kafka-perf-test

請試著佈署一個kafka cluster，並用kafka-producer-perf-test script送資料給一個topic，試著讓一個topic partition的資料夾內的log segment (xxx.log) 出現至少2個，並用kafka-dump-log script看最前面三筆資料。請以截圖搭配說明並作成 pdf 上傳

重現步驟：

使用以下配置起一個 3 個 broker 的 zookeeper mode kafka cluster

```
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:7.2.1
    container_name: zookeeper
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
  kafka1:
    image: confluentinc/cp-kafka:7.2.1
    container_name: kafka1
    ports:
      - "8097:8097"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
EXTERNAL:PLAINTEXT,INTERNAL:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS:
EXTERNAL://localhost:8097,INTERNAL://kafka1:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
  kafka2:
    image: confluentinc/cp-kafka:7.2.1
    container_name: kafka2
    ports:
      - "8098:8098"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 2
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
EXTERNAL:PLAINTEXT,INTERNAL:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS:
EXTERNAL://localhost:8098,INTERNAL://kafka2:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
  kafka3:
    image: confluentinc/cp-kafka:7.2.1
    container_name: kafka3
    ports:
```

```
      - "8099:8099"
    depends_on:
      - zookeeper
    environment:
      KAFKA_BROKER_ID: 3
      KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP:
EXTERNAL:PLAINTEXT,INTERNAL:PLAINTEXT
      KAFKA_ADVERTISED_LISTENERS:
EXTERNAL://localhost:8099,INTERNAL://kafka3:9092
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
```

使用 `kafka-producer-perf-test` 製造 500000 個 record

```
❯ kafka-producer-perf-test --topic test-topic --num-records 500000 --record-size 1000 --thr
oughput -1 --producer-props bootstrap.servers=localhost:8097,localhost:8098,localhost:8099

394577 records sent, 78915.4 records/sec (75.26 MB/sec), 371.3 ms avg latency, 489.0 ms max
 latency.
500000 records sent, 80398.777939 records/sec (76.67 MB/sec), 373.00 ms avg latency, 489.00
 ms max latency, 374 ms 50th, 456 ms 95th, 478 ms 99th, 487 ms 99.9th.

 ~/r/u/playground-2022/ka/kafka-java-getting-started  master ⇡1 *1 !4 ?182
```

使用 kafka-topic 工具找到 partition leader 為 ID = 2 的 broker

```
❯ kafka-topics --describe --bootstrap-server localhost:8098 --topic test-topic
Topic: test-topic       TopicId: rzsa_wagQE6UatT3yYhDMA PartitionCount: 1       Replication
Factor: 1       Configs:
        Topic: test-topic       Partition: 0    Leader: 2       Replicas: 2     Isr: 2
```

進到 zookeeper 使用 `zookeeper-shell` 工具 執行 `get /brokers/ids/2` 找到這個 ID=2 的 broker 所在的 ip `localhost:8098`, 為 `kafka2`

```
❯ docker exec -it zookeeper bash
[appuser@d33ea0746fdc ~]$ zookeeper-shell localhost:2181
Connecting to localhost:2181
Welcome to ZooKeeper!
JLine support is disabled

WATCHER::

WatchedEvent state:SyncConnected type:None path:null
get /brokers/ids/2
{"features":{},"listener_security_protocol_map":{"EXTERNAL":"PLAINTEXT","INTERNAL":"PLAINT
EXT"},"endpoints":["EXTERNAL://localhost:8098","INTERNAL://kafka2:9092"],"jmx_port":-1,"po
rt":8098,"host":"localhost","version":5,"timestamp":"1704159653929"}
```

使用 docker exec 進到 broker `kafka2` 的 test-topic dir, 看到有兩個以上 `xxx.log`

```
[appuser@415811a3c888 ~]$ ls /var/lib/kafka/data/test-topic-0/
00000000000000000000.index      00000000000001060144.snapshot
00000000000000000000.log        00000000000001060144.timeindex
00000000000000000000.timeindex  leader-epoch-checkpoint
00000000000001060144.index      partition.metadata
00000000000001060144.log
[appuser@415811a3c888 ~]$
```

看最前面那個 log 的前三筆資料 (baseOffset: 0, 16, 32)

```
[appuser@415811a3c888 ~]$ kafka-dump-log --files /var/lib/kafka/data/test-topic-0/00000000
000000000000.log | head -n 10
Dumping /var/lib/kafka/data/test-topic-0/00000000000000000000.log
Starting offset: 0
baseOffset: 0 lastOffset: 15 count: 16 baseSequence: 0 lastSequence: 15 producerId: 0 prod
ucerEpoch: 0 partitionLeaderEpoch: 0 isTransactional: false isControl: false deleteHorizon
Ms: OptionalLong.empty position: 0 CreateTime: 1704181086429 size: 16205 magic: 2 compress
codec: none crc: 699985189 isvalid: true
baseOffset: 16 lastOffset: 31 count: 16 baseSequence: 16 lastSequence: 31 producerId: 0 pr
oducerEpoch: 0 partitionLeaderEpoch: 0 isTransactional: false isControl: false deleteHoriz
onMs: OptionalLong.empty position: 16205 CreateTime: 1704181086430 size: 16205 magic: 2 co
mpresscodec: none crc: 1710737397 isvalid: true
baseOffset: 32 lastOffset: 47 count: 16 baseSequence: 32 lastSequence: 47 producerId: 0 pr
oducerEpoch: 0 partitionLeaderEpoch: 0 isTransactional: false isControl: false deleteHoriz
onMs: OptionalLong.empty position: 32410 CreateTime: 1704181086431 size: 16205 magic: 2 co
mpresscodec: none crc: 2105528465 isvalid: true
```