

스프링부트로 RestFulAPI 구현하기

# 14장 EKS

- 클러스터 생성
- IAM 권한 설정

박명회

## 14장 EKS란?

Amazon Elastic Kubernetes Service(Amazon EKS)는 Amazon Web Services(AWS)에 Kubernetes 컨트롤 플레인을 설치, 운영 및 유지 관리할 필요가 없는 관리형 서비스입니다. Kubernetes는 컨테이너화된 애플리케이션의 관리, 규모 조정 및 배포를 자동화하는 오픈 소스 시스템입니다.

보안 네트워킹 및 인증 -Amazon EKS는 Kubernetes 워크로드를 AWS 네트워킹 및 보안 서비스와 통합합니다. 또한 AWS Identity and Access Management(IAM)와의 통합으로 Kubernetes 클러스터에 대한 인증을 제공합니다.

간편한 클러스터 규모 조정 -Amazon EKS를 사용하면 워크로드 수요에 따라 Kubernetes 클러스터 규모를 쉽게 조정할 수 있습니다. Amazon EKS는 CPU 또는 사용자 지정 지표를 기반으로 수평 Pod 자동 규모 조정, 그리고 전체 워크로드 수요를 기반으로 클러스터 자동 규모 조정을 지원합니다.

관리형 Kubernetes 경험 - eksctl, AWS Management Console, AWS Command Line Interface(AWS CLI), API, kubectl 및 Terraform을 사용하여 Kubernetes 클러스터를 변경할 수 있습니다.

높은 가용성 - Amazon EKS는 여러 가용 영역의 컨트롤 플레인에 대한 고가용성을 제공합니다.

AWS 서비스와 통합 - Amazon EKS는 다른 AWS 서비스와 통합되어, 컨테이너화된 애플리케이션을 배포하고 관리하기 위한 포괄적인 플랫폼을 제공합니다. 또한 다양한 관찰성 도구를 통해 Kubernetes 워크로드 문제를 손쉽게 해결할 수 있습니다.

## 14장 EKSCTL

eksctl은 AWS의 관리형 쿠버네티스 서비스인 EKS 클러스터를 생성하고 관리하는 CLI 도구입니다.

내부적으로는 CloudFormation 이라는 AWS의 리소스 프로비저닝 서비스를 사용하여, EKS 구성에 필요한 EC2, IAM Role 등의 AWS 리소스를 자동으로 생성합니다. eksctl을 사용하기 위해서는 클러스터에 접속하기 위한 서버로 사용되는 bastion host에 AWS의 리소스를 사용하기 위한 AWS CLI 설정이 되어 있어야 하고, 설정한 사용자(User) 혹은 역할(Role)이 [eksctl을 사용하는데 필요한 IAM 권한](#)을 가지고 있어야 합니다.

eksctl을 사용하여 EKS 클러스터를 구성하는 방법을 살펴보겠습니다. 클러스터에 접근하기 위한 bastion host로 AWS EC2 (Amazon Linux 2, t2.micro)를 1대 생성하여 사용하였습니다.

## 14장 EKSCTL

eksctl은 AWS의 관리형 쿠버네티스 서비스인 EKS 클러스터를 생성하고 관리하는 CLI 도구입니다.

내부적으로는 CloudFormation 이라는 AWS의 리소스 프로비저닝 서비스를 사용하여, EKS 구성에 필요한 EC2, IAM Role 등의 AWS 리소스를 자동으로 생성합니다. eksctl을 사용하기 위해서는 클러스터에 접속하기 위한 서버로 사용되는 bastion host에 AWS의 리소스를 사용하기 위한 AWS CLI 설정이 되어 있어야 하고, 설정한 사용자(User) 혹은 역할(Role)이 [eksctl을 사용하는데 필요한 IAM 권한](#)을 가지고 있어야 합니다.

eksctl을 사용하여 EKS 클러스터를 구성하는 방법을 살펴보겠습니다. 클러스터에 접근하기 위한 bastion host로 AWS EC2 (Amazon Linux 2, t2.micro)를 1대 생성하여 사용하였습니다.

### 학습

- AWS CLI 사용 방법([https://docs.aws.amazon.com/ko\\_kr/cli/latest/userguide/getting-started-install.html](https://docs.aws.amazon.com/ko_kr/cli/latest/userguide/getting-started-install.html))
- AWS IAM에 대한 이해
- 쿠버네티스의 기본 아키텍처 개념

## 14장 kubectl 및 eksctl 설정

먼저 쿠버네티스의 API 서버와 통신하기 위한 CLI 도구인 kubectl 바이너리를 설치합니다. `curl` 로 바이너리 파일을 받은 다음, 실행 권한 및 PATH 설정을 해 줍니다. 각 설치 단계에 대해 상세히 알고 싶다면 [kubectl 설치](https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/install-kubectl.html) 문서를 참고합니다.

[https://docs.aws.amazon.com/ko\\_kr/eks/latest/userguide/install-kubectl.html](https://docs.aws.amazon.com/ko_kr/eks/latest/userguide/install-kubectl.html)

The screenshot shows the AWS documentation page for installing kubectl and eksctl. The page is in Korean and features a sidebar with navigation links, a main content area with a title and introductory text, and a right-hand sidebar with a table of contents. A blue callout box at the top of the main content area encourages user feedback. The main content area includes a title 'kubectl 및 eksctl 설정', a 'PDF' and 'RSS' link, and a paragraph explaining that kubectl is a command-line tool for interacting with the Kubernetes API server. The right-hand sidebar lists the topics covered in the page: 'kubectl 설치 또는 업데이트', 'eksctl 설치', and '다음 단계'.

aws docs.aws.amazon.com/ko\_kr/eks/latest/userguide/install-kubectl.html

본 가이드 검색

문의하기 한국어 콘솔로 돌아가기

AWS > aws-documentation > Amazon EKS > 사용자 가이드

피드백 설정

Amazon EKS 사용자 가이드

Amazon EKS란 무엇입니까?

▼ 설정

AWS CLI 설정

**kubectl 및 eksctl 설정**

빠른 시작

▶ Amazon EKS 시작하기

▶ 클러스터

▶ 액세스 관리

▶ 노드

▶ 앱 데이터 저장

▶ 네트워킹

▶ 워크로드

▶ 클러스터 관리

이 페이지 개선에 도움 주기

이 사용자 설명서에 기여하고 싶으신가요? 이 페이지 하단으로 스크롤하여 **GitHub에서 이 페이지 편집**을 선택하세요. 여러분의 기여는 모두를 위한 더 나은 사용자 설명서를 만드는 데 도움이 됩니다.

## kubectl 및 eksctl 설정

PDF RSS

Kubectl은 Kubernetes API 서버와 통신하기 위해 사용하는 명령줄 도구입니다. 이 **kubectl** 바이너리는 많은 운영 체제 패키지 관리자에서 사용할 수 있습니다. 설치에 패키지 관리자를 사용하는 것이 수동 다운로드 및 설치 프로세스보다 쉬운 경우가 많습니다. **eksctl** 명령을 사용하면 Amazon EKS 클러스터를 생성하고 수정할 수 있습니다.

이 페이지의 주제는 다음 도구를 설치하고 설정하는 데 도움이 됩니다.

- [kubectl 설치 또는 업데이트](#)
- [eksctl 설치](#)

이 페이지에서

- [kubectl 설치 또는 업데이트](#)
- [eksctl 설치](#)
- [다음 단계](#)

## 14장 kubectl설치

```
$ curl -o kubectl
https://amazon-eks.s3.us-west-2.amazonaws.com/1.21.2/2021-07-05/bin/linux/amd64/kubectl
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left  Speed
100 44.2M  100 44.2M    0     0 2651k      0  0:00:17  0:00:17 --:--:-- 3769k

$ chmod +x ./kubectl

$ mkdir -p $HOME/bin && cp ./kubectl $HOME/bin/kubectl && export PATH=$PATH:$HOME/bin

$ echo 'export PATH=$PATH:$HOME/bin' >> ~/.bashrc

$ kubectl version --short --client
Client Version: v1.21.2-13+d2965f0db10712
```

## 14장 eksctl설치

```
$ 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
```

```
$ eksctl version  
0.94.0
```

## 14장 AWS IAM 설정

EKS와 연관된 AWS 리소스를 생성하기 위해서는 리소스 생성을 요청하는 사용자(User) 혹은 역할(Role)이 그 리소스를 생성할 수 있는 IAM 권한을 가지고 있어야 합니다. 루트 계정이 아니라면 해당 IAM 사용자 혹은 역할이 다음 네가지 정책을 가지고 있어야 합니다.

- AmazonEC2FullAccess (AWS Managed Policy)
- AWSCloudFormationFullAccess (AWS Managed Policy)
- EksAllAccess
- iamFullAccess

4가지 중 위 3개 정책은 AWS Managed 정책이므로 이미 생성되어 있습니다. 따라서 EksAllAccess 1개 정책만 별도로 생성합니다. [IAM 정책 생성](#)의 JSON 탭에서 정책 만들기를 참고하여 1개 정책을 추가합니다. 정책의 JSON 문서는 [Minimum IAM policies](#)에서 확인할 수 있습니다.

그 다음, 4가지 정책을 eksctl을 사용하려는 [사용자, 사용자그룹 혹은 역할 자격 증명에 추가](#)합니다.



## 14장 AWS IAM 설정

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "eks:*",
      "Resource": "*"
    },
    {
      "Action": [
        "ssm:getParameter",
        "ssm:getParameters"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
  ]
}
```

## 14장 AWS CLI 명령어 사용자 생성

모든 권한이 부여되었다면, 이제 bastion host에 AWS CLI 명령을 사용하기 위한 구성 및 자격 증명 파일을 설정할 차례입니다. *aws configure* 명령을 사용하며, 설정할 때에는 액세스 키 및 비밀 액세스 키 정보가 필요합니다. 키 정보를 확인하고 생성하는 방법은 [IAM 사용자의 액세스 키 관리](#) 문서의 액세스 키 관리(콘솔) 섹션을 참고합니다.

The screenshot shows the AWS IAM console interface. The top navigation bar includes the AWS logo, a search bar, and service links like EC2, Elastic Kubernetes Service, Elastic Container Service, CloudFormation, IAM, RDS, S3, and Route 53. The left sidebar shows the 'IAM' menu with 'Users' selected, and a breadcrumb trail 'IAM > 사용자 > 사용자 생성'. The main content area is titled '사용자 세부 정보 지정' (Specify user details). It contains a form for '사용자 세부 정보' (User details) with a text input for '사용자 이름' (User name) containing 'eks040808'. Below the input is a note: '사용자 이름은 최대 64자까지 가능합니다. 유효한 문자: A~Z, a~z, 0~9 및 +, =, ., @, \_ (하이픈)'. There is an unchecked checkbox for 'AWS Management Console에 대한 사용자 액세스 권한 제공 - 선택 사항' (Provide user access to the AWS Management Console - optional) with a note: '사용자에게 콘솔 액세스 권한을 제공하는 경우 IAM Identity Center에서 액세스를 관리하는 것은 모범 사례입니다.' At the bottom of the form is a blue information box: '이 IAM 사용자를 생성한 후 액세스 키 또는 AWS CodeCommit이나 Amazon Keyspaces에 대한 서비스별 보안 인증 정보를 통해 프로그래밍 방식 액세스를 생성할 수 있습니다. 자세히 알아보기'. At the bottom right are '취소' (Cancel) and '다음' (Next) buttons.

aws 서비스 검색 [알트+S]

EC2 Elastic Kubernetes Service Elastic Container Service CloudFormation IAM RDS S3 Route 53

IAM > 사용자 > 사용자 생성

1단계  
사용자 세부 정보 지정

2단계  
권한 설정

3단계  
검토 및 생성

### 사용자 세부 정보 지정

#### 사용자 세부 정보

사용자 이름

eks040808

사용자 이름은 최대 64자까지 가능합니다. 유효한 문자: A~Z, a~z, 0~9 및 +, =, ., @, \_ (하이픈)

☐ AWS Management Console에 대한 사용자 액세스 권한 제공 - 선택 사항  
사용자에게 콘솔 액세스 권한을 제공하는 경우 IAM Identity Center에서 액세스를 관리하는 것은 모범 사례입니다.

이 IAM 사용자를 생성한 후 액세스 키 또는 AWS CodeCommit이나 Amazon Keyspaces에 대한 서비스별 보안 인증 정보를 통해 프로그래밍 방식 액세스를 생성할 수 있습니다. [자세히 알아보기](#)

취소 다음

# 14장 AWS CLI 명령어 사용자 생성

aws

서비스

검색

[알트+S]

EC2

Elastic Kubernetes Service

Elastic Container Service

CloudFormation

IAM

RDS

S3

Route 53

IAM > 사용자 > 사용자 생성

1단계

사용자 세부 정보 지정

2단계

권한 설정

3단계

검토 및 생성

권한 설정

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 직무별로 사용자의 권한을 관리하려면 그룹을 사용하는 것이 좋습니다. 자세히 알아보기

권한 옵션

☒ 그룹에 사용자 추가

기존 그룹에 사용자를 추가하거나 새 그룹을 생성합니다. 그룹을 사용하여 직무별로 사용자 권한을 관리하는 것이 좋습니다.

☐ 권한 복사

기존 사용자의 모든 그룹 멤버십, 연결된 관리형 정책 및 인라인 정책을 복사합니다.

☐ 직접 정책 연결

관리형 정책을 사용자에게 직접 연결합니다. 사용자에게 연결하는 대신, 정책을 그룹에 연결한 후 사용자를 적절한 그룹에 추가하는 것이 좋습니다.

사용자 그룹 (1/2)

그룹 생성

검색

|                                     | 그룹 이름     | 사용자 | 연결된 정책                                    | 생성됨                |
|-------------------------------------|-----------|-----|---|--------------------|
| <input type="checkbox"/>            | 3_3       | 0   | -   | 2023-04-12 (1년 전)  |
| <input checked="" type="checkbox"/> | eksmaster | 2   | AmazonEC2FullAccess, IAMFullAccess 및 1개 더 | 2024-06-12 (1개월 전) |

## 14장 AWS CLI 명령어 사용자 생성

Identity and Access Management(IAM)

Q IAM 검색

대시보드

▼ 액세스 관리

사용자 그룹

사용자

역할

정책

자격 증명 공급자

계정 설정

▼ 보고서 액세스

Access Analyzer

외부 액세스

미사용 액세스

보통기 서적

☑ 사용자가 성공적으로 생성됨

사용자 보기

AWS Management Console에 로그인하기 위한 사용자의 암호와 이메일 지침을 보고 다운로드할 수 있습니다.

IAM > 사용자

사용자 (23) 정보

↻ 삭제 사용자 생성

Q eks X 3 개 일치

< 1 > ⚙

| <input type="checkbox"/> | 사용자 이름 ▲                  | 경로 ▼ | 그룹 ▼ | 마지막 활동 ▼ | MFA ▼ | 암호 수명 ▼ | 콘솔 마지막 로그인 ▼ | 액세스 키 ID ▼              | 활성 키 수명 |
|--------------------------|---------------------------|------|------|----------|-------|---------|--------------|-------------------------|---------|
| <input type="checkbox"/> | <a href="#">eks040808</a> | /    | 1    | -        | -     | -       | -            | -                       | -       |
| <input type="checkbox"/> | <a href="#">eks0613</a>   | /    | 1    | ☑ 55일 전  | -     | -       | -            | Active - AKIAU5ITWDP... | ☑ 55일   |
| <input type="checkbox"/> | <a href="#">eksuser</a>   | /    | 1    | ☑ 56일 전  | -     | -       | -            | Active - AKIAU5ITWDP... | ☑ 56일   |

## 14장 AWS CLI 설치 및 설정

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html>

`snap install aws-cli --classic`

2. Run the following `snap install` command for the AWS CLI.

```
$ snap install aws-cli --classic
```



Depending on your permissions, you may need to add `sudo` to the command.

```
$ sudo snap install aws-cli --classic
```



### Note

To view the snap repository for the AWS CLI, including additional `snap` instructions, see the [aws-cli](#) page in the *Canonical Snapcraft website*.

3. Verify that the AWS CLI installed correctly.

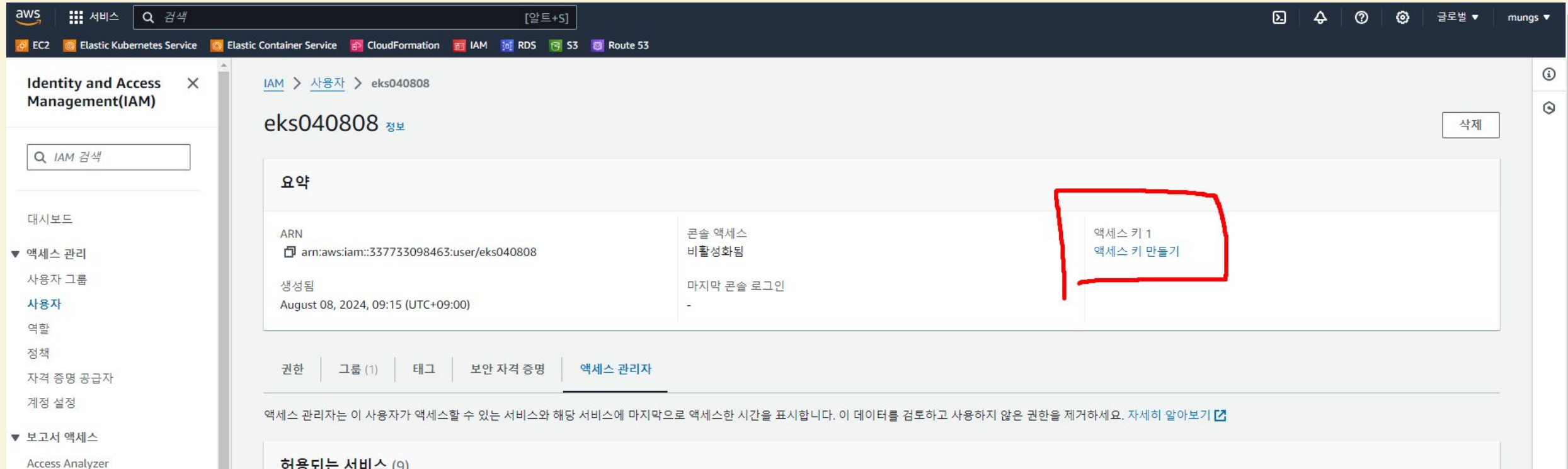
```
$ aws --version
aws-cli/2.17.20 Python/3.11.6 Linux/5.10.205-195.807.amzn2.x86_64 botocore/2.4.5
```



If you get an error, see [Troubleshoot AWS CLI errors](#).

모든 권한이 부여되었다면, 이제 bastion host에 AWS CLI 명령을 사용하기 위한 구성 및 자격 증명 파일을 설정할 차례입니다. `aws configure` 명령을 사용하며, 설정할 때에는 액세스 키 및 비밀 액세스 키 정보가 필요합니다. 키 정보를 확인하고 생성하는 방법은 [IAM 사용자의 액세스 키 관리](#) 문서의 액세스 키 관리(콘솔) 섹션을 참고합니다.

## 14장 AWS CLI 명령어 사용자 생성



The screenshot displays the AWS IAM console interface. The top navigation bar includes the AWS logo, a search bar, and various service icons. The left sidebar shows the 'Identity and Access Management(IAM)' section with a search bar and a list of navigation items: '대시보드', '액세스 관리', '사용자 그룹', '사용자', '역할', '정책', '자격 증명 공급자', '계정 설정', '보고서 액세스', and 'Access Analyzer'. The main content area shows the details for a user named 'eks040808'. The '요약' (Summary) section contains a table with the following information:

| ARN                                      | 콘솔 액세스 비활성화됨 |
|--|--------------|
| arn:aws:iam::337733098463:user/eks040808 |              |

Below the table, the '생성됨' (Created) date is listed as 'August 08, 2024, 09:15 (UTC+09:00)'. The '마지막 콘솔 로그인' (Last console login) is shown as '-'. To the right of the table, a red box highlights the '액세스 키 1' (Access Key 1) section, which includes a link to '액세스 키 만들기' (Create Access Key). Below the summary section, there are tabs for '권한' (Permissions), '그룹 (1)' (Groups), '태그' (Tags), '보안 자격 증명' (Security Credentials), and '액세스 관리자' (Access Manager). The '액세스 관리자' tab is currently selected. Below the tabs, a message states: '액세스 관리자는 이 사용자가 액세스할 수 있는 서비스와 해당 서비스에 마지막으로 액세스한 시간을 표시합니다. 이 데이터를 검토하고 사용하지 않은 권한을 제거하세요. 자세히 알아보기' (Access Manager displays the services and the last time the user accessed the services. Review this data and remove permissions that you are not using. Learn more). At the bottom, there is a section titled '허용되는 서비스 (9)' (Services allowed (9)).

# 14장 AWS CLI 명령어 사용자 생성

aws

서비스

검색

[알트+S]

EC2Elastic Kubernetes ServiceElastic Container ServiceCloudFormationIAMRDSSES3Route 53

1단계

액세스 키 모범 사례 및 대안

2단계 - 선택 사항

설명 태그 설정

3단계

액세스 키 검색

IAM > 사용자 > eks040808 > 액세스 키 만들기

액세스 키 모범 사례 및 대안

정보

보안 개선을 위해 액세스 키와 같은 장기 자격 증명을 사용하지 마세요. 다음과 같은 사용 사례와 대안을 고려하세요.

사용 사례

☐ Command Line Interface(CLI)  
AWS CLI를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ 로컬 코드  
로컬 개발 환경의 애플리케이션 코드를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☒ AWS 컴퓨팅 서비스에서 실행되는 애플리케이션  
Amazon EC2, Amazon ECS 또는 AWS Lambda와 같은 AWS 컴퓨팅 서비스에서 실행되는 애플리케이션 코드를 사용하여 AWS 계정에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ 서드 파티 서비스  
AWS 리소스를 모니터링 또는 관리하는 서드 파티 애플리케이션 또는 서비스에 액세스할 수 있도록 이 액세스 키를 사용할 것입니다.

☐ AWS 외부에서 실행되는 애플리케이션  
이 액세스 키를 사용하여 AWS 리소스에 액세스해야 하는 AWS 외부의 데이터 센터 또는 기타 인프라에서 실행 중인 워크로드를 인증할 것입니다.

☐ 기타  
귀하의 사용 사례가 여기에 나열되어 있지 않습니다.

권장되는 대안

## 14장 AWS CLI 명령어 사용자 생성

aws

서비스

검색

[알트+S]

EC2Elastic Kubernetes ServiceElastic Container ServiceCloudFormationIAMRDSSS3Route 53

☰

🔍 액세스 키 생성됨

지금 아니면 비밀 액세스 키를 보거나 다운로드할 수 없습니다. 나중에 복구할 수 없습니다. 하지만 언제든지 새 액세스 키를 생성할 수 있습니다.

[IAM](#) > [사용자](#) > [eks040808](#) > 액세스 키 만들기

1단계

액세스 키 모범 사례 및 대안

2단계 - 선택 사항

설명 태그 설정

3단계

액세스 키 검색

액세스 키 검색 정보

액세스 키

본실하거나 잊어버린 비밀 액세스 키는 검색할 수 없습니다. 대신 새 액세스 키를 생성하고 이전 키를 비활성화합니다.

| 액세스 키                             | 비밀 액세스 키                              |
|-----------------------------------|---------------------------------------|
| <div>📄 AKIAUSITWDPPU6H6N2W2</div> | <div>📄 ***** <a href="#">표시</a></div> |

액세스 키 모범 사례

- 액세스 키를 일반 텍스트, 코드 리포지토리 또는 코드로 저장해서는 안 됩니다.
- 더 이상 필요 없는 경우 액세스 키를 비활성화하거나 삭제합니다.
- 최소 권한을 활성화합니다.
- 액세스 키를 정기적으로 교체합니다.

액세스 키 관리에 대한 자세한 내용은 [AWS 액세스 키 관리 모범 사례](#)를 참조하세요.

.csv 파일 다운로드

완료



## 14장 AWS CONFIGURE

```
$ aws configure
```

```
AWS Access Key ID [None]: asdasdasd # Sample Access Key
```

```
AWS Secret Access Key [None]: asdasdasdasd # Sample Secret Access Key
```

```
Default region name [None]: ap-northeast-2
```

```
Default output format [None]: json
```

자격 증명 정보가 올바르게 설정되었다면, 홈 디렉토리의 `.aws` 디렉토리 아래 2개 파일이 생성된 것을 확인할 수 있습니다. 구성 옵션은 `config` 파일에, 자격 증명 정보는 `credentials` 파일에 기록됩니다. `eksctl`은 이 정보를 사용하여 AWS 리소스 생성 요청을 하게 됩니다.

```
$ cat .aws/config
```

```
[default]
```

```
output = json
```

```
region = ap-northeast-2
```

```
$ cat .aws/credentials
```

```
[default]
```

```
aws_access_key_id = asdfasdf
```

```
aws_secret_access_key = asdfasdf
```

## 14장 EKS 클러스터 생성

모든 준비가 끝났습니다. 이제 `eksctl`을 사용하여 간단한 예제 EKS 클러스터를 생성해보겠습니다. `eksctl`을 사용하면 아래 한줄의 명령어로 EKS 클러스터를 생성할 수 있습니다.

```
$ eksctl create cluster --name my-sample-cluster --region ap-northeast-2
2024-08-08 00:20:36 [i] eksctl version 0.188.0
2024-08-08 00:20:36 [i] using region ap-northeast-2
2024-08-08 00:20:36 [i] setting availability zones to [ap-northeast-2a ap-northeast-2d ap-northeast-2c]
2024-08-08 00:20:36 [i] subnets for ap-northeast-2a - public:192.168.0.0/19 private:192.168.96.0/19
2024-08-08 00:20:36 [i] subnets for ap-northeast-2d - public:192.168.32.0/19 private:192.168.128.0/19
2024-08-08 00:20:36 [i] subnets for ap-northeast-2c - public:192.168.64.0/19 private:192.168.160.0/19
2024-08-08 00:20:36 [i] nodegroup "ng-dc1b98dc" will use "" [AmazonLinux2/1.30]
2024-08-08 00:20:36 [i] using Kubernetes version 1.30
2024-08-08 00:20:36 [i] creating EKS cluster "my-sample-cluster" in "ap-northeast-2" region with managed nodes
2024-08-08 00:20:36 [i] will create 2 separate CloudFormation stacks for cluster itself and the initial managed nodegroup
2024-08-08 00:20:36 [i] if you encounter any issues, check CloudFormation console or try 'eksctl utils describe-stacks
--region=ap-northeast-2 --cluster=my-sample-cluster'
2024-08-08 00:20:36 [i] Kubernetes API endpoint access will use default of {publicAccess=true, privateAccess=false} for cluster
"my-sample-cluster" in "ap-northeast-2"
2024-08-08 00:20:36 [i] CloudWatch logging will not be enabled for cluster "my-sample-cluster" in "ap-northeast-2"
2024-08-08 00:20:36 [i] you can enable it with 'eksctl utils update-cluster-logging
--enable-types={SPECIFY-YOUR-LOG-TYPES-HERE (e.g. all)} --region=ap-northeast-2 --cluster=my-sample-cluster'
```

맨 아래 EKS cluster is ready 문구가 출력되면 EKS 클러스터 생성이 정상적으로 완료된 것입니다. 클러스터가 Ready 상태가 되기까지 약 17분이 소요된 것을 확인할 수 있습니다.

## 14장 EKS 클러스터 생성

```
eksctl create cluster \  
--name my-sample-cluster \  
--region ap-northeast-2 \  
--with-oidc \  
--nodegroup-name my-sample-ng \  
--nodes 2 \  
--node-type t3.medium \  
--node-volume-size=20 \  
--managed
```

## 14장 CloudFormation 스택 확인

eksctl의 출력 로그를 살펴볼 수도 있지만, 보다 정확하게 어떤 AWS 리소스가 생성되었는지 확인하기 위해 AWS 웹 콘솔에서 CloudFormation 스택을 확인해봅시다.

The screenshot shows the AWS CloudFormation console. The left sidebar contains navigation links for CloudFormation, StackSet, and various AWS services. The main area displays a list of stacks. Two stacks are visible, both in the 'CREATE\_COMPLETE' state.

| 스택 이름  | 상태              | 생성 시간                        | 설명   |
|--|-----------------|------------------------------|--|
| <a href="#">eksctl-my-sample-cluster-nodegroup-ng-dc1b98dc</a> | CREATE_COMPLETE | 2024-08-08 09:29:40 UTC+0900 | EKS Managed Nodes (SSH access: false) [created by eksctl]                              |
| <a href="#">eksctl-my-sample-cluster-cluster</a>               | CREATE_COMPLETE | 2024-08-08 09:20:36 UTC+0900 | EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl] |

eksctl이 2개의 CloudFormation 스택을 생성했습니다. 생성 시간을 보면 -cluster로 끝나는 EKS 컨트롤 플레인을 프로비저닝하는 스택이 먼저 생성된 것을 확인할 수 있습니다. 이 스택부터 살펴보겠습니다.

## 14장 CloudFormation 스택 확인

무려 34개의 AWS 리소스가 EKS 클러스터를 프로비저닝하기 위해 생성된 것을 볼 수 있습니다. *EKS Cluster* 뿐만 아니라 클러스터가 위치할 리전의 *VPC*, 노드가 위치할 *Subnet*, 쿠버네티스 클러스터가 사용할 *IAM Role*과 *IAM Policy*, 인터넷 접근을 위한 *NAT Gateway*와 *Internet Gateway*, 컨트롤 플레인 전용 및 클러스터 전체 공유 2가지 *Security Group* 등이 생성되었습니다.

The screenshot displays the AWS CloudFormation console for the stack 'eksctl-my-sample-cluster-cluster'. The left sidebar shows the 'CloudFormation' navigation menu with options like '스택', '스택 세부 정보', '드리프트', 'StackSet', and '내보내기'. The main content area is divided into two panes. The left pane shows the 'Stacks' list with two stacks: 'eksctl-my-sample-cluster-nodegroup-ng-dc1b98dc' and 'eksctl-my-sample-cluster-cluster', both in a 'CREATE\_COMPLETE' state. The right pane shows the 'Resources' tab for the selected stack, listing 32 resources. The resources are organized into a table with columns for Logical ID, Physical ID, Type, Status, and Module.

| 논리적 ID                         | 물리적 ID                | 유형                             | 상태              | 모듈 |
|--------------------------------|-----------------------|--------------------------------|-----------------|----|
| ClusterSharedNodeSecurityGroup | sg-0834512ec2a421336  | AWS::EC2::SecurityGroup        | CREATE_COMPLETE | -  |
| ControlPlane                   | my-sample-cluster     | AWS::EKS::Cluster              | CREATE_COMPLETE | -  |
| ControlPlaneSecurityGroup      | sg-0e0f73f2e825f1c58  | AWS::EC2::SecurityGroup        | CREATE_COMPLETE | -  |
| IngressDefaultClusterToNodeSG  | sgr-033c395ac3caa0066 | AWS::EC2::SecurityGroupIngress | CREATE_COMPLETE | -  |
| IngressInterNodeGroupSG        | sgr-02705bca62e6602cd | AWS::EC2::SecurityGroupIngress | CREATE_COMPLETE | -  |
| IngressNodeToDefaultClusterSG  | sgr-0a6247d4321c84292 | AWS::EC2::SecurityGroupIngress | CREATE_COMPLETE | -  |
| InternetGateway                | igw-0cd06194e921b78a3 | AWS::EC2::InternetGateway      | CREATE_COMPLETE | -  |
| NATGateway                     | nat-096ce32c17777d24a | AWS::EC2::NatGateway           | CREATE_COMPLETE | -  |

## 14장 쿠버네티스 node 생성 및 클러스터 생성 확인

aws 서비스 검색 [알트+S]

EC2 Elastic Kubernetes Service Elastic Container Service CloudFormation IAM RDS S3 Route 53

인스턴스 (3) 정보

인스턴스를 속성 또는 (case-sensitive) 태그로 찾기

인스턴스 상태 = running 필터 지우기

| <input type="checkbox"/> | Name                               | 인스턴스 ID             | 인스턴스 상태 | 인스턴스 유형  | 상태 검사         | 경보 상태 | 가용 영역           | 퍼블릭 IPv4 DNS             | 퍼블릭 IPv4    |
|--------------------------|------------------------------------|---------------------|---------|----------|---------------|-------|-----------------|--------------------------|-------------|
| <input type="checkbox"/> | my-sample-cluster-ng-dc1b98dc-Node | i-0c6fba7d786fcaf7d | 실행 중    | m5.large | 2/2개 검사 통과... | 경보 보기 | ap-northeast-2d | ec2-43-202-251-143.ap... | 43.202.251. |
| <input type="checkbox"/> | eks테스트                             | i-051d2b36954982151 | 실행 중    | t2.micro | 2/2개 검사 통과... | 경보 보기 | ap-northeast-2a | ec2-3-35-207-111.ap-n... | 3.35.207.11 |
| <input type="checkbox"/> | my-sample-cluster-ng-dc1b98dc-Node | i-0f89e8c30bc6c0ba2 | 실행 중    | m5.large | 2/2개 검사 통과... | 경보 보기 | ap-northeast-2a | ec2-3-36-123-158.ap-n... | 3.36.123.15 |

```
ubuntu@ip-172-31-0-200:~$ aws eks list-clusters
{
  "clusters": [
    "my-sample-cluster"
  ]
}
```



## 14장 kubernetes NGINX

The screenshot displays the Visual Studio Code interface with the 'Kubernetes Templates' extension by lunuan installed. The left sidebar shows the 'EXTENSIONS: MARKETPLACE' view with a search for 'kubern snap'. The main panel shows the extension's details, including its version (v1.3.1), a rating of 5 stars, and an 'Install' button. Below the extension details, the 'Getting Started' section provides instructions on how to use the extension: creating a YAML file with a prefix like 'k8s' or 'kube.', and selecting a resource object from a pop-up list. A sample 'deployment.yaml' file is shown, illustrating the syntax for creating a Kubernetes Deployment. The right sidebar shows the 'Categories' section with 'Snippets' selected, and the 'Resources' section with links to the Marketplace, Repository, License, and the author's profile. The 'More Info' section provides details about the extension's publication date, last release, and identifier.

**EXTENSIONS: MARKETPLACE**

Search:

- Kubernetes** (4.4M, 4.5 stars) - Microsoft - [Install](#)
- Bridge to Kubernetes** (436K, 5 stars) - Microsoft - [Install](#)
- Kubernetes Support** (276K, 3 stars) - ipedrazas - [Install](#)
- Azure Kubernetes Service** (330K, 5 stars) - Microsoft - [Install](#)
- YAML** (17.6M, 3.5 stars) - Red Hat - [Install](#)
- Kubernetes Kind** (75K, 4 stars) - Microsoft - [Install](#)
- Kubernetes Templates** (104K, 5 stars) - lunuan - [Install](#)
- Kubernator** (10K, 5 stars) - SMP - [Install](#)

**Kubernetes Templates** v1.3.1  
lunuan | 104,409 | 5 stars (2)  
[Install](#) ☒ Auto Update [Settings](#)

**Getting Started**

- Create a YAML file, enter prefix `k8s` or `kube.`
- Select the resource object you want to create from the pop-up list

It's that simple

```
deployment.yaml
1 k8s
  k8sCronJob CronJob (kub...) Deployment (kubernetes template x
  k8sDaemonSet DaemonSet (k...
  k8sDeployment Deployment (k...
  k8sEndpoints Endpoints (k...
  k8sIngressTraefik Ingr...
  k8sJob Job (kuberne...
  k8sMiddleware Traefik Midd...
  k8sPersistentVo... PersistentVo...
  k8sPod Pod (kuberne...
  k8sSecret Secret (kub...
  k8sService Service (kub...
  k8sStatefulSet StatefulSet ...
```

**Categories**

[Snippets](#)

**Resources**

- [Marketplace](#)
- [Repository](#)
- [License](#)
- [lunuan](#)

**More Info**

|               |                             |
|---------------|-----------------------------|
| Published     | 2020-06-06, 16:50:14        |
| Last released | 2021-12-16, 01:41:21        |
| Identifier    | lunuan.kubernetes-templates |

## 14장 NGINX

### my-nginx.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-nginx
spec:
  selector:
    matchLabels:
      run: my-nginx
  replicas: 2
  template:
    metadata:
      labels:
        run: my-nginx
    spec:
      containers:
      -name: my-nginx
        image: nginx
        ports:
        -containerPort: 80
```



## 14장 NGINX

my-nginx-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-nginx
  namespace: default
spec:
  selector:
    app: my-nginx
  type: LoadBalancer
  ports:
    - name: my-nginx
      protocol: TCP
      port: 80
      targetPort: 80
```

## 14장 MYSQL

### my-mysql.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-mysql
  namespace: default
  labels:
    app: my-mysql
spec:
  selector:
    matchLabels:
      app: my-mysql
  replicas: 1
  template:
    metadata:
      labels:
        app: my-mysql
    spec:
      containers:
        - name: my-mysql
          image: mysql
          env:
            - name: MYSQL_ROOT_PASSWORD
              value: password
          ports:
            - containerPort: 3306
              name: my-mysql
```

## 14장 NGINX

my-mysql-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-mysql
  namespace: default
spec:
  selector:
    app: my-mysql
  type: LoadBalancer
  ports:
    - name: my-mysql
      protocol: TCP
      port: 3306
      targetPort: 3306
```

## 14장 NGINX-service

my-mysql-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-mysql
  namespace: default
spec:
  selector:
    app: my-mysql
  type: LoadBalancer
  ports:
    - name: my-mysql
      protocol: TCP
      port: 3306
      targetPort: 3306
```

## 14장 mysql-service

my-mysql-svc.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-mysql
  namespace: default
spec:
  selector:
    app: my-mysql
  type: LoadBalancer
  ports:
    - name: my-mysql
      protocol: TCP
      port: 3306
      targetPort: 3306
```

## 14장 springboot.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-springboot
  namespace: default
  labels:
    app: my-springboot
spec:
  selector:
    matchLabels:
      app: my-springboot
  replicas: 1
  template:
    metadata:
      labels:
        app: my-springboot
    spec:
      containers:
        - name: my-springboot
          image: parkmyoung Choi/restful
          env:
            - name: SPRING_DATASOURCE_PASSWORD
              value: password
            - name: SPRING_DATASOURCE_URL
              value: jdbc:mysql://10.100.223.91:3306/pmh
          ports:
            - containerPort: 8080
              name: my-springboot
```

## 14장 springboot-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: my-springboot
  namespace: default
spec:
  selector:
    app: my-springboot
  type: NodePort
  ports:
    - name: my-springboot
      protocol: TCP
      port: 8080
      targetPort: 8080
      nodePort: 30010
```

## 14장 EKS 클러스터 삭제

```
$ eksctl delete cluster --name my-sample-cluster --region ap-northeast-2
```

클러스터와 노드의 사용을 끝낸 후에는 다음 명령으로 클러스터를 삭제합니다. 그렇지 않으면 **EKS** 및 **EC2** 비용이 지속적으로 발생할 수 있습니다.



감사합니다