Project 01 [25 points] Deadline December x th

Introduction

Nowadays, IoT applications and solutions have emerged and are gaining importance in the modern world.

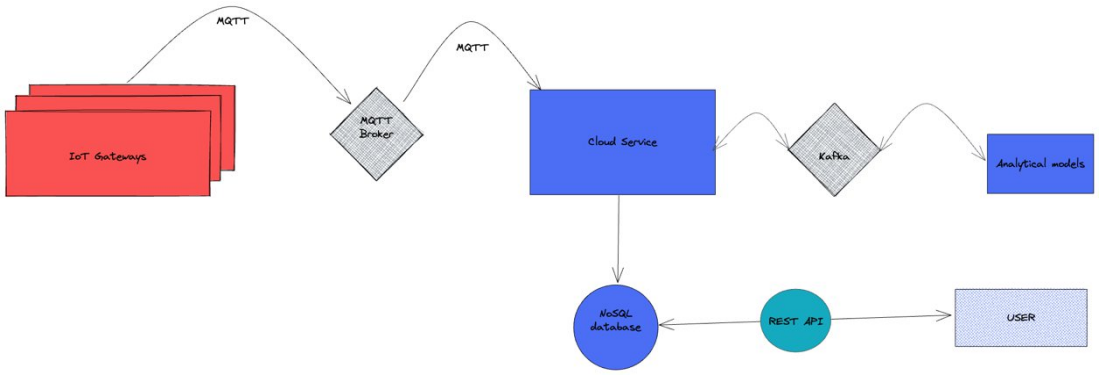
IoT devices, spread all over the world, are capable of collect a large amount of data in many different fields: home, mobility, energy consumption, real-time monitoring, etc.

With all this data, we can create advanced analytics, using various types of modeling: statistical models, neural networks, etc… and provide the corresponding summaries and verified information to provide the necessary knowledge to improve the way things are used.

If some students have a custom project proposal different than the one presented here, they can explain it and the teacher will verify if it can be used for the subject. (A different proposal will be always graded better due to the complexity of working in a real environment)

Project Proposal

This project will consist of the creation of an IoT application to serve the different users who join a home network.



- Each user will have a gateway (simulated in different docker containers) in his home that will send every 10 seconds the measurement of the indoor temperature.

- The data will be sent using one of the most important protocols used in IoT (MQTT).

- A cloud application will receive the data from MQTT and store it in a database to be able to check it latter.

- Additionally, the data received will be sent to an analytics module (The professor will provide the module) containing a simple predictive model and boundaries check. The data will be sent in streaming using Kafka.

- The response of the module (The predicted boundaries and if the data is in them or outside) will also be sent via Kafka back to the cloud and must be stored in the database.

1st level Functions [50%]

- The Gateway can be simulated with dockers. It will contain the CSV provided by the teacher with the temperatures to send, and it has to send the values every 10 seconds following the order. When the last value in the CSV is sent, it will start again from the beginning.
  Each Gateway must send the values using a different userID to identify who is sending the values.
  We will run a random number of gateways, so it has to be made easy to create a new one.
  All data will be sent using MQTT (25%).

- The Cloud Service will receive the information from MQTT and will send it to the Database of your choice. The database must be used so it is easy to get all data sent by a User. (25%).

2on level Functions [30%]

- The Cloud Provider will use Kafka to send the data to the analytics module that will process the information and return the result in another Kafka topic. This information must also be stored in the Database.

3rd level Functions [10 %]

- Create or use some of the visualization tools (like influxDB dashboards) to show the data in the database and see the results of the model.
- Create a docker-compose file that contains all components required.

Quality Features [10%]

The next features provide extra points.

- Data storage is done considering optimality
- The deliverable is well redacted and includes the correct reasoning and explanation of why it is implemented this way.

- A docker-compose file is delivered that runs the project automatically

Considerations

Considerations

You face the development of a distributed application. Your solution should provide correct implementation of the functionalities and consider efficiency, fault tolerance and performance.

You can choose the technology to develop the project, but all decisions must be justified. The technology you choose must consider facilities to efficiently develop distributed applications with access to remote objects or services, etc. The technology chosen cannot be an excuse neither to satisfy the project requirements nor for a correct implementation of the functionalities.

Deliveries

Report content

Create a report with the following contents:

- Description of each component.
- Summarize the main design decisions done in this project such as technology used, database used and data model.
- A well detailed documentation on how to run the project. It is better if all modules are run using a single docker compose.
-

Instructions

Work in pairs to develop the distributed application project. Submit a report with the contents specified above. The final source code and deliverable should be upload to the virtual campus. Do not forget to indicate in the report the time spent on the activity. **There will be a face-to-face presentation with the professors. In the case a group deliver the project after the deadlines there will be penalizations on the score.**