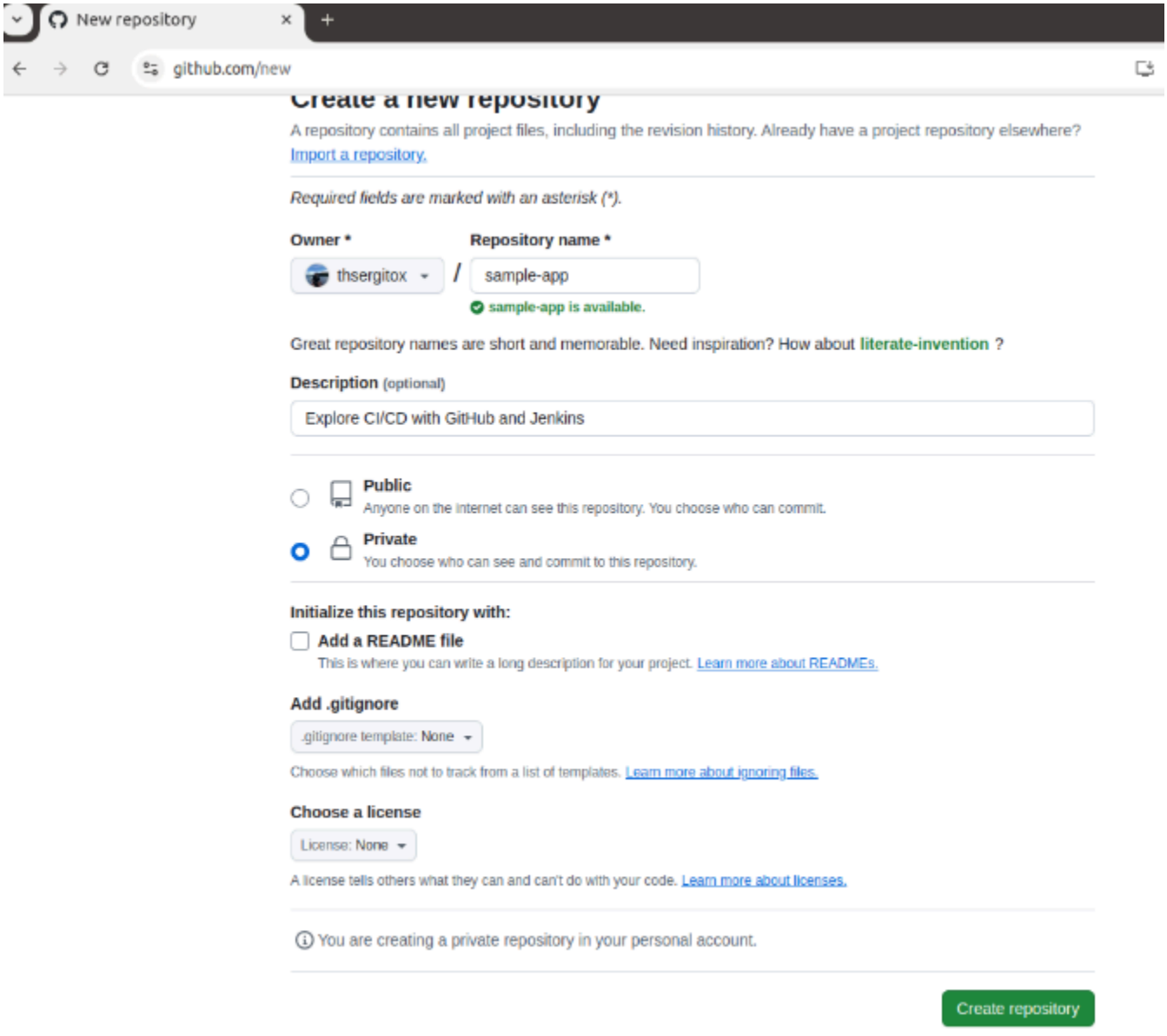
Laboratorio 7

Sergio Sebastian Pezo Jimenez - 20224087G

Confirmar la aplicación de muestra a Git

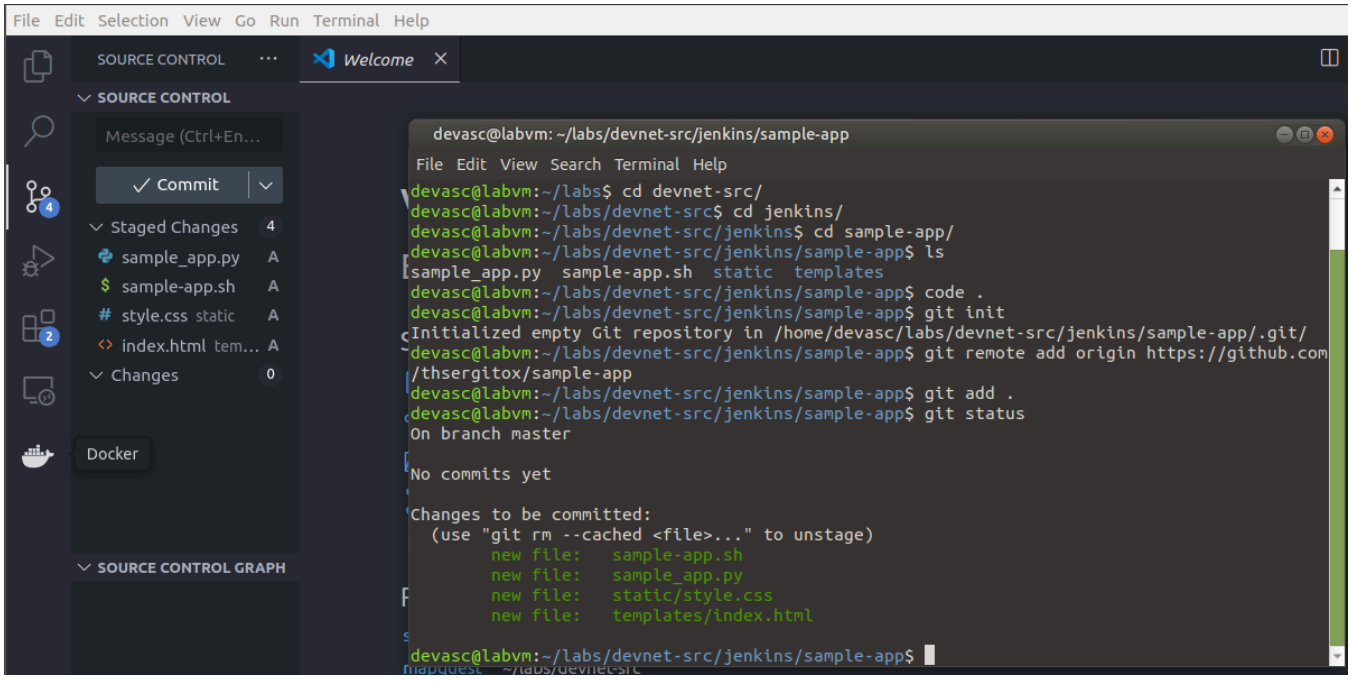
Creamos nuestro repositorio



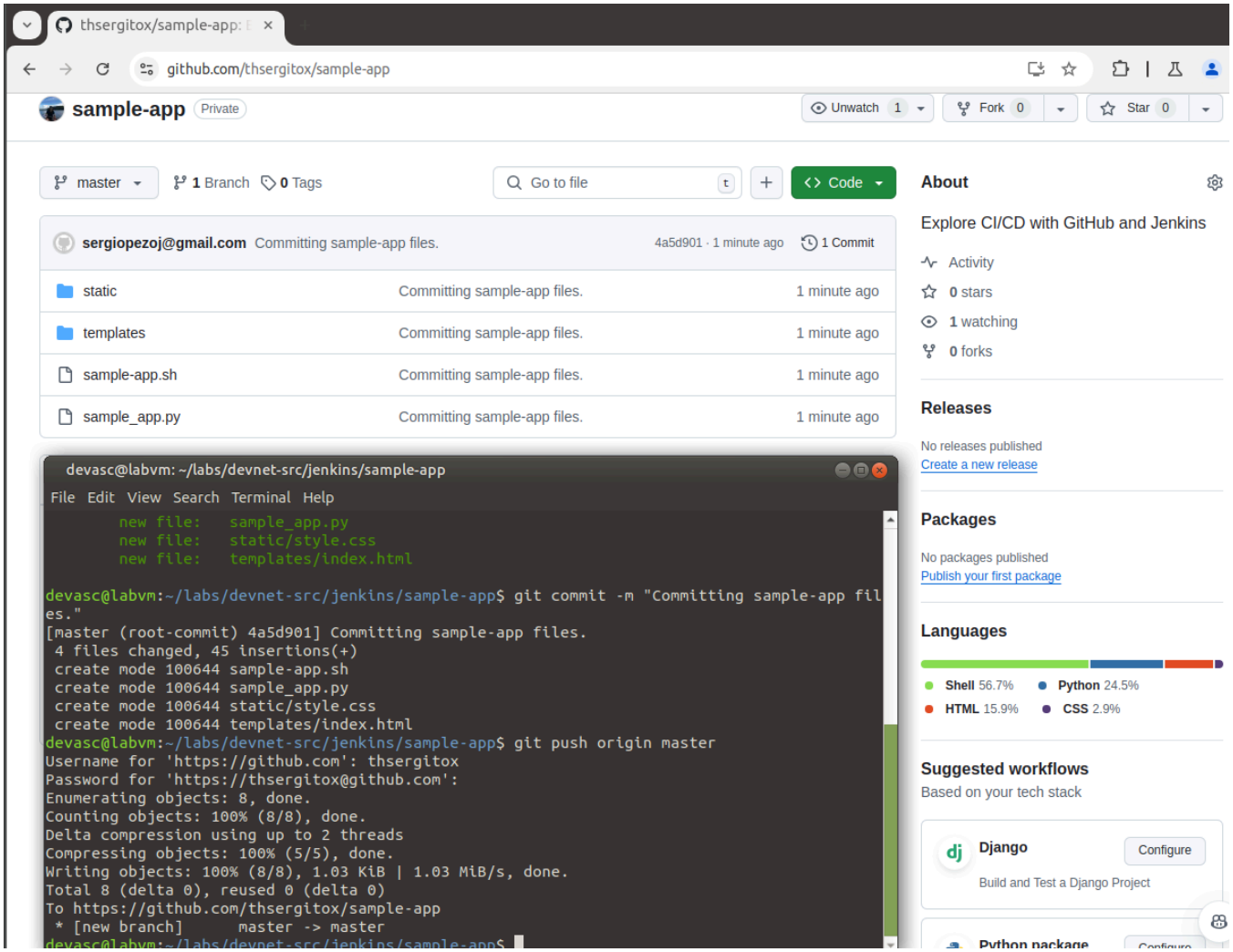
Configuramos nuestras credenciales en la VM

```
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.

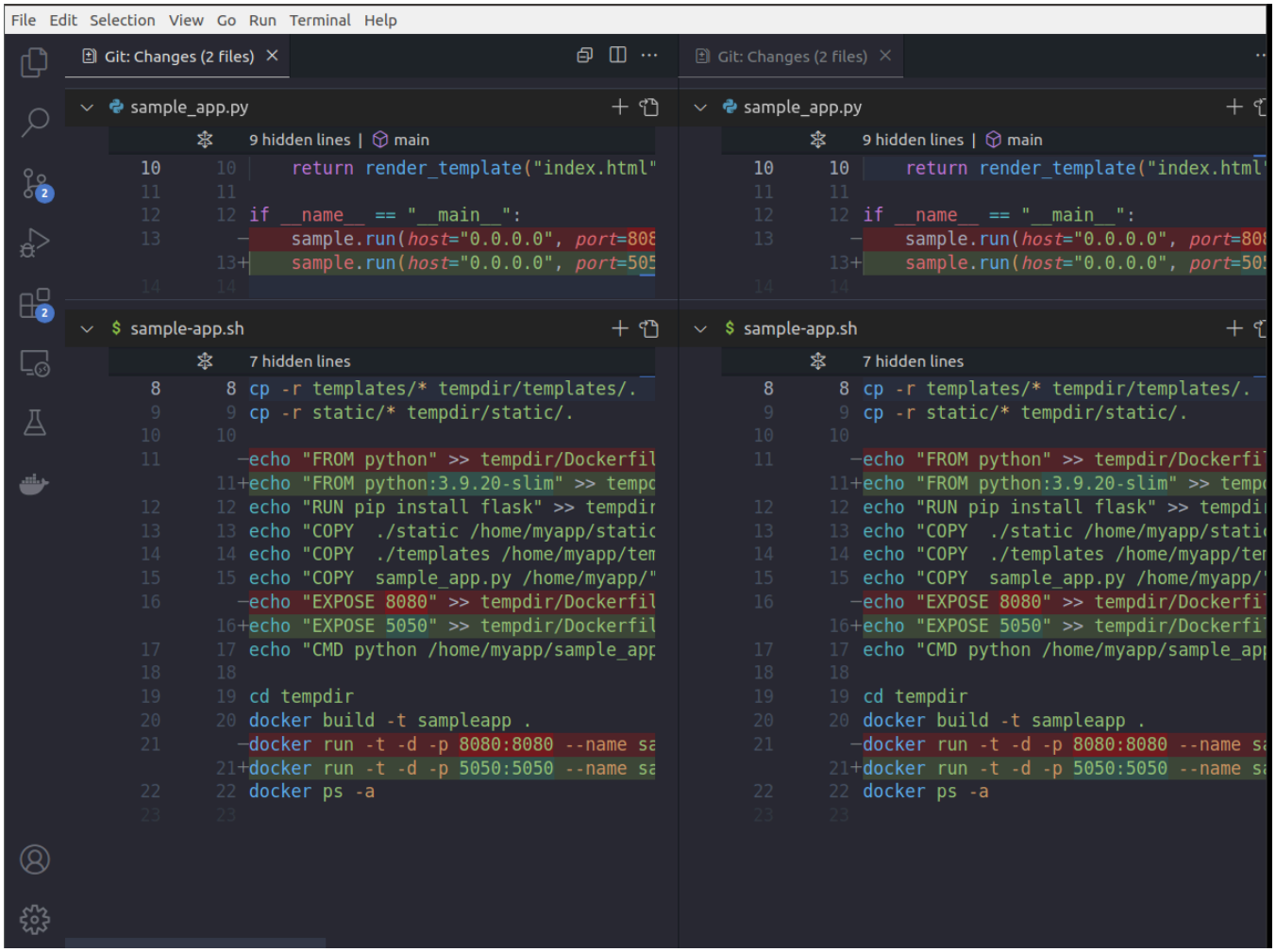


Finalmete lo pusheamos

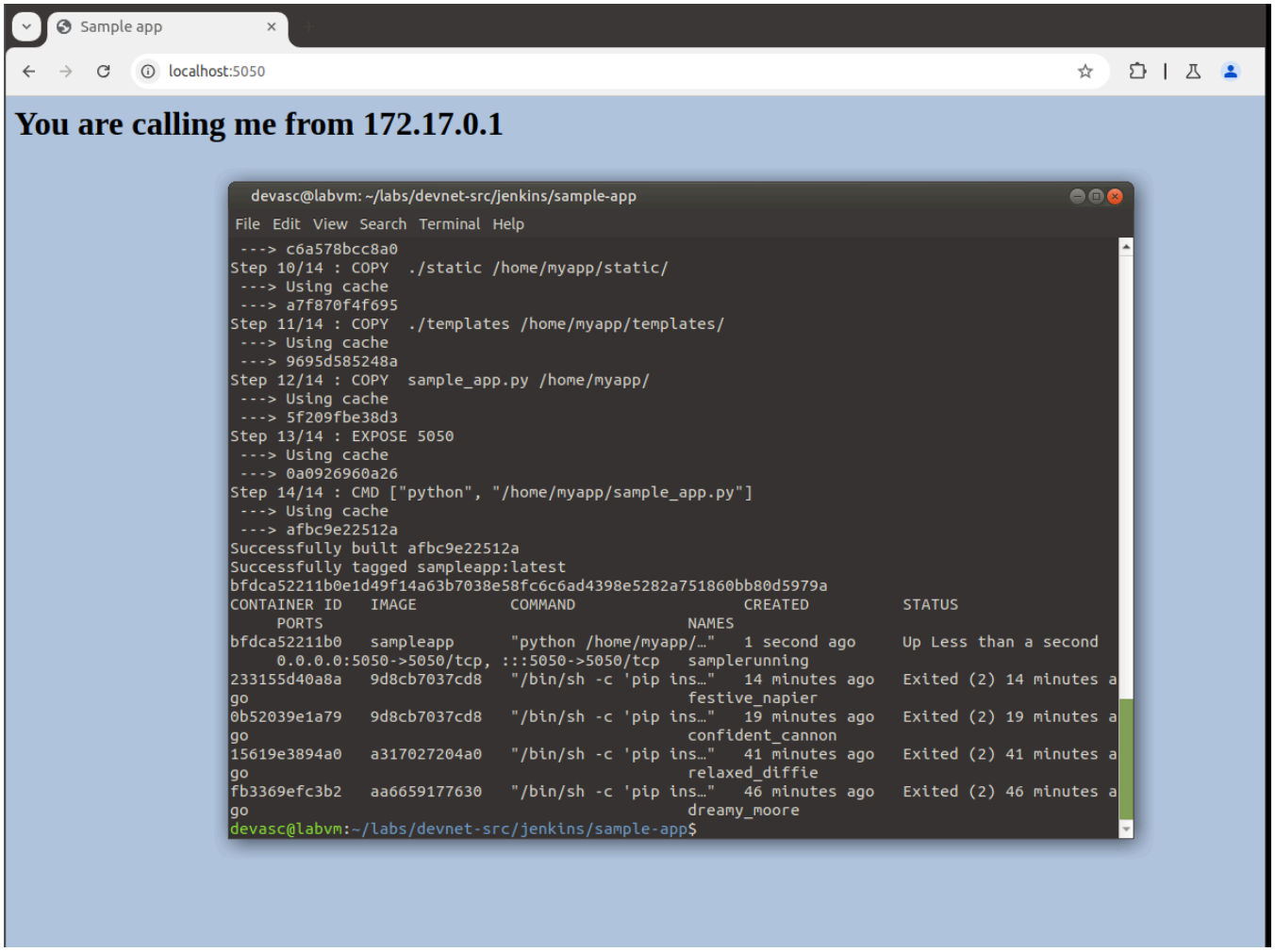


Modificamos los archivos para pushearlos de nuevo

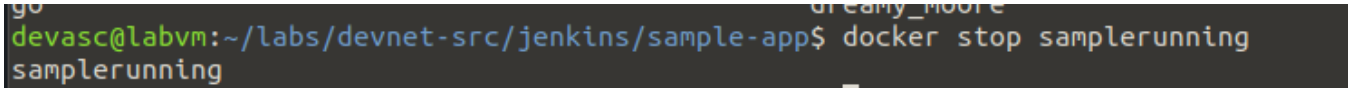
Cambiamos de puerto y versión de python



Construimos nuesra app con docker



Paramos el contenedor



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

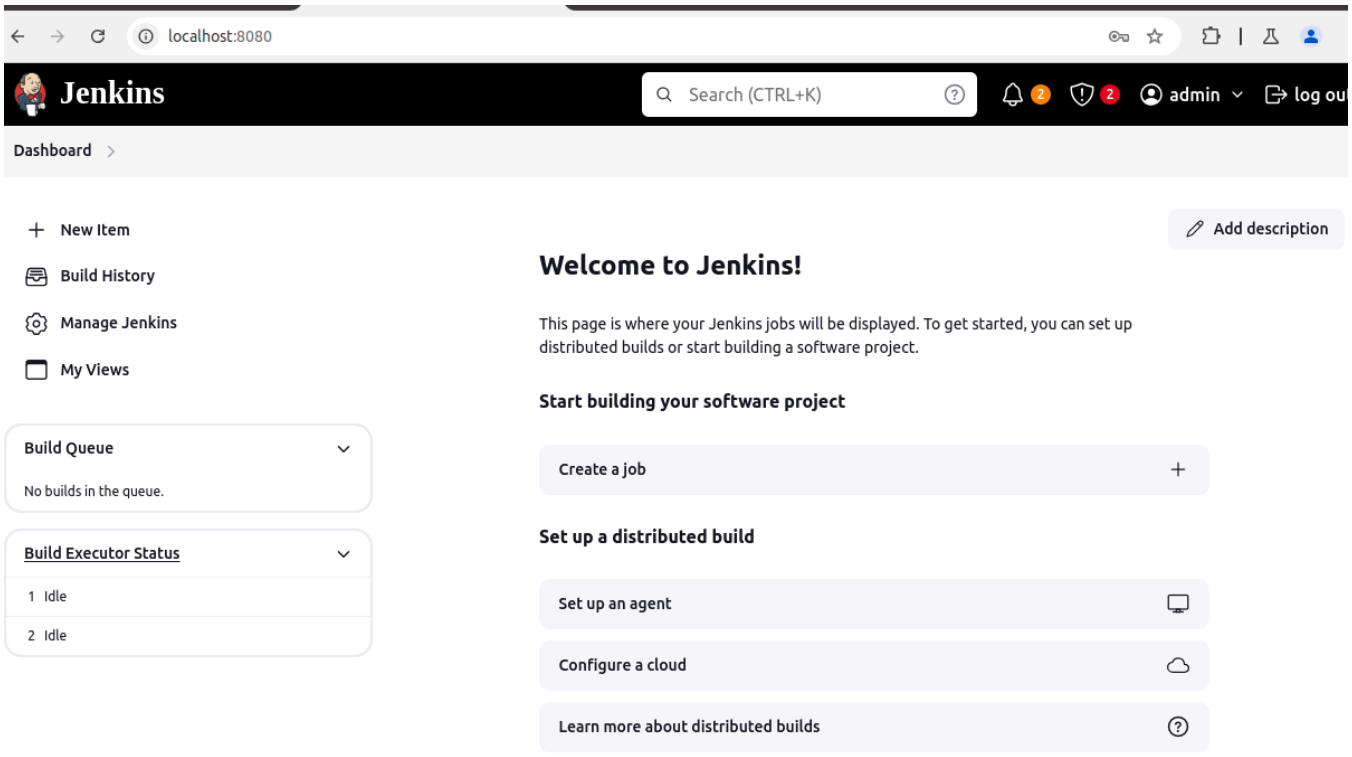
Traemos la imagen de la Docker Hub y la iniciamos, en específico la versión 2.414.3-slim-jdk17, pues mi cuenta de GitHub estaba configurada con SSH, y el plugin para Jenkins estaba para esa versión y todo genial de ahí.

```
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
8351e2eec838: 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 $(which do
ker):/usr/bin/docker -v /var/run/docker.sock:/var/run/docker.sock -v "$HOME":/home --name jenkins_server jenkins/jenkins:2.271
Running from: /usr/share/jenkins/jenkins.war
webroot: EnvVars.masterEnvVars.get("JENKINS_HOME")
```

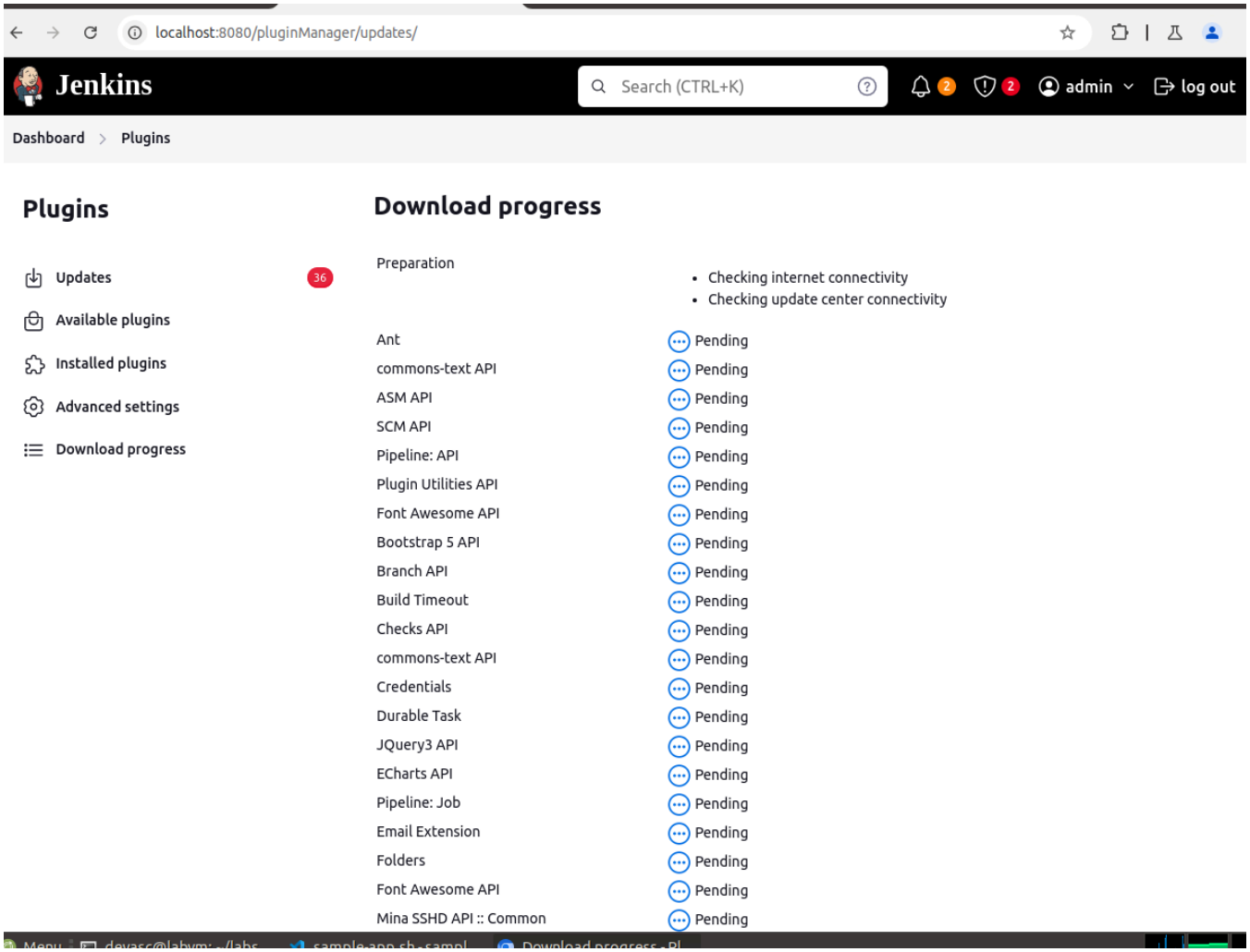
Verificamos

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

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

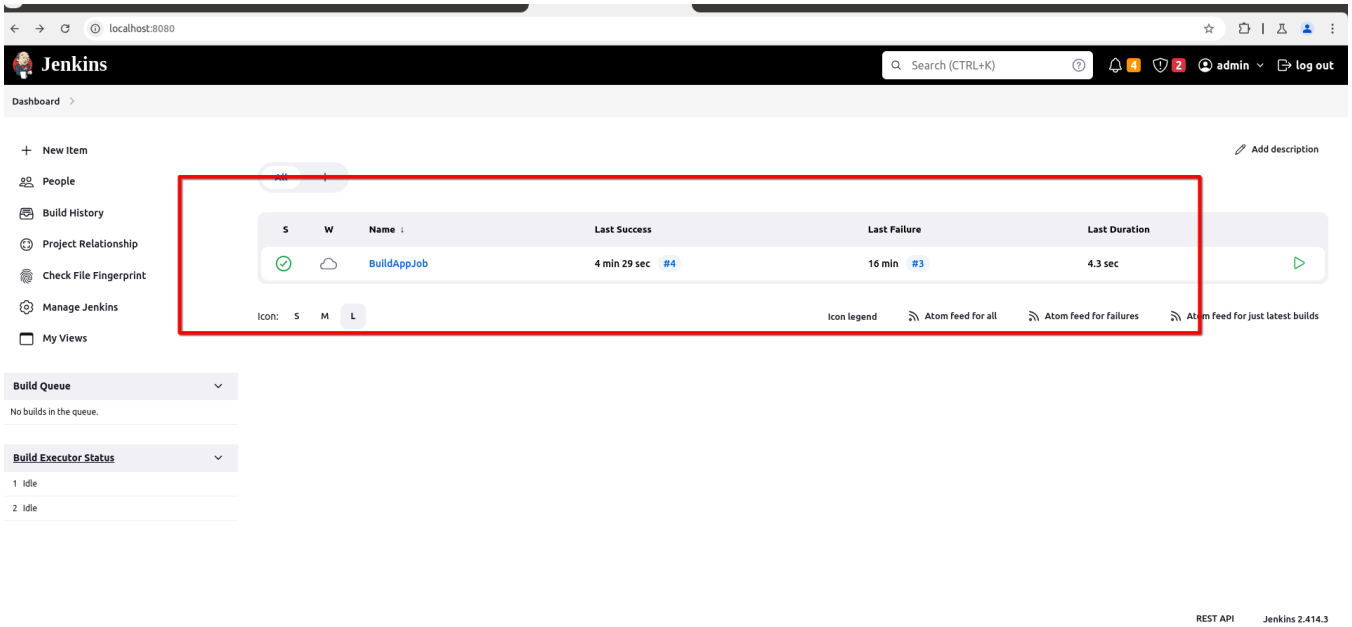


Descargamos todos los plugins recomendados

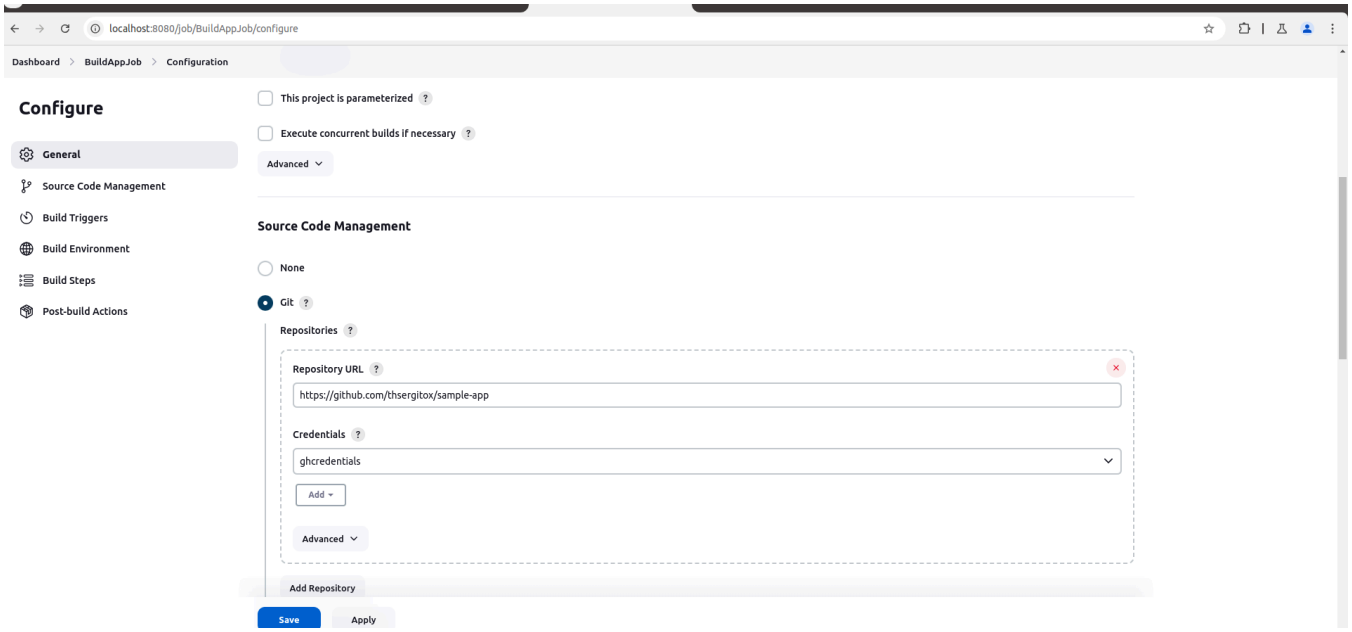


Usamos Jenkins para ejecutar un build de nuestra app

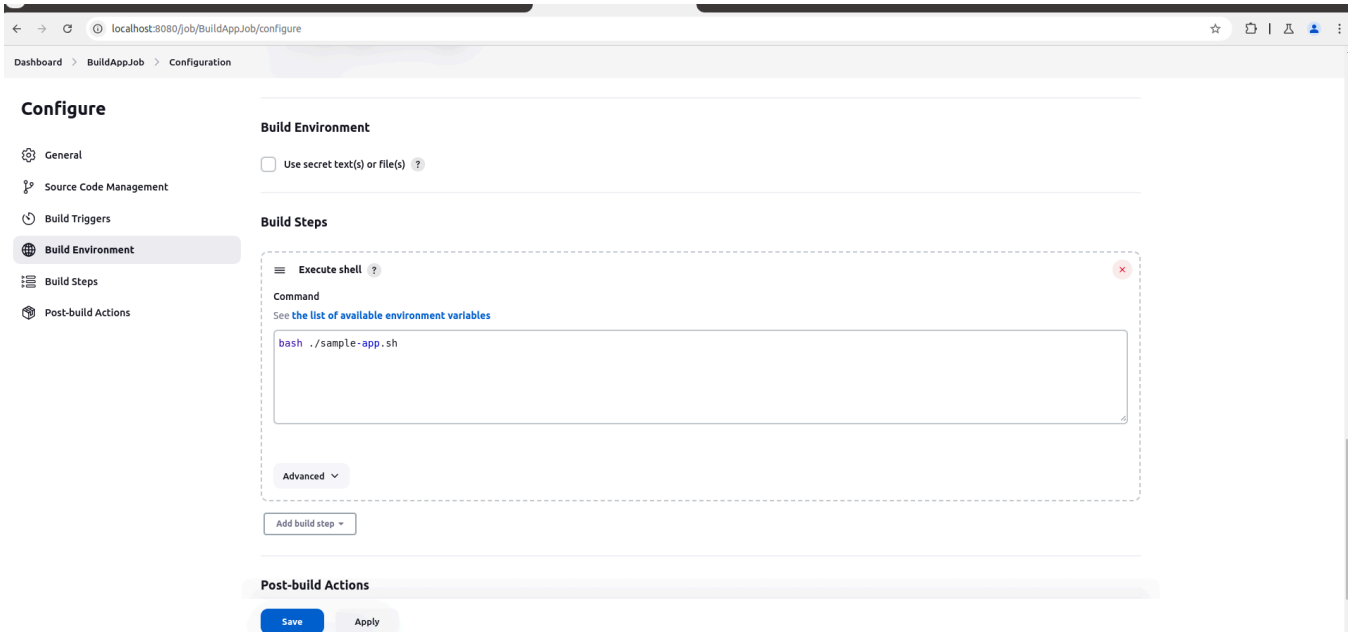
Creamos un item llamado BuildAppJob



Agregamos nuestras credenciales para conectarnos con GitHub y nuestro repositorio.



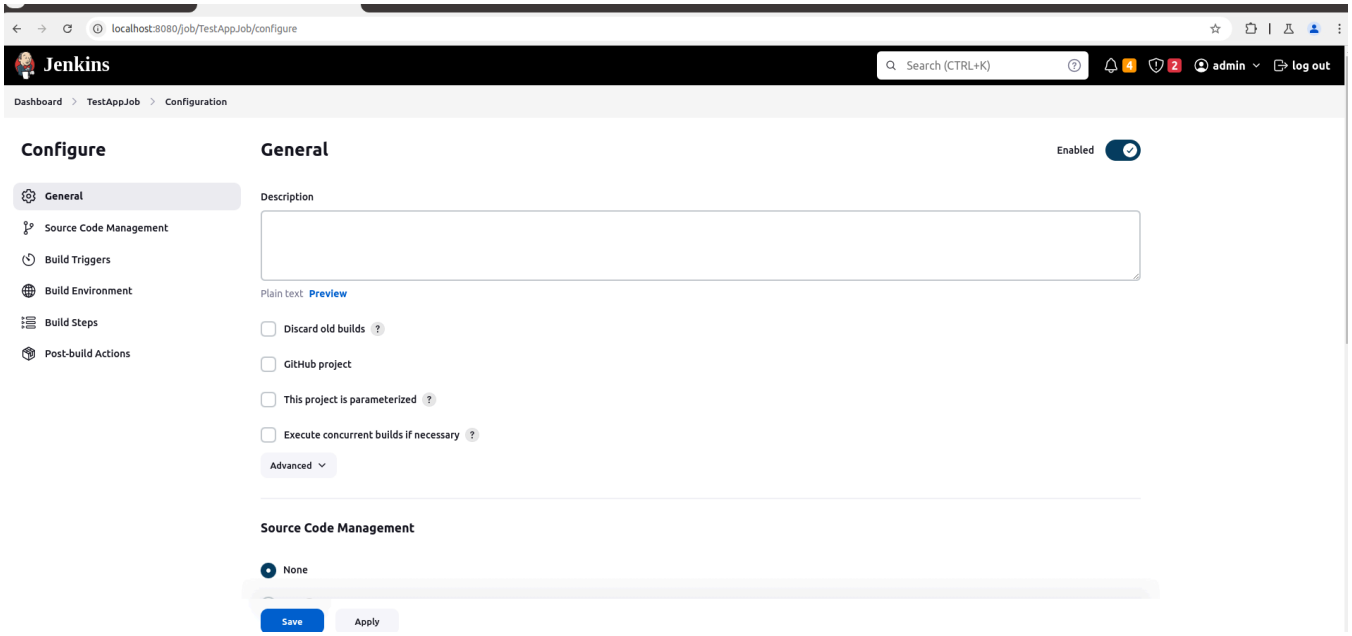
Agregamos un **build step**



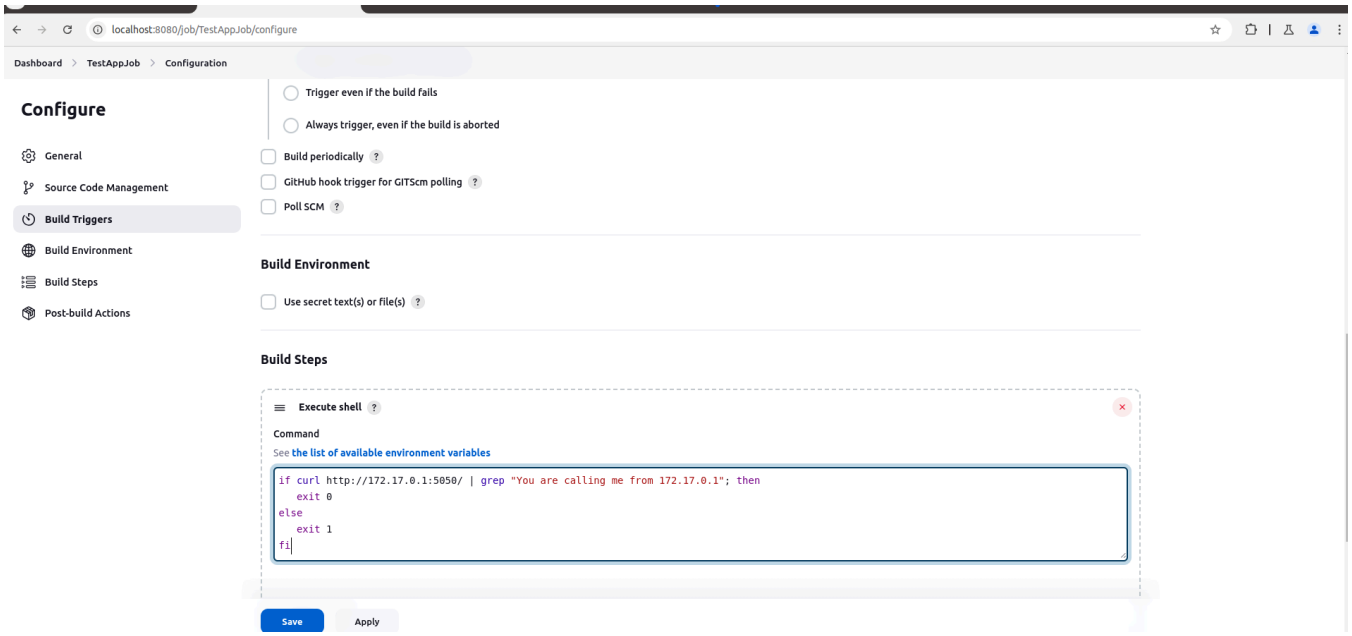
Realizamo un Build y lo ejecutó a la perfección

Usamos Jenkins para testear el Build

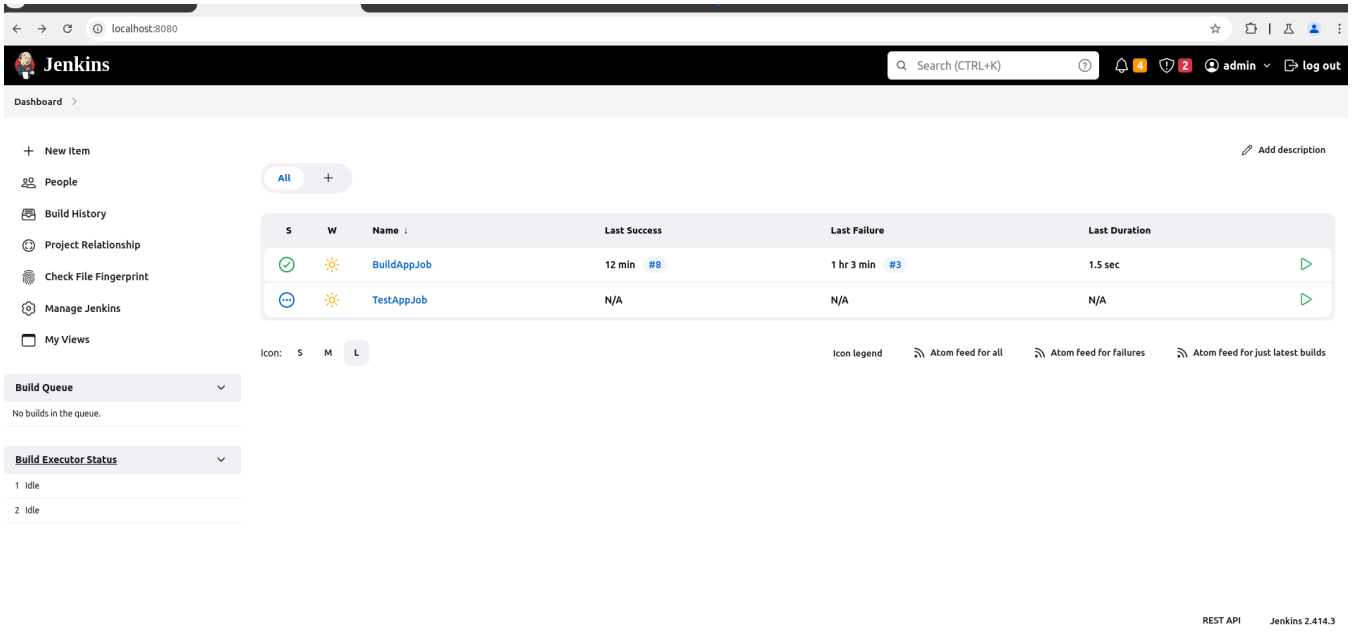
Creamos un nuevo item llamado **TestAppJob**



COnfiguramos nuestro nuevo item.

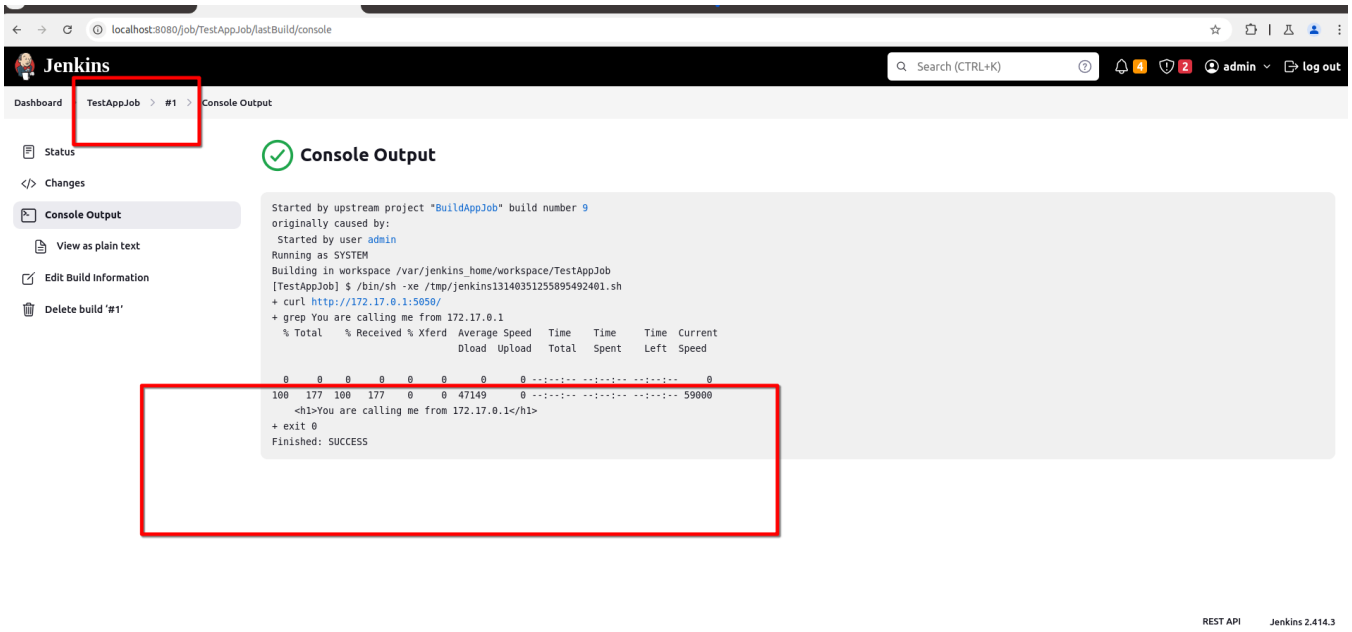


Aqui tenemos los dos



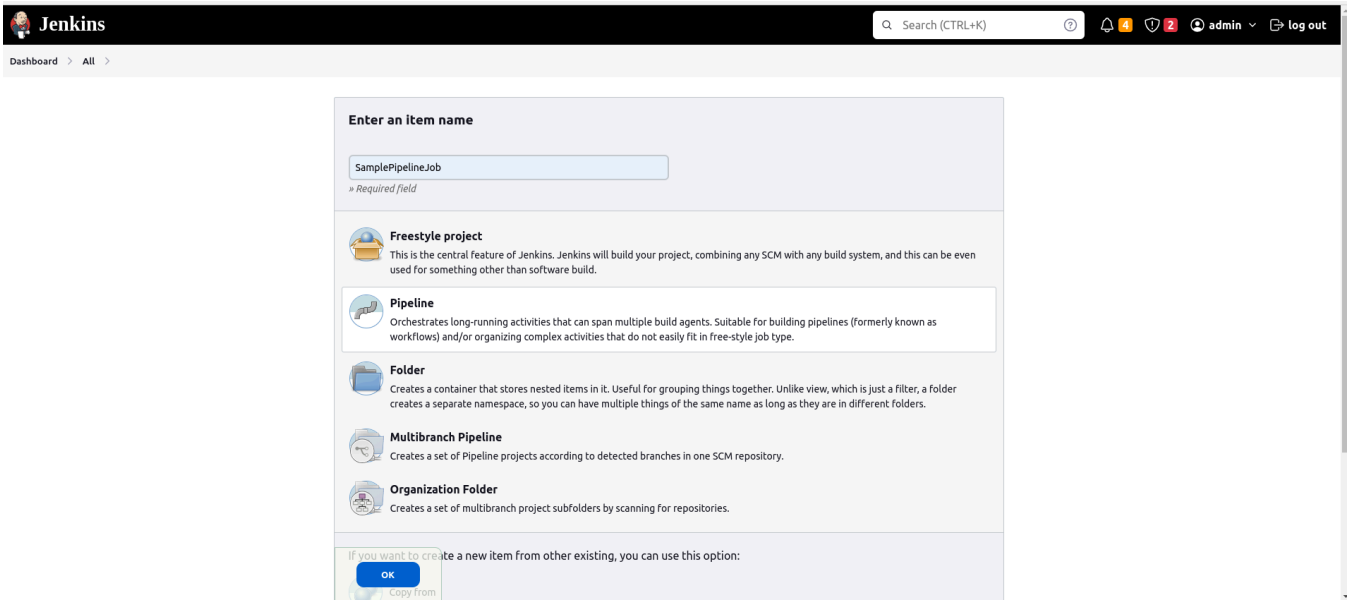
Ejecutamos el build de BuilAppJob y se ejecutó con éxito, tanto ese como el test.

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

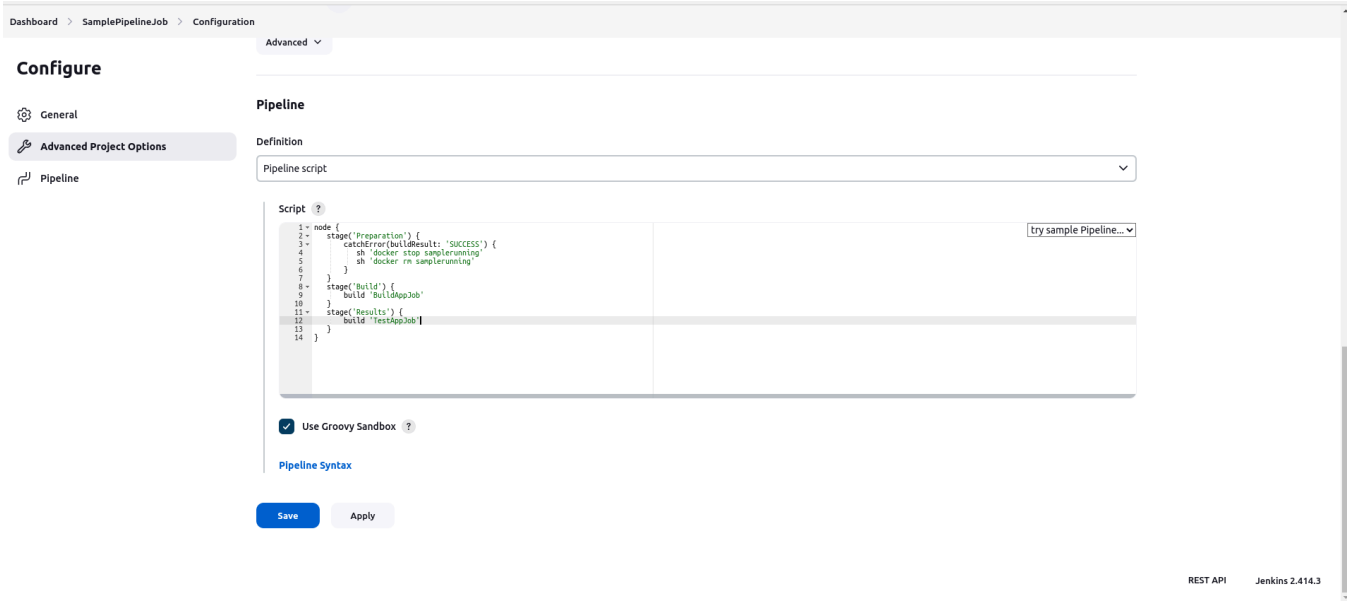


Creamos un pipeline job

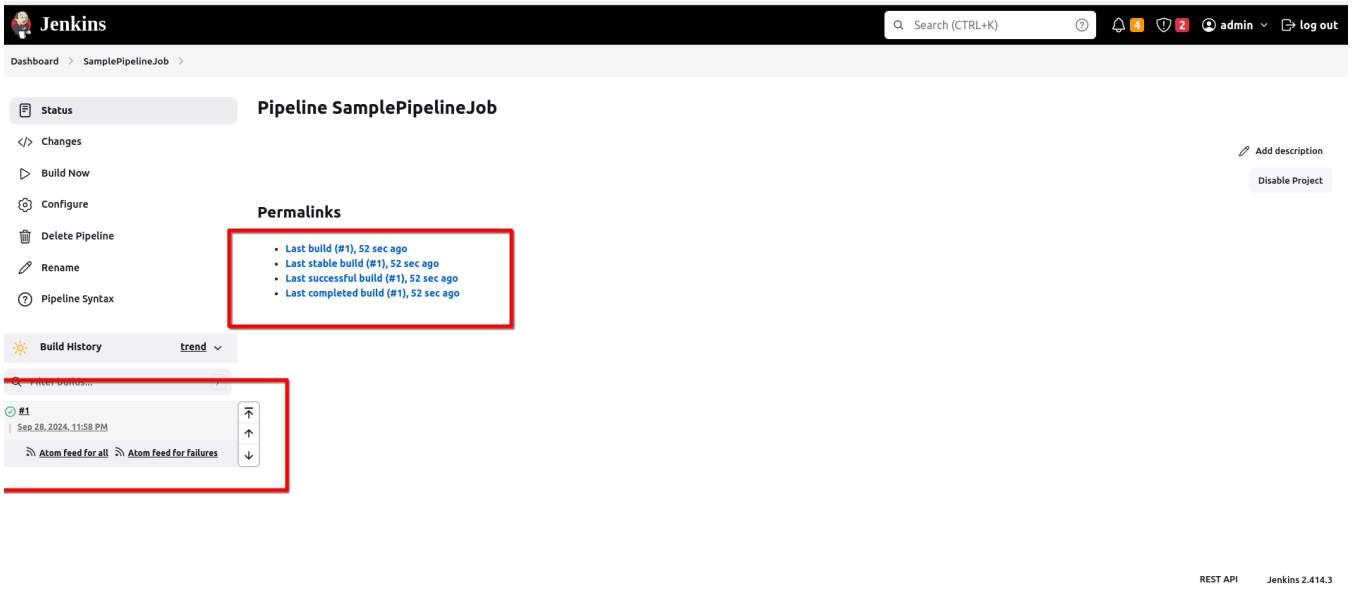
Creamos un nuevo Job de tipo pipeline, instalando el plugin Pipeline.



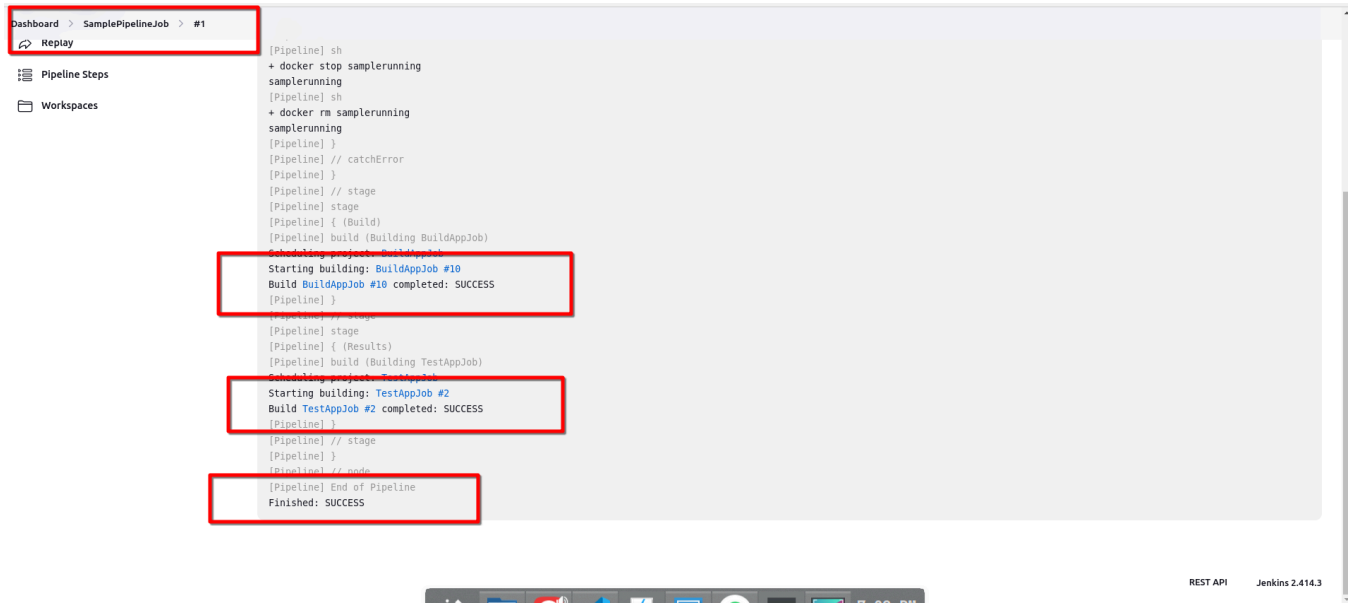
Agregamos el pipeline script



Le damos click a Build Now y funciona



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



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.