Kafka Docker Compose Architecture & Full Explanation

1. Overview of Services

This Docker Compose setup launches a full Kafka ecosystem:

- Zookeeper: Coordination service used by Kafka.

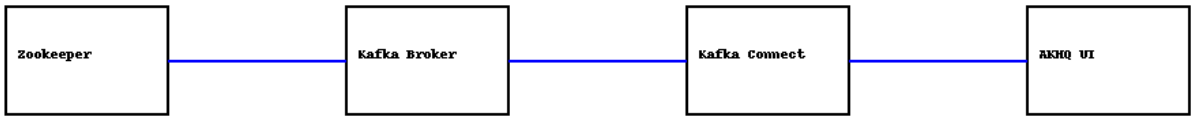
- Kafka Broker: Message broker handling pub-sub communication.

- Kafka Connect: Connector platform to stream data to/from Kafka.

- AKHQ: Web UI to visualize Kafka brokers, topics, and connectors.

All are connected via a single Docker bridge network.

2. Architecture Diagram



3. Docker Compose File - Line by Line Explanation

Kafka Docker Compose Architecture & Full Explanation

```
services:
  zookeeper2:
    image: confluentinc/cp-zookeeper:7.2.1
    ports:
      - "22181:2181"   # Maps host port 22181 to container's 2181
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000  # Zookeeper heartbeat interval

  kafka1:
    image: confluentinc/cp-kafka:7.2.1
    depends_on:
      - zookeeper2     # Ensures Zookeeper starts before Kafka
    ports:
      - "29092:29092"
      - "9092:9092"
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper2:2181
                                                    KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT
                                                                KAFKA_ADVERTISED_LISTENERS:
PLAINTEXT://kafka1:29092,PLAINTEXT_HOST://localhost:9092
      KAFKA_LISTENERS: PLAINTEXT://kafka1:29092,PLAINTEXT_HOST://0.0.0.0:9092
      KAFKA_AUTO_CREATE_TOPICS_ENABLE: 'true'
```

```
connect1:
  image: confluentinc/cp-kafka-connect:7.2.1
  depends_on:
    - kafka1
  ports:
    - '8083:8083'
  environment:
    CONNECT_BOOTSTRAP_SERVERS: kafka1:29092
    CONNECT_REST_ADVERTISED_HOST_NAME: connect1
    CONNECT_REST_PORT: 8083
    CONNECT_CONFIG_STORAGE_TOPIC: docker-connect-configs
    CONNECT_OFFSET_STORAGE_TOPIC: docker-connect-offsets
    CONNECT_STATUS_STORAGE_TOPIC: docker-connect-status
    CONNECT_KEY_CONVERTER: org.apache.kafka.connect.json.JsonConverter
    CONNECT_VALUE_CONVERTER: org.apache.kafka.connect.json.JsonConverter
    CONNECT_KEY_CONVERTER_SCHEMAS_ENABLE: 'false'
    CONNECT_VALUE_CONVERTER_SCHEMAS_ENABLE: 'false'
    CONNECT_PLUGIN_PATH: '/usr/share/java,/usr/share/confluent-hub-components'


akhq1:
  image: tchiotludo/akhq
  ports:
    - 8080:8080
  environment:
```

```
    AKHQ_CONFIGURATION: |
      akhq:
        connections:
          docker-kafka-server:
            properties:
              bootstrap.servers: "kafka1:29092"
            connect:
              - name: "connect"
                url: "http://connect1:8083"

networks:
  kafka-network1:
    name: kafka-network1
    driver: bridge
```

4. Internal Kafka Connect Topics

Kafka Connect uses 3 internal topics:

- docker-connect-configs: Stores connector definitions.

- docker-connect-offsets: Tracks source read progress.

- docker-connect-status: Maintains status of connectors and their tasks.

These are compacted topics and essential for Kafka Connect's fault-tolerance.

5. Common Docker Commands Explained

- docker-compose up --build

Builds and starts containers from docker-compose.yml.

- docker exec -it kafka1 bash

Opens an interactive shell inside the running Kafka container.

- kafka-topics --create

Creates a new Kafka topic from inside the broker container.

- kafka-console-producer / consumer

CLI tools to write to and read from Kafka topics.

- docker-compose down -v

Stops and removes containers, volumes, and network.