Apache Kafka: An Intro

by Fatih Nayebi, Ph.D.

Master of Management Analytics, Desautels Faculty of Management, McGill

Setup

- The first thing to do is to run the landoop/fast-data-dev Docker image. For example:

```
docker run --rm -it -p 2181:2181 -p 3030:3030 -p
8081:8081 -p 8082:8082 -p 8083:8083 -p 9092:9092 -p
9581:9581 -p 9582:9582 -p 9583:9583 -p 9584:9584 -e
ADV_HOST=127.0.0.1 landoop/fast-data-dev:latest
```

- *Note: Please follow the instructions on fast-data-dev README to customize the container.*

Run

- You can execute a bash shell at the running container as follow:

```
docker run --rm -it --net=host landoop/fast-data-dev bash
```

- *Note: Kafka utilities are now available.*

Topics

- You can create a new Kafka topic named **`my-topic`** as follows:

```
kafka-topics --create --zookeeper localhost:2181 --
replication-factor 1 --partitions 3 --topic my-topic
```

Verify the creation of a topic

- You can verify that the `my-topic` topic was successfully created by listing all available topics:

```
kafka-topics --list --zookeeper localhost:2181
```

Add more partitions to the topic

- You can add more partitions as follows:

```
kafka-topics --zookeeper localhost:2181 --alter --topic
my-topic --partitions 16
```

Delete a topic

- You can delete a topic named `my-topic` as follows:

```
kafka-topics --zookeeper localhost:2181 --delete --topic
my-topic
```

Find more details about a topic

- You can find more details about a topic named `cc_payments` as follows:

```
kafka-topics --describe --zookeeper localhost:2181 --
topic cc_payments
```

Find under-replicated partitions

- You can see the under-replicated partitions for all topics as follows:

```
kafka-topics --zookeeper localhost:2181/kafka-cluster --
describe --under-replicated-partitions
```

Producers

- You can produce messages from standard input as follows:

```
kafka-console-producer --broker-list localhost:9092 --
topic my-topic
```

Produce messages from an existing file

- You can produce new messages from an existing file named `messages.txt` as follows:

```
kafka-console-producer --broker-list localhost:9092 --
topic test < messages.txt
```

Produce Avro messages

- You can produce Avro messages as follows:

```
kafka-avro-console-producer --broker-list localhost:
9092 --topic my.Topic --property
value.schema='{"type":"record","name":"myrecord","fields"
:[{"name":"f1","type":"string"}]}' --property
schema.registry.url=http://localhost:8081
```

- You can enter a few new values from the console as follows:

```
{"f1": "value1"}
```

Consumers

Consume messages

- You can begin a consumer from the beginning of the log as follows:

```
kafka-console-consumer --bootstrap-server localhost:9092
--topic my-topic --from-beginning
```

Consume a single message

- You can consume a single message as follows:

```
kafka-console-consumer --bootstrap-server localhost:9092
--topic my-topic  --max-messages 1
```

Download a single message

- You can consume a single message from `__consumer_offsets` as follows:

```
kafka-console-consumer --bootstrap-server localhost:9092
--topic __consumer_offsets --formatter
'kafka.coordinator.GroupMetadataManager$OffsetsMessageFor
matter' --max-messages 1
```

Consume a specific consumer group

- You can consume and specify a consumer group as follows:

```
kafka-console-consumer --topic my-topic --new-consumer --
bootstrap-server localhost:9092 --consumer-property
group.id=my-group
```

Consume Avro messages

- You can consume 10 Avro messages from a topic named `position-reports` as follows:

```
kafka-avro-console-consumer --topic position-reports --
new-consumer --bootstrap-server localhost:9092 --from-
beginning --property schema.registry.url=localhost:8081
--max-messages 10
```

Consume all existing Avro messages

- You can consume all existing Avro messages from a topic named `position-reports` as follows:

```
kafka-avro-console-consumer --topic position-reports --new-consumer --bootstrap-server localhost:9092 --from-beginning --property schema.registry.url=localhost:8081
```

Consumers admin operations

- You can list all groups as follows:

```
kafka-consumer-groups --new-consumer --list --bootstrap-server localhost:9092
```

Describe consumer groups

- You can describe a Group named `testgroup` as follows:

```
kafka-consumer-groups --bootstrap-server localhost:9092
--describe --group testgroup
```

Config

- You can set the retention for a topic as follows:

```
kafka-configs --zookeeper localhost:2181 --alter --
entity-type topics --entity-name my-topic --add-config
retention.ms=3600000
```

Print all configuration overrides

- You can print all configuration overrides for a topic named `my-topic` as follows:

```
kafka-configs --zookeeper localhost:2181 --describe --
entity-type topics --entity-name my-topic
```

Delete a configuration override

- You can delete a configuration override for `retention.ms` for a topic named `my-topic` as follows:

```
kafka-configs --zookeeper localhost:2181 --alter --
entity-type topics --entity-name my-topic --delete-config
retention.ms
```

Performance

- Although Kafka is pretty fast by design, it is good to be able to test its performance.
- You can check the Produce performance of Kafka as follows:

```
kafka-producer-perf-test --topic position-reports --
throughput 10000 --record-size 300 --num-records 20000 --
producer-props bootstrap.servers="localhost:9092"
```

ACLs

- You can add a new *consumer* ACL to an existing topic as follows:

```
kafka-acls --authorizer-properties
zookeeper.connect=localhost:2181 --add --allow-principal
User:Bob --consumer --topic topicA --group groupA
```

Add a new producer ACL

- You can add a new *producer* ACL to an existing topic as follows:

```
kafka-acls --authorizer-properties
zookeeper.connect=localhost:2181 --add --allow-principal
User:Bob --producer --topic topicA
```

List the ACLs

- You can list the ACLs of a topic named **`topicA`** as follows:

```
kafka-acls --authorizer-properties
zookeeper.connect=localhost:2181 --list --topic topicA
```

Zookeeper

- You can enter the zookeeper shell as follows:

```
zookeeper-shell localhost:2182 ls
```