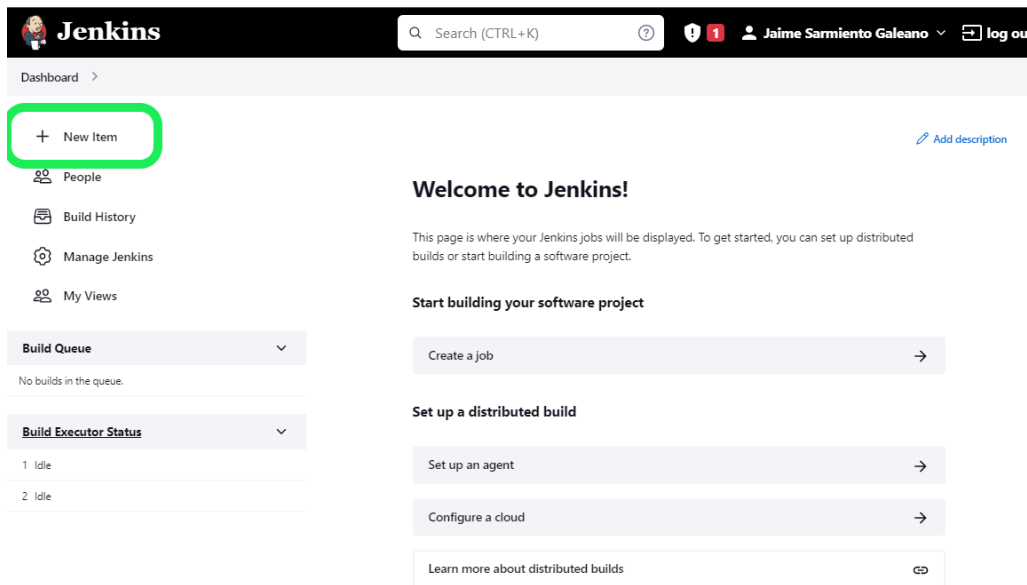


CALCULATOR BUILD - JENKINS STEP BY STEP

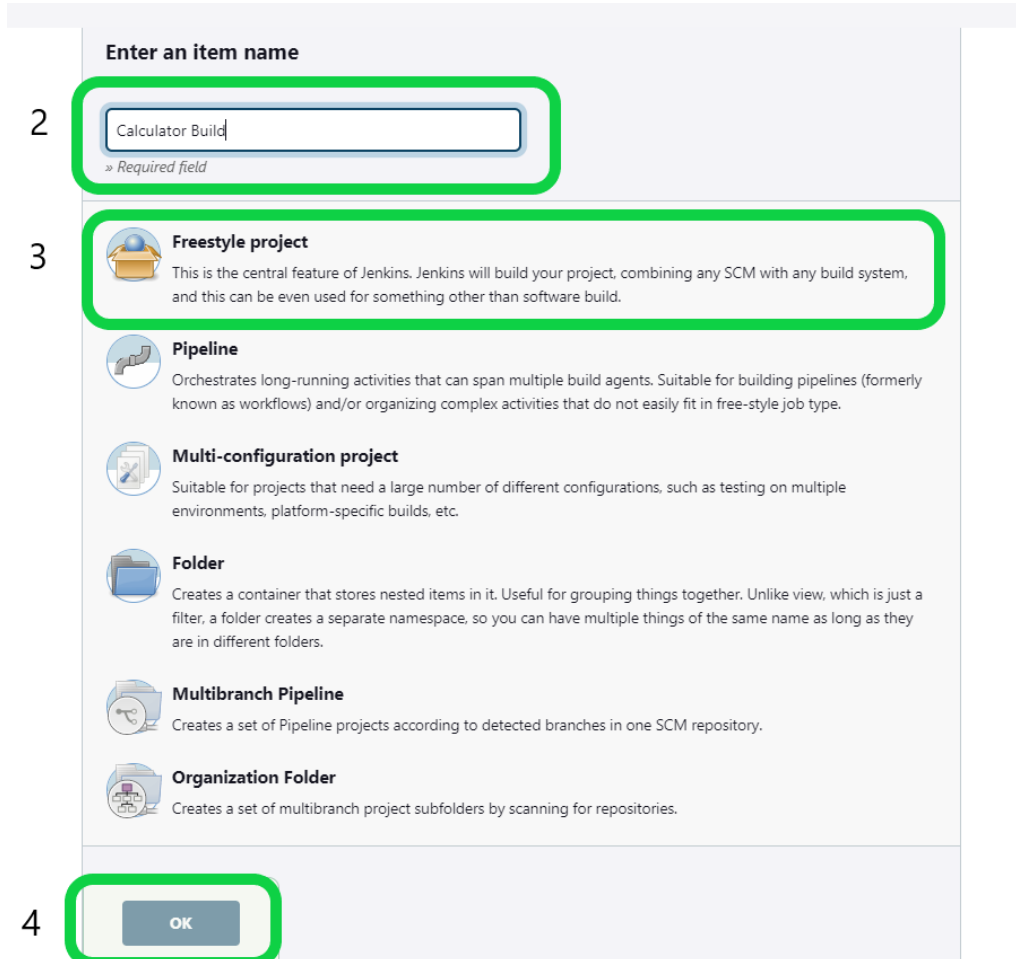
1. Click on New Item to begin.



2. Enter a name for the new item.

3. For this project choose Freestyle project.

4. Click OK.



5. Write the description for the item, in this case, the objectives of the project.

Configuration **General** Enabled

5

Description

Tomando como base el repositorio <https://github.com/vero0311/calculator>

1. Crear un job que haga lo siguiente:

1.1. Descargue el proyecto de la rama master

1.2. Compile el proyecto

1.3. Genere el archivo .jar del proyecto

[Plain text](#) [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

[Advanced...](#)

6. Scroll down and under Source Code Management click on Git, this will display more options.

7. Enter the Git repository that will be cloned.

8. Make sure you are working on the correct branch.

Configuration **Source Code Management**

6

☐ None

☒ Git ?

7

Repository URL ?

<https://github.com/vero0311/calculator>

Please enter Git repository.

Credentials ?

- none -

[+ Add](#)

[Advanced...](#)

[Add Repository](#)

Branches to build ?

8

Branch Specifier (blank for 'any') ?

*/master

9. Scroll down and under Build Steps click on Add build step.

10. From the dropdown menu choose the type of script depending on your preferences, for this example we will use "Execute Windows batch command".

Configuration

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

- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

Build Environment

- ☐ 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

9 Add build step +

10 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

11. In the box that is now active write down the commands needed for the execution. Click save to finish.

Configuration

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

- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant ?

Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
cd src
javac Calculator.java
cd. > Manifest.txt
echo Main-Class: Calculator > Manifest.txt
jar cfm Calculator.jar Manifest.txt Calculator.class
```

Advanced...

Add build step +

Post-build Actions

Add post-build action +

Save Apply

12. In the new screen look for the Build Now option on the left hand menu and Click on it.

13. Below the menu you will see a progress bar while the script runs and a result after it's done, either a green check or a red cross. The green check indicates the build was successful, click on the green check to view the build details.

[↑ Back to Dashboard](#)

[📄 Status](#)

[</> Changes](#)

[📁 Workspace](#)

[▶ Build Now](#)

[⚙️ Configure](#)

[🗑️ Delete Project](#)

[✎ Rename](#)

Project Calculator Build

Tomando como base el repositorio <https://github.com/vero0311/calculator>

1. Crear un job que haga lo siguiente:
 - 1.1. Descargue el proyecto de la rama master
 - 1.2. Compile el proyecto
 - 1.3. Genere el archivo .jar del proyecto

Permalinks

[🔍 Build History](#)
[trend ▼](#)

✓ #1

Nov 1, 2022, 2:29 PM

[Atom feed for all](#)
[Atom feed for failures](#)

14. In the new menu, click on Console Output, this will show every step performed during the build.
15. On the last line of the Console Output you will be able to see if the build was successful, if it was not successful, the Console Output will contain any errors encountered during the build.

[↑ Back to Project](#)

[📄 Status](#)

[</> Changes](#)

[📄 Console Output](#)

[📄 View as plain text](#)

[📄 Edit Build Information](#)

[🗑️ Delete build '#1'](#)

[🔍 Git Build Data](#)

Console Output

```

Started by user Jaime Sarmiento Galeano
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/vero0311/calculator
> git init C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build # timeout=10
Fetching upstream changes from https://github.com/vero0311/calculator
> git --version # timeout=10
> git --version 'git version 2.38.0.windows.1'
> git fetch --tags --force --progress -- https://github.com/vero0311/calculator +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/vero0311/calculator # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision b2ec49664e0aa02b087274ba23f09bd99629c530 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f b2ec49664e0aa02b087274ba23f09bd99629c530 # timeout=10
Commit message: "Calculator project"
First time build. Skipping changelog.
[Calculator Build] $ cmd /c call C:\windows\TEMP\jenkins12534761451185471238.bat

C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build>cd src
C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build\src>javac Calculator.java

C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build\src>cd. 1>Manifest.txt
C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build\src>echo Main-Class: Calculator 1>Manifest.txt
C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build\src>jar cfm Calculator.jar Manifest.txt Calculator.class

C:\ProgramData\Jenkins\jenkins\workspace\Calculator Build\src>exit 0
Finished: SUCCESS

```

TA-DA!!