Air Quality Monitoring System

Using Apache Kafka

By:

Vasumathi R

RA2412044015107

AIDS-B Section

SRMIST, Chennai.

AIM:

To design and implement a real-time air quality monitoring system that continuously collects, processes, and streams air quality data using Apache Kafka to enable actionable insights for environmental monitoring and public health awareness.

INTRODUCTION:

The Air Quality Monitoring System leverages Apache Kafka to collect, process, and stream real-time pollution data. This scalable system provides actionable insights through dashboards and alert mechanisms, empowering timely interventions. By addressing health and sustainability challenges, it highlights technology's role in advancing environmental monitoring and public awareness.

PROCEDURE:

Step 1: Install Apache Kafka in WSL

Step 2: Create Kafka Topic

Step 3: Install Required Libraries

Step 4: Get Open Air Quality Map API Key

Step 5: Create the Kafka Producer

Step 6: Create the Kafka Consumer

Step 7: Create a Real-Time Dashboard Using Streamlit

EXECUTION and SCREENSHOTS

Step 1: Install Apache Kafka in WSL

If you haven’t already installed Kafka, follow these steps:

1.1 Download and Extract Kafka

wget https://archive.apache.org/dist/kafka/3.4.1/kafka_2.13-3.4.1.tgz

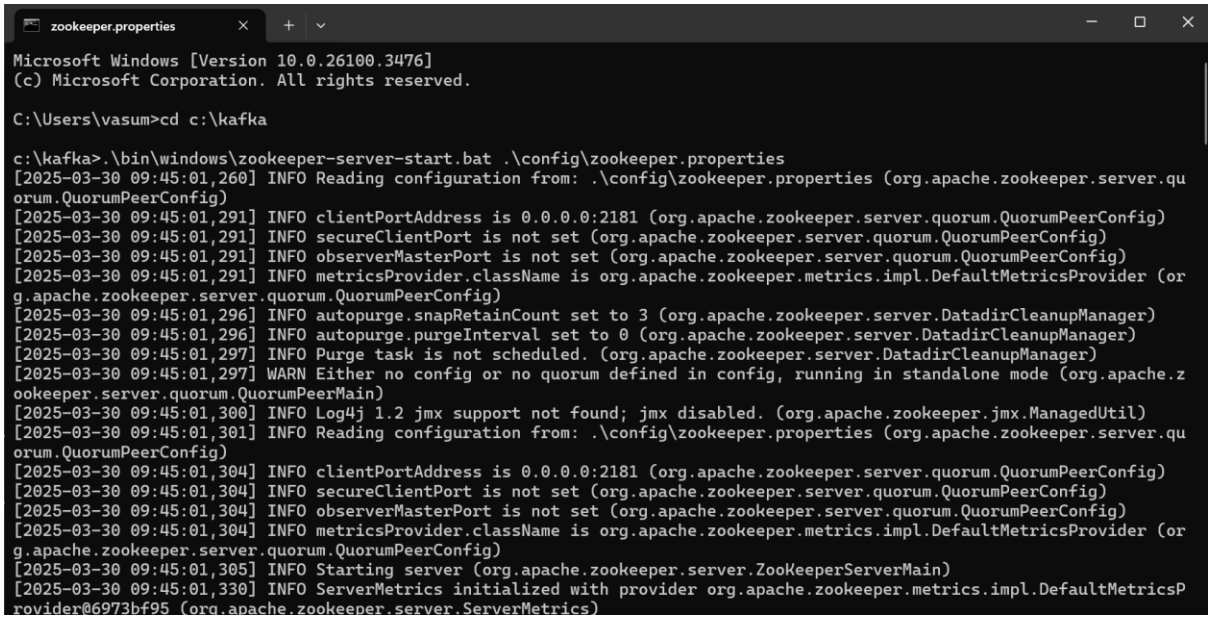
tar -xvzf kafka_2.13-3.4.1.tgz (Extract)

mv kafka_2.13-3.4.1 kafka (Store)

1.2 Start Zookeeper

cd kafka

bin/zookeeper-server-start.sh config/zookeeper.properties



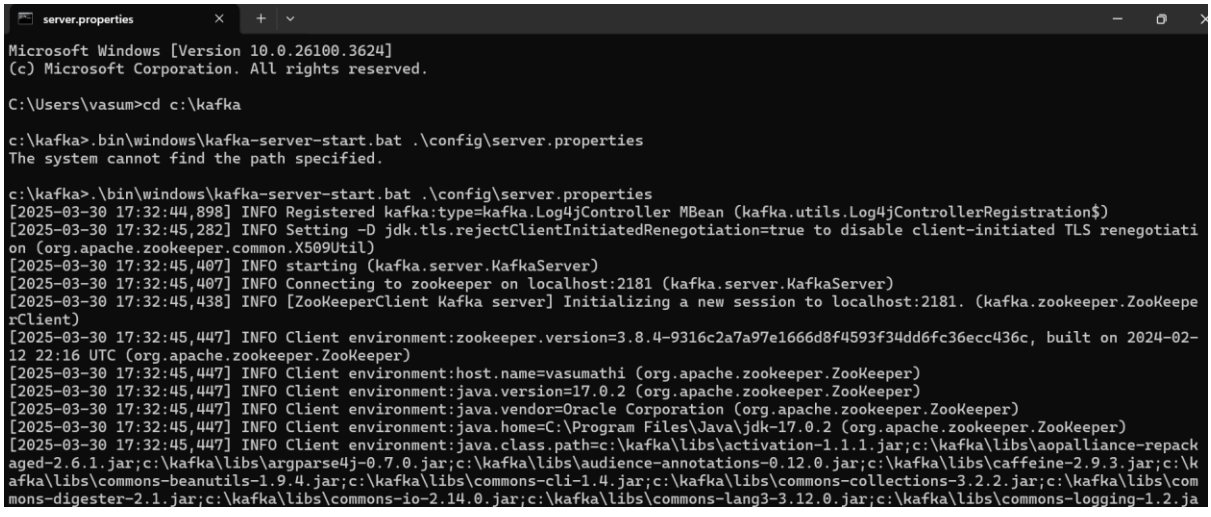
1.3 Start Kafka Server

In a new terminal:

cd kafka

bin/kafka-server-start.sh config/server.properties

Server Running



Step 2: Create Kafka Topic

```
topic
Microsoft Windows [Version 10.0.26100.3624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vasum>cd c:\kafka

c:\kafka>bin\windows\kafka-topics.bat --create --topic air-quality-data --bootstrap-server localhost:9092 --partitions 3 --replicatio
n-factor 1

created topic air-quality-data.
```

Step 3: Install Required Libraries

Virtual Environment Creation

```
VE Creation
Microsoft Windows [Version 10.0.26100.3624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\vasum>python -m venv myenv

C:\Users\vasum>
C:\Users\vasum>myenv\Scripts\activate

(myenv) C:\Users\vasum>pip install kafka-python requests pandas streamlit
Collecting kafka-python
  Obtaining dependency information for kafka-python from https://files.pythonhosted.org/packages/f7/01/e5a16a51093febcee7cd833a6408eb
2bcf44f3282cf96a4e731c036454ff/kafka_python-2.1.4-py2.py3-none-any.whl.metadata
  Downloading kafka_python-2.1.4-py2.py3-none-any.whl.metadata (9.1 kB)
Collecting requests
  Obtaining dependency information for requests from https://files.pythonhosted.org/packages/f9/9b/335f9764261e915ed497fcdeb11df5dfd6
f7bf257d4a6a2a686d80da4d54/requests-2.32.3-py3-none-any.whl.metadata
  Downloading requests-2.32.3-py3-none-any.whl.metadata (4.6 kB)
Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/ed/8c/87ddf1fcb55d11f9f847e3c69bb1c6f8e46e
2f40ab1a2d2abadb2401b007/pandas-2.2.3-cp311-cp311-win_amd64.whl.metadata
  Downloading pandas-2.2.3-cp311-cp311-win_amd64.whl.metadata (19 kB)
Collecting streamlit
  Obtaining dependency information for streamlit from https://files.pythonhosted.org/packages/46/9e/e51e34f504940da00145795b9e8be9c12
9704708b071f672f3626a37d842/streamlit-1.44.0-py3-none-any.whl.metadata
  Downloading streamlit-1.44.0-py3-none-any.whl.metadata (8.9 kB)
```

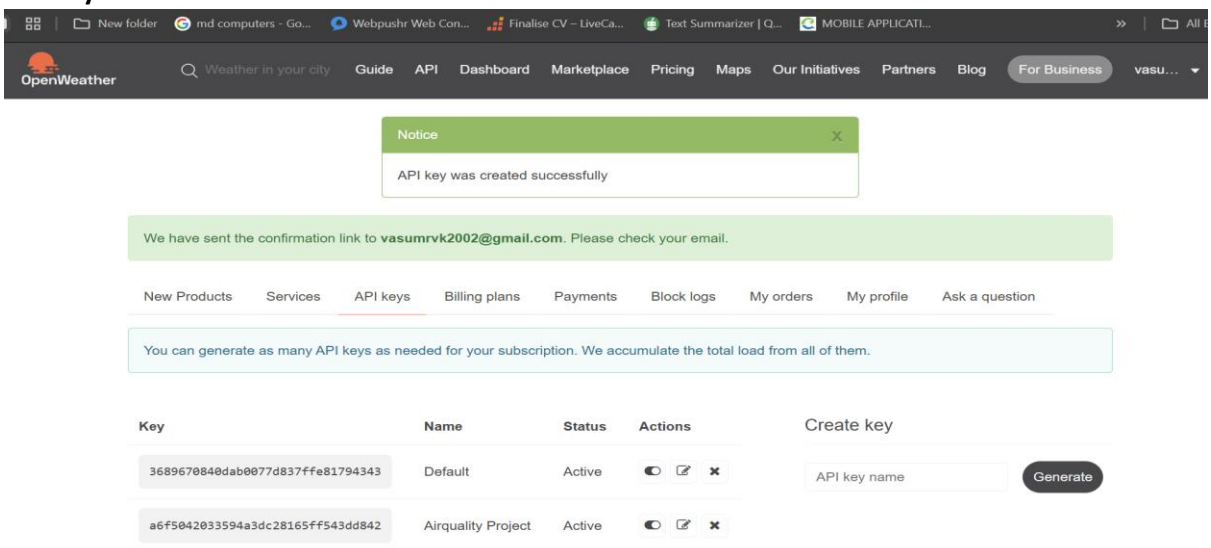
Step 4: Get Open Air Quality Map API Key

1. Go to https://home.openweathermap.org/users/sign_up

2. Sign up for a free account

3. Go to API keys → Generate a new API key

API Key Generation:

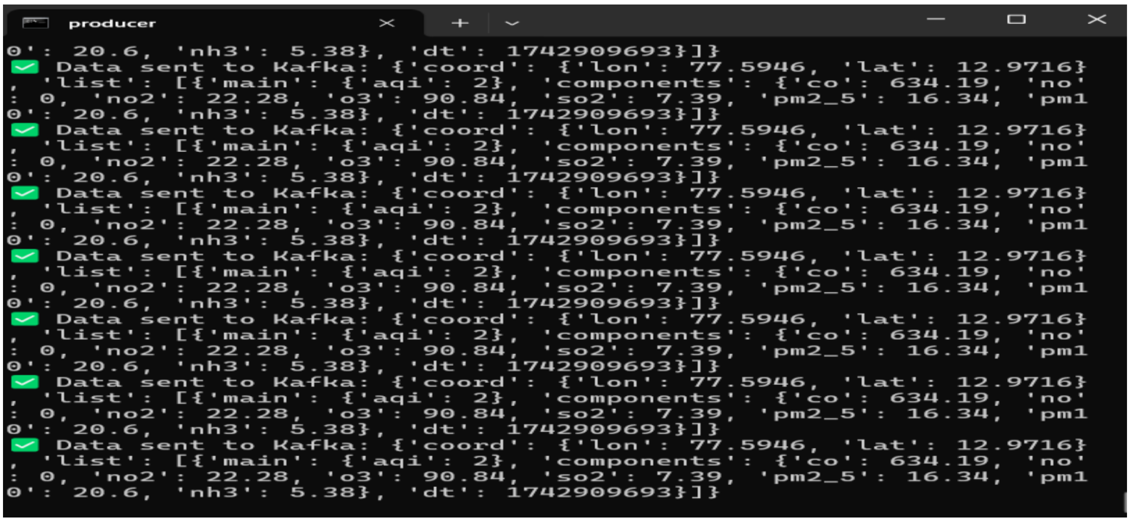


Step 5: Create the Kafka Producer

Kafka Producer: (Producer.py)

```
import json

import requests

from kafka import KafkaProducer

import time

# OpenWeatherMap API details

API_KEY = ' a6f5042033594a3dc28165ff543dd842'

CITY = 'Chennai'  # City name

LAT, LON = 13.0827, 80.2707  # Latitude and Longitude of Chennai

# API URL to fetch air quality data

URL =
f"http://api.openweathermap.org/data/2.5/air_pollution?lat=12.9716&lon=77.5946&appid={API_KE
Y}"

# Kafka Producer configuration

producer = KafkaProducer(

   bootstrap_servers='localhost:9092',  # Kafka server

   value_serializer=lambda v: json.dumps(v).encode('utf-8')  # Convert data to JSON

)

def fetch_air_quality_data():

   """Fetch real-time air quality data from OpenWeatherMap API"""

   try:

      response = requests.get(URL)

      data = response.json()

      return data

   except Exception as e:

      print(f"Error fetching data: {e}")

      return None

# Send data to Kafka at regular intervals

while True:

   air_data = fetch_air_quality_data()
```

```
if air_data:
    producer.send('air-quality-data', value=air_data)
    print("✅ Data sent to Kafka:", air_data)
time.sleep(10)  # Fetch data every 10 seconds
```

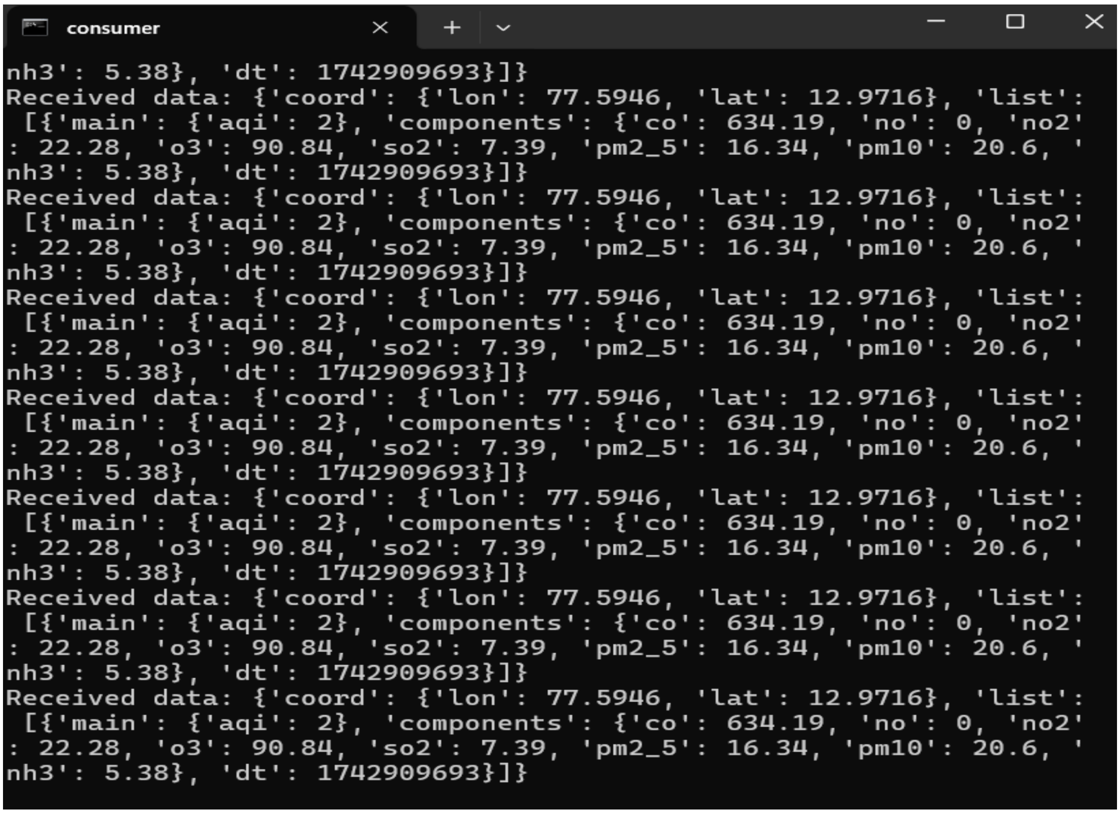


Step 6: Create the Kafka Consumer

Consumer: (consumer.py)

```
from kafka import KafkaConsumer
import json
import pandas as pd

# Corrected Kafka consumer configuration
TOPIC_NAME = 'air-quality-data'  # 👈 Updated topic name
BOOTSTRAP_SERVERS = 'localhost:9092'

# Create a Kafka consumer
consumer = KafkaConsumer(
    TOPIC_NAME,
    bootstrap_servers=BOOTSTRAP_SERVERS,
    value_deserializer=lambda x: json.loads(x.decode('utf-8')),
    auto_offset_reset='earliest',  # Read from the beginning if no offset is committed
    enable_auto_commit=True,
```

```
    group_id='air-quality-group'
)
print("✅ Listening for incoming air quality data...")
# Process incoming data
for message in consumer:
    air_data = message.value
print(f"Received data: {air_data}")
```

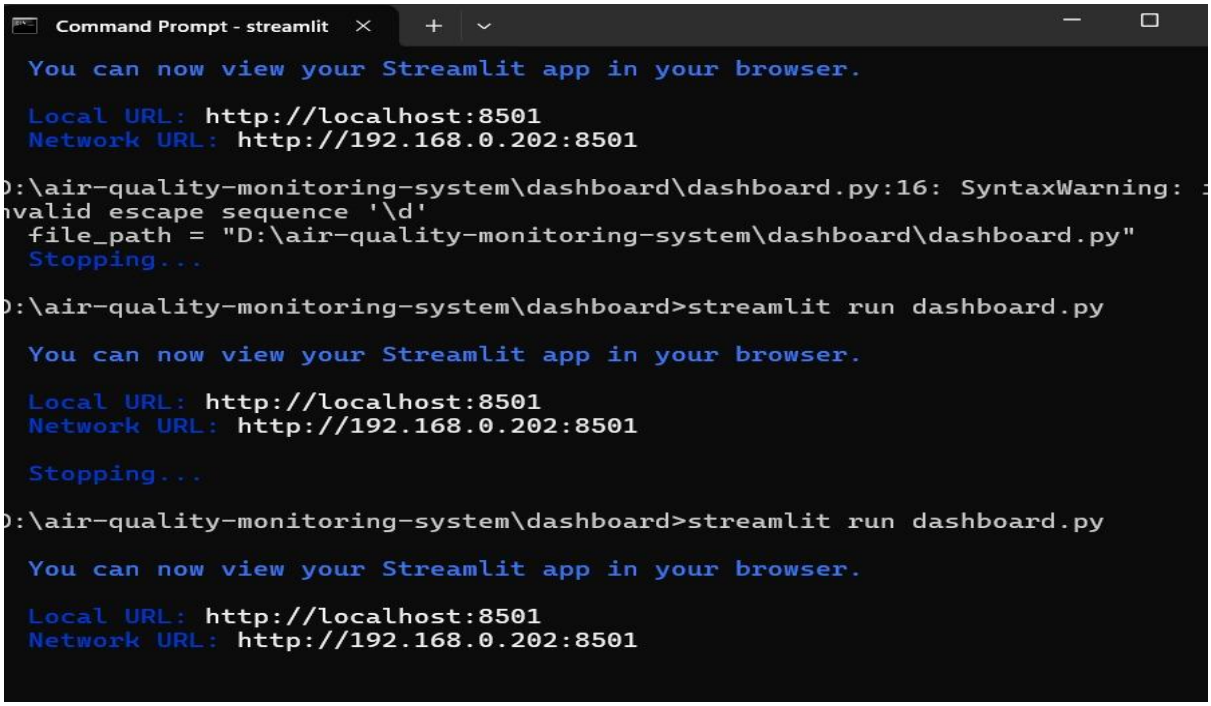


Step 7: Create a Real-Time Dashboard

Using StreamlitDashboard Creation:

```
    import streamlit as st
import pandas as pd
import plotly.express as px
# Page title
st.title(" Air Quality Monitoring Dashboard")
# Sample air quality data
data = {
  "Timestamp": ["2025-03-25 18:00", "2025-03-25 18:05", "2025-03-25 18:10"],
```

```
    "PM2.5": [15.01, 16.34, 18.56],
    "PM10": [18.56, 20.60, 22.78],
    "CO": [594.14, 634.19, 650.43],
    "NO2": [21.08, 22.28, 23.34],
    "O3": [97.28, 90.84, 88.23],
    "AQI": [2, 2, 3]
}
# Create DataFrame
df = pd.DataFrame(data)
# Display data as table
st.write("### 📊 Raw Air Quality Data")
st.dataframe(df)
# Plot AQI over time
fig = px.line(df, x="Timestamp", y="AQI", title="Air Quality Index Over Time")
st.plotly_chart(fig)
# Plot PM2.5 and PM10 levels
fig_pm = px.line(df, x="Timestamp", y=["PM2.5", "PM10"], title="PM2.5 and PM10 Levels Over Time")
st.plotly_chart(fig_pm)
st.success("✅ Dashboard Updated Successfully!")
```



CONCLUSION:

The Air Quality Monitoring System powered by Apache Kafka effectively addresses the need for real-time environmental data processing. By providing actionable insights and timely alerts, it empowers decision-makers to combat air pollution and safeguard public health. This project demonstrates the transformative impact of technology in building a sustainable and healthier future.