\*\*Dockerizar la aplicacion.

* Todo el proyecto se encuentra subido en el repositorio: [jfmarquez23/DevOps: En este repositorio tenemos pruebas de lo siguiente: (github.com)](https://github.com/jfmarquez23/DevOps)



# Construir la imagen en Docker:

$ docker build -t pruebas .

#0 building with "default" instance using docker driver

#1 [internal] load build definition from dockerfile

#1 transferring dockerfile: 1.25kB 0.0s done

#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/python:3.11.3

#2 ...

#3 [auth] library/python:pull token for registry-1.docker.io

#3 DONE 0.0s

#2 [internal] load metadata for docker.io/library/python:3.11.3

#2 DONE 1.2s

#4 [internal] load .dockerignore

#4 transferring context: 2B 0.0s done

#4 DONE 0.0s

#5 [1/8] FROM docker.io/library/python:3.11.3@sha256:3a619e3c96fd4c5fc5e1998fd4dcb1f1403eb90c4c6409c70d7e80b9468df7df

#5 DONE 0.0s

#6 [internal] load build context

#6 transferring context: 100.06kB 0.3s done

#6 DONE 0.3s

#7 [6/8] RUN adduser --disabled-password --gecos '' devsu

#7 CACHED

#8 [2/8] RUN mkdir /app

#8 CACHED

#9 [3/8] WORKDIR /app

#9 CACHED

#10 [4/8] COPY requirements.txt .

#10 CACHED

#11 [5/8] RUN pip install --no-cache-dir -r requirements.txt

#11 CACHED

#12 [7/8] RUN chown -R devsu:devsu /app

#12 CACHED

#13 [8/8] COPY . .

#13 DONE 0.4s

#14 exporting to image

#14 exporting layers 0.1s done

#14 writing image sha256:1c1769e43bb97b8462c37833c7f8dc525f699f4b004920ed69089170287b12d9 done

#14 naming to docker.io/library/pruebas done

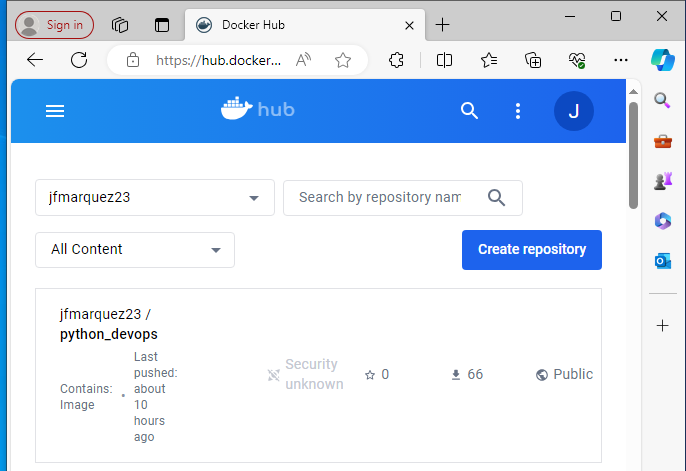
#14 DONE 0.1s

View build details: docker-desktop://dashboard/build/default/default/hyfm5tanj3c49tgec6gtlgemz

**What's Next?**

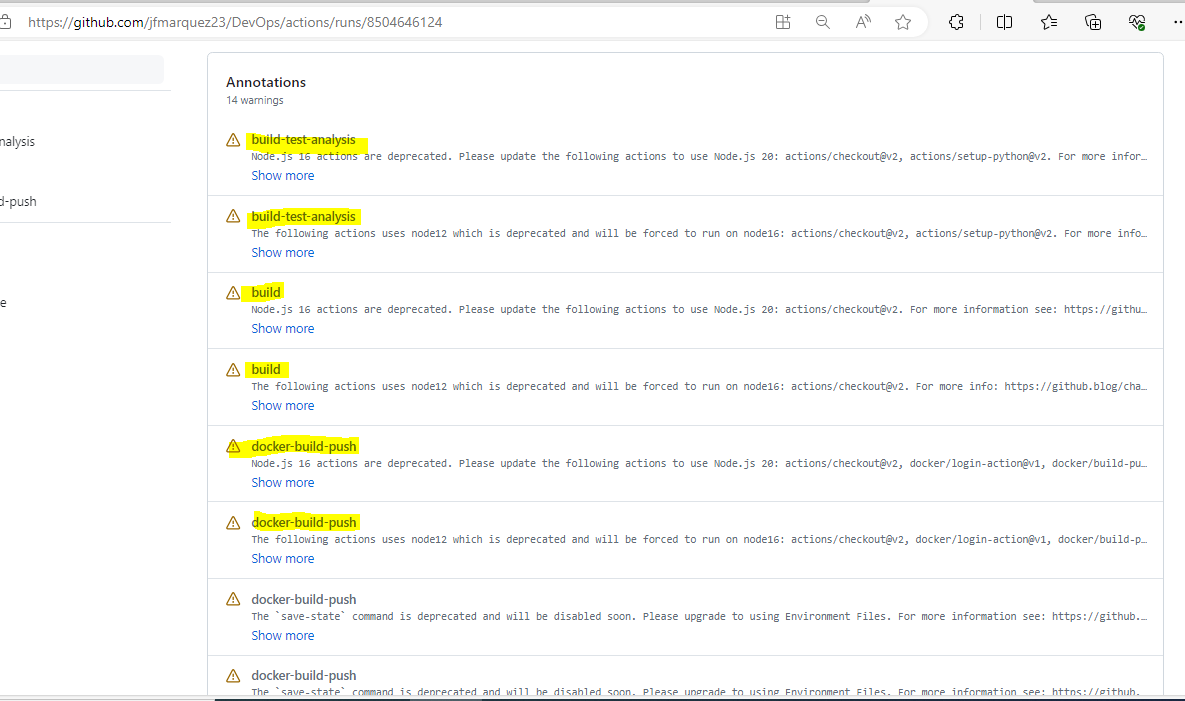
View a summary of image vulnerabilities and recommendations → docker scout quickview

# Imagen subida en docker Hub:



# \*\*Se debe generar un pipeline como código que idealmente incluya los siguientes pasos:

Se crea el pipeline: Pipeline\_Prueba\_CI.yml con Git Actions

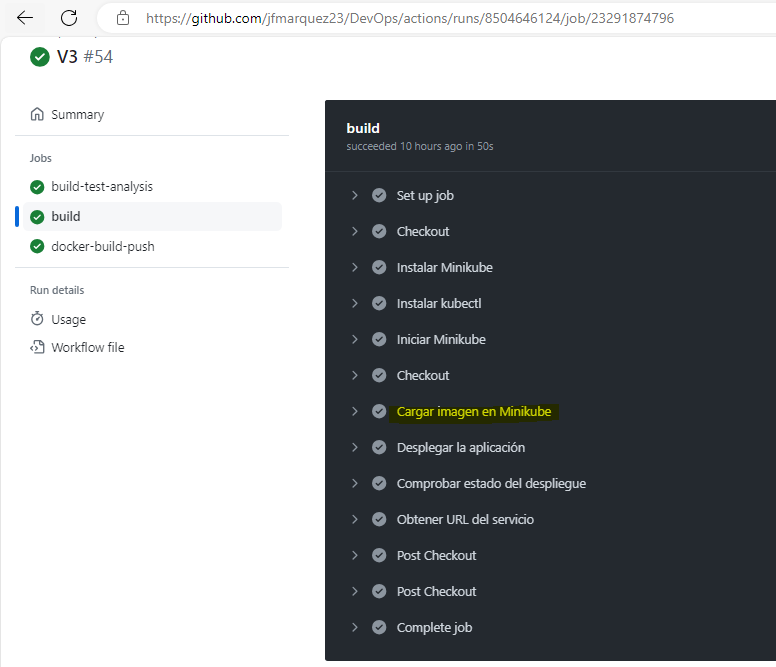


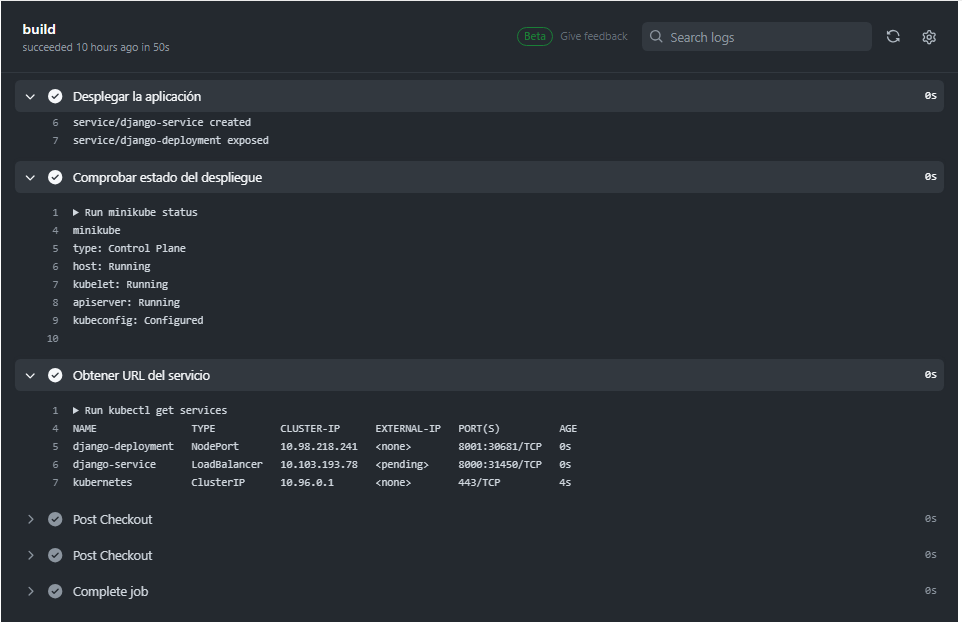
# \*\*Desplegar la aplicacion dockerizada en Kubernetes, no se requiere que utilice un entorno publico, puede utilizar minikube/docker-desktop en un

# entorno local. Intente utilizar todos los recursos que cree necesario, es decir lo que utilizaria para q la app este productiva.

# Se requiere por lo menos 2 réplicas y escalamiento horizontal.

Se despliega la imagen dockerizada en kubernetes en Minikube





## Validación:

La URL es: <http://127.0.0.1:50145/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/workloads?namespace=default>

