Table of Contents

[1. Mini Project 1: Simple ML Model Deployment 2](#_Toc160883095)

[2. Git hub link of stroke detection project: 2](#_Toc160883096)

[3. Console output of building a package 2](#_Toc160883097)

[4. FastApi Application 5](#_Toc160883098)

[5. CI workflow 6](#_Toc160883099)

[6. Dockerization 6](#_Toc160883100)

[7. Docker images 7](#_Toc160883101)

[8. Docker image In VS code 7](#_Toc160883102)

[9. Running Docker app from CLI 7](#_Toc160883103)

[10. Stop app and push to image to docker 7](#_Toc160883104)

[11. Added Docker pass\_token and secret 8](#_Toc160883105)

[12. Full CI pipeline is running successfully 8](#_Toc160883106)

[13. Cloud-9 setup 8](#_Toc160883107)

[14. Git clone in AWS environment 8](#_Toc160883108)

[15. Train and Predict running in AWS 9](#_Toc160883109)

[16. Creating & Launching EC2 9](#_Toc160883110)

[17. Docker Installation 9](#_Toc160883111)

[18. Adding self-hosted runner in Git hub Repo: 11](#_Toc160883112)

[19. Running GitHub Actions: 11](#_Toc160883113)

[20. CD workflow 12](#_Toc160883114)

[21. Running FastApi app on EC2 instance 12](#_Toc160883115)

# Mini Project 1: Simple ML Model Deployment

**Objective:** Deploy a simple ML model using a pre-trained model and serve predictions through an API endpoint.

1. Create a Git repository for the project and include the ML model code and API server code.
2. Implement the API server using a Python web framework like Flask or FastAPI.
3. Set up a GitHub Actions workflow to automate model testing and deployment on each push to the repository.
4. Build a Docker container for the API server with the necessary dependencies and the pre-trained model.
5. Use Docker Hub to store the Docker image.
6. Deploy the Docker container on AWS EC2.
7. Set up an API Gateway to expose the API endpoint securely.

# Git hub link of stroke detection project:

<https://github.com/pycsr/stroke_detection>

# Console output of building a package

(venv) (base) quicktech@psoni stroke\_detection % python -m build

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (setuptools>=42, wheel)

\* Getting build dependencies for sdist...

/Users/quicktech/Documents/IISc\_AIMLOps\_Projects/Part-B\_MiniProject/stroke\_detection

running egg\_info

creating stroke\_detection\_model.egg-info

writing stroke\_detection\_model.egg-info/PKG-INFO

writing dependency\_links to stroke\_detection\_model.egg-info/dependency\_links.txt

writing requirements to stroke\_detection\_model.egg-info/requires.txt

writing top-level names to stroke\_detection\_model.egg-info/top\_level.txt

writing manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

reading manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

warning: no files found matching '\*.txt'

warning: no files found matching '\*.pkl'

warning: no previously-included files found matching '\*.log'

warning: no previously-included files found matching '\*.cfg'

warning: no previously-included files matching '\_\_pycache\_\_' found under directory '\*'

adding license file 'LICENSE'

writing manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

\* Building sdist...

/Users/quicktech/Documents/IISc\_AIMLOps\_Projects/Part-B\_MiniProject/stroke\_detection

running sdist

running egg\_info

writing stroke\_detection\_model.egg-info/PKG-INFO

writing dependency\_links to stroke\_detection\_model.egg-info/dependency\_links.txt

writing requirements to stroke\_detection\_model.egg-info/requires.txt

writing top-level names to stroke\_detection\_model.egg-info/top\_level.txt

reading manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

warning: no files found matching '\*.txt'

warning: no files found matching '\*.pkl'

warning: no previously-included files found matching '\*.log'

warning: no previously-included files found matching '\*.cfg'

warning: no previously-included files matching '\_\_pycache\_\_' found under directory '\*'

adding license file 'LICENSE'

writing manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

running check

creating stroke\_detection\_model-0.0.1

creating stroke\_detection\_model-0.0.1/requirements

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model/config

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model/datasets

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model/processing

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model/trained\_models

creating stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

creating stroke\_detection\_model-0.0.1/tests

copying files to stroke\_detection\_model-0.0.1...

copying LICENSE -> stroke\_detection\_model-0.0.1

copying MANIFEST.in -> stroke\_detection\_model-0.0.1

copying README.md -> stroke\_detection\_model-0.0.1

copying pyproject.toml -> stroke\_detection\_model-0.0.1

copying setup.py -> stroke\_detection\_model-0.0.1

copying ./requirements/requirements.txt -> stroke\_detection\_model-0.0.1/./requirements

copying ./requirements/test\_requirements.txt -> stroke\_detection\_model-0.0.1/./requirements

copying ./stroke\_detection\_model/.DS\_Store -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/VERSION -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/config.yml -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/pipeline.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/predict.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/train\_pipeline.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model

copying ./stroke\_detection\_model/config/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/config

copying ./stroke\_detection\_model/config/core.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/config

copying ./stroke\_detection\_model/datasets/.DS\_Store -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/datasets

copying ./stroke\_detection\_model/datasets/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/datasets

copying ./stroke\_detection\_model/datasets/healthcare-dataset-stroke-data.csv -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/datasets

copying ./stroke\_detection\_model/processing/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/processing

copying ./stroke\_detection\_model/processing/data\_manager.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/processing

copying ./stroke\_detection\_model/processing/features.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/processing

copying ./stroke\_detection\_model/processing/validation.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/processing

copying ./stroke\_detection\_model/trained\_models/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/trained\_models

copying ./stroke\_detection\_model/trained\_models/stroke\_detection\_model\_output\_v0.0.1.pkl -> stroke\_detection\_model-0.0.1/./stroke\_detection\_model/trained\_models

copying stroke\_detection\_model/VERSION -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model/config.yml -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model/pipeline.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model/predict.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model/train\_pipeline.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model

copying stroke\_detection\_model.egg-info/PKG-INFO -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

copying stroke\_detection\_model.egg-info/SOURCES.txt -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

copying stroke\_detection\_model.egg-info/dependency\_links.txt -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

copying stroke\_detection\_model.egg-info/requires.txt -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

copying stroke\_detection\_model.egg-info/top\_level.txt -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

copying stroke\_detection\_model/config/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/config

copying stroke\_detection\_model/config/core.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/config

copying stroke\_detection\_model/datasets/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/datasets

copying stroke\_detection\_model/datasets/healthcare-dataset-stroke-data.csv -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/datasets

copying stroke\_detection\_model/processing/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/data\_manager.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/features.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/validation.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/processing

copying stroke\_detection\_model/trained\_models/\_\_init\_\_.py -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/trained\_models

copying stroke\_detection\_model/trained\_models/stroke\_detection\_model\_output\_v0.0.1.pkl -> stroke\_detection\_model-0.0.1/stroke\_detection\_model/trained\_models

copying tests/test\_features.py -> stroke\_detection\_model-0.0.1/tests

copying tests/test\_predictions.py -> stroke\_detection\_model-0.0.1/tests

copying stroke\_detection\_model.egg-info/SOURCES.txt -> stroke\_detection\_model-0.0.1/stroke\_detection\_model.egg-info

Writing stroke\_detection\_model-0.0.1/setup.cfg

Creating tar archive

removing 'stroke\_detection\_model-0.0.1' (and everything under it)

\* Building wheel from sdist

\* Creating venv isolated environment...

\* Installing packages in isolated environment... (setuptools>=42, wheel)

\* Getting build dependencies for wheel...

/private/var/folders/vs/6l92ypwj3sbc973853q9w9w00000gn/T/build-via-sdist-ca8j0naj/stroke\_detection\_model-0.0.1

running egg\_info

writing stroke\_detection\_model.egg-info/PKG-INFO

writing dependency\_links to stroke\_detection\_model.egg-info/dependency\_links.txt

writing requirements to stroke\_detection\_model.egg-info/requires.txt

writing top-level names to stroke\_detection\_model.egg-info/top\_level.txt

reading manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

warning: no files found matching '\*.txt'

warning: no files found matching '\*.pkl'

warning: no previously-included files found matching '\*.log'

warning: no previously-included files matching '\_\_pycache\_\_' found under directory '\*'

warning: no previously-included files matching '\*.py[co]' found under directory '\*'

adding license file 'LICENSE'

writing manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

\* Installing packages in isolated environment... (wheel)

\* Building wheel...

/private/var/folders/vs/6l92ypwj3sbc973853q9w9w00000gn/T/build-via-sdist-ca8j0naj/stroke\_detection\_model-0.0.1

running bdist\_wheel

running build

running build\_py

creating build

creating build/lib

creating build/lib/stroke\_detection\_model

copying stroke\_detection\_model/train\_pipeline.py -> build/lib/stroke\_detection\_model

copying stroke\_detection\_model/predict.py -> build/lib/stroke\_detection\_model

copying stroke\_detection\_model/\_\_init\_\_.py -> build/lib/stroke\_detection\_model

copying stroke\_detection\_model/pipeline.py -> build/lib/stroke\_detection\_model

creating build/lib/stroke\_detection\_model/config

copying stroke\_detection\_model/config/\_\_init\_\_.py -> build/lib/stroke\_detection\_model/config

copying stroke\_detection\_model/config/core.py -> build/lib/stroke\_detection\_model/config

creating build/lib/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/\_\_init\_\_.py -> build/lib/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/features.py -> build/lib/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/data\_manager.py -> build/lib/stroke\_detection\_model/processing

copying stroke\_detection\_model/processing/validation.py -> build/lib/stroke\_detection\_model/processing

creating build/lib/stroke\_detection\_model/datasets

copying stroke\_detection\_model/datasets/\_\_init\_\_.py -> build/lib/stroke\_detection\_model/datasets

creating build/lib/stroke\_detection\_model/trained\_models

copying stroke\_detection\_model/trained\_models/\_\_init\_\_.py -> build/lib/stroke\_detection\_model/trained\_models

running egg\_info

writing stroke\_detection\_model.egg-info/PKG-INFO

writing dependency\_links to stroke\_detection\_model.egg-info/dependency\_links.txt

writing requirements to stroke\_detection\_model.egg-info/requires.txt

writing top-level names to stroke\_detection\_model.egg-info/top\_level.txt

reading manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

reading manifest template 'MANIFEST.in'

warning: no files found matching '\*.txt'

warning: no files found matching '\*.pkl'

warning: no previously-included files found matching '\*.log'

warning: no previously-included files matching '\_\_pycache\_\_' found under directory '\*'

warning: no previously-included files matching '\*.py[co]' found under directory '\*'

adding license file 'LICENSE'

writing manifest file 'stroke\_detection\_model.egg-info/SOURCES.txt'

copying stroke\_detection\_model/VERSION -> build/lib/stroke\_detection\_model

copying stroke\_detection\_model/config.yml -> build/lib/stroke\_detection\_model

copying stroke\_detection\_model/datasets/healthcare-dataset-stroke-data.csv -> build/lib/stroke\_detection\_model/datasets

copying stroke\_detection\_model/trained\_models/stroke\_detection\_model\_output\_v0.0.1.pkl -> build/lib/stroke\_detection\_model/trained\_models

installing to build/bdist.macosx-10.9-x86\_64/wheel

running install

running install\_lib

creating build/bdist.macosx-10.9-x86\_64

creating build/bdist.macosx-10.9-x86\_64/wheel

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

copying build/lib/stroke\_detection\_model/train\_pipeline.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/config

copying build/lib/stroke\_detection\_model/config/\_\_init\_\_.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/config

copying build/lib/stroke\_detection\_model/config/core.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/config

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/processing

copying build/lib/stroke\_detection\_model/processing/\_\_init\_\_.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/processing

copying build/lib/stroke\_detection\_model/processing/features.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/processing

copying build/lib/stroke\_detection\_model/processing/data\_manager.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/processing

copying build/lib/stroke\_detection\_model/processing/validation.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/processing

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/datasets

copying build/lib/stroke\_detection\_model/datasets/\_\_init\_\_.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/datasets

copying build/lib/stroke\_detection\_model/datasets/healthcare-dataset-stroke-data.csv -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/datasets

copying build/lib/stroke\_detection\_model/predict.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

copying build/lib/stroke\_detection\_model/\_\_init\_\_.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/trained\_models

copying build/lib/stroke\_detection\_model/trained\_models/stroke\_detection\_model\_output\_v0.0.1.pkl -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/trained\_models

copying build/lib/stroke\_detection\_model/trained\_models/\_\_init\_\_.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model/trained\_models

copying build/lib/stroke\_detection\_model/VERSION -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

copying build/lib/stroke\_detection\_model/pipeline.py -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

copying build/lib/stroke\_detection\_model/config.yml -> build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model

running install\_egg\_info

Copying stroke\_detection\_model.egg-info to build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model-0.0.1-py3.11.egg-info

running install\_scripts

creating build/bdist.macosx-10.9-x86\_64/wheel/stroke\_detection\_model-0.0.1.dist-info/WHEEL

creating '/Users/quicktech/Documents/IISc\_AIMLOps\_Projects/Part-B\_MiniProject/stroke\_detection/dist/.tmp-e7ycyhc8/stroke\_detection\_model-0.0.1-py3-none-any.whl' and adding 'build/bdist.macosx-10.9-x86\_64/wheel' to it

adding 'stroke\_detection\_model/VERSION'

adding 'stroke\_detection\_model/\_\_init\_\_.py'

adding 'stroke\_detection\_model/config.yml'

adding 'stroke\_detection\_model/pipeline.py'

adding 'stroke\_detection\_model/predict.py'

adding 'stroke\_detection\_model/train\_pipeline.py'

adding 'stroke\_detection\_model/config/\_\_init\_\_.py'

adding 'stroke\_detection\_model/config/core.py'

adding 'stroke\_detection\_model/datasets/\_\_init\_\_.py'

adding 'stroke\_detection\_model/datasets/healthcare-dataset-stroke-data.csv'

adding 'stroke\_detection\_model/processing/\_\_init\_\_.py'

adding 'stroke\_detection\_model/processing/data\_manager.py'

adding 'stroke\_detection\_model/processing/features.py'

adding 'stroke\_detection\_model/processing/validation.py'

adding 'stroke\_detection\_model/trained\_models/\_\_init\_\_.py'

adding 'stroke\_detection\_model/trained\_models/stroke\_detection\_model\_output\_v0.0.1.pkl'

adding 'stroke\_detection\_model-0.0.1.dist-info/LICENSE'

adding 'stroke\_detection\_model-0.0.1.dist-info/METADATA'

adding 'stroke\_detection\_model-0.0.1.dist-info/WHEEL'

adding 'stroke\_detection\_model-0.0.1.dist-info/top\_level.txt'

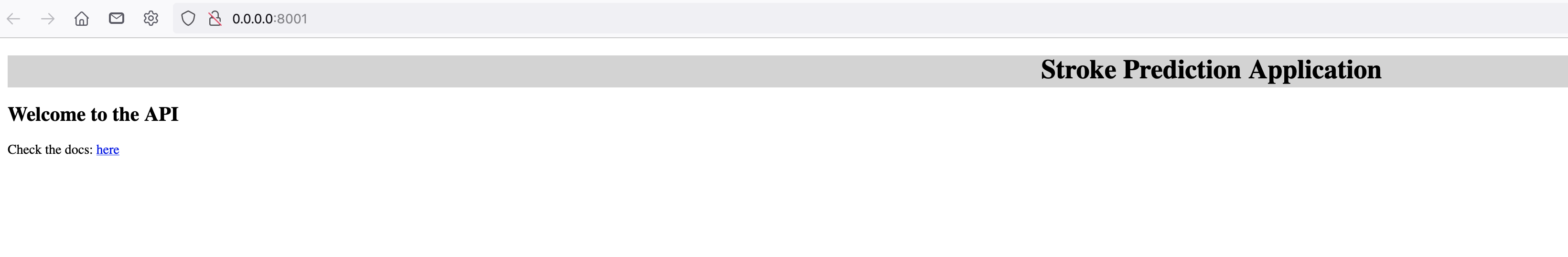
adding 'stroke\_detection\_model-0.0.1.dist-info/RECORD'

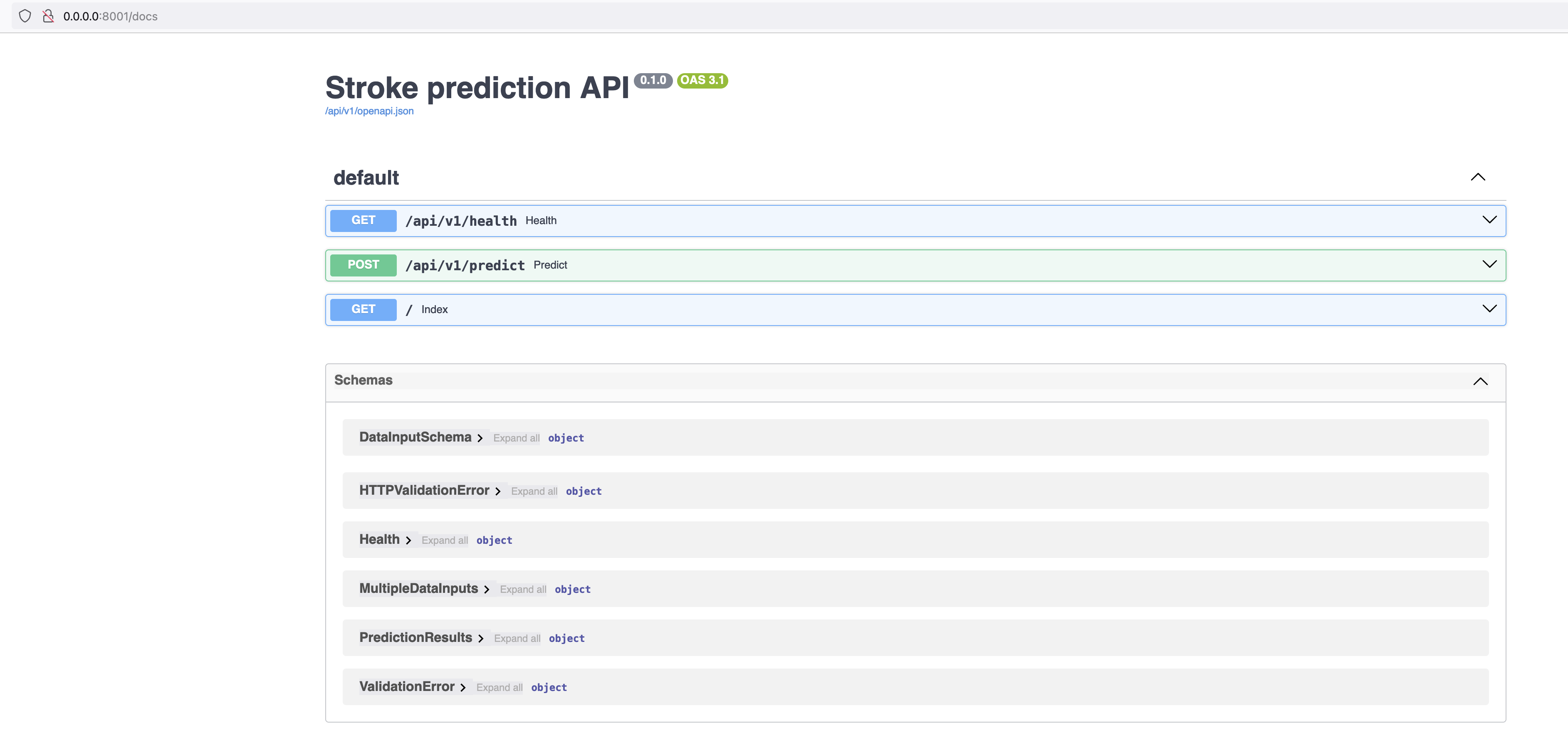
removing build/bdist.macosx-10.9-x86\_64/wheel

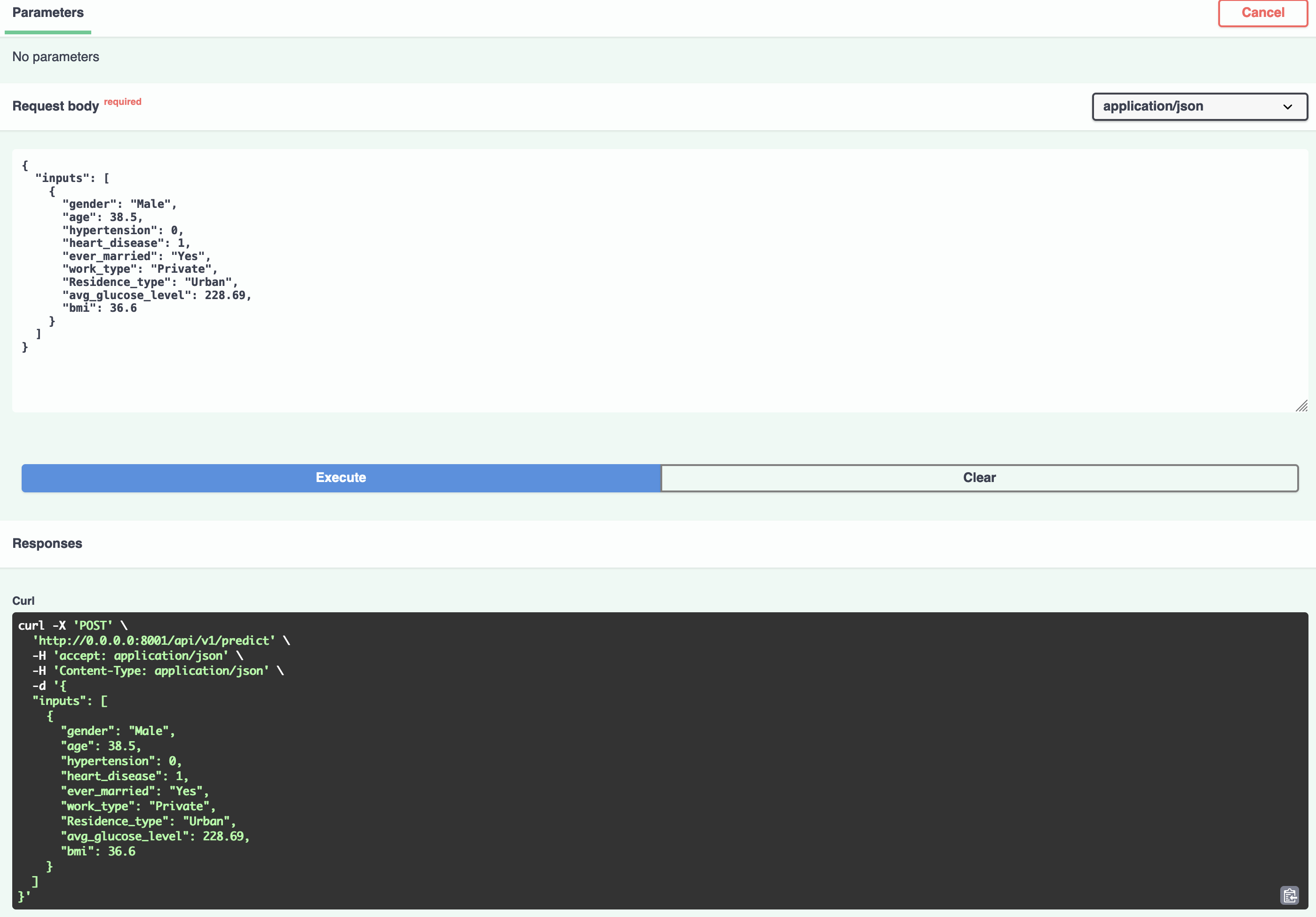
**Successfully built stroke\_detection\_model-0.0.1.tar.gz and stroke\_detection\_model-0.0.1-py3-no**

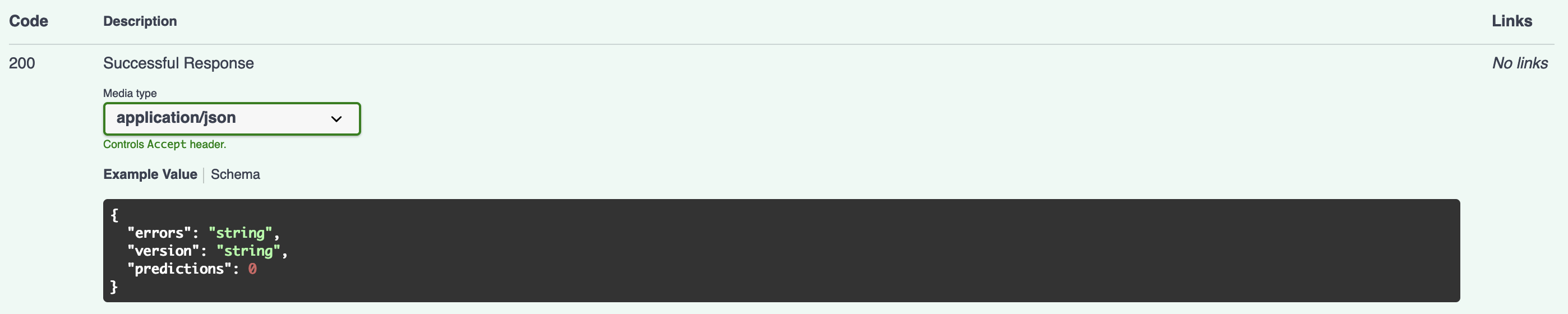
# FastApi Application



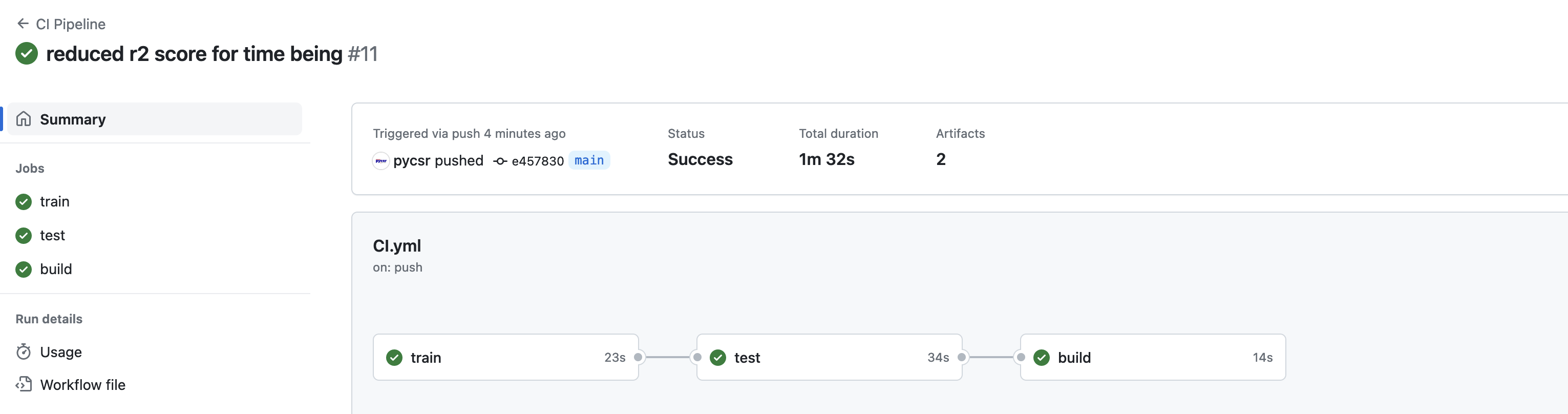
****

****

****

****

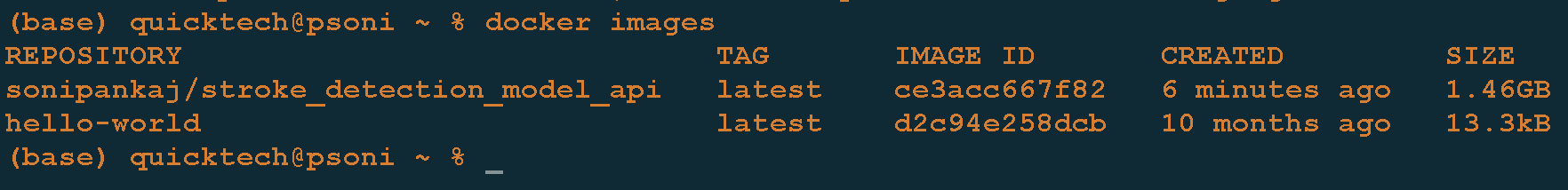
# CI workflow

****

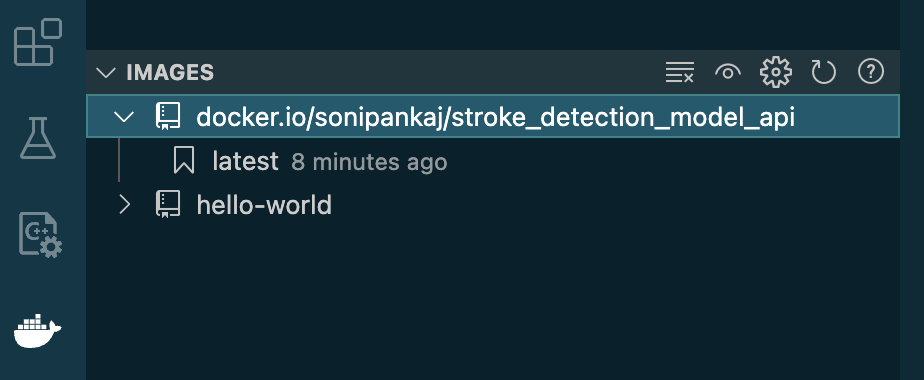
# Dockerization



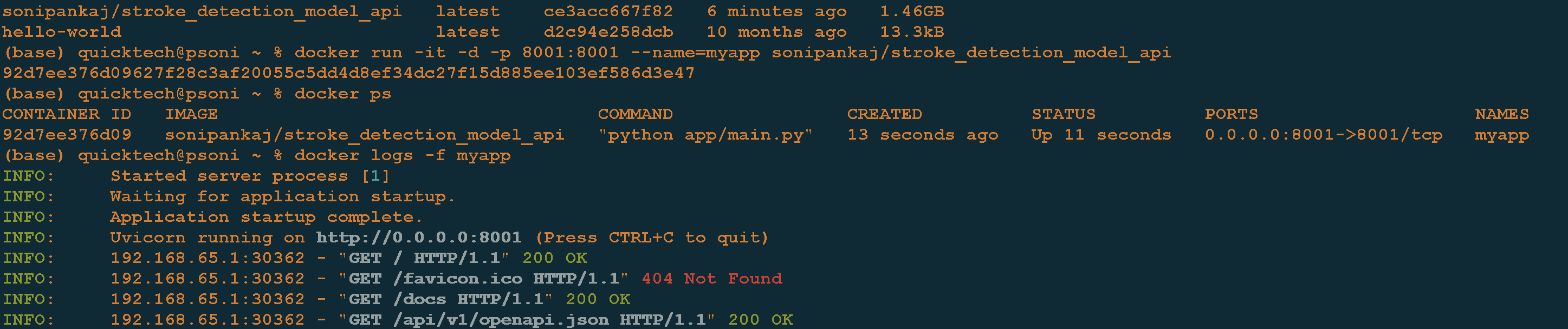
# Docker images

****

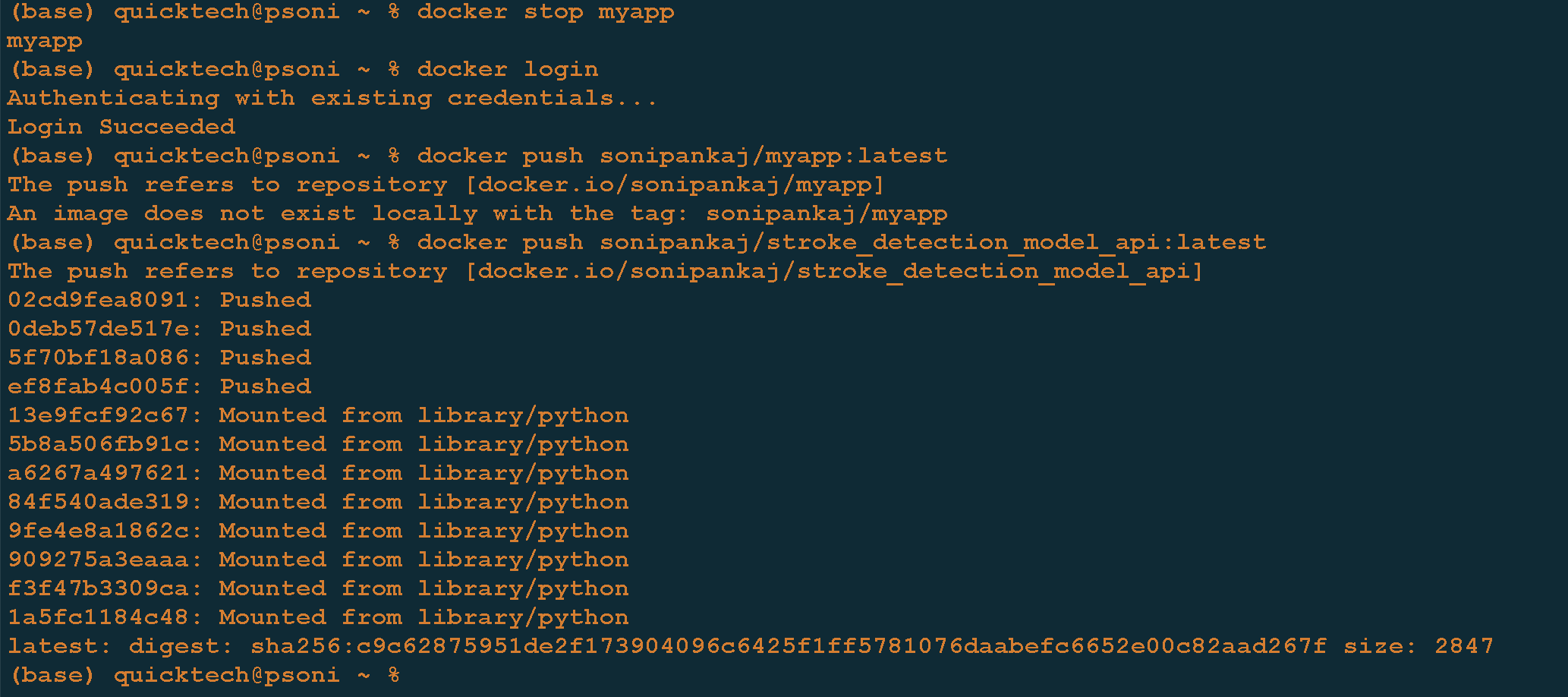
# Docker image In VS code



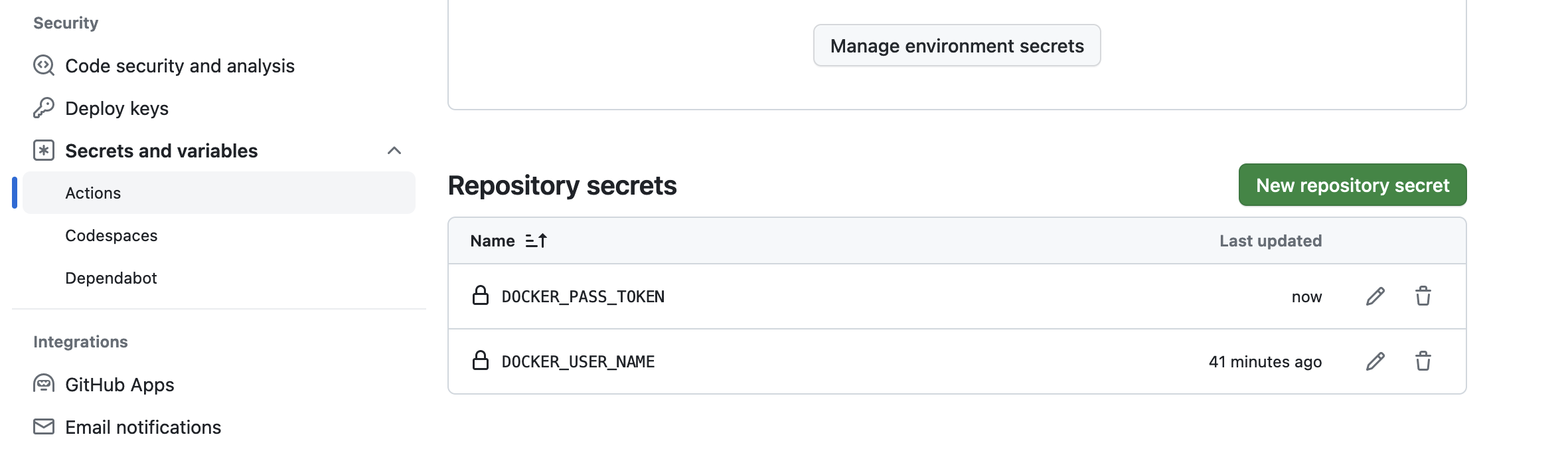
# Running Docker app from CLI



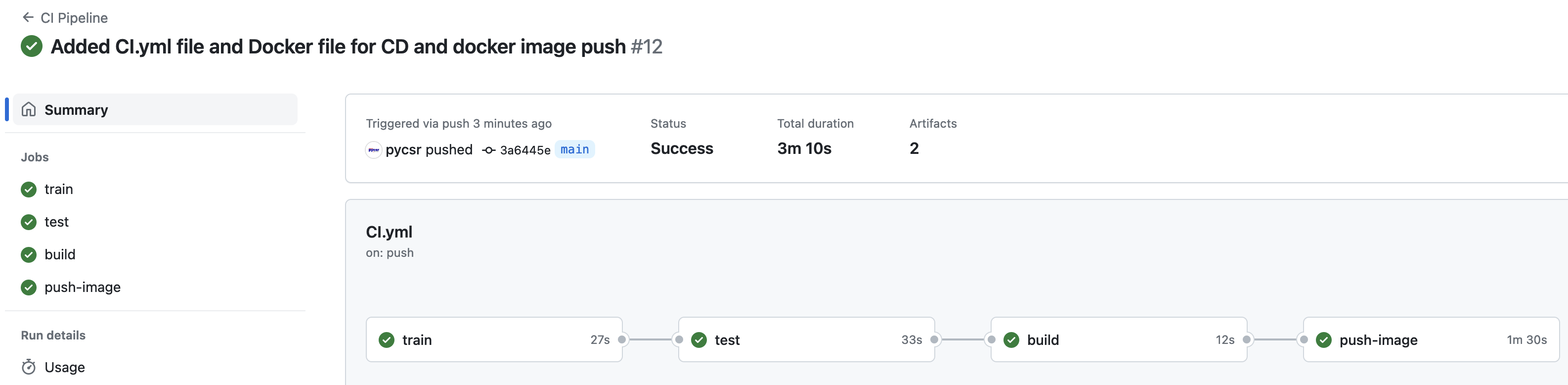
# Stop app and push to image to docker

****

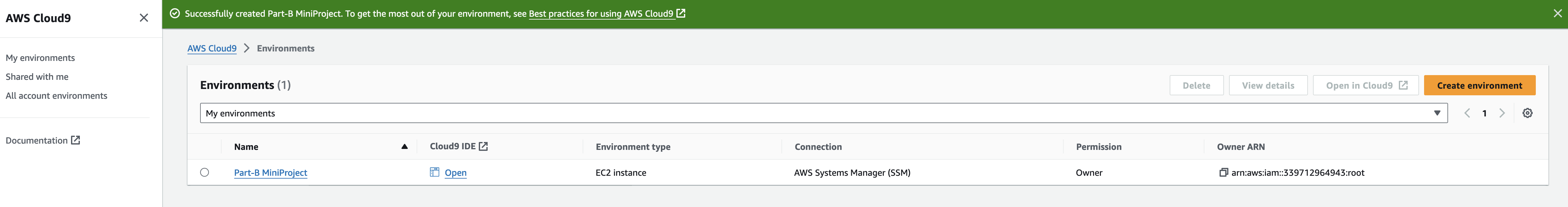
# Added Docker pass\_token and secret

****

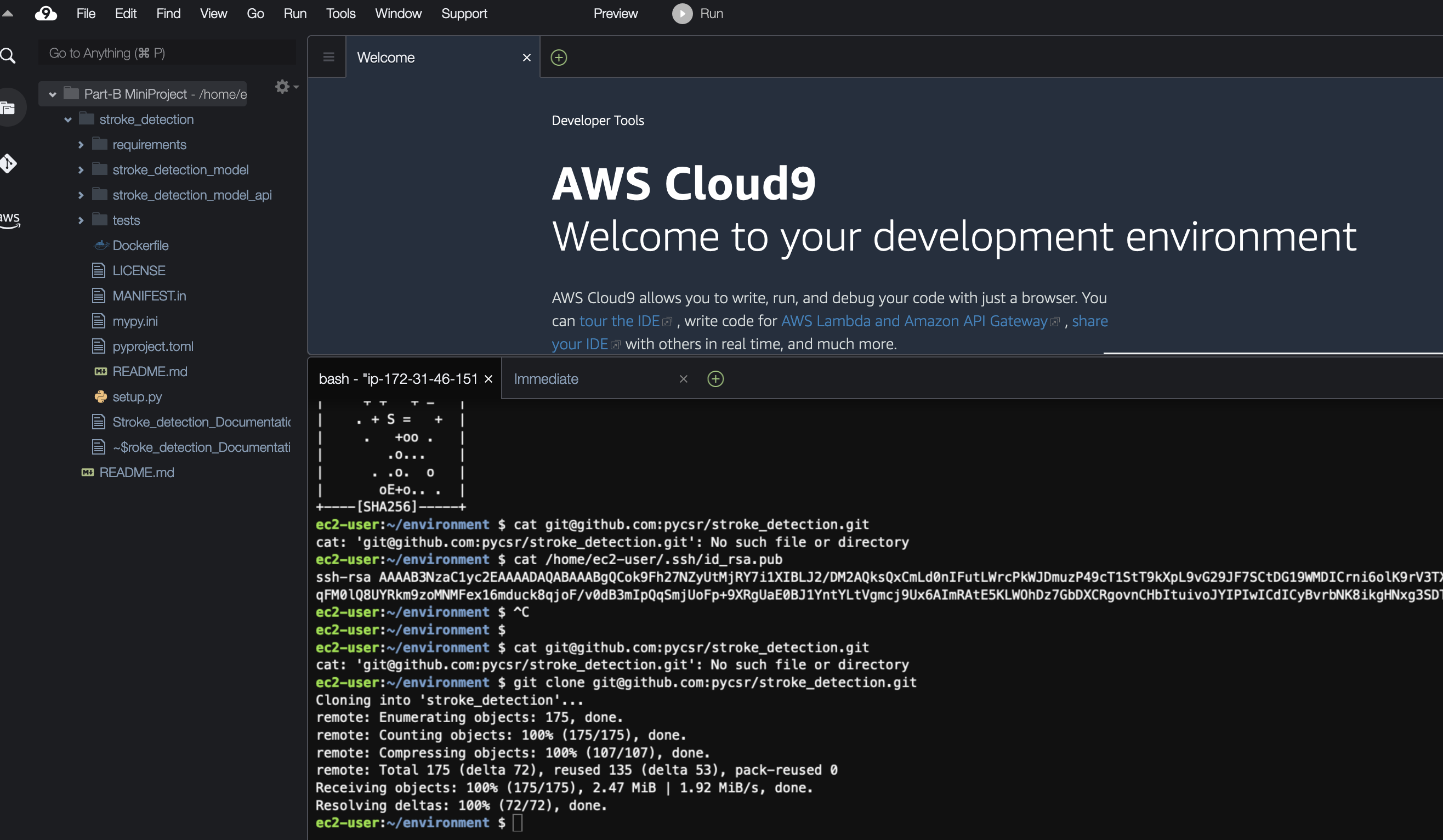
# Full CI pipeline is running successfully



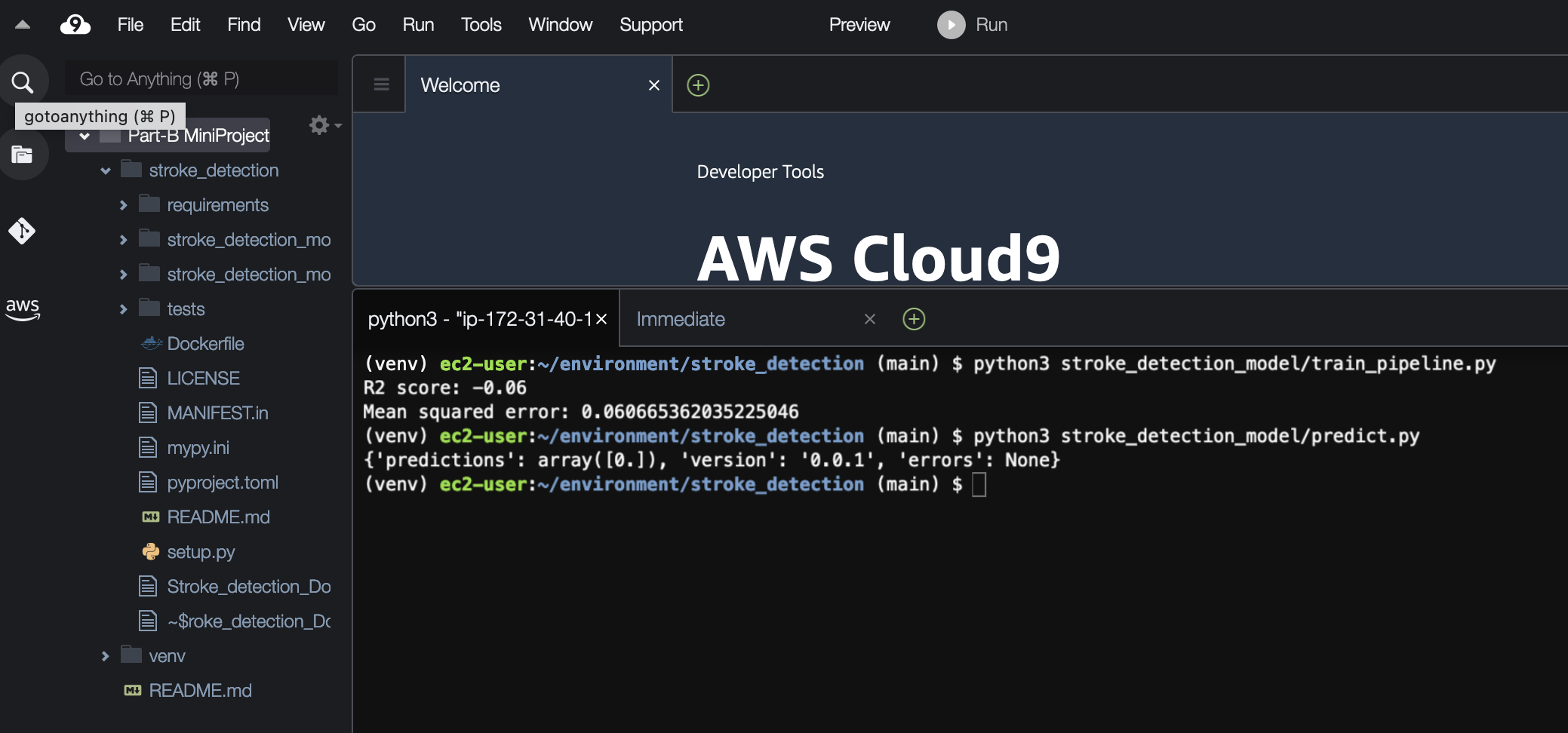
# Cloud-9 setup

****

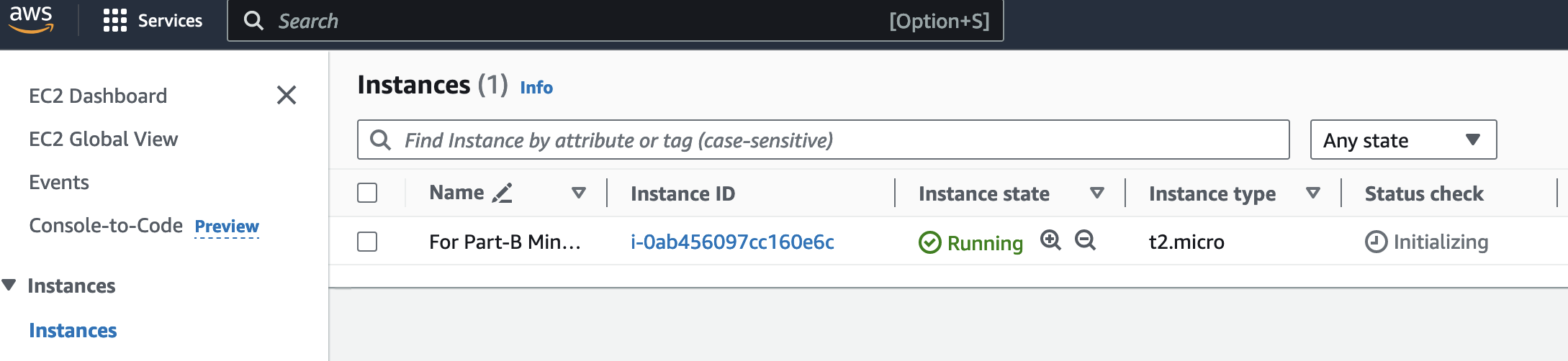
# Git clone in AWS environment

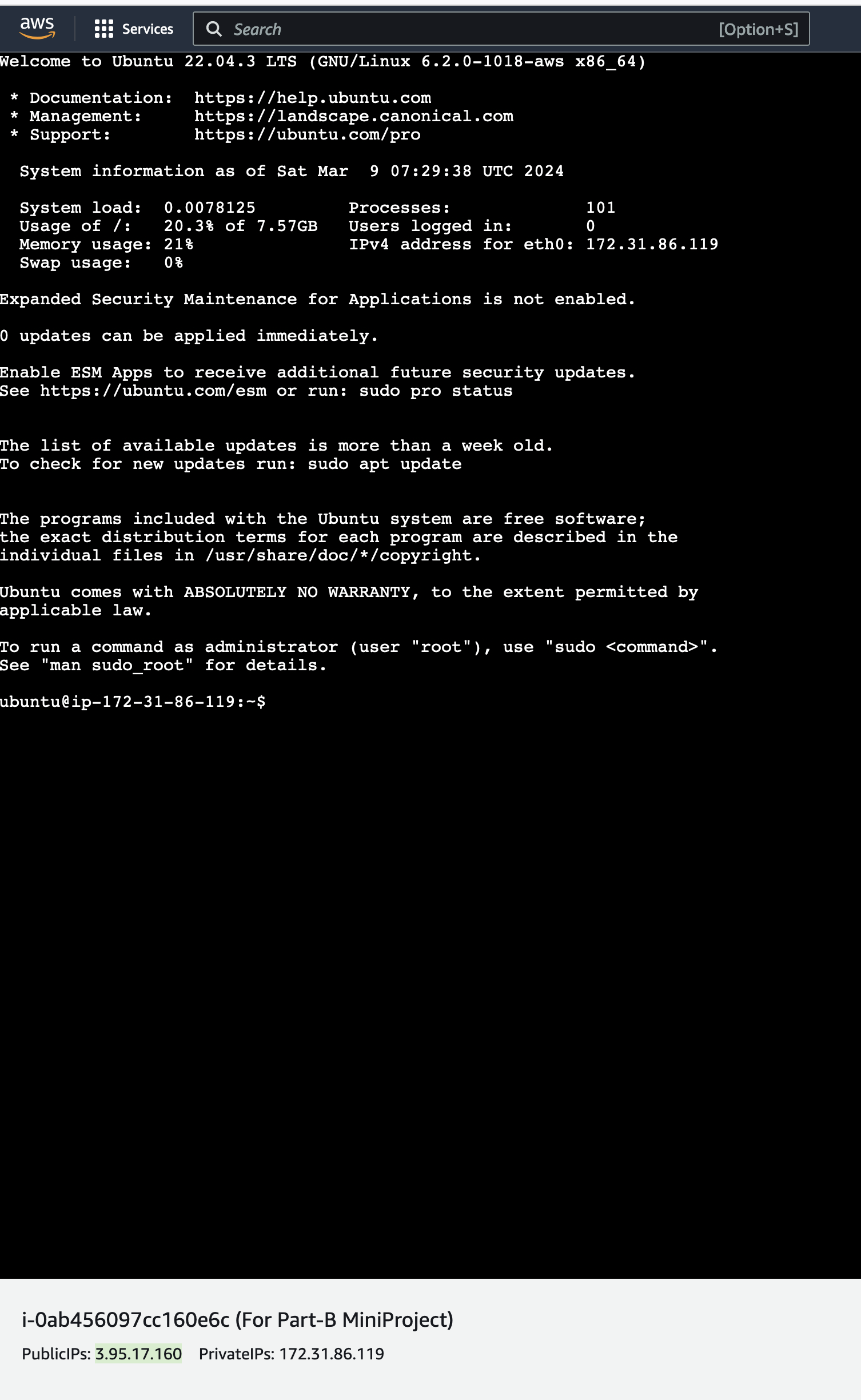
****

# Train and Predict running in AWS

**

# Creating & Launching EC2

**

**

# Docker Installation

**Install Docker on virtual machine / EC2 instance:**

1. Open the terminal.
2. Remove any Docker files that are running in the system, using the following command:

sudo apt-get remove docker docker-engine docker.io

NOTE that we are using the *sudo* keyword along with commands.  
Using *sudo*, users no longer had to change to the root user or log into that account to run administrative commands (such as installing software). Users could run those admin activities through sudo with the same effect as if they were run from the root user account.

1. Get the system up-to-date using the following commands.  
   It will update the package index files on the instance, which contain information about available packages and their versions.

Press enter, if a popup comes saying - Services will be restarted.

1. Install Docker using the following command:

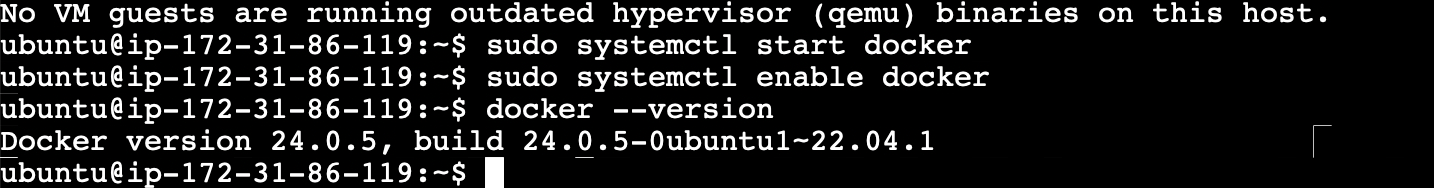
sudo apt install docker.io -y

1. Start docker service and enable the service to start at boot using the following commands:

sudo systemctl start docker

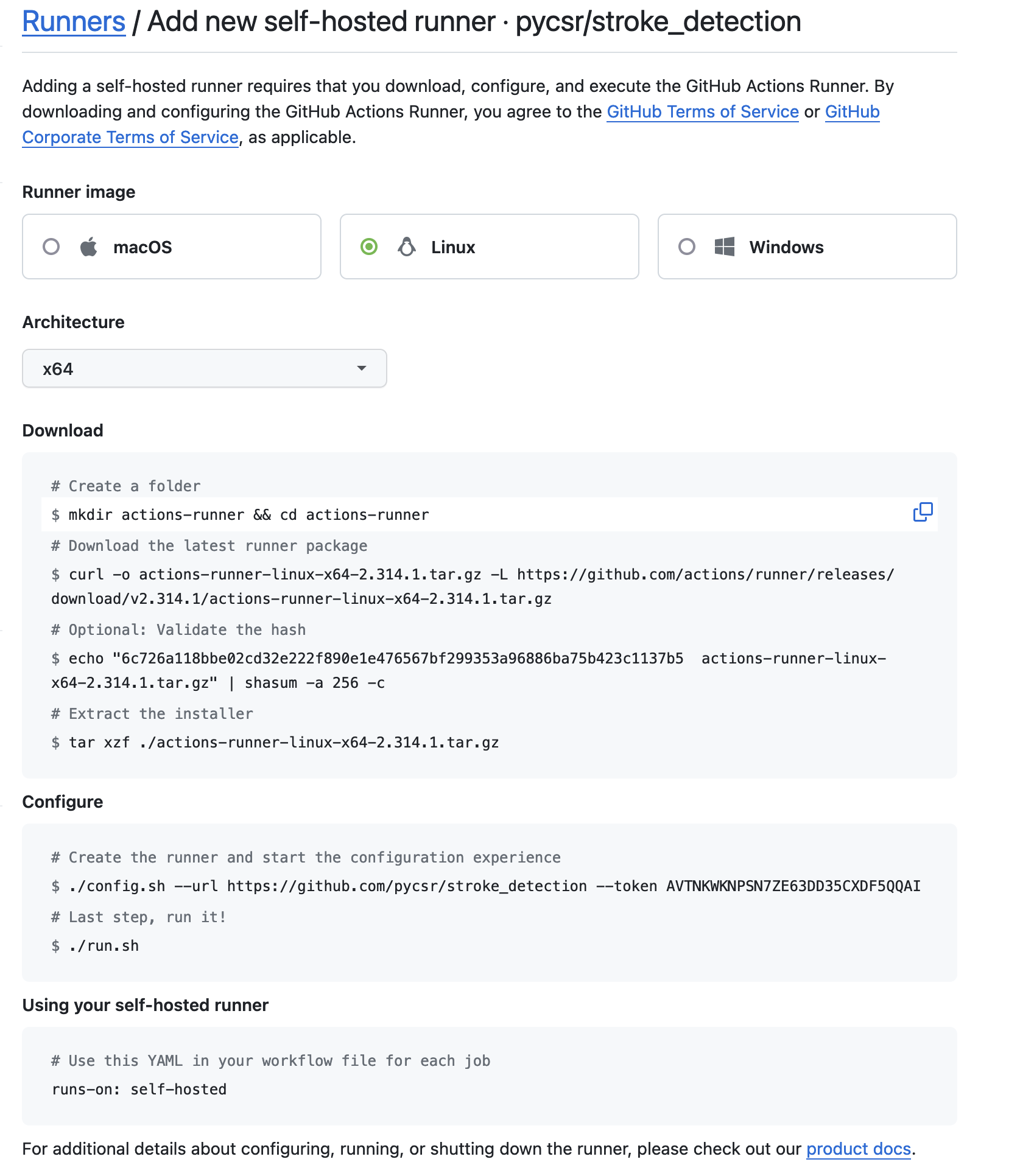
sudo systemctl enable docker

1. Before testing Docker, check the version installed using the following command:

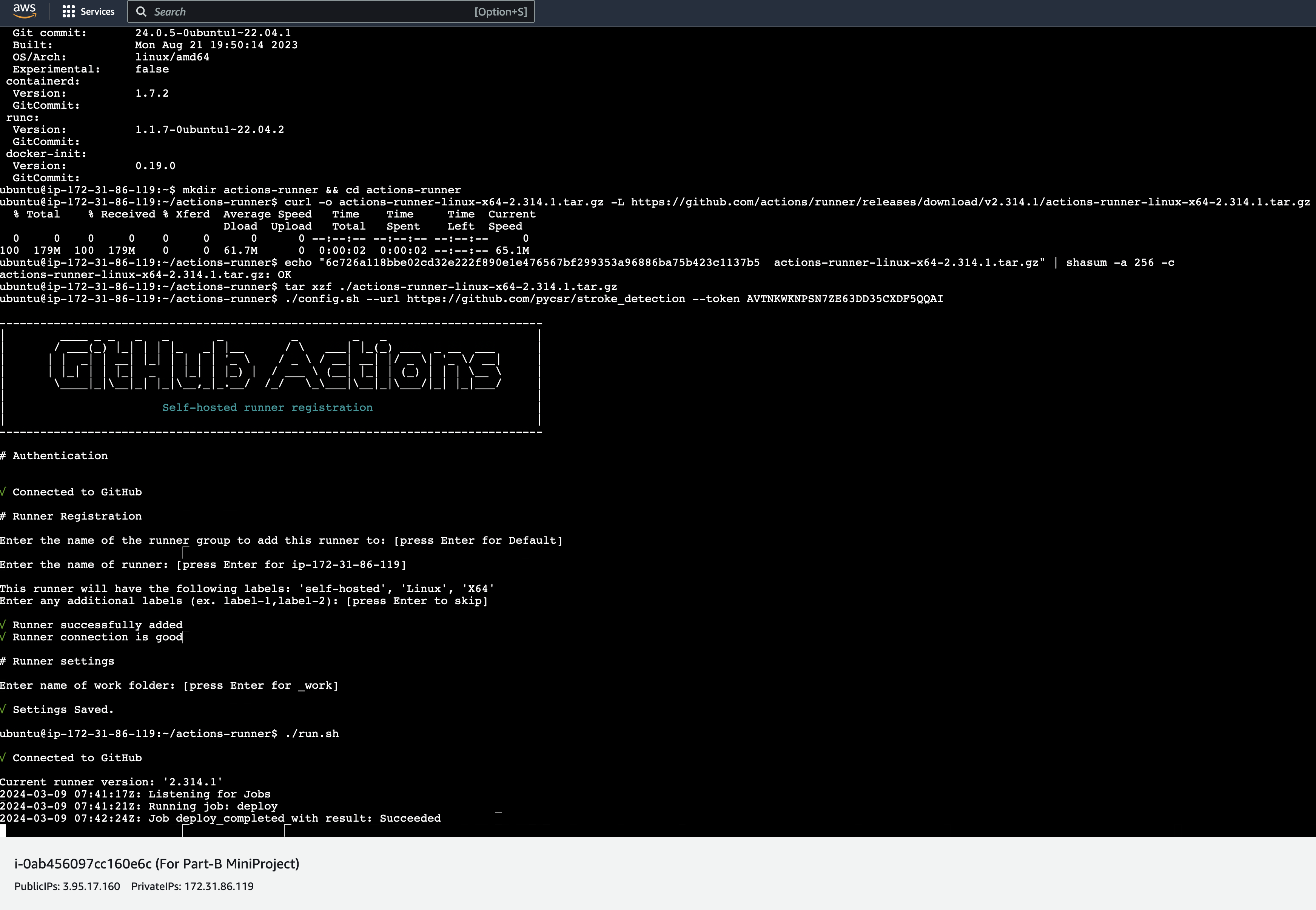




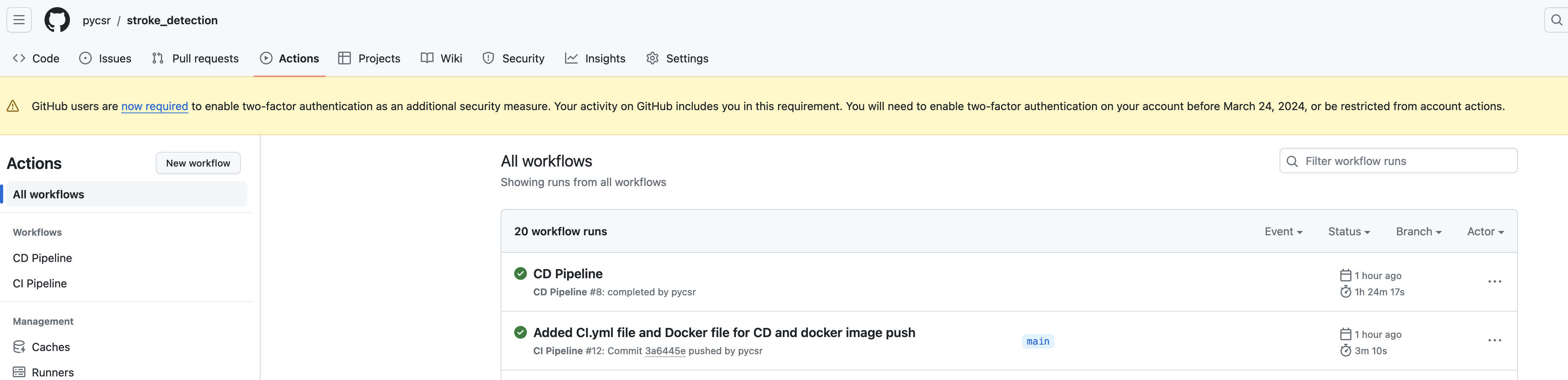
# Adding self-hosted runner in Git hub Repo:

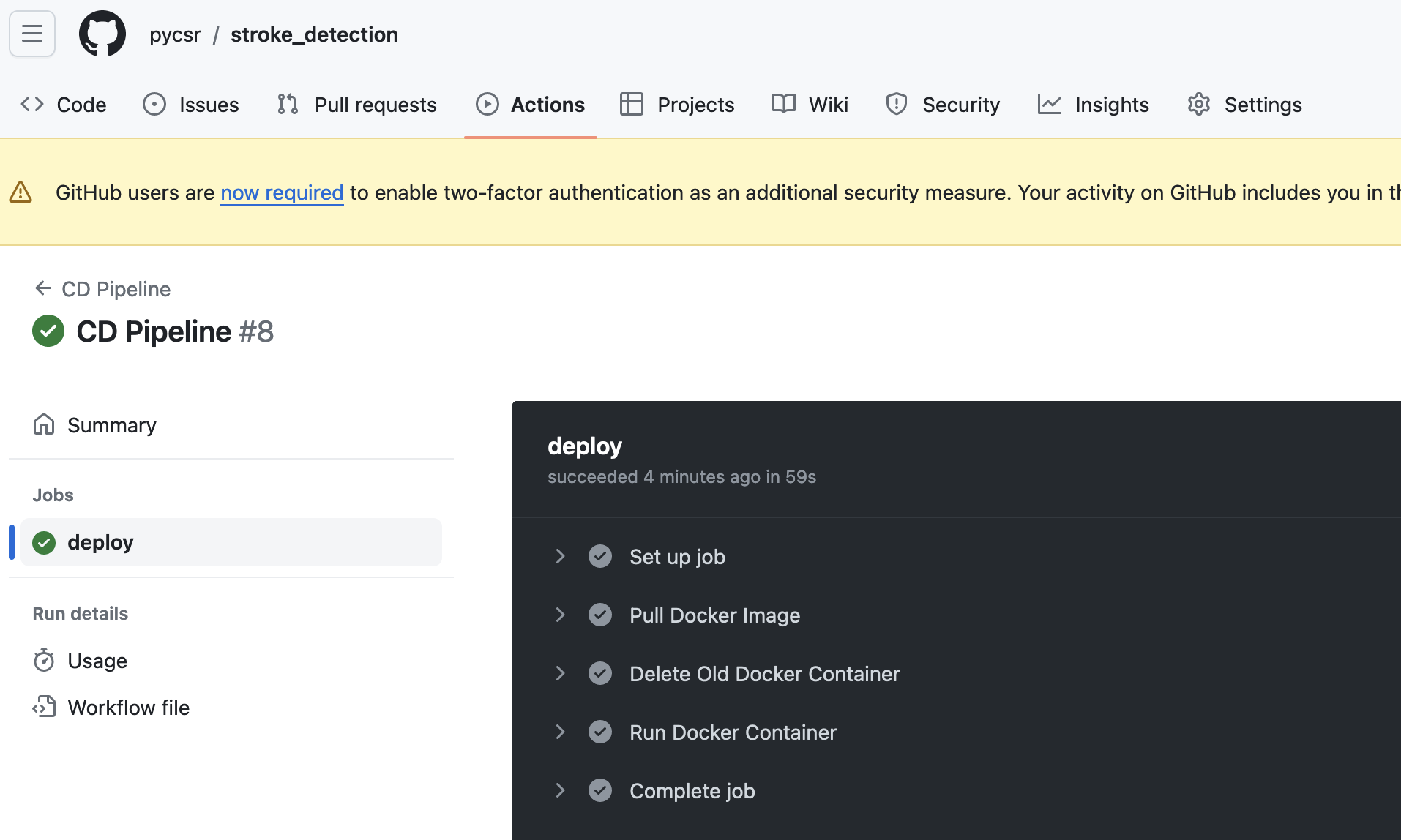
**

# Running GitHub Actions:

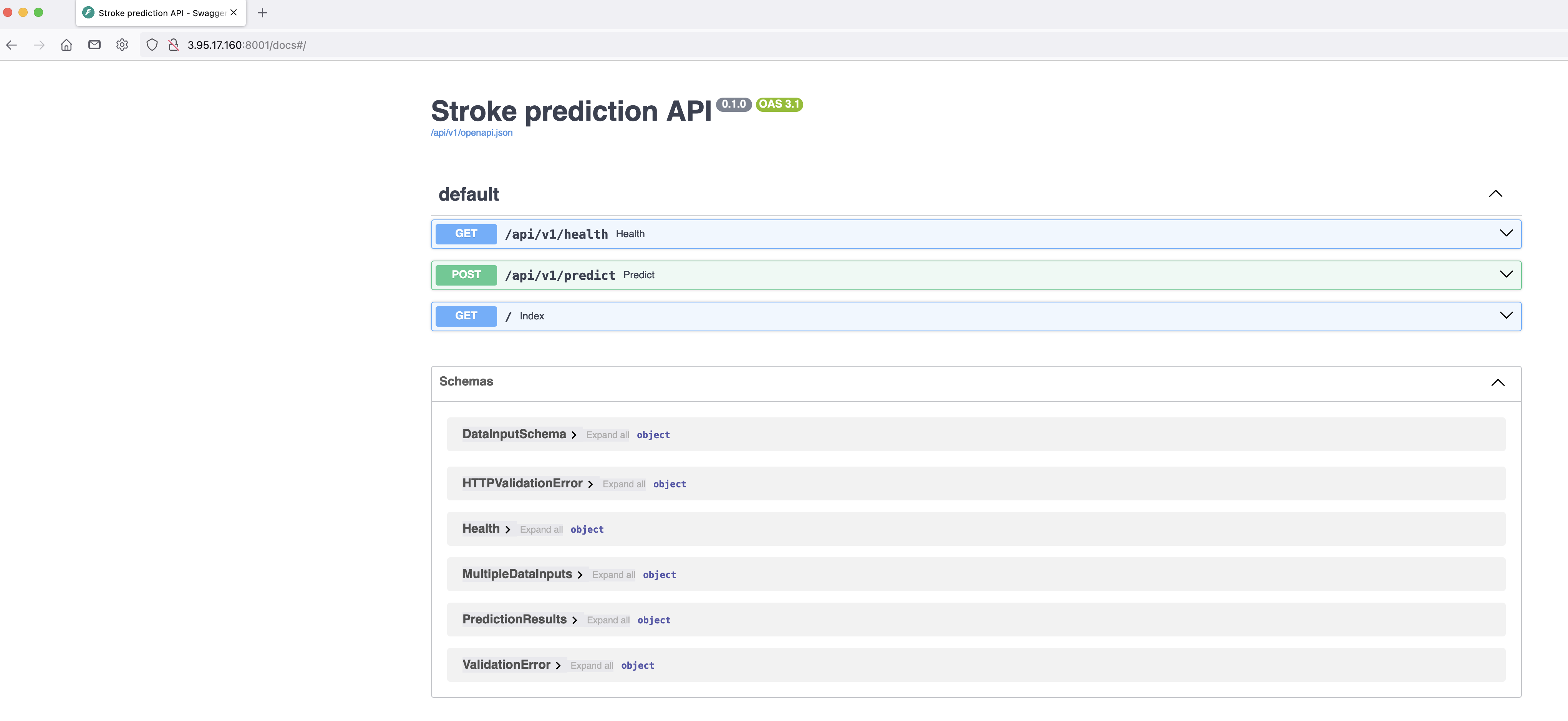
**

# CD workflow

**

**

# Running FastApi app on EC2 instance

**