k8s - 쿠버네티스 아키텍처

k8s는 어떤 사상으로 움직이는지 알아보자

쿠버네티스 특징

1. **선언적**

   → 상태 확인 → 현재 상태가 설정 상태가 아니네? →

   선언된 상태로 변환 (관리비용 감소)

   but, 단순한 내부동작 제어 쉽지않음…(kubectl로 제어 기능 추가됨)

   → 상태복구 (클러스터가 동작 중이라면 원 상태로 자동복구)

   → 단일장애점이 없음

2. **워크로드 분리**

   운영체제처럼 프로세스의 관리를 추상화 해주는 레이어

3. **어디서나 설치 및 사용 가능, 성숙한 커뮤니티**

🧭 ※쿠버네티스의 원리를 알아보자(간단히)

공식문서에서는?

쿠버네티스가 아닌것도 알아야됩니다.

MSA환경과 유사한 환경에서 작업하시는 분 들은 작업 환경이 모두 쿠버네티스 어플리케이션 안에서 이루어지는 줄 알고있습니다. (ex. 대표적으로 이충호)

하드웨어보단 컨테이너 수준에서 운영되기 때문에 쿠버네티스 밖에서 운영되는 모놀리틱 솔루션을 선택해 추가하는 형태로도 많이 운영되고있습니다.

대표적인 모놀리틱 솔루션

1. CI/CD 툴
2. Middle Ware
3. DB

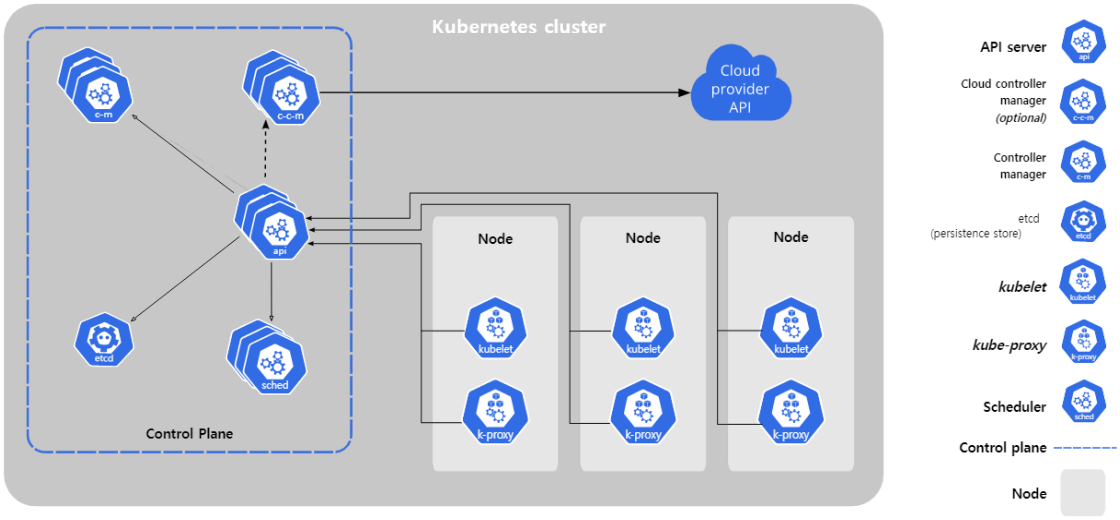
4. Cash / Cluster Storage

등은 쿠버네티스 환경에서 실행되는 추가적인 솔루션이라고 생각하면 됩니다.

단지, 정의된 워크플로우에 따라 입력된 상태를 유지하게 됩니다.

클러스터의 전체 구조

https://hoing.io/archives/131 및
https://kubernetes.io/ko/docs/concepts/overview/components/를 참조하였습니다.

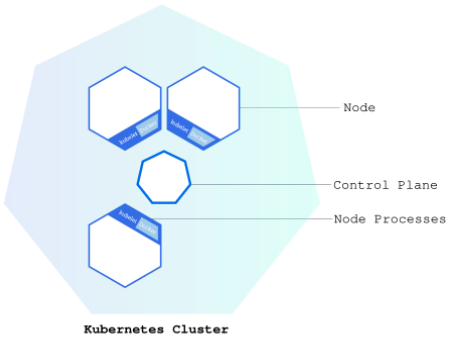


쿠버네티스를 배포하면

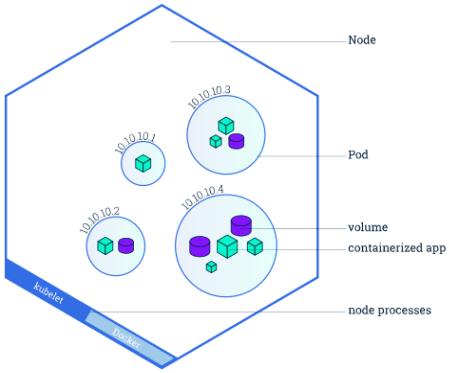
→ **Cluster**를 얻게됩니다.

→ 클러스터는 한개 이상의 **Worker Node**(컨테이너 애플리케이션)를 얻게 됩니다.

→ 워커 노드는 구성요소인 **Pod**를 호스트합니다.



Cluster 다이어그램



Node 및 Pod 다이어그램

Cluster

- 그룹단위 입니다. 90가지 예제로 배우는 예제에서는 Master 3, Worker 2로 구성될 예정입니다.

Controll Plane ≒ Master

- **Worker Node + 클러스터 내부 Pod**를 관리합니다.
- 컨트롤 플레인이 여러 컴퓨터에 걸쳐 실행되고, 일반적으로 여러 노드를 실행함으로 **내결함성**과 **고가용성**이 제공된다고 합니다.
- 실질적으로는 1개 운영되며, 마스터에 문제가 생기면, 다른 워커 노드와 달리 대신할 수 있는 대체자가 없습니다, 대신, 예비노드에서 관리 및 제어로 대체됩니다.

Worker ≒ Node

- Master의 명령을 수행하는 서버입니다. 실제 실행과 중지, 유지등 역할을 수행합니다.
- k8s에서는 워커머신 이지만, 클러스터에 따라 VM or physical Machine이 될 수 있다고 하네요(설치되는 환경에 따라 다른 듯 합니다.)
- Master에 의해 Pod를 관리합니다.

Pod

- Node에 배포된 1..N(일반적으로 1개)의 컨테이너 그룹입니다. 컨테이너를 묶음단위로 관리 가능하게 해줍니다.