Logo

Description automatically generated with low confidence Logo

Description automatically generated

**Devops Lab**

Pre-requisite for the lab

JDK – 8/11 should be installed - Find steps at the end of the document.

Maven – latest version should be installed - Find steps at the end of the document.

**Lab1**

Use Case: Jenkins Install and Configure

**Windows Platform**

Follow the instruction from the following link - <https://www.jenkins.io/doc/book/installing/windows/>

**Note: Step 3 from the above instruction, Logon Type should be *“Run system as localsystem”***

**Linux Platform**

Follow the instruction from the following link - <https://www.jenkins.io/doc/book/installing/linux/#debianubuntu>

For Ubuntu –

Graphical user interface, text, application, email

Description automatically generated

**Note: Execute the following command first before installing any software**

* sudo apt-get update
* sudo apt-get upgrade -y

**Lab2**

Use case: Jenkins-Gitlab with Java Maven Project

Let’s assume a scenario — when a developer completed its coding part on its local machine. To keep versioning of source code he/she committed code to some VCS (let say Gitlab). At the end of the day developer want to see the running integrated code on Dev Server situated somewhere, then developer must manually build his code and put it to Dev server.

For above case we will use the automated solution. Developer commits code to Gitlab then Jenkins will automatically pull the code from Gitlab and automatically builds the project.

Make sure that Jenkins is properly setup and Maven project is on Gitlab.

**Step 1: Configure Jenkins**

1. **Insure that Gitlab Plugin is installed under- Manage Jenkins > Manage Plugins > Installed search for git. If not installed move to Available Tab and search for git and install it.**
2. **Configure Java, Git and Maven for Jenkins**
3. **Navigate to Manage Jenkins > Global Tool Configuration > Under JDK section provide Name and path to JAVA\_HOME, in same way for Git provide Git Name and path to Git executable, same in case of Maven provide Name and MAVEN\_HOME as in below images**

**Graphical user interface, text, application, email

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

**Step 2: Create Job**

1. **Create a new Job by clicking New Item**
2. **Enter your Job Name ‘Jenkins-GitHub’ and select Maven Project then click OK, You will be navigated to configure the Job.**

**Graphical user interface, text, application

Description automatically generated**

**Note:** **If Maven Project is not visible in your Jenkins then install plugin named as ‘Maven Integration’**

**Graphical user interface, text, application

Description automatically generated**

1. **Under Source Code Management section click on Git radio button and provide Repository URL- https://gitlab.com/123Balu42/Jenkins\_Spring.git**
2. **Select branch to build if master- ‘\*/master’ if development- ‘\*/development’**
3. **You can add behavior by selecting value from drop down what to perform in my case I want to clean space before code checkout so I have selected Clean before chekout (you can configure according to your need this is optional)**
4. **Navigate to Build Trigger- If you want to build your project on specific time interval you can configure it under Build Trigger > tick Poll SCM and in text field \* \* \* \* \* (five star separated by space for every minute, see bottom of this article for convention). If you will not set up Poll you will have to manually build the Job.**
5. **Navigate to Build Section > provide path to pom.xml (for maven project) and Goals and OptionsGraphical user interface, text, application, email

   Description automatically generated**
6. **Navigate to that project. You will have to click on Build Now if you have not configured Poll SCM else it will automatically trigger every minute (as we have set \* \* \* \* \*). Your project will build successfully.**
7. **Graphical user interface, text, application

   Description automatically generated**

**By clicking on blue circle you will be able to see logs**

**By Clicking #1 (#<Build Number>) you will be navigation to below window showing various project related information.**

**Graphical user interface, text, application

Description automatically generated**

**Note For Reference:**

**Poll SCM:**

**(\* \* \* \* \* ) represents (minute, hour, day\_of\_month, week, day\_of\_week)**

**range (0–59, 0–23, 1–31, 1–12, 0–7)**

**sunday -0, monday-1, tuesday-2 ,…,saturday-6**

**January-1, Feb-2, …,Dec-12**

**Example :**

**/5 \* \* \* \* — every 15 min**

**0 \* \* \* \* — every hour**

**0 0 \* \* \* — every 00:00 hours**

**00 14 10 3,6,9,12 \* — at 2:00 PM on 10th of March, June, September and December**

**25,50 1 15 \* 2 — at 1:25 AM, 1:50 AM every Tuesday and on 15th of every month.**

**00 21 \* \* Mon,Wed,Fri — at 9.00 PM every Monday, Wednesday, Friday**

**\*/5 \* \* \* 1–5 — every 5 minutes during the 5 working days (Monday — Friday), every week and month**

**\* \*/4 \* \* Sun — every minute past every 4th hour on Sunday**