The following is an assignment to evaluate your ability to develop software. You may use any resources on the internet to guide your solution, however you should have a strong understanding of your implementation.

The code you write for this assignment should be well documented with a readme file explaining how the code works, how to build it, and how to run/test it. Be sure to use programming best practices in your code such as thorough commenting and proper code formatting.

The prompt is intentionally lacking in implementation details. It's up to you to make assumptions and choose implementations of your choice when necessary.

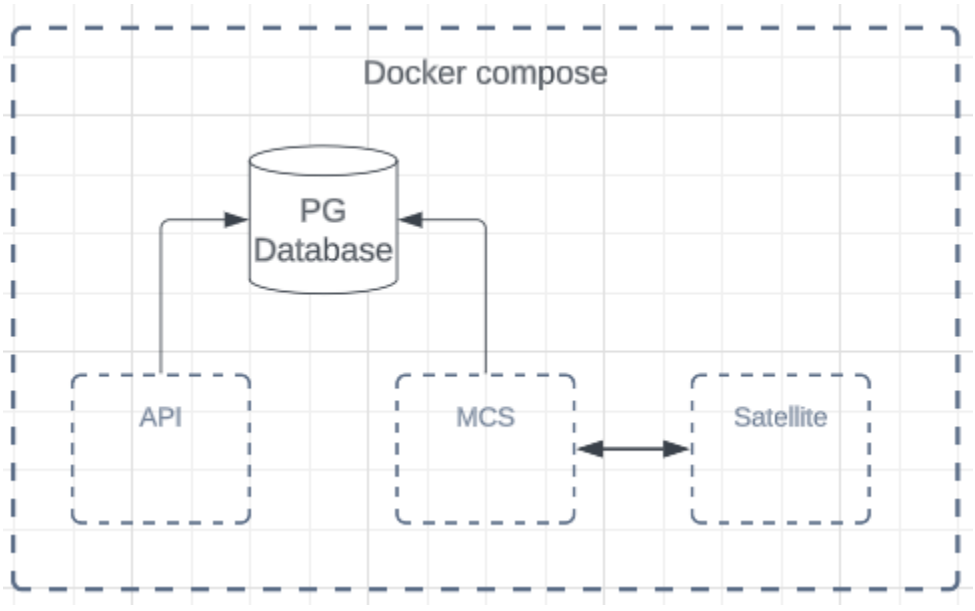
When finished, please send a GitHub link or a zip of the project via email.

Prompt

Build a system that facilitates communication with a satellite and data delivery to the end-user. The code for this system can be written in any language(s). The system will consist of a client facing API to queue imaging requests and view data delivered by a satellite, a mission control software (MCS) to send and receive data to/from the satellite, a satellite, and a database.

The user should be able to interact with this system through the client API.

This project must be completed using Docker, Docker Compose, and PostgreSQL. A base project has been provided for you to get you started. The project comes with a Postgres database and a simple starter MCS container. The readme instructions tell you how to use PG Admin along with some other tips.



Client API

Create this container to provide a REST API that acts as the driver for this system. This API should have the following endpoints exposed:

1. Add a tracked client satellite to the database.
2. Request an image of the tracked satellite in the database. These requests should be inserted into the database.
3. Get images of the tracked satellite that were taken by DROID.

MCS

A starting point container has been provided for you to mock an automated MCS. The MCS should intermittently communicate with the satellite to send imaging requests and download images (this mocks periodic communication when a satellite is over a ground station). The MCS should:

1. Implement a mechanism to intermittently communicate with the satellite. This is essentially mocking when a satellite is over a ground station. This can be very simple.
2. When the MCS is able to communicate with the satellite, send imaging requests to the satellite that the MCS finds in the database.
3. Periodically send a "download all images" command to the satellite to get all images stored on the satellite and insert into the database.

Satellite

Create a container that acts as the satellite for the system that the MCS will communicate to. The satellite can be built anyway you want and can be represented as a REST API if you want. The satellite should be able to:

1. Receive imaging requests from the MCS. When a request is received, the endpoint should "take an image" of the client satellite and store it locally. These can be random images from anywhere.
2. Receive image download requests from the MCS. This will download locally stored images from the satellite to the MCS.

Database

A Postgres database has been provided within the docker container. It is an empty database but should be enough to get you going. Instructions in the readme will help you get connected to it.

Notes

1. Communication between containers can be done via service name instead of an IP address. There is a lot of documentation about this process online.
    a. https://docs.docker.com/compose/networking/#link-containers
2. The client API should be reachable by Postman. This is how we will do our testing (a UI is not required).

Guidelines/Evaluation Metrics

Documentation	• Describe your approach to solving this problem • Comments explaining implementation • Detailed README including documentation for the API endpoints • Detail future improvements/expansions of the solution • Detail steps to build and run your program
Code Style	• How readable is your code? • How easy would it be to integrate this code into a wider code base? • How easy would it be for another developer to continue working on this project? • Usage of object-oriented programming (OOP) where it makes sense • Is your code testable?
Performance	• How well does the solution solve the problem? • Does the API build and run without errors? • Are all endpoints implemented? • Can the API be easily tested with tools like Postman?