# Part 1(a)

**What pros and cons do you see with respect to grabbing readings from an MQTT broker, vs getting them via a REST API?**

Pros:

1. The MQTT broker is optimized for unreliable networks, which makes it more ideal for remote monitoring.
2. Changing from a REST API to an MQTT broker can reduce the data upload time to seconds.
3. The MQTT approach will have reduced complexity, since it will not require a dispatcher on a schedule. It will only have acquisition functions subscribed to certain devices.
4. MQTT broker eliminates the need to connect every time data is fetched, which reduces overhead.

Cons:

1. MQTT operates over TCP, which adds to the power and memory requirements of the devices.

**How would you run such an acquisition function, which subscribes to an MQTT broker? For example would you still trigger it periodically via a dispatcher, or would you have it running as some sort of continuous "listener" function?**

Since an acquisition function subscribing to an MQTT broker receives data as it is published by a certain device, it does not need to run on a schedule. It should run continuously as a listener function, so that any data updates are immediately pushed to the data pipeline.

**What underlying AWS service would you run it on? E.g. EC2 vs Lambda, etc.**

Since a continuous “listener” function will be constantly running, a lambda function will not be a good option, since it has a timeout. On an EC2 server, an acquisition function can run for a long time and continuously pick up live data points published by devices to the broker. Furthermore, lambda (serverless) functions can easily become disorganised as more and more functions are added, which means that an EC2 server would be a better decision in the long run.

# Part 2

a) Considering that you could implement a DevOps cycle from scratch, what are key factors that you would consider before implementing?

- The team: number of people, skills available, roles in the project.
- The timeline: Deadlines, milestones and sprint cycles.

b) What tools would you use and how would they integrate with each other? A specific toolchain as well as a more conceptual view are both okay.

- A program which features a task board, to delegate tasks.
- A reliable channel for communication between team members.

c) How would you define and measure success?

I would define success as reaching the end of a project with team members who are happy with what they have produced and clients who are happy with their experience with the team. I feel that communication is the most important aspect of getting a team to work smoothly together. The factors I would use to measure success in a project would be:

- Satisfaction of team members with what they have produced.
- Satisfaction of customer with team after the project has completed.