

Name – Rachit Shrivastava

Sap Id – 500119571

Batch 3 – DevOps

Lab Exercise 7

Integrating Maven with Jenkins

Objective: To install the Maven plugin in Jenkins for smooth integration and automation of Maven-based build processes within the Jenkins environment

Tools required: Git, GitHub, and Jenkins

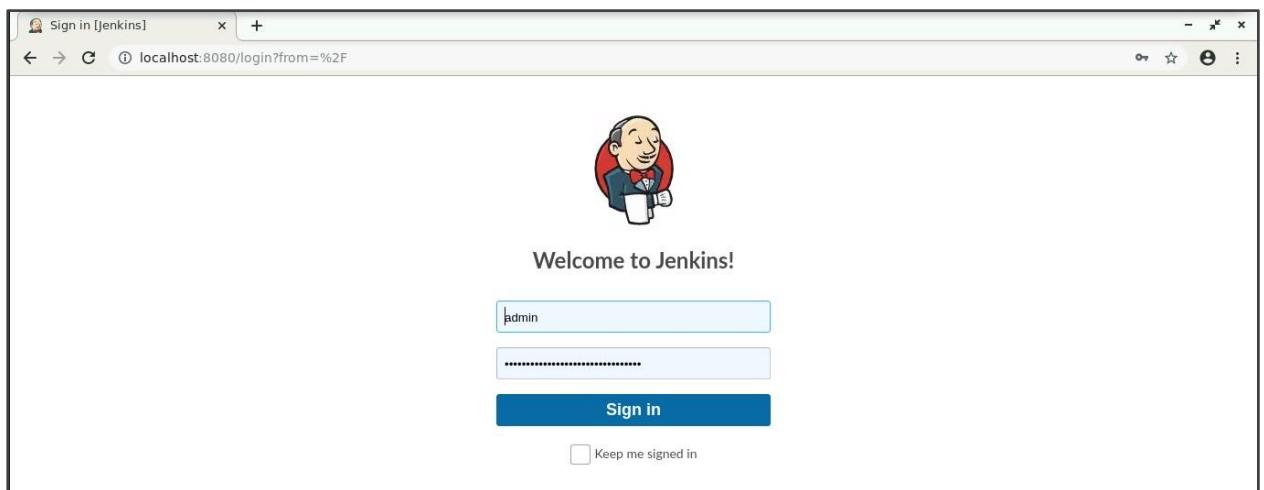
Prerequisites: None

Steps to be followed:

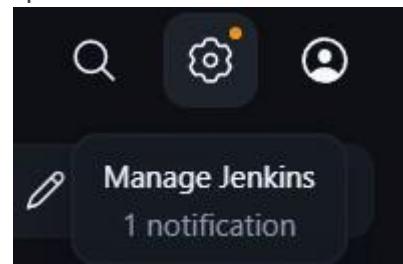
1. Install the Maven plugin
2. Set up Global Tool Configuration
3. Fork a sample repository
4. Integrate Maven with Jenkins

Step 1: Install the Maven plugin

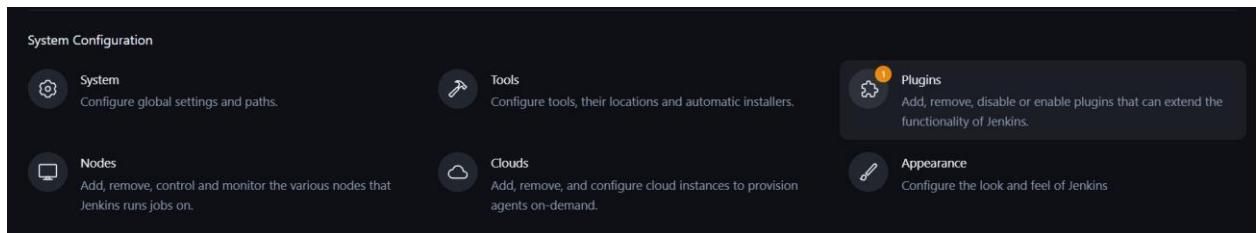
- 1.1 Open the browser, go to the Jenkins Dashboard by typing **localhost:8080** in your browser, provide the credentials, and click the **Sign in** button



1.2 Click on the **Manage Jenkins** option as shown in the screenshot below:



1.3 Click on the **Plugins** option as shown in the screenshot below:



1.4 Click on **Installed plugins** to verify whether the **Maven Integration plugin** has been installed

The screenshot shows the Jenkins Plugins page. At the top, there is a search bar with the placeholder "Name ↴" and two buttons: "Health" and "Enabled". Below the search bar, a list of plugins is displayed. The first item in the list is "Maven Integration plugin 3.27". The description for this plugin states: "This plugin provides a deep integration between Jenkins and Maven. It adds support for automatic triggers between projects depending on SNAPSHOTs as well as the automated configuration of various Jenkins publishers such as Junit." To the right of the description are three icons: a green circle with the number "100", a blue toggle switch with a checkmark, and a red close button.

Note: Maven is already installed in your practice lab environment. If not, click on **Available plugins**, search for the Maven Integration plugin, and install it.

1.5 Use the following command to check the Maven version:

```
mvn -version
```

```
$ mvn -version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcdc97d260186937)
Maven home: C:\maven\mvn
Java version: 17.0.15, vendor: Microsoft, runtime: C:\Program Files\Microsoft\jdk-17.0.15.6-hotspot
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

Step 2: Set up Global Tool Configuration

2.1 Go to the Jenkins Dashboard, click on **Manage Jenkins**, and then select **Tools** from the list of options

The screenshot shows the Jenkins System Configuration page. At the top, there is a title "System Configuration". Below the title, there are several sections: "System" (Configure global settings and paths), "Tools" (Configure tools, their locations and automatic installers, highlighted in a blue box), "Nodes" (Add, remove, control and monitor the various nodes that Jenkins runs jobs on), "Clouds" (Add, remove, and configure cloud instances to provision agents on-demand), "Plugins" (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), and "Appearance" (Configure the look and feel of Jenkins).

2.2 To configure Maven, click on the **Maven installations** button in the Maven section and enter a **Name** and **MAVEN_HOME** path

