



# AWS INNOVATE

ONLINE CONFERENCE 2018

*SPECIAL EDITION - MACHINE LEARNING*

**AWS의 새로운 통합 머신러닝 플랫폼 서비스**  
**Amazon SageMaker**

김무현 AWS 솔루션즈 아키텍트

# 목차



- 머신러닝 프로세스 리뷰
- Amazon SageMaker 소개
- Amazon SageMaker 주요 기능
- 데모

# ML on AWS: Our mission

모든 개발자와 데이터 과학자들이 사용할 수 있는  
서비스를 제공하는 것입니다.

# AWS 머신러닝 스택

## A P P L I C A T I O N S E R V I C E S

### VISION



Amazon  
Rekognition



Amazon  
Rekognition Video



Amazon Kinesis  
Video Streams

### LANGUAGE



Amazon Polly



Amazon Lex

Alexa for Business

Amazon Transcribe

Amazon Translate

Amazon Comprehend

### VR/IR

Amazon Sumerian

## PLATFORM SERVICES

Amazon SageMaker

AWS DeepLens

## FRAMEWORKS AND INTERFACES

### AWS DEEP LEARNING API

Apache MXNet

Caffe2

CNTK

PyTorch

TensorFlow

Theano

Torch

Keras

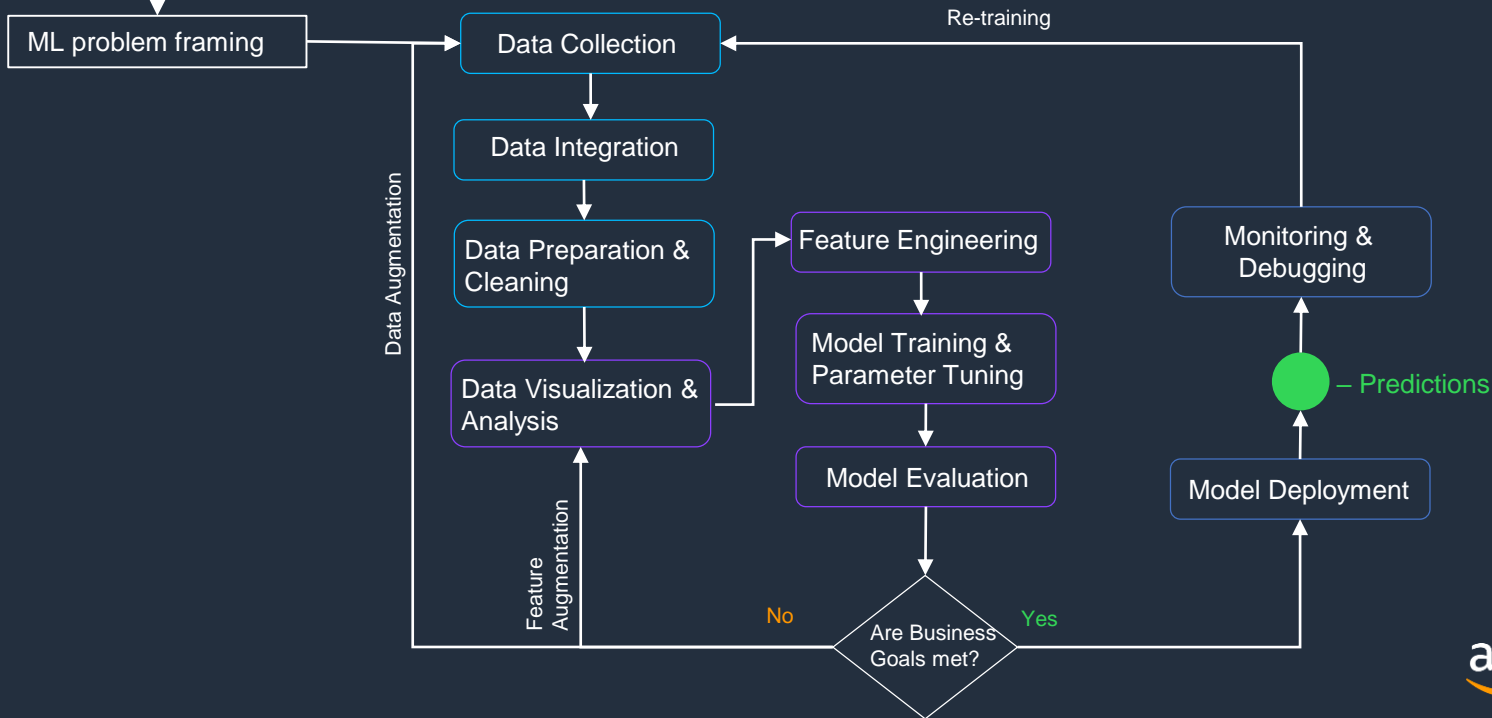
Gluon



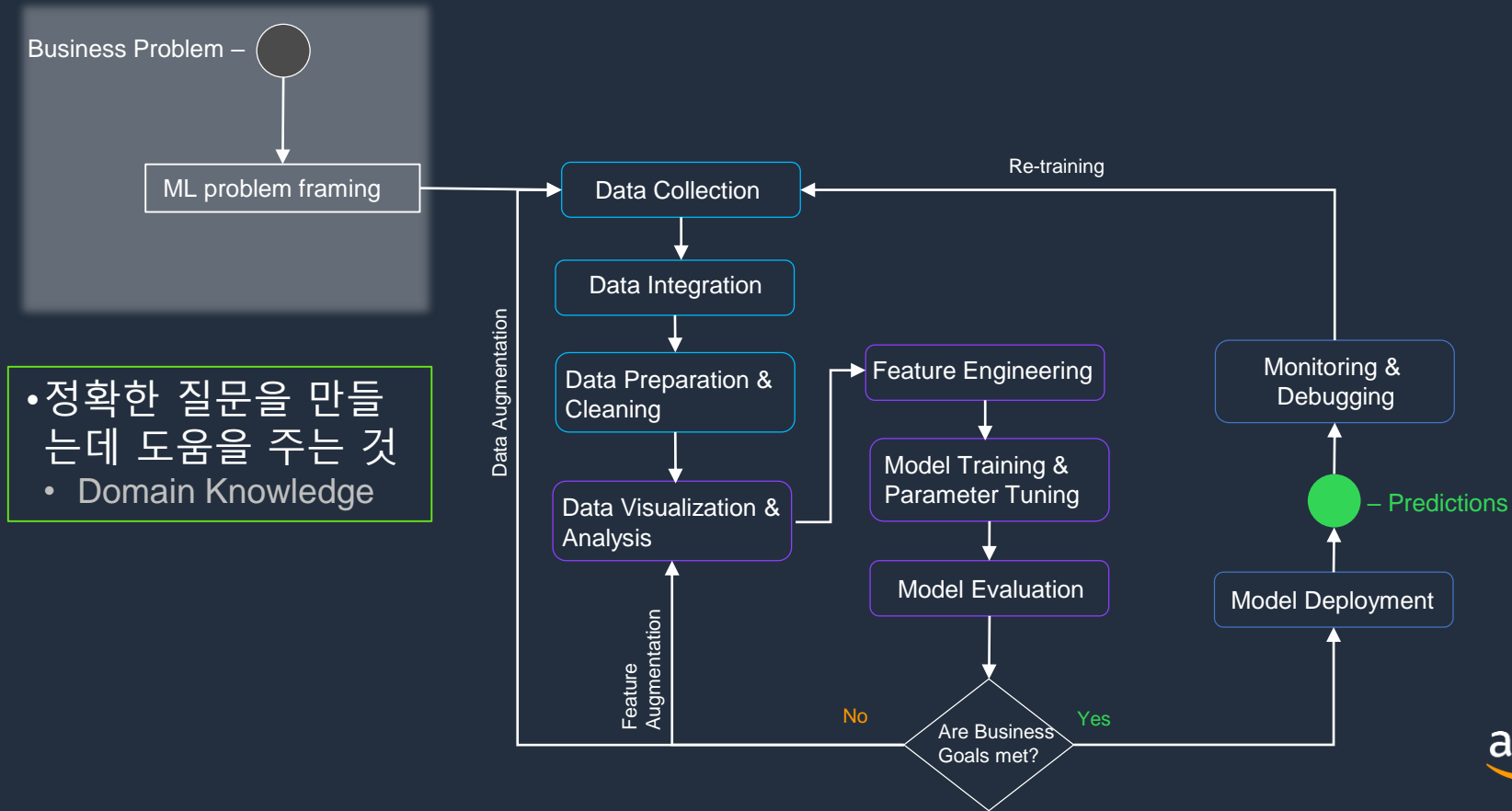
# 머신러닝 프로세스를 함께 볼까요

# 머신러닝 프로세스

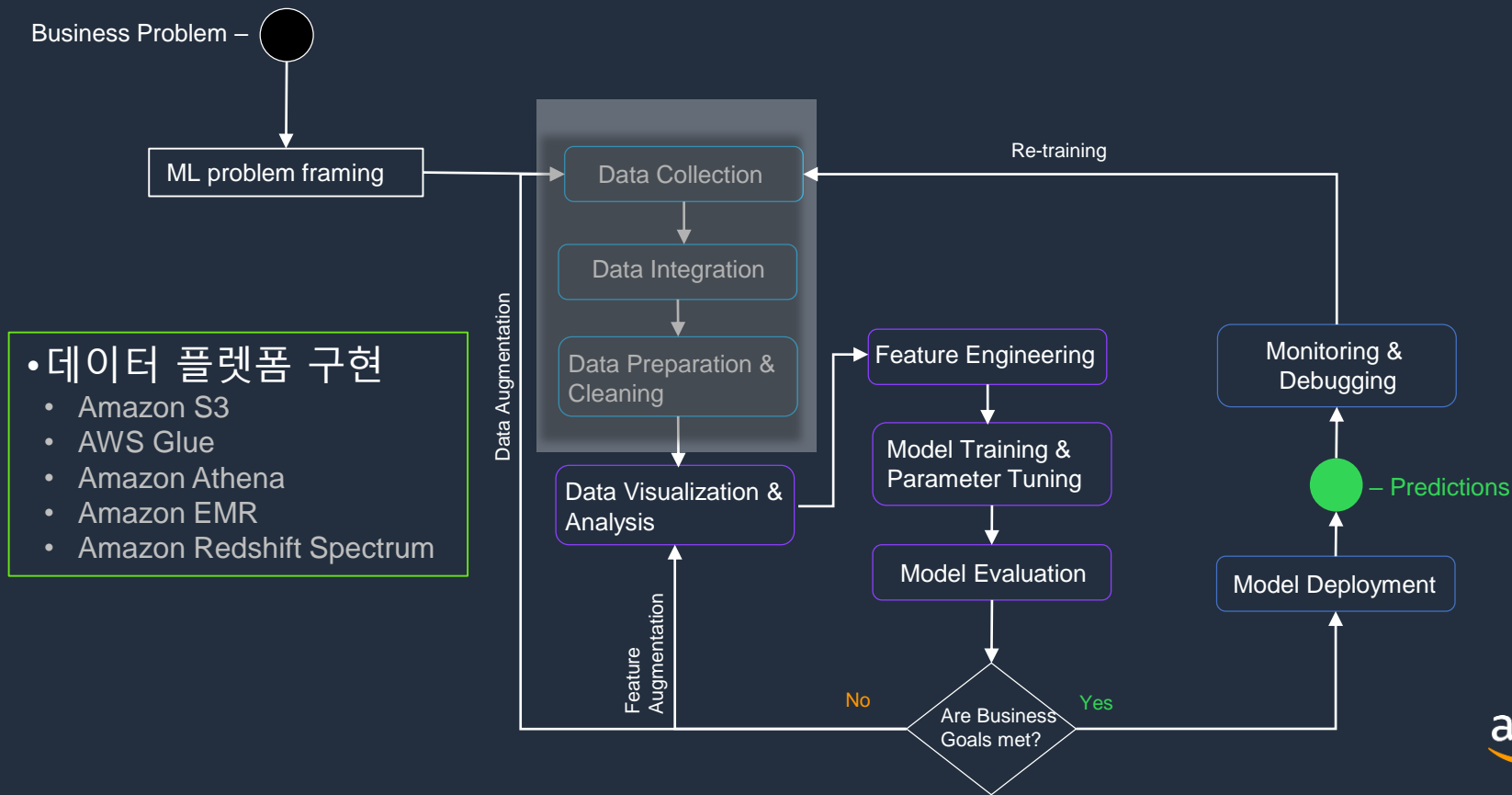
Business Problem – ●



## 탐색: 분석

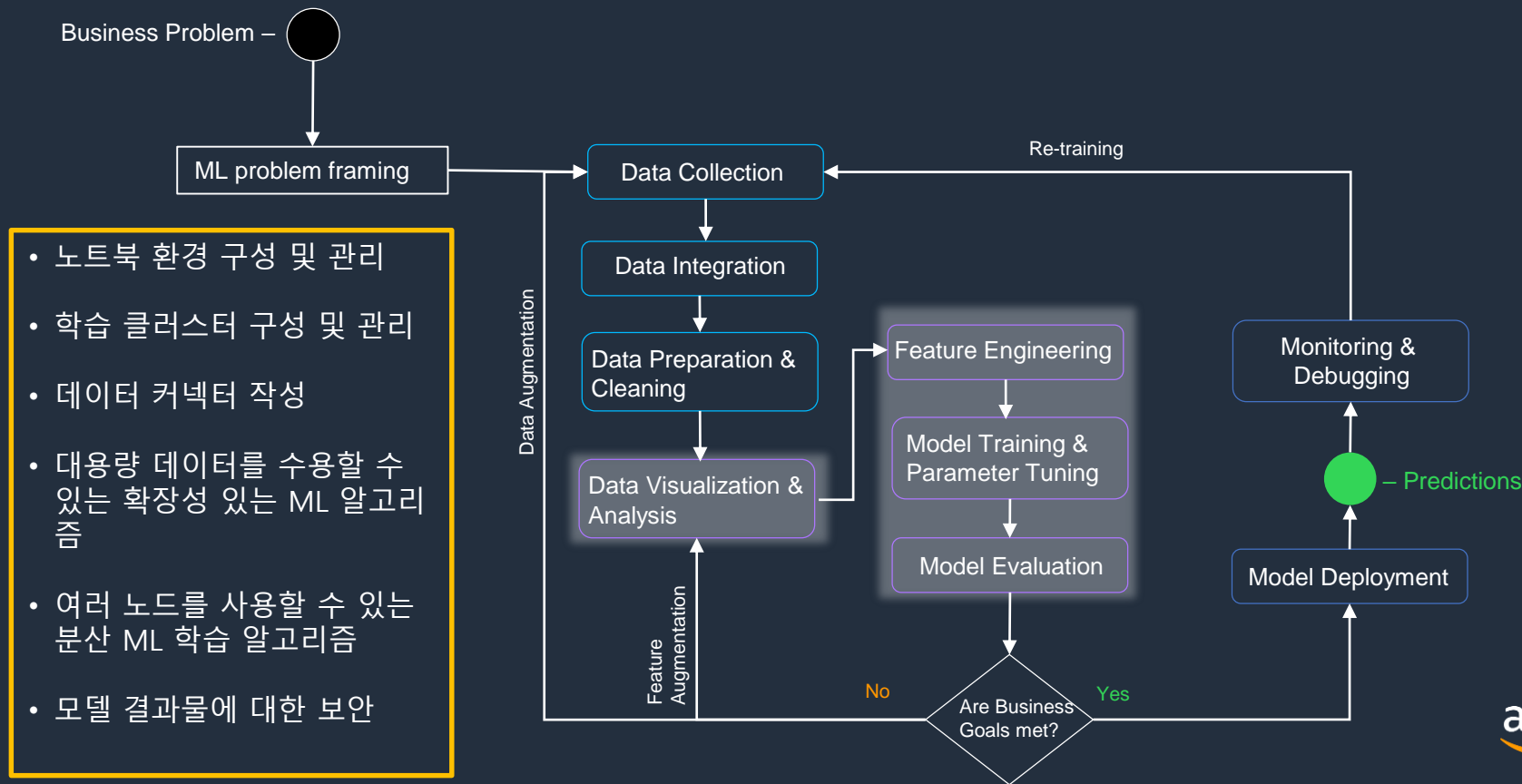


# 통합: 데이터 아키텍처

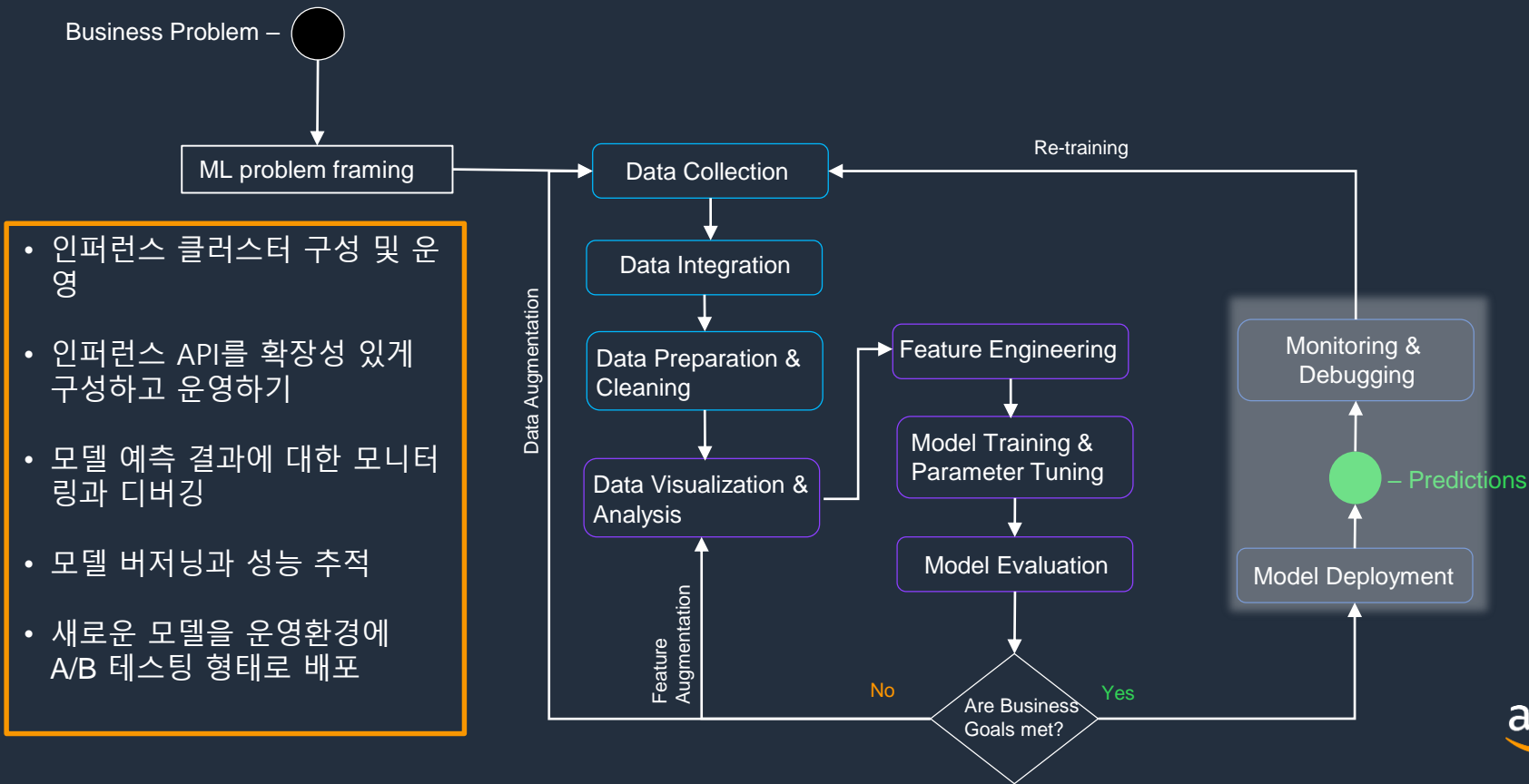




# 왜 SageMaker를 만들었나 – 학습 환경 제공



# 왜 SageMaker를 만들었나 – 배포 환경 제공





# Amazon SageMaker

데이터 과학자와 개발자들이 스마트 어플리케이션에 사용될  
머신러닝 기반의 모델을 빠르고 쉽게 만들도록 해주는  
완전 관리형 서비스

# Amazon SageMaker

구축

사전에 빌드된  
노트북 인스턴스

고도로 최적화된  
머신러닝 알고리즘들



# Amazon SageMaker

## 구축

사전에 빌드된  
노트북 인스턴스

고도로 최적화된  
머신러닝 알고리즘들

한번 클릭으로  
ML, DL,  
커스텀 알고리즘 학습

하이퍼파라미터 최적화를  
통한 손쉬운 학습



## 학습



# Amazon SageMaker

## 구축

사전에 빌드된  
노트북 인스턴스

고도로 최적화된  
머신러닝 알고리즘들



## 학습

한번 클릭으로  
ML, DL,  
커스텀 알고리즘 학습



하이퍼파라미터 최적화를  
통한 손쉬운 학습



확장성이 있는 관전  
관리형 모델 호스팅

엔지니어링 노력  
이 필요없는 배포

## 배포



# Amazon SageMaker

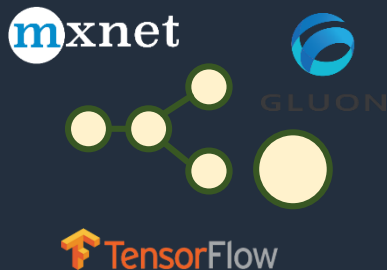
머신러닝 모델을 확장성이 있도록 빌드, 학습, 배포



End-to-End  
머신러닝 플랫폼



제로 셋업



유연한 모델 학습



초당 과금

Client application



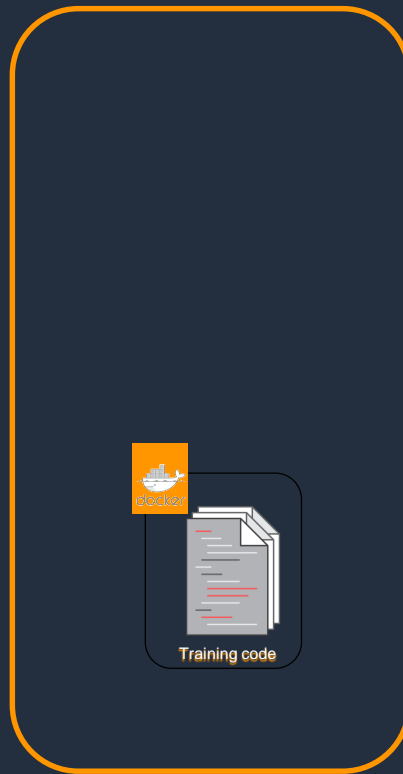
Amazon SageMaker



Amazon ECR



Model Training (on EC2)



Training code

학습 코드 및  
학습 데이터세트  
준비



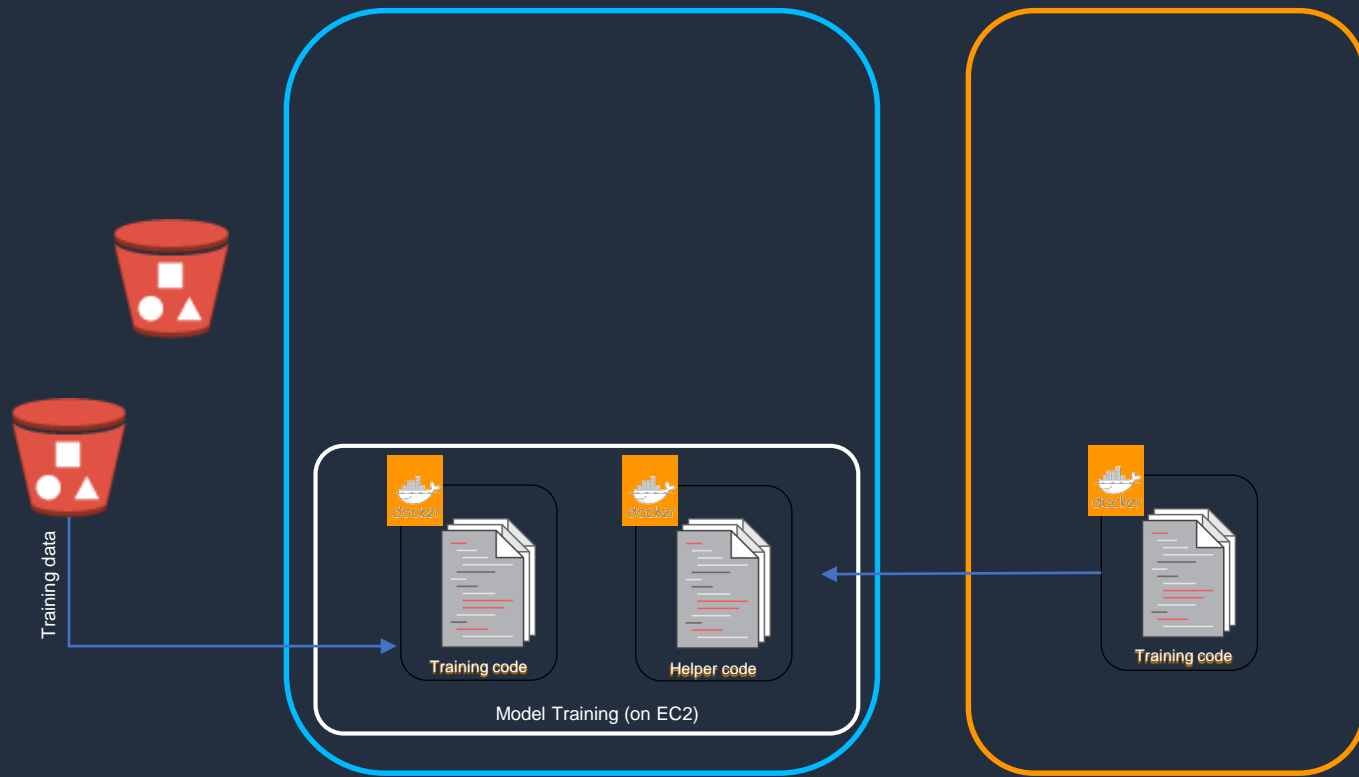
Client application



Amazon SageMaker



Amazon ECR



학습 환경 구성  
및  
학습 수행

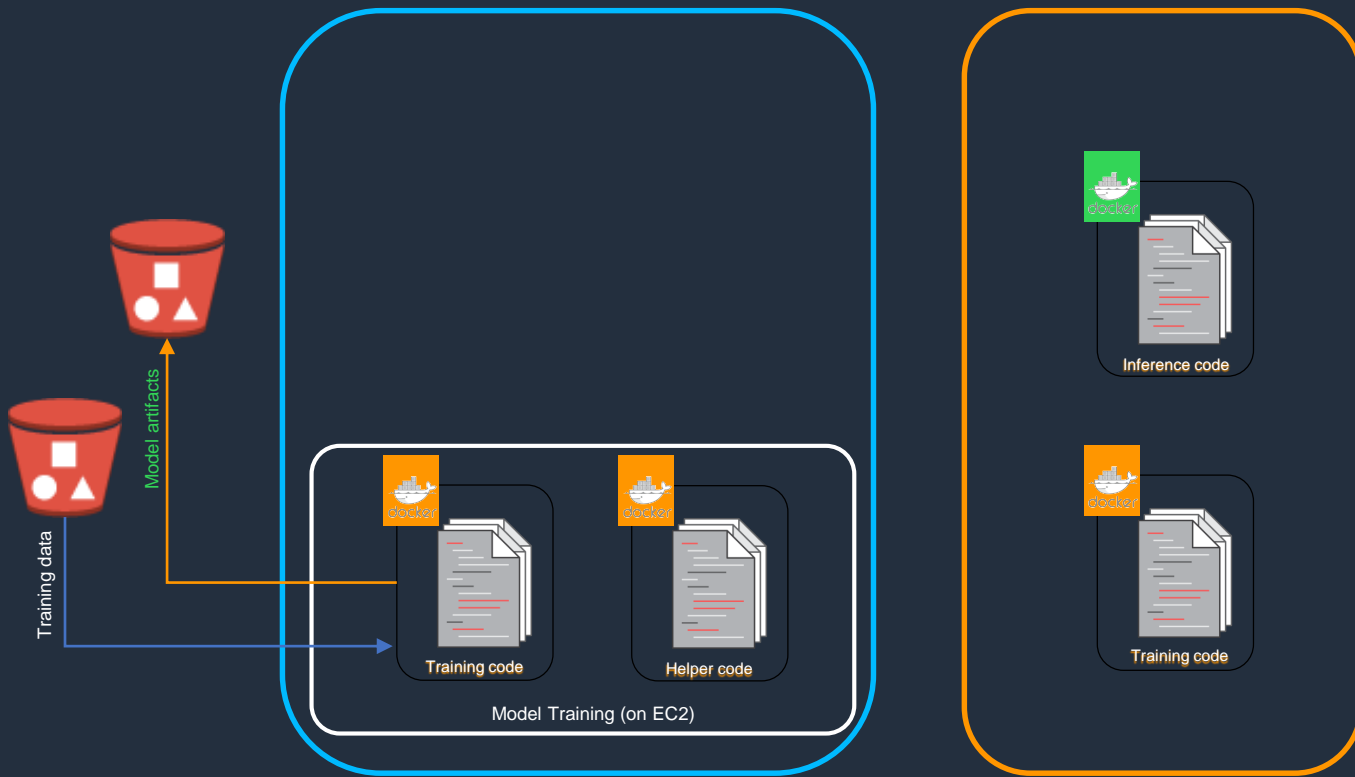
Client application



Amazon SageMaker



Amazon ECR



학습 완료된 모델  
저장  
및  
예측 코드 준비

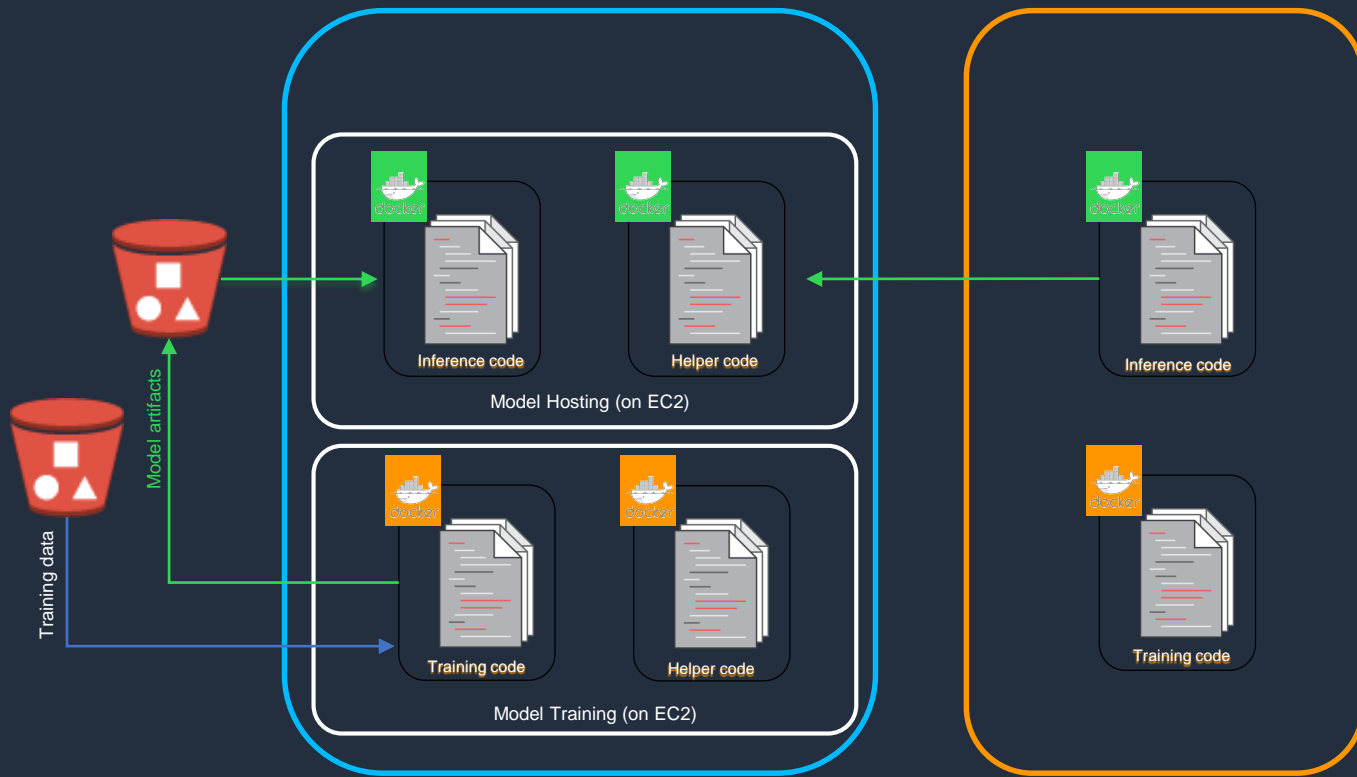
Client application



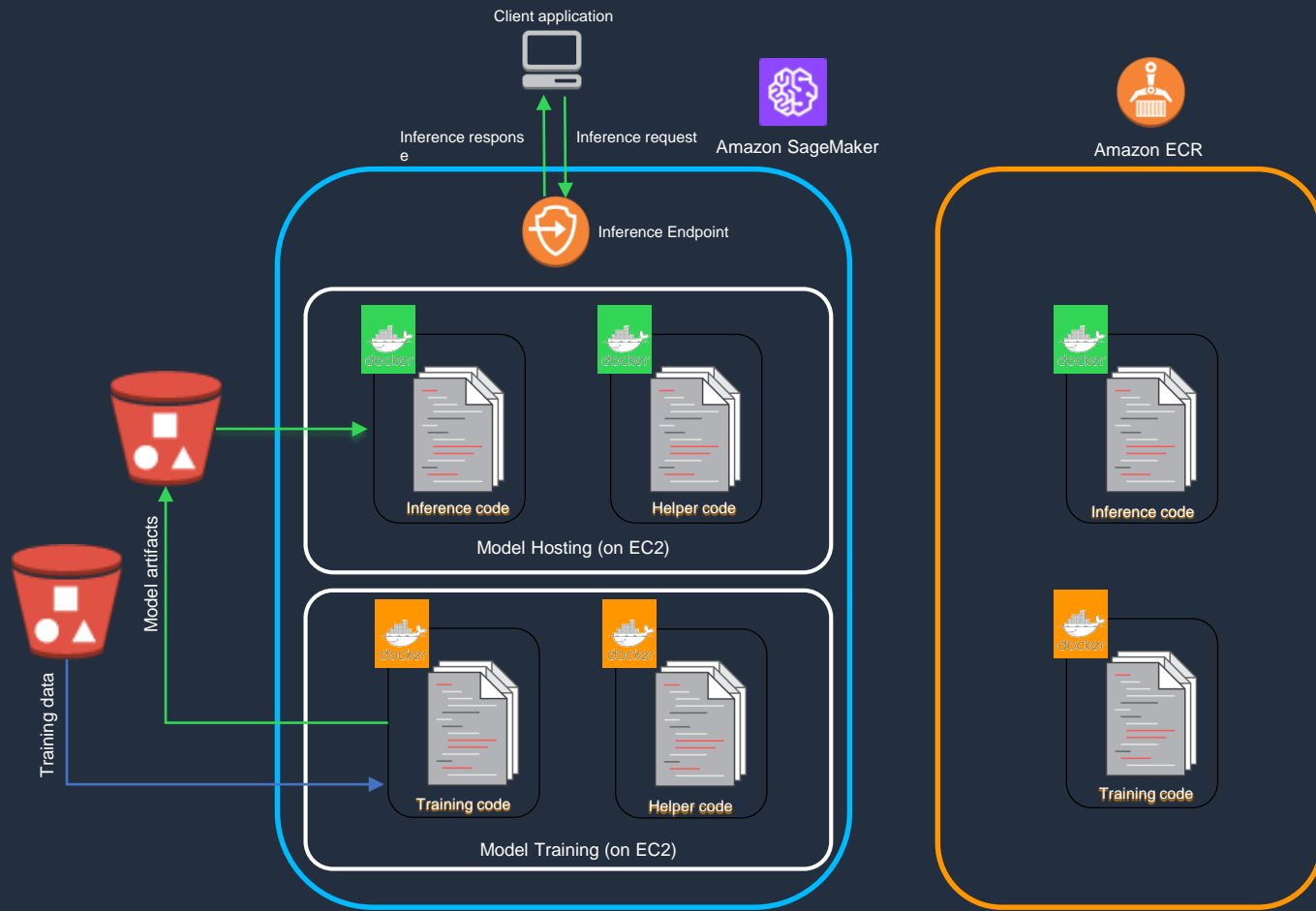
Amazon SageMaker



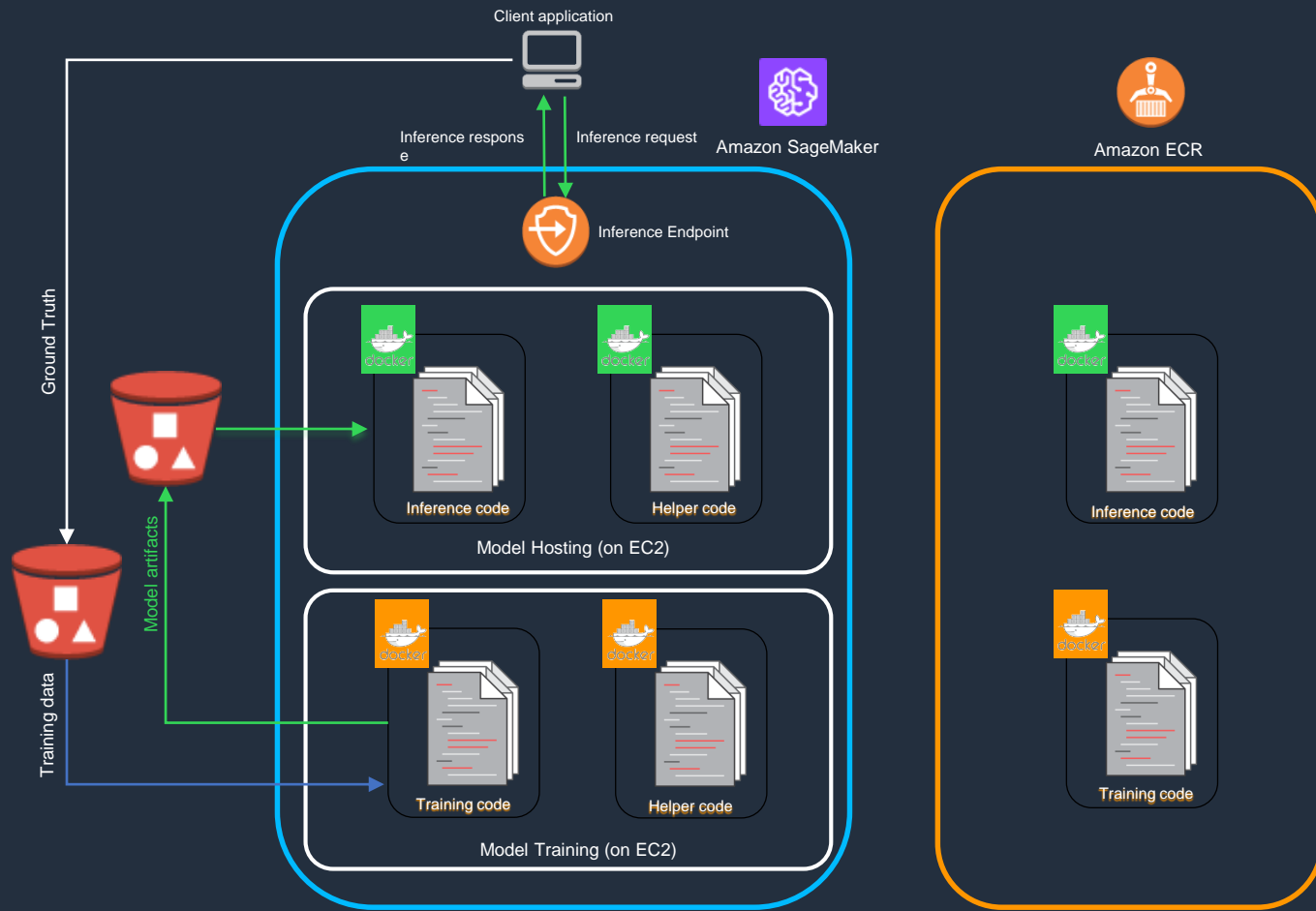
Amazon ECR



예측 환경 구성  
및  
모델 호스팅



예측 엔드포인트  
를 통한 API 서비스 제공



새로운  
학습 데이터 수집  
및 재학습, 배포

...

# Intuit에서 말하는 SageMaker의 장점들

## SageMaker 사용 전

노트북 환경을 Ad-hoc 하게 구성하고 관리  
해야했음

모델 배포 환경 선택의 제약

팀간의 컴퓨트 자원 경쟁

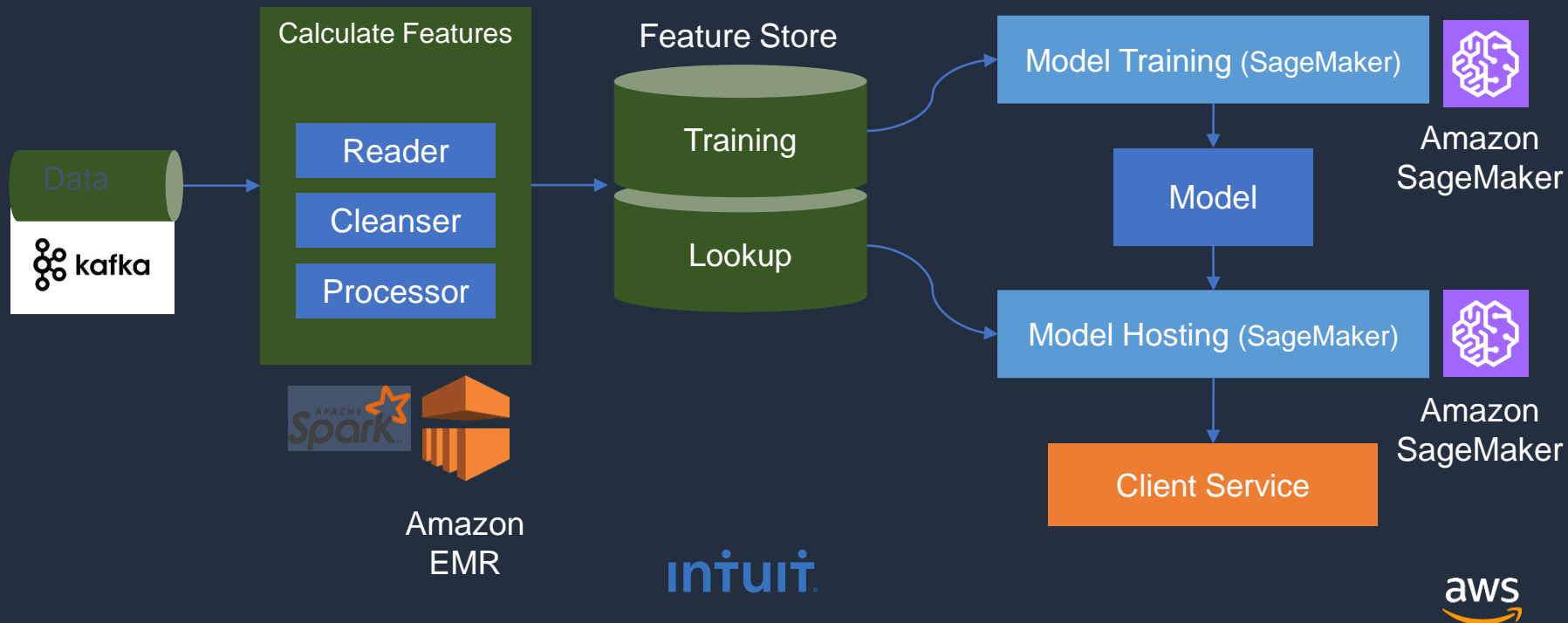
## SageMaker 사용 후

SageMaker 노트북에서 쉬운 데이터 탐색  
가능

가상화를 활용한 유연한 배포 환경 구성

Auto-scale이 지원되는 모델 호스팅 환경

# SageMaker를 이용한 실시간 사기 탐지 기능





# Amazon SageMaker

1



|  
Notebook 인스턴스

2



|  
ML 알고리즘들

3



|  
ML 학습 서비스

4



|  
ML 호스팅 서비스



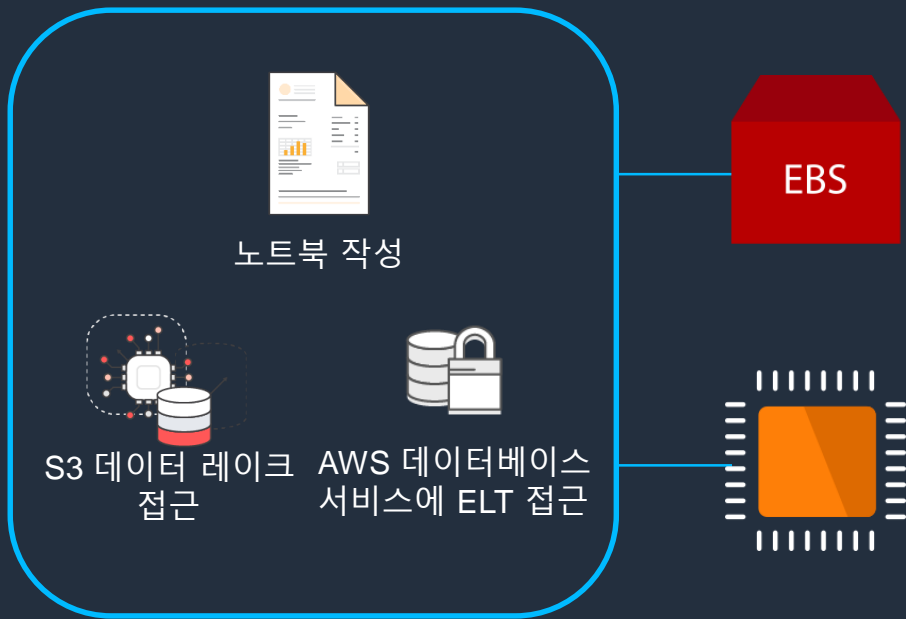
1

# Zero Setup

데이터 탐색을 통한 분석을 바로 수행



Notebook 인스턴스



*"Just add data"*

- 추천/개인화
- 이상 거래 탐지
- 이미지 분류
- 이탈 예측
- 마케팅 이메일, 캠페인 타케팅
- 로그 프로세스 및 이상 탐지
- 음성을 텍스트로 변환
- ...

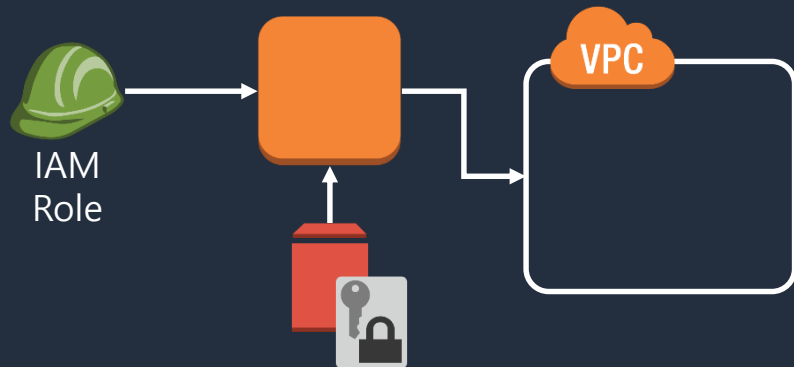
# 예제 - Jupyter 노트북 만들기

## SageMaker가 없다면 ...

1. AWS Deep Learning AMI 선택
2. EC2 인스턴스 생성
3. Jupyter 노트북 서비스 구동
4. SSH 터널링 설정
5. Jupyter 노트북 접속
6. 문제 생기면, 3번부터 반복

## SageMaker를 이용하면 ...

1. Jupyter 인스턴스 생성 요청
2. AWS 콘솔에서 Jupyter 노트북 열기



데모

Amazon SageMaker

Services

Resource Groups

Amazon SageMaker

Muhyun Kim

N. Virginia

Support

Dashboard

Notebook instances

Jobs

Resources

Models

Endpoint configuration

Endpoints

ARTIFICIAL INTELLIGENCE

# Amazon SageMaker

Build, train, and deploy machine learning models at scale

The quickest and easiest way to get ML models from idea to production.

### Get started

Explore AWS data in your notebooks, and use algorithms to create models via training jobs. Leverage Notebook instances in the cloud to begin.

Create notebook instance


[Start with an overview](#)

### Pricing (US)

With Amazon SageMaker, you pay only for what you use. Authoring, training and hosting is billed by the second, with no minimum fees and no upfront commitments.


[Learn more](#)

### How it works




**Build**

Connect to other AWS services and




**Train**

Use Amazon SageMaker's



**Tune**

Amazon SageMaker's automatic



**Deploy**

Once training is completed, models can

Feedback

English (US)

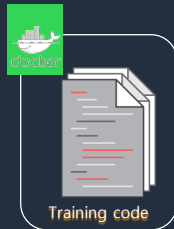
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



## ② 속도와 큰 데이터에 최적화된 ML Algorithm



|  
Algorithms



- Matrix Factorization
- Regression
- Principal Component Analysis
- K-Means Clustering
- Gradient Boosted Trees
- Time-series Prediction
- Image Classification
- 더 많은 알고리즘 추가 예정

Amazon provided Algorithms



Bring Your Own Script  
(SageMaker builds the Container)



SageMaker Estimators  
in Apache Spark



Bring Your Own Algorithm (You build the Container)



|

Algorithms

# Amazon SageMaker ML 알고리즘 특징



데이터셋 스트리밍을  
통한 저렴한 학습  
비용



싱글 패스로  
빠른 학습



아주 큰 데이터셋에  
대한 학습이 가능



다양한  
ML 알고리즘 제공

# 빌트인 ML Algorithm 종류

문제	알고리즘	러닝 형태
Discrete Classification, Regression	Linear Learner	Supervised
	XGBoost Algorithm	Supervised
Discrete Recommendations	Factorization Machines	Supervised
Image Classification	Image Classification Algorithm	Supervised, CNN
Neural Machine Translation	Sequence to Sequence	Supervised, seq2seq
Time-series Prediction	DeepAR Forecasting	Supervised, RNN
Discrete Groupings	K-Means Algorithm	Unsupervised
Dimensionality Reduction	PCA (Principal Component Analysis)	Unsupervised
Topic Determination	Latent Dirichlet Allocation (LDA)	Unsupervised
	Neural Topic Model (NTM)	Unsupervised, Neural Network Based
Word2Vec Embedding	BlazingText	Unsupervised

# 알고리즘을 조금 더 살펴보면 ...

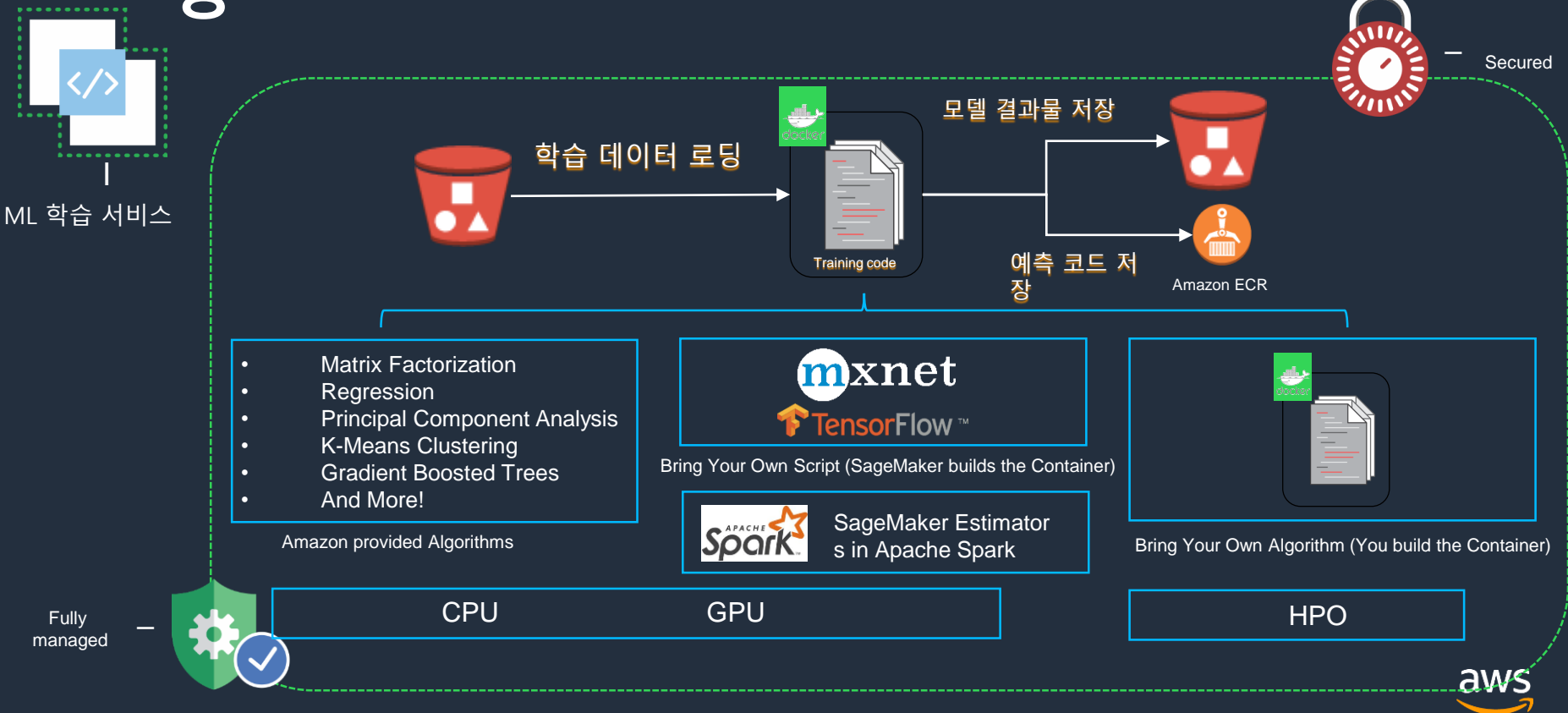
## Neural Machine Translation

- Recurrent Neural Networks (RNNs)와 Convolutional Neural Network (CNN) 모델을 attention과 함께 사용하는 encoder-decoder 아키텍처
- 활용예 - 기계 번역, 텍스트 요약, 음성을 텍스트로 변환

## Time-series Forecasting

- 관련된 여러 Time-series 학습 데이터들로 부터 패턴을 학습해서 정확한 예측 모델을 생성하는 알고리즘
- <https://arxiv.org/abs/1704.04110>
- 특정 시점에 대한 예측 및 확률적인 예측 결과 제공

# ③ 유연한 분산 학습 환경을 관리형 서비스로 제





# SageMaker를 이용한 학습의 단순화

## 1. 학습 환경 구성

- 1) EC2, EBS 생성
- 2) 필요시 클러스터 구성

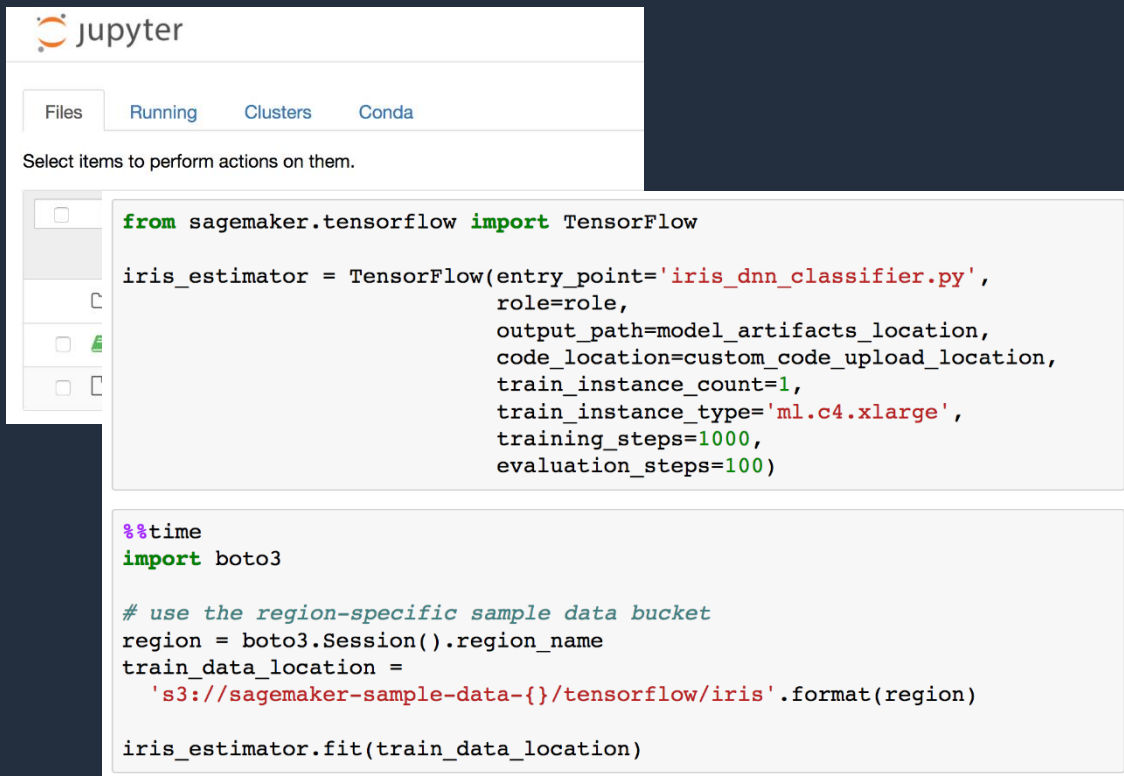
## 2. EC2에 필요한 파일들 복사

- 1) 학습 스크립트 복사
- 2) 학습 데이터 복사

## 3. 학습 수행

## 4. 학습 완료 후,

- 1) 모델을 영구 스토리지로 이동
- 2) 학습 환경 삭제



The screenshot shows the Jupyter web interface with a sidebar on the left containing a file explorer and tabs for 'Files', 'Running', 'Clusters', and 'Conda'. The main area displays two code cells. The first cell imports TensorFlow from SageMaker and creates an estimator. The second cell imports boto3 and configures the training data location in S3 before fitting the estimator.

```
from sagemaker.tensorflow import TensorFlow

iris_estimator = TensorFlow(entry_point='iris_dnn_classifier.py',
                           role=role,
                           output_path=model_artifacts_location,
                           code_location=custom_code_upload_location,
                           train_instance_count=1,
                           train_instance_type='ml.c4.xlarge',
                           training_steps=1000,
                           evaluation_steps=100)

%%time
import boto3

# use the region-specific sample data bucket
region = boto3.Session().region_name
train_data_location =
    's3://sagemaker-sample-data-{}-tensorflow/iris'.format(region)

iris_estimator.fit(train_data_location)
```



# Amazon SageMaker를 Apache Spark와 함께 사용

데이터 전처리

모델 학습

모델 호스팅

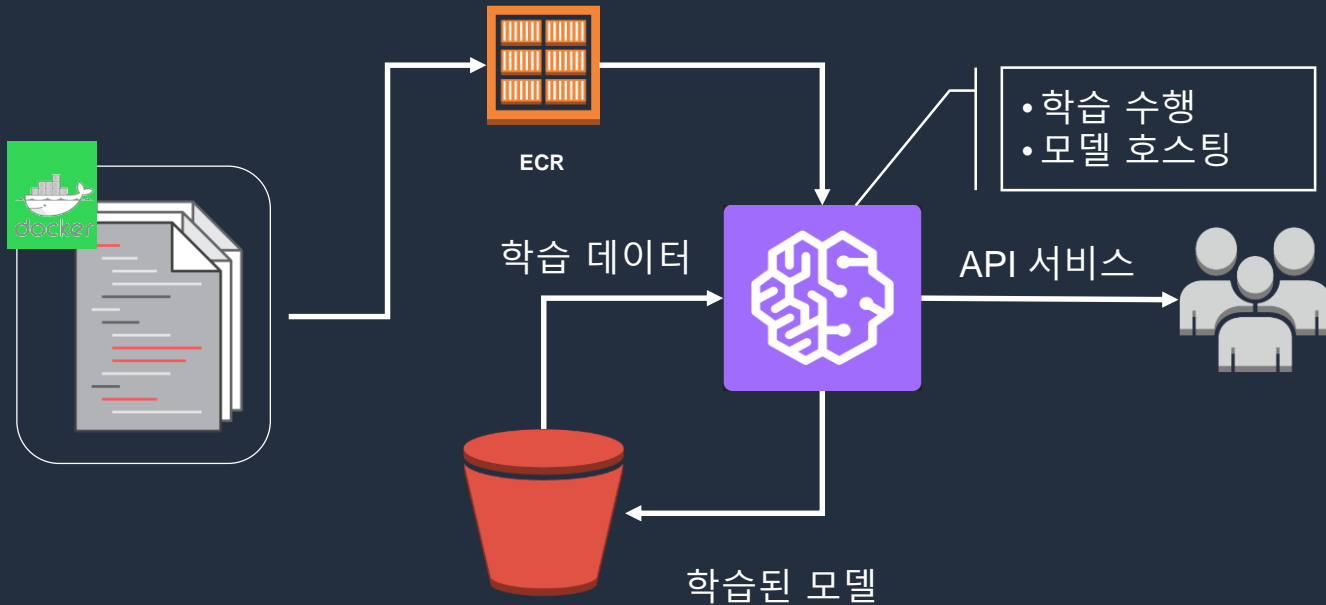
Apache Spark

Amazon SageMaker

- 연동을 위한 SageMaker Spark SDK 제공
- Spark ML 파이프라인에 SageMaker를 통합해서 학습 및 모델 호스팅에 사용할 수 있음

# 어떤 머신러닝/딥러닝 프레임워크와도 함께 사용

- ML/DL 프레임워크
- 학습 스크립트
- 예측 스크립트
- API 설정

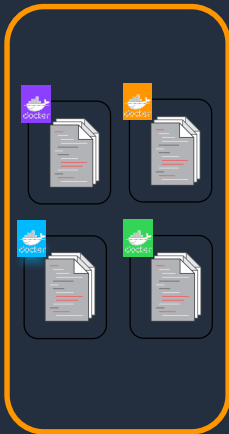


4

# 쉬운 모델 배포



ML Hosting Service



Amazon ECR



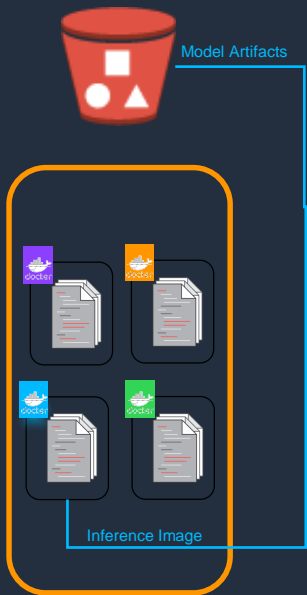
Amazon SageMaker

4

# 쉬운 모델 배포



ML Hosting Service



Amazon ECR



모델 생성



Amazon SageMaker



4

# 쉬운 모델 배포



모델의 버전들 생성

여러 버전의  
모델

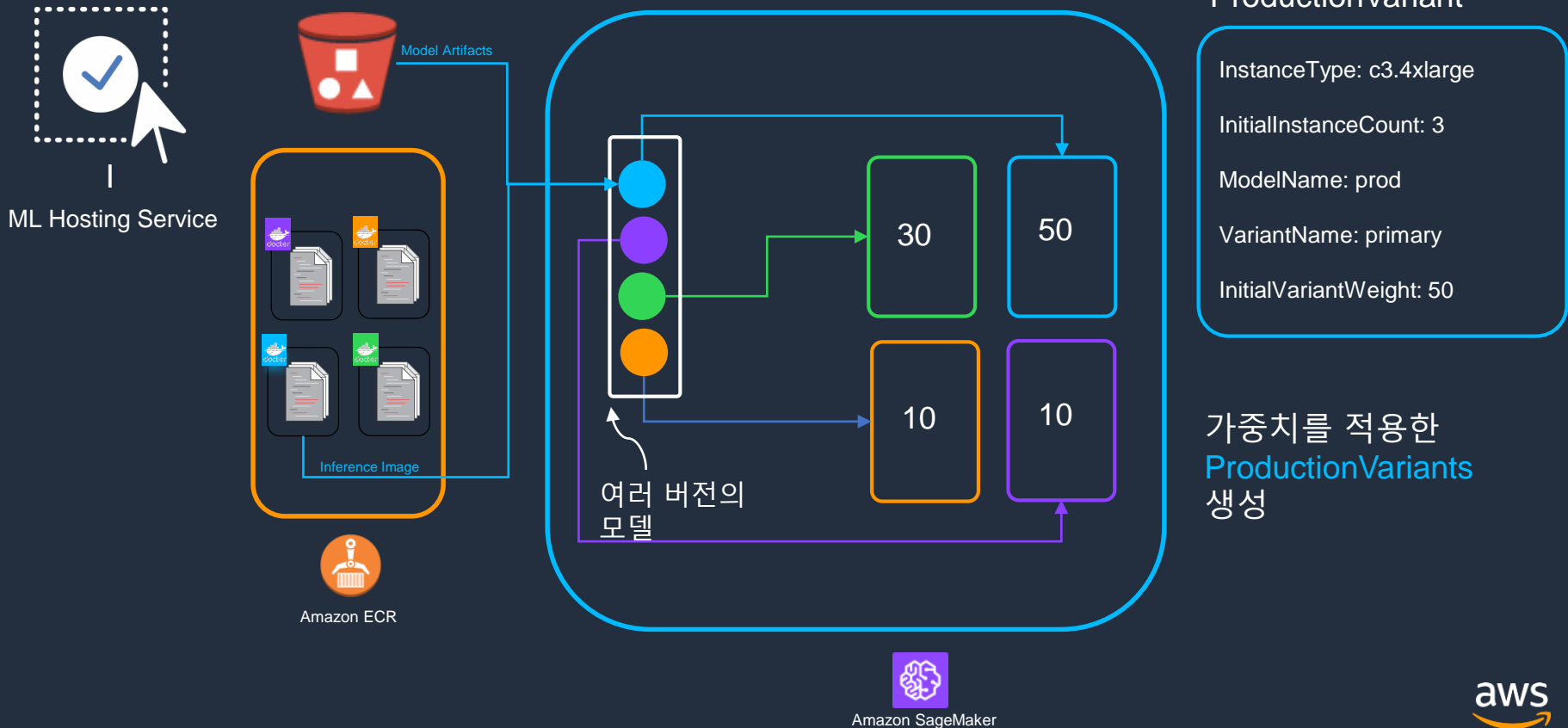


Amazon SageMaker



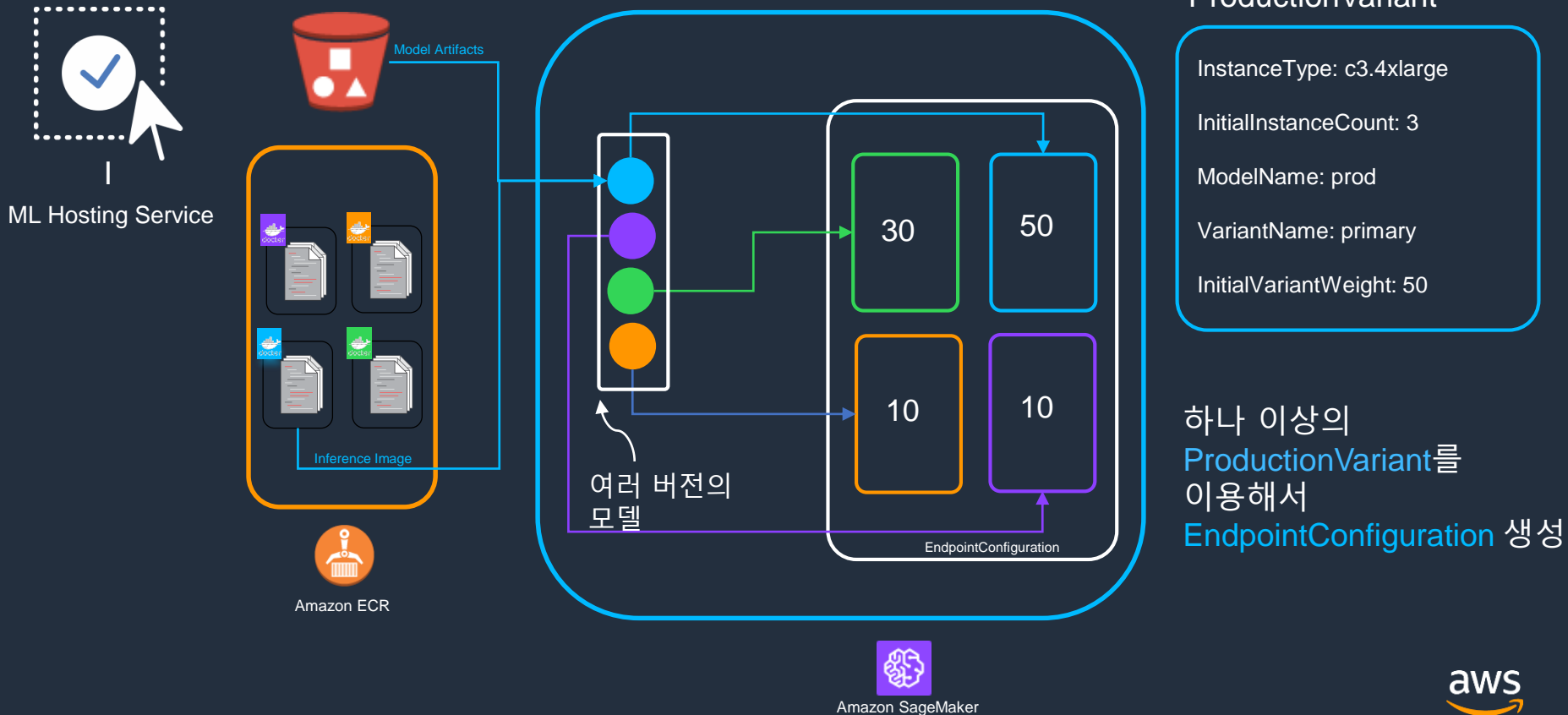
4

# 쉬운 모델 배포



4

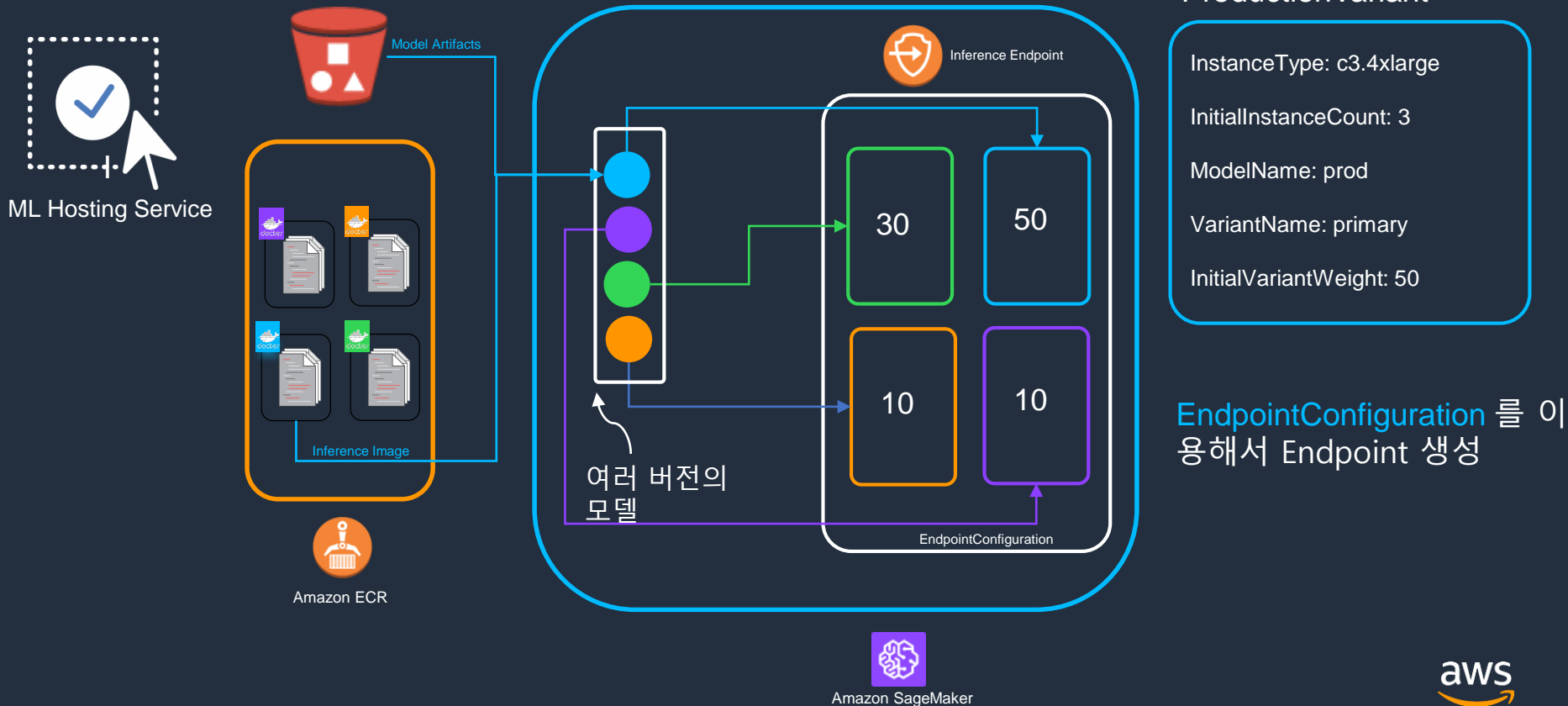
# 쉬운 모델 배포





4

# 쉬운 모델 배포



# 데모: 학습에서 배포까지

# 데모 – 어떤 문제를 풀어볼까요



From left to right, *Iris setosa* (by [Radomil](#), CC BY-SA 3.0), *Iris versicolor* (by [Dlanglois](#), CC BY-SA 3.0), and *Iris virginica* (by [Frank Mayfield](#), CC BY-SA 2.0).

# 데모 – 어떤 문제를 풀어볼까요

Sepal Length	Sepal Width	Petal Length	Petal Width	Species
5.1	3.5	1.4	0.2	0
4.9	3.0	1.4	0.2	0
4.7	3.2	1.3	0.2	0
...	...	...	...	...
7.0	3.2	4.7	1.4	1
6.4	3.2	4.5	1.5	1
6.9	3.1	4.9	1.5	1
...	...	...	...	...
6.5	3.0	5.2	2.0	2
6.2	3.4	5.4	2.3	2
5.9	3.0	5.1	1.8	2

## Species

0 – Iris Setosa  
1 – Iris Versicolor  
2 – Iris Virginica

## 데이터

학습 데이터: 120 샘플 / iris\_training.csv  
테스트 데이터: 30 샘플 / iris\_test.csv

# 데모 - 학습, 예측 스크립트 준비

```
jupyter iris_dnn_classifier.py ✓ 12/15/2017
File Edit View Language Python

1 import numpy as np
2 import
3 import
4
5 INPUT_      8 def estimator_fn(run_config, params):
6              9     feature_columns = [tf.feature_column.numeric_column(INPUT_TENSOR_NAME, shape=[4])]
7              10    return tf.estimator.DNNClassifier(feature_columns=feature_columns,
8              11                                   hidden_units=[10, 20, 10],
9              12                                   n_classes=3,
10             13                                   config=run_config)
11             14
12
13
14
15
16 def serving_input_fn(params):
17     feature_spec = {INPUT_TENSOR_NAME: tf.FixedLenFeature(dtype=tf.float32, shape=[4])}
18     return tf.estimator.export.build_parsing_serving_input_receiver_fn(feature_spec)()
19
20
21 def train_input_fn(training_dir, params):
22     """Returns input function that would feed the model during training"""
23     return _generate_input_fn(training_dir, 'iris_training.csv')
24
25
26 def eval_input_fn(training_dir, params):
27     """Returns input function that would feed the model during evaluation"""
28     return _generate_input_fn(training_dir, 'iris_test.csv')
29
30
31 def _generate_input_fn(training_dir, training_filename):
32     training_set = tf.contrib.learn.datasets.base.load_csv_with_header(
33         filename=os.path.join(training_dir, training_filename),
34         target_dtype=np.int,
35         features_dtype=np.float32)
36
37     return tf.estimator.inputs.numpy_input_fn(
38         x={INPUT_TENSOR_NAME: np.array(training_set.data)},
39         y=np.array(training_set.target),
40         num_epochs=None,
41         shuffle=True)()
42
```

# 데모 – S3 버킷 이름 정의

## Let us first initialize variables

```
In [1]: from sagemaker import get_execution_role

#Bucket location to save your custom code in tar.gz format.
custom_code_upload_location = 's3://sagemaker-muhyun-iad/customcode/tensorflow_iris'

#Bucket location where results of model training are saved.
model_artifacts_location = 's3://sagemaker-muhyun-iad/artifacts'

#IAM execution role that gives SageMaker access to resources in your AWS account.
role = get_execution_role()
```

# 데모 – SageMaker TensorFlow 객체 생성

```
In [2]: from sagemaker.tensorflow import TensorFlow

iris_estimator = TensorFlow(entry_point='iris_dnn_classifier.py',
                             role=role,
                             output_path=model_artifacts_location,
                             code_location=custom_code_upload_location,
                             train_instance_count=1,
                             train_instance_type='ml.c4.xlarge',
                             training_steps=1000,
                             evaluation_steps=100)
```

# 데모 - 학습 수행

```
In [4]: %%time
import boto3

# use the region-specific sample data bucket
region = boto3.Session().region_name
train_data_location = 's3://sagemaker-sample-data-{}-tensorflow/iris'.format(region)

iris_estimator.fit(train_data_location)

INFO:tensorflow:Evaluation [99/100]
INFO:tensorflow:Evaluation [100/100]
INFO:tensorflow:Finished evaluation at 2018-01-15-04:52:46
INFO:tensorflow:Saving dict for global step 1000: accuracy = 0.966719, average_loss = 0.0529209, global_step = 1000, loss = 6.77388
INFO:tensorflow:Restoring parameters from s3://sagemaker-muhyun-iaid/artifacts/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568/checkpoints/model.ckpt-1000
INFO:tensorflow:Assets added to graph.
INFO:tensorflow:No assets to write.
INFO:tensorflow:SavedModel written to: s3://sagemaker-muhyun-iaid/artifacts/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568/checkpoints/export/Servo/temp-1515991967/saved_model.pb
INFO:tensorflow:writing success training
2018-01-15 04:52:49,820 INFO - botocore.vendored.requests.packages.urllib3.connectionpool - Starting new HTTPS connection (1): s3.amazonaws.com
2018-01-15 04:52:49,939 INFO - tf_container.serve - Downloaded saved model at /opt/ml/model/export/Servo/1515991967/saved_model.pb
===== Job Complete =====
CPU times: user 444 ms, sys: 60 ms, total: 504 ms
Wall time: 6min 39s
```



# 데모 - 학습 수행

Amazon SageMaker > Jobs

**Jobs**

Create modelStopAdd/Edit tagsCreate training job

< 1 ... > ⚙

	Name ▼	Creation time ▼	Duration	Status ▼
<input type="radio"/>	sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568	Jan 15, 2018 04:47 UTC	—	🕒 InProgress



# 데모 - 학습 수행

Amazon SageMaker > Jobs > sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

## sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

[Stop](#) [Create model](#)

### Job settings

Job name	Training duration
sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568	—
ARN	IAM role ARN
arn:aws:sagemaker:us-east-1:850550765017:training-job/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568	<a href="#">arn:aws:iam::850550765017:role/service-role/AmazonSageMaker-ExecutionRole-20171221T135887</a> 
Status	
 InProgress	
Creation time	
Jan 15, 2018 04:47 UTC	
Last modified time	
Jan 15, 2018 04:47 UTC	

# 데모 - 모델 배포 및 테스트

## Deploy the trained Model

The `deploy()` method creates an endpoint which serves prediction requests in real-time.

```
In [5]: %%time
iris_predictor = iris_estimator.deploy(initial_instance_count=1,
                                       instance_type='ml.m4.xlarge')
```

INFO:sagemaker:Creating model with name: sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568  
INFO:sagemaker:Creating endpoint with name sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

---

-----!CPU times: user 524 ms, sys: 48 ms, total: 572 ms  
Wall time: 11min 37s

# 데모 - 모델 배포 및 테스트

[Amazon SageMaker](#) > [Endpoint configuration](#) > sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

sagemaker-tensorflow-py2-cpu-  
2018-01-15-04-47-01-568

Delete

Apply to endpoint

## Endpoint configuration settings

Copy and edit

Name  
sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

Creation time  
-

ARN  
arn:aws:sagemaker:us-east-1:850550765017:endpoint-  
config/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

## Production variants

Model name	Variant name	Instance type	Initial instance count	Initial weight
<a href="#">sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568</a>	AllTraffic	ml.m4.xlarge	1	1

# 데모 - 모델 배포 및 테스트

Amazon SageMaker > Endpoints > sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

Delete

## Endpoint settings

Name

sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

ARN

arn:aws:sagemaker:us-east-1:850550765017:endpoint/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568

URL

<https://runtime.sagemaker.us-east-1.amazonaws.com/endpoints/sagemaker-tensorflow-py2-cpu-2018-01-15-04-47-01-568/invocations>

[Learn more about the API](#)

Status

 InService

Creation time

Mon Jan 15 2018 12:56:06 GMT+0800

Last updated

Mon Jan 15 2018 13:07:41 GMT+0800

# 데모 - 모델 배포 및 테스트

## Invoke the Endpoint to get inferences

Invoking prediction:

```
In [6]: iris_predictor.predict([6.4, 3.2, 4.5, 1.5]) #expected label to be 1
```

```
Out[6]: {'result': {'classifications': [{'classes': [{'label': '0',  
    'score': 0.00013661041157320142},  
    {'label': '1', 'score': 0.9990039467811584},  
    {'label': '2', 'score': 0.0008594192913733423}]}]}}
```



## 본 강연이 끝난 후...

1. “Getting started with Amazon SageMaker”를 통한 간단한 실습
2. Amazon SageMaker SDK 사용법 익히기
  - Python: <https://github.com/aws/sagemaker-python-sdk>
  - Spark: <https://github.com/aws/sagemaker-spark>
3. SageMaker 예제들  
<https://github.com/aws-labs/amazon-sagemaker-examples>

# Thank You For Attending AWS Innovate

We hope you found it interesting! A kind reminder to **complete the survey**.  
Let us know what you thought of today's event and how we can improve the event experience for you in the future.



[aws-korea-marketing@amazon.com](mailto:aws-korea-marketing@amazon.com)



[youtube.com/user/AmazonWebServices](https://youtube.com/user/AmazonWebServices)



[twitter.com/AWSCloud](https://twitter.com/AWSCloud)



[slideshare.net/AmazonWebServices](https://slideshare.net/AmazonWebServices)



[facebook.com/AmazonWebServices](https://facebook.com/AmazonWebServices)



[twitch.tv/aws](https://twitch.tv/aws)