



(2) Master nodes (Kube-API Server, ETCD, Controller Manager, Scheduler)

▼ 목차

♥ Kube-API Server ♥

♥ ETCD ♥

ETCD란?

ETCD 설치

쿠버네티스에서 ETCD의 역할

쿠버네티스 배포의 두가지 방법

ETCD - Commands (Optional)

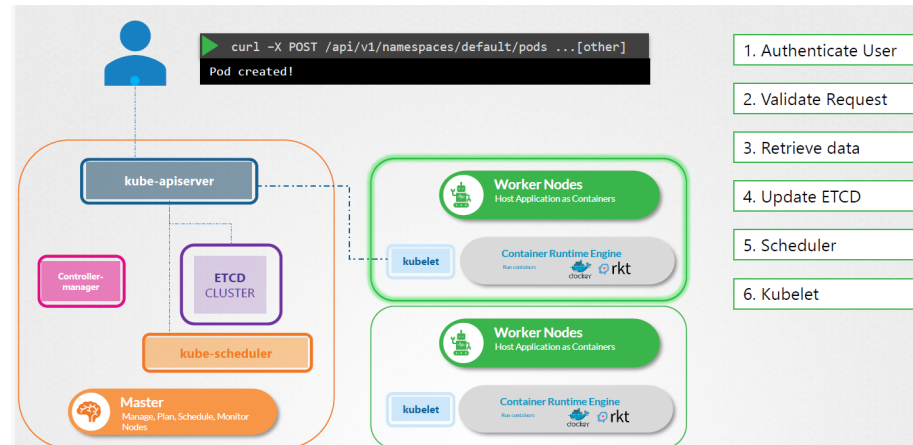
♥ Kube Controller Manager ♥

♥ Kube Scheduler ♥

♥ Kube-API Server ♥

- 주요 관리 구성 요소
- 기타 데이터 저장소와 직접 상호작용하는 유일한 구성요소임!
 - 다른 구성요소(scheduler, kube-controller-manager, kubelet 등)들은 API server를 통해 각 영역의 클러스터에서 업데이트 함

- 쿠버네티스 동작 과정



- kubectll 명령어~

1. **Authenticate User**

: API server가 요청 인증

2. **Validate Request**

: API server가 유효성 검증

3. **Retrieve Data**

: ETCD 클러스터가 데이터 회수, 요청된 정보로 API server에 응답

4. **Update ETCD**

: 변경된 클러스터 정보 업데이트

5. **Scheduler**

: 지속적으로 API 서버 모니터링

→ 올바른 노드 식별, API server에 전달

→ API server는 이를 ETCD 클러스터에 전달, 업데이트함

6. **Kubelet**

: API 서버가 변경 정보를 해당하는 worker node의 kubelet에 전달함

→ kubelet은 노드에 pod 생성, 컨테이너 런타임 엔진에 지시, 앱 이미지 배포

→ 완료 후, kubelet이 현재 상태를 API server로 전달

→ API server는 관련 데이터 업데이트



ETCD란?

: 분산되고, 신뢰할 수 있는 **key-value store**

- 간단, 안전, 빠르다는 특성을 가짐
- **key-value store?**
 - 전통적인 테이블 형식의 데이터베이스와 달리, 문서/페이지의 형태로 정보 저장
 - 어떤 형식/구조로 될 수 있음
 - 한 파일의 변화가 다른 파일에 영향을 주지 X
 - 데이터가 복잡해지면 JSON/YAML 같은 데이터 포맷을 활용함

ETCD 설치

1. Download Binaries

```
curl -L https://github.com/etcd-io/etcd/releases/download/v3.3.11/etcd-v3.3.11-linux-amd64.tar.gz -O etcd-v3.3.11-linux-amd64.tar.gz
```

2. Extract

```
tar xzvf etcd-v3.3.11-linux-amd64.tar.gz
```

3. Run ETCD Service

```
./etcd
```

- ETCD의 CLI 클라이언트: **etcdctl**

- **key-value** 쌍 저장, 회수 가능

```
# 명령어 옵션 확인
./etcdctl
# 버전 확인
./etcdctl --version
```

- version
 - 2 → 3으로 넘어가며 API 버전 변경됨 (etcdctl 명령어가 바뀜)

- API 버전을 3으로 설정하기 위해서는 ETCDCTL_API 환경변수 설정 필요

```
# 방법1) API 버전을 3으로 변경
ETCDCTL_API=3 ./etcdctl version

# 방법2) 전체 세션에 내보내기
export ETCDCTL_API=3
./etcdctl version
```

- API ver.2

```
# 데이터 저장 (key-value)
./etcdctl set key1 value1
# 저장된 데이터 검색
./etcdctl get key1
```

- API ver.3

```
# 데이터 저장 (key-value)
./etcdctl put key1 value1
# 저장된 데이터 검색
./etcdctl get key1
```

쿠버네티스에서 ETCD의 역할

1. 클러스터에 관한 정보 저장

- Nodes, Pods, Configs, Secrets, Accounts, Roles, Bindings 등

2. 클러스터에 변화를 줄 때마다 ETCD가 업데이트 됨

(ETCD에 저장되고 나서야 비로소 클러스터 변화가 완료된 것으로 간주됨)

- ETCD는 클러스터 설정에 따라 배포함

쿠버네티스 배포의 두가지 방법

→ 두 방법의 차이점을 아는 것이 좋다~

1. 직접 배포

- 직접 binaries를 다운로드해 배포함
- 마스터 노드에 직접 binaries 설치, 서비스로 구성함

2. kubeadm 툴 사용

- kubeadm이 기타 서버를 설치함

```
# 쿠버네티스에서 제공하는 etcdctl 명령어를 확인
kubect1 exec etcd-master -n kube-system etcdctl get / --prefix -keys-only
# pods 확인
kubect1 get pods -n kube-system
```

▼ ETCD - Commands (Optional)

(Optional) Additional information about ETCDCTL Utility

ETCDCTL is the CLI tool used to interact with ETCD.

ETCDCTL can interact with ETCD Server using 2 API versions - Version 2 and Version 3. By default its set to use Version 2. Each version has different sets of commands.

For example ETCDCTL version 2 supports the following commands:

```
1. etcdctl backup
2. etcdctl cluster-health
3. etcdctl mk
4. etcdctl mkdir
5. etcdctl set
```

Whereas the commands are different in version 3

```
1. etcdctl snapshot save
2. etcdctl endpoint health
3. etcdctl get
4. etcdctl put
```

To set the right version of API set the environment variable ETCDCTL_API command

```
export ETCDCTL_API=3
```

When API version is not set, it is assumed to be set to version 2. And version 3 commands listed above don't work. When API version is set to version 3, version 2 commands listed above don't work.

Apart from that, you must also specify path to certificate files so that ETCDCTL can authenticate to the ETCD API Server. The certificate files are available in the etcd-master at the following path. We discuss more about certificates in the security section of this course. So don't worry if this looks complex:

```
--cacert /etc/kubernetes/pki/etcd/ca.crt
--cert /etc/kubernetes/pki/etcd/server.crt
--key /etc/kubernetes/pki/etcd/server.key
```

So for the commands I showed in the previous video to work you must specify the ETCDCTL API version and path to certificate files. Below is the final form:

```
kubectl exec etcd-master -n kube-system -- sh -c "ETCDCTL_API=3 etcdctl get / --prefix --keys-only --limit=10 --cacert /etc/kubernetes/pki/etcd/ca.crt --cert /etc/kubernetes/pki/etcd/server.crt --key /etc/kubernetes/pki/etcd/ser
```

💛Kube Controller Manager💛

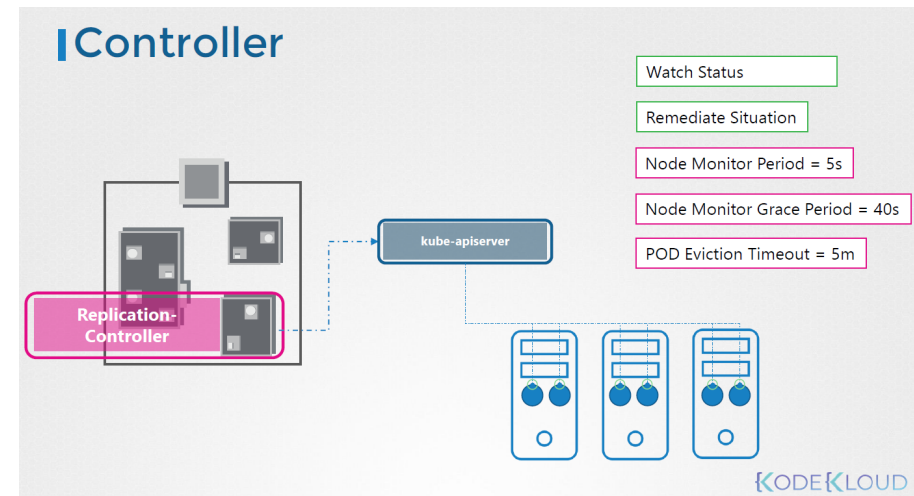
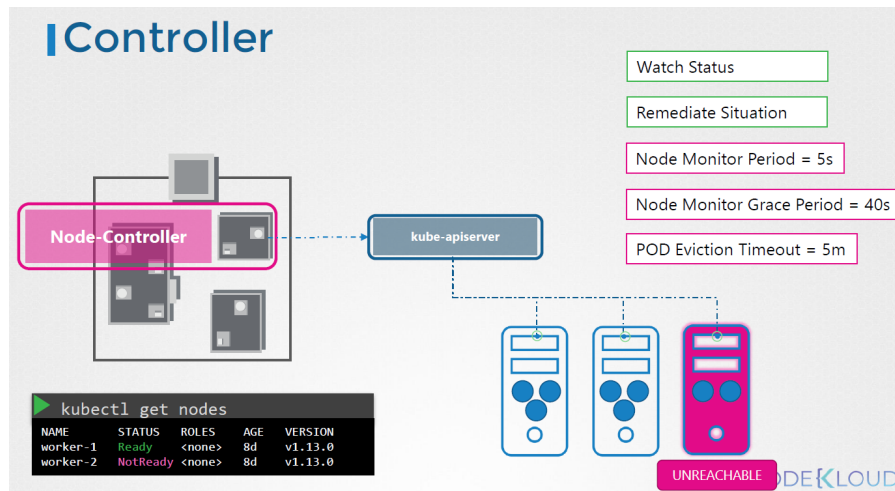
- 쿠버네티스의 다양한 컨트롤러 관리
 1. 상태 모니터링
 2. 상황 재조정
- 쿠버네티스에서의 **Controller**
 - 시스템 내 다양한 구성 요소의 상태를 지속적으로 모니터링, 시스템 전체를 원하는 기능 상태로 만드는 process
- 종류

Node Controller

- Kube API server를 통해 노드의 상태를 모니터링하고, 응용 프로그램이 계속 실행되도록 조치를 취함
 - 5초마다 노드의 상태를 확인함
 - 40초 이후에도 신호가 돌아오지 않는다면, unreachable로 표시
 - 상태가 다시 돌아오는 데 5초가 걸림
 - 돌아오지 않는다면, 해당 pod 제거, ReplicaSet을 할당

Replication Controller

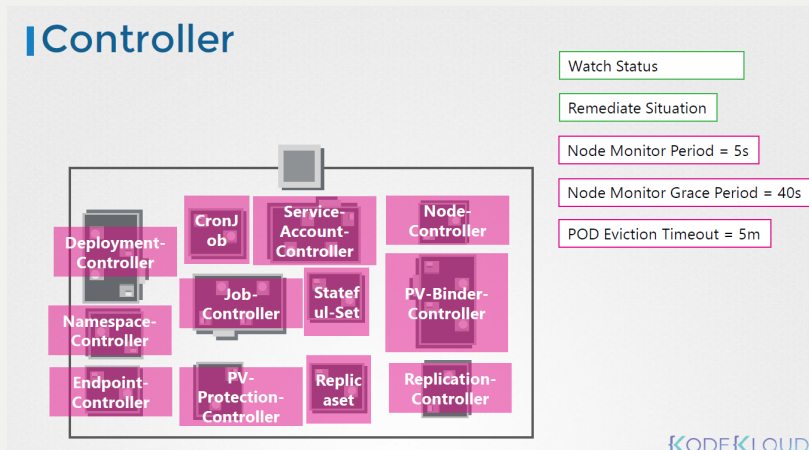
- ReplicaSet의 상태를 모니터링하고, 원하는 수의 pod가 항상 사용 가능하도록 유지함
 - 한 pod가 죽으면, 다른 pod를 새로 생성함





쿠버네티스 내에는 이외에도 **다양한 controller**가 존재함

⇒ 이들을 하나로 묶어 **Kube Controller Manager**라는 프로세스로 패키징되어 있음



♥ Kube Scheduler ♥

- node pod의 일정 관리를 책임짐
- 특정 기준에 따라 어떤 pod이 어떤 node에 들어갈지 결정만 함
(pod을 직접 node에 두는 것 X → 이걸 kubelet이 하는 일!)
- 각 pod을 보고, pod를 놓을 최적의 node를 찾고자 함
 - 다음과 같은 2단계를 거쳐 최적의 node를 찾아냄
 1. 노드를 필터링함
: pod에 맞지 않는 노드를 걸러냄

2. 노드의 순위를 매김

: 우선순위 함수를 이용해, 0부터 10까지의 점수로 노드에 점수를 매겨, 최종적으로 최적의 노드를 선택함

- 선정 기준?

- CPU, 메모리 요구사항, 리소스 요구사항, 제한 변수, taints and toleration, 노드 선택기, 선호도 규칙 등

