1. airflow vm 설치
2. Kafka 설치

---

## *STEP 1*

## Kafka Resource

#KAFKA_RESOURCE

### docker-compose.yaml

```
version: '2'

networks:
  common-network:
    external: true

services:
  zookeeper:
    image: confluentinc/cp-zookeeper:6.1.15
    hostname: zookeeper
    container_name: zookeeper
    ports:
      - "2181:2181"
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
    networks:
      - common-network

  broker:
    image: confluentinc/cp-kafka:6.1.15
    hostname: broker
    container_name: broker
    depends_on:
      - zookeeper
    ports:
      - "19092:19092"
      - "9092:9092"
    environment:
      KAFKA_BROKER_ID: 1
      KAFKA_ZOOKEEPER_CONNECT: 'zookeeper:2181'
      KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: 'INTERNAL:PLAINTEXT,EXTERNAL:PLAINTEXT'
      KAFKA_ADVERTISED_LISTENERS: 'INTERNAL://broker:9092,EXTERNAL://133.186.244.202:19092'
      KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
      KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
      KAFKA_TRANSACTION_STATE_LOG_MIN_ISR: 1
      KAFKA_TRANSACTION_STATE_LOG_REPLICATION_FACTOR: 1
      KAFKA_GROUP_INITIAL_REBALANCE_DELAY_MS: 0
    networks:
      - common-network
```

## Spark Resource

#SPARK_RESOURCE

### Dockerfile

```
FROM bitnami/spark:3.3.0
```

```
USER root

# Install necessary packages
RUN apt-get update && \
    apt-get install -y --no-install-recommends \
        vim \
        curl \
        netcat-openbsd && \
    rm -rf /var/lib/apt/lists/*

# Download Kafka client JAR
RUN curl -o /opt/bitnami/spark/jars/kafka-clients-3.3.0.jar https://repo1.maven.org/maven2/org/apache/kafka/kafka-clients/3.3.0/kafka-clients-3.3.0.jar

# Download Spark Kafka connector JAR
RUN curl -o /opt/bitnami/spark/jars/spark-token-provider-kafka-0-10_2.12-3.3.0.jar
https://repo1.maven.org/maven2/org/apache/spark/spark-token-provider-kafka-0-10_2.12/3.3.0/spark-token-provider-kafka-0-10_2.12-3.3.0.jar

# Download Spark SQL Kafka connector JAR
RUN curl -o /opt/bitnami/spark/jars/spark-sql-kafka-0-10_2.12-3.3.0.jar
https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.12/3.3.0/spark-sql-kafka-0-10_2.12-3.3.0.jar

RUN curl -o /opt/bitnami/spark/jars/commons-pool2-2.11.0.jar
https://repo1.maven.org/maven2/org/apache/commons/commons-pool2/2.11.0/commons-pool2-2.11.0.jar

RUN pip install py4j==0.10.9.5

USER 1001
```

```
sudo docker build -t seunghyejeong/spark:1.0
sudo docker push seunghyejeong/spark:1.0
```

## docker-compose.yaml

```
# Copyright VMware, Inc.
# SPDX-License-Identifier: APACHE-2.0

version: '2'

networks:
  common-network:
    external: true

services:
  spark:
    image: seunghyejeong/spark:1.0
    environment:
      - SPARK_MODE=master
      - SPARK_RPC_AUTHENTICATION_ENABLED=no
      - SPARK_RPC_ENCRYPTION_ENABLED=no
      - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
      - SPARK_SSL_ENABLED=no
      - SPARK_USER=spark
      - SPARK_MASTER_OPTS="-Dspark.rpc.message.maxSize=512"
    ports:
      - '8080:8080'
      - '7077:7077'
    networks:
      - common-network

  spark-worker:
    image: seunghyejeong/spark:1.0
    environment:
      - SPARK_MODE=worker
      - SPARK_MASTER_URL=spark://{MASTER_IP}:7077
      - SPARK_WORKER_MEMORY=1G
      - SPARK_WORKER_CORES=1
      - SPARK_RPC_AUTHENTICATION_ENABLED=no
      - SPARK_RPC_ENCRYPTION_ENABLED=no
      - SPARK_LOCAL_STORAGE_ENCRYPTION_ENABLED=no
      - SPARK_SSL_ENABLED=no
      - SPARK_USER=spark
      - SPARK_WORKER_OPTS="-Dspark.rpc.message.maxSize=512"
    networks:
      - common-network
```

## Create Docker Network

```
$docker network create common-network
c6e200e1d23e453357553975f326462c98dd93bc11d7b958b2a9f7c9ab837a6d
```

## Deploy

```
docker compose up -d
```

## *STEP 2*

#KAFKA_CREATE_TOPIC_CMD

## Create topic *kafka*

```
sudo docker compose exec broker kafka-topics --create --topic devices --bootstrap-server broker:9092 --replication-
factor 1 --partitions 1
```

- 확인하기

```
sudo docker compose exec broker kafka-topics --describe --topic devices --bootstrap-server broker:9092
```

## Consumer topic *spark*

- Spark Worker 접속

```
docker exec -ti f2ec8a60af7a bash
```

- spark.py 생성

```
vim spark.py
```

> *spark.py*

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import from_json
from pyspark.sql.types import StringType, StructField, StructType, ArrayType, LongType

spark = SparkSession \
    .builder \
    .appName("pipeline") \
    .config("spark.streaming.stopGracefullyOnShutdown", True) \
    .config('spark.jars.packages', 'org.apache.spark:spark-sql-kafka-0-10_2.12:3.3.0') \
    .config("spark.sql.shuffle.partitions", 4) \
    .master("local[*]") \
    .getOrCreate()

# Define Kafka connection properties
```

```
kafka_params = {
    "kafka.bootstrap.servers": "125.6.40.186:19092",
    "subscribe": "devices",
    "startingOffsets": "earliest"
}

# Define JSON Schema

json_schema = StructType([
    StructField('customerId', StringType(), True),
    StructField('data', StructType([
        StructField('devices', ArrayType(StructType([
            StructField('deviceId', StringType(), True),
            StructField('measure', StringType(), True),
            StructField('status', StringType(), True),
            StructField('temperature', LongType(), True)
        ]), True), True)
    ]), True),
    StructField('eventId', StringType(), True),
    StructField('eventOffset', LongType(), True),
    StructField('eventPublisher', StringType(), True),
    StructField('eventTime', StringType(), True)
])

# Read Kafka messages
streaming_df = spark \
    .readStream \
    .format("kafka") \
    .options(**kafka_params) \
    .load()

# Parse JSON messages
json_df = streaming_df.selectExpr("CAST(value AS STRING) AS value") \

json_expanded_df = json_df.withColumn("value", from_json(json_df["value"], json_schema)).select("value.*")

#print(json_schema)
#json_expanded_df.printSchema()

from pyspark.sql.functions import explode, col


exploded_df = json_expanded_df \
    .select("customerId", "eventId", "eventOffset", "eventPublisher", "eventTime", "data") \
    .withColumn("devices", explode("data.devices")) \
    .drop("data")

#exploded_df.printSchema()
#exploded_df.show

flattened_df = exploded_df \
    .selectExpr("customerId", "eventId", "eventOffset", "eventPublisher", "cast(eventTime as timestamp) as 
eventTime",
                "devices.deviceId as deviceId", "devices.measure as measure",
                "devices.status as status", "devices.temperature as temperature")

#flattened_df.printSchema()


# Aggregate the dataframes to find the average temparature
# per Customer per device throughout the day for SUCCESS events
from pyspark.sql.functions import to_date, avg

agg_df = flattened_df.where("STATUS = 'SUCCESS'") \
    .withColumn("eventDate", to_date("eventTime", "yyyy-MM-dd")) \
    .groupBy("customerId","deviceId","eventDate") \
    .agg(avg("temperature").alias("avg_temp"))

# Write the output to console sink to check the output
```

```
writing_df = agg_df.writeStream \
    .format("console") \
    .option("checkpointLocation","checkpoint_dir") \
    .outputMode("complete") \
    .start()

writing_df.awaitTermination()
```

## *STEP 3*

## 메세지 Produce, Consume 동작 확인

### kafka

#KAFKA_PRODUCER_CMD
#KAFKA_EXAMPLE_TOPIC

- producer 실행하기

```
sudo docker exec -ti broker bash
```

```
kafka-console-producer --topic devices --bootstrap-server broker:9092
```

- 예제 토픽

```
{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId":
"CI00103", "data": {"devices": [{"deviceId": "D001", "temperature": 15, "measure": "C", "status": "ERROR"},
{"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05
11:13:53.643364"}
```

### spark

```
python spark.py
```

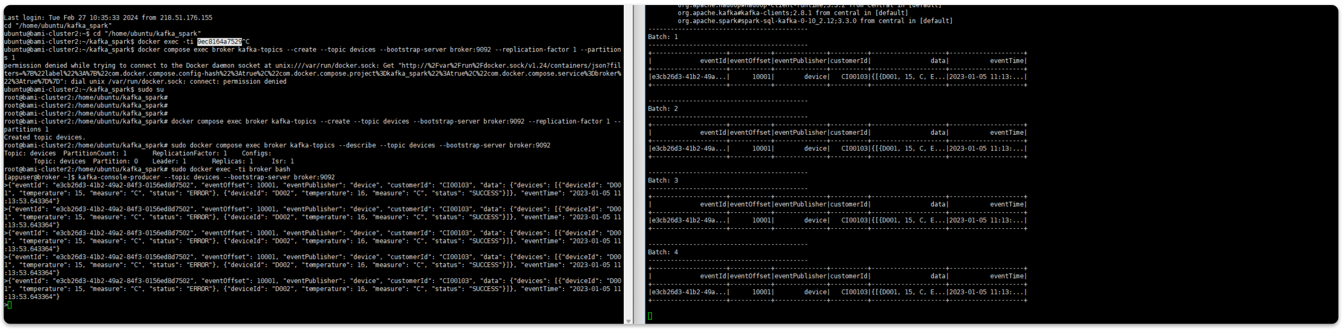
> *output*

```
:: loading settings :: url = jar:file:/opt/bitnami/spark/jars/ivy-
2.5.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /opt/bitnami/spark/.ivy2/cache
The jars for the packages stored in: /opt/bitnami/spark/.ivy2/jars
org.apache.spark#spark-sql-kafka-0-10_2.12 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-3b6554bd-b894-47ae-9cc3-5acbcd98ad5b;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.12;3.3.0 in central
        found org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.3.0 in central
        found org.apache.kafka#kafka-clients;2.8.1 in central
        found org.lz4#lz4-java;1.8.0 in central
        found org.xerial.snappy#snappy-java;1.1.8.4 in central
        found org.slf4j#slf4j-api;1.7.32 in central
        found org.apache.hadoop#hadoop-client-runtime;3.3.2 in central
        found org.spark-project.spark#unused;1.0.0 in central
        found org.apache.hadoop#hadoop-client-api;3.3.2 in central
        found commons-logging#commons-logging;1.1.3 in central
        found com.google.code.findbugs#jsr305;3.0.0 in central
        found org.apache.commons#commons-pool2;2.11.1 in central
downloading https://repo1.maven.org/maven2/org/apache/spark/spark-sql-kafka-0-10_2.12/3.3.0/spark-sql-kafka-0-
10_2.12-3.3.0.jar ...
        [SUCCESSFUL ] org.apache.spark#spark-sql-kafka-0-10_2.12;3.3.0!spark-sql-kafka-0-10_2.12.jar (447ms)
downloading https://repo1.maven.org/maven2/org/apache/spark/spark-token-provider-kafka-0-10_2.12/3.3.0/spark-token-
provider-kafka-0-10_2.12-3.3.0.jar ...
        [SUCCESSFUL ] org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.3.0!spark-token-provider-kafka-0-
```

```
10_2.12.jar (258ms)
downloading https://repo1.maven.org/maven2/org/apache/kafka/kafka-clients/2.8.1/kafka-clients-2.8.1.jar ...
        [SUCCESSFUL ] org.apache.kafka#kafka-clients;2.8.1!kafka-clients.jar (640ms)
downloading https://repo1.maven.org/maven2/com/google/code/findbugs/jsr305/3.0.0/jsr305-3.0.0.jar ...
        [SUCCESSFUL ] com.google.code.findbugs#jsr305;3.0.0!jsr305.jar (251ms)
downloading https://repo1.maven.org/maven2/org/apache/commons/commons-pool2/2.11.1/commons-pool2-2.11.1.jar ...
        [SUCCESSFUL ] org.apache.commons#commons-pool2;2.11.1!commons-pool2.jar (253ms)
downloading https://repo1.maven.org/maven2/org/spark-project/spark/unused/1.0.0/unused-1.0.0.jar ...
        [SUCCESSFUL ] org.spark-project.spark#unused;1.0.0!unused.jar (251ms)
downloading https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-client-runtime/3.3.2/hadoop-client-runtime-
3.3.2.jar ...
        [SUCCESSFUL ] org.apache.hadoop#hadoop-client-runtime;3.3.2!hadoop-client-runtime.jar (1514ms)
downloading https://repo1.maven.org/maven2/org/lz4/lz4-java/1.8.0/lz4-java-1.8.0.jar ...
        [SUCCESSFUL ] org.lz4#lz4-java;1.8.0!lz4-java.jar (259ms)
downloading https://repo1.maven.org/maven2/org/xerial/snappy/snappy-java/1.1.8.4/snappy-java-1.1.8.4.jar ...
        [SUCCESSFUL ] org.xerial.snappy#snappy-java;1.1.8.4!snappy-java.jar(bundle) (284ms)
downloading https://repo1.maven.org/maven2/org/slf4j/slf4j-api/1.7.32/slf4j-api-1.7.32.jar ...
        [SUCCESSFUL ] org.slf4j#slf4j-api;1.7.32!slf4j-api.jar (252ms)
downloading https://repo1.maven.org/maven2/org/apache/hadoop/hadoop-client-api/3.3.2/hadoop-client-api-3.3.2.jar
...
        [SUCCESSFUL ] org.apache.hadoop#hadoop-client-api;3.3.2!hadoop-client-api.jar (1016ms)
downloading https://repo1.maven.org/maven2/commons-logging/commons-logging/1.1.3/commons-logging-1.1.3.jar ...
        [SUCCESSFUL ] commons-logging#commons-logging;1.1.3!commons-logging.jar (252ms)
:: resolution report :: resolve 17431ms :: artifacts dl 5696ms
        :: modules in use:
        com.google.code.findbugs#jsr305;3.0.0 from central in [default]
        commons-logging#commons-logging;1.1.3 from central in [default]
        org.apache.commons#commons-pool2;2.11.1 from central in [default]
        org.apache.hadoop#hadoop-client-api;3.3.2 from central in [default]
        org.apache.hadoop#hadoop-client-runtime;3.3.2 from central in [default]
        org.apache.kafka#kafka-clients;2.8.1 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.12;3.3.0 from central in [default]
        org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.3.0 from central in [default]
        org.lz4#lz4-java;1.8.0 from central in [default]
        org.slf4j#slf4j-api;1.7.32 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.8.4 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   12  |   12  |   12  |   0   ||   12  |   12  |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-3b6554bd-b894-47ae-9cc3-5acbcd98ad5b
        confs: [default]
        12 artifacts copied, 0 already retrieved (56631kB/87ms)
24/02/27 02:07:52 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-
java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
24/02/27 02:08:00 WARN ResolveWriteToStream: Temporary checkpoint location created which is deleted normally when
the query didn't fail: /tmp/temporary-59644b8a-43d4-44e8-8dc1-100eaed02d95. If it's required to delete it under any
circumstances, please set spark.sql.streaming.forceDeleteTempCheckpointLocation to true. Important to know deleting
temp checkpoint folder is best effort.
24/02/27 02:08:00 WARN ResolveWriteToStream: spark.sql.adaptive.enabled is not supported in streaming
DataFrames/Datasets and will be disabled.
-------------------------------------------
Batch: 0
-------------------------------------------
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|             eventId|eventOffset|eventPublisher|customerId|                data|           eventTime|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|e3cb26d3-41b2-49a...|      10001|        device|   CI00103|{[{D001, 15, C, E...|2023-01-05 11:13:...|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
```

## *FINAL*

# 실시간으로 로그가 확인 가능함



```
Last login: Tue Feb 27 10:35:33 2024 from 218.51.176.155
cd "/home/ubuntu/kafka_spark"
ubuntu@bami-cluster2:~$ cd "/home/ubuntu/kafka_spark"
ubuntu@bami-cluster2:~/kafka_spark$ docker exec -ti 9ec8164a7529^C
ubuntu@bami-cluster2:~/kafka_spark$ docker compose exec broker kafka-topics --create --topic devices --bootstrap-server broker:9092 --replication-factor 1 --partition
s 1
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/json?fil
ters=%7B%22label%22%3A%7B%22com.docker.compose.config-hash%22%3Atrue%2C%22com.docker.compose.project%3Dkafka_spark%22%3Atrue%2C%22com.docker.compose.service%3Dbroker%
22%3Atrue%7D%7D": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@bami-cluster2:~/kafka_spark$ sudo su
root@bami-cluster2:/home/ubuntu/kafka_spark#
root@bami-cluster2:/home/ubuntu/kafka_spark#
root@bami-cluster2:/home/ubuntu/kafka_spark#
root@bami-cluster2:/home/ubuntu/kafka_spark# docker compose exec broker kafka-topics --create --topic devices --bootstrap-server broker:9092 --replication-factor 1 --
partitions 1
Created topic devices.
root@bami-cluster2:/home/ubuntu/kafka_spark# sudo docker compose exec broker kafka-topics --describe --topic devices --bootstrap-server broker:9092
Topic: devices  PartitionCount: 1       ReplicationFactor: 1    Configs:
        Topic: devices  Partition: 0    Leader: 1       Replicas: 1     Isr: 1
root@bami-cluster2:/home/ubuntu/kafka_spark# sudo docker exec -ti broker bash
[appuser@broker ~]$ kafka-console-producer --topic devices --bootstrap-server broker:9092
>{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId": "CI00103", "data": {"devices": [{"deviceId": "D00
1", "temperature": 15, "measure": "C", "status": "ERROR"}, {"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05 11
:13:53.643364"}
>{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId": "CI00103", "data": {"devices": [{"deviceId": "D00
1", "temperature": 15, "measure": "C", "status": "ERROR"}, {"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05 11
:13:53.643364"}
>{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId": "CI00103", "data": {"devices": [{"deviceId": "D00
1", "temperature": 15, "measure": "C", "status": "ERROR"}, {"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05 11
:13:53.643364"}
>{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId": "CI00103", "data": {"devices": [{"deviceId": "D00
1", "temperature": 15, "measure": "C", "status": "ERROR"}, {"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05 11
:13:53.643364"}
>{"eventId": "e3cb26d3-41b2-49a2-84f3-0156ed8d7502", "eventOffset": 10001, "eventPublisher": "device", "customerId": "CI00103", "data": {"devices": [{"deviceId": "D00
1", "temperature": 15, "measure": "C", "status": "ERROR"}, {"deviceId": "D002", "temperature": 16, "measure": "C", "status": "SUCCESS"}]}, "eventTime": "2023-01-05 11
:13:53.643364"}
>
```

```
        org.apache.hadoop#hadoop-client-runtime;3.3.2 from central in [default]
        org.apache.kafka#kafka-clients;2.8.1 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.12;3.3.0 from central in [default]
-------------------------------------------
Batch: 1
-------------------------------------------
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|             eventId|eventOffset|eventPublisher|customerId|                data|           eventTime|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|e3cb26d3-41b2-49a...|      10001|        device|   CI00103|{[{D001, 15, C, E...|2023-01-05 11:13:...|
+--------------------+-----------+--------------+----------+--------------------+--------------------+

-------------------------------------------
Batch: 2
-------------------------------------------
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|             eventId|eventOffset|eventPublisher|customerId|                data|           eventTime|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|e3cb26d3-41b2-49a...|      10001|        device|   CI00103|{[{D001, 15, C, E...|2023-01-05 11:13:...|
+--------------------+-----------+--------------+----------+--------------------+--------------------+

-------------------------------------------
Batch: 3
-------------------------------------------
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|             eventId|eventOffset|eventPublisher|customerId|                data|           eventTime|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|e3cb26d3-41b2-49a...|      10001|        device|   CI00103|{[{D001, 15, C, E...|2023-01-05 11:13:...|
+--------------------+-----------+--------------+----------+--------------------+--------------------+

-------------------------------------------
Batch: 4
-------------------------------------------
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|             eventId|eventOffset|eventPublisher|customerId|                data|           eventTime|
+--------------------+-----------+--------------+----------+--------------------+--------------------+
|e3cb26d3-41b2-49a...|      10001|        device|   CI00103|{[{D001, 15, C, E...|2023-01-05 11:13:...|
+--------------------+-----------+--------------+----------+--------------------+--------------------+

```