4-1. EKS Cluster 구성

클라우드 네이티브 운영체계

Exported on 09/15/2022

Table of Contents

1	사전 준비	3
1.1	네트워크 구성	3
	EKS 클러스터 IAM Role 생성	
	kubectl 수행용 Bastion 인스턴스 생성	
1.4	Bastion 서버 접속 및 Credentials 설정	8
2	EKS 클러스터 생성	9
	클러스터 생성	
	Bastion 서버 kubectl 설정	
3	Fargate 생성	13
	Fargate IAM 역할 생성	
	Fargate 프로필 생성	
	CoreDNS pod 수정	

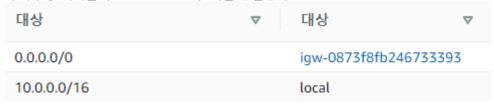
1 사전 준비

1.1 네트워크 구성

* 별도로 준비된 네트워크(VPC)가 있다면 생략한다.

- 1. VPC 생성
 - Name: vpc-sts
 - IPv4 Addr Block: 10.0.0.0/16
- 2. Subnet 생성
 - Public Subnet 1
 - Name: subnet-sts-public-a
 - AZ:az-a
 - IPv4 addr block: 10.0.1.0/24
 - Public Subnet 2
 - Name: subnet-sts-public-c
 - AZ : az-c
 - IPv4 addr block: 10.0.2.0/24
 - Private Subnet 1
 - Name: subnet-sts-private-a
 - AZ:az-a
 - IPv4 addr block: 10.0.3.0/24
 - Private Subnet 2
 - Name: subnet-sts-private-c
 - AZ:az-c
 - IPv4 addr block: 10.0.4.0/24
- 3. IGW 생성
 - Name: igw-sts
 - 1번에서 생성한 vpc에 연결
- 4. NAT Gateway 생성
 - Name: nat-sts
 - Subnet: subnet-sts-public-a
- 5. Route Table 생성
 - Public Route Table
 - Name: rtb-public

• 이 라우팅 테이블에 Public Subnet 두 개를 연결한다.



- Private Route Table
 - Name: rtb-private
 - 이 라우팅 테이블에 Private Subnet 두 개를 연결한다.

대상	∇	대상
0.0.0.0/0		nat-0294a5a1949b33b17
10.0.0.0/16		local

1.2 EKS 클러스터 IAM Role 생성

To create your Amazon EKS cluster role in the IAM console

- 1. Open the IAM console at https://console.aws.amazon.com/iam/ ☑.
- 2. Choose Roles, then Create role.
- 3. Under Trusted entity type, select AWS service.
- 4. From the Use cases for other AWS services dropdown list, choose EKS.
- 5. Choose EKS Cluster for your use case, and then choose Next.
- 6. On the Add permissions tab, choose Next.
- 7. For Role name, enter a unique name for your role, such as eksClusterRole.
- For Description, enter descriptive text such as Amazon EKS Cluster role.
- 9. Choose Create role.

신뢰할 수 있는 엔터티 유형



EC2, Lambda 등의 AWS 서비스가 이 계정에서 작 업을 수행하도록 허용합니 다

AWS 계정

AWS 제'8 사용자 또는 서드 파티에 속한 다른 AWS 계정의 엔 터티가 이 계정에서 작업 을 수행하도록 허용합니 다.

○ 웹 자격 증명

웹 시작 증명 지정된 외부 웹 자격 증명 공급자와 연동된 사용자가 이 역할을 맡아 이 계정에 서 작업을 수행하도록 허 용합니다.

O SAML 2.0 연동

기업 디렉터리에서 SAML 2.0과 연동된 사용자가 이 계정에서 작업을 수행할 수 있도록 허용합니다. ○ 사용자 지정 신뢰 정책

다른 사용자가 이 계정에 서 작업을 수행할 수 있도 록 사용자 지정 신뢰 정책 을 생성합니다.

사용 사례

EC2, Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

일반 사용 사례

O EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

다른 AWS 서비스의 사용 사례:

EKS

 \blacksquare

EKS

Allows EKS to manage clusters on your behalf.

EKS - Cluster

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

IAM → 역할 → 역할 생성

신뢰할 수 있는 엔터티 유형: AWS 서비스

사용 사례 : EKS -Cluster

역할 세부 정보

역할 이름

이 역할을 식별하는 의미 있는 이름을 입력합니다.

EKSClusterRole

최대 64자입니다. 영숫자 및 '+=,.@-_' 문자를 사용하세요.

설명

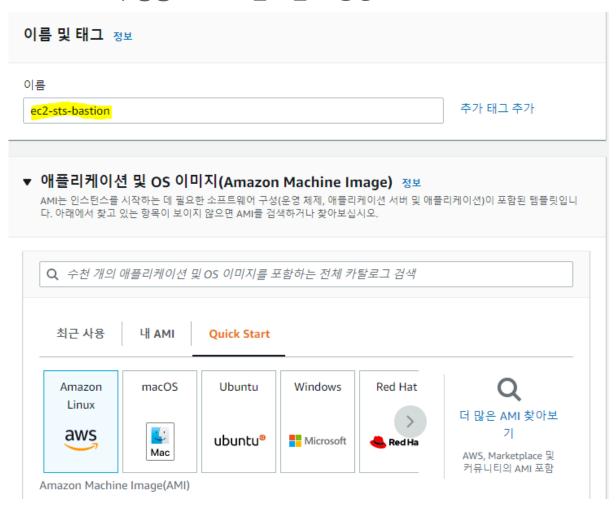
이 역할에 대하여 간단한 설명을 추가합니다.

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

최대 1,000자입니다. 영숫자 및 '+=,.@-_' 문자를 사용하세요.

역할 이름: EKSClusterRole

1.3 kubectl 수행용 Bastion 인스턴스 생성





EC2 → 인스턴스 → 인스턴스 시작

이름 : ec2-sts-bastion 인스턴스 유형 : t3.nano

키 페어 : '새 키 페어 생성' 또는 '기존 키 페어' 선택

VPC: vpc-sts

Subnet : subnet-sts-public-a 퍼블릭 IP 자동 할당 : 활성화

보안 그룹 : 신규 생성 (내 IP로 ssh 접근 가능하도록 설정)

1.4 Bastion 서버 접속 및 Credentials 설정

ssh 접속 후 자신 계정의 credentials 설정

aws configure

2 EKS 클러스터 생성

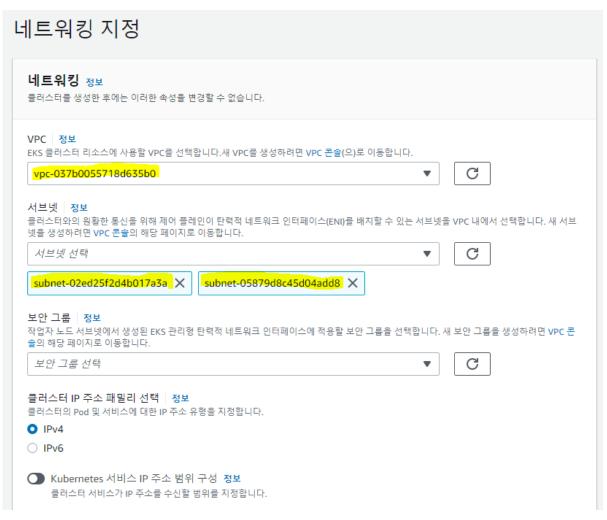
2.1 클러스터 생성



EKS → 클러스터 → EKS 클러스터 생성

이름 : eks-sts-cluster Kubernetes 버전 : 1.22

클러스터 서비스 역할: EKSClusterRole





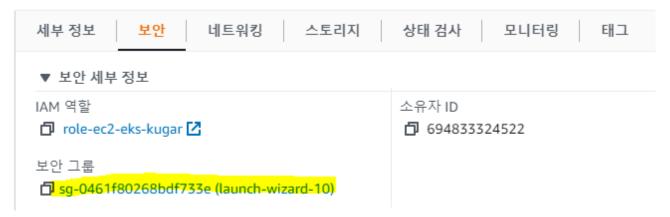
VPC: vpc-sts

Subnet: subnet-sts-private-a, subnet-sts-private-c

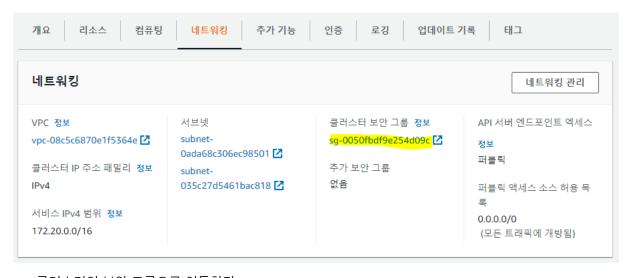
클러스터 엔드포인트 액세스: 프라이빗

2.2 Bastion 서버 kubectl 설정

인스턴스: i-0cd825e971565bb5f(ec2-sts-bastion)



Bastion 서버에서 kubectl을 사용하여 EKS 클러스터로 API를 요청할 수 있도록 443포트를 허용해주도록 한다. Bastion 서버의 보안 그룹을 확인한다.



EKS 클러스터의 보안 그룹으로 이동한다.



인바운드 규칙 편집 → HTTPS에 Bastion SG 추가

bastion 서버 접속하여 아래 명령어를 수행한다.

```
aws eks update-kubeconfig --region [region-code] --name [cluster-name]
```

kubectl 설치

```
curl -o kubectl https://s3.us-west-2.amazonaws.com/amazon-eks/1.22.6/2022-03-09/bin/
linux/amd64/kubectl
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
kubectl get ns
```

3 Fargate 생성

3.1 Fargate IAM 역할 생성

- 1. https://console.aws.amazon.com/iam/ ☑에서 IAM 콘솔을 엽니다.
- 2. 왼쪽 탐색 창에서 역할(Roles)을 선택합니다.
- 3. 역할(Roles) 페이지에서 역할 생성(Create role)을 선택합니다.
- 4. 신뢰할 수 있는 엔터티 선택(Select trusted entity) 페이지에서 다음을 수행합니다.
 - a. 신뢰할 수 있는 엔터티 유형(Trusted entity type) 섹션에서 AWS 서비스(service)를 선택합니다.
 - b. 기타 AWS 서비스에 대한 사용 사례(Use cases for other) 드롭다운에서 EKS를 선택합니다.
 - c. **EKS Fargate pod**를 선택합니다.
 - d. 다음(Next)을 선택합니다.
- 5. 권한 추가(Add permissions) 페이지에서 다음(Next)을 선택합니다.
- 6. 이름, 검토 및 생성(Name, review, and create) 페이지에서 다음을 수행합니다.
 - a. 역할 이름(Role name)에 역할의 고유한 이름(예: AmazonEKSFargatePodExecutionRole)을 입력합니다.
 - b. 태그 추가(Add tags)(선택 사항)에서 태그를 키 값 페어로 연결하여 메타데이터를 역할에 추가합니다. IAM에서 태그 사용에 대한 자세한 내용은 IAM 사용 설명서의 IAM 엔터티 태깅을 참 조하세요.
 - c. 역할 생성(Create role)을 선택합니다.
- 7. 역할(Roles) 페이지에서 AmazonEKSFargatePodExecutionRole에 대한 역할 목록을 검색합니다. 역할을 선택합니다.
- 8. AmazonEKSFargatePodExecutionRole 페이지에서 다음을 수행합니다.
 - a. 신뢰 관계(Trust relationships)를 선택합니다.
 - b. 신뢰 정책 편집(Edit trust policy)을 선택합니다.
- 9. 신뢰 정책 편집(Edit trust policy) 페이지에서 다음 작업을 수행합니다.
 - a. 다음 내용을 복사하여 **신뢰 정책 편집(Edit trust policy)** 양식에 붙여 넣습니다. region-code 을 해당 클러스터가 있는 AWS 리전으로 바꿉니다. 계정의 모든 AWS 리전에서 동일한 역할을 사용하려는 경우 region-code 을 *로 바꾸세요. 111122223333 를 계정 ID로, my-cluster 를 클러스터의 이름으로 바꿉니다. 계정의 모든 클러스터에 대해 동일한 역할을 사용하려는 경우 my-cluster 를 *로 바꾸세요.

b. 정책 업데이트(Update policy)를 선택합니다.

신뢰할 수 있는 엔터티 유형

O AWS 서비스

EC2, Lambda 등의 AWS 서비스가 이 계정에서 작 업을 수행하도록 허용합니 다.

AWS 계정

사용자 또는 서드 파티에 속한 다른 AWS 계정의 엔 터티가 이 계정에서 작업 을 수행하도록 허용합니 다.

○ 웹 자격 증명

집 시 시 O O O 지정된 외부 웹 자격 증명 공급자와 연동된 사용자가 이 역할을 맡아 이 계정에 서 작업을 수행하도록 허 용합니다.

O SAML 2.0 연동

기업 디렉터리에서 SAML 2.0과 연동된 사용자가 이 계정에서 작업을 수행할 수 있도록 허용합니다.

○ 사용자 지정 신뢰 정책

다른 사용자가 이 계정에 서 작업을 수행할 수 있도 록 사용자 지정 신뢰 정책 을 생성합니다.

사용 사례

EC2. Lambda 등의 AWS 서비스가 이 계정에서 작업을 수행하도록 허용합니다.

일반 사용 사례

O EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

다른 AWS 서비스의 사용 사례:

EKS ▼

○ EKS

Allows EKS to manage clusters on your behalf.

O EKS - Cluster

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

O EKS - Nodegroup

Allow EKS to manage nodegroups on your behalf.

EKS - Fargate pod

Allows access to other AWS service resources that are required to run Amazon EKS pods on AWS Fargate

IAM → 역할 → 역할 생성

신뢰할 수 있는 엔터티 유형: AWS 서비스

사용 사례 : EKS - Fargate pod

역할 세부 정보

역할 이름

이 역할을 식별하는 의미 있는 이름을 입력합니다.

EKSFargatePodExecutionRole

최대 64자입니다. 영숫자 및 '+=,.@-_' 문자를 사용하세요.

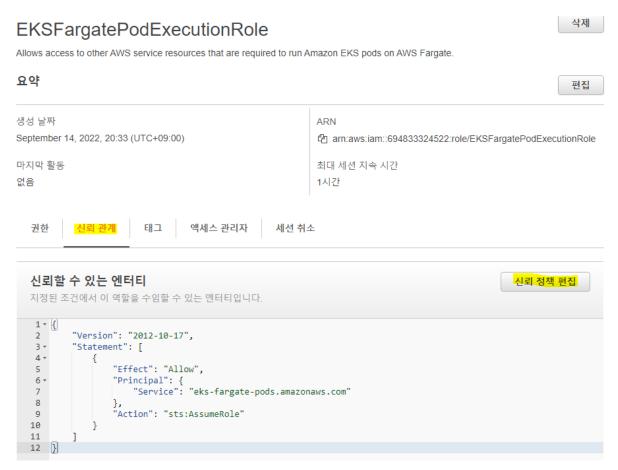
설명

___ 이 역할에 대하여 간단한 설명을 추가합니다.

Allows access to other AWS service resources that are required to run Amazon EKS pods on AWS Fargate.

최대 1,000자입니다. 영숫자 및 '+=,.@-_' 문자를 사용하세요.

역할 이름 : EKSFargatePodExecutionRole



역할(EKSFargatePodExecutionRole) → 신뢰 관계 → 신뢰 정책 편집

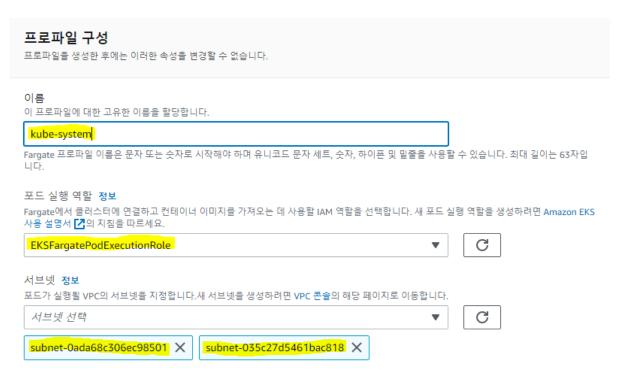
ArnLike에 EKS 클러스터 Arn을 추가한다.

```
1 * {
      "Version": "2012-10-17",
     "Statement": [
4 -
       "Effect": "Allow",
 5
6 +
       "Condition": {
7 -
          "ArnLike": {
      "aws:SourceArn": "arn:aws:eks:ap-northeast-2:694833324522:fargateprofile/eks-sts-cluster/*"
8
9
      }
      },
10
       "Principal": {
11 -
        "Service": "eks-fargate-pods.amazonaws.com"
12
13
      },
14
      "Action": "sts:AssumeRole"
15
16
    ]
17 }
```

3.2 Fargate 프로필 생성



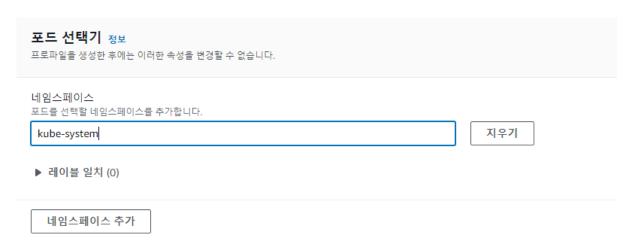
EKS Cluster → 컴퓨팅 탭 > Fargate 프로파일 추가



이름 : kube-system

파드 실행 역할:EKSFargatePodExecutionRole

서브넷: subnet-sts-private-a, subnet-sts-private-c



네임스페이스: kube-system

3.3 CoreDNS pod 수정

default 로 EC2 위에서 수행되도록 설정되어 있기 때문에 Fargate 위에서 수행되도록 수정해준다. eks.amazonaws.com/compute-type¹: ec2 주석 제거

```
kubectl patch deployment coredns \
    -n kube-system \
    --type json \
    -p='[{"op": "remove", "path": "/spec/template/metadata/annotations/
eks.amazonaws.com~1compute-type"}]'
```

재기동

kubectl rollout restart -n kube-system deployment coredns

배포 확인

kubectl get pods -A |grep coredns

¹ http://eks.amazonaws.com/compute-type