Technological Institute of Costa Rica

Software Quality Assurance

Homework #12

Marco Herrera González

2021075403

Professor: Erick Hernández Bonilla

I Semester, 2024

Content

Image Index

Continuous Integration Pipeline

Tools Used

To implement a simple continuous integration (CI) pipeline the following tools were used:

- GitHub Actions: it is a tool to implement continuous integration deliveries. It allows to configure the steps of deploying pipelines, like automatic buildings, testing and deploying to Docker images. The steps of the pipeline, also called actions events, can be configured to be triggered on specific events on the source control repository from GitHub or by schedules events. Each of these actions run inside a cloud container provided by GitHub (called runners) and can be configured to run on parallel or sequentially (GitHub, n.d.). A sequence of actions can also be packaged in a job. Such actions are configured using a file.

- GitHub: it is a source control tool that offers hosting for software development projects using Git (Juviler, 2024). It provides features to create repositories, branches, and forks; and it also manages version control features such as commits, pulls, and merges. It allows teams to collaborate simultaneously in the development of software projects, as several collaborators can be included in a repository. GitHub also provides features to keep secret keys and integrate with services like GitHub Actions to implement CI pipelines.

- Docker Hub: it is a cloud service that allows to create repositories for Docker images. Users can pull the images and create containers with them in local machines. The tool provides features such as collaborative repositories, security scanning, build and deploy Docker images, mirroring, artifact management, and version control management (Docker docs, 2024a).

- Docker: it is a tool that allows to run applications as lightweight, portable, and self-sufficient containers. The containers are built using Docker images, which contain the necessary configurations, code, and commands to build a container of an application. Containers are lightweight, standalone, and executable packages that contain everything needed to run a piece of software, including code, libraries, and dependencies (Docker docs, 2024b). One of the main advantages of containers are their portability, as they can be easily shipped to several environments like development, testing, and production environments.

- Flask: it is a minimalist and lightweight framework for Python that allows to develop simple web applications. The tool allows developers to map URL routes to Python functions, it has a simple built-in templating engine (called Jinja2) to generate dynamic HTML pages, it can handle HTTP requests and parse request and response data, provides features to develop web servers, it is highly extensible and customizable, and it is simple to use and learn (Flask Documentation, n.d.).

- PyTest: it is a framework for Python used for unit testing. It provides features for running tests, make assertions, create and configure mock objects, perform code analysis, mock functions, and run batches of tests automatically (PyTest Documentation, n.d.). Additionally, it can perform integration and acceptance tests, and provides test parametrization to reduce code duplication. The tool is also known to work with most other Python frameworks.

Application Description

The application is programmed in Python. It creates a simple web server using Flask with a single route (‘/’). When the route is accessed, it displays a simple HTML page which contains the label ‘Está funcionando!!!’. To test the page, a single unit test for the endpoint was programmed using Pytest, which verifies that when the endpoint is accessed, it returns the state 200 (which means the endpoint is correctly working).

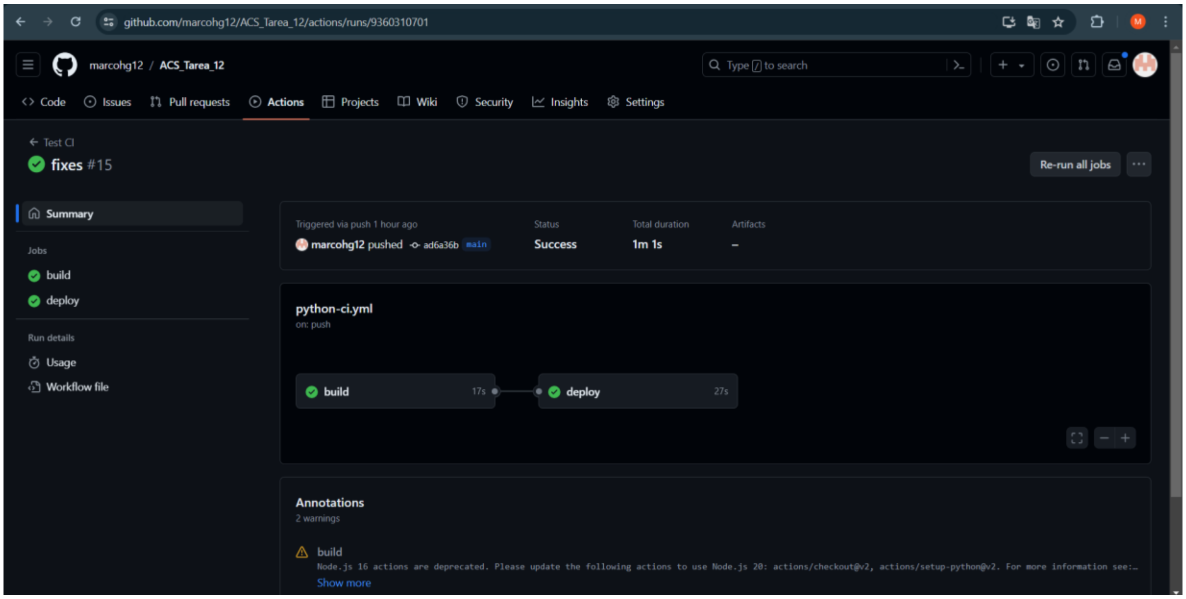
Pipeline Description

First, there is a file called ‘python-ci.yml’, which contains the configuration for GitHub Actions. The file contains two jobs, one for building and testing the application, and the other for deploying the Docker image of the application to Docker Hub. The first job sets up Python and installs the dependencies necessary to run the tests, then, the tests are run. The second job is executed if all tests passed, and if so, it pushes a Docker image of the application into the Docker Hub repository. The image is built using the Dockerfile contained in the repository. After the image is pushed, it can be built into a Docker container so that it can be executed. This is done using Docker in a local machine. This pipeline is performed every time a commit or push action is made into the GitHub repository.

Evidence

This section contains images of evidence of the pipeline, deployment and functionality of the application.

Figure 1 - GitHub Action Performed



fixes #15

Triggered via push 1 hour ago
marcohg12 pushed ad6a36b main

Status: Success

Total duration: 1m 1s

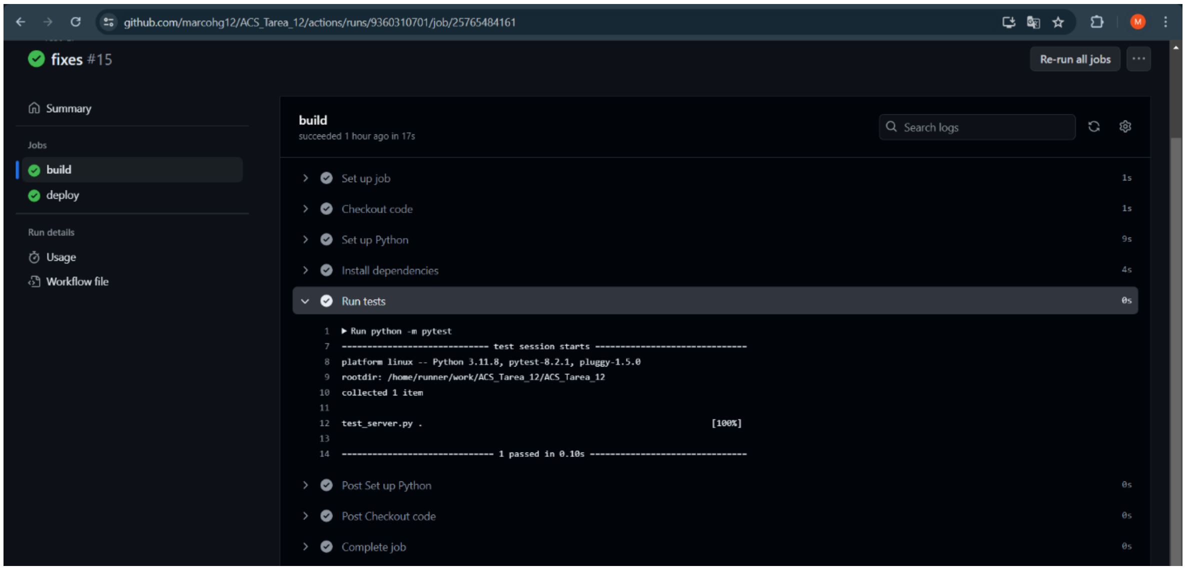
Artifacts: –

python-ci.yml
on: push

Annotations
2 warnings

build
Node.js 16 actions are deprecated. Please update the following actions to use Node.js 20: actions/checkout@v2, actions/setup-python@v2. For more information see:…
Show more

Figure 2 - Testing Step Results



build
succeeded 1 hour ago in 17s

- Set up job 1s
- Checkout code 1s
- Set up Python 9s
- Install dependencies 4s
- Run tests 0s

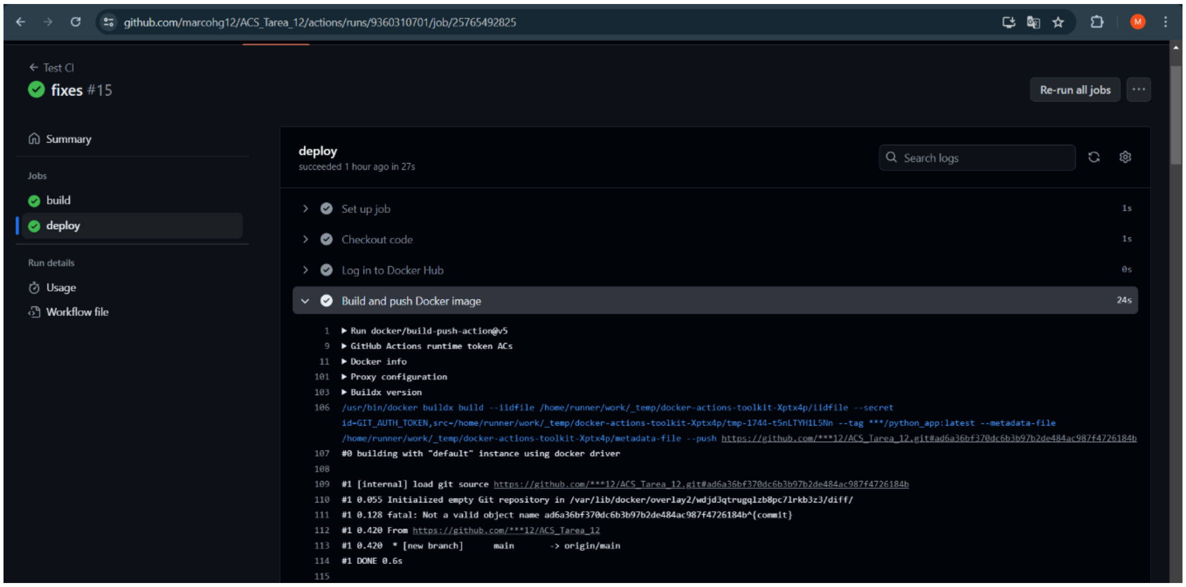
```
▶ Run python -m pytest
---------------------------- test session starts -----------------------------
platform linux -- Python 3.11.8, pytest-8.2.1, pluggy-1.5.0
rootdir: /home/runner/work/ACS_Tarea_12/ACS_Tarea_12
collected 1 item

test_server.py .                                                        [100%]

----------------------------- 1 passed in 0.10s ------------------------------
```

- Post Set up Python 0s
- Post Checkout code 0s
- Complete job 0s

Figure 3 - Docker Image Building



deploy
succeeded 1 hour ago in 27s

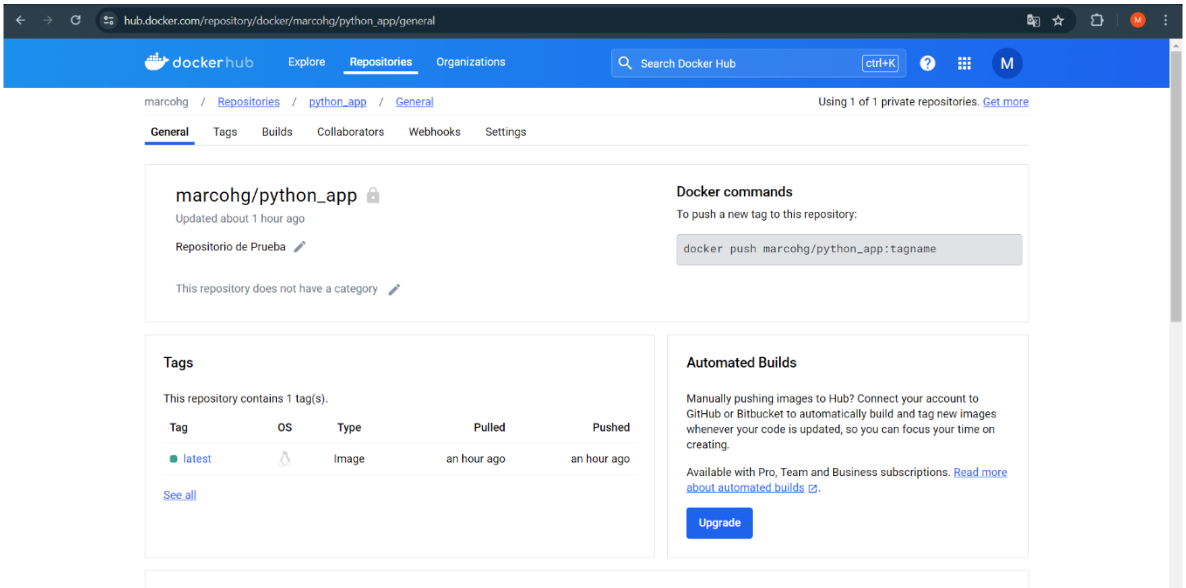
- Set up job 1s
- Checkout code 1s
- Log in to Docker Hub 0s
- Build and push Docker image 24s

```
▶ Run docker/build-push-action@v5
▶ GitHub Actions runtime token ACs
▶ Docker info
▶ Proxy configuration
▶ Buildx version
/usr/bin/docker buildx build --iidfile /home/runner/work/_temp/docker-actions-toolkit-Xptx4p/iidfile --secret
id=GIT_AUTH_TOKEN,src=/home/runner/work/_temp/docker-actions-toolkit-Xptx4p/tmp-1744-t5nLTYH1L5Nn --tag ***/python_app:latest --metadata-file
/home/runner/work/_temp/docker-actions-toolkit-Xptx4p/metadata-file --push https://github.com/***12/ACS_Tarea_12.git#ad6a36bf370dc6b3b97b2de484ac987f4726184b
#0 building with "default" instance using docker driver

#1 [internal] load git source https://github.com/***12/ACS_Tarea_12.git#ad6a36bf370dc6b3b97b2de484ac987f4726184b
#1 0.055 Initialized empty Git repository in /var/lib/docker/overlay2/wdjd3qtrugqlzb8pc7lrkb3z3/diff/
#1 0.128 fatal: Not a valid object name ad6a36bf370dc6b3b97b2de484ac987f4726184b^{commit}
#1 0.420 From https://github.com/***12/ACS_Tarea_12
#1 0.420  * [new branch]      main       -> origin/main
#1 DONE 0.6s

```

Figure 4 - Docker Image Pushed to Docker Hub



hub.docker.com/repository/docker/marcohg/python_app/general
docker hub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
M
marcohg / Repositories / python_app / General
Using 1 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings

marcohg/python_app

Updated about 1 hour ago

Repositorio de Prueba

This repository does not have a category

Docker commands

To push a new tag to this repository:

```
docker push marcohg/python_app:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	an hour ago	an hour ago

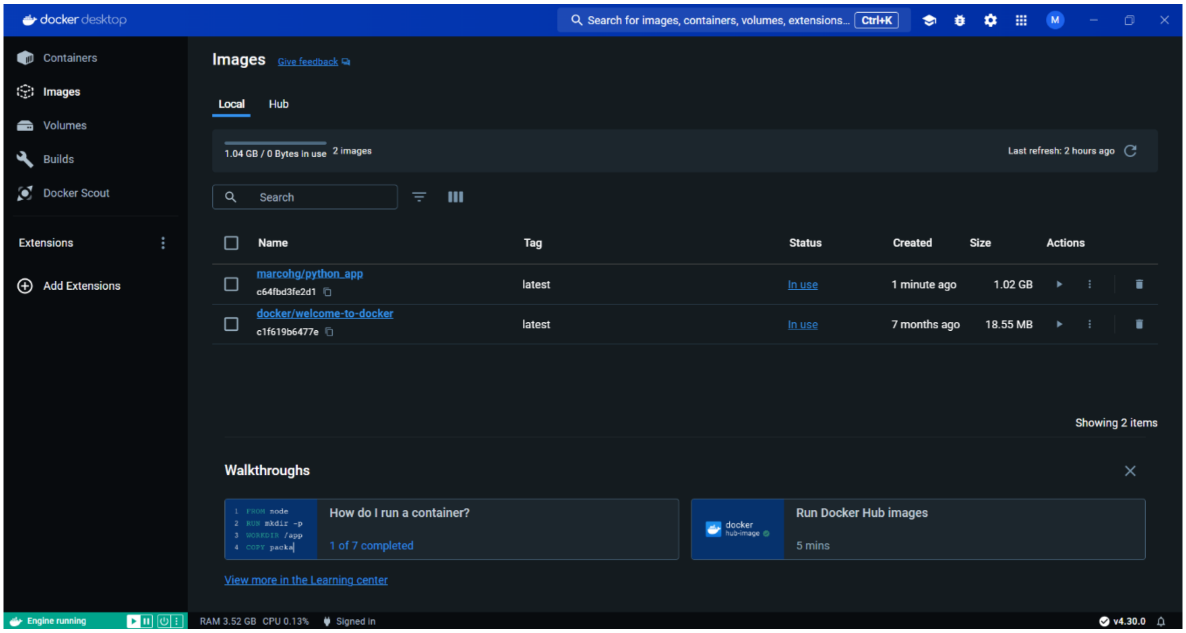
See all

Automated Builds

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

Figure 5 - Docker Image Pulled to Docker



1.04 GB / 0 Bytes in use 2 images

Last refresh: 2 hours ago

Name	Tag	Status	Created	Size	Actions
marcohg/python_app c64fbd3fe2d1	latest	In use	1 minute ago	1.02 GB	
docker/welcome-to-docker c1f619b6477e	latest	In use	7 months ago	18.55 MB	

Showing 2 items

Walkthroughs

How do I run a container?

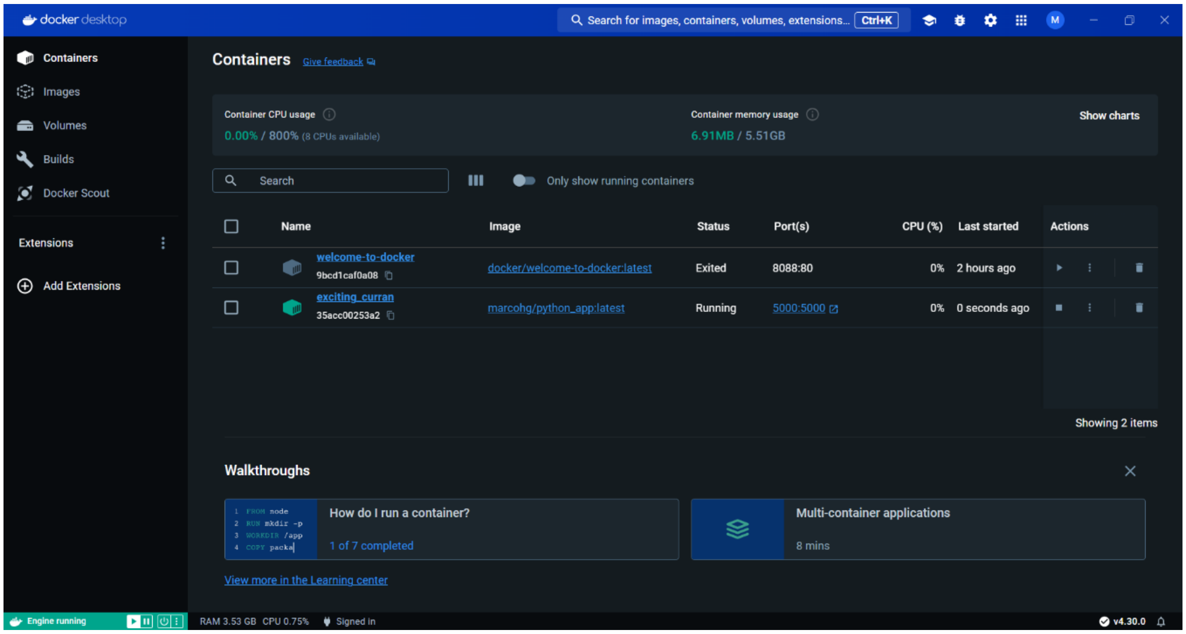
1 of 7 completed

Run Docker Hub images

5 mins

View more in the Learning center

Figure 6 - Docker Container Deployed



Containers

Container CPU usage 0.00% / 800% (8 CPUs available)

Container memory usage 6.91MB / 5.51GB

	Name	Image	Status	Port(s)	CPU (%)	Last started	Actions
□	welcome-to-docker 9bcd1caf0a08	docker/welcome-to-docker:latest	Exited	8088:80	0%	2 hours ago	
□	exciting_curran 35acc00253a2	marcohg/python_app:latest	Running	5000:5000	0%	0 seconds ago	

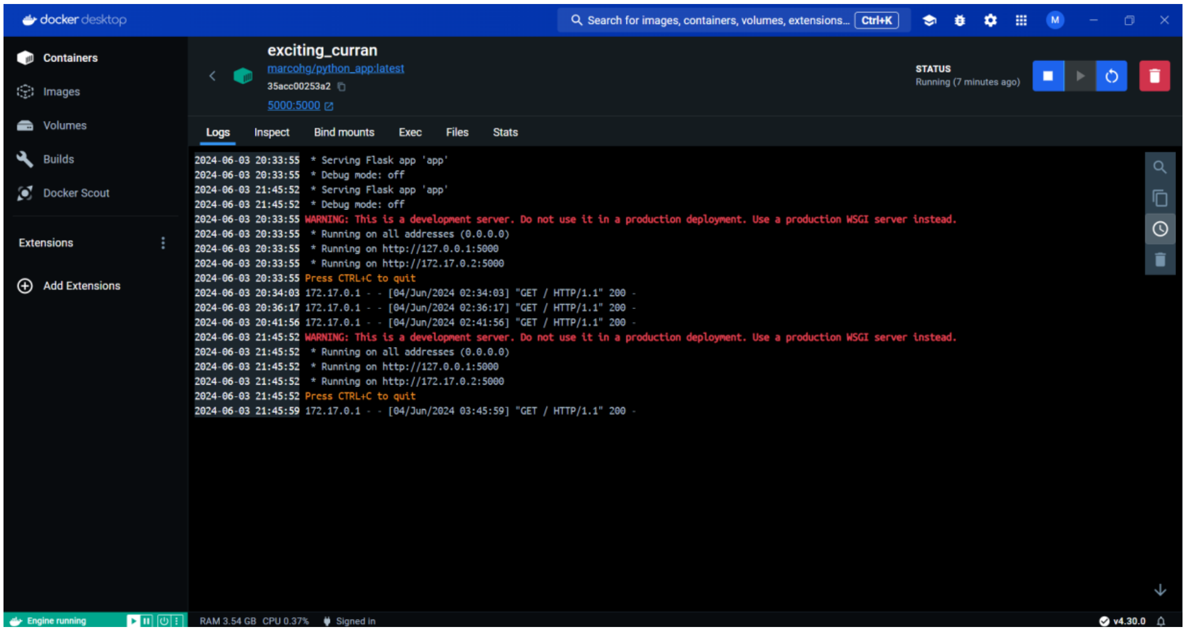
Showing 2 items

How do I run a container? 1 of 7 completed

Multi-container applications 8 mins

RAM 3.53 GB CPU 0.75% Signed in v4.30.0

Figure 7 - Container Console Log



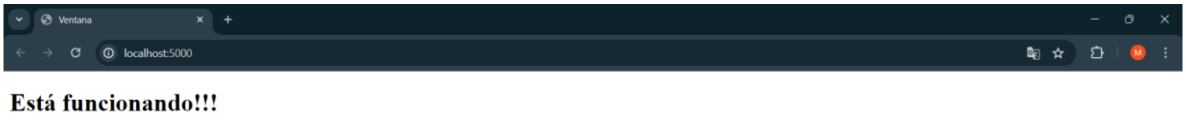
exciting_curran
marcohg/python_app:latest
35acc00253a2
5000:5000

STATUS
Running (7 minutes ago)

```
2024-06-03 20:33:55  * Serving Flask app 'app'
2024-06-03 20:33:55  * Debug mode: off
2024-06-03 21:45:52  * Serving Flask app 'app'
2024-06-03 21:45:52  * Debug mode: off
2024-06-03 20:33:55 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
2024-06-03 20:33:55  * Running on all addresses (0.0.0.0)
2024-06-03 20:33:55  * Running on http://127.0.0.1:5000
2024-06-03 20:33:55  * Running on http://172.17.0.2:5000
2024-06-03 20:33:55 Press CTRL+C to quit
2024-06-03 20:34:03 172.17.0.1 - - [04/Jun/2024 02:34:03] "GET / HTTP/1.1" 200 -
2024-06-03 20:36:17 172.17.0.1 - - [04/Jun/2024 02:36:17] "GET / HTTP/1.1" 200 -
2024-06-03 20:41:56 172.17.0.1 - - [04/Jun/2024 02:41:56] "GET / HTTP/1.1" 200 -
2024-06-03 21:45:52 WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
2024-06-03 21:45:52  * Running on all addresses (0.0.0.0)
2024-06-03 21:45:52  * Running on http://127.0.0.1:5000
2024-06-03 21:45:52  * Running on http://172.17.0.2:5000
2024-06-03 21:45:52 Press CTRL+C to quit
2024-06-03 21:45:59 172.17.0.1 - - [04/Jun/2024 03:45:59] "GET / HTTP/1.1" 200 -
```

RAM 3.54 GB CPU 0.37% Signed in v4.30.0

Figure 8 - Application Running



Está funcionando!!!

References

Docker docs. (2024a). Overview of Docker Hub. https://docs.docker.com/docker-hub/

Docker docs. (2024b). *Docker Overview*. https://docs.docker.com/get-started/overview/

Flask Documentation. (n.d.). *Flask User's Guide*. https://flask.palletsprojects.com/en/3.0.x/

GitHub. (n.d.). *Understand GitHub Actions*. https://docs.github.com/es/actions/learn-github actions/understanding-github-actions

Juviler, J. (2024). *What Is GitHub? (And What Is It Used For?)*. https://blog.hubspot.com/website/what-is-github-used-for

PyTest Documentation. (n.d.). *PyTest: Helps you write better programs*. https://docs.pytest.org/en/8.2.x/