



Running Containers on Amazon EKS

모듈 3:

Amazon EKS 클러스터 구축 및 유지 관리



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

| 강사용 노트

| 이 모듈을 완료하는 데는 약 **1시간 10분** 정도 걸립니다.



Running Containers on Amazon EKS

모듈 3 개요

- Amazon EKS 클러스터 생성
- 노드 배포
- Kubernetes 버전 업그레이드



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



Running Containers on Amazon EKS

Amazon EKS 클러스터 생성

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



| 수강생용 노트

이 주제에서는 AWS Management Console을 사용하여 Amazon Elastic Kubernetes Service(Amazon EKS)에서 클러스터를 생성하는 방법을 알아봅니다.

클러스터 구성 및 관리 도구



aws

Amazon EKS 클러스터를 구성하고 관리하려면 다음 도구를 사용합니다.

- `eksctl`
- AWS Command Line Interface(AWS CLI)
- `kubectl`

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

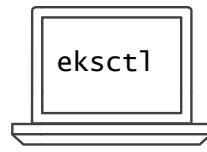
| 수강생용 노트

첫 번째 클러스터를 생성하기 전에 클러스터에 액세스하는 데 사용할 시스템에 필요한 모든 도구를 설치했는지 확인합니다. 이 모듈에서는 `eksctl`을 클러스터 생성 기본 방법으로 사용합니다. '클러스터 생성'이란 Amazon EKS에 의해 관리되는 제어 영역을 배포하는 프로세스를 나타냅니다. 클러스터에 노드를 추가하는 것은 별도의 단계이며 이 모듈의 뒷부분에서 알아봅니다.

이러한 도구를 설치하는 방법에 대한 자세한 내용은 다음 자료를 참조하십시오.

- **Amazon EKS** 사용 설명서의 ‘Amazon EKS 시작하기 – `eksctl`’ (<https://docs.aws.amazon.com/eks/latest/userguide/getting-started-eksctl.html>)
- AWS Command Line Interface 사용 설명서의 ‘최신 버전의 **AWS CLI** 설치 또는 업데이트’ (<https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html>)
- **Amazon EKS** 사용 설명서의 ‘`aws-iam-authenticator` 설치’ (<https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html>)
 - 참고: AWS CLI 버전 1.16.156 이상을 실행하는 경우 인증자를 설치할 필요가 없습니다. 대신 `aws eks get-token` 명령을 사용할 수 있습니다.
- **Amazon EKS** 사용 설명서의 ‘`kubectl` 설치 또는 업데이트’ (<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html>)

클러스터 생성을 위한 인터페이스



AWS Management Console

- 그래픽 인터페이스 내에서 수행되는 대부분의 작업

AWS CLI

- 더 큰 유연성
- 콘솔보다 높은 복잡성

eksctl

- 단일 명령으로 클러스터를 생성하는 강력한 CLI 도구



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

| 수강생용 노트

다음 방법 중 하나 이상을 사용하여 클러스터를 생성할 수 있습니다.

- AWS Management Console**은 클러스터 생성의 여러 복잡성을 자동으로 처리하는 그래픽 인터페이스를 제공합니다. 그러나 이 방법을 사용할 때는 **AWS CLI**를 사용하여 몇 가지 단계를 수행해야 합니다. 이러한 단계는 이 모듈의 뒷부분에 설명되어 있습니다.
- AWS CLI**에서는 가장 많은 사용자 지정이 가능합니다. 또한 3가지 방법 중 가장 복잡합니다.
- eksctl**은 클러스터 생성을 단순화하는 명령줄 도구입니다. 이 방법을 사용하면 단일 명령으로 클러스터를 생성할 수 있습니다. 이 모듈의 뒷부분에서 **eksctl**에 대해 자세히 설명합니다.

특히 프로세스에 자동화를 추가하려는 경우 여러 가지 다른 방법을 사용할 수 있습니다. **AWS API**를 사용하여 스크립트를 작성할 수 있습니다. **AWS CloudFormation** 또는 기타 서드 파티 프로비저닝 도구를 사용하여 클러스터를 배포할 수 있습니다. 사용자 환경에 자동화를 추가하는 방법은 이 과정의 뒷부분에서 알아봅니다.

이 모듈 및 과정 전반에서 **eksctl**을 클러스터 생성 기본 방법으로 사용합니다.

eksctl이란 무엇입니까?

AWS CloudFormation을 사용하여 Amazon EKS에서 클러스터를 생성하기 위한 CLI 도구

```
$ eksctl create cluster -f cluster.yaml
```

또는

```
$ eksctl create cluster \  
--name prod \  
--version 1.27 \  
--region us-west-2 \  
--nodegroup-name standard-workers \  
--node-type t3.medium \  
--nodes 3 \  
--nodes-min 1 \  
--nodes-max 4 \  
--managed
```



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

| 수강생용 노트

Weaveworks는 Amazon EKS에서 Kubernetes 클러스터를 생성하고 관리하는 간단한 명령줄 유틸리티로 eksctl을 개발했습니다. eksctl 명령줄 유틸리티는 Amazon EKS용 노드가 포함된 새 클러스터를 더 빠르고 쉽게 만들 수 있는 방법을 제공합니다. 이 유틸리티는 AWS CloudFormation도 사용합니다. eksctl과 함께 사용할 구성 파일을 만들거나(첫 번째 예) 인수를 추가하여 명령을 사용자 지정할 수 있습니다(두 번째 예).

eksctl 코드를 보려면 GitHub의 eksctl 리포지토리(<https://github.com/eksctl-io/eksctl>)를 참조하십시오.

eksctl에 대한 자세한 내용은 ‘eksctl - The official CLI for Amazon EKS’ (<https://eksctl.io>)를 참조하십시오.

cluster.yaml 파일을 사용하여 eksctl 사용자 정의

```
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: MySecondEKScluster
  region: us-east-1
vpc:
  subnets:
    private:
      us-east-1b: { id: subnet-0fdd9b15cb2da9ce6 }
      us-east-1c: { id: subnet-061bff24d18088fd5 }
nodeGroups:
  - name: ClusterTwoNodes
    instanceType: m5.xlarge
    desiredCapacity: 3
    privateNetworking: true
```



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

| 수강생용 노트

플래그 대신 구성 파일을 사용하여 클러스터를 생성할 수 있으므로 선택하는 옵션을 더 효율적으로 제어할 수 있습니다. **cluster.yaml** 파일을 사용하면 클러스터 생성 프로세스가 명령형이 아닌 선언형으로 바뀝니다. **eksctl**은 구성 파일의 명령을 동등한 AWS CloudFormation 템플릿으로 변환합니다. AWS CloudFormation 템플릿이나 Terraform 자동화 파일과 같은 다른 자산과 동일한 방식으로 **cluster.yaml** 파일을 소스 코드 제어에 넣을 수 있습니다.

이 예에서는 새 VPC를 생성하는 대신 기존 Virtual Private Cloud(VPC)를 사용하도록 **eksctl**을 구성하는 **cluster.yaml** 파일을 만드는 방법을 보여줍니다. 이 예에서는 기본값보다 큰 인스턴스 크기를 사용합니다. 구성 파일은 사용할 리전, 가용 영역, 서브넷과 같이 필요한 세부 정보를 지정합니다.

자세한 내용은 ‘eksctl - The official CLI for Amazon EKS’의 개요(<https://eksctl.io/>)를 참조하십시오.

eksctl은 기본적으로 무엇을 수행합니까?



1. 클러스터 및 노드의 IAM 역할 생성
2. CIDR 192.168.0.0/16을 사용하여 전용 VPC 생성
3. 클러스터 및 노드 그룹 생성
4. API 엔드포인트에 대한 액세스 구성
5. 클러스터의 kubeconfig 파일 작성

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

| 수강생용 노트

eksctl은 클러스터 및 노드 생성에 필요한 많은 단계를 자동화합니다. 다음은 기본 옵션으로 실행될 때 **eksctl**이 수행하는 태스크의 개요입니다.

1. 클러스터 및 노드에 대한 AWS Identity and Access Management(IAM) 역할을 생성합니다.
2. Classless Inter-Domain Routing(CIDR) 192.168.0.0/16을 8개의 서브넷(프라이빗 3개, 퍼블릭 3개, 예약 2개)으로 균등하게 분할하여 전용 VPC를 생성합니다.
3. Amazon EKS 최적화 Amazon Machine Image(AMI)를 사용하여 2개의 m5.large 노드를 포함하는 클러스터와 노드 그룹을 생성합니다.
4. API 엔드포인트에 대한 액세스를 구성합니다.
5. 클러스터의 kubeconfig 파일을 작성합니다.

예(1/4): eksctl

1. eksctl을 설치합니다.

```
$ curl --silent --location
"https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp
$ sudo mv /tmp/eksctl /usr/local/bin
```

2. 클러스터 및 관리형 노드 그룹을 생성합니다.

```
$ eksctl create cluster
[i] eksctl version 0.123.0-dev+53d2577c4.2022-12-15T12:55:26Z
[i] using region us-east-1
[i] setting availability zones to [us-east-1a us-east-1f]
[i] subnets for us-east-1a - public:192.168.0.0/19 private:192.168.64.0/19
[i] subnets for us-east-1f - public:192.168.32.0/19 private:192.168.96.0/19
[i] nodegroup "ng-a3616114" will use "" [AmazonLinux2/1.23]
[i] using Kubernetes version 1.27
[i] creating EKS cluster "unique-painting-1671728673" in "us-east-1" region with managed
nodes
[i] will create 2 separate CloudFormation stacks for cluster itself and the initial
managed nodegroup
```



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

| 수강생용 노트

eksctl을 설치하고 기본 파라미터를 사용하여 클러스터 및 관리형 노드 그룹 생성을 시작하는 예입니다. 이 예에서는 eksctl이 IAM 역할, 서브넷이 포함된 VPC, 관리형 노드 그룹을 생성합니다.

eksctl create cluster의 출력에서 첫 번째 부분은 클러스터를 위해 리전 내에서 고유한 이름(이 경우 us-east-1)이 자동 생성되는 것을 보여줍니다.

이 예에서는 2개의 AWS CloudFormation 스택이 생성됩니다. 첫 번째 스택은 Amazon EKS 제어 영역을 프로비저닝합니다. 두 번째 스택은 데이터 영역을 위한 Amazon EKS 관리형 노드 그룹을 프로비저닝합니다.

예(2/4): eksctl

```
[i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks --region=us-east-1 --cluster=unique-painting-1671728673'  
[i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster "unique-painting-1671728673" in "us-east-1"  
[i] Cloudwatch logging will not be enabled for cluster "unique-painting-1671728673" in "us-east-1"  
[i] you can enable it with 'eksctl utils update-cluster-logging --enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=us-east-1 --cluster=unique-painting-1671728673'  
[i]  
2 sequential tasks: { create cluster control plane "unique-painting-1671728673",  
  2 sequential sub-tasks: {  
    wait for control plane to become ready,  
    create managed nodegroup "ng-a3616114",  
  }  
}
```



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

| 수강생용 노트

이 코드 예는 클러스터 생성 중 출력에 대한 자세한 내용을 보여줍니다. 클러스터를 생성하는 데 사용하는 인터페이스와 관계없이 클러스터를 생성하는 데 약 **15분**이 걸립니다. 따라서 이 프로세스 중에는 출력이 일시 중지될 것으로 예상됩니다.

예(3/4): eksctl

```
[i] building cluster stack "eksctl-unique-painting-1671728673-cluster"
[i] deploying stack "eksctl-unique-painting-1671728673-cluster" "728673-cluster"
[i] waiting for CloudFormation stack "eksctl-unique-painting-1671728673-cluster"
[i] building managed nodegroup stack "eksctl-unique-painting-1671728673-nodegroup-ng-
a3616114"
[i] deploying stack "eksctl-unique-painting-1671728673-nodegroup-ng-a3616114"
[i] waiting for CloudFormation stack "eksctl-unique-painting-1671728673-nodegroup-ng-
a3616114"
[i] waiting for the control plane to become ready
[✓] saved kubeconfig as "/Users/admin/.kube/config"
[i] no tasks
[✓] all EKS cluster resources for "unique-painting-1671728673" have been created
```



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

| 수강생용 노트

이 코드 예는 클러스터 생성 중 출력에 대한 자세한 내용을 보여줍니다(Amazon EKS 제어 영역). 첫 번째 CloudFormation 스택 배포가 충분히 진행되면 이 예의 eksctl은 두 번째 CloudFormation 스택을 시작하여 관리형 노드 그룹을 구축합니다. 제어 영역(Amazon EKS 클러스터)과 데이터 영역(Amazon EKS 관리형 노드 그룹)이 모두 배포되면 eksctl은 kubeconfig(~/.kube/config) 파일을 업데이트합니다. 이는 eksctl create cluster를 시작한 IAM 사용자 또는 역할에게 kubectl, Helm 또는 Kubernetes API를 사용하는 기타 도구를 사용하여 클러스터에 액세스할 수 있는 권한을 부여합니다.

예(3/4): eksctl

```
[i] nodegroup "ng-a3616114" has 2 node(s)
[i] node "ip-192-168-10-227.ec2.internal" is ready
[i] node "ip-192-168-51-216.ec2.internal" is ready
[i] waiting for at least 2 node(s) to become ready in "ng-a3616114"
[i] nodegroup "ng-a3616114" has 2 node(s)
[i] node "ip-192-168-10-227.ec2.internal" is ready
[i] node "ip-192-168-51-216.ec2.internal" is ready
[i] kubectl command should work with "/users/admin/.kube/config", try 'kubectl get nodes'
[✓] EKS cluster "unique-painting-1671728673" in "us-east-1" region is ready
```

3. 노드를 봅니다.

```
$ kubectl get nodes
NAME                  STATUS   ROLES      AGE      VERSION
ip-192-168-23-226.ec2.internal   Ready    <none>    6m26s   v1.15.11-eks-af3caf
ip-192-168-58-209.ec2.internal   Ready    <none>    6m30s   v1.15.11-eks-af3caf
```



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

| 강사용 노트

| 시간이 허락한다면 이 슬라이드 다음에 클러스터 배포에 대한 데모를 진행할 수 있습니다. 원하는 경우 다음 슬라이드를 숨김 해제할 수 있습니다.

| 데모

| 클러스터 구성 및 관리 도구를 설치합니다.

| IAM에서 권한을 구성합니다.

| 필요에 따라 네트워킹을 구성합니다.

| 클러스터를 생성합니다.

| 수강생용 노트

eksctl의 출력에 성공이 표시된 후 kubectl로 생성된 노드의 상태를 검사하여 준비 상태인지 확인합니다.



Running Containers on Amazon EKS

노드 배포

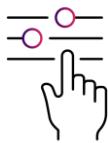


©2024, Amazon Web Services, Inc. or its affiliates. All rights reserved.

| 수강생용 노트

이 주제에서는 클러스터에 노드를 배포하는 방법을 배웁니다. 특히 관리형 노드 그룹을 생성하는 단계를 알아봅니다.

노드용 AMI 선택



Amazon EKS 최적화 Linux

- Amazon EKS용 기본 AMI
- 터키 솔루션

사용자 지정

- Amazon EKS AMI 빌드 사양은 GitHub에서 확인 가능
- AWS Fargate에 대해 지원되지 않음

Bottlerocket

- 컨테이너 실행을 위해 특별히 구축된 오픈 소스, Linux 기반 OS
- 기본 제공 AMI 선택으로 사용할 수 있습니다.

Windows

- Windows 컨테이너에 사용
- Windows Server 2019 및 2022를 기반으로 빌드



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

| 강사용 노트

| 시간이 허락한다면 이 슬라이드 다음에 클러스터 구성을 시연하면 좋습니다. 원하는 경우 다음 슬라이드를 숨김 해제하여 자리 표시자로 사용할 수 있습니다.

| 데모 노트

| 클러스터에 대한 액세스를 제어하는 **kubeconfig** 파일을 생성합니다.

| Amazon EC2 Auto Scaling을 허용하는 IAM 정책을 생성합니다.

| 노드에 대한 IAM 역할을 구성합니다.

| 관리형 노드 그룹을 생성하고 구성합니다.

| Cluster Autoscaler를 배포합니다.

| 수강생용 노트

Amazon EKS는 다양한 유형의 Amazon Machine Image(AMI)를 지원합니다.

Amazon EKS 최적화 Amazon Linux AMI

Amazon EKS 최적화 Amazon Linux AMI는 Amazon Linux 2를 기반으로 빌드되었습니다. 이 AMI는 Amazon EKS 노드의 기본 이미지로 사용하도록 구성됩니다. 이 AMI는 구성 없이 Amazon EKS에 사용할 수 있도록 구성됩니다. 여기에는 **kubelet**, **containerd**, **AWS IAM Authenticator**가 포함됩니다. 그래픽 처리 장치(**GPU**) 지원 유무와 관계없이 사용할 수 있으며 AWS Management Console에서 사용할 수 있습니다.

사용자 정의 Amazon EKS AMI

HashiCorp Packer로 사용자 정의 Amazon EKS AMI를 구축하기 위한 리소스 및 구성 스크립트는 **amazon-eks-ami GitHub** 리포지토리(<https://github.com/awslabs/amazon-eks-ami>)에서 제공됩니다. 이는 Amazon EKS가 공식 Amazon EKS 최적화 AMI를 생성하는 데 사용하는 구성과 동일합니다.

Bottlerocket

Bottlerocket은 Linux 기반 오픈 소스 운영 체제(OS)입니다. AWS는 가상 머신 또는 베어 메탈 호스트에서 컨테이너를 실행하기 위해 Bottlerocket을 구축했습니다. Bottlerocket은 다음과 같은 이점을 제공하도록 설계되었습니다.

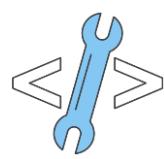
- 가동 시간 증가
- 관리 오버헤드 감소
- 컨테이너 애플리케이션을 위한 향상된 보안 및 최적화된 성능

이제 Bottlerocket은 AWS Management Console에서 선택할 수 있는 기본 제공 AMI입니다.

Windows

Windows Amazon EKS 최적화 AMI는 Windows Server 2019 및 Windows Server 2022를 기반으로 빌드되었습니다. 이 AMI는 Amazon EKS 노드의 기본 이미지로 사용하도록 구성됩니다. 여기에는 **kublet**, **kube-proxy**, **CSI** 프록시, **containerd**, AWS IAM Authenticator가 포함됩니다.

클러스터 생성을 위한 선언적 옵션



AWS CloudFormation

- 코드형 인프라로 취급하여 리소스를 스택으로 빠르고 일관성 있게 프로비저닝, 실행, 구성합니다.

Amazon EKS Blueprints

- 워크로드를 배포 및 운영하는 데 필요한 운영 소프트웨어로 완전히 부트스트래핑된 전체 EKS 클러스터를 구성합니다.

코드형 인프라(IaC)

- AWS SDK 및 서드 파티 IaC 소프트웨어 도구(예: Terraform)를 사용하여 클라우드 서비스를 관리합니다.



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

| 수강생용 노트

Amazon EKS 클러스터 작업을 위한 콘솔 및 명령줄 옵션(AWS CLI, eksctl) 외에도 AWS CloudFormation, Amazon EKS Blueprints, 코드형 인프라(IaC)와 같은 보다 선언형 옵션을 사용할 수도 있습니다.

AWS CloudFormation을 사용하면 인프라를 코드로 취급하여 리소스를 스택으로 빠르고 일관성 있게 프로비저닝, 시작, 구성할 수 있습니다.

EKS Blueprints는 워크로드를 배포 및 운영하는 데 필요한 운영 소프트웨어로 완전히 부트스트래핑된 전체 Amazon EKS 클러스터를 구성하는 데 도움이 됩니다. EKS Blueprints는 AWS Cloud Development Kit(CDK)를 사용하여 구축되었습니다. EKS Blueprints에 대한 자세한 내용은 ‘Amazon EKS Blueprints Quick Start’(<https://aws-quickstart.github.io/cdk-eks-blueprints/>)를 참조하십시오.

AWS SDK 및 Terraform, Pulumi, ClusterAPI 등 여러 서드 파티 IaC 소프트웨어 도구도 사용할 수 있습니다.



Running Containers on Amazon EKS

Kubernetes 버전 업그레이드



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

| 수강생용 노트

이 주제에서는 **Kubernetes** 버전 업그레이드 방법을 알아봅니다.

업그레이드 준비



신중하게 설명서 읽기



업그레이드를 담당할 팀 확인



업그레이드 절차 정의,
테스트, 문서화



환경과 워크로드에
미치는 영향 파악



다운스트림 종속성 확인



API 변경 계획



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

| 수강생용 노트

업그레이드의 성공을 위해서는 철저한 계획과 테스트가 필요합니다. 업그레이드 계획에는 다음 준비 사항이 포함되어야 합니다.

- 설명서를 꼼꼼히 읽은 후에 진행합니다. 클러스터의 **Kubernetes** 버전이나 **Amazon EKS** 버전을 업그레이드하기 전에 변경 사항 및 새 버전과 관련된 업그레이드 정보를 철저히 검토합니다. 다음 자료를 이용하여 조사를 시작합니다.
 - AWS 설명서** - <https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html>.
 - Kubernetes** 변경 로그 - <https://github.com/kubernetes/kubernetes/tree/master/CHANGELOG>.
- 업그레이드를 수행할 팀을 확인합니다. 클러스터 관리에 대한 책임 부서는 한 기업 내에서 여러 부서가 될 수 있습니다. 어떤 기업에는 클러스터를 관리하는 전담팀이 있고, 공동으로 책임을 지는 경우도 있습니다. 업그레이드 계획에 클러스터 구성 요소에 대한 업그레이드를 관리하는 각 팀의 책임을 구체적으로 명시해야 합니다.
- 강력한 업그레이드 절차 세트를 개발합니다. 테스트를 어떻게 실시할 것인가, 룰아웃 시 어떤 전략을 사용하는가 프로세스를 어떻게 자동화하는가 등을 고려합니다. 업그레이드에는 버전 특유의 문제가 발생할 수 있지만, 계획 수립의 토대로 사용할 성숙하고 포괄적인 전략을 시간을 두고 개발하십시오. 계획에는 주요 구성 요소를 업그레이드할 올바른 순서에 대한 결정도 포함될 수 있습니다(일반적으로 설명서에 있음).

- 환경과 워크로드에 미치는 영향을 파악합니다. 워크로드 실행에 미칠 수 있는 영향은 무엇인가, 업그레이드가 진행되는 동안 허용 가능한 서비스 수준을 유지하려면 어떤 절차를 취해야 하는가, 사용자의 환경에 원하는 업그레이드 전략을 수용할 정도로 충분한 리소스(예: 사용 가능한 IP 공간)가 있는가 등을 고려합니다.
- 다운스트림 종속성을 파악합니다. 어떤 종속 구성 요소를 함께 업그레이드해야 하는지 확인합니다. **Kubernetes** 클러스터는 여러 소스에서 나오는 고도로 분산된 구성 요소의 집합으로 만들어집니다. 자세한 내용은 **Kubernetes** 온라인 설명서의 '**Version Skew Policy**'(<https://kubernetes.io/docs/setup/release/version-skew-policy/>)를 참조하십시오. AWS Fargate의 경우 해당 **kubelet** 버전이 클러스터의 현재 업그레이드 전 **Kubernetes** 마이너 버전과 같도록 **Fargate Pod**를 재활용해야 할 수도 있습니다. 이 예에서는 **Fargate pod**의 **kubelet** 버전이 1.28.4가 되도록 **Fargate Pod**를 재활용하게 됩니다.
- API** 변경을 계획합니다. 사용이 중단된 **Kubernetes** 명령을 검토하고, 그 명령을 사용하는 스크립트나 애플리케이션이 있는지 확인하고, 지원되는 명령으로 마이그레이션할 계획을 시작합니다. 마이너 **Kubernetes** 릴리스는 주로 **API**를 변경합니다. 변화의 정도는 매우 다양할 수 있습니다. 자세한 내용은 다음 페이지를 참조하십시오.
 - ‘**Kubernetes Deprecation Policy**’ (<https://kubernetes.io/docs/reference/using-api/deprecation-policy/>)
 - ‘**Deprecated API Migration Guide**’ (<https://kubernetes.io/docs/reference/using-api/deprecation-guide/>)

Kubernetes 버전 업그레이드

- 마이너 버전은 고객이 제어합니다.
- 패치 버전은 제어 영역에 자동으로 적용됩니다.
- Amazon EKS 플랫폼 버전은 Amazon EKS에서 지원되는 Kubernetes 버전을 정의합니다.

v1.28.4

메이저
제품 전체
수정

마이너
새로운
기능

패치
버그 수정
보안



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

| 수강생용 노트

Kubernetes의 신규 버전에는 많은 변화가 적용될 수 있습니다. 이러한 변화의 영향은 같은 유형의 여러 업그레이드 간에도 크게 다를 수 있습니다. 예를 들어, 어떤 마이너 릴리스에는 **Kubernetes API**에 대한 약간의 변경 사항만 포함되어 있지만 또 어떤 마이너 릴리스에는 자주 사용되는 명령에 대한 중요한 변경 사항이 포함되어 있습니다. 마이너 릴리스에도 중요한 변경 사항이 포함될 수 있다는 점을 유념하십시오.

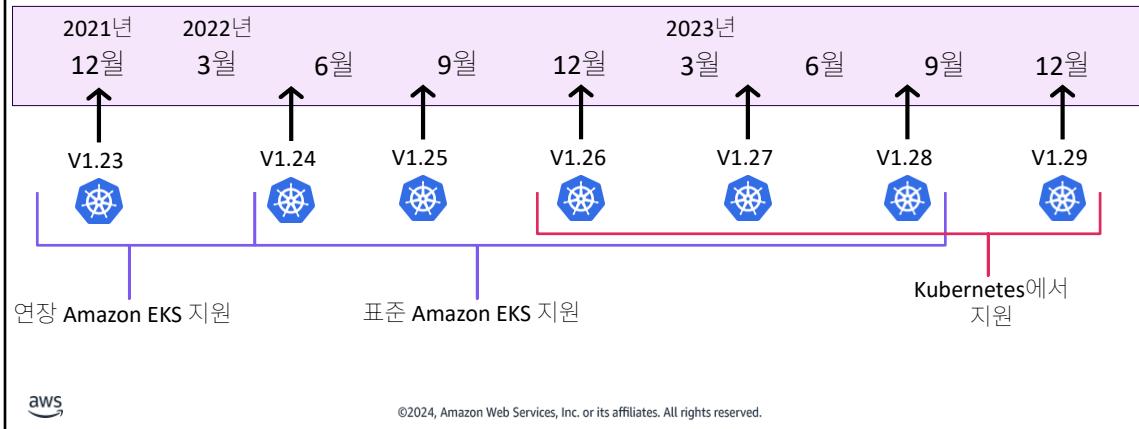
새 Kubernetes 버전 업그레이드 여부를 결정할 때는 다음을 고려하십시오.

- Kubernetes 버전 업그레이드 시 이점은 무엇인가
- Kubernetes 버전 업그레이드는 어느 팀에서 담당하는가
- 노드, Container Network Interface(CNI) 플러그 인과 같은 다운스트림 구성 요소와 역시 업그레이드해야 하는 Cluster Autoscaler와 같은 추가 기능을 파악합니다.
- 다운스트림 종속성을 어떤 순서로 업그레이드해야 하는가
- 업그레이드가 진행되는 동안 애플리케이션에 어떤 영향을 미치는가
- 에코시스템의 애플리케이션이 Kubernetes API를 사용하는가 이러한 애플리케이션에 미치는 영향을 분석하는 것도 고려해봅니다.

업그레이드를 수행하는 단계는 Kubernetes 버전에 따라 다릅니다. 자세한 내용은 **Amazon EKS** 사용 설명서의 ‘Amazon EKS 클러스터 Kubernetes 버전 업데이트’ (<https://docs.aws.amazon.com/eks/latest/userguide/update-cluster.html>)를 참조하십시오.

Kubernetes 릴리스 주기

*2024년 1월 5일 기준



~ALT text

~각 Kubernetes 버전이 출시된 시기와 Kubernetes 및 Amazon EKS에서 지원하는 버전을 보여 주는 2021~2023년의 타임라인입니다.

|수강생용 노트

Kubernetes에서는 적극적으로 릴리스 일정을 운영합니다. 새로운 마이너 버전이 약 3개월마다 릴리스됩니다. Kubernetes 프로젝트는 3~4개의 직전 마이너 릴리스(1.29, 1.28, 1.27, 1.26)에 대한 릴리스 브랜치를 유지 관리합니다.

여기서 다루는 지원은 Cloud Native Computing Foundation(CNCF)에서 제공하는 Kubernetes 자체에 대한 지원입니다. 이것은 AWS가 Amazon EKS를 위해 제공하는 지원이 아닙니다. Kubernetes의 신규 버전에 대한 Amazon EKS 지원은 이와는 다른 일정을 따릅니다.

Amazon EKS의 Kubernetes 버전에 대한 표준 지원은 Amazon EKS에서 Kubernetes 버전이 릴리스될 때 시작되고 릴리스 날짜로부터 14개월 후에 종료됩니다. Kubernetes 버전에 대한 연장 지원은 표준 지원 종료 직후에 시작되고 12개월 후에 종료됩니다. 예를 들어 Amazon EKS의 버전 1.28에 대한 표준 지원은 2024년 11월에 종료됩니다. 버전 1.28에 대한 연장 지원은 2024년 11월에 시작되고 2025년 11월에 종료됩니다.

Amazon EKS 연장 지원에 대한 자세한 내용은

<https://aws.amazon.com/blogs/containers/amazon-eks-extended-support-for-kubernetes-versions-pricing/>을 참조하십시오.

다양한 Kubernetes 구성 요소 간에 지원되는 최대 버전 차이에 대한 자세한 내용은 Kubernetes 온라인 설명서의 ‘Version Skew Policy’ (<https://kubernetes.io/docs/setup/release/version-skew-policy/>)를 참조하십시오.

업그레이드 시퀀스

1 릴리스 노트 검토

5 클러스터 제어 영역 업그레이드

2 클러스터 백업(선택 사항)

6 추가 기능 검토 및 업그레이드

3 API 변경 사항 식별

7 kubectl 업그레이드

4 노드 그룹 버전 확인

8 클러스터 데이터 영역 업그레이드



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

| 수강생용 노트

Kubernetes 버전에는 제어 영역 및 데이터 영역이 모두 포함됩니다. 제어 영역과 데이터 영역 모두 동일한 **Kubernetes** 버전을 실행하는 것이 좋습니다.

- 제어 영역 - 제어 영역 버전은 **Kubernetes API** 서버에 따라 결정됩니다. **Amazon EKS** 클러스터에서는 **AWS API**를 통해 제어 영역 업그레이드를 시작할 수 있습니다.
- 데이터 영역 - 데이터 영역 버전은 개별 노드에서 실행되는 **Kubelet** 버전과 연결됩니다. 동일한 클러스터에 다른 버전을 실행하는 노드가 있을 수 있습니다. **kubectl get nodes**를 실행하면 모든 노드의 버전을 확인할 수 있습니다.

클러스터를 업그레이드하려면 다음 작업을 수행해야 합니다.

1. **Kubernetes** 및 **EKS** 릴리스 노트를 검토합니다.
2. 필요에 따라 클러스터를 백업합니다. 이에 대한 두 가지 제안이 있습니다.
 - **Git** 리포지토리에 클러스터 구성을 저장하여 **GitOps** 접근 방식을 사용합니다.
 - **Velero**를 사용합니다. 자세한 내용은 <https://aws.amazon.com/blogs/containers/backup-and-restore-your-eks-cluster/>

[amazon-eks-cluster-resources-using-velero/를 참조하십시오.](#)

3. 워크로드에서 사용 중단 또는 제거된 API 사용을 식별하고 해결합니다. 자세한 내용은 다음 설명서를 참조하십시오.
 - ‘Kubernetes Deprecation Policy’ (<https://kubernetes.io/docs/reference/using-api/deprecation-policy/>)
 - ‘Deprecated API Migration Guide’ (<https://kubernetes.io/docs/reference/using-api/deprecation-guide/>)
4. 모범 사례로, 노드 그룹을 사용하는 경우 제어 영역과 동일한 Kubernetes 버전이어야 합니다.
5. AWS 콘솔 또는 CLI를 사용하여 클러스터 제어 영역을 업그레이드합니다.
6. 추가 기능 호환성을 검토합니다. 필요에 따라 Kubernetes 추가 기능 및 사용자 지정 컨트롤러를 업그레이드합니다. 자세한 내용은 <https://aws.github.io/eks-best-practices/upgrades/#upgrade-add-ons-and-components-using-the-kubernetes-api>를 참조하십시오.
7. kubectl을 업데이트합니다.
8. 클러스터 데이터 영역을 업그레이드합니다. 노드를 업그레이드된 클러스터와 동일한 Kubernetes 버전으로 업그레이드합니다. 자세한 내용은 <https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>를 참조하십시오.

자세한 내용은 <https://aws.github.io/eks-best-practices/upgrades/>를 참조하십시오.

클러스터의 Kubernetes 버전을 업그레이드하는 동안 어떤 일이 발생합니까?

- 새 API 서버 노드의 배포
- 실패한 경우 자동 롤백
- 마이너 서비스 중단 가능성 있음
- 노드 또는 **Kubernetes** 추가 기능은 변경되지 않은 상태로 유지될 수 있음



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

| 수강생용 노트

Amazon EKS에 새 **Kubernetes** 버전이 제공되면 클러스터를 최신 버전으로 업그레이드할 수 있습니다. Amazon EKS를 사용하는 것의 장점 중 하나는 관리형 서비스라는 점입니다. Amazon EKS는 제어 영역의 업그레이드 프로세스를 관리합니다. 사전 요구 사항을 충족하고 간단한 명령을 실행하기만 하면 됩니다.

API 서버 노드

업그레이드 시 Amazon EKS는 업그레이드된 **Kubernetes** 버전으로 새 API 서버 노드를 시작하여 기존 노드를 대체합니다. Amazon EKS는 이러한 새 노드의 네트워크 트래픽에 대한 표준 인프라 및 준비 상태 확인을 수행하여 예상대로 작동하는지 확인합니다.

자동 롤백

이러한 검사 중 하나라도 실패할 경우 Amazon EKS가 인프라 배포를 되돌리고, 클러스터는 이전 **Kubernetes** 버전으로 남아 있게 됩니다. 실행 중인 애플리케이션은 영향을 받지 않으며 클러스터는 복구할 수 없는 상태로 남아 있지 않습니다. Amazon EKS는 모든 관리형 클러스터를 정기적으로 백업하며 필요한 경우 클러스터를 복구하는 메커니즘이 존재합니다.

マイ너 서비스 중단 가능성 있음

클러스터를 업그레이드하기 위해 **Amazon EKS**는 클러스터를 생성할 때 제공된 서브넷의 IP 주소 2~3개가 필요합니다. 이 서브넷에 사용 가능한 IP 주소가 없으면 업그레이드에 실패할 수 있습니다. 또한 클러스터 생성 동안 제공된 서브넷 또는 보안 그룹이 삭제된 경우 클러스터 업그레이드 프로세스에 실패할 수 있습니다.

노드 및 **Kubernetes** 추가 기능은 변경되지 않은 상태로 유지됨

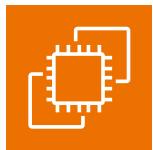
Amazon EKS는 클러스터를 업그레이드할 때 노드, **Amazon EKS** 추가 기능 또는 **Kubernetes** 추가 기능을 수정하지 않습니다.

다음 조건에 해당하는 경우 클러스터를 업그레이드하기 전에 모든 노드를 클러스터의 현재 업그레이드 전 **Kubernetes** 마이너 버전으로 업그레이드하십시오.

- 제어 영역에 비해 노드의 버전이 두 **Kubernetes** 마이너 버전보다 오래되었습니다.

클러스터를 업그레이드한 후 필요에 따라 추가 기능을 업그레이드합니다.

클러스터의 노드를 업그레이드하려면 어떻게 해야 합니까?



셀프 매니지드 노드



관리형 노드 그룹



AWS Fargate

- AWS 기본 도구 또는 서드 파티 도구를 사용하여 업그레이드합니다.

- AWS Management Console 또는 eksctl을 사용하여 업그레이드합니다.

- 업그레이드가 필요하지 않습니다.



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

| 수강생용 노트

Amazon EKS는 Amazon EKS 제어 영역을 업그레이드할 때 **Kubernetes** 노드를 업그레이드하지 않습니다. Amazon EKS 노드를 업데이트할 책임은 사용자에게 있습니다. 노드를 업그레이드하는 절차는 사용 중인 컴퓨팅 리소스의 유형(자체 관리형 노드, 관리형 노드 그룹 또는 **AWS Fargate**)에 따라 다릅니다. 업그레이드는 관리형 서비스의 복잡성을 크게 줄이는 또 하나의 방법입니다.

Amazon EKS 팀은 AWS가 지원하는 각 **Kubernetes** 버전에 해당하는 노드를 위해 일련의 Amazon EKS 최적화 AMI를 릴리스합니다. AMI의 목록은 Amazon EKS 사용 설명서의 ‘Amazon EKS 최적화 Amazon Linux AMI’

(<https://docs.aws.amazon.com/eks/latest/userguide/eks-optimized-ami.html>)를 참조하십시오.

Amazon EKS 노드에 사용할 자체 사용자 정의 AMI를 구축하는 경우 AWS는 구축 단계를 문서화하는 **Packer** 스크립트도 게시합니다. 이러한 스크립트를 사용하여 각 AMI 버전에 포함된 바이너리를 식별할 수 있습니다. Amazon EKS AMI 빌드 사양에 대해서는 GitHub의 ‘amazon-eks-ami’ 리포지토리 (<https://github.com/awslabs/amazon-eks-ami>)를 참조하십시오.

새로운 Amazon EKS 최적화 AMI가 릴리스되면 노드 그룹의 노드를 이 새 AMI로 교체하는 것을 고려해 보십시오. 마찬가지로, Amazon EKS 클러스터용 Kubernetes 버전을 업그레이드한 경우 노드도 업그레이드하여 동일한 Kubernetes 버전의 노드를 사용하십시오.

Amazon EKS 추가 기능 업그레이드

The screenshot shows two windows side-by-side. On the left is the 'Cluster info' page for an EKS cluster, listing the 'Add-ons' section which includes the 'Amazon VPC CNI' add-on. On the right is a modal window titled 'Configure Amazon VPC CNI' where the 'Version' dropdown is set to 'v1.12.5-eksbuild.2'. A purple box highlights the 'Update version' button on the cluster info page and the 'Save changes' button on the modal window.

~ALT text

~kube-proxy 추가 기능을 업데이트할 때 AWS 콘솔에서 수신되는 옵션이 강조표시된 스크린샷입니다.

|수강생용 노트

Amazon EKS 추가 기능을 사용하여 **Kubernetes** 애플리케이션을 지원하기 위한 핵심 기능을 제공하는 운영 소프트웨어를 구성, 배포, 업데이트할 수 있습니다. 모든 Amazon EKS 추가 기능에는 최신 보안 패치 및 버그 수정 사항이 포함되며, Amazon EKS에서 작동하도록 검증됩니다. Amazon EKS 추가 기능을 사용하면 Amazon EKS 클러스터의 보안과 안정성을 일관되게 유지할 수 있습니다. 또한 추가 기능을 사용하면 AWS에서 프로덕션 준비가 완료된 **Kubernetes** 클러스터를 시작하고 관리하는 데 필요한 작업이 줄어듭니다.

Amazon VPC CNI, kube-proxy, CoreDNS와 같은 일부 추가 기능에 대해 앞서 배웠습니다. 이러한 추가 기능은 Amazon EKS API, AWS Management Console, AWS CLI를 사용하여 업데이트할 수 있습니다.



Running Containers on Amazon EKS

지식 확인



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

지식 확인 1

노드 IAM 역할이 부여한 권한을 사용하여 AWS API에 요청을 수행할 수 있는 구성 요소는 무엇입니까? (2개 선택)

보기	응답
A	제어 영역에서 실행되는 API 서버
B	각 노드에서 실행되는 <code>kubelet</code>
C	관리 호스트에서 실행되는 <code>kubectl</code>
D	노드의 Pod에서 실행되는 애플리케이션
E	관리 호스트에서 실행되는 <code>eksctl</code>

지식 확인 1: 정답은 B와 D입니다.

노드 IAM 역할이 부여한 권한을 사용하여 AWS API에 요청을 수행할 수 있는 구성 요소는 무엇입니까? (2개 선택)

보기	응답
A	제어 영역에서 실행되는 API 서버
B	각 노드에서 실행되는 <code>kubelet</code>
C	관리 호스트에서 실행되는 <code>kubectl</code>
D	노드의 Pod에서 실행되는 애플리케이션
E	관리 호스트에서 실행되는 <code>eksctl</code>

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

| 수강생용 노트

정답은 B와 D입니다. 노드 IAM 역할에 의해 부여된 권한은 호스트 수준에서 적용됩니다. 데이터 영역 또는 제어 영역의 다른 호스트에서 실행되는 명령에는 적용되지 않습니다. 하지만 해당 호스트에서 실행 중인 Pod에 적용된다는 것을 기억해야 합니다.

지식 확인 2:

`eksctl`을 사용하여 클러스터가 생성되는 AWS 리전을 지정하려면 어떻게 합니까? (2개 선택)

보기	응답
A	클러스터가 생성된 후 클러스터 속성을 편집합니다.
B	명령을 실행할 때 <code>--region</code> 플래그를 사용합니다.
C	사용자 지정된 <code>cluster.yaml</code> 파일을 사용합니다.
D	클러스터를 생성하려는 리전에 태그를 추가합니다.
E	<code>kubectl</code> 을 사용하여 리전을 지정하는 <code>Kubernetes</code> 레이블을 추가합니다.

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

| 수강생용 노트

정답은 B와 C입니다. 명령을 실행할 때 `--region` 플래그를 사용하거나 특정 속성을 가진 `cluster.yaml` 파일을 만들어 `eksctl`의 동작을 사용자 정의할 수 있습니다.

지식 확인 2: 정답은 B와 C입니다.

`eksctl`을 사용하여
클러스터가 생성되는 AWS
리전을 지정하려면 어떻게
합니까? (2개 선택)

보기	응답
A	클러스터가 생성된 후 클러스터 속성을 편집합니다.
B	명령을 실행할 때 <code>--region</code> 플래그를 사용합니다.
C	사용자 지정된 <code>cluster.yaml</code> 파일을 사용합니다.
D	클러스터를 생성하려는 리전에 태그를 추가합니다.
E	<code>kubectl</code> 을 사용하여 리전을 지정하는 Kubernetes 레이블을 추가합니다.

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

| 수강생용 노트

정답은 B와 C입니다. 명령을 실행할 때 `--region` 플래그를 사용하거나 특정 속성을
가진 `cluster.yaml` 파일을 만들어 `eksctl`의 동작을 사용자 정의할 수 있습니다.

지식 확인 3

API를 제거할 수 있는
Kubernetes 버전 업그레이드의
최소 범주는 무엇입니까?

보기	응답
A	메이저 버전 업그레이드
B	마이너 버전 업그레이드
C	패치 버전 업그레이드
D	API는 사용 중단될 수 있지만 절대 제거되지 않음



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

지식 확인 3: 정답은 B입니다.

API를 제거할 수 있는
Kubernetes 버전 업그레이드의
최소 범주는 무엇입니까?

보기	응답
A	메이저 버전 업그레이드
B	마이너 버전 업그레이드
C	패치 버전 업그레이드
D	API는 사용 중단될 수 있지만 절대 제거되지 않음

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

| 수강생용 노트

정답은 B입니다. Kubernetes 프로젝트에서는 이러한 업그레이드를 마이너 버전이라고 설명하지만 이 업그레이드는 프로덕션에 커다란 영향을 미칠 수 있습니다.

모듈 요약



aws

이 모듈에서 학습한 내용:

- 필요에 따라 클러스터를 생성하는 데 가장 적합한 인터페이스를 선택합니다.
 - `eksctl`
 - AWS Management Console
 - AWS CLI
- 선택한 방법에 대한 전제 조건을 숙지합니다.
- Kubernetes** 버전 업그레이드는 신중하게 계획해야 합니다.
- 관리형 노드 그룹 또는 AWS Fargate를 사용하여 Kubernetes 버전 업그레이드를 간소화할 수 있습니다.

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

| 강사용 노트

| 수강생용 노트



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.