**💻 Lab Exercise 03**

**Create Java Web App using Maven & Setup Java WebApp in GitHub Repo**

**🧩 Project Objective:**

This lab project walks through the process of creating a **Java Web Application using Maven**, configuring **SSH access**, and pushing the code to a **GitHub repository** from an **AWS EC2 instance** (typically used for Jenkins setup). This forms the foundation for DevOps practices like **CI/CD with Jenkins** and **deployment on Tomcat**.

**🔧 Prerequisites:**

* GitHub account
* Git client installed
* AWS EC2 instance (Jenkins pre-installed is ideal)
* Maven installed
* Internet access from EC2 instance

**✅ Step-by-Step Guide**

**🔐 Step 1 – Create a Private GitHub Repository**

1. Login to GitHub
2. Click on **New** to create a repository.
3. Name your repo (e.g., JavaWebApp)
4. Choose **Private** and initialize with a README.

**🔐 Step 2 – Generate SSH Keys on EC2**

1. SSH into your Jenkins EC2 instance.
2. Run the following command to generate SSH keys:

bash

CopyEdit

ssh-keygen

(Press Enter 4 times to skip password setup)

1. View the public key:

bash

CopyEdit

sudo cat ~/.ssh/id\_rsa.pub

**🔐 Step 3 – Add SSH Key to GitHub**

1. Go to GitHub → **Settings**
2. Navigate to **SSH and GPG keys**
3. Click **New SSH Key**
4. Paste the copied public key
5. Click **Add SSH Key**

**📦 Step 4 – Clone Repository Locally on EC2**

1. Copy the SSH URL from your repo (git@github.com:username/repo.git)
2. Run:

bash

CopyEdit

git clone <paste-ssh-url>

1. Move into the repo folder:

bash

CopyEdit

cd JavaWebApp

ls -al

**🌐 Step 5 – Create Java Web App using Maven**

Use Maven to scaffold a Java Web App:

bash

CopyEdit

mvn archetype:generate -DgroupId=com.example.webapp \

-DartifactId=my-webapp \

-DarchetypeArtifactId=maven-archetype-webapp \

-DinteractiveMode=false

**⬆️ Step 6 – Push Java Web App to GitHub**

1. Move into the newly created directory if needed:

bash

CopyEdit

cd my-webapp

1. Add all new files:

bash

CopyEdit

git add \*

1. Configure Git identity:

bash

CopyEdit

git config --global user.name "your-github-username"

git config --global user.email "your-email@example.com"

1. Commit changes:

bash

CopyEdit

git commit -m "Initial commit of Java web app"

1. Push to GitHub:

bash

CopyEdit

git push

**🔍 Step 7 – Verify on GitHub**

1. Go back to your GitHub repo.
2. You should now see your Java web project files.
3. The commit message should also appear.