



MLOps 101

Episode 5: ML 생애주기 (4) 배포/서빙

한석진
마이크로소프트

Episode 5

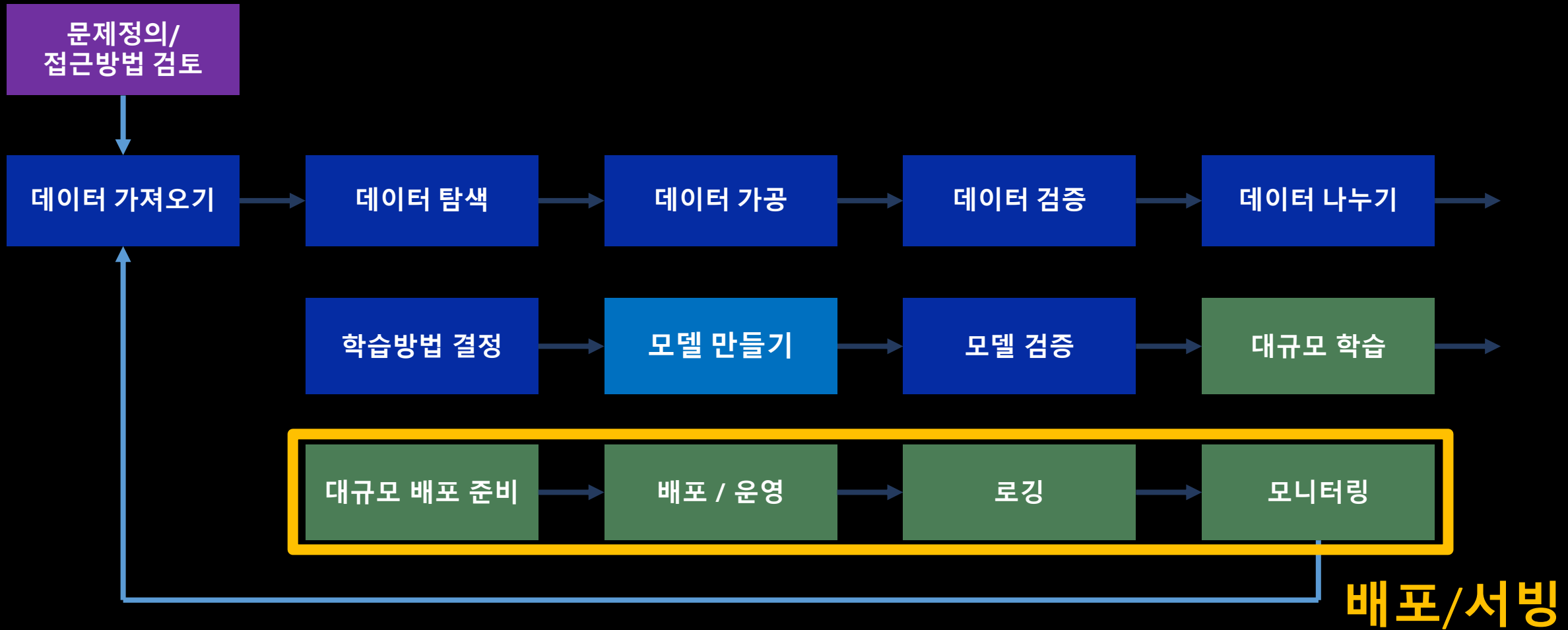
ML 생애주기 (4)

배포/서빙

ML 생애주기 (4) 배포/서빙

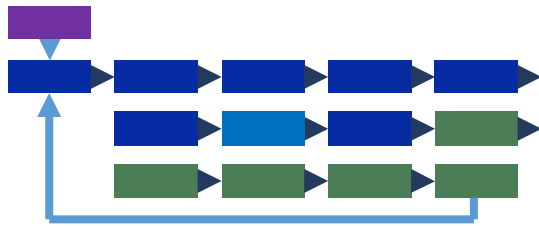
- (복습) MLOps는 누가 하나: Actors
- (복습) 일반적인 DevOps와의 비교
- 애저머신러닝에서 패키징, 배포(서빙) 개념
 - 자동화된 ML에서의 No-code 배포 *DEMO*
- 모델의 모니터링: 데이터 드리프트(Data Drift)
 - 애저머신러닝에서 데이터 드리프트 확인 *DEMO*

ML 생애주기



(복습) MLOps는 누가 하나: Actors

MLOps의 2개의 축



Data Scientists

- 빠르게 실험 반복
- 원하는 ML/DL 프레임워크 사용
- 가장 좋은 Tool
- 머리 아픈 관리는 최소화
- 대용량 (scale): 데이터 가공, 모델학습



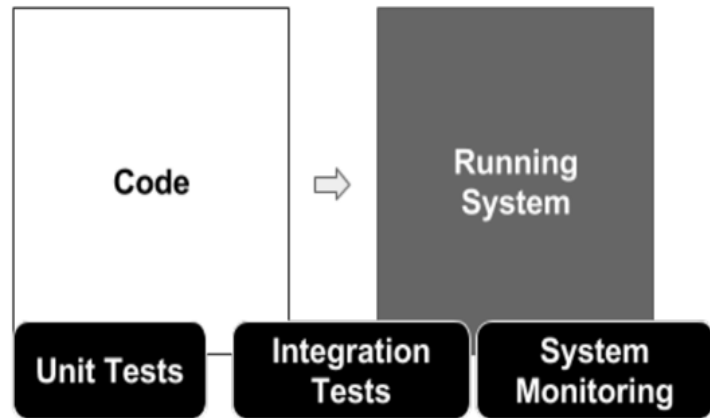
Data/Software Engineers

- Tool과 플랫폼의 재사용
- 전사 정책 (Compliance)
- 모니터링/감사(Audit)
- 죽지 않고 살아남기 (Uptime)

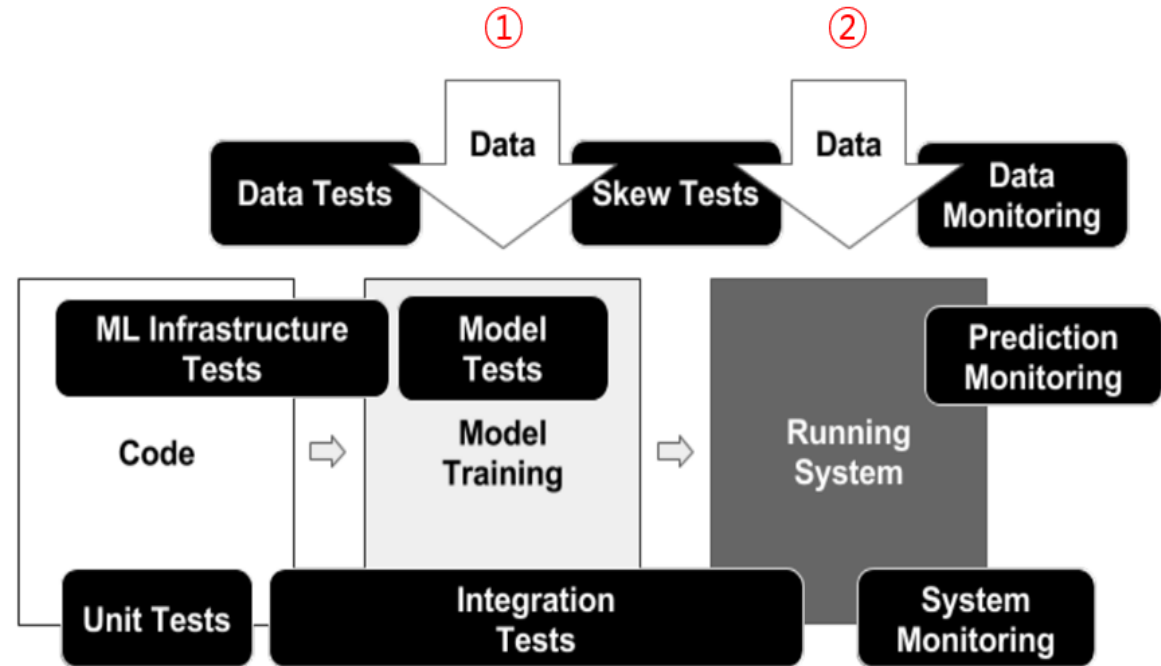


(복습) 일반적인 DevOps와의 비교

ML 기반 시스템의 테스트와 모니터링



Traditional System Testing and Monitoring



ML-Based System Testing and Monitoring

애저머신러닝에서 패키징, 배포 (서빙) 개념

모델 등록

```
model = run.register_model(model_name='sklearn_mnist',
                           tags={'area': 'mnist'},
                           model_path='outputs/sklearn_mnist_model.pkl')
print(model.name, model.id, model.version, sep='\t')
```

Python 패키지 정보 제공

```
name: project_environment
dependencies:
- python=3.6.2
- scikit-learn=0.22.1
- pip:
  # You must list azureml-defaults as a pip dependency
  - azureml-defaults>1.0.45
  - inference-schema[numpy-support]
```

추론로직 제공

```

Example: scikit-learn and Swagger
import json
import numpy as np
import os
from sklearn.externals import joblib
from sklearn.linear_model import Ridge

from inference_schema.schema_decorators import input_schema, output_schema
from inference_schema.parameter_types.numpy_parameter_type import NumpyParameterType

def init():
    global model
    model_path = os.path.join(os.getenv('AZUREML_MODEL_DIR'), 'sklearn_mnist_model.pkl')
    model = joblib.load(model_path)

input_sample = np.array([[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]])
output_sample = np.array([3726.995])

@input_schema('data', NumpyParameterType(input_sample))
@output_schema(NumpyParameterType(output_sample))
def run(data):
    try:
        result = model.predict(data)
        # You can return any data type, as long as it is JSON serializable.
        return result.tolist()
    except Exception as e:
        error = str(e)
        return error

```

Container

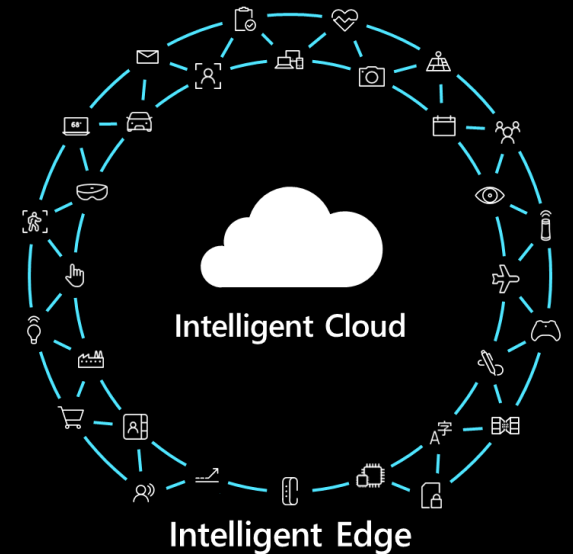
- Image
- 자동생성



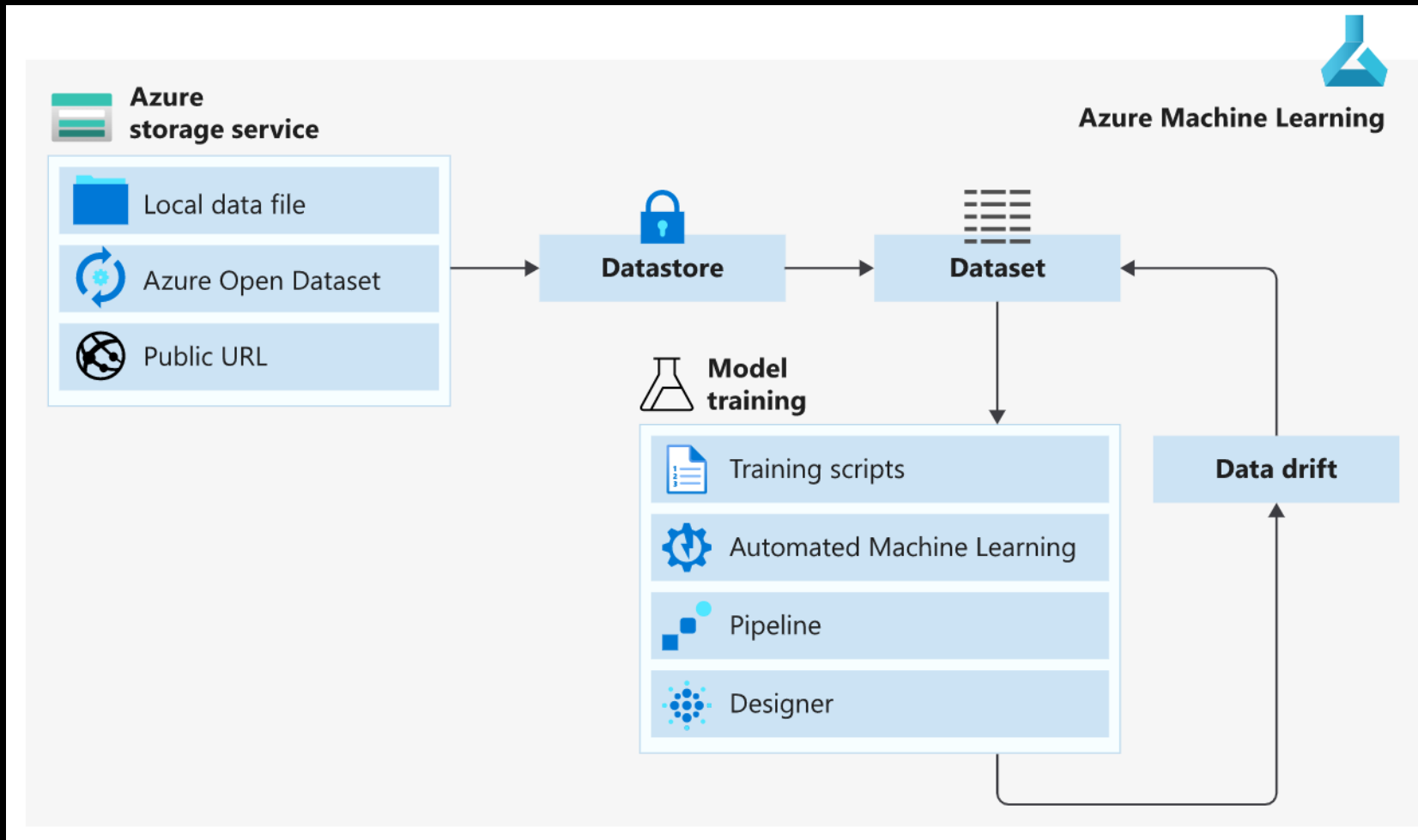
배포환경 지정

배포/서빙

위치	용도	GPU	FPGA	설명
Local Web	Dev/Test			H/W가속화 지원
Compute Instance	Dev/Test			
AKS	실시간 추론	Y	Y	대규모 Production용. Auto-scale 지원.
ACI	Dev/Test			소규모, CPU기반, 48GB 메모리 이내
AML Compute	배치 추론	Y (pipe-line)		Normal 및 Low Priority 지원
App Service	실시간 추론			
Functions	실시간 추론			
Cognitive Search	배치 추론			
IoT Edge	Edge 추론			
DataBoxEdge	Edge 추론		Y	IoT Edge Appliance



모델의 모니터링: 데이터 드리프트(Data Drift)



Data Drift

- 데이터의 패턴 특성이 시간이 지나며 변하는 현상
- 학습에 활용된 당시의 데이터와 차이가 커질 수록 예측 성능이 저하될 가능성
- 따라서 Data Drift 여부를 지속적으로 모니터링하면 모델의 재학습 시점을 판단하는데 간접적인 방법이 될 수 있음

Episode 5

ML 생애주기 (4)

배포/서빙

ML 생애주기 (4) 배포/서빙

- (복습) MLOps는 누가 하나: Actors
- (복습) 일반적인 DevOps와의 비교
- 애저머신러닝에서 패키징, 배포(서빙) 개념
 - 자동화된 ML에서의 No-code 배포 *DEMO*
- 모델의 모니터링: 데이터 드리프트(Data Drift)
 - 애저머신러닝에서 데이터 드리프트 확인 *DEMO*

{다음 시간에는}

Episode 6

MLOps in Action

옛보기

-
- 이제껏 알아본 것들이 실제 어떻게 구현되나
 - 애저머신러닝과 애저데브옵스의 만남 *DEMO*
 - MLOps란 무엇이고 어떻게 준비할 수 있는가