

1 Lab5. Kafka Consumer Application을 ECR에 배포하기

2 1. Amazon ECR에서 리포지토리 생성하기

- 3 1)[서비스] > [컨테이너] > [Elastic Container Registry] 이동한다.
- 4 2)[리포지토리 생성] > [시작하기] 버튼 클릭
- 5 3)[리포지토리 생성] 페이지에서
- 6 -[일반설정] > [표시 여부 설정] : [프라이빗]
- 7 -[일반설정] > [리포지토리 이름] : msk/consumer
- 8 4)나머지는 기본값 그대로 사용한다.
- 9 5)[리포지토리 생성] 버튼 클릭

10 2. 이미지 생성 하기

11 1)Dockerfile과 requirements.txt 파일 복사하기

```
12 $ pwd
13 /home/ec2-user/kafka-client/producer
14
15 $ cp requirements.txt ../consumer/requirements.txt
16 $ cp Dockerfile ../consumer/Dockerfile
```

17 2)consumer 디렉토리의 Dockerfile 수정

```
18 FROM python:3
19
20 ENV PYTHONUNBUFFERED 1
21 RUN mkdir /kafka-client
22 WORKDIR /kafka-client
23 COPY requirements.txt /kafka-client/
24 RUN pip install -r requirements.txt
25 COPY consumer.py /kafka-client/consumer.py <----여기 수정
26 CMD ["python", "./consumer.py"] <----여기 수정
```

27 3)Docker Image 생성하기

```
28 $ docker build -t kafka-consumer . <---마지막 . 주의
```

29 4)생성된 Image 확인하기

| REPOSITORY   | TAG    | IMAGE ID     | CREATED            | SIZE  |
|--|--------|--------------|--------------------|-------|
| kafka-consumer   | latest | 0057736d6552 | 5 seconds ago      | 936MB |
| 789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/producer | latest | 0726d0a60267 | About a minute ago | 936MB |
| kafka-producer   | latest | 0726d0a60267 | About a minute ago | 936MB |
| python   | 3      | 815c8c75dfc0 | 5 days ago         | 920MB |

30 5)이미지에 tag 달기

```
31 $ docker tag kafka-consumer:latest {Amazon ECR 리포지토리 URI}:latest
32 ex) $ docker tag kafka-consumer:latest 789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/consumer
```

33 6>tag 확인하기

```
34 [ec2-user@ip-172-16-1-206 consumer]$ docker images
35 REPOSITORY TAG IMAGE ID CREATED SIZE
36 789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/consumer latest 0057736d6552 About a minute ago 936MB
37 kafka-consumer latest 0057736d6552 About a minute ago 936MB
38 789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/producer latest 0726d0a60267 2 minutes ago 936MB
39 kafka-producer latest 0726d0a60267 2 minutes ago 936MB
40 python 3 815c8c75dfc0 5 days ago 920MB
```

41 3. ECR에 배포하기

42 1)ECR에 로그인하기

```
43 $ aws ecr get-login --no-include-email --region ap-northeast-2
44 docker login -u AWS -p
45 eyJwYXlsb2FkIjoia09QWJbB1Zk5kL1RUWkNYUEozNWNwYTl1YjFDbEZFc1Q2a3NNVS9vSII5RHRscHk2elNDDeEdaMmZQUVdsOTTh3
46 SmxBd3dXM21zWDhyT1RrUmtxd0M5bzNHV2VWVWRyeS82Ny8yc2gwb1BVNmEyanErVXlwak94dXYwWDNqMGIDSkdkTzVITjM
47 3YzdQMTUxL29kZ1dNN1RvU29sdldhoZXPHE9iVnhqc1hIU3c2eFhKSXVwZIRYzGxxUW4wekZLWmxQaEVvZmh0REViNEdvoEgV
48 RVJ0d2NsWGJyS1V4ZU0wd1ZGdkZkZ2ZUcDBvVnhGNkhjcVlmbWgrelExVGZxSIBtOHV4SnJ1enlzOGIibXNIMC8vY21mTnFwbWt
49 xVFFEd0pCZzF0amF5L0JoTWQ0M0xMYWJ0Nm4vWHVZS1IGZkhldjZSWFZQTfHqUllkUVlkUjlpTEc5WHI3TlVoZWWhxRlV2ZmQwU2
50 1zSHB2R1dOWUN0NkUyWm9lejV0emIFakVSRVZMUTZSeWFZY3V1MDdMemtBNkVHU210RnBoVmsyZcJveKoyOUg0N2QzY2Y3R
51 DEvdmg2eFV3UHZvd2xhNWVRUmxGa3pMcHdnUmwvc1IK0pYCTjXVUNyZVRPUkRjZVp6S01PdU1NTEpPU1dQbKxIVEprRnQ0V
52 Wo1M2JialFFbjk1bDJKv1ZYVzRvWm1VeFhQSERONUx0eW4rcnE0VFPoSmRqNEV1UVVZcDhwRmNUTGtsdKtaK3lvUkJIL0dXY2JP
53 NmVKNm5qTmZIU1dwWnBHR29vK3BqYzlnZHNHMKf3cEx6NEtWZ1V5aDRNd1BWZGRtVmVuVHFYZDdPUkFIMXhkZGdVR2xiY1
54 RsWkI3b2RvN3ZwTUtvckJuMTJaRTIWFVGMGlRyJRT2t3YWd4cUpvcWE5dnPjBXdUNm01T3NPWThTWGQyeGtSbWd6ejZVRDJ
55 YVZFI0pDwKRlK2ZReHJxOXV2SmtYd002UVB0eGhmdzlhRWxURS9DWHduWGNUem9vS0NmbEJmZnlCaU9lc1k4M1I2YUZhbn
56 d4Wfd5M3YyNUtDOHljdU94TkFoRmRjdVpET3o4eE9aMHIUc1pzL3pNOTJYJh0WHpacnRaSmR6ZTcxUzdzQkJs1IWcTV3TEhjbnpN
57 WIVCY2ZXZzc2a3dUcUcHJHNXBWVEtkZHY4MHBOMTY3Ni9NSDdzQ3FtNm9Sc0ZFa0IHwJ6RitvSTY0bTBVSkZiYUd0Y2NsbkjhUIBo
58 S0UyOTFHeE1GYUhYEFF5LzBrM0c5RCtGdENTNG9tWWdSdTQzbU94MDZpZ1BjNDlieVdaQjErS3VzdzM4VWt3d1NUMGhZ2ZhoZ
59 G5PeTR0Nyt4K2pjTnB1Q3pLejJ3cVM3Y0sxcjNWM0JCeU1kQjZ0UW5lSVY5QUhiRitzUFARA1raExzRkFGYy83UTJLaHZscm5nYm
60 RqTFZVTzVQL1VaZWdyT0VodjloL0FtQWFUblRnYmcyRfhyRzBhSUhHaUg0aVdONE13QklmZXljWm81ajU3OWhhTHlqcUZYdkQza
61 TdZQnlaZmxaSnI4VWJwbm10bUZIzZfTaGhWSHZ1RWVEWStvbm54c1RYUUVLKzlpbGxsc1Y3WFJEbmJPYnJxazExSWpTa3hDel
62 htYUirekZyei9kdmhQR3dvMXV4N1FLa3l6VjRXU2hmSm96ZW1IbStKUXhpQ1NXVzF1UW1ER3grL24yZDI4UIZ4R29SZFNuVDRIT
63 3NOKy9wVm1hckpTelovSHdhVFZ6TGdpYVWVM3gwZDY0a2hmQUE4NXRIbW1UbNvUU1IEYII2dXdsWlN5VjA4UmhMb2Mrb1pw
64 Z2UzOHowVjM3YkpjNGQrOFRVSGwxejhiY2piUzd0UUCvRkpxNjdBeG9kcDdXWGpK2JETXpJZWw0cG53VTJOM3dyN2l1anV0b3B
65 rUDFvU25KYndESERMNTdscUExCHEzanZnYzREZEo5YmhobG1yUEISRmpoZ3F1ZXIOMjI3SWhpS1dsRTRsNTIPYVJmNU5WZ3dW
66 U2Y2ZDF4WfPcV1JBUSTpNmV0cEzUvXfSS2tISfkwVnVsDWhRcFbnWkNZb3djdnnpjZlXZSxRiWUtPVXhIUeJ5Sk9Ra3Z4WTRKYVI
```

```
XRjBQKzBqSjJSQWZpc0Z3OTFQdEtaemd2YndSS0pkMmRyUGpKTTZVQkJDUmNCOXFNMittYXh1dmdsNTJHWTZYbURRaXMrQi9
QZmx6Qmowem4vOVNCA09WZ0dNbUtycTVMZ1Z2OGkxRDVxT0grTS9qTkI2U1BpZUFIrDhLTWFmUEF5TG9XcE9Rb2tOM215Y
k1KRkJ3YitjVytWSnQrNTBoTnVvTU90VlpMMXEyTGVyS2NvYk94ejl1b1ZDRzdpa0gxM0gzOWpKTXZ2S3JhRytWK1ZncjJlcTI5Y0dj
a2VUL0J5eXBtN0hQNXE3N2FzSHc1amF3clZIMWMrNnRWY3dERXNTU3BSVtIkCWIMZWZjK2wxUmNHQXRDTzAwamZ5cW1Fd1E
3S09IQjRJZXMMyNVVtM21wSmpCUWUxQVB1TDdOeXRCNXBRbXN0QXFzU3pObjFVQWRaTE0yY0QzaklHc3haeHArZjJHb1lOcWRJ
eHFWVvK1OMERsU0pXY3FMiwiZGF0YWtleSI6IkFRSUJBSGhBT3NhVzJnWk4wOVdOdE5Ha1ljOHFwMTF4U2haL2RyRUVveTFIazh
MWFDnRjBXa3pYYVdVTGV0NU5obHBMB1cxUkFBQUFmakI4QmdrcWhraUc5dzBCQndhZ2J6QnRBZ0VBTUdnR0NTcUdTTSWIzRFF
FSEFUQWVCZ2xnaGtnQlpRTUVBUzR3RVFRTXNteGVGOTdtWlY0ZFRJU3IBZ0VRZ0R1REhnOHY0eExjY2MydTmWk2VvTjRxTEl3bl
F3dEZrZ3BrQlpoTDBEazhKbEJHT2xuZXc4SDJubDhHKzdvdTHVxMVVIQnNIMG9oQTRCTG5PdZ09IiwidmVyc2lvdjI6IjIiLCJ0eXBIIjo
iREFUQV9LRVkiLCJleHBpcmF0aW9uIjoxNjgzNjQyOTQ5fQ== https://789534828835.dkr.ecr.ap-northeast-2.amazonaws.com
```

2)-받은 키를 이용하여 로그인하기

```
$ $(aws ecr get-login --no-include-email --region ap-northeast-2)
```

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

3)ECR에 push하기

```
$ docker push {Amazon ECR 리포지토리 URI}:latest
```

```
ex) $ docker push 789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/consumer:latest
```

The push refers to repository [789534828835.dkr.ecr.ap-northeast-2.amazonaws.com/msk/consumer]

f0e9d1ee8620: Pushed

ef6477ec5c6e: Pushed

5c2378dd1da8: Pushed

cbdbb83b75a5: Pushed

889828f94722: Pushed

b0f747a5176b: Pushed

7dac77253a2b: Pushed

0007505dc811: Pushed

f43725f97b9f: Pushed

9c42af2c6418: Pushed

d96e248f10e6: Pushed

d925e0fae4e6: Pushed

latest: digest: sha256:7790944be5c6332c50705fd3a893e6dbb19d280df5af6c52041d29ad66e1ee66 size: 2840

4)ECR에서 Image 확인하기

-Amazon ECR > 리포지토리 > msk/consumer에서 이미지 확인

4. ECS에서 이미지 사용을 위한 새 Task 정의 생성

1)ECS 좌측 메뉴 중 [태스크 정의] 클릭

2)[태스크 정의]페이지에서 [새 태스크 정의 생성] > [새 태스크 정의 생성] 선택

3)[태스크 정의 및 컨테이너 구성] 페이지에서

-[태스크 정의 구성] > [태스크 정의 패밀리] : {계정}-consumer-task

-[컨테이너 - 1]

--[컨테이너 세부 정보] > [이름] : {계정}-kafka-consumer-container

--[컨테이너 세부 정보] > [이미지 URI] : {Amazon ECR 리포지토리 msk/consumer의 이미지 URI}

--나머지는 기본값 그대로

--[다음] 버튼 클릭

4)[환경, 스토리지, 모니터링 및 태그 구성] 페이지에서,

-[환경]

--[앱 환경] : [AWS Fargate(서버리스)]

--[운영 체제/아키텍처] : Linux/X86\_64

--[CPU] : .25 vCPU

--[메모리] : .5GB

-나머지는 기본값 그대로 사용

-[다음] 버튼 클릭

5)[검토 및 생성] 페이지에서 [생성] 버튼 클릭

5. 생성한 태스크 정의로 태스크 생성 후 실행하기

1)방금 생성한 태스크 {계정}-consumer-task:1 상세페이지에서 [배포] > [태스크 실행] 클릭

2)[생성] 페이지에서

-[환경]

--[기존 클러스터] : {계정}-ecs-cluster

--[컴퓨팅 옵션] : [시작 유형] 선택

--[시작 유형] : FARGATE

--[플랫폼 버전] : LATEST

--[배포 구성] > [애플리케이션 유형] > [태스크]

-[네트워킹]

--[VPC] : {계정}-datalake-vpc

--[서브넷] : {계정}-datalake-subnet-2a

--[보안 그룹] > [기존 보안 그룹 선택] > {계정}-datalake-sg

-나머지 값은 기본값 그대로

-[생성] 클릭

3)[태스크] 목록에서 방금 생성한 태스크 확인

-[마지막 상태] : "실행 중"이 될 때까지 대기

4)consumer는 producer가 발행해야 메시지를 출력하기 때문에 Terminal에서 producer.py를 실행한다.

```
$ python3 producer.py
```

135 5)[태스크] 값의 링크를 클릭한다.  
136 -[로그] 탭에서 결과를 확인한다.  
137  
138  
139 6. 두개의 Container 테스트해보기  
140 1){계정}-ecs-cluser > [태스크 정의] 페이지에서  
141 1){계정}-producer-task 선택후, [배포] > [태스크 실행] 을 실행해서 [마지막 상태] : "실행 중"이 될 때까지 대기  
142 2){계정}-consumer-task 선택후, [배포] > [태스크 실행] 을 실행해서 [마지막 상태] : "실행 중"이 될 때까지 대기  
143  
144 2)2개의 태스크가 각각 [마지막 상태]가 "실행 중"이 되면 각 태스크 상세 페이지의 [로그] 탭에서 결과 값 확인