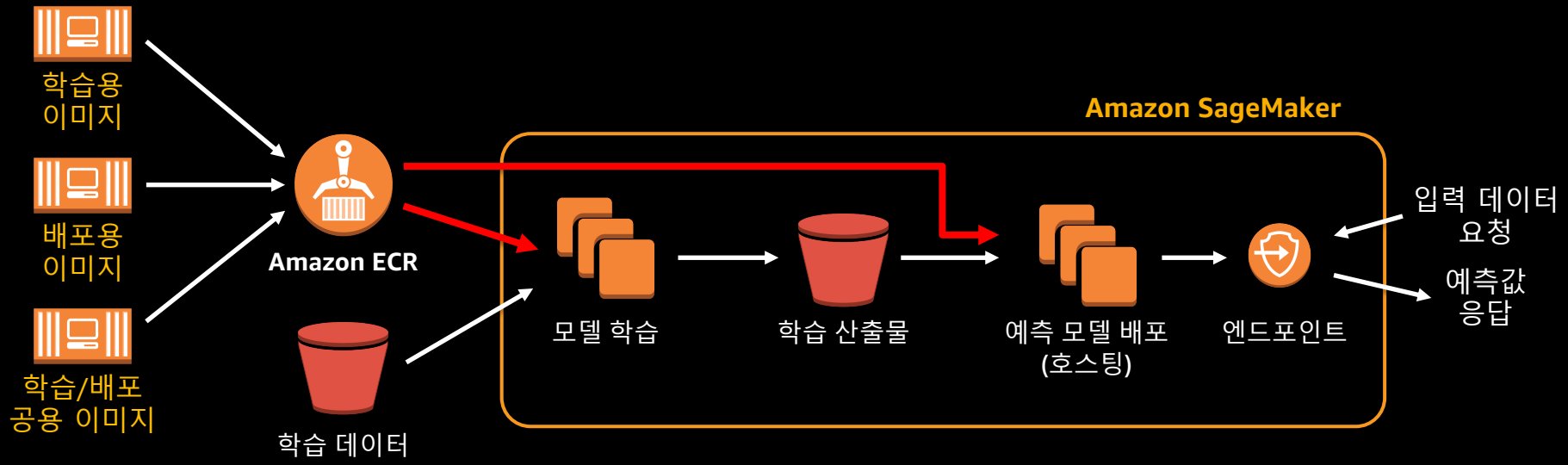
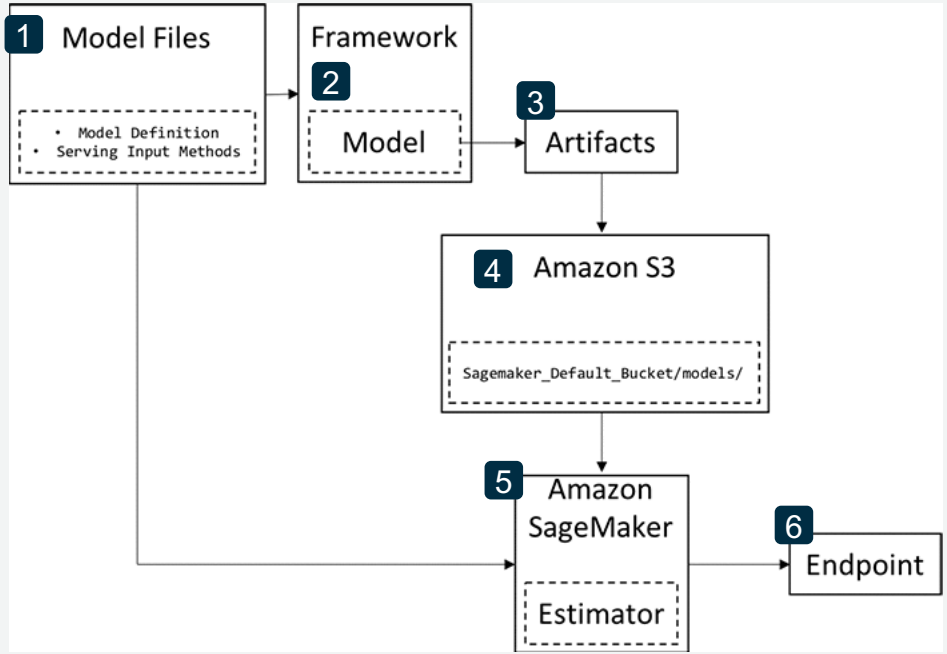
SageMaker - Bring Your Own Container

학습 모델과 추론 모델을 Docker 컨테이너로
어떤 딥러닝 프레임워크, 어떤 프로그래밍 언어라도 사용 가능



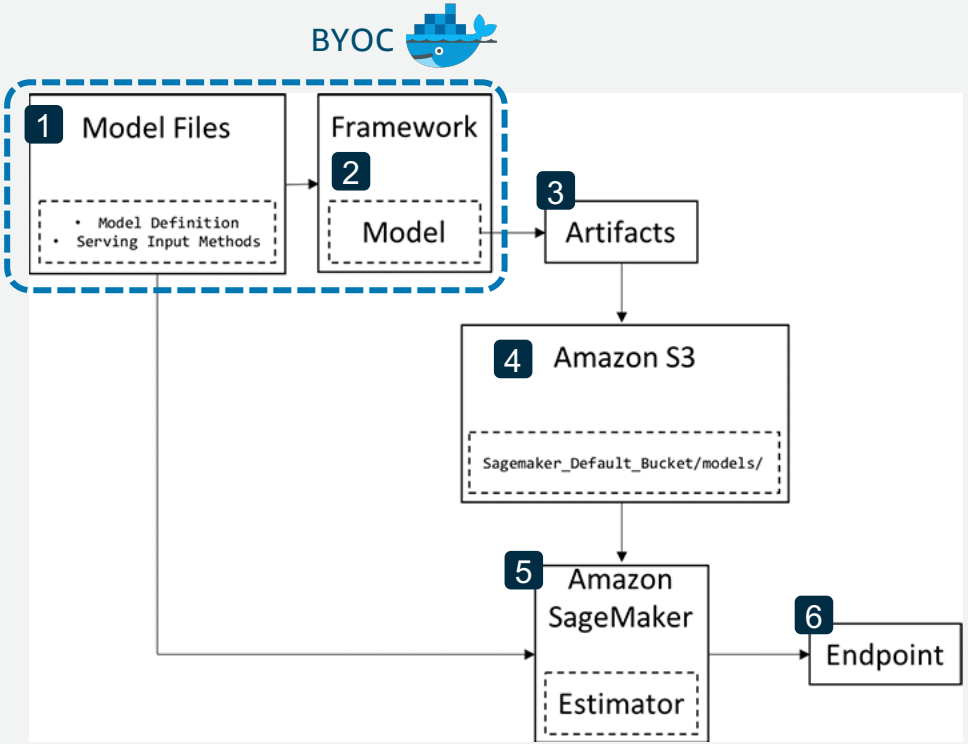
Sagemaker – Training and deploying process

1. Model definitions are written in a framework of choice.
2. The model is trained in that framework.
3. The model is exported and model artifacts that can be understood by Amazon SageMaker are created.
4. Model artifacts are uploaded to an Amazon S3 bucket.
5. Using the model definitions, artifacts, and the Amazon SageMaker Python SDK, a SageMaker model is created.
6. The SageMaker model is deployed as an endpoint.

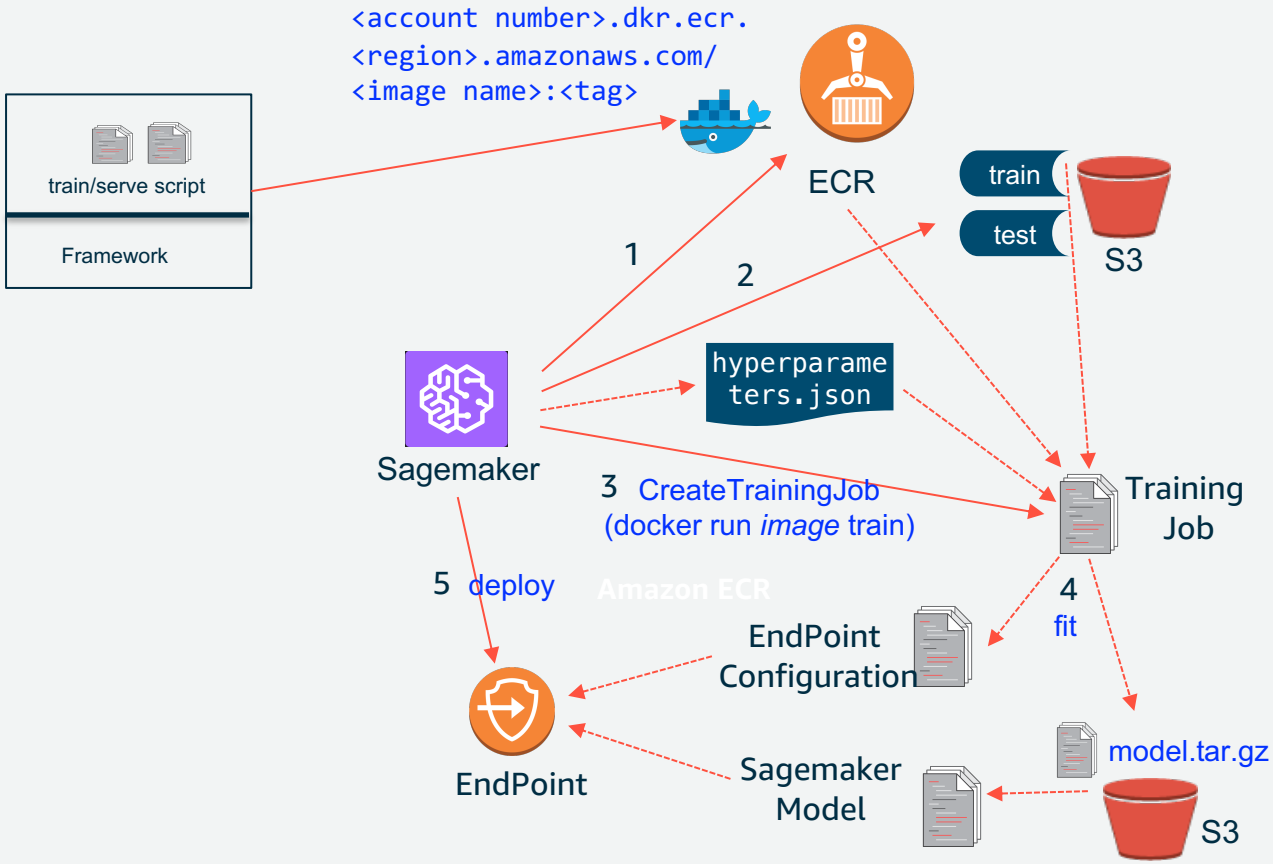


Sagemaker – When do you use BYOC

1. A specific version isn't supported.
2. Configure and install your dependencies and environment.
3. Use a different training/hosting solution than provided.
4. If the code that implements your algorithm is quite complex on its own
5. you need special additions to the framework



Sagemaker – BYOC Training



1. **Docker 이미지 등록**
   - Dependency 설치
   - train/servie 코드 정의
2. **S3 버킷에 Input 데이터 업로드**
   - Train/Test 구분하여 저장
3. **Create Job**
   - Algorithm hyperparameters
   - Training container image
   - instance type, instant count
4. **Fit**
   - IO path 지정
   - model.tar.gz 파일 생성
   - Sagemaker model 생성)
5. **Deploy**
   - Endpoint configurations 및 Endpoint 생성

aws

Sagemaker – BYOC – Preparing Docker Image

Dev environment (i.e. Notebook)

```
./container
├── Dockerfile
├── build_and_push.sh
└── code
    ├── nginx.conf
    ├── predictor.py
    ├── serve
    ├── train
    └── wsgi.py
```

Dockerfile

```
FROM ubuntu:16.04
MAINTAINER Amazon AI <sage-learner@amazon.com>
RUN apt-get -y update && apt-get install -y --no-install-recommends \
        wget \
        python \
        nginx \
        ca-certificates \
    && rm -rf /var/lib/apt/lists/*

RUN wget https://bootstrap.pypa.io/get-pip.py && python get-pip.py && \
    pip install numpy==1.15.1 scipy scikit-learn pandas flask gevent
gunicorn tensorflow Keras-Applications Keras-Preprocessing &&  \
        rm -rf /root/.cache

PYTHONDONTWRITEBYTECODE

ENV PYTHONUNBUFFERED=TRUE
ENV PYTHONDONTWRITEBYTECODE=TRUE
ENV PATH="/opt/program:${PATH}"

COPY code /opt/program
WORKDIR /opt/program
```

<account number>.dkr.ecr.<region>.amazonaws.com/<image name>:<tag>

ECR

Sagemaker

S3

aws

Sagemaker – BYOC – Custom program(1/2)

Dev environment (i.e. Notebook)

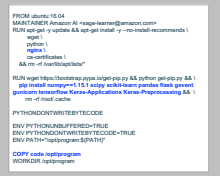
```
./container
├── Dockerfile
├── build_and_push.sh
└── code
    ├── nginx.conf
    ├── predictor.py
    ├── serve
    ├── train
    └── wsgi.py
```

Container

```
/opt/ml
├── input
│   ├── config
│   │   ├── hyperparameters.json
│   │   └── resourceConfig.json
│   └── data
│       └── <channel_name(train/test)>
│           └── <input data>
├── model
│   └── <model files>
└── output
    └── failure
```

```
/opt/program
├── nginx.conf
├── predictor.py
├── serve
├── train
└── wsgi.py
```

Dependencies thru Dockerfile

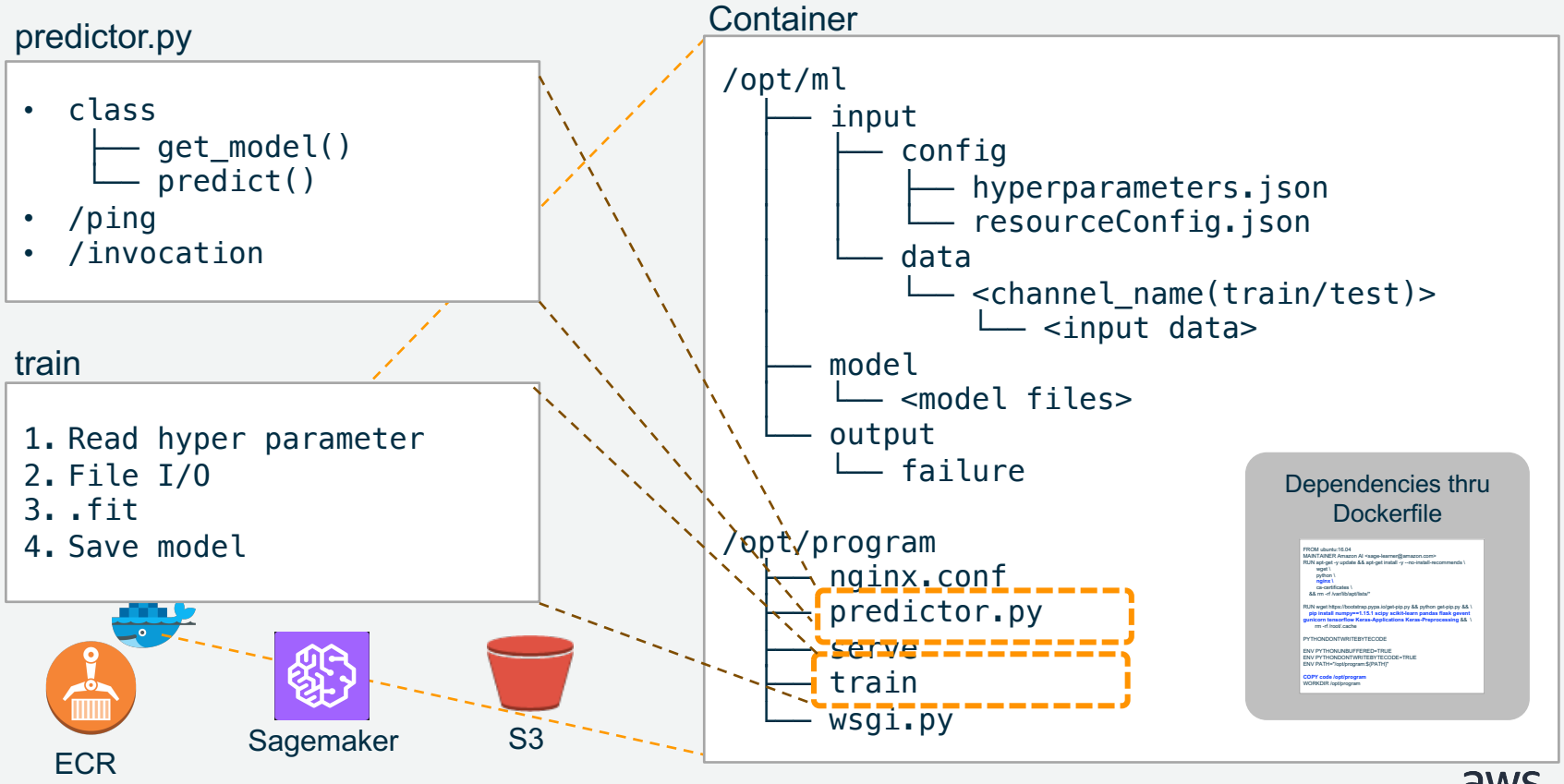


ECR

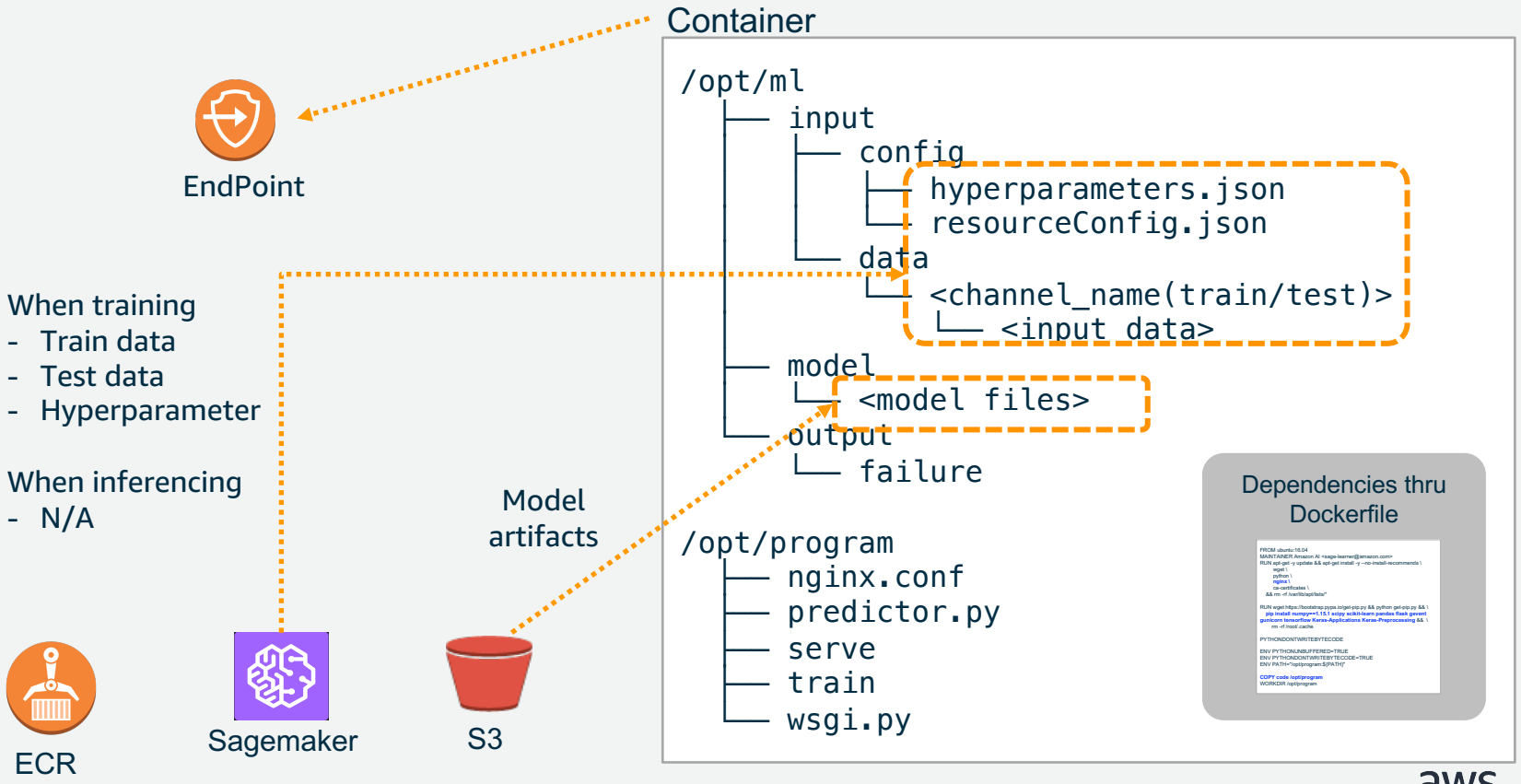
Sagemaker

S3

Sagemaker – BYOC – Custom program(2/2)



Sagemaker – BYOC – Deploying Endpoint



BYOC – Preparing Docker Image from Prebuilt Docker Image

Dev environment (i.e. Notebook)

```
./container
├── Dockerfile
├── build_and_push.sh
└── code
    ├── train/serve
```

<account number>.dkr.ecr.<region>.amazonaws.com/<image name>:<tag>

ECR

Sagemaker

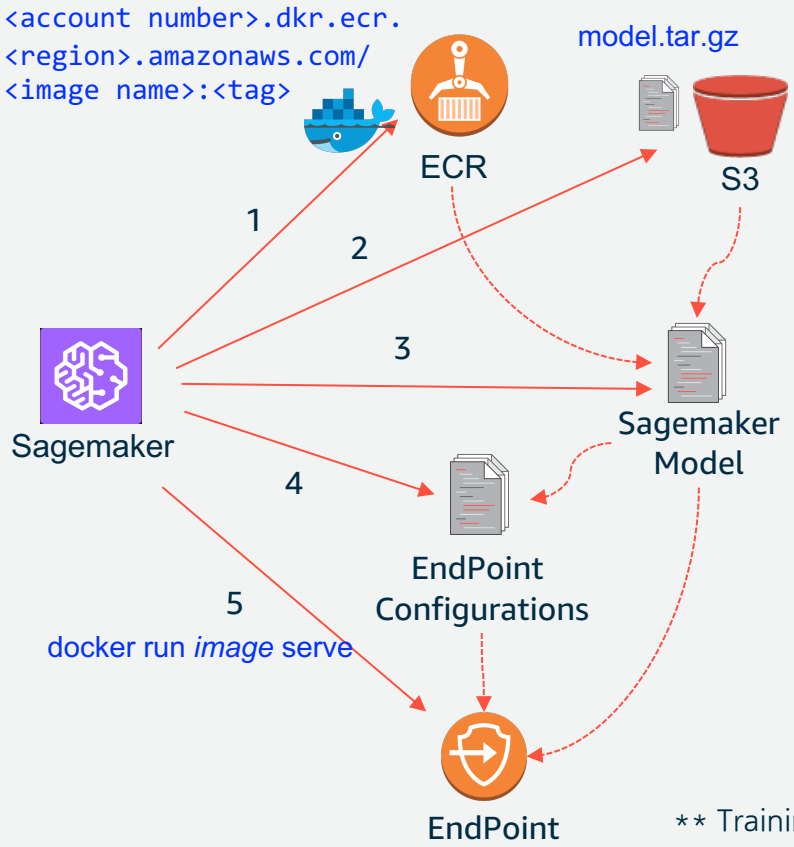
S3

Dockerfile

```
FROM 366743142698.dkr.ecr.ap-northeast-2.amazonaws.com/sagemaker-scikit-learn:0.20.0-cpu-py3

# install python package
RUN pip install joblib


ENV PYTHONUNBUFFERED=TRUE
ENV PYTHONDONTWRITEBYTECODE=TRUE

ENV PATH="/opt/ml/code:${PATH}"

# Copy training code
COPY code/* /opt/ml/code/

WORKDIR /opt/ml/code

# In order to use SageMaker Env varaibles, use the statement below
ENV SAGEMAKER_PROGRAM pca_byoc_train.py
```


aws

Sagemaker – BYOM Deployment



1. **Docker 이미지 등록**
   - Dependency 설치
   - predictor.py 코드 copy
2. **S3 버킷에 Custom 모델 Export****
   - Training이 완료된 모델
   - export방식은 알고리즘별로 개별 구현
3. **Sagemaker Model 생성****
   - 모델이 export된 S3 경로 지정
   - 사용할 Docker 이미지 지정
4. **Sagemaker Endpoint Configurations 생성**
   - 실행 instance type
   - initial instance count & weight
5. **Sagemaker Endpoint 생성**
   - Model과 endpoint configurations 활용

** Training 단계에서 Sagemaker 기능을 활용한 경우에는 자동 생성됨

aws

Lessons Learned

- 로컬에서 디버깅 필수!! (Dockerfile, predictor.py 외.)
  - Dockerfile을 이용하여 로컬에서 이미지 Build해보기 (Dependency debugging)
  - 로컬에서 Docker image Run 하고 serve 실행, postman 등으로 request 보내기
  - 나중엔 아예 로컬 Docker 이미지에 Jupyter notebook 설치하고 거기서 개발함 (이 방법 권장!!!)
- 도커도 가이드
  - https://subicura.com/2017/02/10/docker-guide-for-beginners-create-image-and-deploy.html