Airmeet SRE take-home assignment

Task

Implement and Containerize a REST API server that ingests metrics from its clients and generates on-demand stats reports.

Description

Imagine you're writing a metrics ingestion server as part of an Infrastructure monitoring tool. Implement an HTTP server that exposes the following REST API:

1. Ingestion

```
Method: POST
Path: /metrics
```

Headers:

```
content-type: application/json
```

JSON body structure:

```
{
    "percentage_cpu_used": <integer between 0-100>,
    "percentage_memory_used": <integer between 0-100>
}
```

Responses:

1. 200 If api has successfully ingested the data supplied
2. 500 if anything goes wrong

Sample curl request (assuming your api is deployed locally and bound to port 8080):

```
$ curl \
    -XPOST \
    -H "Content-Type: application/json" \
    --data '{"percentage_cpu_used": 55, "percentage_memory_used": 90}' \
    http://127.0.0.1:8080/metrics
```

The API accepts a json object containing info about the sender's cpu and memory percentage utilization. The api must store this information in-memory along with the IP address of the sender.
A sender can send their metrics to the API at different points of time. For eg- sender A might send metrics 25 times at 2-second intervals, B might send 500 times at 0.5 sec intervals.
For this task, you can ignore concurrency issues like race conditions while managing the data in-memory.

2. Report Generation

```
Method: GET
Path: /report
```

Headers:

```
content-type: application/json
```

Responses:

1. 500 if anything goes wrong
2. 200 and a json payload with the following structure:

```
[
    {
        "ip": <IP address of machine>,
        "max_cpu": <maximum cpu %>,
        "max_memory": <maximum memory %>
    },
    ...
]
```

The JSON payload is an array in which each object corresponds to a unique IP whose metrics were ingested by the API. You need to return the maximum cpu & memory utilizations that the client ever reached.

Sample curl request:

```
$ curl \
    -XGET \
    -H "Content-Type: application/json" \
    http://127.0.0.1:8080/report

[
    {
        "ip": “10.12.10.16”,
        "max_cpu": 87,
        "max_memory": 98
    },
    {
        "ip": “10.12.10.78”,
        "max_cpu": 44,
        "max_memory": 50
    }
]
```

Write a simple Dockerfile to containerise the app. The container must expose the API over port **8080**.

Notes

You’re free to use any language, libraries and framework of your choice. You don’t need to write any tests.

Please upload your code to a git repository and share it with us. Include a README that briefly describes the project and documents how to run the container and sample curl requests for the APIs.

What we’re looking for:

- Correctness of program
- How the metrics data is organized in memory
- Code comments wherever relevant
- Clean organization & readability of code

Best of Luck!