

MSA by AWS EKS

<목차>

<목차>

1. 프로젝트 개요

2. 프로젝트 주요 기능

3. 프로젝트 구성

 3.1. Architecture

 3.2. YAML

4. 프로젝트 구현

 4.1. Create EKS Cluster(Control Plane)

 4.1.1. Create & Associate IAM OIDC Provider for our EKS Cluster

 4.1.2. Create EC2 Key Pair

 4.2. Create EKS Node Group in Private Subnets

 4.5. Create RDS Database

 4.5.1. Create DB Security Group

 4.5.2. Create DB Subnet Group

 4.5.3. Create RDS

 4.5.4. Create Kubernetes externalName Service and Deploy

 4.4. AWS Load Balancer Controller

 4.4.3. Step-03: Create an IAM role for the AWS LoadBalancer Controller and attach the role to the Kubernetes service account

 4.4.4. Step-03-02: Verify using eksctl cli

 4.5. External DNS

 4.5.1. Step-02: Create IAM Policy

 4.5.3. Deploy External DNS

Route 53 (도메인 등록)

AWS Certificated Manager (SSL/TLS 인증서 등록)

 4.8. Amazon Simple Email Service

 4.9. Deploy SMTP ExternalName Service

CloudWatch

5. 이슈리스트

6. References

1. 프로젝트 개요

- 프로젝트 제목: MSA by AWS EKS
- 프로젝트 개요: Kubernetes & AWS EKS 및 다양한 AWS 서비스를 활용한 클라우드 인프라 설계 및 구현 목적

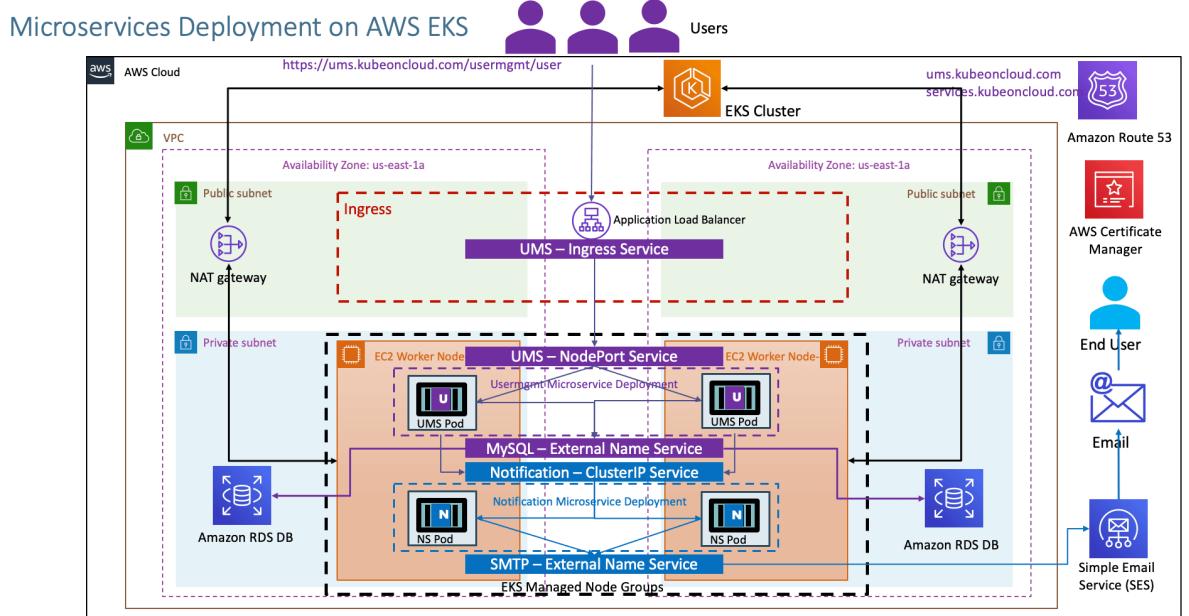
- 프로젝트 주제 선정 목적: k8s와 AWS 서비스를 혼합하여 사용하여 두 가지 개념에 대해 친숙해지며 서버 증설 시간이 오래걸려 트래픽 대응에 좋지못한 모놀로식 아키텍처가 아닌 인프라 운영 및 애플리케이션 배포에 이점이 있는 마이크로 서비스 아키텍처로 설계 하려고 시도해보았다.
- 프로젝트 구성원: 김성년 (1명)

2. 프로젝트 주요 기능

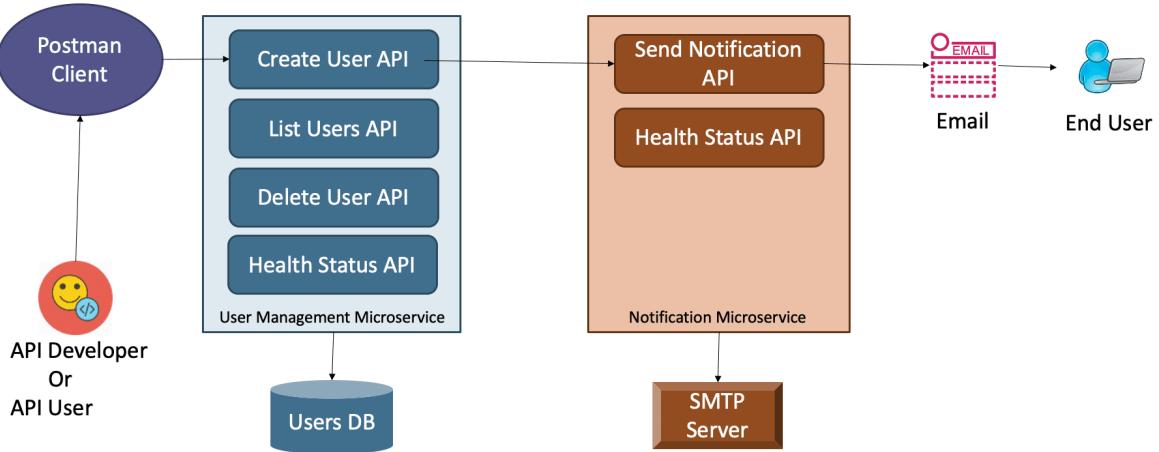
- k8s
 - Service: ingress / nodePort/ ExternalName / ClusterIP
 - Deployment / DeamonSet (CloudWatch)
 - externalDNS
- EKS:
 - 완전 관리형 서비스로 최소한의 인원 및 운영부담 최소화 가능. 또한 손쉽게 AWS의 다른 서비스들과 통합 가능
 - 배포 자동화와 서비스간의 통신, 문제가 발생했을 경우 자가 복구 등 인프라 운영에 효과적인 기능 제공
- ALB
- IAM
- Route 53 & AWS Certifcate Managers(SSL/TLS 인증서) 도메인 호스팅 기능
- RDS
- SES
- NFS: 영구 스토리지를 위한
- Elasticache/opensearch (hae 실제 사례 참고)
- CloudWatch

3. 프로젝트 구성

3.1. Architecture



Microservices



3.2. YAML

- Notification Microservice

```
Notification-MS-Deployment.yaml
1 # NotificationMicroservice-Deployment.yaml
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: notification-microservice
6   labels:
7     app: notification-restapp
8 spec:
9   replicas: 1
10   selector:
11     matchLabels:
12       app: notification-restapp
13   template:
14     metadata:
15       labels:
16         app: notification-restapp
17     spec:
18       containers:
19         - name: notification-service
20           image: stacktakimy/kube-notifications-microservice:1.0.0
21           ports:
22             - containerPort: 8096
23           imagePullPolicy: Always
24   env:
25     - name: AWS_MAIL_SERVER_HOST
26       value: "smtp.sendgrid.net"
27     - name: AWS_MAIL_SERVER_USERNAME
28       value: "AKIAIYKNCW0B20XPZGGB"
29     - name: AWS_MAIL_SERVER_PASSWORD
30       value: "B#0Ww-xK!20Ba95zL+R#b1lDE)x51W#fRgGaAdeCv"
31     - name: AWS_MAIL_SERVER_FROM_ADDRESS
32       value: "v1999vvv@gmail.com" Email will be initiated from here
```

```
1 Notification-NS-ClusterIP-Service.yaml
2
3 apiVersion: v1
4 kind: Service
5 metadata:
6   name: notification-clusterip-service
7   labels:
8     app: notification-restapp
9   spec:
10    type: ClusterIP
11    selector:
12      app: notification-restapp
13    ports:
14      - port: 8086
15        targetPort: 8086
```

```
[Notification-MS-ClusterIP-Service.yaml
1 #Notification-MS-ClusterIP-Service.yaml
2
3 apiVersion: v1
4 kind: Service
5 metadata:
6   name: notification-clusterip-service
7   labels:
8     app: notification-restapp
9   spec:
10    type: ClusterIP
11    selector:
12      app: notification-restapp
13    ports:
14      - port: 8096
15        targetPort: 8095
```

4. 프로젝트 구현

4.1. Create EKS Cluster(Control Plane)

```
# Create Cluster
eksctl create cluster --name=ksn-eks-cluster \
                      --region=ap-northeast-1 \
                      --zones=ap-northeast-1a,ap-northeast-1b \
                      --without-nodegroups
```

eksctl get cluster

```
[ksn@KIMui-MacBookPro ~ % eksctl get cluster
  NAME      REGION      EKSCTL CREATED
  ksn-eks-cluster  ap-northeast-2  True
```

4.1.1. Create & Associate IAM OIDC Provider for our EKS Cluster

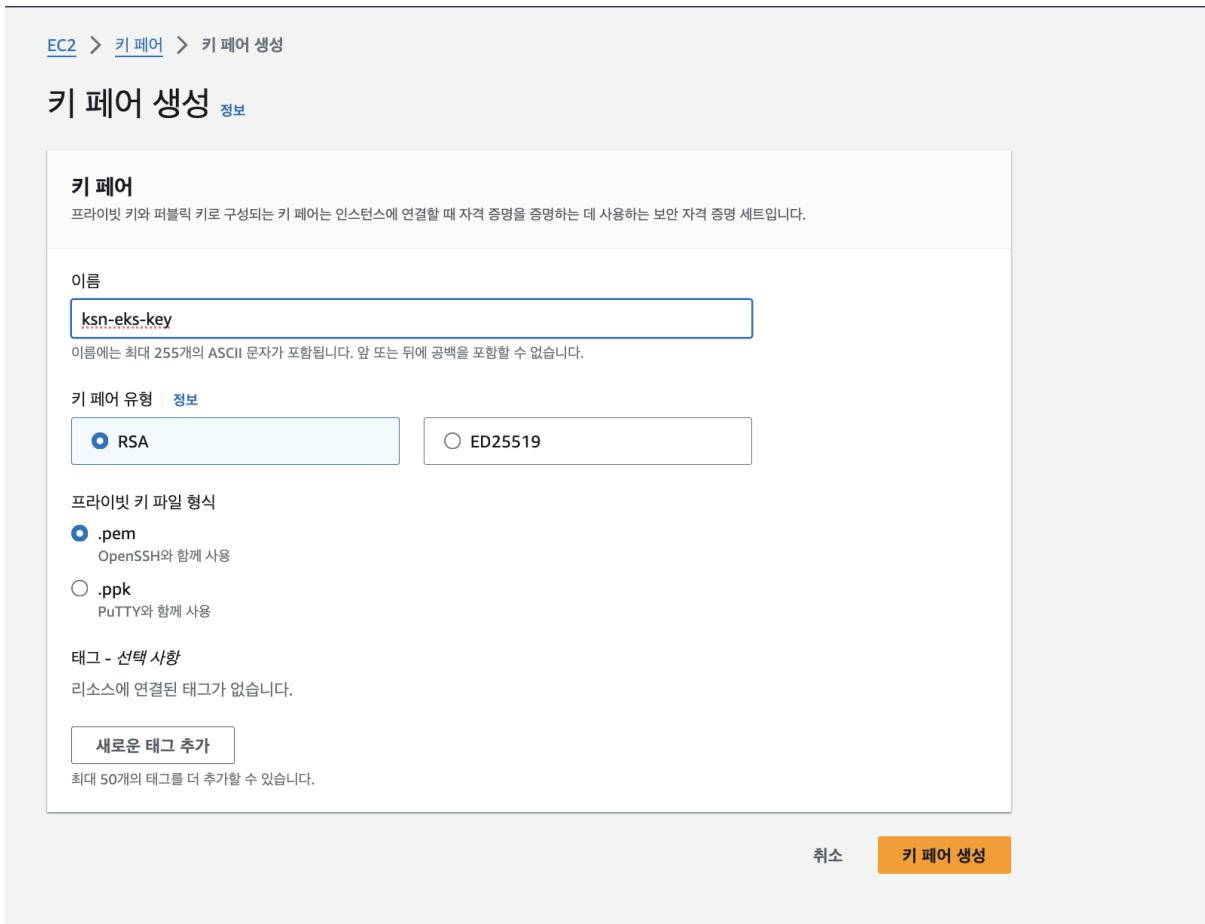
- To enable and use AWS IAM roles for Kubernetes service accounts on our EKS cluster, we must create & associate OIDC identity provider.
- To do so using `eksctl` we can use the below command.
- Use latest eksctl version (as on today the latest version is `0.21.0`)

```
eksctl utils associate-iam-oidc-provider \
    --region ap-northeast-2 \
    --cluster ksn-eks-cluster \
    --approve
```

```
ksn@KIMui-MacBookPro ~ % eksctl utils associate-iam-oidc-provider \
    --region ap-northeast-2 \
    --cluster ksn-eks-cluster \
    --approve
[ 2024-06-20 14:45:04 [i]  will create IAM Open ID Connect provider for cluster "ksn-eks-cluster" in "ap-northeast-2"
2024-06-20 14:45:04 [✓]  created IAM Open ID Connect provider for cluster "ksn-eks-cluster" in "ap-northeast-2"
```

4.1.2. Create EC2 Key Pair

- EKS Node Group을 생성할 때 사용할 Key Pair를 생성해준다. (EC2 > 키 페어 > 키 페어 생성)



4.2. Create EKS Node Group in Private Subnets

- We are going to deploy workloads on the private node group wherein workloads will be running private subnets and load balancer gets created in public subnet and accessible via internet.

```
eksctl create nodegroup --cluster=ksn-eks-cluster \
--region=ap-northeast-2 \
--name=eks-ng-private1 \
--node-type=t3.medium \
--nodes-min=2 \
--nodes-max=4 \
--node-volume-size=20 \
--ssh-access \
--ssh-public-key=ksn-eks-key \ #생성한
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
```

```
--appmesh-access \
--alb-ingress-access \
--node-private-networking
```

```
ksn@KIMui-MacBookPro ~ % eksctl create nodegroup --cluster=ksn-eks-cluster \
--region=ap-northeast-2 \
--name=eks-ng-private1 \
--node-type=t3.medium \
--nodes-min=2 \
--nodes-max=4 \
--node-volume-size=20 \
--ssh-access \
--ssh-public-key=ksn-eks-key \
--managed \
--asg-access \
--external-dns-access \
--full-ecr-access \
--appmesh-access \
--alb-ingress-access \
--node-private-networking
2024-06-20 16:29:44 [i] will use version 1.29 for new nodegroup(s) based on control plane version
2024-06-20 16:29:45 [i] nodegroup "eks-ng-private1" will use "" [AmazonLinux2/1.29]
2024-06-20 16:29:45 [i] using EC2 key pair "ksn-eks-key"
2024-06-20 16:29:45 [i] 1 nodegroup (eks-ng-private1) was included (based on the include/exclude rules)
2024-06-20 16:29:45 [i] will create a CloudFormation stack for each of 1 managed nodegroups in cluster "ksn-eks-cluster"
2024-06-20 16:29:45 [i]
2 sequential tasks: { fix cluster compatibility, 1 task: { 1 task: { create managed nodegroup "eks-ng-private1" } } }
2024-06-20 16:29:45 [i] checking cluster stack for missing resources
2024-06-20 16:29:46 [i] cluster stack has all required resources
2024-06-20 16:29:46 [i] building managed nodegroup stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:29:46 [i] deploying stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:29:46 [i] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:30:16 [i] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:30:53 [i] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:32:41 [i] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-nodegroup-eks-ng-private1"
2024-06-20 16:32:41 [i] no tasks
2024-06-20 16:32:41 [✓] created 0 nodegroup(s) in cluster "ksn-eks-cluster"
2024-06-20 16:32:41 [i] nodegroup "eks-ng-private1" has 2 node(s)
2024-06-20 16:32:41 [i] node "ip-192-168-111-27.ap-northeast-2.compute.internal" is ready
2024-06-20 16:32:41 [i] node "ip-192-168-69-164.ap-northeast-2.compute.internal" is ready
2024-06-20 16:32:41 [i] waiting for at least 2 node(s) to become ready in "eks-ng-private1"
2024-06-20 16:32:41 [i] nodegroup "eks-ng-private1" has 2 node(s)
2024-06-20 16:32:41 [i] node "ip-192-168-111-27.ap-northeast-2.compute.internal" is ready
2024-06-20 16:32:41 [i] node "ip-192-168-69-164.ap-northeast-2.compute.internal" is ready
2024-06-20 16:32:41 [✓] created 1 managed nodegroup(s) in cluster "ksn-eks-cluster"
2024-06-20 16:32:42 [i] checking security group configuration for all nodegroups
2024-06-20 16:32:42 [i] all nodegroups have up-to-date cloudformation templates
```

- Node Group 생성 확인

```
ken@KIMui-MacBookPro ~ % kubectl get nodes -o wide
NAME           STATUS    ROLES   AGE     VERSION      INTERNAL-IP      EXTERNAL-IP      OS-IMAGE       KERNEL-VERSION      CONTAINER-RUNTIME
ip-192-168-111-27.ap-northeast-2.compute.internal   Ready    <none>   5m48s   v1.29.3-eks-ae9a62a   192.168.111.27   <none>        Amazon Linux 2   8.10.218-208.862.amzn2.x86_64   containerd://1.7.11
ip-192-168-69-164.ap-northeast-2.compute.internal   Ready    <none>   5m45s   v1.29.3-eks-ae9a62a   192.168.69.164  <none>        Amazon Linux 2   8.10.218-208.862.amzn2.x86_64   containerd://1.7.11
```

4.5. Create RDS Database

4.5.1. Create DB Security Group

- Create security group to allow access for RDS Database on port 3306
- Security group name: eks_rds_db_sg

- Description: Allow access for RDS Database on Port 3306
- VPC: eksctl-eksdemo1-cluster/VPC
- **Inbound Rules**
 - Type: MySQL/Aurora
 - Protocol: TCP
 - Port: 3306
 - Source: Anywhere (0.0.0.0/0)
 - Description: Allow access for RDS Database on Port 3306
- **Outbound Rules**
 - Leave to defaults

The screenshot shows the AWS EC2 Security Groups creation interface. In the 'Inbound Rules' section, there is one rule defined:

유형	프로토콜	포트 범위	소스	설명
MySQL/Aurora	TCP	3306	Anywhere (0.0.0.0/0)	Allow access for RDS port on 3306

4.5.2. Create DB Subnet Group

DB 서브넷 그룹 생성

새 서브넷 그룹을 생성하려면 이름과 설명을 입력하고 기존 VPC를 선택합니다. 그러면 해당 VPC와 관련된 서브넷을 추가할 수 있습니다.

서브넷 그룹 세부 정보

이름

서브넷 그룹이 생성된 후에는 이름을 수정할 수 없습니다.

eks-rds-db-subnetgroup

1~255자로 구성되어야 합니다. 영숫자, 공백, 하이픈, 밑줄 및 마침표를 사용할 수 있습니다.

설명

Subnet group for RDS db

VPC

DB 서브넷 그룹에 사용할 서브넷에 해당하는 VPC 식별자를 선택합니다. 서브넷 그룹이 생성된 후에는 다른 VPC 식별자를 선택할 수 없습니다.

eksctl-ksn-eks-cluster-cluster/VPC (vpc-03d8108a943490477)



서브넷 추가

가용 영역

추가할 서브넷이 포함된 가용 영역을 선택합니다.

가용 영역 선택



ap-northeast-2a X

ap-northeast-2c X

서브넷

추가할 서브넷을 선택합니다. 목록에는 선택한 가용 영역의 서브넷이 포함됩니다.

서브넷 선택



subnet-017f6dc90273934c5 (192.168.64.0/19) X

subnet-045fddbc7d2c354e6 (192.168.96.0/19) X

4.5.3. Create RDS

- 데이터베이스 생성 방식 : 표준 생성
- 엔진 옵션 : MySQL
- 에디션: MySQL Community
- 엔진 버전 : 8.0.35
- 템플릿 : 프리 티어

설정

DB 인스턴스 사용자 정보

DB 인스턴스 이름을 입력하세요. 이름은 현재 AWS 환경에서 AWS 계정이 소유하는 모든 DB 인스턴스에 대해 고유해야 합니다.

mpgdb

DB 인스턴스 사용자는 대소문자를 구분하지 않지만 `mpgdb`라는 값이 모두 소문자로 저장됩니다. 패스워드는 8자리 이상으로 구성되어야 합니다. 첫 번째 문자는 대문자이며 끝이면 2자가 연속될 수 있습니다. 패스워드로 광고를 수 있습니다.

▼ 자격 증명 설정

마스터 사용자 이름 **생성**

DB 인스턴스에서 사용자 로그인 ID를 입력하세요.

dbadmin

1~16자리의 문자, 숫자 및 번역 문자는 글자이며 됩니다.

자격 증명 관리

AWS Secrets Manager 사용하거나 마스터 사용자 자격 증명을 관리할 수 있습니다.

AWS Secrets Manager에서 관리 - **생성** 키워드 선택

RDS는 자동으로 암호를 생성하고 AWS Secrets Manager를 사용하여 대체 자격 증명을 관리합니다.

암호 생성

Amazon RDS에서 사용으로 암호를 생성하거나 사용자가 직접 암호를 지정할 수 있습니다.

마스터 암호 **생성**

최소 계약 조건: 8자 이상의 인데 가능한 ASCII 문자 사용합니다. / " @ 기호는 포함될 수 있습니다.

마스터 암호 확인 **생성**

인스턴스 구성

마스터 DB 인스턴스 구성을 위한 선택한 항목에서 지원하는 값으로 설정합니다.

DB 인스턴스 품질스 **생성**

▼ 필수 속성

Amazon RDS 최적화된 서버를 지원하는 인스턴스 클래스 품질스 풋지 **생성**

Amazon RDS 최적화된 서버는 주어진 시기 차량(throughput)을 최대 2배 높입니다.

이런 세트 품질스 조정

(*) 스탠다드 품질스(= 기본 품질)

(*) 에디티드 품질스(= 높은 품질)

인스턴스별 품질스 품질스 조정

db.t3.2xlarge
2xGPUs | 16 GB RAM | 대체값 2,200Mbps

스토리지

스토리지 유형: **Amazon S3 (Standard)** 스토리지 활용을 사용할 수 있습니다.

방법 **S3(Standard)**
클라우드 대시 보드 설정

클라우드 스토리지 **생성**

XO
GB

클라우드 대시 보드 설정은 144GB입니다.

▶ 스토리지 사용 조정

0에 인스턴스의 스토리지를 수정하면 DB 인스턴스의 상태가 스토리지 확장과 상태가 됩니다. 스토리지 확장 작업이 완료되어도 인스턴스는 계속 사용할 수 있습니다. [자세히 알아보기](#)

연결 정보

컴퓨팅 리소스
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정할지를 선택합니다. 연결을 설정하면 컴퓨팅 리소스가 이 데이터베이스에 연결할 수 있도록 연결 설정이 자동으로 변경됩니다.

EC2 컴퓨팅 리소스에 연결 안 함
이 데이터베이스의 컴퓨팅 리소스에 대한 연결을 설정하지 않습니다.
다. 나중에 컴퓨팅 리소스에 대한 연결을 수동으로 설정할 수 있습니다.

EC2 컴퓨팅 리소스에 연결
이 데이터베이스의 EC2 컴퓨팅 리소스에 대한 연결을 설정합니다.

Virtual Private Cloud(VPC) 정보
VPC를 선택합니다. VPC는 이 DB 인스턴스의 가상 네트워킹 환경을 정의합니다.

eksctl-ksn-eks-cluster-cluster/vPC (vpc-03d8108a943490477)
4 서브넷, 2 가용 영역

해당 DB 서브넷 그룹이 있는 VPC만 나열됩니다.

DB 서브넷 그룹 정보
DB 서브넷 그룹을 선택합니다. DB 서브넷 그룹은 선택한 VPC에서 DB 인스턴스가 어떤 서브넷과 IP 범위를 사용할 수 있는지를 정의합니다.

eks-rds-db-subnetgroup
2 서브넷, 2 가용 영역

퍼블릭 액세스 정보

예
RDS는 데이터베이스에 퍼블릭 IP 주소를 할당합니다. VPC 외부의 Amazon EC2 인스턴스 및 다른 리소스가 데이터베이스에 연결할 수 있습니다. VPC 내부의 리소스도 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

아니요
RDS는 퍼블릭 IP 주소를 데이터베이스에 할당하지 않습니다. VPC 내부의 Amazon EC2 인스턴스 및 다른 리소스만 데이터베이스에 연결할 수 있습니다. 데이터베이스에 연결할 수 있는 리소스를 지정하는 VPC 보안 그룹을 하나 이상 선택합니다.

VPC 보안 그룹(방화벽) 정보
데이터베이스에 대한 액세스를 허용할 VPC 보안 그룹을 하나 이상 선택합니다. 보안 그룹 규칙이 적절한 수신 트래픽을 허용하는지 확인합니다.

기존 항목 선택
기존 VPC 보안 그룹 선택

새로 생성
새 VPC 보안 그룹 생성

기존 VPC 보안 그룹

하나 이상의 옵션 선택
eks-rds-sg X

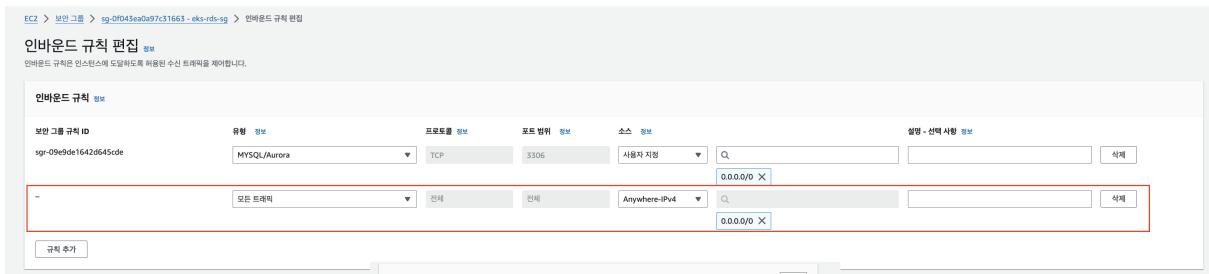
가용 영역 정보

ap-northeast-2a

Edit Database Security to Allow Access from 0.0.0.0/0

- Go to **EC2** → **Security Groups** → **eks-rds-db-securitygroup**
- **Edit Inbound Rules**

- **Source: Anywhere (0.0.0.0/0) (Allow access from everywhere for now)**



4.5.4. Create Kubernetes externalName Service and Deploy

```
apiVersion: v1
kind: Service
metadata:
  name: mysql
spec:
  type: ExternalName
  externalName: mgmtdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com
```

Deploy Manifest

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f ksn-MySQL-externalName-Service.yaml
service/mysql created
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get svc
NAME      TYPE        CLUSTER-IP   EXTERNAL-IP
kubernetes  ClusterIP  10.100.0.1  <none>
mysql     ExternalName <none>       mgmtdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com
```

PORT(S)	AGE
443/TCP	54m
<none>	12s

4.5.5. DB 접속 및 작동 확인

```
kubectl run -it --rm --image=mysql:latest --restart=Never \
mysql-client -- mysql -h mgmtdb.chi680k06fii.ap-northeast-2.
rds.amazonaws.com -u dbadmin -pdbpassword
```

```

ksn@KIMui-MacBookPro ksn-k8s-project % kubectl run -it --rm --image=mysql:latest --restart=Never mysql-client -- mysql -h mgmtdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com -u dbadmin -pdbpassword
If you don't see a command prompt, try pressing enter.

mysql> show schemas;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> create database mgmtdb;
Query OK, 1 row affected (0.01 sec)

mysql> show schemas;
+-----+
| Database |
+-----+
| information_schema |
| mgmtdb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

```

4.4. AWS Load Balancer Controller

4.4.1. Verify any IAM Service Accounts present in EKS Cluster

```
eksctl get iamserviceaccount --cluster=eksdemo1
```

```

ksn@KIMui-MacBookPro ksn-k8s-project % eksctl get iamserviceaccount --cluster=ksn-eks-cluster
No iamserviceaccounts found

```

4.4.2. Step-02: Create IAM Policy

- Create IAM policy for the AWS Load Balancer Controller that allows it to make calls to AWS APIs on your behalf.
- As on today [2.3.1](#) is the latest Load Balancer Controller
- We will download always latest from main branch of Git Repo

```
curl -o iam_policy_latest.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/main/docs/install/iam_policy.json
```

```

ksn@KIMui-MacBookPro ksn-k8s-project % curl -o iam_policy_latest.json https://raw.githubusercontent.com/kubernetes-sigs/aws-load-balancer-controller/main/docs/install/iam_policy.json
  % Total    % Received % Xferd  Average Speed   Time   Time     Time  Current
                                 Dload  Upload Total   Spent   Left  Speed
100  8446  100  8446    0     0 19844      0  --:--:-- 0:00:01  --:--:-- 19872
ksn@KIMui-MacBookPro ksn-k8s-project % ls -ltr
total 72
-rw-r--r--  1 ksn  staff  2262  6 17 17:09 ksn-ManagementMS-Deployment.yaml
-rw-r--r--  1 ksn  staff   492  6 17 17:10 ksn-Management-NodePort-Service.yaml
-rw-r--r--  1 ksn  staff   928  6 17 17:10 ksn-NotificationsMS-Deployment.yaml
-rw-r--r--  1 ksn  staff   196  6 17 17:10 ksn-NotificationsMS-Redirect-ExternalName-Service.yaml
-rw-r--r--  1 ksn  staff    0  6 17 17:12 ksn-ExternalDNS-Service-IP-Service.yaml
-rw-r--r--  1 ksn  staff  2078  6 17 17:12 ksn-ALB-Ingress-SSL-Redirect-ExternalDNS.yaml
-rw-r--r--  1 ksn  staff   168  6 19 01:39 ksn-MySQL-externalName-Service.yaml
-rw-r--r--  1 ksn  staff  8446  6 19 09:28 iam_policy_latest.json

```

```

1 {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Allow",
6             "Action": [
7                 "iam:CreateServiceLinkedRole"
8             ],
9             "Resource": "*",
10            "Condition": {
11                "StringEquals": {
12                    "iam:AWSServiceName": "elasticloadbalancing.amazonaws.com"
13                }
14            }
15        },
16        {
17            "Effect": "Allow",
18            "Action": [
19                "ec2:DescribeAccountAttributes",
20                "ec2:DescribeAddresses",
21                "ec2:DescribeAvailabilityZones",
22                "ec2:DescribeInternetGateways",
23                "ec2:DescribeVpcs",
24                "ec2:DescribeVpPeeringConnections",
25                "ec2:DescribeSubnets",
26                "ec2:DescribeSecurityGroups",
27                "ec2:DescribeInstances",
28                "ec2:DescribeNetworkInterfaces",
29                "ec2:DescribeTags",
30                "ec2:GetCoiPoolsUsage",
31                "ec2:DescribeCoiPools",
32                "elasticloadbalancing:DescribeLoadBalancers",
33                "elasticloadbalancing:DescribeLoadBalancerAttributes",
34                "elasticloadbalancing:DescribeListeners",
35                "elasticloadbalancing:DescribeListenerCertificates",
36                "elasticloadbalancing:DescribeSSLPolicies",
37                "elasticloadbalancing:DescribeRules",
38                "elasticloadbalancing:DescribeTargetGroups",
39                "elasticloadbalancing:DescribeTargetGroupAttributes",
40                "elasticloadbalancing:DescribeTargetHealth",
41                "elasticloadbalancing:DescribeTags",
42                "elasticloadbalancing:DescribeTrustStores"
43            ],
44            "Resource": "*"
45        }
46    }

```

```

# Create IAM Policy using policy downloaded
aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file://iam_policy_latest.json

```

```

ksn@KIMui-MacBookPro ksn-k8s-project %
aws iam create-policy \
--policy-name AWSLoadBalancerControllerIAMPolicy \
--policy-document file://iam_policy_latest.json
{
    "Policy": {
        "PolicyName": "AWSLoadBalancerControllerIAMPolicy",
        "PolicyId": "ANPAXYKJW3R2VDLEJWY7Z",
        "Arn": "arn:aws:iam::533267405941:policy/AWSLoadBalancerControllerIAMPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2024-06-19T00:39:42+00:00",
        "UpdateDate": "2024-06-19T00:39:42+00:00"
    }
}

```

- **Make a note of Policy ARN as we are going to use that in next step when creating IAM Role.**

arn:aws:iam::533267405941:policy/AWSLoadBalancerControllerIAMPolicy

AWSLoadBalancerControllerIAMPolicy 정보

정책 세부 정보

유형 고객 관리형	생성 시간 June 19, 2024, 09:39 (UTC+09:00)	편집 시간 June 19, 2024, 09:39 (UTC+09:00)	ARN <code>arn:aws:iam::533267405941:policy/AWSLoadBalancerControllerIAMPolicy</code>
--------------	---	---	---

이 정책에 정의된 권한 정보

이 정책 문서에 정의된 권한은 허용되거나 거부되는 작업을 지정합니다. IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 대한 권한을 정의하려면 여기에 정책을 연결합니다.

허용(서비스 418개 중 9개)

서비스	액세스 수준	리소스	요청 조건
Certificate Manager	전체: 나열 제한적: 읽기	모든 리소스	None
Cognito User Pools	제한적: 읽기	모든 리소스	None
EC2	전체: 태그 지정 제한적: 나열, 읽기, 쓰기	Multiple	Multiple
ELB	전체: 나열, 태그 지정 제한적: 읽기, 쓰기	Multiple	Multiple
ELB v2	전체: 태그 지정 제한적: 읽기, 쓰기	Multiple	Multiple
IAM	제한적: 나열, 읽기, 쓰기	모든 리소스	<code>iam:AWSServiceName = elasticloadbalancing.amazonaws.com</code>
Shield	제한적: 읽기, 쓰기	모든 리소스	None
WAF Regional	제한적: 읽기, 쓰기	모든 리소스	None
WAF V2	제한적: 읽기, 쓰기	모든 리소스	None

4.4.3. Step-03: Create an IAM role for the AWS LoadBalancer Controller and attach the role to the Kubernetes service account

- Applicable only with `eksctl` managed clusters
- This command will create an AWS IAM role
- This command also will create Kubernetes Service Account in k8s cluster
- In addition, this command will bind IAM Role created and the Kubernetes service account created

Step-03-01: Create IAM Role using eksctl

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get sa -n kube-system
NAME                      SECRETS   AGE
attachdetach-controller    0         11h
aws-cloud-provider        0         11h
aws-node                  0         11h
certificate-controller    0         11h
clusterrole-aggregation-controller 0         11h
coredns                   0         11h
cronjob-controller        0         11h
daemon-set-controller     0         11h
default                   0         11h
deployment-controller     0         11h
disruption-controller     0         11h
eks-vpc-resource-controller 0         11h
endpoint-controller       0         11h
endpointslice-controller  0         11h
endpointslicemirroring-controller 0         11h
ephemeral-volume-controller 0         11h
expand-controller          0         11h
generic-garbage-collector 0         11h
horizontal-pod-autoscaler 0         11h
job-controller             0         11h
kube-proxy                 0         11h
legacy-service-account-token-cleaner 0         11h
namespace-controller      0         11h
node-controller            0         11h
persistent-volume-binder  0         11h
pod-garbage-collector     0         11h
pv-protection-controller  0         11h
pvc-protection-controller 0         11h
replicaset-controller     0         11h
replication-controller    0         11h
resourcequota-controller  0         11h
root-ca-cert-publisher   0         11h
service-account-controller 0         11h
service-controller         0         11h
statefulset-controller    0         11h
tagging-controller         0         11h
ttl-after-finished-controller 0         11h
ttl-controller             0         11h
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get sa aws-load-balancer-controller -n kube-system
Error from server (NotFound): serviceaccounts "aws-load-balancer-controller" not found
```

```
eksctl create iamserviceaccount \
--cluster=ksn-eks-cluster \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--attach-policy-arn=arn:aws:iam::533267405941:policy/AWSLoadBalancerControllerIAMPolicy \
--override-existing-serviceaccounts \
--approve
```

#Note: K8S Service Account Name that need to be bound to newly created IAM Role

```
[ksn@KIMui-MacBookPro ksn-k8s-project % eksctl create iamserviceaccount \
--cluster=ksn-eks-cluster \
--namespace=kube-system \
--name=aws-load-balancer-controller \
--attach-policy-arn=arn:aws:iam::533267405941:policy/AWSLoadBalancerControllerIAMPolicy \
--override-existing-serviceaccounts \
--approve
2024-06-28 17:24:28 [1] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) was included (based on the include/exclude rules)
2024-06-28 17:24:28 [1] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2024-06-28 17:24:28 [1] task: (
  2 steps in total:
    1 create IAM role for serviceaccount "kube-system/aws-load-balancer-controller",
    2 create serviceaccount "kube-system/aws-load-balancer-controller",
) 2024-06-28 17:24:28 [1] building iamserviceaccount stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-06-28 17:24:28 [1] deploying stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-06-28 17:24:50 [1] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-kube-system-aws-load-balancer-controller"
2024-06-28 17:24:51 [1] created serviceaccount "kube-system/aws-load-balancer-controller"
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get sa aws-load-balancer-controller -n kube-system
NAME          SECRETS   AGE
aws-load-balancer-controller   0          24m
```

4.4.4. Step-03-02: Verify using eksctl cli

```
# Get IAM Service Account
eksctl get iamserviceaccount --cluster eksdemo1
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % eksctl get iamserviceaccount --cluster eksCluster
NAMESPACE      NAME           ROLE ARN
kube-system    aws-load-balancer-controller  arn:aws:iam::533267405941:role/eksctl-eksCluster-addon-iamserviceaccount-kub-Role1-mpPq1XiR4vqT
```

IAM Role 생성 확인

The screenshot shows the IAM Roles page with the following details:

- Role Name:** eksctl-ksn-eks-cluster-addon-iamserviceaccoun-Role1-Ea1M4gauGsim
- ARN:** arn:aws:iam::533267405941:role/eksctl-ksn-eks-cluster-addon-iamserviceaccoun-Role1-Ea1M4gauGsim
- Creation Date:** June 20, 2024, 17:24 (UTC+09:00)
- Last Activity:** -
- Max Session Duration:** 1시간

Policies (1) Details

One policy is listed: **AWSLoadBalancerControllerIAMPolicy**. The policy details are as follows:

- Policy Type:** 고객 관리형
- Condition:** oidc.eks.ap-northeast-2.amazonaws.com/id/00DAD97AAC554E1FBD59B2DA47DF4C7E:aud: "sts.amazonaws.com", oidc.eks.ap-northeast-2.amazonaws.com/id/00DAD97AAC554E1FBD59B2DA47DF4C7E:sub: "system:serviceaccount:kube-system:aws-load-balancer-controller"

The screenshot shows the IAM Policies page with the following details:

Policy Name: eksctl-ksn-eks-cluster-addon-iamserviceaccoun-Role1-Ea1M4gauGsim

Policy Type: 고객 관리형

Policy Document:

```

1 - {
2     "Version": "2012-10-17",
3     "Statement": [
4         {
5             "Effect": "Allow",
6             "Principal": [
7                 "Federated": "arn:aws:iam::533267405941:oidc-provider/oidc.eks.ap-northeast-2.amazonaws.com/id/00DAD97AAC554E1FBD59B2DA47DF4C7E"
8             ],
9             "Action": "sts:AssumeRoleWithWebIdentity",
10            "Condition": {
11                "StringEquals": [
12                    "oidc.eks.ap-northeast-2.amazonaws.com/id/00DAD97AAC554E1FBD59B2DA47DF4C7E:aud": "sts.amazonaws.com",
13                    "oidc.eks.ap-northeast-2.amazonaws.com/id/00DAD97AAC554E1FBD59B2DA47DF4C7E:sub": "system:serviceaccount:kube-system:aws-load
14                    -balancer-controller"
15                ]
16            }
17        }
18    ]
}
```

4.4.5. Step-03-04: Verify k8s Service Account using kubectl

```
# Describe Service Account aws-load-balancer-controller
kubectl describe sa aws-load-balancer-controller -n kube-system
```

```
ksn@KIMi-MacBookPro ksn-k8s-project % kubectl get sa aws-load-balancer-controller -n kube-system
NAME          SECRETS   AGE
aws-load-balancer-controller   0        3m5s
ksn@KIMi-MacBookPro ksn-k8s-project % kubectl describe sa aws-load-balancer-controller -n kube-system
Name:           aws-load-balancer-controller
Namespace:      kube-system
Labels:         app.kubernetes.io/manage-by=eksctl
Annotations:    iam.amazonaws.com/role-arn: arn:aws:iam::533267405941:role/eksctl-ksn-eks-cluster-addon-iamserviceaccount-Role1-Ea1M4gauGsm
Image pull secrets: <none>
Mountable secrets: <none>
Tokens:         <none>
Events:         <none>
```

4.4.6. Install the ALB Controller using Helm

```
ksn@KIMi-MacBookPro ksn-k8s-project % brew install helm
Auto-updating Homebrew...
Adjust how often this is run with HOMEBREW_AUTO_UPDATE. Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
==> Downloading https://ghcr.io/v2/homebrew/portable-ruby/portable-ruby/blobs/sha256:49847c7a13f7094b211f6d002590dd2371be07dac894a3d6941d7696296306
#####
# Extracting portable-ruby-3.3.3.arm64_big_sur.bottle.tar.gz
#####
Auto-updated Homebrew!
Updated 3 taps (weaveworks/tap, homebrew/core and homebrew/cask).
  New Formulae
age-plugin-se      chrc      dpg-tree     geniso      jscrontoolkit    llama.cpp      osicl      pedump      rustlu-ffi      typstyle      zfind
ansibrew            cloudflare-clia  displayplacer  gorilla-cli  kubeletIn      mactop      olls-for-unix  poollet      spann-lite      vedic
apache-flink-cdc  codecov-cil   fern-api      goroxxy0.2.8 libpassw1      nsync      openfa      poutine      stripe-cli      vextl
batt                cycil      geni         iamb       libvirt-python  nvtcp      openjdk@21    qshell      toipe       yara-x
  New Casks
sclom                font-gq-magali      font-playwright-dk-loopet      font-playwright-pj      masymbolicator
andromeda-wallet   font-palemonasmufi-bold      font-playwright-dk-uloopt      font-playwright-pj      material-maker
bias-fx              font-palemonasmufi-italic      font-playwright-es      font-playwright-pt      nans?
blitz-9g             font-palemonasmufi-italic      font-playwright-es-deco      font-playwright-ro      nessus
canon-utrii-driver  font-palemonasmufi-regular      font-playwright-fr-moderne  font-playwright-sk      phocus
chime                font-playwright-ar      font-playwright-hr      font-playwright-tz      proton-pass
dlib@sep             font-playwright-aut      font-playwright-hu      font-playwright-tz      ubilibrefo
core-tunnel          font-playwright-aut-nsw      font-playwright-hu-ljivea  font-playwright-us-modern
elgato-capture-device-utility  font-playwright-aqid      font-playwright-id      font-playwright-us-trad  quick-app-ide
emclient@beta        font-playwright-au-sa      font-playwright-ie      font-playwright-zs      quicktune
font-alumni-sans-collegiate-one-sc  font-playwright-au-tas      font-playwright-in      font-wittgenstein      screaming-frog-log-file-analyser
font-archer          font-playwright-be-v1      font-playwright-in      font-zain      semeru-jdk-open@21
font-arsenal-sc      font-playwright-bev10      font-playwright-it      font-zain      ugg
font-baskerville-sc  font-playwright-be-wal      font-playwright-it-trad  font-zine      voice
font-beirut          font-playwright-ca      font-playwright-mx      hopper-disassembler  warcraft-logs-uploader
font-bodoni-modabsc  font-playwright-cl      font-playwright-ng-modern  impel      xnapper
font-bona-nova-sc   font-playwright-co      font-playwright-nl      little-snitch@6
font-edu-ku-vc-wa-nt-hand  font-playwright-cr      font-playwright-nz      loop      mac-mouse-fix@2
font-futura          font-playwright-de-la      font-playwright-pj      mac-mouse-fix@2

You have 7 outdated formulae installed.
  Downloading https://ghcr.io/v2/homebrew/core/helm@manifests/2.15.2
#####
# Fetching helm
#####
# Downloading https://ghcr.io/v2/homebrew/core/helm/blobs/sha256:acf31d718a351224eaddc70677c88b7da41fdeca6568b21b7393fd76247c2
#####
# helm@arm64_sonoma.bottle.tar.gz
#####
# helm@arm64_sonoma.bottle.tar.gz
  Caveats
zsh completions have been installed to:
  /opt/homebrew/share/zsh/site-functions
  Summary
  ▶  /opt/homebrew/share/Gallery/Helm/2.15.2: 66 files, 50.2MB
  Running 'brew cleanup helm'.
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
  brew cleanup has not been run in the last 30 days, running now...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
Removing: /Users/ksn/Library/Caches/Homebrew/aws-iam-authenticator---0.6.14... (13.9MB)
Removing: /Users/ksn/Library/Caches/Homebrew/eksctl---0.177.0.tar.gz... (34.7MB)
Removing: /Users/ksn/Library/Caches/Homebrew/kubernetes---cli-1.38.1... (16.2MB)
Removing: /Users/ksn/Library/Caches/Homebrew/openidp---2.6.7... (2.7MB)
Removing: /Users/ksn/Library/Logs/Homebrew/eksctl... (117B)
```

```
ksn@KIMi-MacBookPro ksn-k8s-project % helm version
version.BuildInfo{Version:"v3.15.2", GitCommit:"1a500d5625419a524fdae4b33de351cc4f58ec35", GitTreeState:"clean", GoVersion:"go1.22.4"}
```

```
# Add the eks-charts repository.
```

```
helm repo add eks https://aws.github.io/eks-charts
```

```
# Update your local repo to make sure that you have the mos
```

```
t recent charts.
```

```
helm repo update
```

```
# Install the AWS Load Balancer Controller.  
## Template  
helm install aws-load-balancer-controller eks/aws-load-balancer-controller \  
-n kube-system \  
--set clusterName=ksn-eks-cluster \  
--set serviceAccount.create=false \  
--set serviceAccount.name=aws-load-balancer-controller \  
--set region=ap-northeast-2 \  
--set vpcId=vpc-03d8108a943490477 \  
--set image.repository=602401143452.dkr.ecr.ap-northeast-  
2.amazonaws.com/amazon/aws-load-balancer-controller
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % helm install aws-load-balancer-controller eks/aws-load-balancer-controller \  
-n kube-system \  
--set clusterName=ksn-eks-cluster \  
--set serviceAccount.create=false \  
--set serviceAccount.name=aws-load-balancer-controller \  
--set region=ap-northeast-2 \  
--set vpcId=vpc-03d8108a943490477 \  
--set image.repository=602401143452.dkr.ecr.ap-northeast-2.amazonaws.com/amazon/aws-load-balancer-controller  
NAME: aws-load-balancer-controller  
LAST DEPLOYED: Fri Jun 21 01:15:37 2024  
NAMESPACE: kube-system  
STATUS: deployed  
REVISION: 1  
TEST SUITE: None  
NOTES:  
AWS Load Balancer controller installed!
```

```

|ksn@KIMui-MacBookPro ksn-k8s-project % kubectl -n kube-system describe deployment aws-load-balancer-controller
Name:           aws-load-balancer-controller
Namespace:      kube-system
CreationTimestamp: Fri, 21 Jun 2024 01:15:38 +0900
Labels:          app.kubernetes.io/instance=aws-load-balancer-controller
Annotations:    app.kubernetes.io/managed-by=Helm
                app.kubernetes.io/name=aws-load-balancer-controller
                app.kubernetes.io/version=v2.8.1
                helm.sh/chart=aws-load-balancer-controller-1.8.1
                deployment.kubernetes.io/revision: 1
                meta.helm.sh/release-name: aws-load-balancer-controller
                meta.helm.sh/release-namespace: kube-system
Selector:        app.kubernetes.io/instance=aws-load-balancer-controller,app.kubernetes.io/name=aws-load-balancer-controller
Replicas:        2 desired | 2 updated | 2 total | 2 available | 0 unavailable
StrategyType:   RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:      app.kubernetes.io/instance=aws-load-balancer-controller
                app.kubernetes.io/name=aws-load-balancer-controller
  Annotations: prometheus.io/port: 8080
                prometheus.io/scrape: true
  Service Account: aws-load-balancer-controller
  Containers:
    aws-load-balancer-controller:
      Image:       602401143452.dkr.ecr.ap-northeast-2.amazonaws.com/amazon/aws-load-balancer-controller:v2.8.1
      Ports:       9443/TCP, 8080/TCP
      Host Ports: 0/TCP, 0/TCP
      Args:
        --cluster-name=ksn-eks-cluster
        --ingress-class=alb
        --aws-region=ap-northeast-2
        --aws-vpc-id=vpc-03d8108a943490477
      Liveliness:  http-get http://:61779/healthz delay=30s timeout=10s period=10s #success=1 #failure=2
      Readiness:   http-get http://:61779/readyz delay=10s timeout=10s period=10s #success=1 #failure=2
      Environment: <none>
      Mounts:
        /tmp/k8s-webhook-server/serving-certs from cert (ro)
  Volumes:
    cert:
      Type:        Secret (a volume populated by a Secret)
      SecretName: aws-load-balancer-tls
      Optional:   false
  Priority Class Name: system-cluster-critical
  Node-Selectors:      <none>
  Tolerations:        <none>
Conditions:
  Type     Status  Reason
  ----  -----
  Available  True    MinimumReplicasAvailable
  Progressing True   NewReplicaSetAvailable
OldReplicaSets: <none>
NewReplicaSet:  aws-load-balancer-controller-b6b97f465 (2/2 replicas created)
Events:
  Type     Reason     Age   From            Message
  ----  -----  ----  ----
  Normal  ScalingReplicaSet  2m2s  deployment-controller  Scaled up replica set aws-load-balancer-controller-b6b97f465 to 2

```

Step-05: Ingress Class Concept

- Understand what is Ingress Class
- Understand how it overrides the default deprecated annotation `#kubernetes.io/ingress.class: "alb"`
- [Ingress Class Documentation Reference](#)
- [Different Ingress Controllers available today](#)

Step-06: Review IngressClass Kubernetes Manifest

- **File Location:** `08-01-Load-Balancer-Controller-Install/kube-manifests/01-ingressclass-resource.yaml`
- Understand in detail about annotation `ingressclass.kubernetes.io/is-default-class: "true"`

```

|ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f ingressclass-resource.yaml
ingressclass.networking.k8s.io/my-aws-ingress-class created

```

4.5. External DNS

Step-01: Introduction

- **External DNS:** Used for Updating Route53 RecordSets from Kubernetes
- We need to create IAM Policy, k8s Service Account & IAM Role and associate them together for external-dns pod to add or remove entries in AWS Route53 Hosted Zones.
- Update External-DNS default manifest to support our needs
- Deploy & Verify logs

4.5.1. Step-02: Create IAM Policy

- This IAM policy will allow external-dns pod to add, remove DNS entries (Record Sets in a Hosted Zone) in AWS Route53 service
- Go to Services → IAM → Policies → Create Policy
 - Click on **JSON** Tab and copy paste below JSON
 - Click on **Visual editor** tab to validate
 - Click on **Review Policy**
 - **Name:** AllowExternalDNSUpdates
 - **Description:** Allow access to Route53 Resources for ExternalDNS
 - Click on **Create Policy**

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "route53:ChangeResourceRecordSets"  
            ],  
            "Resource": [  
                "arn:aws:route53:::hostedzone/*"  
            ]  
        },  
    ]  
},
```

```
{
    "Effect": "Allow",
    "Action": [
        "route53>ListHostedZones",
        "route53>ListResourceRecordSets"
    ],
    "Resource": [
        "*"
    ]
}
```

IAM > 정책 > 정책 생성

1단계
권한 지정

2단계
검토 및 생성

권한 지정

서비스, 작업, 리소스 및 조건을 선택하여 권한을 추가합니다. JSON 편집기를 사용하여 권한 설명문을 작성합니다.

정책 편집기

시작적 JSON 작업 □

문 편집

문 선택

정책에서 기존 문을 선택하거나 새 문을 추가합니다.

+ 새 문 추가

```

1▼ [
2  "Version": "2012-10-17",
3  "Statement": [
4    {
5      "Effect": "Allow",
6      "Action": [
7        "route53:ChangeResourceRecordSets"
8      ],
9      "Resource": [
10        "arn:aws:route53:::hostedzone/*"
11      ],
12    },
13    {
14      "Effect": "Allow",
15      "Action": [
16        "route53>ListHostedZones",
17        "route53>ListResourceRecordSets"
18      ],
19      "Resource": [
20        "*"
21      ],
22    }
23  ]
24 }
25

```

1단계
권한 지정

2단계
검토 및 생성

검토 및 생성

권한을 검토하고 세부 정보 및 태그를 지정합니다.

정책 세부 정보

정책 이름
이 정책을 사용하는 의미 있는 이름을 입력합니다.
AllowExternalDNSUpdates
최대 128자입니다. 영숫자 및 "+", "-", "@" 문자를 사용하세요.

설명 - 선택 사항
이 정책에 대하여 간단한 설명을 추가합니다.
Allow access to Route53 Resources for ExternalDNS

최대 1,000자입니다. 영숫자 및 "+", "-", "@" 문자를 사용하세요.

이 정책에 정의된 권한

편집

이 정책 문서에 정의된 권한은 적용되거나 거부되는 작업을 지정합니다. IAM 자격 증명(사용자, 사용자 그룹 또는 역할)에 대한 권한을 정의하려면 여기에 정책을 연결합니다.

Q. 검색

허용(서비스 418개 중 1개)

나머지 서비스 417개 표시

서비스	액세스 수준	리소스	요청 조건
Route 53	제한적: 나열, 쓰기	Multiple	None

The screenshot shows the AWS IAM 'AllowExternalDNSUpdates' policy details page. At the top right, there are '편집' (Edit) and '삭제' (Delete) buttons. Below the title, it says 'Allow access to Route53 Resources for ExternalDNS'. The policy has one condition: '제한적: 나일, 쓰기' (Restrictive: Read, Write) for the 'Route_53' service. The ARN is highlighted with a red box: `arn:aws:iam::533267405941:policy/AllowExternalDNSUpdates`.

4.5.2. Step-03: Create IAM Role, k8s Service Account & Associate IAM Policy

- As part of this step, we are going to create a k8s Service Account named `external-dns` and also a AWS IAM role and associate them by annotating role ARN in Service Account.
- In addition, we are also going to associate the AWS IAM Policy `AllowExternalDNSUpdates` to the newly created AWS IAM Role.

Step-03-01: Create IAM Role, k8s Service Account & Associate IAM Policy

```
# Template
eksctl create iamserviceaccount \
--name external-dns \
--namespace default \
--cluster ksn-eks-cluster \
--attach-policy-arn arn:aws:iam::533267405941:policy/AllowExternalDNSUpdates \
--approve \
--override-existing-serviceaccounts
```

```
ksnKMu1-MacBookPro ksn-k8s-project % kubectl apply -f ingressclass-resource.yaml
ingressclass networking.k8s.io/my-aws-ingress-class created
ksnKMu1-MacBookPro ksn-k8s-project % eksctl create iamserviceaccount \
--name external-dns \
--namespace default \
--cluster ksn-eks-cluster \
--attach-policy-arn arn:aws:iam::533267405941:policy/AllowExternalDNSUpdates \
--approve \
--override-existing-serviceaccounts
2024-06-21 00:03:32 [!] 1 iamserviceaccount (kube-system/aws-load-balancer-controller) will be excluded
2024-06-21 00:03:32 [!] 1 iamserviceaccount (default/external-dns) was included (based on the include/exclude rules)
2024-06-21 00:03:32 [!] metadata of serviceaccounts that exist in Kubernetes will be updated, as --override-existing-serviceaccounts was set
2024-06-21 00:03:32 [!] 1 task:
  2 sequential sub-tasks:
    create IAM role for serviceaccount "default/external-dns",
    create serviceaccount "default/external-dns",
  } 2024-06-21 00:03:32 [!] building iamserviceaccount stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-default-external-dns"
2024-06-21 00:03:32 [!] defining CloudFormation stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-default-external-dns"
2024-06-21 00:03:32 [!] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-default-external-dns"
2024-06-21 00:03:32 [!] waiting for CloudFormation stack "eksctl-ksn-eks-cluster-addon-iamserviceaccount-default-external-dns"
2024-06-21 00:04:02 [!] created serviceaccount "default/external-dns"
```

Step-03-02: Verify the Service Account

- Verify external-dns service account, primarily verify annotation related to IAM Role

```
# List Service Account
kubectl get sa external-dns

# Describe Service Account
kubectl describe sa external-dns
Observation:
1. Verify the Annotations and you should see the IAM Role is present on the Service Account
```

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get sa external-dns
NAME      SECRETS   AGE
external-dns  0          111s
```

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get sa external-dns
NAME      SECRETS   AGE
external-dns  0          111s
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl describe sa external-dns
Name:           external-dns
Namespace:      default
Labels:         app.kubernetes.io/managed-by=eksctl
Annotations:    eks.amazonaws.com/role-arn: arn:aws:iam::533267405941:role/eksctl-eksCluster-addon-iamserviceaccount-def-Role1-21bwokc90kjC
Image pull secrets: <none>
Mountable secrets: <none>
Tokens:         <none>
Events:        <none>
```

4.5.3. Deploy External DNS

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f Deploy-ExternalDNS.yaml
Warning: resource servicename/external-dns is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created directly by kubectl or --save-config or kubectl apply. The missing annotation will be patched automatically.
serviceaccount/external-dns configured
clusterrole.rbac.authorization.k8s.io/external-dns created
clusterrolebinding.rbac.authorization.k8s.io/external-dns-viewer created
deployment.apps/external-dns created
```

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get all
NAME                 READY   STATUS    RESTARTS   AGE
pod/external-dns-85468db78b-95sx  1/1     Running   0          56s

NAME              TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
service/kubernetes  ClusterIP  10.100.0.1   <none>        443/TCP   8h
service/mysql       ExternalName <none>        mgntdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com  <none>    7h32m

NAME            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/external-dns  1/1     1           1           3m4s

NAME            DESIRED   CURRENT   READY   AGE
replicaset.apps/external-dns-85468db78b  1        1        1        56s
```

Route 53 (도메인 등록)

Route 53 > 등록된 도메인 > 도메인 등록

도메인 등록 정보

도메인 이름의 요금은 최상위 도메인(TLD)에 따라 다릅니다. 여러 TLD의 가격 [\[링크\]](#)에서 자세한 내용을 확인하세요.

도메인 검색	선택된 도메인 (1/5)
도메인의 가용성 확인 <input type="text" value="ksncloud.com"/> <input type="button" value="검색"/> <input type="button" value="X"/>	도메인 등록 수수료 ksncloud.com <input type="button" value="제거"/>

AWS Certificated Manager (SSL/TLS 인증서 등록)

AWS Certificate Manager > 인증서 > 인증서 요청

인증서 요청

인증서 유형 정보

ACM 인증서는 인터넷 또는 내부 네트워크 내에서 안전한 통신 액세스를 설정하는 데 사용할 수 있습니다. ACM이 제공할 인증서 유형을 선택합니다.

퍼블릭 인증서 요청
Amazon으로부터 퍼블릭 SSL/TLS 인증서를 요청합니다. 기본적으로 브라우저 및 운영 체제는 퍼블릭 인증서를 신뢰합니다.

프라이빗 인증서 요청
발급할 수 있는 프라이빗 CA가 없습니다.

프라이빗 인증서를 요청하려면 Private Certificate Authority(CA)를 생성해야 합니다. Private CA를 생성하려면 다음을 참조하십시오.
[AWS Private Certificate Authority \[링크\]](#)

wildcard certificate → all the sub domains are supported here

퍼블릭 인증서 요청

도메인 이름

인증서에 대해 하나 이상의 도메인 이름을 제공합니다.

완전히 정규화된 도메인 이름 [정보](#)

*.ksncloud.com

이 인증서에 다른 이름 추가

이 인증서에 이름을 추가할 수 있습니다. 예를 들어, 'www.example.com'에 대한 인증서를 요청하는 경우 고객이 두 이름 중 하나로 사이트에 접속할 수 있도록 'example.com'이라는 이름을 추가할 수 있습니다.

검증 방법

[정보](#)
도메인 소유권을 검증하기 위한 방법 선택

DNS 검증 – 권장

인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한이 있는 경우 이 옵션을 선택합니다.

이메일 검증

인증서 요청에서 도메인에 대한 DNS 구성을 수정할 권한을 소유하지 않거나 획득할 수 없는 경우 이 옵션을 선택합니다.

키 알고리즘

[정보](#)
암호화 알고리즘을 선택합니다. 일부 알고리즘은 일부 AWS 서비스에서 지원되지 않을 수 있습니다.

RSA 2048

RSA는 가장 널리 사용되는 키 유형입니다.

ECDSA P256

암호화 강도는 RSA 3072와 동일합니다.

ECDSA P384

암호화 강도는 RSA 7680와 동일합니다.

wildcard certificate → all the sub domains are supported here

① ID가 있는 인증서를 요청했습니다. d3da00d4-c117-4486-9155-4284d049e7c5
확인 대기 중 상태의 인증서 요청이 생성되었습니다. 인증서의 검증 및 승인을 완료하려면 추가 작업이 필요합니다.

AWS Certificate Manager > 인증서 > [d3da00d4-c117-4486-9155-4284d049e7c5](#) >
Amazon Route 53에서 DNS 레코드 생성

Amazon Route 53에서 DNS 레코드 생성 (1/1)

도메인 검색

1 일치

검증 상태 = 검증 대기 중

검증 상태 = 실패

도메인이 Route 53에 있습니까? = 예

필터 지우기

< 1 >

도메인

검증 상태

도메인이 Route 53에 있습니까?

*.ksncloud.com

④ 검증 대기 중

예

취소

레코드 생성

인증서 (1)								
		도메인 이름	유형	상태	사용 중	갱신 자격	키 알고리즘	요청
<input type="checkbox"/>	인증서 ID	d3da00d4-c117-4486-9155-4284d049e7c5	*.ksncloud.com	Amazon 발급	발급됨	아니요	부작격	RSA 2048

4.8. Amazon Simple Email Service

SMTP 설정

Amazon SES > SMTP 설정

SMTP(Simple Mail Transfer Protocol) 설정

SMTP 지원 프로그래밍 언어, 이메일 서버 또는 애플리케이션을 사용하여 Amazon SES SMTP 인터페이스에 연결할 수 있습니다. 다음에서 이 이메일 전송 방법을 구성하려면 다음 정보와 SMTP 자격 증명 세트가 필요합니다. 아시아 태평양(서울).

SMTP(Simple Mail Transfer Protocol) 설정 정보

SMTP 엔드포인트 email-smtp.ap-northeast-2.amazonaws.com	STARTTLS 포트 25, 587 또는 2587	기존 SMTP 보안 인증 관리
전송 계층 보안(TLS) 필수 항목	사용자 지정 SSL 클라이언트 지원 -	SMTP 인터페이스에 액세스하려면 Amazon SES SMTP 사용자 이름과 암호가 있어야 합니다. 이러한 보안 인증은 AWS 액세스 키와 다르며 리전별로 고유합니다.
	TLS 래퍼 포트 465 또는 2465	기존 SMTP 보안 인증 관리를

IAM > 사용자 > 사용자 생성

1단계 사용자 세부 정보 지정

2단계 SMTP 자격 증명 검색

사용자 세부 정보 지정

SMTP 사용자 생성

Amazon SES에서 SMTP 인증을 위한 SMTP 자격 증명을 사용하여 IAM 사용자를 생성합니다.

사용자 이름
`ses-smtp-user.20240619-143033`

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

사용자에 대한 권한 정책

이 권한 정책은 사용자에게 AWS SES에 액세스할 수 있는 권한을 부여합니다.

```

1 - {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "ses:SendRawEmail",
7       "Resource": "*"
8     }
9   ]
10 }
  
```

- SMTP 자격 증명
 - SMTP 엔드포인트: `email-smtp.ap-northeast-2.amazonaws.com`
 - IAM 사용자 이름: `ses-smtp-user.20240619-143033` /
 - SMTP 사용자 이름: `AKIAXYKJW3R23FXPGZGB. / AKIAXYKJW3R2Y6EU4GGE`
 - SMTP 비밀번호: `BDsWrsKH2Q8aoL9SZf+RzBIIJDEjz51FWrFngGaAdeCV / BEJ6tbbTcl0uyN7UouFpOflyswn+NARdF4MkImQZVM5c`

IAM > 사용자 > ses-smtp-user.20240619-143033

ses-smtp-user.20240619-143033 정보

삭제

요약

ARN arn:aws:iam::533267405941:user/ses-smtp-user.20240619-143033	콘솔 액세스 비활성화됨	액세스 키 1 AKIAKYKJW3R23FXPGZGB - Active ④ 사용된 적 없음. 13시간 기준.
생성됨 June 19, 2024, 14:39 (UTC+09:00)	마지막 콘솔 로그인 -	액세스 키 2 액세스 키 만들기

권한 | 그룹 | 태그 (1) | 보안 자격 증명 | 액세스 관리자

권한 정책 (1)

사용자에게 직접 연결된 정책을 통해 또는 그룹을 통해 권한을 정의합니다.

필터링 기준 유형
검색 모든 유형

선택 이름	유형	연결 방식
<input type="checkbox"/> AmazonSesSendingAccess	고객 인라인	인라인

< 1 > ⌂

Amazon SES > 구성: 자격 증명 > 자격 증명 생성

자격 증명 생성

자격 증명은(는) Amazon SES를 통해 이메일을 전송할 때 사용하는 도메인, 하위 도메인 또는 이메일 주소입니다. 도메인 수준의 자격 증명 확인은 확인된 하나의 도메인 자격 증명으로 모든 이메일 주소까지 확장됩니다.

자격 증명 세부 정보

보안 인증 유형

도메인
도메인의 소유권을 확인하려면 DNS 설정에 액세스하여 필요한 레코드를 추가해야 합니다.

이메일 주소
이메일 주소의 소유권을 확인하려면 확인 이메일을 열 수 있는 받은 편지함에 액세스할 수 있어야 합니다.

ⓘ 도메인 자격 증명을 확인하지 않고 이메일 주소 자격 증명으로 이메일을 보내면 도메인의 DMARC 정책에 따라 메시지가 차단되거나 거부될 수 있습니다. [DMARC 및 도메인의 DMARC 정책을 조회하는 방법에 대해 자세히 알아보세요.](#)

이메일 주소

이메일 주소에는 더하기 기호(+), 등호(=) 및 밑줄(_)을 포함하여 최대 320자를 포함할 수 있습니다.

기본 구성 세트 활성화
이 옵션을 활성화하면 전송 시 구성 세트가 지정되지 않을 때마다 활성화된 구성 세트가 기본적으로 이 자격 증명에서 전송된 메시지에 적용됩니다.

자격 증명 생성

자격 증명은(는) Amazon SES를 통해 이메일을 전송할 때 사용하는 도메인, 하위 도메인 또는 이메일 주소입니다. 도메인 수준의 자격 증명 확인은 확인된 하나의 도메인 자격 증명으로 모든 이메일 주소까지 확장됩니다.

자격 증명 세부 정보

보안 인증 유형

도메인
도메인의 소유권을 확인하려면 DNS 설정에 액세스하여 필요한 레코드를 추가해야 합니다.

이메일 주소
이메일 주소의 소유권을 확인하려면 확인 이메일을 열 수 있는 받은 편지함에 액세스할 수 있어야 합니다.

도메인 자격 증명을 확인하지 않고 이메일 주소 자격 증명으로 이메일을 보내면 도메인의 DMARC 정책에 따라 메시지가 차단되거나 거부될 수 있습니다. [DMARC 및 도메인의 DMARC 정책을 조회하는 방법에 대해 자세히 알아보세요.](#)

이메일 주소

seongnyeon.kim@ipageon.com

이메일 주소에는 더하기 기호(+)와 등호(=) 및 밑줄(_)을 포함하여 최대 320자를 포함할 수 있습니다.

기본 구성 세트 할당
이 옵션을 활성화하면 전송 시 구성 세트가 지정되지 않을 때마다 할당된 구성 세트가 기본적으로 이 자격 증명에서 전송된 메시지에 적용됩니다.

4.9. Deploy SMTP ExternalName Service

AWS Certificate Manager > 인증서 > d3da00d4-c117-4486-9155-4284d049e7c5

d3da00d4-c117-4486-9155-4284d049e7c5

삭제

인증서 상태

식별자: 17-4486-9155-4284d049e7c5
클린보드에 복사됨

상태: 발급됨

arn:aws:acm:ap-northeast-2:533267405941:certificate/d3da00d4-c117-4486-9155-4284d049e7c5

유형: Amazon 발급

도메인 (1)

Route 53에서 레코드 생성 CSV로 내보내기

도메인	상태	갱신 상태	유형	CNAME 이름	CNAME 값
*.ksncloud.com	성공	-	CNAME	_722be10815b6380de8c8e335f0b01247.ksncloud.com.	_eece8efbca27d82a0400211a75f validations.aws.

세부 정보

사용 중 아니요	일련 번호: 0b:44:cb:2f:28:b2:3f:8a:64:74:91:3:a:4:c:78:f0	요청 시간: 6월 19, 2024, 13:47:2 (UTC+09:00)	갱신 차례: 부적격
도메인 이름	퍼블릭 키 정보	발급 시간	

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f SMTP-ExternalName-Service.yaml
service/smtp-service created
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f Notification-MS-Deployment.yaml -f Notification-MS-ClusterIP-Service.yaml
deployment.apps/notification-microservice created
service/notification-clusterip-service created
```

```
! Notification-MS-Deployment.yaml
1 # NotificationMicroservice-Deployment.yaml
2 apiVersion: apps/v1
3 kind: Deployment
4 metadata:
5   name: notification-microservice
6   labels:
7     app: notification-restapp
8 spec:
9   replicas: 1
10  selector:
11    matchLabels:
12      app: notification-restapp
13  template:
14    metadata:
15      labels:
16        app: notification-restapp
17  spec:
18    containers:
19      - name: notification-service
20        image: stackimplify/kube-notifications-microservice:1.0.0
21        ports:
22          - containerPort: 8096
23        imagePullPolicy: Always
24        env:
25          - name: AWS_MAIL_SERVER_HOST
26            value: "smtp-service" # SMTP-ExternalName-Servie.yaml 에서 metadata.name 과 일치
27          - name: AWS_MAIL_SERVER_USERNAME
28            value: "AKIAIXYKJW3R23FXPGZGB"
29          - name: AWS_MAIL_SERVER_PASSWORD
30            value: "BDsWrskH2Q8aoL9SZf+RzBllJDEjz51FWrFngGaAdeCV"
31          - name: AWS_MAIL_SERVER_FROM_ADDRESS
32            value: "v1999vvv@gmail.com" # Email will be initiated from here
```

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f UserManagement-MS-Deployment.yaml -f UserManagement-NodePort-Service.yaml
deployment.apps/usermgmt-microservice created
secret/mysql-db-password created
service/usermgmt-restapp-nodeport-service created
```

```
[ksn@KIMui-MacBookPro ksn-k8s-project % kubectl apply -f ALB-Ingress-SSL-Redirect-ExternalDNS.yaml
ingress.networking.k8s.io/eks-microservices-demo created
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get pods
NAME                  READY   STATUS    RESTARTS   AGE
external-dns-85468db78b-95srx   1/1     Running   0          70m
notification-microservice-9fd66ddd-d7z9f   1/1     Running   0          16m
notification-microservice-9fd66ddd-p67bp   1/1     Running   0          16m
usermgmt-microservice-55d4689598-b48lw   1/1     Running   0          3m34s
```

EC2 > 대상 그룹

대상 그룹 (1/1) 정보

이름	ARN	포트	프로토콜	대상 유형	로드 밸런서	VPC ID
k8s-default-usermgmt-87e43dfc3c	arn:aws:elasticloadbalancing:ap-northeast-2:1231060885:targetgroup/k8s-default-usermgmt-87e43dfc3c	31859	HTTP	인스턴스	eks-microservices-demo	vpc-03d8108a943490477

대상 그룹: k8s-default-usermgmt-87e43dfc3c

세부 정보 | 대상 | 모니터링 | 상태 검사 | 속성 | 태그

등록된 대상 (2) 정보

대상 그룹은 지정한 프로토콜 및 포트 번호를 사용하여 등록된 개별 대상으로 요청을 리우팅합니다. 상태 확인은 대상 그룹의 상태 확인 설정에 따라 등록된 모든 대상에 대해 수행됩니다. 이상 탐지는 정상 대상이 3개 이상 있는 HTTP/HTTPS 대상 그룹에 자동으로 적용됩니다.

인스턴스 ID	이름	포트	영역	상태 확인	상태 확인 세부 정보	시작 시간
i-064f9df31e483f28c	ksn-eks-cluster... 31859	ap-northeast-2a	Healthy	-	2024년 6월 20일, 16:31 (UTC+09:00)	
i-0e3c4cd7e7c3fb070	ksn-eks-cluster... 31859	ap-northeast-2c	Healthy	-	2024년 6월 20일, 16:31 (UTC+09:00)	

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get ingress
NAME          CLASS      HOSTS    ADDRESS          PORTS   AGE
eks-microservices-demo   my-aws-ingress-class   *       eks-microservices-demo-1231060885.ap-northeast-2.elb.amazonaws.com   80      18m
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % nslookup eks-microservices-demo-1231060885.ap-northeast-2.elb.amazonaws.com
Server: 210.220.163.82
Address: 210.220.163.82#53

Non-authoritative answer:
Name: eks-microservices-demo-1231060885.ap-northeast-2.elb.amazonaws.com
Address: 3.39.127.230
Name: eks-microservices-demo-1231060885.ap-northeast-2.elb.amazonaws.com
Address: 15.165.122.72
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % nslookup services.ksncloud.com
Server: 210.220.163.82
Address: 210.220.163.82#53

Non-authoritative answer:
Name: services.ksncloud.com
Address: 3.39.127.230
Name: services.ksncloud.com
Address: 15.165.122.72

ksn@KIMui-MacBookPro ksn-k8s-project % nslookup ums.ksncloud.com
Server: 210.220.163.82
Address: 210.220.163.82#53

Non-authoritative answer:
Name: ums.ksncloud.com
Address: 3.39.127.230
Name: ums.ksncloud.com
Address: 15.165.122.72
```

https://services.ksncloud.com/usermgmt/health-status

User Management Service UP and RUNNING - V1

services.ksncloud.com/usermgmt/notification-service-info

```
Notification Service IP Address: 192.168.81.56
Hostname: notification-microservice-9fd66ddd-p67bp
Application Version: V1
```

The screenshot shows the Postman interface with the following details:

- Collection:** AWS-EKS-Microservices
- Request Type:** GET
- URL:** {{url}}/usermgmt/health-status
- Params:** Key: Value
- Body:** Status: 200 OK, Time: 96 ms, Size: 392 B
- Test Results:** 1 User Management Service UP and RUNNING - V1

The browser window displays the following information:

- Address bar: https://services.ksncloud.com/usermgmt/users
- Content pane: A large black area indicating no data is present.

The screenshot shows the Postman interface with the following details:

- Collection:** AWS-EKS-Microservices
- Request Type:** POST
- URL:** {{url}}/usermgmt/user
- Body:** JSON


```

1   {
2     "username": "MSA-TEST1",
3     "email": "seongnyeon.kim@ipageon.com",
4     "role": "ROLE_ADMIN",
5     "enabled": true,
6     "firstname": "SeongNyeon",
7     "lastname": "Kim",
8     "password": "Pass@123"
9   }
      
```
- Body:** Status: 200 OK, Time: 551 ms, Size: 308 B

Stack Simplify account creation.



받은편지함

성년

나 오후 7:22

받는사람: 나 ▾



...



한국어 번역



Hello SeongNyeon, your stack simplify account is created successfully!

CloudWatch

The screenshot shows the AWS IAM Policy Editor interface. A search bar at the top contains the text "CloudWatchAgentServer". Below the search bar, a table lists policy results. The first result is "CloudWatchAgentServerPolicy", which is selected and highlighted with a blue border. The table includes columns for "정책 이름" (Policy Name), "설명" (Description), and "AWS 관리형" (AWS-managed). At the bottom right of the table, there are buttons for "취소" (Cancel) and "권한 추가" (Add permissions).

<https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container->

Insights-setup-EKS-quickstart.html

```
curl -s https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/ksn-eks-cluster/;s/{{region_name}}/ap-northeast-2/" | kubectl apply -f -
```

```
|ksn@KMu-MacBookPro ksn-k8s-project % curl -s https://raw.githubusercontent.com/aws-samples/amazon-cloudwatch-container-insights/latest/k8s-deployment-manifest-templates/deployment-mode/daemonset/container-insights-monitoring/quickstart/cwagent-fluentd-quickstart.yaml | sed "s/{{cluster_name}}/ksn-eks-cluster/;s/{{region_name}}/ap-northeast-2/" | kubectl apply -f -
```

```
serviceaccount/cloudwatch-agent created
serviceaccount/cloudwatch-agent-role created
clusterrole.rbac.authorization.k8s.io/cloudwatch-agent-role created
clusterrolebinding.rbac.authorization.k8s.io/cloudwatch-agent-role-binding created
configmap/cwagent-config created
daemonset/fluentd-cloudwatch created
configmap/cluster-created created
serviceaccount/fluent created
clusterrole.rbac.authorization.k8s.io/fluentd-role created
clusterrolebinding.rbac.authorization.k8s.io/fluentd-role-binding created
configmap/fluentd-config created
daemonset.apps/fluentd-cloudwatch created
```

```
kubectl -n amazon-cloudwatch get daemonsets
```

```
|ksn@KMu-MacBookPro ksn-k8s-project % kubectl -n amazon-cloudwatch get daemonsets
NAME      DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
cloudwatch-agent   2         2         2         2           2          kubernetes.io/os=linux   63s
fluentd-cloudwatch 2         2         2         2           2          <none>        63s
```

5. 이슈리스트

```
|ksn@KMu-MacBookPro ksn-k8s-project % kubectl run -it --rm --image=mysql:latest --restart=Never mysql-client -- mysql -h umgtdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com -u dbadmin -pdbpassword
mysql: [Warning] Using a password on the command line interface can be insecure.
ERROR 2005 (HY000): Unknown MySQL server host 'umgtdb.chi680k06fii.ap-northeast-2.rds.amazonaws.com' (-2)
pod "mysql-client" deleted
pod default/mysql-client terminated (Error)
```

```
|ksn@KMu-MacBookPro ksn-k8s-project % kubectl apply -f ALB-Ingress-SSL-Redirect-ExternalDNS.yaml
error: resource mapping not found for name: "eks-microservices" namespace: "" from "ALB-Ingress-SSL-Redirect-ExternalDNS.yaml": no matches for kind "Ingress" in version "extensions/v1beta1"
ensure CRDs are installed first
```

<https://kubernetes.io/docs/reference/using-api/deprecation-guide/>

apiVersion: extensions/v1beta1 → [networking.k8s.io/v1](https://kubernetes.io/docs/reference/api/networking.k8s.io/v1/)

```
|ksn@KMu-MacBookPro ksn-k8s-project % kubectl apply -f ALB-Ingress-SSL-Redirect-ExternalDNS.yaml
Error from server (BadRequest): error when creating "ALB-Ingress-SSL-Redirect-ExternalDNS.yaml": Ingress in version "v1" cannot be handled as a Ingress: strict decoding error: unknown field "spec.rules[0].http.paths[0].backend.serviceName", unknown field "spec.rules[0].http.paths[0].backend.servicePort", unknown field "spec.rules[0].http.paths[1].backend.serviceName", unknown field "spec.rules[0].http.paths[1].backend.servicePort"
```

```
spec:
  rules:
    - http:
        paths:
          - path: /* # SSL Redirect Setting
            backend:
              serviceName: ssl-redirect
              servicePort: use-annotation
          - path: /*
            backend:
              serviceName: usermgmt-restapp-nodeport-service
              #UserManagement-NodePort-Service.yaml의 metadata.name과 일치
              #왜냐하면 UserManagement NodePort Service로 트래픽을 보내기때문
              servicePort: 8095
```

```
# SSL Redirect Setting
alb.ingress.kubernetes.io/ssl-redirect: '443'
```

```
rules:
  - http:
      paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: usermgmt-restapp-nodeport-service #UserManagement-NodePort-Service.yaml의 metadata.name과 일치
              #왜냐하면 UserManagement NodePort Service로 트래픽을 보내기때문
              port:
                number: 8095
```

```
ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get pods
NAME           READY   STATUS    RESTARTS   AGE
notification-microservice-9fd66ddd-4swlp  1/1     Running   0          3h16m
notification-microservice-9fd66ddd-xtl16  1/1     Running   0          3h16m
usermgmt-microservice-cdd654c6d-8qcpx    0/1     Pending   0          3h15m
```

kubectl describe pod <pod 이름>

```

ksn@KIMUi-MacBookPro ksn-k8s-project % kubectl describe pod usermgmt-microservice-cdd654c6d-8qcpk
Name:           usermgmt-microservice-cdd654c6d-8qcpk
Namespace:      default
Priority:      0
Service Account: default
Node:          <none>
Labels:        app=usermgmt-restapp
               pod-template-hash=cdd654c6d
Annotations:   <none>
Status:        Pending
IP:            <none>
Controlled By: ReplicaSet/usermgmt-microservice-cdd654c6d
Init Containers:
  init-db:
    Image:    busybox:1.31
    Port:    <none>
    Host Port: <none>
    Command:
      sh
      -c
      echo -e "Checking for the availability of MySQL Server deployment"; while ! nc -z mysql 3306; do sleep 1; printf "-"; done; echo -e "  >> MySQL DB Server has started";
    Environment: <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-7896n (ro)
Containers:
  usermgmt-restapp:
    Image:    stacksmplify/kube-usermanagement-microservice:1.0.0
    Port:    8095/TCP
    Host Port: 8095
    Liveness: exec [/bin/sh -c nc -z localhost 8095] delay=60s timeout=1s period=10s #success=1 #failure=3
    Readiness: http-get http://:8095/usermgmt/health-status delay=60s timeout=1s period=10s #success=1 #failure=3
    Environment:
      DB_HOSTNAME: mysql
      DB_PORT: 3306
      DB_TYPE: usermgmt
      DB_USERNAME: useradmin
      DB_PASSWORD: <set to the key 'db-password' in secret 'mysql-db-password'> Optional: false
      NOTIFICATION_SERVICE_HOST: notification-clusterip-service
      NOTIFICATION_SERVICE_PORT: 8096
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-7896n (ro)
Conditions:
  Type        Status
  PodScheduled  False
Volumes:
  kube-api-access-7896n:
    Type:      Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds: 3607
    ConfigMapName:  kube-root-ca.crt
    ConfigMapOptional: <nil>
    DownwardAPI: true
  QoS Class:  BestEffort
  Node-Selectors: <none>
  Tolerations: node.kubernetes.io/not-ready:NoExecute opnExists for 300s
                 node.kubernetes.io/unreachable:NoExecute opnExists for 300s
Events:
  Type     Reason     Age         From               Message
  ----     ----     --         --               --
  Warning  FailedScheduling  25s (x42 over 3h25m)  default-scheduler  0/2 nodes are available: 2 Too many pods. preemption: 0/2 nodes are available: 2 No preemption victims found for incoming pod.

```

```

ksn@KIMUi-MacBookPro ksn-k8s-project % aws ec2 describe-instance-types --filters "Name=instance-type,Values=t2.micro" --query "InstanceTypes[].[Type: InstanceType, MaxENI: NetworkInfo.MaximumNetworkInterfaces, IPv4: add: NetworkInfo.Ipv4AddressesPerInterface, IPv6addr: NetworkInfo.Ipv6AddressesPerInterface]" --output table
+-----+-----+-----+-----+
|             DescribeInstanceTypes             |
+-----+-----+-----+-----+
| IPv4addr | IPv6addr | MaxENI | Type  |
+-----+-----+-----+-----+
| 2       | 2       | 2      | t2.micro |
+-----+-----+-----+-----+

```

참고: <https://repost.aws/ko/articles/ARDH72Ep54QKii5DST3SInLA/aws-eks에서-node의-ec-2-타입과-max-pods-설정간-관계>

```

ksn@KIMUi-MacBookPro ksn-k8s-project % kubectl get pods
NAME                      READY   STATUS    RESTARTS   AGE
external-dns-854468db78b-95srx   1/1    Running   0          58m
notification-microservice-9fd66ddd-d7z9f   1/1    Running   0          4m26s
notification-microservice-9fd66ddd-p67bp   1/1    Running   0          4m26s
usermgmt-microservice-786b6d9b97-zrwb2   0/1    CrashLoopBackOff 4 (53s ago) 3m57s

```

```

ksn@KIMUi-MacBookPro ksn-k8s-project % kubectl logs -f $(kubectl get po | egrep -o 'usermgmt-microservice-[A-Za-z0-9-]*')
Default container "usermgmt-restapp" out of: usermgmt-restapp, init-db (init)

:: Spring Boot ::
(v2.1.4.RELEASE)

2024-06-28 16:31:28.327 INFO 1 --- [           main] c.s.r.s.UserManagementApplication      : Starting UserManagementApplication v1.0.0 on usermgmt-microservice-786b6d9b97-zrwb2 with PID 1 (/app.jar started by root)
2024-06-28 16:31:28.331 INFO 1 --- [           main] c.s.r.s.UserManagementApplication      : The following profilers are active: mysqlaws
2024-06-28 16:31:33.335 INFO 1 --- [           main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-06-28 16:31:33.335 INFO 1 --- [           main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.17]
2024-06-28 16:31:33.501 INFO 1 --- [           main] o.a.c.c.C.[localhost].usermgmt       : Initializing Spring embedded WebApplicationContext
2024-06-28 16:31:35.786 ERROR 1 --- [           main] com.zaxxer.hikari.pool.HikariPool      : HikariPool-1 - Exception during pool initialization.

java.sql.SQLSyntaxErrorException: Unknown database 'usermgmt'
at com.mysql.cj.jdbc.exceptions.SQLError.createSQLEception(SQLError.java:120) ~[mysql-connector-java-8.0.15.jar!/]
at com.mysql.cj.jdbc.exceptions.SQLError.createSQLEception(SQLError.java:97) ~[mysql-connector-java-8.0.15.jar!/]
at com.mysql.cj.jdbc.exceptions.SQLExceptionsMapping.translateException(SQLExceptionsMapping.java:122) ~[mysql-connector-java-8.0.15.jar!/]
at com.mysql.cj.jdbc.ConnectionImpl.init(ConnectionImpl.java:455) ~[mysql-connector-java-8.0.15.jar!/]
at com.mysql.cj.jdbc.ConnectionImpl.getInstance(ConnectionImpl.java:248) ~[mysql-connector-java-8.0.15.jar!/]
at com.mysql.cj.jdbc.NonRegisteringDriver.connect(NonRegisteringDriver.java:199) ~[mysql-connector-java-8.0.15.jar!/]
at com.zaxxer.hikari.HikariDriver.getConnection(HikariDriver.java:136) ~[HikariCP-3.2.0.jar!/]
at com.zaxxer.hikari.HikariPool$ProxyFactory.createProxy(HikariPool$ProxyFactory.java:103) ~[HikariCP-3.2.0.jar!/]
at com.zaxxer.hikari.pool.PoolBase.newPoolEntry(PoolBase.java:198) ~[HikariCP-3.2.0.jar!/]
at com.zaxxer.hikari.pool.HikariPool.createPoolEntry(HikariPool.java:447) ~[HikariCP-3.2.0.jar!/]
at com.zaxxer.hikari.pool.HikariPool.<init>(HikariPool.java:116) ~[HikariCP-3.2.0.jar!/]
at com.zaxxer.hikari.HikariPool$ProxyFactory.createProxy(HikariPool$ProxyFactory.java:103) ~[HikariCP-3.2.0.jar!/]
at org.springframework.jdbc.datasource.DataSourceUtils.fetchConnection(DataSourceUtils.java:157) [spring-jdbc-5.1.6.RELEASE.jar!/]
at org.springframework.jdbc.datasource.DataSourceUtils.doGetConnection(DataSourceUtils.java:151) [spring-jdbc-5.1.6.RELEASE.jar!/]
at org.springframework.jdbc.datasource.DataSourceUtils.getConnection(DataSourceUtils.java:78) [spring-jdbc-5.1.6.RELEASE.jar!/]
at org.springframework.jdbc.support.JdbcUtils.extractDatabaseMetaData(JdbcUtils.java:319) [spring-jdbc-5.1.6.RELEASE.jar!/]
at org.springframework.jdbc.support.JdbcUtils.extractDatabaseMetaData(JdbcUtils.java:356) [spring-jdbc-5.1.6.RELEASE.jar!/]

```

```

UserManagement-MS-Deployment.yaml
UserManagement-MS-Deployment.yaml
ALB-Ingress-SSL-Redirect-ExternalDNS.yaml

8   spec:
13     template:
17       spec:
22         containers:
23           - name: usermgmt-restapp
24             ports:
25               - containerPort: 8095
26             env:
27               - name: #RDS
28                 value: "mysql"
29               - name: DB_PORT
30                 value: "3306"
31               - name: DB_NAME
32                 value: "usermgmt"
33               - name: DB_USERNAME
34                 value: "dbadmin"
35               - name: DB_PASSWORD
36                 valueFrom:
37                   secretKeyRef:
38                     name: mysql-db-password
39                     key: db-password
40               - name: NOTIFICATION_SERVICE_HOST
41                 value: "notification-clusterip-service" #Notification-MS-ClusterIP-Service.yaml의 metadata.name과 일치
42               - name: NOTIFICATION_SERVICE_PORT
43                 value: "8096"
44               livenessProbe:
45                 exec:
46                   command:
47                     - /bin/sh
48                     - -c
49                     - nc -z localhost 8095
50               initialDelaySeconds: 60
51               periodSeconds: 10
52             readinessProbe:
53               httpGet:
54                 path: /usermgmt/health-status
55               port: 8095

```

```

ksn@KIMui-MacBookPro ksn-k8s-project % kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
external-dns-85468db78b-95srx   1/1     Running   0          67m
notification-microservice-9fd66ddd-d7z9f   1/1     Running   0          13m
notification-microservice-9fd66ddd-p67bp   1/1     Running   0          13m
usermgmt-microservice-55d4689598-b48lw   0/1     Running   0          33s
usermgmt-microservice-786b6d9b97-zrwb2   0/1     CrashLoopBackOff 6 (4m43s ago) 12m

```

6. References

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

<https://docs.aws.amazon.com/eks/latest/userguide/setting-up.html#installing-eksctl>

<https://docs.aws.amazon.com/eks/latest/userguide/enable-iam-roles-for-service-accounts.html>

<https://docs.aws.amazon.com/eks/latest/userguide/what-is-eks.html>

<https://repost.aws/ko/knowledge-center/eks-persistent-storage>

<https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.4/guide/ingress/annotations/>

<https://github.com/kubernetes-sigs/aws-load-balancer-controller>

<https://kubernetes-sigs.github.io/aws-alb-ingress-controller/guide/ingress/annotation/>

<https://github.com/kubernetes-sigs/external-dns/releases>