

Procesamiento en Batch

- **Realizar el cargue del dataset en la herramienta Hadoop.**

Requisitos previos: Tener instalado spark en Ubuntu

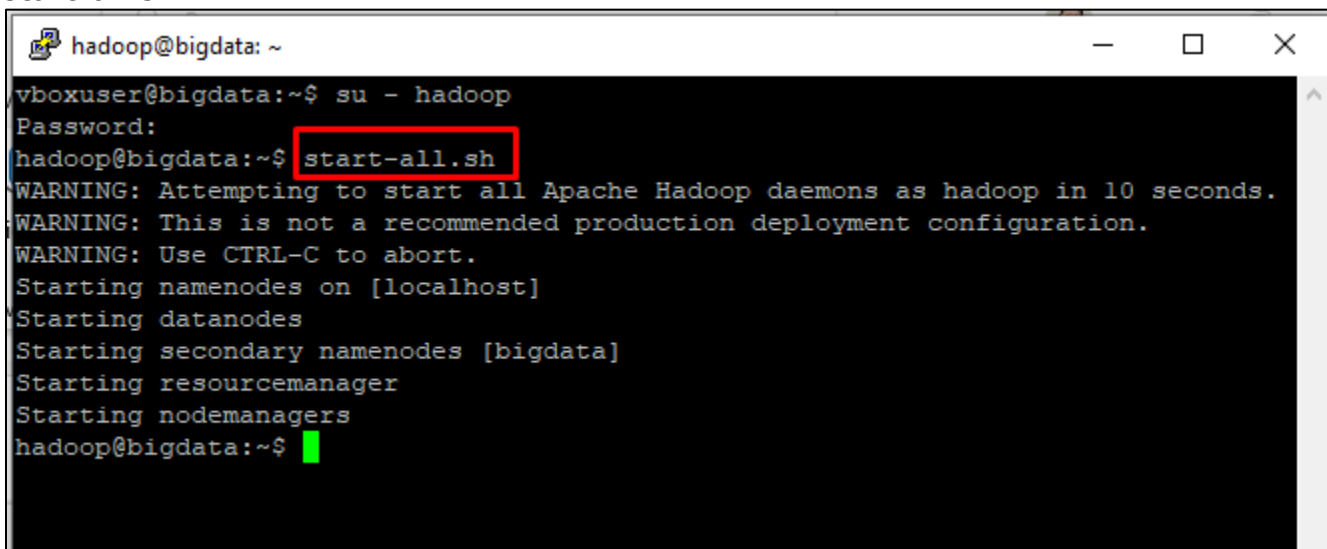
Iniciamos una sesión por putty e ingresamos con el usuario:

Usuario: hadoop

Password: hadoop

Iniciamos el clúster de Hadoop con el siguiente comando:

start-all.sh

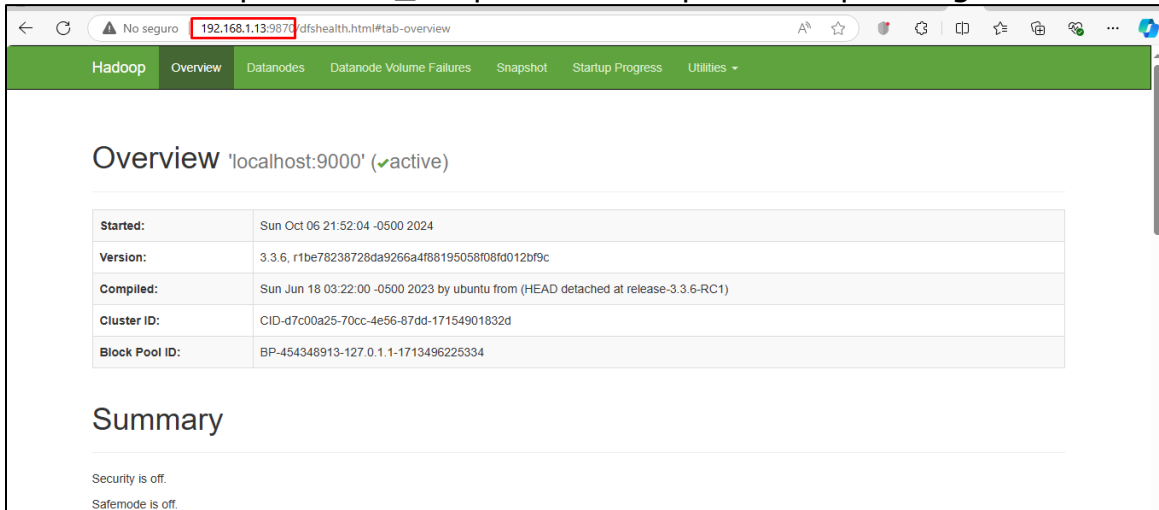
A terminal window titled 'hadoop@bigdata: ~' showing the execution of the 'start-all.sh' script. The user 'vboxuser' switches to 'hadoop' using 'su - hadoop' and enters the password. Then, 'start-all.sh' is executed. The terminal displays several warning messages about starting all daemons and non-recommended configuration, followed by status messages for starting namenodes, datanodes, secondary namenodes, resource manager, and node managers. The prompt returns to 'hadoop@bigdata:~\$' with a green cursor.

```
hadoop@bigdata: ~  
vboxuser@bigdata:~$ su - hadoop  
Password:  
hadoop@bigdata:~$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [bigdata]  
Starting resource manager  
Starting node managers  
hadoop@bigdata:~$
```

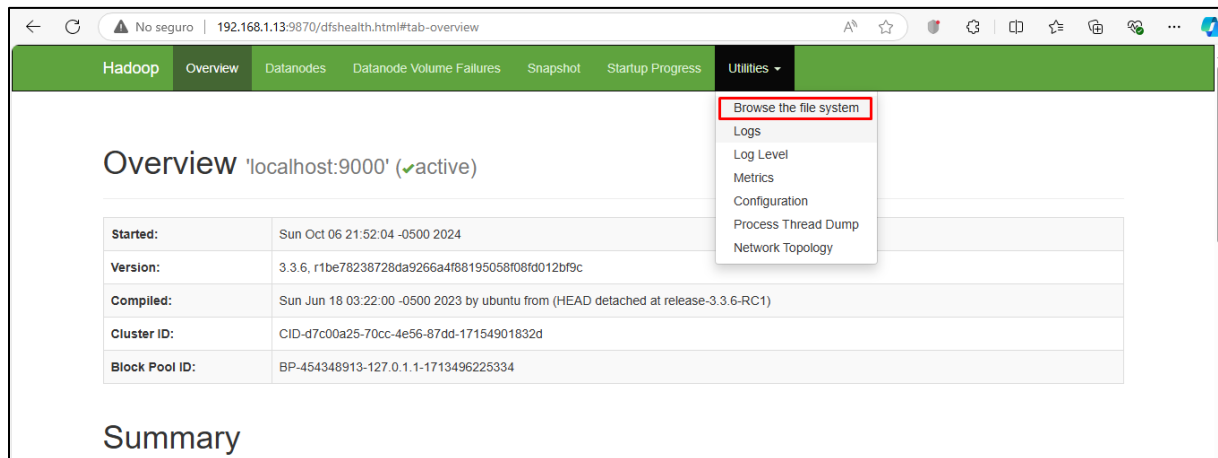
Una vez iniciados todos los servicios, puede acceder al Hadoop en:

http://IP_MaquinaVirtual:9870

Importante: *Reemplazar "IP_MaquinaVirtual" por la IP que tenga el servidor*

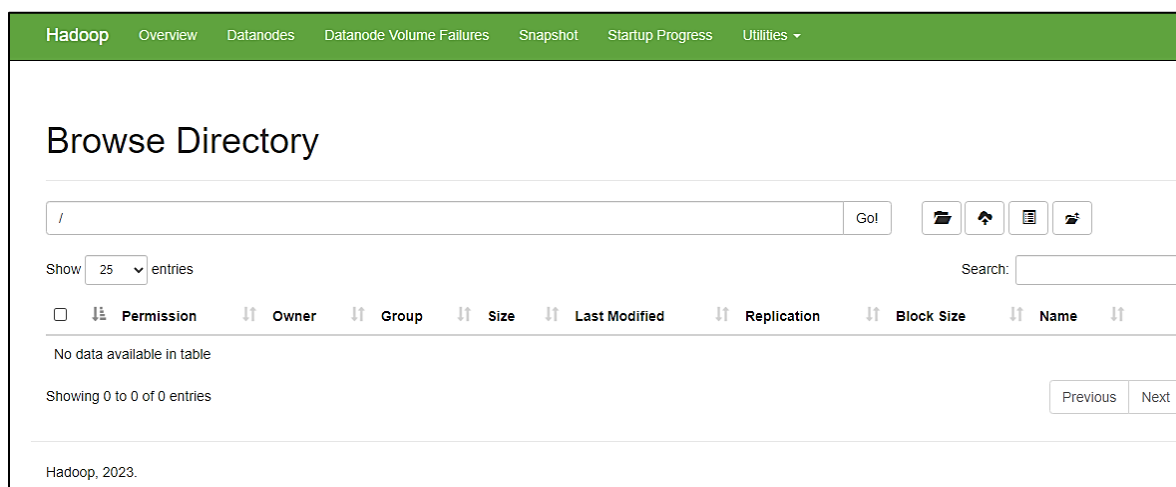


Puede visualizar el sistema de archivos HDFS en la opción Utilities-Browse the file system



The screenshot shows the Hadoop web interface at `192.168.1.13:9870/dfshealth.html#tab-overview`. The 'Utilities' dropdown menu is open, highlighting 'Browse the file system'. The main content area displays the 'Overview' for 'localhost:9000' (active), including a table with system details.

| Property | Value |
|----------------|--|
| Started: | Sun Oct 06 21:52:04 -0500 2024 |
| Version: | 3.3.6, r1be78238728da9266a4f88195058f08fd012bf9c |
| Compiled: | Sun Jun 18 03:22:00 -0500 2023 by ubuntu from (HEAD detached at release-3.3.6-RC1) |
| Cluster ID: | CID-d7c00a25-70cc-4e56-87dd-17154901832d |
| Block Pool ID: | BP-454348913-127.0.1.1-1713496225334 |



The screenshot shows the 'Browse Directory' page in the Hadoop web interface. It features a search bar, a 'Go!' button, and a table with columns for Permission, Owner, Group, Size, Last Modified, Replication, Block Size, and Name. The table is currently empty, displaying 'No data available in table'.

Ahora se crea una carpeta en el sistema HDFS utilizando el comando:

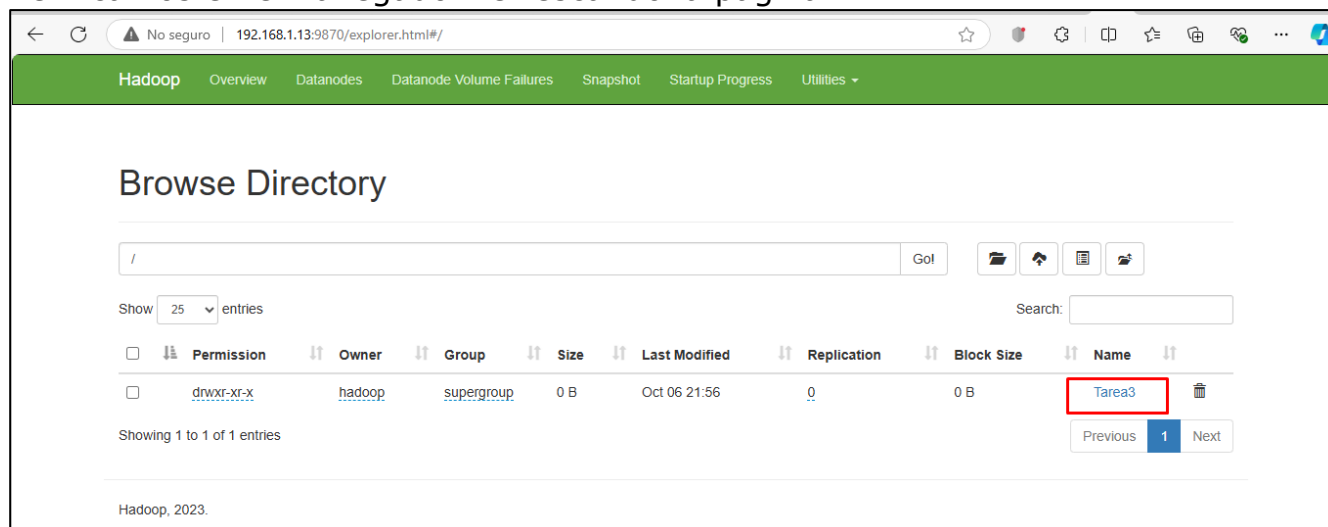
```
hdfs dfs -mkdir /Tarea3
```

con esto creamos una carpeta llamada Tarea3 en el sistema HDFS:



```
hadoop@bigdata: ~  
vboxuser@bigdata:~$ su - hadoop  
Password:  
hadoop@bigdata:~$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [bigdata]  
Starting resourcemanager  
Starting nodemanagers  
hadoop@bigdata:~$ hdfs dfs -mkdir /Tarea3  
hadoop@bigdata:~$
```

Verificamos en el navegador refrescando la página:



Con la siguiente dirección WEB y usando el comando wget, descargamos el dataset que utilizaremos para el ejercicio:

<https://www.datos.gov.co/api/views/g4vd-w4ip/rows.csv>

Comando:

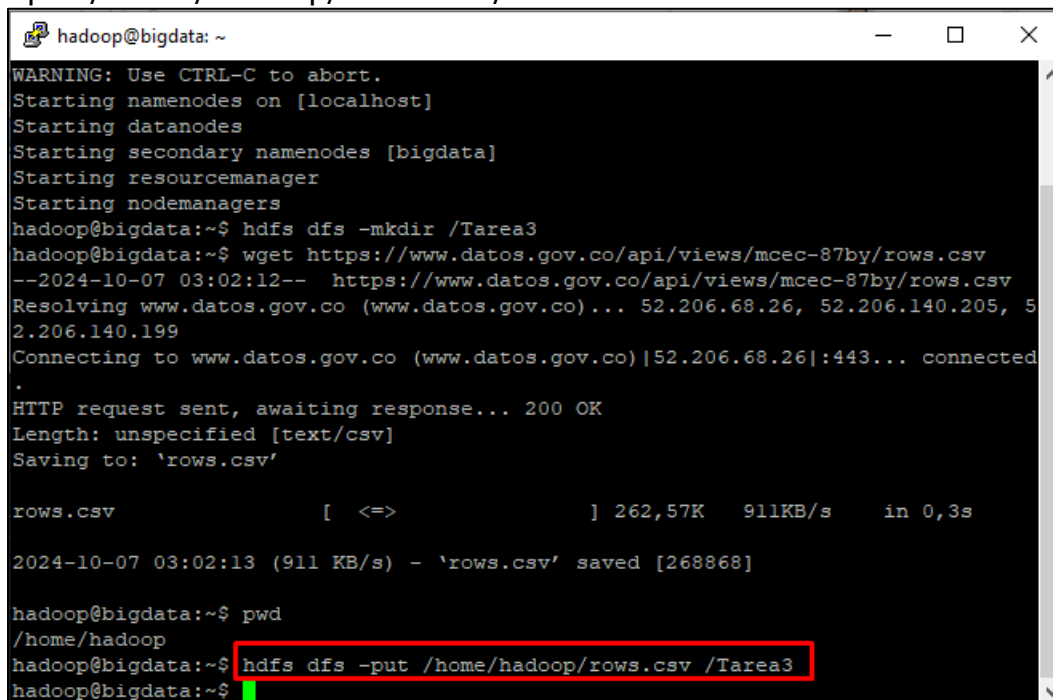
wget <https://www.datos.gov.co/api/views/mcec-87by/rows.csv>

Ejemplo:

```
hadoop@bigdata: ~  
hadoop@bigdata:~$ start-all.sh  
WARNING: Attempting to start all Apache Hadoop daemons as hadoop in 10 seconds.  
WARNING: This is not a recommended production deployment configuration.  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [bigdata]  
Starting resourcemanager  
Starting nodemanagers  
hadoop@bigdata:~$ hdfs dfs -mkdir /Tarea3  
hadoop@bigdata:~$ wget https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
--2024-10-07 03:02:12-- https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
Resolving www.datos.gov.co (www.datos.gov.co)... 52.206.68.26, 52.206.140.205, 52.206.140.199  
Connecting to www.datos.gov.co (www.datos.gov.co)|52.206.68.26|:443... connected  
.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/csv]  
Saving to: 'rows.csv'  
  
rows.csv          [ <=>          ] 262,57K   911KB/s   in 0,3s  
  
2024-10-07 03:02:13 (911 KB/s) - 'rows.csv' saved [268868]  
hadoop@bigdata:~$
```

Ahora copiamos el archivo del Dataset descargado a la carpeta HDFS que creamos anteriormente con el siguiente comando:

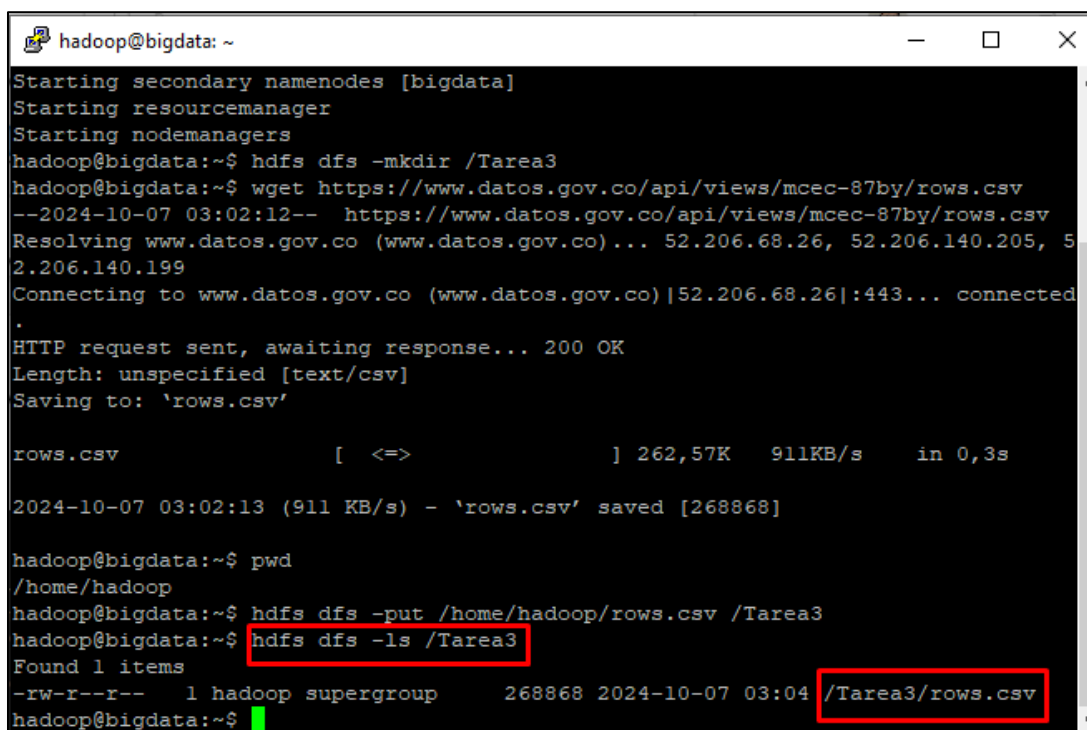
```
hdfs dfs -put /home/hadoop/rows.csv /Tarea3
```

A terminal window titled 'hadoop@bigdata: ~' showing the process of setting up HDFS and uploading a file. The commands executed are: 'hdfs dfs -mkdir /Tarea3', 'wget https://www.datos.gov.co/api/views/mcec-87by/rows.csv', and 'hdfs dfs -put /home/hadoop/rows.csv /Tarea3'. The output shows the directory creation, file download details (262,57K, 911KB/s), and the successful upload to HDFS. The 'hdfs dfs -put' command is highlighted with a red box.

```
hadoop@bigdata: ~  
WARNING: Use CTRL-C to abort.  
Starting namenodes on [localhost]  
Starting datanodes  
Starting secondary namenodes [bigdata]  
Starting resourcemanager  
Starting nodemanagers  
hadoop@bigdata:~$ hdfs dfs -mkdir /Tarea3  
hadoop@bigdata:~$ wget https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
--2024-10-07 03:02:12-- https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
Resolving www.datos.gov.co (www.datos.gov.co)... 52.206.68.26, 52.206.140.205, 52.206.140.199  
Connecting to www.datos.gov.co (www.datos.gov.co)|52.206.68.26|:443... connected  
.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/csv]  
Saving to: 'rows.csv'  
  
rows.csv          [  <=>          ] 262,57K   911KB/s   in 0,3s  
  
2024-10-07 03:02:13 (911 KB/s) - 'rows.csv' saved [268868]  
  
hadoop@bigdata:~$ pwd  
/home/hadoop  
hadoop@bigdata:~$ hdfs dfs -put /home/hadoop/rows.csv /Tarea3  
hadoop@bigdata:~$
```

Con el siguiente comando podemos ver el archivo del dataset en la carpeta HDFS llamada Tarea 3:

```
hdfs dfs -ls /Tarea3
```

A terminal window titled 'hadoop@bigdata: ~' showing the same HDFS setup as the previous screenshot, followed by the command 'hdfs dfs -ls /Tarea3'. The output shows the file 'rows.csv' with permissions '-rw-r--r--', owner 'hadoop', group 'supergroup', size '268868', and timestamp '2024-10-07 03:04'. The file path '/Tarea3/rows.csv' is highlighted with a red box.

```
hadoop@bigdata: ~  
Starting secondary namenodes [bigdata]  
Starting resourcemanager  
Starting nodemanagers  
hadoop@bigdata:~$ hdfs dfs -mkdir /Tarea3  
hadoop@bigdata:~$ wget https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
--2024-10-07 03:02:12-- https://www.datos.gov.co/api/views/mcec-87by/rows.csv  
Resolving www.datos.gov.co (www.datos.gov.co)... 52.206.68.26, 52.206.140.205, 52.206.140.199  
Connecting to www.datos.gov.co (www.datos.gov.co)|52.206.68.26|:443... connected  
.  
HTTP request sent, awaiting response... 200 OK  
Length: unspecified [text/csv]  
Saving to: 'rows.csv'  
  
rows.csv          [  <=>          ] 262,57K   911KB/s   in 0,3s  
  
2024-10-07 03:02:13 (911 KB/s) - 'rows.csv' saved [268868]  
  
hadoop@bigdata:~$ pwd  
/home/hadoop  
hadoop@bigdata:~$ hdfs dfs -put /home/hadoop/rows.csv /Tarea3  
hadoop@bigdata:~$ hdfs dfs -ls /Tarea3  
Found 1 items  
-rw-r--r--    1 hadoop supergroup    268868 2024-10-07 03:04 /Tarea3/rows.csv  
hadoop@bigdata:~$
```

NOTA: En este punto queda cargado el dataset en el sistema de archivos HDFS, esto es muy importante para ejecutar el código.

- **Ejecución del código**

Ejecutamos una nueva sesión por putty sin cerrar la que ya está establecida e ingresamos con el usuario:

Usuario: vboxuser

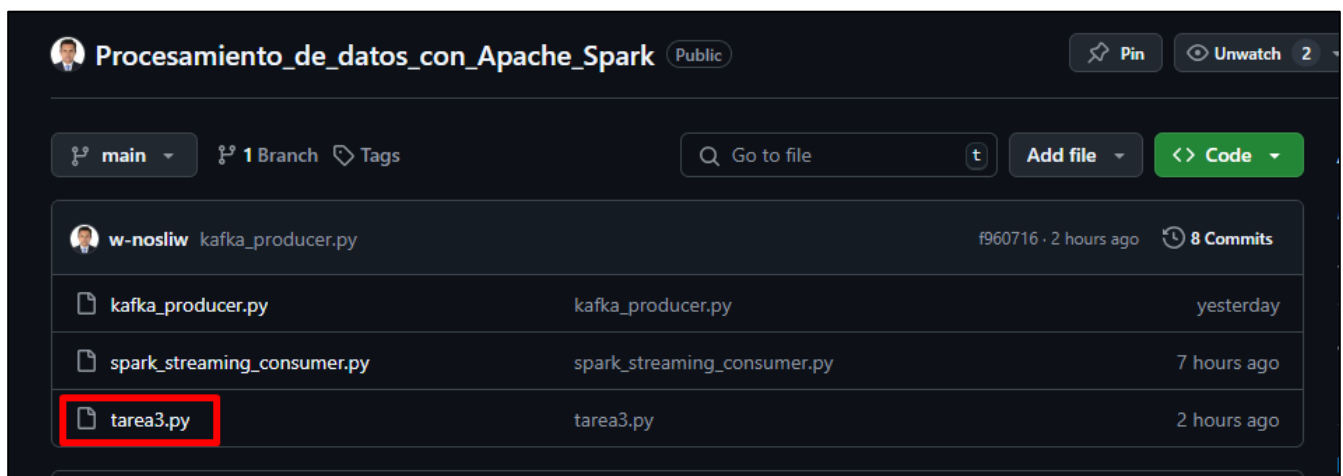
Contraseña: bigdata

Se crea un archivo de python con extension .py llamado tarea3.py, donde colocaremos el código del programa que se conectará al dataset descargado dentro del sistema HDFS, luego convertiremos estos datos en un dataframe de Spark y sobre este dataframe se realizarán diferentes operaciones y consultas. Para crea este archivo se ejecuta el siguiente comando:

nano tarea3.py

A terminal window with a black background and white text. The prompt is 'vboxuser@bigdata: ~'. The command 'nano tarea3.py' is entered and highlighted with a red rectangular box. A green cursor is visible at the end of the command.

Pegamos el código que se encuentra en este repositorio de GitHub con el nombre tarea3.py:



Importamos librerías necesarias

from pyspark.sql import SparkSession, functions as F

Inicializa la sesión de Spark

spark = SparkSession.builder.appName('Tarea3').getOrCreate()

```

# Define la ruta del archivo CSV
file_path = 'hdfs://localhost:9000/Tarea3/rows.csv'

# Lee el archivo CSV
df = spark.read.format('csv').option('header',
'true').option('inferSchema', 'true').load(file_path)

# Muestra el esquema del DataFrame
df.printSchema()

# Limpiar duplicados y eliminar valores nulos en columnas clave
df_clean = df.dropDuplicates().na.drop(subset=["EDAD", "SEXO", "TIPO
DE VEHÍCULO", "COSTOS", "FECHA DE ACCIDENTE"])

# ----- Análisis Exploratorio de Datos (EDA) -----
-----

# 1. Estadísticas básicas del conjunto de datos
print("Resumen de estadísticas del conjunto de datos:")
df_clean.summary().show()

# 2. Accidentes por tipo de vehículo
accidentes_por_vehiculo = df_clean.groupBy("TIPO DE
VEHÍCULO").count().orderBy(F.col("count").desc())
print("Accidentes por tipo de vehículo:")
accidentes_por_vehiculo.show()

# 3. Distribución de accidentes por día de la semana
accidentes_por_dia = df_clean.groupBy("DIA
SEMANA").count().orderBy(F.col("count").desc())
print("Distribución de accidentes por día de la semana:")
accidentes_por_dia.show()

# 4. Estadísticas de edad según el tipo de vehículo
edad_por_vehiculo = df_clean.groupBy("TIPO DE VEHÍCULO").agg(
    F.mean("EDAD").alias("Edad Promedio"),
    F.min("EDAD").alias("Edad Mínima"),
    F.max("EDAD").alias("Edad Máxima")
)
print("Estadísticas de edad por tipo de vehículo:")
edad_por_vehiculo.show()

# 5. Filtrado: Accidentes donde el usuario es conductor
accidentes_conductor = df_clean.filter(F.col("RELACION
USUARIO/ACCIDENTE") == "CONDUCTOR")
print("Accidentes donde el usuario es conductor:")
accidentes_conductor.show(5)

```

6. Accidentes por grupo etario

```
accidentes_por_grupo_etario = df_clean.groupBy("GRUPO  
ETAREO").count().orderBy(F.col("count").desc())  
print("Accidentes por grupo etario:")  
accidentes_por_grupo_etario.show()
```

7. Promedio de tiempo de atención (en horas) en accidentes

```
promedio_atencion_horas = df_clean.groupBy("TIPO DE  
VEHÍCULO").agg(  
    F.mean("OPORTUNIDAD DE LA ATENCIÓN EN HORAS").alias("Tiempo  
Promedio Atención (Horas)")  
)  
print("Promedio de tiempo de atención en horas según tipo de  
vehículo:")  
promedio_atencion_horas.show()
```

8. Accidentes filtrados por altos costos de atención médica

```
accidentes_costos_altos = df_clean.filter(F.col("COSTOS") >  
100000).select("NUMERO", "COSTOS", "TIPO DE VEHÍCULO", "SEXO")  
print("Accidentes con costos mayores a 100,000:")  
accidentes_costos_altos.show()
```

9. Número de accidentes por mes

```
accidentes_por_mes =  
df_clean.groupBy("MES").count().orderBy(F.col("count").desc())  
print("Número de accidentes por mes:")  
accidentes_por_mes.show()
```

10. Porcentaje de accidentes por sexo

```
accidentes_por_sexo =  
df_clean.groupBy("SEXO").count().withColumn("Porcentaje",  
(F.col("count") / df_clean.count()) * 100)  
print("Porcentaje de accidentes por sexo:")  
accidentes_por_sexo.show()
```

Detener la sesión de Spark

```
spark.stop()
```

Una vez pegado el código oprimir Crtl+O - enter y finalmente Crtl+X para salir del archivo.

Ejecutar el archivo con el siguiente comando:

```
python3 tarea3.py
```

```
vboxuser@bigdata: ~  
vboxuser@bigdata:~$ nano tarea3.py  
vboxuser@bigdata:~$ python3 tarea3.py  
24/10/07 03:23:27 WARN Utils: Your hostname, bigdata resolves to a loopback address: 127.0.1.1; using 192.168.1.13 instead (on interface enp0s3)  
24/10/07 03:23:27 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
24/10/07 03:23:29 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

Al terminar la ejecución del archivo debemos obtener resultados similares a los siguientes:

```
vboxuser@bigdata:~$ nano tarea3.py  
vboxuser@bigdata:~$ python3 tarea3.py  
24/10/17 01:36:52 WARN Utils: Your hostname, bigdata resolves to a loopback address: 127.0.1.1; using 192.168.130.43 instead (on interface enp0s3)  
24/10/17 01:36:52 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address  
Setting default log level to "WARN".  
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).  
24/10/17 01:36:53 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable  
root  
|-- Numero: integer (nullable = true)  
|-- EDAD: string (nullable = true)  
|-- SEXO: string (nullable = true)  
|-- TIPO DE VEHICULO: string (nullable = true)  
|-- RELACION USUARIO/ACCIDENTE: string (nullable = true)  
|-- FECHA DE ACCIDENTE: string (nullable = true)  
|-- FECHA DE INGRESO IPS: string (nullable = true)  
|-- HORA DE ACCIDENTE: string (nullable = true)  
|-- HORA DE INGRESO IPS: string (nullable = true)  
|-- FECHA DE ATENCION MEDICA: string (nullable = true)  
|-- HORA DE ATENCION MEDICA: string (nullable = true)  
|-- OPORTUNIDAD DE LA ATENCION EN HORAS: string (nullable = true)  
|-- OPORTUNIDAD DE LA ATENCION EN MINUTOS: string (nullable = true)  
|-- REQUERIMIENTO DE CX: string (nullable = true)  
|-- REFERIDO A OTRA IPS: string (nullable = true)  
|-- REGION ANATOMICA MAS AFECTADA: string (nullable = true)  
|-- POLITRAUMATISMO: string (nullable = true)  
|-- TIPO AFILIACION: string (nullable = true)  
|-- EPS USUARIO: string (nullable = true)  
|-- CONDICION EGRESO: string (nullable = true)  
|-- OPORTUNIDAD CITAS DE CONTROL: string (nullable = true)  
|-- COSTOS: string (nullable = true)  
|-- MES: string (nullable = true)  
|-- IPS: string (nullable = true)  
|-- GRUPO ETAREO: string (nullable = true)  
|-- CURSO DE VIDA: string (nullable = true)  
|-- AÑO: integer (nullable = true)  
|-- DIA SEMANA: string (nullable = true)  
|-- Código Administradora: string (nullable = true)  
|-- Administradora: string (nullable = true)  
|-- Tipo Administradora: string (nullable = true)  
|-- Coigo RIPS: string (nullable = true)  
|-- Descripción RIPS: string (nullable = true)  
  
Resumen de estadísticas del conjunto de datos:  
24/10/17 01:37:03 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spar
```


[illegible]

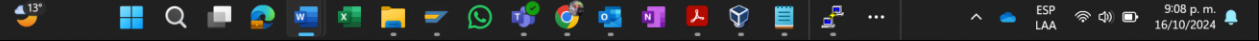
```
vboxuser@bigdata: ~
Accidentes por grupo etario:
+-----+
|GRUPO ETAREO|count|
+-----+
|29 a 59|16402|
|19 a 28|13748|
|60 y mas|1746|
|12 a 18|1704|
|7 a 11|270|
|0 a 6|206|
|#:NUM|2|
|ERROR: #N/A|1|
+-----+

Promedio de tiempo de atención en horas según tipo de vehiculo:
+-----+
|TIPO DE VEHÍCULO|Tiempo Promedio Atención (Horas)|
+-----+
|BICICLETA|NULL|
|BUS|2322.8|
|SIN INFORMACION|10543.804347826086|
|CARRO|552.8729508196722|
|MOTO|289.36148148148146|
|CAMIONETA|1693.857142857143|
|PEATON|NULL|
|CAMION|206.75|
|METROLINEA|NULL|
|AMBULANCIA|NULL|
|VOLQUETA|NULL|
+-----+

Accidentes con costos mayores a 100,000:
+-----+
|NUMERO|COSTOS|TIPO DE VEHÍCULO|SEXO|
+-----+
|20265|506646|MOTO|MASCULINO|
|20860|302820|MOTO|FEMENINO|
|21591|7666576|MOTO|FEMENINO|
|21696|222315|MOTO|MASCULINO|
|21891|215084|CAMION|FEMENINO|
|20381|212004|MOTO|MASCULINO|
|20490|1346395|MOTO|MASCULINO|
|21490|273600|MOTO|MASCULINO|
|21806|155705|MOTO|MASCULINO|
+-----+
```

Finalmente visualizamos un reporte del número de accidentes por mes y el porcentaje de accidentes por genero

```
vboxuser@bigdata: ~  
| 20490|1346395| MOTO|MASCULINO|  
| 21490| 273600| MOTO|MASCULINO|  
| 21806| 155705| MOTO|MASCULINO|  
| 20269| 282664| MOTO|MASCULINO|  
| 21032| 147889| CAMION|MASCULINO|  
| 21862| 135733| MOTO| FEMENINO|  
| 19863| 129197| MOTO|MASCULINO|  
| 19905| 172150| MOTO|MASCULINO|  
| 20266| 128650| MOTO|MASCULINO|  
| 20372| 278696| MOTO|MASCULINO|  
| 20427| 135346| MOTO|MASCULINO|  
| 20701| 101000| CARRO|MASCULINO|  
| 20781| 170450| MOTO|MASCULINO|  
| 20837| 326810| MOTO| FEMENINO|  
+-----+  
only showing top 20 rows  
  
Número de accidentes por mes:  
+-----+  
| MES|count|  
+-----+  
| 10. OCTUBRE| 3358|  
| 11. NOVIEMBRE| 3104|  
| 01. ENERO| 3101|  
| 03. MARZO| 3088|  
| 07. JULIO| 3069|  
| 02. FEBRERO| 3019|  
| 09. SEPTIEMBRE| 2980|  
| 08. AGOSTO| 2818|  
| 05. MAYO| 2664|  
| 06. JUNIO| 2612|  
| 04. ABRIL| 2607|  
| 12. DICIEMBRE| 1659|  
+-----+  
  
Porcentaje de accidentes por sexo:  
+-----+  
| SEXO|count| Porcentaje|  
+-----+  
| FEMENINO|11231|32.95577921887379|  
| MASCULINO|22848|67.04422078112621|  
+-----+  
  
vboxuser@bigdata:~$
```



Procesamiento en tiempo real (Spark Streaming & Kafka):

Requisitos previos: Tener instalado Spark y Kafka en ubuntu

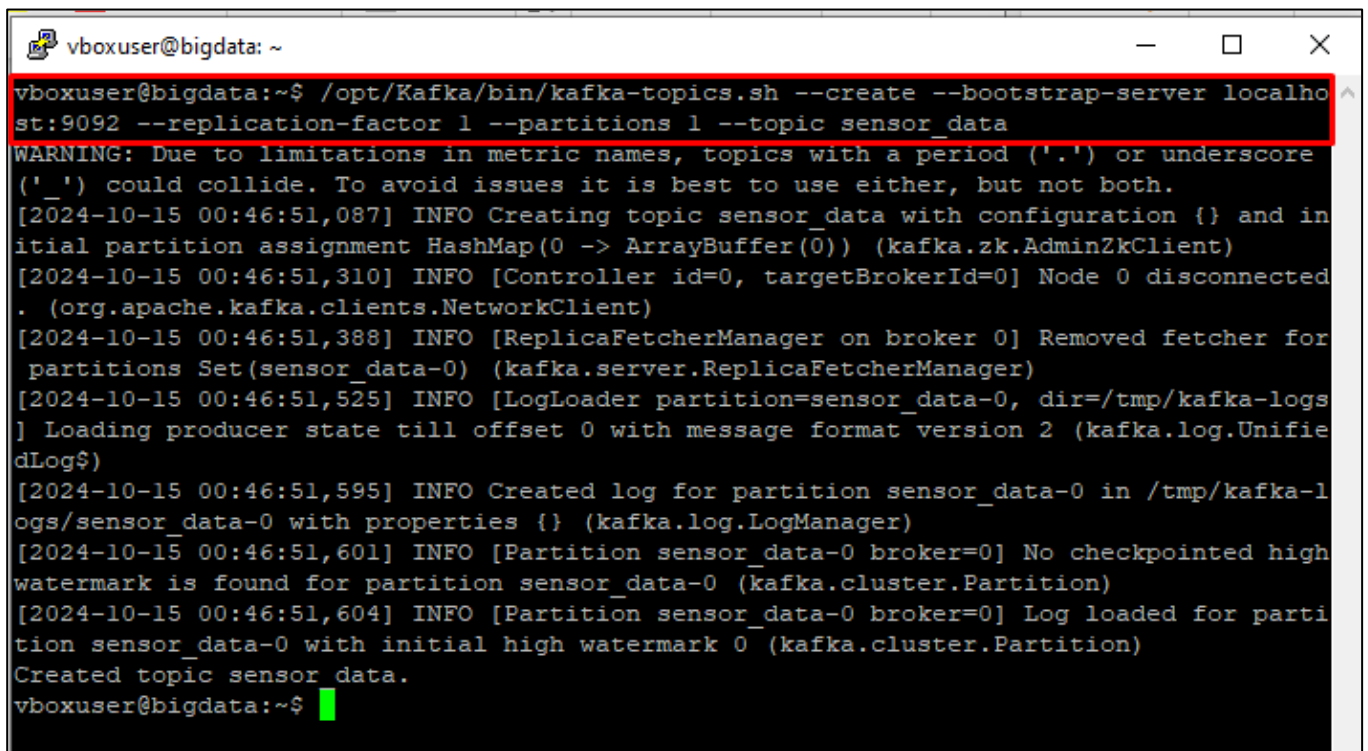
Ejecutamos una sesión por putty e ingresamos con el usuario:

Usuario: vboxuser

Contraseña: bigdata

Creamos un tema (topic) de Kafka, el tema se llamará sensor_data con el siguiente comando:

```
/opt/Kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic sensor_data
```

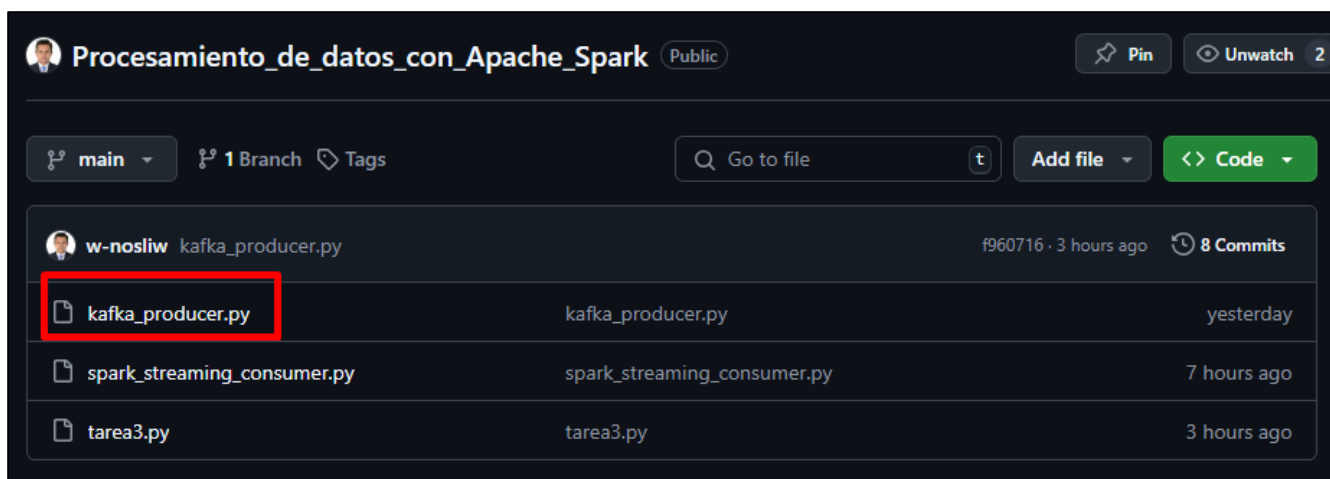
A terminal window titled 'vboxuser@bigdata: ~' showing the execution of the command `/opt/Kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic sensor_data`. The command is highlighted with a red box. Below the command, the terminal displays several log messages from Kafka, including a warning about metric names, and finally confirms 'Created topic sensor_data.' before returning to the prompt `vboxuser@bigdata:~$`.

```
vboxuser@bigdata:~$ /opt/Kafka/bin/kafka-topics.sh --create --bootstrap-server localhost:9092 --replication-factor 1 --partitions 1 --topic sensor_data
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
[2024-10-15 00:46:51,087] INFO Creating topic sensor_data with configuration {} and inital partition assignment HashMap(0 -> ArrayBuffer(0)) (kafka.zk.AdminZkClient)
[2024-10-15 00:46:51,310] INFO [Controller id=0, targetBrokerId=0] Node 0 disconnected. (org.apache.kafka.clients.NetworkClient)
[2024-10-15 00:46:51,388] INFO [ReplicaFetcherManager on broker 0] Removed fetcher for partitions Set(sensor_data-0) (kafka.server.ReplicaFetcherManager)
[2024-10-15 00:46:51,525] INFO [LogLoader partition=sensor_data-0, dir=/tmp/kafka-logs] Loading producer state till offset 0 with message format version 2 (kafka.log.UnifiedLog)
[2024-10-15 00:46:51,595] INFO Created log for partition sensor_data-0 in /tmp/kafka-logs/sensor_data-0 with properties {} (kafka.log.LogManager)
[2024-10-15 00:46:51,601] INFO [Partition sensor_data-0 broker=0] No checkpointed high watermark is found for partition sensor_data-0 (kafka.cluster.Partition)
[2024-10-15 00:46:51,604] INFO [Partition sensor_data-0 broker=0] Log loaded for partition sensor_data-0 with initial high watermark 0 (kafka.cluster.Partition)
Created topic sensor_data.
vboxuser@bigdata:~$
```

Creamos un archivo llamado kafka_producer.py con el siguiente comando:

```
nano kafka_producer.py
```

Pegamos el código que se encuentra en este repositorio de GitHub con el nombre kafka_producer.py:



```
import time
import json
import random
from kafka import KafkaProducer

def generate_sensor_data():
    return {
        "sensor_id": random.randint(1, 10),
        "temperature": round(random.uniform(20, 30), 2),
        "humidity": round(random.uniform(30, 70), 2),
        "pressure": round(random.uniform(900, 1100), 2), # Presión en
hPa
        "battery_level": round(random.uniform(10, 100), 2), # Nivel de
batería en %
        "wind_speed": round(random.uniform(0, 20), 2), # Velocidad del
viento en m/s
        "timestamp": int(time.time())
    }

producer = KafkaProducer(bootstrap_servers=['localhost:9092'],
                        value_serializer=lambda x: json.dumps(x).encode('utf-
8'))

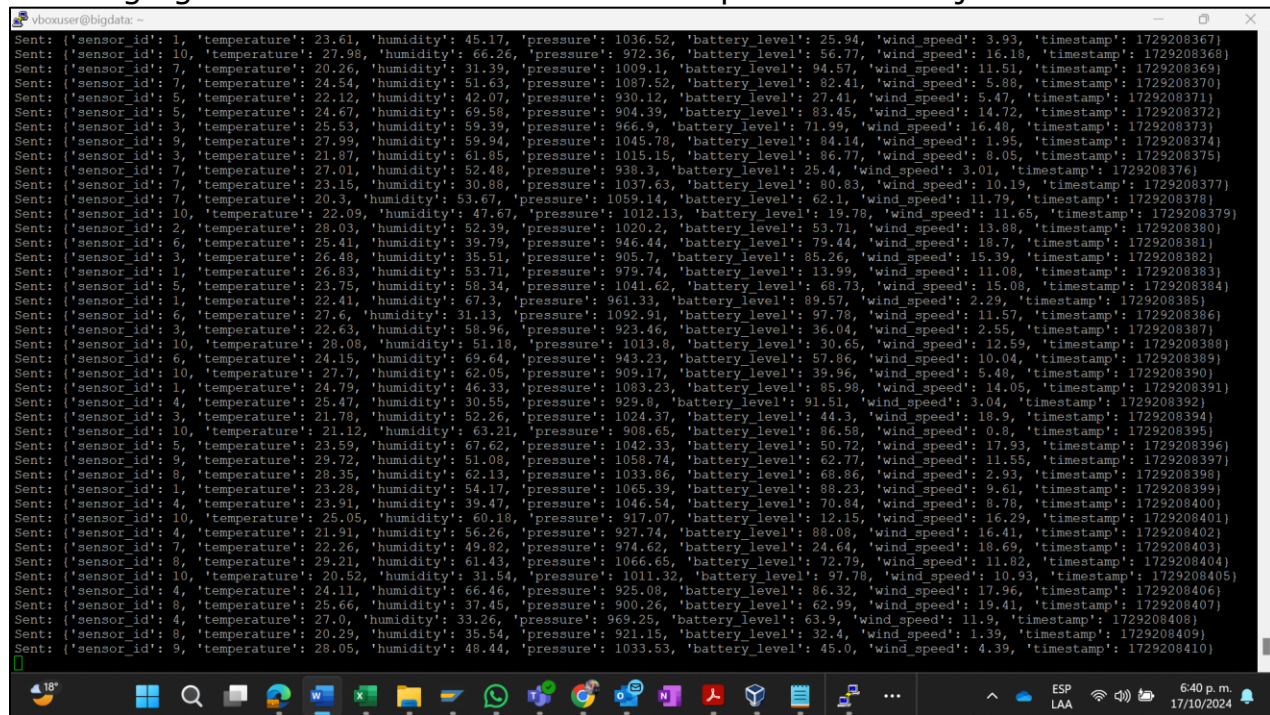
while True:
    sensor_data = generate_sensor_data()
    producer.send('sensor_data', value=sensor_data)
    print(f"Sent: {sensor_data}")
    time.sleep(1)
```

Una vez pegado el código oprimir Ctrl+O - enter y finalmente Ctrl+X para salir del archivo.

Ejecutar el archivo con el siguiente comando:

```
python3 kafka_producer.py
```

Este código generará datos automáticamente para nuestro ejercicio



```
vboxuser@bigdata: ~  
Sent: {'sensor_id': 1, 'temperature': 23.61, 'humidity': 45.17, 'pressure': 1036.52, 'battery_level': 25.94, 'wind_speed': 3.93, 'timestamp': 1729208367}  
Sent: {'sensor_id': 10, 'temperature': 27.98, 'humidity': 66.26, 'pressure': 972.36, 'battery_level': 56.77, 'wind_speed': 16.18, 'timestamp': 1729208368}  
Sent: {'sensor_id': 7, 'temperature': 20.26, 'humidity': 31.39, 'pressure': 1009.1, 'battery_level': 94.57, 'wind_speed': 11.51, 'timestamp': 1729208369}  
Sent: {'sensor_id': 7, 'temperature': 24.54, 'humidity': 51.63, 'pressure': 1087.52, 'battery_level': 82.41, 'wind_speed': 5.88, 'timestamp': 1729208370}  
Sent: {'sensor_id': 5, 'temperature': 22.12, 'humidity': 42.07, 'pressure': 930.12, 'battery_level': 27.41, 'wind_speed': 5.47, 'timestamp': 1729208371}  
Sent: {'sensor_id': 5, 'temperature': 24.67, 'humidity': 69.58, 'pressure': 904.39, 'battery_level': 83.45, 'wind_speed': 14.72, 'timestamp': 1729208372}  
Sent: {'sensor_id': 3, 'temperature': 25.53, 'humidity': 59.39, 'pressure': 966.9, 'battery_level': 71.99, 'wind_speed': 16.48, 'timestamp': 1729208373}  
Sent: {'sensor_id': 9, 'temperature': 27.99, 'humidity': 59.94, 'pressure': 1045.78, 'battery_level': 84.14, 'wind_speed': 1.95, 'timestamp': 1729208374}  
Sent: {'sensor_id': 3, 'temperature': 21.87, 'humidity': 61.85, 'pressure': 1015.15, 'battery_level': 86.77, 'wind_speed': 8.05, 'timestamp': 1729208375}  
Sent: {'sensor_id': 7, 'temperature': 27.01, 'humidity': 52.48, 'pressure': 938.3, 'battery_level': 25.4, 'wind_speed': 3.01, 'timestamp': 1729208376}  
Sent: {'sensor_id': 7, 'temperature': 23.15, 'humidity': 30.88, 'pressure': 1037.63, 'battery_level': 80.83, 'wind_speed': 10.19, 'timestamp': 1729208377}  
Sent: {'sensor_id': 7, 'temperature': 20.3, 'humidity': 53.67, 'pressure': 1059.14, 'battery_level': 62.1, 'wind_speed': 11.79, 'timestamp': 1729208378}  
Sent: {'sensor_id': 10, 'temperature': 22.09, 'humidity': 47.67, 'pressure': 1012.13, 'battery_level': 19.78, 'wind_speed': 11.65, 'timestamp': 1729208379}  
Sent: {'sensor_id': 2, 'temperature': 28.03, 'humidity': 52.39, 'pressure': 1020.2, 'battery_level': 53.71, 'wind_speed': 13.88, 'timestamp': 1729208380}  
Sent: {'sensor_id': 6, 'temperature': 25.41, 'humidity': 39.79, 'pressure': 946.44, 'battery_level': 79.44, 'wind_speed': 18.7, 'timestamp': 1729208381}  
Sent: {'sensor_id': 3, 'temperature': 26.48, 'humidity': 35.51, 'pressure': 905.7, 'battery_level': 85.26, 'wind_speed': 15.39, 'timestamp': 1729208382}  
Sent: {'sensor_id': 1, 'temperature': 26.83, 'humidity': 53.71, 'pressure': 979.74, 'battery_level': 13.99, 'wind_speed': 11.08, 'timestamp': 1729208383}  
Sent: {'sensor_id': 5, 'temperature': 23.75, 'humidity': 58.34, 'pressure': 1041.62, 'battery_level': 68.73, 'wind_speed': 15.08, 'timestamp': 1729208384}  
Sent: {'sensor_id': 1, 'temperature': 22.41, 'humidity': 67.3, 'pressure': 961.33, 'battery_level': 89.57, 'wind_speed': 2.29, 'timestamp': 1729208385}  
Sent: {'sensor_id': 6, 'temperature': 27.6, 'humidity': 31.13, 'pressure': 1092.91, 'battery_level': 97.78, 'wind_speed': 11.57, 'timestamp': 1729208386}  
Sent: {'sensor_id': 3, 'temperature': 22.63, 'humidity': 58.96, 'pressure': 923.46, 'battery_level': 36.04, 'wind_speed': 2.55, 'timestamp': 1729208387}  
Sent: {'sensor_id': 10, 'temperature': 28.08, 'humidity': 51.18, 'pressure': 1013.8, 'battery_level': 30.65, 'wind_speed': 12.59, 'timestamp': 1729208388}  
Sent: {'sensor_id': 6, 'temperature': 24.15, 'humidity': 69.64, 'pressure': 943.23, 'battery_level': 57.86, 'wind_speed': 10.04, 'timestamp': 1729208389}  
Sent: {'sensor_id': 10, 'temperature': 27.7, 'humidity': 62.05, 'pressure': 909.17, 'battery_level': 39.96, 'wind_speed': 5.48, 'timestamp': 1729208390}  
Sent: {'sensor_id': 1, 'temperature': 24.79, 'humidity': 46.33, 'pressure': 1083.23, 'battery_level': 85.98, 'wind_speed': 14.05, 'timestamp': 1729208391}  
Sent: {'sensor_id': 4, 'temperature': 25.47, 'humidity': 30.55, 'pressure': 929.8, 'battery_level': 91.51, 'wind_speed': 3.04, 'timestamp': 1729208392}  
Sent: {'sensor_id': 3, 'temperature': 21.78, 'humidity': 52.26, 'pressure': 1024.37, 'battery_level': 44.3, 'wind_speed': 18.9, 'timestamp': 1729208394}  
Sent: {'sensor_id': 10, 'temperature': 21.12, 'humidity': 63.21, 'pressure': 908.65, 'battery_level': 86.58, 'wind_speed': 0.8, 'timestamp': 1729208395}  
Sent: {'sensor_id': 5, 'temperature': 23.59, 'humidity': 67.62, 'pressure': 1042.33, 'battery_level': 50.72, 'wind_speed': 17.93, 'timestamp': 1729208396}  
Sent: {'sensor_id': 9, 'temperature': 29.72, 'humidity': 51.08, 'pressure': 1058.74, 'battery_level': 62.77, 'wind_speed': 11.55, 'timestamp': 1729208397}  
Sent: {'sensor_id': 8, 'temperature': 28.35, 'humidity': 62.13, 'pressure': 1033.86, 'battery_level': 68.86, 'wind_speed': 2.93, 'timestamp': 1729208398}  
Sent: {'sensor_id': 1, 'temperature': 23.28, 'humidity': 54.17, 'pressure': 1065.39, 'battery_level': 88.23, 'wind_speed': 9.61, 'timestamp': 1729208399}  
Sent: {'sensor_id': 4, 'temperature': 23.91, 'humidity': 39.47, 'pressure': 1046.54, 'battery_level': 70.84, 'wind_speed': 8.78, 'timestamp': 1729208400}  
Sent: {'sensor_id': 10, 'temperature': 25.05, 'humidity': 60.18, 'pressure': 917.07, 'battery_level': 12.15, 'wind_speed': 16.29, 'timestamp': 1729208401}  
Sent: {'sensor_id': 4, 'temperature': 21.91, 'humidity': 56.26, 'pressure': 927.74, 'battery_level': 88.08, 'wind_speed': 16.41, 'timestamp': 1729208402}  
Sent: {'sensor_id': 7, 'temperature': 22.26, 'humidity': 49.82, 'pressure': 974.62, 'battery_level': 24.64, 'wind_speed': 18.69, 'timestamp': 1729208403}  
Sent: {'sensor_id': 8, 'temperature': 29.21, 'humidity': 61.43, 'pressure': 1066.65, 'battery_level': 72.79, 'wind_speed': 11.82, 'timestamp': 1729208404}  
Sent: {'sensor_id': 10, 'temperature': 20.52, 'humidity': 31.54, 'pressure': 1011.32, 'battery_level': 97.78, 'wind_speed': 10.93, 'timestamp': 1729208405}  
Sent: {'sensor_id': 4, 'temperature': 24.11, 'humidity': 66.46, 'pressure': 925.08, 'battery_level': 86.32, 'wind_speed': 17.96, 'timestamp': 1729208406}  
Sent: {'sensor_id': 8, 'temperature': 25.66, 'humidity': 37.45, 'pressure': 900.26, 'battery_level': 62.99, 'wind_speed': 19.41, 'timestamp': 1729208407}  
Sent: {'sensor_id': 4, 'temperature': 27.0, 'humidity': 33.26, 'pressure': 969.25, 'battery_level': 63.9, 'wind_speed': 11.9, 'timestamp': 1729208408}  
Sent: {'sensor_id': 8, 'temperature': 20.29, 'humidity': 35.54, 'pressure': 921.15, 'battery_level': 32.4, 'wind_speed': 1.39, 'timestamp': 1729208409}  
Sent: {'sensor_id': 9, 'temperature': 28.05, 'humidity': 48.44, 'pressure': 1033.53, 'battery_level': 45.0, 'wind_speed': 4.39, 'timestamp': 1729208410}
```

Ejecutamos una nueva sesión por putty sin cerrar la actual donde se está ejecutando el archivo producer e ingresamos con el usuario:

Usuario: vboxuser

Contraseña: bigdata

Ahora, crearemos un consumidor(consumer) utilizando Spark Streaming para procesar los datos en tiempo real. Crea un archivo llamado `spark_streaming_consumer.py` con el siguiente comando:

```
nano spark_streaming_consumer.py
```

Pegamos el código que se encuentra en este repositorio de GitHub con el nombre `spark_streaming_consumer.py`:


Procesamiento_de_datos_con_Apache_Spark
Public
Pin
Unwatch 2

main
1 Branch
Tags
Go to file
Add file
Code


w-nosliw
kafka_producer.py
f960716 · 3 hours ago
8 Commits

| | | |
|--|-----------------------------|-------------|
|  kafka_producer.py | kafka_producer.py | yesterday |
|  spark_streaming_consumer.py | spark_streaming_consumer.py | 7 hours ago |
|  tarea3.py | tarea3.py | 3 hours ago |

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json, col, window
from pyspark.sql.types import StructType, StructField, IntegerType,
FloatType, TimestampType
```

```
# Configura el nivel de log a WARN para reducir los mensajes INFO
spark=SparkSession.builder\
    .appName("KafkaSparkStreaming")\
    .getOrCreate()
```

```
spark.sparkContext.setLogLevel("WARN")
```

```
# Definir el esquema para los datos de entrada
schema = StructType([
    StructField("sensor_id", IntegerType()),
    StructField("temperature", FloatType()),
    StructField("humidity", FloatType()),
    StructField("pressure", FloatType()),      # Nuevo campo de presión
    StructField("battery_level", FloatType()), # Nuevo campo de nivel de
batería
    StructField("wind_speed", FloatType()),    # Nuevo campo de
velocidad del viento
    StructField("timestamp", TimestampType())
])
```

```
# Crear una sesión de Spark
spark = SparkSession.builder\
    .appName("SensorDataAnalysis")\
    .getOrCreate()
```

```
# Configurar el lector de streaming para leer desde Kafka
df = spark\
```



```

.readStream\
.format("kafka")\
.option("kafka.bootstrap.servers", "localhost:9092")\
.option("subscribe", "sensor_data")\
.load()

# Parsear los datos JSON
parsed_df = df.select(from_json(col("value").cast("string"),
schema).alias("data")).select("data.*")

# Calcular estadísticas por ventana de tiempo (promedios por sensor y
por minuto)
windowed_stats=parsed_df\
.groupBy(window(col("timestamp"), "1 minute"), "sensor_id")\
.agg({
    "temperature":"avg",
    "humidity":"avg",
    "pressure":"avg",
    "battery_level":"avg",
    "wind_speed":"avg"
})

# Escribir los resultados en la consola
query=windowed_stats\
.writeStream\
.outputMode("complete")\
.format("console")\
.start()

query.awaitTermination()

```

Una vez pegado el código oprimir Ctrl+O - enter y finalmente Ctrl+X para salir del archivo.

Ejecutar el archivo con el siguiente comando:

```

spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.5.3
spark_streaming_consumer.py

```

Una vez ejecutado el código, empezará a recibir los datos proporcionados por el producer, los analiza y finalmente nos muestra los datos procesados el lotes de tiempo:


```
vboxuser@bigdata: ~  
[2024-10-18 15:52:...] 2|1050.4600219726562|27.145000457763672| 55.89999961853027| 7.150000095367432|30.934999465942383|  
[2024-10-18 15:53:...] 5| 999.8920166015625|26.711999893188477| 45.52800064086914| 8.285999774932861| 38.10999984741211|  
[2024-10-18 15:53:...] 2| 967.4014282226562|25.155714307512557| 44.27428599766323|10.694285801478795| 41.72000026702881|  
[2024-10-18 15:53:...] 10|1000.2188991970486| 23.71444426642524| 43.68888960944282| 7.816666735543145| 51.2322245110406|  
[2024-10-18 15:56:...] 1| 972.5950012207031| 25.3050003051758156| 755001068115234|7.6300002336502075| 34.20500087738037|  
[2024-10-18 15:55:...] 4| 987.2624893188477|24.364999771118164| 56.77375078201294| 7.230000019073486| 52.56499981880188|  
[2024-10-18 15:54:...] 10|1021.4600118001301|25.175000190724863| 50.0716673531217| 10.3766654586792|52.64666644685875|  
[2024-10-18 15:55:...] 7| 949.714963378906|26.03999961853027| 47.85500144358496| 10.0600004196167| 50.6299991607666|  
[2024-10-18 15:56:...] 7| 996.179926757812| 24.38499925720752|57.060001373291016|17.010000228881836|63.540000915527344|  
[2024-10-18 15:55:...] 3|1021.7799987792969| 23.0625|48.942501068115234| 6.625| 38.72250032424927|  
[2024-10-18 15:54:...] 3| 980.5650024414062|26.824999809265137| 44.5275011062622|11.300000041723251| 66.54250049591064|  
only showing top 20 rows  
Batch: 2  
window|sensor_id| avg (pressure)| avg (temperature)| avg (humidity)| avg (wind_speed)| avg (battery_level)|  
[2024-10-18 15:56:...] 4|1009.1640014648438|25.655999755859376| 49.01600112915039|11.854000139236451| 56.48200035095215|  
[2024-10-18 15:57:...] 4|1055.8440185546874| 23.8439998626709|48.934000396728514|11.519999980926514| 53.65000057220459|  
[2024-10-18 15:54:...] 2| 992.2149963378906|22.010000228881836| 50.71000099182129| 8.245000123977661| 19.31999969482422|  
[2024-10-18 15:52:...] 7| 960.614990234375| 21.99000072479248| 47.06999969482422| 5.460000157356262| 50.26499938964844|  
[2024-10-18 15:56:...] 6|1020.3299865722656|25.295000394185383| 52.65833409627279|10.32499998079071|53.710000356038414|  
[2024-10-18 15:56:...] 5|1011.5762557983398| 24.93750023841858|53.916250228881836| 6.597500070929527| 61.43000078201294|  
[2024-10-18 15:54:...] 5|1003.3940185546875| 25.66800030517581| 55.04300041198731|12.521000003814697| 54.01599979400635|  
[2024-10-18 15:52:...] 10| 974.5866495768229|24.676666895548504| 59.0533332824707|13.096666733423868| 38.96999899546305|  
[2024-10-18 15:57:...] 5| 926.3049926757812|27.269999504089355| 54.15999984741211| 5.519999980926514| 65.52499961853027|  
[2024-10-18 15:53:...] 9| 1004.393310546875|27.156667073567707| 46.0366667011719|13.54666640083305| 68.7966670989902|  
[2024-10-18 15:55:...] 8| 994.096357865767|24.84544302701082| 49.90818127718839| 7.388181897726926| 52.91909070448442|  
[2024-10-18 15:52:...] 2|1050.4600219726562|27.145000457763672| 55.89999961853027| 7.150000095367432|30.934999465942383|  
[2024-10-18 15:58:...] 8|1034.5399780273438|22.932499885559082|44.750000953674316| 9.885000050067902| 47.84249973297119|  
[2024-10-18 15:53:...] 5| 999.8920166015625|26.711999893188477| 45.52800064086914| 8.285999774932861| 38.10999984741211|  
[2024-10-18 15:53:...] 2| 967.4014282226562|25.155714307512557| 44.27428599766323|10.694285801478795| 41.72000026702881|  
[2024-10-18 15:53:...] 10|1000.2188991970486| 23.71444426642524| 43.68888960944282| 7.816666735543145| 51.2322245110406|  
[2024-10-18 15:56:...] 1|1014.1324920654297|23.820000171661377| 52.34250068664551|10.477500140666962| 36.72250032424927|  
[2024-10-18 15:55:...] 4| 987.2624893188477|24.364999771118164| 56.77375078201294| 7.230000019073486| 52.56499981880188|  
[2024-10-18 15:58:...] 7| 972.8249816894531| 23.31166648864746| 55.04666646321615|10.391666531562805| 49.39000002543131|  
[2024-10-18 15:57:...] 9| 964.2099975585937|25.707999801635744| 42.741999816894534| 9.017999839782714| 57.70799942016602|  
only showing top 20 rows
```

Podemos observar ambos códigos trabajando al mismo tiempo: el generador y el consumidor:

```
vboxuser@bigdata: ~  
Sent: {'sensor_id': 4, 'temperature': 28.66, 'humidity': 66.47, 'pressure': 939.29, 'battery_level': 81.78, 'wind_speed': 12.47, 'timestamp': 1729267688}  
Sent: {'sensor_id': 7, 'temperature': 22.0, 'humidity': 46.52, 'pressure': 997.84, 'battery_level': 83.35, 'wind_speed': 13.46, 'timestamp': 1729267689}  
Sent: {'sensor_id': 9, 'temperature': 29.94, 'humidity': 60.17, 'pressure': 919.18, 'battery_level': 67.2, 'wind_speed': 6.06, 'timestamp': 1729267690}  
Sent: {'sensor_id': 4, 'temperature': 27.83, 'humidity': 44.04, 'pressure': 906.65, 'battery_level': 27.72, 'wind_speed': 6.65, 'timestamp': 1729267691}  
Sent: {'sensor_id': 2, 'temperature': 22.79, 'humidity': 65.77, 'pressure': 1029.41, 'battery_level': 56.7, 'wind_speed': 13.81, 'timestamp': 1729267692}  
Sent: {'sensor_id': 4, 'temperature': 23.19, 'humidity': 54.75, 'pressure': 1038.77, 'battery_level': 22.05, 'wind_speed': 6.8, 'timestamp': 1729267693}  
Sent: {'sensor_id': 2, 'temperature': 27.1, 'humidity': 49.86, 'pressure': 1093.67, 'battery_level': 65.74, 'wind_speed': 5.84, 'timestamp': 1729267694}  
Sent: {'sensor_id': 7, 'temperature': 22.51, 'humidity': 64.21, 'pressure': 985.37, 'battery_level': 79.09, 'wind_speed': 5.02, 'timestamp': 1729267695}  
Sent: {'sensor_id': 3, 'temperature': 27.44, 'humidity': 49.09, 'pressure': 1018.38, 'battery_level': 53.88, 'wind_speed': 17.0, 'timestamp': 1729267696}  
Sent: {'sensor_id': 9, 'temperature': 24.69, 'humidity': 51.64, 'pressure': 979.19, 'battery_level': 23.59, 'wind_speed': 17.5, 'timestamp': 1729267698}  
Sent: {'sensor_id': 2, 'temperature': 24.01, 'humidity': 39.04, 'pressure': 1015.12, 'battery_level': 37.97, 'wind_speed': 11.59, 'timestamp': 1729267699}  
Sent: {'sensor_id': 5, 'temperature': 28.81, 'humidity': 42.76, 'pressure': 973.78, 'battery_level': 38.4, 'wind_speed': 7.25, 'timestamp': 1729267700}  
Sent: {'sensor_id': 4, 'temperature': 27.43, 'humidity': 31.52, 'pressure': 969.59, 'battery_level': 65.86, 'wind_speed': 2.87, 'timestamp': 1729267701}  
Sent: {'sensor_id': 3, 'temperature': 27.33, 'humidity': 49.25, 'pressure': 1025.16, 'battery_level': 92.43, 'wind_speed': 0.74, 'timestamp': 1729267702}  
Sent: {'sensor_id': 8, 'temperature': 26.98, 'humidity': 69.04, 'pressure': 973.12, 'battery_level': 86.81, 'wind_speed': 1.54, 'timestamp': 1729267703}  
Sent: {'sensor_id': 3, 'temperature': 27.75, 'humidity': 48.25, 'pressure': 960.96, 'battery_level': 37.12, 'wind_speed': 18.47, 'timestamp': 1729267704}  
Sent: {'sensor_id': 3, 'temperature': 28.76, 'humidity': 44.19, 'pressure': 972.26, 'battery_level': 35.14, 'wind_speed': 15.99, 'timestamp': 1729267705}  
Sent: {'sensor_id': 3, 'temperature': 26.42, 'humidity': 61.44, 'pressure': 972.95, 'battery_level': 35.14, 'wind_speed': 19.89, 'timestamp': 1729267706}  
Sent: {'sensor_id': 9, 'temperature': 26.45, 'humidity': 47.53, 'pressure': 1050.02, 'battery_level': 34.25, 'wind_speed': 8.44, 'timestamp': 1729267707}  
Sent: {'sensor_id': 7, 'temperature': 24.42, 'humidity': 65.13, 'pressure': 903.18, 'battery_level': 86.6, 'wind_speed': 10.4, 'timestamp': 1729267708}  
Sent: {'sensor_id': 6, 'temperature': 29.91, 'humidity': 56.78, 'pressure': 1078.29, 'battery_level': 21.32, 'wind_speed': 17.81, 'timestamp': 1729267709}
```

```
vboxuser@bigdata: ~  
[2024-10-18 15:55:...] 8| 994.096357865767|24.84544302701082| 49.908|  
[2024-10-18 15:52:...] 2|1050.4600219726562|27.145000457763672| 55.899|  
[2024-10-18 15:58:...] 8|1034.5399780273438|22.932499885559082|44.7500|  
[2024-10-18 15:53:...] 5| 999.8920166015625|26.711999893188477| 45.528|  
[2024-10-18 15:53:...] 2| 967.4014282226562|25.155714307512557| 44.274|  
[2024-10-18 15:53:...] 10|1000.2188991970486| 23.71444426642524| 43.688|  
[2024-10-18 15:56:...] 1|1014.1324920654297|23.820000171661377| 52.342|  
[2024-10-18 15:55:...] 4| 987.2624893188477|24.364999771118164| 56.773|  
[2024-10-18 15:58:...] 7| 972.8249816894531| 23.31166648864746| 55.046|  
[2024-10-18 15:57:...] 9| 964.2099975585937|25.707999801635744| 42.741|  
only showing top 20 rows  
Batch: 3  
window|sensor_id| avg (pressure)| avg (temperature)| av  
[2024-10-18 15:56:...] 4|1009.1640014648438|25.655999755859376| 49.016|  
[2024-10-18 16:01:...] 3| 980.7150065104166|23.208333333333332| 42.650|  
[2024-10-18 15:57:...] 4|1055.8440185546874| 23.8439998626709|48.9340|  
[2024-10-18 16:00:...] 8| 1014.875714983259|24.85142844063895| 46.441|  
[2024-10-18 16:02:...] 1|1027.7371390206474|25.375714438302175| 48.717|  
[2024-10-18 15:54:...] 2| 992.2149963378906|22.010000228881836| 50.710|  
[2024-10-18 16:02:...] 3|1033.6174926757812|23.789999961853027| 51.904|  
[2024-10-18 15:52:...] 7| 960.614990234375| 21.99000072479248| 47.069|  
[2024-10-18 15:56:...] 6|1020.3299865722656|25.295000394185383| 52.658|  
[2024-10-18 15:56:...] 5|1011.5762557983398| 24.93750023841858|53.9162|  
[2024-10-18 15:54:...] 5|1003.3940185546875| 25.66800030517581| 55.043|  
[2024-10-18 15:52:...] 10| 974.5866495768229|24.676666895548504| 59.053|  
[2024-10-18 15:57:...] 5| 926.3049926757812|27.269999504089355| 54.159|  
[2024-10-18 16:02:...] 5|1040.8899841308594|24.265000104904175| 47.972|  
[2024-10-18 15:53:...] 9| 1004.393310546875|27.156667073567707| 46.036|  
[2024-10-18 15:55:...] 8| 994.096357865767|24.84544302701082| 49.908|  
[2024-10-18 16:02:...] 10|1014.1866861979166|24.729999542236328| 54.873|  
[2024-10-18 15:52:...] 2|1050.4600219726562|27.145000457763672| 55.899|  
[2024-10-18 15:58:...] 8|1034.5399780273438|22.932499885559082|44.7500|  
[2024-10-18 15:53:...] 5| 999.8920166015625|26.711999893188477| 45.528|  
only showing top 20 rows
```