

Cluster API & Metadata API

개요

❖ 종류

✓ 클러스터 API

- 노드
- 네임스페이스
- 영구 볼륨
- 리소스 쿼터
- 서비스 어카운트
- 룰
- 클러스터롤
- 룰바인딩
- 클러스터롤바인딩
- 네트워크정책



Cluster API

❖ Node

- ✓ 노드(Node) 리소스는 다른 리소스와 달리 기본적으로 사용자가 생성하거나 삭제하는 리소스는 아니지만 쿠버네티스에 리소스로 등록되어 있음
- ✓ 쿠버네티스를 운영하면서 자주 확인하는 리소스
- ✓ 노드 정보 상세 표시: `kubectl get nodes -o wide`

NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE
		KERNEL-VERSION		CONTAINER-RUNTIME			
master	Ready	control-plane	5d18h	v1.30.14	10.0.2.15	<none>	Ubuntu
		24.04.3 LTS		6.8.0-79-generic	containerd://1.7.28		
worker1	Ready	<none>	5d18h	v1.30.14	10.0.2.16	<none>	Ubuntu
		24.04.3 LTS		6.8.0-79-generic	containerd://1.7.28		
worker2	Ready	<none>	5d18h	v1.30.14	10.0.2.17	<none>	Ubuntu
		24.04.3 LTS		6.8.0-79-generic	containerd://1.7.28		

Cluster API

❖ Node

- ✓ 특정 노드 정보 상세 표시: `kubectl get nodes worker1 -o yaml`

apiVersion: v1

kind: Node

metadata:

annotations:

flannel.alpha.coreos.com/backend-data: '{"VNI":1,"VtepMAC":"7e:22:19:d5:c9:fb"}'

flannel.alpha.coreos.com/backend-type: vxlan



Cluster API

❖ Node

- ✓ 특정 노드 정보 상세 표시: `kubectl get nodes worker1 -o yaml`

apiVersion: v1

kind: Node

metadata:

annotations:

flannel.alpha.coreos.com/backend-data: '{"VNI":1,"VtepMAC":"7e:22:19:d5:c9:fb"}'

creationTimestamp: "2026-02-10T08:16:15Z"

labels:

beta.kubernetes.io/arch: arm64

beta.kubernetes.io/os: linux

kubernetes.io/arch: arm64

kubernetes.io/hostname: worker1

kubernetes.io/os: linux

name: worker1

spec:

podCIDR: 10.244.2.0/24

podCIDRs:

- 10.244.2.0/24

status:

addresses:

- address: 10.0.2.16

type: InternalIP

- address: worker1

Cluster API

❖ Node

- ✓ 특정 노드 정보 상세 표시: `kubectl describe node worker1`

Non-terminated Pods: (3 in total)

Namespace	Name	CPU Requests	CPU Limits	Memory
Requests	Memory Limits	Age		
-----	----	-----	-----	-----
default	sample-hostpath	0 (0%)	0 (0%)	0
(0%) 50m				
kube-flannel	kube-flannel-ds-7vnfd	100m (10%)	0 (0%)	50Mi (2%)
0 (0%) 5d18h				
kube-system	kube-proxy-tghff	0 (0%)	0 (0%)	0 (0%)
(0%) 5d18h				0

Cluster API

❖ Namespace

✓ 개요

- 쿠버네티스에는 네임스페이스라는 가상의 쿠버네티스클러스터 분리 기능이 있음
- 초기 상태에서 default/kube-system/kube-public/kube-node-lease와 같이 네 가지 네임스페이스가 생성되며 직접 생성도 가능
- 네임스페이스는 리소스의 쿼터를 설정하는 리소스 쿼터나 인증을 수행하는 RBAC (Role Based Access Control) 에서도 설정 범위를 지정할 때 사용할 수 있고 다양한 쿠버네티스 주변 에코시스템을 도입할 때도 시스템마다 네임스페이스를 분리하여 생성하는 방법을 많이 사용



Cluster API

❖ Namespace

✓ 생성

- 매니페스트를 이용한 생성
 - ◆ 매니페스트 파일: sample-namespace.yaml

```
apiVersion: v1
kind: Namespace
metadata:
  name: sample-namespace
```
 - ◆ 리소스 생성: `kubectl apply -f sample-namespace.yaml`
- 명령어로 생성: `kubectl create namespace sample-namespace`



Cluster API

❖ Namespace

✓ 리소스 획득

- 네임스페이스를 지정하여 리소스를 가져오는 경우 `-n` 또는 `-namespace` 옵션을 사용
- `sample-namespace` 네임스페이스의 파드 목록을 가져옴: `kubectl get pods -n sample-namespace`
- 모든 네임스페이스의 파드 목록을 가져옴: `kubectl get pods --all-namespaces` 또는 `kubectl get pods -A`

