



실습

▼ Lab 1. Amazon MSK 구성하기

1. MSK Cluster 생성하기

| 클러스터 이름 | 상태 | 인증 | Apache... | 브로커 유형 | 영역당... | 영역 | 생성 |
|----------------|----|-----------------------|-----------|----------------|--------|----|------|
| mk-msk-cluster | 활성 | Unauthenticated, I... | 2.8.1 | kafka.t3.small | 1 | 2 | 2023 |

2. MSK Client 생성하기

| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 | 가용 영역 | 피블릭 IPv4 DNS |
|----------------|---------------------|---------|----------|-------|-------|-----------------|-----------------|
| kmk-msk-client | i-046e6ae040a11efd9 | 실행 중 | t2.micro | 초기화 | 경보 없음 | ap-northeast-2a | ec2-3-35-134-20 |

3. Tabby로 MSK Client 연결하여 Topic 생성하기

```
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-topics.sh --list --bootstrap-server b-1.mkmkcluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092,b-2.mkmkcluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092
2
amazon_msk_canary
consumer_offsets
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-topics.sh --create --bootstrap-server b-1.mkmkcluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092,b-2.mkmkcluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092 --replication-factor 2 --partitions 1 --topic kmk-topic
Created topic kmk-topic.
[ec2-user@ip-172-16-1-19 bin]$
```

4. 생성한 Topic으로 연결 테스트하기

producer 창

```
[ec2-user@ip-172-16-1-19 bin]$ pwd  
/home/ec2-user/kafka_2.12-2.8.1/bin  
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-console-producer.sh --bootstrap-server b-1.mkscluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092,b-2.mkscluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092 --topic kmk-topic  
>Hello  
>반가워요
```

```
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-console-producer.sh --bootstrap-serv  
9092 --topic kmk-topic  
>Hello  
>반가워요
```

consumer 창

```
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-console-consumer.sh --bootstrap-serv  
9092 --topic kmk-topic --from-beginning  
>Hello  
>반가워요
```

▼ Lab 2. Kafka Producer 애플리케이션 개발

Tabby에서 producer.py 파일 작성

```
from kafka import KafkaProducer  
from json import dumps  
import time  
  
producer = KafkaProducer(  
    acks=0,  
    bootstrap_servers =[  
        'b-2.mkscluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092',  
        'b-1.mkscluster.3t5mfs.c4.kafka.ap-northeast-2.amazonaws.com:9092'  
    ],  
    value_serializer=lambda x : dumps(x).encode('utf-8')  
)  
topic = 'kmk-topic'  
start = time.time()  
for i in range(20) :  
    data = {'num' : str(i)}  
    producer.send(topic, value=data)  
    producer.flush()  
    print('sended data : ', data)  
    time.sleep(3)  
  
print('elapsed : ', time.time() - start)
```

producer 창에서 producer.py 실행

```
[ec2-user@ip-172-16-1-19 ~]$ python3 producer.py
sended data : {'num': '0'}
sended data : {'num': '1'}
sended data : {'num': '2'}
sended data : {'num': '3'}
sended data : {'num': '4'}
sended data : {'num': '5'}
sended data : {'num': '6'}
sended data : {'num': '7'}
sended data : {'num': '8'}
sended data : {'num': '9'}
sended data : {'num': '10'}
sended data : {'num': '11'}
sended data : {'num': '12'}
sended data : {'num': '13'}
sended data : {'num': '14'}
sended data : {'num': '15'}
sended data : {'num': '16'}
sended data : {'num': '17'}
sended data : {'num': '18'}
sended data : {'num': '19'}
elpased : 60.07647228240967
```

consumer 창에서 메시지 들어오는 것 확인

```
[ec2-user@ip-172-16-1-19 bin]$ ./kafka-console-consumer.sh --bootstrap-server 172.16.1.19:9092 --topic kmk-topic
{"num": "1"}
{"num": "2"}
{"num": "3"}
{"num": "4"}
{"num": "5"}
{"num": "6"}
{"num": "7"}
{"num": "8"}
{"num": "9"}
{"num": "10"}
{"num": "11"}
{"num": "12"}
{"num": "13"}
{"num": "14"}
{"num": "15"}
{"num": "16"}
{"num": "17"}
{"num": "18"}
{"num": "19"}
```

▼ Lab 3. 2개 인스턴스로 Kafka Application 실행하기

1. MSK-Client Image 생성하기

| 스냅샷 (1) 정보 | | | | | | |
|--------------------------|---------------------|---------------|--------|------------------------------|--------|---|
| 내 소유 ▾ | | 검색 | | | | |
| | Name | 스냅샷 ID | 크기 | 설명 | 스토리... | 스냅샷 상태 |
| <input type="checkbox"/> | kmk-msk-client1-img | snap-08082... | 30 GiB | Created by CreateImage(i...) | 표준 |  완료됨 |

| 인스턴스 (1) 정보 | | | | | | |
|---------------------------------------|----------------|---------------------|--|--|---|----|
| Q 인스턴스를 속성 또는 (case-sensitive) 태그로 찾기 | | | | | | |
| | Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 연결 |
| <input type="checkbox"/> | kmk-msk-client | i-046e6ae040a11efd9 |  실행 중 |  t2.micro |  초기화 | |

2. 앞에서 생성한 이미지를 이용하여 새 인스턴스 생성하기

이름

추가 태그 추가

▼ 애플리케이션 및 OS 이미지(Amazon Machine Image) 정보

AMI는 인스턴스를 시작하는 데 필요한 소프트웨어 구성(운영 체제, 애플리케이션 서버 및 애플리케이션)이 포함된 템플릿입니다. 아래에서 찾고 있는 항목이 보이지 않으면 AMI를 검색하거나 찾아보십시오.

수천 개의 애플리케이션 및 OS 이미지를 포함하는 전체 카탈로그 검색

최근 사용 내 AMI Quick Start

내 소유 나와 공유됨

 더 많은 AMI 찾아보기

AWS, Marketplace 및 커뮤니티의 AMI 포함

Amazon Machine Image(AMI)

| |
|---|
| msk-client-img ami-05a0466b345c7b5d7 |
|---|

- 인스턴스 2개 확인

| 인스턴스 (2) 정보 | | | | | | | | | |
|--|---------------------|-------------------------------|----------|---------------|-------|-------------------|---------------|-----------|--|
| <input type="text"/> 인스턴스를 속성 또는 (case-sensitive) 태그로 찾기 | | <input type="button"/> 연결 | | 인스턴스 상태 ▾ | | 작업 ▾ | | 인스턴스 시작 ▾ | |
| <input type="button"/> 인스턴스 상태 = running X | | <input type="button"/> 필터 지우기 | | | | | | | |
| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 | 가용 영역 | 퍼블릭 IPv4 | | |
| kmk-msk-client | i-046e6ae040a11efd9 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | ec2-13-125-11 | | |
| kmk-msk-client-1 | i-00074c56f6ec76f97 | 실행 중 | t2.micro | 초기화 | 경보 없음 | + ap-northeast-2a | ec2-3-38-210- | | |

3. 인스턴스 [Name] 변경하기

태그 관리 정보

태그는 AWS 리소스에 할당하는 사용자 지정 레이블입니다. 태그를 사용하여 인스턴스를 정리하고 식별할 수 있습니다.

| 키 | 값 - 선택 사항 |
|----------------------------------|---|
| <input type="text"/> Name X | <input type="text"/> kmk-kafka-producer X <input type="button"/> 제거 |
| <input type="button"/> 새로운 태그 추가 | |

최대 49개의 태그를 더 추가할 수 있습니다.

태그 관리 정보

태그는 AWS 리소스에 할당하는 사용자 지정 레이블입니다. 태그를 사용하여 인스턴스를 정리하고 식별할 수 있습니다.

| 키 | 값 - 선택 사항 |
|----------------------------------|---|
| <input type="text"/> Name X | <input type="text"/> kmk-kafka-consumer X <input type="button"/> 제거 |
| <input type="button"/> 새로운 태그 추가 | |

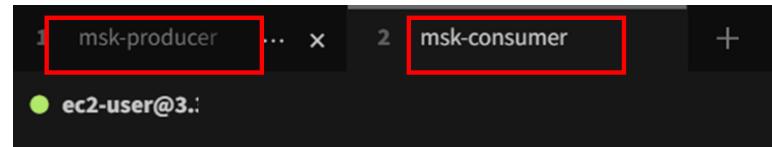
최대 49개의 태그를 더 추가할 수 있습니다.

- 인스턴스 목록에서 producer와 consumer 인스턴스 확인

| 인스턴스 (2) 정보 | | | | | | | | | |
|--|---------------------|-------------------------------|----------|---------------|-------|-------------------|---------------|-----------|--|
| <input type="text"/> 인스턴스를 속성 또는 (case-sensitive) 태그로 찾기 | | <input type="button"/> 연결 | | 인스턴스 상태 ▾ | | 작업 ▾ | | 인스턴스 시작 ▾ | |
| <input type="button"/> 인스턴스 상태 = running X | | <input type="button"/> 필터 지우기 | | | | | | | |
| Name | 인스턴스 ID | 인스턴스 상태 | 인스턴스 유형 | 상태 검사 | 경보 상태 | 가용 영역 | 퍼블릭 IPv4 | | |
| kmk-kafka-producer | i-046e6ae040a11efd9 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | ec2-13-125-11 | | |
| kmk-kafka-consumer | i-00074c56f6ec76f97 | 실행 중 | t2.micro | 2/2개 검사 통과... | 경보 없음 | + ap-northeast-2a | ec2-3-38-210- | | |

4. Tabby connection 설정 변경

- Tabby 창 2개(producer, consumer)



- msk-producer의 hostname 변경하기

```

1 msk-producer ... x 2 msk-consumer + 🖥
● ec2-user@

Last login: Tue May 23 15:06:55 2023 from 124.194.99.173
[ec2-user@ip-172-16-1-19 ~]$ sudo hostnamectl set-hostname msk-producer #msk-producer로 이름을 바꿈
[ec2-user@ip-172-16-1-19 ~]$ hostnamectl #이름 바뀌었는지 확인
Static hostname: msk-producer
Icon name:
Chassis:
Machine ID:
Boot ID:
Virtualization:
Operating System:
CPE OS Name:
Kernel:
Architecture: ...
[ec2-user@ip-172-16-1-19 ~]$ sudo reboot #재부팅(프롬포트에서는 이름이 안 바껴서)

SSH 13.125.11.140: session closed
Press any key to reconnect

```

- msk-consumer의 hostname 변경하기

```
1 msk-producer ... x 2 msk-consumer + └
● ec2-user@3.38.210.30:22

Last login: Tue May 23 15:06:55 2023 from 124.194.99.173
[ec2-user@ip-172-16-1-117 ~]$ sudo hostnamectl set-hostname msk-consumer #msk-consumer로 이름을 바꿈
[ec2-user@ip-172-16-1-117 ~]$ hostnamectl #이름 바꿔있는지 확인
static hostname: msk-consumer
    Icon name:
        Chassis:
    Machine ID:
        Boot ID:
Virtualization:
Operating System:
    CPE OS Name:
        Kernel:
Architecture:
[ec2-user@ip-172-16-1-117 ~]$ sudo reboot #재부팅(프롬포트에서는 이름이 안 바껴서)

SSH 3.38.210.30: session closed
Press any key to reconnect
```