실시간 적재 기능 구현

실시간 적재 파일럿 실행 3단계-실시간 적재 기능 구현

카프카 Spout 기능 구현
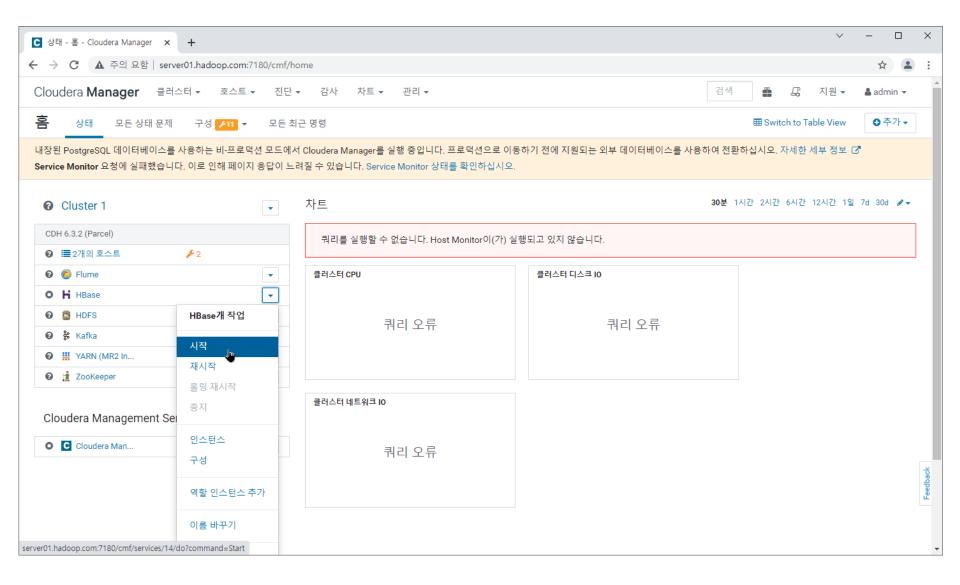
예제 5.2 카프카 Spout 소스 - SmartCarDriverTopology.java

```java
TopologyBuilder driverCarTopologyBuilder = new TopologyBuilder();

String zkHost = "server02.hadoop.com:2181";                                    ❶
BrokerHosts brkBost = new ZkHosts(zkHost);


String topicName = "SmartCar-Topic";
String zkPathName = "/SmartCar-Topic";


SpoutConfig spoutConf = new SpoutConfig(brkBost, topicName, zkPathName, UUID.randomUUID().
toString());                                                                   ❷
spoutConf.scheme = new SchemeAsMultiScheme(new StringScheme());
KafkaSpout kafkaSpout = new KafkaSpout(spoutConf);


driverCarTopologyBuilder.setSpout("kafkaSpout", kafkaSpout, 1);                 ❸


driverCarTopologyBuilder.setBolt("splitBolt", new SplitBolt(),1).allGrouping("kafkaSpout");
driverCarTopologyBuilder.setBolt("esperBolt", new EsperBolt(),1).allGrouping("kafkaSpout");  ❹
```

실시간 적재 파일럿 실행 3단계-실시간 적재 기능 구현

❶ 카프카에 접속하기 위해 먼저 카프카와 연결된 주키퍼의 정보를 설정하고, BrokerHosts 객체를 생성한다.

❷ KafkaSpout 객체를 생성하는 단계다. 앞서 정의한 주키퍼 연결 정보와 카프카의 토픽 정보, 주키퍼에 설정된 토픽의 Path 정보 등을 설정한다.

❸ KafkaSpout 객체를 스톰의 Topology에 설정한다. "고유 ID", "Kafka Spout 객체", "병렬 처리 힌트"를 설정한다.

❹ KafkaSpout가 수신받은 데이터를 어떻게 라우팅할지에 대한 설정이다. 정확히 말하면 Spout의 설정이 아니라 스톰 Topology의 그루핑 설정으로 All-Grouping 기능을 이용해 앞서 설정한 "kafkaSpout"로부터 받은 데이터를 두 개의 Bolt(Split/Esper)에 동일하게 전달하는 것이다.
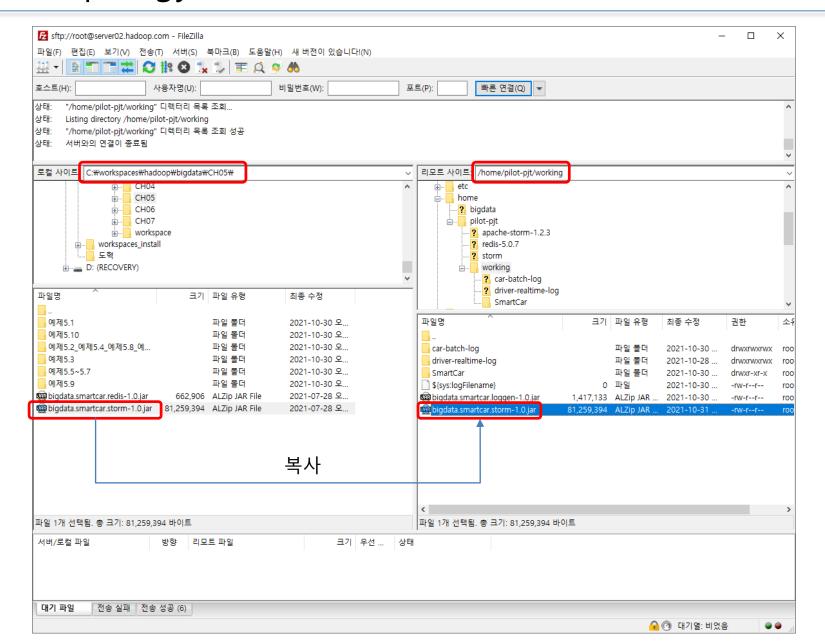
HBase 테이블 생성

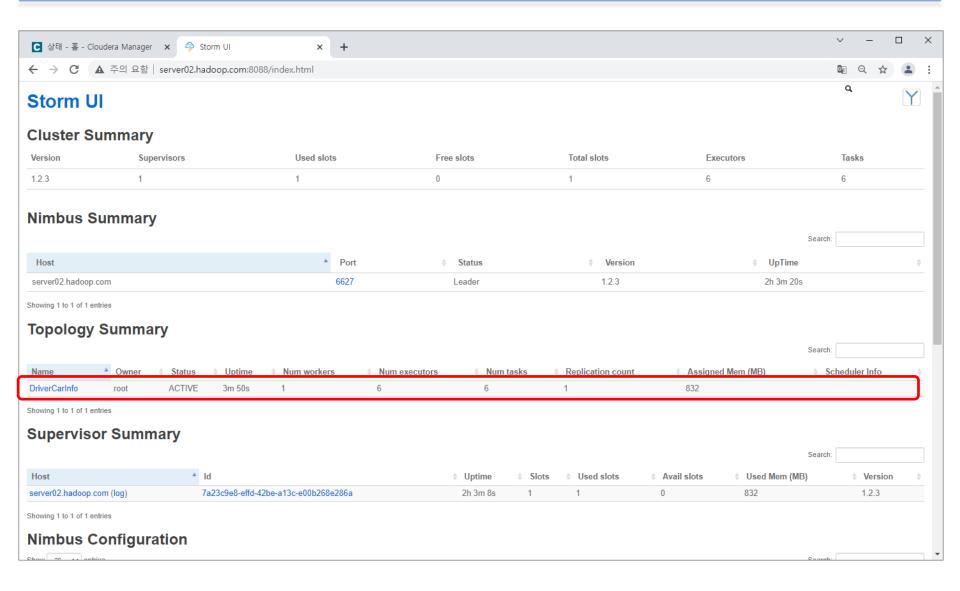# HBase 테이블 생성

스톰 Topology 배포

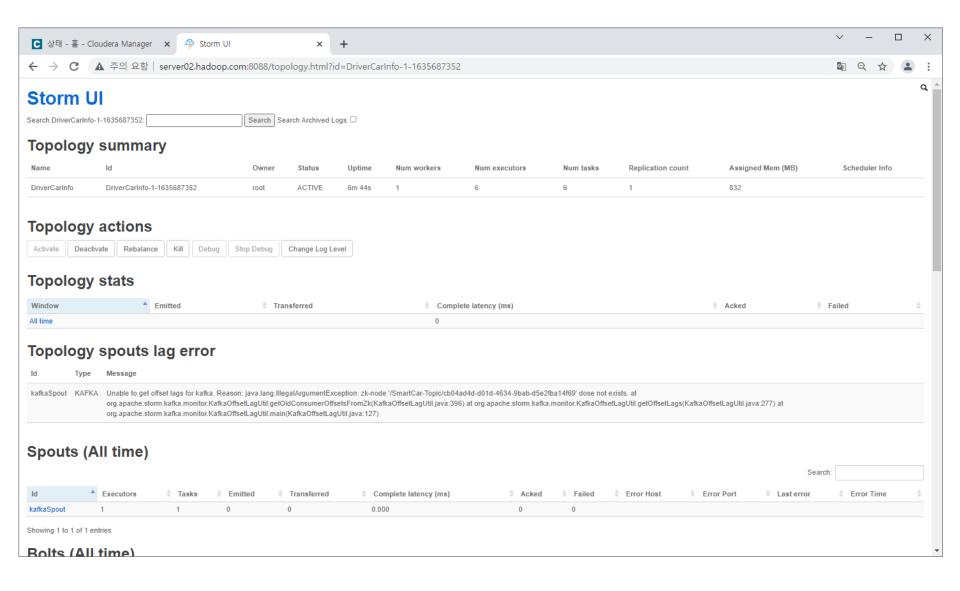# 스톰 Topology 배포


```
root@server02:/home/pilot-pjt/working                          —  □  ✕

[root@server02 ~]# cd /home/pilot-pjt/working/
[root@server02 working]# storm jar bigdata.smartcar.storm-1.0.jar com.wikibook.bigdata.sm
artcar.storm.SmartCarDriverTopology DriverCarInfo

        ...

18687 [main] INFO  o.a.s.StormSubmitter - Successfully uploaded topology jar to assigned
location: /home/pilot-pjt/storm/data/nimbus/inbox/stormjar-05b8bacc-407b-4f83-8598-4377ca
3e5d8f.jar
18688 [main] INFO  o.a.s.StormSubmitter - Submitting topology DriverCarInfo in distribute
d mode with conf {"storm.zookeeper.topology.auth.scheme":"digest","storm.zookeeper.retry.
times":5,"storm.zookeeper.port":2181,"nimbus.thrift.port":6627,"topology.debug":true,"sto
rm.zookeeper.session.timeout":20000,"storm.zookeeper.retry.interval":1000,"storm.zookeepe
r.topology.auth.payload":"-9204397587414060732:-8731124369090014868","storm.zookeeper.ser
vers":["server02.hadoop.com"],"HBASE_CONFIG":{"hbase.rootdir":"hdfs:\/\/server01.hadoop.c
om:8020\/hbase"},"storm.zookeeper.connection.timeout":15000}
18689 [main] WARN  o.a.s.u.Utils - STORM-VERSION new 1.2.3 old 1.2.3
22417 [main] INFO  o.a.s.StormSubmitter - Finished submitting topology: DriverCarInfo
[root@server02 working]#
```
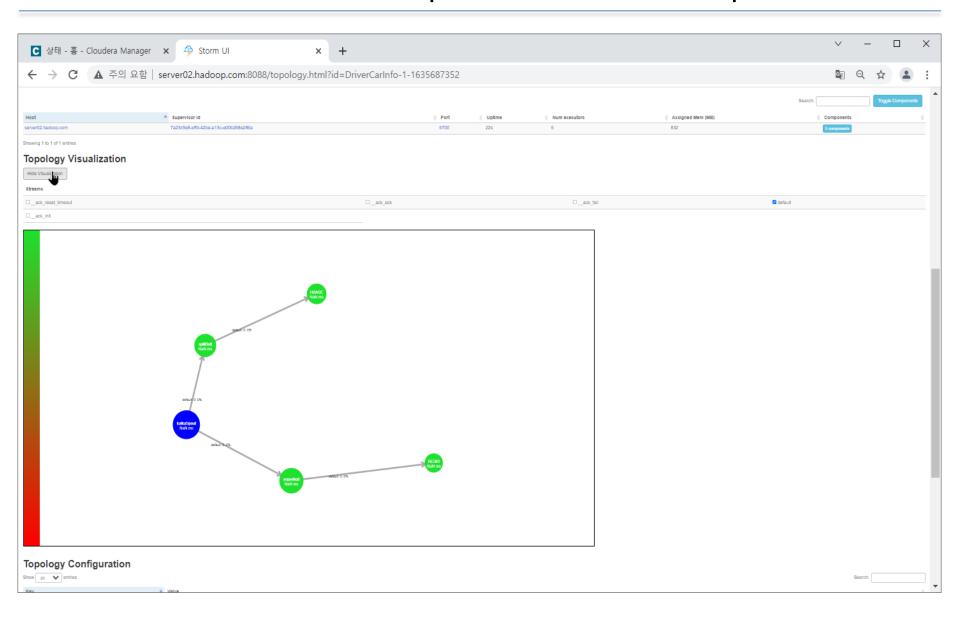
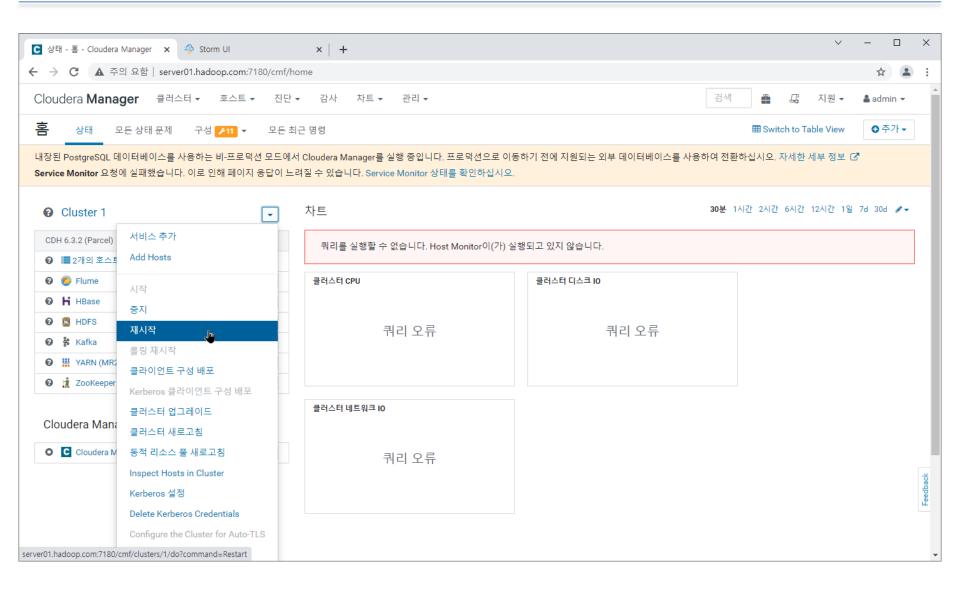# 스톰 관리자 UI 접속 – http://server02.Hadoop.com:8088

스톰 관리자 UI 접속 – http://server02.Hadoop.com:8088

스톰 관리자 UI 접속 – http://server02.Hadoop.com:8088

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 로그 시뮬레이터 작동

```
root@server02:/home/pilot-pjt/working                              —    □    ×

[root@server02 working]# service storm-nimbus status
storm-nimbus is running (pid is 25030).
[root@server02 working]# service storm-supervisor status
storm-supervisor is running (pid is 25166).
[root@server02 working]# service storm-ui status
storm-ui is running (pid is 834).
[root@server02 working]# service redis_6379 status
Redis is running (31187)
[root@server02 working]# cd /home/pilot-pjt/working/
[root@server02 working]# java -cp bigdata.smartcar.loggen-1.0.jar com.wikibook.bigdata.sm
artcar.loggen.DriverLogMain 20500101 10 &
[1] 28351
[root@server02 working]#
[root@server02 working]#
```

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 새로운 server02 putty 실행 - 실시간 로그 발생 확인.

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- Hbase에 적재 데이터 확인

```
root@server02:/home/pilot-pjt/working/driver-realtime-log                          —    □    ×

[root@server02 driver-realtime-log]# hbase shell
HBase Shell
Use "help" to get list of supported commands.
Use "exit" to quit this interactive shell.
For Reference, please visit: http://hbase.apache.org/2.0/book.html#shell
Version 2.1.0-cdh6.3.2, rUnknown, Fri Nov  8 05:44:07 PST 2019
Took 0.1062 seconds
hbase(main):001:0> count 'DriverCarInfo'
Current count: 1000, row: 20230010100502-W0001
Current count: 2000, row: 40530010100502-H0003
Current count: 3000, row: 60840010100502-W0001
Current count: 4000, row: 81150010100502-F0002
4757 row(s)
Took 97.9228 seconds
=> 4757
hbase(main):002:0>
```

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- Hbase에 적재 데이터 확인

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- Hbase에 적재 데이터 확인

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 레디스에 적재 데이터 확인

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 레디스 클라이언트 어플리케이션 작동

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 레디스 클라이언트 어플리케이션 작동

```
root@server02:/home/pilot-pjt/working                                    —    □    ×

[root@server02 ~]# cd /home/pilot-pjt/working/
[root@server02 working]# java -cp bigdata.smartcar.redis-1.0.jar com.wikibook.bigdata.smartc
ar.redis.OverSpeedCarInfo 20500101
#################################################
#####    Start of The OverSpeed SmartCar    #####
#################################################


[ Try No.1]
W0001-20500101013246
F0002-20500101001438
S0002-20500101003342


#################################################
######    End of The OverSpeed SmartCar    ######
#################################################




#################################################
#####    Start of The OverSpeed SmartCar    #####
#################################################


[ Try No.2]
```

실시간 적재 파일럿 실행 4단계-실시간 적재 기능 테스트

- 운전자 로그 발생 어플리케이션 중지