Recitation 1

Apache Kafka

Shreyans Sheth
May 20th, 2020

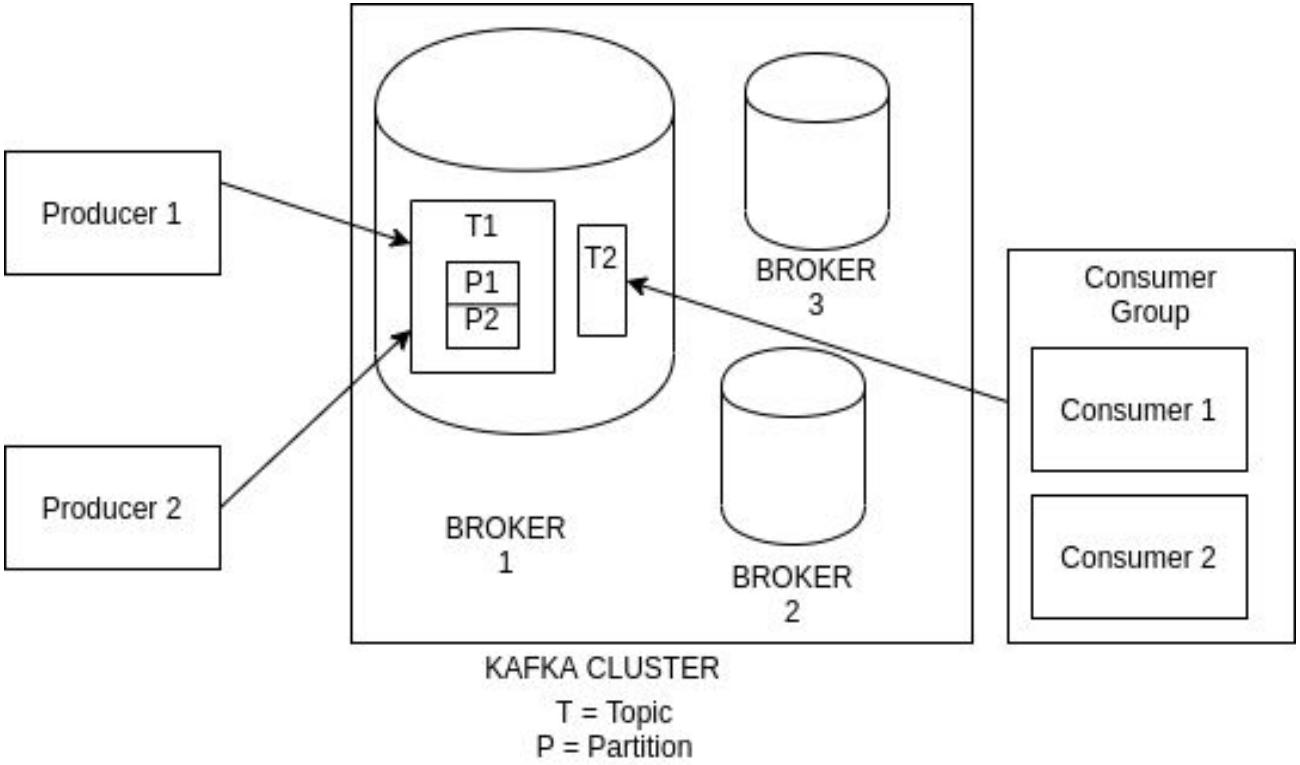
Agenda

- Kafka Basics
- Kafka Architecture Overview
- Software Setup
- Simple programming exercise

Apache Kafka - Introduction

- **What is it** ?
  - *Essentially*, a publish subscribe system in it’s canonical use case
  - Designed for scalability, reliability and high throughput
  - 3 primary concepts - Produces, Topics, Consumers
- **Why use it ?**
  - Used for real time data stream processing
  - Eg. Logging, Metrics, high volume of real time activities, etc.
- **Why learnt it ?**
  - You will need it in your assignments :)

Apache Kafka - Architecture Overview



- Publishers
- Consumers
- Clusters
- Brokers
- Topics
- Partitions
  - Why ? Scalability
- Consumers
  - Offsets
- Groups
  - One consumer per group per topic

Setup

- Open a new terminal
- Run ***ssh -L 9092:localhost:9092 tunnel@128.2.204.215 -NT***
  - This command forwards the port 9092 of our server to your local machine (just leave it running in the background)
  - Password given during recitation
  - Alternatively, use the ssh key found here and run *ssh -L 9092:localhost:9092 tunnel@128.2.204.215 -NT -i id_rsa*
- Install kafkacat (CLI tool for Kafka)
  - ***brew install kafkacat*** OR ***sudo apt-get install kafkacat***
- Test your connection
  - ***kafkacat -b localhost -L1***

Setup - Successful Output

```
Metadata for all topics (from broker 1: localhost:9092/1):
 1 brokers:
  broker 1 at localhost:9092
 5 topics:
  topic "test" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "movielog" with 4 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
    partition 2, leader 1, replicas: 1, isrs: 1
    partition 3, leader 1, replicas: 1, isrs: 1
    partition 1, leader 1, replicas: 1, isrs: 1
  topic "_confluent-license" with 1 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
  topic "_confluent-metrics" with 12 partitions:
    partition 0, leader 1, replicas: 1, isrs: 1
    partition 5, leader 1, replicas: 1, isrs: 1
    partition 10, leader 1, replicas: 1, isrs: 1
    partition 2, leader 1, replicas: 1, isrs: 1
    partition 8, leader 1, replicas: 1, isrs: 1
```

Exercise - Bootstrapping the project

1. Create a directory:
   - ***mkdir seai-recitation-1***
2. Navigate to the director:
   - ***cd seai-recitation-1***
3. Install pip:
   - *(sudo)* ***pip install virtualenv***
4. Setup your virtualenv:
   - ***virtualenv -p python3 venv***
5. Activate the virtualenv
   - ***source venv/bin/activate***
6. Install kafka library for python
   - *(sudo)* ***pip install kafka-python***

Exercise - Writing to kafka (gist)

Create a python script - ***touch producer.py***

```
from time import sleep
from json import dumps
from kafka import KafkaProducer

# Create a producer to write data to kafka
producer = KafkaProducer(bootstrap_servers=['localhost:9092'],
                         value_serializer=lambda x: dumps(x).encode('utf-8'))

# Write data via the producer
for e in range(10):
    data = {'number' : e}
    producer.send(topic='numtest-<andrewid>', value=data)
    sleep(1)
```

Exercise - Test output via kafkacat!

- ***kafkacat -b localhost -t numtest***

  Output

```
% Auto-selecting Consumer mode (use -P or -C to override)
{"number": 0}
{"number": 1}
{"number": 2}
{"number": 3}
{"number": 4}
{"number": 5}
{"number": 6}
{"number": 7}
{"number": 8}
{"number": 9}
% Reached end of topic numtest [0] at offset 10
```

Exercise - Reading from Kafka (gist)

```
from kafka import KafkaConsumer
from json import loads

# Create a consumer to read data from kafka
consumer = KafkaConsumer(
    'numtest-<andrewid>',
    bootstrap_servers=['localhost:9092'],
    # Read from the start of the topic; Default is latest
    auto_offset_reset='earliest'
)

# Prints all messages, again and again!
for message in consumer:
    # Default message.value type is bytes!
    print(loads(message.value))
```

Exercise - Reading (smartly) from Kafka

How would you make reads more fault tolerant ?

```
consumer = KafkaConsumer(
    'numtest',
    bootstrap_servers=['localhost:9092'],
    auto_offset_reset='earliest',
    # Consumer group id
    group_id='numtest-group-<andrewid>',
    # Commit that an offset has been read
    enable_auto_commit=True,
    # How often to tell Kafka, an offset has been read
    auto_commit_interval_ms=1000
)

# Prints messages once, then only new ones. Run again and see!
for message in consumer:
    print(loads(message.value))
```

Refer to official Kafka Python docs for more useful API methods and advanced use cases - KafkaConsumer

Thanks!

References

- https://www.youtube.com/watch?v=JalUUBKdcA0
- https://towardsdatascience.com/kafka-python-explained-in-10-lines-of-code-800e3e07dad1