

Tarea 2 Sistemas Operativos

Alejandro Díaz

Introducción:

En esta experiencia se solicita implementar Apache Kafka como sistema distribuido para gestionar los procesos del gremio de sopaipilleros de Chile.

Desarrollo:

Se inicializan los contenedores necesarios, siendo principalmente el contenedor de Kafka, el contenedor de la api que contendrá rutas que actuarán como producers, el contenedor de zookeeper, dependencia necesaria de Kafka y por último todos los contenedores que actuarán como consumers.

```
version: '3.1'

services:
  api:
    container_name: api
    build: ./api
    restart: always
    depends_on:
      - kafka
      - zookeeper
    ports:
      - "3000:3000"
    environment:
      PORT: 3000
      kafkaHost: kafka:9092

  zookeeper:
    container_name: zookeeper
    image: 'bitnami/zookeeper:latest'
    environment:
      ALLOW_ANONYMOUS_LOGIN: "yes"
      ZOOKEEPER_CLIENT_PORT: 2181
    ports:
      - 2181:2181
      - 2888:2888
      - 3888:3888

  kafka:
    container_name: kafka
    image: 'bitnami/kafka:latest'
    depends_on:
      - zookeeper
    environment:
      KAFKA_ADVERTISED_HOST_NAME: "kafka"
      KAFKA_ADVERTISED_PORT: "9092"
      ALLOW_PLAINTEXT_LISTENER: "yes"
      KAFKA_CFG_ZOOKEEPER_CONNECT: "zookeeper:2181"
      KAFKA_CFG_NUM_PARTITIONS: 2
    ports:
      - 9092:9092
```

Figura 1: Configuración de DockerFile.

Se dispondrá de 5 contenedores de consumers los cuales estarán suscritos al tópico respectivo necesario para su función, luego desde la api los producers podrán entregar la información a los tópicos correspondientes

```
Id":"ubicacion", "memberId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "leaderId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "isLeader":true, "memberAssignment":{"ubicacion":[0,1]}, "groupProtocol":"RoundRobinAssigner", "duration":4502}
kafka | [2022-11-04 09:32:07,959] INFO [GroupCoordinator 1001]: Dynamic member with unknown member id joins group registroVenta in Empty state. Created a new member id kafkajs-d3f28761-94a7-43ac-8ab6-ab0cecc7ab5b and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
kafka | [2022-11-04 09:32:07,963] INFO [GroupCoordinator 1001]: Preparing to rebalance group registroVenta in state PreparingRebalance with old generation 0 (__consumer_offsets-9) (reason: Adding new member kafkajs-d3f28761-94a7-43ac-8ab6-ab0cecc7ab5b with group instance id None; client reason: not provided) (kafka.coordinator.group.GroupCoordinator)
kafka | [2022-11-04 09:32:07,965] INFO [GroupCoordinator 1001]: Stabilized group registroVenta generation 1 (__consumer_offsets-9) with 1 members (kafka.coordinator.group.GroupCoordinator)
kafka | [2022-11-04 09:32:07,972] INFO [GroupCoordinator 1001]: Assignment received from leader kafkajs-d3f28761-94a7-43ac-8ab6-ab0cecc7ab5b for group registroVenta for generation 1. The group has 1 members, 0 of which are static. (kafka.coordinator.group.GroupCoordinator)
registroVenta_1 | {"level":"INFO", "timestamp":"2022-11-04T09:32:07.981Z", "logger":"kafkajs", "message":"[ConsumerGroup] Consumer has joined the group", "groupId":"registroVenta", "memberId":"kafkajs-d3f28761-94a7-43ac-8ab6-ab0cecc7ab5b", "leaderId":"kafkajs-d3f28761-94a7-43ac-8ab6-ab0cecc7ab5b", "isLeader":true, "memberAssignment":{"registroVenta":[0,1]}, "groupProtocol":"RoundRobinAssigner", "duration":5234}
kafka | [2022-11-04 09:32:07,994] INFO [GroupCoordinator 1001]: Dynamic member with unknown member id joins group ubicacion in Stable state. Created a new member id kafkajs-9ada2f38-297c-4d14-b407-bfa86d44a1bf and request the member to rejoin with this id. (kafka.coordinator.group.GroupCoordinator)
kafka | [2022-11-04 09:32:08,009] INFO [GroupCoordinator 1001]: Preparing to rebalance group ubicacion in state PreparingRebalance with old generation 1 (__consumer_offsets-33) (reason: Adding new member kafkajs-9ada2f38-297c-4d14-b407-bfa86d44a1bf with group instance id None; client reason: not provided) (kafka.coordinator.group.GroupCoordinator)
ubicacion_1 | {"level":"ERROR", "timestamp":"2022-11-04T09:32:12.271Z", "logger":"kafkajs", "message":"[Connection] Response Heartbeat(key: 12, version: 3)", "broker":"bf8db65da6b7:9092", "clientId":"kafkajs", "error":"The group is rebalancing, so a rejoin is needed", "correlationId":11, "size":10}
ubicacion_1 | {"level":"ERROR", "timestamp":"2022-11-04T09:32:12.274Z", "logger":"kafkajs", "message":"[Connection] Response Heartbeat(key: 12, version: 3)", "broker":"bf8db65da6b7:9092", "clientId":"kafkajs", "error":"The group is rebalancing, so a rejoin is needed", "correlationId":12, "size":10}
ubicacion_1 | {"level":"WARN", "timestamp":"2022-11-04T09:32:12.276Z", "logger":"kafkajs", "message":"[Runner] The group is rebalancing, re-joining", "groupId":"ubicacion", "memberId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "error":"The group is rebalancing, so a rejoin is needed"}
kafka | [2022-11-04 09:32:12,280] INFO [GroupCoordinator 1001]: Stabilized group ubicacion generation 2 (__consumer_offsets-33) with 2 members (kafka.coordinator.group.GroupCoordinator)
kafka | [2022-11-04 09:32:12,288] INFO [GroupCoordinator 1001]: Assignment received from leader kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96 for group ubicacion for generation 2. The group has 2 members, 0 of which are static. (kafka.coordinator.group.GroupCoordinator)
agentextra_1 | {"level":"INFO", "timestamp":"2022-11-04T09:32:12.292Z", "logger":"kafkajs", "message":"[ConsumerGroup] Consumer has joined the group", "groupId":"ubicacion", "memberId":"kafkajs-9ada2f38-297c-4d14-b407-bfa86d44a1bf", "leaderId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "isLeader":false, "memberAssignment":{"ubicacion":[1]}, "groupProtocol":"RoundRobinAssigner", "duration":19249}
ubicacion_1 | {"level":"INFO", "timestamp":"2022-11-04T09:32:12.299Z", "logger":"kafkajs", "message":"[ConsumerGroup] Consumer has joined the group", "groupId":"ubicacion", "memberId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "leaderId":"kafkajs-64ef68a0-3b0d-416a-b933-8510cd027c96", "isLeader":true, "memberAssignment":{"ubicacion":[0]}, "groupProtocol":"RoundRobinAssigner", "duration":22}
```

Figura 2: Consumers suscritos a los topics correspondientes.

Finalmente no se logra el desarrollo más allá de la suscripción de los tópicos.

Conclusión:

Se concluye esta experiencia respondiendo las preguntas planteadas:

¿Cómo Kafka puede escalar tanto vertical como horizontalmente? Relacione su respuesta con el problema asociado, dando un ejemplo para cada uno de los tipos de escalamiento.

R: Kafka puede escalar verticalmente configurando los parámetros de particiones y tópicos a los que acceden los consumidores, y puede escalar horizontalmente añadiendo más máquinas que actúen como brokers ayudando al balance de carga.

¿Qué características puede observar de Kafka como sistema distribuido? ¿Cómo se reflejan esas propiedades en la arquitectura de Kafka?

R: Kafka es un sistema que cuenta con 2 principales agentes, producers y consumers, el broker principal se encarga de gestionar el balance de carga que le llega a estos a través de las particiones correspondientes a cada topic, esto se ve reflejado en el funcionamiento necesario que se debió configurar, cada producer debe entregar la información a un topic y cada consumer que se informe desde ese topic debe suscribirse, sin embargo no se observa una conexión directa entre producer-consumer, lo que funciona como sistema distribuido a través del balance de carga que genera el broker.