Apache Kafka and Its Ecosystem

~Prerak Gada (prerakgada.in)

About the Speaker

- **Prerak Gada, Founder & CEO at Engaze.in**

- Software Development Freelancer for 3 years
- Led 12+ Seminars and Hands-on
- Developed 6+ Apps with DevOps
- Pursuing 3rd year of B.E. in Electronics & Computer Scier
- App Lead at Google Developer Student Club – ACE
- Tech Head at IEEE Students Chapter – ACE

Prerak Gada
mail@prerakgada.in
https://prerakgada.in

Agenda

01 Introduction to Kafka

02 Core Concepts

03 Use Cases

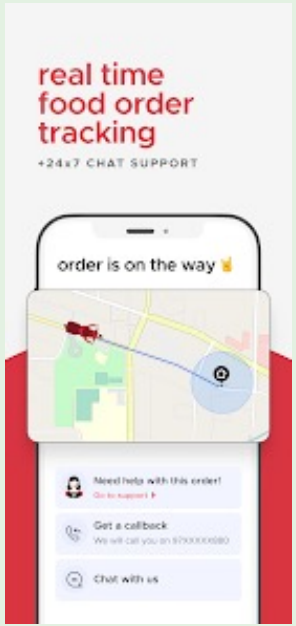
04 Hands-on

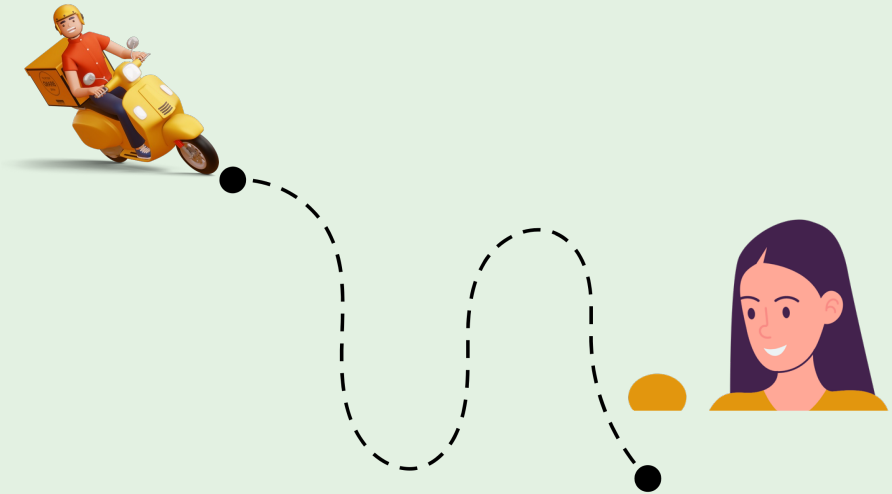
What is Apache Kafka?

Apache Kafka is a distributed streaming Open-source platform developed by Apache Software Foundation, written in Scala and Java

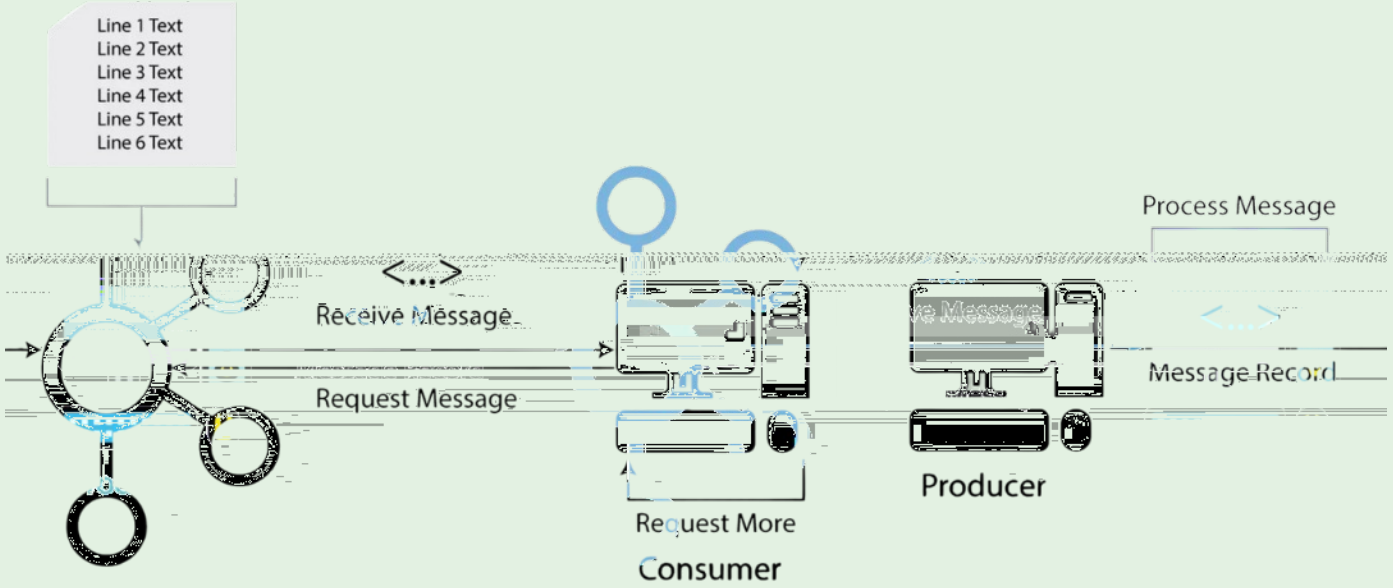
- A distributed system is a system with multiple components located on different machines that communicate and coordinate actions via messages to appear as a single system to the end-user.
- Create real time data streams
- Processing real time data stream

Let’s Understand with Swiggy!



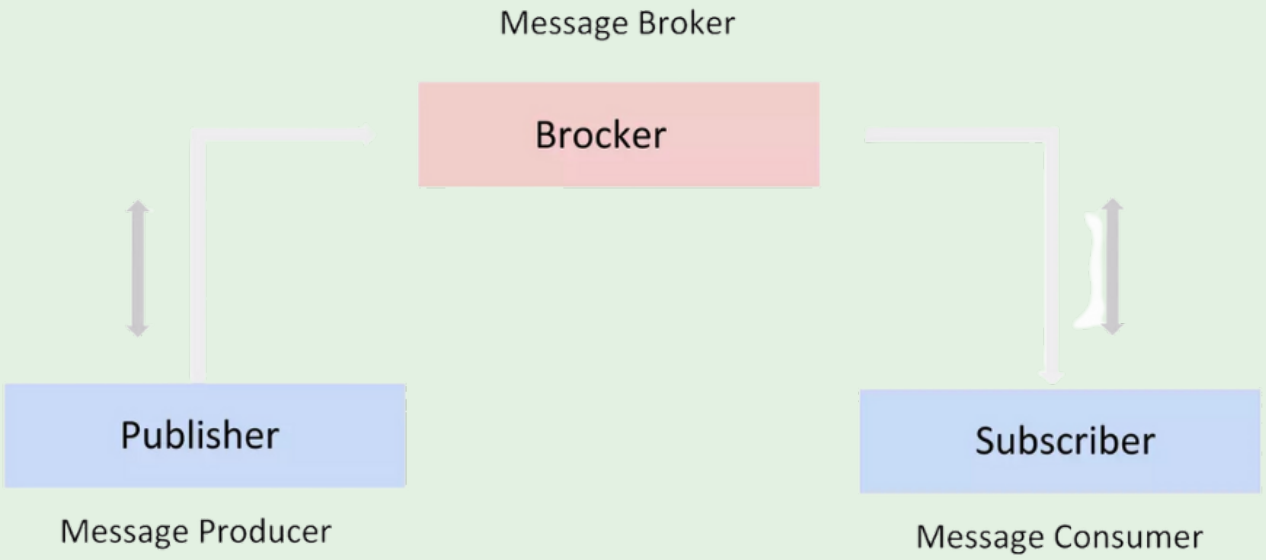


How Kafka Works?

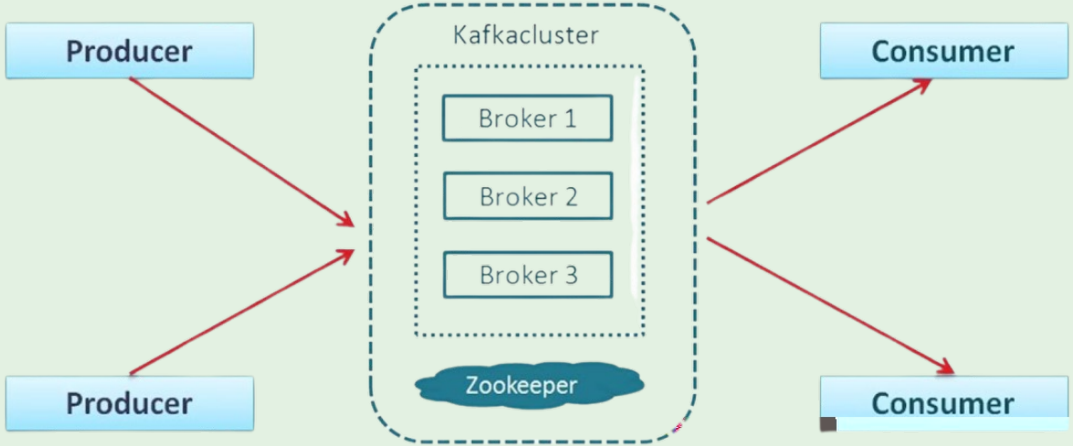


Core Concepts

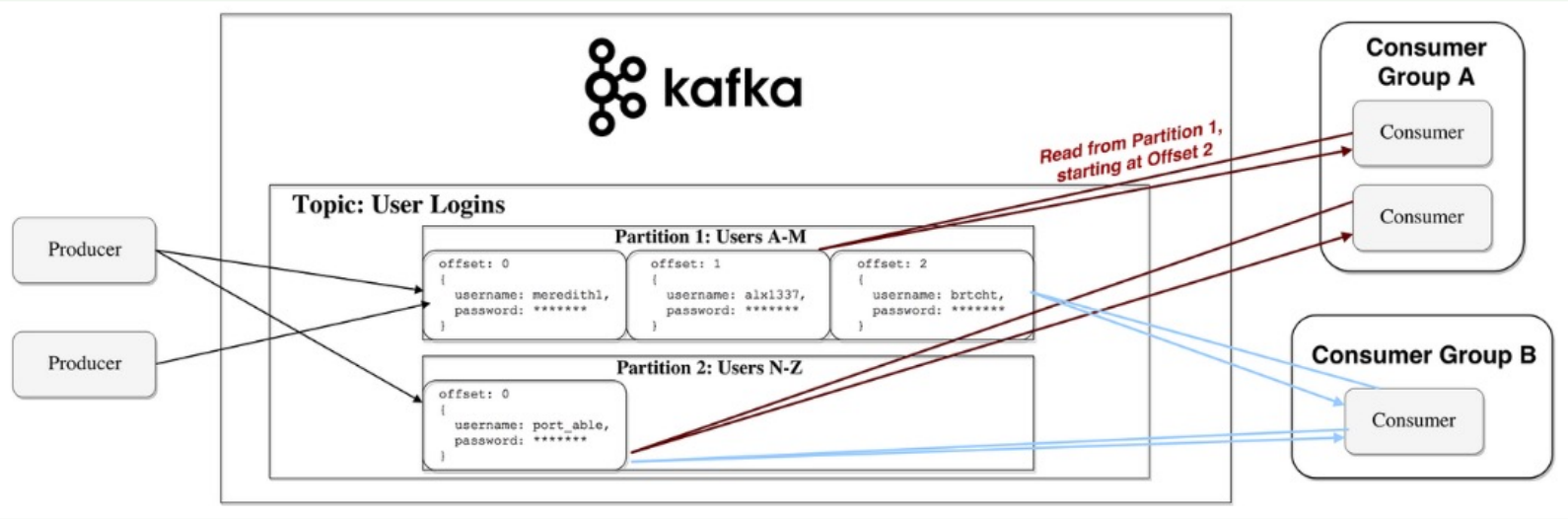
Producer, Consumer, and Broker



Kafka Cluster and Zookeeper



Use Case


kafka
Topic: User Logins
Partition 1: Users A-M
offset: 0
{
username: meredith1,
password: *******
}
offset: 1
{
username: alx1337,
password: *******
}
offset: 2
{
username: brtcht,
password: *******
}
Partition 2: Users N-Z
offset: 0
{
username: port_able,
password: *******
}
Producer
Producer
Read from Partition 1, starting at Offset 2
Consumer Group A
Consumer
Consumer
Consumer Group B
Consumer

Hands-on!

Docker Commands

- Start Zookeper Container and expose PORT 2181.
- docker run -p 2181:2181 zookeeper

- Start Kafka Container, expose PORT 9092 and setup ENV variables.

```
docker run -p 9092:9092 \
-e KAFKA_ZOOKEEPER_CONNECT=<PRIVATE_IP>:2181 \
-e KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://<PRIVATE_IP>:9092 \
-e KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR=1 \
confluentinc/cp-kafka
```

alient.js

```
const { Kafka } = require("kafkajs");

exports.kafka = new Kafka({
  clientId: "my-app",
  brokers: ["<PRIVATE_IP>:9092"],
});
```

admin.js

```
const { kafka } = require("./client");

async function init() {
  const admin = kafka.admin();
  console.log("Admin connecting...");
  admin.connect();
  console.log("Adming Connection Success...");

  console.log("Creating Topic [rider-updates]");
  await admin.createTopics({
    topics: [
      {
        topic: "rider-updates",
        numPartitions: 2,
      },
    ],
  });
  console.log("Topic Created Success [rider-updates]");

  console.log("Disconnecting Admin..");
  await admin.disconnect();
}

init();
```

producer.js

```
const { kafka } = require("./client");
const readline = require("readline");

const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout,
});

async function init() {
  const producer = kafka.producer();

  console.log("Connecting Producer");
  await producer.connect();
  console.log("Producer Connected Successfully");

  rl.setPrompt("> ");
  rl.prompt();

  rl.on("line", async function (line) {
    const [riderName, location] = line.split(" ");
    await producer.send({
      topic: "rider-updates",
      messages: [
        {
          partition: location.toLowerCase() === "north" ? 0 : 1,
          key: "location-update",
          value: JSON.stringify({ name: riderName, location }),
        },
      ],
    });
  }).on("close", async () => {
    await producer.disconnect();
  });
}

init();
```

consumer.js

```
const { kafka } = require("./client");
const group = process.argv[2];

async function init() {
  const consumer = kafka.consumer({ groupId: group });
  await consumer.connect();

  await consumer.subscribe({ topics: ["rider-updates"],
fromBeginning: true });

  await consumer.run({
    eachMessage: async ({ topic, partition, message, heartbeat,
pause }) => {
      console.log(
        `${group}: [${topic}]: PART:${partition}:`,
        message.value.toString()
      );
    },
  });
}

init();
```

Thank You