Install AWS Boto3 in EC2 machine

sudo yum update -y

sudo yum install -y python3-pip python3 python3-setuptools

pip3 install boto3 --user

Ejercicio en clase

Esto es un ejemplo de como un sensor de temperatura y humedad enviá lo que mide a la nube directamente o a traves de un IoT Gateway:

```
{
        "sensor_id": "THS-001",
        "timestamp": "2024-01-01T12:30:00Z",
        "temperature": 23.5,
        "humidity": 60.2,
        "location": {
                "latitude": 37.7749,
                "longitude": -122.4194
        },
        "battery_level": 95
}
```

donde:

- sensor_id: Identificador único del sensor.
- timestamp: Marca de tiempo de la lectura (en formato ISO 8601).
- temperature: Temperatura en grados Celsius.
- humidity: Humedad relativa en porcentaje.
- location: Coordenadas geográficas del sensor.
- battery_level: Nivel de batería del sensor en porcentaje.

1. Hacer un código en python que permita simular el comportamiento de varios sensores. Para esto el código en python deberá generar varios archivos .json. Cada archivo json debe tener un array de mediciones como se muestra en el ejemplo de abajo.

Cada medición individual se debe simular con datos aleatorios (usar datos razonables para temperatura, humedad y nivel de batería. La fecha (o timestamp) se debe

generar en intervalos de máximo un minuto. La latitud y longitud deben ser fijas por cada sensor, dado que estos sensores no son moviles.

El json que agrupa varias mediciones se debe subir a un bucket S3 que usted creara con el nombre que usted defina. El intervalo entre archivos json puede ser de máximo un minuto también. Su programa debe generar al menos 5 archivos a S3. Y cada archivo debe tener no menos de 10 diferentes mediciones de diferentes dispositivos.

La idea de que un archivo tenga varias mediciones es implementar un comportamiento del IoT Gateway, de agrupar varias mediciones, crear un paquete y enviarlo a la nube.

Ejemplo:

```json
{
    "measurements": [
        {
            "sensor_id": "THS-001",
            "timestamp": "2024-01-01T12:30:00Z",
            "temperature": 23.5,
            "humidity": 60.2,
            "location": {
                "latitude": 37.7749,
                "longitude": -122.4194
            },
            "battery_level": 95
        },
        {
            "sensor_id": "THS-003",
            "timestamp": "2024-01-01T12:31:00Z",
            "temperature": 23.6,
            "humidity": 60.3,
            "location": {
                "latitude": 37.7749,
                "longitude": -122.4194
            },
            "battery_level": 94
        },
        {
            "sensor_id": "THS-002",
            "timestamp": "2024-01-01T12:32:00Z",
            "temperature": 24.6,
            "humidity": 61.3,
            "location": {
                "latitude": 37.7749,
                "longitude": -122.4194
            },
            "battery_level": 93
        },
        {
            "sensor_id": "THS-001",
            "timestamp": "2024-01-01T12:31:00Z",
            "temperature": 24.5,
            "humidity": 61.2,
            "location": {
                "latitude": 37.7749,
                "longitude": -122.4194
            },
            "battery_level": 80
        }
    ]
}
```

2. Hacer un programa en python que lea todos los archivos que hay en el bucket, abra cada uno de los archivos que se enviaron en el punto uno y cargue la información a una base de datos postgresql. Diseñe un DDL para guardar la información.

3. Haga un query SQL que calcule el promedio de temperatura de cada dispositivo.

4. Haga un query que calcule cuantas mediciones llegaron de cada dispositivo.

Nota:

- Guardar el código en su GitHub o repositorio personal.
- Recuerde que la sesión del Learner Lab dura máximo 4 horas.

Recomendaciones:

- Adelante su código primero antes de iniciar el laboratorio del Learner Lab.
- Pueden trabajarlo en equipos de máximo 3 personas.