Lab Exercise # 2 - CICD setup - Install Jenkins in One instance and Tomcat on second EC2 instance in AWS Cloud

**🔧 1. Java**

**✅ What is it?**

Java is a **programming language and platform** used to build applications. It is **required to run many Java-based tools**, including Maven and Jenkins.

**💡 Why is it important?**

* Maven is written in Java.
* Jenkins is also written in Java.
* Many enterprise applications (like Spring, Tomcat apps, etc.) are Java-based.

**➕ You install Java first because:**

**Maven and Jenkins need Java to work.**

**📦 2. Maven**

**✅ What is it?**

Maven is a **build automation and dependency management tool** used primarily for Java projects.

**💡 What it does:**

* Compiles Java code
* Downloads project dependencies (like external libraries)
* Packages the app (e.g., into a .jar or .war file)
* Runs unit tests

**➕ Maven needs Java to run:**

It uses the **Java Development Kit (JDK)** to compile and build projects.

**⚙️ 3. Jenkins**

**✅ What is it?**

Jenkins is an **open-source automation server** used for **continuous integration and continuous delivery (CI/CD)**.

**💡 What it does:**

* Automates building, testing, and deploying applications.
* Integrates with version control tools (like Git).
* Executes scripts like Maven commands (mvn clean install, etc.).
* Can trigger builds when new code is pushed to GitHub.

**➕ Jenkins also requires Java to run.**

**🔄 How They Work Together**

**🔁 Workflow Connection:**

[Developer writes Java code]

⬇️

[Code is committed to GitHub]

⬇️

[Jenkins detects new commit]

⬇️

[Jenkins runs a build job]

⬇️

[Jenkins executes Maven commands]

⬇️

[Maven compiles Java, runs tests, builds WAR file]

⬇️

[Jenkins deploys it (e.g., to Tomcat)]

**✅ Installation Order (with reason)**

1. **Java**
   * Required to run Maven and Jenkins.
2. **Maven**
   * Used to build Java projects. Jenkins will call Maven commands.
3. **Jenkins**
   * Controls the entire build pipeline. Automates and monitors everything.

**🔗 In Summary**

| **Tool** | **Role** | **Depends On** | **Purpose** |
| --- | --- | --- | --- |
| **Java** | Runtime & Compiler | — | Needed by Maven & Jenkins |
| **Maven** | Build Tool | Java | Builds Java projects |
| **Jenkins** | CI/CD Automation | Java, Maven (for Java projects) | Ru |

**🌐 Connect to Tomcat EC2 Instance for Installing Tomcat**

This step is part of your infrastructure setup in the DevOps pipeline. Here's what it means and how it fits into the big picture.

**💡 What is Apache Tomcat?**

Apache Tomcat is an **open-source Java servlet container** — in simple terms, it is a **web server for running Java web applications**, especially .war files generated by Maven builds.

**🔧 Why Connect to the EC2 Instance?**

When you launch an EC2 instance in AWS to host Tomcat, it’s just a **virtual machine running Linux (Ubuntu)**. You need to:

1. **SSH into that instance** (i.e., connect remotely via terminal).
2. **Install Tomcat manually** on the instance.
3. **Deploy your .war file** (from your Maven build, via Jenkins or manually).

**✅ Prerequisites to Connect:**

* You have an EC2 instance (Ubuntu) running.
* Port **8080** is open in the **security group** (because Tomcat uses port 8080).
* You have a **.pem key file** to SSH into the instance.

**🔗 Steps in the Workflow:**

1. **Provision EC2 Instance (Tomcat Server)**
   * OS: Ubuntu 22.04
   * Instance Type: t2.small
2. **Open Terminal / Git Bash**
   * Navigate to where the .pem key is saved.
3. **SSH into the instance**

ssh -i "your-key.pem" ubuntu@<public-ip>

1. **Install Java (Tomcat needs Java)**

sudo apt update

sudo apt install default-jdk -y

1. **Download and Install Apache Tomcat**

wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.80/bin/apache-tomcat-9.0.80.tar.gz

tar -xvzf apache-tomcat-9.0.80.tar.gz

1. **Start Tomcat**

cd apache-tomcat-9.0.80/bin

./startup.sh

1. **Access Tomcat in Browser**

http://<your-ec2-public-ip>:8080

**🔄 How It Connects with Java, Maven, and Jenkins**

* **Java**: Tomcat runs Java web applications, so Java must be installed.
* **Maven**: Builds the .war file that will be deployed to Tomcat.
* **Jenkins**: Can be used to automate the deployment of the .war file to the Tomcat server.