**# Continuous Integration and Continuous Delivery using Jenkins CI tool**

## Table of contents

[1. Installation of Jenkins + Jenkins Configuration + Plugin Management](1.%20Jenkins%20Installation%2C%20Configuration%20and%20Plugins)

[2. Creating pipeline of jobs in Jenkins](2.%20Chain%20of%20Jenkins%20job)

[3. Build Maven Project hosted on GitHub](3.%20Building%20Maven%20Project%20hosted%20on%20GitHub)

[4. BlueOcean Plugin for Pipeline management](4.%20BlueOcean%20Pipeline%20Plugin)

[5. Full Stack Project Management](5.%20Full%20Stack%20using%20Nexus%20Git%20Java%20Maven%20Jenkins)

[6. SonarQube Analysis](6.%20SonarQube%20Analysis)

[7. XAMPP](7.%20XAMPP/src)

[8. PHP MySQL Connection on XAMPP Server](8.%20PHP\_MySQL)

[9. Sessions in PHP](9.%20Sessions%20in%20PHP)

## How to Contribute

All contributions welcomed. Please follow these contribution [guidelines](CONTRIBUTING.md).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Jenkins Installation + Configuration + Plugin Management**

1. Download Generic Jenkins .war file from [Jenkins website](https://jenkins.io/download/)

2. Store \*\*jenkins.war\*\* file on any path you want.

3. Go to the path where jenkins.war file is stored and type following command on command prompt:

```

$ java -jar jenkins.war

```

4. Open \*\*localhost:8080\*\* on the browser.

5. Authenticate Jenkins using authentication key generated on command line.

6. Select \*\*install suggested plugins.\*\*

7. Signup with administrator account.

## Note

If Jenkins does not executes successfully and opens up after step 3, try changing the port using the following command:

```

$ java -jar jenkins.war --httpPort=<portnumber>

```

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Creating Chain of Jenkins job**

1. Create multiple jobs doing some tasks (Example: Executing Windows Batch command)

2. Connect each task with build trigger by successful build of previous task in pipeline.

3. Build Initial Task.

4. Download Delivery Pipeline Plugin.

5. Create new view -> Delivery Pipeline view -> Add componenent -> Initial Job == First Task.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Using Jenkins to build a Maven project hosted on GitHub**

1. Create new item -> Free Style project.

2. In Source Code Management, choose Git and specify URL of the repository where the Maven project is stored.

3. In build triggers, specify the path of root POM.xml file in \*\*Trigger builds remotely via scripts.\*\*

4. In Build commands, specify Windows Batch command execution and write Maven commands:

```

$ mvn clean

$ mvn compile

$ mvn install

```

5. Close project and Build it.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# BlueOcean Pipeline Plugin**

1. Create new pipeline.

2. Use GitHub for hosting.

3. Select account on Git if you have multiple accounts.

4. Select the repository where you want to create the pipeline.

5. Store the Jenkinsfile on the specified repository.

6. Create the pipeline.

7. Build the pipeline and see the results in console output for each stage.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Full Stack Project**

```

Programming Language: Java

Unit Testing: JUnit

Source Code Management: GitHub for Git

Build Tool: Apache Maven

CI and CD: Jenkins

Artifact Repository: Nexus

```

## Start Nexus - In bin folder

```

$ nexus.exe /run

```

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Static Code Analysis using SonarQube :smile:**

Check my blog for details of SonarQube at [this link](https://iq.opengenus.org/sonarqube-for-code-coverage-analysis/)

## Syntax:

```

$ mvn clean install sonar:sonar -Dsonar.host.url=http://localhost:9000 -Dsonar.analysis.mode=publish

```

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# XAMPP**

\* \*\*:X\*\* Works on both W (Windows) and L (Linux)

\* \*\*A:\*\* Apache Server

\* \*\*M:\*\* MySQL

\* \*\*P:\*\* PHP Scripting Language

\* \*\*P:\*\* Pearl Language

## Task

1. Create an HTML form and write its PHP Script to perform Simple Interest calculation.

1. Create an HTML form and write its PHP Script to perform Compound Interest calculation.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Using XAMPP server to demonstrate connection of HTML and PHP with MySQL Database**

**## Tasks**

1. Create a Database in MySQL (localhost/phpmyadmin) and create a table to store student details.

2. Create a Registration HTML and PHP page for insertion of student record in Table.

3. Create a select HTML and PHP page to select all rows one by one from the Database.

4. Create a login module using SAP ID and Name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**# Session in PHP**

Session are used to manage inputs from one or more HTML pages using PHP Script.

In this activity, three HTML pages are used to take inputs of a student and created a chain of them using PHP.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_