



Laboratorio 7

Sergio Sebastian Pezo Jimenez - 20224087G

Confirmar la aplicación de muestra a Git

Creamos nuestro repositorio

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner * **Repository name ***

thsergitox / sample-app

sample-app is available.

Great repository names are short and memorable. Need inspiration? How about [literate-invention](#) ?

Description (optional)

Explore CI/CD with GitHub and Jenkins

☐ **Public**
Anyone on the Internet can see this repository. You choose who can commit.

☒ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

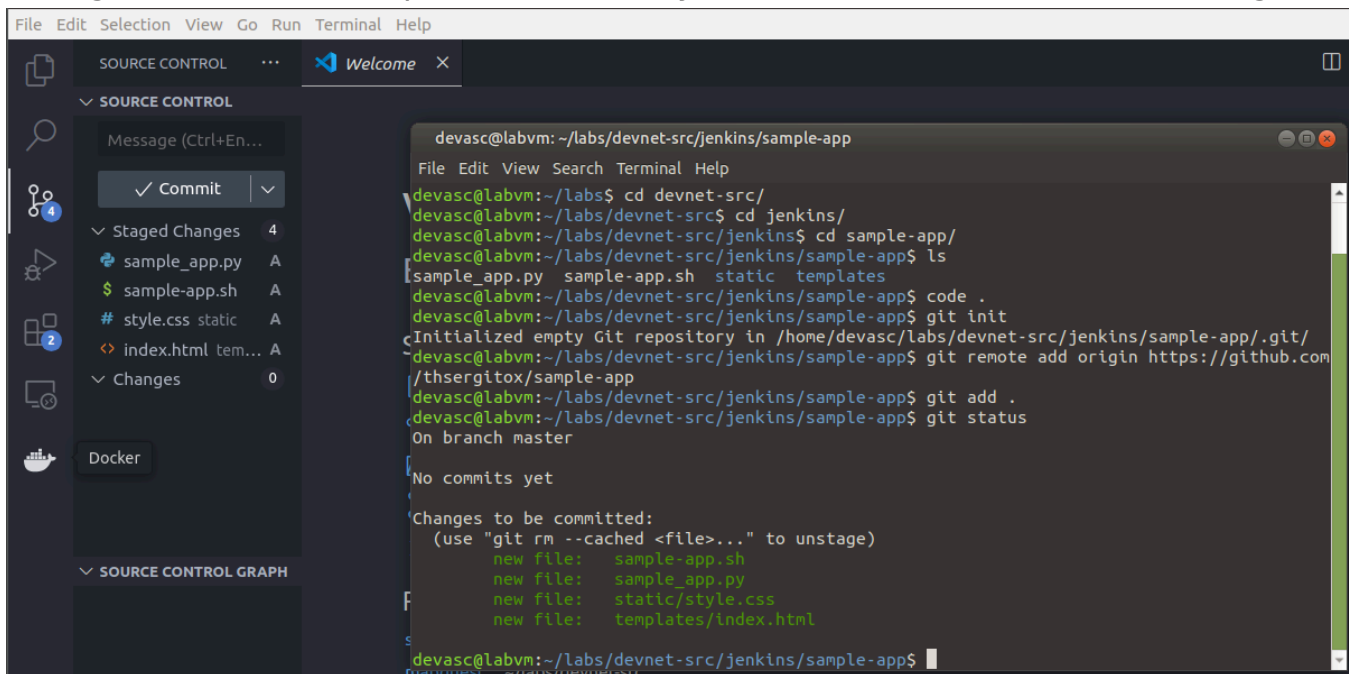
You are creating a private repository in your personal account.

Create repository

Configuramos nuestras credenciales en la VM

```
devasc@labvm: ~  
File Edit View Search Terminal Help  
devasc@labvm:~$ git config --global user.name "thsergitox"  
devasc@labvm:~$ git config --global user.name "sergiopezoj@gmail.com"  
devasc@labvm:~$
```

Configuramos nuestro repositorio remoto y colocamos nuestro archivos al staged.



```
devasc@labvm: ~/labs/devnet-src/jenkins/sample-app  
File Edit View Search Terminal Help  
devasc@labvm:~/labs$ cd devnet-src/  
devasc@labvm:~/labs/devnet-src$ cd jenkins/  
devasc@labvm:~/labs/devnet-src/jenkins$ cd sample-app/  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ ls  
sample_app.py sample-app.sh static templates  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ code .  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git init  
Initialized empty Git repository in /home/devasc/labs/devnet-src/jenkins/sample-app/.git/  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git remote add origin https://github.com/thsergitox/sample-app  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git add .  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git status  
On branch master  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   sample-app.sh  
    new file:   sample_app.py  
    new file:   static/style.css  
    new file:   templates/index.html  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Finalmete lo pusheamos

The screenshot shows a GitHub repository named 'sample-app' by user 'thsergitox'. The repository is private and has 1 branch and 0 tags. A commit by 'sergiopezoj@gmail.com' is shown, committing 'sample-app files.' with 4 files changed and 45 insertions. The files are: static, templates, sample-app.sh, and sample_app.py. Below the commit list is a terminal window showing the commands used to create the files, commit them, and push them to the master branch. The terminal output shows the successful completion of these actions.

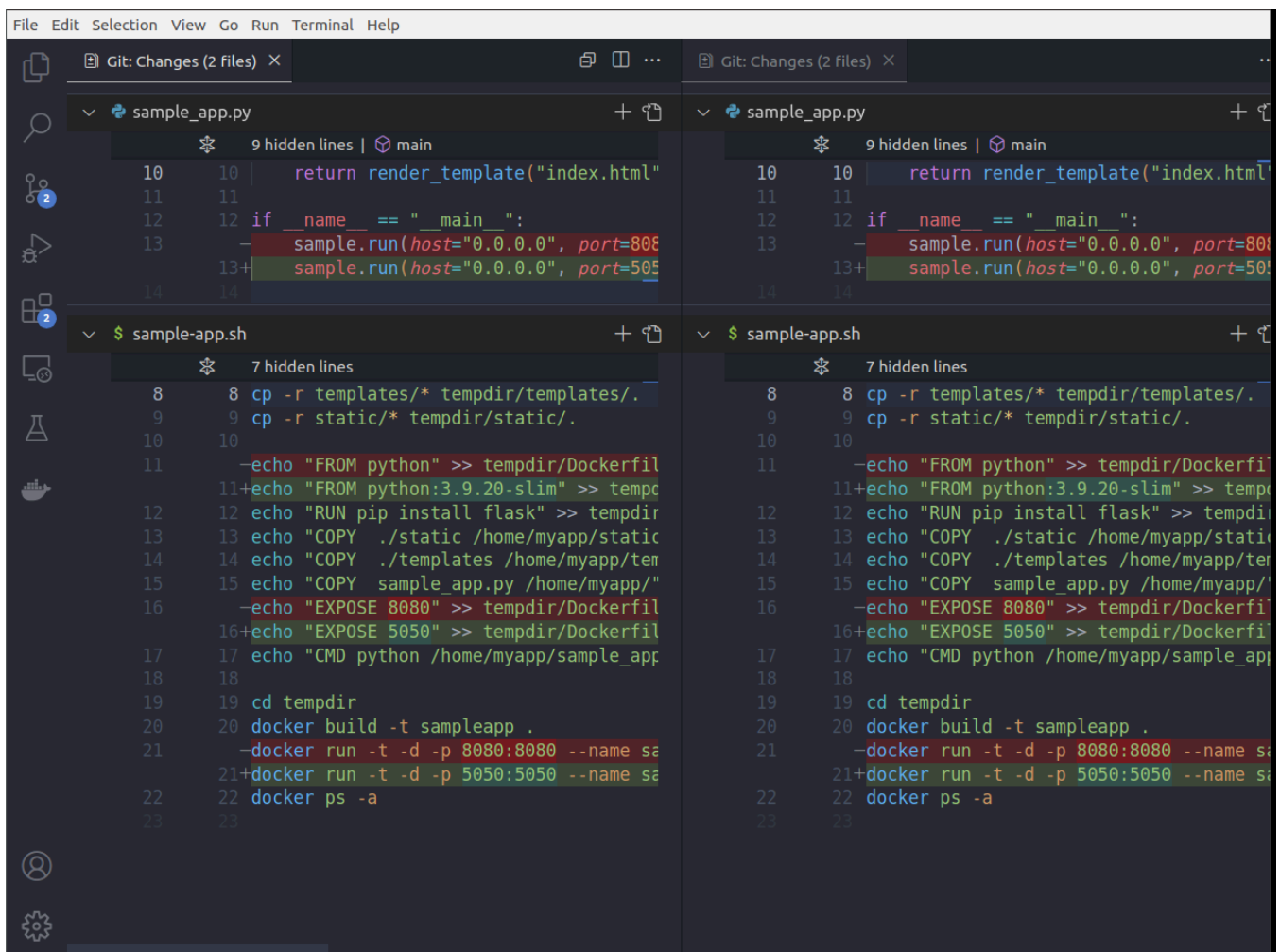
terminal

```
devasc@labvm: ~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help
new file: sample_app.py
new file: static/style.css
new file: templates/index.html

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git commit -m "Committing sample-app files."
[master (root-commit) 4a5d901] Committing sample-app files.
4 files changed, 45 insertions(+)
create mode 100644 sample-app.sh
create mode 100644 sample_app.py
create mode 100644 static/style.css
create mode 100644 templates/index.html
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git push origin master
Username for 'https://github.com': thsergitox
Password for 'https://thsergitox@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.03 KiB | 1.03 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/thsergitox/sample-app
 * [new branch] master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Modificamos los archivos para pushearlos de nuevo

Cambiamos de puerto y versión de python



Construimos nuestra app con docker

Sample app

localhost:5050

You are calling me from 172.17.0.1

```

devasc@labvm: ~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help
--> c6a578bcc8a0
Step 10/14 : COPY ./static /home/myapp/static/
--> Using cache
--> a7f870f4f695
Step 11/14 : COPY ./templates /home/myapp/templates/
--> Using cache
--> 9695d585248a
Step 12/14 : COPY sample_app.py /home/myapp/
--> Using cache
--> 5f209fbe38d3
Step 13/14 : EXPOSE 5050
--> Using cache
--> 0a0926960a26
Step 14/14 : CMD ["python", "/home/myapp/sample_app.py"]
--> Using cache
--> afbc9e22512a
Successfully built afbc9e22512a
Successfully tagged sampleapp:latest
bfdca52211b0e1d49f14a63b7038e58fc6c6ad4398e5282a751860bb80d5979a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS
bfdca52211b0   sampleapp  "python /home/myapp/..." 1 second ago   Up Less than a second
0.0.0.0:5050->5050/tcp, :::5050->5050/tcp  samplerunning
233155d40a8a   9d8cb7037cd8  "/bin/sh -c 'pip ins..." 14 minutes ago   Exited (2) 14 minutes ago
go          festive_napier
0b52039e1a79   9d8cb7037cd8  "/bin/sh -c 'pip ins..." 19 minutes ago   Exited (2) 19 minutes ago
go          confident_cannon
15619e3894a0   a317027204a0  "/bin/sh -c 'pip ins..." 41 minutes ago   Exited (2) 41 minutes ago
go          relaxed_diffie
fb3369efc3b2   aa6659177630  "/bin/sh -c 'pip ins..." 46 minutes ago   Exited (2) 46 minutes ago
go          dreamy_moore
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$

```

Paramos el contenedor

```

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker stop samplerunning
samplerunning

```

Pusheamos todo

```

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git add .
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   requirements.txt
    modified:   sample-app.sh
    modified:   sample_app.py
    new file:   tempdir/Dockerfile
    new file:   tempdir/sample_app.py
    new file:   tempdir/static/style.css
    new file:   tempdir/templates/index.html

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git commit -m "Changed port from 8080 to 5050."
[master 673831e] Changed port from 8080 to 5050.
 7 files changed, 42 insertions(+), 4 deletions(-)
 create mode 100644 requirements.txt
 create mode 100644 tempdir/Dockerfile
 create mode 100644 tempdir/sample_app.py
 create mode 100644 tempdir/static/style.css
 create mode 100644 tempdir/templates/index.html
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git push origin master
Username for 'https://github.com': thsergitox
Password for 'https://thsergitox@github.com':
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 880 bytes | 880.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/thsergitox/sample-app
 4a5d901..673831e  master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$

```

Descargamos y corremos la imagen de Jenkins

Traemos la imagen de la Docker Hub y la iniciamos, en específico la versión 2.414.3-slim-jdk17, pues mi cuenta de GitHub estaba configurada con SSH, y el plugin para Jenkins estaba para esa versión y todo genial de ahí.

```

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker pull jenkins/jenkins:2.271
2.271: Pulling from jenkins/jenkins
3192219afd04: Pull complete
17c160265e75: Pull complete
cc4fe40d0e61: Pull complete
9d647f502a07: Pull complete
d108b8c498aa: Pull complete
1bfe918b8aa5: Pull complete
dafa1a7c0751: Pull complete
a10933ea2f2f: Pull complete
dacec5718df4: Pull complete
0cd1192f374e: Pull complete
bac0875b818f: Pull complete
fc7126ecc5e0: Pull complete
65580027eff4: Pull complete
ea01a82194c6: Pull complete
be9da8492eef: Pull complete
8351e2ec838: Pull complete
909eab257f21: Pull complete
9107ef57f0b5: Pull complete
f925e67af94f: Pull complete
54b9422507ad: Pull complete
Digest: sha256:7648cdca09867d87462a82e6a2aac39942bd2c6deb687da2025cae3f928966cb
Status: Downloaded newer image for jenkins/jenkins:2.271
docker.io/jenkins/jenkins:2.271
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker run --rm -u root -p 8080:8080 -v jenkins-data:/var/jenkins_home -v $(which docker):/usr/bin/docker -v /var/run/docker.sock:/var/run/docker.sock -v "$HOME":/home --name jenkins_server jenkins/jenkins:2.271
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")

```

Verificamos

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

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

2d3e06f5022a46baaedaf4db02539bd4

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

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

2024-09-28 20:55:26.363+0000 [id=26] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2024-09-28 20:55:26.378+0000 [id=20] INFO hudson.WebAppMain$3#run: Jenkins is fully up and running
2024-09-28 20:55:26.734+0000 [id=41] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.task
aven.MavenInstaller
2024-09-28 20:55:26.734+0000 [id=41] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at
attempt #1
2024-09-28 20:55:26.736+0000 [id=41] INFO hudson.model.AsyncPeriodicWork#lambda$doRun$0: Finished Download metadata. 20,295 ms
```

Configuramos Jenkins

The screenshot shows the Jenkins Dashboard in a web browser. The browser address bar shows 'localhost:8080'. The Jenkins logo is on the left, and a search bar is in the center. On the right, there are notification icons (2), a warning icon (2), and a user profile 'admin' with a 'log out' button. The left sidebar contains a 'Dashboard' link and a list of items: 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below this, there are two expandable sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing two 'Idle' executors). The main content area has a 'Welcome to Jenkins!' heading, followed by a paragraph explaining the dashboard's purpose. Below this, there are three sections: 'Start building your software project' with a 'Create a job' button; 'Set up a distributed build' with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

Dashboard >

- + New Item
- Build History
- Manage Jenkins
- My Views

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

Descargamos todos los plugins recomendados

← → ↻ localhost:8080/pluginManager/updates/ ☆ | 🔍 | 👤 | ⚙️

Jenkins 🔍 Search (CTRL+K) 🔍 🔔 2 🛡️ 2 👤 admin ▾ ➡️ log out

Dashboard > Plugins

Plugins

- 📁 Updates
- 📁 Available plugins
- ⚙️ Installed plugins
- ⚙️ Advanced settings
- ☰ Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity

| | |
|-------------------------|-----------|
| Ant | ⋮ Pending |
| commons-text API | ⋮ Pending |
| ASM API | ⋮ Pending |
| SCM API | ⋮ Pending |
| Pipeline: API | ⋮ Pending |
| Plugin Utilities API | ⋮ Pending |
| Font Awesome API | ⋮ Pending |
| Bootstrap 5 API | ⋮ Pending |
| Branch API | ⋮ Pending |
| Build Timeout | ⋮ Pending |
| Checks API | ⋮ Pending |
| commons-text API | ⋮ Pending |
| Credentials | ⋮ Pending |
| Durable Task | ⋮ Pending |
| JQuery3 API | ⋮ Pending |
| ECharts API | ⋮ Pending |
| Pipeline: Job | ⋮ Pending |
| Email Extension | ⋮ Pending |
| Folders | ⋮ Pending |
| Font Awesome API | ⋮ Pending |
| Mina SSHD API :: Common | ⋮ Pending |

Menu: 🏠 devops@labvm: ~/labs 🔍 sample-app-ch sample-app-ch Download progress - PL

Usamos Jenkins para ejecutar un build de nuestra app

Creamos un item llamado BuildAppJob

← → ↻ localhost:8080 ☆ | 🔍 | 👤 | ⚙️

Jenkins 🔍 Search (CTRL+K) 🔍 🔔 4 🛡️ 2 👤 admin ▾ ➡️ log out

Dashboard >

+ New Item Add description

👤 People

📁 Build History

🔗 Project Relationship

🔍 Check File Fingerprint

⚙️ Manage Jenkins

📁 My Views

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|---|-------------|-----------------|--------------|---------------|
| ✓ | ☁ | BuildAppJob | 4 min 29 sec #4 | 16 min #3 | 4.3 sec |

Icon: S W M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

Build Queue

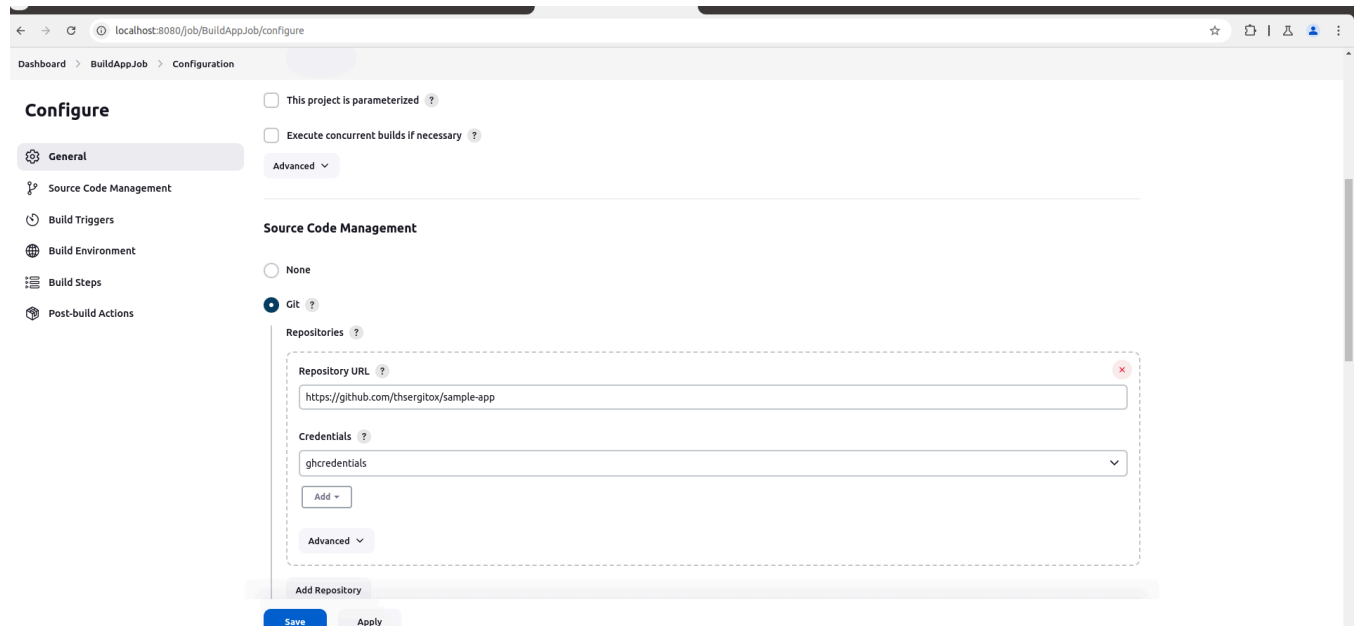
No builds in the queue.

Build Executor Status

1 Idle

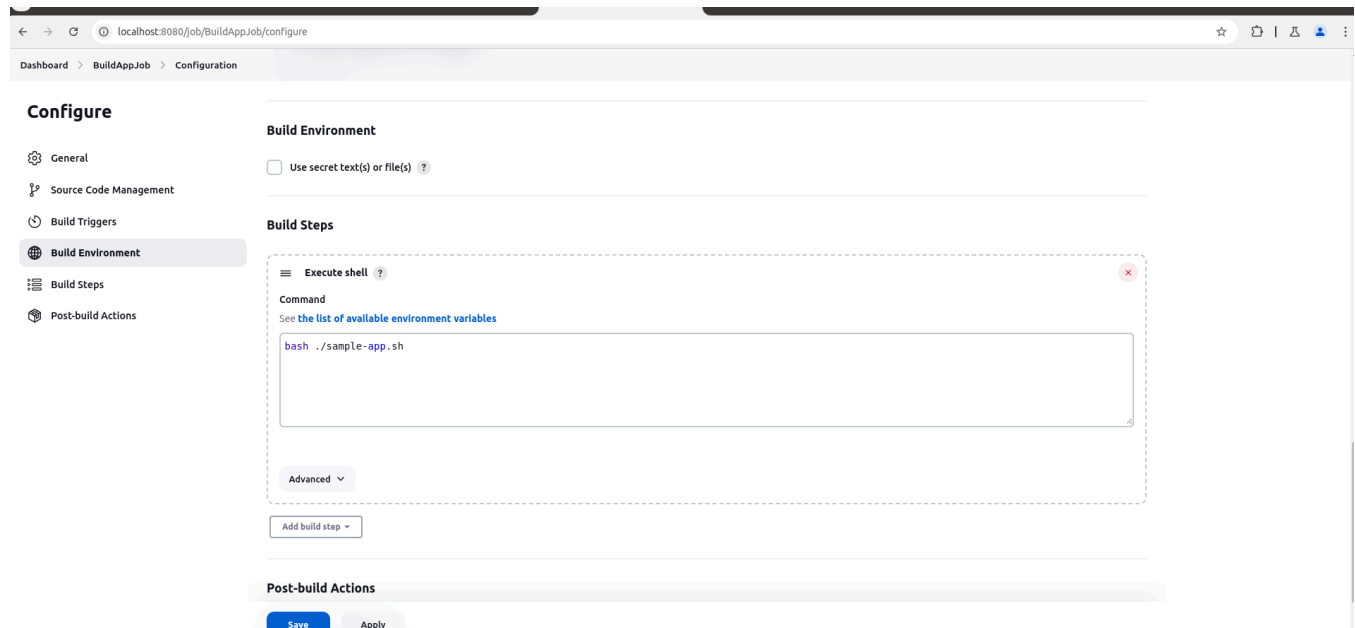
2 Idle

Agregamos nuestras credenciales para conectarnos con GitHub y nuestro repositorio.



The screenshot shows the Jenkins configuration page for a job named 'BuildAppJob'. The 'Source Code Management' section is active, showing the 'Git' provider selected. A repository is configured with the URL 'https://github.com/thsergitox/sample-app' and credentials 'ghcredentials'. The 'Advanced' dropdown is visible. The left sidebar shows the configuration menu with 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'.

Agregamos un **build step**

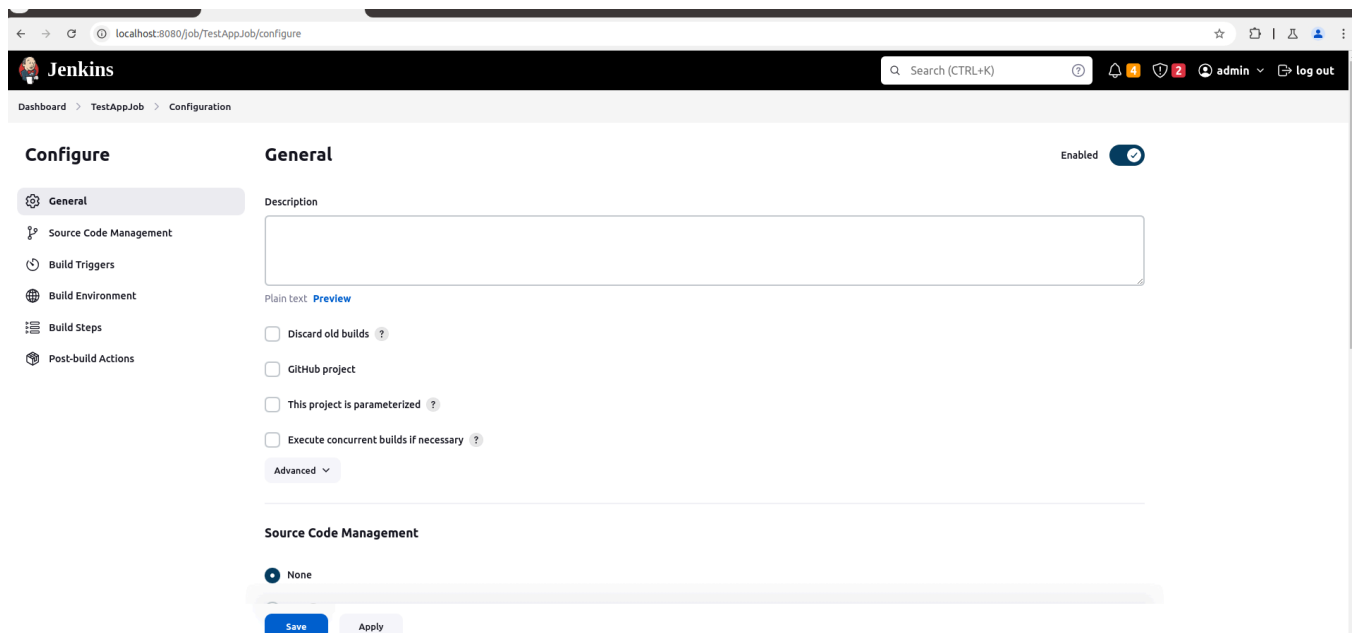


The screenshot shows the Jenkins configuration page for the same job, now in the 'Build Steps' section. A 'Execute shell' build step has been added with the command 'bash ./sample-app.sh'. The 'Advanced' dropdown is visible. The left sidebar shows the configuration menu with 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'.

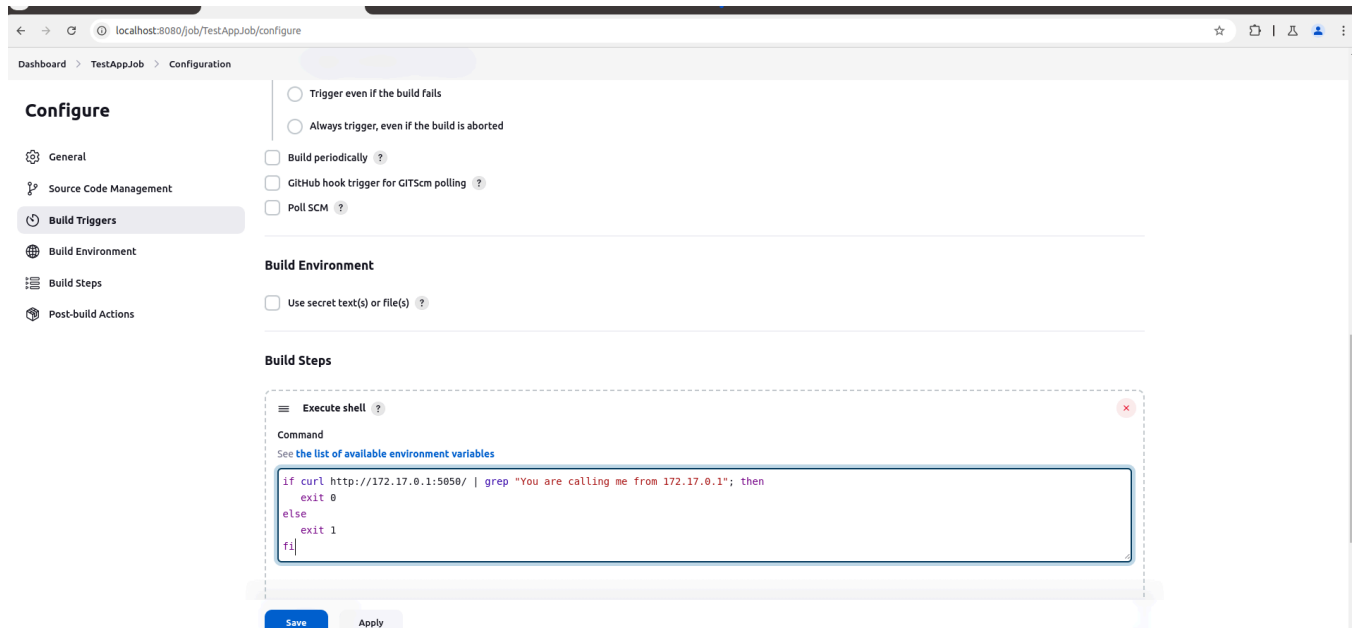
Realizamos un Build y lo ejecutó a la perfección

Usamos Jenkins para testear el Build

Creamos un nuevo item llamado **TestAppJob**



Configuramos nuestro nuevo item.



Aqui tenemos los dos

The screenshot shows the Jenkins Dashboard at localhost:8080. The left sidebar contains links for New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area displays a table of builds with columns for Status (S), Waiver (W), Name, Last Success, Last Failure, and Last Duration. Two builds are listed: BuildAppJob (Last Success: 12 min #8, Last Failure: 1 hr 3 min #3, Last Duration: 1.5 sec) and TestAppJob (Last Success: N/A, Last Failure: N/A, Last Duration: N/A). Below the table, there are links for Icon legend, Atom feed for all, Atom feed for failures, and Atom feed for just latest builds. On the left, the Build Queue section shows 'No builds in the queue.' and the Build Executor Status section shows two idle executors.

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|---|-------------|--------------|---------------|---------------|
| ✓ | ☀ | BuildAppJob | 12 min #8 | 1 hr 3 min #3 | 1.5 sec |
| ⋮ | ☀ | TestAppJob | N/A | N/A | N/A |

REST API Jenkins 2.414.3

Ejecutamos el build de BuilAppJob y se ejecutó con éxito, tanto ese como el test.

| S | W | Name | Last Success | Last Failure | Last Duration |
|---|---|-------------|-----------------|---------------|---------------|
| ✓ | ☀ | BuildAppJob | 2 min 28 sec #9 | 1 hr 7 min #3 | 1.9 sec |
| ✓ | ☀ | TestAppJob | 2 min 17 sec #1 | N/A | 34 ms |

The screenshot shows the Jenkins Console Output for TestAppJob #1. The breadcrumb navigation shows Dashboard > TestAppJob > #1 > Console Output. The console output text is as follows:

```

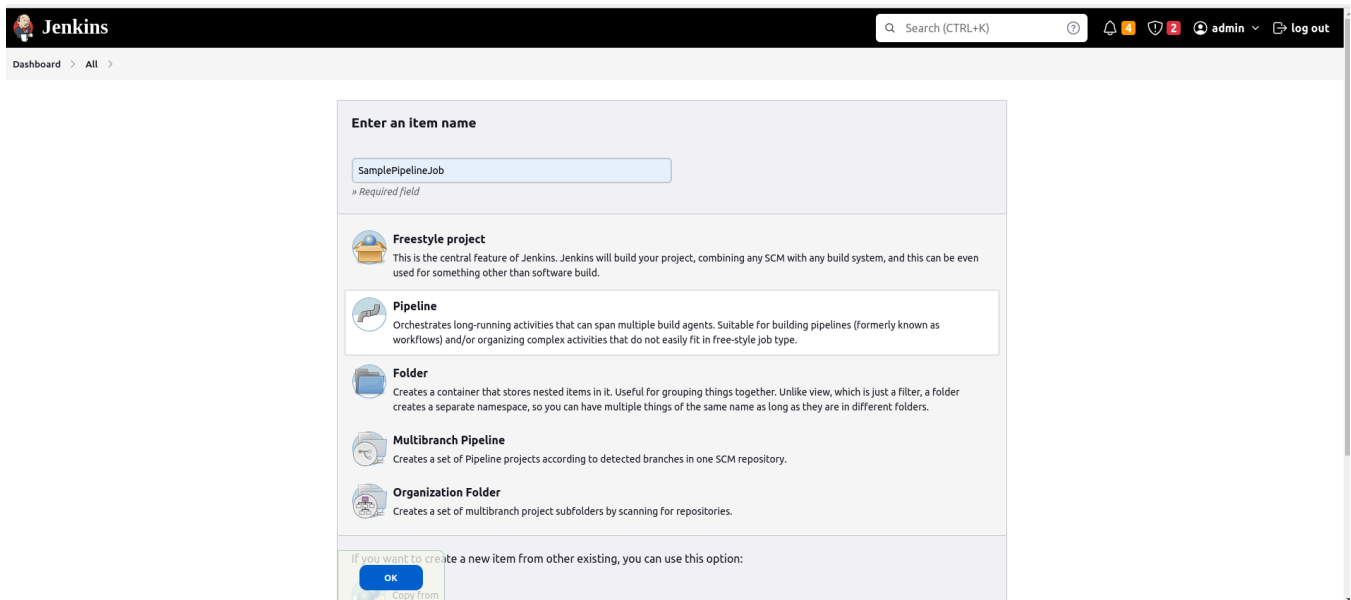
Started by upstream project "BuildAppJob" build number 9
originally caused by:
  Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/TestAppJob
[TestAppJob] $ /bin/sh -xe /tmp/jenkins1314035125895492401.sh
+ curl http://172.17.0.1:5050/
+ grep You are calling me from 172.17.0.1
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
  0     0    0     0    0     0      0      0  0 --:--:-- --:--:-- --:--:--  0
100  177  100  177    0    0  47140    0 --:--:-- --:--:-- --:--:-- 59000
<h1>You are calling me from 172.17.0.1</h1>
+ exit 0
Finished: SUCCESS

```

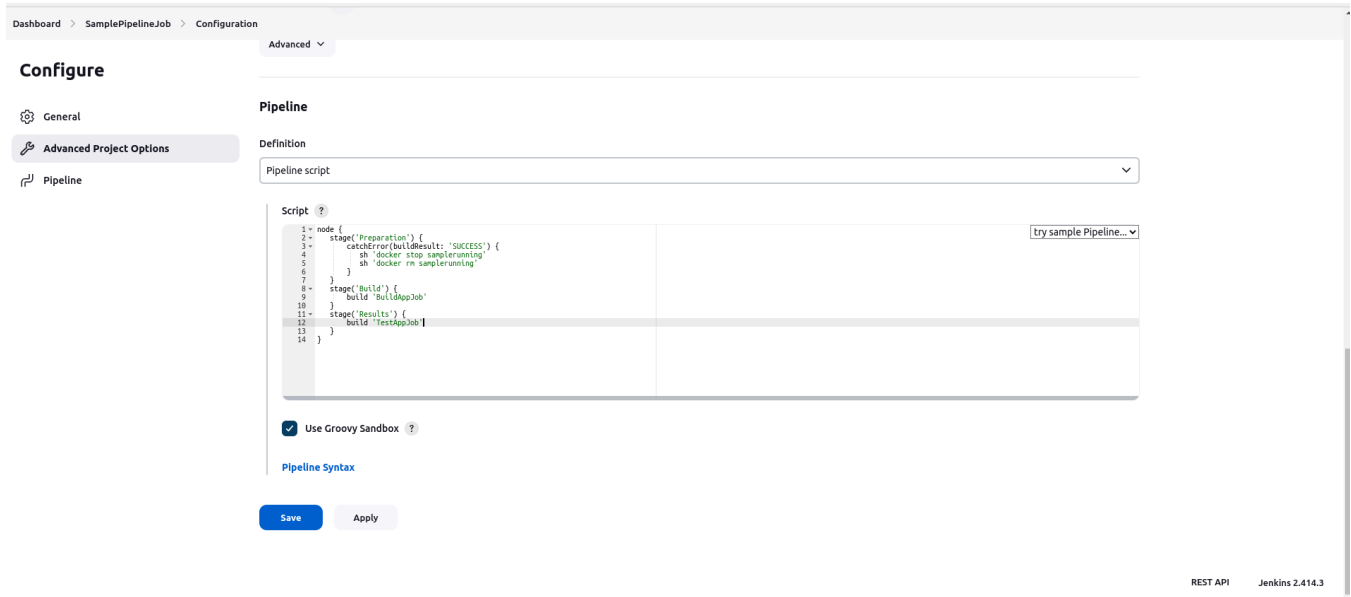
REST API Jenkins 2.414.3

Creamos un pipeline job

Creamos un nuevo Job de tipo pipeline, instalando el plugin Pipeline.



Agregamos el pipeline script



Le damos click a Build Now y funciona

Jenkins

Dashboard > SamplePipelineJob

Pipeline SamplePipelineJob

Permalinks

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

Build History

| #1 | 52 sec ago |
|-------------------|------------------------|
| Atom feed for all | Atom feed for failures |

REST API Jenkins 2.414.3

Entramos al último Permalink y vemos como está el console output, y todo SUCCESS

Dashboard > SamplePipelineJob > #1

Replay

```
[Pipeline] sh
+ docker stop samplerunning
samplerunning
[Pipeline] sh
+ docker rm samplerunning
samplerunning
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] build (Building BuildAppJob)
Starting building: BuildAppJob #10
Build BuildAppJob #10 completed: SUCCESS
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Results)
[Pipeline] build (Building TestAppJob)
Starting building: TestAppJob #2
Build TestAppJob #2 completed: SUCCESS
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.414.3

Finalizado.

Conclusiones

- Se realizaron modificaciones en la aplicación, incluyendo cambios en el puerto y la versión de Python, y se construyó la aplicación usando Docker.
- Se descargó y configuró Jenkins en una imagen de Docker, incluyendo la instalación de plugins necesarios.

- Se configuraron y ejecutaron jobs en Jenkins para construir y testear la aplicación, verificando su correcto funcionamiento.
- Se creó y ejecutó un pipeline en Jenkins, confirmando el éxito del proceso de CI/CD.