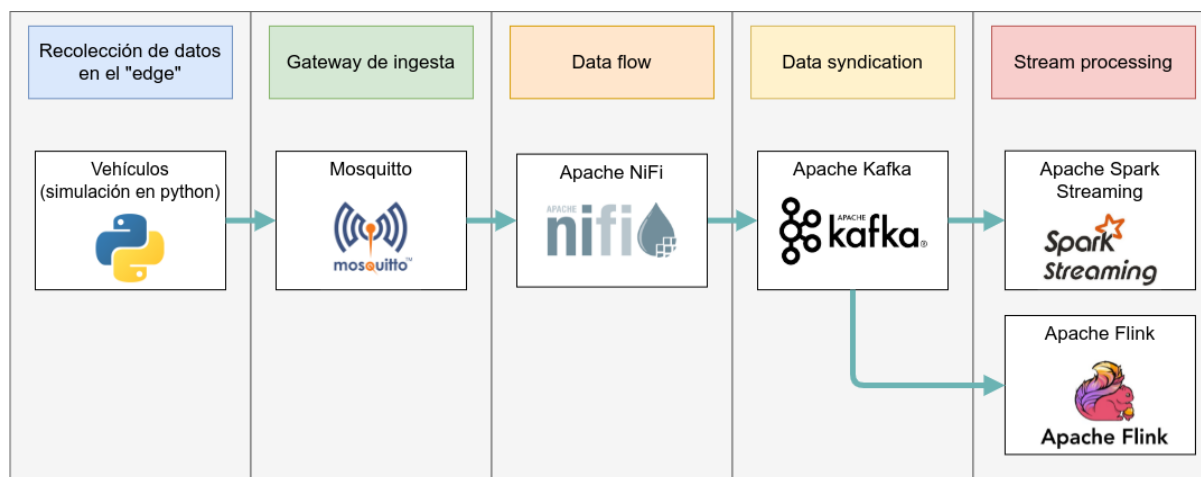


Introducción

En este práctico utilizaremos un generador de datos de GPS de vehículos, que los enviará a un broker MQTT Mosquitto.

Este será el primer paso para la implementación de nuestro sistema de streaming basado en una arquitectura de referencia para IoT:



Parte 1

Descargar de Aulas los archivos:

- Productor Mosquitto.ipynb
- Consumidor Mosquitto.ipynb

Subir esos archivos de notebooks a Jupyter Notebook con la opción upload.

En Jupyter, abrir ambos notebooks cargados.

Iniciar ambas notebooks y observar lo que sucede.

Al finalizar interrumpir la ejecución y apagar las notebooks.

Parte 2

Descargar el simulador

En una terminal SSH, ejecutar los siguientes comando para descargar el simulador de datos:

```
wget https://github.com/nfornaro/ARQ-Simulador/archive/refs/heads/master.zip  
unzip master.zip  
rm master.zip  
mv ARQ-Simulador-master/ simulador
```

Generación de datos

Para generar los datos debemos ejecutar:

```
cd simulador  
python generador.py
```

Y cuando solicite por el script, indicar "script.txt" con las comillas incluidas.

Ejemplo de salida:

```
[azureuser@tatr-vm simulador]$ python generador.py  
Cargando dataset ...  
84507 puntos cargados  
script? "script.txt"  
[azureuser@tatr-vm simulador]$
```

Envío de datos y consumo de datos

Configuración del envío

Editar el archivo reproductor.py (por ejemplo, con vi/vim).

En la sección de código:

```
if mode == 'mqtt':  
    self.client = mqtt.Client()  
    self.client.connect("127.0.0.1", 8000, 60)
```

Cambiar el número 8000 por 1883, que indica el puerto en el que ejecuta mosquitto:

```
if mode == 'mqtt':  
    self.client = mqtt.Client()  
    self.client.connect("127.0.0.1", 1883, 60)
```

Configuración del consumidor

Abrir el notebook Consumidor Mosquito.

En el primer bloque, cambiar el nombre del tópico de “test-topic” a “gps”:

```
topic = "test-topic"
```

Por:

```
topic = "gps"
```

Ejecución

Ejecutar todo el notebook.

Ejecutar el reproductor:

```
/home/azureuser/conda/bin/python reproductor.py
```

Elegir el intervalo de segundos, por ejemplo, 2.

Se deberá ver en el notebook la información entrante.

Para cancelar el envío, interrumpir el proceso ejecutando control + C.

Detener y apagar el notebook.