



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**INFORME DE LABORATORIO N° 06
“INGESTA DE DATOS EN AWS CON KINESIS
DATA FIREHOSE”**

Curso: Inteligencia de Negocios

Docente: Ing. Patrick Cuadros Quiroga

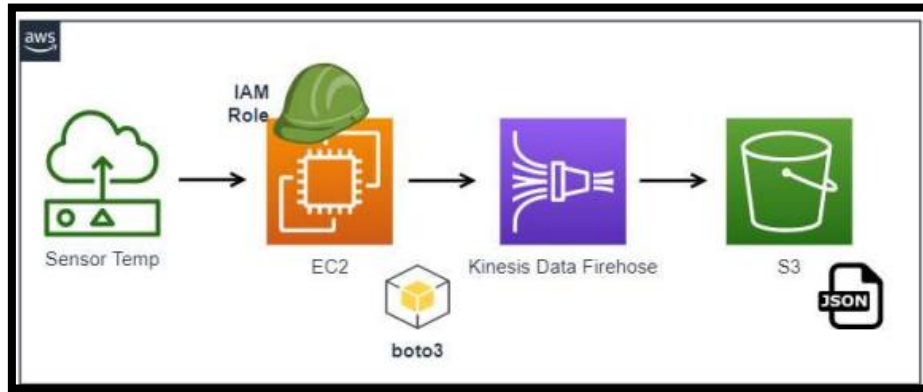
Anahua Huayhua, Jenny Karen (2018062150)

**Tacna – Perú
2022**

PRACTICA DE LABORATORIO N° 06**TEMA: INGESTA DE DATOS EN AWS CON KINESIS DATA FIREHOSE**

Realizar los siguientes pasos para el laboratorio

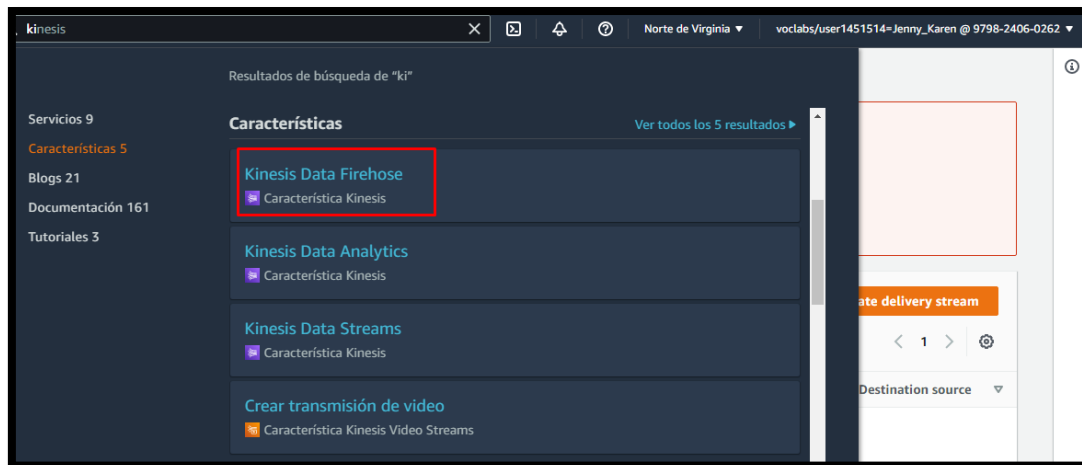
1. ingestando datos a Firehose mediante el SDK de AWS



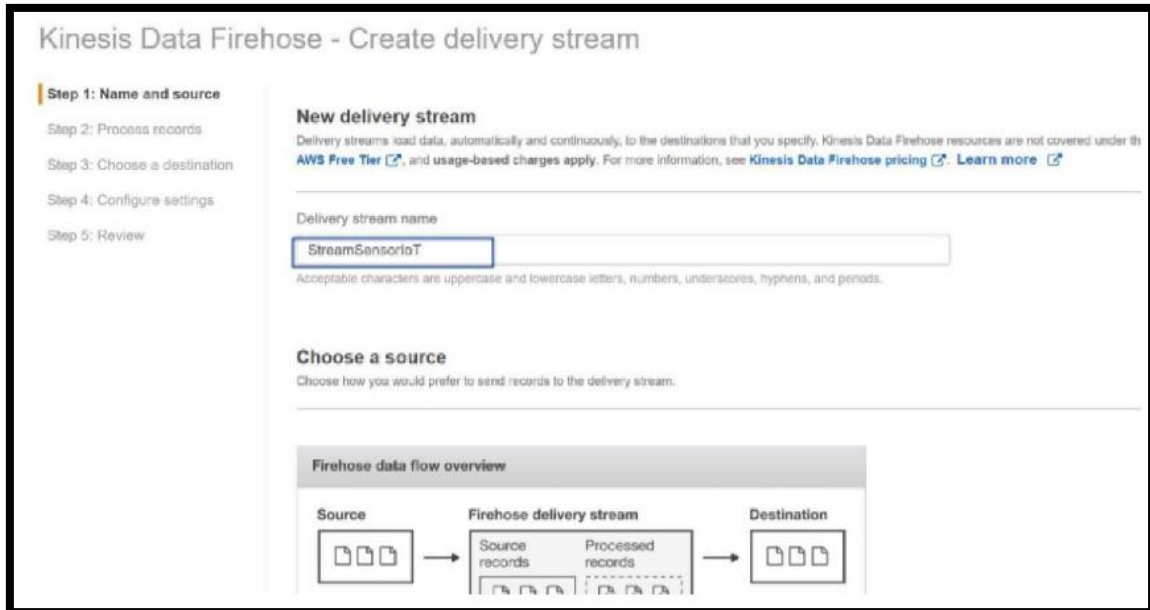
<https://www.kaggle.com/atulanandjha/temperature-readings-iot-devices>

Para este flujo near real time, leeremos un dataset descargado de Kaggle. Este conjunto de datos contiene las lecturas de temperatura de los dispositivos IOT instalados fuera y dentro de una sala.

- a) Entrar a la consola de AWS.
- b) Ir al servicio de Kinesis Firehose, clic en Create delivery stream



- c) Crear el stream con el nombre de StreamSensorIoT, y luego siguiente y otra vez siguiente
- d) En la siguiente ventana, dejamos marcado S3, porque es ahí donde almacenaremos los datos que se agregarán a Kinesis Data Firehose.



Kinesis Data Firehose - Create delivery stream

Step 1: Name and source

Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review

New delivery stream

Delivery streams load data, automatically and continuously, to the destinations that you specify. Kinesis Data Firehose resources are not covered under the [AWS Free Tier](#), and usage-based charges apply. For more information, see [Kinesis Data Firehose pricing](#). [Learn more](#)

Delivery stream name

Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.

Choose a source

Choose how you would prefer to send records to the delivery stream.

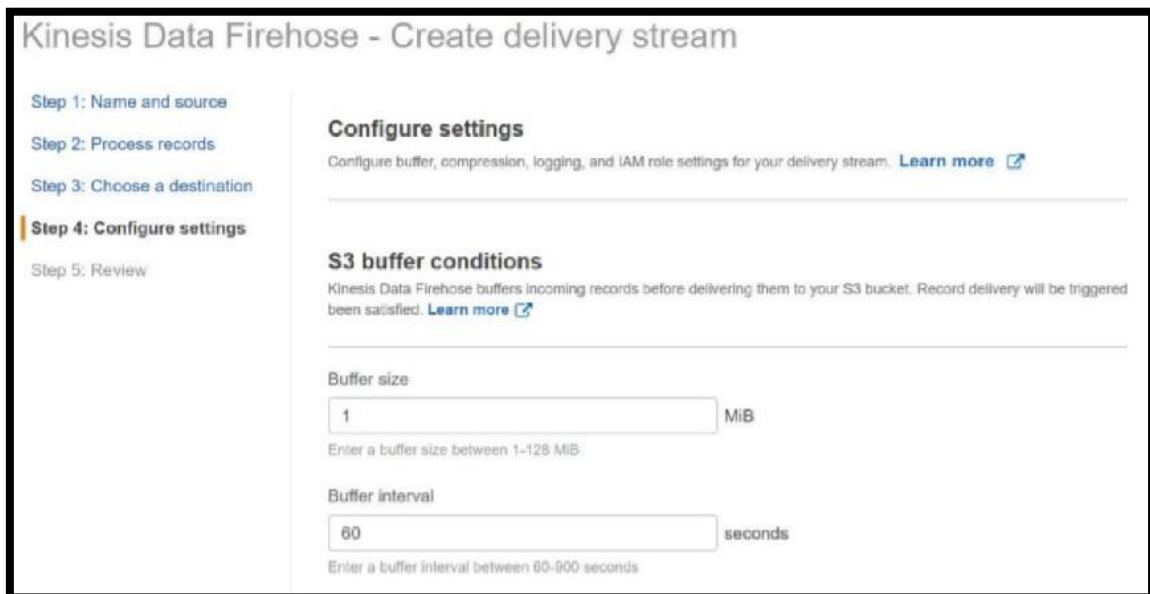
Firehose data flow overview

Source → Firehose delivery stream → Destination

The diagram shows a flow from a Source (represented by three document icons) to a Firehose delivery stream (represented by a box divided into 'Source records' and 'Processed records' with document icons), and finally to a Destination (represented by three document icons).

e) Creamos un bucket en la siguiente pantalla. (Los nombres de los bucket son únicos globalmente)

f) En la siguiente ventana, definimos que el tamaño del búfer será de 1MB y el intervalo de tiempo es de 60 segundos. (Recordar que al cumplirse una de estas dos condiciones los datos se guardarán en S3)



Kinesis Data Firehose - Create delivery stream

Step 1: Name and source
Step 2: Process records
Step 3: Choose a destination
Step 4: Configure settings
Step 5: Review

Configure settings

Configure buffer, compression, logging, and IAM role settings for your delivery stream. [Learn more](#)

S3 buffer conditions

Kinesis Data Firehose buffers incoming records before delivering them to your S3 bucket. Record delivery will be triggered when the buffer size or buffer interval has been satisfied. [Learn more](#)

Buffer size
 MiB
Enter a buffer size between 1-128 MiB

Buffer interval
 seconds
Enter a buffer interval between 60-900 seconds

g) Seleccionamos la opción de crear un rol de IAM y Next. (Este rol, permitirá escribir los resultados en S3)

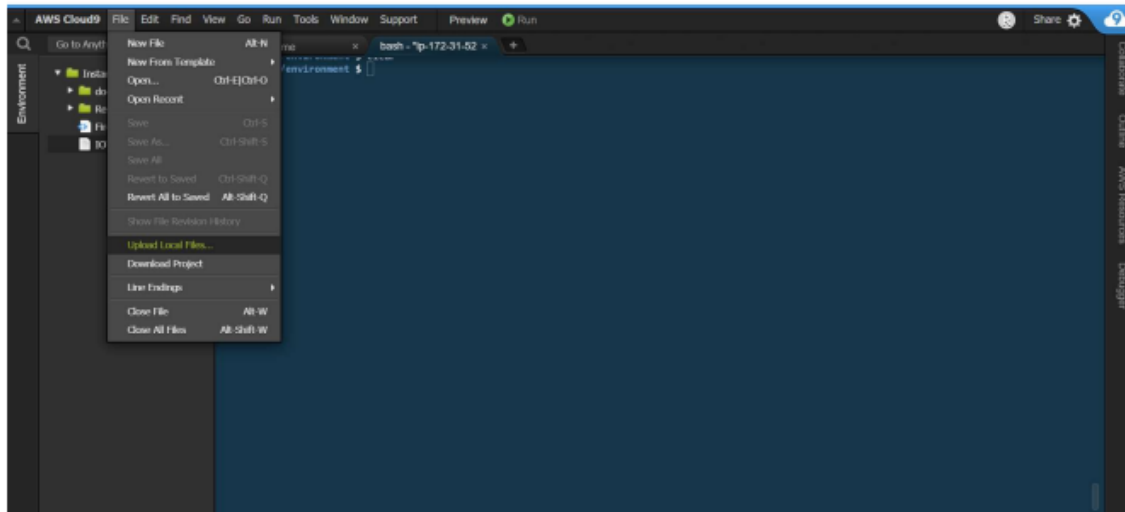
h) Siguiente y Crear delivery stream.

i) El stream en Kinesis Data Firehose se ha creado.

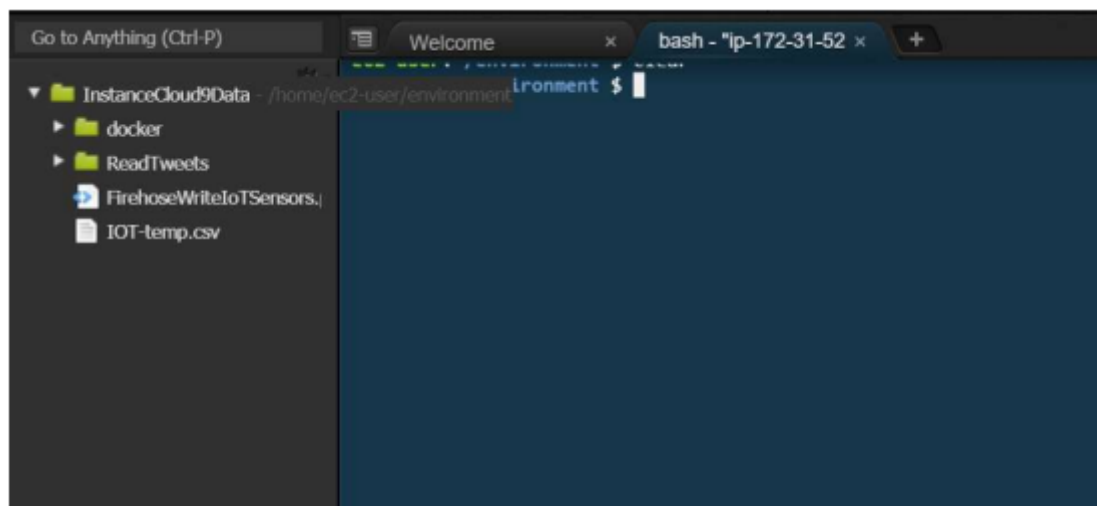
j) Entramos a Cloud9, clic en Open IDE.



Subiremos los archivos del laboratorio: - FirehoseWriteloTSensors.py - IOT-temp.csv Clic en File
-> Upload Local Files



Aparecerá el siguiente popup, arrastramos los archivos en Drag & Drop y listo. Debería quedar así.



Ejecutar los siguientes comandos en la línea de comandos:

curl -O <https://bootstrap.pypa.io/get-pip.py> # Get the install script.

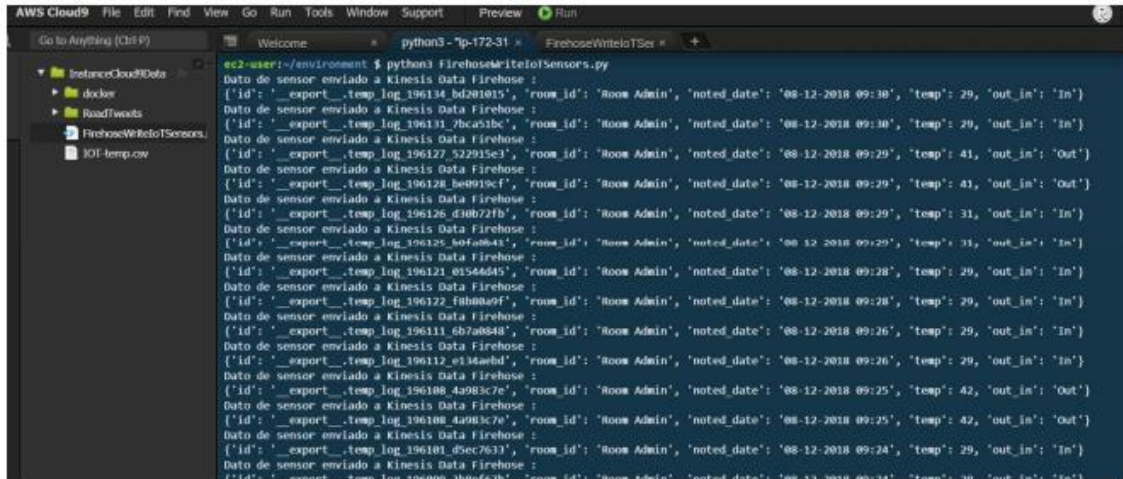
sudo python3 get-pip.py # Install pip for Python 3.6.

python -m pip --version # Verify pip is installed.

rm get-pip.py

sudo python3 -m pip install pandas

Ejecutar el siguiente Python: `python3 FirehoseWriteloTSensors.py`



```
ec2-user:~/environment $ python3 FirehoseWriteToSensors.py
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196134_bd201015', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:30', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196131_7bca51bc', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:30', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196127_522915e1', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:29', 'temp': 41, 'out_in': 'Out'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196128_b00910e1', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:29', 'temp': 41, 'out_in': 'Out'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196126_d30b72fb', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:29', 'temp': 31, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196125_b0f0a041', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:29', 'temp': 31, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196121_01544445', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:28', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196122_f8bba09f', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:28', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196111_6b7a0848', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:26', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196112_013a0ebd', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:26', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196108_4a983c7e', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:25', 'temp': 42, 'out_in': 'Out'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196108_4a983c7e', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:25', 'temp': 42, 'out_in': 'Out'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196101_d5ec7633', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:24', 'temp': 29, 'out_in': 'In'}
Dato de sensor enviado a Kinesis Data Firehose :
{'id': '__export__temp_log_196099_3b0cf03b', 'room_id': 'Room Admin', 'noted_date': '08-12-2018 09:24', 'temp': 29, 'out_in': 'In'}
```

Después de un minuto, en S3 debemos tener archivos con la información de los sensores.

Esperamos aproximadamente un minuto que se cumpla el umbral de tiempo del stream de Firehose y genere los archivos en S3, una vez veamos los registros en S3, en el terminal hacemos Ctrl + C para matar el proceso y no seguir ingestado. (Realizar esta acción para evitar gastos innecesarios)

Conclusión

Mediante el presente laboratorio se pudo realizar ingesta de datos en AWS con kinesis data firehose, siguiendo los pasos detallados del docente.