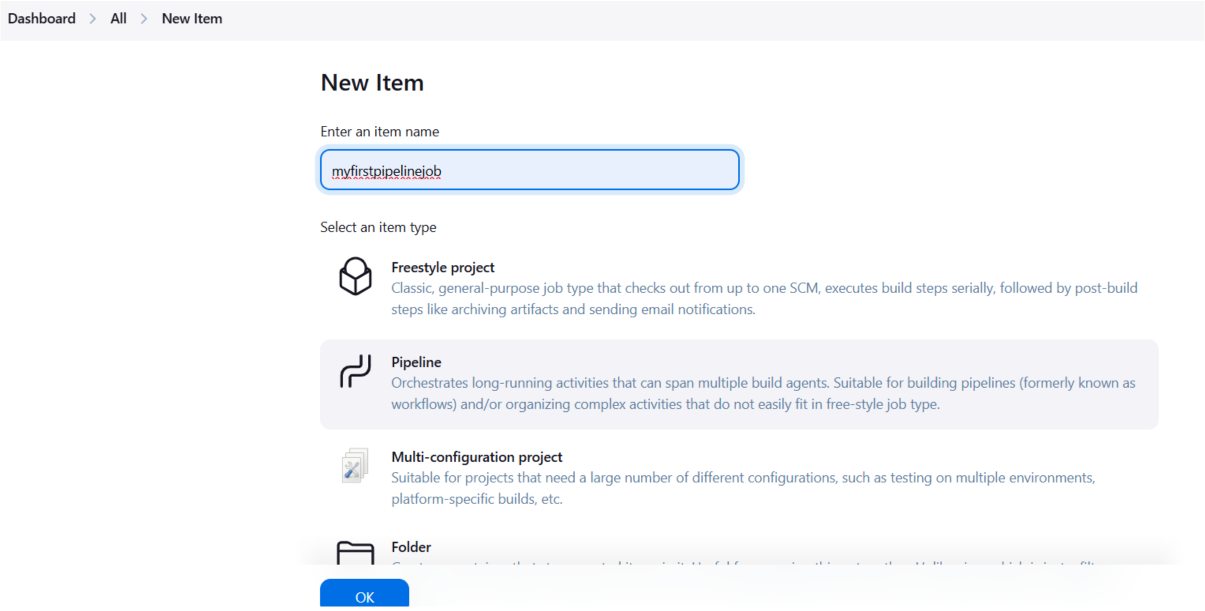
Jenkins:

Checkout private repository.



Dashboard > All > New Item

New Item

Enter an item name

myfirstpipelinejob

Select an item type

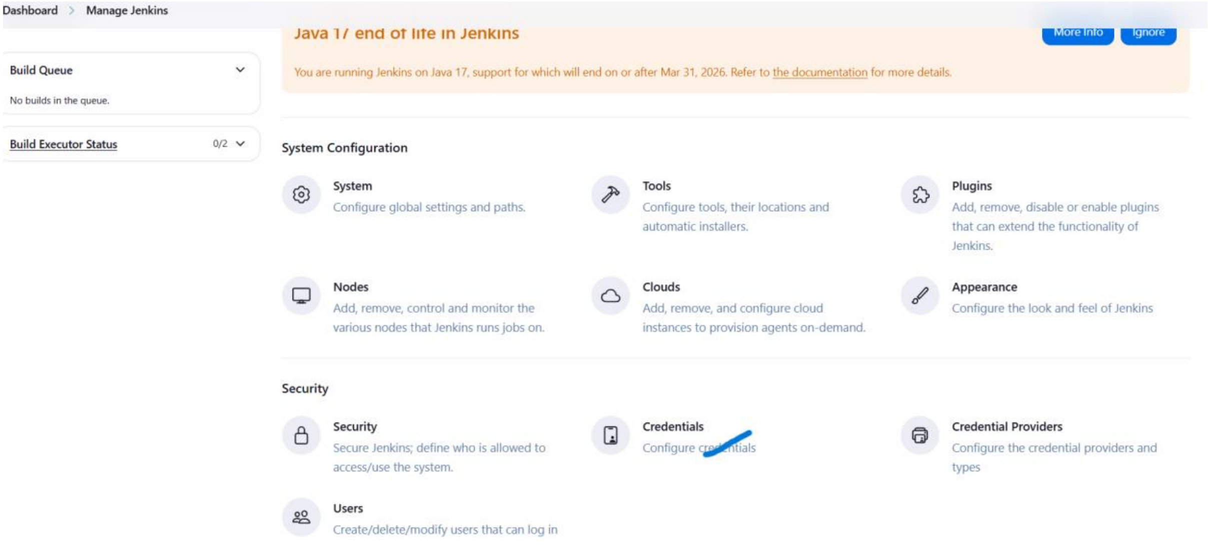
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

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.

Folder

For this first we need to add credentials of github to provide access.



Dashboard > Manage Jenkins

Build Queue
No builds in the queue.

Build Executor Status 0/2

Java 17 end of life in Jenkins

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to the documentation for more details.

System Configuration

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

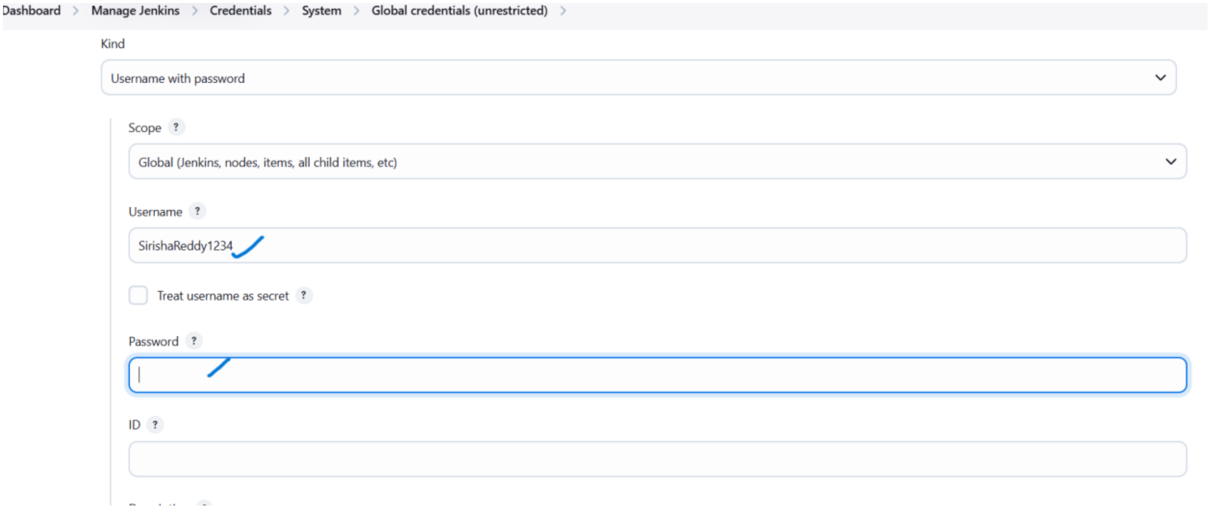
Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in

For password we need to When configuring GitHub authentication, we'll use our GitHub username and a Personal Access Token (PAT) as the password, which can be generated in the GitHub settings under 'Developer settings' > 'Personal access tokens generate one personal access token shown below.



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

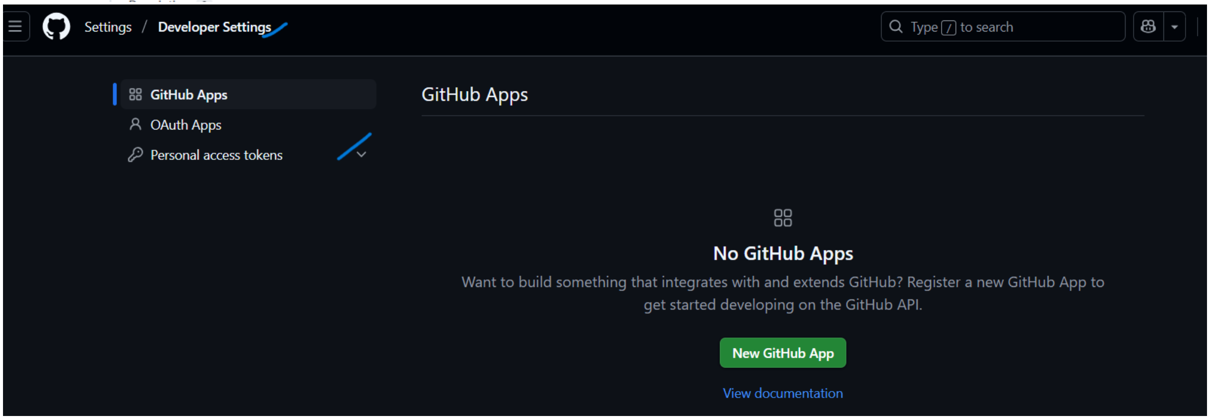
Username ?

SirishaReddy1234

□ Treat username as secret ?

Password ?

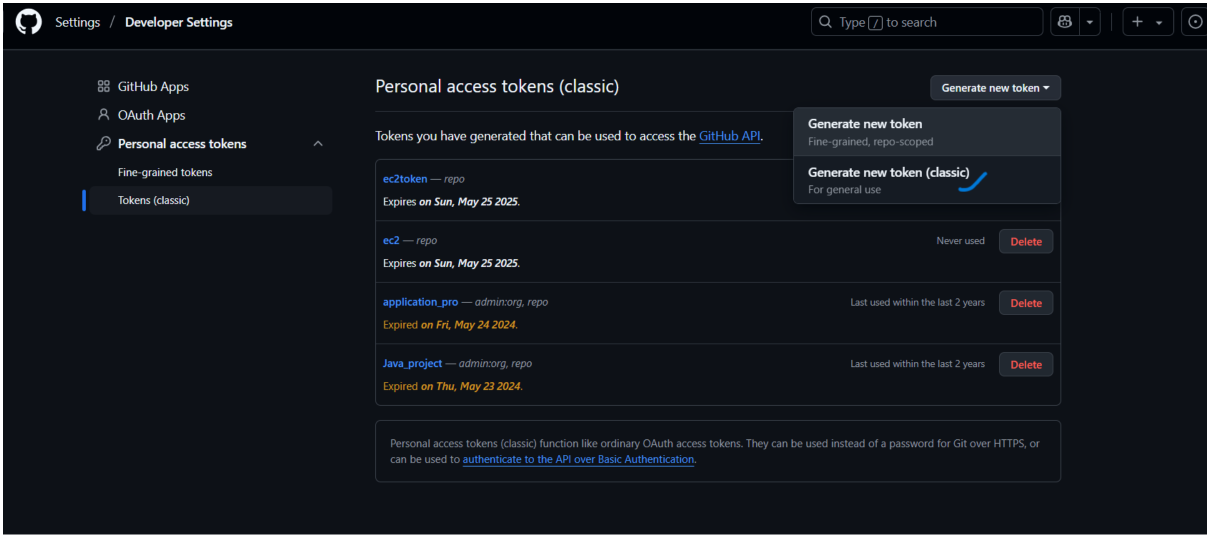
ID ?



GitHub Apps

No GitHub Apps

Want to build something that integrates with and extends GitHub? Register a new GitHub App to get started developing on the GitHub API.



Personal access tokens (classic)

Generate new token
Fine-grained, repo-scoped

Generate new token (classic)
For general use

Tokens you have generated that can be used to access the GitHub API.

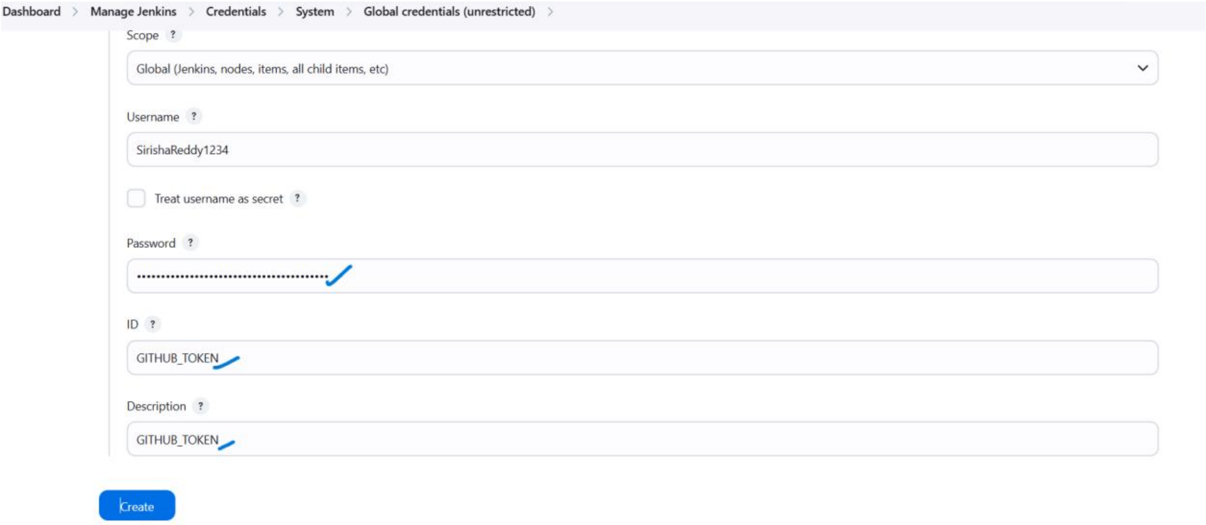
ec2token — *repo*
Expires *on Sun, May 25 2025.*

ec2 — *repo* Never used Delete
Expires *on Sun, May 25 2025.*

application_pro — *admin:org, repo* Last used within the last 2 years Delete
Expired *on Fri, May 24 2024.*

Java_project — *admin:org, repo* Last used within the last 2 years Delete
Expired *on Thu, May 23 2024.*

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

SirishaReddy1234

□ Treat username as secret ?

Password ?

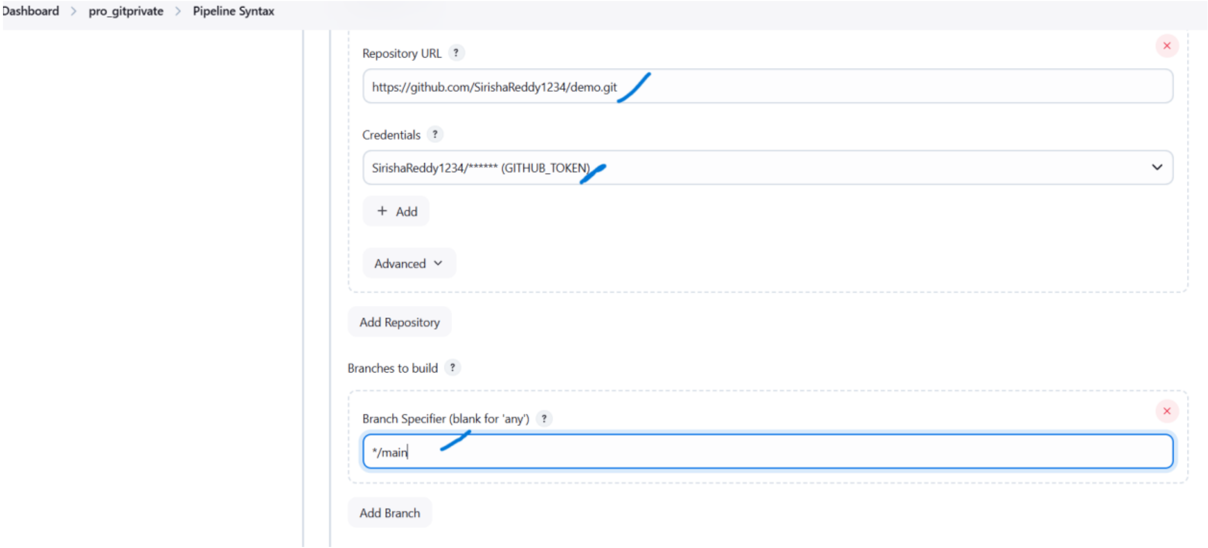
••

ID ?

GITHUB_TOKEN

Description ?

GITHUB_TOKEN



Dashboard > pro_gitprivate > Pipeline Syntax

Repository URL ?

https://github.com/SirishaReddy1234/demo.git

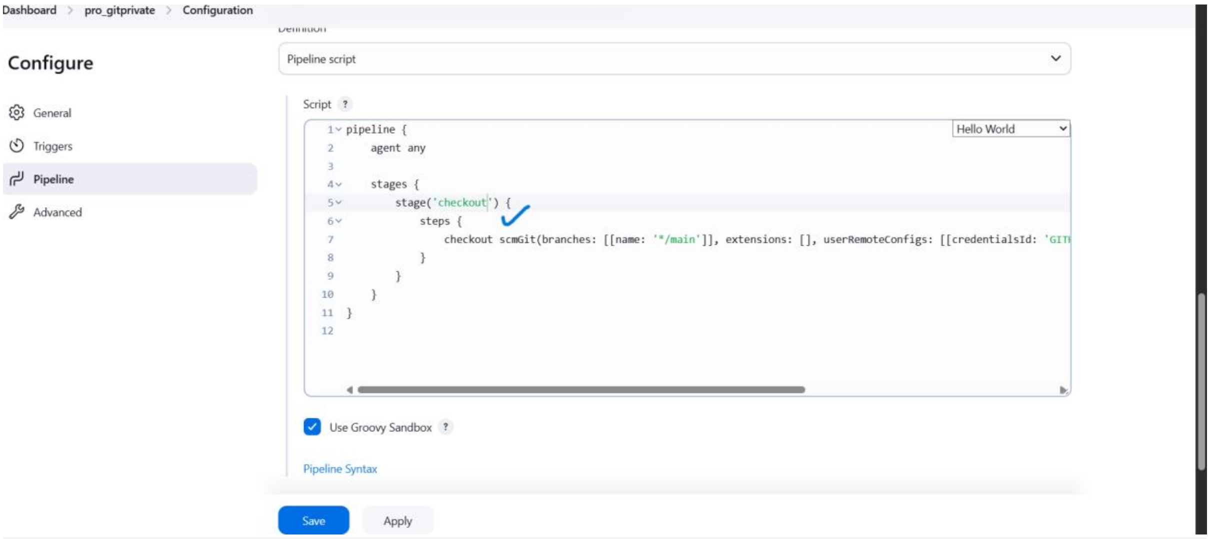
Credentials ?

SirishaReddy1234/****** (GITHUB_TOKEN)

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main



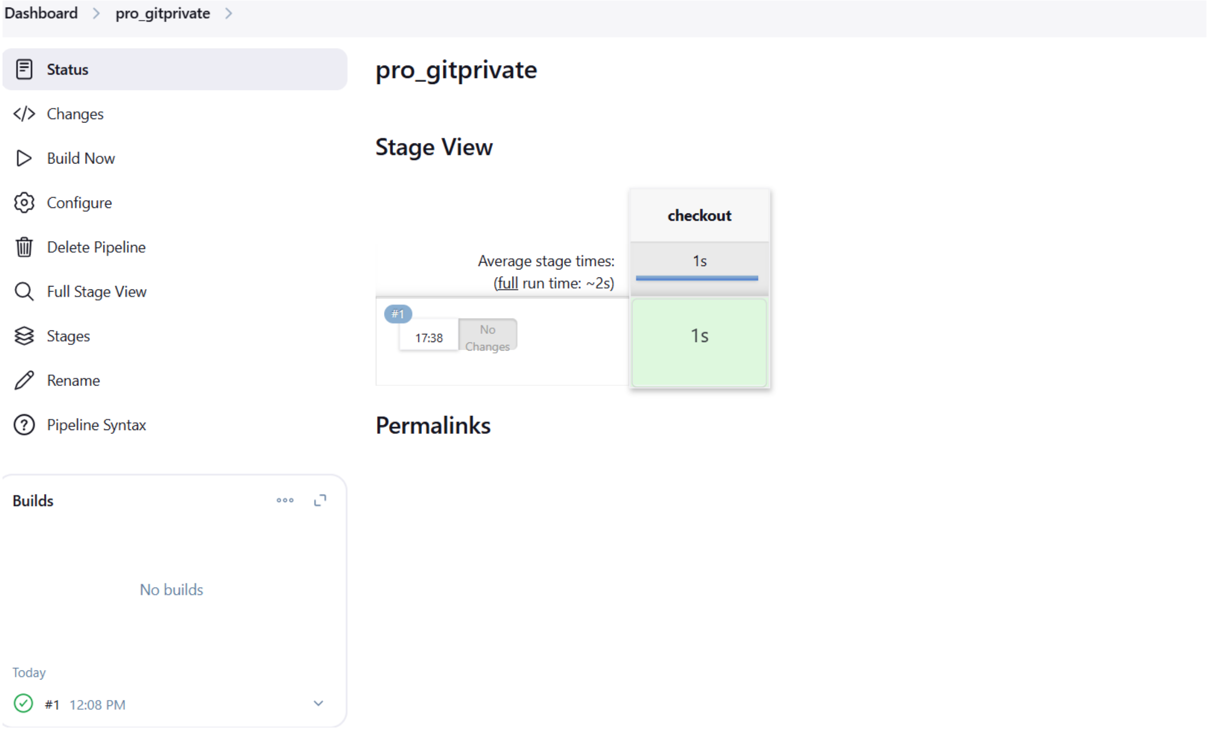
Pipeline script

Script

```
pipeline {
    agent any

    stages {
        stage('checkout') {
            steps {
                checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[credentialsId: 'GIT
            }
        }
    }
}
```

☑ Use Groovy Sandbox



	checkout
Average stage times: (full run time: ~2s)	1s
#1 17:38 No Changes	1s

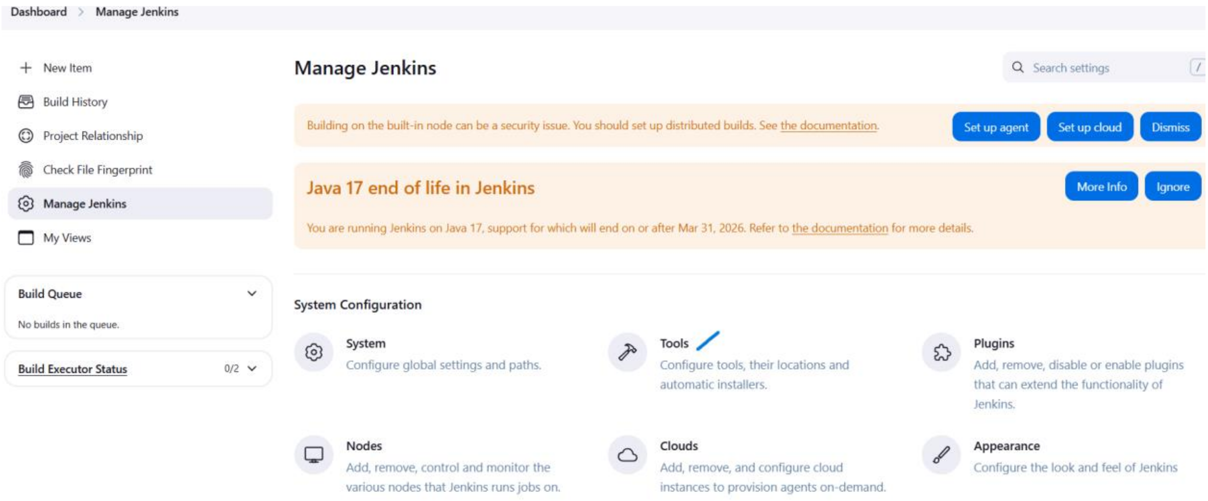
Builds

No builds

Today

✓ #1 12:08 PM

To run Maven commands, we need to configure the Maven path in the tools section of our Jenkins configuration.



Manage Jenkins

Building on the built-in node can be a security issue. You should set up distributed builds. See the documentation.

Java 17 end of life in Jenkins

You are running Jenkins on Java 17, support for which will end on or after Mar 31, 2026. Refer to the documentation for more details.

No builds in the queue.

System Configuration

System
Configure global settings and paths.

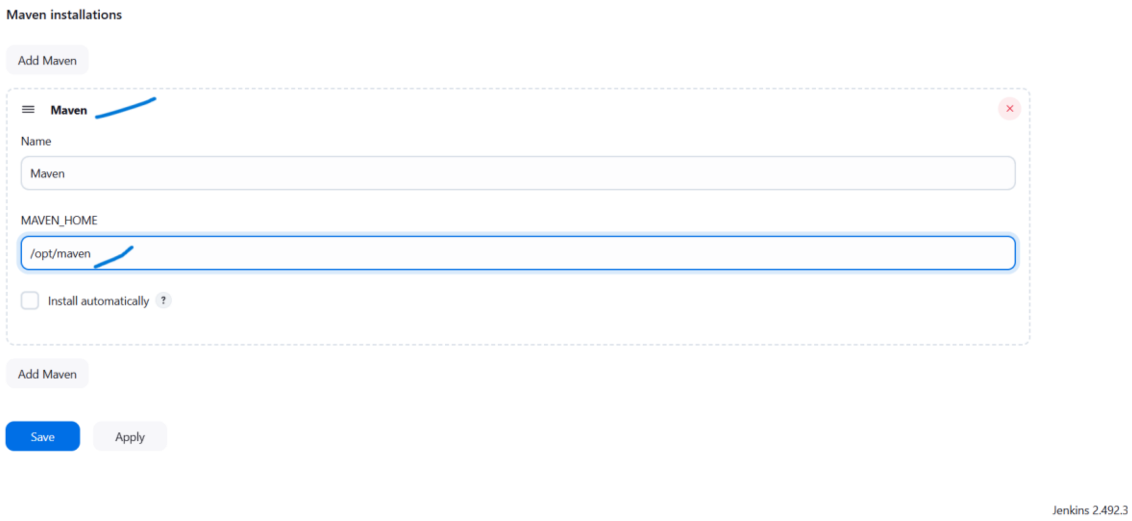
Tools
Configure tools, their locations and automatic installers.

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



Maven installations

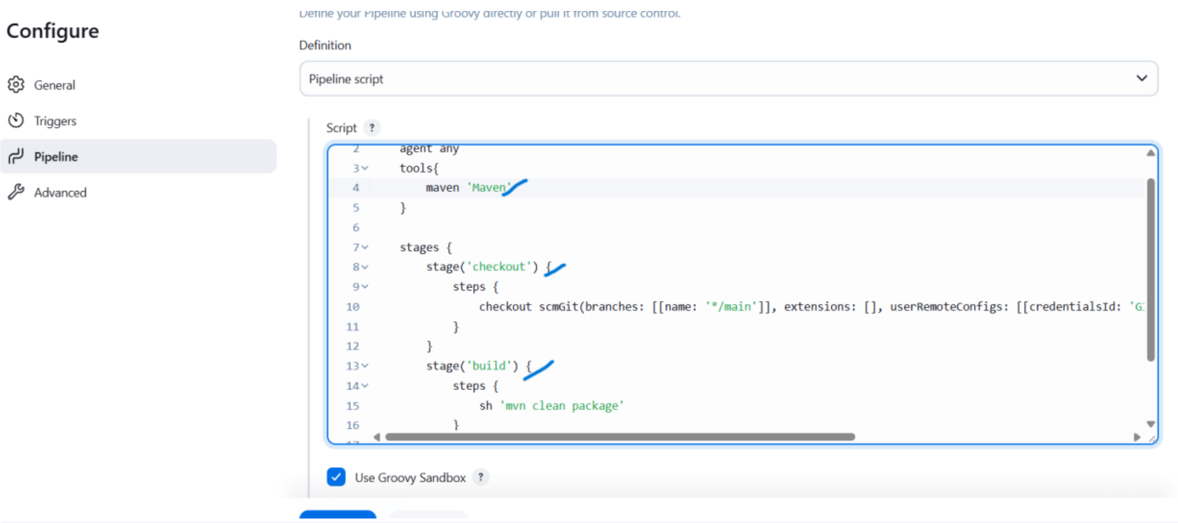
Maven

Name: Maven

MAVEN_HOME: /opt/maven

□ Install automatically

Jenkins 2.492.3



Define your Pipeline using Groovy directly or pull it from source control.

Definition

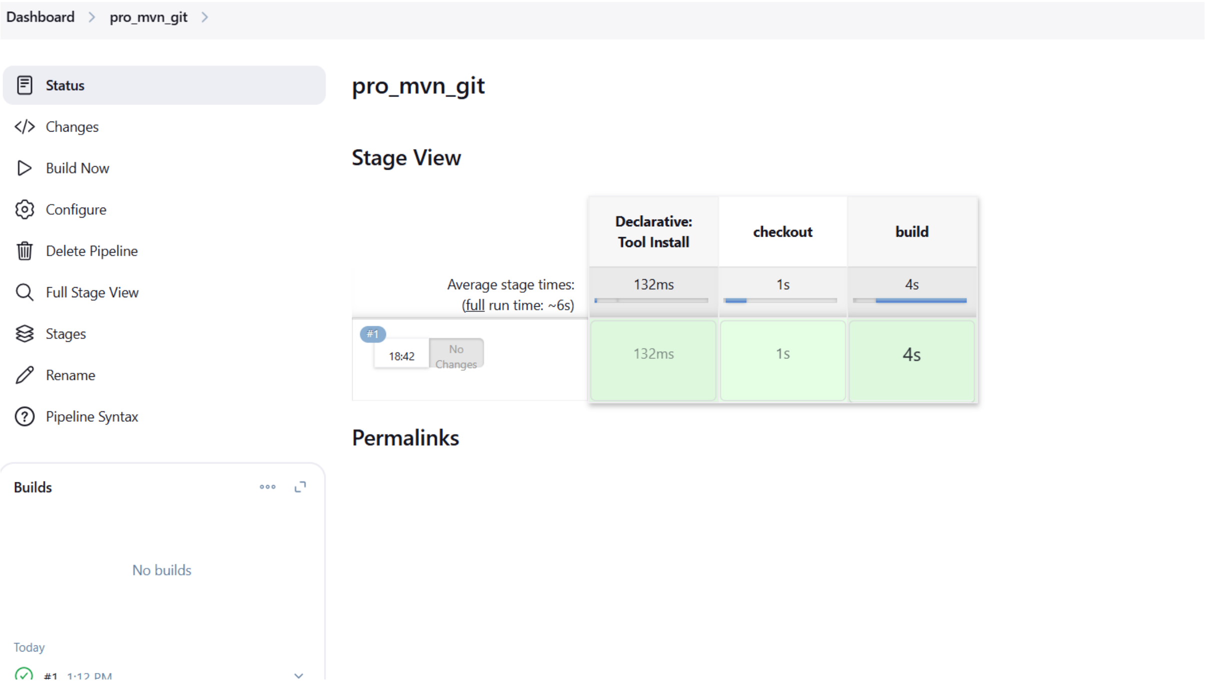
Pipeline script

Script

```
    agent any
    tools{
        maven 'Maven'
    }

    stages {
        stage('checkout') {
            steps {
                checkout scmGit(branches: [[name: '*/main']], extensions: [], userRemoteConfigs: [[credentialsId: 'G
            }
        }
        stage('build') {
            steps {
                sh 'mvn clean package'
            }
```

☑ Use Groovy Sandbox

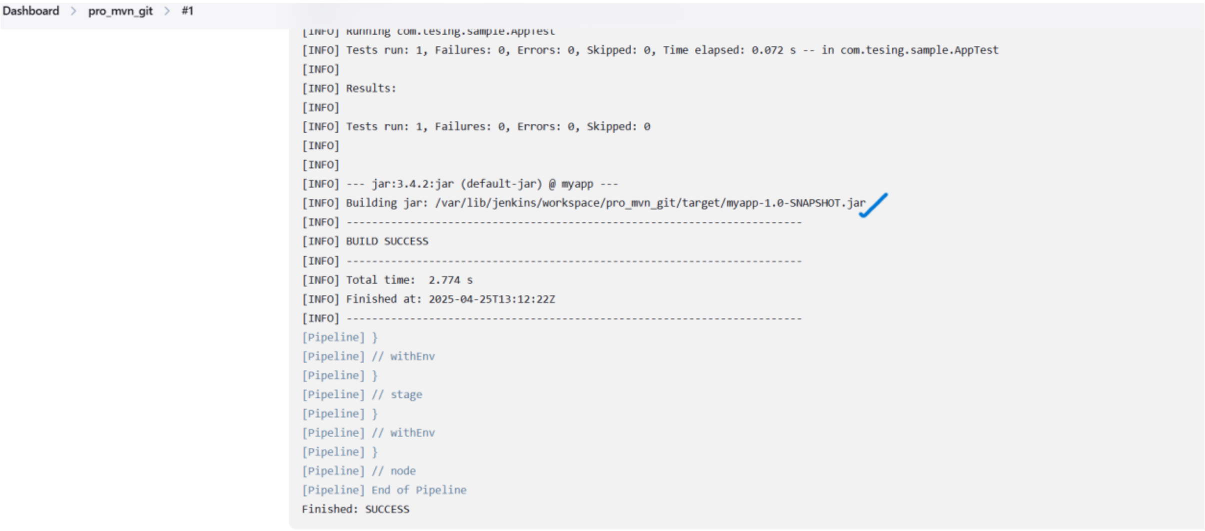


pro_mvn_git

Stage View

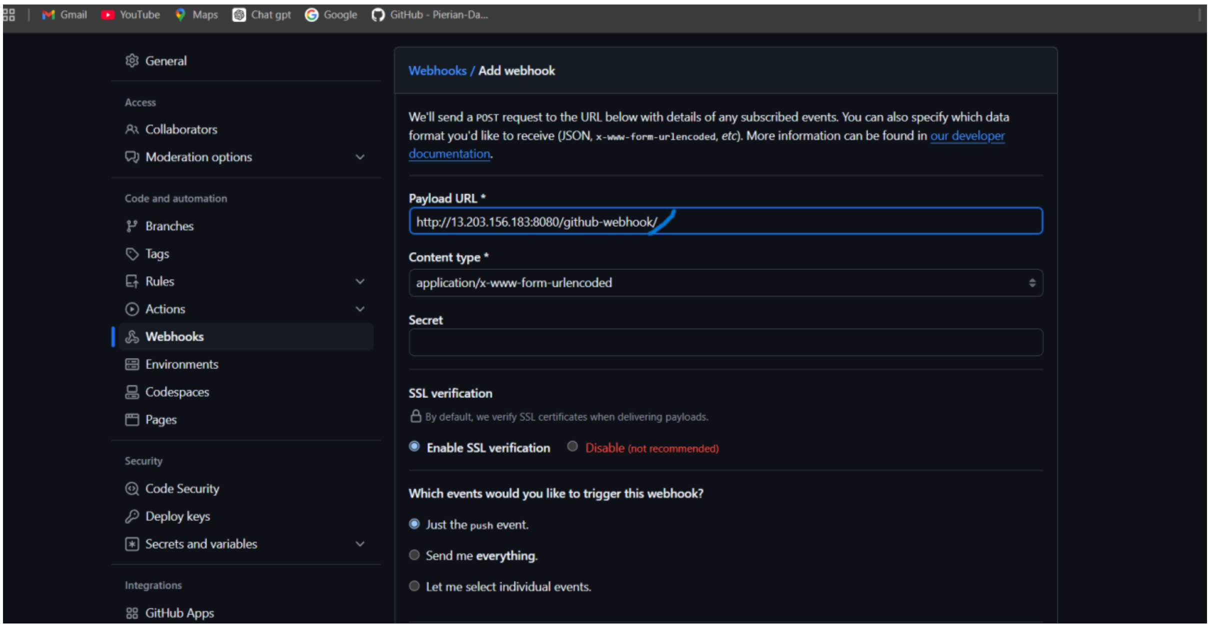
	Declarative: Tool Install	checkout	build
Average stage times: (full run time: ~6s)	132ms	1s	4s
#1 18:42 No Changes	132ms	1s	4s

Permalinks



```
[INFO] Running com.tesing.sample.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.072 s -- in com.tesing.sample.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.2:jar (default-jar) @ myapp ---
[INFO] Building jar: /var/lib/jenkins/workspace/pro_mvn_git/target/myapp-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.774 s
[INFO] Finished at: 2025-04-25T13:12:22Z
[INFO] ------------------------------------------------------------------------
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

To trigger automatic deployments to the server when developers commit changes to GitHub, we need to set up webhooks in GitHub



Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, `x-www-form-urlencoded`, *etc*). More information can be found in our developer documentation.

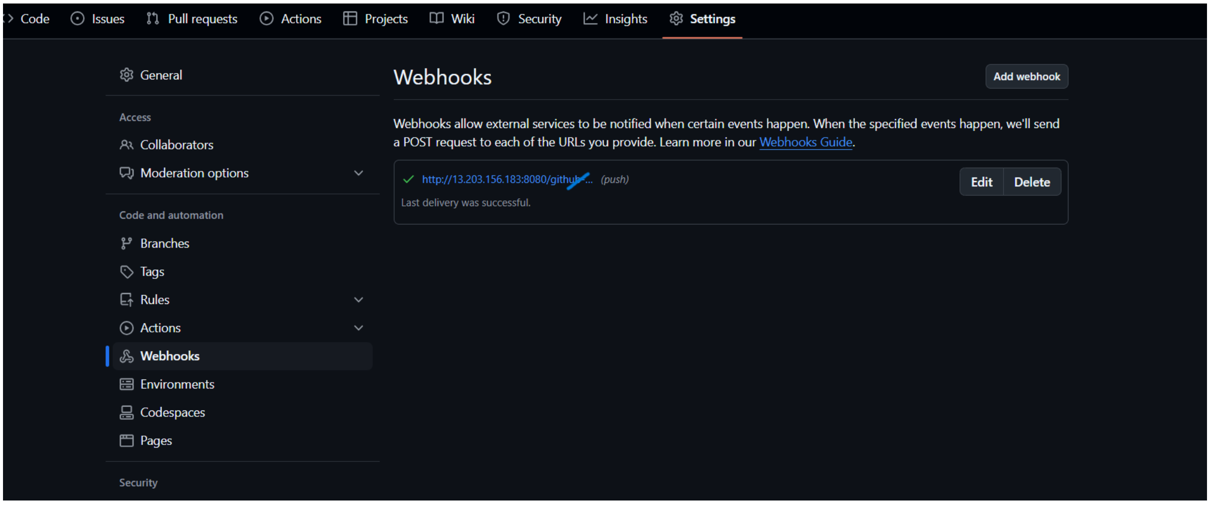
Payload URL *
http://13.203.156.183:8080/github-webhook/

Content type *
application/x-www-form-urlencoded

Secret

SSL verification
By default, we verify SSL certificates when delivering payloads.
● Enable SSL verification ○ Disable (not recommended)

Which events would you like to trigger this webhook?
● Just the push event.
○ Send me **everything**.
○ Let me select individual events.



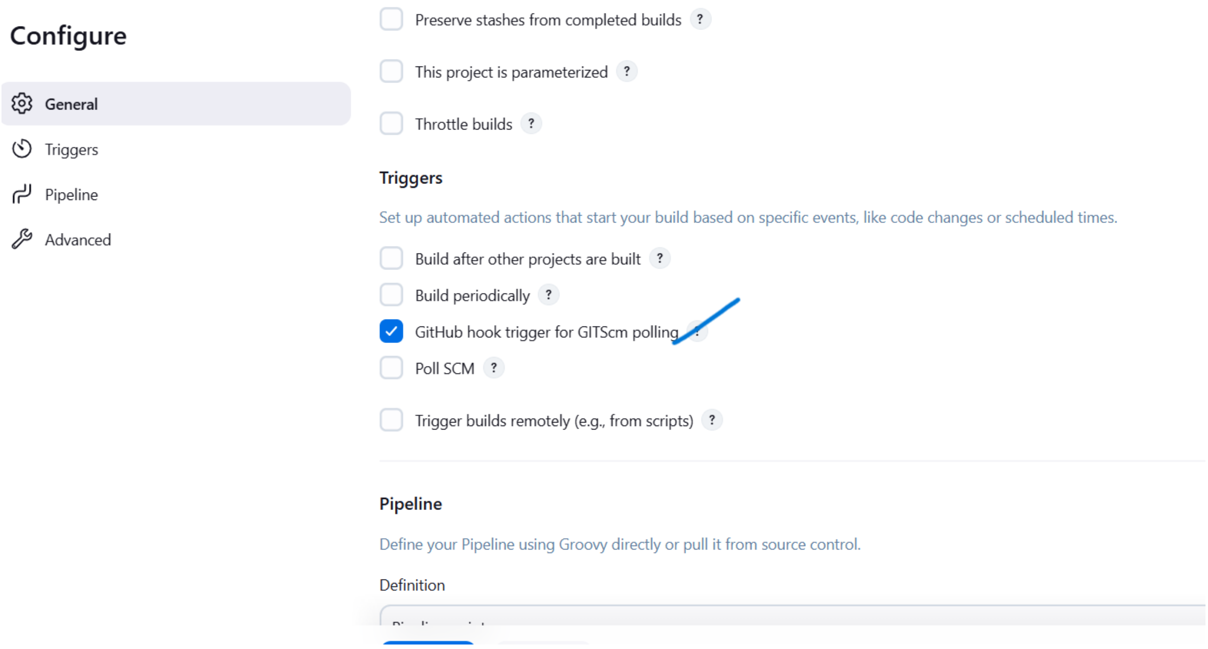
Webhooks

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our Webhooks Guide.

✓ http://13.203.156.183:8080/github... (push)
Last delivery was successful.

Edit Delete



Configure

□ Preserve stashes from completed builds ?

□ This project is parameterized ?

□ Throttle builds ?

Triggers

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

□ Build after other projects are built ?

□ Build periodically ?

☑ GitHub hook trigger for GITScm polling ?

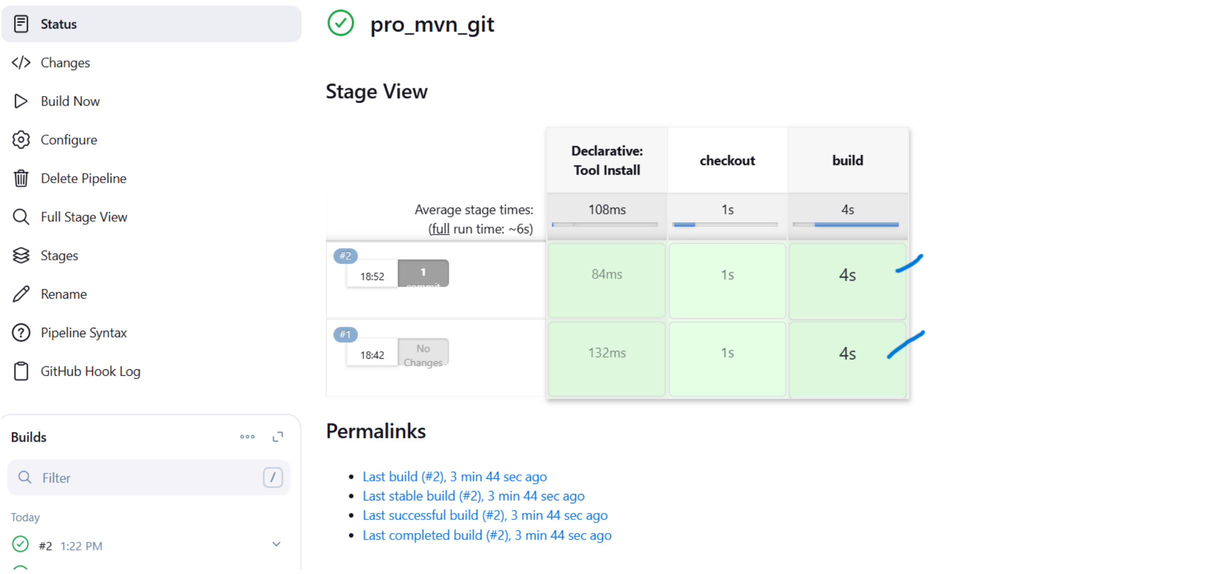
□ Poll SCM ?

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

Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition



pro_mvn_git

Stage View

	Declarative: Tool Install	checkout	build
Average stage times: (full run time: ~6s)	108ms	1s	4s
#2 18:52 1	84ms	1s	4s
#1 18:42 No Changes	132ms	1s	4s

Permalinks

- Last build (#2), 3 min 44 sec ago
- Last stable build (#2), 3 min 44 sec ago
- Last successful build (#2), 3 min 44 sec ago
- Last completed build (#2), 3 min 44 sec ago