# Lab - Continuous Integration with Jenkins

In this lab you will work with Jenkins which is a Continuous Integration (**CI**) tool.

The project supplied is written in Java, but Jenkins' CI pipeline can work for any buildable software.

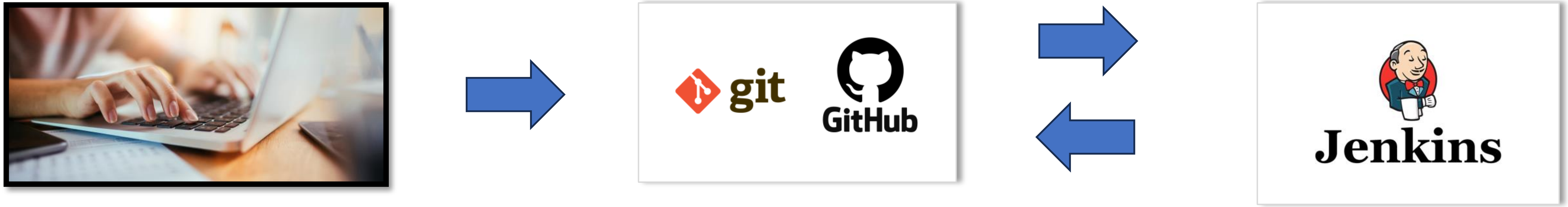
## Objectives

- To get Jenkins up and running using Docker
- To explore the options Jenkins gives us for connecting to other services
- To explore the Jenkins dashboard

### Prerequisites

- Docker desktop is installed
- Git CLI
- You have a GitHub account

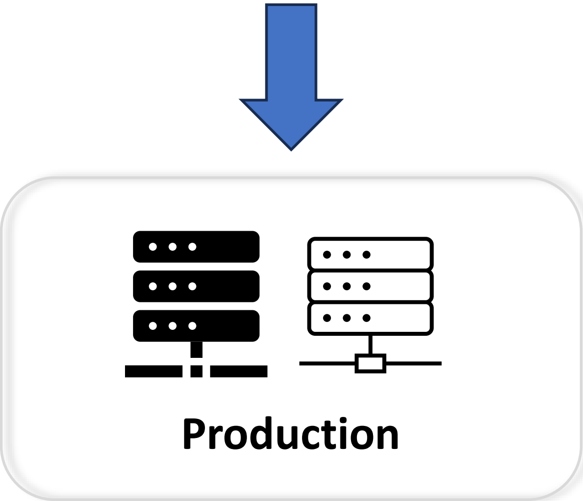
# Objective of this exercise



Write code
Make changes
Check the code in

Detects the change
Raises a build event

Build and runs tests
Deploy the code
Report result to GitHub



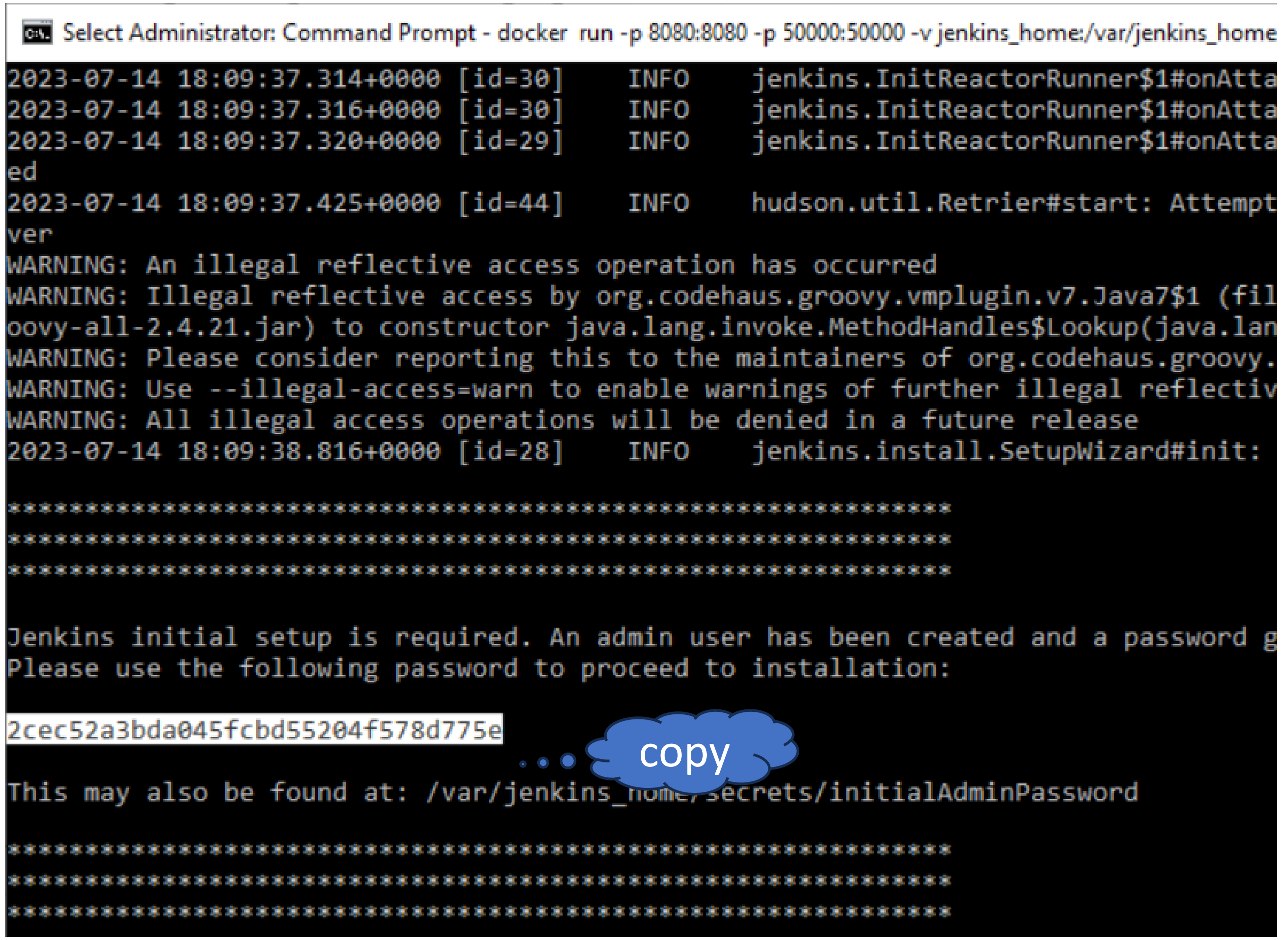
### In this lab, you will

- Deploy a Maven app from GitHub to Jenkins
- Build and test the app using Jenkins
- Make changes to your code and deploy the changes to GitHub
- Set up Jenkins to detect the change in code and rebuild the app

# Run Jenkins using Docker

In a terminal window, type

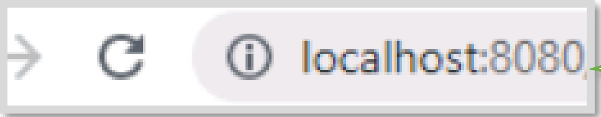
```
docker run -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk11
```



Select Administrator: Command Prompt - docker run -p 8080:8080 -p 50000:50000 -v jenkins_home:/var/jenkins_home

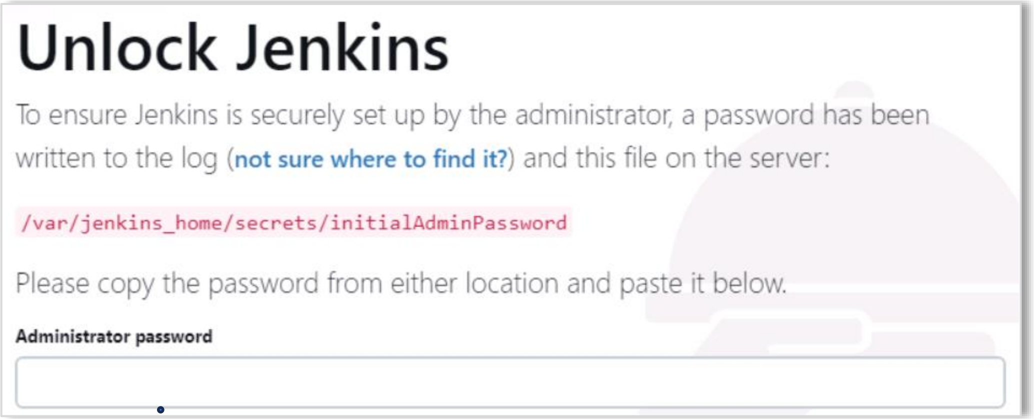
```
2023-07-14 18:09:37.314+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAtta
2023-07-14 18:09:37.316+0000 [id=30]    INFO    jenkins.InitReactorRunner$1#onAtta
2023-07-14 18:09:37.320+0000 [id=29]    INFO    jenkins.InitReactorRunner$1#onAtta
ed
2023-07-14 18:09:37.425+0000 [id=44]    INFO    hudson.util.Retrier#start: Attempt
ver
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.codehaus.groovy.vmplugin.v7.Java7$1 (fil
oovy-all-2.4.21.jar) to constructor java.lang.invoke.MethodHandles$Lookup(java.lan
WARNING: Please consider reporting this to the maintainers of org.codehaus.groovy.
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflectiv
WARNING: All illegal access operations will be denied in a future release
2023-07-14 18:09:38.816+0000 [id=28]    INFO    jenkins.install.SetupWizard#init:

*************************************************************
*************************************************************
*************************************************************

Jenkins initial setup is required. An admin user has been created and a password g
Please use the following password to proceed to installation:

2cec52a3bda045fcbd55204f578d775e

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*************************************************************
*************************************************************
*************************************************************
```

copy



localhost:8080

Run Jenkins in a browser



## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

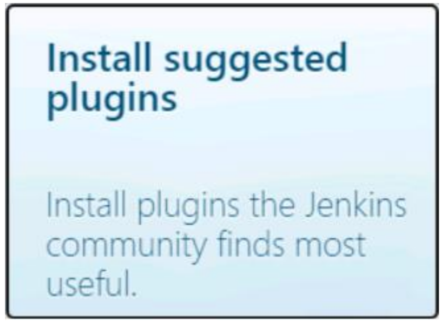
`/var/jenkins_home/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

**Administrator password**

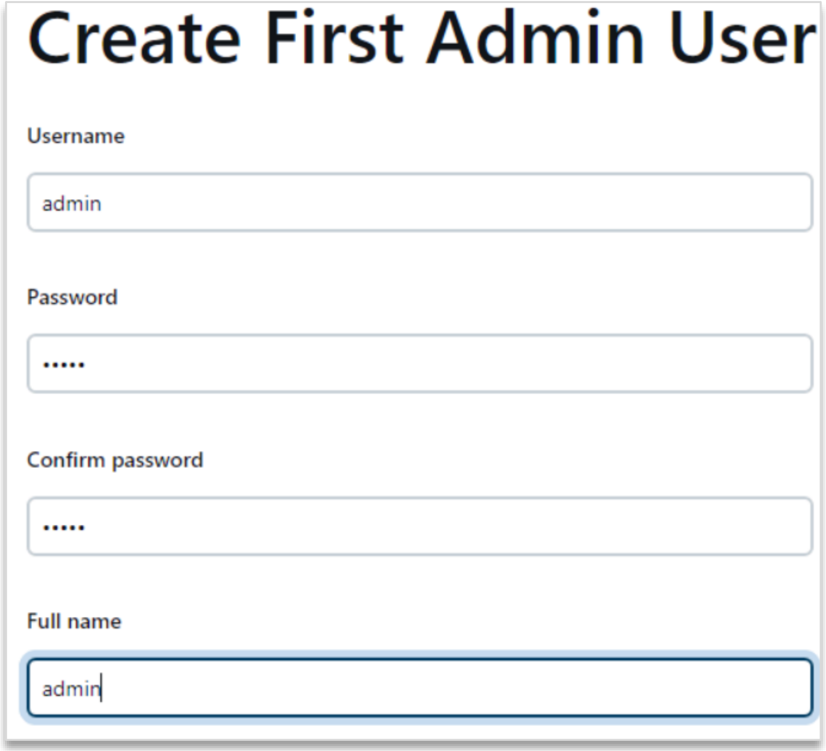
paste

Jenkins



Install suggested plugins

Install plugins the Jenkins community finds most useful.



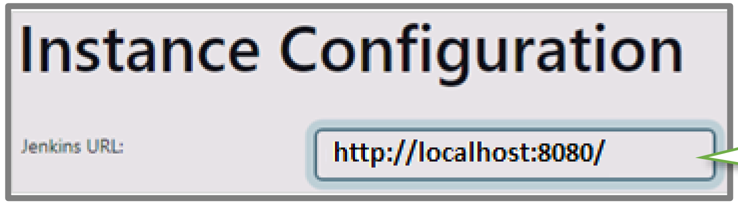
Create First Admin User

Username: admin

Password: •••••

Confirm password: •••••

Full name: admin

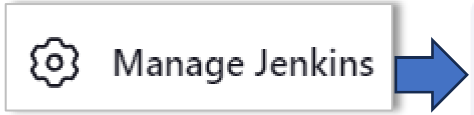


Instance Configuration

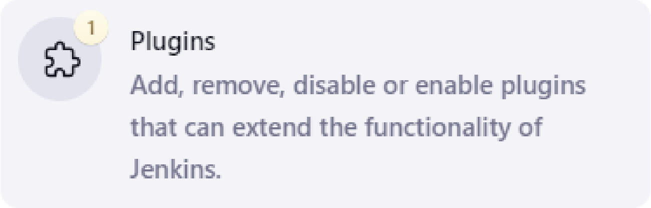
Jenkins URL: http://localhost:8080/

Accept or change the port

# Jenkins

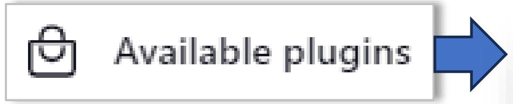


Manage Jenkins

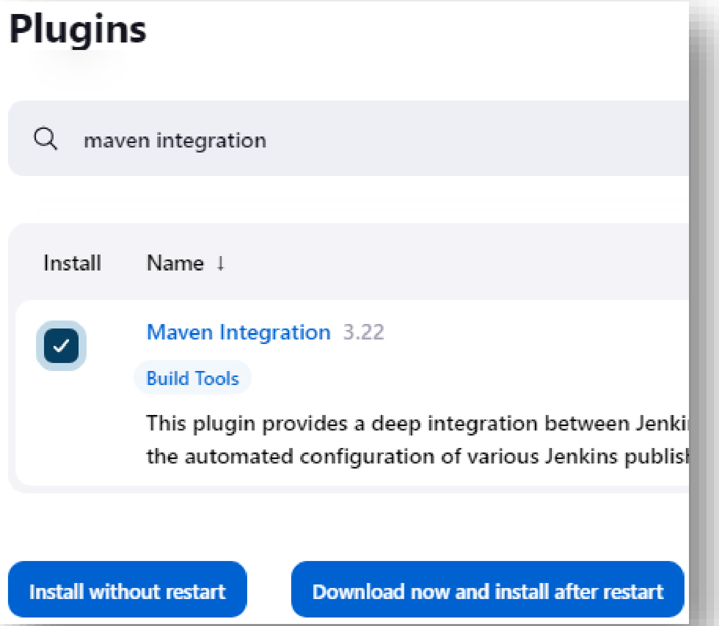


Plugins

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



Available plugins



## Plugins

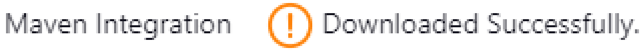
maven integration

| Install | Name ↓ |
|---|---|
| ☑ | Maven Integration 3.22<br>Build Tools<br>This plugin provides a deep integration between Jenki<br>the automated configuration of various Jenkins publis |

Install without restart

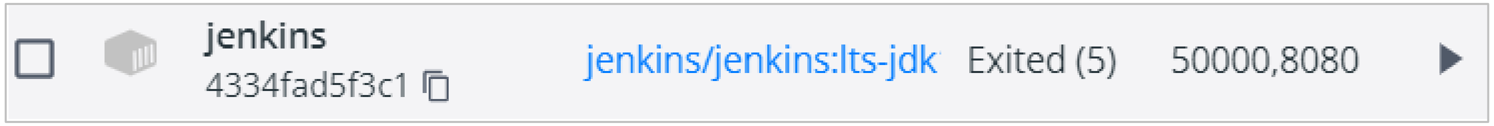
Download now and install after restart

Click



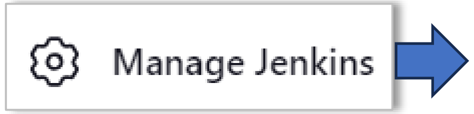
Maven Integration ! Downloaded Successfully.

## Stop and start Jenkins in Docker desktop

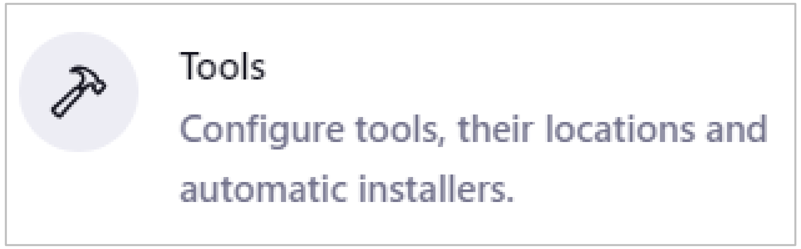


| | | | | | |
|---|---|---|---|---|---|
| □ | jenkins<br>4334fad5f3c1 | jenkins/jenkins:lts-jdk | Exited (5) | 50000,8080 | ▶ |

Jenkins



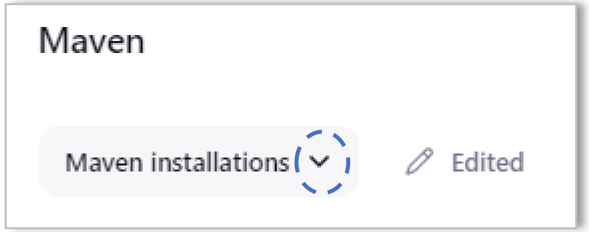
Manage Jenkins



Tools

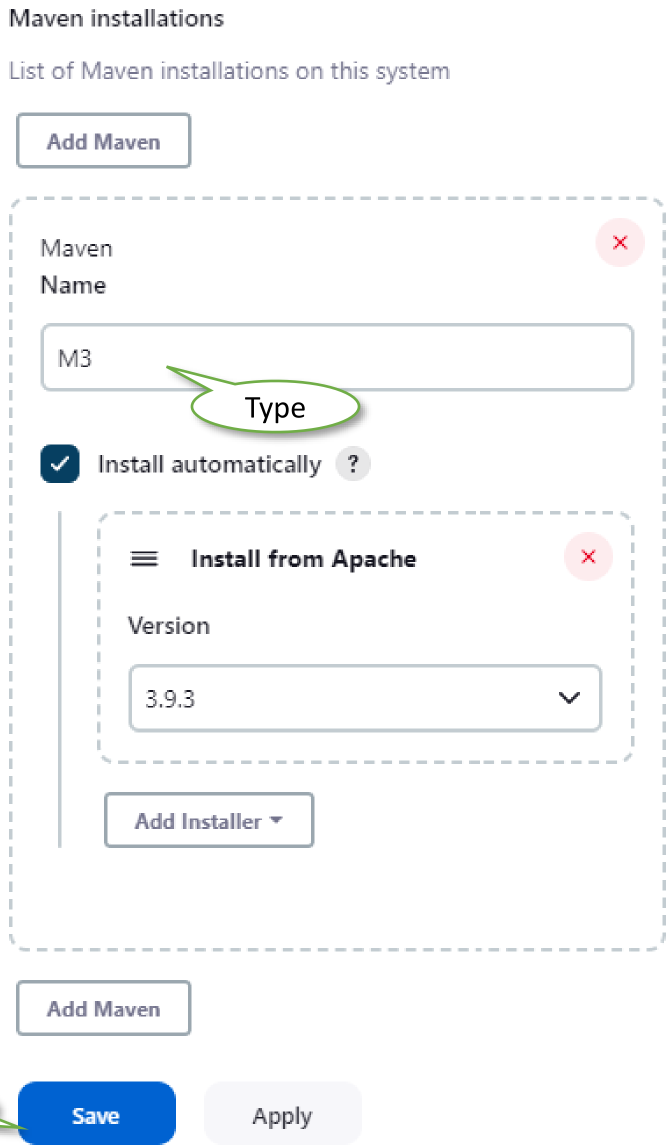
Configure tools, their locations and automatic installers.

**Scroll down until you find the `Maven` section**



Maven

Maven installations ⌄ Edited



Maven installations

List of Maven installations on this system

Add Maven

Maven

Name

M3

Type

☑ Install automatically ?

Install from Apache

Version

3.9.3

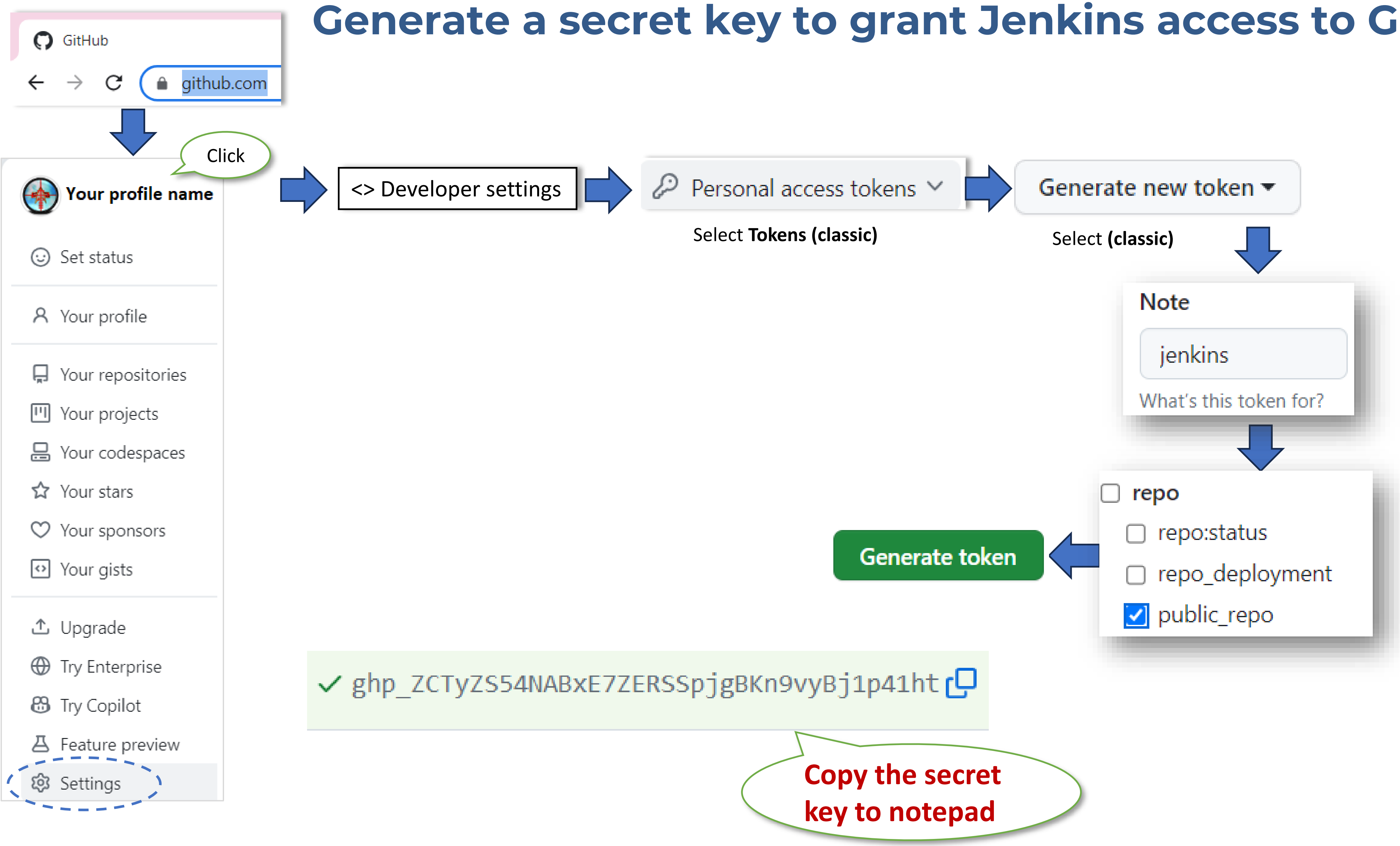
Add Installer ▾

Add Maven

Click

Save Apply

# Generate a secret key to grant Jenkins access to GitHub



Click

Select **Tokens (classic)**

Select **(classic)**

Note

jenkins

What's this token for?

□ repo

□ repo:status

□ repo_deployment

☑ public_repo

✓ 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

Copy the secret key to notepad

# Create a repository for your code

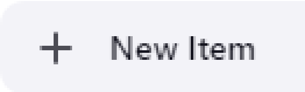
1. Copy the **calculator-jenkins** folder from the **Jenkins** folder to **C:\calculator-Jenkins**
2. Open a terminal window, and navigate to C:\calculator-Jenkins
   `cd c:\calculator-jenkins`
3. Initialise the folder as a new Git repo: **`git init`**
4. Add all of the files: **`git add .`**
5. Commit the files: **`git commit -m "Initial commit"`**
6. Create a new repository in GitHub and call it **calculator-Jenkins**
7. Type **git branch -M main** (in the terminal window)

   **git remote add origin** [https://github.com/…./calculator-jenkins.git](https://github.com/…./calculator-jenkins.git)

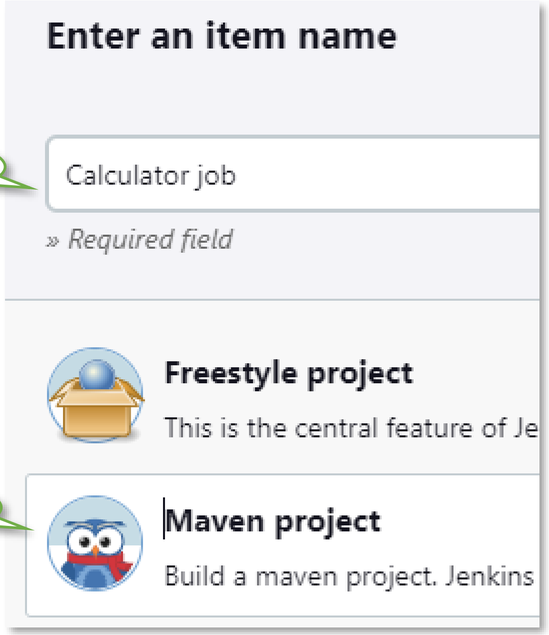
   **git push -u origin main**

Copy the address

# Create a new Jenkins job



+ New Item



Enter an item name

Type

Calculator job

» *Required field*

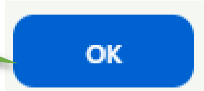
**Freestyle project**

This is the central feature of Je

Select

**Maven project**

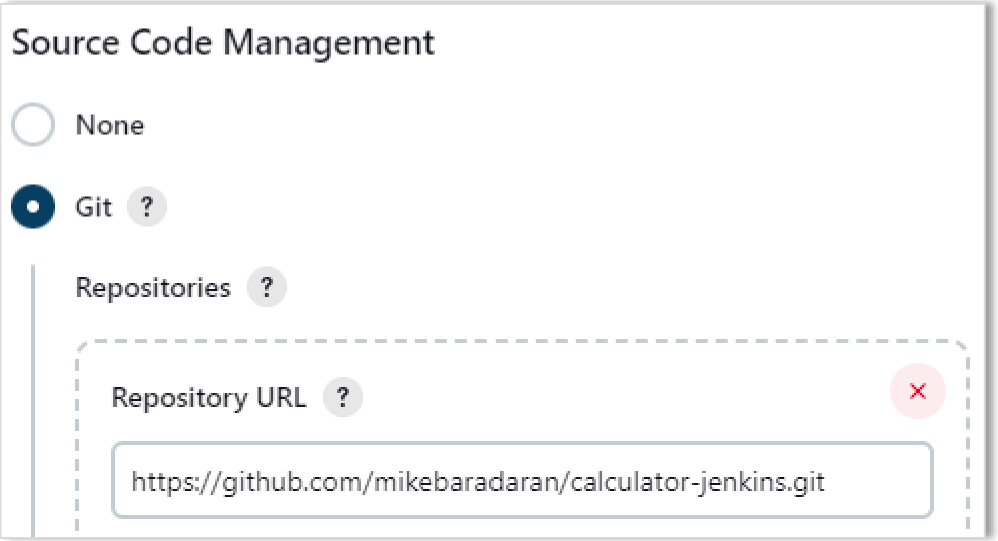
Build a maven project. Jenkins



Click

OK

On the screen below



Source Code Management

○ None

● Git ?

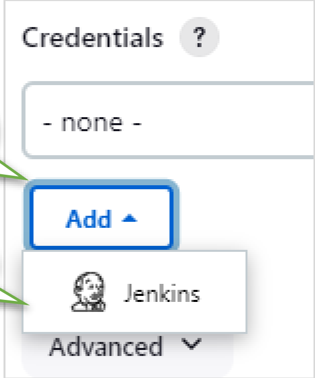
Repositories ?

Repository URL ?

https://github.com/mikebaradaran/calculator-jenkins.git

Continue on the next slide…

Jenkins



Credentials ?

- none -

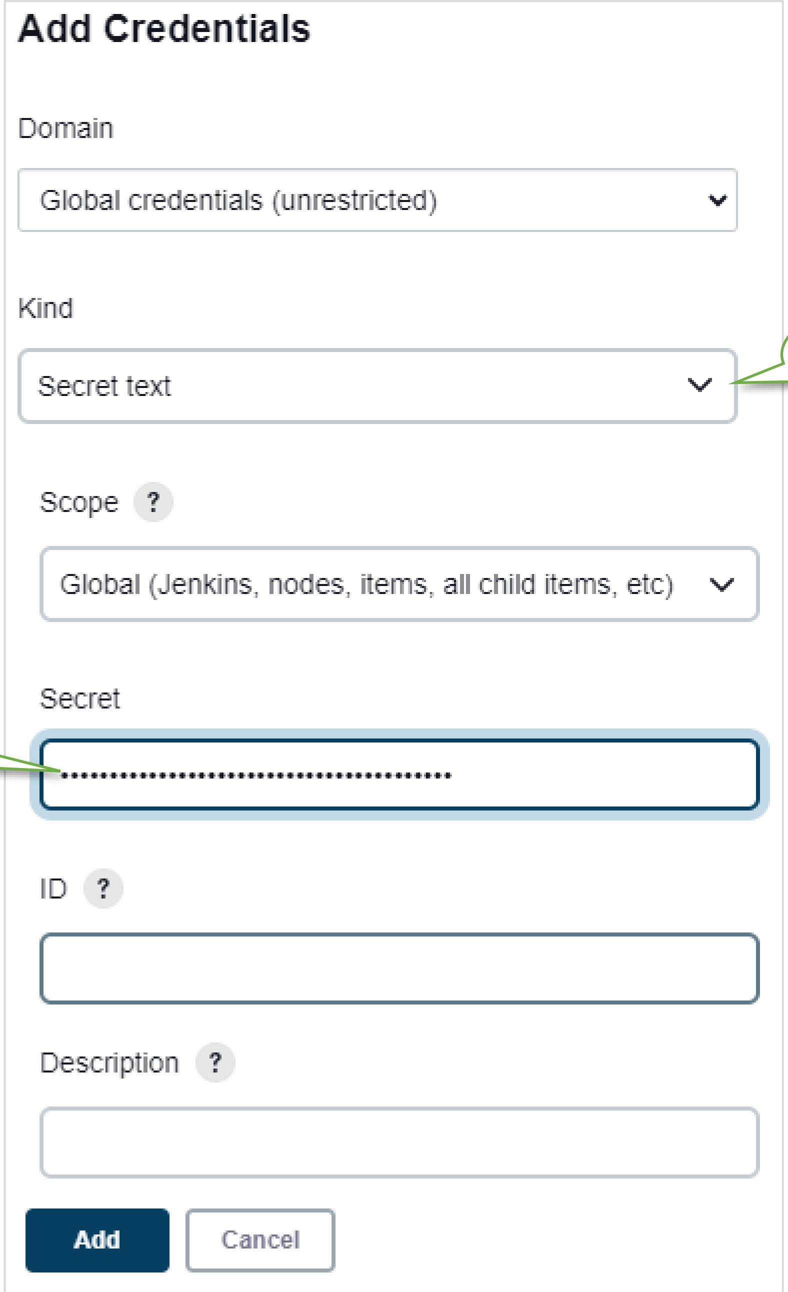
Add ▲

Jenkins

Advanced ⌄

Click the arrow

Then Click



## Add Credentials

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope ?

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

Secret

••••••••••••••••••••••••••••••••••••••••

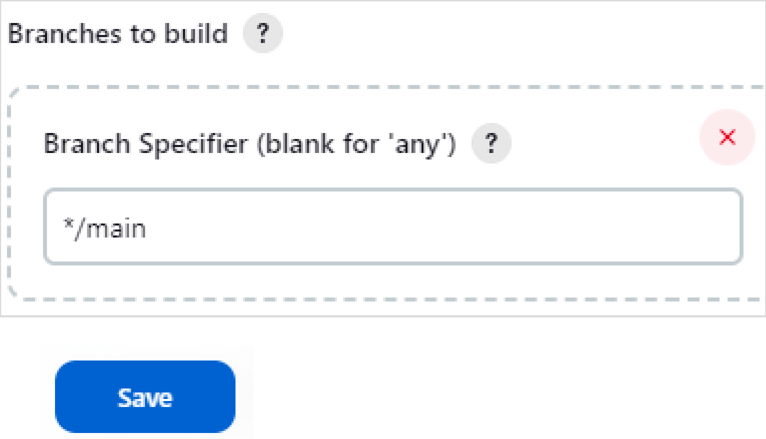
ID ?

Description ?

Add | Cancel

Select

Personal access token from GitHub



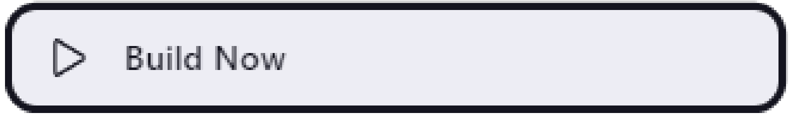
Branches to build ?

Branch Specifier (blank for 'any') ?

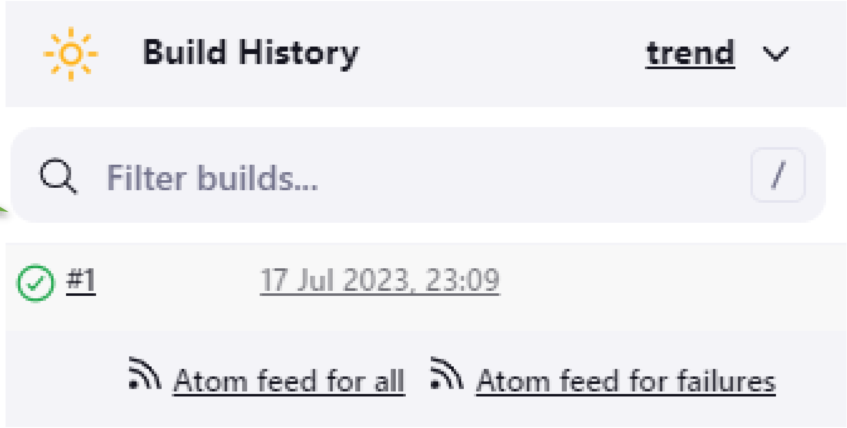
*/main

Save

Jenkins



Build Now



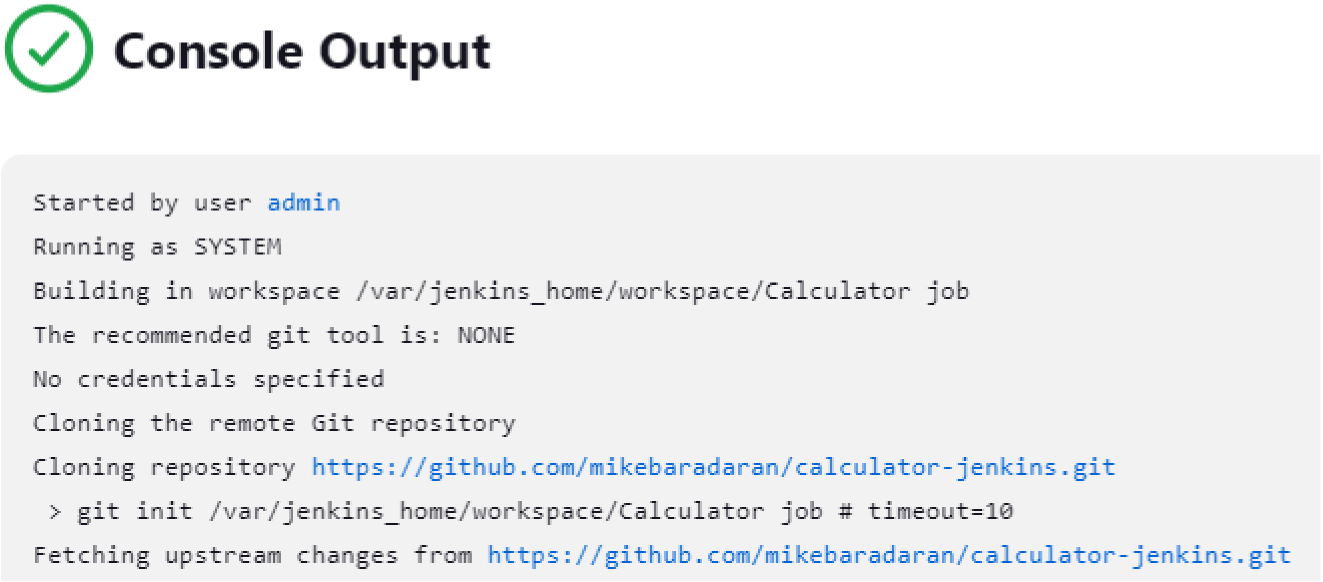
Build History trend

Filter builds...

#1 17 Jul 2023, 23:09

Atom feed for all Atom feed for failures

Click to see log



## Console Output

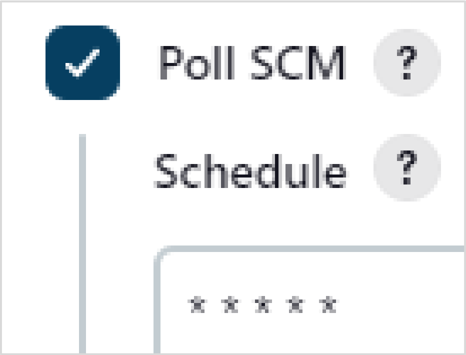
```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/Calculator job
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/mikebaradaran/calculator-jenkins.git
 > git init /var/jenkins_home/workspace/Calculator job # timeout=10
Fetching upstream changes from https://github.com/mikebaradaran/calculator-jenkins.git
```

# Configure Jenkins’ Build Triggers

1. In the **Calculator job** window, select Configure
2. Click on Build Triggers
3. Check the **`Poll SCM`** (Source Control Management) box

   Enter `* * * * *` into the **`Schedule`**

   to build every minute every time a commit is made to this repo



☑ Poll SCM ?

Schedule ?

```
* * * * *
```

Let's test this on the next page

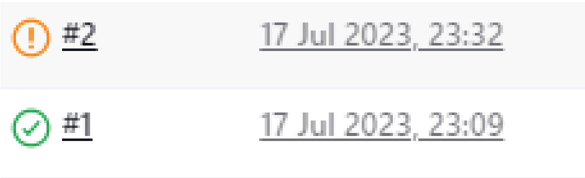
# Run the Trigger (Terminal window)

1. In the C:\ drive `/src/test/java/com/agile/calculator/CalculatorTest.java`
2. Change the value of **`expected`** on line 17 to **`42`** (`makes the test fail`)
3. Save the file
4. Add and commit this file to the repo

```
git add .
git commit –m "change test result"
git push
```

5. **Wait for a minute** (resist the urge to press the Build now button!)

This time the build fails, and the log indicates the reason for this failure



! #2 17 Jul 2023, 23:32

✓ #1 17 Jul 2023, 23:09

Start Jenkins container with volume:

```
docker run -d -v C:/jenkins_calc:/var/jenkins_home/workspace/calculator/host-path jenkins/jenkins
```

Add a Jenkins post build command:

```
cp -r target/* /var/jenkins_home/workspace/example-job/host-path/
```