Let's Add Health Monitoring to the FastAPI Application for MNIST digit prediction and Dockerize it.

Building on the previous assignment, let's add prometheus monitoring hooks into the application to track the API usage and to monitor the App's health. Let's add Grafana visualization to work with Prometheus metrics. After making the single instance of the application working, let's dockerize it and try to reproduce seamless working. When the dockerized version works fine, let's spice it up by creating multiple instances (cluster) of the docker image and monitor the cluster's health.

Task 1: [25 pts]

1. Take a FastAPI module from the previous assignment.
2. Setup Prometheus & Grafana for monitoring the FastAPI Application.
3. Add Counters for tracking the API usage from different client IP addresses.
4. Add Gauges to monitor the running time of the API with respect to the length of the input text. For a 'L' sized input text (just length of the string), if the time taken is 'T' mS, the effective processing time is T/L measured in "micro-sec per character". The length of input, total time taken of the API and the T/L time (plus the client IP details) should be exported by respective Gauges.
5. Now, add visualization in Grafana to display the usage counters, API run time, API T/L time, API memory utilization, API CPU utilization (rate), API network I/O bytes (and rate).
6. You client application can be Swagger UI, Postman or curl, but test it from different machines.

Task 2: [25 pts]

1. Setup the Docker environment and start the Docker services.
2. Define the dockerization configuration to dockerize the fastapi application with monitoring. Ensure the FastAPI and Prometheus ports are mapped appropriately.
3. Build the docker image.
4. Run the docker container. Check if Task 1 works seamlessly with the docker run.
5. While starting the docker container, setup CPU utilization limit to 1 CPU (you can use command line options)
6. Now, spin 2 or more instances of docker container based on the #CPUs in your laptop. You should adjust the port mapping to avoid conflict across multiple containers.
7. You have a cluster of FastAPI servers now. Configure your prometheus/grafana rig to monitor the cluster for various metrics. Have fun!!

Important Pointers

1. You are expected to submit the python scripts with the necessary inline comments and respective YAML files as well as a report.
2. Additionally, snapshot of your grafana visualization should be submitted.
3. You have to maintain this project in github with proper documentation and folder structure.
4. Here is the allocation of points per task
   20% for a clean coding style
   40% for the correctness of the implementation
   20% for github project
   20% for input arguments' validation/boundary check

Best wishes.