

PHÂN TÍCH LUỒNG DỮ LIỆU LỚN VỚI KAFKA & SPARK

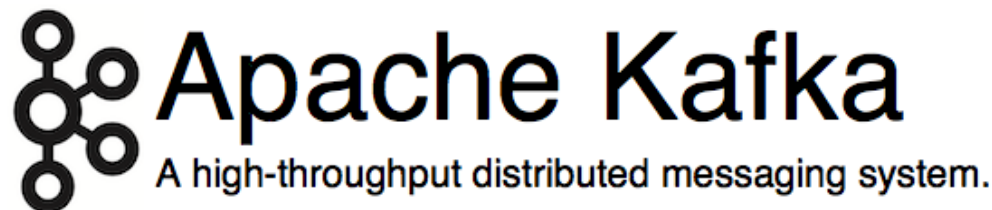
Giảng viên: **TS. Nguyễn Văn Quyết**



- Giới thiệu chung về Kafka
- Một số khái niệm cơ bản trong Kafka
- Kiến trúc của Kafka
- Kafka Workflow
- Thực hành với Kafka
- Hỏi & đáp

What is Kafka?

- Apache Kafka là một hệ thống message pub/sub (**distributed publish-subscribe messaging system**) và có khả năng truyền một lượng lớn các message theo thời gian thực.
- Kafka messages được lưu trữ trên ổ đĩa và được nhân bản trên cụm để phòng tránh việc mất dữ liệu.
- Kafka được “built on top of the ZooKeeper”.
- <http://kafka.apache.org/>



Tại sao nên sử dụng Kafka?

- **Tin cậy (Reliability):** phân tán, phân mảnh, sao lưu và khả năng chịu lỗi
- **Khả năng mở rộng (Scalability):** mở rộng theo chiều ngang.
- **Tính an toàn (Durability):** dữ liệu được lưu trữ an toàn cho đến khi bên nhận sẵn sàng nhận nó.
- **Hiệu năng (Performance):** Kafka thích hợp cho việc xử lý dòng dữ liệu thời gian thực với thông lượng cao cho cả hoạt động pub/sub messages.

Kafka Usecases

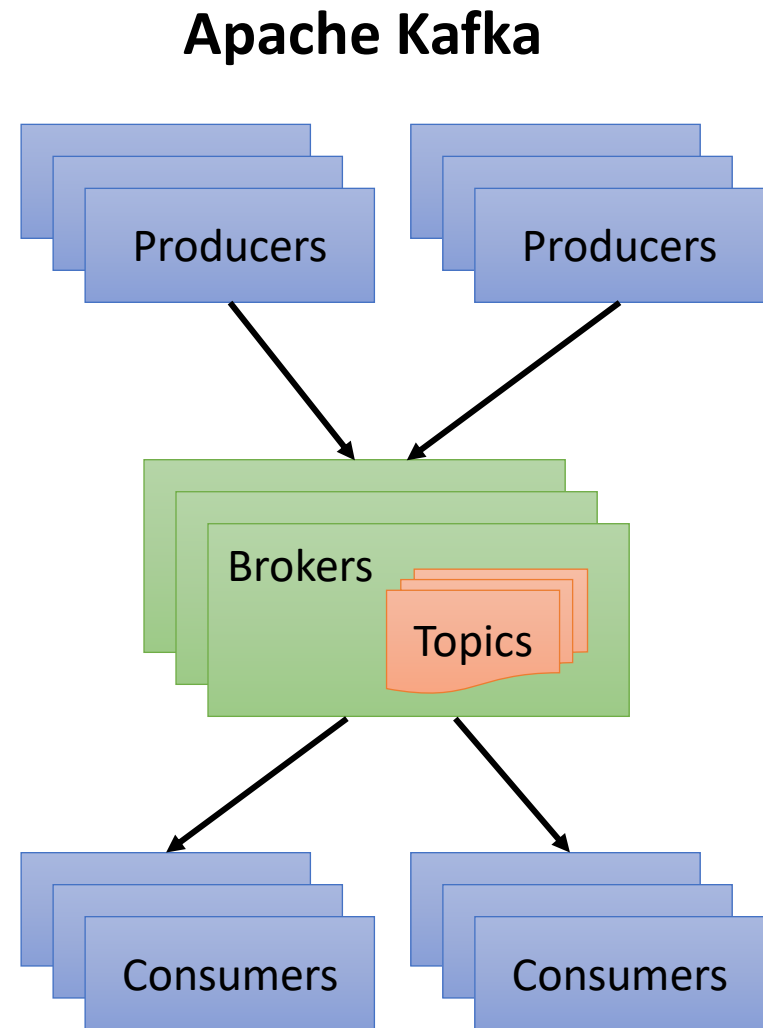
- Messaging
- Website Activity Tracking
- Metrics
- Log Aggregation
- Stream Processing
- Commit Log

Các khái niệm cơ bản

- A first look
- Topics, partitions, offsets, replicas
- Brokers, producers, consumers
- Putting it all together

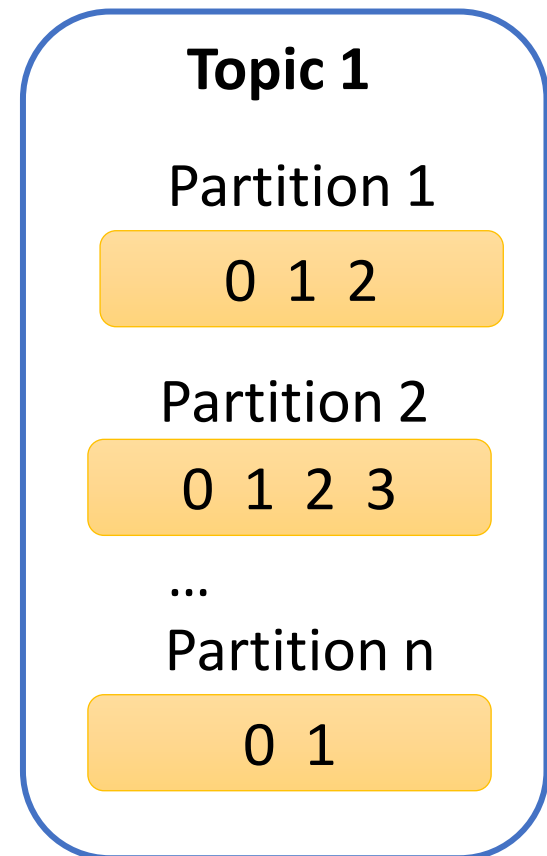
A first look

- **Producers** ghi dữ liệu tới các **brokers**.
- Dữ liệu được lưu trong các **topics**.
- **Consumers** đọc dữ liệu từ các **brokers**.



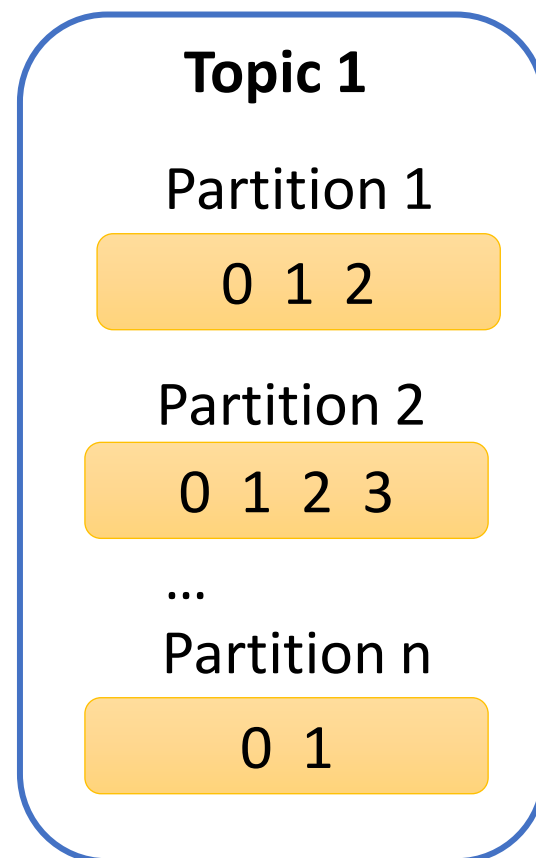
Topics (1/5)

- Một **topic** là một chủ đề mà các bản ghi (dữ liệu) được đưa vào để publish.
- Một topic trong Kafka có thể có nhiều subscriber
- Một topic có thể được chia làm nhiều **partitions**



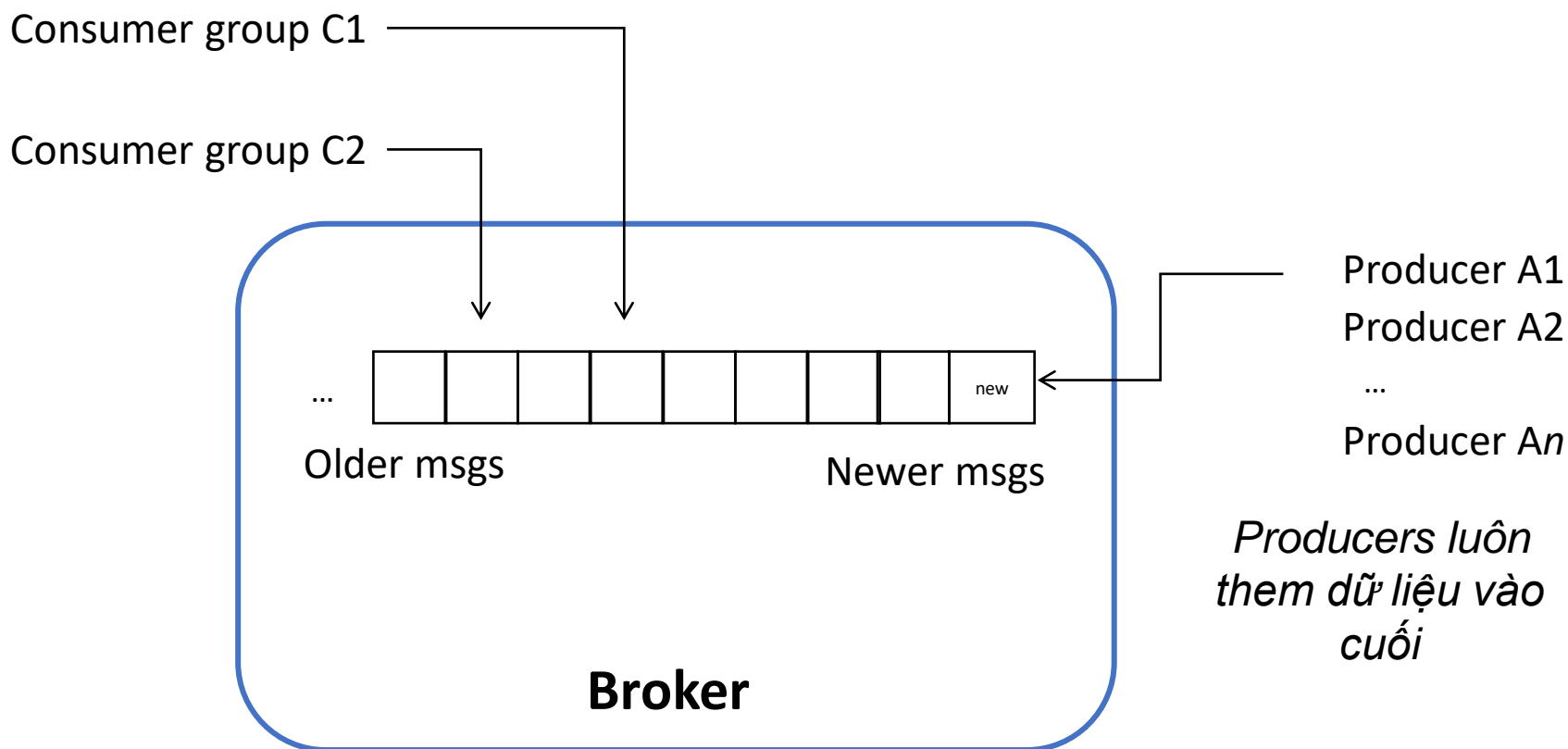
Topics (2/5)

- **Partitions:** chứa các messages được sắp xếp tuần tự không thay đổi được. Nó được thiết lập là tập các file có kích thước bằng nhau.
- Mỗi partition có một ID duy nhất gọi là **partition offset**.



Topics (3/5)

Consumers sử dụng một “offset pointer” để theo dõi và điều khiển việc đọc của nó



Topics (4/5)

● **Replicas:** bản sao lưu của một partition

- Nó được sử dụng để phòng tránh việc mất dữ liệu.
- Nó **KHÔNG** trợ giúp các producer hoặc consumer trong việc song song hóa!
- Kafka có khả năng thay thế các “dead brokers” (*numReplicas - 1*)
 - LinkedIn: `numReplicas == 2` → 1 broker có thể hỏng

Topics (5/5)

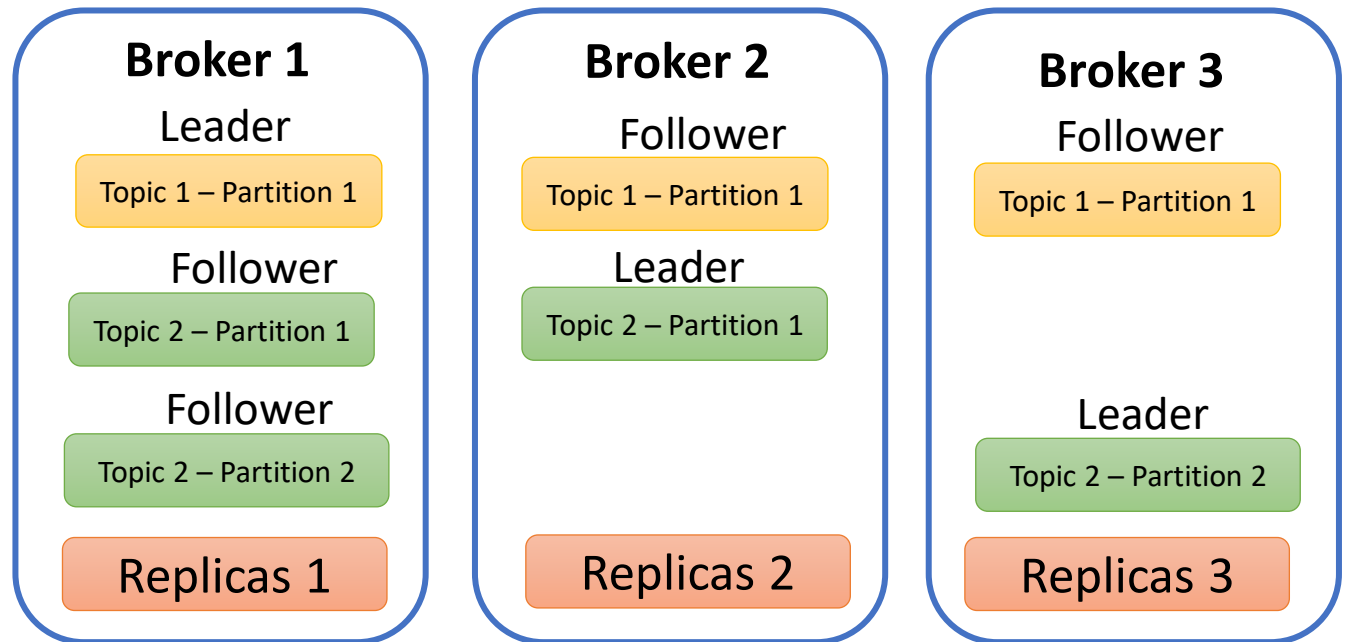
- Mỗi partition có một server (node) giữ vai trò làm "**leader**" và có thể có 0 hoặc nhiều "**followers**".
- **Leader:** là node chịu trách nhiệm đọc và ghi dữ liệu trên một partition nó .
 - Every partition has one server acting as a leader.
- **Follower:** là node hoạt động theo chỉ dẫn của leader
 - Nếu leader fails, một follower sẽ được tự động chọn làm leader.

Brokers

- **Brokers:** nắm giữ các published data.
 - Mỗi broker có thể có 0 hoặc nhiều partitions / topic.

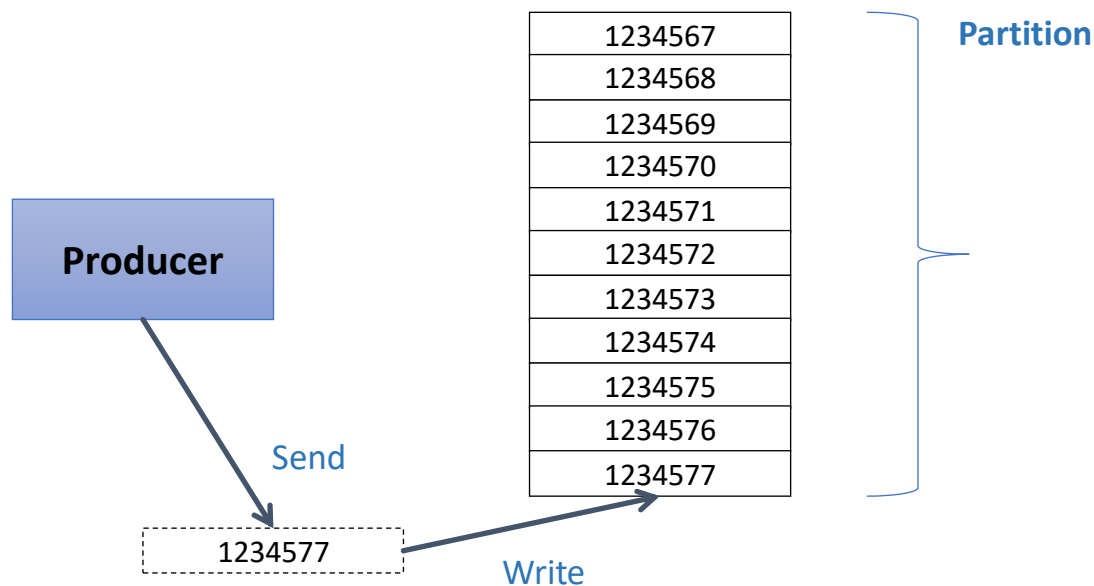
Topic:	Topic 1
Partitions:	1
Replicas:	3

Topic:	Topic 2
Partitions:	2
Replicas:	2



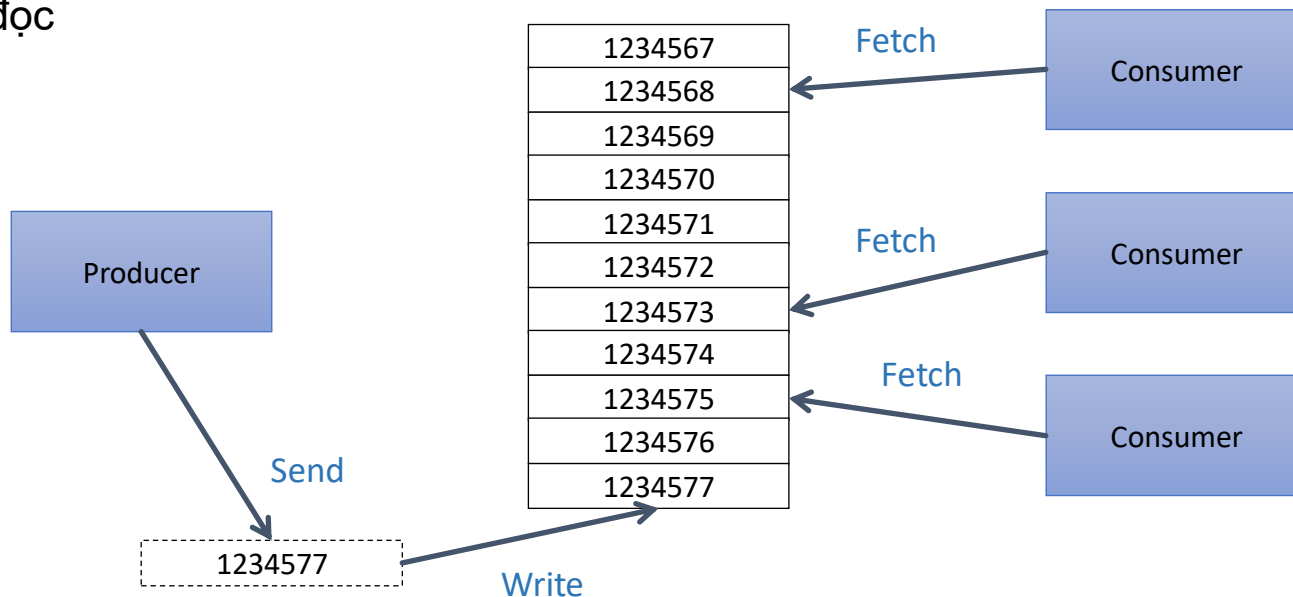
Producers

- **Producers:** đưa các messages to một hoặc nhiều Kafka topics.
 - Producers gửi dữ liệu tới các Kafka brokers.
 - Broker ghi nối tiếp các message vào cuối mỗi segment file.
 - Producer gửi các messages tới một partition mà nó chọn.



Consumers

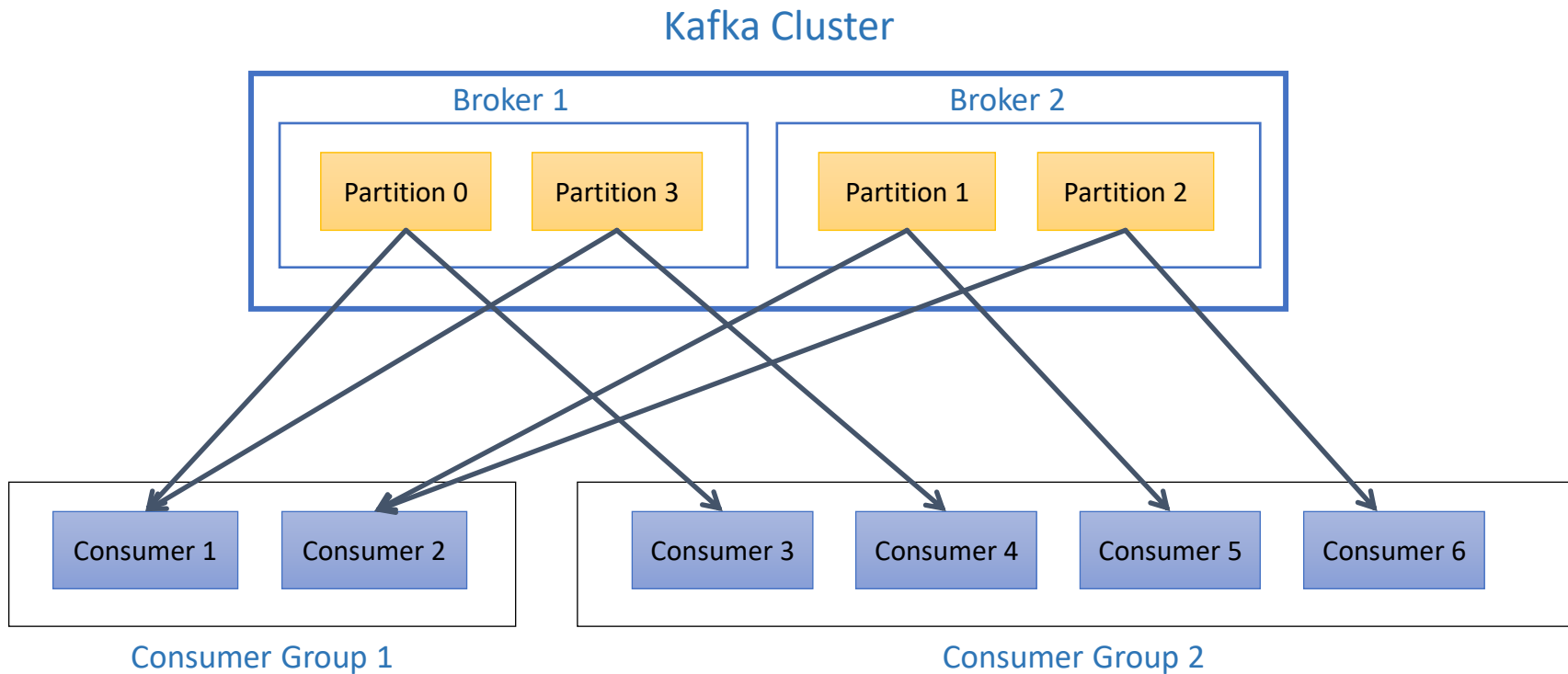
- Consumers đọc dữ liệu từ brokers
- Consumers đăng ký nhận/đọc dữ liệu từ một hoặc nhiều topics
 - Nhiều consumers có thể đọc dữ liệu trên cùng topic
 - Mỗi consumer chịu trách nhiệm quản lý các địa chỉ offset mà
 - Các messages được lưu giữ trong Kafka...chúng không bị xóa sau khi được đọc



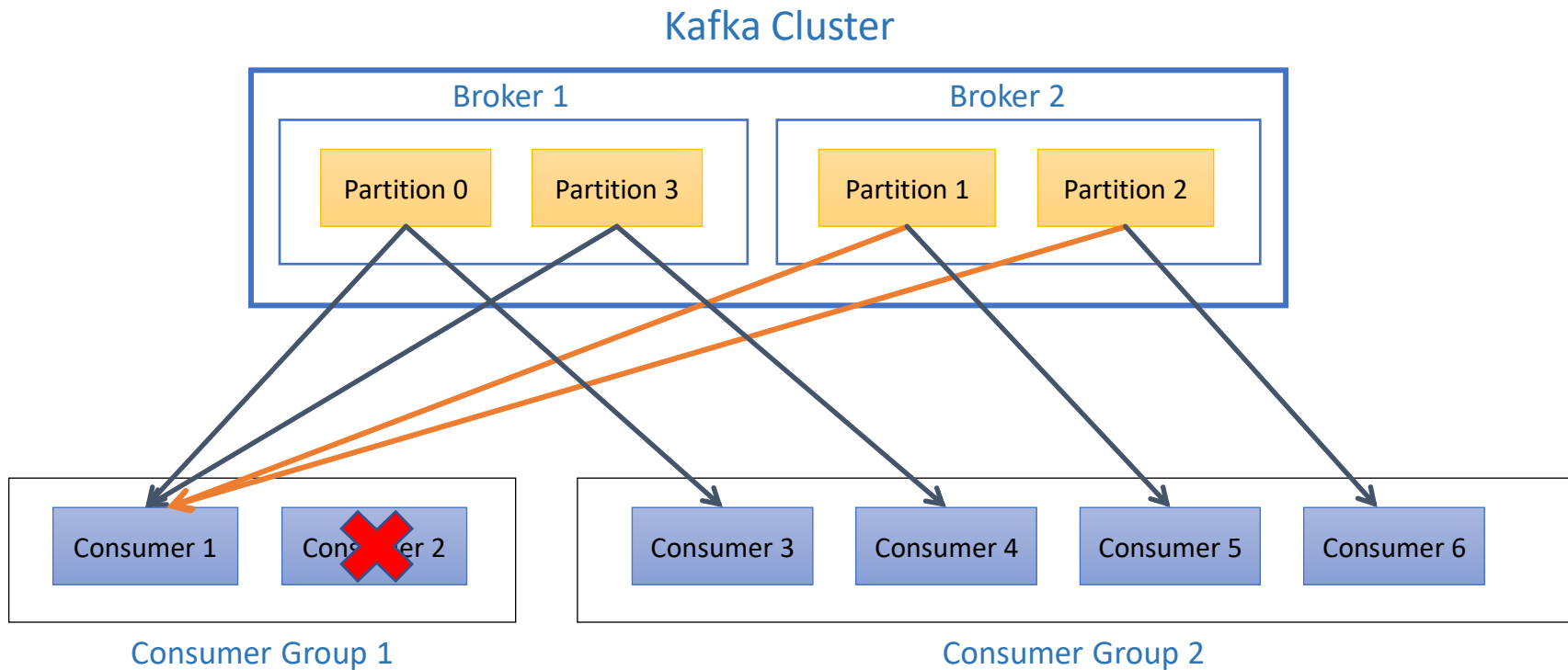
Consumers - Group

- Consumers có thể được tổ chức thành nhóm
- Common Patterns:
 - Tất cả các consumer trong các nhóm khác nhau
 - **Nhiều consumer trong một nhóm**
 - Các Consumers được thêm vào để tăng khả năng mở rộng và chịu lỗi
 - Mỗi consumer đọc từ một hoặc nhiều partitions của một topic
 - Số Consumer một nhóm không nhiều hơn số partitions

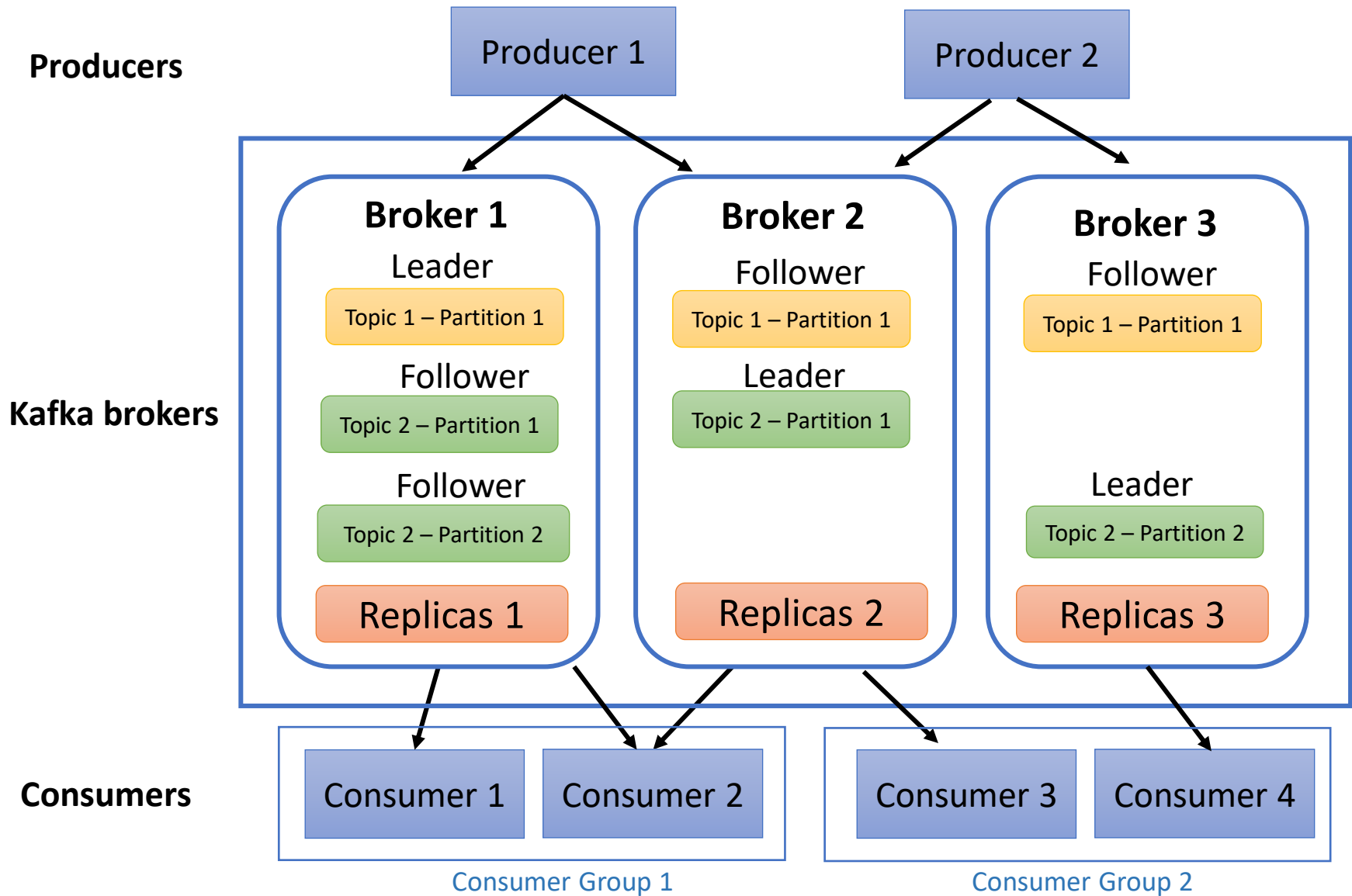
Consumers - Group



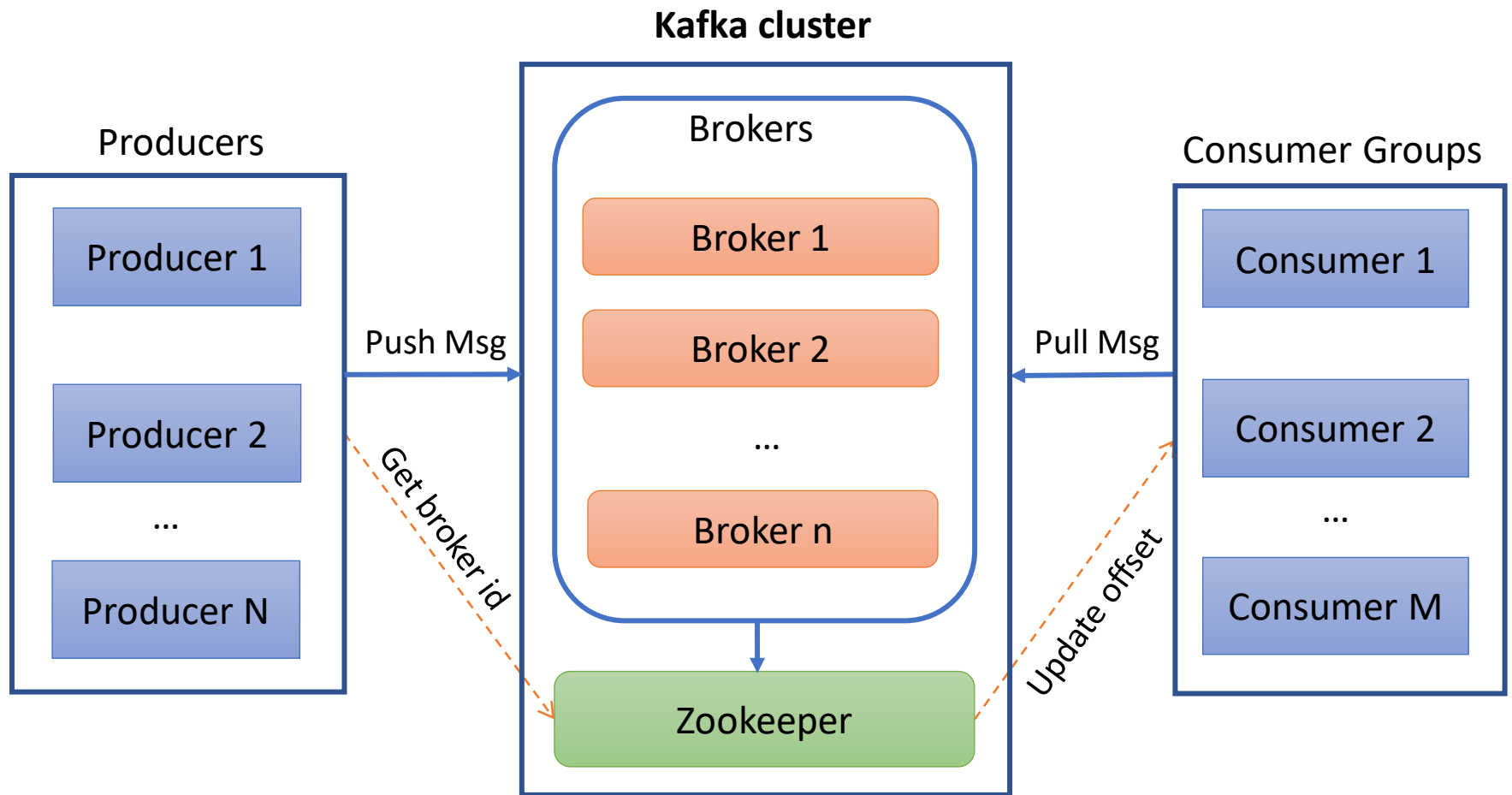
Consumers - Group



Put it all together



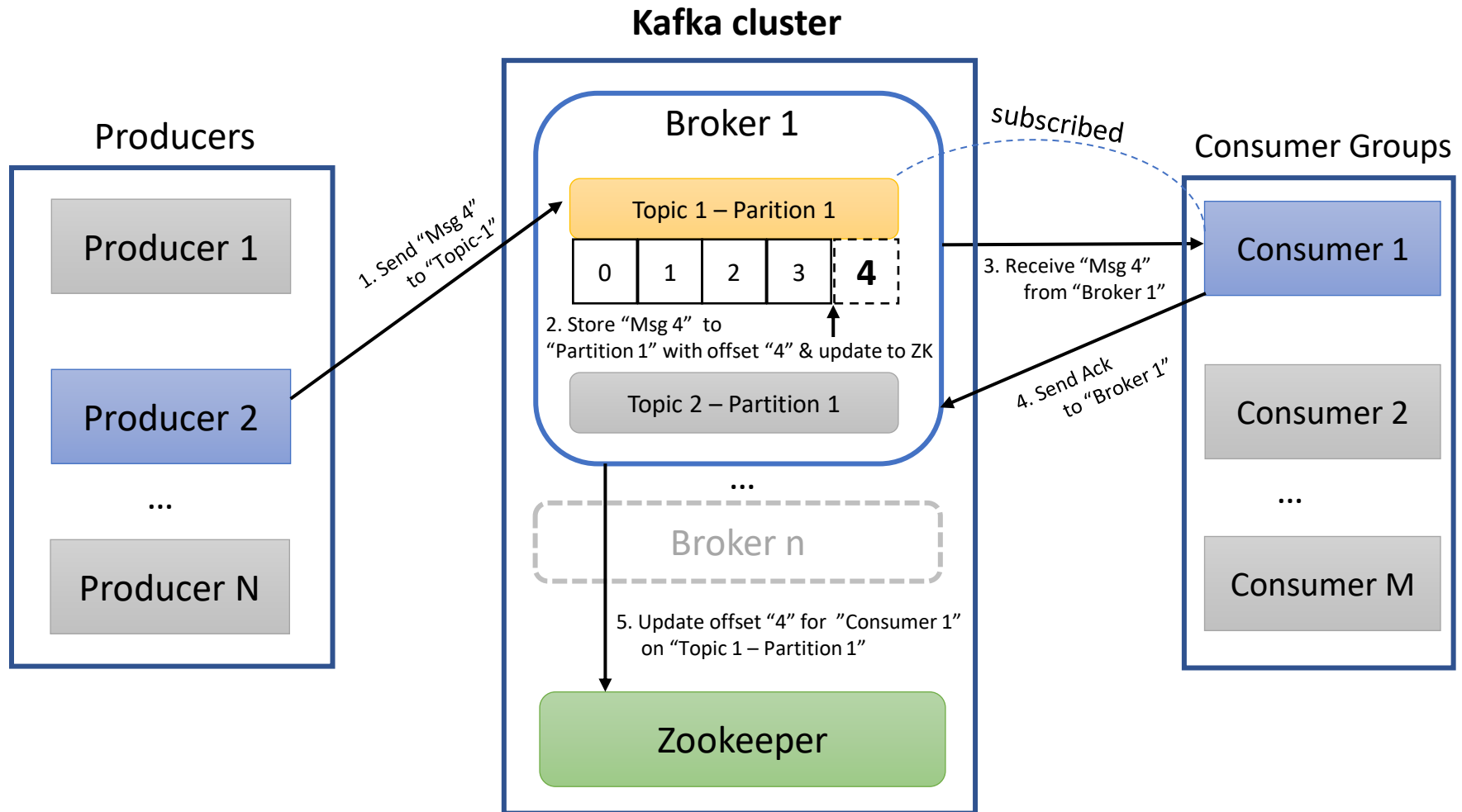
Kiến trúc của Kafka



Vai trò của Zookeeper

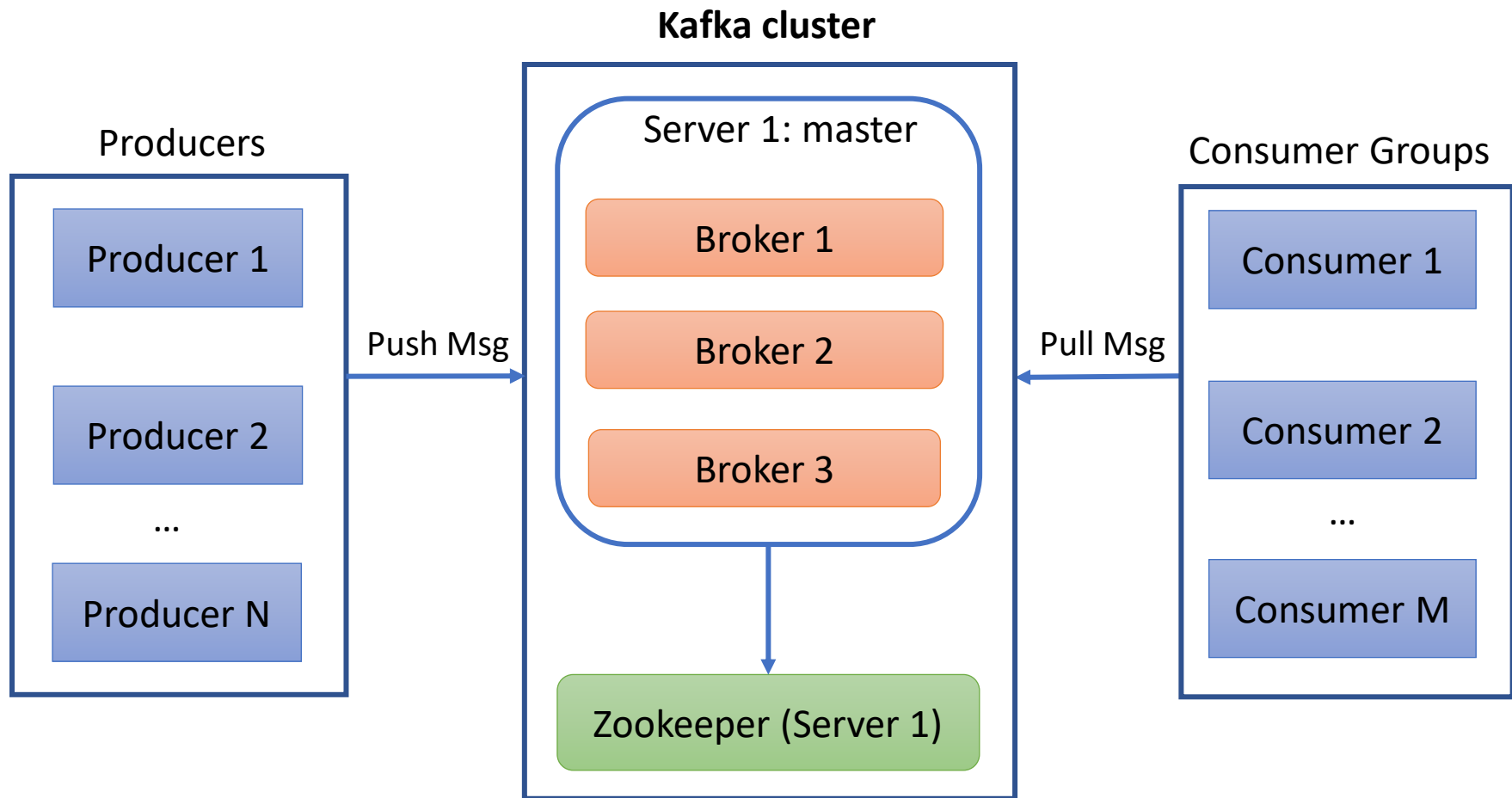
- ZooKeeper được sử dụng để thông báo cho **producer** và **consumer** về sự có mặt của bất kỳ **new broker** hoặc **lỗi xảy ra trên broker** trong hệ thống.
- Zookeeper dùng để lưu trữ thông tin metadata của Kafka như về topics, brokers, và consumer offsets.
- Lựa chọn **leader** giữa các Kafka brokers trong khi một broker bị lỗi.

Kafka Workflow



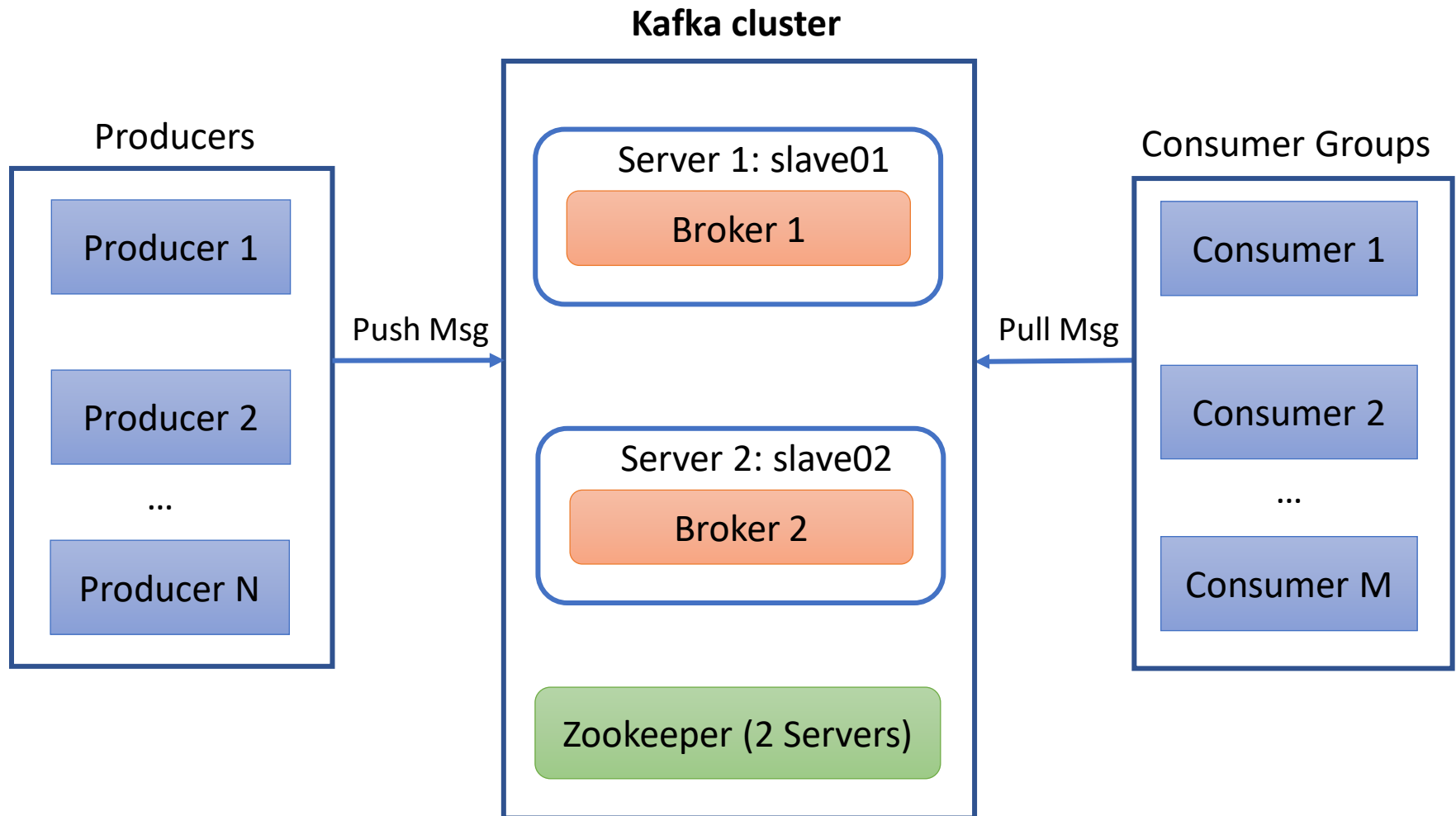
Chuẩn bị môi trường – Kiến trúc hệ thống

● Multi Brokers on Single Node



Chuẩn bị môi trường – Kiến trúc hệ thống

- Multi Brokers on Multi Nodes → Dùng trong bài thực hành



Chuẩn bị môi trường – Cài đặt Zookeeper

● Bước 1: Tải và cài đặt Zookeeper (trên máy master)

```
hduser@master:~$ cd /usr/local
```

```
hduser@master:/usr/local$ sudo wget
```

```
http://apache.tt.co.kr/zookeeper/stable/apache-zookeeper-3.6.3-bin.tar.gz
```

```
hduser@master:/usr/local$ sudo tar -xzvf apache-zookeeper-3.6.3-bin.tar.gz
```

```
hduser@master:/usr/local$ sudo mv apache-zookeeper-3.6.3-bin zookeeper
```

```
hduser@master:/usr/local$ sudo chown -R hduser:hadoop zookeeper
```

Chuẩn bị môi trường – Cài đặt Zookeeper

● Bước 2: Thiết lập biến môi trường cho Zookeeper (trên tất cả các máy)

- Thêm vào file `~/.bashrc` các lệnh sau:

```
# Set Zookeeper related variables
```

```
export ZOOKEEPER_HOME="/usr/local/zookeeper"
```

```
# Add Zookeeper bin/ directory to PATH
```

```
export PATH=$PATH:$ZOOKEEPER_HOME/bin
```

- Chạy lệnh sau để áp dụng

```
$ source ~/.bashrc
```

Chuẩn bị môi trường – Cài đặt Zookeeper

- Bước 3: Tạo thư mục lưu dữ liệu của Zookeeper **(trên tất cả các máy)**

```
hduser@master:~$ sudo mkdir -p /app/zookeeper/data
```

```
hduser@master:~$ sudo chown -R hduser:hadoop /app/zookeeper/
```

```
hduser@slave01:~$ sudo mkdir -p /app/zookeeper/data
```

```
hduser@slave01:~$ sudo chown -R hduser:hadoop /app/zookeeper/
```

```
hduser@slave02 :~$ sudo mkdir -p /app/zookeeper/data
```

```
hduser@slave02:~$ sudo chown -R hduser:hadoop /app/zookeeper/
```

Chuẩn bị môi trường – Cài đặt Zookeeper

- Bước 4: Cấu hình Zookeeper
 - Cấu hình file **zoo.cfg** (trên máy master)

```
hduser@master:~$ cd /usr/local/zookeeper
```

```
hduser@master:/usr/local/zookeeper$ cp conf/zoo_sample.cfg conf/zoo.cfg
```

```
hduser@master:/usr/local/zookeeper$ sudo gedit conf/zoo.cfg
```

// Sửa đường dẫn trở đến thư mục chứa dữ liệu của Zookeeper trong tệp
`dataDir=/app/zookeeper/data`

Chuẩn bị môi trường – Cài đặt Zookeeper

● Bước 5: Copy Zookeeper từ master sang các slaves (1/2)

// Tạo và cấp quyền thư mục zookeeper trên máy slave01

```
hduser@master:~$ ssh slave01
```

```
hduser@slave01:~$ sudo mkdir /usr/local/zookeeper
```

```
hduser@slave01:~$ sudo chown -R hduser:hadoop /usr/local/zookeeper
```

```
hduser@slave01:~$ exit
```

// Tạo và cấp quyền thư mục zookeeper trên máy slave02

```
hduser@master:~$ ssh slave02
```

```
hduser@slave02:~$ sudo mkdir /usr/local/zookeeper
```

```
hduser@slave02:~$ sudo chown -R hduser:hadoop /usr/local/zookeeper
```

```
hduser@slave02:~$ exit
```

Chuẩn bị môi trường – Cài đặt Zookeeper

● Bước 5: Copy Zookeeper từ master sang các slaves (2/2)

// Copy thư mục Hbase từ máy master sang các máy slaves

```
hduser@master:~$          scp          -r          /usr/local/zookeeper/*  
hduser@slave01:/usr/local/zookeeper/
```

```
hduser@master:~$          scp          -r          /usr/local/zookeeper/*  
hduser@slave02:/usr/local/zookeeper/
```

Chuẩn bị môi trường – Cài đặt Zookeeper

● Bước 6: Khởi chạy Zookeeper

- Chạy Zookeeper:

```
hduser@master:~$ cd /usr/local/zookeeper
```

```
hduser@master:/usr/local/zookeeper$ bin/zkServer.sh start
```

```
hduser@master:/usr/local/zookeeper$ bin/zkServer.sh start
ZooKeeper JMX enabled by default
Using config: /usr/local/zookeeper/bin/../conf/zoo.cfg
Starting zookeeper ... STARTED
hduser@master:/usr/local/zookeeper$ jps
3187 QuorumPeerMain
3221 Jps
hduser@master:/usr/local/zookeeper$
```

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 1: Tải và cài đặt Kafka

```
hduser@master:~$ cd /usr/local
```

```
hduser@master:/usr/local$
```

sudo

wget

```
https://downloads.apache.org/kafka/2.8.0/kafka\_2.12-2.8.0.tgz
```

```
hduser@master:/usr/local$ sudo tar -xzf kafka_2.12-2.8.0.tgz
```

```
hduser@master:/usr/local$ sudo mv kafka_2.12-2.8.0.tgz kafka
```

```
hduser@master:/usr/local$ sudo chown -R hduser:hadoop kafka
```


Chuẩn bị môi trường – Cài đặt Kafka

- Bước 2: Thiết lập biến môi trường cho Kafka (trên tất cả các máy)

- Thêm vào file `~/.bashrc` các lệnh sau:

```
# Set Kafka related variables
```

```
export KAFKA_HOME="/usr/local/kafka"
```

```
# Add Kafka bin/ directory to PATH
```

```
export PATH=$PATH:$KAFKA_HOME/bin
```

- Chạy lệnh sau để áp dụng

```
$ source ~/.bashrc
```

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 3: Copy Kafka từ master sang các slaves (1/2)

// Tạo và cấp quyền thư mục kafka trên máy slave01

```
hduser@master:~$ ssh slave01
```

```
hduser@slave01:~$ sudo mkdir /usr/local/kafka
```

```
hduser@slave01:~$ sudo chown -R hduser:hadoop /usr/local/kafka
```

```
hduser@slave01:~$ exit
```

// Tạo và cấp quyền thư mục kafka trên máy slave02

```
hduser@master:~$ ssh slave02
```

```
hduser@slave02:~$ sudo mkdir /usr/local/kafka
```

```
hduser@slave02:~$ sudo chown -R hduser:hadoop /usr/local/kafka
```

```
hduser@slave01:~$ exit
```

Chuẩn bị môi trường – Cài đặt Kafka

- Bước 3: Copy Kafka từ master sang các slaves (2/2)

// Copy thư mục Hbase từ máy master sang các máy slaves

```
hduser@master:~$ scp -r /usr/local/kafka/* hduser@slave01:/usr/local/kafka
```

```
hduser@master:~$ scp -r /usr/local/kafka/* hduser@slave02:/usr/local/kafka
```

Chuẩn bị môi trường – Cài đặt Kafka

- Bước 4: Cấu hình Kafka trên các máy slaves (1/3)
 - Multi Brokers on Multi Nodes (tạo 2 Brokers trên 2 nodes)

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/kafka
```

```
hduser@slave01:/usr/local/kafka$ cp config/server.properties  
config/slave01-1.properties
```

// Trên máy Slave02

```
hduser@slave02:~$ cd /usr/local/kafka
```

```
hduser@slave02:/usr/local/kafka$ cp config/server.properties  
config/slave02-1.properties
```

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 4: Cấu hình Kafka trên các máy slaves (2/3)

- Kiểm tra sửa/thêm vào các cấu hình sau

// Trên máy Slave01

hduser@slave01:/usr/local/kafka\$ sudo gedit config/slave01-1.properties

- broker.id=1
- port=9092
- log.dirs=/tmp/kafka/slave01-1-logs
- host.name=10.0.2.196
- zookeeper.connect= 10.0.2.196:2181, 10.0.2.197:2181

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 4: Cấu hình Kafka trên các máy slaves (3/3)

- Kiểm tra sửa/thêm vào các cấu hình sau

// Trên máy Slave02

hduser@slave02:/usr/local/kafka\$ sudo gedit config/slave02-1.properties

- broker.id=2
- port=9093
- log.dirs=/tmp/kafka/slave02-1-logs
- host.name=10.0.2.197
- zookeeper.connect= 10.0.2.196:2181, 10.0.2.197:2181

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 5: Khởi chạy Zookeeper trên các slaves

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/zookeeper
```

```
hduser@slave01:/usr/local/zookeeper$ bin/zkServer.sh start
```

```
hduser@slave01:/usr/local/zookeeper$ bin/zkCli.sh -server 10.0.2.196:2181
```

// Trên máy Slave02

```
hduser@slave02:~$ cd /usr/local/zookeeper
```

```
hduser@slave02:/usr/local/zookeeper$ bin/zkServer.sh start
```

```
hduser@slave02:/usr/local/zookeeper$ bin/zkCli.sh -server 10.0.2.197:2181
```

Chuẩn bị môi trường – Cài đặt Kafka

● Bước 6: Chạy Kafka server trên các slaves

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/kafka
```

```
hduser@slave01:/usr/local/kafka$ ./bin/kafka-server-start.sh -daemon  
config/slave01-1.properties
```

// Trên máy Slave02

```
hduser@slave02:~$ cd /usr/local/kafka
```

```
hduser@slave02:/usr/local/kafka$ ./bin/kafka-server-start.sh -daemon  
config/slave02-1.properties
```


Chuẩn bị môi trường – Cài đặt Kafka

● Bước 7: Kiểm tra Kafka server trên các slaves

// Trên máy Slave01

hduser@slave01:~\$ jps

```
hduser@slave01:/usr/local$ cd kafka/  
hduser@slave01:/usr/local/kafka$ bin/kafka-server-start.sh -daemon config/slave01-1.properties  
hduser@slave01:/usr/local/kafka$ jps  
2304 ZooKeeperMain  
2712 Jps  
2267 QuorumPeerMain  
2685 Kafka  
hduser@slave01:/usr/local/kafka$
```

// Tương tự, kiểm tra Kafka trên máy Slave02

```
hduser@slave02:/usr/local/kafka$ jps  
2114 QuorumPeerMain  
7763 Kafka  
8166 Jps  
4620 ZooKeeperMain  
hduser@slave02:/usr/local/kafka$
```

Thử nghiệm một hệ thống Pub/Sub

● Bước 1: Tạo topics

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/kafka
```

```
hduser@slave01:/usr/local/kafka$ ./bin/kafka-topics.sh --zookeeper  
10.0.2.196:2181,10.0.2.197:2181 --create --topic weather --replication-factor  
1 --partitions 1
```

```
hduser@slave01:/usr/local/kafka$ ./bin/kafka-topics.sh --zookeeper 10.0.2.196:2181,10.0.2.197  
:2181 --create --topic weather --replication-factor 1 --partitions 1  
Created topic weather.
```

Thử nghiệm một hệ thống Pub/Sub

● Bước 2: Xem thông tin topic

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/kafka
```

```
hduser@slave01:/usr/local/kafka$ ./bin/kafka-topics.sh --zookeeper  
10.0.2.196:2181,10.0.2.197:2181 --describe --topic weather
```

```
hduser@slave01:/usr/local/kafka$ kafka-topics.sh --zookeeper 10.0.2.197:2181,10.0.2.196:2181  
--describe --topic weather  
Topic: weather TopicId: u_P0QWH2Sxa20tjKvhSLrg PartitionCount: 1 ReplicationFactor: 1  
Configs:  
Topic: weather Partition: 0 Leader: 1 Replicas: 1 Isr: 1  
hduser@slave01:/usr/local/kafka$
```

Thử nghiệm một hệ thống Pub/Sub

● Bước 3: Chạy một producer

// Trên máy Slave01

```
hduser@slave01:~$ cd /usr/local/kafka
```

```
hduser@slave01:/usr/local/kafka$ ./bin/kafka-console-producer.sh --broker-list 10.0.2.196:9092,10.0.2.197:9093 --topic weather
```

```
hduser@slave01:/usr/local/kafka$ kafka-console-producer.sh --broker-list 10.0.2.196:9092,10.0.2.197:9093 --topic weather
>21
>23
>
```

Thử nghiệm một hệ thống Pub/Sub

● Bước 4: Chạy một consumer

// Trên máy Slave02

```
hduser@slave02:~$ cd /usr/local/kafka
```

```
hduser@slave02:/usr/local/kafka$ ./bin/kafka-console-consumer.sh --bootstrap-server 10.0.2.196:9092,10.0.2.197:9093 --topic weather
```

```
hduser@slave02:/usr/local/kafka$ ./bin/kafka-console-consumer.sh --bootstrap-server 10.0.2.196:9092,10.0.2.197:9093 --topic weather --from-beginning
21
23
█
```

Trân trọng cảm ơn!
Q&A