

클라우드컴퓨팅AI SageMaker 보강

2023 1학기 14주차

AI융합학부

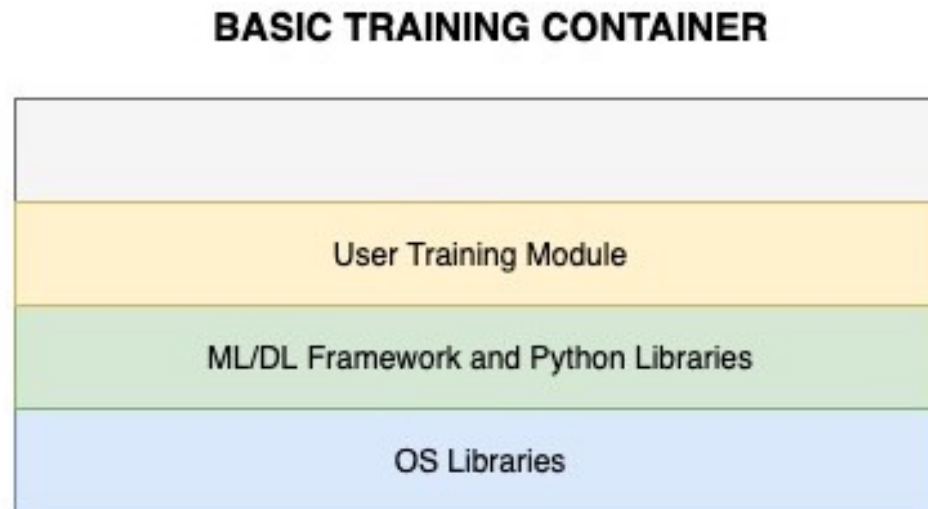
교수 오장민

개요

- Amazon SageMaker Custom Training Containers
- 참고
 - <https://github.com/aws/amazon-sagemaker-examples.git>
 - 해당 리포 아래: advanced_functionality/custom-training-containers/
- 절차
 - CLI 수준에서 on-demand 훈련 요청
 - 학습이 수행될 실행 환경: 컨테이너
 - SageMaker가 리소스 할당 후 컨테이너 기동
 - 컨테이너 내에서 학습 코드 실행
 - 훈련 종료되면 리소스 제거
 - 사용한만큼만 과금

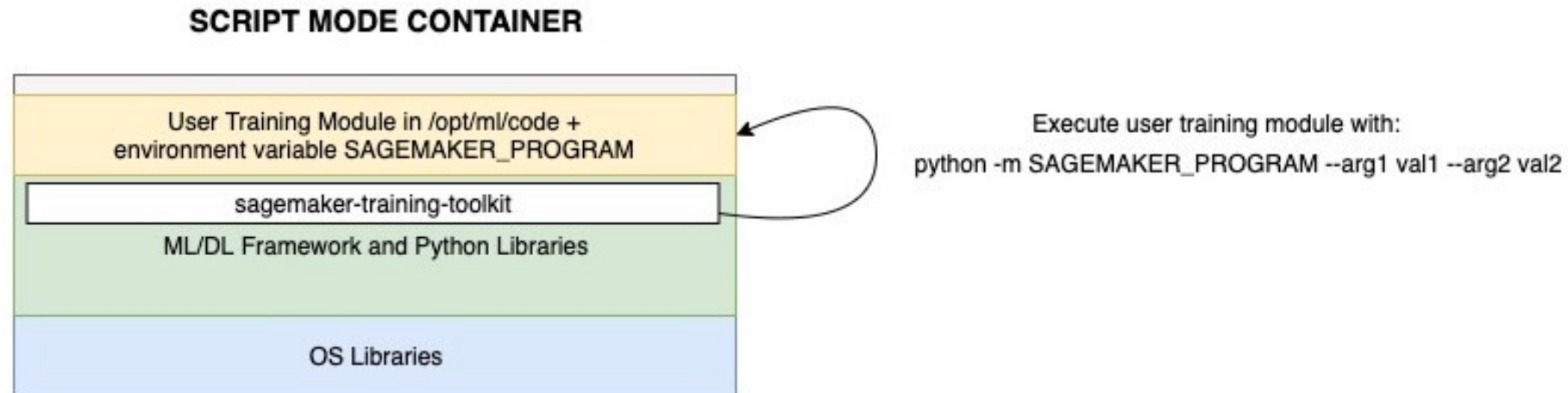
Basic Training Container

- 학습 모듈까지 모든 것을 도커 이미지화
 - 도커 이미지 제작시 학습 스크립트를 ENTRYPOINT로 지정
 - 컨테이너 초기화 후 ENTRYPOINT가 실행됨



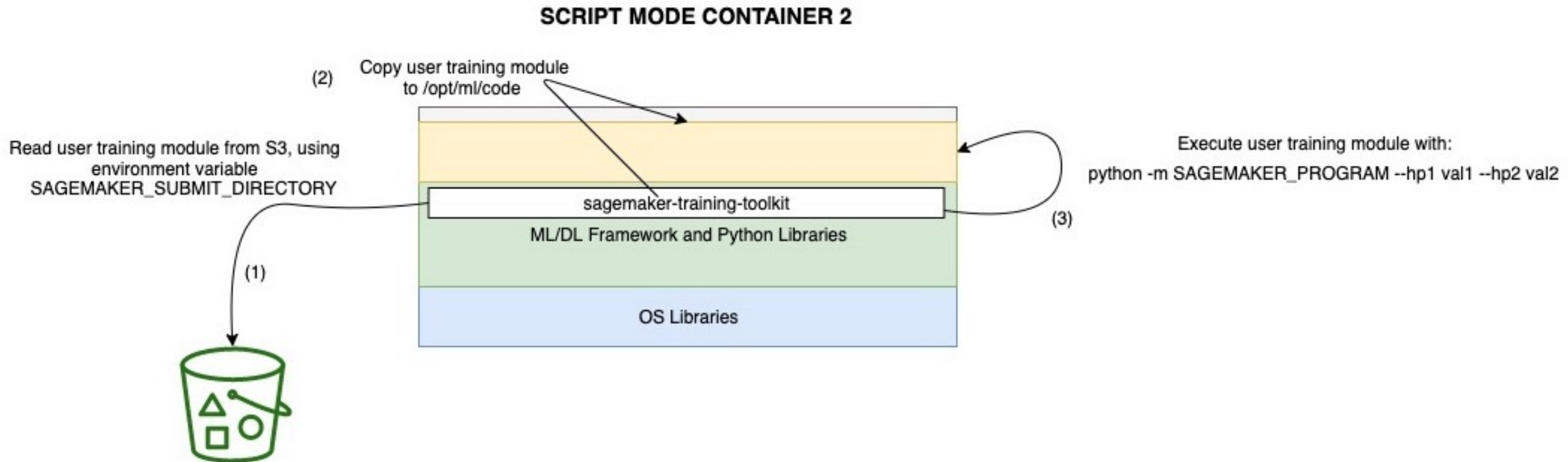
Script Mode Container

- SageMaker Training Toolkit 을 포함한 도커 이미지
- 컨테이너 내 특정 폴더에서 학습 스크립트를 선택하여 실행
 - SageMaker Training Toolkit이 ENTRYPOINT 역할



Script Mode Container 2

- 이전 슬라이드 (Script Mode Container)와 유사
- 학습 스크립트를 S3에 업로드
- 이를 /opt/ml/code 폴더에 다운로드하고 실행



예시) Script Mode Container 2

• Dockerfile

공용 도커 이미지 용 도커 파일

- 오장민이 docker-hub에 푸시해둔 범용 학습 환경 제작용 이미지
 - `jangminnature/mlops:speech-v1.0-22.09-py3`
 - cuda, cudnn
 - pytorch 1.13, transformers-dev, scikit-learn, ray, librosa, gradio 등
 - 추가 설치
 - deepctr-torch
 - sagemaker-training

```
1
2 FROM jangminnature/mlops:speech-v1.0-22.09-py3
3
4 WORKDIR /workspace
5
6 RUN pip install -U deepctr-torch sagemaker-training
7
8 # Setting some environment variables.
9 ENV PYTHONDONTWRITEBYTECODE=1 \
10     PYTHONUNBUFFERED=1 \
11     PYTHONIOENCODING=UTF-8 \
12     LANG=C.UTF-8 \
13     LC_ALL=C.UTF-8
14
```

예시) Script Mode Container 2

- AWS ECR에 도커 이미지 푸시

ECR 내 작성된 도커 이미지 URI

``126537332169.dkr.ecr.ap-northeast-2.amazonaws.com/sagemaker-training-containers/deepctr-training-container:latest``

예시) Script Mode Container 2

- SageMaker를 통한 training-job 과정 요약

SageMaker의 training-job 을 생성 및 진행 되는 원리 (외부에서 학습 스크립트를 제공하는 경우)

- 지정된 도커 이미지로부터 컨테이너 생성
 - 생성에 문제가 없으면 training-job 요청은 바로 리턴된다.
- 생성된 컨테이너의 기존 엔트리 포인트 무시 (무효화)
- 제공할 외부 학습 스크립트는 tar.gz로 압축하여 s3 버킷에 업로드 후 URI를 지정하여 제공
 - 새로운 엔트리 포인트를 지정
- 컨테이너 호스트로부터 특정 볼륨이 마운트 되는데 컨테이너 내부에서는 `/opt/ml/` 의 경로로 지정됨
 - 그 하부에 여러 폴더들이 만들어지며 학습/테스트 데이터의 경우 s3 버킷으로부터 자동으로 내려 받는다.
 - `input/data/train` : 학습 데이터
 - `input/data/test` : 테스트 데이터
 - `input/config` : 하이퍼파라미터 포함 설정
 - `model` : 모델 경로
 - 초기화가 끝난 후 학습 스크립트가 시작됨
- 학습 과정에서 생성되는 내용 (모델 포함)은 training-job 생성시 지정한 출력용 s3 버킷에 업로드 된다.
- 컨테이너가 종료되고, 최종적으로 training-job이 마무리 된다.

예시) Script Mode Container 2

- Training-job 에 지정할 내용

AlgorithmSpecification

- ECR의 도커 이미지 정보 지정

HyperParameters

- 외부 제공 학습 스크립트 엔트리 포인트
- 압축된 전체 학습 스크립트 묶음에 대한 s3 버킷 URI
- 학습 스크립트가 사용할 파라미터들

RoleArn

- 실행 권한: IAM의 역할 이름을 지정해야 함

InputDataConfig

- 입력 파일의 s3 버킷 URI 등 정보 제공
- 채널이라는 개념으로 구분됨: train, test 등

OutputDataConfig

- 출력 s3 버킷 URI
 - 학습의 결과물이 이 주소로 업로드 됨 (압축되어서)

ResourceConfig

- 2) • 컨테이너의 호스트 인스턴스 사양 등 설정

예시) Script Mode Container 2

- Training-job 생성: AWS CLI 의 경우

```
#!/bin/bash
```

```
export JOB_NAME=deep-ctr-trainjob-$(date +%Y-%m-%d-%H-%M)
```

```
aws sagemaker create-training-job --training-job-name $JOB_NAME --cli-input-json file://train_spec.json
```

예시) Script Mode Container 2

- train_spec.json

```
1 {  
2   "TrainingJobName": "deep-ctr-trainjob",  
3   "RoleArn": "arn:aws:iam::126537332169:role/service-role/SageMaker-oz-datascientist",  
4   "AlgorithmSpecification": {  
5     "TrainingInputMode": "File",  
6     "TrainingImage": "126537332169.dkr.ecr.ap-northeast-2.amazonaws.com/sagemaker-t  
7   },
```

예시) Script Mode Container 2

- train_spec.json

```
8  "InputDataConfig": [  
9    {  
10      "ChannelName": "train",  
11      "DataSource": {  
12        "S3DataSource": {  
13          "S3DataType": "S3Prefix",  
14          "S3Uri": "s3://sagemaker-ap-northeast-2-126537332169/katchers-recom",  
15          "S3DataDistributionType": "FullyReplicated"  
16        }  
17      }  
18    },  
19    {  
20      "ChannelName": "test",  
21      "DataSource": {  
22        "S3DataSource": {  
23          "S3DataType": "S3Prefix",  
24          "S3Uri": "s3://sagemaker-ap-northeast-2-126537332169/katchers-recommen",  
25          "S3DataDistributionType": "FullyReplicated"  
26        }  
27      }  
28    }  
29  ],
```

예시) Script Mode Container 2

- train_spec.json

```
30  "OutputDataConfig": {
31      "S3OutputPath": "s3://sagemaker-ap-northeast-2-126537332169/katchers-recommenda
32  },
33  "ResourceConfig": {
34      "VolumeSizeInGB": 64,
35      "InstanceCount": 1,
36      "InstanceType": "ml.m5.large"
37  },
38  "StoppingCondition": {
39      "MaxRuntimeInSeconds": 86400
40  },
41  "HyperParameters": {
42      "sagemaker_program": "main.py",
43      "sagemaker_submit_directory": "s3://sagemaker-ap-northeast-2-126537332169/katcher
44      "batch_size": "64",
45      "epochs": "4",
46      "verbose": "2",
47      "validation-split": "0.2",
48      "l2_reg": "1e5"
49  }
50 }
51
```

예시) Script Mode Container 2

- 모니터링

Amazon SageMaker > 훈련 작업

훈련 작업 정보

🔍 훈련 작업 검색

작업 ▼ 훈련 작업 생성

< 1 ... > ⚙️

	이름 ▼	생성 시간 ▼	기간	작업 상태 ▼	원 풀 상태	남은 시간
<input type="radio"/>	deep-ctr-trainjob-20230510-1403	2023. 5. 10. 오후 11:03:19	10 minutes	✅ Completed	-	-
<input type="radio"/>	deep-ctr-trainjob-2023-05-10-13-47-23	2023. 5. 10. 오후 10:47:23	10 minutes	✅ Completed	-	-
<input type="radio"/>	deep-ctr-trainjob-2023-05-10-13-23-25	2023. 5. 10. 오후 10:23:26	9 minutes	✅ Completed	-	-
<input type="radio"/>	deep-ctr-trainjob-2023-05-10-13-08-21	2023. 5. 10. 오후 10:08:21	10 minutes	❌ Failed	-	-
<input type="radio"/>	deep-ctr-trainjob-2023-05-10-12-57-44	2023. 5. 10. 오후 9:57:44	9 minutes	❌ Failed	-	-

각 training-job 별 상세 화면에서

- 상세 로그 조회 가능
- 실행 정보 등 확인 가능