AWS Builders Korea Program 200

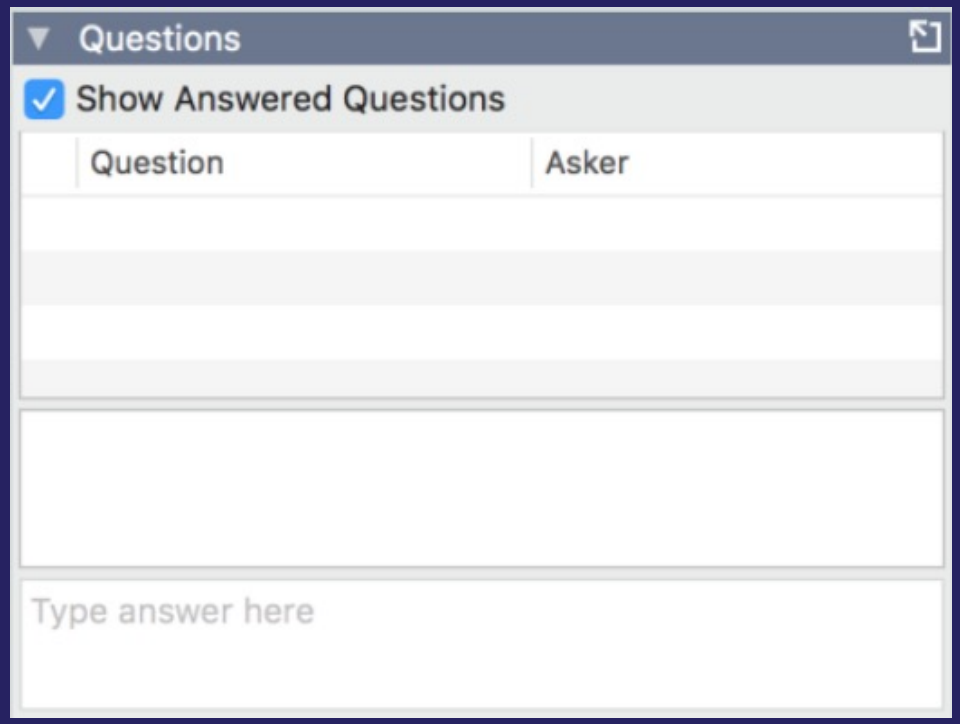
Amazon Elastic Container Service (ECS)

Sanghyun Kim

Solutions Architect

강연 중 질문하는 방법

- AWS Builders Go to Webinar “Questions” 창에 자신이 질문한 내역이 표시됩니다. 기본적으로 모든 질문은 공개로 답변 됩니다만 본인만 답변을 받고 싶으면 (비공개)라고 하고 질문해 주시면 됩니다.



고지 사항 (Disclaimer)

본 컨텐츠는 고객의 편의를 위해 AWS 서비스 설명을 위해 온라인 세미나용으로 별도로 제작, 제공된 것입니다. 만약 AWS 사이트와 컨텐츠 상에서 차이나 불일치가 있을 경우, AWS 사이트(aws.amazon.com)가 우선합니다. 또한 AWS 사이트 상 에서 한글 번역문과 영어 원문에 차이나 불일치가 있을 경우(번역의 지체로 인한 경우 등 포함), 영어 원문이 우선합니다.

실습 시작 전 준비 사항

AWS 계정으로 시작

1. 실습 전 계정을 꼭 신청해주세요 : https://portal.aws.amazon.com/billing/signup#/start

2. AWS 계정이 없으신 경우, 행사 참여 전에 미리 AWS 계정 생성 가이드를 확인하시고 AWS 계정을 생성해 주시길 바랍니다.

   - *AWS 계정 생성 가이드: https://aws.amazon.com/ko/premiumsupport/knowledge-center/create-and-activate-aws-account/

3. 웨비나 종료 후 설문조사에 참여해주신 분들께는 실습 비용 지원을 위한 AWS 크레딧(1인당 $50 크레딧, 전체 세션당 1회 제공)을 추가로 지원드립니다. 해당 AWS 크레딧은 등록하신 이메일 계정으로 00월 중 발송 드릴 예정입니다.

4. 검증된 호환성을 위하여 실습 시 사용할 웹 브라우저는
   Mozilla Firefox 또는 Google Chrome Browser로 진행 부탁드립니다.

실습 시작 전 준비 사항 (Credit)

- AWS 계정으로 시작하실 경우, **금일 실습에서 발생하는 비용은 당월 과금이 되는 점** 미리 확인 부탁 드립니다.
- 웨비나 종료 후 **설문 조사에 참여해주신 분들께는** 실습 비용 지원을 위한 **AWS 크레딧(1인당 $50 USD 크레딧, 전체 세션당 1회 제공)을 추가**로 지원드립니다.
- 해당 **AWS 크레딧**은 등록하신 이메일 계정으로 **1개월 내** 발송 드릴 예정이며, 전달 받은 AWS 크레딧은 바로 사용 가능합니다.

실습 마무리 및 설문 참여 방법

- 실습이 모두 끝난 후에는 **자원 삭제**를 잊지 마세요. 직접 준비하신 AWS 계정으로 실습을 진행하신 고객 분들의 경우, 가이드에 따라 자원 삭제를 진행하셔야 합니다. 또한, 기존에 사용하시던 자원이 있으신 고객 분들의 경우, **오늘 생성한 자원만 삭제**하는 것에 주의 부탁드립니다.

- **가이드:** (세션별 제공)

- 마지막으로 세션이 끝난 후, **GoToWebinar 창을 종료하면 설문 조사 창**이 나옵니다.
  이때, **설문 조사를 진행해 주셔야 AWS 크레딧**(1인당 $50 크레딧, 전체 세션당 1회 제공) 을 제공받으실 수 있습니다.

AWS는 고객 피드백을 기반으로 의사 결정을 수행하며 이러한 피드백은 추후에 진행할 세션 방향을 결정합니다.
더 나은 세션을 위하여 여러분들의 소중한 의견을 부탁드립니다.

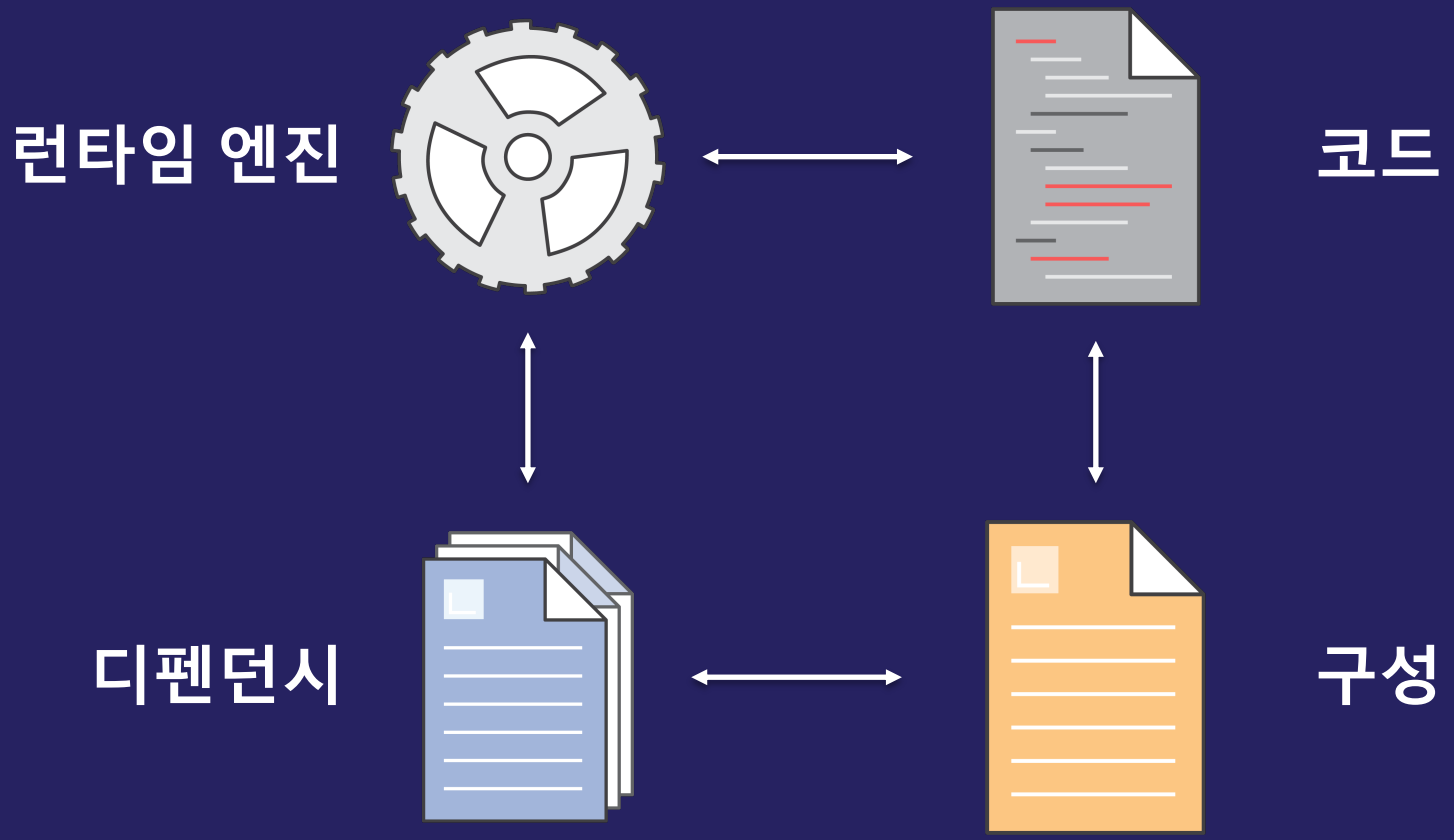
감사합니다.

더 나은 세미나를 위해
여러분의 의견을 남겨주세요!

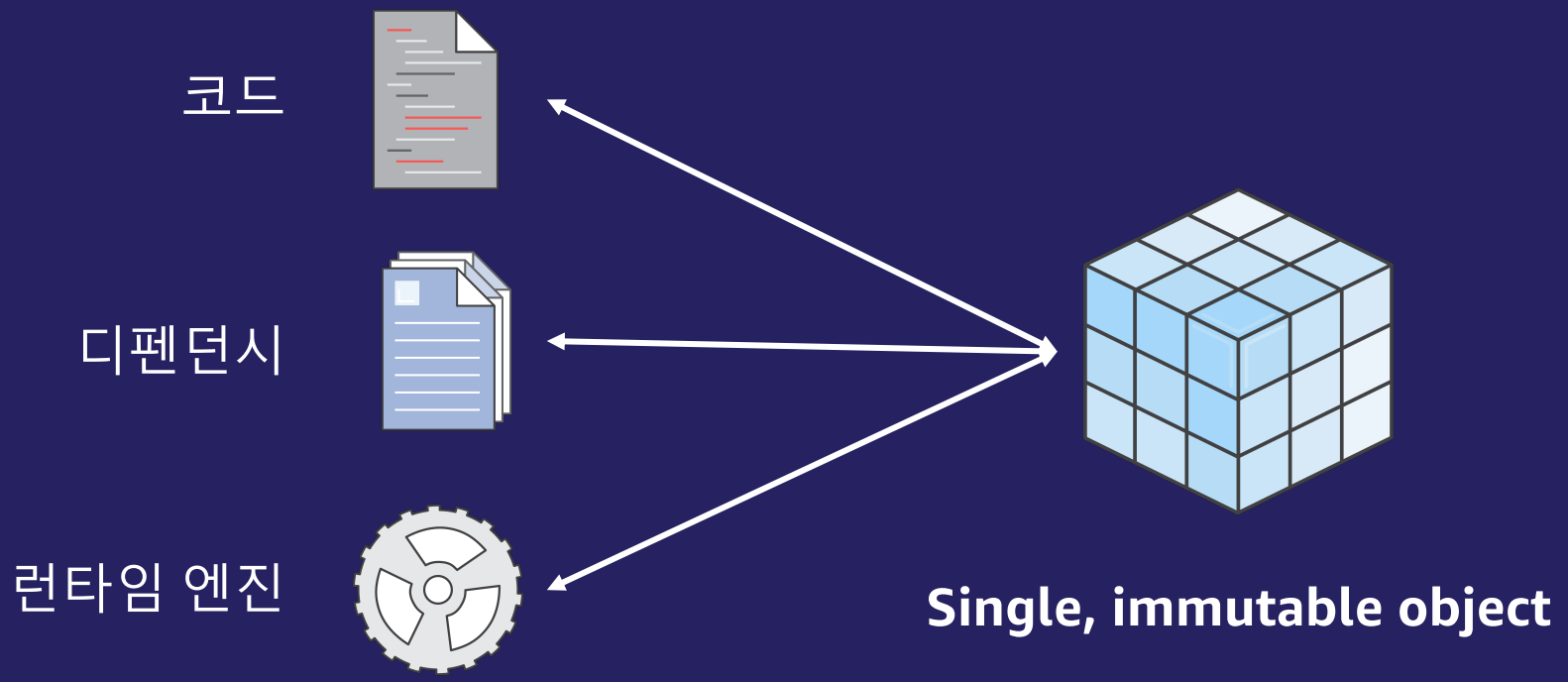
- 질문에 대한 답변 드립니다.

Container Overview

애플리케이션 주요 구성 요소

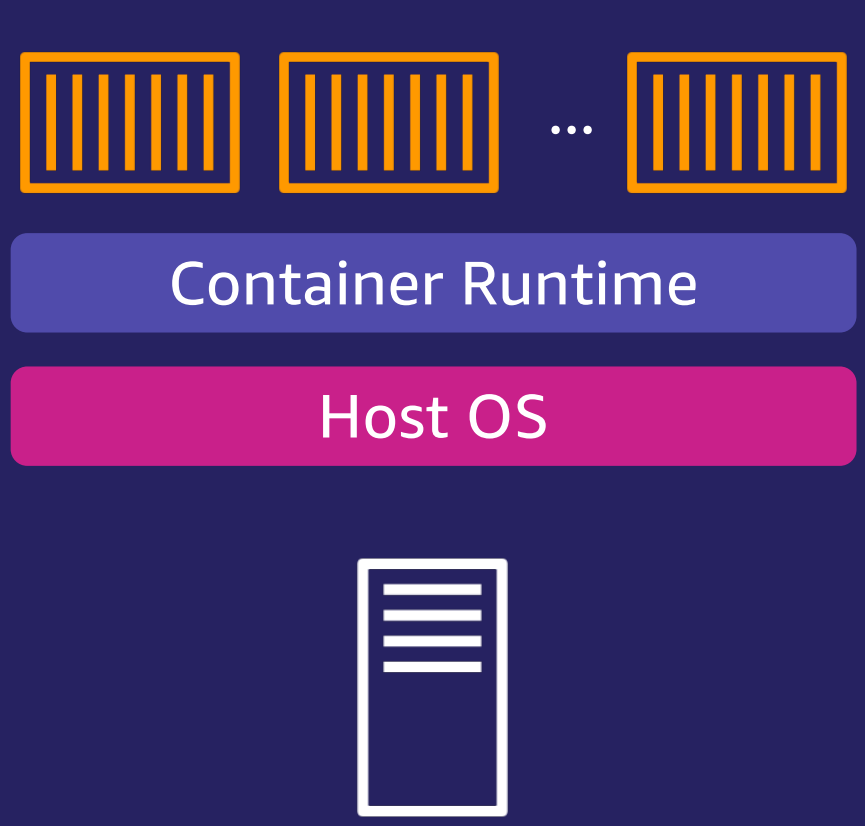


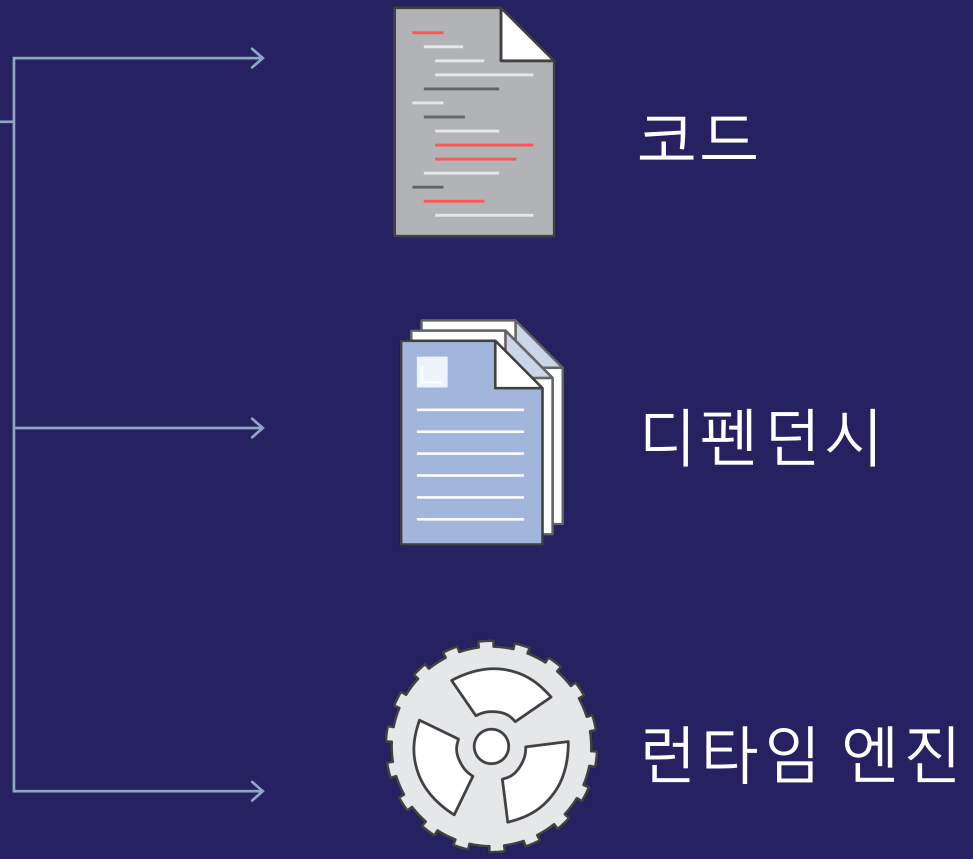
컨테이너와 컨테이너 이미지



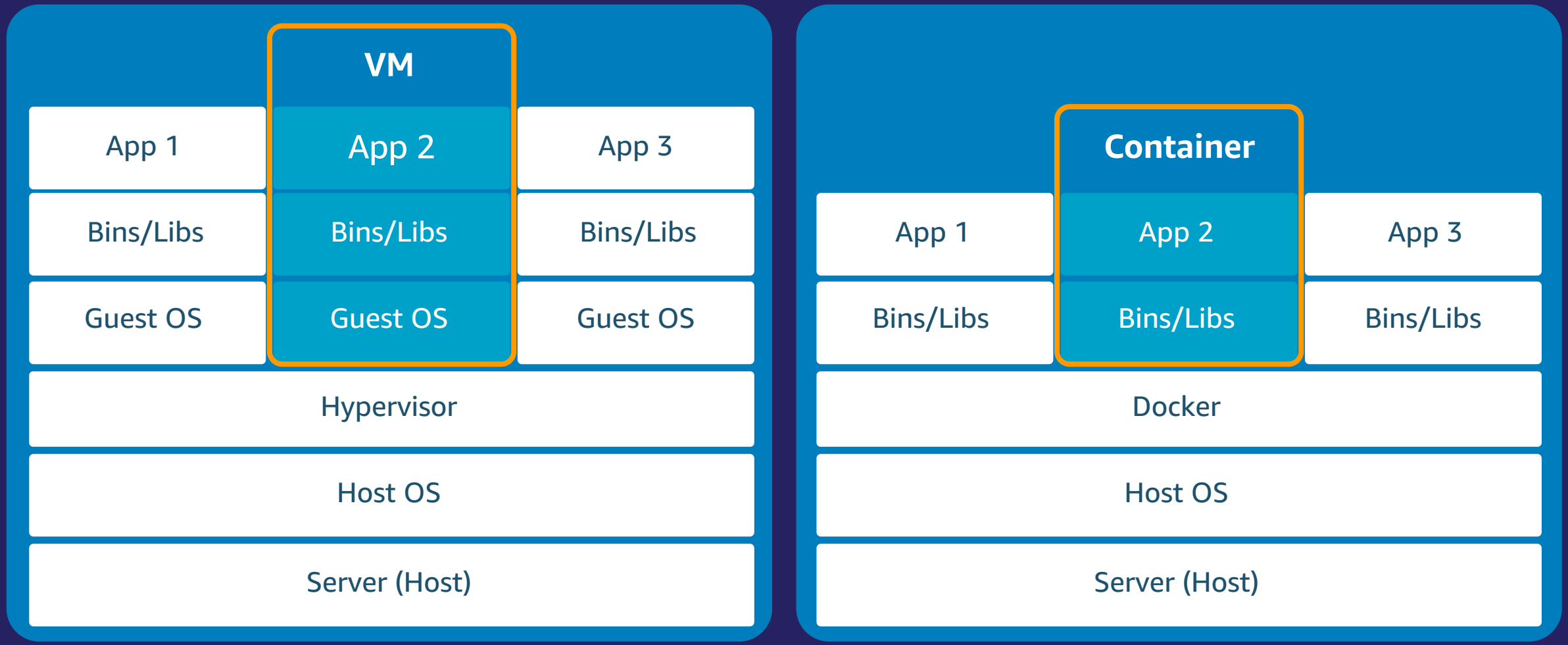
애플리케이션이 컴퓨팅 환경에서 실행되기 위해 필요한 코드와 모든 의존성을 패키징한 소프트웨어 단위

컨테이너

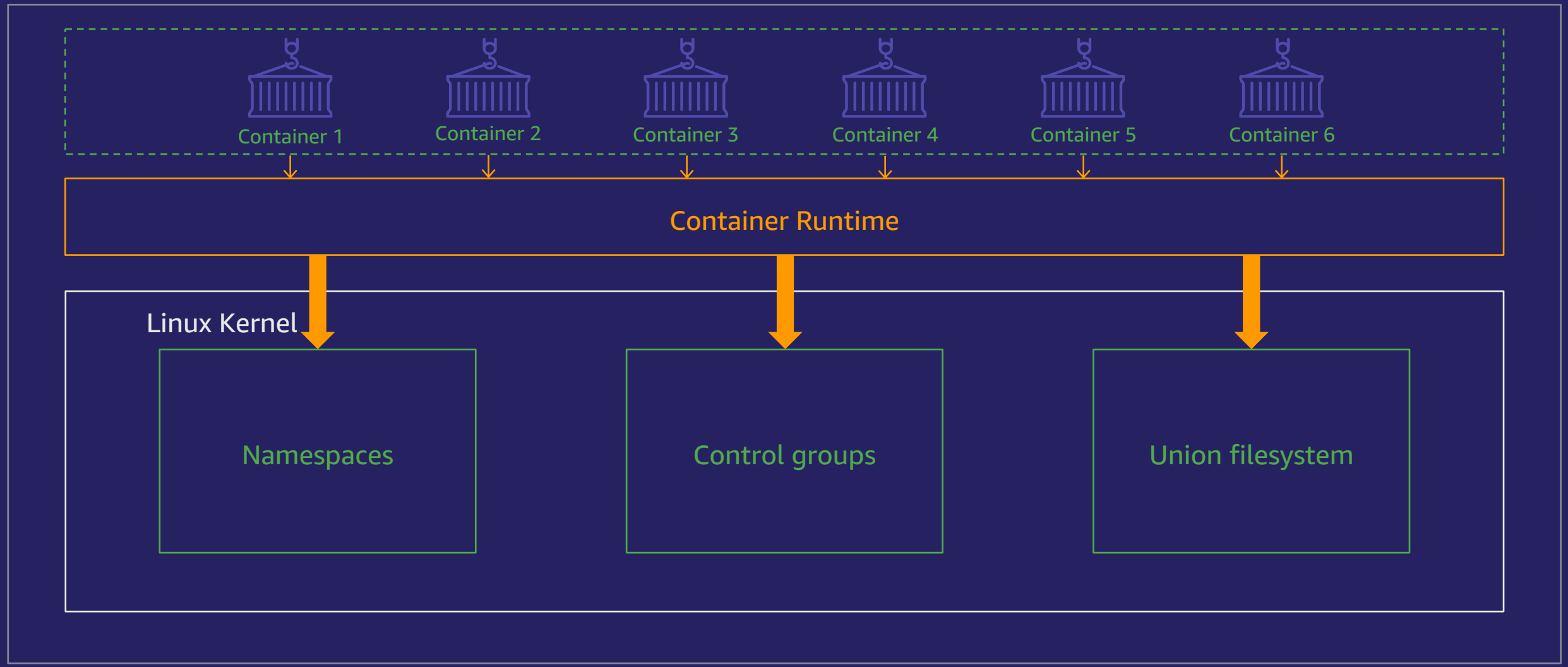




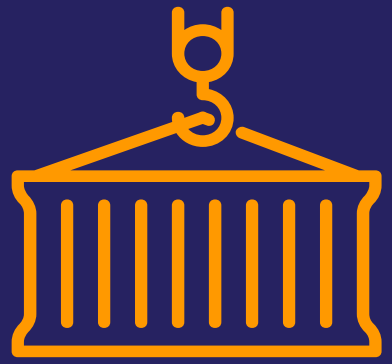
가상 머신과 컨테이너

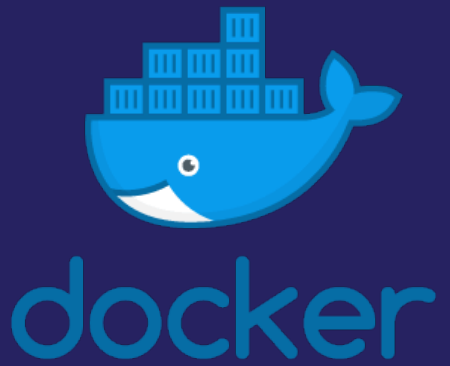


컨테이너 기반 기술



컨테이너 = docker?





Docker

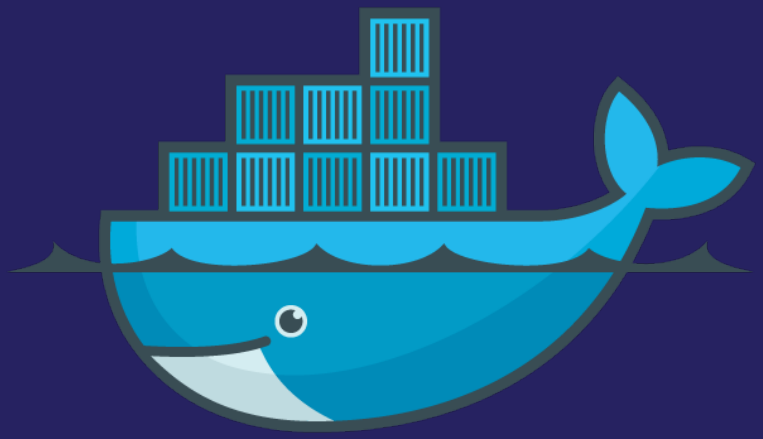
경량의 컨테이너 가상화 기술

애플리케이션을 배포하고 관리하는 도구

Apache 2.0 license

Built by Docker, Inc.

Moby: Open Source Project



Docker Image

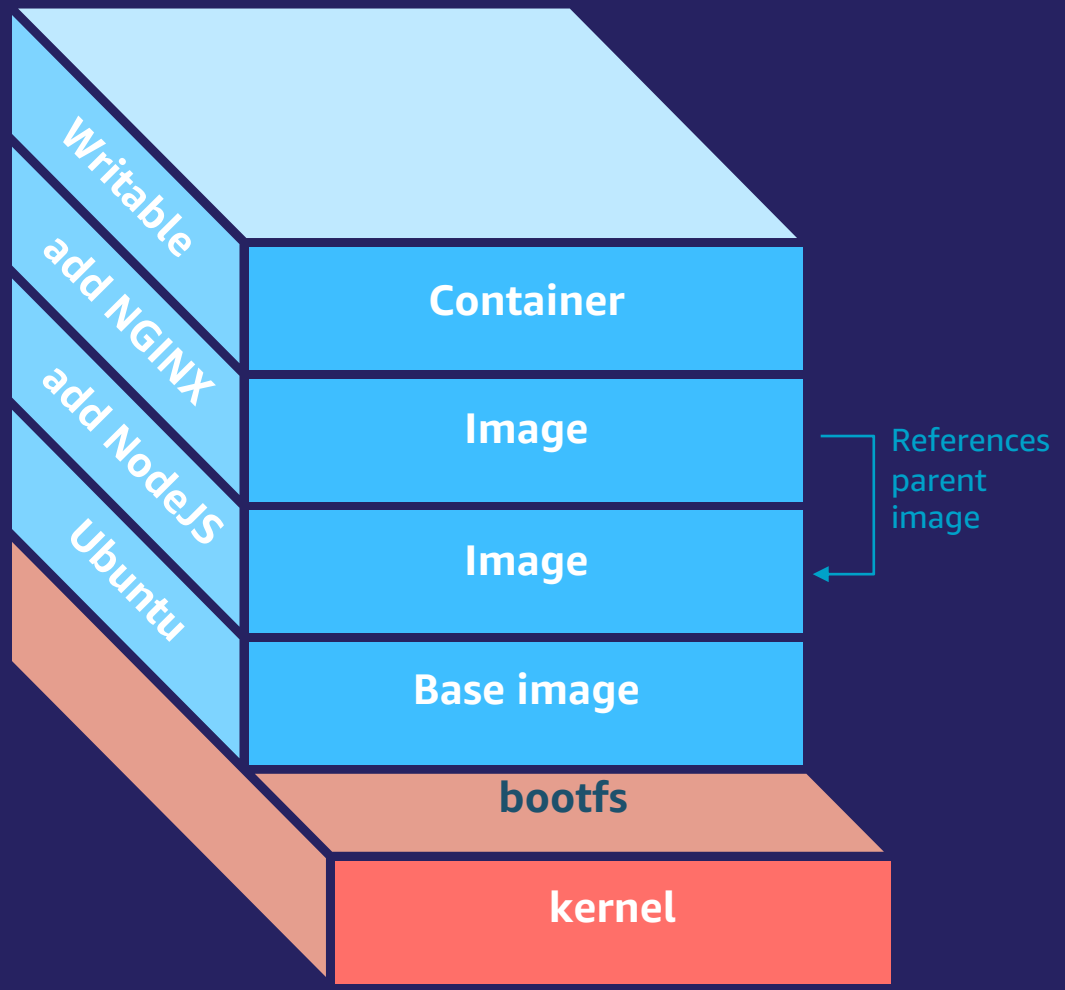
Read-only template

컨테이너를 구동하는데 사용됨

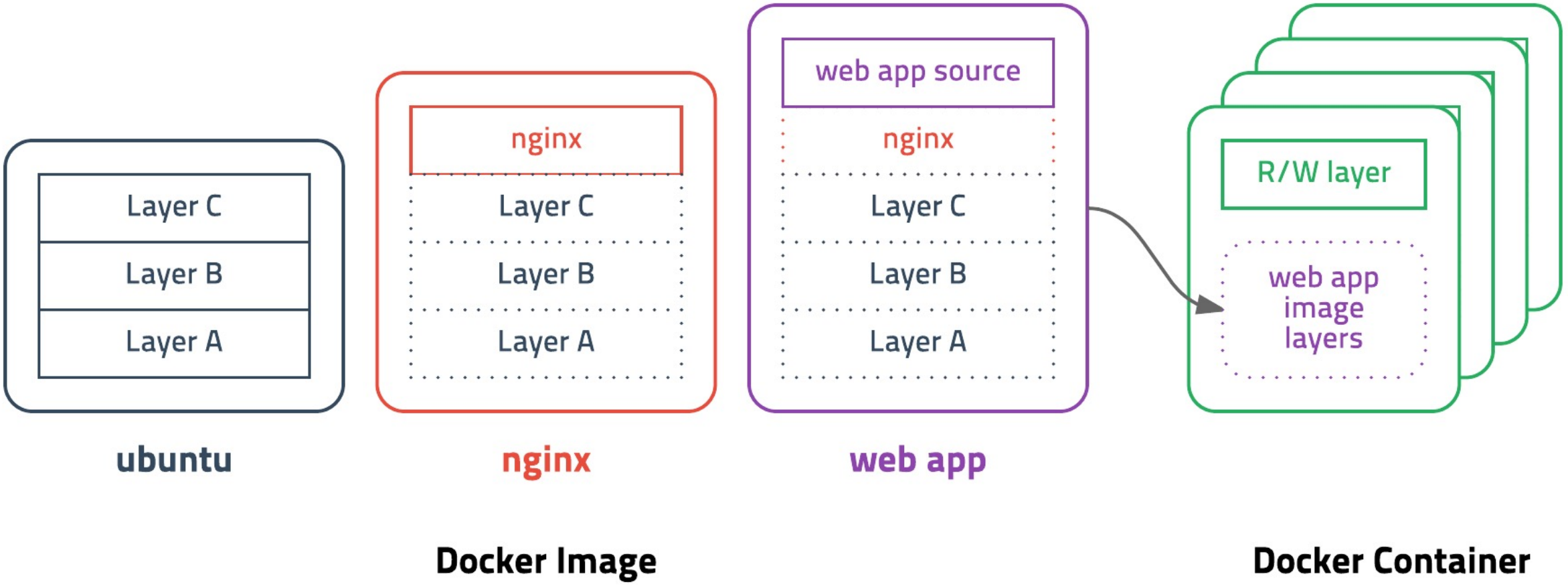
UnionFS(overylay2)를 이용,
다른 레이어들을 하나의 이미지로 합침

Base image위에 사용자 instruction
레이어들이 올라감

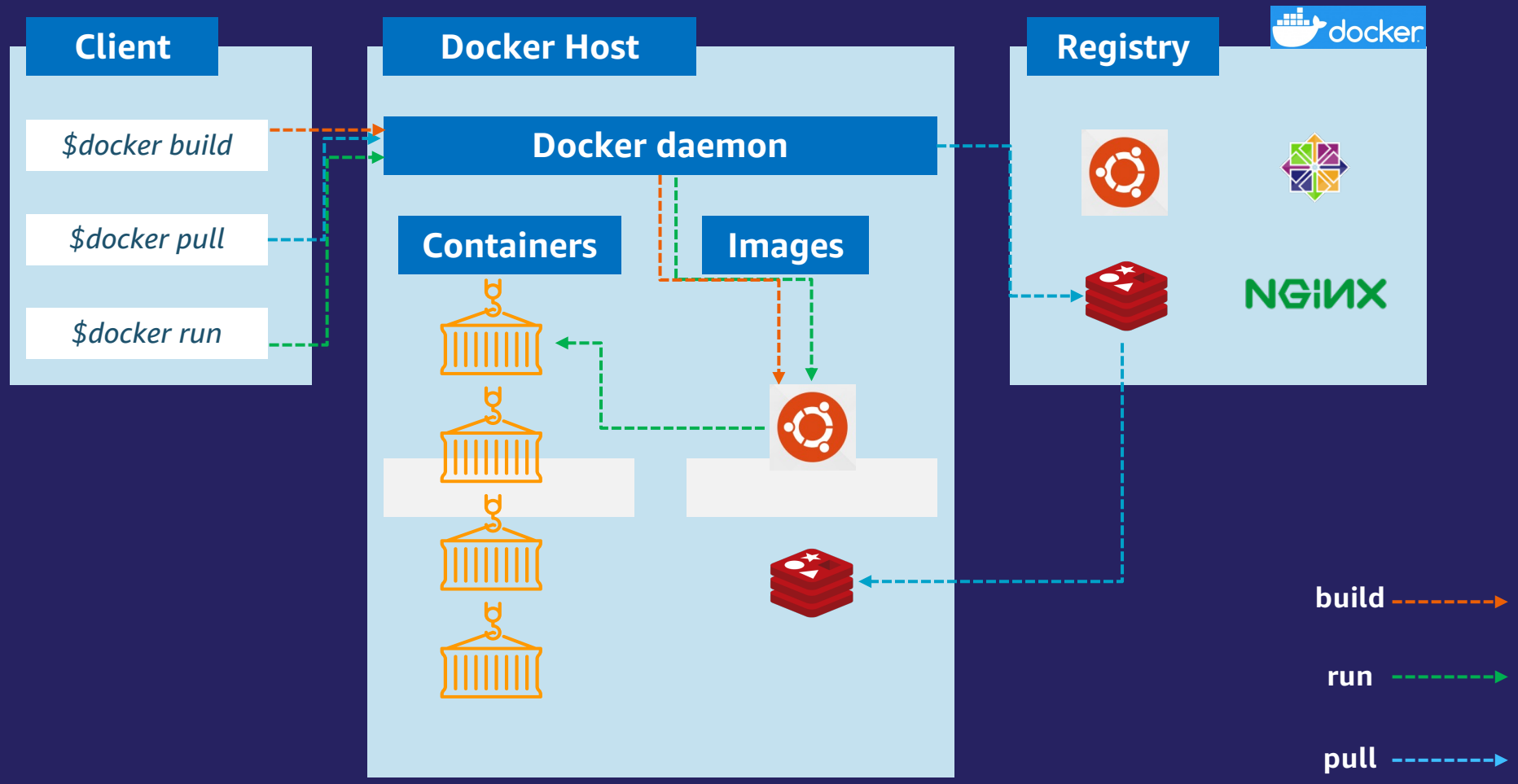
사용자 Instruction은 Dockerfile로 작성



Docker Image



Docker 구조



컨테이너 사용 사례

Applications

모바일, 웹 애플리케이션

백엔드 웹 서비스

IoT

데이터 프로세싱

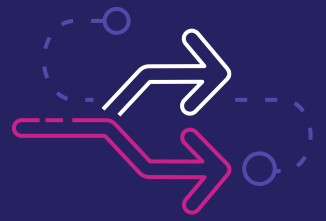
Shared services platform

CI/CD

IaaS

관리, 보안, 거버넌스

로깅, 모니터링



Enterprise app migration

.NET 윈도우 기반

리눅스 애플리케이션

3rd party 애플리케이션

Machine learning

추천 엔진

사기 탐지

챗봇

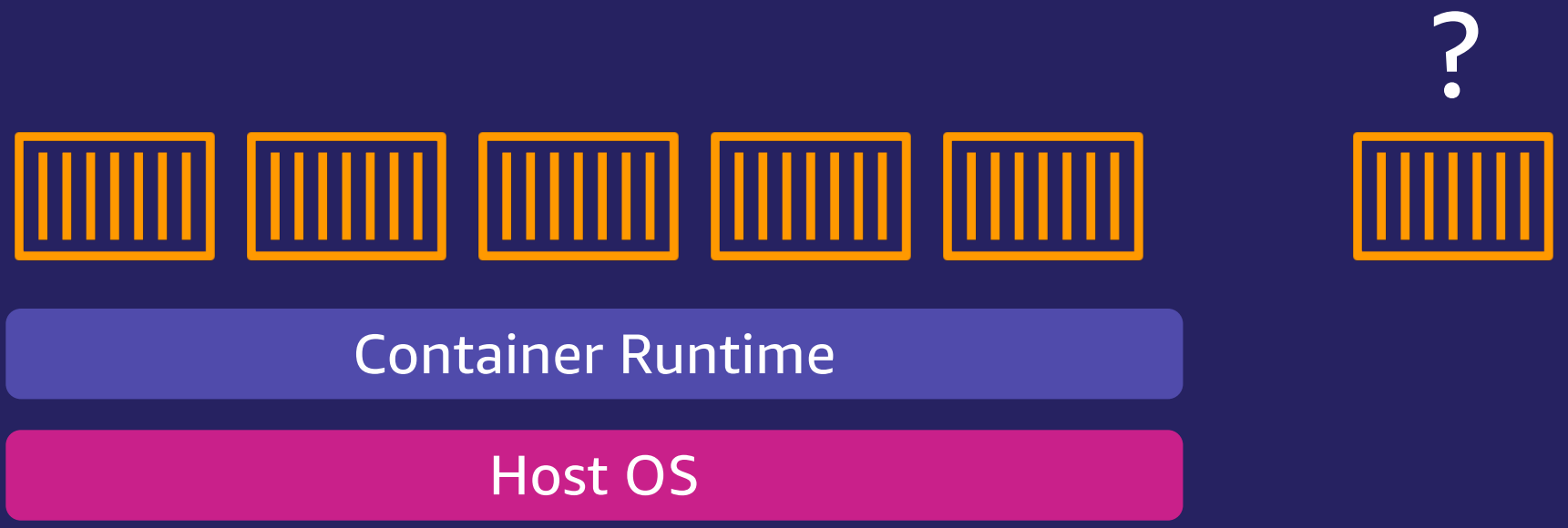
Container Ochestration

하나의 컨테이너를 올렸을 때

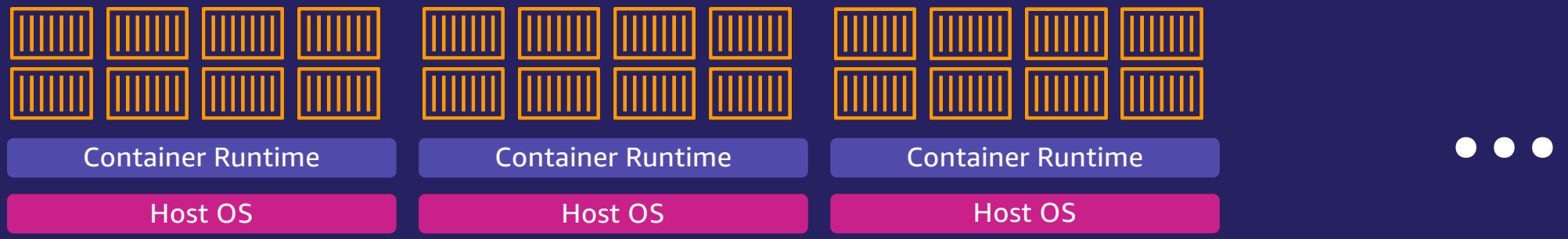
Container Runtime

Host OS

하나의 호스트 OS에 최대의 컨테이너를 올렸을 때



개발한 시스템이 프로덕션으로 간다면?



컨테이너 오케스트레이션 툴의 필요성

컨테이너를 호스트에 어떻게 배포하지?

컨테이너 내부/외부 통신은 어떻게 하지?

시크릿 관리는 어떻게 하지?

컴퓨팅 자원 풀을 최적화 할 수 있는 방법은 무엇일까?

컨테이너의 생명 주기를 어떻게 관리하지?

무중단, 블루/그린 배포는 어떻게 할 수 있을까?

컨테이너 오케스트레이션 도구의 구조

사용자가 원하는 상태(Desired State)로 동작하도록 관리(Schedule)하는 것

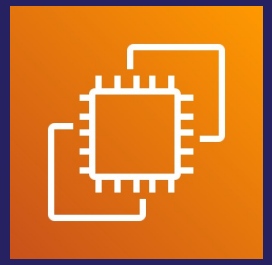
AWS 환경 위에서의 컨테이너

Management
컨테이너화된 애플리케이션을 배포, 스케줄링, 스케일링 및 관리

Amazon Elastic Container Service

Amazon Elastic Kubernetes Service

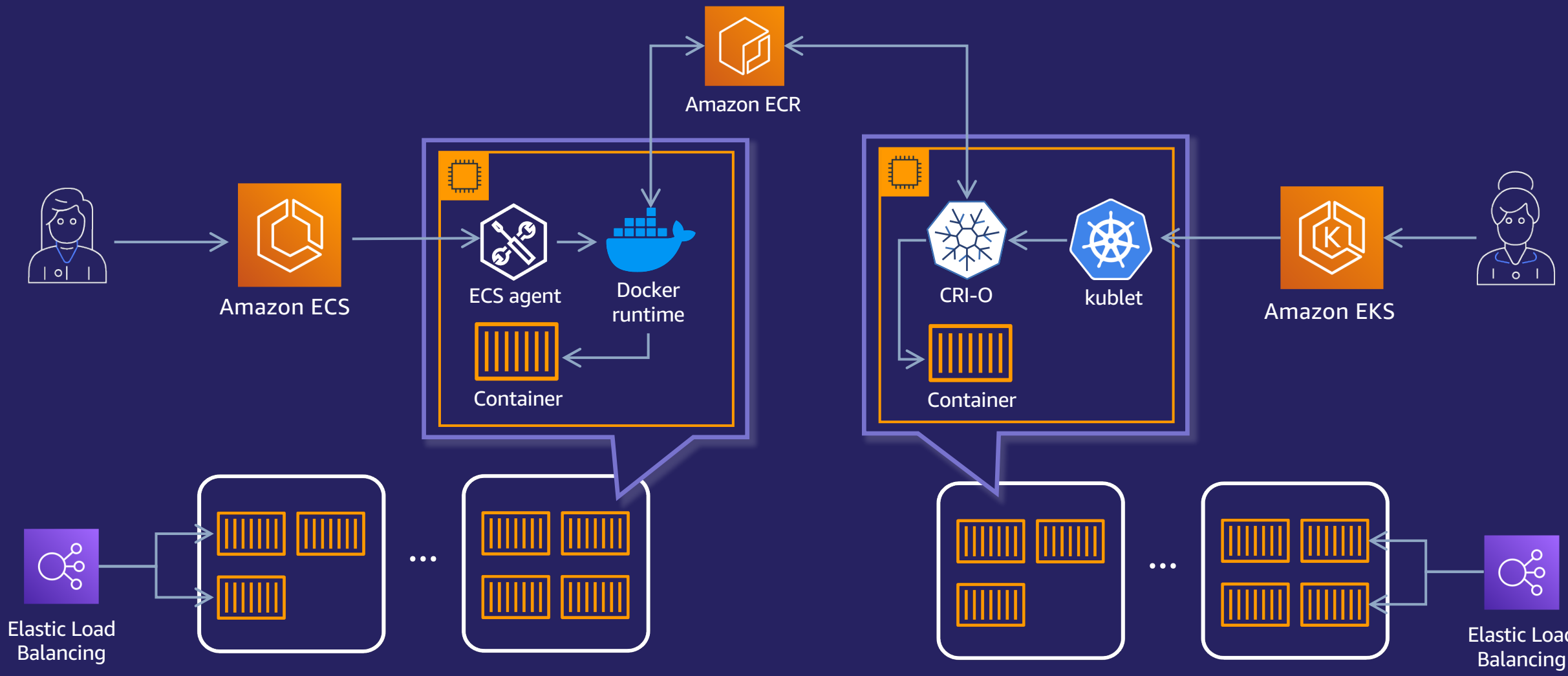
Hosting
컨테이너가 실행되는 곳



Amazon EC2

AWS Fargate

AWS 관리형 컨테이너 서비스의 구조



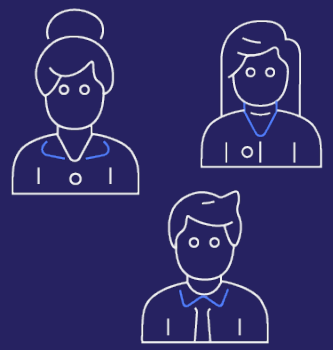
Amazon ECS

Amazon Elastic Container Service

Amazon Elastic Container Service

AWS 환경에서 컨테이너 애플리케이션을 쉽게 실행, 확장 및 보호할 수 있는 컨테이너 오케스트레이션 서비스

Amazon ECS 아키텍처



Administrators

AWS CLI, Cloudformation, or CDK

Amazon ECS

Amazon ECS Agent

EC2 Instances

Fargate Instances

Amazon ECS 구성 요소

ECS Task

컨테이너 최소 실행 단위 ≈ Pod

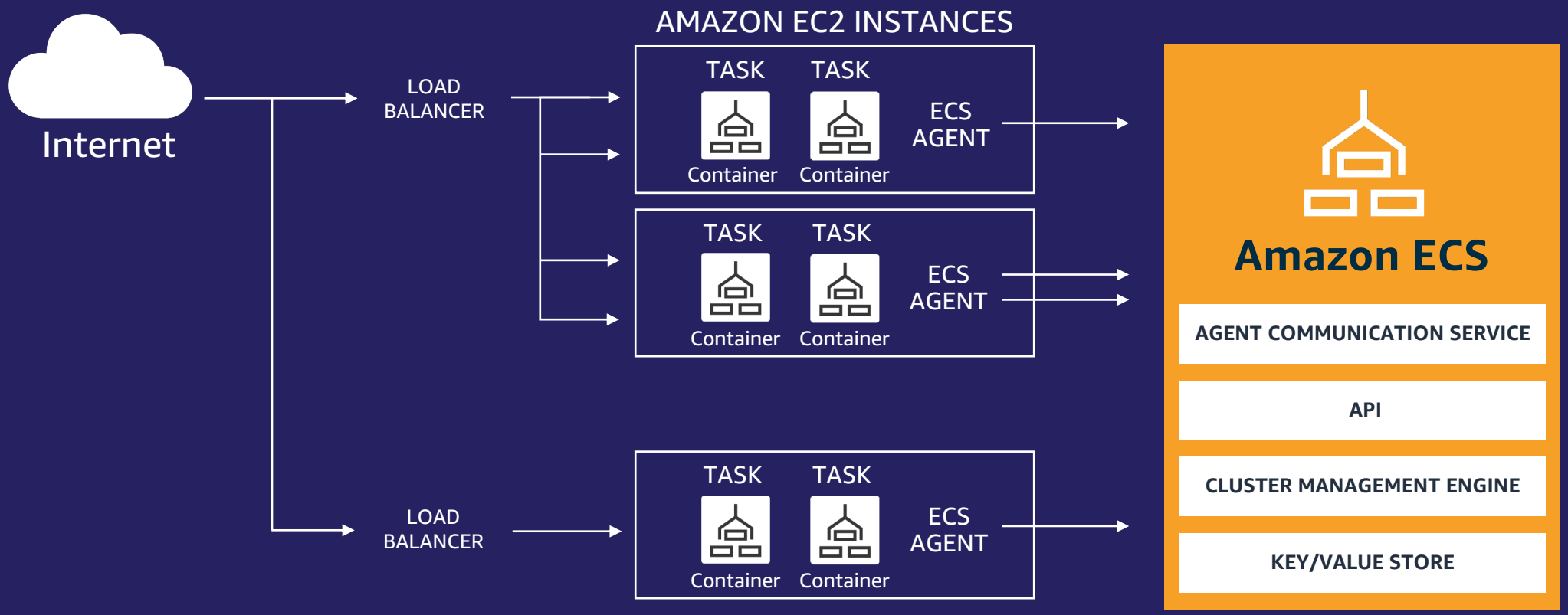
ECS Service

필요한 태스크 수를 유지
ELB와 연동되어 외부로 서비스 노출 ≈ Deployment or Service

ECS Cluster

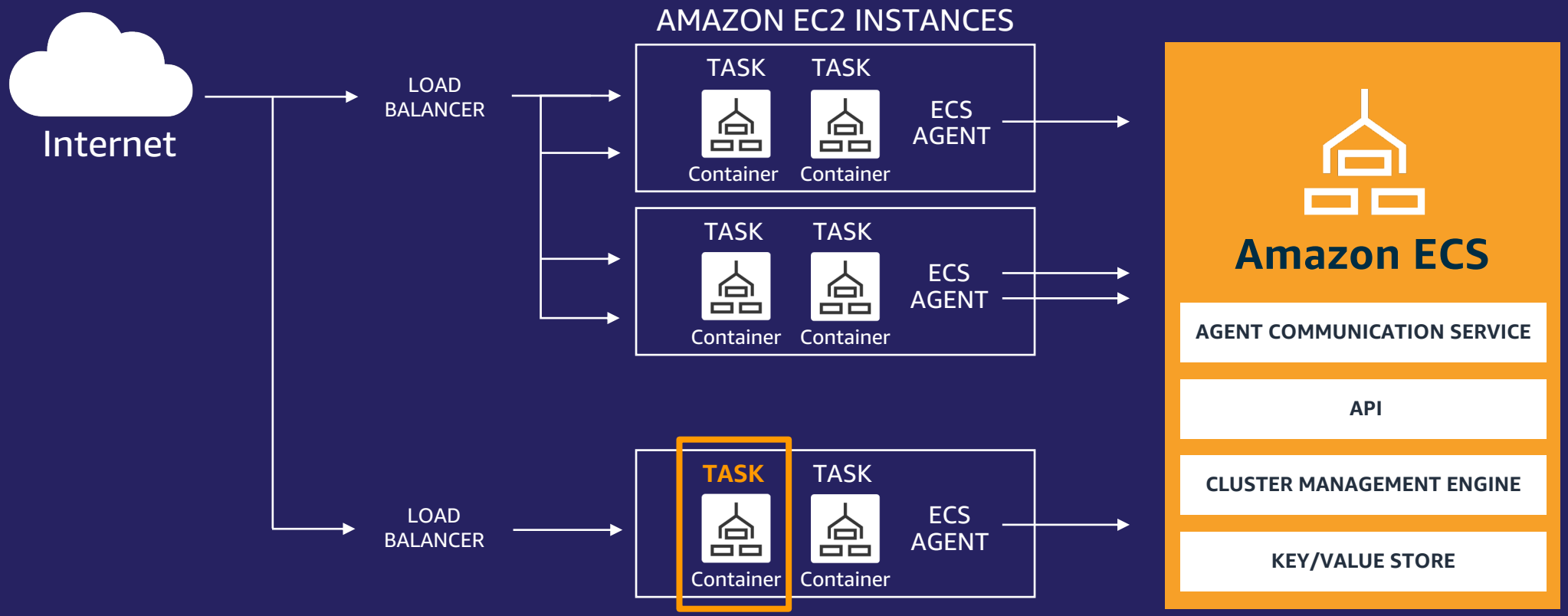
태스크가 실행되는 논리적 그룹 ≈ Cluster or Namespace

Amazon ECS

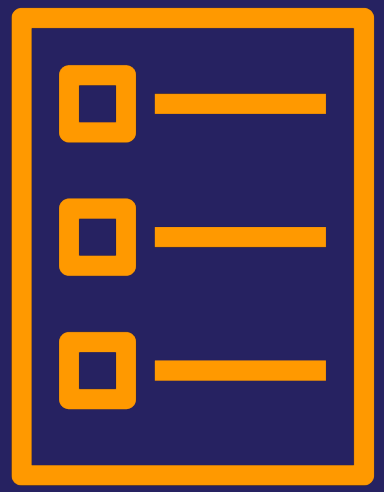


Amazon ECS – 작업 (Task)

- ECS에서 **최소 실행 단위**
- **작업 정의** 내용을 기반으로 ECS 클러스터에 속하는 **인스턴스** 및 **Fargate** 배포
- **한 작업** 당 최대 10개 컨테이너 가능, 모두 같은 호스트에 배치



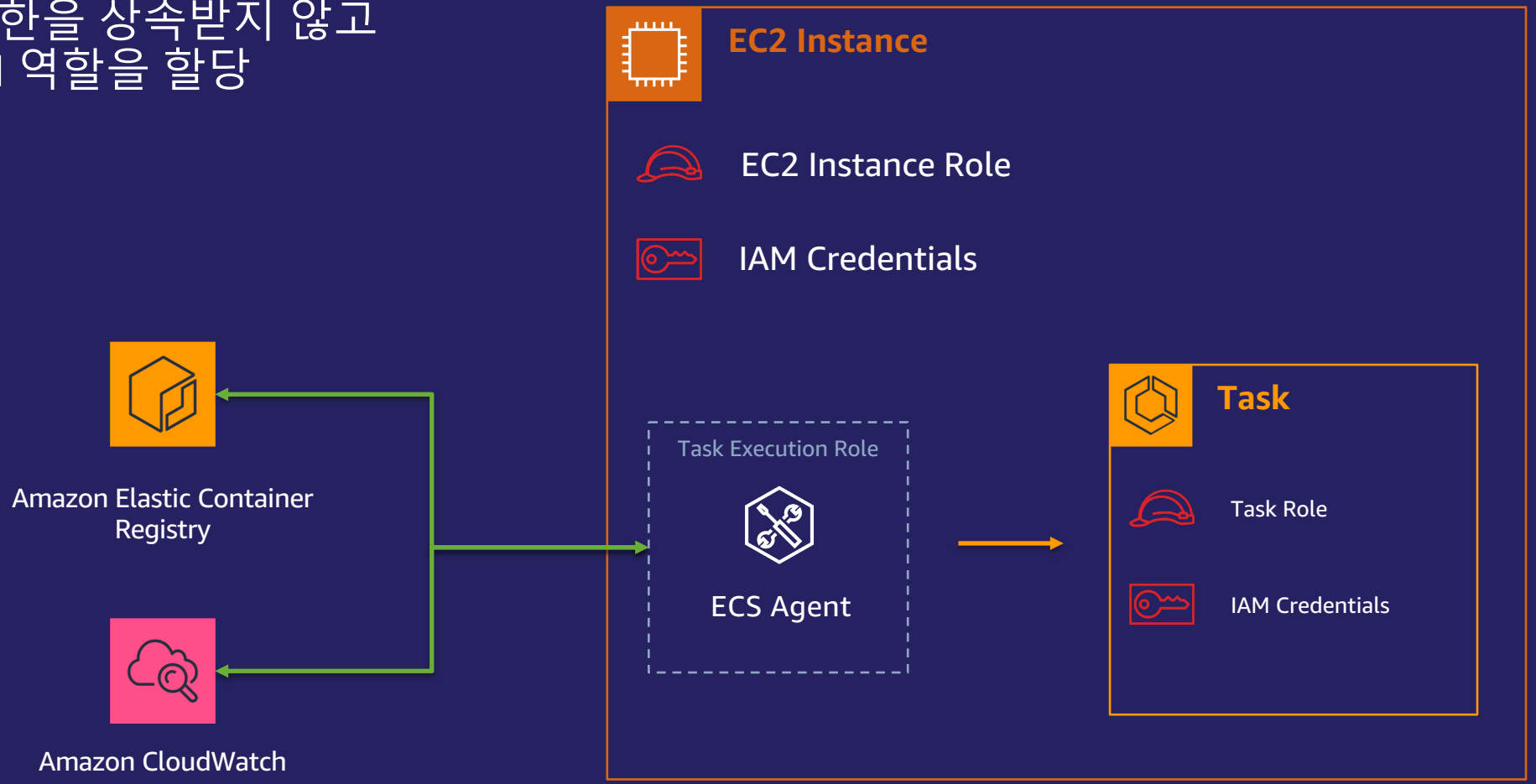
작업 정의 (Task Definition)



- ECS에서 컨테이너를 실행하려면 작업 정의가 필요
- 컨테이너 이미지 맵핑을 통한 컨테이너 정의
- 배포 타입 설정 가능: Fargate, EC2
- 작업 역할을 부여해 API 요청을 받을 때 권한에 따라 동작 가능
- 작업 크기 설정 가능
- App Mesh, FireLens 등과 통합 활성화 가능
- 다양한 볼륨 선택권 제공

ECS 작업에 부여하는 IAM 역할

호스트에 있는 권한을 상속받지 않고
작업(Task)에 IAM 역할을 할당



다른 서비스들과의 손쉬운 연동

```
"containerDefinitions":[
    {
      "name": "wordpress",
      "image": "wordpress:latest",
      "logConfiguration": {
          "logDriver": "awslogs",
          "options": {
            "awslogs-group": "awslogs-wordpress”,
            "awslogs-region":  " ap-northeast-2",
            "awslogs-stream-prefix": "awslogs-example”
          }
      }
    }
]
```

CloudWatch Logs

AWS App Mesh

Amazon S3

저장

Amazon Kinesis

스트리밍

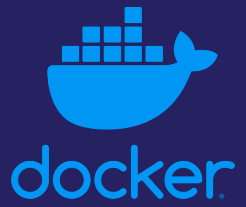
AWS Lambda

처리

Amazon OpenSearch

검색

다양한 볼륨 선택권



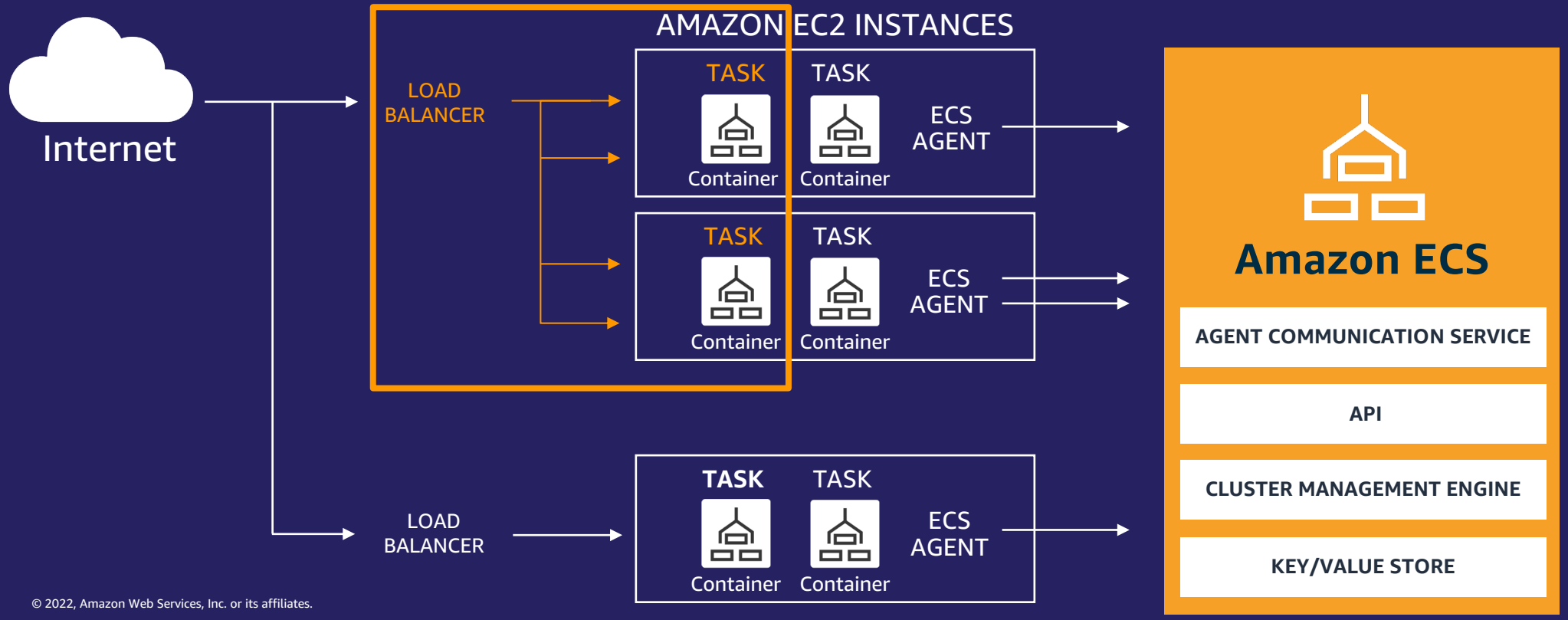
컨테이너 인스턴스 /var/lib/docker/volumes 위치에 도커 관리형 볼륨이 생성됨
도커 볼륨 드라이버(플러그인이라고도 함)들은 Amazon EBS와 같은 외부 스토리지 시스템과 통합하는데 사용됨

호스트 머신에 있는 파일 혹은 디렉토리를 컨테이너에 마운트
EC2 혹은 Fargate 런치 타입에 상관없이 호스트 볼륨 바인딩 지원

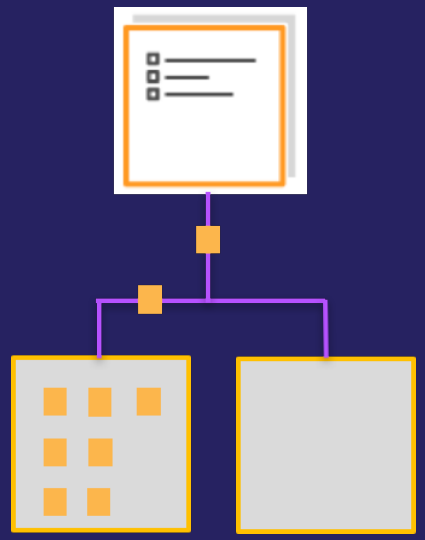
Amazon Elastic File System (EFS)를 컨테이너에 마운트

Amazon ECS – 서비스 (Service)

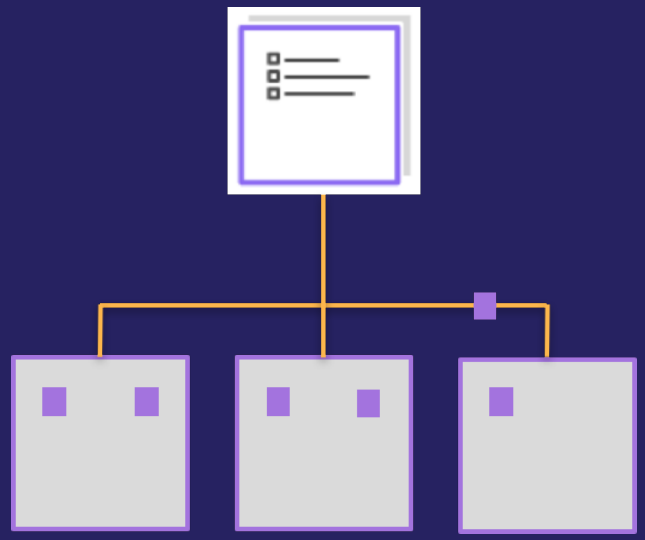
- 설정한 작업 수를 자동 유지 및 복구
- 서비스 유형(replica, daemon), 작업 배치 전략 선택 가능
- 배포 유형: 롤링 업데이트(ECS로 제어), 블루/그린 배포(CodeDeploy로 제어)
- ALB를 통해 효율적으로 부하 분산
- 서비스 Auto Scaling 적용 가능



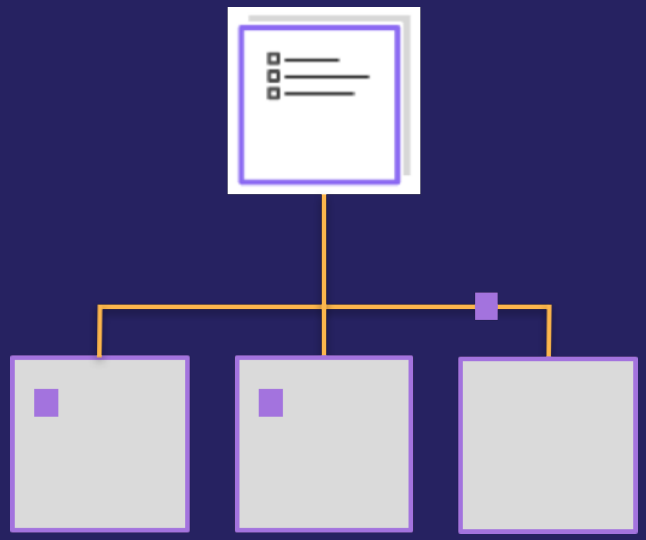
다양한 작업 배치 전략



Binpacking



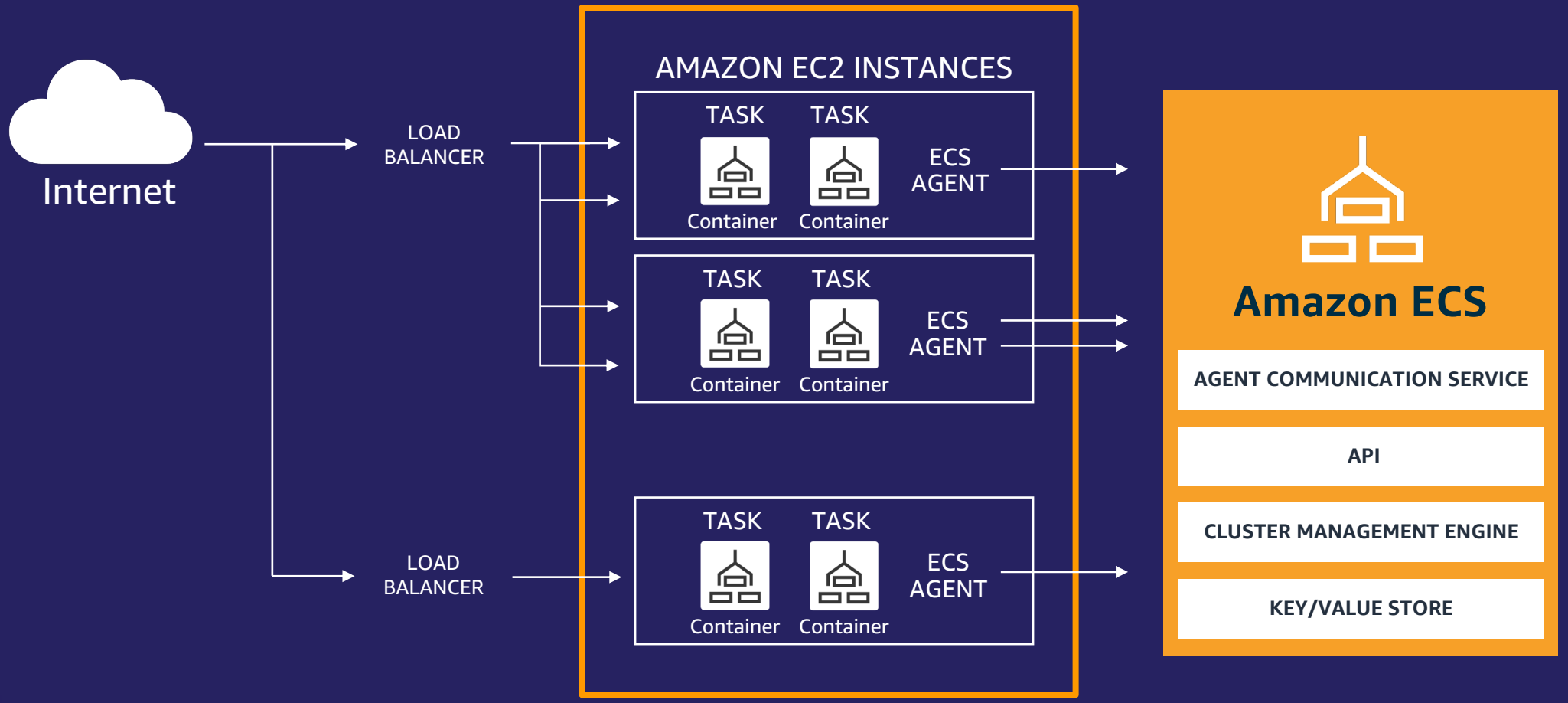
Spread



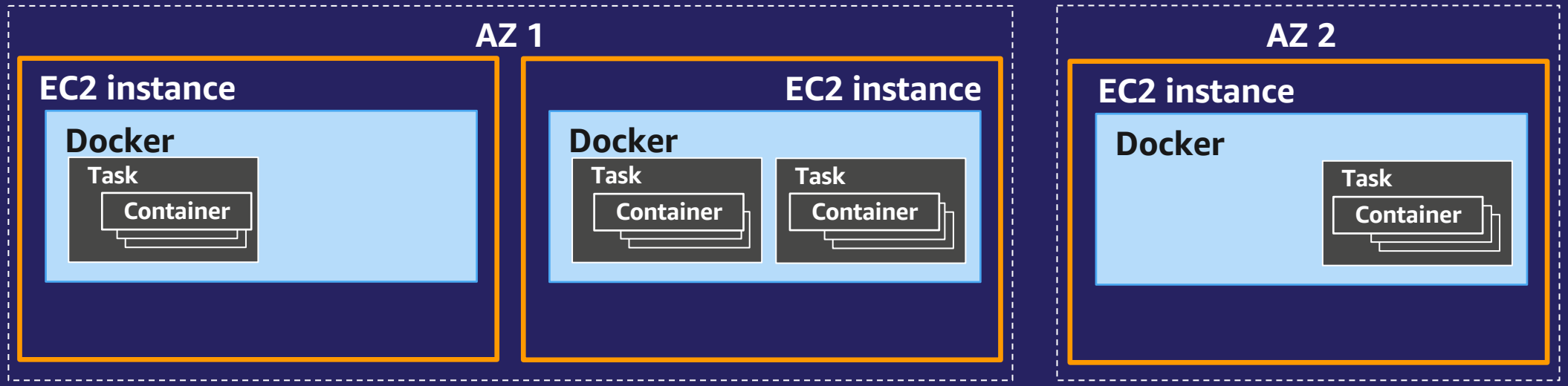
Random

Amazon ECS – 클러스터 (Clsuter)

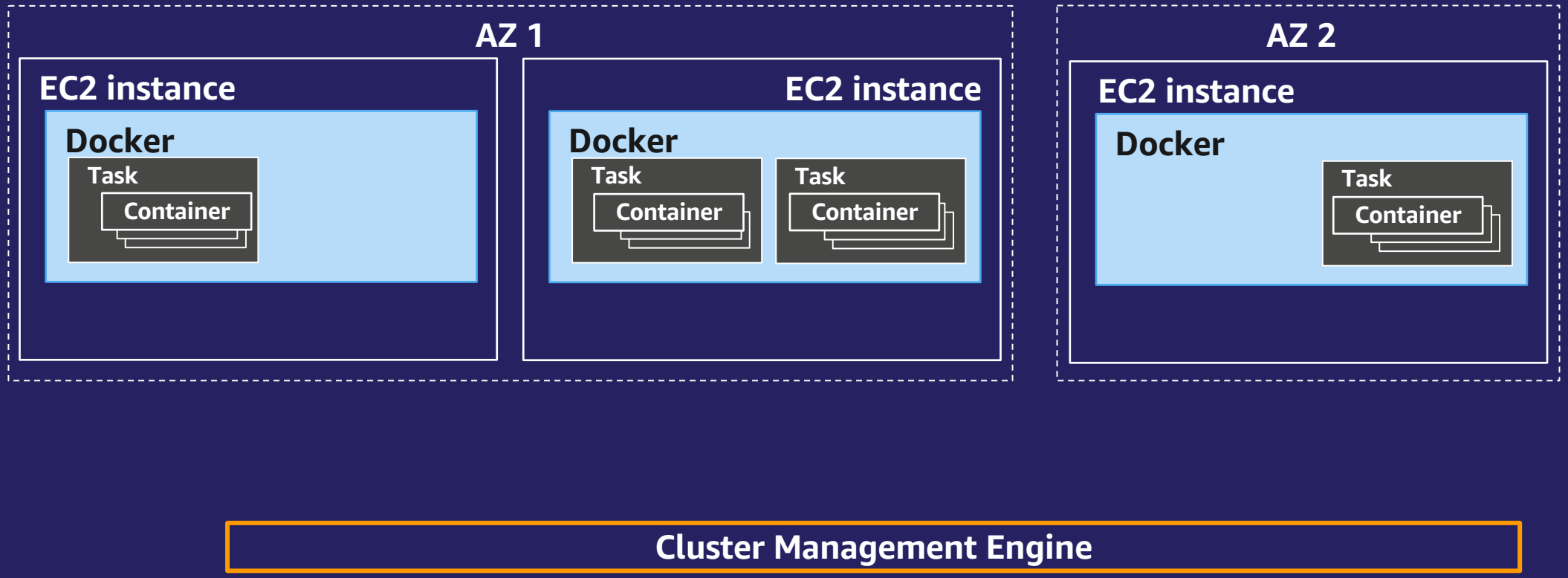
- 작업이 실행되는 **논리적인 그룹**
- 클러스터 생성 시, EC2 인스턴스 구축 혹은 생성 후, 클러스터에 추가 인스턴스 등록 가능
- 클러스터 템플릿을 선택하여 배포



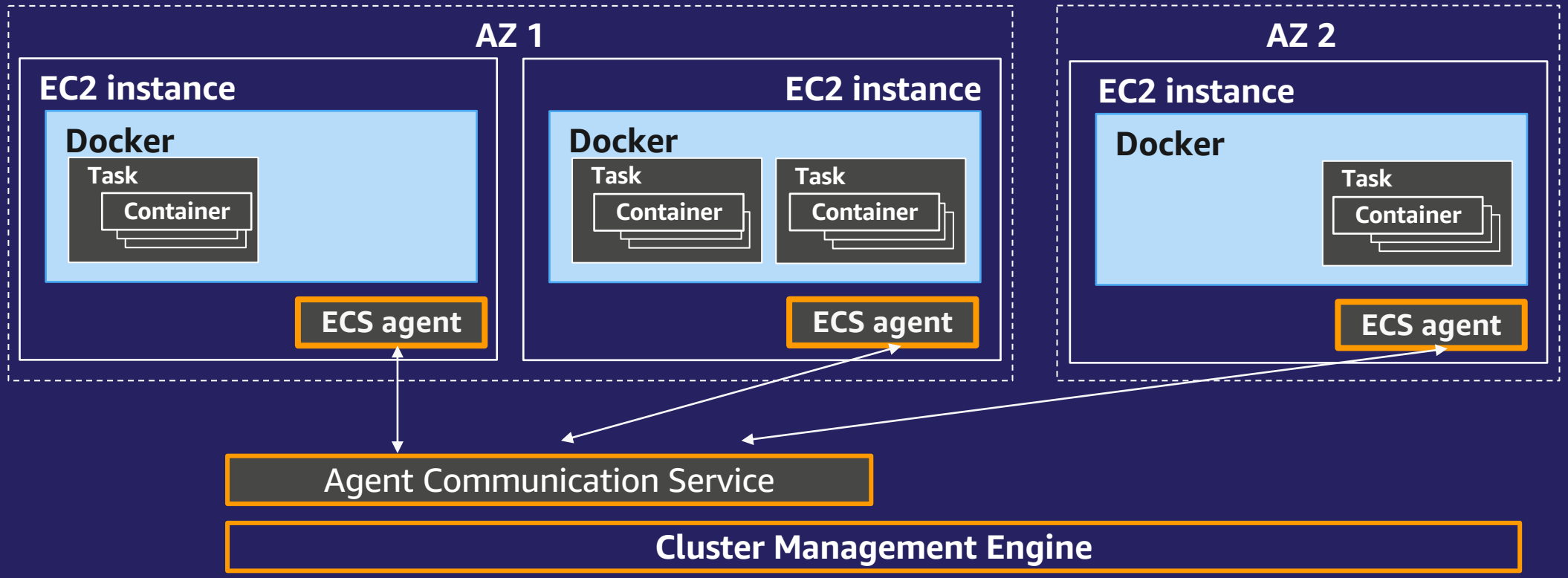
Amazon ECS 구성 요소: 클러스터



Amazon ECS 구성 요소: 클러스터 관리 엔진



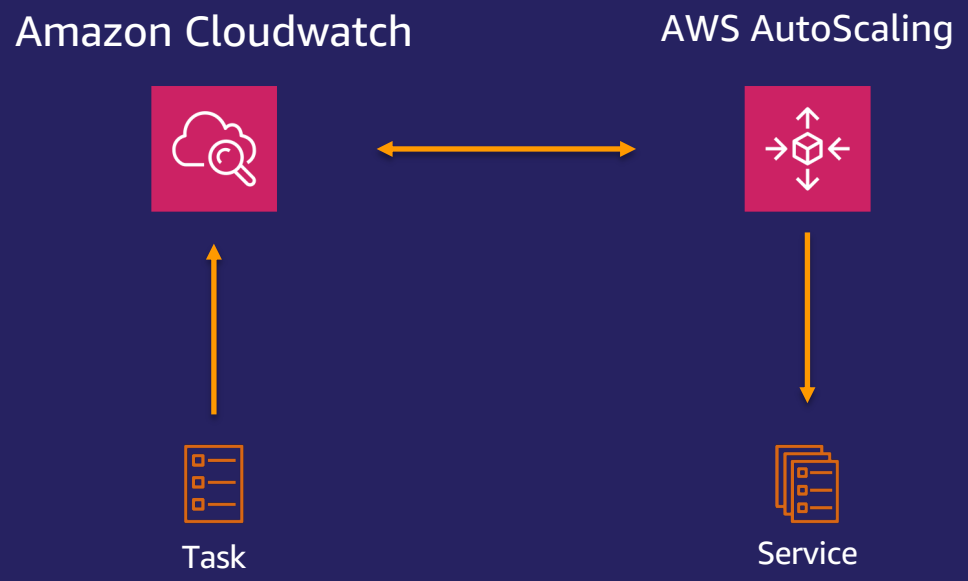
Amazon ECS 구성 요소: ECS 에이전트



Amazon ECS Auto Scaling

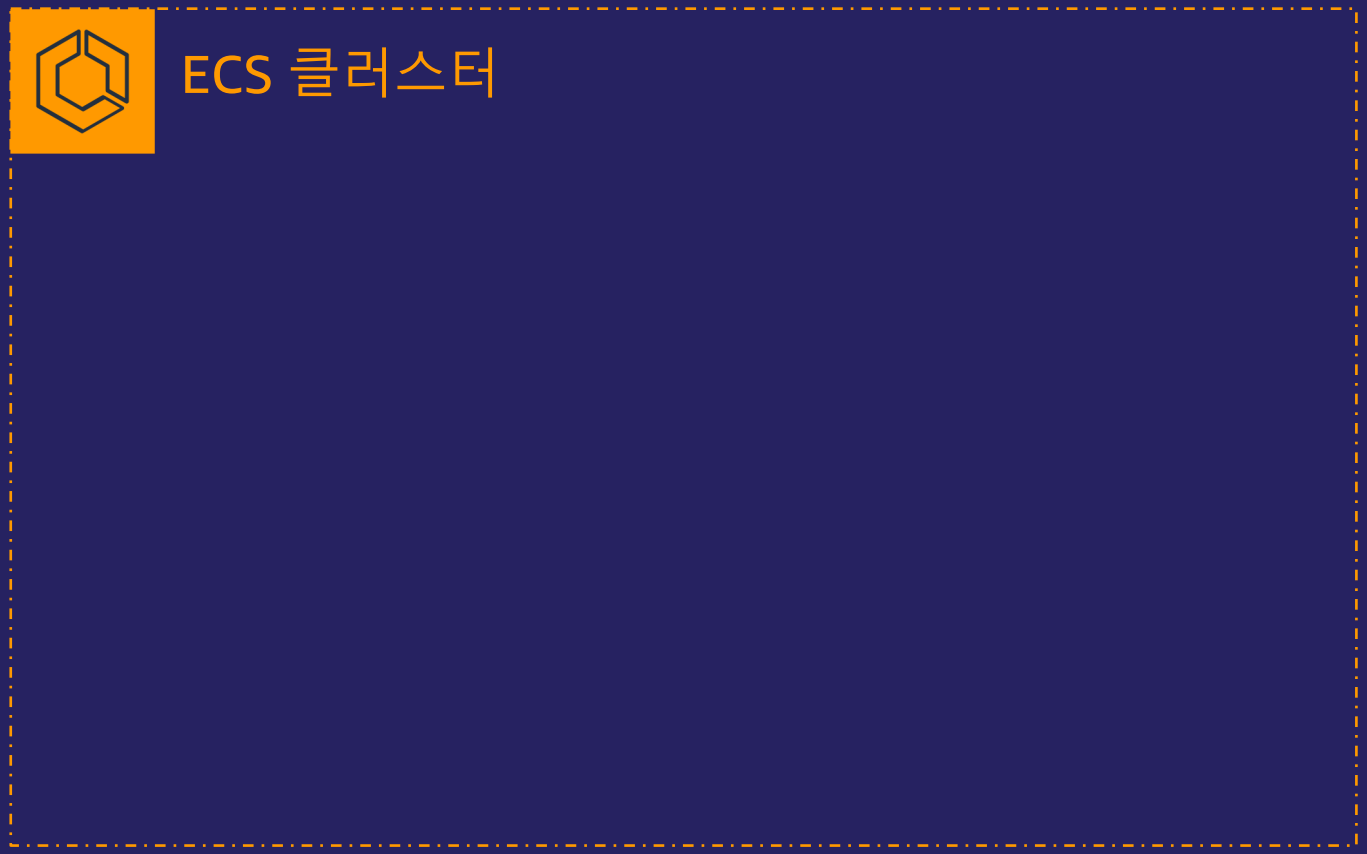
Auto Scaling

서비스 내의 작업 수를 자동으로 늘리거나 줄이는 역할을 함
ECS는 CPU 및 메모리 통계를 CloudWatch에 게시하고, Target Tracking, Step Scaling, 그리고 Scheduled Scaling을 지원함



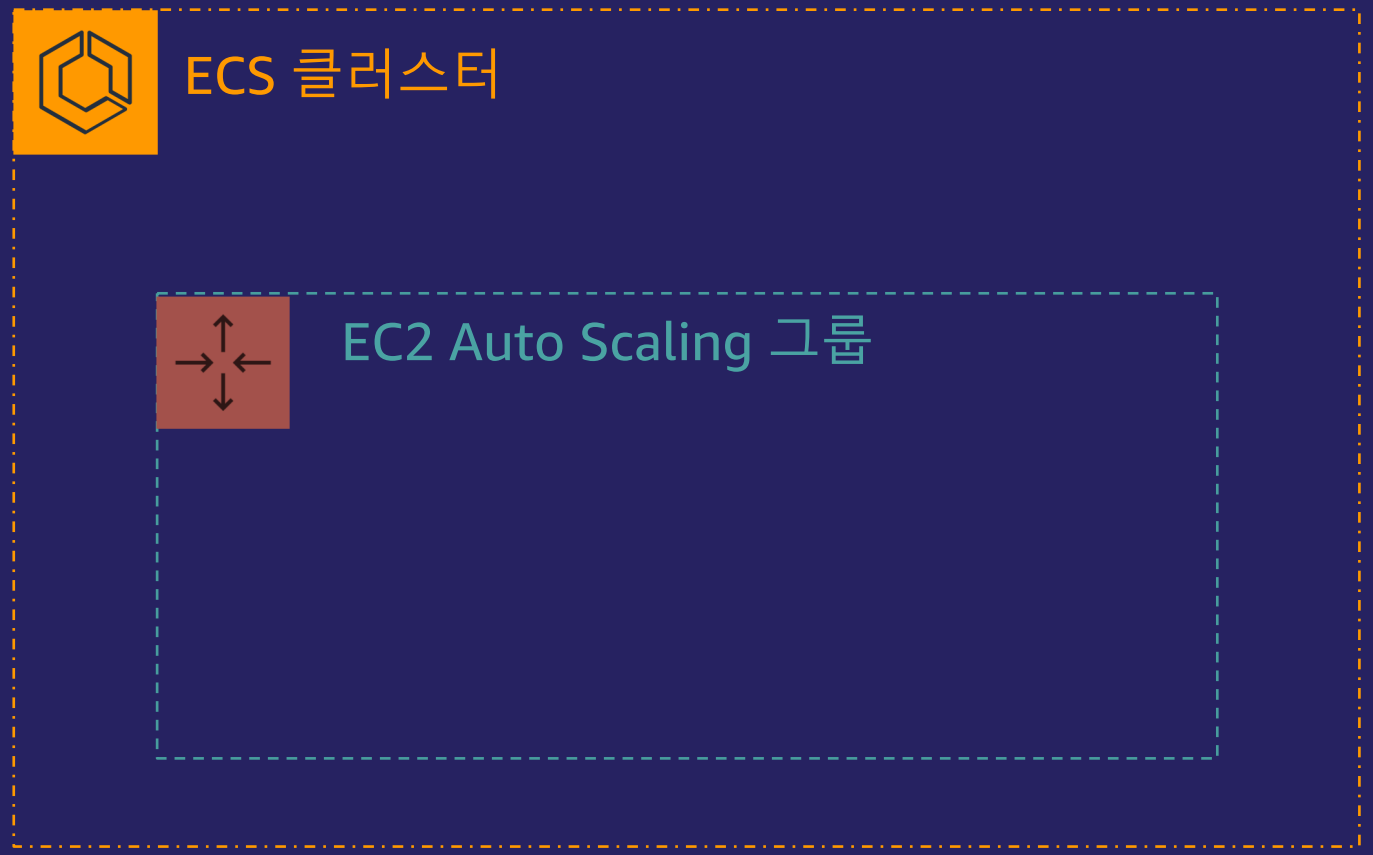
- AWS에서는 CPU 메트릭 기반의 Target Tracking 사용을 권고
- Target Tracking은 확장을 원활하게 수행
- Task는 CloudWatch로 데이터를 전송하고, CloudWatch는 서비스 작업 수를 늘리거나 줄이기 위해 AutoScaling을 트리거시킴

ECS 클러스터와 Amazon EC2 Auto Scaling 그룹



클러스터 생성

ECS 클러스터와 Amazon EC2 Auto Scaling 그룹



클러스터 생성

ASG 생성

ECS 클러스터와 Amazon EC2 Auto Scaling 그룹

클러스터 생성

ASG 생성

인스턴스 시작

ECS 클러스터와 Amazon EC2 Auto Scaling 그룹

클러스터 생성

ASG 생성

인스턴스 시작

작업 수행

클러스터 스케일링

클러스터 스케일링

ECS는 CloudWatch 지표를 반영

클러스터 스케일링

ECS는 CloudWatch 지표를 반영

경보는 ASG 스케일링

클러스터 스케일링

ECS는 CloudWatch 지표를 반영

경보는 ASG 스케일링

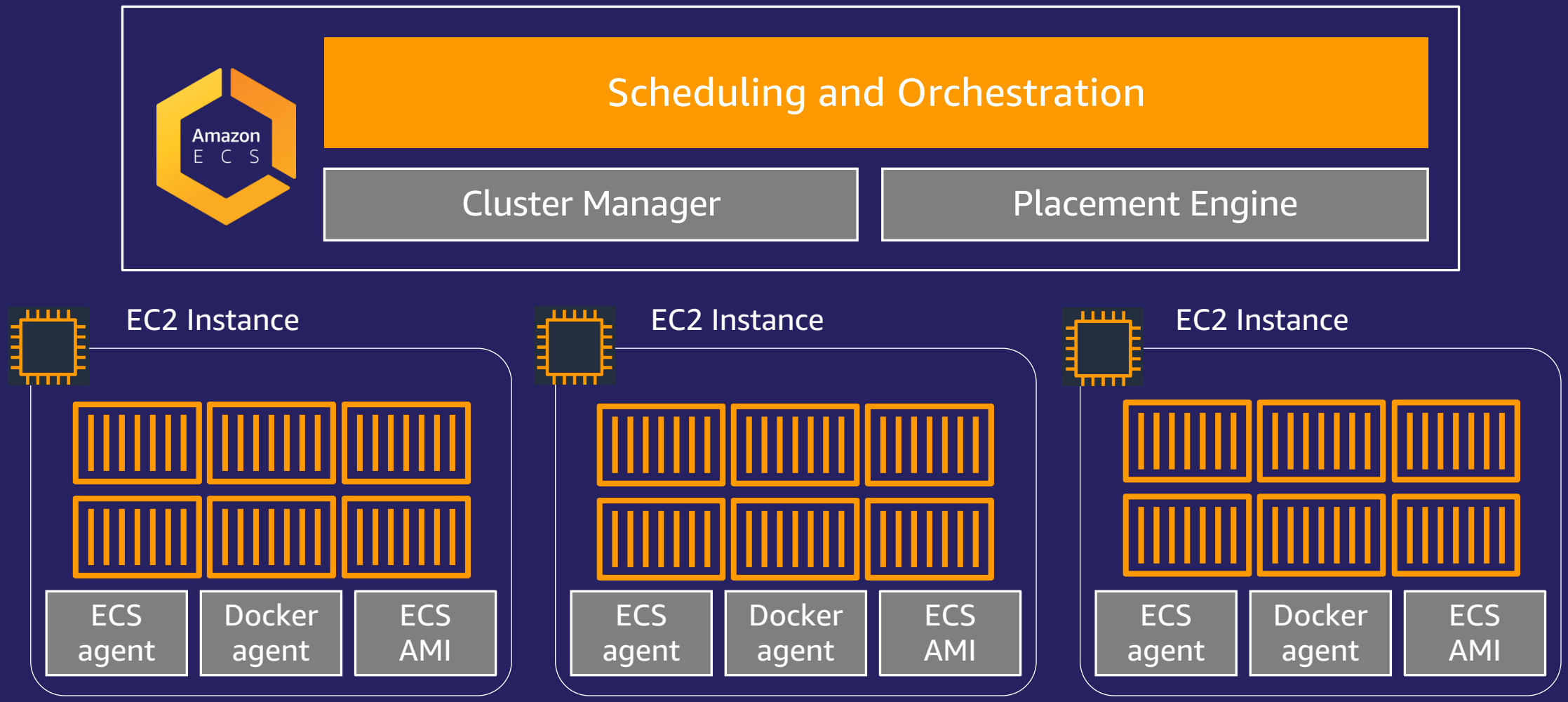
클러스터 스케일링

ECS는 CloudWatch 지표를 반영

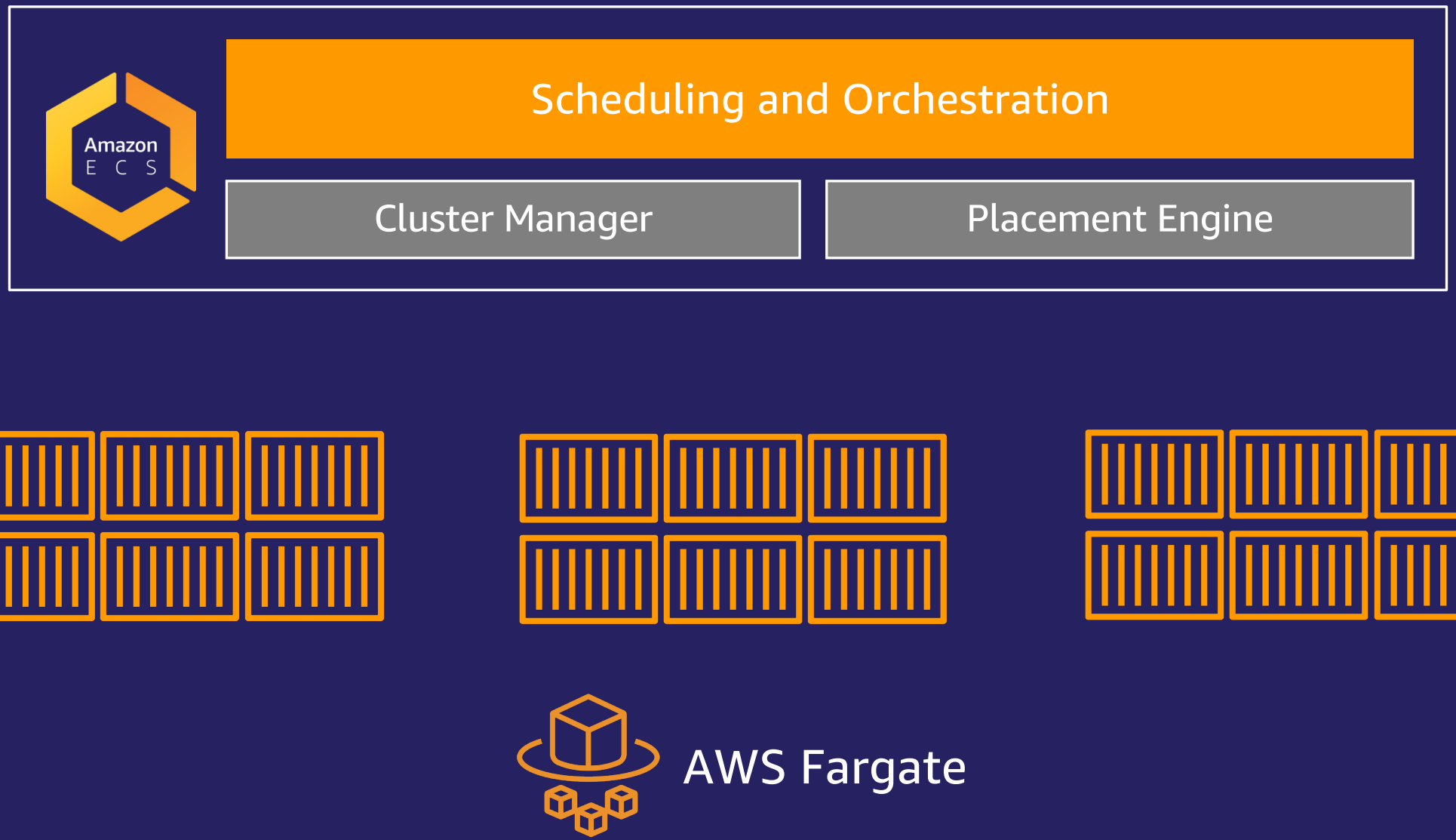
경보는 ASG 스케일링

AWS Fargate on Amazon ECS

Amazon ECS on EC2



Amazon ECS on Fargate



Fargate의 장점

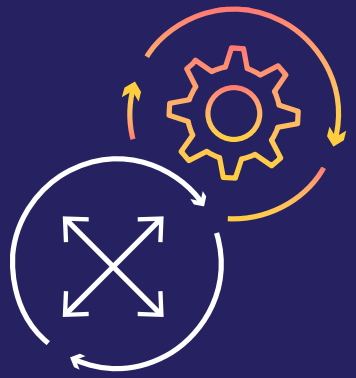
관리 업무로 인한 오버헤드 경감

까다로운 컨테이너 클러스터 관리를 AWS에 위임함으로써 고객은 애플리케이션에만 집중

기존 컨테이너 그대로 배포 가능

기존의 컨테이너 변경 불필요

현재 쿠버네티스, ECS 클러스터의 서비스, 워크플로우 그대로 이용

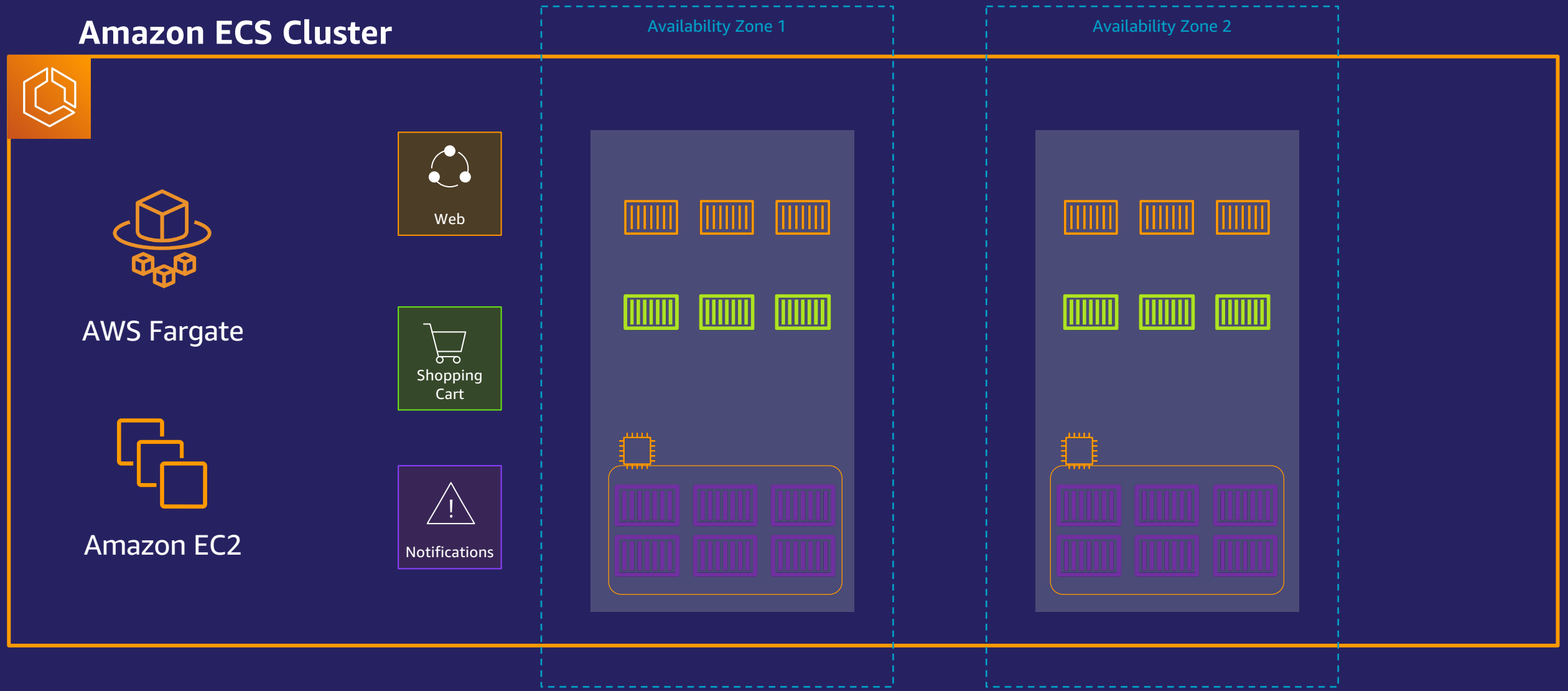


필요한 만큼만 & 쉬운 연동

컨테이너 실행에 필요한 자원 만큼만 비용 지불

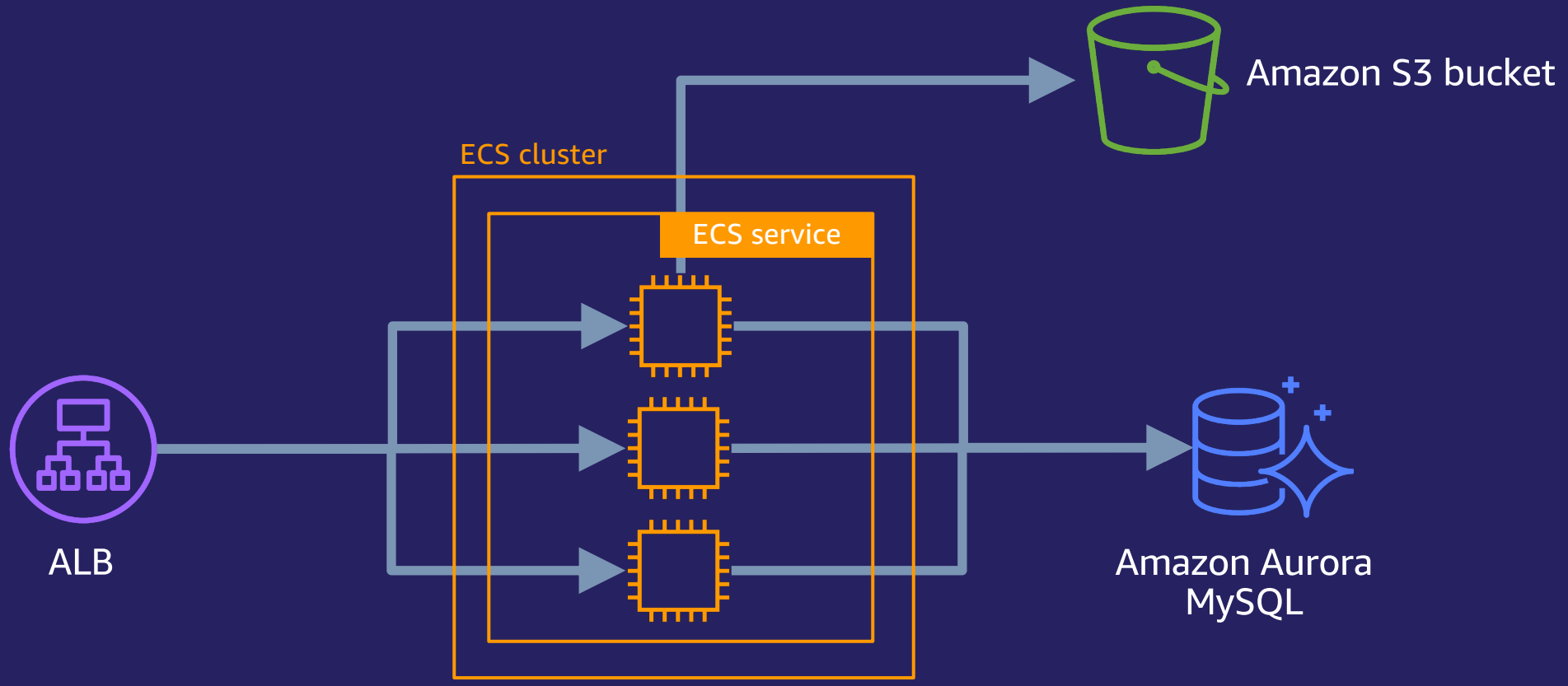
기존 AWS 네트워크, 보안과 네이티브하게 통합하여 사용

Amazon EC2와 Fargate의 Hybrid Architecture

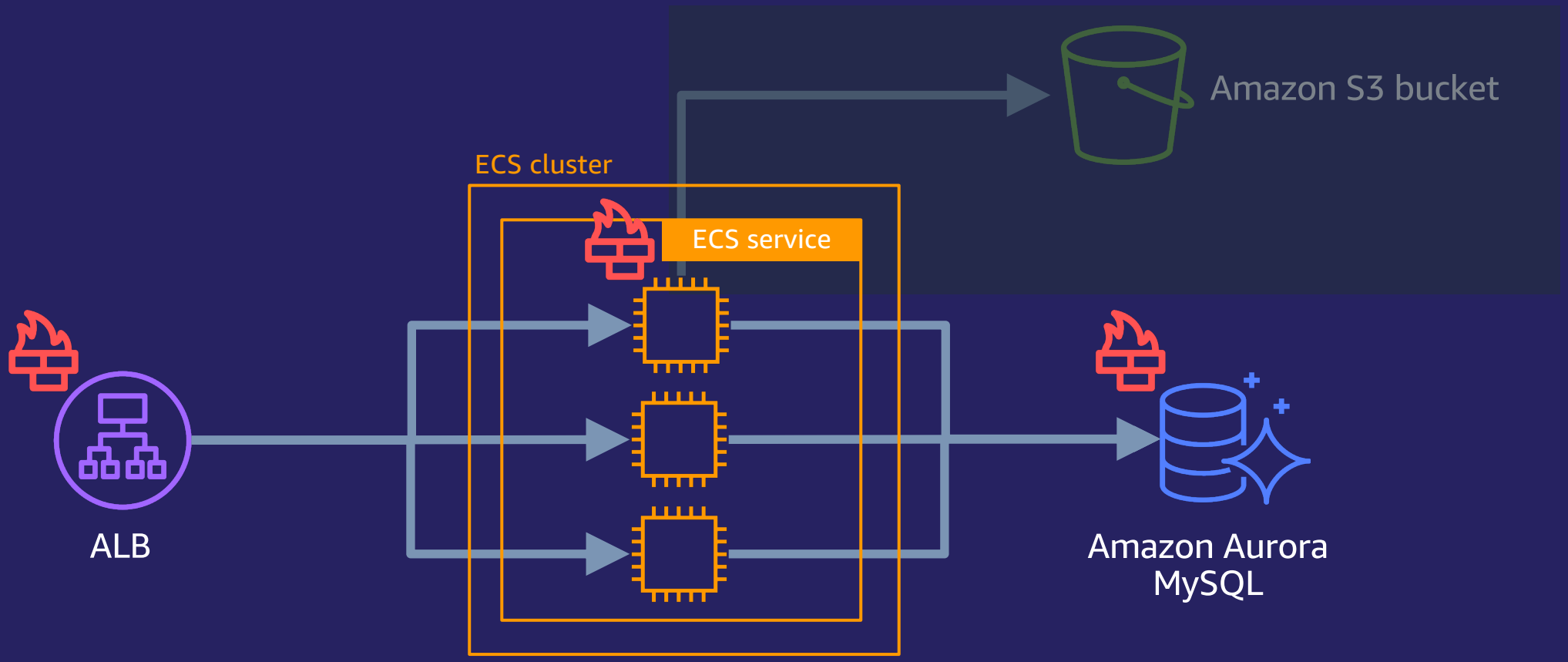


컨테이너 보안

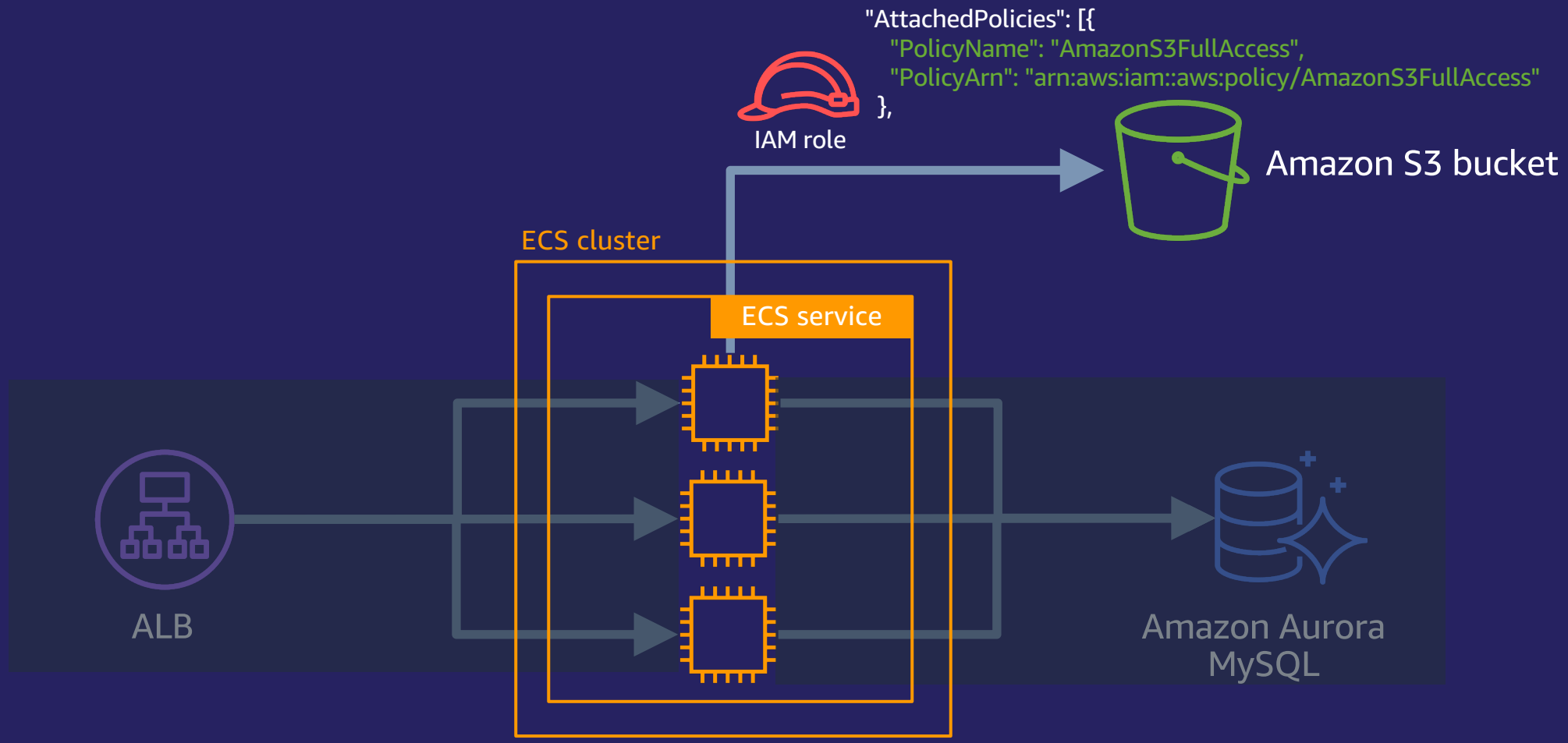
ECS 기반의 애플리케이션



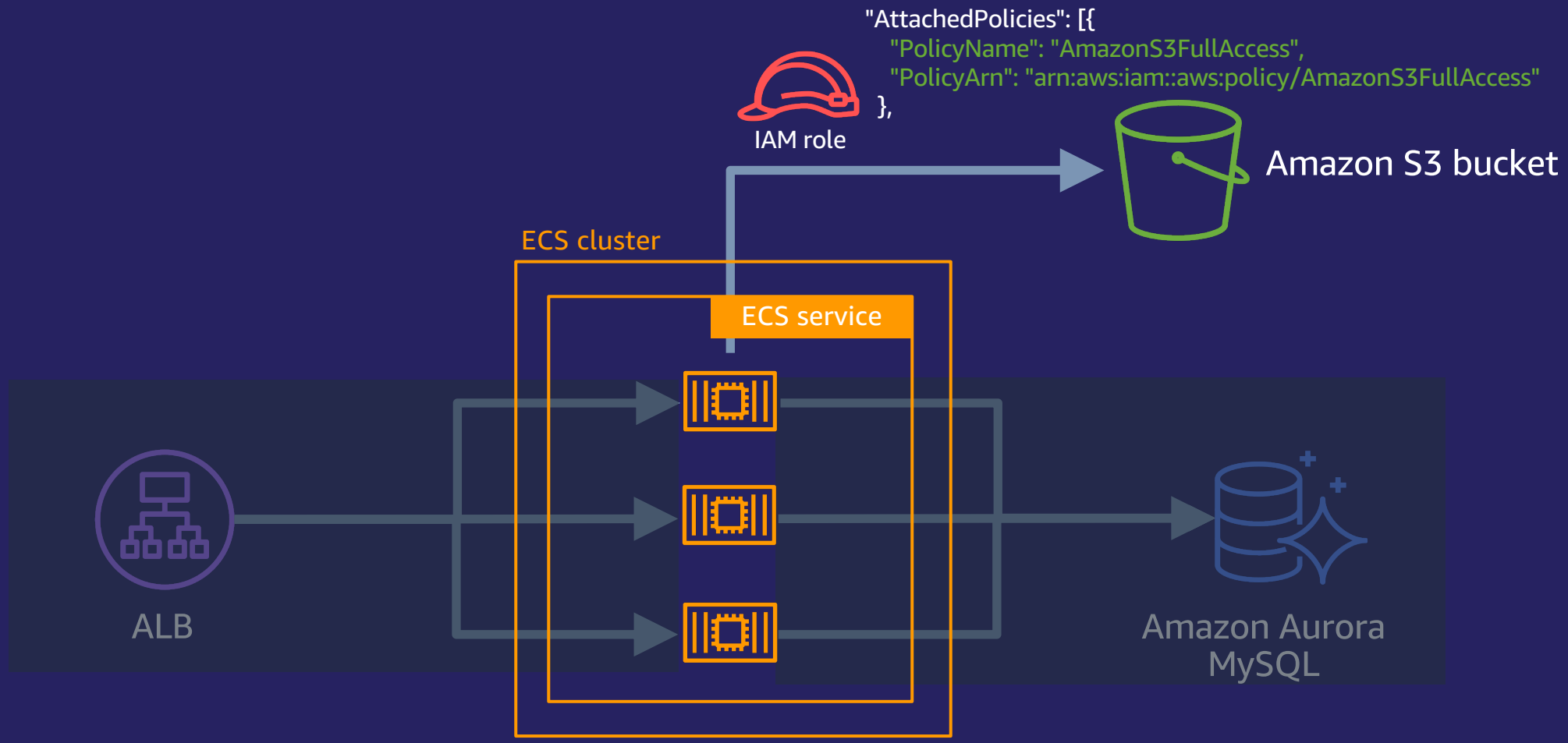
보안 원칙: 접근 제어 및 권한 관리



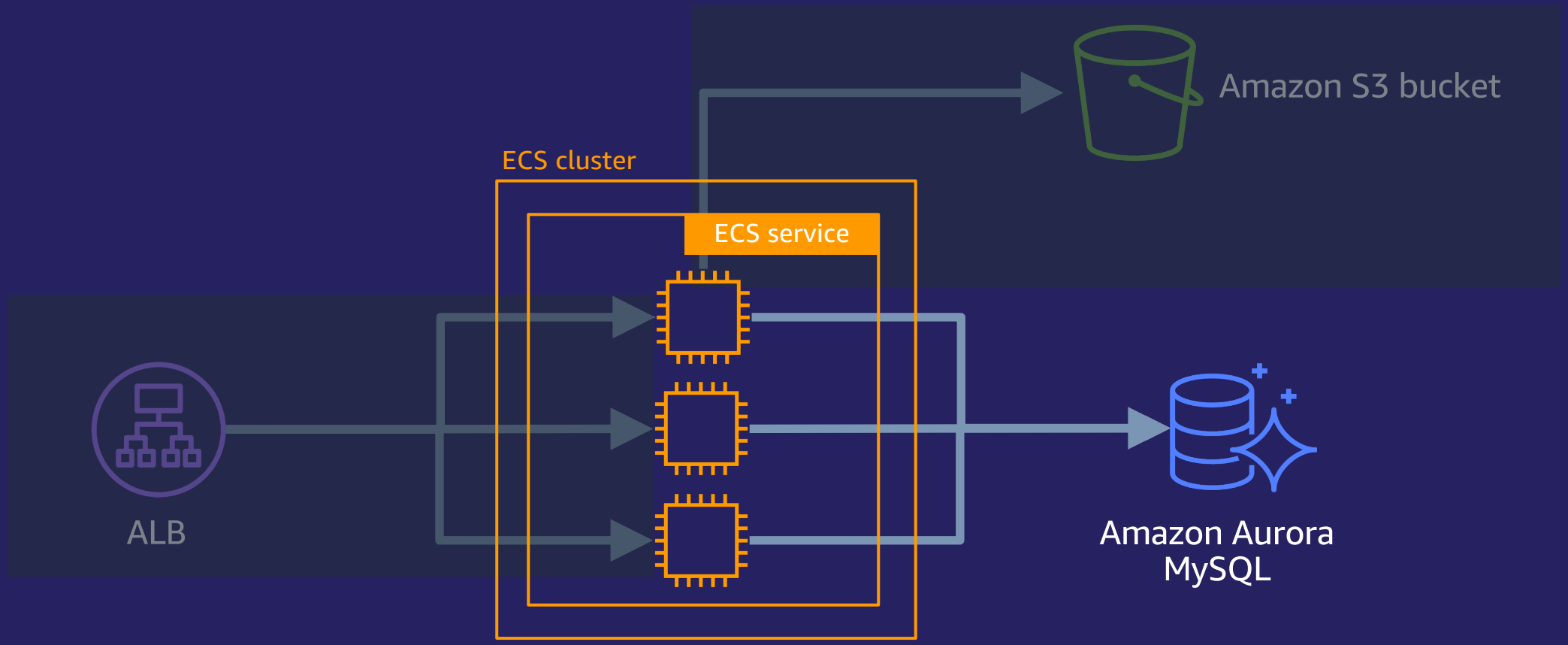
보안 원칙: 접근 제어 및 권한 관리



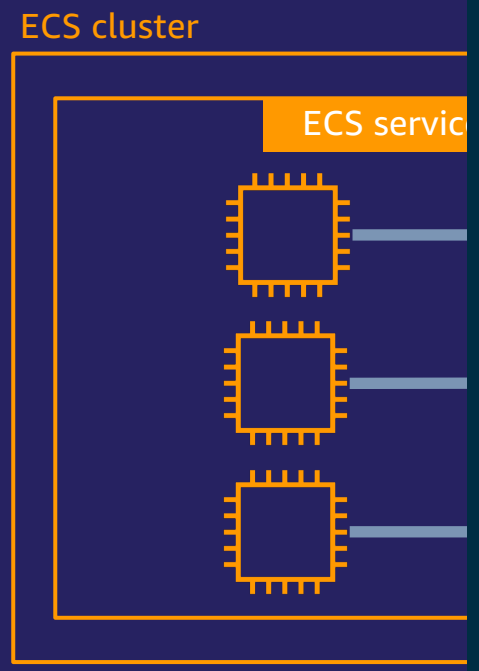
보안 원칙: 접근 제어 및 권한 관리



보안 원칙: 시크릿 관리



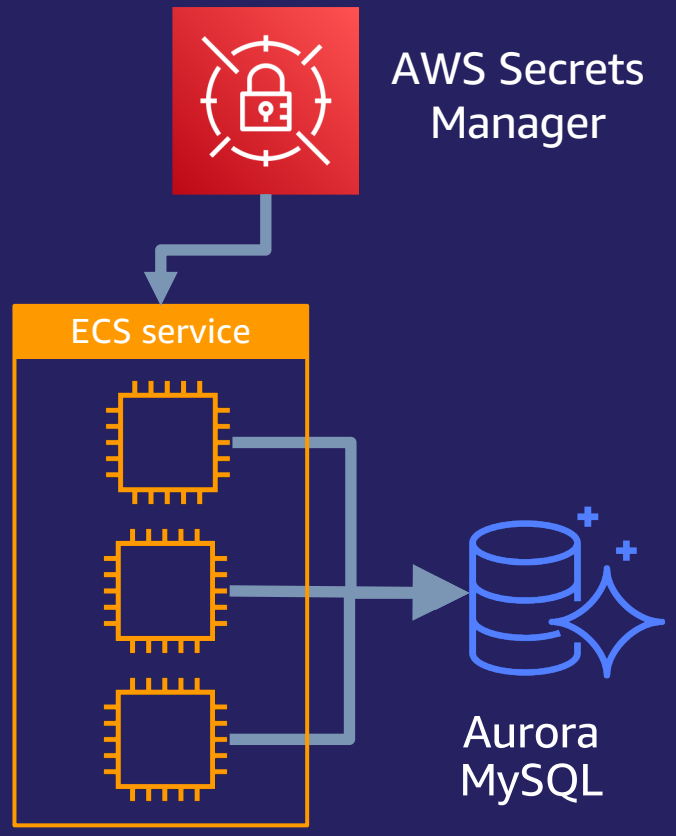
보안 원칙: 시크릿 관리



ECS task definition

```
{
  ... snip ...
  "containerDefinitions":[{
    "name":"my-web-app",
    "image":"my-web-app:v1",
      ... snip ...
    "environment": [{
      "name": "DB_USER",
      "value": "myappdbuser"
    },{
      "name": "DB_PASSWD",
      "value": "mysupersecretpasswd"
    },{
      "name": "DB_HOST",
      "value": "my-db...rds.amazonaws.com"
    },{
      "name": "DB_PORT",
      "value": "3306"
    }],
      ... snip ...
  }],
  ... snip ...
}
```

보안 원칙: 시크릿 관리



ECS task definition

```
{
  ... snip ...
  "containerDefinitions":[{
    "name":"my-app",
    "image":"my-web-app:v1",
      ... snip ...
    "secrets": [{
      "name": "DB_USER",
      "valueFrom": "...secretsmanager:...:my-db-secret:username::"
    },{
      "name": "DB_PASSWD",
      "valueFrom": "...secretsmanager:...:my-db-secret:password::"
    }],
    "environment": [{
      "name": "DB_HOST",
      "value": "my-db...rds.amazonaws.com"
    },{
      "name": "DB_PORT",
      "value": "3306"
    }]
      ... snip ...
}
```

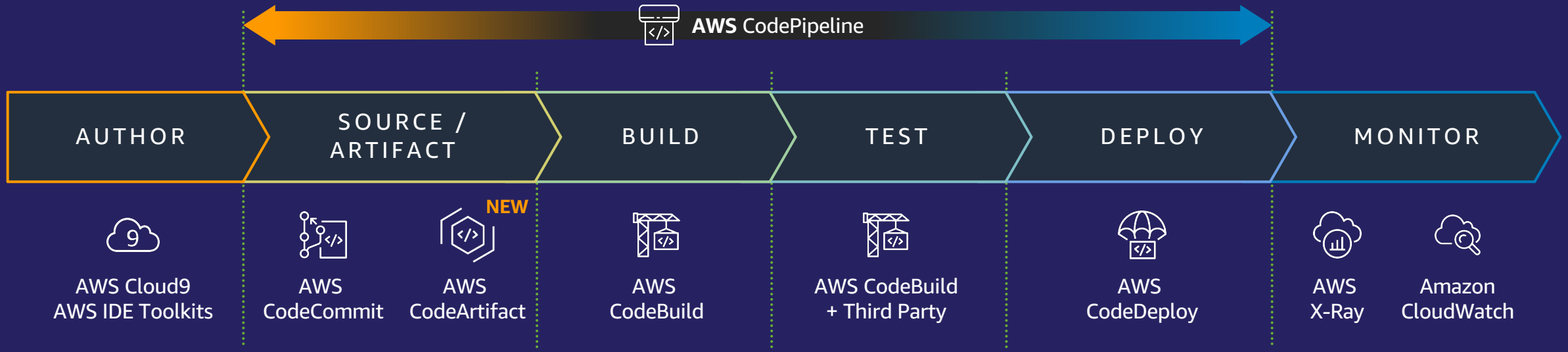
컨테이너 관련 AWS 서비스들

Amazon Elastic Container Registry (ECR)

- 완전 관리형 **컨테이너 레지스트리**
- 배포 워크플로우 간소화

  - Amazon EKS, Amazon ECS, Amazon Lambda, AWS Fargate
- 권한 제어 및 수명 주기 정책 규칙 설정
- 이미지 취약성 스캐닝 기능
- 퍼블릭/프라이빗 레파지토리 및 퍼블릭 갤러리
- 교차 리전/교차 계정 복제

AWS Developer Tool

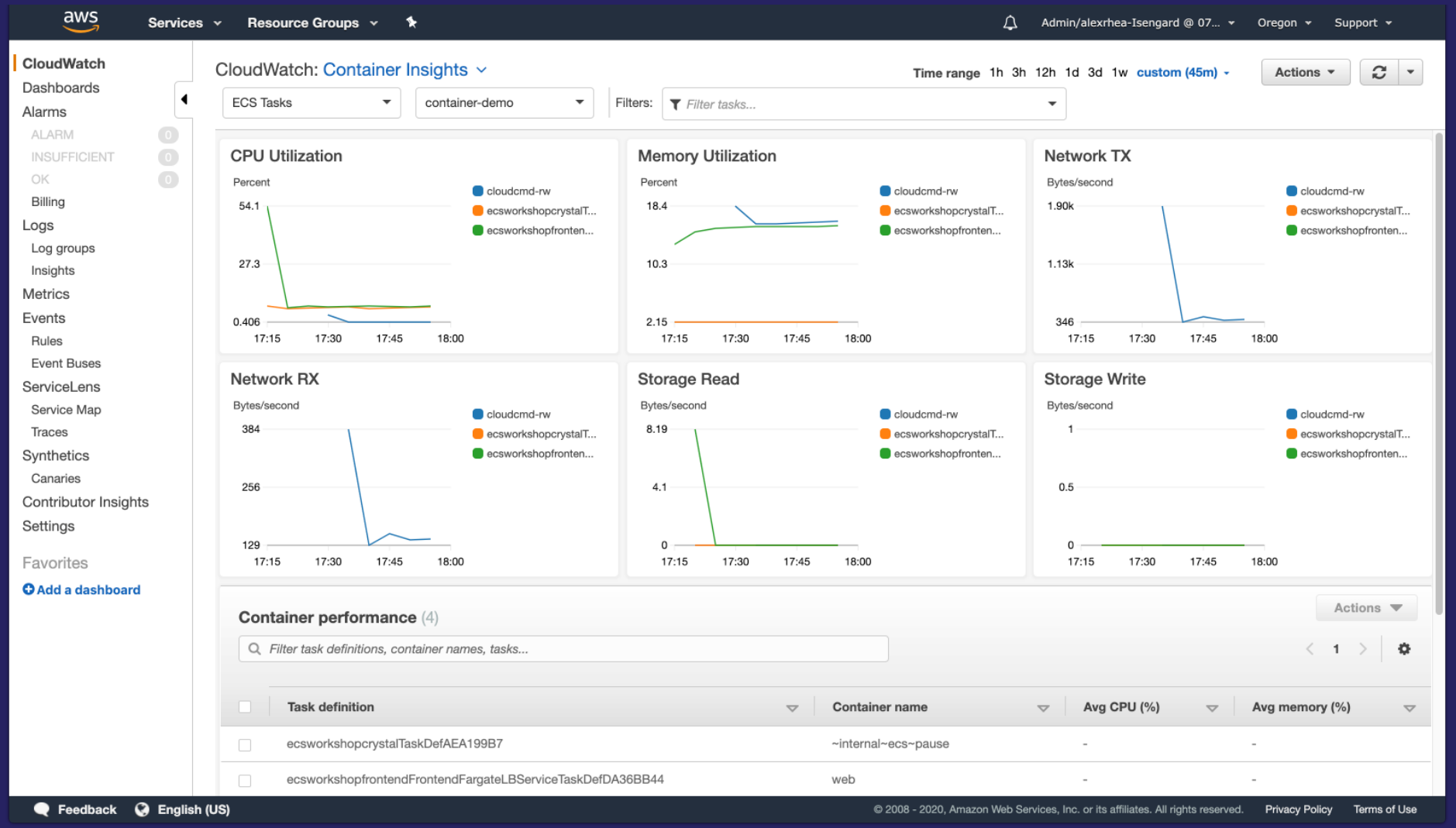


Amazon CloudWatch & AWS X-Ray

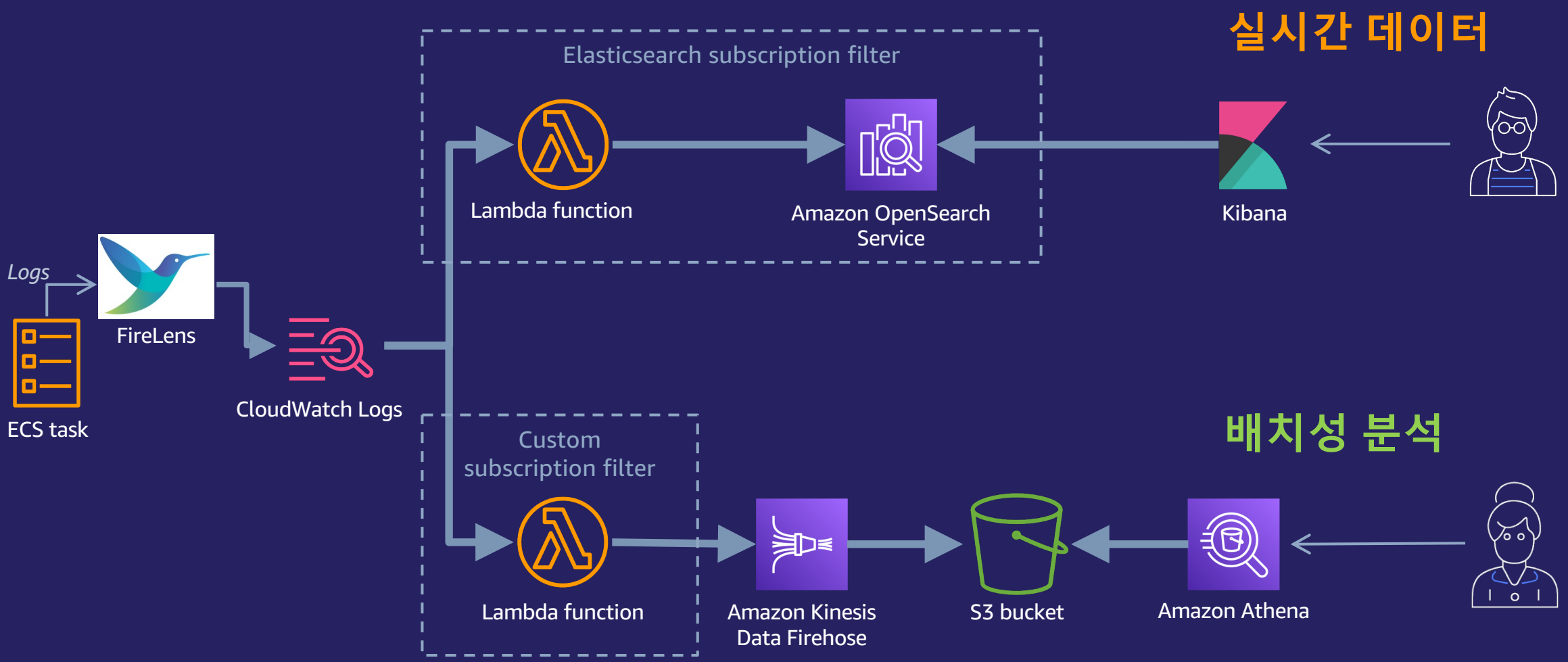
Amazon CloudWatch

AWS X-Ray

Amazon CloudWatch Container Insights



컨테이너 로그 수집: AWS FireLens



더 나은 세미나를 위해 여러분의 의견을 남겨주세요!

- 질문에 대한 답변 드립니다.

Thank you!