□□□ - Kafka □□ □□ □□ □□

- 1. Kafka □□ □□
  - Kafka □□ □□ □□
- 2. Kafka □□
  - Kafka, Zookeeper □□ □ □□

1. Kafka □□ □□

Kafka 필수 개념 정리

Kafka 기본 개념
Kafka는 고성능 분산 메시징 시스템으로, 대량의 데이터를 처리하고 분산된 환경에서 안정적으로 운영할 수 있도록 설계되었습니다.

Kafka에서의 메시지는 데이터를 의미합니다.

1. Kafka의 주요 구성 요소

1.1. 브로커(Broker)

Kafka 데이터를 저장하고 클라이언트의 요청을 처리하는 서버
여러 브로커가 모여 Kafka 클러스터를 형성

1.2. 토픽(Topic)

메시지가 저장되는 논리적 구분 단위
각 토픽은 여러 파티션으로 나뉘어 데이터를 분산 저장

1.3. 파티션(Partition)

토픽을 구성하는 물리적 단위
데이터를 병렬로 처리할 수 있도록 분산 저장

1.4. 복제(Replication)

내결함성을 위해 파티션 데이터를 여러 브로커에 복제
replication-factor 로 복제본 수 설정

1.5. 프로듀서(Producer)

데이터를 Kafka에 전송하는 클라이언트
특정 토픽으로 메시지를 보냄

1.6. 컨슈머(Consumer)

Kafka에서 데이터를 소비하는 클라이언트
특정 토픽에서 메시지를 읽음

1.7. ZooKeeper (Kafka를 실행 시키기 위해 우선 실행)

Kafka 메타데이터를 관리하는 도구
Kafka 브로커의 상태를 모니터링하고 클러스터 구성을 관리

2. Kafka □□

2. Kafka 설치

Kafka, Zookeeper 설치 및 설정

1. 필수 요구사항 확인

1.1. Java 설치 확인

Kafka는 Java 기반으로 동작합니다. Java 8 이상이 필요하며, 설치 여부를 확인하세요.

```
java -version
```

- 설치되지 않은 경우, OpenJDK 또는 Oracle JDK를 설치합니다.

```
brew update
brew install openjdk@11
```

1.2. 디스크 공간 확인

Kafka는 로그 데이 `/tmp/kafka-logs` 공간이 필요합니다. 기본적으로

---

2. ZooKeeper 설치 및 실행

2.1. ZooKeeper 설치

ZooKeeper는 Kafka가 의존하는 분산 코디네이션 도구입니다. 아래 명령어로 설치합니다.

```
brew install zookeeper
```

2.2. ZooKeeper 실행

ZooKeeper를 시작하려면 다음 명령어를 사용합니다.

```
brew services start zookeeper
```

2.3. ZooKeeper 실행 상태 확인

ZooKeeper 실행 상태를 확인하려면 아래 명령어를 사용하세요.

```
brew services list
```

2.4. ZooKeeper 설정 파일 주요 항목

ZooKeeper 설 `/etc/zookeeper/conf/zoo.cfg` 에 위치합니다. 아래는 주요 설정 항목입니다

```
dataDir=/var/lib/zookeeper
clientPort=2181
maxClientCnxns=60
initLimit=10
syncLimit=5
server.1=localhost:2888:3888
```

- **`dataDir`**:
  - ZooKeeper 데이터(스냅샷과 트랜잭션 로그)가 저장되는 디렉터리입니다
  - ZooKeeper가 실행 중인 서버의 디스크 경로를 지정합니다
  - 예: `/usr/local/var/run/zookeeper`
- **`clientPort`**:
  - 클라이언트가 ZooKeeper에 연결하기 위한 네트워크 포트 번호입니다
  - [`2181` 이며, 클라이언트가 ZooKeeper에 요청을 보내는 진입점 역할을 합니다
- **`maxClientCnxns`**:
  - ZooKeeper에 연결할 수 있는 클라이언트의 최대 연결 수를 제한합니다
  - 하나의 클라이언트(또는 IP 주소)에서 과도한 연결을 방지하는 역할을 합니다
- **`initLimit`**:
  - ZooKeeper 클러스터에서 팔로워(follower)가 리더(leader)와 초기 데이터 동기화를 완료하는 데 필요한 시간을 설정합니다
  - 이 시간이 초과되면 동기화가 실패로 간주됩니다
- **`syncLimit`**:
  - ZooKeeper 팔로워가 리더와 데이터 동기화를 유지할 수 있는 시간(틱 수)을 설정합니다
  - 리더와의 동기화가 이 시간보다 오래 중단되면 팔로워가 재시작됩니다
- **`server.x`**:
  - ZooKeeper 서버가 클러스터 모드로 동작할 때, 각 서버의 정보를 설정합니다
  - 형 `server.<ID>=<호스트>:<follower-to-leader>:<leader-to-follower>`
    - `<ID>`: ZooKeeper 서버 ID
    - `<호스`:>ZooKeeper 서버의 호스트 이름 또는 IP
    - `<follower-to-leader>`: 팔로워가 리더에게 요청을 보낼 때 사용하는 포트
    - `<leader-to-follower>`: 리더가 팔로워와 연결할 때 사용하는 포트

---

3. Kafka 다운로드 및 설치

3.1. Kafka 다운로드

```
brew install kafka
```

3.2. Kafka □□

Kafka □□□□ □□□□□ ZooKeeper□ □□ □□□ □□□ □ □□ □□□□ □□

```
brew services start kafka
```

---

4. □□□□ □□

4.1. □□□□ □□

`~/.bashrc` □ `~/.zshrc` □□□ □□ □□ □□□ □□□□□.

```
export KAFKA_HOME=/usr/local/opt/kafka
export PATH=$PATH:$KAFKA_HOME/bin
```

4.2. □□□□ □□

```
source ~/.bashrc
```

---

5. Kafka □□ □□ □□

5.1. □□ □□ □□

Kafka □□ `/usr/local/etc/kafka/server.properties` □ □□□□□.

5.2. □□ □□ □□

```
broker.id=0
log.dirs=/tmp/kafka-logs
zookeeper.connect=localhost:2181
num.partitions=1
default.replication.factor=1
listeners=PLAINTEXT://0.0.0.0:9092
```

```
advertised.listeners=PLAINTEXT://localhost:9092
```

- `broker.id` : 브로커 ID를 설정합니다. 클러스터에서 고유해야 합니다
- `log.dirs` : Kafka 로그 데이터가 저장될 디렉터리 경로
- `zookeeper.connect` : ZooKeeper 주소 및 포트 정보
- `num.partitions` : 새로 생성되는 토픽의 기본 파티션 수
- `default.replication.factor` : 새로 생성되는 토픽의 기본 복제본 수
- `advertised.listeners` **:**
  - 역Kafka 브로커가 클라이언트에게 자신을 알릴 때 사용하는 주소를 설정합니다. 클라이언트는 이 주소를 통해 브로커와 통신
  - 값: `PLAINTEXT://localhost:9092`
  - `localhost` : 브로커가 클라이언트에게 공개하는 IP 주소입니다. 보통 외부 네트워크에서 접근 가능하도록 구성합니다.
  - `9092` : 클라이언트와 통신할 포트 번호입니다.
  - `PLAINTEXT` : 클라이언트가 이 브로커와 일반 텍스트 프로토콜로 통신함을 나타냅니다.
- `listners` **:**
  - 역Kafka 브로커가 요청을 수신할 때 사용할 네트워크 주소를 설정합니다.
  - 값: `PLAINTEXT://0.0.0.0:9092`
    - `PLAINTEXT` : 이 브로커가 클라이언트와 통신할 때 암호화되지 않은 일반 텍스트 프로토콜을 사용함을 나타냅니다.
    - `0.0.0.0` : 브로커가 모든 네트워크 인터페이스에서 연결을 수락하도록 설정합니다. 즉, 외부에서 접근할 수 있도록 열려 있는 상태입니다.
    - `9092` : 브로커가 사용할 포트 번호입니다.