TKS Cloud Service

Cluster API로 멀티 테넌시를 고려한 관리형 쿠버네티스 (EKS) 배포하기

SK telecom

Container Solution개발팀 엄주관

개요

- Cluster API로 EKS 배포
- AWS에서 멀티테넌시 EKS 클러스터 구성

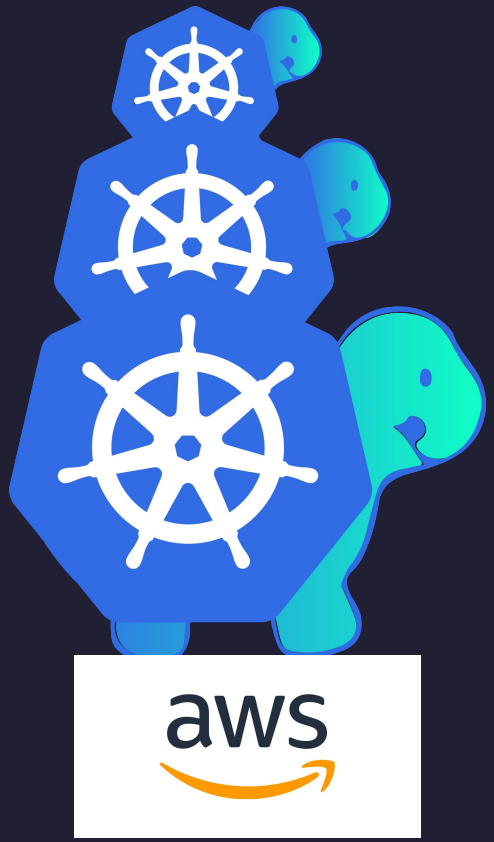
TKS는 AWS뿐만 아니라 다양한 클라우드 인프라에 쿠버네티스를 생성/관리하기 위해 Cluster-api를 표준기술로 사용하고 있습니다.

2022

엄주관
SK텔레콤

Application을 넘어 Infrastructure와 Kubernetes Infrastructure도 GitOps로 관리하기

Cluster API로 EKS 배포



Cluster API란?

Cluster API는 Kubernetes 클러스터를 프로비저닝, 업그레이드 및 운영하기 위한 **선언적 API 및 도구**를 제공하는 Kubernetes 하위 프로젝트입니다 ⇒ K8S를 K8S로 배포/관리
(이제는 많이 익숙해진)커스텀 리소스를 사용해 Cluster API 자원을 정의하고 관리

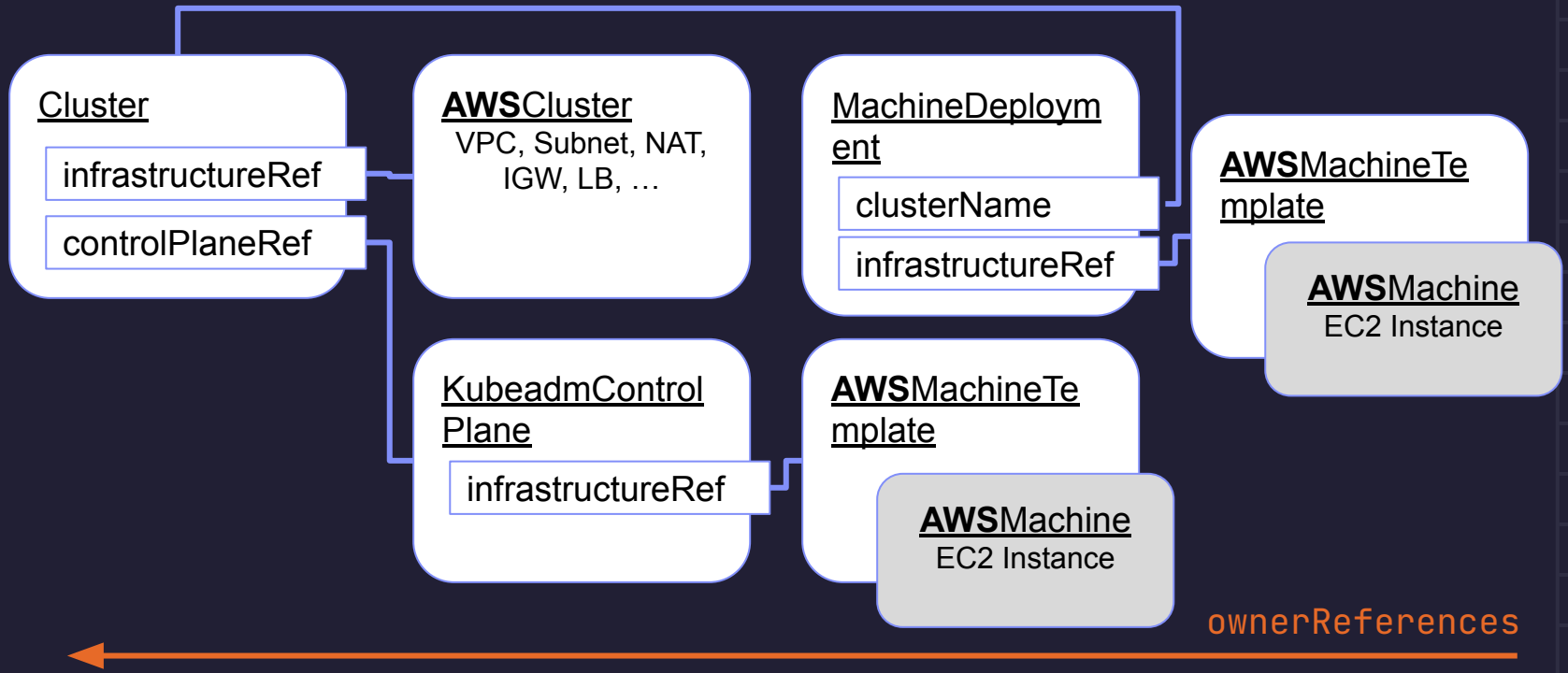
- Application: Deployment → ReplicaSet -> Pod
- Kubernetes Cluster: Cluster → ControlPlane / MachineDeployment → MachineSet → Machine

⇒ 어플리케이션 배포에 쓰이는 기술, 방법들을 동일하게 적용 가능

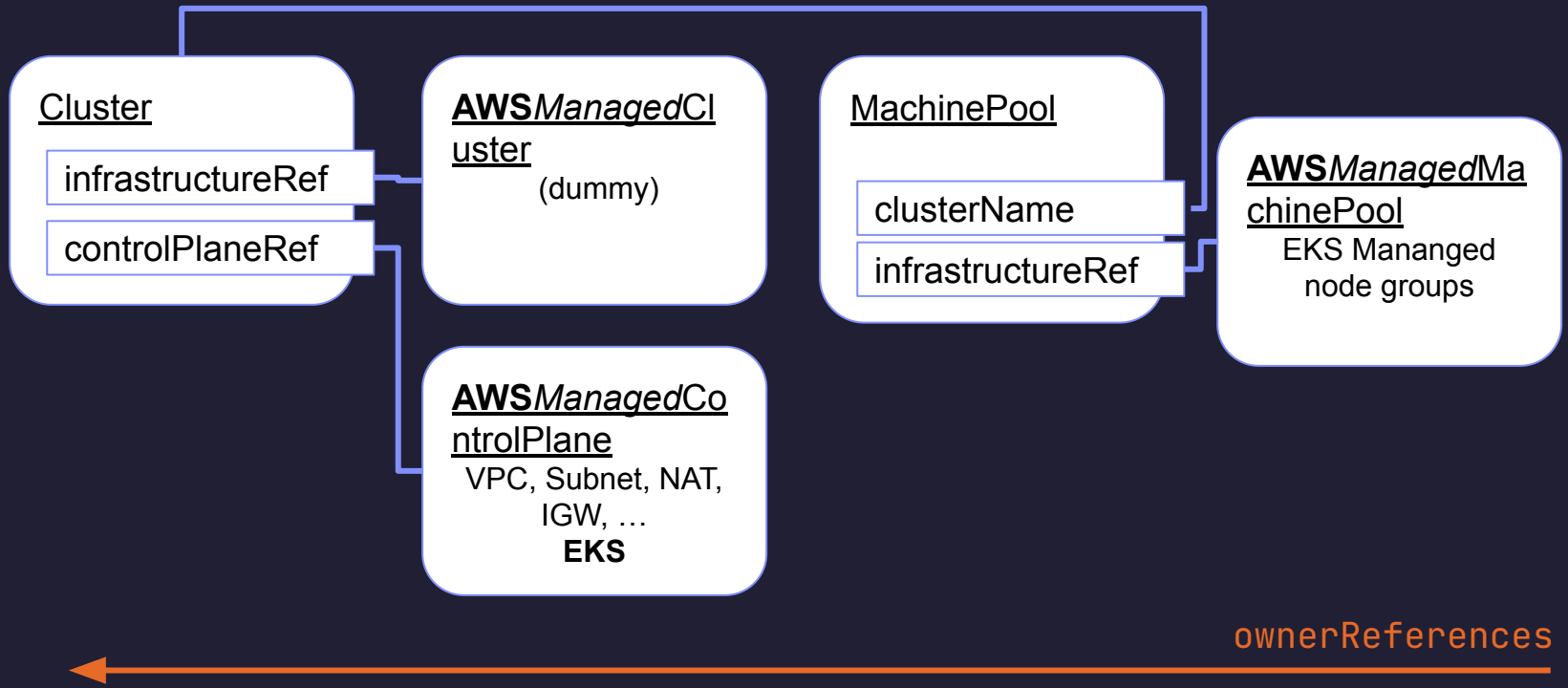
Cluster API 구성 요소

- Management Cluster: Cluster API 자원이 배포되고 컨트롤러가 실행되는 클러스터
- Workload Cluster: Cluster API 컨트롤러에 의해서 생성되는 클러스터
- Infrastructure Provider: 퍼블릭/프라이빗 클라우드 환경, 솔루션에 맞추어 클러스터 또는 머신에 필요한 인프라/컴퓨팅 리소스의 프로비저닝을 담당
  - AWS (Cluster API Provider AWS, CAPA), Azure, Google, VMware, OpenStack, KubeVirt, Meta3(Baremetal), ...
  - Cluster API 상위 프로젝트와는 **독립적인** 커스텀 리소스 정의와 컨트롤러 제공

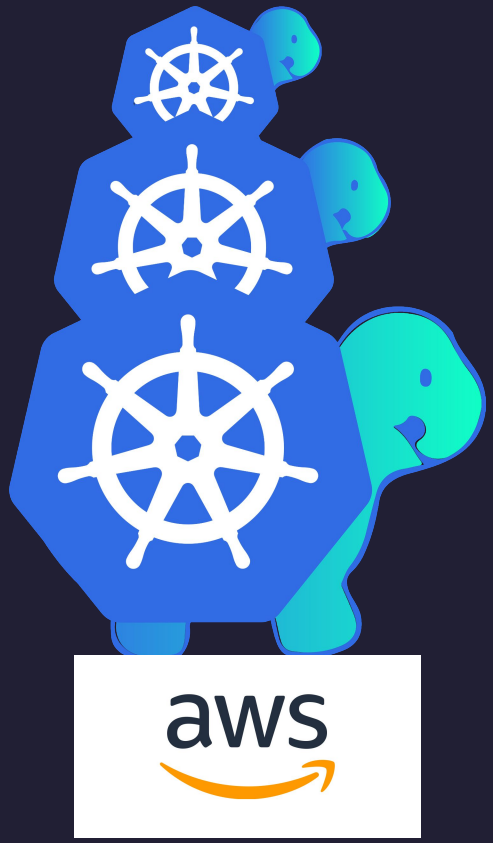
Cluster API 커스텀 리소스 (Self-provisioned)



Cluster API 커스텀 리소스 (EKS)



AWS에서 멀티테넌시 EKS 클러스터 구성



ChatGPT가 알려주는 멀티 테넌시

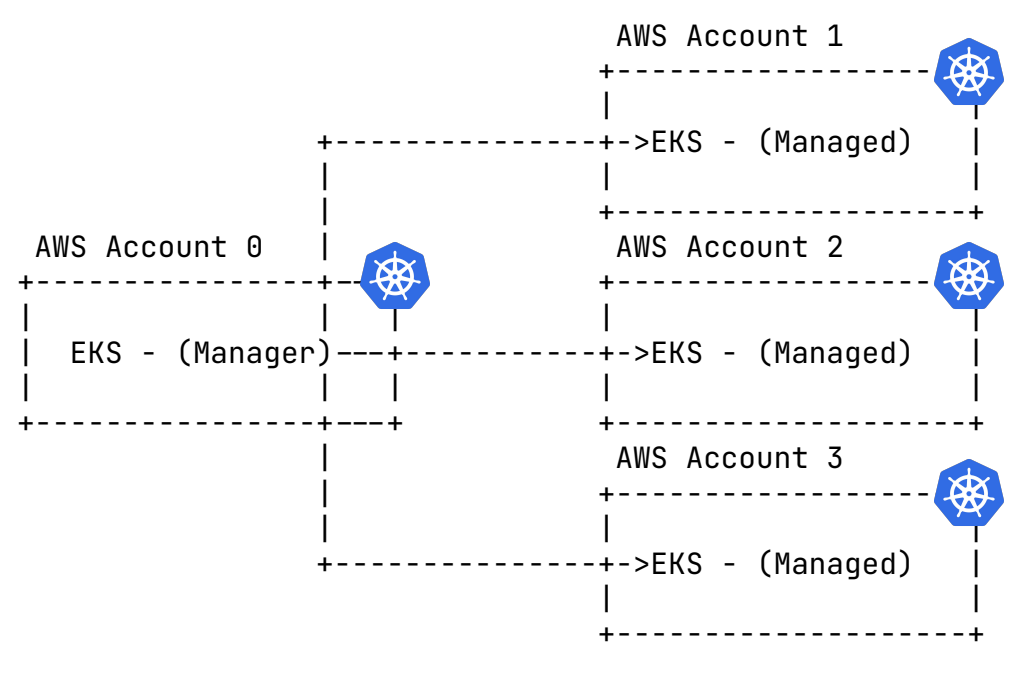
멀티 테넌시가 무슨 뜻이야?

멀티 테넌시란 하나의 소프트웨어나 시스템이 여러 사용자 또는 여러 조직에서 동시에 사용될 수 있는 능력을 말합니다. 즉, 각각의 사용자 또는 조직은 자신만의 독립적인 환경을 갖고, 서로의 데이터나 자원에 영향을 주지 않으면서 공통된 소프트웨어나 시스템을 사용할 수 있습니다.

예를 들어, 클라우드 컴퓨팅에서는 하나의 물리적 서버를 여러 사용자나 조직이 공유하며, 각각의 사용자나 조직은 자신만의 가상 서버 환경을 갖게 됩니다. 이렇게 하면 자원을 효율적으로 활용할 수 있으며, 관리와 유지보수 비용을 줄일 수 있습니다.

또한 멀티 테넌시는 SaaS(소프트웨어 서비스) 모델에서도 많이 사용됩니다. 이 경우에는 하나의 소프트웨어 애플리케이션이 여러 조직이나 사용자에게 제공되며, 각각의 조직이나 사용자는 자신만의 데이터와 설정을 갖게 됩니다.

Cluster API Provider AWS에서 멀티 테넌시



매니지먼트 클러스터와 다른 AWS 계정으로 워크로드 클러스터를 배포

- 조직 별 다른 AWS 계정을 사용
- 용도(개발, 운영)별 다른 AWS 계정을 사용
- **쿠버네티스 as a 서비스를 제공**

AWS에서 API 인증

AWS 요청 주체 (Principal: IAM User)는 아래와 같은 "상시" 자격 증명 (Credential) 정보로 인증 후 AWS API 사용

Cluster API 매니지먼트 클러스터를 구성하는 도구인 clusterctl 실행 전에 아래와 같이 AWS credential 환경 변수를 설정

```
export AWS_ACCESS_KEY_ID=<your-access-key>
export
AWS_SECRET_ACCESS_KEY=<your-secret-access-key>
```

Cluster API Provider AWS 컨트롤러

AWS Crendential 환경 변수 정보는 이후 Cluster API Provider AWS (CAPA) 컨트롤러가 참조하는 Secret으로 저장

```
$ kubectl  get secret -n capa-system
capa-manager-bootstrap-credentials
-ojsonpath='{.data.credentials}' | base64 -d
[default]
aws_access_key_id = XXXXXXXXXXXXXXXXXXX
aws_secret_access_key =
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
region = ap-northeast-2
```

Cluster API, AWS에서 멀티 테넌시

각 계정의 AWS Crendential 정보를 Secret으로 생성하여 N개의 계정마다 N개의 capa-controller-manager를 실행하면 되겠구나!

문제점

- N개의 controller-manager: 자원 낭비?
- AWS Credential이 만료된다면?
- Manager 클러스터가 (악의적인 내/외부에게) 노출되었을 때 N+1개의 AWS Credential이 유출

⇒ AWS Credential을 저장하여 사용하지 않는 방안이 필요

AWS Assume Role

*An **IAM role** is an IAM identity that you can create in your account that has specific permissions.* *<https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html>*

- 정의된 권한 범위 내 AWS API를 사용할 수 있는 "임시" 자격 증명
- IAM Role을 사용하면 권한을 사용자 계정마다 설정하지 않고 사용자 권한을 공유하거나 매번 필요한 권한을 직접 부여 불필요
  https://youtu.be/c70gIL9Znzs

- **Assume**
  - AWS Principal이 임시 Credential을 발급 받아 IAM Role에 정의된 권한 내에서 액션을 수행할 수 있음
  - 서로 다른 AWS 계정 간에도 가능: Z 계정 Role Y → A 계정 B 사용자

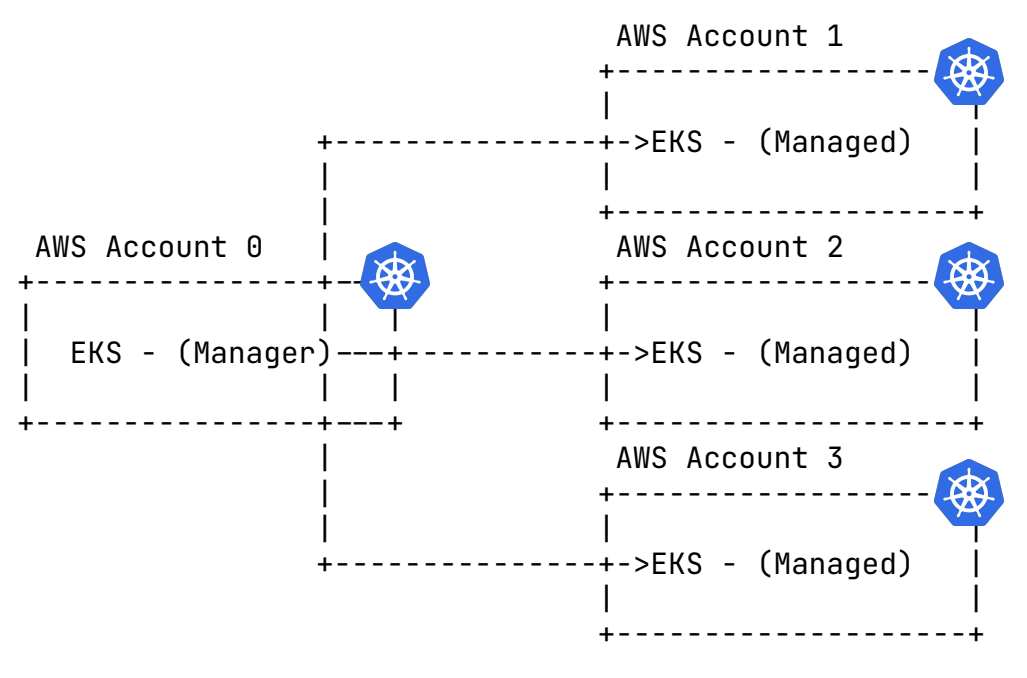
AWS IAM resources for CAPA

```
clusterawsadm bootstrap iam create-cloudformation-stack
```

CAPA 동작에 필요한 AWS IAM 자원을 CloudFormation을 통해 생성

- Role
  - Controller for capa-controller-manager
    `controllers.cluster-api-provider-aws.sigs.k8s.io` (Policy)
    - ➔ `controllers-eks.cluster-api-provider-aws.sigs.k8s.io`
    - ➔ `controllers.cluster-api-provider-aws.sigs.k8s.io`
  - Node: ControlPlane/EKSControlPlane, EKSNodegroup/Nodes
- InstanceProfile
- Policy

Cluster API, AWS에서 멀티 테넌시



capa-controller가 사용하는 **Account 0** AWS credential의 **AWS User**가 **Account 1,2,3**에 생성된 **controller AWS IAM Role**을 **Assume**하여 EKS 클러스터 생성, 관리

Cluster API, AWS에서 멀티 테넌시

CAPA는 인프라 클러스터 자원(AWSCluster, AWSManagedControlPlane) 스펙의 identityRef 필드를 통해 다른 계정 정보를 지정할 수 있음

- AWSClusterStaticIdentity: AWS Credential(AccessKeyID, SecretAccessKey)를 K8S 시크릿으로 저장하고 이를 참조
- **AWSClusterRoleIdentity**: CAPA 컨트롤러가 자신 혹은 다른 AWS 계정의 Role 권한을 위임(assume)받아 수행 → AssumeRole(AWS API) 사용

AWS IAM 준비 for multi-tenancy

Manager 클러스터의 CAPA 컨트롤러가 사용하는 AWS User

- AssumeRole 이 가능하도록 권한 추가

Managed 클러스터가 생성될 AWS 계정의 controller IAM Role

- Manager 클러스터의 CAPA 컨트롤러 AWS User에게 위임 설정
- Cluster API에서 제공하는 도구가 아닌 별도로 IAM Role을 생성하는 경우
  위임하는 계정의 Role이 CAPA 컨트롤러 동작에 필요한 모든 권한을 보유해야 함

```
{
    "Version": "2012-10-17",
    "Statement": [,
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123443211234:user/tks-capa"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Manager CAPA 컨트롤러 AWS IAM User ARN

AWSClusterRoleIdentity 설정

```
apiVersion: infrastructure.cluster.x-k8s.io/v1beta1
kind: AWSClusterRoleIdentity
metadata:
  name: customer-a-account-role
spec:
  allowedNamespaces:
    selector:
      matchLabels:
        customer: customer-a
  roleARN:
"arn:aws:iam::987698769876:role/controllers.cluster-api-provider-aws.sigs.k8s.io"
  sourceIdentityRef:
    kind: AWSClusterControllerIdentity # use the singleton for root auth
    name: default
```

```
apiVersion:
controlplane.cluster.x-k8s.io/v1beta2
kind: AWSManagedControlPlane
metadata:
  name: capa-eks
  namespace: customer-a
spec:
…
identityRef:
    kind: AWSClusterRoleIdentity
    name: customer-a-account-role
```

클러스터 접근/사용

CAPA가 자동 생성하는 kubeconfigs → K8S Secret

- [cluster-name]-user-kubeconfig: aws-iam-authenticator 이용 인증
- [cluster-name]-kubeconfig used internally by Cluster API
    - 유효 시간이 10분인 토큰

EKS 클러스터는 클러스터를 생성한 주체 (IAM User, IAM Role)만이 k8s admin으로 접근 가능하며 그 외 사용자가 접근하기 위해서는 다음의 안내 문서에 따라 설정이 필요합니다.

- AWS EKS User Guide: Cluster authentication: https://docs.aws.amazon.com/eks/latest/userguide/cluster-auth.html

SK telecom

Questions?

SK telecom

Thank you!

참고

https://cluster-api-aws.sigs.k8s.io/topics/multitenancy.html