

DSC 650

Inman, Gracie

Week 7

04/28/24

## Kafka Introduction

- My Topic Screenshot

```
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker compose up -d
[+] Running 2/3
  ✓ Network kafka_default Created
  ✓ Container kafka-kafka-1 Started
  ✓ Container kafka-zookeeper-1 Started
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     STATUS      PORTS                               NAMES
f13cd57e3833   nasheeb/zookee /docker-entrypoint..." 6 seconds ago Up 6 seconds 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 8080/tcp  kafka-zookeeper-1
aec2e20ea74a   nasheeb/kafka:1.0 "start-kafka.sh"         6 seconds ago Up 6 seconds 0.0.0.0:32858->9092/tcp, :::32858->9092/tcp  kafka-kafka-1
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-topics.sh --create --topic my-topic --bootstrap-server localhost:9092
Created topic my-topic.
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-topics.sh --list --bootstrap-server localhost:9092
my-topic
root@aec2e20ea74a:/#
```

- Producer and consumer terminals

```
Terminal Shell Edit View Window Help
gracieinman -- gracieinman@big-data-week1-instance: ~/dsc650-infra-main/bellevue-bigdata/kafka -- ssh gracieinman@34.16.209.77 -- 209x26

CONTAINER ID   IMAGE          COMMAND                  CREATED     STATUS      PORTS                               NAMES
a7677ba8889c   nasheeb/kafka:1.0 "start-kafka.sh"         20 minutes ago Restarting (1) 20 seconds ago 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 8080/tcp  kafka-zookeeper-1
c117ef8b0712   nasheeb/zookee /docker-entrypoint..." 20 minutes ago Up 20 minutes 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 8080/tcp  kafka-zookeeper-1
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker compose down
[+] Running 3/3
  ✓ Container kafka-kafka-1 Removed
  ✓ Container kafka-zookeeper-1 Removed
  ✓ Network kafka_default Removed
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker compose up -d
[+] Running 2/3
  ✓ Network kafka_default Created
  ✓ Container kafka-kafka-1 Started
  ✓ Container kafka-zookeeper-1 Started
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     STATUS      PORTS                               NAMES
f13cd57e3833   nasheeb/zookee /docker-entrypoint..." 6 seconds ago Up 6 seconds 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 8080/tcp  kafka-zookeeper-1
aec2e20ea74a   nasheeb/kafka:1.0 "start-kafka.sh"         6 seconds ago Up 6 seconds 0.0.0.0:32858->9092/tcp, :::32858->9092/tcp  kafka-kafka-1
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker exec -it kafka-kafka-1 bash
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-topics.sh --create --topic my-topic --bootstrap-server localhost:9092
Created topic my-topic.
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-topics.sh --list --bootstrap-server localhost:9092
my-topic
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-console-producer.sh --topic my-topic --from-beginning --bootstrap-server localhost:9092
Hi my name is Gracie and I am trying to learn kafka. I think it is going okay.
>
>

gracieinman -- gracieinman@big-data-week1-instance: ~/dsc650-infra-main/bellevue-bigdata/kafka -- ssh gracieinman@34.16.209.77 -- 208x30

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.
https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.
19 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sun Apr 28 17:59:45 2024 from 73.212.143.78
gracieinman@big-data-week1-instance:~$ cd dsc650-infra-main
gracieinman@big-data-week1-instance:~/dsc650-infra-main$ cd bellevue-bigdata
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata$ cd kafka
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     STATUS      PORTS                               NAMES
f13cd57e3833   nasheeb/zookee /docker-entrypoint..." About a minute ago Up About a minute 2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 8080/tcp  kafka-zookeeper-1
aec2e20ea74a   nasheeb/kafka:1.0 "start-kafka.sh"         About a minute ago Up About a minute 0.0.0.0:32858->9092/tcp, :::32858->9092/tcp  kafka-kafka-1
gracieinman@big-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker exec -it kafka-kafka-1 bash
root@aec2e20ea74a:/# /opt/kafka_2.13-2.8.1/bin/kafka-console-producer.sh --topic my-topic --bootstrap-server localhost:9092
>Hi my name is Gracie and I am trying to learn kafka. I think it is going okay.
>
>
```

- Performance test

```
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-producer-perf-test.sh --topic my-topic --num-records 50000 --record-size 100 --throughput 1000 --producer-props bootstrap.servers=localhost:9092 key.serializer=org.apache.kafka.common.serialization.StringSerializer value.serializer=org.apache.kafka.common.serialization.StringSerializer
5001 records sent, 1000.0 records/sec (0.10 MB/sec), 0.7 ms avg latency, 407.0 ms max latency.
5002 records sent, 1001.0 records/sec (0.10 MB/sec), 0.8 ms avg latency, 15.0 ms max latency.
5002 records sent, 1000.0 records/sec (0.10 MB/sec), 0.7 ms avg latency, 9.0 ms max latency.
5002 records sent, 1000.2 records/sec (0.10 MB/sec), 0.6 ms avg latency, 3.0 ms max latency.
5000 records sent, 1000.0 records/sec (0.10 MB/sec), 0.6 ms avg latency, 6.0 ms max latency.
5001 records sent, 1000.2 records/sec (0.10 MB/sec), 0.5 ms avg latency, 1.0 ms max latency.
5001 records sent, 1000.0 records/sec (0.10 MB/sec), 0.5 ms avg latency, 5.0 ms max latency.
5003 records sent, 1000.4 records/sec (0.10 MB/sec), 0.5 ms avg latency, 5.0 ms max latency.
5001 records sent, 1000.0 records/sec (0.10 MB/sec), 0.5 ms avg latency, 5.0 ms max latency.
50000 records sent, 999.720078 records/sec (0.10 MB/sec), 0.92 ms avg latency, 407.00 ms max latency, 1 ms 50th, 1 ms 95th, 4 ms 99th, 36 ms 99.9th.
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-consumer-perf-test.sh --broker-list localhost:9092 --topic my-topic --messages 50000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2024-04-28 18:32:33.722, 2024-04-28 18:32:38.638, 4.7685, 0.6599, 50003, 69836.5922, 418, 298, 16.0017, 167795.3028
root@ec2-2e20ea74a:/#
```

The performance test indicates how well Kafka is performing and can identify possible issues to ensure efficient processing. The results of the producer performance test indicate that 50000 records were sent at a rate of 1000 records per second. The average latency that was experienced was 0.92 milliseconds and the max latency was 407 milliseconds. The results of the consumer performance test indicated that 4.7685 MB of data was consumed at a rate of 6.66 MB/second. There were 50003 messages consumed at a rate of 69836 messages per second. The rebalance time was 418 milliseconds, the fetch time was 298 milliseconds, and the fetch rate was 16.0017 MD per second.

- Scaled Performance test.

```
gracie@man01g-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker-compose scale kafka=3
WARNING: The scale command is deprecated. Use the up command with the --scale flag instead.
Creating kafka_kafka_2 ... done
Creating kafka_kafka_3 ... done
gracie@man01g-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker exec -it kafka_kafka_1 bash
Error response from daemon: No such container: kafka_kafka_1
gracie@man01g-data-week1-instance:~/dsc650-infra-main/bellevue-bigdata/kafka$ docker exec -it kafka-kafka-1 bash
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-topics.sh --create --topic my-partitioned-topic --replication-factor 3 --partitions 3 --bootstrap-server localhost:9092
Created topic my-partitioned-topic.
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-producer-perf-test.sh --topic my-partitioned-topic --num-records 50000 --record-size 100 --throughput 1000 --producer-props bootstrap.servers=localhost:9092 key.serializer=org.apache.kafka.common.serialization.StringSerializer value.serializer=org.apache.kafka.common.serialization.StringSerializer
4905 records sent, 996.0 records/sec (0.09 MB/sec), 12.9 ms avg latency, 397.0 ms max latency.
5024 records sent, 1004.6 records/sec (0.10 MB/sec), 4.7 ms avg latency, 71.0 ms max latency.
5005 records sent, 1000.8 records/sec (0.10 MB/sec), 3.2 ms avg latency, 41.0 ms max latency.
5004 records sent, 1000.6 records/sec (0.10 MB/sec), 1.8 ms avg latency, 17.0 ms max latency.
5000 records sent, 1000.0 records/sec (0.10 MB/sec), 2.0 ms avg latency, 71.0 ms max latency.
5004 records sent, 1000.4 records/sec (0.10 MB/sec), 1.4 ms avg latency, 44.0 ms max latency.
5004 records sent, 1000.6 records/sec (0.10 MB/sec), 1.1 ms avg latency, 28.0 ms max latency.
5001 records sent, 1000.0 records/sec (0.10 MB/sec), 0.9 ms avg latency, 12.0 ms max latency.
5003 records sent, 1000.6 records/sec (0.10 MB/sec), 0.9 ms avg latency, 7.0 ms max latency.
50000 records sent, 999.720078 records/sec (0.10 MB/sec), 3.48 ms avg latency, 397.00 ms max latency, 1 ms 50th, 12 ms 95th, 38 ms 99th, 133 ms 99.9th.
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-consumer-perf-test.sh --broker-list localhost:9092 --topic my-partitioned-topic --messages 50000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
root@ec2-2e20ea74a:/# /opt/kafka-2.13-2.8.1/bin/kafka-consumer-perf-test.sh --broker-list localhost:9092 --topic my-partitioned-topic --messages 50000
start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2024-04-28 19:05:11.450, 2024-04-28 19:05:12.211, 4.7684, 0.2659, 50000, 65703.0223, 434, 327, 14.5822, 152905.1988
root@ec2-2e20ea74a:/# ex
```

Compared to the unscaled performance test, the scaled performance test had a lower average latency, and a higher fetch rate, but slightly higher rebalance and fetch times. This could be useful when working with large amounts of data. However, it would be important to consider the tradeoff of the slightly higher rebalance and fetch times for the lower latency and fetch rate.