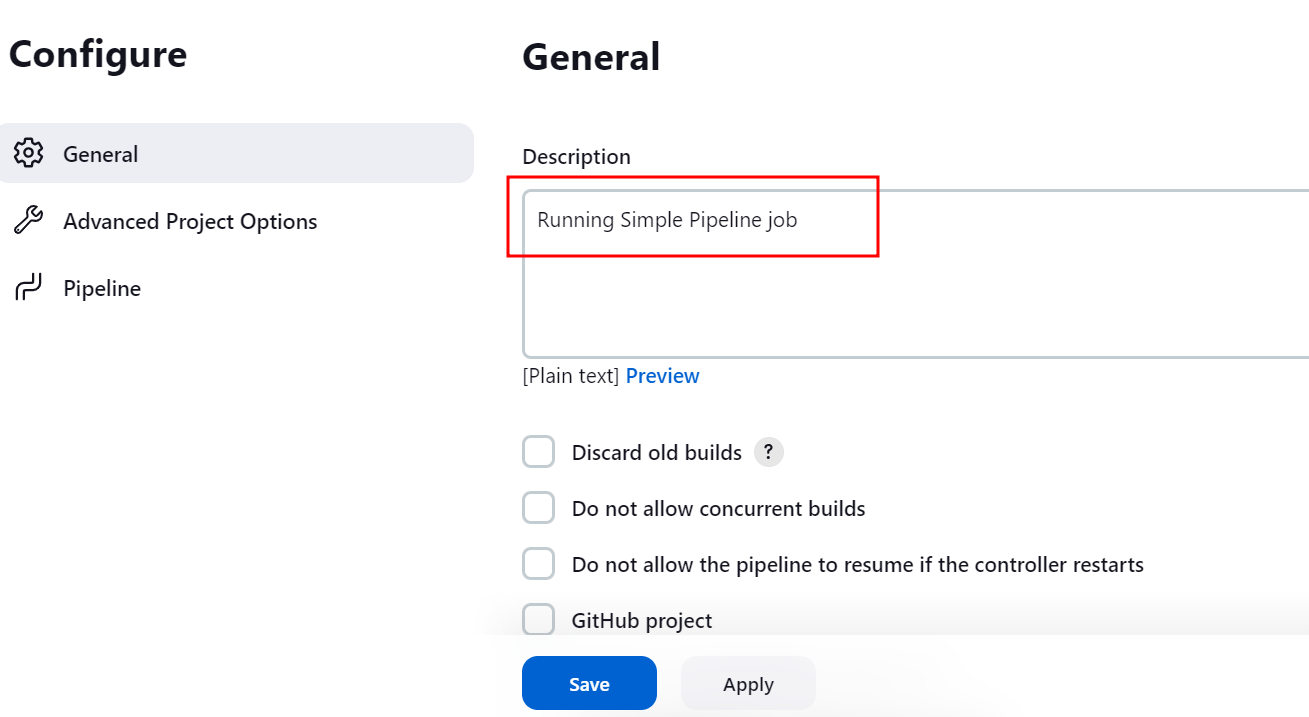
* Open **localhost:8080** in your browser**.**
* Enter your username, password and click on **login.**

**Step 3.3.2:** Setting up a Sample Project in Jenkins for continuous integration

* Open Jenkins dashboard page on <http://localhost:8080>.
* Login as admin user.
* From the left bar, click **Manage Jenkins.**
* Go back to the Dashboard page.
* In the Dashboard page, click on **New Item** from the left bar.



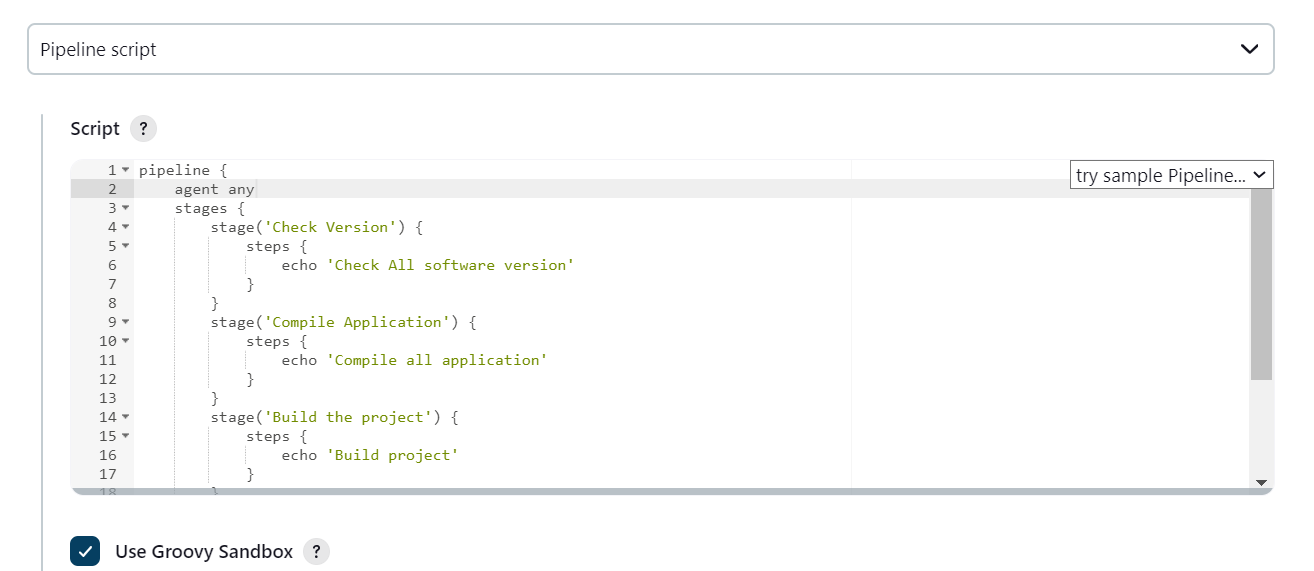
* Enter name as **Ok.**
* This will open the Configuration screen.
* In the **General** tab enter a description of your choice**.**



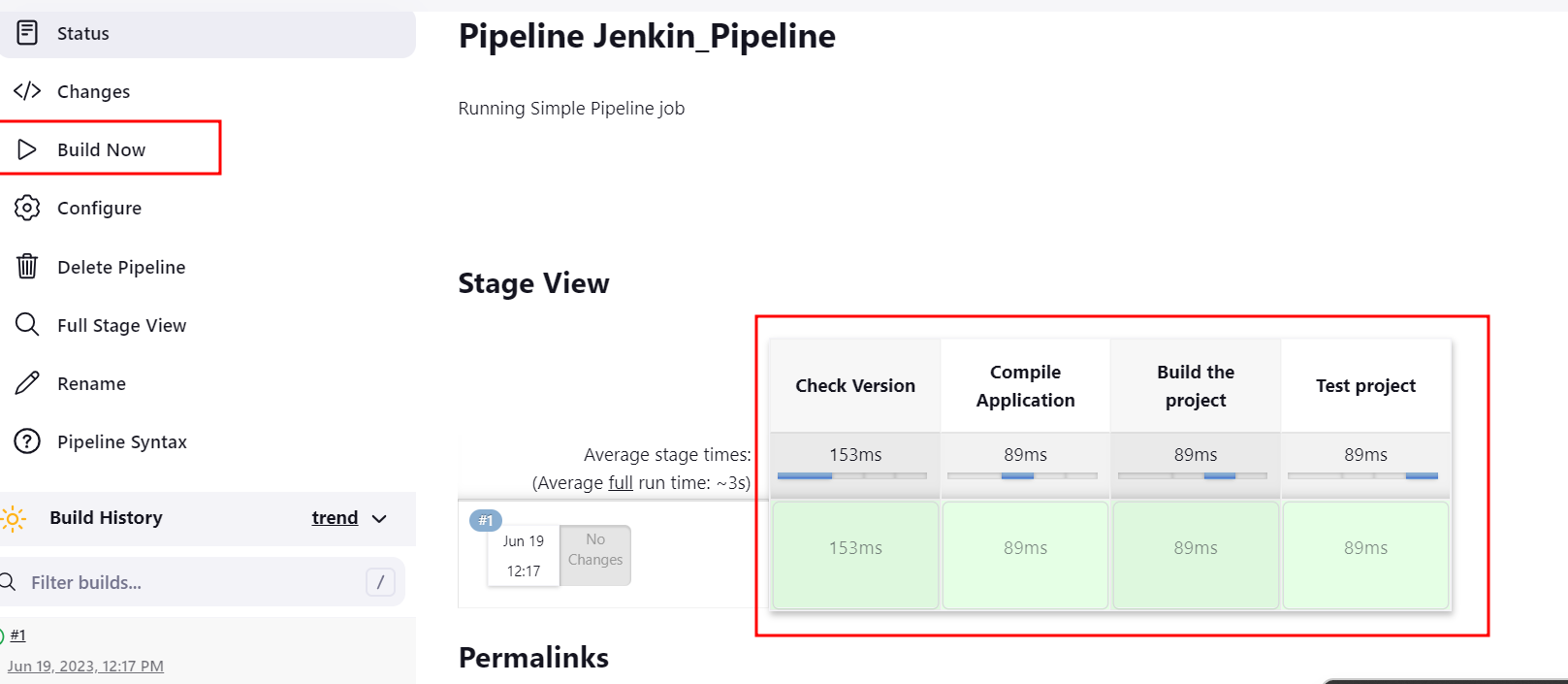
* Now click on pipeline option



* You can edit the stages in place of echo we can execute set of commands.



* Click **Save.**
* From the left bar, click **Build Now** to build the project. This will generate a build in the Jenkins workspace. Once the **Build Number** is displayed click on it to see the workspace.

****

**Step 3.3.3:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add .

Commit the changes using the following command:

git commit -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master