

<https://gist.github.com/piyushgarg-dev/32cadf6420c452b66a9a6d977ade0b01>  
<https://kafka.js.org/docs/admin#create-partitions>

- Start Zookeeper 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
```

<https://kafka.js.org/>

```
npm install --global yarn
```

```
Create a folder and run - yarn init
```

```
Then run - yarn add kafkajs
```

client.js -

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

exports.kafka = new Kafka({
  clientId: "my-app",
  brokers: ["192.168.1.7:9092"],
});
```

admin.js

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

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

```

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

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-status",
  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-status"], fromBeginning: true });

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

init();

```

- Run Multiple Consumers

`node consumer.js <GROUP_NAME>` — example - `node consumer.js group-1` (run this twice on two terminals so that there are two consumers in the same group called group-1. All data from partition 0 will be north, anything else will be in partition 1

- Create Producer

`node producer.js`

> tony south

> tony north