

Objectives & Instructions:

Continuous Assessment:

Implement a simple IoT pipeline with AWS Cloud platform and visualise the data. We will simulate two small IoT setups that record and push data from two jet engines. Students need to write a guideline on how they set up the pipeline (which the pictures with their Matric No. inside)

Requirements:

AWS Free Tier account (visa/master is required for registration).

A laptop with Anaconda 3 installed (Python \geq 3.5).

Importance:

Put your Matric No. in all the naming convention in AWS setup: thing names, dataframe column and DynamoDB table name, so that scoring can be done.

Students will implement a simple pipeline of pushing IoT data from devices to AWS and visualise it. The instructions for a "just-enough" submission are given as followed:

1. (Optional, highly recommended) Follow the AWS instruction to understand how to populate, send data from a computer to AWS through MQTT protocol.

(<https://docs.aws.amazon.com/iot/latest/developerguide/iot-plant-module1.html>)

2. [4 scores] (Publish pre-defined engine data to AWS.

- a. Download data for engines FD001 and FD002 from Github repo https://github.com/iceberg12/NUS_guest_lecture/tree/master/input. Understand more about the jet engine data (operating settings and sensors) from the document there.
- b. Modify the Jupyter notebook in Step 1 to read and publish data from trainFD001.txt to your thing under AWS IoT platform at the rate of 10 seconds per row. Overwrite column 'id' of the engine as 'FD001' + id (e.g. FD001_12); add one more column 'timestamp' as timestamp in UTC (e.g. UTC 2019-01-28 14:41:15.237); also add one more column that contains your Matric number.
- c. Setup an Act under AWS IoT platform to define a rule that splits the message into columns of a DynamoDB table (see <https://docs.aws.amazon.com/iot/latest/developerguide/iot-ddb-rule.html>). The primary key of this table should consist of 'id' as the partition key and 'timestamp' as the sort key.
- d. Check under AWS DynamoDB service if your table is populated correctly.
- e. Stop the Jupyter notebook.

3. [4 scores] We are going to simulate two IoT things now.
 - a. Add one more thing under AWS IoT platform and one more certificate. Create a copy of the Jupyter notebook above, renaming the client name, certificate, data source train_FD002.txt. Modify the rule under AWS IoT platform to be triggered by '\$aws/things/+/shadow/update/accepted' (+ means anything, both the thing in Step 2 and in this step). Now, this rule should help to push data from both things.
 - b. Let the two Jupyter notebooks run at the same time, simulating the two "things" to run in parallel to publish data with a sampling rate of one record per 10 seconds in each thing.
4. [2 scores] Visualise the data for the two engines for all the sensors by querying the data from AWS DynamoDB. You are free to use any tools but need to explain clearly how you do it.
5. [Optional, 3 scores] Use other data sources.
6. [Optional, 7 scores] Ingest real-time data from embedded systems.
7. Only after complete this module: cleaning all resources created to avoid a recurring charge from AWS :)

References:

<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html><https://docs.aws.amazon.com/iot/latest/developerguide/topics.html>

Evaluation:

You are encouraged to try other ideas outside of this engine data.

You are encouraged to form a group of maximum 2 students and submit the same report (use two Matric No. in the IoT setup). For these groups, the point for each student is $\text{Group score} / 2 * 1.5$.

Example:

Group A: 1 student, scores 8 by following specifications => Get 8 points.

Group B: 2 students, scores 20 by doing projects out of the box => $\text{Get } 20 / 2 * 1.5 = 15$ points.

Group C: 2 students, scores 10 by following specifications => $\text{Get } 10 / 2 * 1.5 = 7.5$.

Report:

Deadline: 21 Sept 2019, 11:59:59 AM.

1. The report should describe step by step of your process, including screenshots with entities named with your Matric number.
2. Post a copy of your report to Medium post. Upload your code to Github.
3. Reports are to be sent to elenht@nus.edu.sg. Inside the report, indicate links to your Medium post and Github folder.