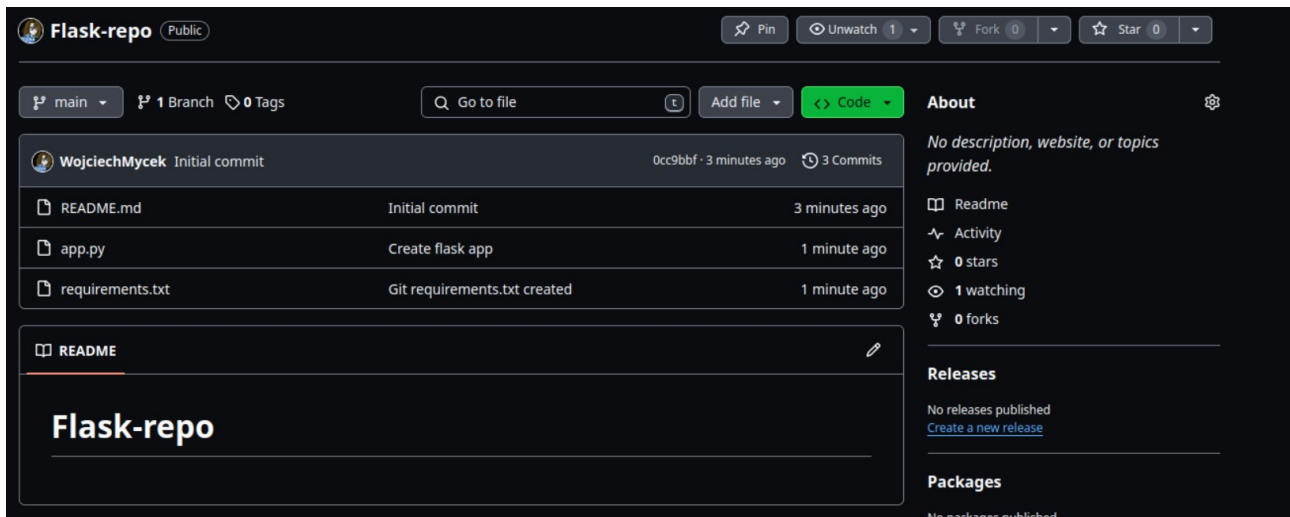


## 1. Utworzenie repo na githubie



## 2. Sklonowanie repo lokalnie I utworzenie wcześniej zdefiniowanych plików:

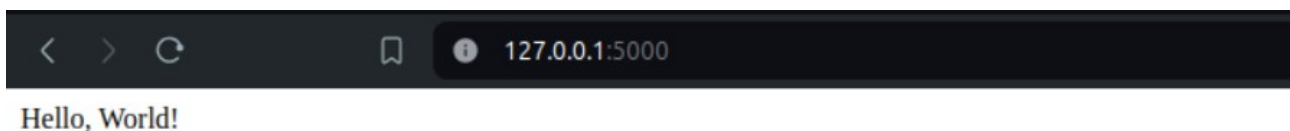
```
wojtek@wojtek-ThinkPad-T480:~/Desktop/GitHub/Flask/Flask-repo$ ls -lah
total 24K
drwxrwxr-x 3 wojtek wojtek 4.0K Jun  4 19:17 .
drwxrwxr-x 3 wojtek wojtek 4.0K Jun  4 19:15 ..
-rw-rw-r-- 1 wojtek wojtek 163 Jun  4 19:16 app.py
drwxrwxr-x 8 wojtek wojtek 4.0K Jun  4 19:19 .git
-rw-rw-r-- 1 wojtek wojtek 12 Jun  4 19:15 README.md
-rw-rw-r-- 1 wojtek wojtek 2.0K Jun  4 19:17 requirements.txt
wojtek@wojtek-ThinkPad-T480:~/Desktop/GitHub/Flask/Flask-repo$
```

Repo zostało sklonowane do katalogu lokalnego na maszynie. Aby to osiągnąć została dodana para kluczy SSH na GitHub, aby umożliwić bezproblemowo sklonowanie go.

## 3. Działanie aplikacji flask:

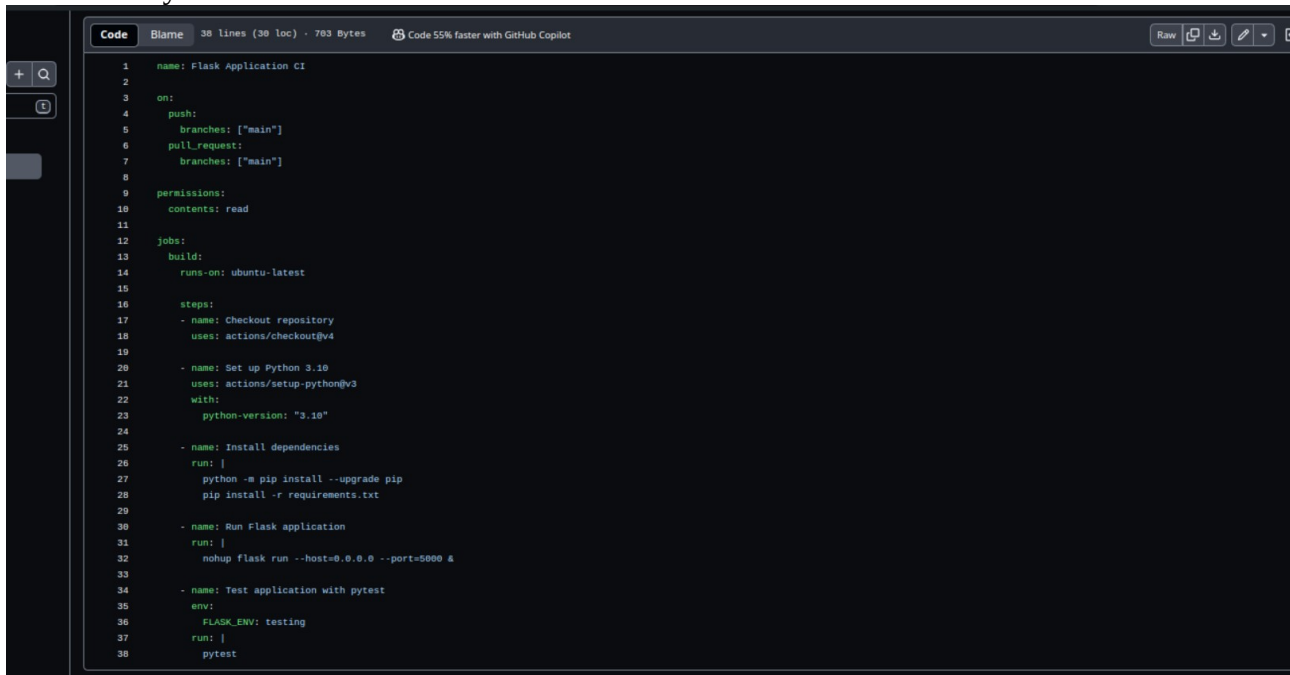
```
wojtek@wojtek-ThinkPad-T480:~/Desktop/GitHub/Flask/Flask-repo$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 158-879-459
```

Widok z przeglądarki:



Hello World jest widocznie na stronie

#### 4. Dodanie .yaml file



```
1 name: Flask Application CI
2
3 on:
4   push:
5     branches: ["main"]
6   pull_request:
7     branches: ["main"]
8
9 permissions:
10  contents: read
11
12 jobs:
13   build:
14     runs-on: ubuntu-latest
15
16     steps:
17       - name: Checkout repository
18         uses: actions/checkout@v4
19
20       - name: Set up Python 3.10
21         uses: actions/setup-python@v3
22         with:
23           python-version: "3.10"
24
25       - name: Install dependencies
26         run: |
27           python -m pip install --upgrade pip
28           pip install -r requirements.txt
29
30       - name: Run Flask application
31         run: |
32           nohup flask run --host=0.0.0.0 --port=5000 &
33
34       - name: Test application with pytest
35         env:
36           FLASK_ENV: testing
37         run: |
38           pytest
```

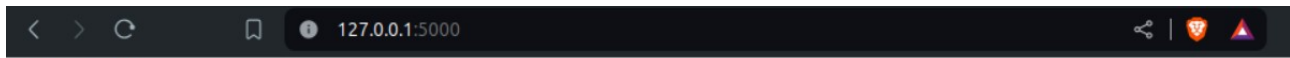
<https://github.com/WojciechMycek/Flask-repo/blob/main/.github/workflows/python-app.yml>

#### 5. Utworzenie DockerFile

```
1 FROM python:3.10-slim
2
3 ENV PYTHONUNBUFFERED=1
4
5 WORKDIR /app
6
7 COPY requirements.txt .
8
9 RUN pip install --no-cache-dir -r requirements.txt
10
11 COPY . .
12
13 EXPOSE 5000
14
15 CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]
16
17
```

Budowanie obrazu:

Efekt:



```
https://docs.docker.com/go/buildx/
Sending build context to Docker daemon 144.9kB
Step 1/8 : FROM python:3.10-slim
--> 61b0911887c0
Step 2/8 : ENV PYTHONUNBUFFERED=1
--> Using cache
--> 9422bb28baf0
Step 3/8 : WORKDIR /app
--> Using cache
--> d9e7eb9a702a
Step 4/8 : COPY requirements.txt .
--> Using cache
--> e6795336219c
Step 5/8 : RUN pip install --no-cache-dir -r requirements.txt
--> Using cache
--> 479ad2f10a14
Step 6/8 : COPY . .
--> Using cache
--> 1c9fbaa0dc55
Step 7/8 : EXPOSE 5000
--> Using cache
--> ef774bedfa11
Step 8/8 : CMD ["gunicorn", "--bind", "0.0.0.0:5000", "app:app"]
--> Using cache
--> 0638548f40db
Successfully built 0638548f40db
Successfully tagged flask-app:latest
```

