Amazon SageMaker

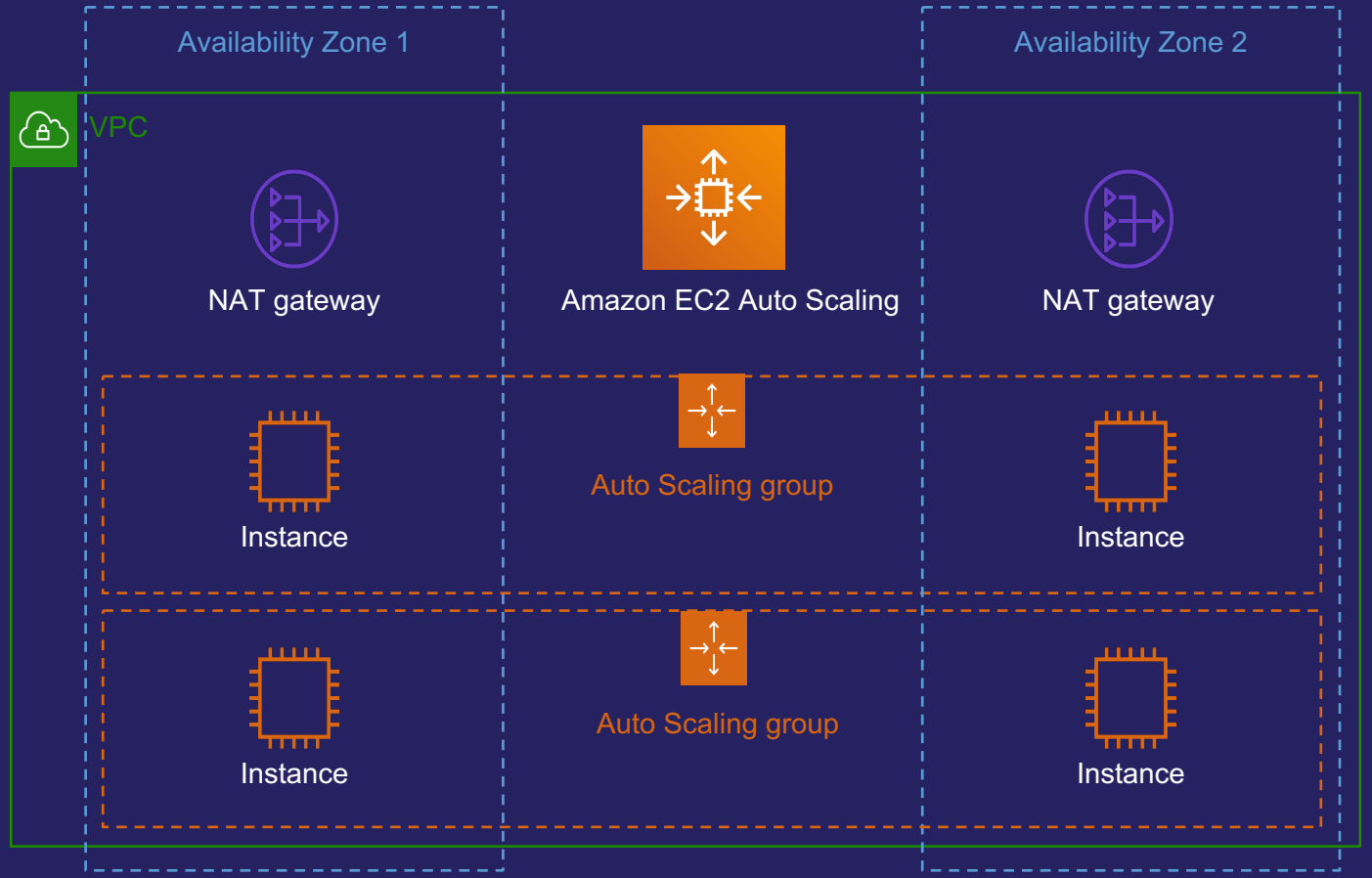
Deployment

최영준
AIML Expert SA
AWS

개요

밑바닥부터 모델 서빙을 구현하려면?

- 인프라 설정
- 고가용성
- 프레임워크 버전 관리는?
- 트래픽이 몰린다면?
- A/B 테스트
- 보안을 고려하려면?
- 비용을 아끼고 싶은데?
- ….



클라우드 네이티브 모델 서빙의 이점은 무엇일까요?

손쉬운 모델 배포 및 관리

몇 분 안에 시작할 수 있는 엔드포인트endpoint

인프라 관리, 패치 및 기본 제공 업데이트

Amazon CloudWatch에서 엔드포인트에 대한 지표 및 로그 수집

최고의 가격 대비 성능 절충안

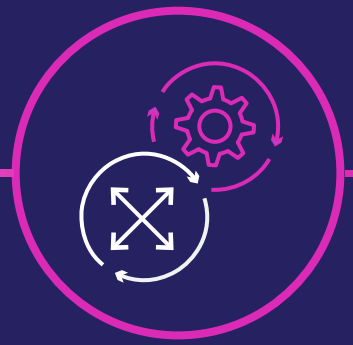
99.99% 서비스 가용성 SLA

70개 이상의 SageMaker 머신 러닝 인스턴스

트래픽 기반 오토스케일링

엔드포인트에 다중 모델 배포

MLOps 통합



CI/CD: SageMaker 파이프라인 및 프로젝트

모델 레지스트리Model Registry: 카탈로그 모델, 버전 관리, 승인 워크플로

모델 모니터Model Monitor: 데이터 및 모델 드리프트에 대한 경고

SageMaker 추론inference 포트폴리오

Amazon SageMaker 빌트인

Amazon SageMaker

SAGEMAKER STUDIO IDE

리얼타임 추론 | 비동기 추론 | 배치 추론 | 멀티 모델 엔드포인트 | 멀티 컨테이너 엔드포인트 | 추론 파이프라인 | 모델 버전 관리 | CI/CD | 모델 모니터링 | CloudWatch 지표 및 로깅

머신 러닝 프레임워크

TensorFlow | PyTorch | mxnet | GLUON | | scikit learn | XGBoost | ONNX

모델 서버

AWS Deep Learning Containers | TensorFlow Serving | TorchServe | NVIDIA Triton Inference Server | AWS Multi Model Server (MMS) | Nginx + gunicorn

머신 러닝 인스턴스

CPUs | GPUs | Inferentia | Graviton (ARM)

컴퓨팅 가속기

SageMaker Neo | NVIDIA TensorRT/ cuDNN | Intel oneDNN | ARM Compute Library

SageMaker 빌트인 모델 서빙 4가지 주요 패턴

리얼타임 추론

Multi-container Endpoint:
2021.03 GA
NVIDIA Triton Inference:
2021.11 GA

- Low latency
- Ultra high throughput
- A/B testing

배치 추론

- Process large datasets
- Job-based system

비동기 추론

2021.08 GA

- Near real-time
- Large payloads (1 GB)
- Long timeouts (15 min)

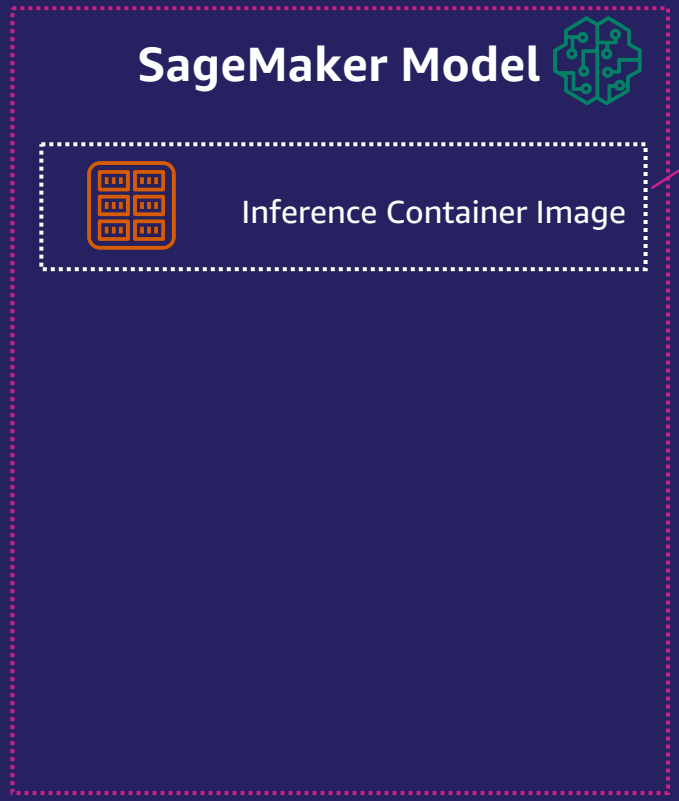
서버리스 추론

Lambda Serverless Inference:
2021.Q3 GA
SageMaker Serverless Inference:
2021.12 Public Preview

- Automatic scaling
- Pay-per-use pricing

빌트인 모델 서빙 동작 원리

1 모델Model 생성

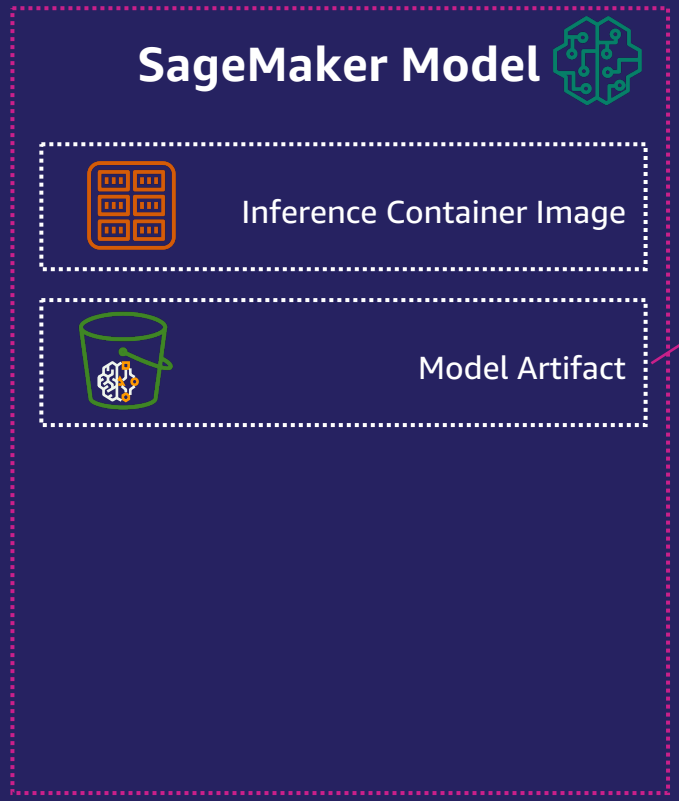


ECR 또는 Private Docker Registry에 저장된
SageMaker 추론 이미지의 경로

배포를 위한 모델 패키징

빌트인 모델 서빙 동작 원리

1 모델Model 생성



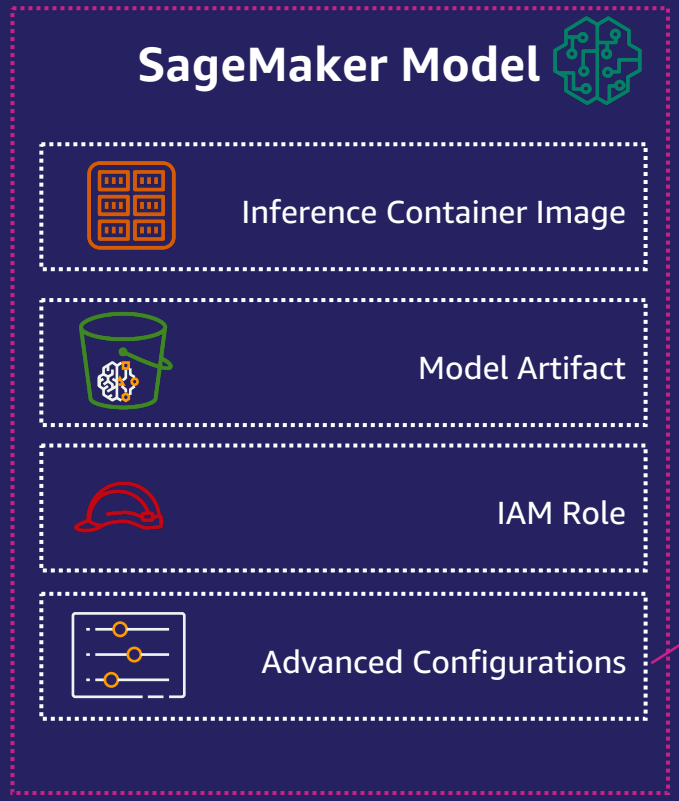
훈련된 모델 아티팩트에 대한 S3 경로

***SageMaker 빌트인 알고리즘에 필요*

배포를 위한 모델 패키징

빌트인 모델 서빙 동작 원리

1 모델Model 생성



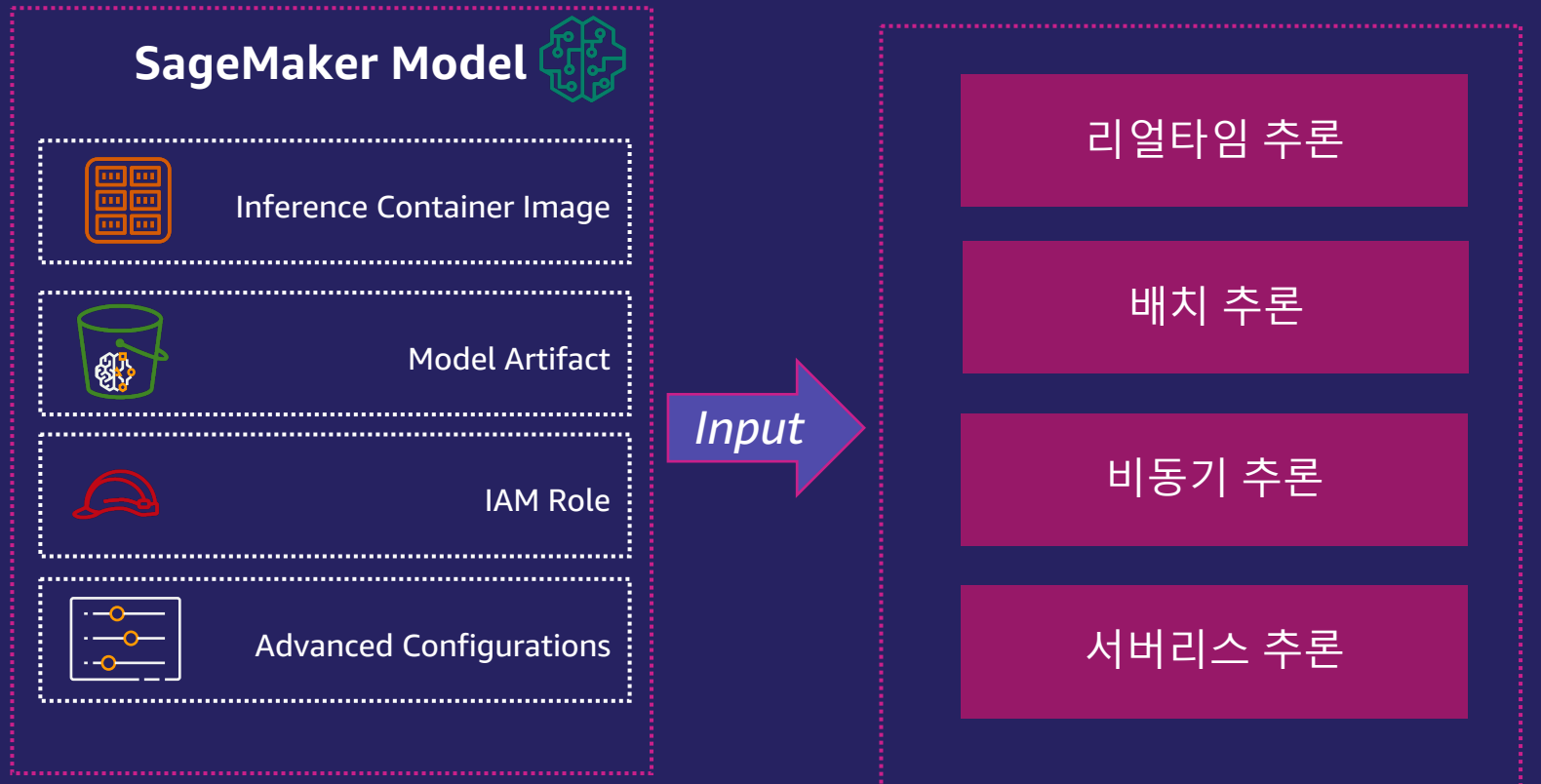
배포를 위한 모델 패키징

고급 구성 옵션은 선택한 배포 옵션에 따라 다릅니다.
예: VPC 구성, 다중 컨테이너 및 멀티 모델 배포.

빌트인 모델 서빙 동작 원리

1 모델Model 생성

2 엔드포인트 배포



배포를 위한 모델 패키징

4가지 주요 패턴 살펴보기 – 리얼타임 추론

SageMaker Deployment – 실시간 추론

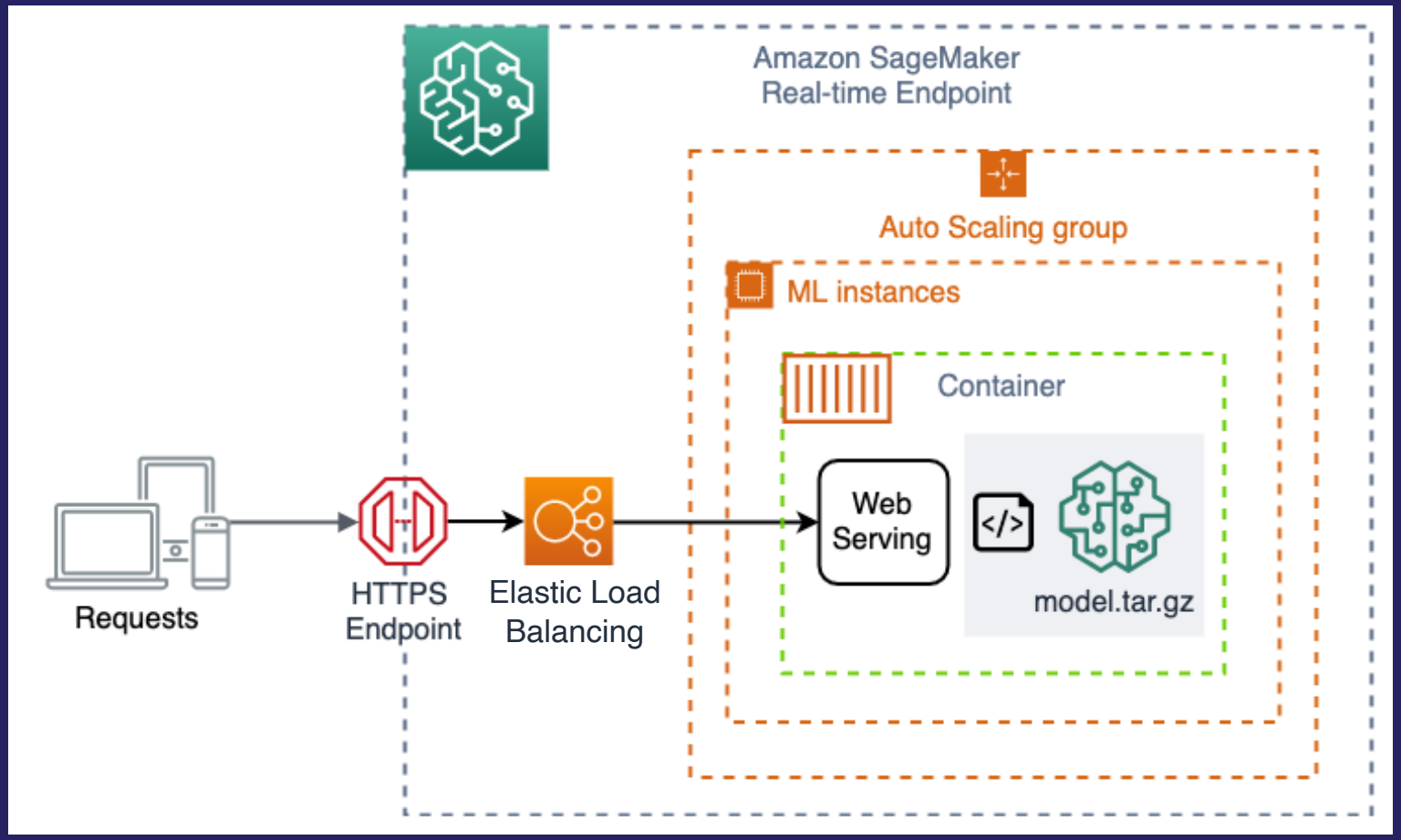
SageMaker Real-time Inference

장기 실행되는 마이크로 서비스 생성

최대 6MB 페이로드에 대한 실시간 응답

최대 60초의 타임아웃

Autoscaling



엔드포인트 생성 3단계

모델Model 생성

```
aws sagemaker create-model
  --model-name model1
  --primary-container '{"Image": "123.dkr.ecr.amazonaws.com/algo",
                        "ModelDataUrl": "s3://bkt/model1.tar.gz"}'
  --execution-role-arn arn:aws:iam::123:role/me
```

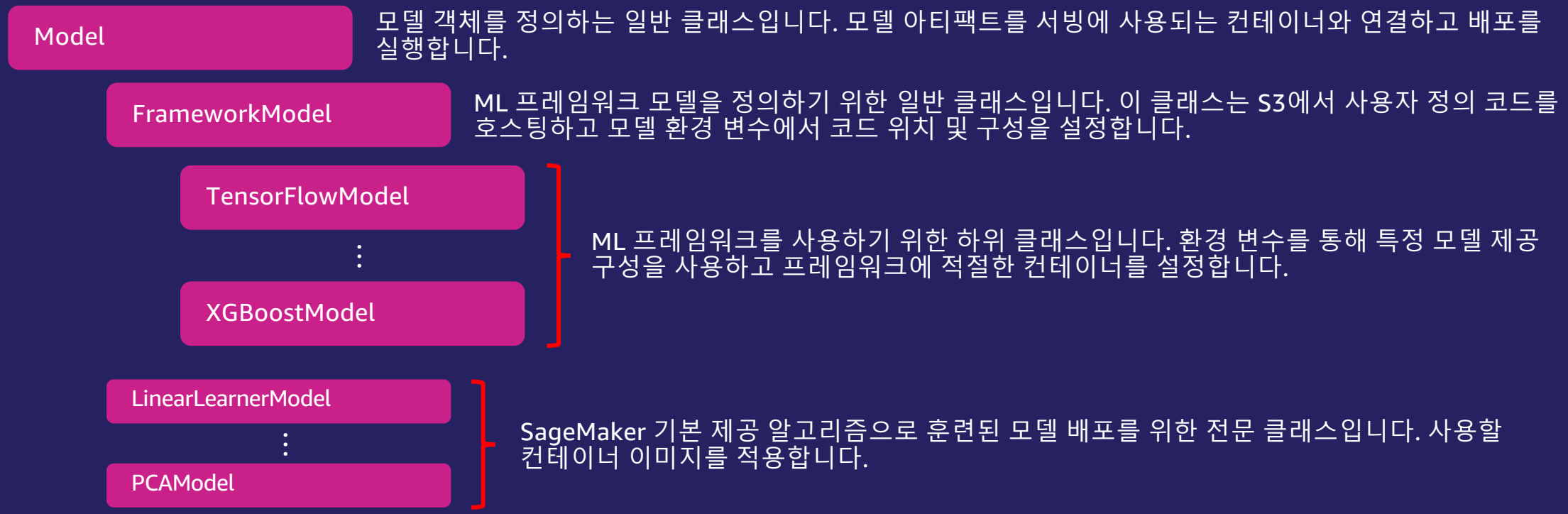
엔드포인트 구성
EndpointConfig 생성

```
aws sagemaker create-endpoint-config
  --endpoint-config-name model1-config
  --production-variants '{"InitialInstanceCount": 2,
                          "InstanceType": "ml.m4.xlarge",
                          "InitialVariantWeight": 1,
                          "ModelName": "model1",
                          "VariantName": "AllTraffic"}'
```

엔드포인트
Endpoint 생성

```
aws sagemaker create-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name model1-config
```

SageMaker SDK 핵심 클래스 - Model

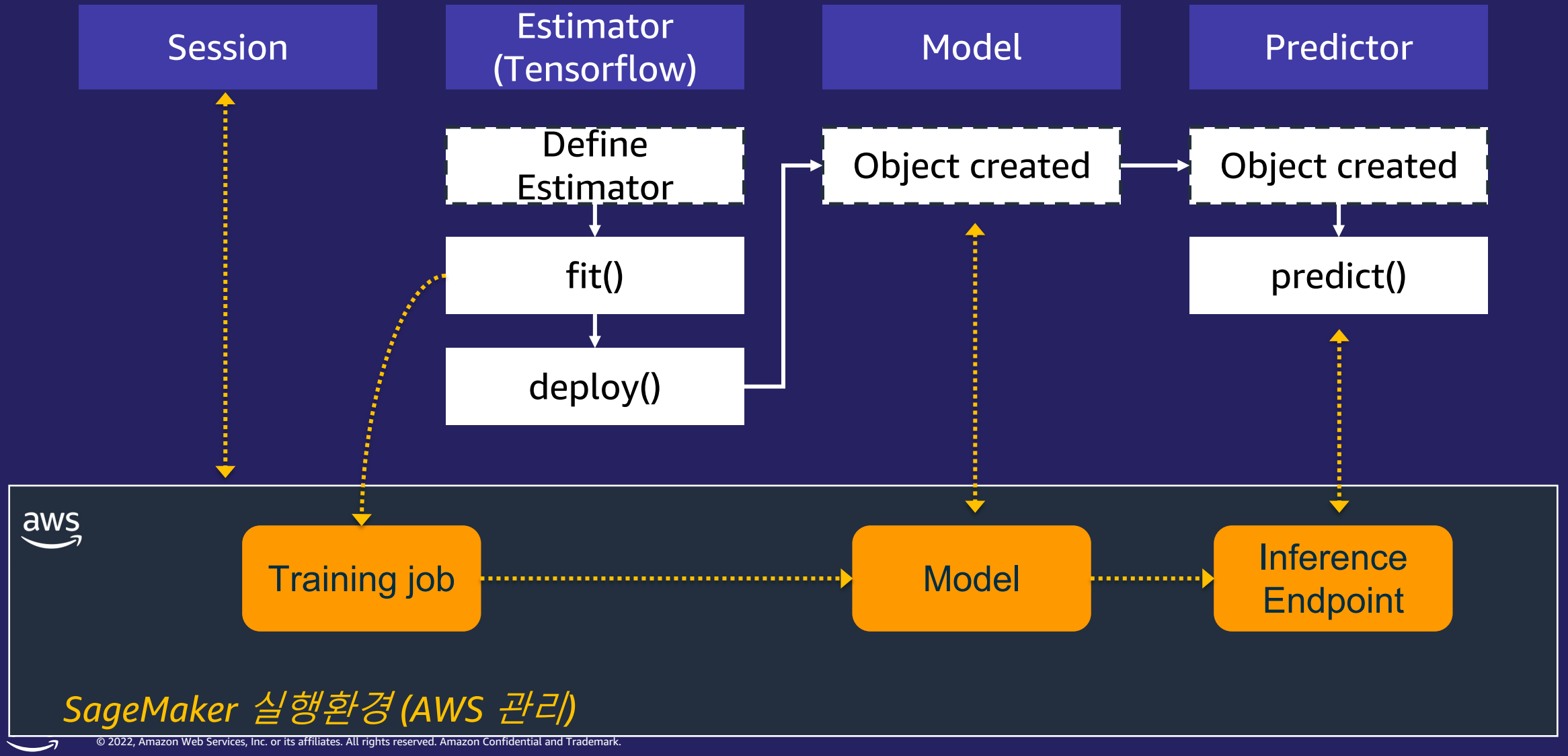



```
from sagemaker.tensorflow import TensorFlowModel

model = TensorFlowModel(model_data='s3://mybucket/model.tar.gz', role='MyRole')

predictor = model.deploy(initial_instance_count=1, instance_type='ml.c5.xlarge')
```

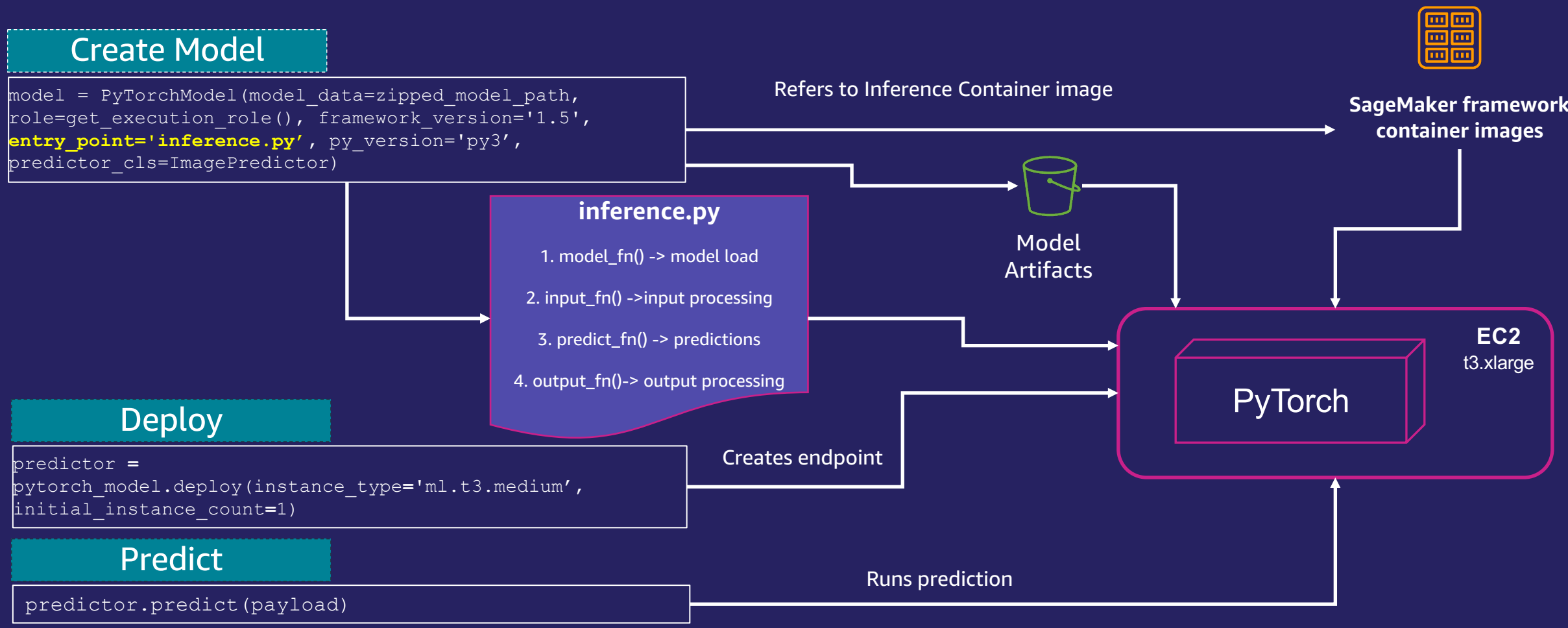
SageMaker SDK 핵심 클래스 - Predictor

```
from sagemaker.serializers import CSVSerializer

xgb_predictor = xgb.deploy(initial_instance_count=1,     --> Returns predictor object
                           instance_type='ml.m4.xlarge')

xgb_predictor.serializer = CSVSerializer()

predictions = xgb_predictor.predict(inf_data).decode('utf-8')
```

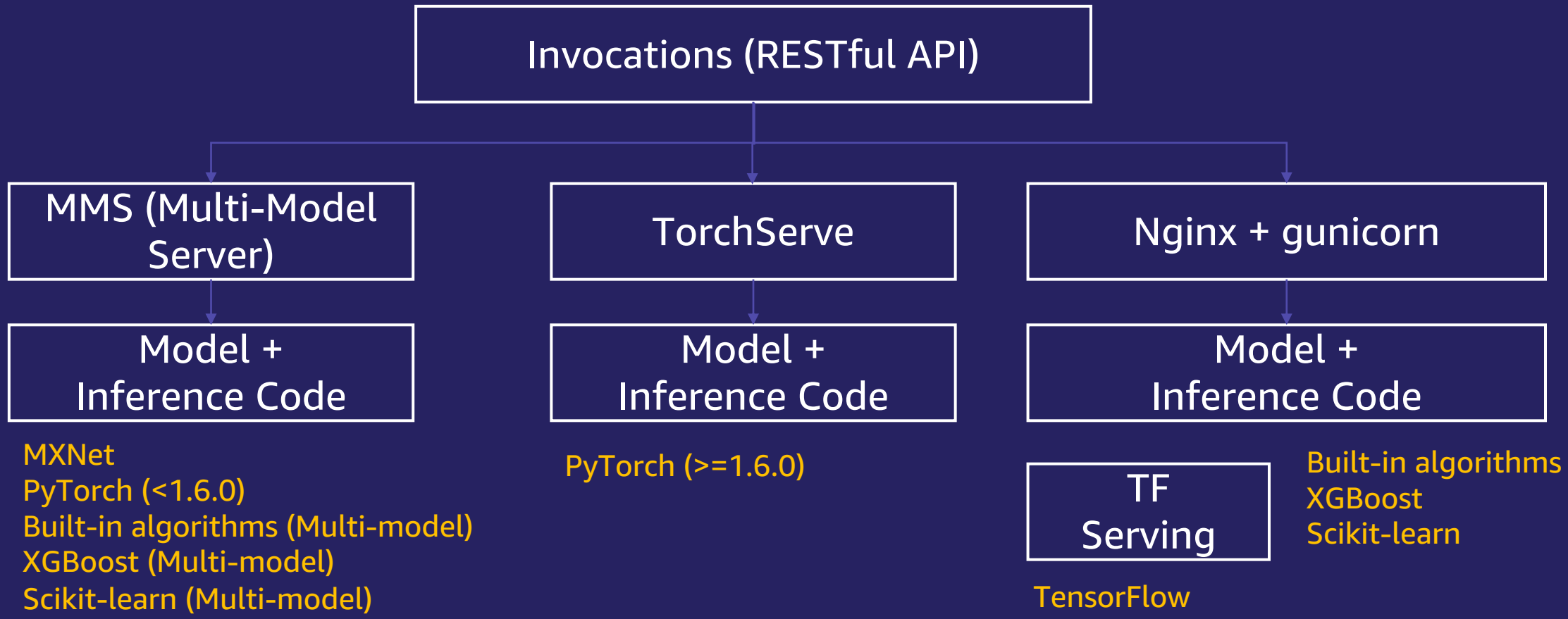
SageMaker SDK 핵심 클래스 관계



SageMaker SDK 엔드포인트 생성 과정 요약

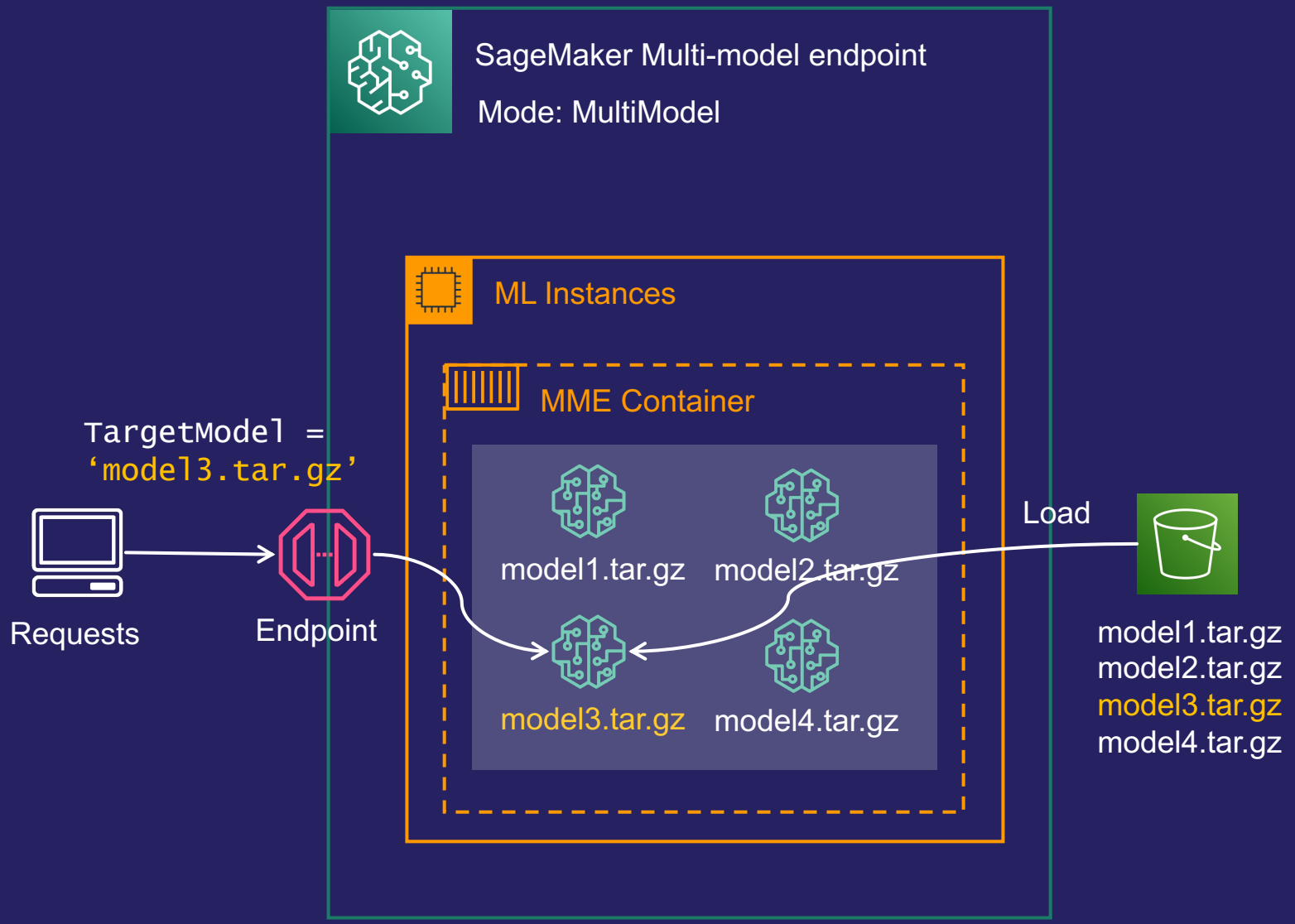


SageMaker 빌트인 웹서버



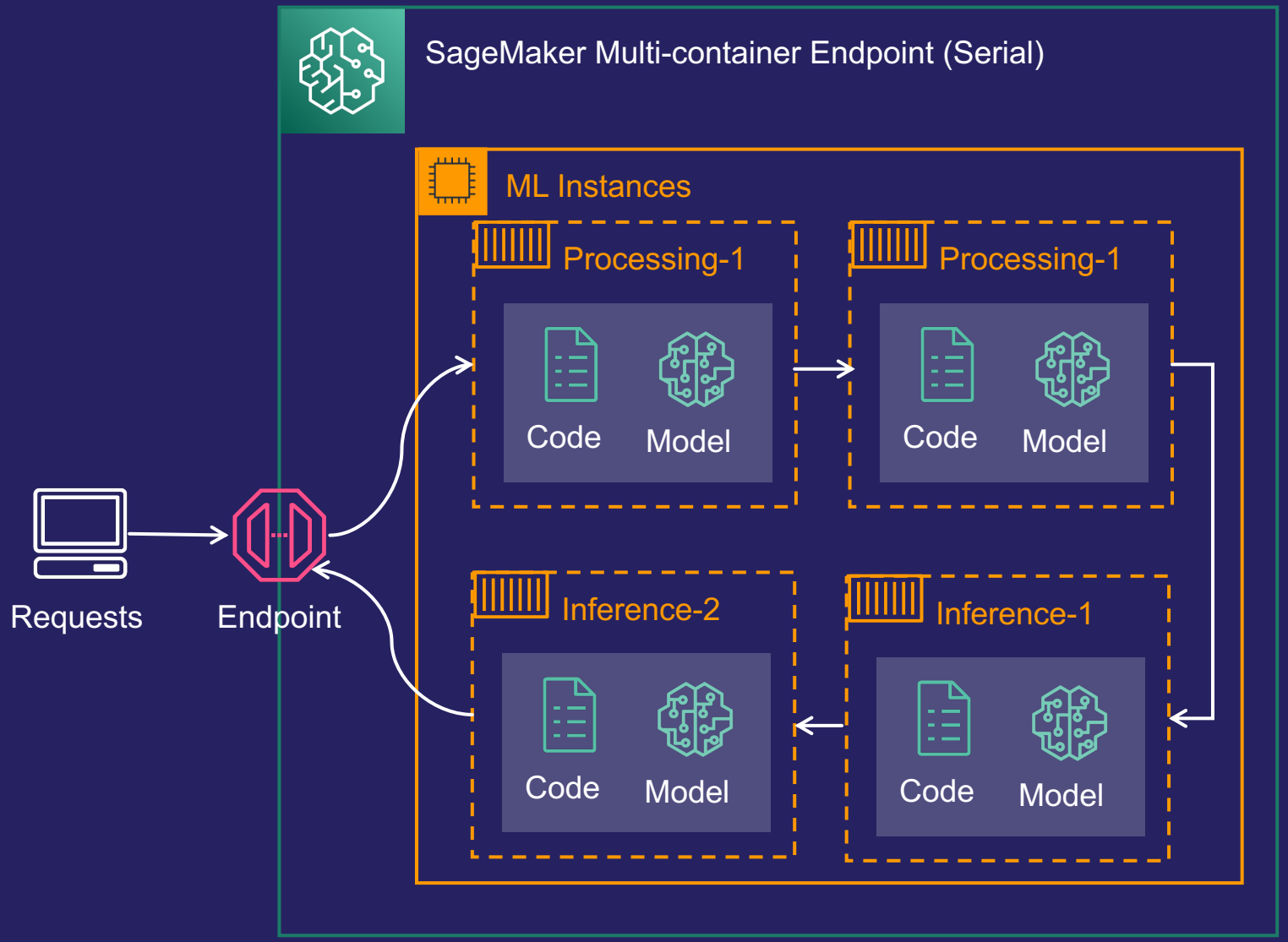
멀티 모델 엔드포인트 Multi-model Endpoint

- 단일 컨테이너에서 여러 모델 호스팅
- 타겟 모델에 대한 직접 호출direct invocations
- Amazon S3에서 동적으로 모델 로딩
- 콜드 스타트



멀티 컨테이너 엔드포인트multi-container Endpoint

- 최대 15개의 개별 컨테이너 호스팅 지원
- 직접 또는 직렬serial 호출
- 1GB 페이로드, 15분 타임아웃
- 콜드 스타트 없음



Code snippets

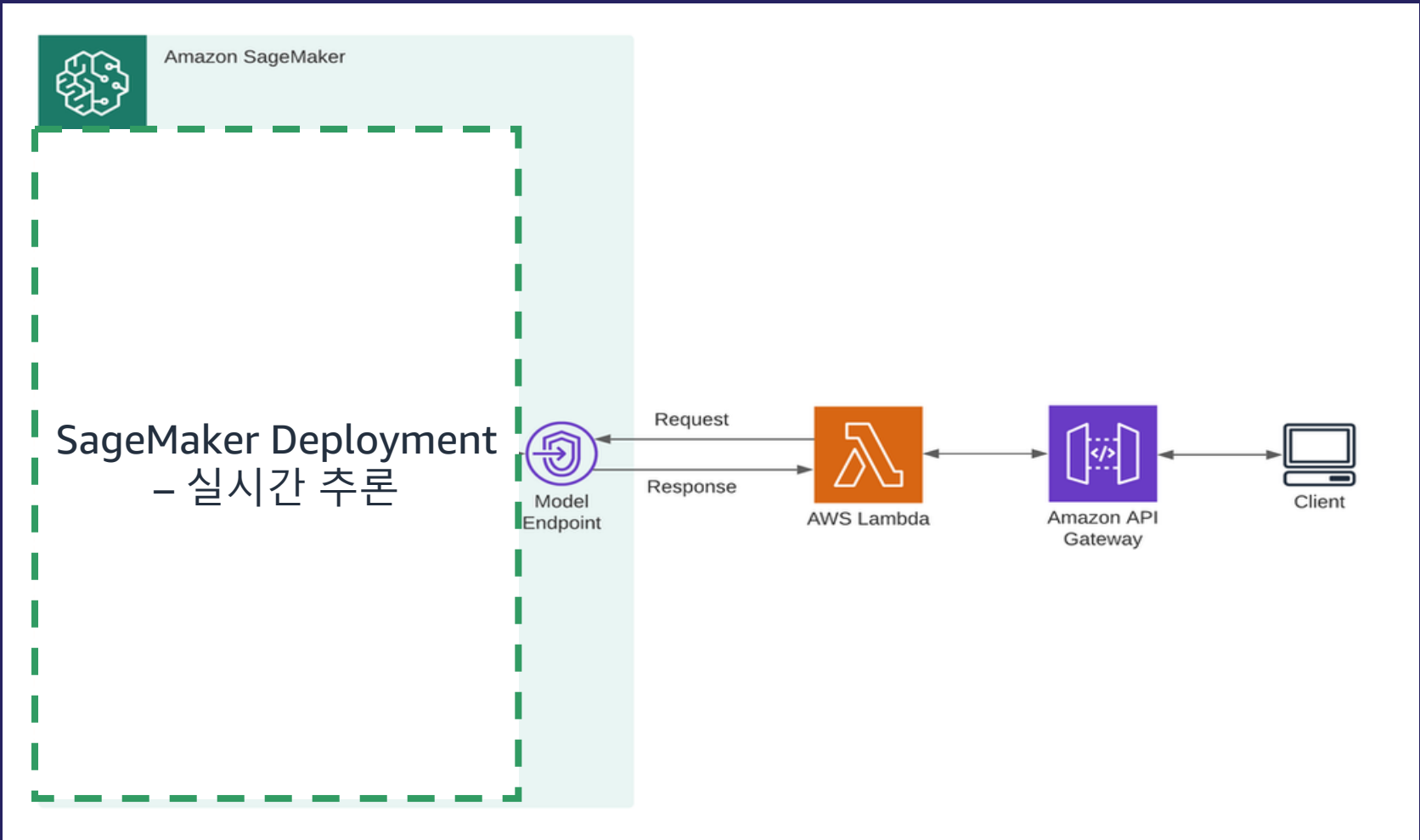
멀티 모델

```
container = {
   'Image': mme-supported-image,
  'ModelDataUrl': 's3://my-bucket/folder-of-tar-gz',
  'Mode': 'MultiModel'}

sm.create_model(
    Containers = [container], ...)

sm.create_endpoint_config()
sm.create_endpoint()

smrt.invoke_endpoint(
    EndpointName = endpoint_name,
    TargetModel = 'model-007.tar.gz',
    Body = body, ...
```

멀티 컨테이너

```
container1 = {
    'Image': ecr-image1,
    'ContainerHostname': 'firstContainer'}; ...

container2 = {
    'Image': ecr-image2,
    'ContainerHostname': 'secondtContainer'}; ...

sm.create_model(
    InferenceExecutionConfig = {'Mode': 'Direct'},
    Containers = [container1, container2, ...], ...)

sm.create_endpoint_config()
sm.create_endpoint()

smrt.invoke_endpoint(
    EndpointName = endpoint_name,
    TargetContainerHostname = 'firstContainer',
    Body = body, ...)
```

SageMaker Deployment – 실시간 추론



4가지 주요 패턴 살펴보기 – 배치 추론, 비동기 추론, 서버리스 추론

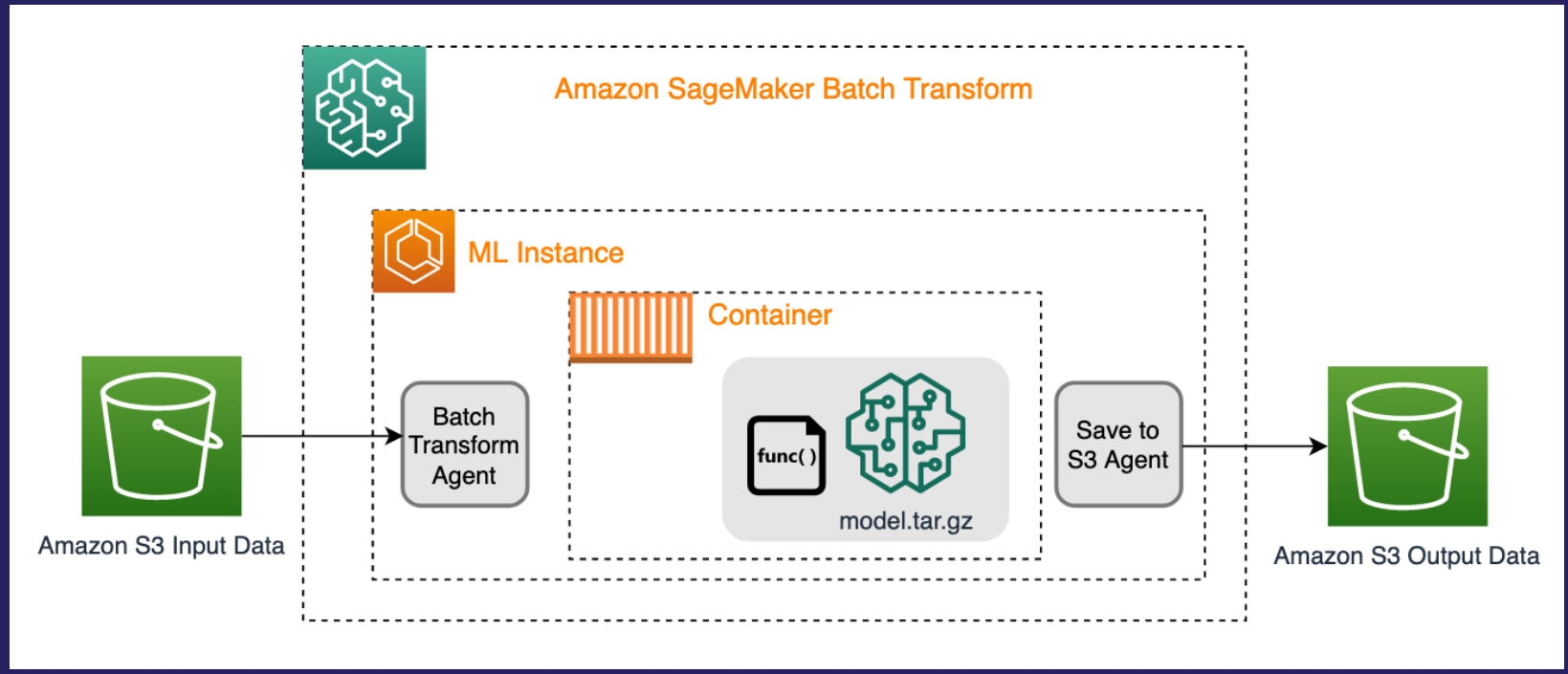
SageMaker Deployment – Batch Inference

SageMaker Batch Transform

Fully managed mini-batching for large data

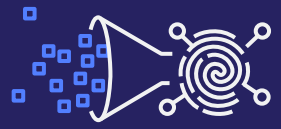
Pay only for what you use

Suitable for periodic arrival of large data



SageMaker Deployment – Async Inference

SageMaker Asynchronous Inference

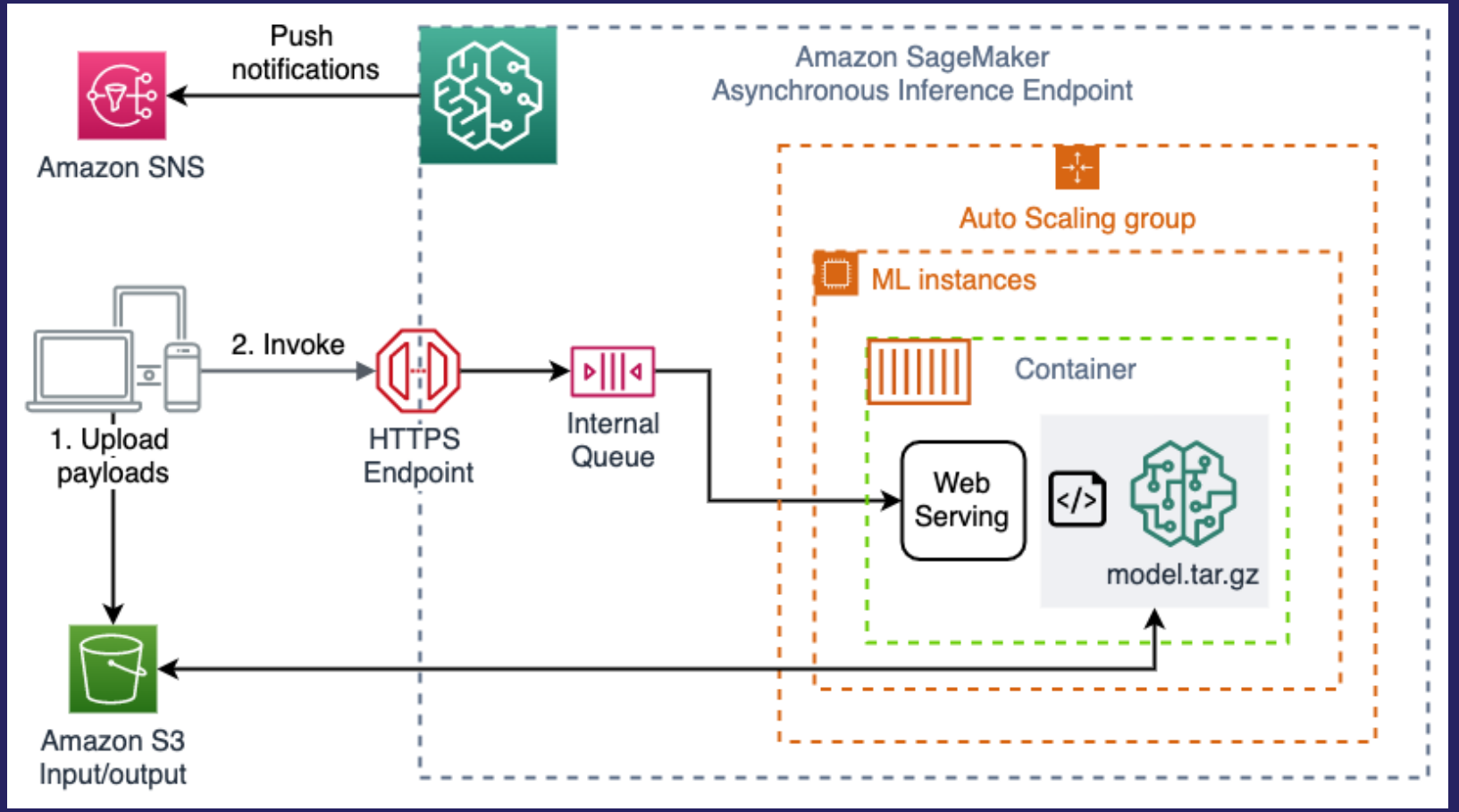


Ideal for large payload up to 1GB

Longer processing timeout up to 15 min

Autoscaling (down to 0 instance)

Suitable for CV/NLP use cases



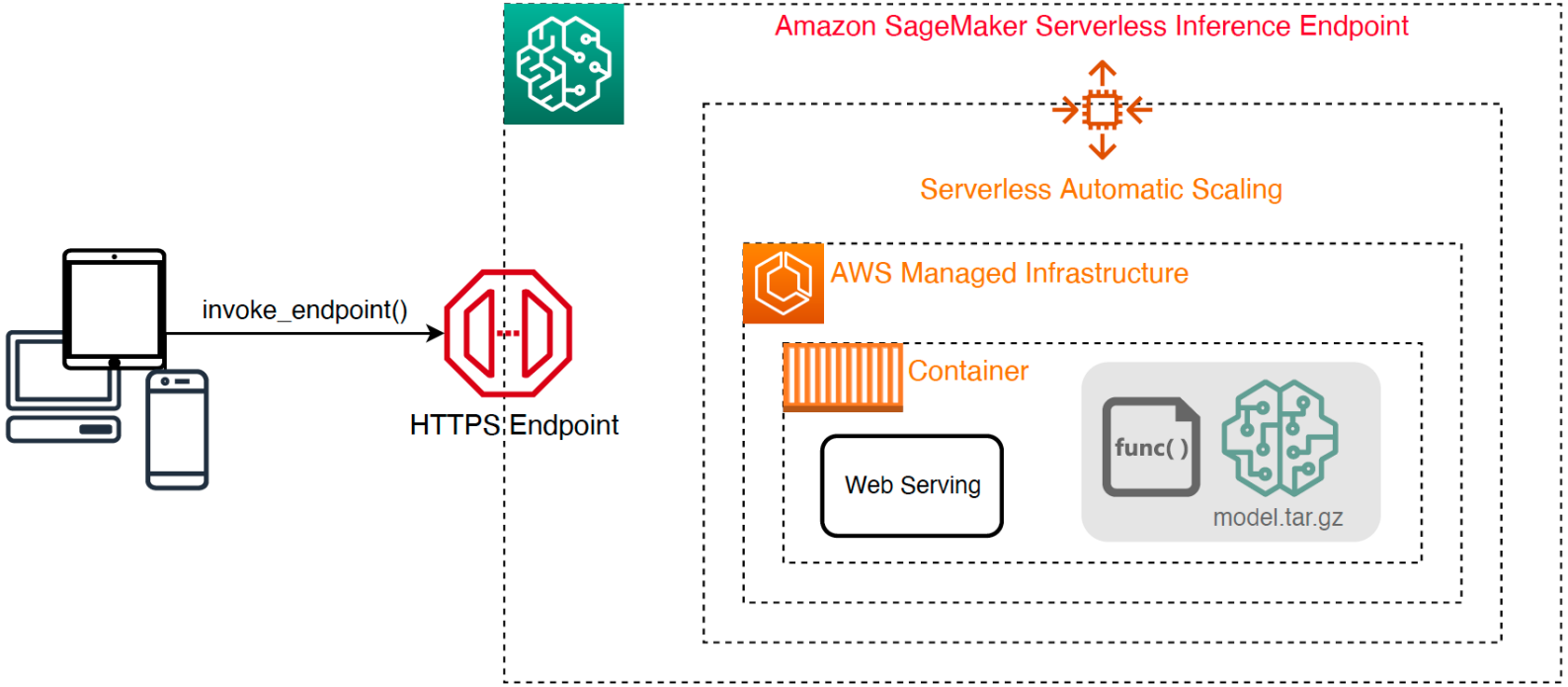
SageMaker Deployment – Serverless Inference

SageMaker Serverless Inference

Ideal for unpredictable prediction traffic

Workload tolerable to cold start

Autoscaling (down to 0 instance)



SageMaker 서버리스 추론

모델Model 생성

추론 컨테이너 이미지

ML 모델

엔드포인트 구성Endpoint Configuration 생성

Serverless 설정만 추가:

```
create_endpoint_config(
  ...
  "ServerlessConfig": {
    "MemorySizeInMB": 2048,
    "MaxConcurrency": 20
  }
)
```

엔드포인트 생성

기존과 동일

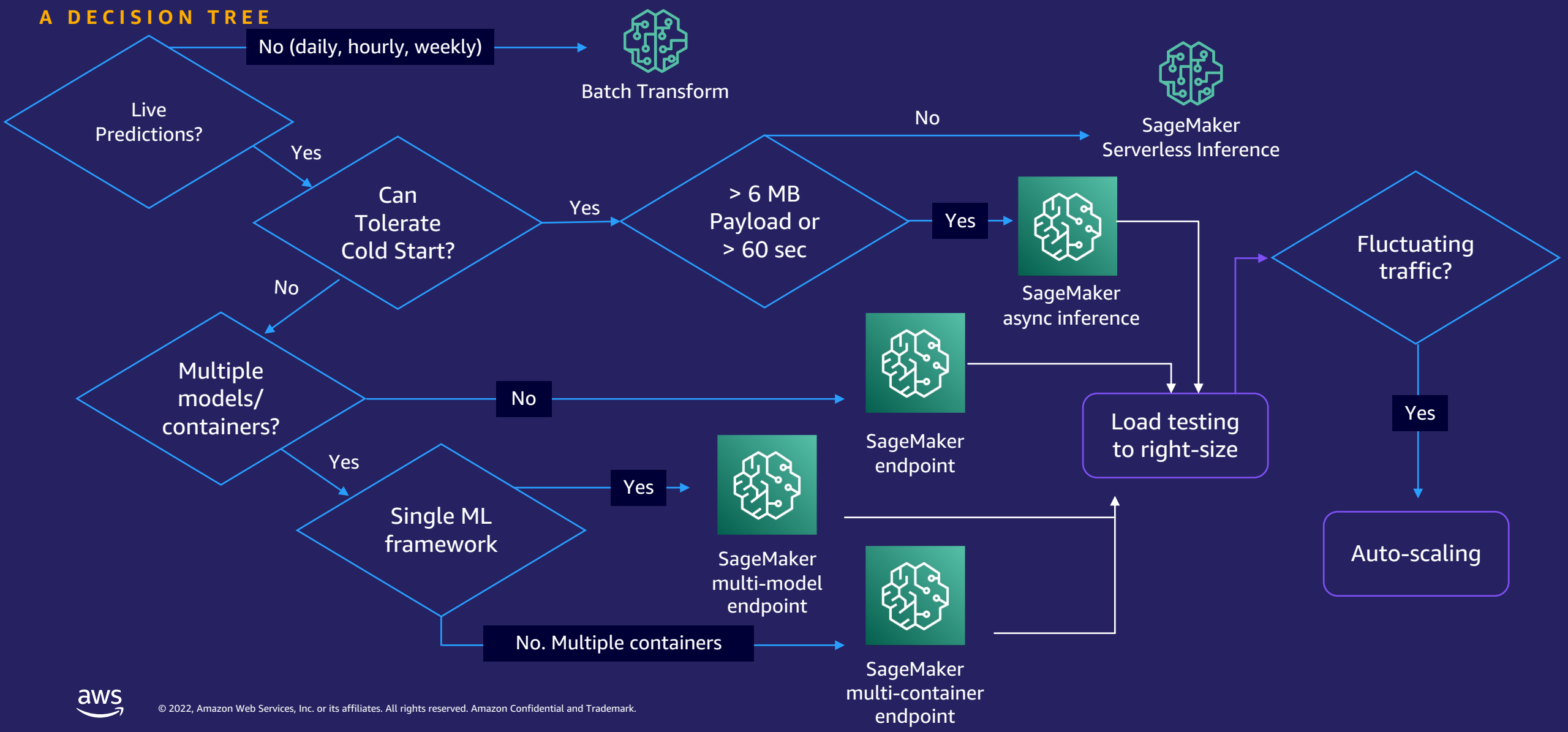
SageMaker Serverless Endpoint

```
response = sm_client.create_endpoint_config(
    EndpointConfigName="[YOUR-ENDPOINT-CONFIG]"
    ProductionVariants=[
        {
           "ModelName": "[YOUR-MODEL-NAME]",
           "VariantName": "AllTraffic",
           "ServerlessConfig": {
                "MemorySizeInMB": 2048,
                "MaxConcurrency": 20
           }

        }
    ]
)
```

사용 가능한 메모리 크기:
1GB/2GB/3GB/4GB/5GB/6GB

How to choose your Deployment Strategy

A DECISION TREE



프로덕션 적용

프로덕션 요구 사항

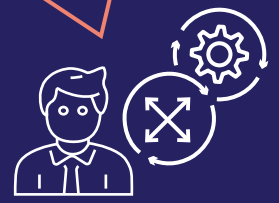
70개 이상 인스턴스 유형들 중 어떤 유형을 사용해야 하나요? 트래픽이 몰릴 때나 적을 때 어떻게 대처해야 하나요?

Product Owner
CDO/CTO

모델 드리프트/데이터 드리프트를 지속적으로 모니터링하고 싶어요.

데이터 과학자

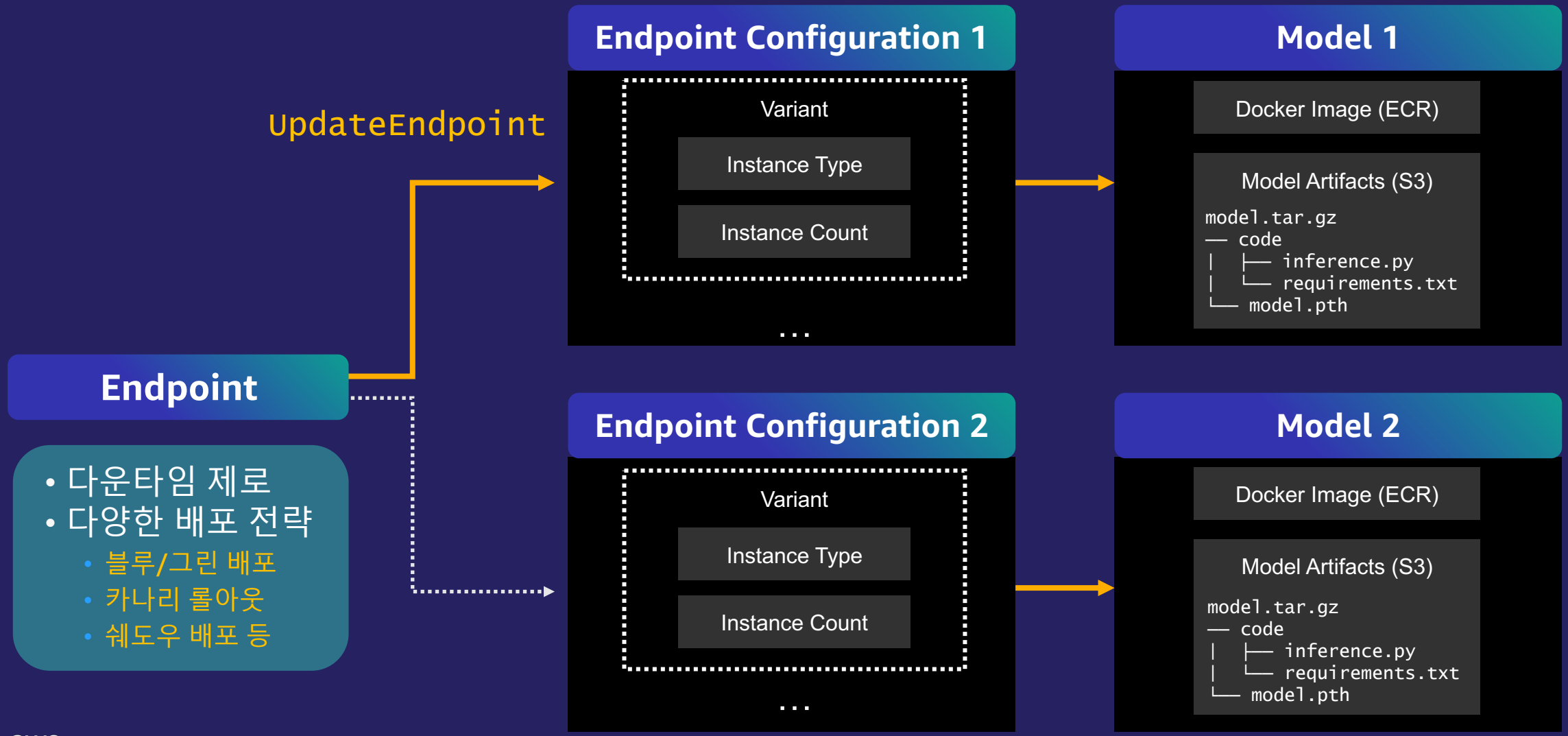
무중단 배포를 쉽게 할 수 있나요? 로드 테스트를 위해 별도의 서드파티 툴킷을 사용해야 하나요? CI/CD 파이프라인을 구성해야 해요.



MLOps 엔지니어

복잡한 솔루션 & 구현이 아닌 간편한 방법을 원해요!

엔드포인트 업데이트



엔드포인트 업데이트

모델Model 생성

```
aws sagemaker create-model
 --model-name model2
 --primary-container ‘{“Image”: “123.dkr.ecr.amazonaws.com/algo”,
                       “ModelDataUrl”: “s3://bkt/model2.tar.gz”}
 --execution-role-arn arn:aws:iam::123:role/me
```

엔드포인트 구성
EndpointConfig 생성

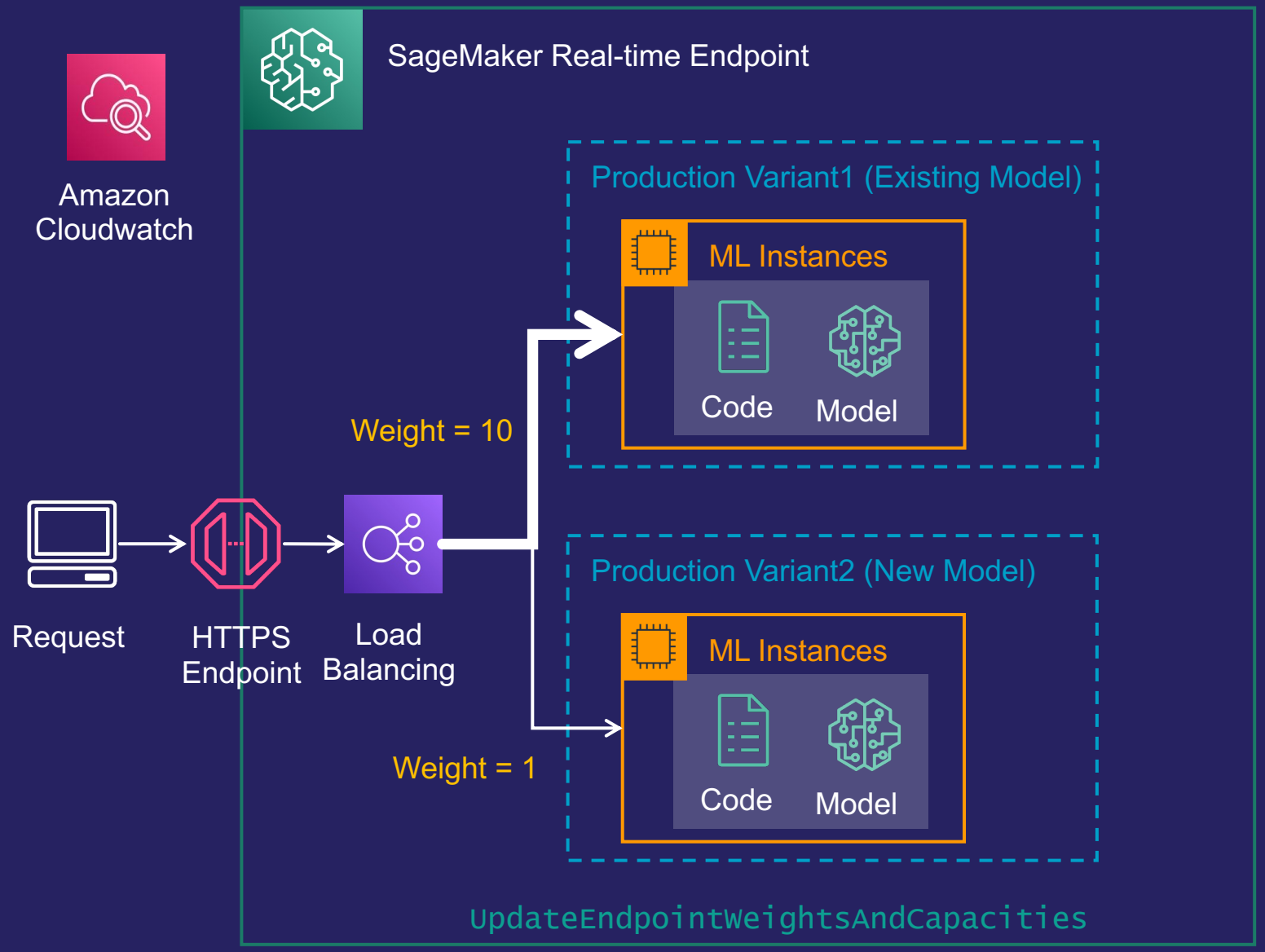
```
aws sagemaker create-endpoint-config
  --endpoint-config-name model2-config
  --production-variants ‘{“InitialInstanceCount”: 2,
                          “InstanceType”: “ml.m4.xlarge”,
                          ”InitialVariantWeight”: 1,
                          ”ModelName”: “model2”,
                          ”VariantName”: “AllTraffic”}’
```

엔드포인트
Endpoint 수정

```
aws sagemaker update-endpoint
  --endpoint-name my-endpoint
  --endpoint-config-name model2-config
```

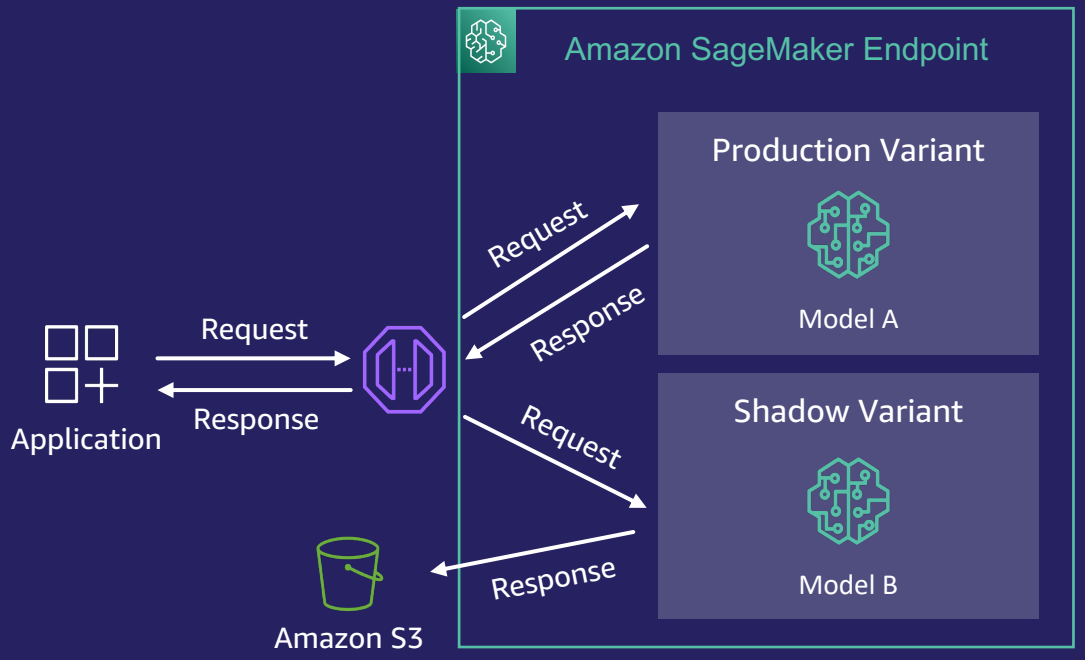
A/B 테스트

- 1~10개의 프로덕션 변형production variants
- 입출력 스키마가 동일해야 함
- 서비스 중단 없이 엔드포인트 수정



Shadow testing

실시간으로 모델을 비교하기 위한 SHADOW 테스트를 지원



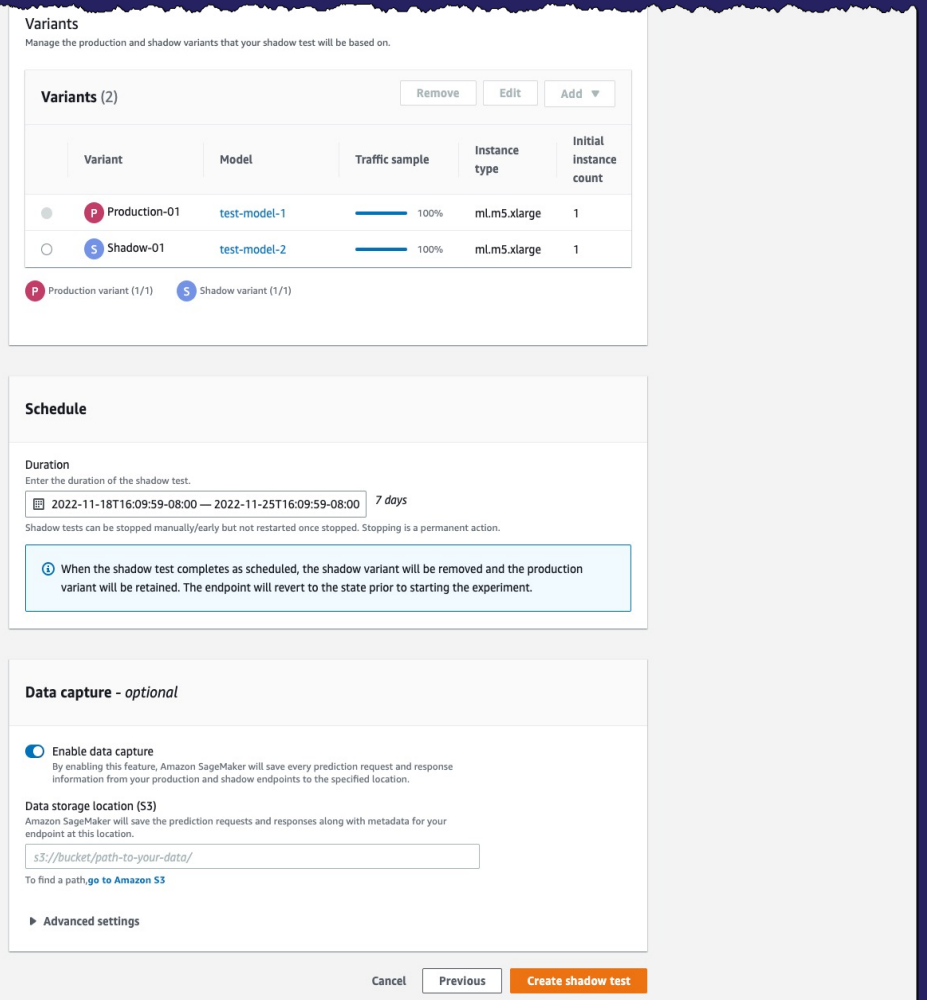
Accessible through AWS console, CLI, APIs

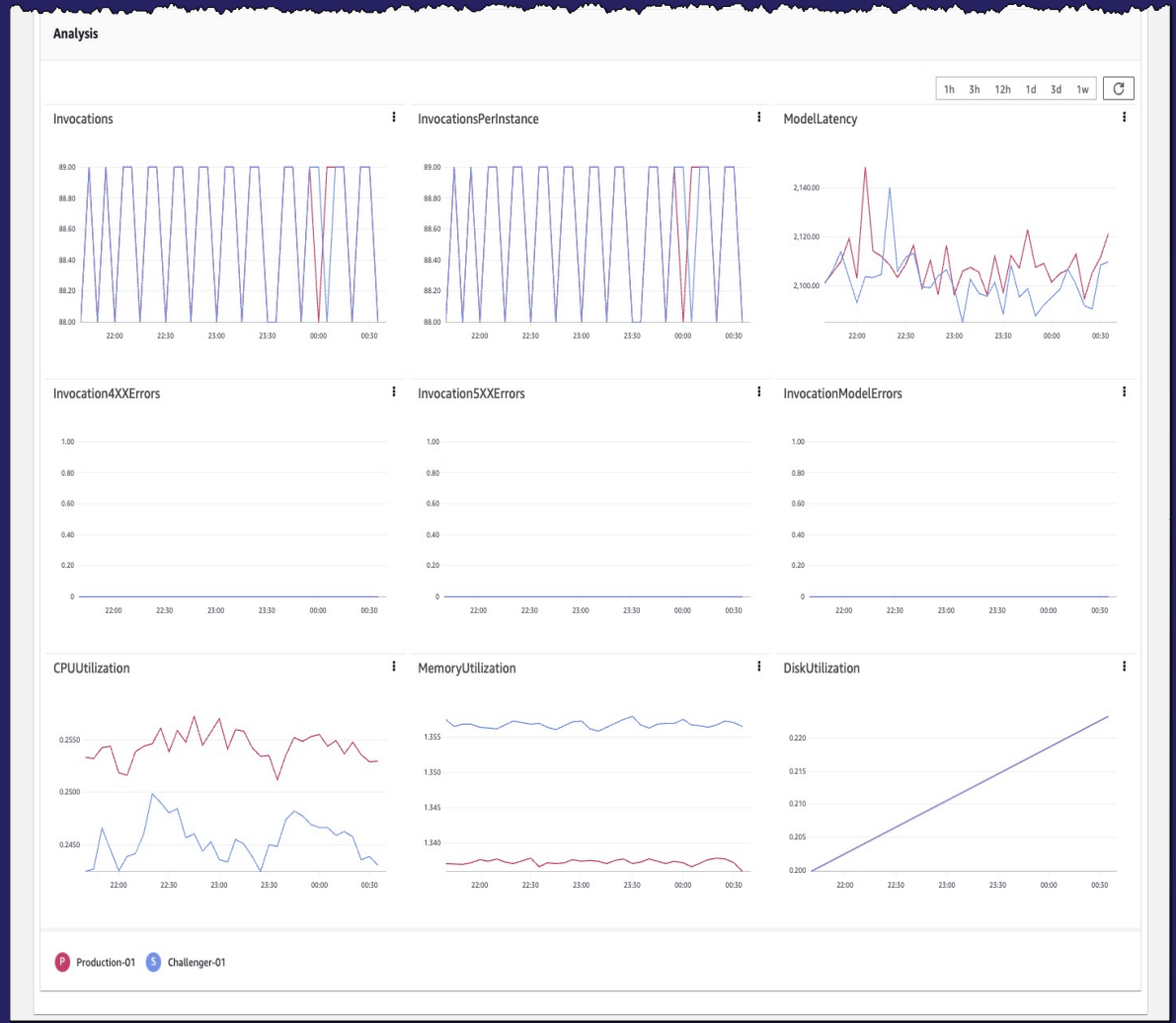
- 프로덕션 모델과 비교하여 모델의 성능 검증
- 세이지메이커가 미러링 요청을 처리
- 최종 사용자에게 영향을 미치기 전에 잠재적인 구성 오류 및 성능 문제 파악
- 라이브 대시보드를 통해 shadow 테스트 진행률 및 지연 시간 및 오류율과 같은 성능 메트릭 모니터링
- 프로덕션 모델에 대한 추론요청의 사본을 shadow 모델로 라우팅, 결과는 S3에 저장 (분석 용도)

Ref :
https://docs.aws.amazon.com/sagemaker/latest/dg/shadow-tests.html
https://aws.amazon.com/blogs/aws/new-for-amazon-sagemaker-perform-shadow-tests-to-compare-inference-performance-between-ml-model-variants/

Shadow testing

LATENCY와ERROR RATE 같은 성능 메트릭과 SHADOW TEST의 진행 모니터링

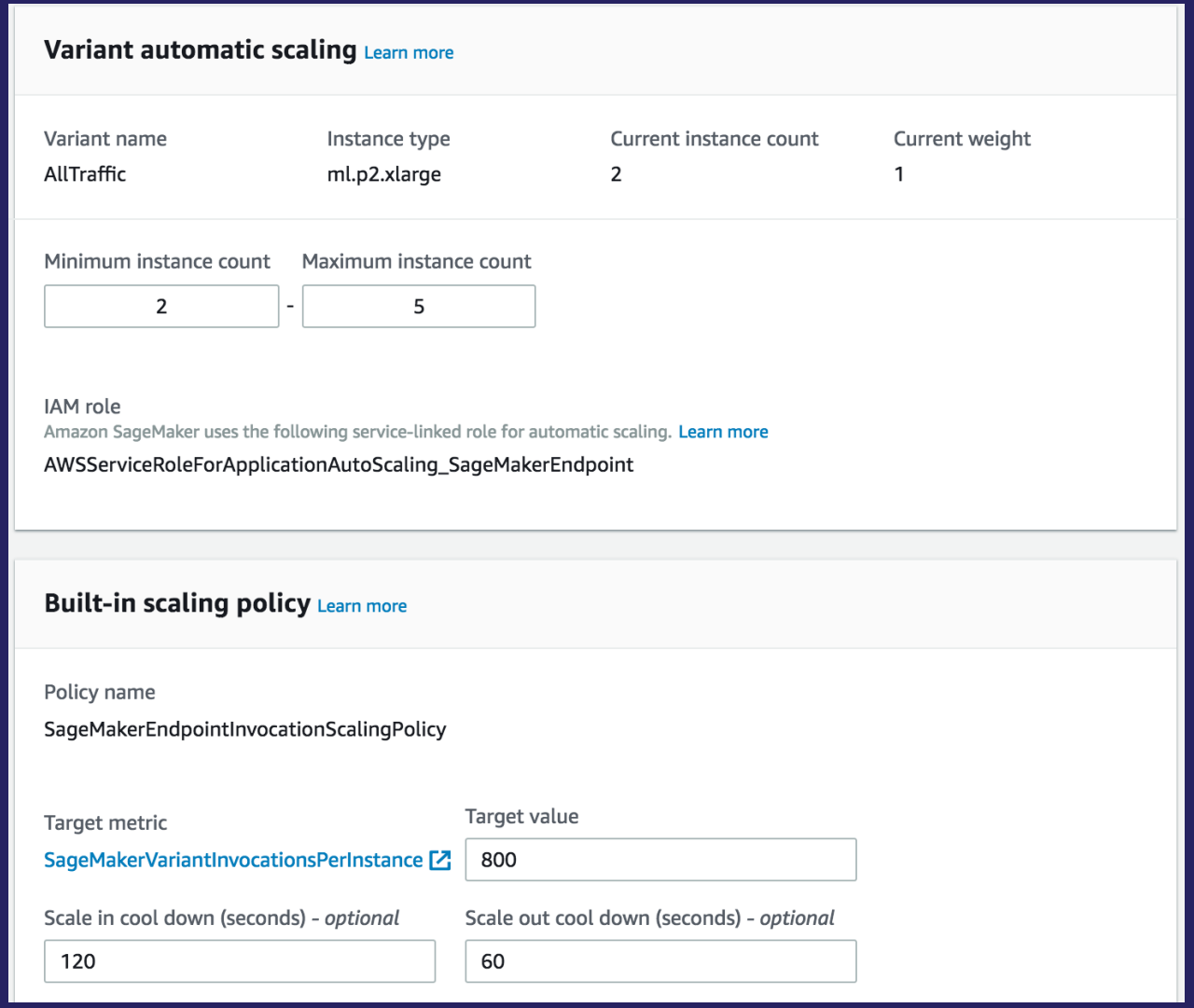




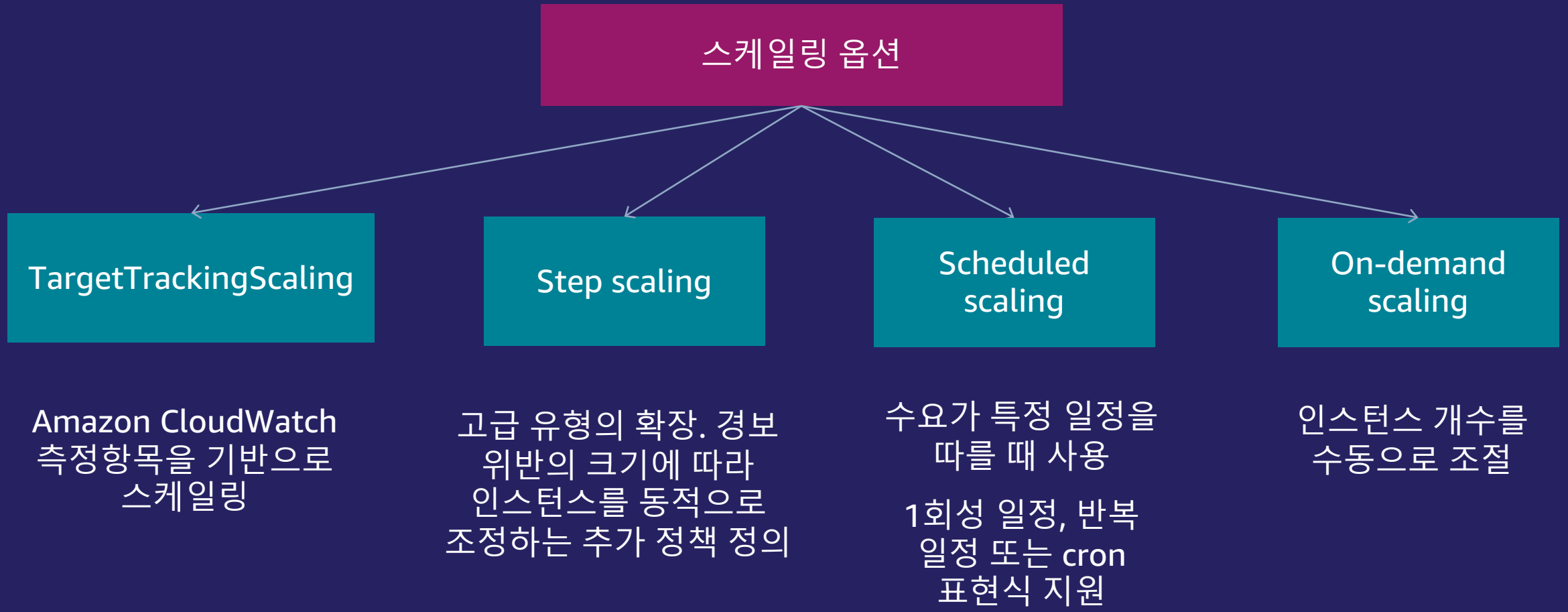
Ref :
https://docs.aws.amazon.com/sagemaker/latest/dg/shadow-tests.html
https://aws.amazon.com/blogs/aws/new-for-amazon-sagemaker-perform-shadow-tests-to-compare-inference-performance-between-ml-model-variants/

오토스케일링Auto-scaling 엔드포인트

- 엔드포인트 인스턴스의 Amazon CloudWatch 지표를 기반으로 자동 스케일링
  - Min and max instances
  - Target invocations per instance
  - Scaling cooldowns
- 빌트인 & 커스텀 스케일링 정책



오토스케일링Auto-scaling 엔드포인트



오토스케일링Auto-scaling 엔드포인트

오토스케일링 타겟 등록

```
aws application-autoscaling register-scalable-target
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --min-capacity 2
  --max-capacity 5
```

스케일링 정책 생성

```
aws application-autoscaling put-scaling-policy
  --policy-name model2-scaling
  --service-namespace sagemaker
  --resource-id endpoint/my-endpoint/variant/model2
  --scalable-dimension sagemaker:variant:DesiredInstanceCount
  --policy-type TargetTrackingScaling
  --target-tracking-scaling-policy-configuration
    '{"TargetValue": 50,
      "CustomizedMetricSpecification":
        {"MetricName": "CPUUtilization",
         "Namespace": "/aws/sagemaker/Endpoints",
         "Dimensions":
           [{"Name": "EndpointName", "Value": "my-endpoint"},
            {"Name": "VariantName","Value": "model2"}],
        "Statistic": "Average",
        "Unit": "Percent"}}'
```

4가지 주요 패턴 요약

	리얼타임 추론	배치 추론	비동기 추론	서버리스 추론
GPU 지원	O	O	O	X
오토스케일링	O	N/A	O	O
Scale to Zero	X	N/A	O	O
멀티컨테이너	O	X	X	X
멀티모델	O	X	X	X
페이로드 크기	6MB		1GB	4MB
타임아웃	60초	N/A	15분	60초
블루그린 가드레일	O	X	X	1-step만 지원
PrivateLink 지원	O	O	O	X
AB 테스트 (다중 Production Variants)	O	X	X	X

Thank you!


aws