



AWS BUILDERS KOREA PROGRAM 100

# Container 와 EKS 시작하기

방신철

Solutions Architect  
Amazon Web Services

# 강연 중 질문하는 방법

AWS Builders Korea Go to Webinar “Questions (질문) ” 창에 자신이 질문한 내역이 표시됩니다. 본인만 답변을 받고 싶으실 경우 (비공개)라고 하고 질문해 주시면 됩니다.

질문 주신 사항에 대해서는 질문창을 통해 답변을 드립니다.

▼ Questions

☒ Show Answered Questions

Question	Asker

Type answer here

## 고지 사항 (Disclaimer)

본 콘텐츠는 고객의 편의를 위해 AWS 서비스 설명을 위해 온라인 세미나용으로 별도로 제작, 제공된 것입니다. 만약 AWS 사이트와 콘텐츠 상에서 차이나 불일치가 있을 경우, AWS 사이트([aws.amazon.com](https://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.

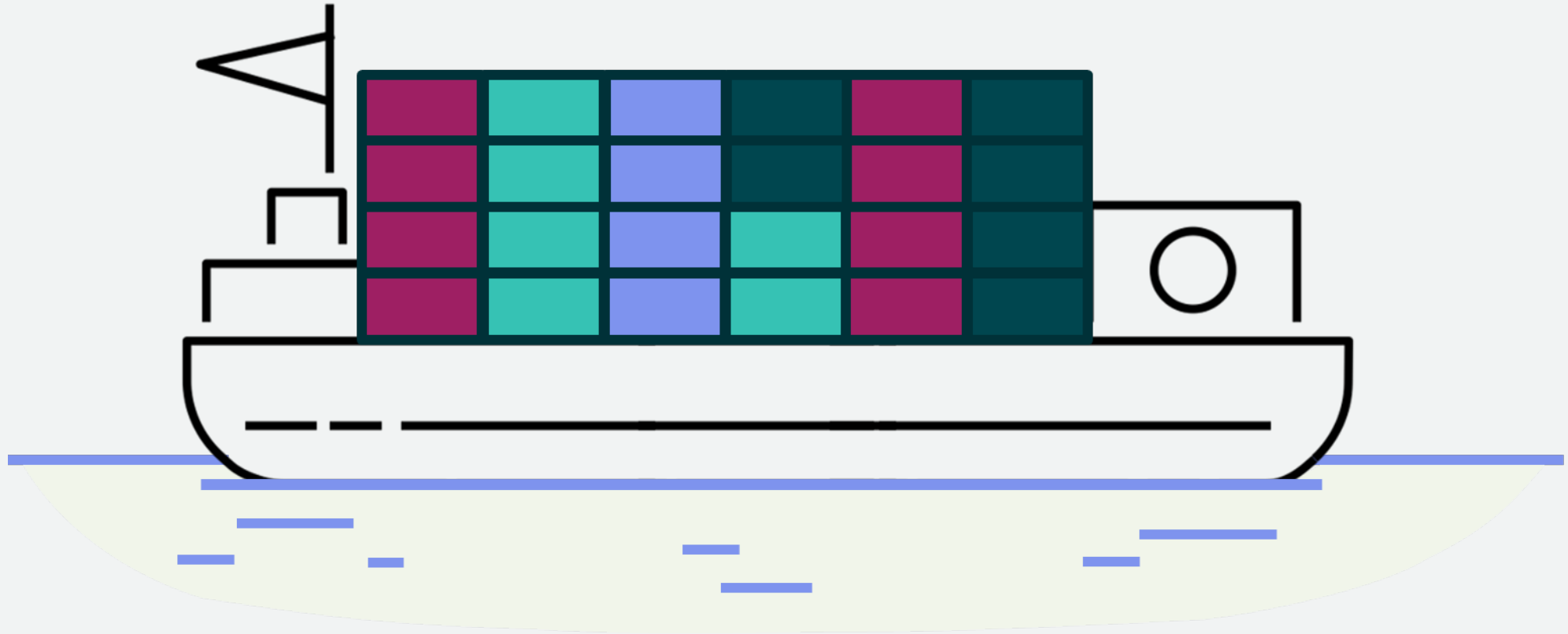
AWS는 본 콘텐츠에 포함되거나 콘텐츠를 통하여 고객에게 제공된 일체의 정보, 콘텐츠, 자료, 제품(소프트웨어 포함) 또는 서비스를 이용함으로써 인하여 발생하는 어떠한 종류의 손해에 대하여 어떠한 책임도 지지 아니하며, 이는 직접 손해, 간접 손해, 부수적 손해, 징벌적 손해 및 결과적 손해를 포함하되 이에 한정되지 아니합니다.

# 목차

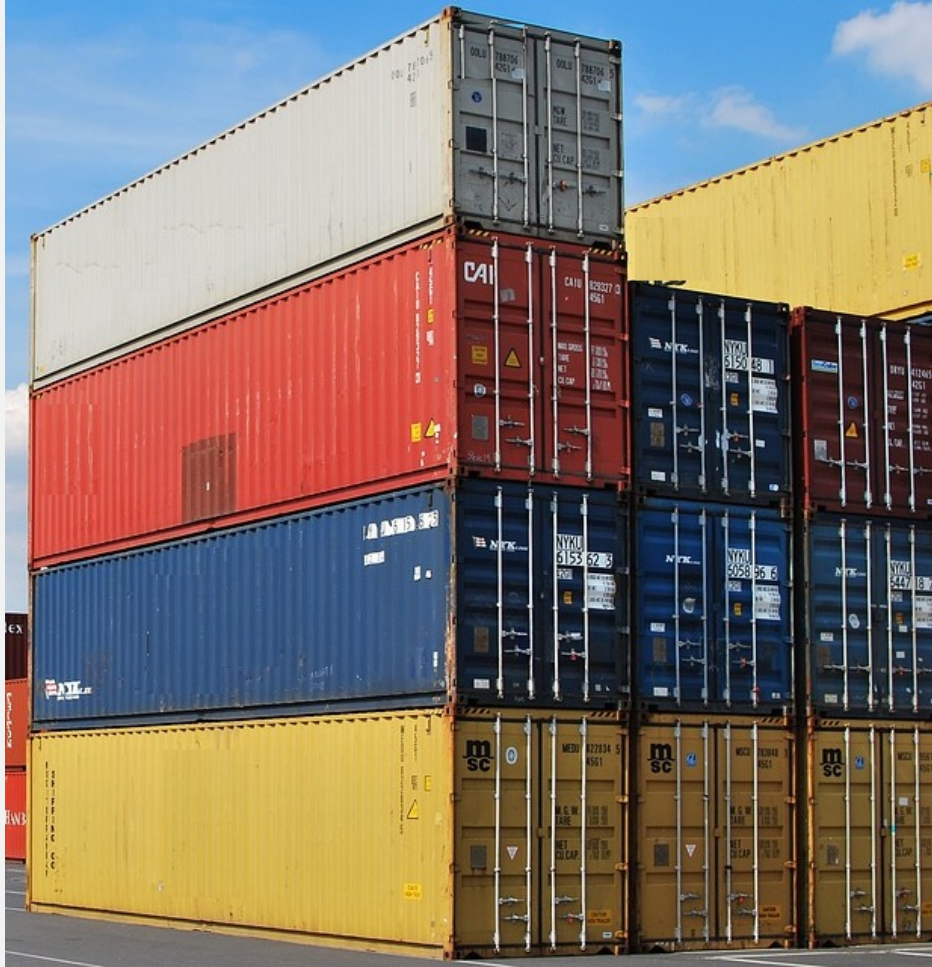
- 컨테이너를 왜 사용하는가?
- ECR 을 왜 사용하는가?
- EKS, ECS 를 왜 사용하는가?
- Fargate 를 왜 사용하는가?
- Kubernetes Object



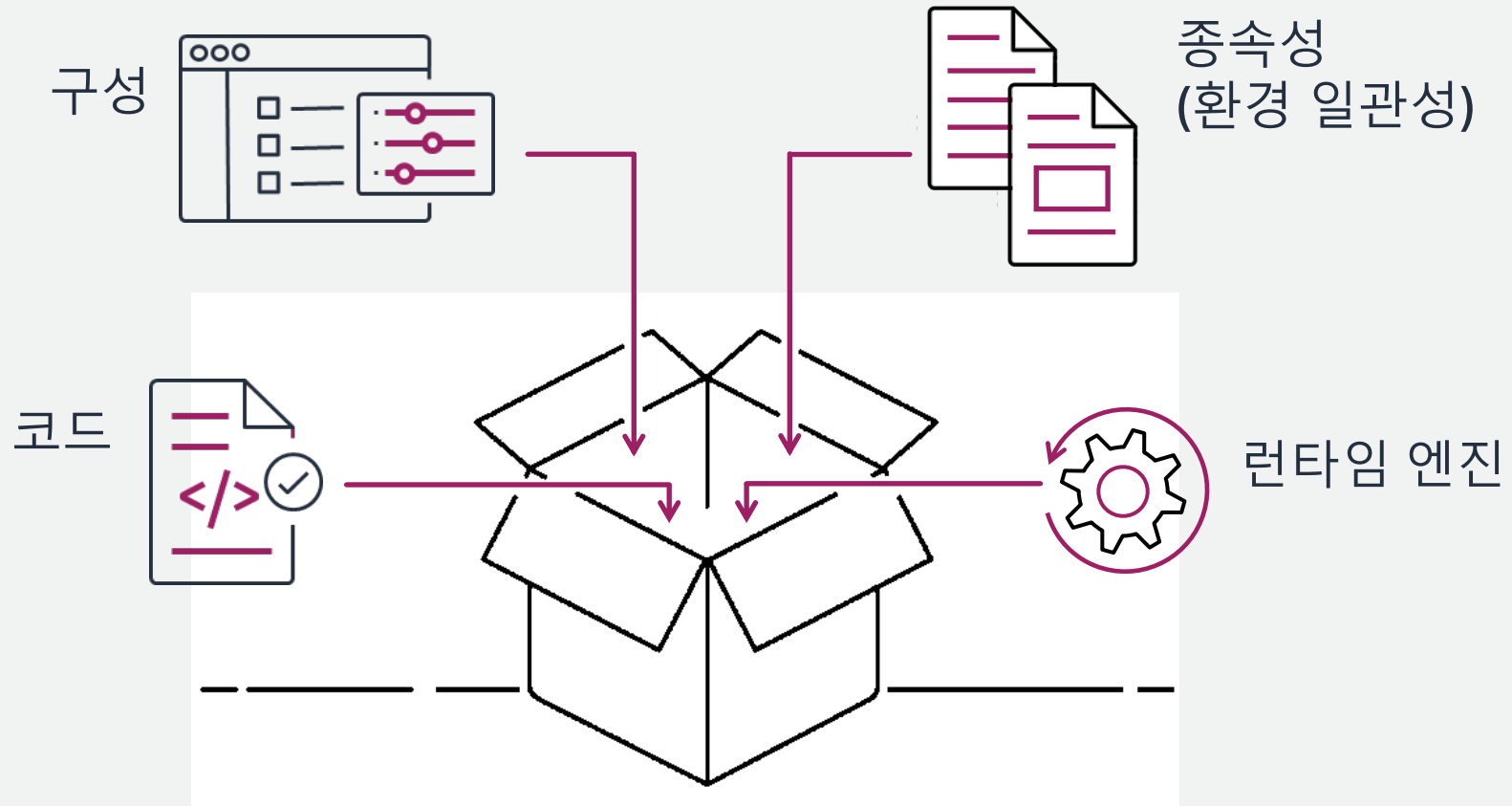
# 표준화된 스토리지 단위



# 컨테이너로 운송



# 컨테이너란 무엇입니까?



# 컨테이너의 특성:

- 반복 가능
- 독립적 환경
- 민첩성 - VM보다 더 빠른 가동/중단 속도
- 이동성
- 확장성

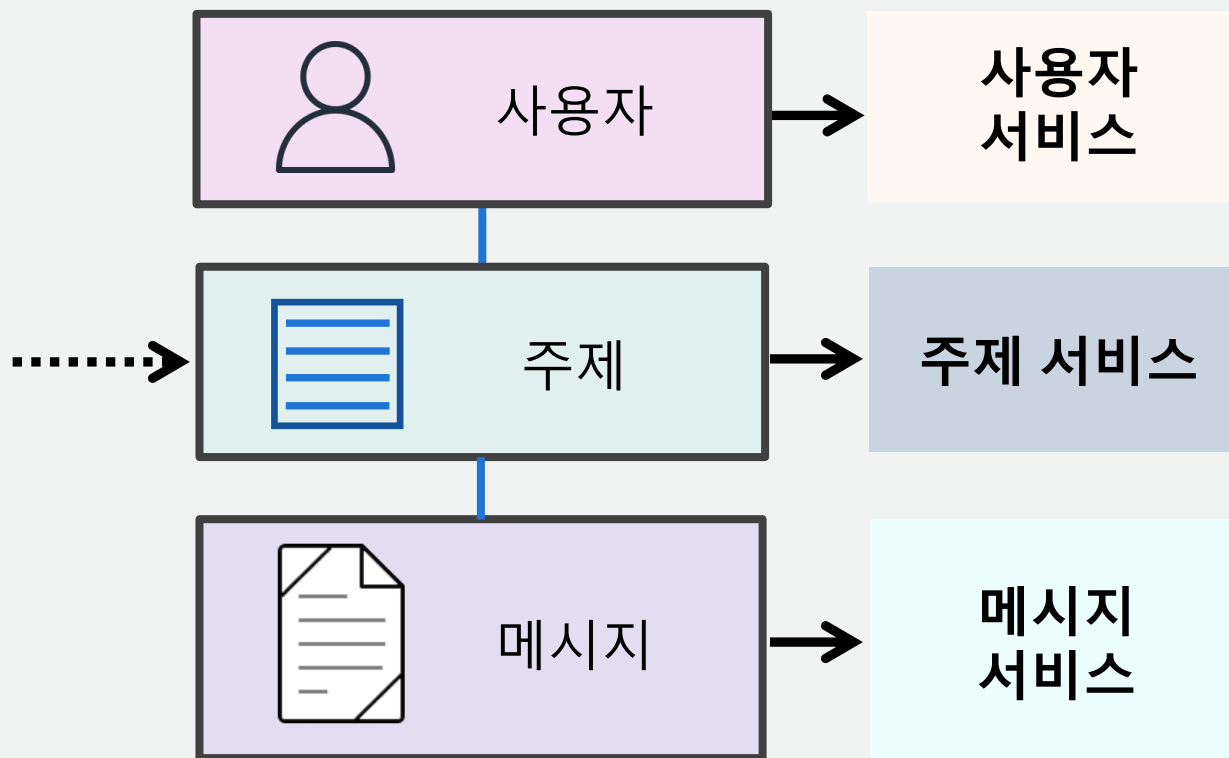


# 마이크로서비스

모놀리식 애플리케이션

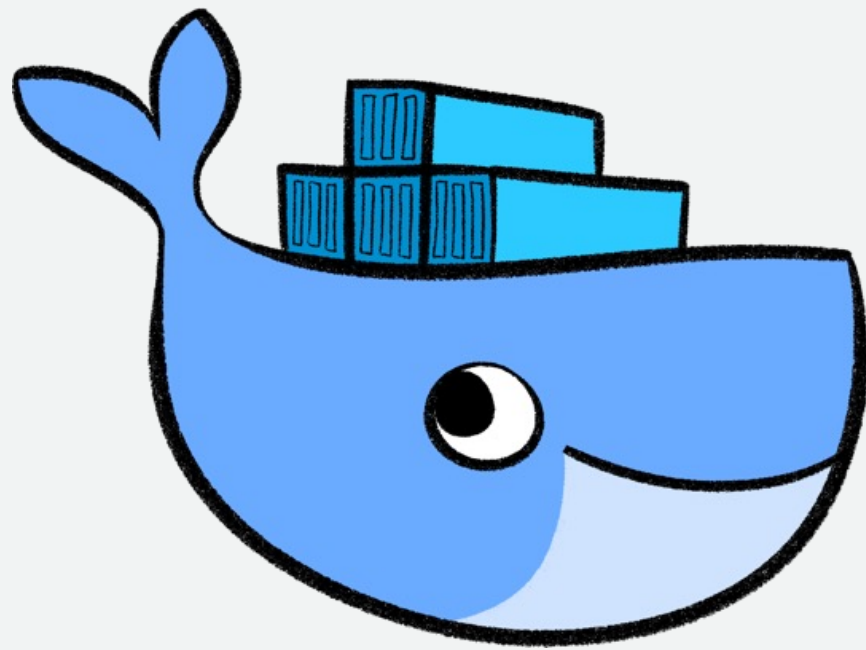


마이크로서비스 애플리케이션



# 가상화 플랫폼인 Docker

- 경량 컨테이너 가상화 플랫폼
- 컨테이너를 생성, 저장, 관리 및 실행할 수 있는 도구 제공
- 자동화된 구축, 테스트 및 배포 파이프라인과 쉽게 통합



# 예: Dockerfile

```
# Use an official Python runtime as a base image
FROM python:3.7-alpine

# Set the working directory to /app
WORKDIR /app

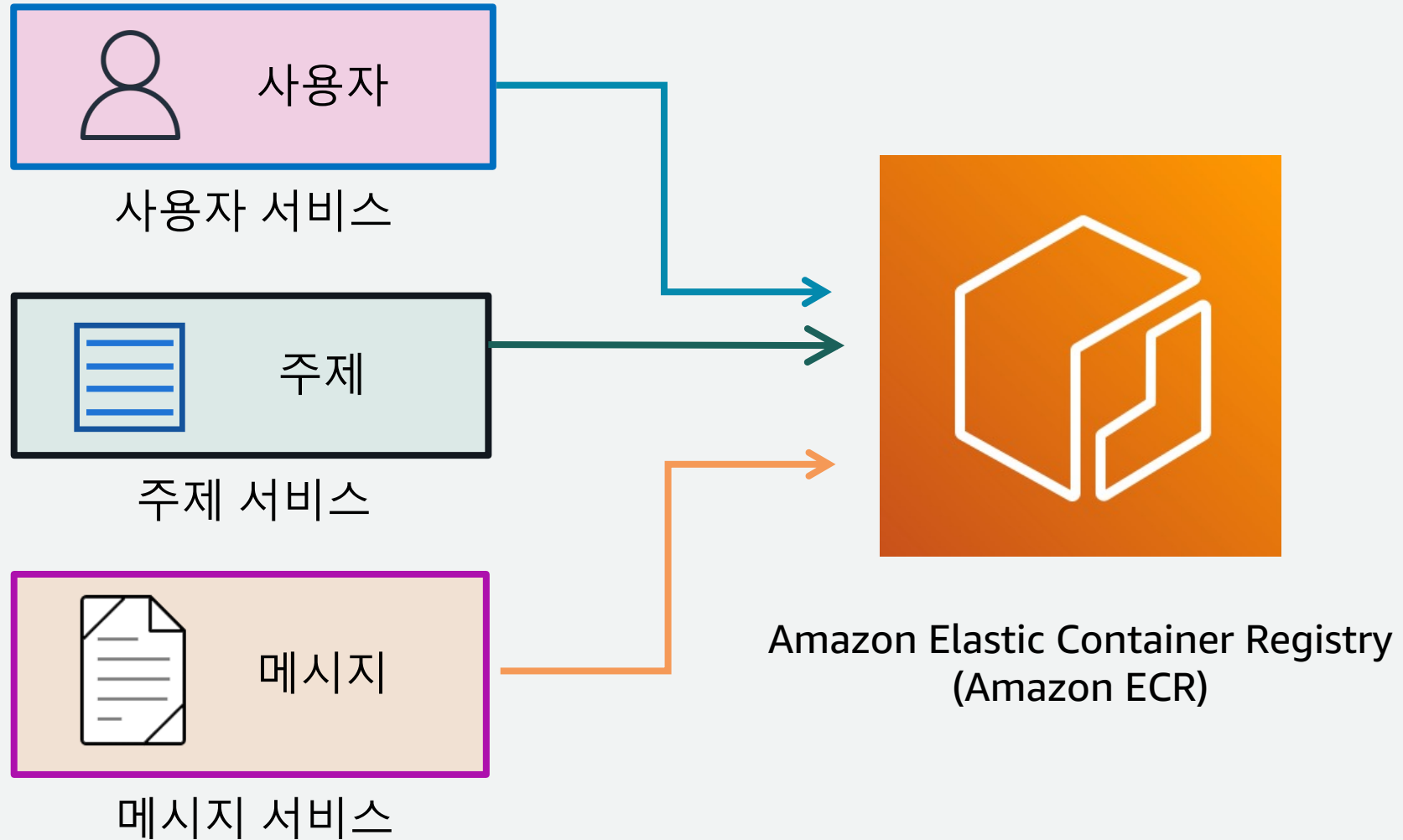
# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Run gunicorn when the container launches
CMD gunicorn -w 4 app:app -b 0.0.0.0:80
```

# Amazon ECR



# 하나의 컨테이너로 시작



# 수백 개의 호스트에 수천 개의 컨테이너



# 오케스트레이션 도구 필요



Amazon Elastic Container Service  
(Amazon ECS)



Amazon Elastic Kubernetes Service  
(Amazon EKS)

# Kubernetes 컨트롤 플레인 과 데이터 플레인

## Kubernetes Cluster





# EKS 컨트롤 플레인 과 데이터 플레인

## EKS Cluster



# EKS 컨트롤 플레인 과 데이터 플레인

EKS Cluster

컨트롤 플레인



Amazon EKS

완전 자동 관리

데이터 플레인



AWS Fargate

완전 자동 관리

# 컨테이너에 적합한 서버리스 컴퓨팅



AWS Fargate

- 기본 (Native)
- 적정 크기
- 빠르고 간편
- 완전 관리형
- 최적화

# AWS 관리형 컨테이너 서비스로 배포

## 관리형 컨테이너 서비스

오케스트레이션 도구 선택

Amazon ECS



Amazon EKS



시작 유형 선택

Amazon EC2



Fargate



Amazon EC2



Fargate



# Kubernetes Object



# Pod(파드)

쿠버네티스에서 생성하고 관리할 수 있는 배포 가능한 가장 작은 컴퓨팅 단위  
(하나 이상의 컨테이너 그룹)

```
apiVersion: v1
kind: Pod
metadata:
  name: pod-example
spec:
  containers:
  - name: ubuntu
    image: ubuntu:18.04
[...]
```

# ReplicaSet(레플리카셋)

파드 집합의 실행을 항상 안정적으로 유지하는 것이다. 이처럼 레플리카셋은 보통 명시된 동일 파드 개수에 대한 가용성을 보증하는데 사용

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: ubuntu-replicaset
spec:
  replicas: 3
  selector:
    matchLabels:
      app: ubuntu
  template:
    metadata:
      labels:
        app: ubuntu
```

```
spec:
  containers:
    - name: ubuntu
      image: Ubuntu:latest
      ports:
        - containerPort: 80
```

# Deployment(디플로이먼트)

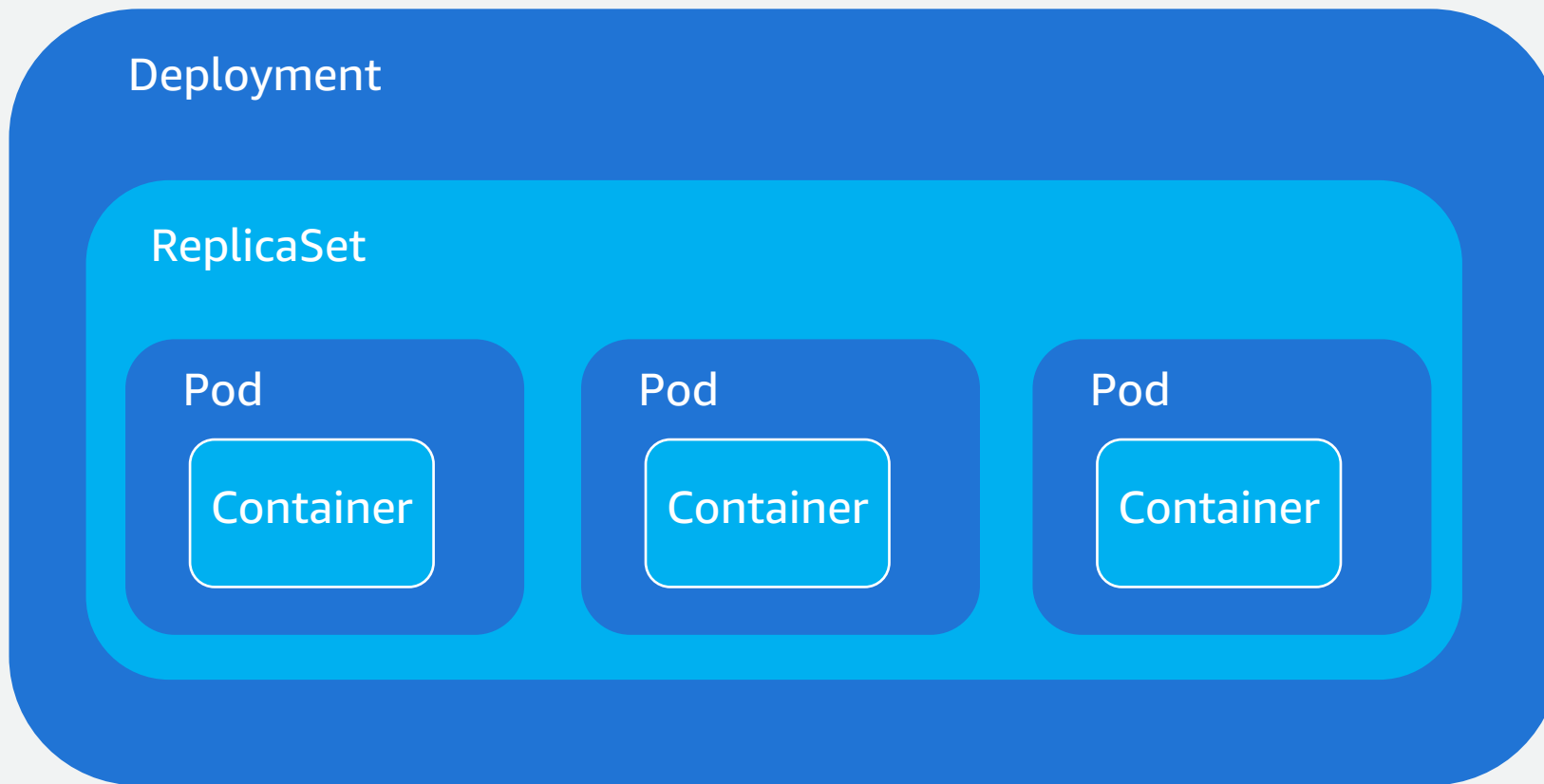
파드와 레플리카셋(ReplicaSet)에 대한 선언적 업데이트를 제공

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ubuntu-deployment
spec:
  replicas: 3
  selector:
    matchLabels:
      app: ubuntu
  template:
    metadata:
      labels:
        app: ubuntu
```

```
spec:
  containers:
    - name: ubuntu
      image: Ubuntu:latest
      ports:
        - containerPort: 80
```



# 디플로이먼트 구조

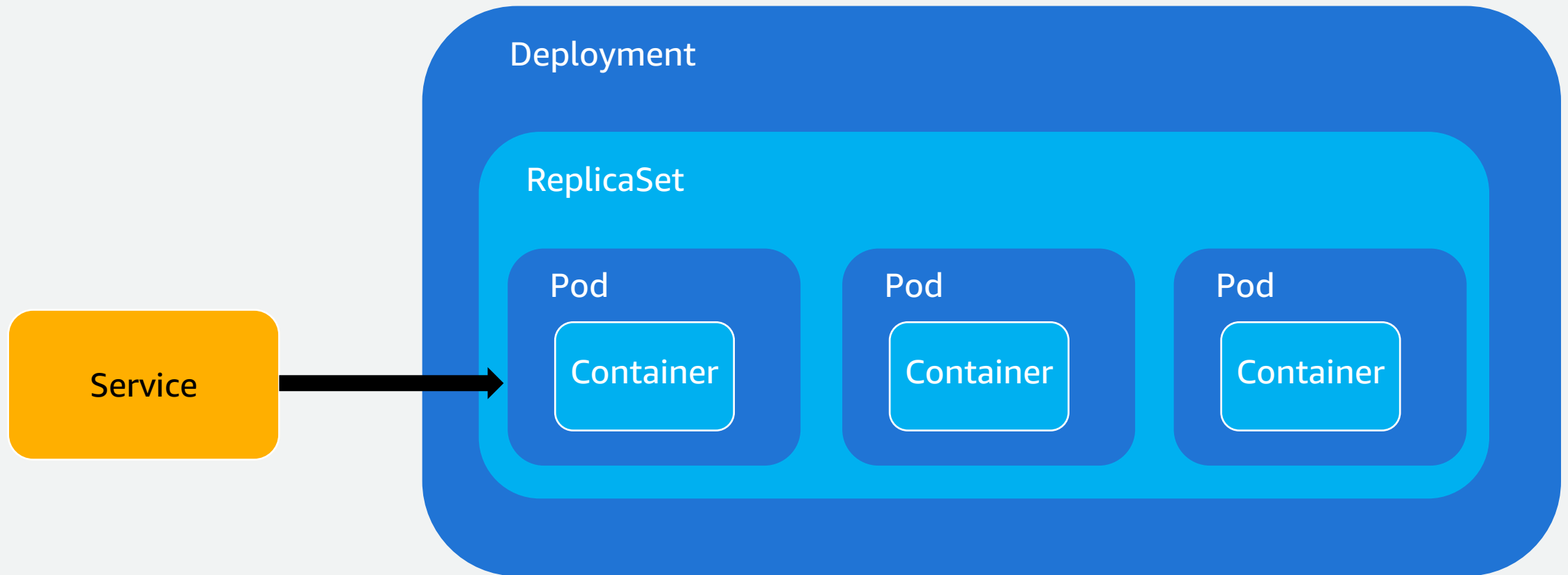


# Service(서비스)

파드 집합에서 실행중인 애플리케이션을 네트워크 서비스로 노출하는 추상화 방법

```
apiVersion: v1
kind: Service
metadata:
  name: ubuntu-service
spec:
  type: LoadBalancer
  ports:
    - port: 80
      targetPort: 80
      protocol: TCP
  selector:
    matchLabels:
      app: ubuntu
```

# 서비스의 구조



# 실습 안내



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

# 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 여러분의 소중한 의견을 부탁드립니다.

감사합니다.



# 크레딧 안내

- AWS 계정으로 시작하실 경우, **금일 실습에서 발생하는 비용은 당월 과금이 되는 점** 미리 확인 부탁드립니다.
- 웨비나 종료 후 **설문 조사에 참여해주신 분들께는 AWS 크레딧 바우처** (1인당 \$50 USD 크레딧, 전체 세션당 1회 제공)를 드립니다.
- 해당 **AWS 크레딧**은 등록하신 이메일 계정으로 **행사 종료 후 1개월 내** 발송 드릴 예정이며, 전달 받은 AWS 크레딧은 바로 사용 가능합니다.

# 감사 메일 & 참석 증명서

- AWS Builders Korea 세션에 참석해 주신 분들께 행사 종료 후 1개월 내 감사메일과 참석 증명서가 순차 발송됩니다.
- 등록 진행 후 참석하지 않으실 경우 별도 메일 및 증명서는 발급되지 않습니다.

## 감사 메일 예시

**AWS Builders Korea Program 온라인 세미나에  
참석해 주셔서 감사합니다.**

AWS Builders Korea Program에 참석하고 피드백을 공유해주셔서  
감사드립니다. 세미나 자료는 아래 링크를 통해 확인하실 수 있습니다.

[자료 확인하기](#)

## 참석 증명서 예시

### 참석 증명서

AWS Builders Korea Program에 참석해 주셔서 감사합니다.

홍길동

2023년 3월 20일 - 3월 24일





# 강연 다시보기

aws builders korea program 다시보기



<https://kr-resources.awscloud.com/aws-builders-korea-program>

## AWS Builders Korea 프로그램 정보

aws builders korea program



<https://aws.amazon.com/ko/events/seminars/aws-builders/>





# Thank you!

방신철

shinchul@amazon.com