```
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': '6621326',   # 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
```

Libraries Used:

- psutil: A Python library that helps you get information about your system's resources, like CPU, memory, and disk usage.
- GPUtil: A library used to get information about your system's GPU (graphics card) usage.
- mysql.connector: A Python connector that lets you interact with a MySQL database.
- confluent_kafka: A Python client for Kafka, used to send and receive data streams (messages).

Key Steps in the Code

1. Kafka Setup

- Kafka is like a "message bus" that allows systems to send messages to one another. Here, we set it up as a producer that will send system stats as messages.

```
# Kafka broker configuration
bootstrap_servers = 'localhost:9092'  # The server where Kafka is running
topic = 'system_stats'                # The topic (like a channel) to send the data to
```

The Producer is created here:

```
producer = Producer({'bootstrap.servers': bootstrap_servers})
```

This producer will send messages (system stats) to Kafka.

2. MySQL Database Setup

MySQL is used to store the data (like CPU and memory stats) permanently so it can be accessed later.

```
db_config = {
    'user': 'root',  # Your MySQL username
    'password': '6621326',  # Your MySQL password
    'host': 'localhost',  # The MySQL server location (localhost = your computer)
    'database': 'system_stats'  # The name of the database you want to store data in
}
```

Here, the code sets up a connection to the MySQL database.

3. Collecting System Stats

```
This part of the code gathers data about your computer’s current resource usage.
cpu_usage = psutil.cpu_percent(interval=0.5)  # CPU usage as a percentage
memory = psutil.virtual_memory()  # Memory details like used, free memory
disk_usage = psutil.disk_usage('/')  # Disk usage (how full your hard drive is)

gpus = GPUtil.getGPUs()  # Get GPU info
gpu_memory_used = gpus[0].memoryUsed if gpus else 0  # If there's a GPU, get memory used
gpu_memory_free = gpus[0].memoryFree if gpus else 0  # GPU free memory
gpu_load = gpus[0].load * 100 if gpus else 0  # GPU usage as a percentage
```

This function collects details about:

- CPU usage
- Memory (RAM) usage
- Disk usage
- GPU (graphics card) usage, if available

It puts these details into a dictionary (a data structure in Python):

```
stats = {
    'time': time.strftime('%Y-%m-%d %H:%M:%S'),  # Current time
    'cpu_usage': cpu_usage,  # Collected CPU usage
    'memory_usage': memory.percent,  # Memory usage percentage
    'memory_used': memory.used,  # Amount of memory used in bytes
    'memory_free': memory.free,  # Free memory
    'disk_usage': disk_usage.percent,  # Disk usage percentage
    'gpu_memory_used': gpu_memory_used,  # GPU memory used
    'gpu_memory_free': gpu_memory_free,  # GPU memory free
    'gpu_load': gpu_load,  # GPU load percentage
}
```

4. Sending Data to Kafka

Once the stats are collected, they are sent to the Kafka broker. The producer.produce line does that:

```
producer.produce(topic, value=json.dumps(system_stats), callback=delivery_report)
producer.produce(topic, value=json.dumps(system_stats), callback=delivery_report)
```

- topic: Specifies the Kafka topic where the message (stats) is sent.
- value: The actual data (system stats) in JSON format (a commonly used data format).
- callback: A function (delivery_report) that checks if the message was successfully sent□ topic: Specifies the Kafka topic where the message (stats) is sent.

- value: The actual data (system stats) in JSON format (a commonly used data format).
- callback: A function (delivery_report) that checks if the message was successfully sent.

5. Inserting Data into MySQL

This function stores the collected stats in a MySQL database table called `performance`. The stats are inserted into columns like `cpu_usage`, `memory_usage`, `gpu_memory_used`, etc.

```
def insert_into_db(stats):
    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)
    """
```

The insert_query is a SQL command that adds the collected data into the performance table. Each stat is placed into a corresponding column.

6. Main Loop

The code continuously collects system stats and sends them to both Kafka and MySQL every 0.5 seconds.

```
while True:
    system_stats = get_system_stats()  # Collect the stats
    producer.produce(topic, value=json.dumps(system_stats), callback=delivery_report)  #
Send to Kafka
    producer.poll(1)  # Wait for Kafka to confirm the message is sent
    insert_into_db(system_stats)  # Insert the stats into the database
    time.sleep(0.5)  # Wait for 0.5 seconds before repeating
```

- The system stats are collected.
- The stats are sent to Kafka and stored in MySQL.
- It repeats every 0.5 seconds until you stop the program.

Conclusion:

This program is designed to **monitor system performance** (CPU, memory, GPU) and **send that data** to both Kafka and a MySQL database in real-time. Kafka can be used to stream the data to other systems or dashboards, while the MySQL database keeps the data for future analysis or reporting.

If you were explaining this to a layman:

- Think of **Kafka** as a message delivery service where you send real-time data.
- The **MySQL database** is like a storage box where you save the data for later.
- The program gathers system information, sends it to Kafka, stores it in MySQL, and does this repeatedly.

This pipeline helps track and record system stats continuously without missing any real-time data.