날씨데이터 실측치와 예측치의 차이 비교분석

DATA ENGINEER

3조 야수의 심장

김정호 이윤재 이현범 한유정

목차

프로젝트 개요

빅데이터 기술 함양

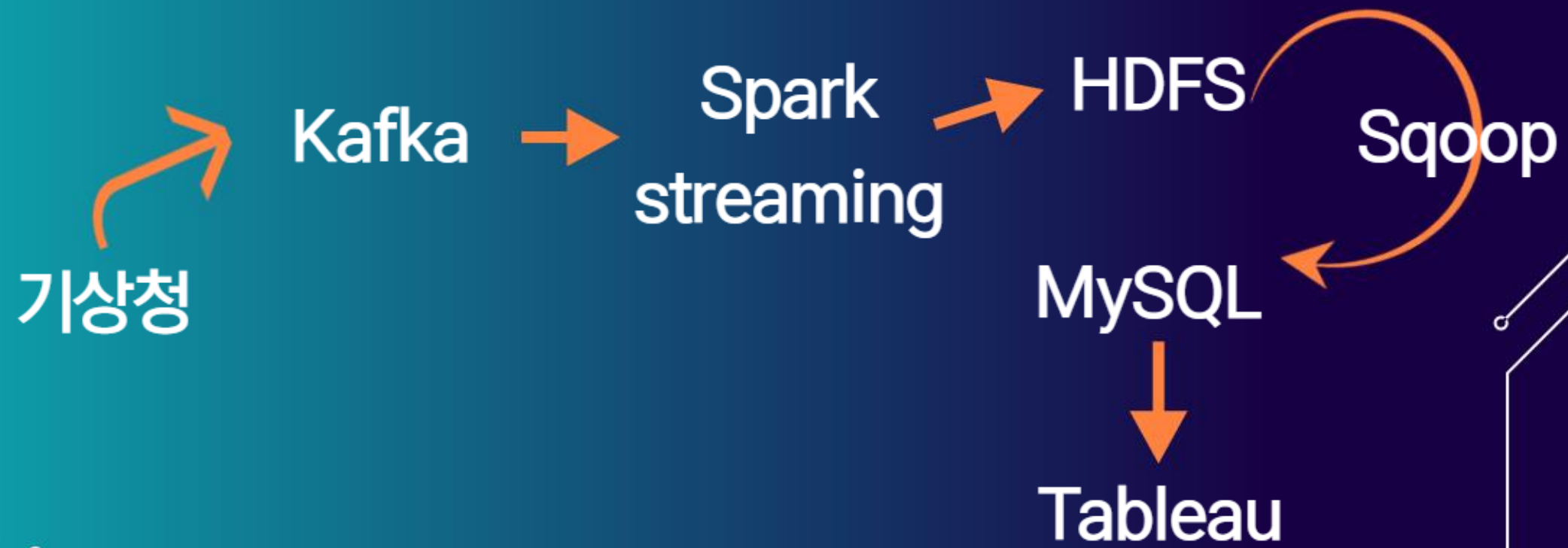
환경구축
local / AWS

수집
정형 / 비정형 데이터

전송
Kafka, Streaming

저장
HDFS, MySQL

PIPELINE



환경구축

DATAENGINEER

시스템 환경변수 편집

```
vim /etc/profile.d/hadoop.sh

export HADOOP_HOME=/opt/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop/
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop/

vim /etc/profile.d/spark.sh
export SPARK_HOME=/opt/spark
export PATH=$SPARK_HOME/bin:$PATH

vim /etc/profile.d/zookeeper.sh
export ZOOKEEPER_HOME=/opt/zookeeper
export PATH=$PATH:$ZOOKEEPER_HOME/bin

vim /etc/profile.d/kafka.sh
export KAFKA_HOME=/opt/kafka
export PATH=$PATH:$KAFKA_HOME/bin
```

zoo.cfg 설정

```
cd /opt/zookeeper/conf
cp -r zoo_sample.cfg zoo.cfg
```

kafka server.properties 설정

```
cd /opt/kafka/config/
vim server.properties
zookeeper.connect=localhost:2181/localhost
```

zookeeper, kafka 실행

```
zkServer.sh start
kafka-server-start.sh $KAFKA_HOME/config/server.properties
```

환경구축 데이터 전처리

```python
import pandas as pd
import pymysql
import warnings

warnings.filterwarnings(action='ignore')
pymysql.install_as_MySQLdb()

import MySQLdb
from sqlalchemy import create_engine
import sqlalchemy


file= pd.read_csv('./gayang_1_24_result.csv',encoding='utf-8')
```

```python
for i in file['r_hour'].index:
    if file['r_hour'][i] >= 24:
        if file['r_hour'][i] - 24 >= 1:
            file['avg(value)'][i+3] = (file['avg(value)'][i+3] + file['avg(value)'][i])/2
            file['day'][i] = 'na'
        else:
            file['r_hour'][i] = 0
            file['day'][i] += 1
    else:
        pass

file.dropna(axis=0,inplace=True)
```

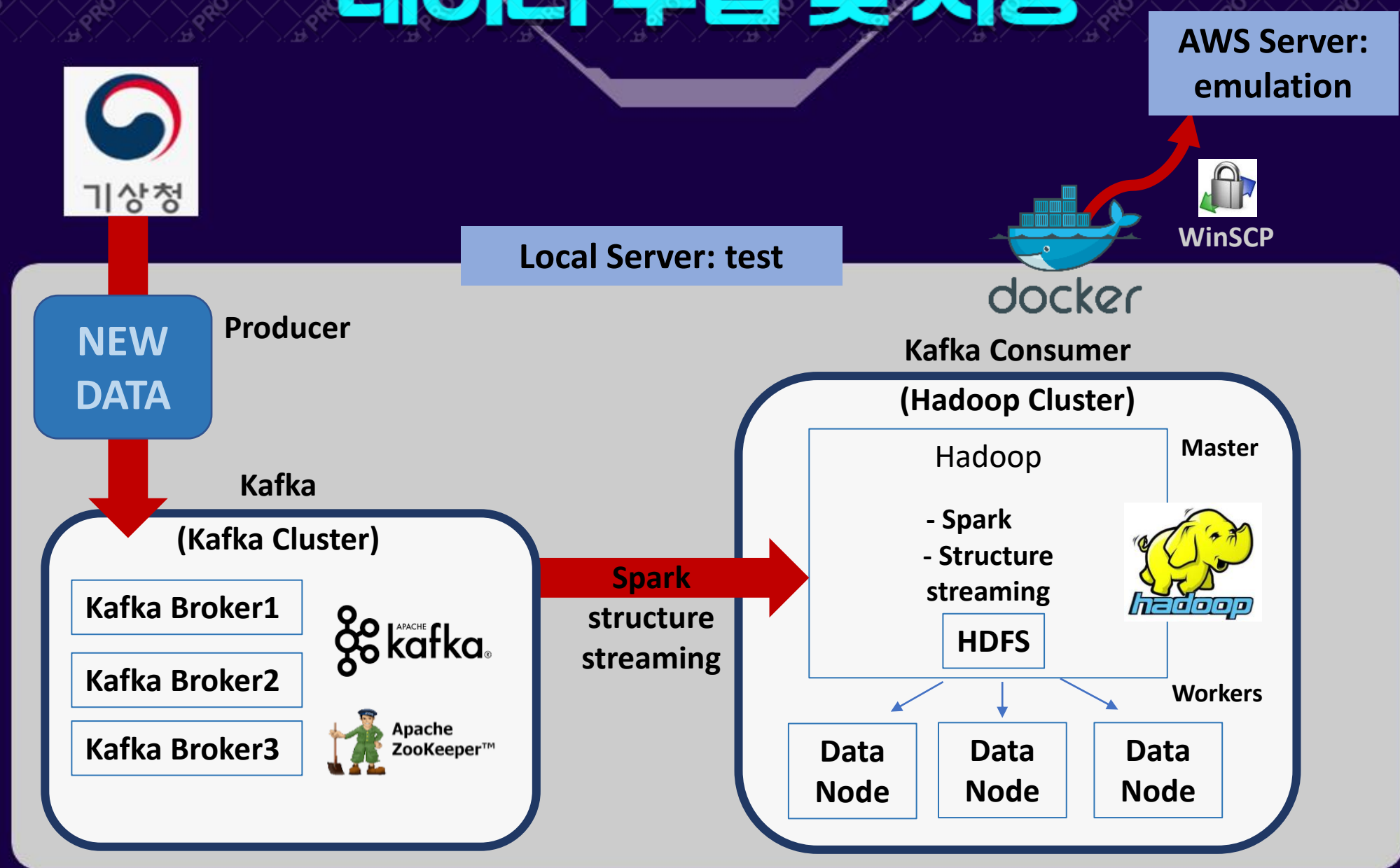
```python
idx = file[file['day']=='na'].index
file = file.drop(idx)
```

```python
engine = create_engine("mysql+mysqldb://Serena H:"+""+"hanu1004@127.0.0.1/celcius", encoding='utf-8')
conn = engine.connect()


file.to_sql(name='celcius_24_result', con=engine, if_exists='append',index = False,dtype = {
    'year':sqlalchemy.types.INT(),
'month':sqlalchemy.types.INT(),
'day':sqlalchemy.types.INT(),
'r_hour':sqlalchemy.types.INT(),
'avg(value)': sqlalchemy.types.FLOAT(6)
})

conn.close()
```

DATAENGINEER

```mysql
use celcius;
select year, month, day, r_hour, avg(value) from celcius group by r_hour, year, month, day;

select *, value from celcius_24_result left join celcius_real.value
on celcius_24_result.year = celcius_real.year and celcius_24_result.month = celcius_real.month and
celcius_24_result.day = celcius_real.day;
```

데이터 수집 및 저장


AWS Server: emulation
기상청
WinSCP
Local Server: test
docker
NEW DATA
Producer
Kafka Consumer
(Hadoop Cluster)
Kafka
(Kafka Cluster)
Hadoop
Master
- Spark
- Structure streaming
HDFS
Kafka Broker1
Kafka Broker2
Kafka Broker3
kafka
Apache ZooKeeper™
Spark structure streaming
Workers
Data Node
Data Node
Data Node

데이터 수집 및 저장

AWS Server Kafka → Spark → HDFS

```
": "26.000000 "}, {" format: day": " 15", "hour": "1930", "forecast": "+6", "val
ue location:60_127 Start : 20210701 ": "27.000000 "}, {" format: day": " 15", "h
our": "2030", "forecast": "+1", "value location:60_127 Start : 20210701 ": "22.0
00000 "}, {" format: day": " 15", "hour": "2030", "forecast": "+2", "value locat
ion:60_127 Start : 20210701 ": "22.000000 "}, {" format: day": " 15", "hour": "2
030", "forecast": "+3", "value location:60_127 Start : 20210701 ": "23.000000 "}
, {" format: day": " 15", "hour": "2030", "forecast": "+4", "value location:60_1
27 Start : 20210701 ": "26.000000 "}, {" format: day": " 15", "hour": "2030", "f
orecast": "+5", "value location:60_127 Start : 20210701 ": "27.000000 "}, {" for
mat: day": " 15", "hour": "2030", "forecast": "+6", "value location:60_127 Start
 : 20210701 ": "28.000000 "}, {" format: day": " 15", "hour": "2130", "forecast"
: "+1", "value location:60_127 Start : 20210701 ": "22.000000 "}, {" format: day
": " 15", "hour": "2130", "forecast": "+2", "value loc
701 ": "23.000000 "}, {" format: day": " 15", "hour":
"value location:60_127 Start : 20210701 ": "25.000000
, "hour": "2130", "forecast": "+4", "value location:60
27.000000 "}, {" format: day": " 15", "hour": "2130",
ocation:60_127 Start : 20210701 ": "28.000000 "}, {" f
: "2130", "forecast": "+6", "value location:60_127 Sta
0 "}, {" format: day": " 15", "hour": "2230", "forecas
60_127 Start : 20210701 ": "23.000000 "}, {" format: d
, "forecast": "+2", "value location:60_127 Start : 202
 format: day": " 15", "hour": "2230", "forecast": "+3"
tart : 20210701 ": "26.000000 "}, {" format: day": " 1
ast": "+4", "value location:60_127 Start : 20210701 ":
 day": " 15", "hour": "2230", "forecast": "+5", "value
0210701 ": "30.000000 "}, {" format: day": " 15", "hou
6", "value location:60_127 Start : 20210701 ": "30.000
 15", "hour": "2330", "forecast": "+1", "value locatio
": "25.000000 "}, {" format: day": " 15", "hour": "233
ue location:60_127 Start : 20210701 ": "26.000000 "},
our": "2330", "forecast": "+3", "value location:60_127
00000 "}, {" format: day": " 15", "hour": "2330", "for
ion:60_127 Start : 20210701 ": "30.000000 "}, {" forma
330", "forecast": "+5", "value location:60_127 Start :
, {" format: day": " 15", "hour": "2330", "forecast":
27 Start : 20210701 ": "31.000000 "}, {" format: day":
ast": null, "value location:60_127 Start : 20210701 ":
Processed a total of 27 messages
root@ip-172-31-15-42:/opt/kafka#
```

root@ip-172-31-15-42: /opt/kafka

```
ocalhost:9092 --topic Jim_topic --from-beginning
root@ip-172-31-15-42:/opt/kafka# bin/kafka-topics.sh --list --zookeeper localhos
t:2181
Jim_Topic
Jim_topic
__consumer_offsets
test
root@ip-172-31-15-42:/opt/kafka# bin/kafka-console-consumer.sh --bootstrap-serve
r localhost:9092 --topic Jim_Topic --from-beginning
[" format: day,hour,value location:60_127 Start : 20210701 "]
[" 1, 0000, 27.299999 "]
[" 1, 0100, 27.600000 "]
[" 1, 0200, 29.400000 "]
[" 1, 0300, 30.299999 "]
[" 1, 0400, 30.600000 "]
[" 1, 0500, 30.700001 "]
[" 1, 0600, 30.600000 "]
[" 1, 0700, 30.200001 "]
[" 1, 0800, 29.700001 "]
[" 1, 0900, 28.700001 "]
[" 1, 1000, 27.600000 "]
[" 1, 1100, 26.799999 "]
[" 1, 1200, 26.200001 "]
[" 1, 1300, 25.600000 "]
```

데이터 수집 및 저장

Kafka Producer 코드

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

producer = KafkaProducer(acks=0, compression_type='gzip',
                         bootstrap_servers=['172.17.0.6:9092','172.17.0.7:9092','172.17.0.8:9092'],
                         value_serializer=lambda x: dumps(x).encode('utf-8')
                        )
with open('/home/lab07/교남동_기온_20210701_20210815.csv', 'r') as file:
    reader = csv.reader(file, delimiter = '₩t')
    for messages in reader:
        if 'Start' in ''.join(messages):
            continue
        producer.send('hdtest', ','.join(messages))
        producer.flush()
```

Structure Streaming → HDFS 저장

```
df = spark.readStream.format("kafka").option("kafka.bootstrap.servers",'172.17.0.6:9092,172.17.0.7:9092,172.17.0.8:9092') ₩
.option("subscribe","hdtest").load()
```

```
df = df.selectExpr("CAST(value AS STRING)")
```

```
df.writeStream.trigger(processingTime='5 seconds').outputMode("append").format("text") ₩
.option("path", "/streaming/out").option("checkpointLocation", "/streaming/checkpointLocation").start().awaitTermination()
```

데이터 수집 및 저장

HDFS 저장 결과

```
root@master:/opt/hadoop# bin/hdfs dfs -ls /streaming/out
Found 14 items
drwxr-xr-x   - root supergroup          0 2021-08-20 03:49 /streaming/out/_spark_metadata
-rw-r--r--   3 root supergroup      15475 2021-08-20 03:49 /streaming/out/part-00000-25334965-5849-4d3d-8996-169294e796c1-c000.txt
-rw-r--r--   3 root supergroup          0 2021-08-20 03:48 /streaming/out/part-00000-5fa3dcdc-23c0-4594-8e86-8f4c8f5befe6-c000.txt
-rw-r--r--   3 root supergroup      12034 2021-08-20 03:49 /streaming/out/part-00000-88823619-4220-4e85-8d03-a6a262bb4aa0-c000.txt
-rw-r--r--   3 root supergroup      15689 2021-08-20 03:49 /streaming/out/part-00001-553092fb-65a9-4a9c-9889-5e250ededfa8-c000.txt
-rw-r--r--   3 root supergroup      10977 2021-08-20 03:49 /streaming/out/part-00001-f8ba96c0-7af6-48ee-a464-314f09b56547-c000.txt
-rw-r--r--   3 root supergroup      12034 2021-08-20 03:49 /streaming/out/part-00002-39181702-2666-4576-9318-6fd0f3a0e684-c000.txt
-rw-r--r--   3 root supergroup      14478 2021-08-20 03:49 /streaming/out/part-00002-53eee7f8-12b3-450f-a83a-c9c5048da2dd-c000.txt
-rw-r--r--   3 root supergroup      14772 2021-08-20 03:49 /streaming/out/part-00003-1429f494-b302-4862-808b-7f9ecdc7e345-c000.txt
-rw-r--r--   3 root supergroup      11929 2021-08-20 03:49 /streaming/out/part-00003-736ff819-eba5-4e06-bd9f-e344287e766c-c000.txt
-rw-r--r--   3 root supergroup      12429 2021-08-20 03:49 /streaming/out/part-00004-21044489-ece9-4d9e-973b-0b05c57e5e4a-c000.txt
-rw-r--r--   3 root supergroup      15704 2021-08-20 03:49 /streaming/out/part-00004-607ca0cc-9f20-486f-8f4c-a695c8e7653f-c000.txt
-rw-r--r--   3 root supergroup      12705 2021-08-20 03:49 /streaming/out/part-00005-b43101c9-6320-4580-b0a8-80f52f2ab438-c000.txt
-rw-r--r--   3 root supergroup      14786 2021-08-20 03:49 /streaming/out/part-00005-b69e44fc-d64c-4952-8d06-0805c98c3848-c000.txt
root@master:/opt/hadoop# bin/hdfs dfs -cat  /streaming/out/part-00000-25334965-5849-4d3d-8996-169294e796c1-c000.txt
" 1,0030,+2,28.000000 "
" 1,0130,+3,30.000000 "
" 1,0330,+1,29.000000 "
" 1,0330,+2,29.000000 "
" 1,0430,+4,30.000000 "
" 1,0430,+6,29.000000 "
" 1,0730,+2,29.000000 "
" 1,0930,+4,26.000000 "
" 1,1030,+6,25.000000 "
" 1,1130,+3,25.000000 "
" 1,1130,+5,25.000000 "
" 1,1230,+6,24.000000 "
" 1,1330,+2,24.000000 "
" 1,1530,+5,23.000000 "
" 1,1630,+1,24.000000 "
" 1,1930,+1,23.000000 "
" 1,1930,+2,23.000000 "
```

Kafka에서 데이터를 보내고
spark structured streaming에서 받아서 HDFS에 저장

데이터 수집 및 저장

HDFS에 json 형식 저장 결과

```
root@master:/opt/hadoop# bin/hdfs dfs -ls /streaming/out
Found 32 items
drwxr-xr-x   - root supergroup          0 2021-08-20 10:34 /streaming/out/_spark_metadata
-rw-r--r--   3 root supergroup      88832 2021-08-20 10:34 /streaming/out/part-00000-0271062d-5699-45fe-90ba-9f929f3610a6-c000.json
-rw-r--r--   3 root supergroup     348381 2021-08-20 10:34 /streaming/out/part-00000-1f89aeef-faa3-480e-9f4b-951f604b3f49-c000.json
-rw-r--r--   3 root supergroup      69807 2021-08-20 10:34 /streaming/out/part-00000-4326e48c-0e53-4d00-a76b-605b99525d01-c000.json
-rw-r--r--   3 root supergroup      97224 2021-08-20 10:34 /streaming/out/part-00000-74c7e16a-d6d1-4ce5-a762-d5eaea002786-c000.json
-rw-r--r--   3 root supergroup      15652 2021-08-20 10:34 /streaming/out/part-00000-78fbee39-0bca-473f-86f3-e5d7f54edf32-c000.json
-rw-r--r--   3 root supergroup          0 2021-08-20 10:33 /streaming/out/part-00000-9aede235-4dd8-47c7-9e11-99940211661e-c000.json
-rw-r--r--   3 root supergroup      14588 2021-08-20 10:34 /streaming/out/part-00001-0c711ecd-1e2b-4956-919b-efcb784ed4ae-c000.json
-rw-r--r--   3 root supergroup     340602 2021-08-20 10:34 /streaming/out/part-00001-316b6140-273b-4acd-8dbe-fa4630288dcb-c000.json
-rw-r--r--   3 root supergroup      94551 2021-08-20 10:34 /streaming/out/part-00001-6adce98f-e9da-4a2d-aaf0-9f04a3329341-c000.json
-rw-r--r--   3 root supergroup      82841 2021-08-20 10:34 /streaming/out/part-00001-af2aa1c0-d8e7-4c16-b528-e16e379fe9cf-c000.json
-rw-r--r--   3 root supergroup      80580 2021-08-20 10:34 /streaming/out/part-00001-d1ab7539-da6b-4244-8840-131ac537cdca-c000.json
-rw-r--r--   3 root supergroup      98775 2021-08-20 10:34 /streaming/out/part-00002-1510f261-c66e-40b9-a1bd-6e34e6377c85-c000.json
-rw-r--r--   3 root supergroup      13371 2021-08-20 10:34 /streaming/out/part-00002-2a8d688f-d322-4817-a7a7-8323e9e11f01-c000.json
-rw-r--r--   3 root supergroup      93070 2021-08-20 10:34 /streaming/out/part-00002-48e3e073-6aea-4fce-964d-748aa299c4c1-c000.json
-rw-r--r--   3 root supergroup      80064 2021-08-20 10:34 /streaming/out/part-00002-4dda43a8-a989-4560-bf0b-34540548a759-c000.json
-rw-r--r--   3 root supergroup     337020 2021-08-20 10:34 /streaming/out/part-00002-705418b3-9bfd-45f2-91ad-72b110a59388-c000.json
-rw-r--r--   3 root supergroup      95932 2021-08-20 10:34 /streaming/out/part-00003-208c36ac-7ac9-4afd-a19e-37b76afa5a94-c000.json
-rw-r--r--   3 root supergroup     347143 2021-08-20 10:34 /streaming/out/part-00003-3a8689c3-3895-46a0-8cfc-d92173d1a45d-c000.json
-rw-r--r--   3 root supergroup      86866 2021-08-20 10:34 /streaming/out/part-00003-74954ee0-a8c6-4ab2-a615-fe99f27b4241-c000.json
-rw-r--r--   3 root supergroup      81664 2021-08-20 10:34 /streaming/out/part-00003-955ba2b3-7728-4a8b-9b1d-a60743aeff0a-c000.json
-rw-r--r--   3 root supergroup      11490 2021-08-20 10:34 /streaming/out/part-00003-e502621b-4b6b-4e35-b183-58841d259009-c000.json
-rw-r--r--   3 root supergroup      92947 2021-08-20 10:34 /streaming/out/part-00004-311d14d4-8258-4397-9bda-f103b8847531-c000.json
-rw-r--r--   3 root supergroup      77856 2021-08-20 10:34 /streaming/out/part-00004-608a27ff-c2b2-4f2a-a30b-715a57b1b9c0-c000.json
-rw-r--r--   3 root supergroup      93309 2021-08-20 10:34 /streaming/out/part-00004-7dd5a6ed-292b-4864-8d68-f5c6fc249197-c000.json
-rw-r--r--   3 root supergroup      13780 2021-08-20 10:34 /streaming/out/part-00004-8b077e08-9caa-4c84-9630-7281f089d8be-c000.json
-rw-r--r--   3 root supergroup     330182 2021-08-20 10:34 /streaming/out/part-00004-9a89e25a-bbbd-4b43-9d33-83949f4e7d60-c000.json
-rw-r--r--   3 root supergroup      10177 2021-08-20 10:34 /streaming/out/part-00005-21fb2c2a-ed23-47ea-85d5-4addba85ce8e-c000.json
-rw-r--r--   3 root supergroup      87736 2021-08-20 10:34 /streaming/out/part-00005-7b525703-db52-4323-938e-f90fa3b9ffd7-c000.json
-rw-r--r--   3 root supergroup     340622 2021-08-20 10:34 /streaming/out/part-00005-81371dfa-fcdf-48f9-af43-17236ad34d65-c000.json
-rw-r--r--   3 root supergroup      95140 2021-08-20 10:34 /streaming/out/part-00005-b52edc3f-163d-4a00-b0e0-b16734fea5ed-c000.json
-rw-r--r--   3 root supergroup      70915 2021-08-20 10:34 /streaming/out/part-00005-d4d6d884-e153-4d63-9609-8d4860696db0-c000.json
root@master:/opt/hadoop# bin/hdfs dfs -cat /streaming/out/part-00005-21fb2c2a-ed23-47ea-85d5-4addba85ce8e-c000.json
```

데이터 수집 및 저장

json 파일 open

```
root@master:/opt/hadoop# bin/hdfs dfs -cat /streaming/out/part-00005-21fb2c2a-ed23-47ea-85d5-4addba85ce8e-c000.json
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"1\", \"r_hour\": \"7\", \"avg(value)\": \"25.39999962\", \"real_value\": \"2
2\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"1\", \"r_hour\": \"12\", \"avg(value)\": \"24.25\", \"real_value\": \"21.1\"
}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"1\", \"r_hour\": \"17\", \"avg(value)\": \"22.39999962\", \"real_value\": \"
20.8\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"1\", \"r_hour\": \"21\", \"avg(value)\": \"24\", \"real_value\": \"21.3\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"1\", \"r_hour\": \"0\", \"avg(value)\": \"24.25\", \"real_value\": \"24\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"2\", \"r_hour\": \"5\", \"avg(value)\": \"26.20000013\", \"real_value\": \"2
3.7\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"2\", \"r_hour\": \"10\", \"avg(value)\": \"24.5\", \"real_value\": \"22.9\"}
"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"2\", \"r_hour\": \"17\", \"avg(value)\": \"23.53333346\", \"real_value\": \"
23.2\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"2\", \"r_hour\": \"21\", \"avg(value)\": \"23\", \"real_value\": \"24.5\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"3\", \"r_hour\": \"11\", \"avg(value)\": \"25.5\", \"real_value\": \"24.9\"}
"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"3\", \"r_hour\": \"20\", \"avg(value)\": \"23.96666654\", \"real_value\": \"
23.9\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"3\", \"r_hour\": \"0\", \"avg(value)\": \"27\", \"real_value\": \"28.9\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"4\", \"r_hour\": \"3\", \"avg(value)\": \"30\", \"real_value\": \"30.7\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"4\", \"r_hour\": \"22\", \"avg(value)\": \"24.25\", \"real_value\": \"24.4\"
}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"5\", \"r_hour\": \"5\", \"avg(value)\": \"28.10000038\", \"real_value\": \"2
9.9\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"5\", \"r_hour\": \"19\", \"avg(value)\": \"22.90000057\", \"real_value\": \"
23.1\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"5\", \"r_hour\": \"0\", \"avg(value)\": \"23\", \"real_value\": \"21.6\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"6\", \"r_hour\": \"11\", \"avg(value)\": \"22.69999949\", \"real_value\": \"
22\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"6\", \"r_hour\": \"13\", \"avg(value)\": \"21.75\", \"real_value\": \"21.3\"
}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"6\", \"r_hour\": \"21\", \"avg(value)\": \"20\", \"real_value\": \"19.8\"}"}
{"value":"{\"year\": \"2018\", \"month\": \"7\", \"day\": \"6\", \"r_hour\": \"23\", \"avg(value)\": \"21.16666667\", \"real_value\": \"
22.4\"}"}
```

DATAENGINEER

데이터 수집 및 저장

MySQL → Sqoop
(테이블 정보 수신)

```
root@master:/opt/sqoop# sqoop list-databases --connect jdbc:mysql://3.34.160.1:3306 --username root --password '12345678'
Warning: /opt/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /opt/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /opt/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /opt/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2021-08-20 00:19:55,514 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7
2021-08-20 00:19:55,553 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
2021-08-20 00:19:55,674 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatic
ally registered via the SPI and manual loading of the driver class is generally unnecessary.
mysql
information_schema
performance_schema
sys
```

Hadoop → MySQL 어려움

DATAENGINEER

데이터 수집 및 저장

HDFS → Spark SQL→ MySQL

```
mysql> select * from weather
    -> ;
```

year	month	day	r_hour	avg_value	real_value
2020	6	11	23	24	26.1
2020	6	12	3	26.66666667	25.1
2020	6	12	5	25.8	25.8
2020	3	28	6	10.83333333	11.9
2020	6	12	12	23.5	23.9
2020	3	28	8	11	10.7
2020	6	12	18	22.5	21.2
2020	3	28	15	5.333333333	3.5
2020	6	13	4	28	29.6
2020	6	13	7	28.75	28.6
2020	3	29	13	8.5	8.7
2020	6	13	16	24	23.4
2020	3	29	18	7.333333333	7.3
2020	6	13	18	23.5	23.2
2020	3	30	19	5	4.6
2020	6	13	21	22.83333333	22.7
2020	6	14	5	27.4	27.5
2020	3	30	22	5.75	7.1
2020	6	14	6	27.83333333	27.4
2020	6	14	10	24.5	23.3
2020	6	14	17	19.4	19.6
2020	6	14	18	19.33333333	19.1
2020	3	31	6	18.83333333	20.4
2020	6	15	2	24.91666667	27.5
2020	6	15	13	21.5	21.4
2020	6	15	17	20.2	20
2020	6	15	18	20.33333333	19.7
2020	6	16	10	27	25.5
2020	6	16	13	23.25	23.7
2020	6	16	19	20	20.5
2020	3	31	8	18.8	18.1
2020	6	16	0	22	25.5
2020	6	17	5	27.4	27.7
2020	3	31	11	15.6	12.6
2020	6	17	6	27.83333333	26.3
2020	3	31	16	10	10.2

weather

Don't Limit

```
1 • SELECT * FROM multi_db.weather;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	month	day	r_hour	avg_value	real_value
2020	6	11	23	24	26.1
2020	6	12	3	26.66666667	25.1
2020	6	12	5	25.8	25.8
2020	3	28	6	10.83333333	11.9
2020	6	12	12	23.5	23.9
2020	3	28	8	11	10.7
2020	6	12	18	22.5	21.2
2020	3	28	15	5.333333333	3.5
2020	6	13	4	28	29.6

DATAENGINEER

데이터 수집 및 저장

Docker GUI

Hadoop Spark Container, Kafka Container
모두 SSH 통신을 통해서 데이터를 주고 받아서
분산 환경을 구축

DATAENGINEER

데이터 수집 및 저장

Spark Web UI

Spark Master at spark://master:7077

URL: spark://master:7077
Alive Workers: 3
Cores in use: 12 Total, 12 Used
Memory in use: 86.7 GiB Total, 3.0 GiB Used
Resources in use:
Applications: 1 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20210819155704-172.17.0.3-41575	172.17.0.3:41575	ALIVE	4 (4 Used)	28.9 GiB (1024.0 MiB Used)	
worker-20210819155704-172.17.0.4-46789	172.17.0.4:46789	ALIVE	4 (4 Used)	28.9 GiB (1024.0 MiB Used)	
worker-20210819155704-172.17.0.5-40959	172.17.0.5:40959	ALIVE	4 (4 Used)	28.9 GiB (1024.0 MiB Used)	

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20210819162803-0002 (kill)	myFirstApp	12	1024.0 MiB		2021/08/19 16:28:03	root	RUNNING	18 min

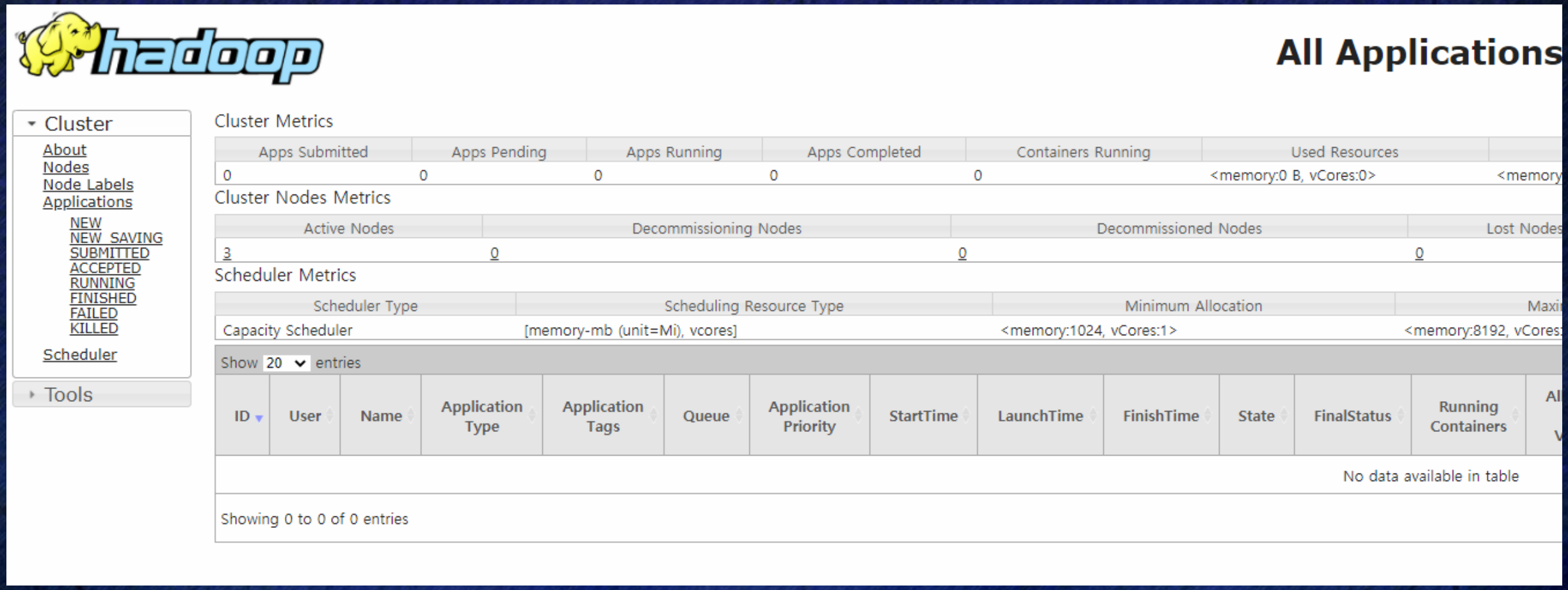
Completed Applications (2)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
app-20210819162735-0001	myFirstApp	12	1024.0 MiB		2021/08/19 16:27:35	root	FINISHED	19 s
app-20210819155744-0000	myFirstApp	12	1024.0 MiB		2021/08/19 15:57:44	root	FINISHED	30 min

DATAENGINEER

데이터 수집 및 저장

YARN Web UI



데이터 수집 및 저장

HDFS Web UI

Hadoop | Overview | Datanodes | Datanode Volume Failures | Snapshot | Startup Progress | Utilities

Overview 'master:9000' (✔active)

Started:	Thu Aug 19 15:56:18 +0900 2021
Version:	3.3.1, ra3b9c37a397ad4188041dd80621bdeefc46885f2
Compiled:	Tue Jun 15 14:13:00 +0900 2021 by ubuntu from (HEAD detached at release-3.3.1-RC3)
Cluster ID:	CID-66c89eee-78f1-4014-a218-44edf64e648c
Block Pool ID:	BP-2142135247-172.17.0.2-1629256639537

Summary

Security is off.

Safemode is off.

186 files and directories, 187 blocks (187 replicated blocks, 0 erasure coded block groups) = 373 total filesystem object(s).

Heap Memory used 196.14 MB of 828 MB Heap Memory. Max Heap Memory is 6.64 GB.

Non Heap Memory used 58.9 MB of 60.31 MB Commited Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	2.84 TB
Configured Remote Capacity:	0 B
DFS Used:	16.57 GB (0.57%)

데이터 수집 및 저장

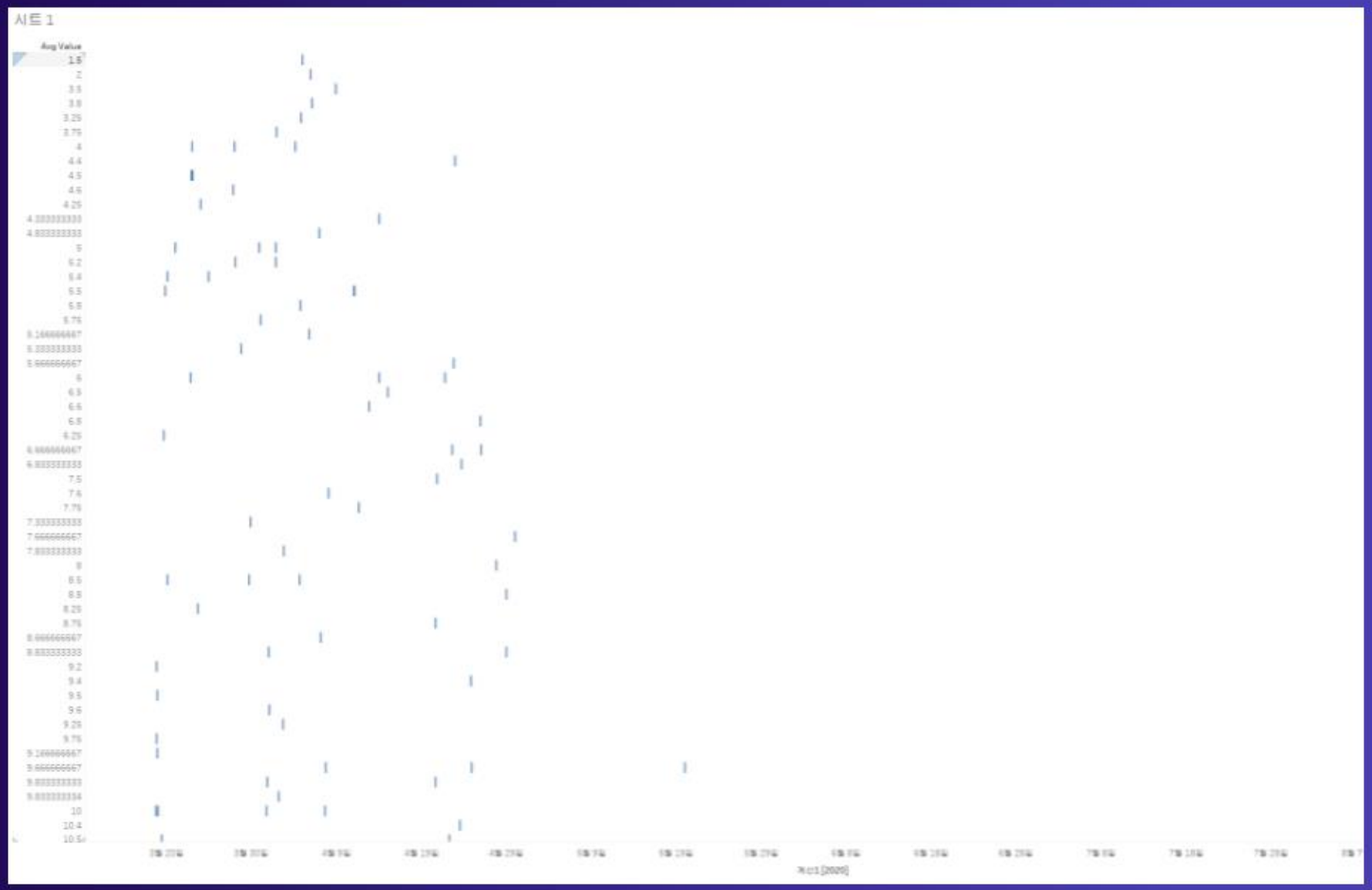
HDFS Web UI

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	root	supergroup	0 B	Aug 20 03:49	0	0 B	_spark_metadata
-rw-r--r--	root	supergroup	15.11 KB	Aug 20 03:49	3	128 MB	part-00000-25334965-5849-4d3d-8996-169294e796c1-c000.txt
-rw-r--r--	root	supergroup	0 B	Aug 20 03:48	3	128 MB	part-00000-5fa3dcdc-23c0-4594-8e86-8f4c8f5befe6-c000.txt
-rw-r--r--	root	supergroup	11.75 KB	Aug 20 03:49	3	128 MB	part-00000-88823619-4220-4e85-8d03-a6a262bb4aa0-c000.txt
-rw-r--r--	root	supergroup	15.32 KB	Aug 20 03:49	3	128 MB	part-00001-553092fb-65a9-4a9c-9889-5e250ededfa8-c000.txt
-rw-r--r--	root	supergroup	10.72 KB	Aug 20 03:49	3	128 MB	part-00001-f8ba96c0-7af6-48ee-a464-314f09b56547-c000.txt
-rw-r--r--	root	supergroup	11.75 KB	Aug 20 03:49	3	128 MB	part-00002-39181702-2666-4576-9318-6fd0f3a0e684-c000.txt
-rw-r--r--	root	supergroup	14.14 KB	Aug 20 03:49	3	128 MB	part-00002-53eee7f8-12b3-450f-a83a-c9c5048da2dd-c000.txt
-rw-r--r--	root	supergroup	14.43 KB	Aug 20 03:49	3	128 MB	part-00003-1429f494-b302-4862-808b-7f9ecdc7e345-c000.txt
-rw-r--r--	root	supergroup	11.65 KB	Aug 20 03:49	3	128 MB	part-00003-736ff819-eba5-4e06-bd9f-e344287e766c-c000.txt
-rw-r--r--	root	supergroup	12.14 KB	Aug 20 03:49	3	128 MB	part-00004-21044489-ece9-4d9e-973b-0b05c57e5e4a-c000.txt
-rw-r--r--	root	supergroup	15.34 KB	Aug 20 03:49	3	128 MB	part-00004-607ca0cc-9f20-486f-8f4c-a695c8e7653f-c000.txt
-rw-r--r--	root	supergroup	12.41 KB	Aug 20 03:49	3	128 MB	part-00005-b43101c9-6320-4580-b0a8-80f52f2ab438-c000.txt
-rw-r--r--	root	supergroup	14.44 KB	Aug 20 03:49	3	128 MB	part-00005-b69e44fc-d64c-4952-8d06-0805c98c3848-c000.txt

Showing 1 to 14 of 14 entries

Previous 1 Next

시각화



프로젝트 후기

김정호

처음 다뤄보는것들이 많아서 어려웠고
팀원분들이 너무 고생해주셔서 마무리가
가능했습니다

더 열심히 다음 프로젝트 준비를 하겠습니다

이윤재

맨땅에 헤딩하는격으로 시작한 프로젝트였기
때문에 많이 힘들었지만,
다른 팀원들 모두 열심히 해주셔서 도움이 많이
되었습니다

이현범

사전지식이 거의 없는 상태여서 구현이 많이
힘들었지만 어떻게든 한 것 같다

팀장으로써 거의 한 게 없는데 열심히 해준
팀원들에게 고마운 마음 뿐이다

한유정

데이터 수집과 저장에 관한 새로운 기술들을
적용하는 프로젝트에 참여 할 생각에 들떴지만,
생각 이상으로 적용하는 것이 어려웠다.
그래서 이번 프로젝트에서 함께 참여하는 느낌이
적어서 아쉽고 미안한 마음이 컸다. 또 한편으로는
열심히 해주는 팀원분들 모두에게 존경심도 들었다.

그리고 어렵게 따라 가면서 데이터 수집과 저장의
과정에 대한 개념 이해를 많이 할 수 있어서 좋았다.

날씨데이터 실측치와 예측치의 차이 비교분석

THANK YOU

3조 야수의 심장

김정호 이윤재 이현범 한유정