Experiment 6 - Continuous Integration with Jenkins: Setting Up a CI Pipeline, Integrating Jenkins with Maven/Gradle, Running Automated Builds and Tests

Objective

To set up a Continuous Integration (CI) pipeline in Jenkins, integrate it with Git, and run Selenium Java tests using Maven.

Prerequisites

Before proceeding, ensure the following:

- ✅ **Jenkins is installed and running** on your system. If not, refer to [Experiment 5].
- ✅ **Git is installed** and configured in Jenkins. (Verify with git --version).
- ✅ **Maven is installed** and configured in Jenkins. (Check with mvn -version).
- ✅ **Selenium Maven Project is ready** with test cases (src/test/java).
- ✅ **The project is stored in two places**:
  - Locally on your system (e.g., D:\Idea Projects\MVNGRDLDEMO).
  - Pushed to **GitHub** with a valid repository link.
- ✅ **Jenkins has access to the GitHub repository** (via credentials).

1. Configuring Jenkins & Git Integration

Step 1: Verify Git Installation in Jenkins

1. Open Jenkins Dashboard → Manage Jenkins → Global Tool Configuration.
2. Under Git, verify the installation path (e.g., C:\Program Files\Git\bin\git.exe).
3. Click Save.

Step 2: Add GitHub Credentials in Jenkins

1. Navigate to Manage Jenkins → Manage Credentials.
2. Select Global credentials (unrestricted) → Click Add Credentials.
3. Choose Username with password or SSH Key, provide details, and click OK.

2. Running a Selenium Java Test from a Local Maven Project

Step 1: Create a New Jenkins Job

1. Go to Jenkins Dashboard → Click New Item.
2. Enter a project name → Select Freestyle Project.
3. Click OK.

Step 2: Configure the Build Step

1. Scroll to Build → Click Add build step → Execute Windows Batch Command.
2. Enter the following commands (ensure correct navigation to project directory):

```
cd D:\Idea Projects\MVNGRDLDEMO
mvn test

```

3.Click **Save** → Click **Build Now** to execute the test.

3. Running Selenium Tests from a GitHub Repository via Jenkins

Step 1: Set Up a New Jenkins Job for GitHub Project

1. Go to Jenkins Dashboard → Click New Item.
2. Enter a project name → Select Freestyle Project.
3. Click OK.

Step 2: Configure Git Repository in Jenkins

1. Under Source Code Management, select Git.
2. Enter your GitHub repository URL (e.g., https://github.com/your-repo-name.git).
3. Select the Git credentials configured earlier.

Step 3: Add Build Step for Maven

1. Scroll to Build → Click Add build step → Execute Windows Batch Command.
2. Enter the Maven test command:

```
mvn test

```

3. Click **Save**.

Step 4: Trigger the Build

1. Click **Build Now** to fetch the code from GitHub and execute the Selenium tests.
2. Check the **Console Output** to verify test execution.

Basic Git Commands

Step 1: Create a new folder and open Git Bash/Cmd and go to the folder location

Step 2: Run these commands for git configuration

```
git config --global user.email “sampleGitHub@email.com “
git config --global user.name “sampleGitHub_username “
```

Step 3: Initialize git using this command

```
git init
```

Step 4: Add some sample files in the folder by using this command:

```
git touch <filename1.txt>
git touch <filename2.html>
git touch <filename3.py>
git touch <filename4.js>
```

Step 5: Run these commands to check status, add files, and commit your changes/updates

```
git status                          [ To check status]
git add <filename>                  [ To add a particular file]
git add .                           [ To add all the files]
git commit -m “Commit Message “     [ To commit all your changes]
git remote add origin “github-url “ [ To add remote repo link to local repo]
git push -u origin master           [ To push your local changes to the remote]
git pull origin master –rebase      [ To pull from remote]
```

To add team members to GitHub, the steps depend on whether you're working in:

1. A personal repository, or
2. An organization (best for team collaboration)

Let’s look at both cases:

✅ If You’re Using a Personal GitHub Repository:

You can’t create teams, but you can invite collaborators.

Steps:

1. Go to your repository on GitHub.
2. Click on Settings (top menu).
3. In the left sidebar, click Collaborators (under "Access").
4. Click "Add people".
5. Enter their GitHub username or email and select them.
6. Choose the permission level (Read / Write / Admin).
7. Click Add.

They’ll get an invitation to collaborate.

✅ If You’re in a GitHub Organization:

Organizations let you manage teams and permissions more efficiently.

Step 1: Create or Go to the Organization

1. Visit: https://github.com/organizations/new to create one.
2. Or go to your existing organization.

Step 2: Add Members to Organization

1. Go to the organization page.
2. Click "People" → "Invite member".
3. Enter GitHub usernames.
4. Choose role (Member / Owner).
5. Optionally assign to teams.

6. Click Send Invitation.

📌 Step 3: Create a Team

1. Go to your organization.
2. Click "Teams" > New team.
3. Give your team a name and description.
4. Set repo access (you can assign specific repos to the team).
5. Add members.

```
Sharath@e401473 MINGW64 /f
$ mkdir gitproject1

Sharath@e401473 MINGW64 /f
$ cd gitproject1

Sharath@e401473 MINGW64 /f/gitproject1
$ vi s2.sh

Sharath@e401473 MINGW64 /f/gitproject1
$ ls
s1.txt.txt  s2.sh

Sharath@e401473 MINGW64 /f/gitproject1
$ ls
s1.txt  s2.sh

Sharath@e401473 MINGW64 /f/gitproject1
$ git config --global uesr.name "sharath2707"

Sharath@e401473 MINGW64 /f/gitproject1
$ git config --global user.email "sharathbabu.sura@gmail.com"

Sharath@e401473 MINGW64 /f/gitproject1
$ git init
Initialized empty Git repository in F:/gitproject1/.git/
```

```
Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        s1.txt
        s2.sh

nothing added to commit but untracked files present (use "git add" to track)

Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git add .
warning: in the working copy of 's2.sh', LF will be replaced by CRLF the next time Git touches it

Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git commit -m "committed message"
[master (root-commit) c84cdfa] committed message
 2 files changed, 1 insertion(+)
 create mode 100644 s1.txt
 create mode 100644 s2.sh
```

```
Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git remote add origin https://github.com/sharath2707/repo1.git

Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git push -u origin master
remote: Permission to sharath2707/repo1.git denied to sharath2707.
fatal: unable to access 'https://github.com/sharath2707/repo1.git/': The requested URL returned error: 40
3
```

```
Sharath@e401473 MINGW64 /f/gitproject1 (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 261 bytes | 261.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sharath2707/repo1.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Sharath@e401473 MINGW64 /f/gitproject1 (master)
$
```