

Práctica de laboratorio 7: Build a CI/CD Pipeline Using Jenkins

Part 1: Launch the DEVASC VM



Part 2: Commit the Sample App to Git

Creando repositorio en GitHub:

The screenshot shows the GitHub interface for creating a new repository. At the top, there's a navigation bar with icons for search, issues, pull requests, and user profile. Below it, the main title is "Create a new repository". A note says "A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository." It also specifies that required fields are marked with an asterisk (*).

Repository template: A dropdown menu is set to "No template". A note below it says "Start your repository with a template repository's contents.".

Owner *: A dropdown menu shows "A-PachecoT". **Repository name ***: A text input field contains "sample-app". A note below it says "✓ sample-app is available."

Description (optional): A text area contains "Explore CI/CD with GitHub and Jenkins".

Visibility: Two radio button options are shown: "Public" (unchecked) and "Private" (checked). A note for "Public" says "Anyone on the internet can see this repository. You choose who can commit." A note for "Private" says "You choose who can see and commit to this repository."

Initialize this repository with: An unchecked checkbox for "Add a README file". A note below it says "This is where you can write a long description for your project. [Learn more about READMEs](#)".

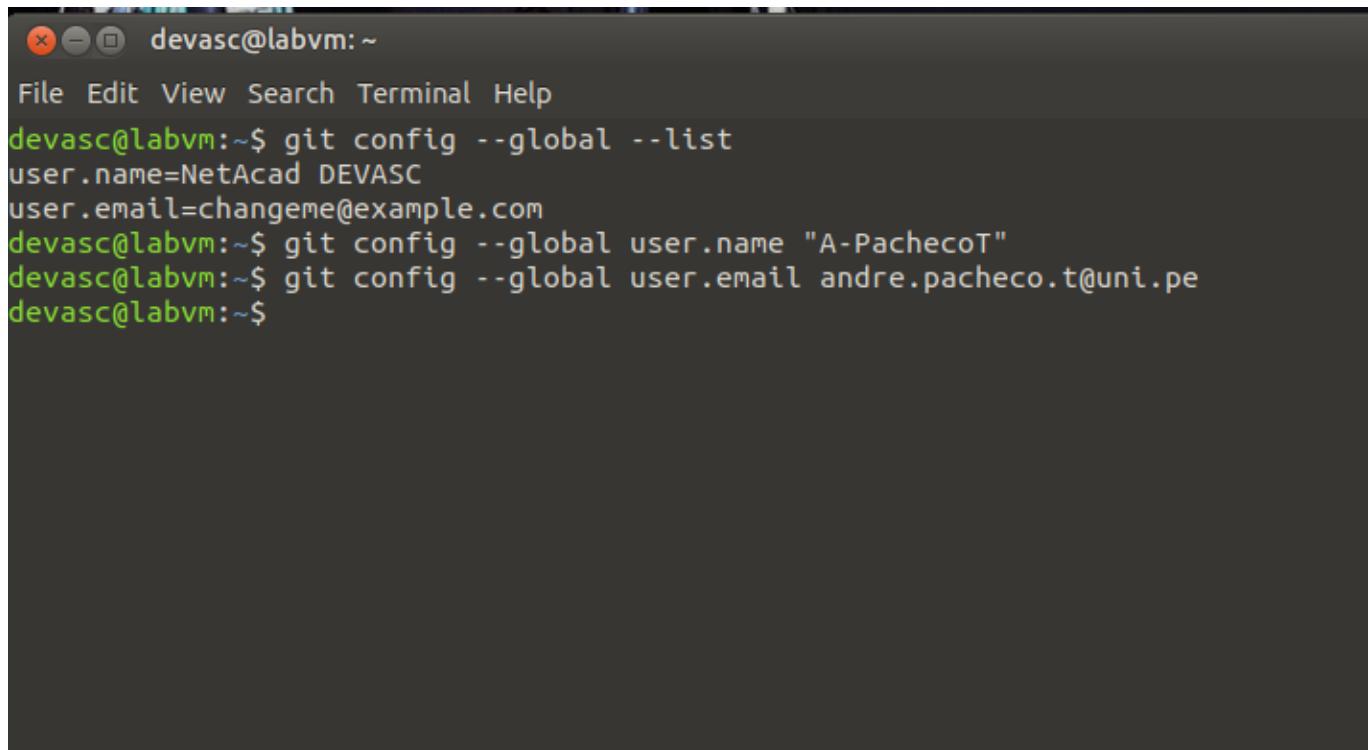
Add .gitignore: A dropdown menu is set to ".gitignore template: None". A note below it says "Choose which files not to track from a list of templates. [Learn more about ignoring files](#)".

Choose a license: A dropdown menu is set to "License: None". A note below it says "A license tells others what they can and can't do with your code. [Learn more about licenses](#)".

Information: A note says "ⓘ You are creating a private repository in your personal account."

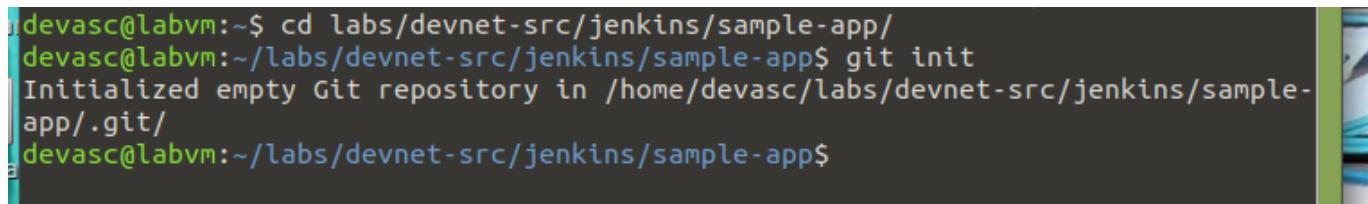
Create repository: A large green button at the bottom right.

Configurando credenciales en la máquina virtual:



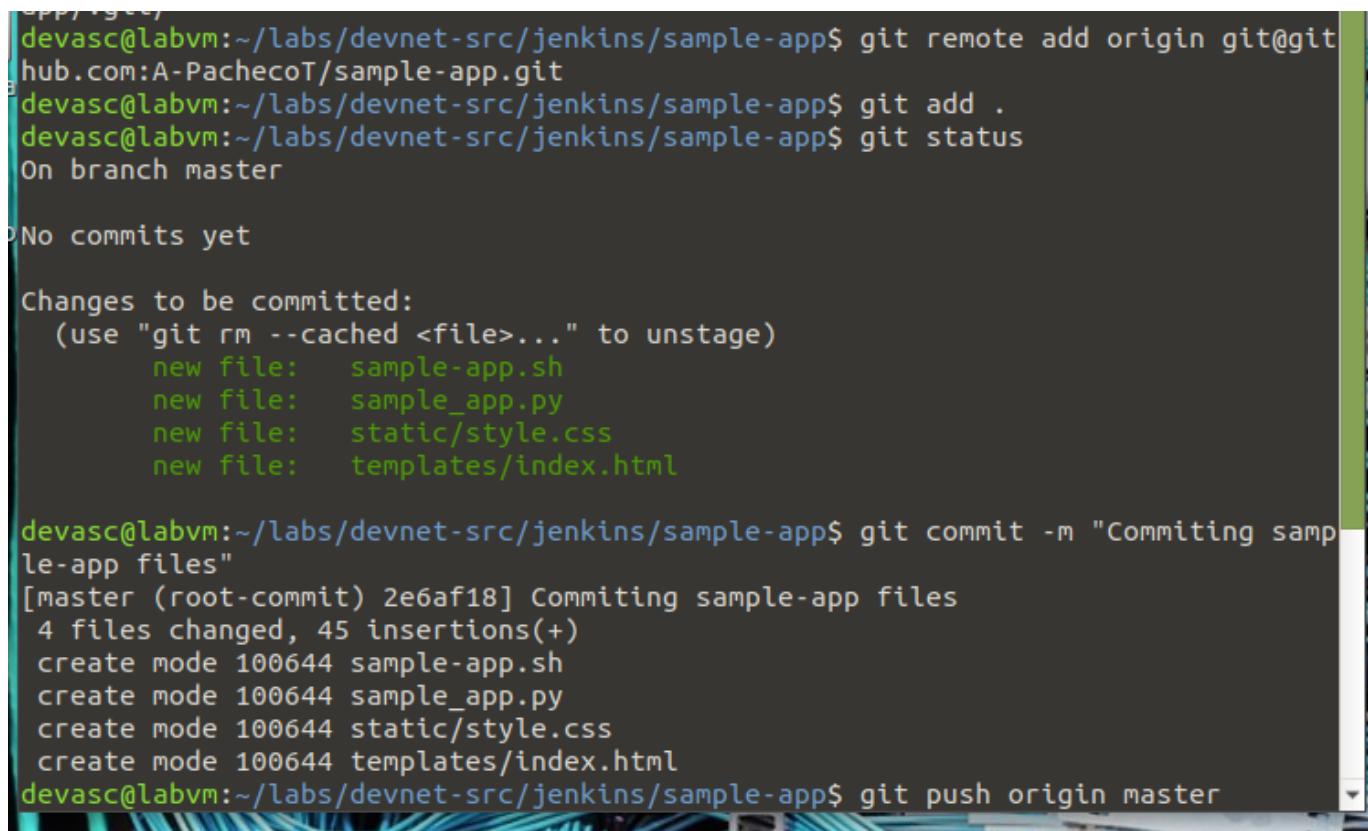
```
devasc@labvm:~$ git config --global --list
user.name=NetAcad DEVASC
user.email=changeme@example.com
devasc@labvm:~$ git config --global user.name "A-PachecoT"
devasc@labvm:~$ git config --global user.email andre.pacheco.t@uni.pe
devasc@labvm:~$
```

Inicializando repositorio local:



```
devasc@labvm:~$ cd labs/devnet-src/jenkins/sample-app/
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$
```

Enlazando con el repositorio remoto y agregando los archivos con add, commiteando y puseando a master:



```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git remote add origin git@git.hub.com:A-PachecoT/sample-app.git
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$ git commit -m "Committing sample-app files"
[master (root-commit) 2e6af18] 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
```

El uso de contraseñas para la autenticación es más fácil de implementar, pero Github deprecó el uso de contraseñas para la autenticación por lo que se uso claves ssh. Por ello tengo que crear una clave ssh:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ ssh-keygen -t ed25519 -C "andre.pacheco.t@uni.pe"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/devasc/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/devasc/.ssh/id_ed25519
Your public key has been saved in /home/devasc/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:agvkBEgHfh9R8DNanICpRKpEQ1ZYNVEZnxx6PyleLew andre.pacheco.t@uni.pe
The key's randomart image is:
++-[ED25519 256]---+
|o0=++B=+o.          |
|*+oo *.= o          |
|+oo.. . 0 =          |
|o.... + + o o       |
|. oo S. B .         |
| + ... + o          |
| o o . E            |
| o .                |
| .                  |
+---[SHA256]---+
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ eval "$(ssh-agent -s)"
Agent pid 4488
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ ssh-add ~/.ssh/id_ed25519
Identity added: /home/devasc/.ssh/id_ed25519 (andre.pacheco.t@uni.pe)
```

con

```
cat ~/.ssh/id_rsa.pub
```

vemos la clave. Esta clave se añade a las llaves de mi cuenta de github:

The screenshot shows the GitHub Settings interface. On the left, there's a sidebar with links like 'Public profile', 'Account', 'Appearance', 'Accessibility', 'Notifications', 'Access' (with 'Billing and plans' expanded), 'Emails', 'Password and authentication', 'Sessions', 'SSH and GPG keys' (which is selected and highlighted in blue), 'Organizations', 'Enterprises', and 'Moderation'. Below the sidebar, there are sections for 'Code, planning, and automation' with 'Repositories' and 'Actions'. At the top right, there's a search bar, a '+' button, a circular icon, and a user profile picture. The main content area is titled 'Add new SSH Key'. It has fields for 'Title' (set to 'devasc-vm'), 'Key type' (set to 'Authentication Key'), and a large text area for the key itself. The key text is: 'ssh-ed25519 POR SEGURIDAD NO PONGO TODA LA CLASE AQUI...vJ0 andre.pacheco.t@uni.pe'. At the bottom right of this area is a green 'Add SSH key' button.

Finalmente puedo pushear a mi repositorio remoto:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git push origin master
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 github.com:A-PachecoT/sample-app.git
 * [new branch]      master -> master
```

Part 3: Modify the Sample App and Push Changes to Git

Ahora cambiemos archivos...

Se cambia el puerto en el .py y .hs:

```
x - devasc@labvms: ~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help
# Add to this file for the sample app lab
from flask import Flask
from flask import request
from flask import render_template

sample = Flask(__name__)

@app.route("/")
def main():
    return render_template("index.html")

if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=5050)
~
```

```
x - devasc@labvms: ~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help

cp sample_app.py tempdir/.
cp -r templates/* tempdir/templates/.
cp -r static/* tempdir/static/.

echo "FROM python" >> tempdir/Dockerfile
echo "RUN pip install flask" >> tempdir/Dockerfile
echo "COPY ./static /home/myapp/static/" >> tempdir/Dockerfile
echo "COPY ./templates /home/myapp/templates/" >> tempdir/Dockerfile
echo "COPY sample_app.py /home/myapp/" >> tempdir/Dockerfile
echo "EXPOSE 5050" >> tempdir/Dockerfile
echo "CMD python /home/myapp/sample_app.py" >> tempdir/Dockerfile

cd tempdir

docker build -t sampleapp .

docker run -t -d -p 5050:5050 --name samplerunning sampleapp
docker ps -a
~
```

Tenía el contenedor creado así que tuve que eliminar y volver a crearlo:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             NAMES
6e45c30c5ceb      e6554b917808      "/bin/sh -c 'python3..."   41 hours ago    samplerunning
          Exited (137) 40 hours ago
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker rm ^C
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker rm 6e45c30c5ceb
6e45c30c5ceb
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             NAMES
STATUS              PORTS              NAMES

```

Y vuelvo a ejecutar el .sh:

```
devasc@labvm: ~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help
---> d4a34b986a46
Step 7/14 : CMD python /home/myapp/sample_app.py
---> Using cache
---> 8f24c207da6b
Step 8/14 : FROM python
---> ea2ebd905ab2
Step 9/14 : RUN pip install flask
---> Using cache
---> 65a193e541eb
Step 10/14 : COPY ./static /home/myapp/static/
---> Using cache
---> 285bad4ca062
Step 11/14 : COPY ./templates /home/myapp/templates/
---> Using cache
---> 413d2b236dff
Step 12/14 : COPY sample_app.py /home/myapp/
---> Using cache
---> 2599f082c490
Step 13/14 : EXPOSE 5050
---> Using cache
---> d4a34b986a46
Step 14/14 : CMD python /home/myapp/sample_app.py
---> Using cache
---> 8f24c207da6b
Successfully built 8f24c207da6b
Successfully tagged sampleapp:latest
0615e1c27e0ffc4c4f921361ce2b2606373e7629e399b6c304d83e598b7c1487
CONTAINER ID        IMAGE               COMMAND                  CREATED             NAMES
      STATUS              PORTS              NAMES
0615e1c27e0f      sampleapp           "/bin/sh -c 'python ..."   1 second ago    samplerunning
          Up Less than a second  0.0.0.0:5050->5050/tcp  samplerunning
devasc@labvm: ~/labs/devnet-src/jenkins/sample-app$
```

Listo, ya tenemos el contenedor funcionando con el nuevo puerto.

Por ultimo subimos los cambios a mi repositorio remoto:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app
File Edit View Search Terminal Help
Up Less than a second 0.0.0.0:5050->5050/tcp samplerunning
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)
    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 "chore: changed port from 8080 to 5050"
[master 9f5a281] chore: changed port from 8080 to 5050
6 files changed, 42 insertions(+), 3 deletions(-)
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
Warning: Permanently added the ECDSA host key for IP address '140.82.112.4' to the list of known hosts.
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 785 bytes | 98.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:A-PachecoT/sample-app.git
  2e6af18..9f5a281 master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Efectivamente se pusieron los cambios.

A-PachecoT chore: changed port from 8080 to 5050
static Committing sample-app files
tempdir chore: changed port from 8080...
templates Committing sample-app files
sample-app.sh chore: changed port from 8080...
sample_app.py chore: changed port from 8080...

About

Explore CI/CD with GitHub and Jenkins

Activity 0 stars 1 watching 0 forks

Releases

No releases published [Create a new release](#)

Packages

Part 4: Download and Run the Jenkins Docker Image

Descargamos la imagen Its de jenkins con el comando:

```
docker pull jenkins/jenkins:lts
```

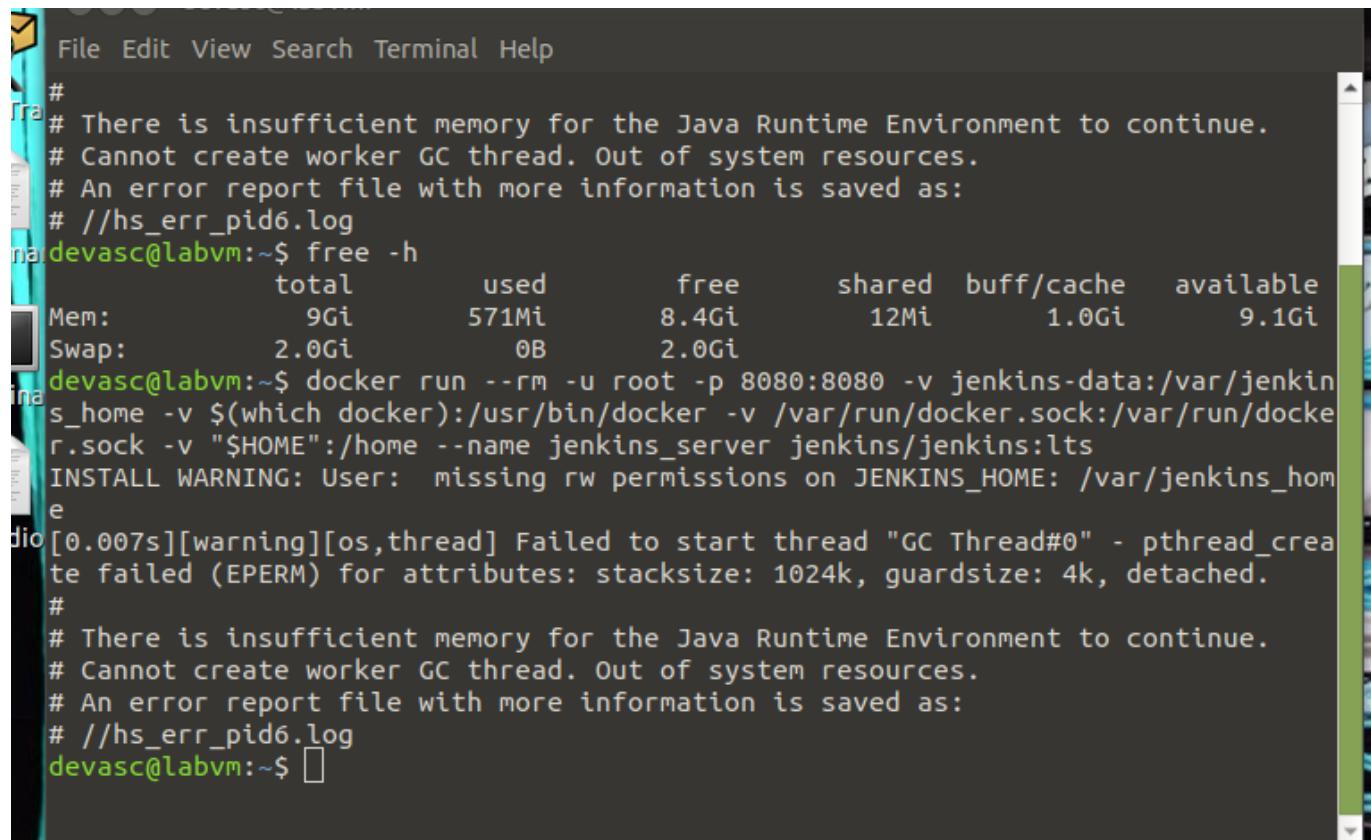
```
2e6af18..9f5a281 master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
903681d87777: Pull complete
f76fc73f1c48: Pull complete
fc1d2482b243: Pull complete
5bae62448211: Pull complete
9020ffff6008: Pull complete
8a9191d56587: Pull complete
da374eff6f05: Pull complete
31aceeb653c9: Pull complete
061dcfd72fbac: Pull complete
79716cc251e4: Pull complete
628c862ab449: Pull complete
d7dec4cb14f6: Pull complete
Digest: sha256:95313257a8cddbef83c74e3d577ea139aeae30c3c014ddcaa83a72b60409bbe1
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Empezamos el contenedor con el comando:

```
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:lts
```

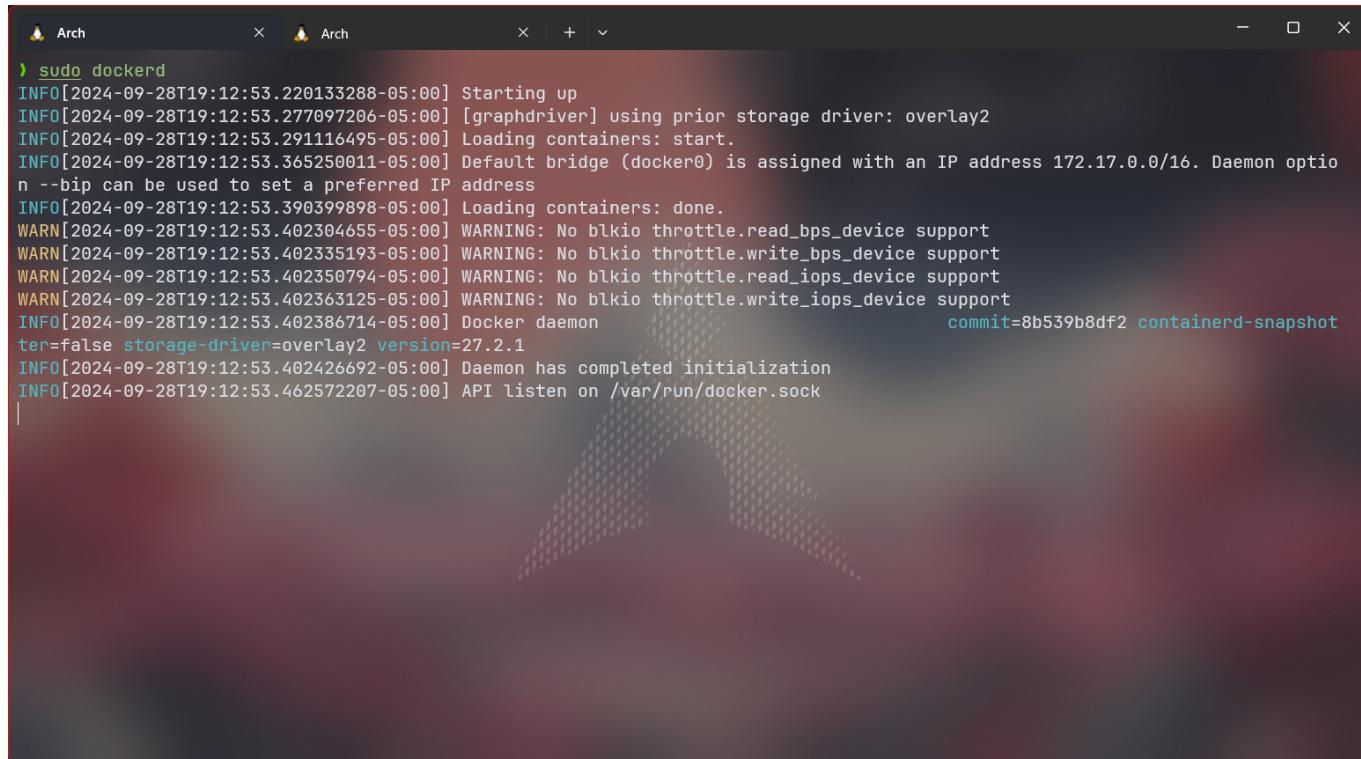
Tuve un error al iniciar el contenedor, por falta de memoria RAM; pero por más que le puse +10GB de RAM no me dejaba iniciar el contenedor.



```
File Edit View Search Terminal Help
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Cannot create worker GC thread. Out of system resources.
# An error report file with more information is saved as:
# //hs_err_pid6.log
devasc@labvm:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:          9Gi       571Mi       8.4Gi       12Mi       1.0Gi       9.1Gi
Swap:        2.0Gi          0B       2.0Gi
devasc@labvm:~$ 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:lts
INSTALL WARNING: User: missing rw permissions on JENKINS_HOME: /var/jenkins_home
dio[0.007s][warning][os,thread] Failed to start thread "GC Thread#0" - pthread_create failed (EPERM) for attributes: stacksize: 1024k, guardsize: 4k, detached.
#
# There is insufficient memory for the Java Runtime Environment to continue.
# Cannot create worker GC thread. Out of system resources.
# An error report file with more information is saved as:
# //hs_err_pid6.log
devasc@labvm:~$ 
```

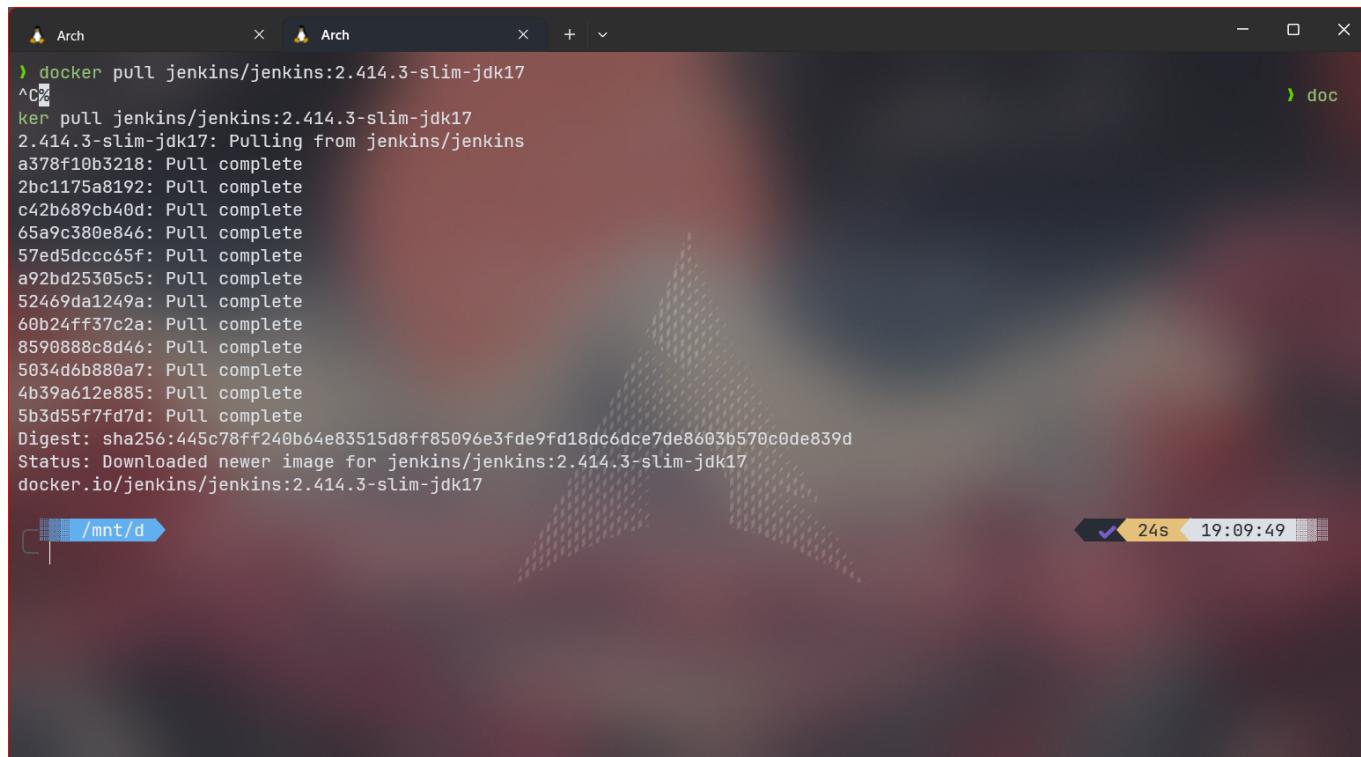
Luego de insistentes errores al seguir con el laboratorio, decidí proseguir usando el Windows Subsystem for Linux (WSL). En particular estoy usando Arch Linux WSL.

Prendo docker daemon



```
> sudo dockerd
INFO[2024-09-28T19:12:53.220133288-05:00] Starting up
INFO[2024-09-28T19:12:53.277097206-05:00] [graphdriver] using prior storage driver: overlay2
INFO[2024-09-28T19:12:53.291116495-05:00] Loading containers: start.
INFO[2024-09-28T19:12:53.365250011-05:00] Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a preferred IP address
INFO[2024-09-28T19:12:53.390399898-05:00] Loading containers: done.
WARN[2024-09-28T19:12:53.402304655-05:00] WARNING: No blkio throttle.read_bps_device support
WARN[2024-09-28T19:12:53.402335193-05:00] WARNING: No blkio throttle.write_bps_device support
WARN[2024-09-28T19:12:53.402350794-05:00] WARNING: No blkio throttle.read_iops_device support
WARN[2024-09-28T19:12:53.402363125-05:00] WARNING: No blkio throttle.write_iops_device support
INFO[2024-09-28T19:12:53.402386714-05:00] Docker daemon commit=8b539b8df2 containerd-snapshotter=false storage-driver=overlay2 version=27.2.1
INFO[2024-09-28T19:12:53.402426692-05:00] Daemon has completed initialization
INFO[2024-09-28T19:12:53.462572207-05:00] API listen on /var/run/docker.sock
```

Descargo la imagen de jenkins 2.414.3-slim-jdk17 que es compatible con la clave ssh con la que enlazé mi repositorio remoto:



```
> docker pull jenkins/jenkins:2.414.3-slim-jdk17
^C%
ker pull jenkins/jenkins:2.414.3-slim-jdk17
2.414.3-slim-jdk17: Pulling from jenkins/jenkins
a378f10b3218: Pull complete
2bc1175a8192: Pull complete
c42b689cb40d: Pull complete
65a9c380e846: Pull complete
57ed5dccc65f: Pull complete
a92bd25305c5: Pull complete
52469da1249a: Pull complete
60b24ff37c2a: Pull complete
8590888c8d46: Pull complete
5034d6b880a7: Pull complete
4b39a612e885: Pull complete
5b3d55f7fd7d: Pull complete
Digest: sha256:445c78ff240b64e83515d8ff85096e3fde9fd18dc6dce7de8603b570c0de839d
Status: Downloaded newer image for jenkins/jenkins:2.414.3-slim-jdk17
docker.io/jenkins/jenkins:2.414.3-slim-jdk17
```

Ejecuto el contenedor con el comando:

```
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:2.414.3-slim-jdk17
```

```

❯ 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:2.414.3-slim-jdk17
Running from: /usr/share/jenkins/jenkins.war
webroot: /var/jenkins_home/war
2024-09-29 00:16:28.853+0000 [id=1]     INFO    winstone.Logger#logInternal: Beginning extraction from war file
2024-09-29 00:16:28.925+0000 [id=1]     WARNING o.e.j.s.handler.ContextHandler#setContextPath: Empty contextPath
2024-09-29 00:16:28.985+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: jetty-10.0.17; built: 2023-10-02T04:04:10.314Z; git: a0f5f05abaa6c3aabb7c3d35f10a6f412ab8b05f; jvm 17.0.8.1+1
2024-09-29 00:16:29.158+0000 [id=1]     INFO    o.e.j.w.StandardDescriptorProcessor#visitServlet: NO JSP Support for /, did not find org.eclipse.jetty.jsp.JettyJspServlet
2024-09-29 00:16:29.196+0000 [id=1]     INFO    o.e.j.s.s.DefaultSessionIdManager#doStart: Session workerName=node0
2024-09-29 00:16:29.571+0000 [id=1]     INFO    hudson.WebAppMain#contextInitialized: Jenkins home directory: /var/jenkins_home found at: EnvVars.masterEnvVars.get("JENKINS_HOME")
2024-09-29 00:16:29.655+0000 [id=1]     INFO    o.e.j.s.handler.ContextHandler#doStart: Started w.@56781d96{Jenkins v2.414.3, /, file:///var/jenkins_home/war/,AVAILABLE}{/var/jenkins_home/war}
2024-09-29 00:16:29.668+0000 [id=1]     INFO    o.e.j.server.AbstractConnector#doStart: Started ServerConnector@1329eff{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2024-09-29 00:16:29.679+0000 [id=1]     INFO    org.eclipse.jetty.server.Server#doStart: Started Server@be35cd9{STARTING}[10.0.17,sto=0] @1175ms
2024-09-29 00:16:29.680+0000 [id=27]     INFO    winstone.Logger#logInternal: Winstone Servlet Engine running: controlPort=disabled
2024-09-29 00:16:29.833+0000 [id=34]     INFO    jenkins.InitReactorRunner$1#onAttained: Started initialization
2024-09-29 00:16:29.876+0000 [id=54]     INFO    jenkins.InitReactorRunner$1#onAttained: Listed all plugins
2024-09-29 00:16:30.446+0000 [id=37]     INFO    jenkins.InitReactorRunner$1#onAttained: Prepared all plugins
2024-09-29 00:16:30.450+0000 [id=37]     INFO    jenkins.InitReactorRunner$1#onAttained: Started all plugins
2024-09-29 00:16:30.454+0000 [id=37]     INFO    jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2024-09-29 00:16:30.643+0000 [id=34]     INFO    jenkins.InitReactorRunner$1#onAttained: System config loaded
2024-09-29 00:16:30.644+0000 [id=43]     INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2024-09-29 00:16:30.645+0000 [id=43]     INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2024-09-29 00:16:30.646+0000 [id=32]     INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated

```

Ahora sí me salió el mensaje con la contraseña:

```

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****
2024-09-29 00:16:39.372+0000 [id=44]     INFO    jenkins.InitReactorRunner$1#onAttained: Completed initialization
2024-09-29 00:16:39.432+0000 [id=26]     INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running

```

1c50ae0907ff46828edce82287cd93e7

También me cierro que esa es la contraseña al entrar al contenedor:

```

❯ docker exec -it jenkins_servers /bin/bash
Error response from daemon: No such container: jenkins_servers
❯ docker exec -it jenkins_server /bin/bash
root@a0aca5013037:/# cat /var/jenkins_home/secrets/initialAdminPassword
1c50ae0907ff46828edce82287cd93e7
root@a0aca5013037:/# exit
exit

```

Configuramos jenkins:

Entramos con la contraseña que obtuvimos en localhost:8080

Getting Started

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

Instalamos los plugins sugeridos.

Skipeamos crear un usuario.

Dejé la config como estaba.

Getting Started

Jenkins is ready!

You have skipped the **setup of an admin user**.

To log in, use the username: "admin" and the administrator password you used to access the setup wizard.

Your Jenkins setup is complete.

[Start using Jenkins](#)

Listo! Ya estamos en el panel de jenkins.

The screenshot shows the Jenkins dashboard interface. On the left, there's a sidebar with various icons and links: 'Dashboard' (selected), 'New Item', 'People' (with 3657 items), 'Build History', 'Manage Jenkins', 'My Views', 'Build Queue' (empty), 'Build Executor Status' (1 Idle, 2 Idle), and 'Add description'. The main content area features a large 'Welcome to Jenkins!' message, a 'Start building your software project' button, and a 'Create a job' button. At the bottom, there's a 'Set up a distributed build' link.

Dashboard

- + New Item
- People (3657)
- Build History
- Manage Jenkins
- My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Add description

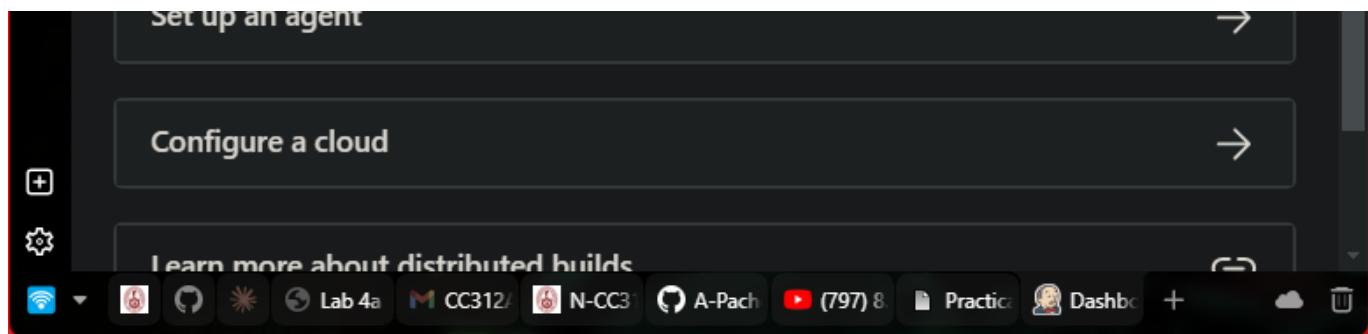
Welcome to Jenkins!

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



Usamos jenkins para construir nuestra app:

Creamos un job BuildAppJob. Se eligió un Freestyle project.

Dashboard >

Enter an item name

BuildAppJob

» Required field

Se establecieron el link del repositorio remoto y las credenciales de github en la configuración del job.

Dashboard > BuildAppJob > Configuration

General

Enabled

Description

My first Jenkins job

Plain text [Preview](#)

Discard old builds [?](#)

GitHub project

Project url [?](#)

<https://github.com/A-PachecoT/sample-app>

Advanced ▾

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Advanced ▾

Source Code Management

None

Git ?

Repositories ?

Repository URL ?

<https://github.com/A-PachecoT/sample-app.git>



Credentials ?

A-PachecoT/*****



Add ▾

Advanced ▾

Se agregó un Build Step de tipo Execute shell con el comando de construcción del contenedor:

Build Steps

The screenshot shows the 'Execute shell' build step configuration in Jenkins. It includes a command input field containing 'bash ./sample-app.sh', an 'Advanced' dropdown, and an 'Add build step' button.

```
bash ./sample-app.sh
```

Advanced ▾

Add build step ▾

Se guardó la configuración.

Usamos jenkins para testear la construcción de nuestra app:

Le di a Build Now. Salida de consola:

Console Output

Progress:  X

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/BuildAppJob
The recommended git tool is: NONE
using credential 1dc7f1d7-da96-4caa-ab09-c19131bd89e3
Cloning the remote Git repository
Cloning repository https://github.com/A-PachecoT/sample-app.git
> git init /var/jenkins_home/workspace/BuildAppJob # timeout=10
Fetching upstream changes from https://github.com/A-PachecoT/sample-app.git
> git --version # timeout=10
> git --version # 'git version 2.39.2'
using GIT_ASKPASS to set credentials
> git fetch --tags --force --progress -- https://github.com/A-PachecoT/sample-app.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/A-PachecoT/sample-app.git # 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 9f5a281a2e5c0780bf4bbfb65ff11176fdbba3cb (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9f5a281a2e5c0780bf4bbfb65ff11176fdbba3cb # timeout=10
Commit message: "chore: changed port from 8080 to 5050"
First time build. Skipping changelog.
[BuildAppJob] $ /bin/sh -xe /tmp/jenkins5486945177744948488.sh
+ bash ./sample-app.sh
mkdir: cannot create directory 'tempdir': File exists
mkdir: cannot create directory 'tempdir/templates': File exists
mkdir: cannot create directory 'tempdir/static': File exists
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
    Install the buildx component to build images with BuildKit:
    https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 6.656kB

Step 1/21 : FROM python
latest: Pulling from library/python
cdd62bf39133: Pulling fs layer
a47cff7f31e9: Pulling fs layer
a173f2aee8e9: Pulling fs layer
01272fe8adba: Pulling fs layer
ca852fa4243b: Pulling fs layer
8e0f50619958: Pulling fs layer
fc5ef6d7bb7c: Pulling fs layer
ca852fa4243b: Waiting
01272fe8adba: Waiting
fc5ef6d7bb7c: Waiting
```

```

--> bdb9fe7ff969
Successfully built bdb9fe7ff969
Successfully tagged sampleapp:latest
3986465cee017e53f37a6a48edb6c48d21b0bc6c268f4805de799f6545fc6c73
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              NAMES
STATUS          PORTS          NAMES
3986465cee01   sampleapp      "/bin/sh -c 'python ...'"   Less than a second ago
Up Less than a second   0.0.0.0:5050->5050/tcp, :::5050->5050/tcp   samplerunning
a0aca5013037   jenkins/jenkins:2.414.3-slim-jdk17  "/usr/bin/tini -- /u..."  31 minutes ago
Up 31 minutes    0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp  jenkins_server
Finished: SUCCESS

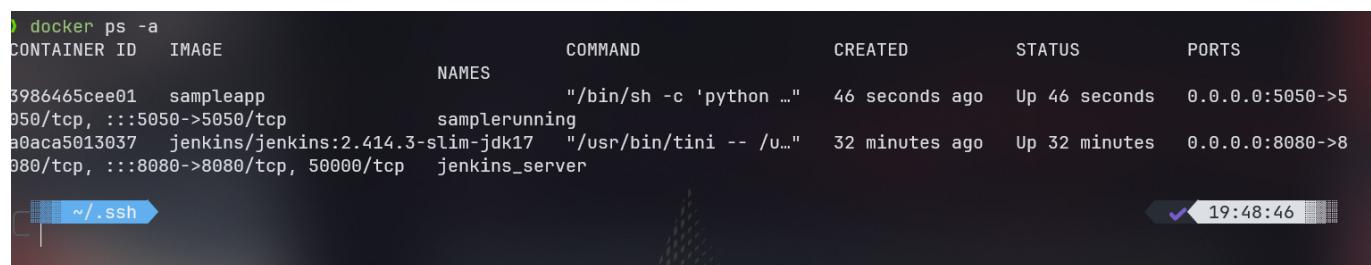
```

Además observamos que se creó el contenedor `sampleapp` con el build. Se ve en la consola y también en mi WSL:

```

$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED             STATUS              PORTS
NAMES
3986465cee01   sampleapp      "/bin/sh -c 'python ...'"   46 seconds ago   Up 46 seconds   0.0.0.0:5050->5
050/tcp, :::5050->5050/tcp   samplerunning
a0aca5013037   jenkins/jenkins:2.414.3-slim-jdk17  "/usr/bin/tini -- /u..."  32 minutes ago   Up 32 minutes   0.0.0.0:8080->8
080/tcp, :::8080->8080/tcp, 50000/tcp  jenkins_server

```



Y se puede acceder a la app:



Usamos jenkins para testear nuestra app

Se para y remueve el contenedor:

```

$ docker stop samplerunning
samplerunning
$ docker rm samplerunning
samplerunning

```

Creamos nuevo job para testing TestAppJob. Como Freestyle project.

Enter an item name

TestAppJob
» *Required field*

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining system, and this can be even used for something other than software build.

 **Pipeline**

Configuramos el job:

Configure General

Enabled

General

Description
My first Jenkins test

Plain text [Preview](#)

Discard old builds ?

GitHub project

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Advanced ▾

Source Code Management

None

Git ?

Build Triggers

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

Build after other projects are built ?

Projects to watch
BuildAppJob

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Always trigger, even if the build is aborted

Build Steps

Execute shell

Command

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

```
if curl http://172.17.0.1:5050/ | grep "You are calling me from 172.17.0.1"; then
    exit 0
else
    exit 1
fi
```

Advanced ▾

[Add build step](#) ▾

Post-build Actions

[Add post-build action](#) ▾

[Save](#) [Apply](#)

Corremos el BuildAppJob:

S	W	Name ↓	Last Success	Last Failure	Last Duration	Schedule a Build for BuildAppJob
✓	☀️	BuildAppJob	12 min #1	N/A	53 sec	
...	☀️	TestAppJob	N/A	N/A	N/A	

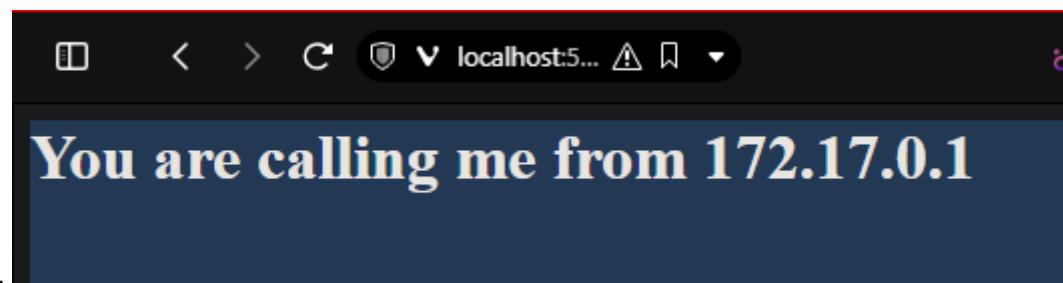
Como vemos se ejecutaron ambos jobs con éxito:

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀️	BuildAppJob	39 sec #2	N/A	1.8 sec	
✓	☀️	TestAppJob	31 sec #1	N/A	28 ms	

Log del testing:

✓ Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/TestAppJob
[TestAppJob] $ /bin/sh -xe /tmp/jenkins1881255256384345467.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
100  177  100  177      0      0  96721      0 --::-- --::-- --::-- 172k
<h1>You are calling me from 172.17.0.1</h1>
+ exit 0
Finished: SUCCESS
```



Funcionando la app!

Creamos pipeline en jenkins:

Creamos el item SamplePipeline de tipo Pipeline.

Enter an item name

SamplePipeline
» Required field

 **Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

 **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**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

 **Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

 **Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

 **Copy from**
Type to autocomplete

OK

Configuramos la pipeline:

Agregamos Script:

Pipeline

Definition

Pipeline script



Script ?

```
1 ▾ node {  
2 ▾   stage('Preparation') {  
3 ▾     catchError(buildResult: 'SUCCESS') {  
4       sh 'docker stop samplerunning'  
5       sh 'docker rm samplerunning'  
6     }  
7   }  
8 ▾   stage('Build') {  
9     build 'BuildAppJob'  
10  }  
11 ▾   stage('Results') {  
12     build 'TestAppJob'  
13   }  
14 }  
15 |
```

try sample Pipeline... ▾

Use Groovy Sandbox ?

Pipeline Syntax

Save

Apply

Ejecutamos el pipeline:

Status

Pipeline SamplePipeline

</> Changes Build scheduled

Build Now

Add description

Configure

Delete Pipeline

Full Stage View

Rename

Average stage times:

Pipeline Syntax

Preparation 499ms

#1 Sep 28 20:10 No Changes 499ms

Build History trend ▾

No builds

Atom feed for all Atom feed for failures

Permalinks

Resultados:

Stage View: alt text

Logs:

Console Output

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/SamplePipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Preparation)
[Pipeline] catchError
[Pipeline] {
[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)
Scheduling project: BuildAppJob
Starting building: BuildAppJob #3
Build BuildAppJob #3 completed: SUCCESS
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Results)
[Pipeline] build (Building TestAppJob)
Scheduling project: TestAppJob
Starting building: TestAppJob #4
Build TestAppJob #4 completed: SUCCESS
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Además se comprobó que la app funcionaba.