

# Rockset과 Kafka 연결하기

## About

Rockset은 카프카에서 처리한 데이터를 일종의 데이터 베이스로 저장해서 BI를 이용한 시각화를 쉽게 하도록 도와주는 하나의 툴이다. 우선 이번 포스팅에서는 Rockset과 카프카를 kafka connect을 이용해서 연결하는 방법에 대해 알아본다.

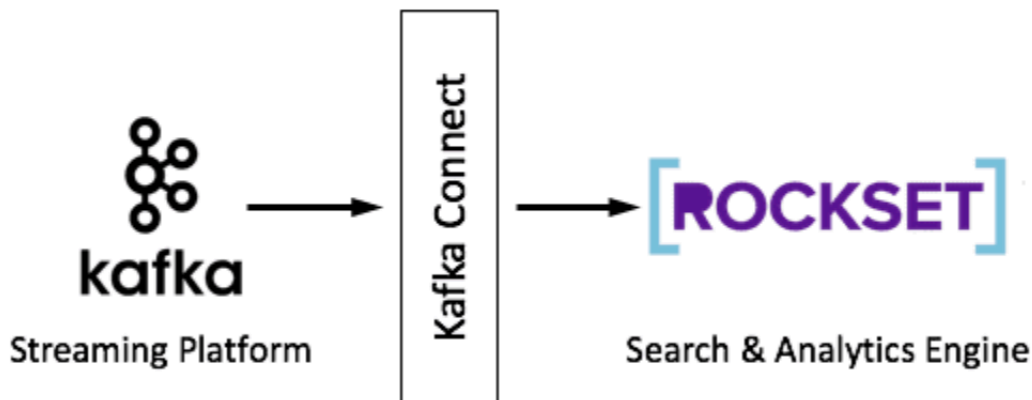
## Prerequisite

- 이미 카프카 클러스터를 구축했다고 가정한다.
- 여기서는 standalone 모드로 진행하므로 여러 서버가 아닌, 로컬 한 대의 카프카 브로커만 있어도 된다.

## Let's Start!

### Kafka Connect

어떤 툴과 다른 툴을 서로 연결할 때는 도식도를 보면 이해가 빠르다. 아래의 그림을 살펴보자.



카프카에서 스트리밍 데이터를 받으면 이를 카프카 커넥트를 이용해서 Rockset으로 넘겨준다. 데이터가 Rockset으로 넘어오면 SQL 쿼리를 이용해서 다양한 EDA를 수행할 수 있다.

원래 카프카는 실시간 데이터 처리보다, 중앙 허브 역할을 수행하는 데이터 플랫폼으로 개발되었다. 하지만 카프카의 성능이 워낙 좋고 데이터를 빨리 처리할 수 있다보니 실시간으로 데이터를 처리하고자 하는 수요에 맞춰서 실시간 데이터를 처리하는 플랫폼의 성격 또한 지니게 되었다.

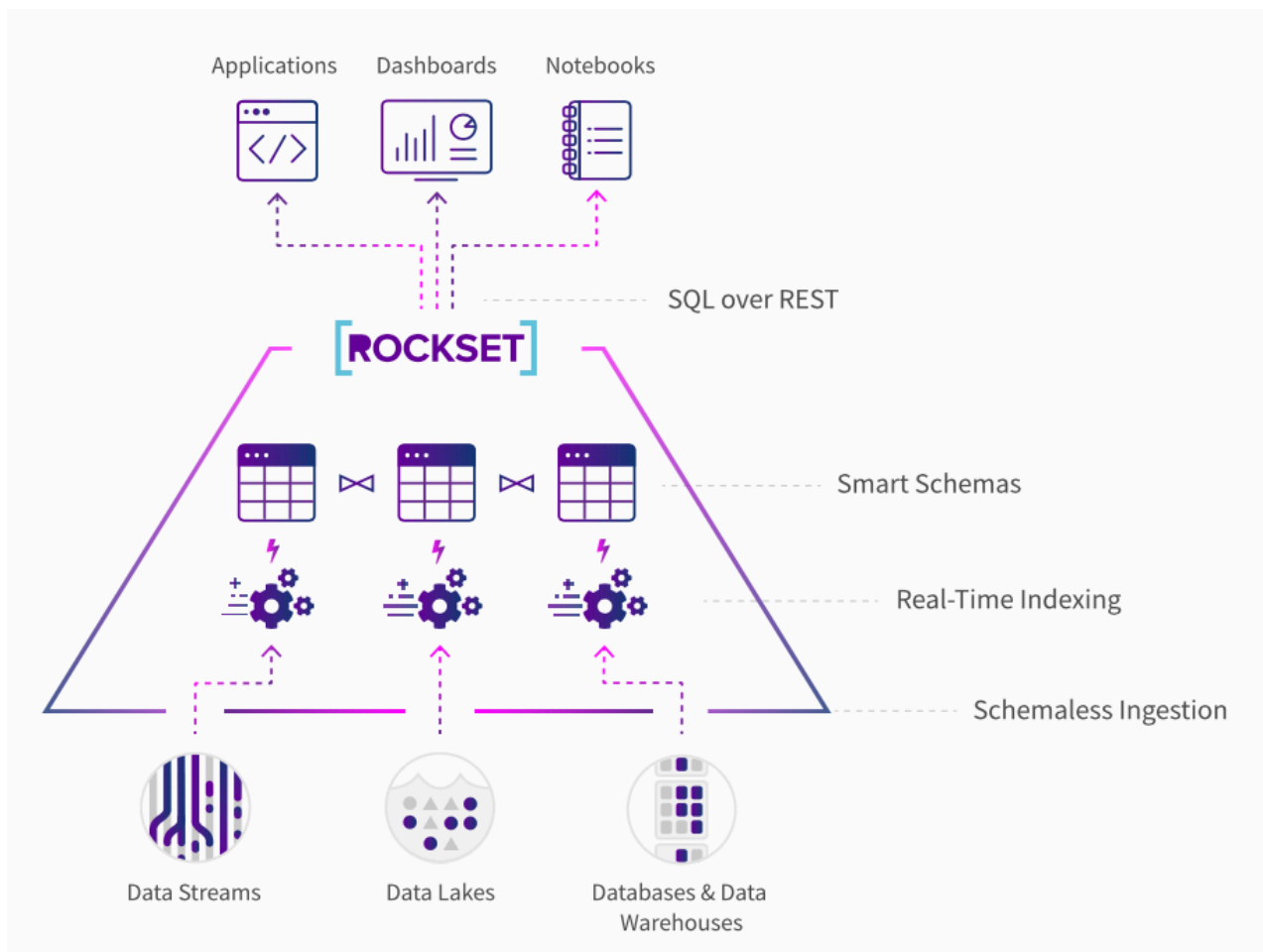
실시간 데이터를 처리하기 위해서 카프카 스트림즈를 사용할 수도 있는데 카프카 스트림즈는 다른 툴을 사용하기보다 직접 어플리케이션을 만들어서 실시간으로 처리하는 느낌이다. 반면 카프카 커넥트를 이용하면 다른 툴들, 예를 들어서 Tableau 등의 BI와 연계도 가능하기 때문에 목적에 따라서 카프카 스트림즈, 카프카 커넥트를 취사선택 하면 될 것으로 보인다.

## Rockset: As a consumer

카프카 브로커에서 받은 메시지를 받는 컨슈머 역할을 수행한다. 공식 API 문서의 정의를 살펴보자.

Rockset is a real-time analytics solution enabling low latency search, aggregations, and joins on massive semi-structured data, without operational burden. Rockset automatically indexes your data – structured, semi-structured, geo and time series data – for real-time search and analytics at scale.

뭔가 실시간 처리를 하는데 특화된 솔루션이라는 것 같다. 특히 어떤 외부의 data source로부터 데이터를 받아와서 SQL로 데이터 처리를 하는데 특화된 것으로 보인다. 도식도는 아래와 같다.



Data Streams, Data Lakes 또는 Database로부터 데이터를 받으면 Real-Time Indexing을 수행하고 결과를 SQL을 이용하여 대시보드나 기타 어플리케이션으로 전송해준다.

Rockset이 사용할 수 있는 Data Sources는 아래와 같다.

- Amazon DynamoDB
- Amazon Kinesis
- Amazon Redshift
- Amazon S3
- Apache Kafka
- Google Cloud Storage

- MongoDB Atlas

이 중에서 여기서는 Apache Kafka를 data source로 하여 tableau와 연결하는 것이다.

## Rockset Console 생성하기

아래 그림에서 Integration으로 들어가자

Yonsei University | Your Free Trial ends in 13 days. | Enter Payment Details | Docs | Live Chat | Log out

**[ROCKSET] Trial** Hide tips ^

**Ingest**  
 Create collections from your data sources.  
[Create Collection](#)

**Query**  
 Execute queries on your ingested data.  
[Run Query](#)

**Collaborate**  
 Invite your team members.  
[Invite Users](#)

Total Documents Ingested  
**20**

Total Queries Performed  
**12**

Active Documents Size  
**2.617 KiB**

**Recent Collections**

Name	Workspace	Source Type	Last Updated	Created By	Documents	Status
rockset-topic	commons	Apache Kafka	12:30 am	sbh0613@yonsei.ac.kr	20	Ready

integration을 추가해주자

Integrations ⓘ

Search:








[+ Add Integration](#)

Name	Description	Collections	Created By	Type	Created
rockset-topic		rockset-topic	sbh0613@yonsei.ac.kr	Apache Kafka	Feb 1

카프카를 선택하고 시작

## Integrate with an External Service

Once created, you can use an integration to load data into one or more Rockset collections from one of the services below. Don't see the service you're looking for? [Tell us about it.](#)

 MongoDB	 Amazon DynamoDB	 Apache Kafka	 Amazon Kinesis
 Amazon Redshift <span>Beta</span>	 Google Cloud Storage	 Amazon S3	

### Apache Kafka

An integration is required to securely connect Rockset with your Kafka brokers. Each integration can access multiple topics and can be used to create multiple Rockset collections.

Recommended: Map each Kafka topic to a separate collection and use SQL queries to join across collections.

#### Setup Time

15 mins

#### Permissions Required

Set up or modify a Kafka Connect Cluster

Cancel

Start

integration 이름을 써주고 kafka topic 이름도 써준다

## Set up your Apache Kafka Integration

An integration is required to securely connect Rockset with your Kafka brokers. Each integration can access multiple topics and can be used to create multiple Rockset collections.

Recommended: Map each Kafka topic to a separate collection and use SQL queries to join across collections.

### 1. Configure Rockset-Kafka Integration

#### Integration Name

Choose a descriptive name for your integration. You will require this to create collections as well as to view integration details later on.

Integration Name

rockset-topic

**Required:** Choose a descriptive name for your integration.

Description

description...

Optionally describe your integration.

#### Data Format

All Kafka topics in this integration must send data in the same format.

Select Data Format

☒ JSON ☐ AVRO

#### Kafka Topics

Select the JSON Kafka topics that this integration will access.

Recommended: Create separate integrations for topics that send data in a different formats.

Kafka Topic (JSON) 1

rockset-topic



Cancel

Save Integration and Continue

상황에 맞는 조건을 선택한다.

## 1. Configure Rockset-Kafka Integration

**Integration Name:** rockset-topic2  
**Data Format:** JSON  
**Topic Names:**  
rockset-topic

## 2. Install and configure Kafka Connect

### Tell us about your Kafka Setup

Answer questions about your Kafka Setup to get customized instructions.

**Where is your Kafka Cluster hosted?**

☒ On Premises

☐ Cloud

**Which platform?**

Confluent Platform

Apache Kafka Open  
Source

### Choose the setup type of your Kafka Connect Cluster

Select "New" if you don't have a pre-existing Kafka connect cluster.

Recommended: Use the standalone mode for development or to do quick POCs. Use distributed mode for production use cases.

[Talk to a Product Specialist](#)

☐ New — no pre-existing Kafka Connect cluster

☒ Standalone — for building prototypes

☐ Distributed — for a production environment

이미 카프카를 깔았기 때문에 여기서 step 1은 할 필요가 없다.

step 2에 카프카를 띄운 주소와 포트를 적어주면 된다. 만약에 여러대의 카프카 브로커를 구축했다면 차례대로 적어주면 된다. 모두 적었으면 Download Apache Kafka Connect Properties를 클릭하자.

또한 step 3의 Download Rockset Sink Connector와 Download Rockset Sink Connector Properties를 클릭하자.


이러면 총 세 개의 파일이 다운받아진다.

- connect-standalone.properties
- connect-rockset-sink.properties
- kafka-connect-rockset-1.2.0-jar-with-dependencies.jar

## Set up your Kafka Connect

The following instructions will help you set up your Kafka Connect cluster based on the configurations selected above.

### Step 1: Download Apache Kafka Connect

 [Download Apache Kafka Connect](#)


### Step 2: Configure Apache Kafka Connect

Address of Apache Kafka Broker 1

0.0.0.0:9092



 [Download Apache Kafka Connect Properties](#)


If you are using Confluent Cloud, you may need to add additional properties as outlined in the [Confluent Documentation](#) .

Place the downloaded properties file in the same directory as above, such that your directory tree resembles the following:

```
/
├── connect-standalone.properties
└── kafka_2.11-2.3.0/
```

### Step 3: Download and Configure Rockset Sink Connector

 [Download Rockset Sink Connector](#)

 [Download Rockset Sink Connector Properties](#)

Place the downloaded files in the same directory as above, such that your directory tree resembles the following:

```
/
├── connect-standalone.properties
├── connect-rockset-sink.properties
├── kafka-connect-rockset-1.2.0-jar-with-dependencies.jar
└── kafka_2.11-2.3.0/
```

connect-standalone.properties는 원래 kafka config 디렉토리에 있는 설정이다. connect-rockset-sink-properties는 rockset과 kafka를 연결해주는 설정인 것 같다.

## 여기서 주의할 점!

connect-standalone.properties의 plugin.path를 절대경로로 설정해줘야 한다!!!!

```
bootstrap.servers=0.0.0.0:9092
key.converter=org.apache.kafka.connect.json.JsonConverter
value.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=false
value.converter.schemas.enable=false
offset.storage.file.filename=/tmp/connect.offsets
offset.flush.interval.ms=10000
plugin.path=/usr/local/Cellar/kafka/config2/kafka-connect-rockset-1.2.0-jar-with-dependencies.jar
```

상대경로로 지정하면 인식을 못함

여태까지 kafka-connect를 연결하기 위해 세 개의 파일을 다운 받고 수정했다.

- connect-standalone.properties
- connect-rockset-sink.properties
- kafka-connect-rockset-1.2.0-jar-with-dependencies.jar

## kafka, zookeeper 실행

```
▶ jps
99862 Jps
40744 QuorumPeerMain
98218 Kafka
```

jps를 쳤을 때 위와 같이 두 개의 java process가 뜨면 된다.

## connect-standalone 실행

```
▶ ./bin/connect-standalone.sh ./config2/connect-standalone.pr
operties ./config2/connect-rockset-sink.properties █
```

알맞은 경로를 지정하고 실행해준다.

```
▶ jps
2114 Jps
1315 ConnectStandalone
40744 QuorumPeerMain
98218 Kafka
```

kafka-connect가 잘 띄워졌음을 확인할 수 있다.

## kafka topic이랑 rockset이랑 연동되는지 확인하기

우선 rockset에서 받아오는 데이터 형태를 JSON으로 지정했기 때문에 kafka topic으로 데이터 보낼때에도 JSON 형식으로 보내야한다. 근데 여기서 serialization도 해야한다. 이를 파이썬에서 아래와 같이 구현하였다.



```

from kafka import KafkaProducer
import json

msg={"id":"test","tel":"11111","regDate":"20200201"}
producer = KafkaProducer(
    acks=1, compression_type='gzip',
    value_serializer=lambda m: json.dumps(msg).encode('ascii'),
    bootstrap_servers='localhost:9092'
    # bootstrap_servers = 'test-broker01:9092,test-broker02:9092'
)
for i in range(10):
    producer.send('rockset-topic', {'key': 'value'})
producer.close()

```

이렇게 데이터를 보내면 kafka-standalone에서는 아래와 같이 로그가 뜬다.

```

[2021-02-02 21:48:13,318] INFO [Consumer clientId=consumer-8, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,318] INFO [Consumer clientId=consumer-5, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,318] INFO [Consumer clientId=consumer-5, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,318] INFO [Consumer clientId=consumer-8, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,318] INFO [Consumer clientId=consumer-6, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,319] INFO [Consumer clientId=consumer-9, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,320] INFO [Consumer clientId=consumer-6, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,320] INFO [Consumer clientId=consumer-1, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,320] INFO [Consumer clientId=consumer-9, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,321] INFO [Consumer clientId=consumer-3, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,321] INFO [Consumer clientId=consumer-7, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,321] INFO [Consumer clientId=consumer-7, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,321] INFO [Consumer clientId=consumer-3, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,322] INFO [Consumer clientId=consumer-10, groupId=connect-rockset-topic] Successfully joined group with generation 12 (org.apache.kafka.clients.consumer.internals.AbstractCoordinator:450)
[2021-02-02 21:48:13,322] INFO [Consumer clientId=consumer-10, groupId=connect-rockset-topic] Setting newly assigned partitions [] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:48:13,322] INFO [Consumer clientId=consumer-1, groupId=connect-rockset-topic] Setting newly assigned partitions [rockset-topic-0] (org.apache.kafka.clients.consumer.internals.ConsumerCoordinator:289)
[2021-02-02 21:55:42,818] INFO WorkerSinkTask{id=rockset-topic-0} Committing offsets asynchronously using sequence number 45: {rockset-topic-0=OffsetAndMetadata{offset=151, leaderEpoch=null, metadata=''}} (org.apache.kafka.connect.runtime.WorkerSinkTask:346)
[2021-02-02 21:55:52,819] INFO WorkerSinkTask{id=rockset-topic-0} Committing offsets asynchronously using sequence number 46: {rockset-topic-0=OffsetAndMetadata{offset=161, leaderEpoch=null, metadata=''}} (org.apache.kafka.connect.runtime.WorkerSinkTask:346)

```

잘은 모르겠는데 어쨌든 성공했다고 한다.

Rockset의 콘솔로 들어가서 확인해보면 데이터를 아래와 같이 SQL로 뽑을 수 있다.

The screenshot displays the Rockset Query Editor interface. On the left, the 'Workspace' is set to 'commons' and 'Collections' shows '0 of 2 selected'. The main area contains a SQL query in a text editor:

```
1 SELECT
2 *
3 FROM
4 commons."rockset-topic"
5 WHERE
6 tel = '11111'
7 ORDER BY
8 regDate
```

Below the query editor, there are tabs for 'Results', 'Parameters', 'Performance', and 'Develop'. The 'Results' tab is active, showing a table with 4 columns: 'regDate', 'tel', '\_event\_time', and 'id'. The table contains 20 rows of data. To the right of the query editor, there is a 'Query Log' panel showing a list of executed queries with timestamps and SQL snippets.

regDate	tel	_event_time	id
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'
'20200201'	'11111'	'2021-02-02T12:55:41.125000Z'	'test'

여기까지 했으니, 이제 Rockset이랑 Tableau랑 연결하는 것까지 하면, Kafka - Kafka Connect - Rockset - Tableau의 연결고리가 완성된다!

## Reference

- 카프카 커넥트에 관한 Rockset 공식 문서: <https://docs.rockset.com/apache-kafka/>
- 카프카와 Rockset을 같이 사용하는 것에 대한 confluent 블로그 글: <https://www.confluent.io/blog/analytcs-with-apache-kafka-and-rockset/>
- kafka-connect-rockset의 github: <https://github.com/rockset/kafka-connect-rockset> 애는 별 도움이 되지 못했으나 그래도 첨부함
- json 형태의 데이터 인코딩하기: <https://programmerdaddy.tistory.com/90>