



Running Containers on Amazon EKS

## 모듈 2:

### Amazon Elastic Kubernetes Service(Amazon EKS) 기본 사항



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



#### ~Dev Notes

~This module was pulled from ....

~

| 강사용 노트

| 이 모듈을 완료하는 데는 약 **1시간 15분** 정도 걸립니다. 이 모듈을 마치면 곧바로 실습 **1**을 진행해야 합니다.



Running Containers on Amazon EKS

## 모듈 2 개요

- Amazon EKS 소개
- Amazon EKS 제어 영역
- Amazon EKS 데이터 영역
- 2개의 API: Kubernetes 및 Amazon EKS
- 실습 1 준비



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.



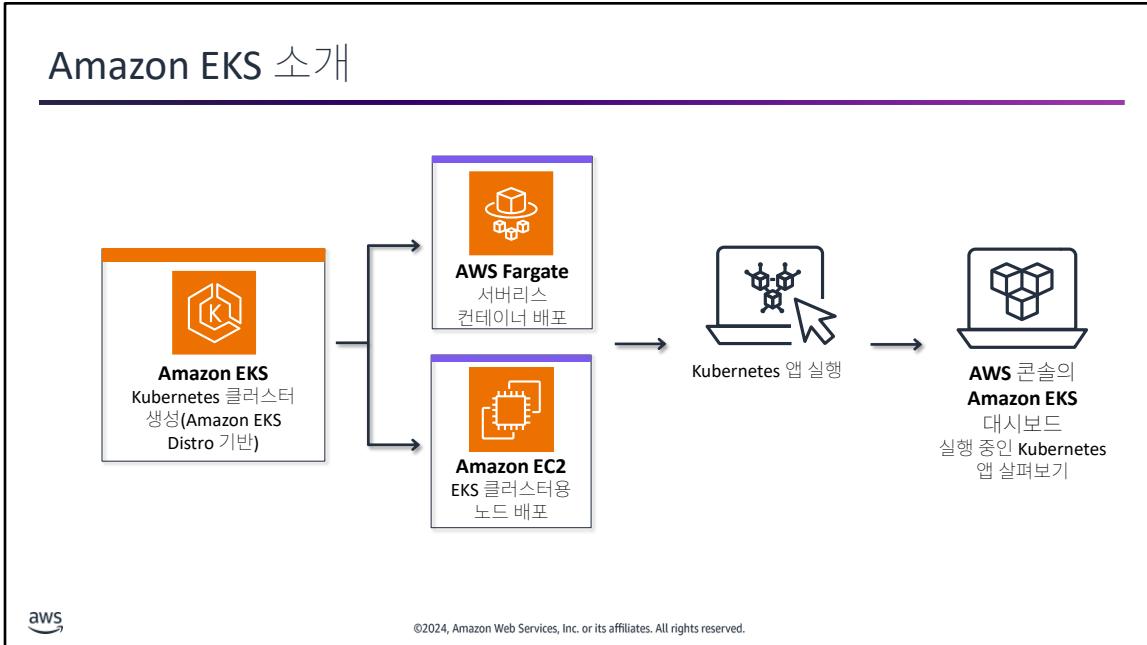
Running Containers on Amazon EKS

# Amazon EKS 소개



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## Amazon EKS 소개



### | 수강생용 노트

이전 모듈에서는 **Kubernetes**가 컨테이너 워크로드를 관리하는 효과적인 방법임을 알아보았습니다.

어렵지는 않지만 **Kubernetes**를 시작하고 실행하는 것은 버거울 수 있습니다. 특히 프로덕션 배포를 위해 고가용성 제어 영역을 수동으로 구성할 때 더욱 그렇습니다. AWS는 개발자가 **Kubernetes** 컨테이너를 실행하는 데 도움이 되도록 **Amazon EKS**를 개발했습니다.

**Amazon EKS**는 AWS에서 **Kubernetes**를 사용하여 컨테이너식 애플리케이션을 배포, 관리, 크기 조정할 수 있는 관리형 **Kubernetes** 제어 영역 서비스입니다. **Kubernetes** 제어 영역은 **Kubernetes**가 클러스터와 통신하는 방식을 관리한다는 점을 기억하십시오. 데이터 영역에서 컨테이너가 실행됩니다. **Amazon EKS**는 **Kubernetes** 클러스터를 위한 제어 영역을 생성하고 관리합니다. **Amazon EKS**는 여러 AWS 가용 영역에 걸쳐 **Kubernetes** 관리 인프라를 실행하여 단일 장애 지점을 제거합니다.

**Amazon EKS**는 제어 영역을 관리하는 것 외에도 사용자가 원할 경우 데이터 영역(노드)의 요소를 관리할 수 있습니다.

**Amazon EKS**는 다음과 같은 다른 AWS 서비스와 긴밀하게 통합되어 있습니다.

- 로드 분산을 위한 **Application Load Balancer**
- 역할 기반 액세스 제어를 위한 **AWS Identity and Access Management(IAM)**
- Pod 네트워킹을 위한 **Amazon Virtual Private Cloud(Amazon VPC)**

AWS는 Amazon EKS 사용자가 AWS 서비스와 기능을 활용하는 데 도움이 되는 **Kubernetes** 코드 베이스에 기여하는 등 **Kubernetes** 커뮤니티와 적극적으로 협력하고 있습니다.

Amazon EKS는 네이티브 업스트림 **Kubernetes**를 실행합니다. Amazon EKS는 공인 **Kubernetes** 준수 서비스이므로 **Kubernetes** 커뮤니티의 기존 플러그 인과 도구를 모두 사용할 수 있습니다. Amazon EKS에서 실행되는 애플리케이션은 모든 표준 **Kubernetes** 환경에서 실행되는 애플리케이션과 완벽하게 호환됩니다. 온프레미스 데이터 센터에서 실행되든 퍼블릭 또는 프라이빗 클라우드에서 실행되든 마찬가지입니다. 즉, 코드를 리팩터링하지 않고 표준 **Kubernetes** 애플리케이션을 Amazon EKS로 마이그레이션할 수 있습니다.



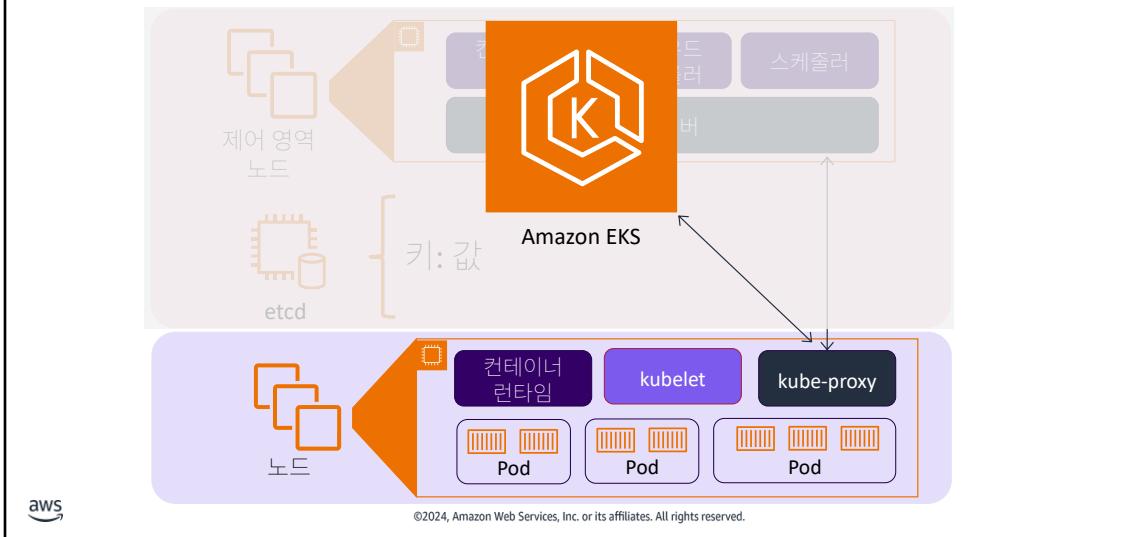
Running Containers on Amazon EKS

# Amazon EKS 제어 영역



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## Amazon EKS: 관리형 Kubernetes 제어 영역



### ~ALT text

~데이터 영역: 여러 Pod와 함께 kube-proxy, 컨테이너 런타임, kubelet을 포함하는 노드가 있는 데이터 영역.

~제어 영역: 컨트롤러 관리자, 클라우드 컨트롤러, 스케줄러, API 서버 및 etcd로 구성된 제어 영역 노드가 있는 제어 영역.

~EKS: 제어 영역에서 Amazon EKS 작업 인수를 보여 주는 애니메이션.

| 강사용 노트 및 애니메이션

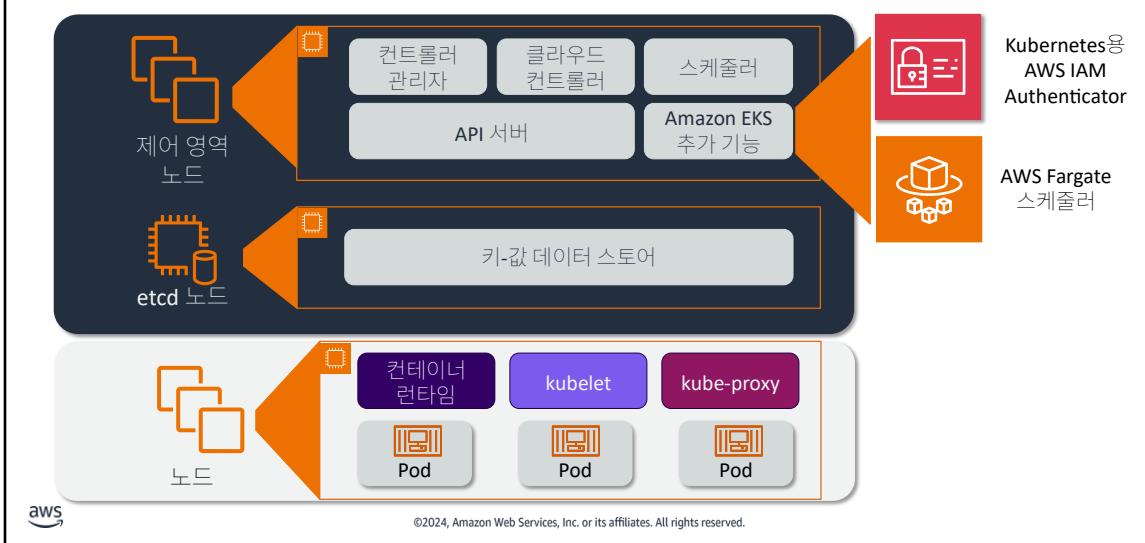
| <1> 제어 영역이 회색으로 표시되고 Amazon EKS로 대체됩니다.

| 수강생용 노트

표준 Kubernetes 배포에서 사용자는 제어 영역의 모든 요소를 설계, 구현, 유지 관리하고 노드를 관리할 책임이 있습니다. 이 모든 관리는 Pod에서 실행되는 애플리케이션을 생성하고 컨테이너화하는 작업에 추가됩니다.

Amazon EKS는 확장 가능하고 가용성이 뛰어난 제어 영역을 제공합니다. Amazon EKS는 각 클러스터의 etcd 지속성 계층과 Kubernetes API 서버의 가용성과 확장성을 자동으로 관리합니다. 이렇게 하면 데이터 영역과 Pod를 실행하는 데 더 많은 시간을 집중할 수 있습니다.

## Amazon EKS 제어 영역 내부



### | 수강생용 노트

Kubernetes 제어 영역 구성 요소 외에도 Amazon EKS 관리형 제어 영역에는 Kubernetes 제어 영역에 대한 몇 가지 추가 기능이 포함되어 있습니다. 여기에는 다음이 포함됩니다.

- Kubernetes用 AWS IAM Authenticator
- AWS Fargate 스케줄러
- OpenID Connect(OIDC) 발급자 URL(슬라이드에는 표시되지 않음)

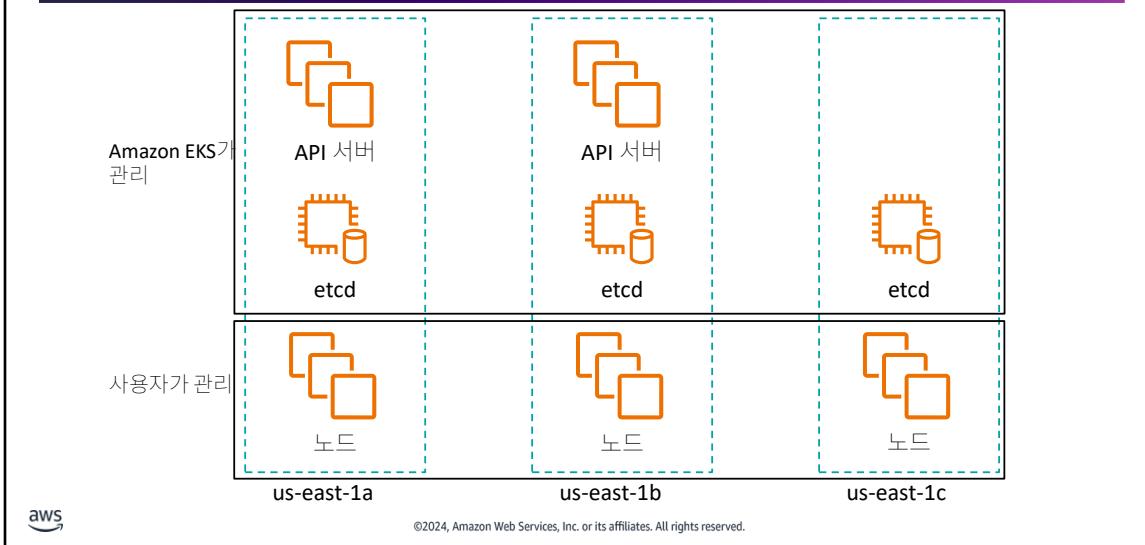
이러한 확장에 대한 자세한 내용은 이 과정의 후반부에서 살펴보겠습니다.

Kubernetes 데이터 영역을 호스팅하는 방법을 선택할 수 있습니다.

- Amazon EC2 인스턴스를 사용하는 셀프 매니지드 노드
- Kubernetes 노드 소프트웨어에 대한 관리형 업그레이드를 제공하고 Amazon EC2 Auto Scaling 그룹을 사용하여 Amazon EC2 인스턴스를 관리하는 Amazon EKS 관리형 노드 그룹
- 노드당 1개 Pod의 서비스 호스팅을 위한 AWS Fargate

이 모듈의 뒷부분에서 데이터 영역에 대해 자세히 알아봅니다.

## 가용성 유지 관리



### ~ALT text

~여러 가용 영역에 걸친 제어 영역 및 여러 노드를 나타내는 아키텍처로 노트에 자세히 설명되어 있습니다.

### |수강생용 노트

Amazon EKS는 Kubernetes 제어 영역을 실행합니다. 제어 영역은 3개의 가용 영역에 걸친 최소 2개의 API 서버(제어 영역) 노드와 3개의 etcd 노드로 구성됩니다. Amazon EKS는 비정상 제어 영역 노드를 자동으로 탐지하여 교체하므로 Kubernetes 실행에 따른 운영 부담이 대폭 제거됩니다. 이 기능을 사용하면 AWS 인프라 관리 대신 애플리케이션 구축에 집중할 수 있습니다.

Amazon EKS를 시작하려면 노드 클러스터를 프로비저닝합니다. 클러스터를 프로비저닝하는 방법에 대한 자세한 내용은 이후 모듈에서 알아봅니다. Amazon EKS는 가용성이 높고 안전한 구성으로 Kubernetes 제어 영역의 프로비저닝, 크기 조정, 관리를 처리합니다. 그러면 사용자는 그래픽 또는 명령줄 인터페이스(CLI)를 사용하여 Amazon EKS 클러스터에 연결합니다. Amazon EKS 클러스터에 연결한 후에는 Kubernetes 애플리케이션을 Amazon EKS 클러스터에 배포할 수 있습니다. 다른 Kubernetes 환경에서와 동일한 방식으로 이 작업을 수행할 수 있습니다.

## 지식 확인 1

항상 Amazon EKS가 관리하는  
구성 요소는 무엇입니까?

보기	응답
A.	Kubernetes 제어 영역
B.	Kubernetes 데이터 영역
C.	Kubernetes 컨테이너
D.	Kubernetes 서비스



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## 지식 확인 1: 정답은 A입니다.

항상 Amazon EKS가 관리하는  
구성 요소는 무엇입니까?

보기  
정답

A.  
B.  
C.  
D.

Kubernetes 제어 영역

Kubernetes 데이터 영역

Kubernetes 컨테이너

Kubernetes 서비스



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### | 수강생용 노트

정답은 A입니다. Amazon EKS는 제어 영역을 관리합니다. 사용자는 데이터 영역의 노드를 관리합니다.

## 지식 확인 2

Amazon EKS를 사용할 때 etcd의  
가용성 관리는 누가  
담당합니까?

보기	응답
A.	Amazon EKS
B.	Kubernetes
C.	사용자



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## 지식 확인 2: 정답은 A입니다.

Amazon EKS를 사용할 때 etcd의  
가용성 관리는 누가  
담당합니까?

- | 보기       | 응답         |
|----------|------------|
| A.<br>정답 | Amazon EKS |
| B.       | Kubernetes |
| C.       | 사용자        |

 ©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### | 수강생용 노트

정답은 A입니다. Amazon EKS는 제어 영역을 관리합니다. 관리에는 **etcd** 지속성 계층 외에 **API 서버**(제어 영역) 노드가 포함됩니다. 사용자는 데이터 영역의 노드를 관리합니다.



Running Containers on Amazon EKS

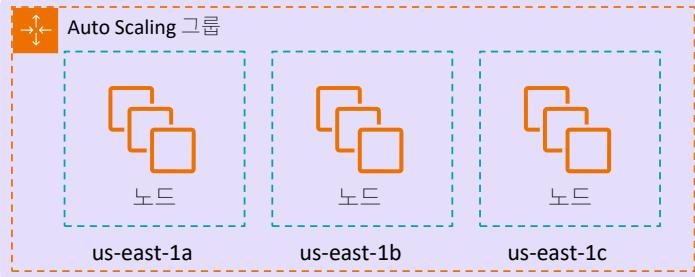
# Amazon EKS 데이터 영역



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

# Amazon EKS 데이터 영역 관리

Amazon EKS 제어 영역



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## ~ALT text

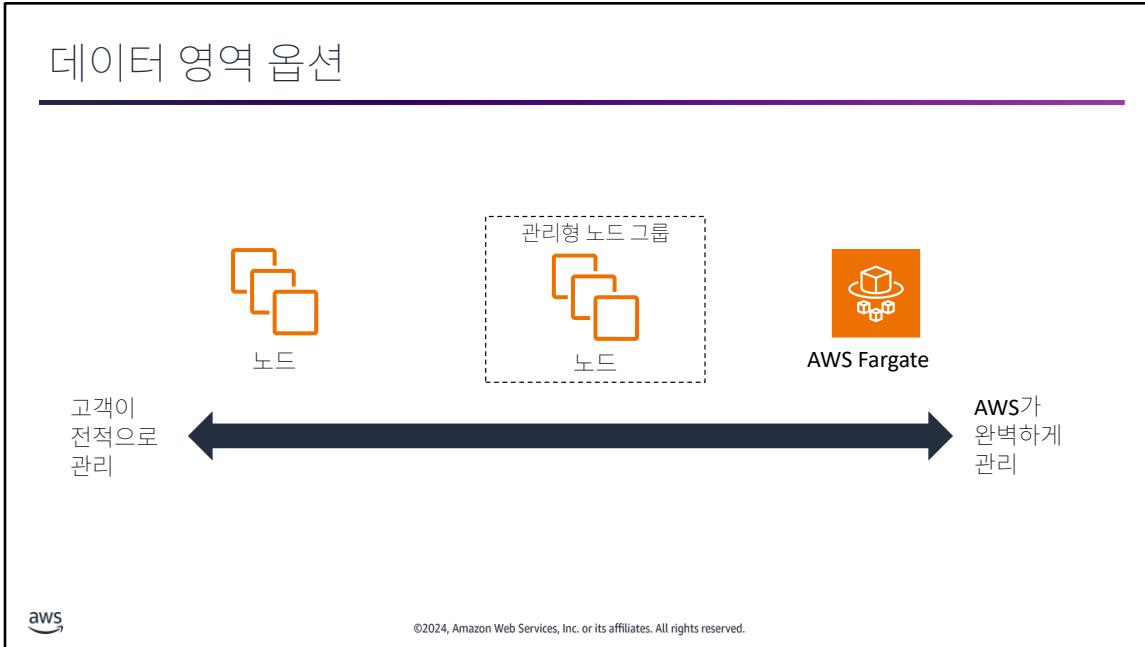
~Auto Scaling 그룹의 노드가 있는 EKS API 및 제어 영역을 나타내는 아키텍처로 노트에 자세히 설명되어 있습니다.

### | 수강생용 노트

Amazon EKS로 데이터 영역을 관리하는 이유는 무엇입니까?

노드가 많은 복잡한 인프라를 관리하고 자동 크기 조정 및 업데이트를 신경 써야 하는 일은 쉽지 않습니다. 또한 클러스터에 노드를 프로비저닝하는 여러 팀이 있을 수 있으며, 이들의 프로비저닝 방식은 각각 다를 수 있습니다. 이러한 차이로 인해 표준화가 어려워집니다. Amazon EKS를 통해 데이터 영역의 일부 또는 전부를 관리하면 인프라를 간소화하고 표준화를 유지할 수 있습니다.

## 데이터 영역 옵션



### ~ALT text

~Amazon EKS가 데이터 영역을 관리하도록 허용할 수 있는 범위를 보여주는 그래픽으로 노트에 자세히 설명되어 있습니다.

#### |수강생용 노트

Kubernetes 제어 영역은 Amazon EKS가 관리하지만 Amazon EKS가 데이터 영역을 관리하는 정도는 사용자가 제어합니다. Amazon EKS를 사용하여 다음 3가지 유형의 노드를 관리할 수 있습니다.

- 전적으로 사용자가 관리하는 셀프 매니지드 노드
- Amazon EKS가 부분적으로 관리하고 사용자가 리소스에 대한 제어를 할 수 있는 관리형 노드 그룹
- Amazon EKS에서 전적으로 관리하는 Fargate 노드

인프라 관리를 줄이고 싶다면 모든 Pod가 Fargate에서 실행되도록 구성할 수 있습니다. 하지만 유연성을 발휘할 수도 있습니다. Fargate에서 잘 작동하는 Pod도 있지만 특정 요구 사항이 있는 Pod도 있습니다. 이러한 경우에는 노드에서의 Pod 배치에 대한 더 많은 제어가 필요합니다. 각 Pod에 가장 적합한 방법을 선택할 수 있습니다.

다음으로 관리형 노드 그룹과 Fargate 노드를 자세히 살펴보겠습니다.

## 관리형 노드 그룹

- 프로비저닝을 지원
- 관리를 지원
- 업데이트를 지원
- 크기 조정 제어
- eksctl과 함께 작동



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### ~ALT text

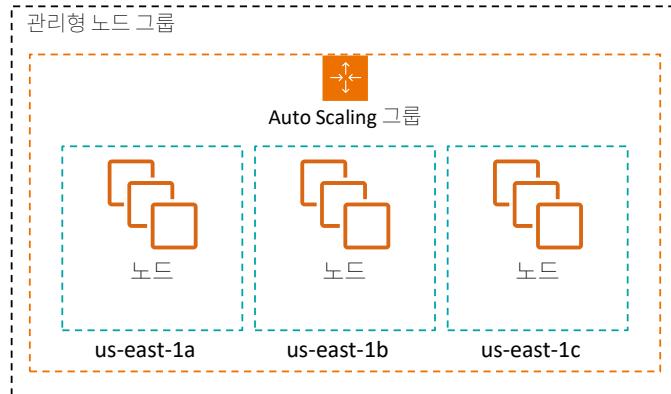
~노드를 관리하기 위해 Amazon EKS API를 사용하는 제어 영역 및 관리형 노드 그룹입니다.

#### | 수강생용 노트

관리형 노드 그룹은 Amazon EKS API를 사용하여 Amazon EKS 클러스터용 컨테이너를 실행하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스를 시작하고 관리합니다. 관리형 노드 그룹은 자동으로 시작되고 관리되지만 EC2 인스턴스 및 Auto Scaling 그룹과 같이 AWS 계정에서 사용 중인 모든 리소스는 계속 볼 수 있습니다. 작업은 줄이면서 제어, 보안, 가시성은 모두 확보할 수 있습니다.

- 프로비저닝을 지원 - Amazon EKS에 최적화되고 Auto Scaling 그룹이 지원하는 최신 Amazon Machine Image(AMI)를 사용하여 Multi-AZ 노드 그룹을 모두 클릭 한 번으로 배포할 수 있습니다.
- 관리를 지원 - Amazon EKS는 관리형 노드 그룹의 상태 모니터링을 처리합니다.
- 업데이트를 지원 - 하나의 명령으로 관리형 노드 그룹을 클러스터 버전에 맞는 최신 AMI 버전으로 업데이트할 수 있습니다. 원하는 시점에 자유롭게 업데이트할 수 있으면서도 Amazon EKS가 업데이트를 대신 처리해 주는 이점을 누릴 수 있습니다.
- 크기 조정 제어 - 관리형 노드 그룹은 노드 크기 조정을 처리합니다. 하지만 Kubernetes 레이블, AWS 태그, 노드 그룹의 크기와 같은 크기 조정 파라미터는 사용자가 여전히 제어할 수 있습니다.
- eksctl과 함께 작동 - eksctl을 사용하여 관리형 노드 그룹을 프로비저닝할 수 있습니다.

## 관리형 노드 그룹 배포



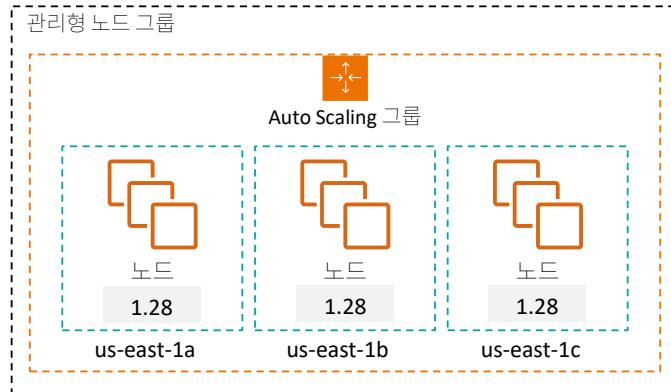
### ~ALT text

~다른 가용 영역에 Auto Scaling 그룹 및 노드가 있는 관리형 노드 그룹을 나타내는 아키텍처입니다.

### |수강생용 노트

하나의 명령으로 관리형 노드 그룹을 배포할 수 있습니다. 그러면 Amazon EKS가 Amazon EKS에 최적화된 최신 AMI를 사용하여 노드를 생성합니다. AWS 서비스는 이를 여러 가용 영역에 배포하고 Auto Scaling 그룹으로 지원합니다. 크기 조정 파라미터를 변경할 수 있습니다.

# 관리형 노드 그룹 업데이트



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## ~ALT text

~노드에서 롤링 업데이트를 수행하는 관리형 노드 그룹을 나타내는 아키텍처로 노트에 자세히 설명되어 있습니다.

| 강사용 노트 및 애니메이션

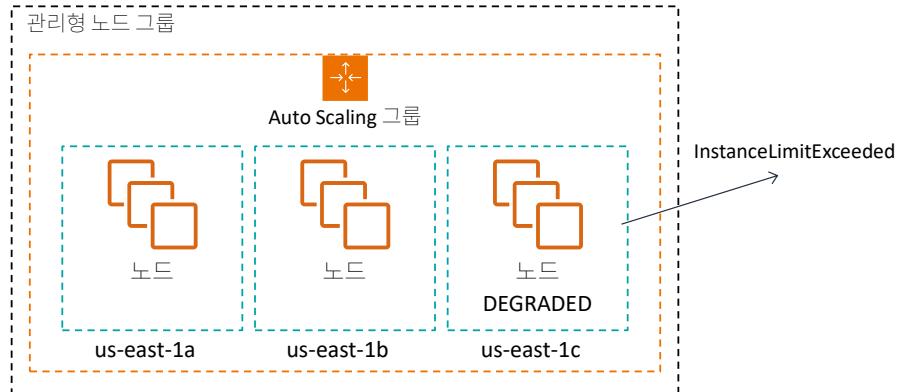
| <1> 각 노드는 1.27 버전으로 종료되고 1.28 버전으로 반환됩니다.

| 수강생용 노트

하나의 명령으로 원하는 시점에 관리형 노드 그룹을 업데이트할 수 있습니다. **Amazon EKS**는 롤링 업데이트 시 노드를 종료하고 **Kubernetes** 버전에 맞는 최신 **AMI** 버전으로 자동 업데이트합니다. 노드 그룹이 제어 영역(**1.28**)과 동일한 **Kubernetes** 버전인 경우 최신 **AMI** 릴리스 버전으로 업데이트할 수 있습니다. **AMI** 릴리스 버전은 사용 중인 **AMI** 유형의 **Kubernetes** 버전(**1.28**)에 대한 버전입니다.

노드 그룹이 클러스터(**1.27**)보다 이전 버전의 **Kubernetes** 버전을 실행 중인 경우 노드 그룹을 최신 **AMI** 릴리스 버전으로 업데이트할 수 있습니다. 이 **AMI** 릴리스 버전은 노드 그룹의 **Kubernetes** 버전(**1.28**)과 일치하는 버전입니다. 또는 제어 영역의 **Kubernetes** 버전(**1.28**)과 일치하는 최신 **AMI** 릴리스 버전으로 업데이트할 수 있습니다.

## 관리형 노드 그룹 모니터링



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### ~ALT text

~노드를 제어하는 Auto Scaling 그룹이 있는 관리형 노드 그룹을 나타내는 아키텍처입니다. 한 그룹은 DEGRADED 레이블이 지정되어 있습니다.

#### | 수강생용 노트

Amazon EKS는 관리형 노드 그룹의 상태를 지속적으로 모니터링합니다. Amazon EKS는 삭제 중이거나 연결할 수 없거나 사용할 수 없는 필수 리소스를 포함한 문제를 사용자에게 자동으로 알려줍니다. Amazon EKS는 업데이트 문제, 한도 초과, 생성 또는 삭제 실패에 대해서도 알려줍니다.

상태 문제가 있는 관리형 노드 그룹은 DEGRADED로 label이 지정됩니다. 관리형 노드 그룹을 사용하면 노드 수준의 Secure Shell(SSH) 액세스, 오픈 소스 로그 라우터 또는 Amazon CloudWatch에서 로그를 가져올 수도 있습니다. 모든 관리형 노드 그룹 이벤트도 AWS CloudTrail에 기록됩니다.

## AWS Fargate

- 기본
- 적정 규모
- 빠르고 간단함
- 투명함
- 최적화됨



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### ~ALT text

~AWS Fargate가 관리하는 데이터 영역입니다.

#### | 수강생용 노트

관리형 노드 그룹을 사용하면 인프라 관리에 쓰는 시간이 줄어듭니다. 하지만 애플리케이션 생성에만 집중하고 **Amazon EKS**가 데이터 영역을 완전히 관리하도록 하는 것이 좋습니다. 이는 **AWS Fargate**에서 **Pod**를 실행하면 가능합니다.

**AWS Fargate**는 **Kubernetes** 데이터 영역의 전체 인프라를 관리합니다. **Pod** 실행에 대해서만 신경 쓰면 됩니다.

- 네이티브 - **AWS Fargate**는 네이티브 **Kubernetes Pod**를 실행합니다. **AWS**를 위해 변경하거나 구성할 필요가 없습니다.
- 적정 규모 - **AWS Fargate**는 **Pod** 및 리소스에 필요한 리소스를 동적으로 프로비저닝하며 그 이상도 그 이하도 아닙니다.
- 빠르고 간단함 - **AWS Fargate**는 빠르게 크기를 조정할 수 있습니다. **Cluster Autoscaler**를 설정할 필요가 없습니다.
- 최적화 - 실행한 **Pod** 비용만 지불하고 **Pod** 수준에서 청구를 확인할 수 있습니다.

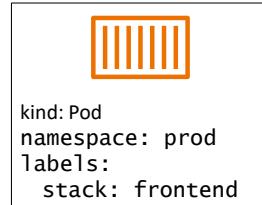
**AWS Fargate**에서 실행되는 **Pod**는 **EC2** 인스턴스에서 실행되는 **Pod**와 동일하므로 **PodSpec** 파일을 **AWS Fargate**로 이동하기 위해 변경할 필요가 없을 수 있습니다. **PodSpec** 내의 가상 **CPU(vCPU)** 및 메모리 예약은 **Fargate**가 **Pod**에 프로비저닝하는 **CPU** 및 메모리 양을 결정합니다. **Amazon EKS**는 버전 및 업데이트도 관리합니다.

**Fargate**에서 실행되는 각 **Pod**는 20GB의 컨테이너 이미지 계층 스토리지를 수신합니다. **Pod** 스토리지는 휘발성입니다. **Pod**가 중지되면 스토리지는 삭제됩니다. 기본적으로 **Pod**는 임시 스토리지 볼륨의 암호화를 사용하도록 설정되어 있습니다.

자세한 내용은 **Amazon EKS** 사용 설명서의 ‘**AWS Fargate**’ (<https://docs.aws.amazon.com/eks/latest/userguide/fargate.html>)를 참조하십시오.

## Fargate 프로파일

```
{  
    "fargateProfileName": myprofile,  
    "clusterName": mycluster,  
    "PodExecutionRoleArn": iam-role-xyz,  
    "subnets": subnet-0ad888345,  
    "selectors": [  
        {  
            "namespace": prod,  
            "labels": {  
                stack: frontend  
            }  
        }  
    ]  
}
```



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### ~ALT text

~Fargate 프로파일이 연결된 Pod입니다.

#### | 수강생용 노트

Amazon EKS에서 AWS Fargate를 활용하려면 Fargate 프로파일을 생성해야 합니다. Fargate 프로파일은 Fargate에서 예약할 Pod를 지정합니다. AWS Fargate에서 Pod를 모두 실행하거나 일부만 실행하도록 선택할 수 있습니다. Fargate 프로파일은 네임스페이스와 레이블이 포함된 선택기를 사용합니다. 프로파일에 있는 selectors의 네임스페이스 및 모든 레이블과 일치하는 모든 Pod는 Fargate로 스케줄링됩니다. 프로파일당 5개의 selector만 허용됩니다. Fargate 프로파일은 Pod가 시작될 서브넷도 지정합니다. 프라이빗 서브넷만 허용됩니다.

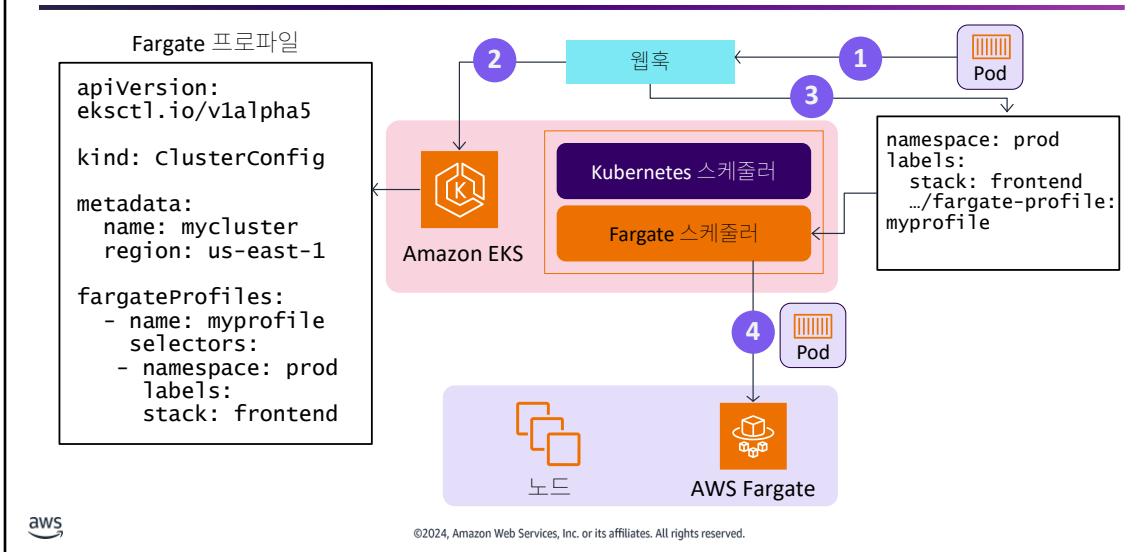
#### IAM 역할

각 Fargate 프로파일에는 PodExecutionRole도 포함됩니다. Pod가 AWS API를 호출할 수 있는 권한을 결정하는 IAM 역할입니다. 이 IAM 역할은 권한 부여를 위해 Kubernetes 역할 기반 액세스 제어(RBAC)에도 추가됩니다. 이렇게 하면 Fargate 인프라에서 실행 중인 kubelet이 Amazon EKS 클러스터에 등록되어 클러스터에 노드로 표시될 수 있습니다.

### **Fargate** 프로파일 지정

Pod가 여러 Fargate 프로파일과 일치하는 경우 Amazon EKS는 임의로 프로파일을 선택합니다. Pod에 대한 프로파일을 지정하려는 경우 Pod 사양 ([eks.amazonaws.com/fargate-profile: profile\\_name](https://eks.amazonaws.com/fargate-profile: profile_name))에 Kubernetes 레이블을 추가할 수 있습니다. 프로파일을 사용하려면 Pod가 네임스페이스 및 selector의 모든 레이블과 일치해야 합니다.

## Fargate 프로파일을 사용한 Pod 예약(1/2)



### ~ALT text

~기본: Fargate 및 EC2 인스턴스를 사용하는 Amazon EKS 클러스터를 보여주는 아키텍처입니다.

~1단계 및 2단계: 1단계 및 2단계로 노트에 자세히 설명되어 있습니다.

~3단계: 3단계로 노트에 자세히 설명되어 있습니다.

~4단계: 4단계로 노트에 자세히 설명되어 있습니다.

| 강사용 노트 및 애니메이션

| <1> 1단계와 2단계는 Pod가 웹 흐름을 통하여 Fargate 프로파일을 확인하는 Amazon EKS로 전송되는 Pod를 보여줍니다.

| <2> Fargate 프로파일이 일치했음을 보여주는 3단계가 표시되고 웹 흐름이 Fargate 프로파일을 레이블로 포함하도록 Pod를 변경하고 이를 Fargate 스케줄러를 통해 전송합니다.

| <3> AWS Fargate에서 Pod를 예약하는 Fargate 스케줄러를 보여주는 4단계가 나타납니다.

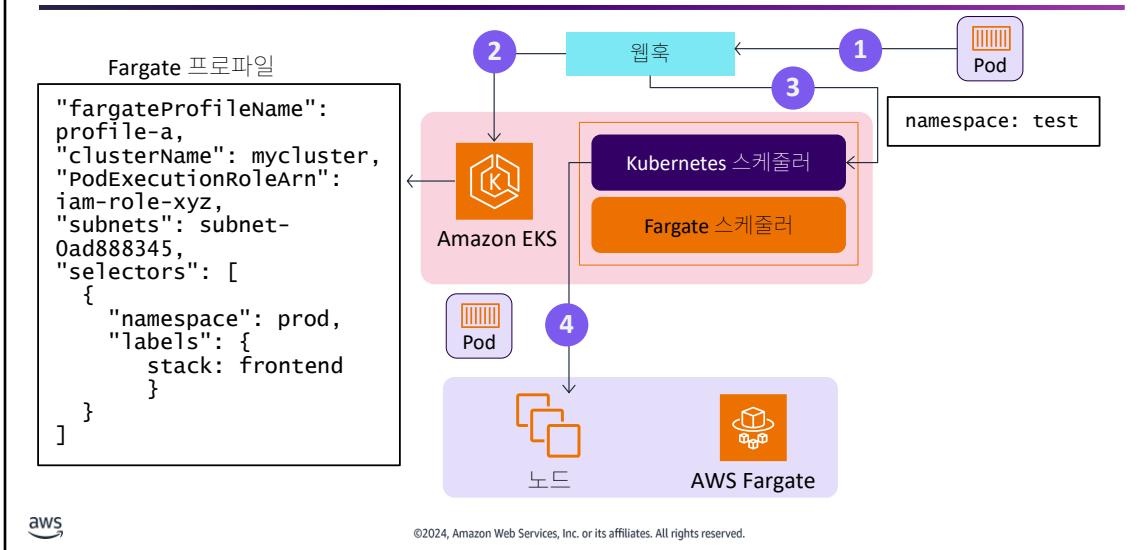
| 수강생용 노트

이는 Fargate 및 EC2 인스턴스를 모두 사용하는 Amazon EKS 클러스터의 예입니다.

1. 새 Pod가 Kubernetes에 배포되면 웹 흐름이 요청을 받아 Amazon EKS로 보냅니다.
2. 그런 다음 Amazon EKS는 Pod에 대한 Fargate 프로파일에 일치하는 항목이 있는지 확인합니다. 이 경우 Pod는 스택 레이블이 프런트엔드와 동일한 **prod** 네임스페이스에 배포됩니다. 이는 Fargate 프로파일과 일치합니다.

3. Pod가 Fargate 프로파일과 일치하므로 웹혹은 Fargate 프로파일을 레이블로 포함하도록 Pod를 변경한 다음(예: eks.amazonaws.com/fargate-profile: profile-a) Fargate 스케줄러를 사용합니다. Fargate 스케줄러는 AWS가 Kubernetes를 기반으로 구축한 사용자 정의 스케줄러입니다. (이는 모듈 1에서 학습한 대로 사용자 정의 객체와 사용자 정의 컨트롤러로 일반 Kubernetes 클러스터를 확장하여 수행됩니다.)
4. 그런 다음 Fargate 스케줄러는 AWS Fargate에서 Pod를 스케줄링합니다.

## Fargate 프로파일을 사용한 Pod 예약(2/2)



### ~ALT text

~기본: Fargate 및 EC2 인스턴스를 사용하는 Amazon EKS 클러스터를 보여주는 아키텍처입니다.

~1단계 및 2단계: 1단계 및 2단계에서는 Pod가 웹훅을 통해 Amazon EKS로 이동하여 Fargate 프로파일을 확인합니다.

~3단계: Pod가 Fargate 프로파일과 일치하지 않으므로 Kubernetes 스케줄러로 전송됩니다.

~4단계: 그런 다음 Pod가 적절한 EC2 인스턴스로 스케줄링됩니다.

| 강사용 노트 및 애니메이션

| <1> 1단계와 2단계가 표시되고 웹훅을 통해 Pod를 Amazon EKS로 보낸 다음 Fargate 프로파일을 확인합니다.

| <2> 3단계가 표시되고 Fargate 프로파일이 일치하지 않으므로 Kubernetes 스케줄러를 통해 Pod가 전송됩니다.

| <3> 4단계가 표시된 다음 Pod가 적절한 EC2 인스턴스로 스케줄링됩니다.

| 시간이 있다면 이 슬라이드 다음에 AWS Fargate를 시연하면 좋습니다.

| 원하는 경우 다음 슬라이드를 숨김 해제하여 데모 슬라이드를 표시할 수 있습니다.

| Fargate 데모 노트:

| 요구 사항:

| - Amazon EKS 클러스터 및 관리형 노드 그룹

| - Fargate 프로파일

| Fargate를 대상으로 하는 Pod의 프로파일 세부 정보를 살펴봅니다.

| Fargate에서 시작되는 Pod를 생성합니다.

- | 관리형 노드 그룹에서 시작되는 **Pod**를 생성합니다.
  - | 필요에 따라 데모를 확장하고 시간이 허락한다면 다음 슬라이드에서 자연스럽게 토론으로 넘어가는 것을 고려하십시오.
- | **수강생용 노트**
- Pod**가 **Fargate** 프로파일과 일치하지 않는 경우 웹 혹은 **Pod**를 수정하지 않고 대신 **Kubernetes** 스케줄러로 보냅니다.

그런 다음 **Kubernetes** 스케줄러는 모듈 1에서 학습한 일반 예약 규칙에 따라 적절한 **EC2** 인스턴스에서 **Pod**를 스케줄링합니다.

## 데이터 영역 옵션 비교

	Fargate	관리형 노드	셀프 매니지드 노드
작업 단위	Pod	Pod 및 EC2	Pod 및 EC2
요금 단위	Pod	EC2	EC2
호스트 수명 주기	표시되는 호스트 없음	AWS(SSH가 허용됨)	고객
호스트 : Pod	일대일	일대다	일대다



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### ~ALT text

~Fargate, 관리형 노드, 셀프 매니지드 노드에 대한 데이터 영역 옵션을 비교하는 테이블입니다.

#### | 수강생용 노트

##### 작업 단위

선택한 데이터 영역 옵션에 따라 다른 객체로 작업하게 됩니다. 셀프 매니지드 노드 및 관리형 노드를 사용하면 Pod 및 EC2 인스턴스로 작업할 수 있습니다. 반면, Fargate 노드는 Pod로만 작업할 수 있도록 관리됩니다.

##### 요금 단위

일반적으로 Amazon EKS에서는 EC2 인스턴스 사용량에 대한 요금이 부과됩니다. AWS Fargate에서는 Pod에 대해서만 요금이 부과됩니다.

##### 호스트 수명 주기

AWS Fargate를 사용할 때 호스트는 AWS에서 관리하므로 표시되는 호스트가 없다는 점을 기억하십시오. 셀프 매니지드 노드를 사용하면 수명 주기를 제어하고 호스트의 AMI를 선택할 수 있습니다. 관리형 노드를 사용하면 AWS가 수명 주기를 관리하고 사용자는 EKS 최적화된 AMI를 사용합니다.

## AWS Fargate 및 노드 그룹

### Fargate를 고려해야 할 상황:

- 고급 Pod 예약의 복잡성을 줄여야 함
- 규정 준수 및 보안을 개선해야 함 – Pod 대 노드 비율 일대일
- 변동비를 절감해야 함
- 인프라 관리 부담을 줄여야 함

### 노드 그룹을 고려해야 할 상황:

- 권한 있는 Pod를 허용
- 퍼블릭 서브넷에 노드를 배포
- 워크로드 중단을 관리
- Pod 시작 시간을 개선



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### | 수강생용 노트

Amazon EKS는 AWS Fargate, EC2 인스턴스 또는 이 둘의 조합 등 애플리케이션 실행 방법을 결정하는데 유연성을 제공합니다. Amazon EKS를 사용하면 언제든지 여러 Fargate 프로파일 및 노드 그룹을 실행하도록 클러스터를 사용자 지정할 수 있습니다. 애플리케이션 실행 방법에 대한 결정은 애플리케이션 및 조직 요구 사항에 따라 다릅니다.

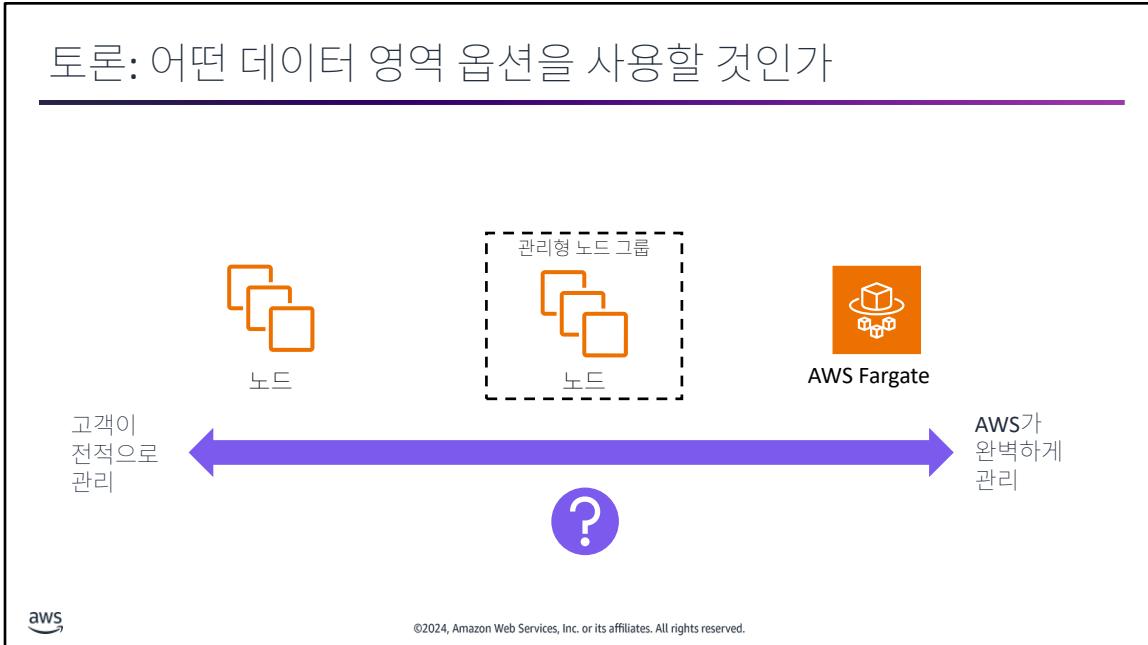
다음 조건에서는 AWS Fargate를 고려해야 합니다.

- Pod가 클러스터 내에서 분산되는 방식을 면밀히 관리할 필요가 없습니다. 예를 들어 일부 마이크로서비스는 동일한 노드에서 실행되어야 하거나 동일한 노드에서 실행될 필요가 없습니다(선후도/비선후도).
- 조직이 애플리케이션이 심층 방어 보안 모델(예: 커널을 넘어서는 추가 경계)을 따르도록 요구합니다. AWS Fargate는 특정 규정 준수 요구 사항을 충족하는 부담을 경감하는 데에도 도움이 됩니다. 자세한 내용은 ‘규정 준수 프로그램 제공 AWS 범위 내 서비스’(<https://aws.amazon.com/compliance/services-in-scope/>)를 참조하십시오.
- AMI 관리, 보안 및 규정 준수 요구 사항 충족, 애플리케이션에 과다 할당된 컴퓨팅 리소스와 같은 가변 비용을 절감하기를 원합니다.
- 클러스터 구성 요소에 대한 업데이트 및 패치를 포함하여 가상 인프라를 관리하는 책임을 원치 않습니다.
- 더 많은 고려 사항(예: AWS Fargate가 지원할 수 있는 사항과 지원하지 않는 사항)을 보려면 <https://docs.aws.amazon.com/eks/latest/userguide/fargate.html#fargate-considerations>를 참조하십시오.

다음 조건에서는 **EC2**를 고려해야 합니다.

- 클러스터에 권한 있는 **Pod**가 필요합니다.
- 퍼블릭 서브넷에 노드를 배포하기를 원합니다. **AWS Fargate Pod**는 프라이빗 서브넷에서만 실행되는 반면, **EC2** 인스턴스는 프라이빗 서브넷과 퍼블릭 서브넷 모두 유연하게 선택할 수 있습니다.
- 애플리케이션에서 업그레이드와 같은 워크로드 중단이 발생하는 방식을 제어하기를 원합니다.
- **Pod**의 시작 시간을 개선하기를 원합니다. **AWS Fargate**에서 실행하도록 예약된 각 **Pod**에 대해 새로운 가상 머신(**VM**)이 생성됩니다. 그 결과 **Pod**의 시작 시간이 느려집니다. **EC2** 인스턴스에서 실행되는 **Pod**는 일반적으로 시작 시간이 더 짧습니다.

## 토론: 어떤 데이터 영역 옵션을 사용할 것인가



### | 강사용 노트

- | 수강생들에게 자신이 개발한 애플리케이션을 생각해 보도록 격려합니다.  
애플리케이션 요구 사항에 어떤 데이터 영역인 옵션이 적합할지 설명해 달라고  
요청합니다. 필요에 따라 수정 피드백을 제공합니다.
- | 아마도 수강생들은 이미 **Amazon EKS**에서 애플리케이션을 실행하고 있을 것입니다.  
현재 데이터 영역 옵션을 선택한 이유를 설명해 달라고 요청하십시오. 지금이라면 다른  
선택을 하게 될까요?
- | 토론의 요점은 수강생들이 애플리케이션 요구 사항에 적합한 데이터 영역 옵션과  
적합하지 않은 옵션을 고려하도록 하는 것입니다.

활동



aws

권한 검토

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트

| 수강생용 노트

## 활동: 권한 검토(1/5)

### 작업

클러스터에서 애플리케이션이 실행되는 방식을 적용합니다.

API 객체에 자동으로 주석을 추가합니다.

Kubernetes API 서버에 연결합니다.

클러스터에 노드를 추가합니다.

### 구성

승인 컨트롤러

kubeconfig 파일

IAM 역할 및 kubeconfig 파일

MutatingAdmissionWebhook



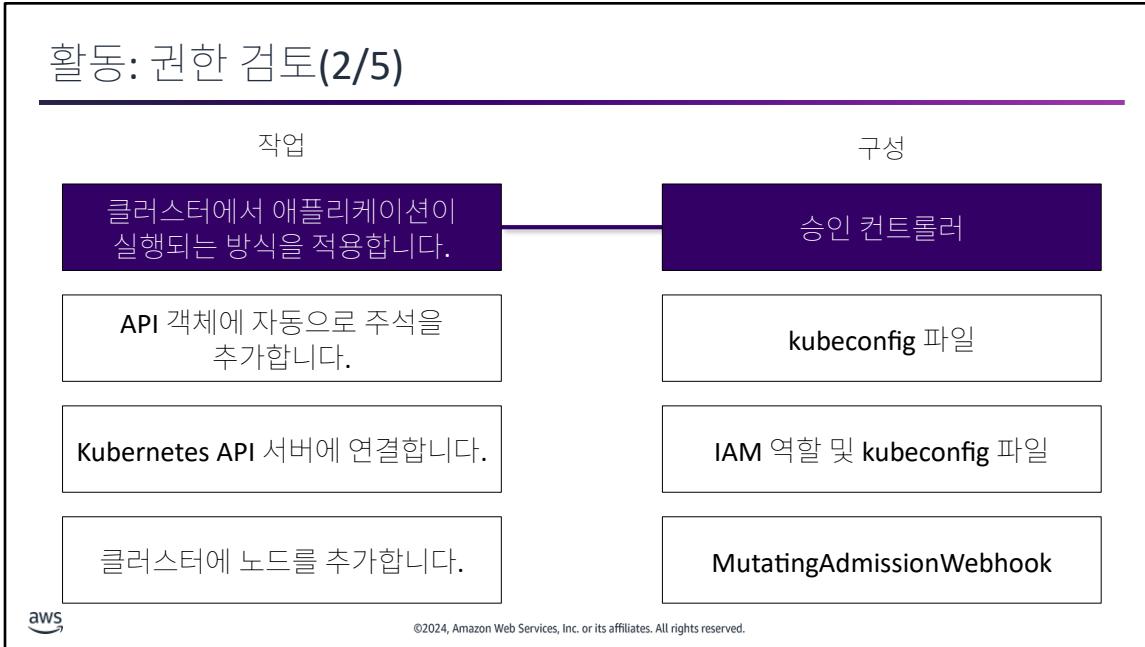
©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### | 수강생용 노트

이 활동에서는 네 가지 ‘작업’을 각각 해당하는 ‘구성’과 일치시킵니다.

나열된 작업은 ‘애플리케이션이 클러스터에서 실행되는 방식을 적용’, ‘API 객체에 자동으로 주석을 추가’, ‘Kubernetes API 서버에 연결’, ‘클러스터에 노드를 추가’입니다. 나열된 구성은 승인 컨트롤러, kubeconfig 파일, IAM 역할 및 kubeconfig 파일, MutatingAdmissionWebhook입니다.

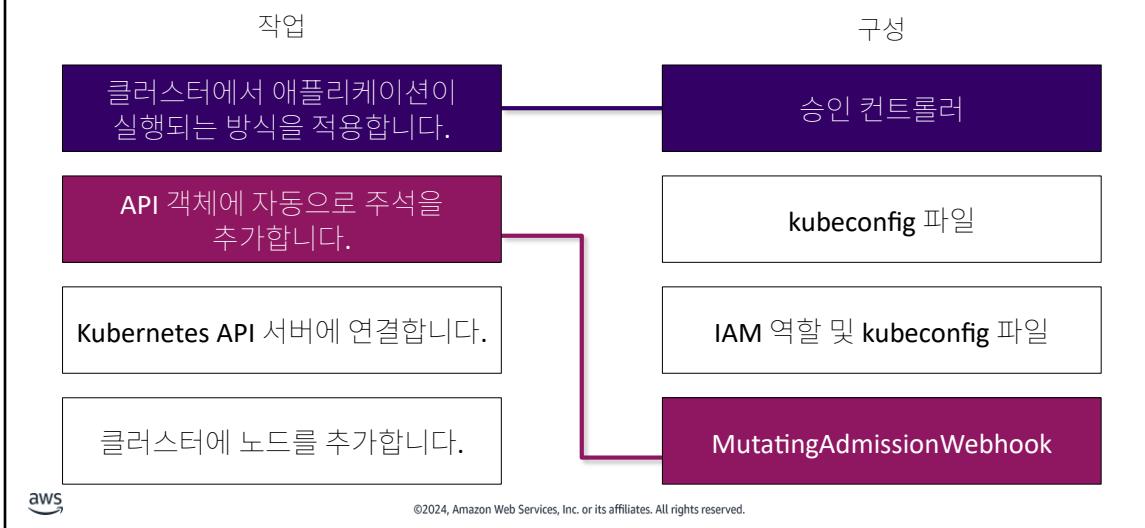
## 활동: 권한 검토(2/5)



### | 수강생용 노트

첫 번째 작업인 ‘클러스터에서 애플리케이션이 실행되는 방식을 적용’은 승인 컨트롤러 구성과 일치합니다.

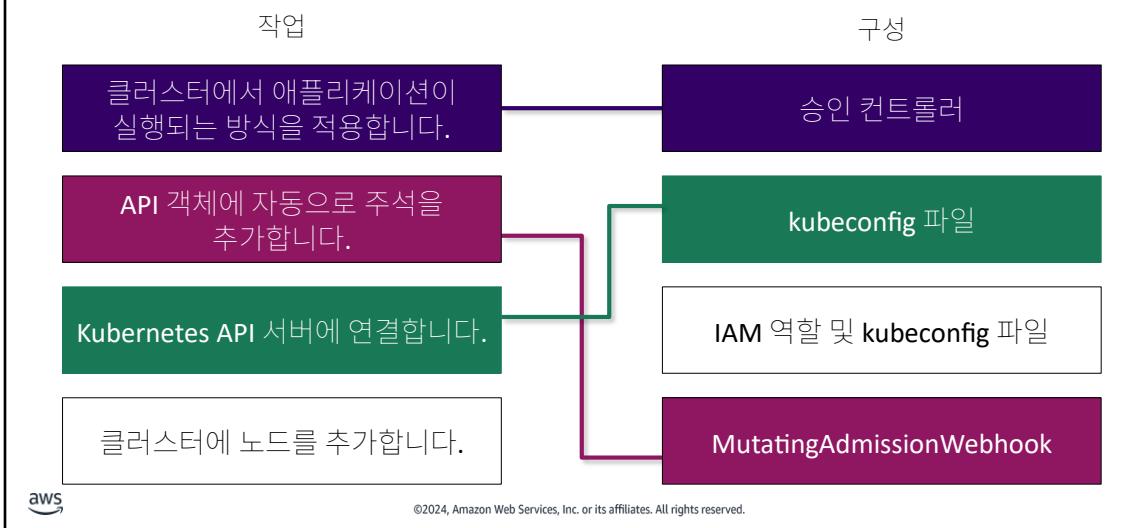
## 활동: 권한 검토(3/5)



### | 수강생용 노트

두 번째 작업인 ‘API 객체에 자동으로 주석을 추가’는 MutatingAdmissionWebhook 구성과 일치합니다.

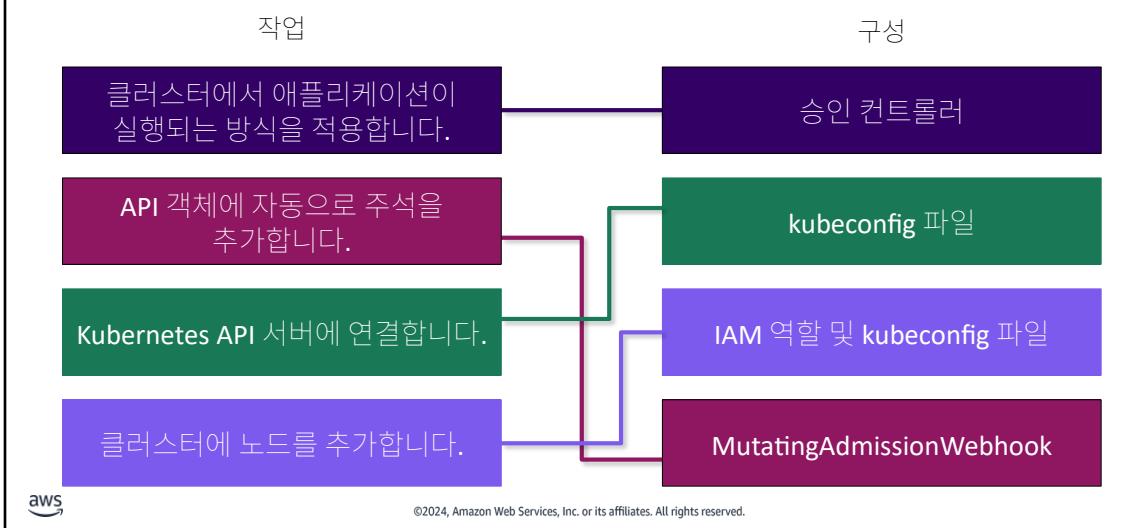
## 활동: 권한 검토(4/5)



### | 수강생용 노트

세 번째 작업인 ‘Kubernetes API 서버에 연결’은 **kubeconfig** 파일 구성과 일치합니다.

## 활동: 권한 검토(5/5)



### | 수강생용 노트

마지막 네 번째 작업인 ‘클러스터에 노드를 추가’는 **IAM 역할** 및 **kubeconfig** 파일 구성과 일치합니다.



Running Containers on Amazon EKS

## 2개의 API: Kubernetes 및 Amazon EKS



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## API 이해

### Kubernetes API

Amazon EKS 클러스터에서 리소스를 관리:

- Kubernetes 객체: Pod, 배포, 네임스페이스 등
- 레이블링 및 주석 지정
- 사용자 정의 리소스

### Amazon EKS API

Amazon EKS 클러스터의 인프라를 관리:

- 클러스터
- 관리형 노드 그룹
- Fargate 프로파일
- 추가 기능
- 자격 증명 공급자 구성
- 태그 지정



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

### | 수강생용 노트

#### Kubernetes API

Kubernetes API는 Pod, 배포, 네임스페이스와 같은 Kubernetes 객체를 처리하는 데 사용됩니다.

예:

```
kubectl get pods
```

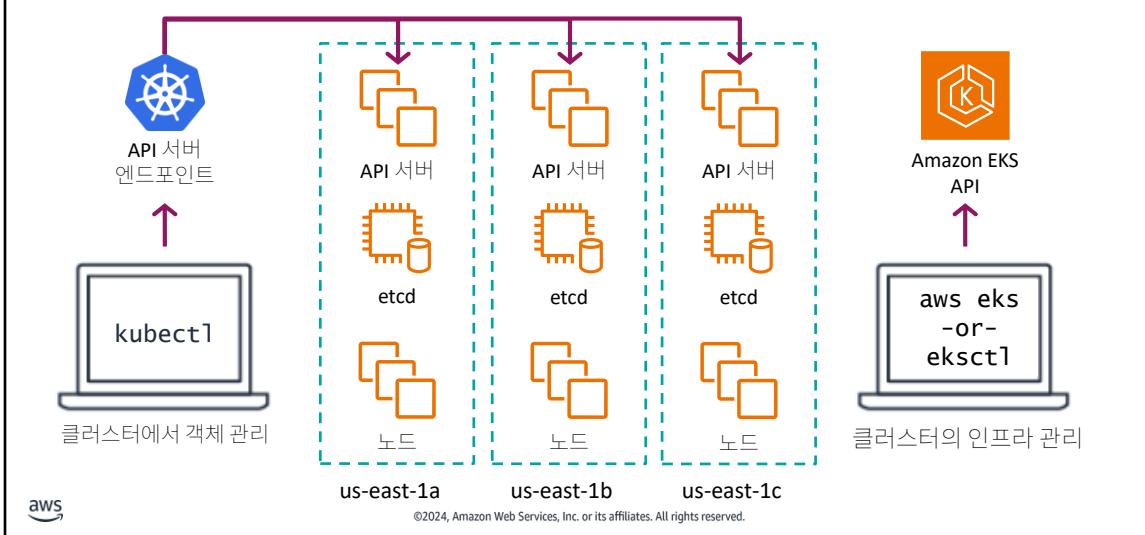
#### Amazon EKS API

Amazon EKS API는 제어 영역(클러스터 처리), 관리형 노드 그룹, Fargate 프로파일 등 Amazon EKS가 관리하는 모든 것에 사용됩니다.

예:

```
eksctl create cluster
```

## 2개의 API: Kubernetes API 및 Amazon EKS API



### | 수강생용 노트

Amazon EKS 클러스터를 관리할 때는 2개의 서로 다른 API로 작업하고 있다는 점을 기억하십시오. Kubernetes 명령을 실행할 때 **kubectl**을 통해 Kubernetes API와 상호 작용하게 됩니다. Amazon EKS 명령을 실행할 때는 콘솔, AWS CLI 또는 **eksctl**에서 AWS API를 사용해 작업합니다. **eksctl**에 대한 자세한 내용은 ‘**eksctl - The official CLI for Amazon EKS**’(<https://eksctl.io/#>)를 참조하십시오.

요약하자면, 클러스터에서 Kubernetes 객체를 관리하려면 Kubernetes API를 사용하십시오. 코드에서 Kubernetes API를 직접 호출할 수도 있고, **kubectl** 및 **helm**과 같은 도구를 사용할 수도 있습니다. DevOps 파이프라인은 **kubectl** 및 **helm**을 사용하여 Kubernetes 클러스터에서 워크로드 관리를 자동화할 수 있다는 점을 기억하십시오. Amazon EKS 클러스터의 인프라를 관리하려면 Amazon EKS API를 사용하십시오. AWS 콘솔, AWS CLI 또는 **eksctl**과 같은 도구를 사용하여 Amazon EKS 클러스터를 생성, 삭제 및 관리할 수 있습니다. AWS CloudFormation, HashCorp Terraform, AWS Cloud Development Kit(AWS CDK), CDK for Terraform과 같은 도구를 사용하여 클러스터 인프라의 생성, 삭제 및 관리를 자동화할 수 있습니다.

## 모듈 요약



이 모듈에서 학습한 내용:

- Amazon EKS는 제어 영역을 관리합니다.
- 관리형 노드 그룹은 데이터 영역을 관리하는 동시에 사용자에게도 제어권을 부여합니다.
- AWS는 AWS Fargate 노드를 완전히 관리합니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트

| 수강생용 노트



Running Containers on Amazon EKS

## 실습 1: Kubernetes Pod 배포



©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 강사용 노트

| 이 실습을 완료하는 데는 약 **45분** 정도 걸립니다.

| 수강생용 노트

실습 1에서는 애플리케이션을 Amazon EKS 클러스터에 배포합니다.

## 실습 1



aws

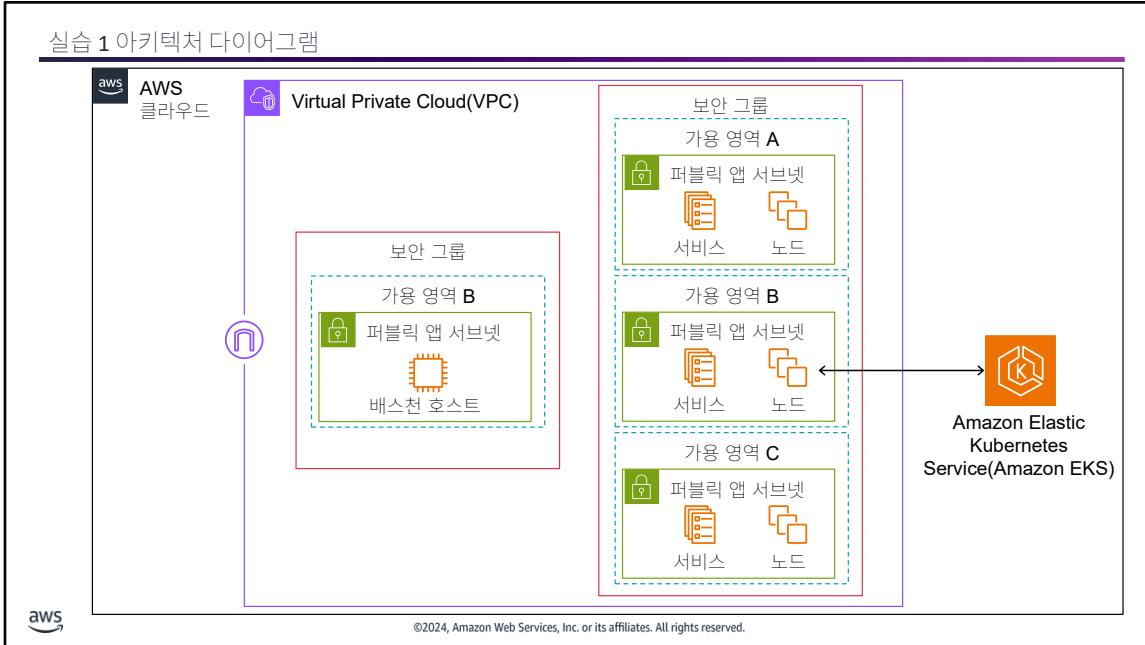
### Kubernetes Pod 배포

이 실습에서는 다음 과제를 수행합니다.

- **Kubernetes** 애플리케이션을 생성하고 배포합니다.
- 배포, 서비스 및 네임스페이스 리소스를 생성합니다.
- 네임스페이스에서 리소스를 봅니다.
- 서비스 및 배포의 세부 정보를 이해합니다.
- **Pod**의 세부 정보를 이해합니다.
- **Pod**에서 명령을 실행합니다.
- 애플리케이션을 삭제합니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## 실습 1 아키텍처 다이어그램



### |수강생용 노트

이 다이어그램은 실습 1의 아키텍처를 보여줍니다.

[이미지 설명: 3노드 Amazon EKS 클러스터가 포함된 VPC를 보여주는 아키텍처입니다. 각 노드는 서로 다른 가용 영역에 있습니다. 한 애플리케이션에 각 노드에서 실행되는 Pod가 있으며 이 애플리케이션은 서비스를 통해 사용할 수 있습니다. 클러스터에 대한 액세스를 제공하기 위해 별도의 배스천 호스트가 가용 영역 중 하나에 배포됩니다.  
설명 끝]



감사합니다.



수정 사항이나 피드백 또는 기타 질문이 있으십니까?  
<https://support.aws.amazon.com/#/contacts/aws-training>에서 문의해  
주십시오. 모든 상표는 해당 소유자의 자산입니다.

©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.