

# 3.6 수집 파일럿 실행 4단계 - 카프카 기능

## 카프카 Producer / Consumer 사용

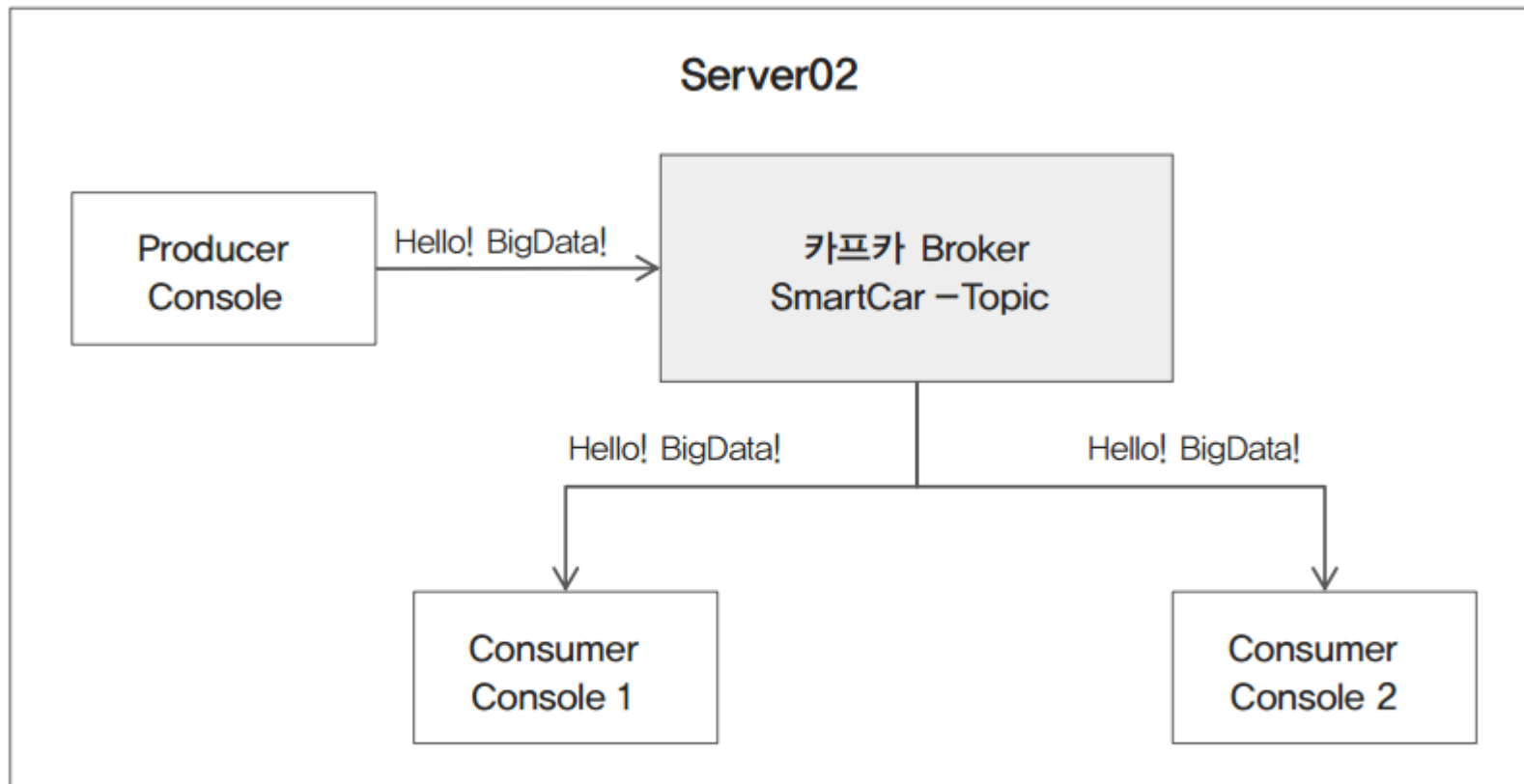


그림 3.30 카프카 Producer와 Consumer의 메시지 송수신 점검

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

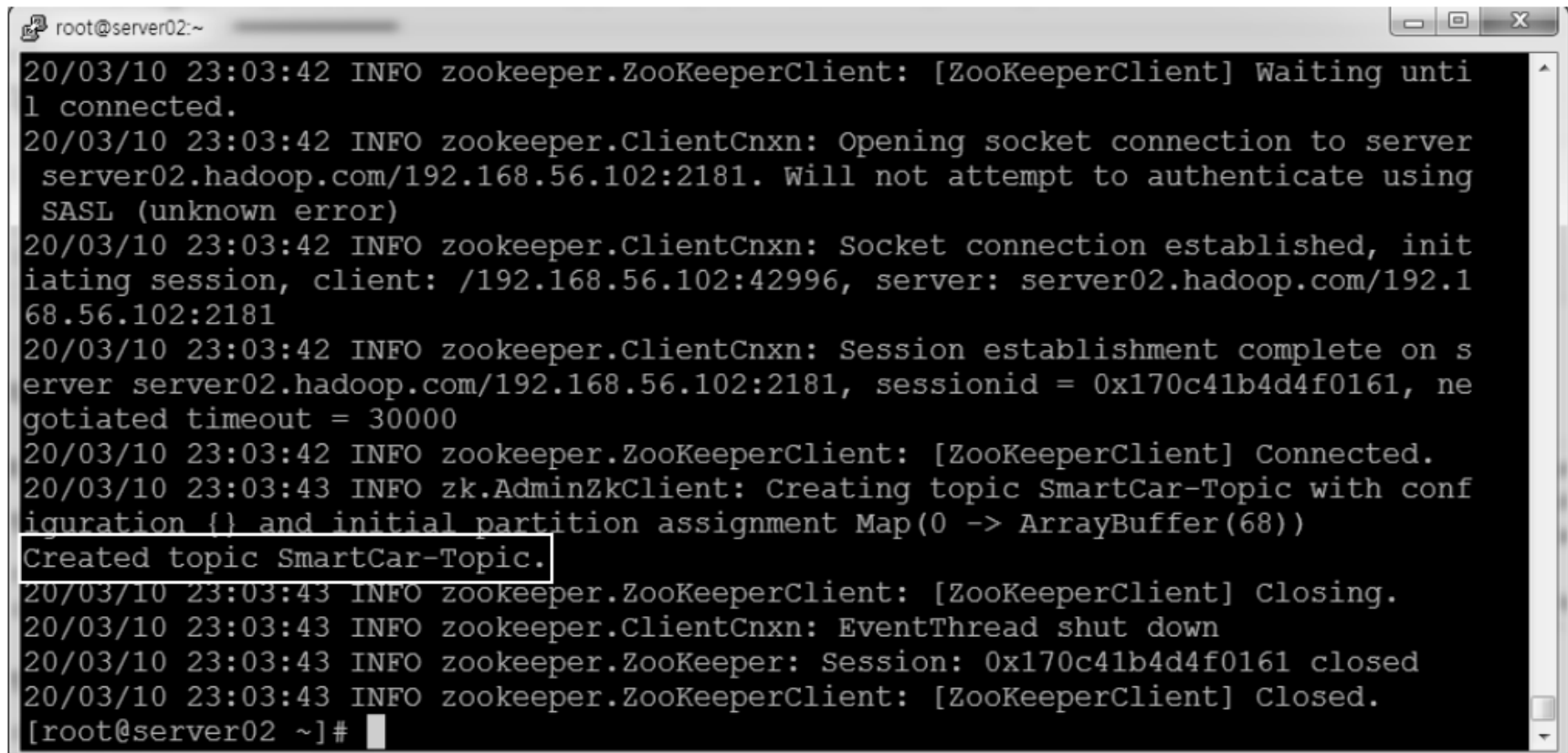
### 카프카 Topic 생성 (1/2)

```
$ kafka-topics --create --zookeeper server02.hadoop.com:2181 --replication-factor 1 --partitions 1 --topic SmartCar-Topic
```

- replication-factor 옵션은 카프카를 다중 Broker로 만들고 전송한 데이터를 replication-factor 개수만큼 복제하게 되는데, 파일럿 프로젝트에서는 단일 카프카 브로커이므로 복제 개수는 1개만 설정한다.
- partitions 옵션은 해당 Topic에 데이터들이 partitions의 개수만큼 분리 저장하게 된다. 이 역시 다중 Broker에서 쓰기/읽기 성능 향상을 위해 사용하는 옵션이다. 파일럿 환경에서는 1로만 설정 한다.

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

### 카프카 Topic 생성 (2/2)

A terminal window titled 'root@server02:~' showing the logs of a Kafka ZooKeeper client. The logs indicate a successful connection to a ZooKeeper server at server02.hadoop.com/192.168.56.102:2181. The client then creates a new topic named 'SmartCar-Topic' with a configuration of {} and an initial partition assignment of Map(0 -> ArrayBuffer(68)). The topic is successfully created, and the client then closes the connection. The final line shows the prompt '[root@server02 ~]#'.

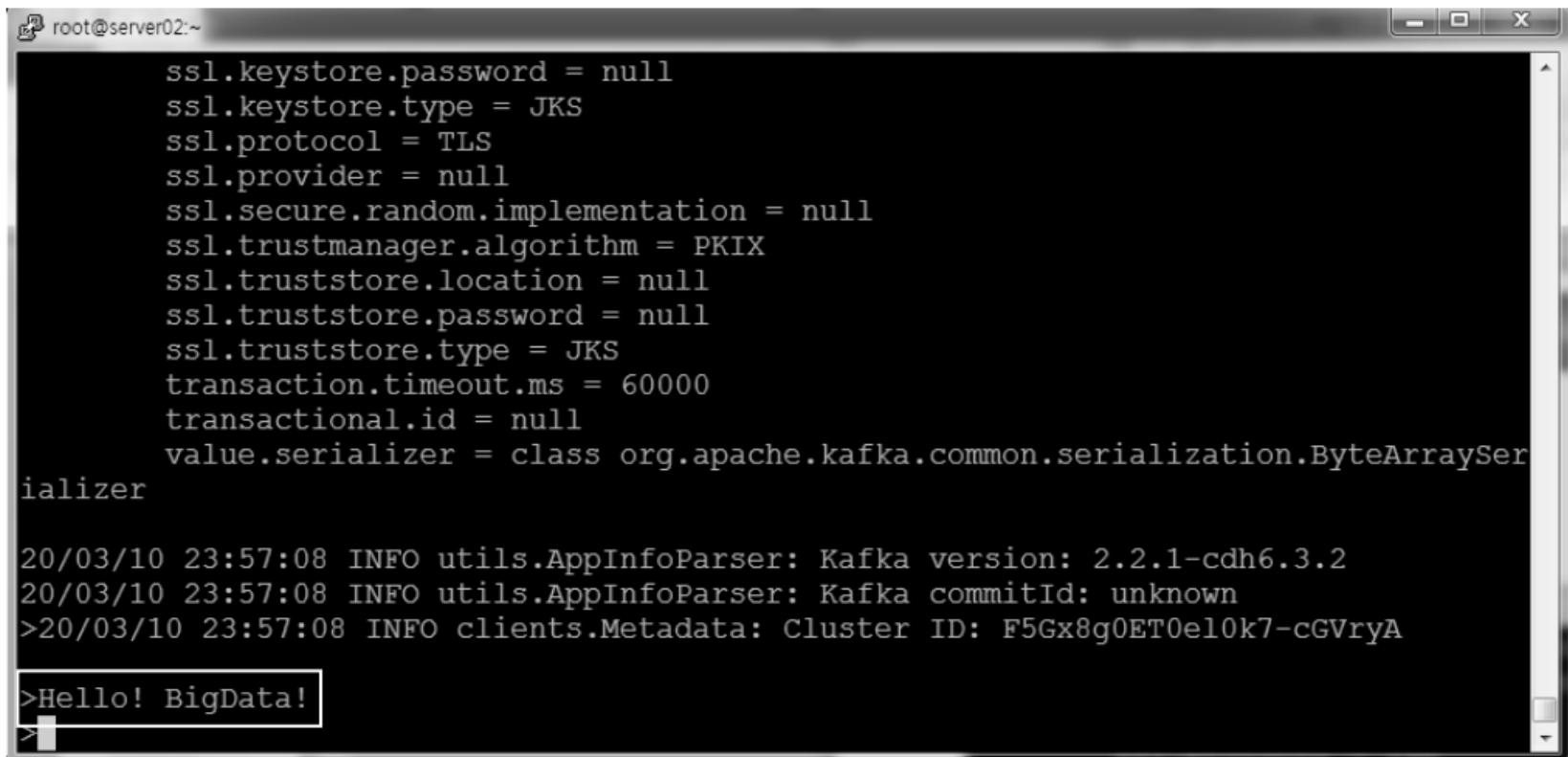
```
root@server02:~
20/03/10 23:03:42 INFO zookeeper.ZooKeeperClient: [ZooKeeperClient] Waiting until connected.
20/03/10 23:03:42 INFO zookeeper.ClientCnxn: Opening socket connection to server server02.hadoop.com/192.168.56.102:2181. Will not attempt to authenticate using SASL (unknown error)
20/03/10 23:03:42 INFO zookeeper.ClientCnxn: Socket connection established, initiating session, client: /192.168.56.102:42996, server: server02.hadoop.com/192.168.56.102:2181
20/03/10 23:03:42 INFO zookeeper.ClientCnxn: Session establishment complete on server server02.hadoop.com/192.168.56.102:2181, sessionId = 0x170c41b4d4f0161, negotiated timeout = 30000
20/03/10 23:03:42 INFO zookeeper.ZooKeeperClient: [ZooKeeperClient] Connected.
20/03/10 23:03:43 INFO zk.AdminZkClient: Creating topic SmartCar-Topic with configuration {} and initial partition assignment Map(0 -> ArrayBuffer(68))
Created topic SmartCar-Topic.
20/03/10 23:03:43 INFO zookeeper.ZooKeeperClient: [ZooKeeperClient] Closing.
20/03/10 23:03:43 INFO zookeeper.ClientCnxn: EventThread shut down
20/03/10 23:03:43 INFO zookeeper.ZooKeeper: Session: 0x170c41b4d4f0161 closed
20/03/10 23:03:43 INFO zookeeper.ZooKeeperClient: [ZooKeeperClient] Closed.
[root@server02 ~]#
```

그림 3.28 카프카 Topic 생성

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

### 카프카 Producer 사용

```
$ kafka-console-producer --broker-list server02.hadoop.com:9092 -topic SmartCar-Topic
```



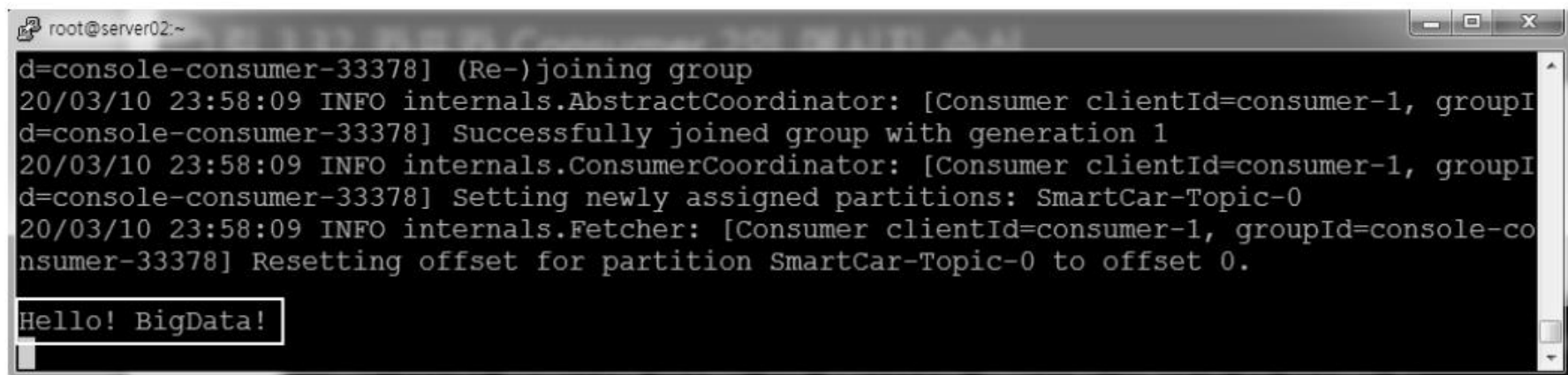
```
root@server02:~  
ssl.keystore.password = null  
ssl.keystore.type = JKS  
ssl.protocol = TLS  
ssl.provider = null  
ssl.secure.random.implementation = null  
ssl.trustmanager.algorithm = PKIX  
ssl.truststore.location = null  
ssl.truststore.password = null  
ssl.truststore.type = JKS  
transaction.timeout.ms = 60000  
transactional.id = null  
value.serializer = class org.apache.kafka.common.serialization.ByteArraySer  
ializer  
  
20/03/10 23:57:08 INFO utils.AppInfoParser: Kafka version: 2.2.1-cdh6.3.2  
20/03/10 23:57:08 INFO utils.AppInfoParser: Kafka commitId: unknown  
>20/03/10 23:57:08 INFO clients.Metadata: Cluster ID: F5Gx8g0ET0el0k7-cGVryA  
  
>Hello! BigData!  
>
```

그림 3.29 카프카 Producer로 메시지 생성

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

### 카프카 Consumer-1 사용

```
$ kafka-console-consumer --bootstrap-server server02.hadoop.com:9092 --topic SmartCar-Topic  
--partition 0 --from-beginning
```



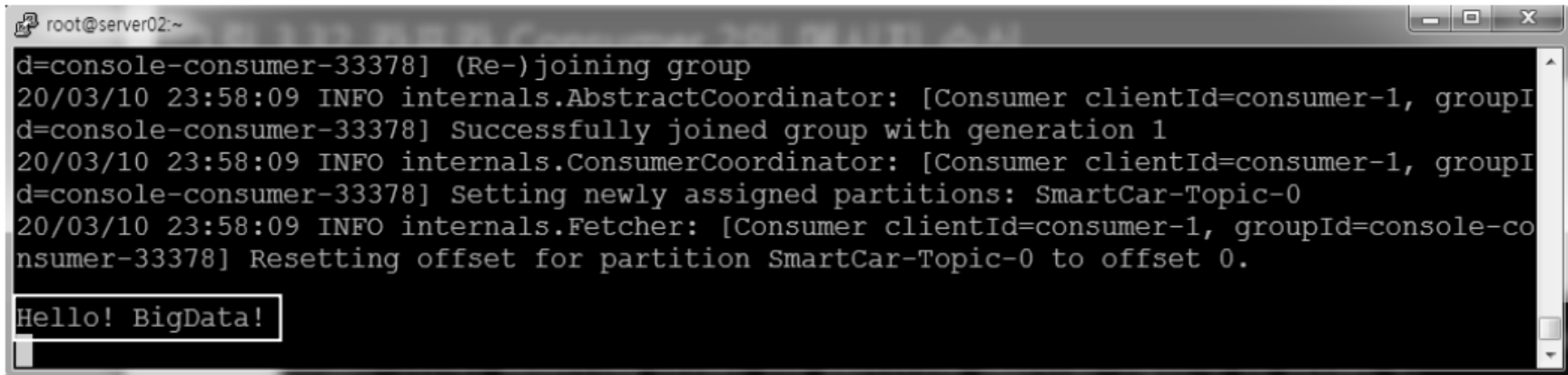
```
root@server02:~  
d=console-consumer-33378] (Re-)joining group  
20/03/10 23:58:09 INFO internals.AbstractCoordinator: [Consumer clientId=consumer-1, groupI  
d=console-consumer-33378] Successfully joined group with generation 1  
20/03/10 23:58:09 INFO internals.ConsumerCoordinator: [Consumer clientId=consumer-1, groupI  
d=console-consumer-33378] Setting newly assigned partitions: SmartCar-Topic-0  
20/03/10 23:58:09 INFO internals.Fetcher: [Consumer clientId=consumer-1, groupId=console-co  
nsumer-33378] Resetting offset for partition SmartCar-Topic-0 to offset 0.  
  
Hello! BigData!
```

그림 3.31 카프카 Consumer 1의 메시지 수신

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

### 카프카 Consumer-2 사용

```
$ kafka-console-consumer --bootstrap-server server02.hadoop.com:9092 --topic SmartCar-Topic  
--partition 0 --from-beginning
```



```
root@server02:~  
d=console-consumer-33378] (Re-)joining group  
20/03/10 23:58:09 INFO internals.AbstractCoordinator: [Consumer clientId=consumer-1, groupId=console-consumer-33378] Successfully joined group with generation 1  
20/03/10 23:58:09 INFO internals.ConsumerCoordinator: [Consumer clientId=consumer-1, groupId=console-consumer-33378] Setting newly assigned partitions: SmartCar-Topic-0  
20/03/10 23:58:09 INFO internals.Fetcher: [Consumer clientId=consumer-1, groupId=console-consumer-33378] Resetting offset for partition SmartCar-Topic-0 to offset 0.  
Hello! BigData!
```

그림 3.32 카프카 Consumer 2의 메시지 수신

## 3.6 수집 파일럿 실행 4단계 - 카프카 기능

# 실습