

Complete Data Pipeline Code

```
import time
import json
import psutil
import GPUtil
import mysql.connector
from confluent_kafka import Producer, Consumer

# Kafka broker configuration
bootstrap_servers = 'localhost:9092'
topic = 'system_stats'

# MySQL database connection
db_config = {
    'user': 'root',          # Replace with your MySQL username
    'password': 'XXXXX',    # Replace with your MySQL password
    'host': 'localhost',
    'database': 'system_stats'
}

# Create a Kafka producer
producer = Producer({'bootstrap.servers': bootstrap_servers})

def get_system_stats():
    """Gather system and GPU statistics."""
    cpu_usage = psutil.cpu_percent(interval=0.5)
    memory = psutil.virtual_memory()
    disk_usage = psutil.disk_usage('/')

    # Get GPU statistics
    gpus = GPUtil.getGPUs()
    gpu_memory_used = gpus[0].memoryUsed if gpus else 0
    gpu_memory_free = gpus[0].memoryFree if gpus else 0
    gpu_load = gpus[0].load * 100 if gpus else 0

    # Create a dictionary of the stats
    stats = {
        'time': time.strftime('%Y-%m-%d %H:%M:%S'),
        'cpu_usage': cpu_usage,
        'memory_usage': memory.percent,
        'memory_used': memory.used,
        'memory_free': memory.free,
        'disk_usage': disk_usage.percent,
        'gpu_memory_used': gpu_memory_used,
        'gpu_memory_free': gpu_memory_free,
        'gpu_load': gpu_load,
    }
    return stats

def delivery_report(err, msg):
    """Callback function to handle delivery reports."""
    if err is not None:
        print(f"Message delivery failed: {err}")
    else:
        print(f"Produced: {msg.value} to Kafka topic: {msg.topic()}")

def insert_into_db(stats):
    """Insert collected stats into the MySQL database."""
    db_connection = mysql.connector.connect(**db_config)
    db_cursor = db_connection.cursor()

    insert_query = """
    INSERT INTO performance (time, cpu_usage, memory_usage, memory_used, memory_free,
    disk_usage, gpu_memory_used, gpu_memory_free, gpu_load)
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
    """
```

```

try:
    data_to_insert = (
        stats['time'],
        stats['cpu_usage'],
        stats['memory_usage'],
        stats['memory_used'],
        stats['memory_free'],
        stats['disk_usage'],
        stats['gpu_memory_used'],
        stats['gpu_memory_free'],
        stats['gpu_load'],
    )
    db_cursor.execute(insert_query, data_to_insert)
    db_connection.commit()
    print(f"Inserted at {stats['time']} into the database.")
except mysql.connector.Error as e:
    print(f"Error inserting into database: {e}")
finally:
    db_cursor.close()
    db_connection.close()

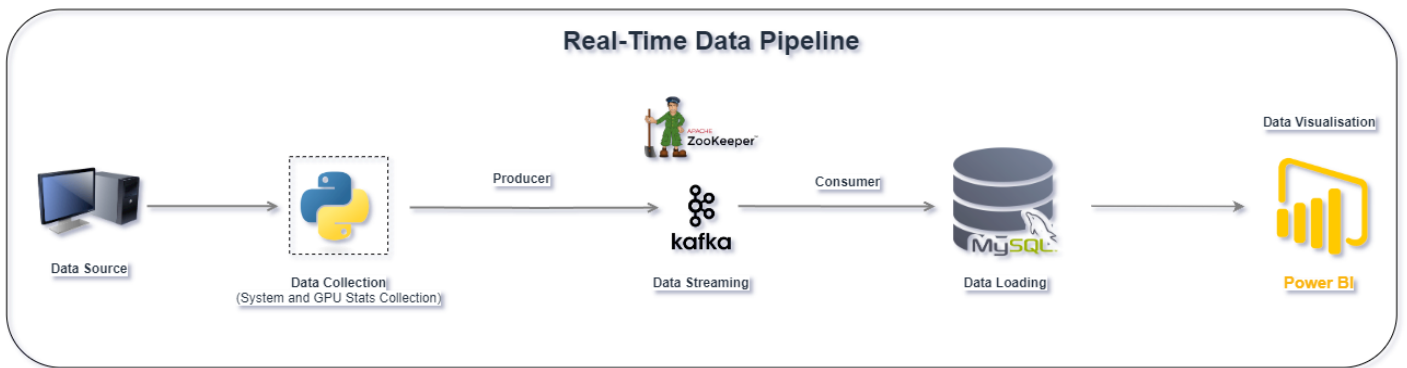
# Main loop for producer and consumer
try:
    # Start the producer
    while True:
        system_stats = get_system_stats()
        producer.produce(topic, value=json.dumps(system_stats), callback=delivery_report)
        producer.poll(1) # Serve delivery reports
        insert_into_db(system_stats) # Insert into MySQL
        time.sleep(0.5) # Send data every 5 seconds
except KeyboardInterrupt:
    print("Producer stopped.")
finally:
    producer.flush() # Wait for any outstanding messages to be delivered

```

Running the Script

1. Make sure to replace your_mysql_user and your_mysql_password with your actual MySQL credentials.
2. Ensure your Kafka broker is running and the database schema is set up as described previously.
3. Run the script:

```
python data_pipeline.py
```



This code sets up a data pipeline that continuously collects system and GPU statistics, sends them to Kafka, and stores them in a MySQL database. Here's a breakdown of the pipeline process:

1. Data Collection

- **System Statistics Gathering:**

- The code collects system statistics such as CPU usage, memory usage, disk usage, and GPU statistics every 0.5 seconds using psutil for system stats and GPUtil for GPU stats.

- **Collected Data:**

- The collected stats include CPU usage (%), memory usage (used and free memory in bytes), disk usage (%), and GPU stats like memory used, memory free, and GPU load (%).

2. Data Transmission to Kafka

- **Kafka Producer:**

- A Kafka producer sends the collected system statistics to a Kafka topic named system_stats.
- The producer encodes the data as JSON and sends it to Kafka, where it can be consumed by other services or processes.

- **Delivery Report:**

- The delivery_report function confirms whether each message is successfully sent to Kafka or if there's an error.

3. Data Storage in MySQL

- **Database Connection:**
 - The code connects to a local MySQL database named `system_stats`.
- **Data Insertion:**
 - The `insert_into_db` function inserts the system statistics into a table named `performance`. It executes an SQL INSERT query to store each set of statistics.
 - After the data is inserted, the database connection is closed to ensure resources are properly managed.

4. Continuous Execution

- **Loop:**
 - The pipeline runs in an infinite loop, where it gathers system statistics, sends them to Kafka, and stores them in MySQL every 0.5 seconds.
- **Graceful Shutdown:**
 - If the script is interrupted (e.g., using Ctrl+C), the producer flushes any remaining messages, ensuring no data is lost.