



# AWS 기반 컨테이너

# 학습 내용

## 강의의 핵심

배울 내용은 다음과 같습니다.

- 컨테이너의 목적과 기능, 컨테이너 사용을 지원하는 AWS 서비스 설명하기
- 컨테이너 및 컨테이너의 이점 특성화하기
- Docker 개요 및 Docker 배포 검토하기
- AWS 기반의 다양한 컨테이너 서비스 비교 및 상관 관계 찾기
- Kubernetes와 Amazon Elastic Kubernetes Service(Amazon EKS) 구별하기
- AWS Fargate 및 컨테이너 관리 과제 식별하기

**데모:** Amazon Elastic Container Service(Amazon ECS)에 도커(Docker) 컨테이너 배포



주요 용어:

- 컨테이너
- Docker
- 무분별한 컨테이너 난립
- 좀비 컨테이너
- Amazon Elastic Container Service(Amazon ECS)
- AWS Fargate
- 클러스터
- Kubernetes
- Amazon Elastic Kubernetes Service(Amazon EKS)
- Amazon Elastic Container Registry(Amazon ECR)



# 문제

# 신규 및 업데이트된 소프트웨어 문제에 대한 몇 가지 일반적인 응답

- "제 시스템에서 실행했어요!"
- "컴퓨터의 운영 체제를 업데이트해야 합니다!"
- "실행에 필요한 일부 소프트웨어만 있습니다!"
- ...이외 여러 응답이 있습니다!

신규 및 업데이트된 소프트웨어로 이러한 문제 및 기타 다른 문제를 어떻게 해결할 수 있습니까?



컨테이너

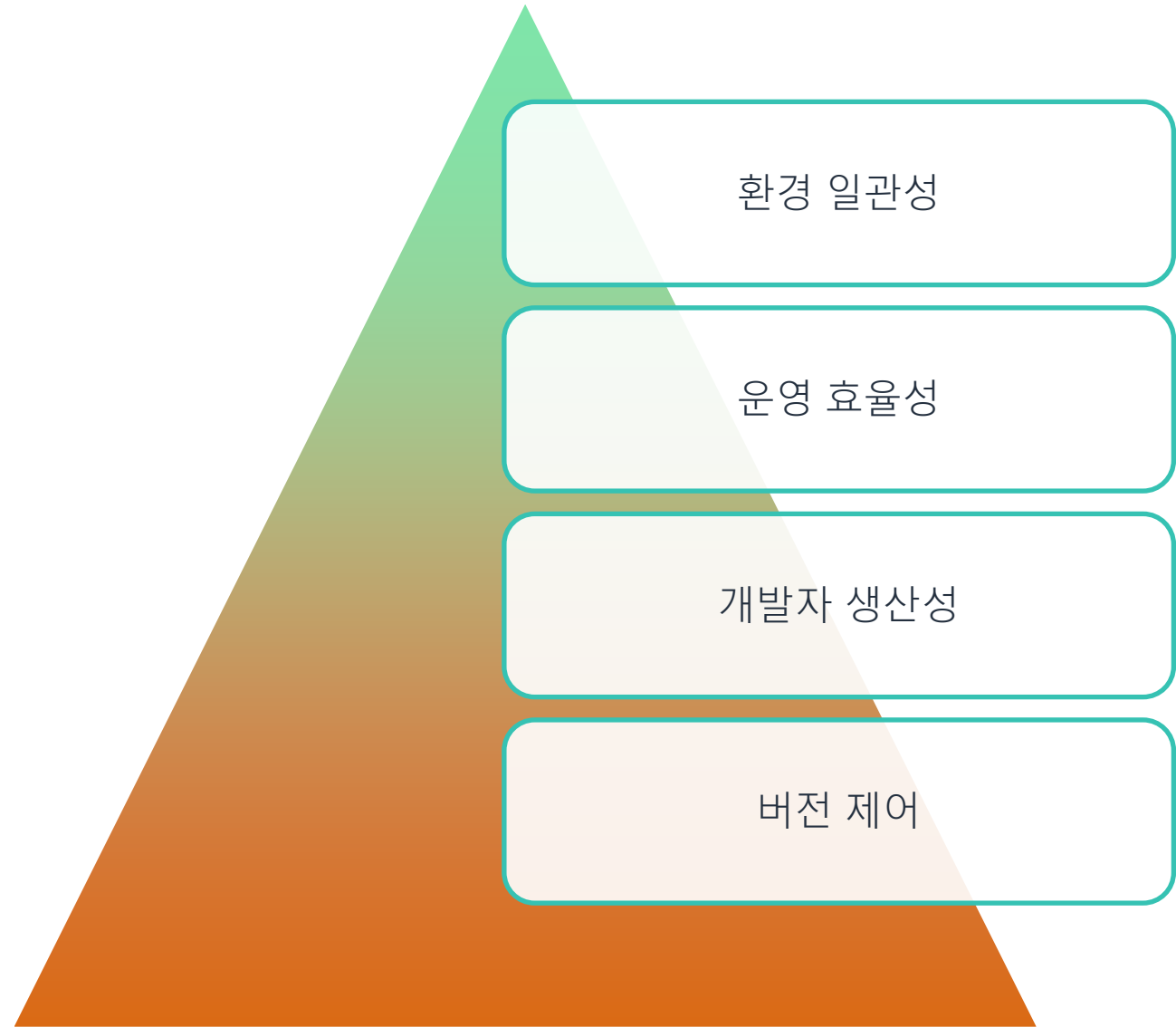
# 컨테이너 소개

컨테이너의 특성은 다음과 같습니다.

- 운영 체제(OS)를 가상화하는 방법
- 리소스 격리 프로세스에서 실행할 수 있는 애플리케이션 및 해당 종속성



# 컨테이너 사용의 이점



# 활용 방법

컨테이너는 신규 및 업데이트된 소프트웨어의 일부 문제를 해결하는데 도움이 됩니다.

하지만 다음 항목을 어떻게 수행할 수 있습니까?

- 컨테이너 생성
- 컨테이너 실행
- 여러 컨테이너 관리 및 배포







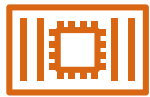
# Docker

# Docker

- Docker는 컨테이너를 생성, 관리 및 실행하는 데 사용되는 애플리케이션 플랫폼입니다.
- **Docker**를 사용하면 개발자와 엔지니어가 컨테이너를 빌드, 테스트, 배포 및 실행할 수 있습니다.



컨테이너 1



컨테이너 2



컨테이너 3

컨테이너에는 소프트웨어를 실행하는 데 필요한 모든 항목이 포함되어 있습니다.

라이브러리

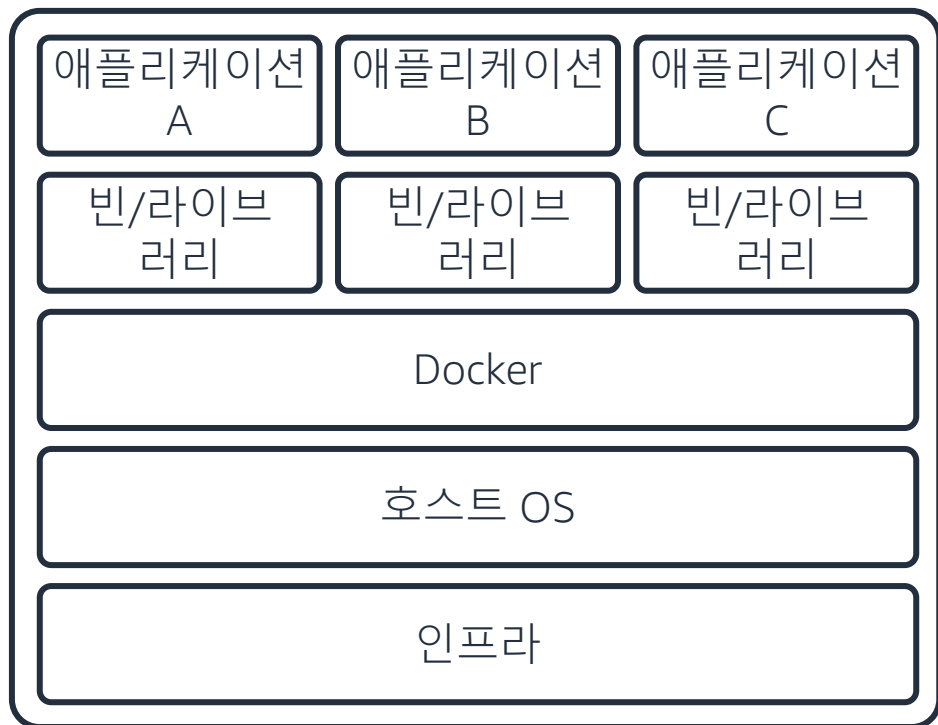
시스템 도구

코드

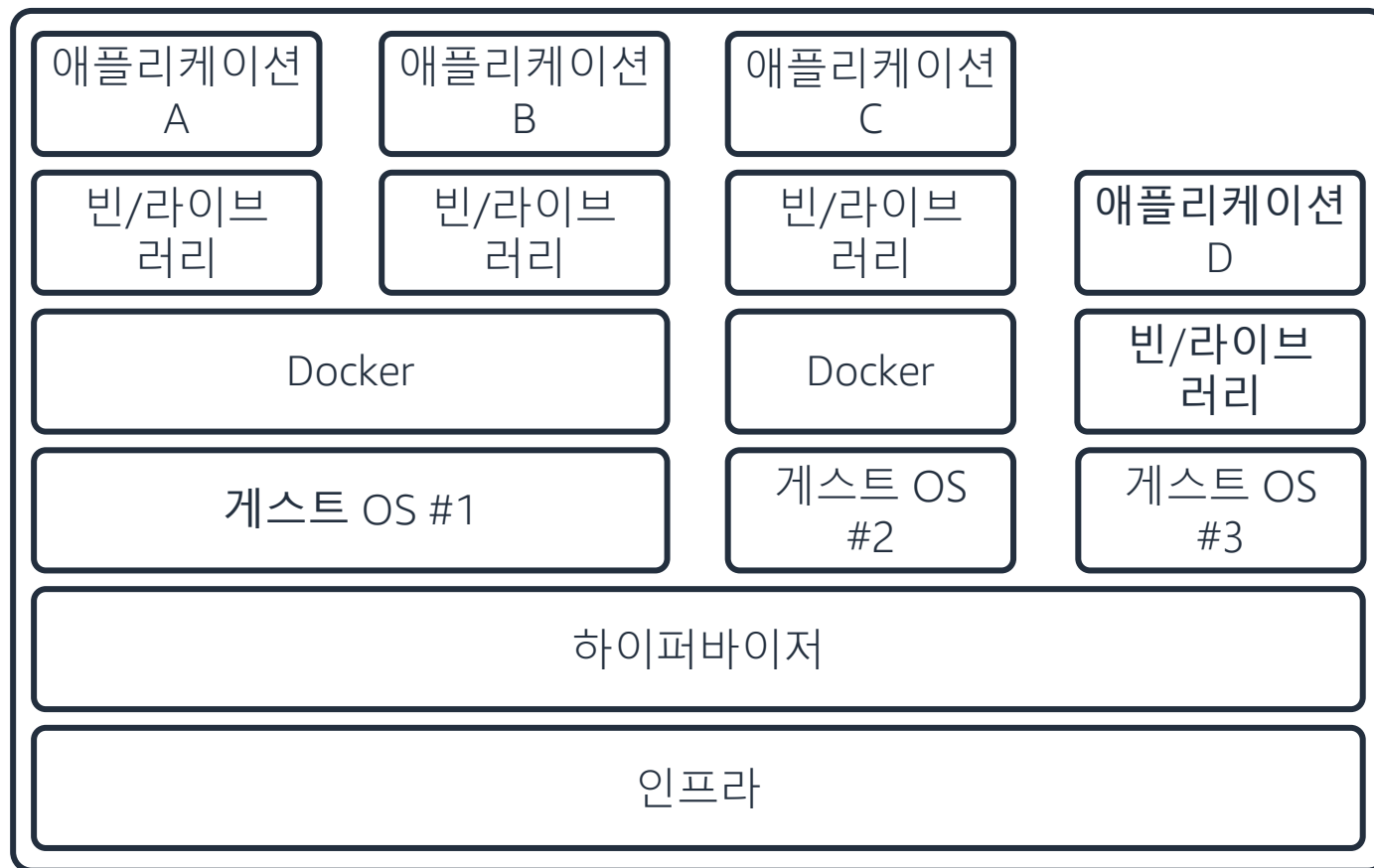
런타임

# Docker 예제 배포

Docker가 설치된 단일 호스트 운영 체제 및  
Docker가 온프레미스에서 실행되는 방식



3개의 게스트 운영 체제에서 호스팅되는 4개의 애플리케이션과  
AWS에서 실행되는 방법





AWS 기반 Docker:

Amazon Elastic Container Registry(Amazon ECR)

Amazon Elastic Container Service(Amazon ECS)

# Amazon Elastic Container Registry(Amazon ECR)

**Amazon ECR**은 개발자가 Docker 컨테이너 이미지를 손쉽게 저장, 관리 및 배포할 수 있도록 돕는 완전관리형 Docker 컨테이너 레지스트리입니다.



Amazon Elastic  
Container Registry



이미지



레지스트리



# Amazon Elastic Container Service(Amazon ECS)

## Amazon Elastic Container Service(Amazon ECS)의 특성

- 확장성이 뛰어난 고성능 컨테이너 관리 서비스
- Docker 컨테이너 지원



# Amazon ECS 이점

Amazon ECS 이점은 다음과 같습니다.



# 데모: Amazon Elastic Container Service(Amazon ECS)에 도커(Docker) 컨테이너 배포





# 컨테이너 관리



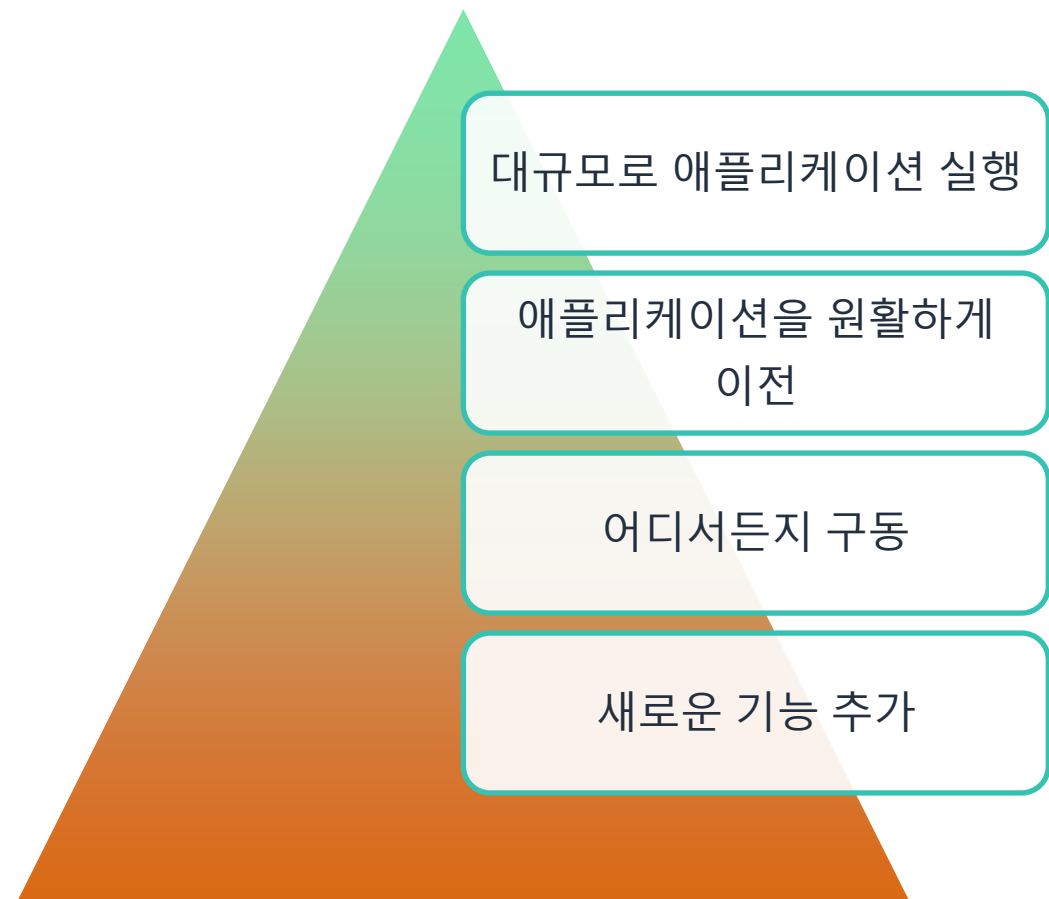
# Kubernetes

# Kubernetes

Kubernetes는 컨테이너 관리 및 오케스트레이션을 위한 오픈 소스 소프트웨어입니다.

Kubernetes를 사용해야 하는 이유:

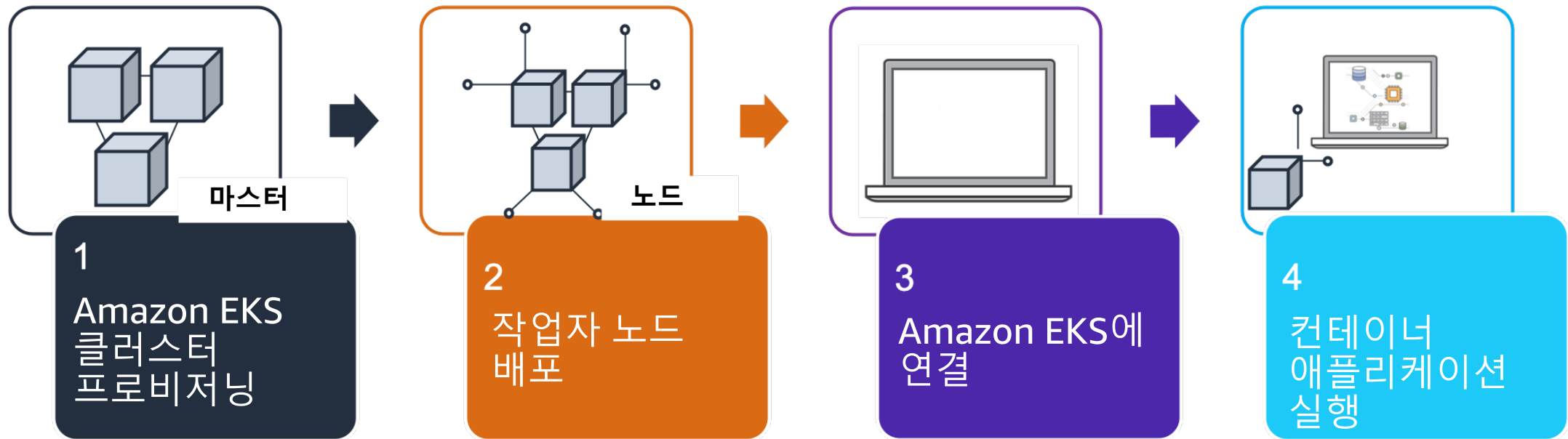
- EC2 컴퓨팅 인스턴스의 클러스터 관리
- 배포, 유지 관리 및 확장 프로세스를 통해 해당 인스턴스에서 컨테이너 실행





# Amazon Elastic Kubernetes Service(Amazon EKS): Kubernetes로 컨테이너 관리

# Amazon Elastic Kubernetes Service(Amazon EKS)



# AWS Fargate: 서버를 관리하지 않고 컨테이너 실행

# AWS Fargate



AWS Fargate

AWS Fargate는 서버나 클러스터를 관리할 필요 없이 컨테이너를 실행할 수 Amazon ECS용 기술입니다.

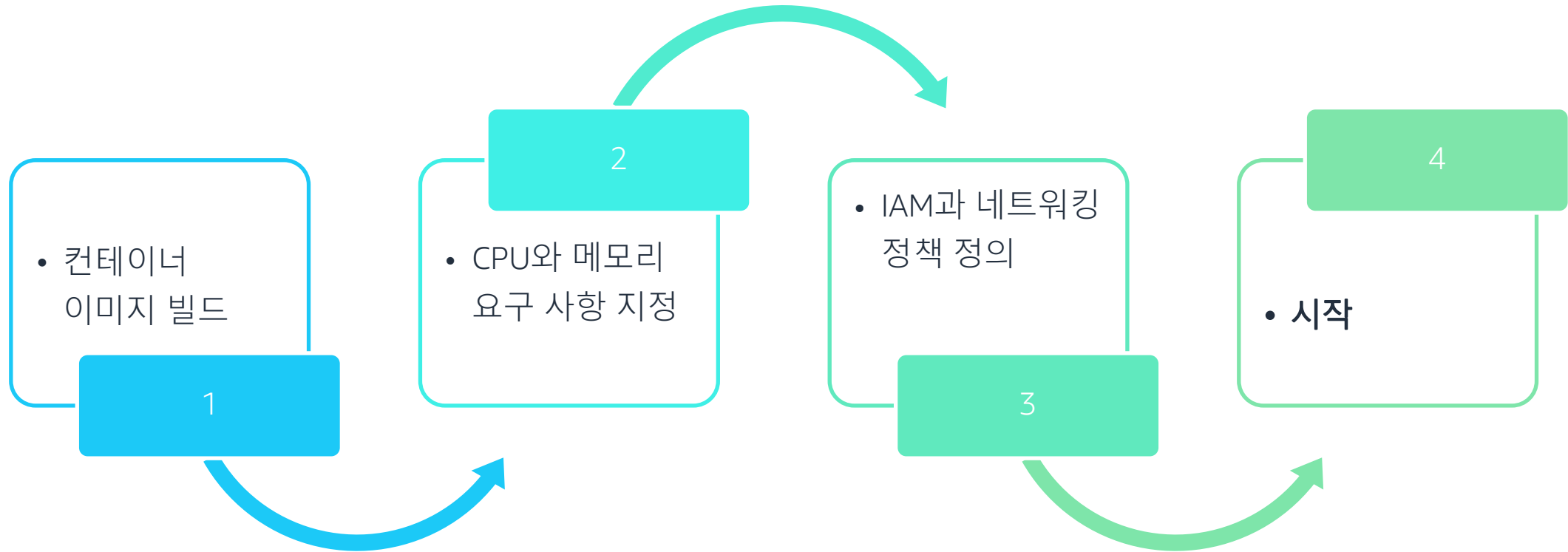


# AWS Fargate 및 Amazon ECS 스택





# AWS Fargate 시작 유형 사용





# 컨테이너 관리 과제

# 컨테이너 관리 과제(1/2)

컨테이너 관리의 당면 과제는 다음과 같습니다.

컨테이너 사용의 급격한 증가

무분별한 컨테이너 난립 방지

버전 차이

소유권

## 컨테이너 관리 과제(2/2)

컨테이너 관리의 당면 과제는 다음과 같습니다.

빈 패킹

더 이상 필요하지 않은 컨테이너(좀비 컨테이너)

사라지는 컨테이너

# 학습 내용 확인



가상 머신과 컨테이너의 차이점은 무엇입니까?

Docker란 무엇입니까?

Kubernetes란 무엇입니까?

# 핵심 사항



- 컨테이너에는 소프트웨어를 실행하는 데 필요한 모든 항목이 포함되어 있습니다.
- Docker는 소프트웨어를 컨테이너로 패키징하는 소프트웨어 플랫폼입니다.
- 단일 애플리케이션이 여러 컨테이너에 걸쳐 구현될 수 있습니다.
- Kubernetes는 컨테이너 오케스트레이션을 위한 오픈 소스 소프트웨어입니다.

필수 항목:  **이해도 점검**