Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Apache Kafka 구성 및 관리

3 Installation, Cluster Configuration

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Apache Kafka Cluster 설치 및 구성

본 강의에 사용된 설치 소프트웨어 및 설치 환경 정보

1. Confluent Platform(CP) 사용
2. Ubuntu 18.04 환경에서 진행
3. Production 환경을 가정(Zookeeper 3 대, Broker 3 대, Control Center[1] 1 대 구성)
   - Kafka Broker 1 개만 사용할 경우(개발 환경), 무료로 모든 기능 사용 가능
   - Broker 2개 이상의 클러스터로 구성할 경우, 30일 기한의 라이선스가 자동 생성
   - 계속 사용이 필요할 경우 라이선스를 구매해야 함
4. Confluent Platform의 시스템 요구사항을 확인할 것
   https://docs.confluent.io/platform/current/installation/system-requirements.html
5. OpenJDK 혹은 OracleJDK 1.8 / 1.11 사용 가능(본 강의에서는 1.8 버전 사용)
6. 머신 간 NTP 시간 동기화

1) Confluent Control Center : Kafka 모니터링 및 관리를 위한 Web UI 도구

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

설치 파일 다운로드
Confluent Platform 다운로드

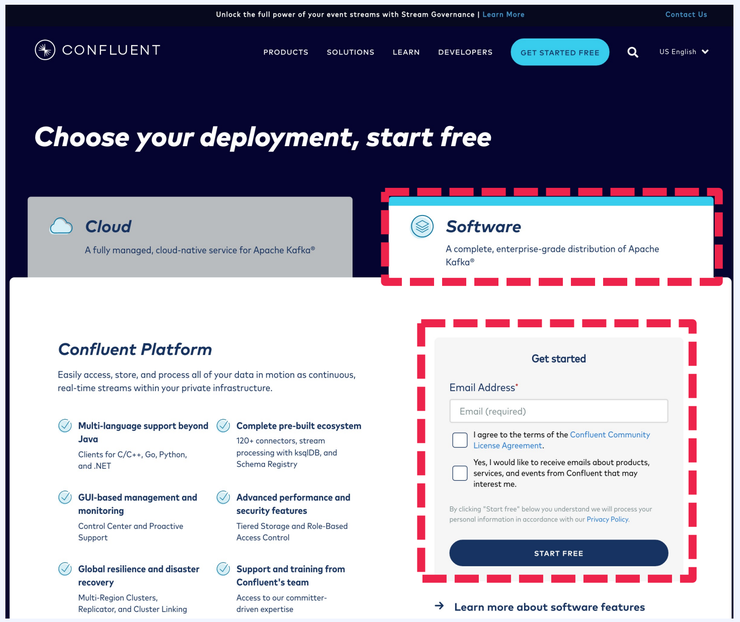
[Download URL]

https://www.confluent.io/get-started 에 접속

Software 탭을 클릭 후

Confluent Platform 다운로드를 위해 email을 입력한 후

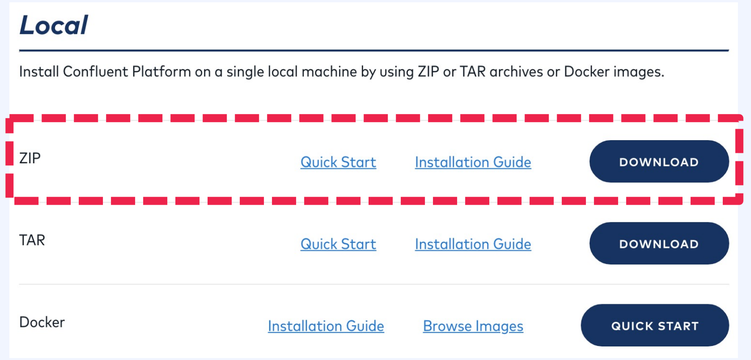
START FREE 버튼을 클릭



[Download Format]

ZIP, TAR, Docker 중에서 선택 가능

(본 강의에는 **ZIP 파일** 다운로드)



Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Kafka 설치를 위한 OS 관련 구성
Zookeeper 3 대, Broker 3 대, Control Center 1 대 모두 동일

1) Linux 계정 생성

Confluent Platform(Apache Kafka)은 Linux 환경에서 동작 : 관리 편의성을 위해 별도의 계정 생성을 권장

```
그룹 생성 : 예) groupadd confluent
사용자 생성 : 예) useradd -g confluent confluent
```

RHEL, CentOS, Debian, Ubuntu 지원[1)]
XFS File System 사용

2) JDK 구성

OpenJDK 혹은 Oracle JDK 1.8.0_202(64bit)를 다운로드[2)] 받은 후,

로그인한 계정의 Home 디렉토리에 압축파일을 옮기고 해당파일의 압축을 해제

Linux 계정의 환경변수에 Java Home 지정하고 bin 디렉토리를 PATH에 추가

```
$ vi ~/.bashrc
export JAVA_HOME=/home/confluent/jdk1.8.0_202          #압축파일을 푼 경로
export PATH=${JAVA_HOME}/bin:${PATH}

$ vi ~/.bash_profile
source ~/.bashrc

$  source ~/.bashrc                                    #환경변수 적용
```

1) https://docs.confluent.io/platform/current/installation/system-requirements.html#operating-systems
2) https://www.oracle.com/kr/java/technologies/javase/javase8-archive-downloads.html

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Confluent Platform 설치
Zookeeper 3 대, Broker 3 대, Control Center 1 대 모두 동일

3) Confluent Platform 압축 파일의 압축 해제

Confluent Platform ZIP 파일을 홈 디렉토리로 복사한 다음, 압축을 해제

계정의 환경 변수에 Confluent Platform Home을 지정하고 bin 디렉토리를 PATH에 추가

```
$ cp confluent-6.2.1.zip /home/confluent/

$ cd /home/confluent

$ unzip confluent-6.2.1.zip

$ vi ~/.bashrc
export CONFLUENT_HOME=/home/confluent/confluent-6.2.1      #confluent platform파일이 있는 경로
export PATH=${CONFLUENT_HOME}/bin:${PATH}

$  source ~/.bashrc                                                    # 환경변수 적용
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Zookeeper Node 구성
3 대 구성

${CONFLUENT_HOME}/etc/kafka/zookeeper.properties 파일 내의 중요한 파라미터

- Zookeeper Node들은 구성 파일이 동일해야 함

```
tickTime=2000
dataDir=/data/zookeeper/
clientPort=2181
initLimit=5
syncLimit=2
autopurge.snapRetainCount=3
autopurge.purgeInterval=24
# server.<myid>=<hostname>:<leaderport>:<electionport>
server.1=zookeeper1:2888:3888
server.2=zookeeper2:2888:3888
server.3=zookeeper3:2888:3888
```

- dataDir 파라미터에 설정한 디렉토리가 생성되어 있어야 하며, Linux 계정(confluent)을 디렉토리에 대한 소유자로 설정
- server.<myid>=<hostname>:<leaderport>:<electionport>
  - myid : 서버식별번호 → dataDir에 “myid”라는 이름의 파일을 생성하고 myid 값(예, 1 또는 2 또는 3)을 입력 후 저장
  - hostname : DNS 또는 /etc/hosts 에 등록된 zookeeper node의 hostname
  - leaderport : Follower가 Leader에 연결되는데 사용 (모든 zookeeper node 간에 연결되어야 함)
  - electionport : Zookeeper Leader 선출에 사용 (모든 zookeeper node 간에 연결되어야 함)

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Broker Node 구성
3 대 구성

${CONFLUENT_HOME}/etc/kafka/server.properties 파일 내의 중요한 파라미터

```
zookeeper.connect=zookeeper1:2181,zookeeper2:2181,zookeeper3:2181
broker.id=1
log.dirs=/data/broker
listeners=PLAINTEXT://broker1:9092
default.replicaiton.factor=3
min.insync.replicas=2
unclean.leader.election.enable=false
metric.reporters=io.confluent.metrics.reporter.ConfluentMetricsReporter
confluent.metrics.reporter.bootstrap.servers=broker1:9092,broker2:9092,broker3:9092
```

- zookeeper.connect : Broker에서 Zookeeper 를 연결하기 위한 파라미터
  zookeeper1:2181에서 zookeeper1은 DNS 또는 /etc/hosts 에 등록된 zookeeper node의 hostname
- broker.id : 정수로 표현되는 숫자 : 같은 Kafka cluster 내의 Broker들은 서로 다른 고유한 broker.id를 가져야 함
- log.dirs 파라미터에 설정한 디렉토리가 생성되어 있어야 하며, Linux 계정(confluent)을 디렉토리에 대한 소유자로 설정
- listeners : Broker가 수신할 URI의 쉼표로 구분된 목록
  broker1은 DNS 또는 /etc/hosts 에 등록된 broker1 node의 hostname
  kafka client가 Broker node와 연결할 때 사용하는 정보

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Control Center Node 구성
1 대 구성

${CONFLUENT_HOME}/etc/confluent-control-center/control-center-production.properties

파일 내의 중요한 파라미터

```
bootstrap.servers=broker1:9092,broker2:9092,broker3:9092
zookeeper.connect=zookeeper1:2181,zookeeper2:2181,zookeeper3:2181
confluent.controlcenter.data.dir=/data/controlcenter
confluent.controlcenter.schema.registry.url=http://schema1:8081
confluent.controlcenter.usage.data.collection.enable=true
# confluent.controlcenter.ksql.ksqldb.url=http://ksqldb1:8088,http://ksqldb2:8088
# confluent.controlcenter.ksql.ksqldb.advertised.url=http://ksqldb1:8088,http://ksqldb2:8088
```

- 개발 환경: 1대로 구성 가능
- 운영 환경: 2대 이상 Machine상에 설치,구성 권장(HA 목적)
- Control Center는 Zookeeper 및 Broker와 별개의 머신에 설치, 구성해야 함

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Confluent Platform 실행
순서대로 수행

1. Zookeeper 실행 (3 대의 Zookeeper Node에서 실행)

```
$ ${CONFLUENT_HOME}/bin/zookeeper-server-start \
${CONFLUENT_HOME}/etc/kafka/zookeeper.properties
```

2. Kafka Broker 실행 (3 대의 Broker Node에서 실행)

```
$ ${CONFLUENT_HOME}/bin/kafka-server-start \
${CONFLUENT_HOME}/etc/kafka/server.properties
```

3. Control Center 실행 (1 대의 Control Center Node에서 실행)

```
$ ${CONFLUENT_HOME}/bin/control-center-start \
${CONFLUENT_HOME}/etc/confluent-control-center/control-center.properties
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

nohup 스크립트 생성 및 실행

Kafka Cluster의 더욱 편리한 실행을 위해 스크립트를 사용

nohup Scripts[1)]

별도의 디렉토리를 생성 후 nohup 명령어 script를 만들어서 사용하는 것을 권장

예) start_[실행할 component 명].sh

stop_[실행할 component 명].sh

log_[실행할 component 명].sh

각 script는 각 컴포넌트를 실행하는 각 머신에서만 작성(예, Broker 머신에는 Broker용 스크립트만 작성)

- confluent 계정의 Home 디렉토리에서 bin 폴더 생성 후 bin 폴더로 이동, 작성 후 실행권한 부여

```
$ mkdir ~/bin
$ cd ~/bin
```

- 실행 권한 부여 (start 스크립트의 예)

```
$ chmod u+x start_[실행할 component 명].sh
```

1) nohup 은 “no hangups” 라는 의미로, Linux/UNIX 에서 Shell Script 를 데몬 형태로 실행시키는 명령어. 터미널이 끊겨도 실행한 프로세스는 계속 동작함

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Zookeeper용 Start nohup 스크립트

$ vi start_zookeeper1.sh

```
#!/usr/bin/sh

# process running
PID=`ps -efx | grep java | grep ZOOKEEPER1 | awk '{print $1}'`
if [ "$PID" != "" ]; then
        echo ZOOKEEPER1 ${PID} is running !!!
        exit
fi

mkdir -p /logs/zookeeper1
mv /logs/zookeeper1/zookeeper1.log /logs/zookeeper1/zookeeper1.log_`date +'%Y%m%d_%H%M%S'`

mkdir -p /logs/zookeeper1/gc

export KAFKA_GC_LOG_OPTS="-DZOOKEEPER1 -verbose:gc \
-Xloggc:/logs/zookeeper1/gc/verbosegc_zookeeper1.log_`date +'%Y%m%d_%H%M%S'` \
-XX:+PrintTenuringDistribution -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps \
-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=100 -XX:GCLogFileSize=50M"

export KAFKA_HEAP_OPTS="-Xms2g -Xmx2g -XX:+UseG1GC -XX:MaxGCPauseMillis=20 \
-XX:InitiatingHeapOccupancyPercent=35 -XX:MinMetaspaceFreeRatio=50"

nohup ${CONFLUENT_HOME}/bin/zookeeper-server-start \
${CONFLUENT_HOME}/etc/kafka/zookeeper.properties > /logs/zookeeper1/zookeeper1.log 2>&1 &
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Zookeeper용 Stop nohup 스크립트

$ vi stop_zookeeper1.sh

```
#!/usr/bin/sh

# process running
PID=` ps -efx | grep java | grep DZOOKEEPER1 | grep -v grep | awk '{print $1}'`
if [ "$PID" == "" ]; then
        echo ZOOKEEPER1 is NOT running !!!
        exit
fi

echo Process ${PID} will be stopped !!!

${CONFLUENT_HOME}/bin/zookeeper-server-stop
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Broker용 Start nohup 스크립트

$ vi start_broker1.sh

```
#!/usr/bin/sh

# process running
PID=`ps -efx | grep java | grep BROKER1 | awk '{print $1}'`
if [ "$PID" != "" ]; then
        echo BROKER1 ${PID} is running !!!
        exit
fi

mkdir -p /logs/broker1
mv /logs/broker1/broker1.log /logs/broker1/broker1.log_`date +'%Y%m%d_%H%M%S'`

mkdir -p /logs/broker1/gc

export KAFKA_GC_LOG_OPTS="-DBROKER1 -verbose:gc \
-Xloggc:/logs/broker1/gc/verbosegc_broker1.log_`date +'%Y%m%d_%H%M%S'` \
-XX:+PrintTenuringDistribution -XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps \
-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=100 -XX:GCLogFileSize=50M"

export KAFKA_HEAP_OPTS="-Xms6g -Xmx6g -XX:MetaspaceSize=96m -XX:+UseG1GC -XX:MaxGCPauseMillis=20 \
-XX:InitiatingHeapOccupancyPercent=35 -XX:G1HeapRegionSize=16M -XX:MinMetaspaceFreeRatio=50 \
-XX:MaxMetaspaceFreeRatio=80"

nohup ${CONFLUENT_HOME}/bin/kafka-server-start ${CONFLUENT_HOME}/etc/kafka/server.properties >
/logs/broker1/broker1.log 2>&1 &
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Broker용 Stop nohup 스크립트

$ vi stop_broker1.sh

```
#!/usr/bin/sh

# process running
PID=`ps -efx | grep java | grep BROKER1 |awk '{print $1}'`
if [ "$PID" == "" ]; then
        echo BROKER1 is NOT running !!!
        exit
fi

echo Process ${PID} will be stopped !!!

${CONFLUENT_HOME}/bin/kafka-server-stop
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Control Center용 Start nohup 스크립트

$ vi start_c3.sh

```
#!/usr/bin/sh

# process running
PID=`ps -efx | grep java | grep CONTROL_CENTER1 | awk '{print $1}'`
if [ "$PID" != "" ]; then
        echo CONTROL_CENTER1 is running !!!
        exit
fi

mkdir -p /logs/controlcenter1
mv /logs/controlcenter1/controlcenter1.log /logs/controlcenter1/controlcenter1.log_`date +'%Y%m%d_%H%M%S'`

mkdir -p /logs/controlcetner1/gc

export CONTROL_CENTER_HEAP_OPTS="-Xms8g -Xmx8g -DCONTROL_CENTER1 \
-Xloggc:/logs/controlcenter1/gc/verbosegc_controlcenter1.log_`date +'%Y%m%d_%H%M%S'` \
-XX:-PrintGCDetails -XX:-PrintGCTimeStamps -XX:-PrintTenuringDistribution -XX:-UseGCLogFileRotation \
-XX:NumberOfGCLogFiles=100 -XX:GCLogFileSize=10M"

export CONTROL_CENTER_OPTS="-Djava.io.tmpdir=/logs/controlcenter1"

nohup ${CONFLUENT_HOME}/bin/control-center-start ${CONFLUENT_HOME}/etc/confluent-control-
center/control-center-production.properties > /logs/controlcenter1/controlcenter1.log 2>&1 &
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Control Center용 Stop nohup 스크립트

$ vi stop_c3.sh

```
#!/usr/bin/sh

# process running
PID=`ps -efx | grep java | grep CONTROL_CENTER1 | awk '{print $1}'`
if [ "$PID" == "" ]; then
        echo CONTROL_CENTER1 is NOT running !!!
        exit
fi

echo Process ${PID} will be stopped !!!

${CONFLUENT_HOME}/bin/control-center-stop
```

Fast campus Copyright FASTCAMPUS Corp.All Rights Reserved

Summary

- Confluent Platform Download
- Installation
- Configuration
- nohup scripts