



Apache Kafka 설치 및 구성 (단일 서버 구성)

References

- [Apache Kafka \(다운로드\)](#)
- [Apache Kafka Quick Start \(공식 문서\)](#)
- [Zookeeper vs KRaft](#)
- [Kafka 컨슈머 그룹 이해하기 \(참고\)](#)
- [Kafka Auto Commit 이란 \(참고\)](#)

STEP 1. Apache Kafka 다운로드

Binary downloads:

- Scala 2.12 - [kafka_2.12-3.4.1.tgz \(asc, sha512\)](#)
- Scala 2.13 - [kafka_2.13-3.4.1.tgz \(asc, sha512\)](#)

We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Kafka 3.4.1 fixes 58 issues since the 3.4.0 release. For more information, please read the detailed [Release Notes](#)

3.4.0

- Released Feb 7, 2023
- [Release Notes](#)
- Source download: [kafka-3.4.0-src.tgz \(asc, sha512\)](#)
- Binary downloads:
 - Scala 2.12 - [kafka_2.12-3.4.0.tgz \(asc, sha512\)](#)
 - Scala 2.13 - [kafka_2.13-3.4.0.tgz \(asc, sha512\)](#)

We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Kafka 3.4.0 includes a significant number of new features and fixes. For more information, please read our [blog post](#) and the detailed [Release Notes](#).

3.3.2

- Released Jan 23, 2023
- [Release Notes](#)
- Source download: [kafka-3.3.2-src.tgz \(asc, sha512\)](#)

안정적인 버전 테스트를 위해 3.4.0 버전을 설치, Scala 2.13 권장

STEP 2. Kafka 설치

다운로드 받은 tgz 파일 서버에 업로드 후 압축 해제 하여 설치

```
# 편의상 root 계정으로 실행
$ su - root

# Kafka 디렉토리 생성
$ mkdir /kafka

# 다운로드 받은 파일 /kafka 아래 업로드 이후 tar 파일 압축 해제
$ cd /kafka
$ tar -zxvf kafka_2.13-3.4.0.tgz
...
```

STEP 3. Kafka 라이브러리 확인, 설정 파일 확인

Kafka 에서 사용하는 라이브러리를 확인하고

별도의 Client 구성 시 해당 디렉터리에서 jar 파일 다운로드 하여 사용한다.

```
$ cd /kafka/kafka_2.13-3.4.0/libs/
$ ll
-rw-r--r--. 1 root root    69409  6월 30  2022 activation-1.1.1.jar
-rw-r--r--. 1 root root    27006  6월 30  2022 aopalliance-repackaged-2.6.1.jar
```

```
-rw-r--r--. 1 root root 90347 6월 30 2022 argparse4j-0.7.0.jar
-rw-r--r--. 1 root root 20437 6월 30 2022 audience-annotations-0.5.0.jar
-rw-r--r--. 1 root root 53820 6월 30 2022 commons-cli-1.4.jar
-rw-r--r--. 1 root root 501879 9월 15 2022 commons-lang3-3.8.1.jar
-rw-r--r--. 1 root root 102608 2월 1 2023 connect-api-3.4.0.jar
-rw-r--r--. 1 root root 19532 2월 1 2023 connect-basic-auth-extension-3
-rw-r--r--. 1 root root 15749 2월 1 2023 connect-file-3.4.0.jar
-rw-r--r--. 1 root root 32231 2월 1 2023 connect-json-3.4.0.jar
...
```

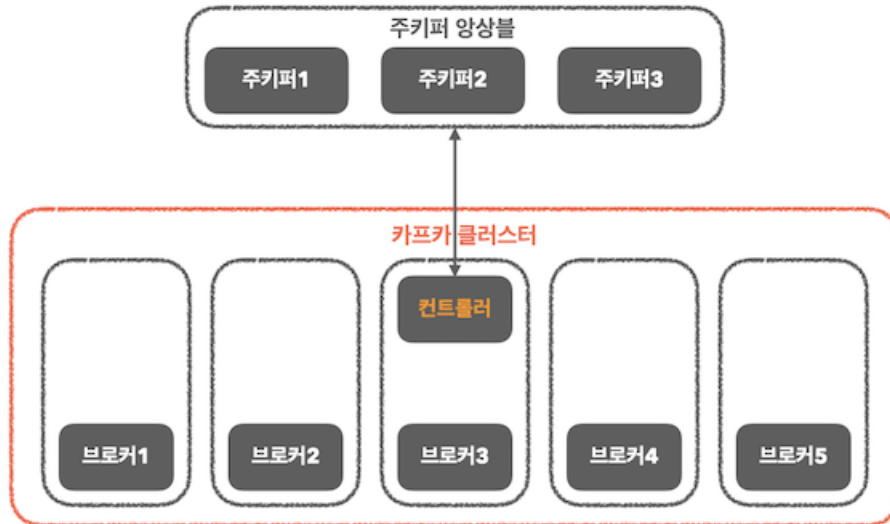
STEP 4. Kafka 설정 파일 확인 및 수정

Kafka 에 사용되는 컴포넌트들의 설정 파일을 확인하고, 수정한다.

```
$ cd /kafka/kafka_2.13-3.4.0/config/
$ ll
-rw-r--r--. 1 root root 906 2월 1 2023 connect-console-sink.properties
-rw-r--r--. 1 root root 909 2월 1 2023 connect-console-source.properties
-rw-r--r--. 1 root root 5475 2월 1 2023 connect-distributed.properties
-rw-r--r--. 1 root root 883 2월 1 2023 connect-file-sink.properties
-rw-r--r--. 1 root root 881 2월 1 2023 connect-file-source.properties
-rw-r--r--. 1 root root 2103 2월 1 2023 connect-log4j.properties
-rw-r--r--. 1 root root 2540 2월 1 2023 connect-mirror-maker.properties
-rw-r--r--. 1 root root 2262 2월 1 2023 connect-standalone.properties
-rw-r--r--. 1 root root 1221 2월 1 2023 consumer.properties
drwxr-xr-x. 2 root root 85 2월 1 2023 kraft
-rw-r--r--. 1 root root 4917 2월 1 2023 log4j.properties
-rw-r--r--. 1 root root 2065 2월 1 2023 producer.properties
-rw-r--r--. 1 root root 6904 11월 25 11:24 server.properties
-rw-r--r--. 1 root root 1032 2월 1 2023 tools-log4j.properties
-rw-r--r--. 1 root root 1169 2월 1 2023 trogdor.conf
-rw-r--r--. 1 root root 1205 2월 1 2023 zookeeper.properties
```

Kafka 는 **Apache Zookeeper** 또는 **KRaft (Apache Kafka Raft) 모드**를 사용하여 서버를 기동.

주키퍼 모드



- Zookeeper 란 ?

- Zookeeper 는 다수의 노드의 애플리케이션을 관리하기 위한 분산처리 도구로,
- Kafka 에서는 메타데이터, 브로커 상태, 토픽, 컨트롤러 등을 관리
- 어떤 브로커가 특정 파티션 및 토픽의 리더인지 결정하고, 리더 선택 (Quorum 아키텍처)

→ Kafka가 아닌 외부에서 메타데이터를 관리하여 확장성에 제한

(ex. 데이터 중복, 브로커의 메타데이터와 Zookeeper 의 메타데이터 불일치, 시스템 복잡성 증가, 서버 자원 필요 등)

KRaft 모드



• KRaft 모드 란?

- Zookeeper와의 의존성을 제거, Kafka를 단순화 되어 확장성, 안정성, 일관성에 도움
- 단일 Kafka 애플리케이션 내에서 메타데이터 관리 기능을 수행하는 독립적 구조로 변경
- Zookeeper 모드에서 한개 였던 컨트롤러가 세개로 늘어나고 하나의 컨트롤러가 리더 역할, 리더 역할의 컨트롤러가 Write 담당
- 카프카 내부의 별도 토픽을 사용하여 메타데이터 관리
- Zookeeper 에 비해 성능이 크게 향상

```
# zookeeper 설정 파일 확인
$ vi zookeeper.properties
...
# the directory where the snapshot is stored.
dataDir=/tmp/zookeeper
# the port at which the clients will connect
clientPort=2181
# disable the per-ip limit on the number of connections since this is a non-p
maxClientCnxns=0
# Disable the adminserver by default to avoid port conflicts.
# Set the port to something non-conflicting if choosing to enable this
admin.enableServer=false
# admin.serverPort=8080
~
```

```
# KRaft 모드 설정 파일들 확인
$ cd kraft/
$ ll
합계 24
-rw-r--r--. 1 root root 6136  2월  1  2023 broker.properties
-rw-r--r--. 1 root root 5765  2월  1  2023 controller.properties
-rw-r--r--. 1 root root 6340  2월  1  2023 server.properties
```

```
# Kafka 서버 설정파일 확인 (Zookeeper 사용)
$ vi server.properties
...
broker.id=0

##### Socket Server Settings #####

# The address the socket server listens on. If not configured, the host name
# java.net.InetAddress.getCanonicalHostName(), with PLAINTEXT listener name,
#   FORMAT:
```

```
# listeners = listener_name://host_name:port
# EXAMPLE:
# listeners = PLAINTEXT://your.host.name:9092
#listeners=PLAINTEXT://localhost:9092

# Listener name, hostname and port the broker will advertise to clients.
# If not set, it uses the value for "listeners".
advertised.listeners=PLAINTEXT://192.168.101.72:9092

# Kafka 와 외부 Client (Producer, Consumer) 간 연계를 위해
# 리스너를 IP 주소로 설정한 예시
```

Note. 리스너 설정 방법

- **listeners**
 - 카프카 브로커가 내부적으로 바인딩하는 주소.
- **advertised.listeners**
 - 카프카 Producer, Consumer 에게 노출할 주소.
 - 미 설정 시 디폴트로 listeners 설정이 적용된다.
 - 외부에서 호출 가능하도록 ip, 호스트 명 또는 도메인 주소를 입력하여 세팅

STEP 5. 단일 Kafka 서버 기동 (Zookeeper 사용)

```
$ cd /kafka/kafka_2.13-3.4.0/bin/

# Zookeeper 시작
$ nohup ./zookeeper-server-start.sh ../config/zookeeper.properties &

# Kafka 서버 시작
$ cd ..
$ nohup ./bin/kafka-server-start.sh config/server.properties &
```

STEP 6. Topic 생성 및 확인

```
$ cd /kafka/kafka_2.13-3.4.0/bin/
$ ./kafka-topics.sh --create --topic {Topic 명}
\ --bootstrap-server 192.168.101.72:9092

# quickstart-events 토픽 생성 예시 스크립트
$ ./kafka-topics.sh --create --topic quickstart-events --bootstrap-server loc
```

```
# 생성 된 토픽 확인 (describe 옵션)  
$ ./kafka-topics.sh --describe --topic quickstart-events --bootstrap-server 1
```