EXPERIMENT-6

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

How to Make a CI-CD Pipeline in Jenkins?

Pre-requisites: Jenkins

DevOps professionals mostly work with pipelines because pipelines can automate the processes like building, testing, and deploying the application. Doing manually by UI takes lots of time and effort which will affect productivity. With the help of Continuous Integration / Continuous Deployment (CI/CD) Pipeline scripts we can automate the whole process which will increase productivity and save lots of time for the organization and can deliver quality applications to the end users.

What is A CI/CD Pipeline?

- CI/CD stands for Continuous Integration / Continuous Deployment
- In computing, a pipeline is a set of stages or processes linked together to form a processing system.
- Each stage in the pipeline takes an input, processes it in accordance with a set of rules, and then sends the outputs to the stage that follows. Frequently, the pipeline's overall output is its final step's output. like the procedures outlined below
  - Test code
  - Build Application
  - Push Repository
  - Deploy to Server

What is Continuous Integration (CI)?

- Continuous integration means whenever new code is committed to remote repositories like GitHub, GitLab, etc. Continuous Integration (CI) will continuously build, tested, and merged into a shared repository.

Benefits of Continuous Integration (CI)

- We can maintain the reports of the projects
- Deployments can be made within the given time
- Bugs can be found quickly.

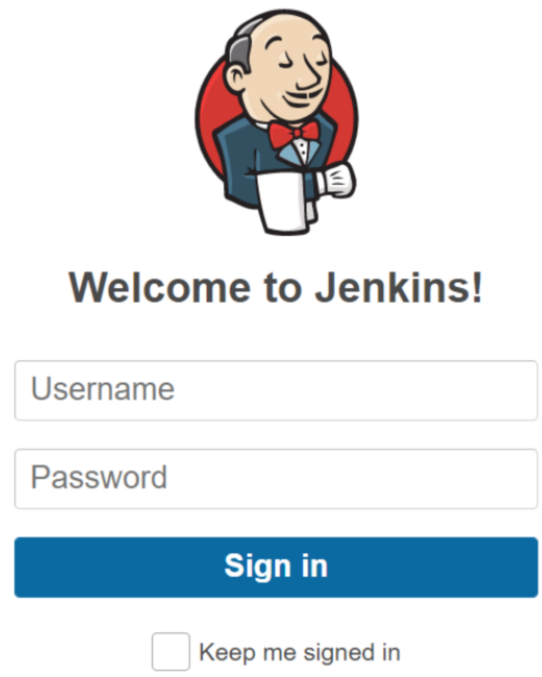
Continuous Deployment(CD)?

Continuous Deployment means automating the further stages of the pipeline automatically or manually deploying the application/code to different environments like Dev, Test, and Production. Automating the build is the main component of Continuous Integration and Continuous Deployment.

Sample Project – To Build A CI/CD Pipeline Using Jenkins

In this project, we will try to learn a basic CI/CD pipeline that is required to build a Java web application .war file by using Jenkins. To install Jenkins refers to the Jenkins installation.

Step 1: Login into your Jenkins account as shown below.
Step 2. Once logged in, the user will be redirected to the Jenkins console, here's the reference for the same.



Welcome to Jenkins!

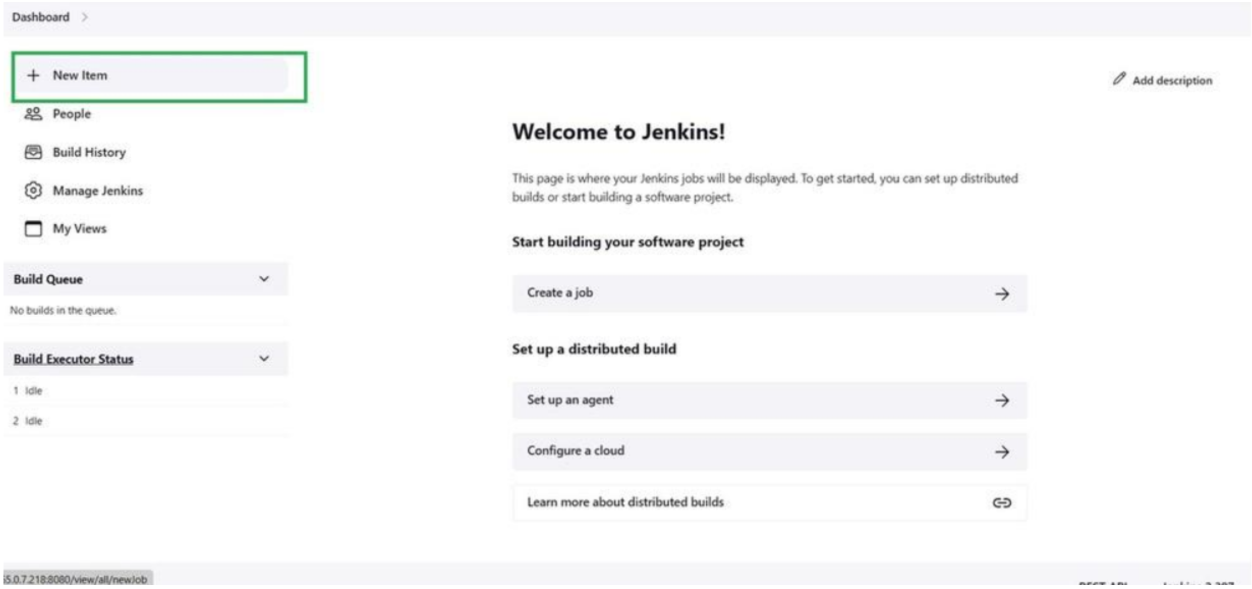
Username

Password

Sign in

□ Keep me signed in

Step 3. To create a new project select the option available in the Dashboard which is **“New Item”** Refer to the image provided below:



Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job →

Set up a distributed build

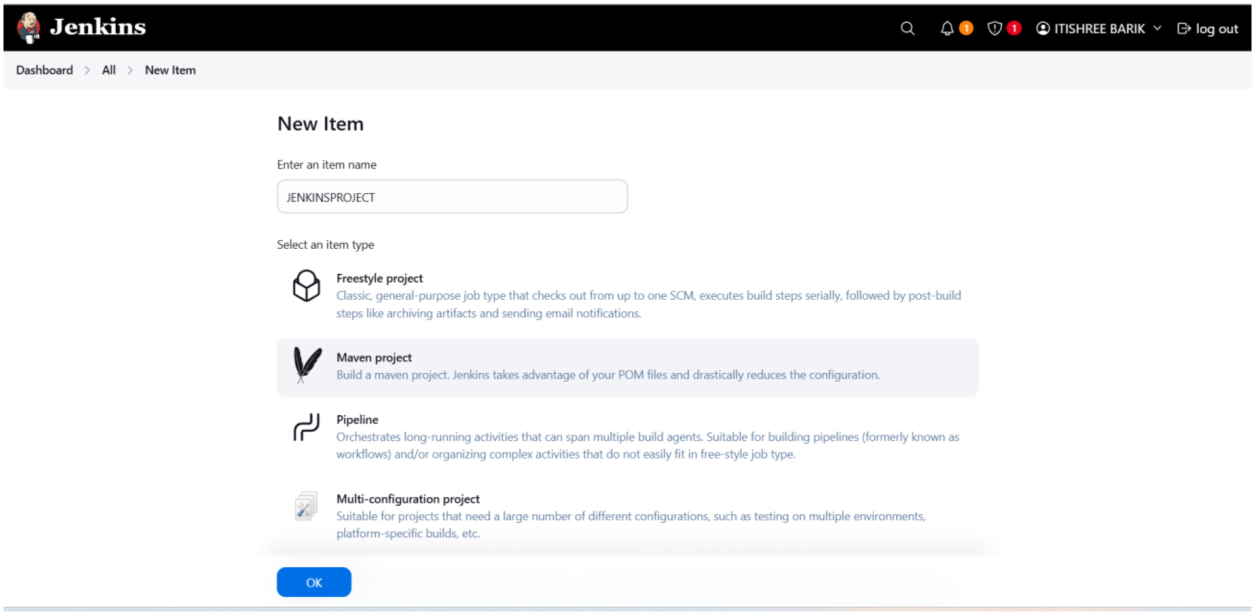
Set up an agent →

Configure a cloud →

Learn more about distributed builds

5.0.7.218:8080/view/all/newJob

Step 4. Now a list of options will be visible on the screen, along with a field to name the maven project. Add a suitable name and select the freestyle project option to proceed.



New Item

Enter an item name

JENKINSPROJECT

Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

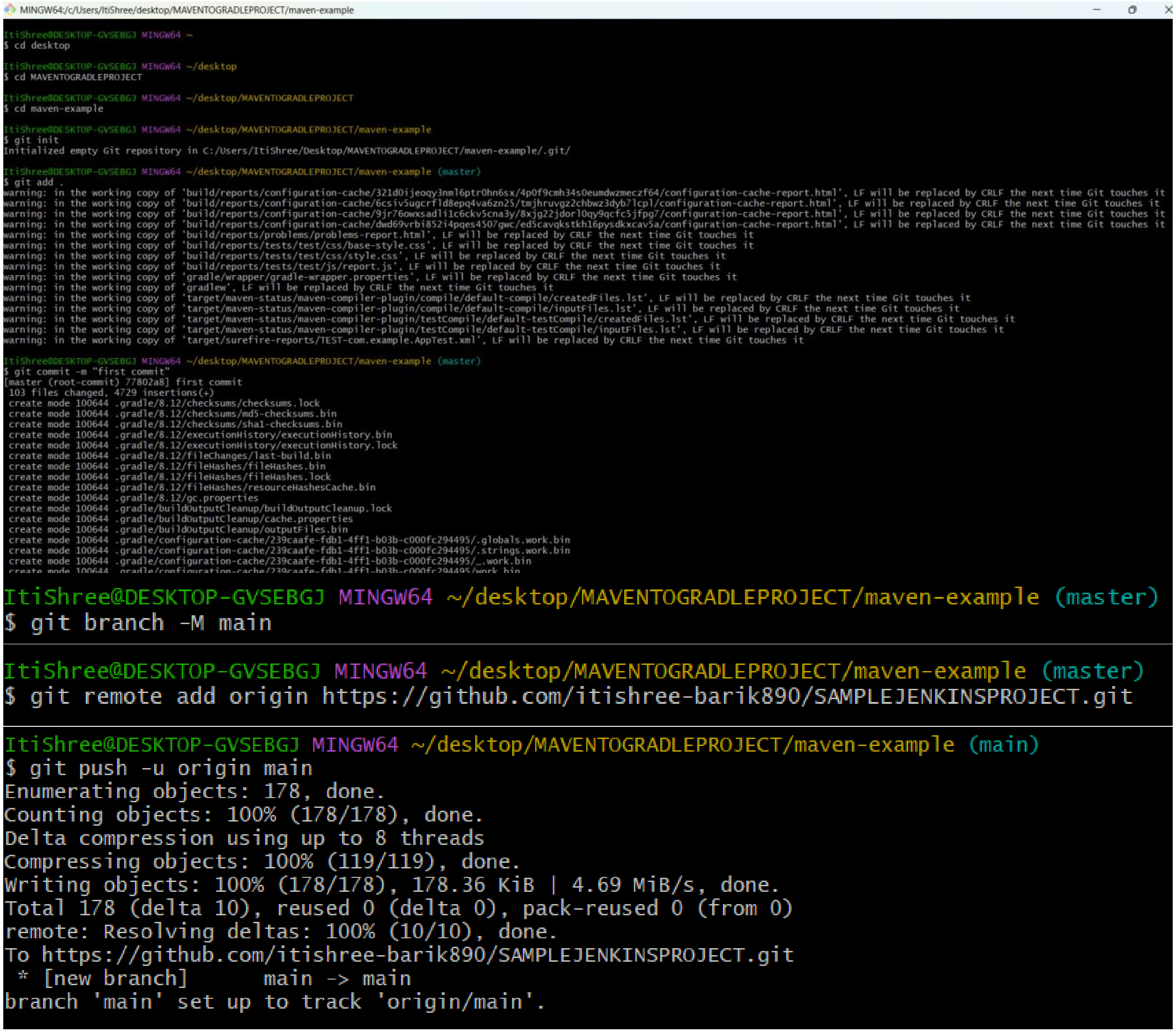
OK

Step 5:open github create a new repository give a Name [SAMPLEJENKINSPROJECT]   CHOOSE Freestyle project.

Step 6: GO to gitbash

```
ItiShree@DESKTOP-GVSEBGJ MINGW64 ~
$ cd desktop

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop
$ cd MAVENTOGRADLEPROJECT

ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT
$ cd maven-example
```

```
ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT/maven-example
$ git init
Initialized empty Git repository in
C:/Users/ItiShree/Desktop/MAVENTOGRADLEPROJECT/maven-example/.git/

ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT/maven-example
(master)
$ git add .
ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT/maven-example
(master)
$ git branch -M main
ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT/maven-example
(main)
$ git remote add origin https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git
ItiShree@DESKTOP-GVSEBGJ MINGW64
~/desktop/MAVENTOGRADLEPROJECT/maven-example
(main)
$ git push -u origin main
```



MINGW64:/c/Users/ItiShree/desktop/MAVENTOGRADLEPROJECT/maven-example

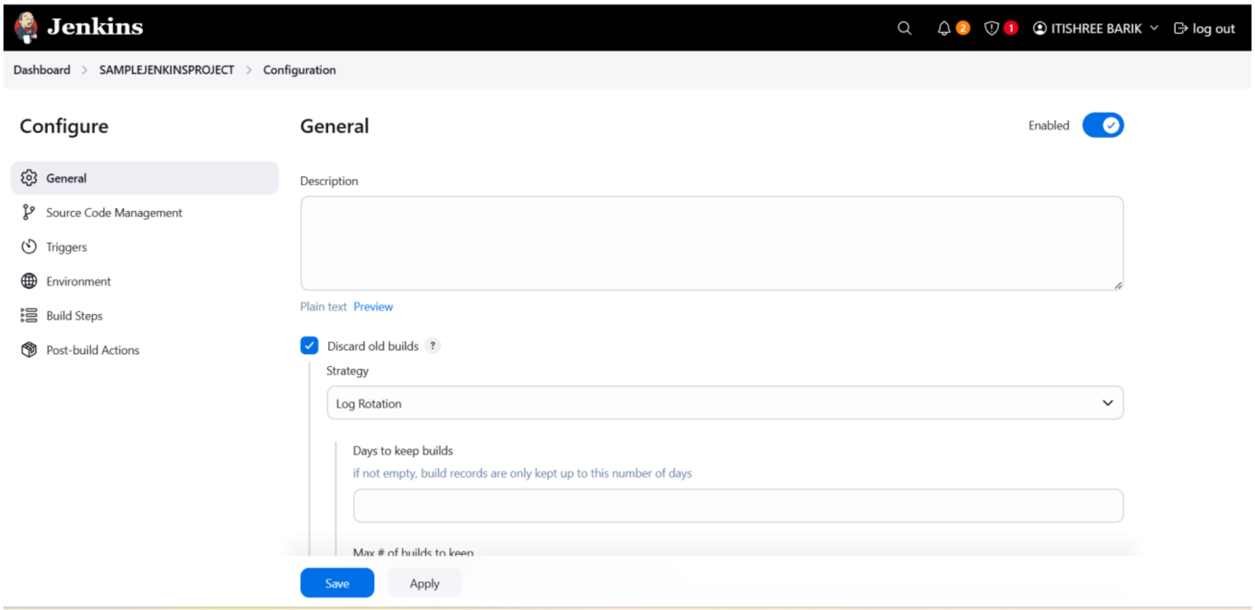
```
ItiShree@DESKTOP-GVSEBGJ MINGW64 ~
$ cd desktop

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop
$ cd MAVENTOGRADLEPROJECT

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT
$ cd maven-example

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example
$ git init
Initialized empty Git repository in C:/Users/ItiShree/Desktop/MAVENTOGRADLEPROJECT/maven-example/.git/

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example (master)
$ git add .
warning: in the working copy of 'build/reports/configuration-cache/321d0ijeoqy3nml6ptr0hn6sx/4p0f9cmh34s0eumdwzmeczf64/configuration-cache-report.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/configuration-cache/6csiv5ugcrfld8epq4va6zn25/tmjhruvgz2chbwz3dyb7lcpl/configuration-cache-report.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/configuration-cache/9jr76owxsadli1c6ckv5cna3y/8xjg22jdorl0qy9qcfc5jfpg7/configuration-cache-report.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/configuration-cache/dwd69vrbi852i4pqes4507gwc/ed5cavqkstkh16pysdkxcav5a/configuration-cache-report.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/problems/problems-report.html', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/tests/test/css/base-style.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/tests/test/css/style.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/reports/tests/test/js/report.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradle/wrapper/gradle-wrapper.properties', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'gradlew', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-status/maven-compiler-plugin/compile/default-compile/createdFiles.lst', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles.lst', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/createdFiles.lst', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'target/surefire-reports/TEST-com.example.AppTest.xml', LF will be replaced by CRLF the next time Git touches it

ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example (master)
$ git commit -m "first commit"
[master (root-commit) 77802a8] first commit
 103 files changed, 4729 insertions(+)
 create mode 100644 .gradle/8.12/checksums/checksums.lock
 create mode 100644 .gradle/8.12/checksums/md5-checksums.bin
 create mode 100644 .gradle/8.12/checksums/sha1-checksums.bin
 create mode 100644 .gradle/8.12/executionHistory/executionHistory.bin
 create mode 100644 .gradle/8.12/executionHistory/executionHistory.lock
 create mode 100644 .gradle/8.12/fileChanges/last-build.bin
 create mode 100644 .gradle/8.12/fileHashes/fileHashes.bin
 create mode 100644 .gradle/8.12/fileHashes/fileHashes.lock
 create mode 100644 .gradle/8.12/fileHashes/resourceHashesCache.bin
 create mode 100644 .gradle/8.12/gc.properties
 create mode 100644 .gradle/buildOutputCleanup/buildOutputCleanup.lock
 create mode 100644 .gradle/buildOutputCleanup/cache.properties
 create mode 100644 .gradle/buildOutputCleanup/outputFiles.bin
 create mode 100644 .gradle/configuration-cache/239caafe-fdb1-4ff1-b03b-c000fc294495/.globals.work.bin
 create mode 100644 .gradle/configuration-cache/239caafe-fdb1-4ff1-b03b-c000fc294495/.strings.work.bin
 create mode 100644 .gradle/configuration-cache/239caafe-fdb1-4ff1-b03b-c000fc294495/_.work.bin
 create mode 100644 .gradle/configuration-cache/239caafe-fdb1-4ff1-b03b-c000fc294495/work.bin
```

```
ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example (master)
$ git branch -M main
```

```
ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example (master)
$ git remote add origin https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git
```

```
ItiShree@DESKTOP-GVSEBGJ MINGW64 ~/desktop/MAVENTOGRADLEPROJECT/maven-example (main)
$ git push -u origin main
Enumerating objects: 178, done.
Counting objects: 100% (178/178), done.
Delta compression using up to 8 threads
Compressing objects: 100% (119/119), done.
Writing objects: 100% (178/178), 178.36 KiB | 4.69 MiB/s, done.
Total 178 (delta 10), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (10/10), done.
To https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

Step 8:Go to Jenkins dashboard click on source code management →choose git→paste github Repo URL.



General

Enabled ☑

Description

☑ Discard old builds ?

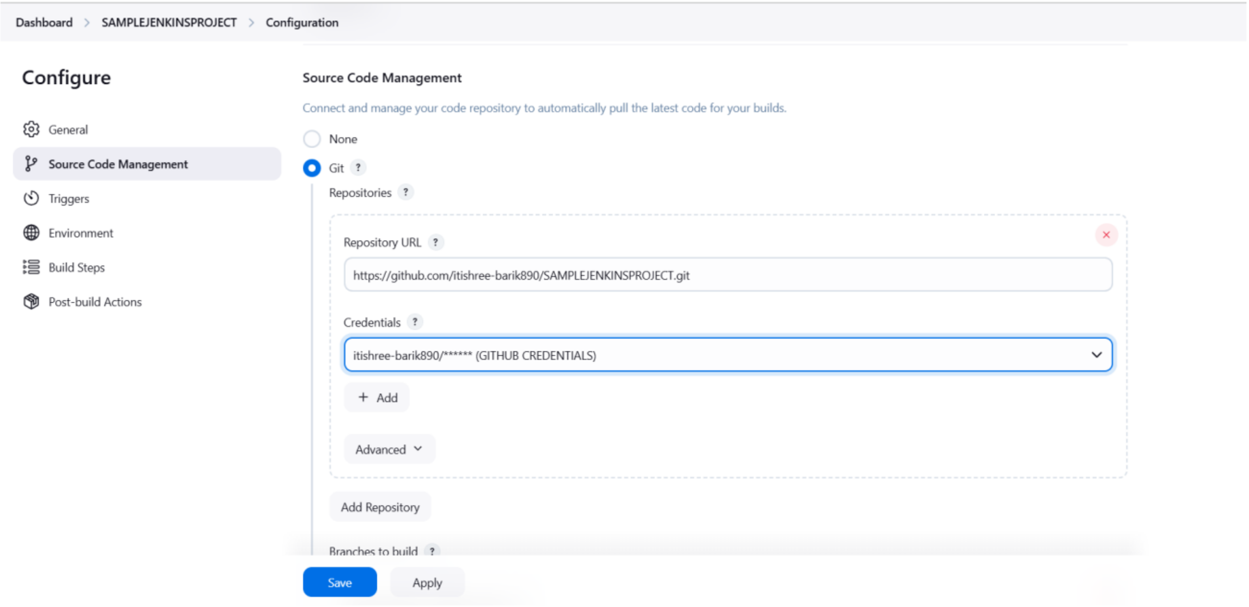
Strategy

Log Rotation

Days to keep builds

if not empty, build records are only kept up to this number of days

Max # of builds to keep



Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

○ None

● Git ?

Repositories ?

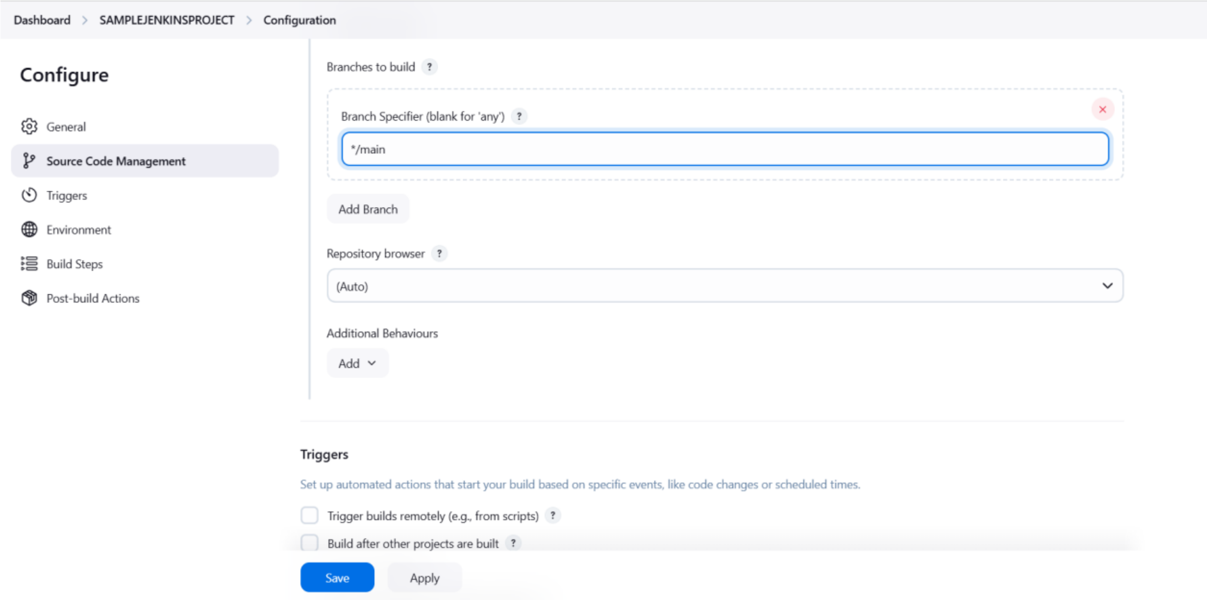
Repository URL ?

https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git

Credentials ?

itishree-barik890/****** (GITHUB CREDENTIALS)

Branches to build ?



Dashboard > SAMPLEJENKINSPROJECT > Configuration

Configure

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Repository browser ?

(Auto)

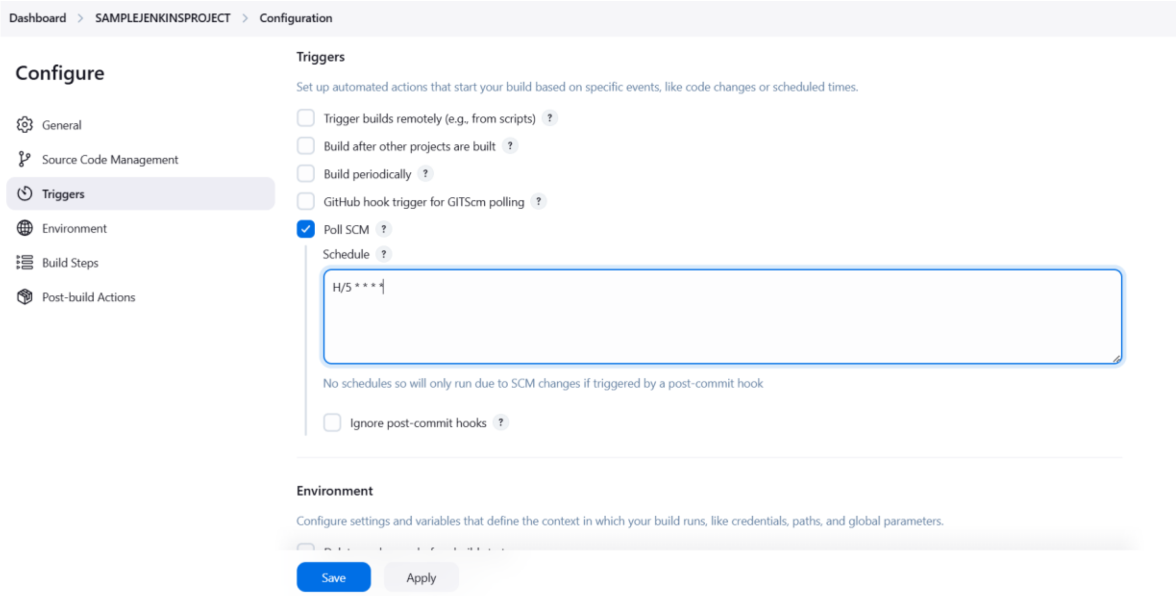
Additional Behaviours

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?



Dashboard > SAMPLEJENKINSPROJECT > Configuration

Configure

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

□ Trigger builds remotely (e.g., from scripts) ?

□ Build after other projects are built ?

□ Build periodically ?

□ GitHub hook trigger for GITScm polling ?

☑ Poll SCM ?

Schedule ?

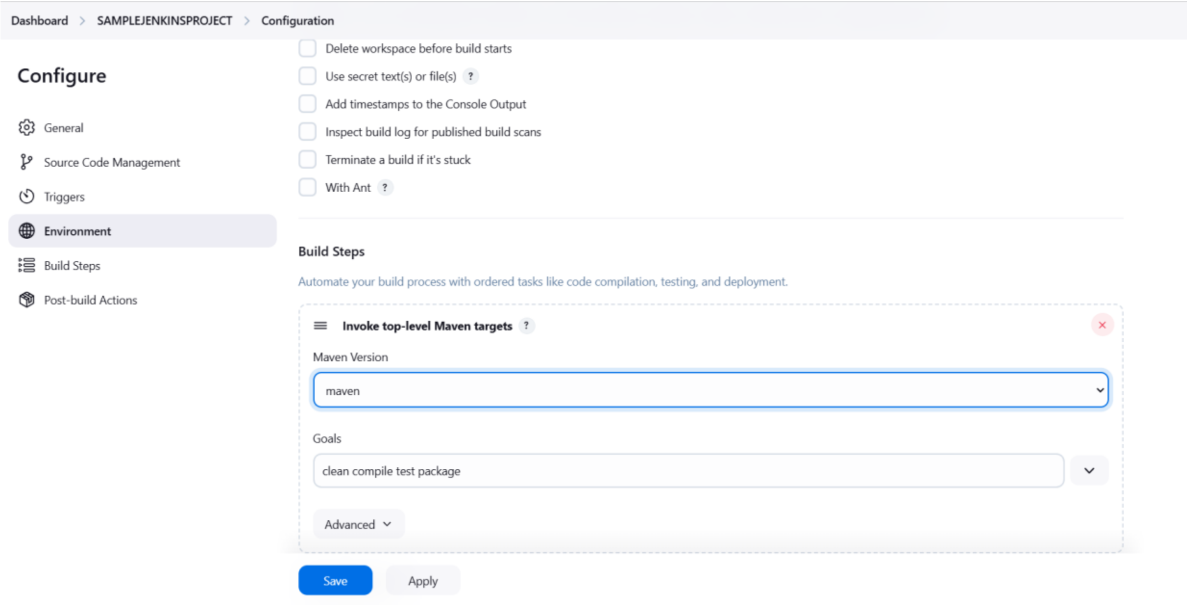
```
H/5 * * * *
```

No schedules so will only run due to SCM changes if triggered by a post-commit hook

□ Ignore post-commit hooks ?

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.



Dashboard > SAMPLEJENKINSPROJECT > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

□ Delete workspace before build starts
□ Use secret text(s) or file(s) ?
□ Add timestamps to the Console Output
□ Inspect build log for published build scans
□ Terminate a build if it's stuck
□ With Ant ?

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.

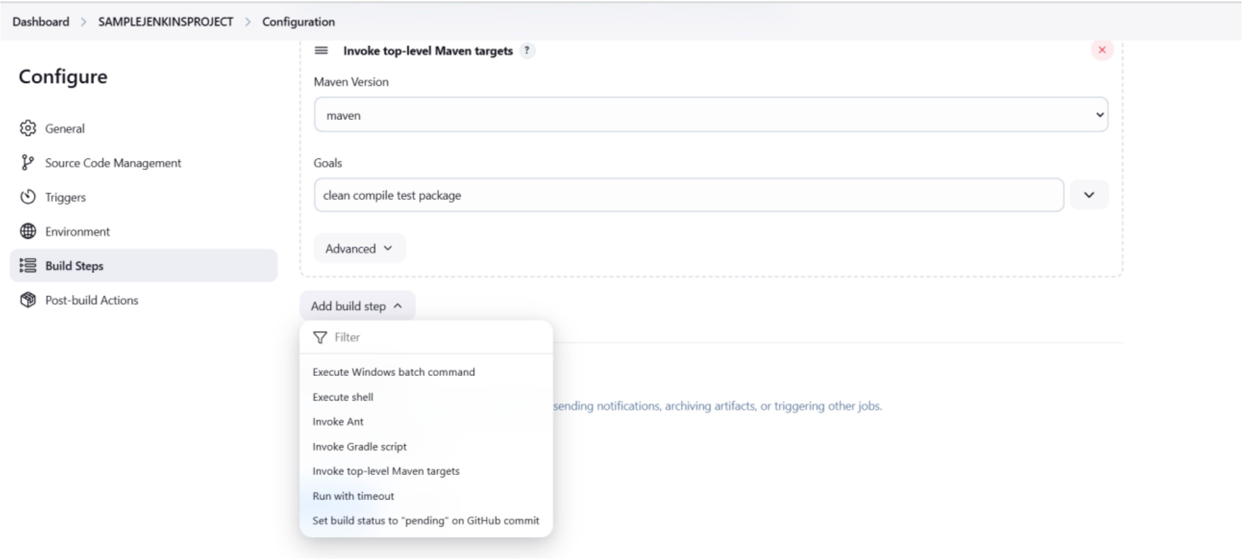
Invoke top-level Maven targets ?

Maven Version: maven

Goals: clean compile test package

Advanced

Save | Apply



Dashboard > SAMPLEJENKINSPROJECT > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Invoke top-level Maven targets ?

Maven Version: maven

Goals: clean compile test package

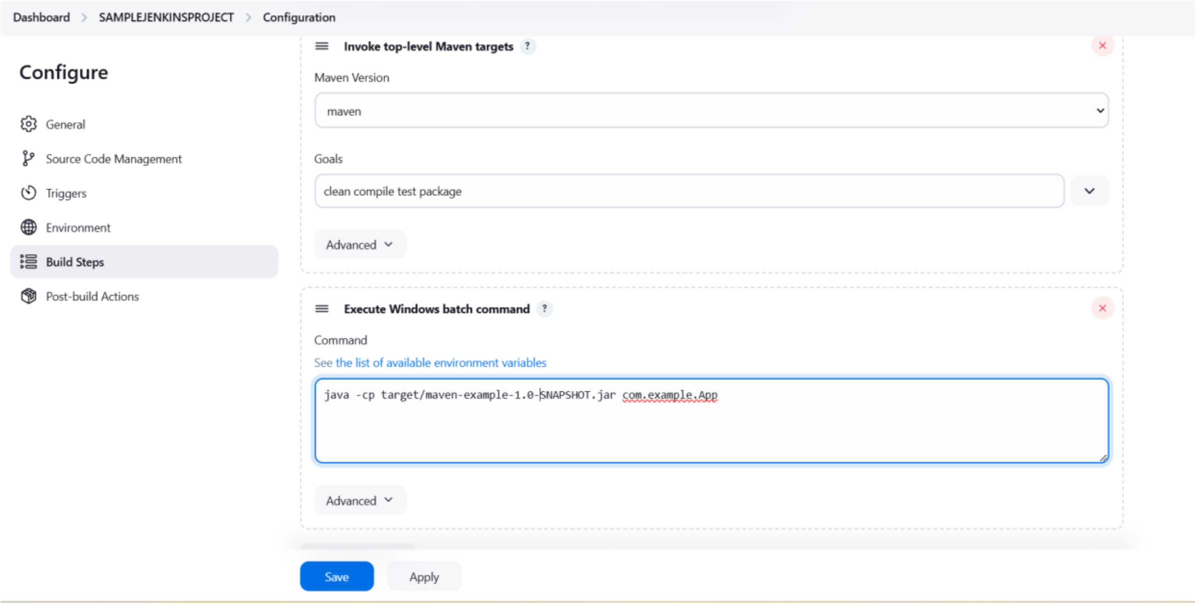
Advanced

Add build step

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

sending notifications, archiving artifacts, or triggering other jobs.



Invoke top-level Maven targets

Maven Version: maven

Goals: clean compile test package

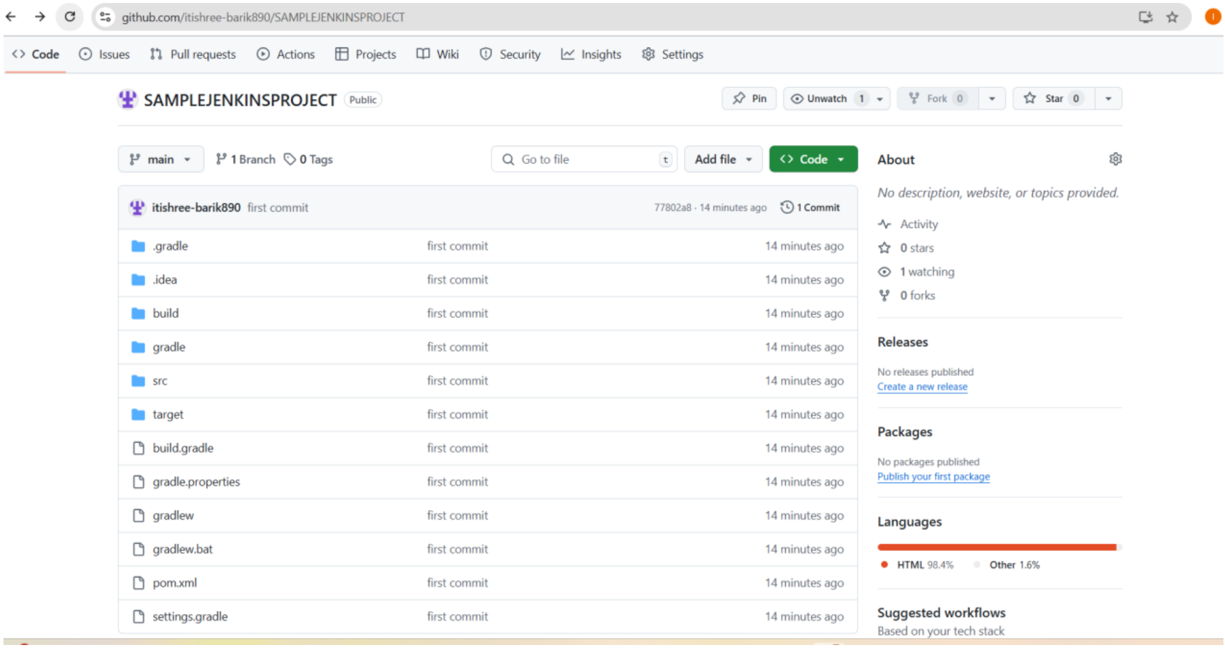
Execute Windows batch command

Command

See the list of available environment variables

```
java -cp target/maven-example-1.0-SNAPSHOT.jar com.example.App
```

NOTE:Now you can see your maven project file added on github.



itishree-barik890 first commit 77802a8 · 14 minutes ago 1 Commit

.gradle	first commit	14 minutes ago
.idea	first commit	14 minutes ago
build	first commit	14 minutes ago
gradle	first commit	14 minutes ago
src	first commit	14 minutes ago
target	first commit	14 minutes ago
build.gradle	first commit	14 minutes ago
gradle.properties	first commit	14 minutes ago
gradlew	first commit	14 minutes ago
gradlew.bat	first commit	14 minutes ago
pom.xml	first commit	14 minutes ago
settings.gradle	first commit	14 minutes ago

About

No description, website, or topics provided.

Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

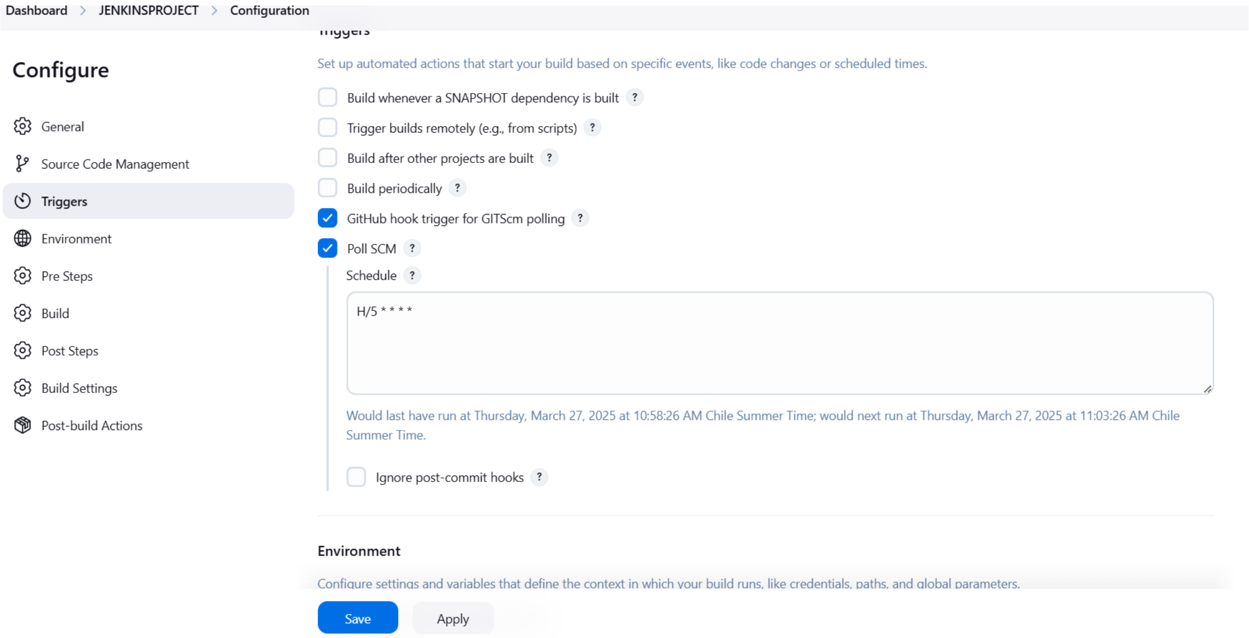
No packages published
Publish your first package

Languages

● HTML 98.4% ● Other 1.6%

Suggested workflows
Based on your tech stack

Step 9: Now comes the second section, i.e. **"Build triggers".** Here, we need to specify the branch and repository and give the credentials too. And add additional behaviors if required so far.



Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

☐ Build whenever a SNAPSHOT dependency is built ?
☐ Trigger builds remotely (e.g., from scripts) ?
☐ Build after other projects are built ?
☐ Build periodically ?
☑ GitHub hook trigger for GITScm polling ?
☑ Poll SCM ?

Schedule ?

```
H/5 * * * *
```

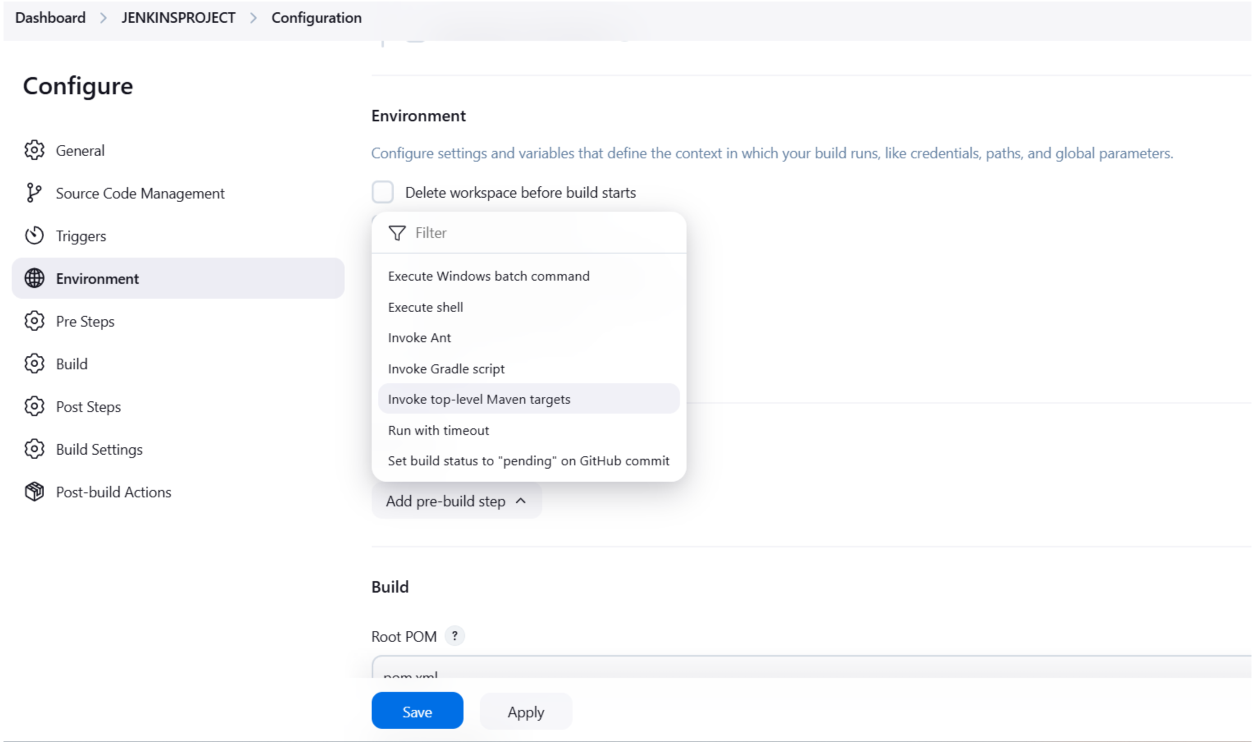
Would last have run at Thursday, March 27, 2025 at 10:58:26 AM Chile Summer Time; would next run at Thursday, March 27, 2025 at 11:03:26 AM Chile Summer Time.

☐ Ignore post-commit hooks ?

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

Step 10: GO To environment click on invoke top level maven target



Dashboard > JENKINSPROJECT > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Pre Steps
- Build
- Post Steps
- Build Settings
- Post-build Actions

Environment

Configure settings and variables that define the context in which your build runs, like credentials, paths, and global parameters.

□ Delete workspace before build starts

Filter

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit

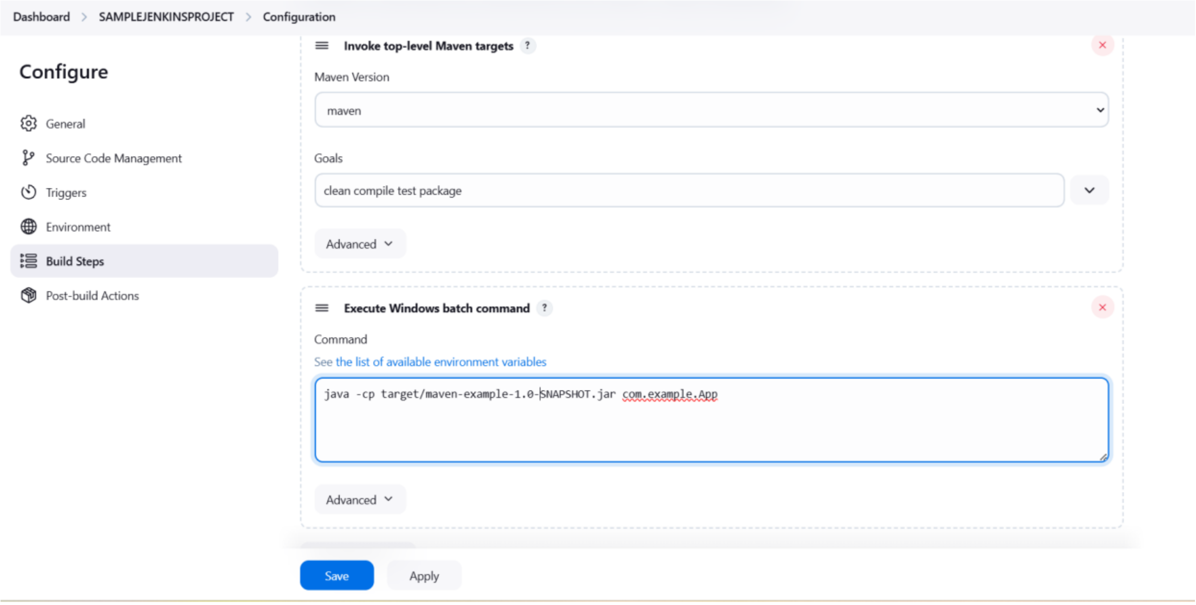
Add pre-build step

Build

Root POM ?

pom.xml

Save Apply



Dashboard > SAMPLEJENKINSPROJECT > Configuration

Configure

- General
- Source Code Management
- Triggers
- Environment
- Build Steps
- Post-build Actions

Invoke top-level Maven targets ?

Maven Version: maven

Goals: clean compile test package

Advanced

Execute Windows batch command ?

Command

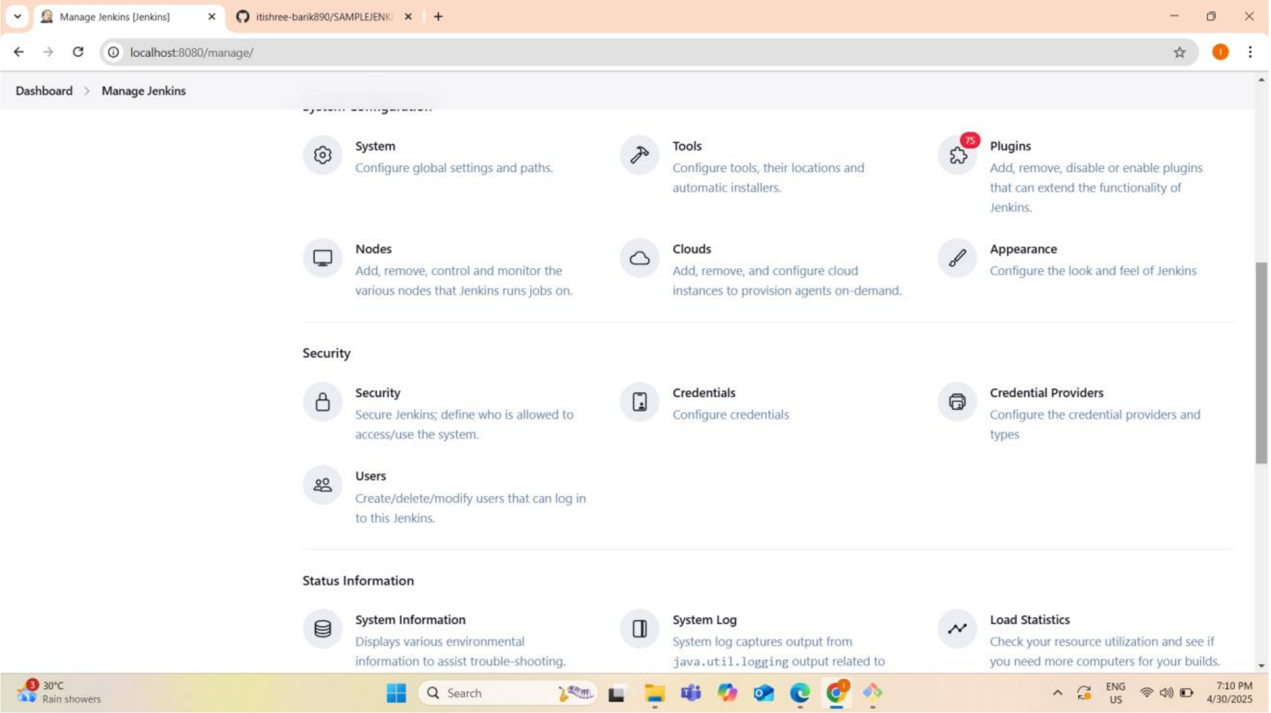
See the list of available environment variables

```
java -cp target/maven-example-1.0-SNAPSHOT.jar com.example.App
```

Advanced

Save Apply

Step 11 : Add plugin



System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

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.

Appearance
Configure the look and feel of Jenkins

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in to this Jenkins.

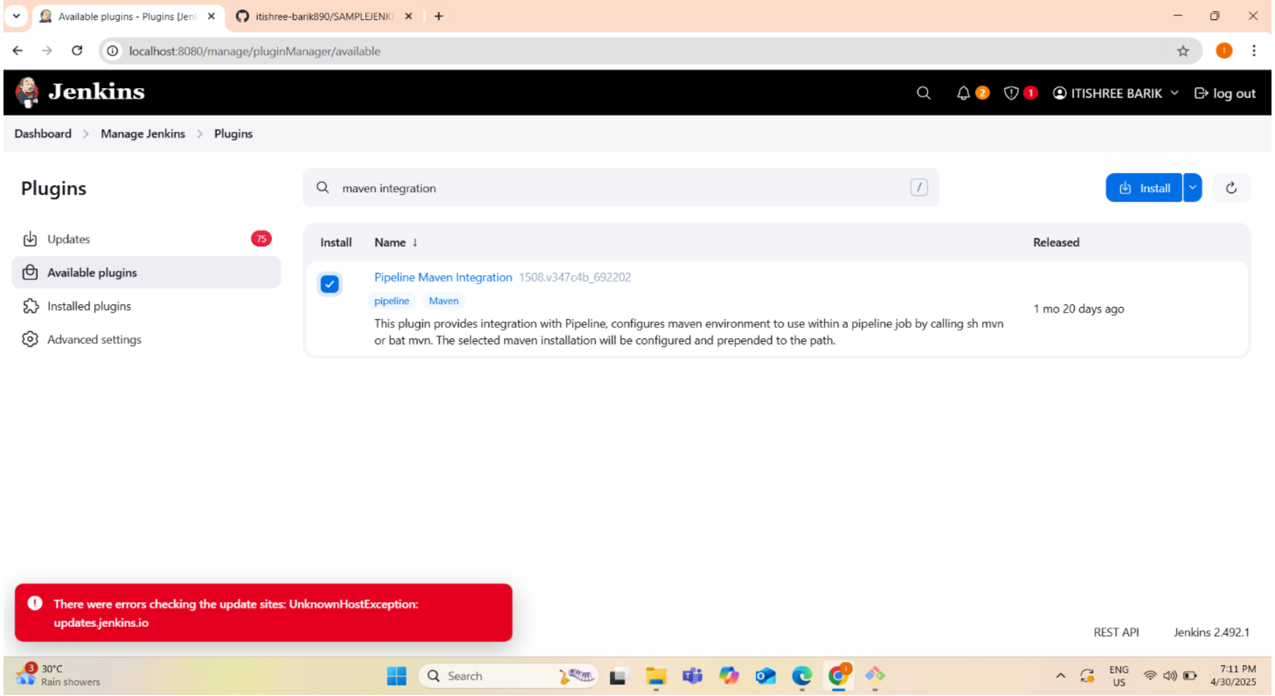
Status Information

System Information
Displays various environmental information to assist trouble-shooting.

System Log
System log captures output from `java.util.logging` output related to

Load Statistics
Check your resource utilization and see if you need more computers for your builds.

30°C
Rain showers
Search
ENG
US
7:10 PM
4/30/2025


Available plugins - Plugins [Jenkins]
itishree-barik890/SAMPLEJENKI
localhost:8080/manage/pluginManager/available
Jenkins
ITISHREE BARIK
log out
Dashboard > Manage Jenkins > Plugins

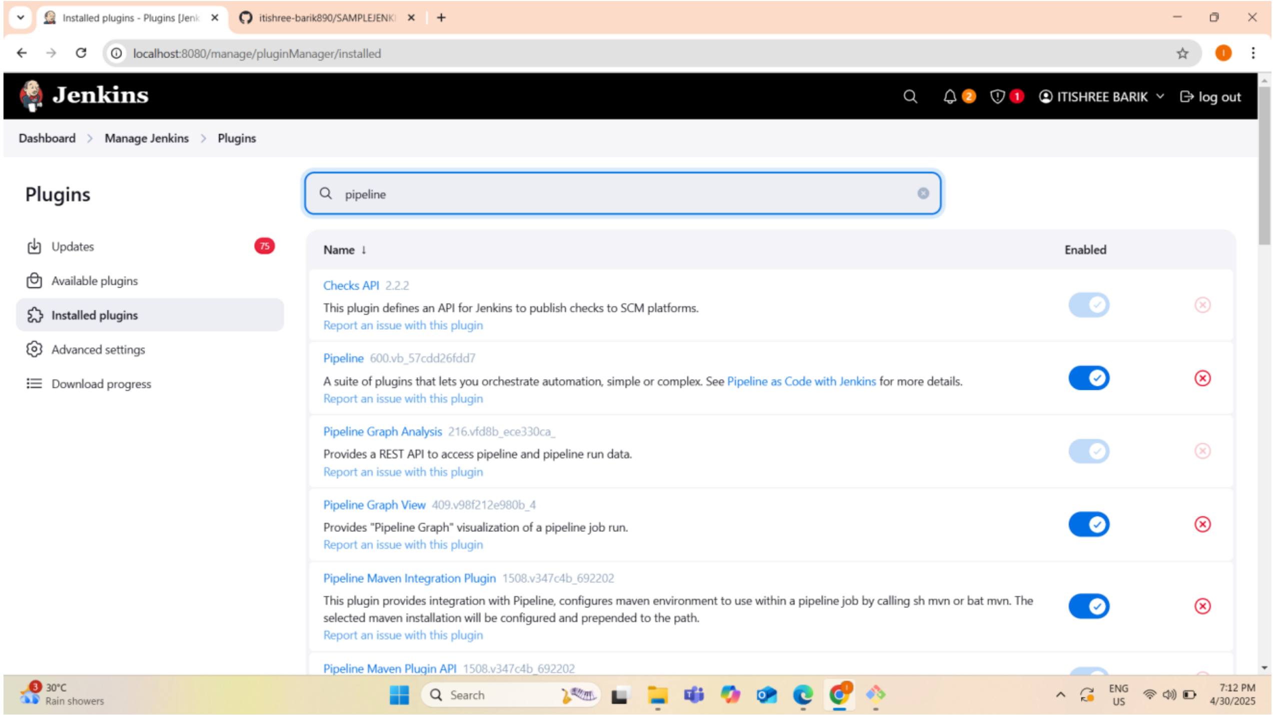
Plugins

Updates 75
Available plugins
Installed plugins
Advanced settings
Install

maven integration

Install	Name ↓	Released
☑	Pipeline Maven Integration 1508.v347c4b_692202 pipeline Maven This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path.	1 mo 20 days ago

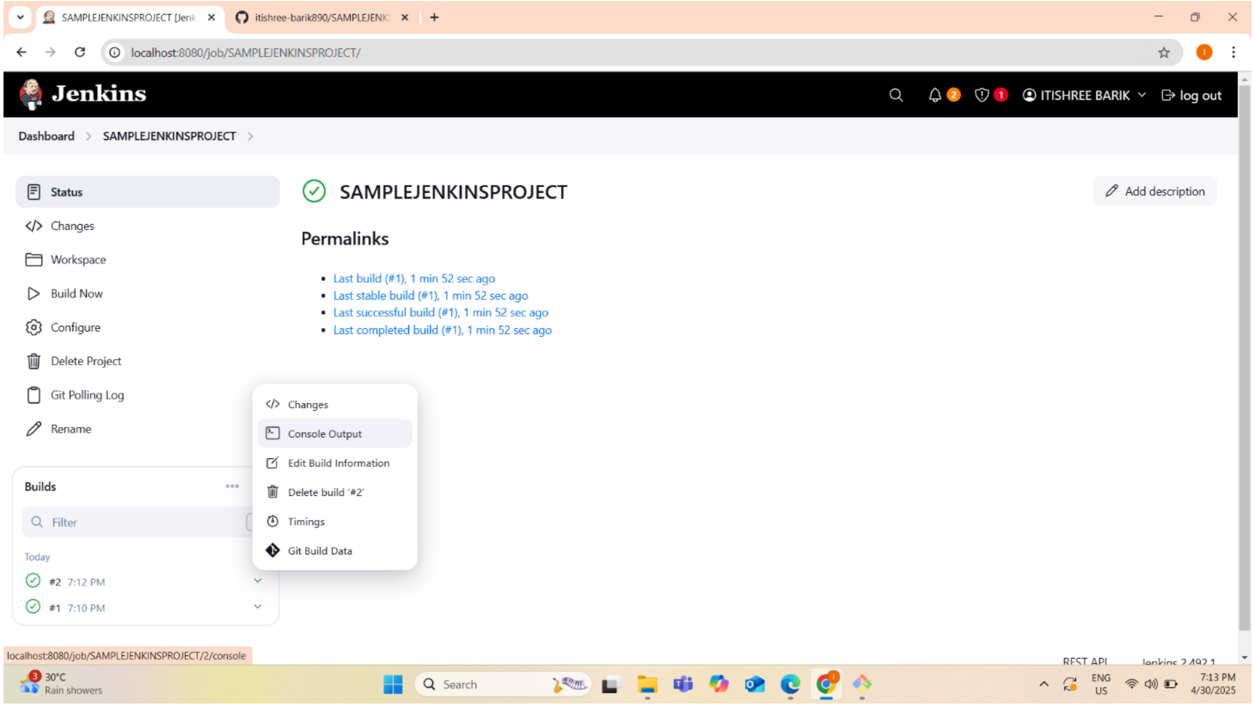
There were errors checking the update sites: UnknownHostException: updates.jenkins.io



Name ↓	Enabled
Checks API 2.2.2 This plugin defines an API for Jenkins to publish checks to SCM platforms. Report an issue with this plugin	✓
Pipeline 600.vb_57cdd26fdd7 A suite of plugins that lets you orchestrate automation, simple or complex. See Pipeline as Code with Jenkins for more details. Report an issue with this plugin	✓
Pipeline Graph Analysis 216.vfd8b_ece330ca_ Provides a REST API to access pipeline and pipeline run data. Report an issue with this plugin	✓
Pipeline Graph View 409.v98f212e980b_4 Provides "Pipeline Graph" visualization of a pipeline job run. Report an issue with this plugin	✓
Pipeline Maven Integration Plugin 1508.v347c4b_692202 This plugin provides integration with Pipeline, configures maven environment to use within a pipeline job by calling sh mvn or bat mvn. The selected maven installation will be configured and prepended to the path. Report an issue with this plugin	✓
Pipeline Maven Plugin API 1508.v347c4b_692202	

Step 10. click on save it will be directly redirected to the Dashboard of the project there we can use, the **“Build Now”** option to run the project and check if it is successful or not, by using stage view or console output.

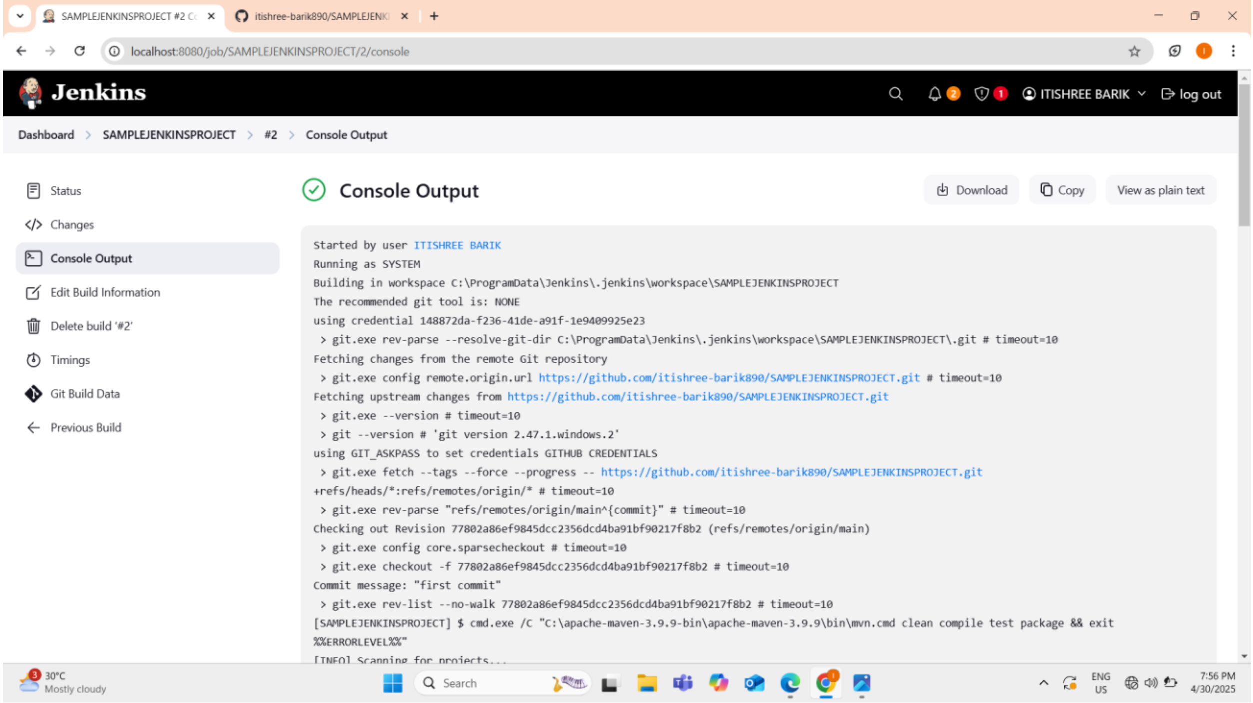
After building project successfully Click on the console output to see the logs of each and every stage that is performed by using the pipeline.



SAMPLEJENKINSPROJECT

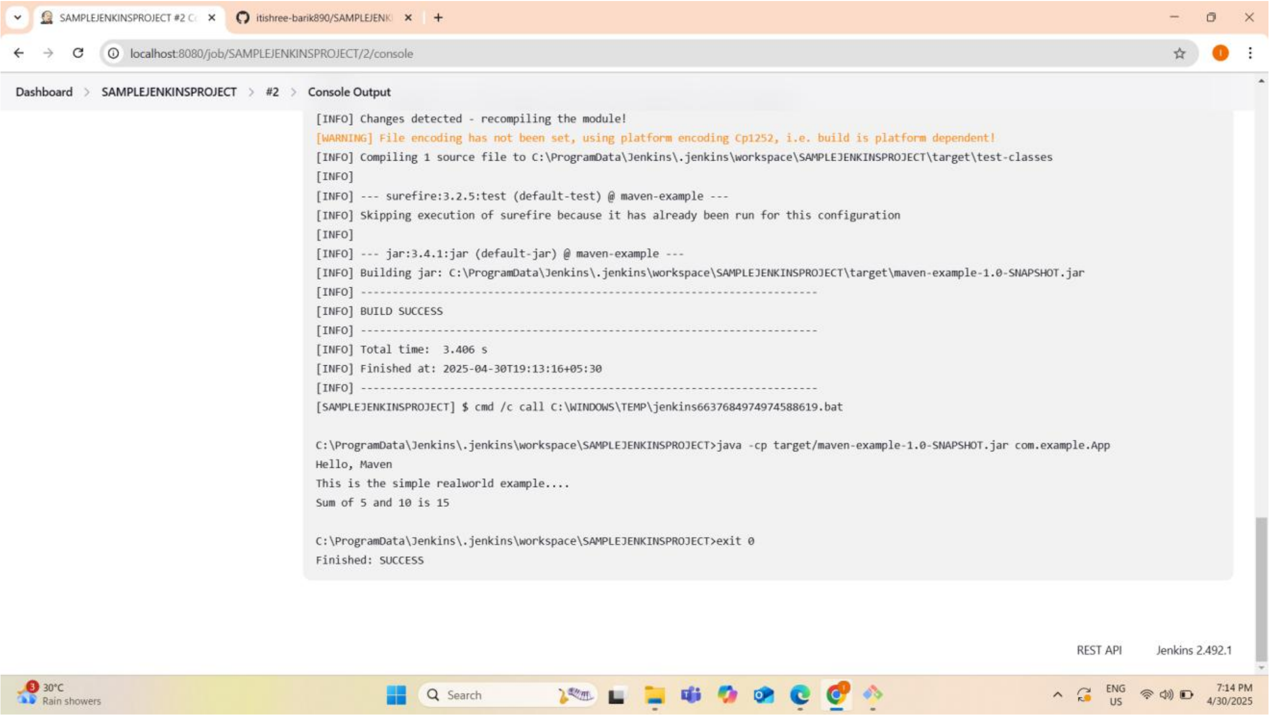
Permalinks

- Last build (#1), 1 min 52 sec ago
- Last stable build (#1), 1 min 52 sec ago
- Last successful build (#1), 1 min 52 sec ago
- Last completed build (#1), 1 min 52 sec ago



Console Output

```
Started by user ITISHREE BARIK
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT
The recommended git tool is: NONE
using credential 148872da-f236-41de-a91f-1e9409925e23
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git # timeout=10
Fetching upstream changes from https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.47.1.windows.2'
using GIT_ASKPASS to set credentials GITHUB CREDENTIALS
 > git.exe fetch --tags --force --progress -- https://github.com/itishree-barik890/SAMPLEJENKINSPROJECT.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 77802a86ef9845dcc2356dcd4ba91bf90217f8b2 (refs/remotes/origin/main)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f 77802a86ef9845dcc2356dcd4ba91bf90217f8b2 # timeout=10
Commit message: "first commit"
 > git.exe rev-list --no-walk 77802a86ef9845dcc2356dcd4ba91bf90217f8b2 # timeout=10
[SAMPLEJENKINSPROJECT] $ cmd.exe /C "C:\apache-maven-3.9.9-bin\apache-maven-3.9.9\bin\mvn.cmd clean compile test package && exit
%%ERRORLEVEL%%"
[INFO] Scanning for projects...
```



SAMPLEJENKINSPROJECT #2 C
itishree-barik890/SAMPLEJENK
localhost:8080/job/SAMPLEJENKINSPROJECT/2/console
Dashboard > SAMPLEJENKINSPROJECT > #2 > Console Output

```
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent!
[INFO] Compiling 1 source file to C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT\target\test-classes
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ maven-example ---
[INFO] Skipping execution of surefire because it has already been run for this configuration
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ maven-example ---
[INFO] Building jar: C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT\target\maven-example-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.406 s
[INFO] Finished at: 2025-04-30T19:13:16+05:30
[INFO] ------------------------------------------------------------------------
[SAMPLEJENKINSPROJECT] $ cmd /c call C:\WINDOWS\TEMP\jenkins6637684974974588619.bat

C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT>java -cp target/maven-example-1.0-SNAPSHOT.jar com.example.App
Hello, Maven
This is the simple realworld example....
Sum of 5 and 10 is 15

C:\ProgramData\Jenkins\.jenkins\workspace\SAMPLEJENKINSPROJECT>exit 0
Finished: SUCCESS
```