

EXPERIMENT-05

AIM: Demonstrate Continuous Integration and Development Using Jenkins

Objective: To demonstrate continuous integration and continuous development (CI/CD) by automating build and deployment processes using Jenkins.

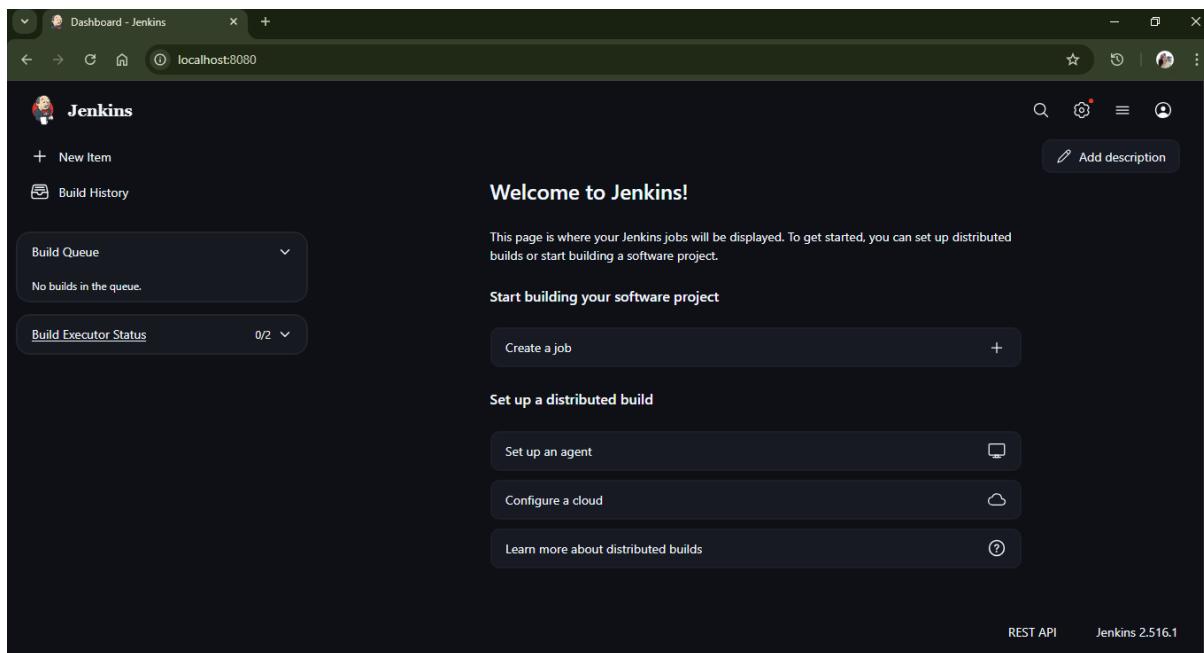
Software Requirements

- Jenkins installed and configured (<https://www.jenkins.io/download/>)
 - Java installed (JDK 11 or higher)
 - Git installed (<https://git-scm.com/downloads>)
 - A sample project (Java/Python/Node.js etc.)
 - Web browser
-

Procedure

1. Configure Jenkins for CI/CD

- Ensure Jenkins is installed and accessible via `http://localhost:8080`.
- Log in to Jenkins dashboard.



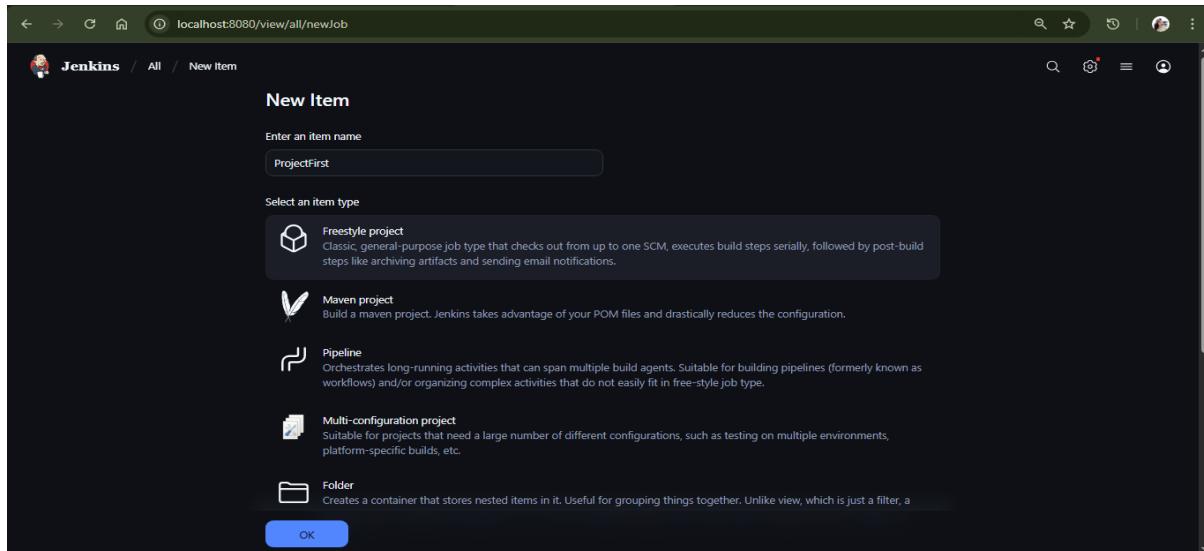
2. Install Required Plugins

- Go to **Settings > Manage Jenkins > Plugins**.
- Install the following plugins if not already installed:
 - Git Plugin

- Pipeline Plugin
 - Any relevant build tool plugin (Maven, Gradle, etc.)
-

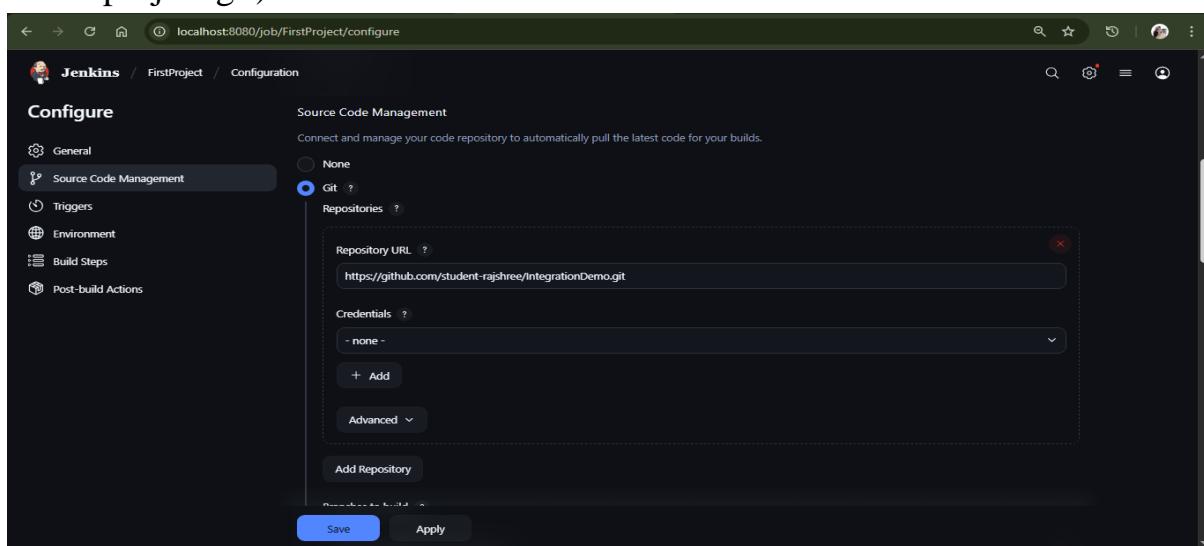
3. Create New Freestyle Project

- Click on **New Item**.
- Enter project name and select **Freestyle project**.
- Click **OK**.



4. Configure Source Code Management

- In the project configuration, go to **Source Code Management**.
- Select **Git**.
- Enter repository URL (e.g., <https://github.com/your-username/sample-project.git>).



5. Configure Build Triggers

- Enable **Poll SCM** or **Build periodically** as needed.
- Alternatively, configure **GitHub webhook** for automatic builds.

The screenshot shows the Jenkins job configuration page for 'FirstProject'. The left sidebar has 'Triggers' selected. Under 'Triggers', 'Build periodically' is checked, and its schedule is set to '*****' (every minute). A warning message states: '⚠ Do you really mean "every minute" when you say "*****"? Perhaps you meant "H * * * *" to poll once per hour'. Below the schedule, there are checkboxes for 'GitHub hook trigger for GITScm polling' and 'Poll SCM', neither of which is checked. At the bottom are 'Save' and 'Apply' buttons.

6. Define Build Steps

- Go to **Build** section.
- Select appropriate build step (e.g., Execute shell, Invoke Maven targets).
- Example build command:
mvn clean install
javac filename.java
java filename

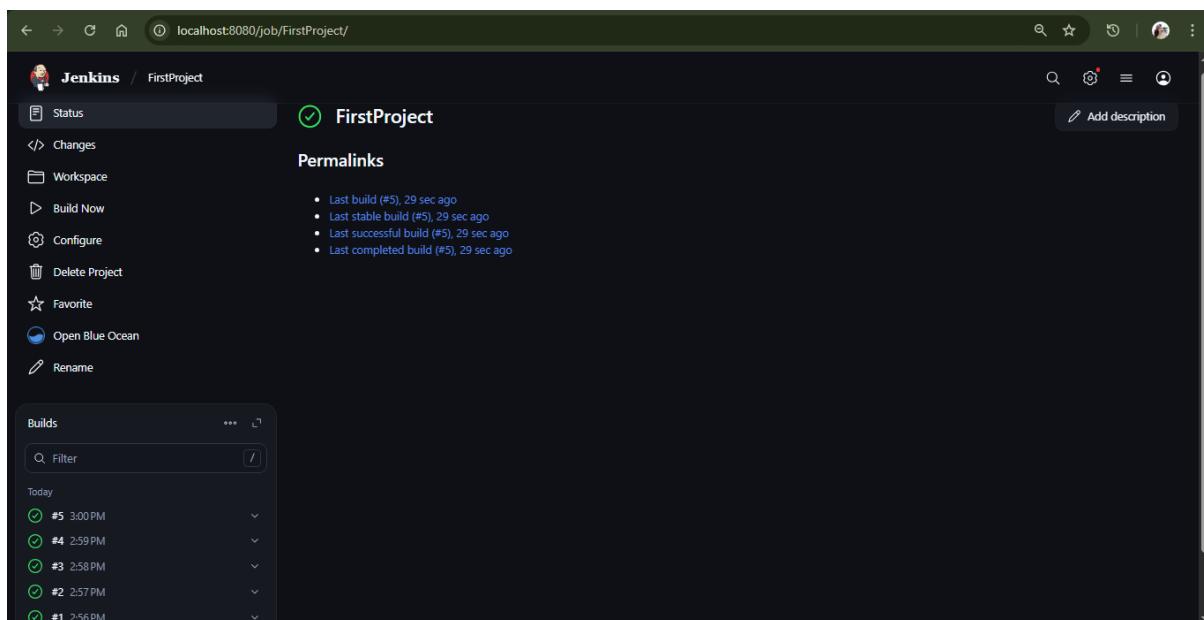
The screenshot shows the Jenkins job configuration page for 'FirstProject'. The left sidebar has 'Build Steps' selected. A single step is defined: 'Execute Windows batch command' with the command 'javac Factorial.java' and 'java Factorial'. Below the build steps is a 'Post-build Actions' section with a 'Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons.

7. Optional: Post-build Actions

- Add post-build steps like:
 - Archive artifacts
 - Email notifications
-

8. Save and Build

- Save the project.
- Click **Build Now**.
- Monitor the build process in **Build History**.



9. View Console Output

- Click on build number.
- View **Console Output** to check build logs.

The screenshot shows the Jenkins interface for a job named 'FirstProject'. The build number is '#5'. The 'Console Output' tab is selected. The log output shows the following steps:

```
Started by timer
Running as SYSTEM
Building in workspace C:\Users\User\.jenkins\workspace\FirstProject
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\Users\User\.jenkins\workspace\FirstProject\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/student-rajshree/IntegrationDemo.git # timeout=10
Fetching upstream changes from https://github.com/student-rajshree/IntegrationDemo.git
> C:\Program Files\Git\bin\git.exe --version # timeout=10
> git --version # git version 2.56.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/student-rajshree/IntegrationDemo.git +refs/heads/*:refs/remotes/origin/*
# timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 405c88982da72c70bb95c7349dd42b2323f764c7 (refs/remotes/origin/main)
> C:\Program Files\Git\bin\git.exe sparsecheckout -f 405c88982da72c70bb95c7349dd42b2323f764c7 # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f 405c88982da72c70bb95c7349dd42b2323f764c7 # timeout=10
Commit message: "Delete DemoIntegration.java"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk 405c88982da72c70bb95c7349dd42b2323f764c7 # timeout=10
[FIRSTPROJECT] $ cmd /c call C:\Users\User\AppData\Local\Temp\jenkins7049028522153200510.bat

C:\Users\User\.jenkins\workspace\FirstProject>javac Factorial.java

C:\Users\User\.jenkins\workspace\FirstProject>java Factorial
Factorial of 10 = 3628800
C:\Users\User\.jenkins\workspace\FirstProject>exit 0
Finished: SUCCESS
```

10. Automate Deployment (Optional)

- Add shell scripts or deployment tasks as build steps to automate deployment after build.