Guide to Run the Dockerized GasLeakage Application

Overview:

This guide walks you through two methods for deploying the Dockerized GasLeakage Application:

1. Pull Images from DockerHub
2. Build Images Locally

Prerequisites Ensure Docker is installed on your system. If it isn't, download Docker from the official Docker Get Started page and follow the setup instructions.

Method 1: Pull Images from DockerHub

This guide walks you through the steps to install and run the GasLeakage application using Docker.

Step 1: Install Docker

Before you begin, make sure Docker is installed on your system. If Docker isn't already installed, download it from the official Docker Get Started page and follow the instructions provided.

Step 2: Clone the Repository from the Master Branch

To access the GasLeakage application code, clone the master branch of the repository. Open a terminal (or Command Prompt) and enter the following command:

```
git clone -b master https://github.com/sanajabbarweb/GasLeakage.git
```

This will download the repository's master branch into a folder called `GasLeakage`.

Step 3: Navigate to the Project Directory

Once cloned, navigate to the project directory by executing:

```
cd GasLeakage
```

Step 4: Download images from Dockerhub

```
docker pull sanajabbar88/gasdet-streamlit_service:latest
```

```
docker pull sanajabbar88/gasdet-flask_service:latest
```

This will download the docker images.

Step 5: Modify Camera Stream URL for Live Streaming

If you want to use a live camera feed instead of the default camera, follow these steps:

- Open the `flask_app.py` file.
- Locate line 231, where `rtsp_url` is defined.
- Replace `url` with the IP URL of your live camera feed in `rtsp_url` to enable live streaming.

Step 6: Build and Start the Application Using Docker Compose

To build and start the application, use Docker Compose with the command below:

```
sudo docker-compose up --build
```

The `--build` flag ensures that Docker Compose rebuilds the application images if there are any updates to the Dockerfile or configuration files.

Application Execution

Docker Compose will automatically start the application, including mapping the necessary ports. This setup removes the need for any additional `docker run` commands.

Step 7: Access the Application

http://localhost:8501/

Step 8:Down Streamlit App

Close App using cltr+c then stop docker using command below.

```
docker-compose down
```

Method 2: Build Images Locally

This guide walks you through the steps to install and run the GasLeakage application using Docker.

Step 1: Install Docker

Before you begin, make sure Docker is installed on your system. If Docker isn't already installed, download it from the official Docker Get Started page and follow the instructions provided.

Step 2: Clone the Repository from the Master Branch

To access the GasLeakage application code, clone the branch of the repository to build docker images locally. Open a terminal (or Command Prompt) and enter the following command:

```
git clone -b local_dockeriamge
https://github.com/sanajabbarweb/GasLeakage.git
```

This will download the repository's master branch into a folder called `GasLeakage`.

Step 3: Navigate to the Project Directory

Once cloned, navigate to the project directory by executing:

```
cd GasLeakage-
```

Step 4: Modify Camera Stream URL for Live Streaming

If you want to use a live camera feed instead of the default camera, follow these steps:

- Open the `flask_app.py` file.
- Locate line 199 where `camera_url` is defined, `set your camera IP here`

```
camera_url = os.getenv("CAMERA_IP_URL") # set your camera IP here
```

- Open the `docker-compose.yaml` file.
- Locate lines 13 and 33 where `CAMERA_IP_UR` is defined.

```
CAMERA_IP_URL=http://<ip_address>:<port>/path  # Set the IP
camera URL here

CAMERA_IP_URL=http://<ip_address>:<port>/path  # Set the IP
camera URL here
```

Step 5: Build and Start the Application Using Docker Compose

To build and start the application, use Docker Compose with the command below:

```
sudo docker-compose up --build
```

The `--build` flag ensures that Docker Compose rebuilds the application images if there are any updates to the Dockerfile or configuration files.

Application Execution

Docker Compose will automatically start the application, including mapping the necessary ports. This setup removes the need for any additional `docker run` commands.

Step 6: Access the Application

http://localhost:8501/

Step 7:Down Streamlit App

Close App using cltr+c then stop docker using command below.

```
docker-compose down
```