Check Visitor's IP App

Task:

Please create a simple web application that returns the IP address of the requesting client. The application should respect the Accept request HTTP header (supported formats: xml, yaml, html, txt). The application should also be able to return the list of client IP addresses that have queried the app in the past.

The application is to be written in Python and run as a Docker container. Please keep the app simple and minimalistic.

Further suggested extensions, if possible:

- If needed, handle errors in any simple way
- Add rate limiting per client IP address
- Optimize the size of the Docker image
- Make it easy to deploy the app in Kubernetes or Openshift
- Make Kubernetes/Openshift able to restart the container if it does not work properly

Document (in English) how to build it and run it. If anything in your development remains unfinished, please leave comments explaining what is to be improved.

Deliver the app to PCSS as a git repository.

Goals:

- Receive IP address of the requesting client

- Obtain log of visitors IP

- Python as a main programming language

- Run application in a Docker

- Request rate limit per client(in this case 3 attempts)

- Handle errors in any simple way

- Publish app as a git repository

Implementation steps:

- Installation of Docker Api, Python

- Create virtual environment to test application

- Install required dependencies (flask, requests)

- Create main.py that contains main source code of web application written in Python

- Create ip_list.txt containing history of visitors IP

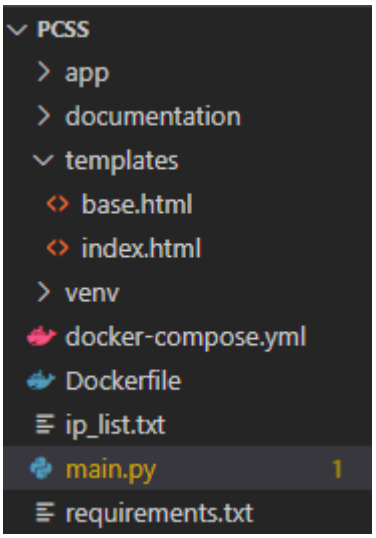
- For a Docker implementation:

- Create Dockerfile, a text document that contains all the commands a user could call on the command line to assemble an image

- Create requirements.txt that contains all mandatory packages

- Create docker-compose.yml that define the services and with a single command

Application structure:



How to run application:

There are two ways to run the project:

- in virtual environment:

Command to run application

python -u "project_path"

After successful compilation, console logs should look like:

```
 * Detected change in 'c:\\Users\\Lipczan\\Desktop\\pcss\\main.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 338-523-687
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://192.168.0.69:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [21/Feb/2022 15:37:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Feb/2022 15:37:17] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [21/Feb/2022 15:37:18] "GET /favicon.ico HTTP/1.1" 404 -
```

And Web App:

Check your IP

Your IP address: Too many request from the same source IP! Received: 2022-02-21 15:37:17

IP Log:

46.187.246.142 - Requested: 2022-02-20 17:49:48

46.187.246.142 - Requested: 2022-02-20 23:35:21

89.151.35.80 - Requested: 2022-02-20 23:36:01

89.151.35.80 - Requested: 2022-02-20 23:36:10

89.151.35.80 - Requested: 2022-02-20 23:36:11

82.118.29.92 - Requested: 2022-02-21 00:17:55

82.118.29.92 - Requested: 2022-02-21 00:18:17

127.0.0.1 - Requested: 2022-02-21 15:31:51

127.0.0.1 - Requested: 2022-02-21 15:36:24

127.0.0.1 - Requested: 2022-02-21 15:36:26

31.0.48.101 - Requested: 2022-02-21 15:36:53

31.0.48.101 - Requested: 2022-02-21 15:36:54

31.0.48.101 - Requested: 2022-02-21 15:36:57

In the attached picture, clearly see that app returns public IP of a visitor, also web prevents from ‘resfresh spam’ and does not save logs (maximum 3 logs) generated by the same client

- in Docker container

Command to run and build container that includes image with app:

docker compose up –build

After successful compilation, console logs should look like:

```
PS C:\Users\Lipczan\Desktop\pcss> docker compose up --build
[+] Building 2.9s (10/10) FINISHED
 => [internal] load build definition from Dockerfile                                                                0.0s
 => => transferring dockerfile: 32B                                                                                 0.0s
 => [internal] load .dockerignore                                                                                   0.0s
 => => transferring context: 2B                                                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.8                                                       2.2s
 => [1/5] FROM docker.io/library/python:3.8@sha256:eb6bb612babb3bcb3b846e27904807f0fd2322b8d3d832b84dbc244f8fb25068  0.0s
 => [internal] load build context                                                                                   0.2s
 => => transferring context: 168.23kB                                                                               0.2s
 => CACHED [2/5] WORKDIR /app                                                                                       0.0s
 => CACHED [3/5] COPY ./requirements.txt /app/requirements.txt                                                      0.0s
 => CACHED [4/5] RUN pip install -r requirements.txt                                                                0.0s
 => [5/5] COPY . /app                                                                                               0.2s
 => exporting to image                                                                                              0.3s
 => => exporting layers                                                                                             0.2s
 => => writing image sha256:098fa31f41bd640541277a1af98ffc76a9e46cd305f2048de2a480b809c76bf7                        0.0s
 => => naming to docker.io/library/pcss_web                                                                         0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 1/1
 - Container pcss-web-1  Recreated                                                                                  0.2s
Attaching to pcss-web-1
pcss-web-1  |  * Serving Flask app 'main' (lazy loading)
pcss-web-1  |  * Environment: production
pcss-web-1  |    WARNING: This is a development server. Do not use it in a production deployment.
pcss-web-1  |    Use a production WSGI server instead.
pcss-web-1  |  * Debug mode: on
pcss-web-1  |  * Running on all addresses.
pcss-web-1  |    WARNING: This is a development server. Do not use it in a production deployment.
pcss-web-1  |  * Running on http://172.19.0.2:5000/ (Press CTRL+C to quit)
pcss-web-1  |  * Restarting with stat
pcss-web-1  |  * Debugger is active!
pcss-web-1  |  * Debugger PIN: 353-988-425
```

Occurred problems:

- Application ran in Docker container return address of a Docker Gateway, instead public visitor IP (possible problem caused by Docker network configuration that I'm not able to resolve it)

Check your IP

Your IP address: 172.19.0.1 Received: 2022-02-21 14:55:23

IP Log:

46.187.246.142 - Requested: 2022-02-20 17:49:48

46.187.246.142 - Requested: 2022-02-20 23:35:21

89.151.35.80 - Requested: 2022-02-20 23:36:01

89.151.35.80 - Requested: 2022-02-20 23:36:10

89.151.35.80 - Requested: 2022-02-20 23:36:11

82.118.29.92 - Requested: 2022-02-21 00:17:55

82.118.29.92 - Requested: 2022-02-21 00:18:17

127.0.0.1 - Requested: 2022-02-21 15:31:51

127.0.0.1 - Requested: 2022-02-21 15:36:24

127.0.0.1 - Requested: 2022-02-21 15:36:26

31.0.48.101 - Requested: 2022-02-21 15:36:53

31.0.48.101 - Requested: 2022-02-21 15:36:54

31.0.48.101 - Requested: 2022-02-21 15:36:57

172.19.0.1 - Requested: 2022-02-21 14:55:18

172.19.0.1 - Requested: 2022-02-21 14:55:21

Undone tasks:

- Optimize the size of the Docker image
- Make it easy to deploy the app in Kubernetes or Openshift
- Make Kubernetes/Openshift able to restart the container if it does not work properly

Due to the lack of my knowledge and experience, the above goals were not achieved.