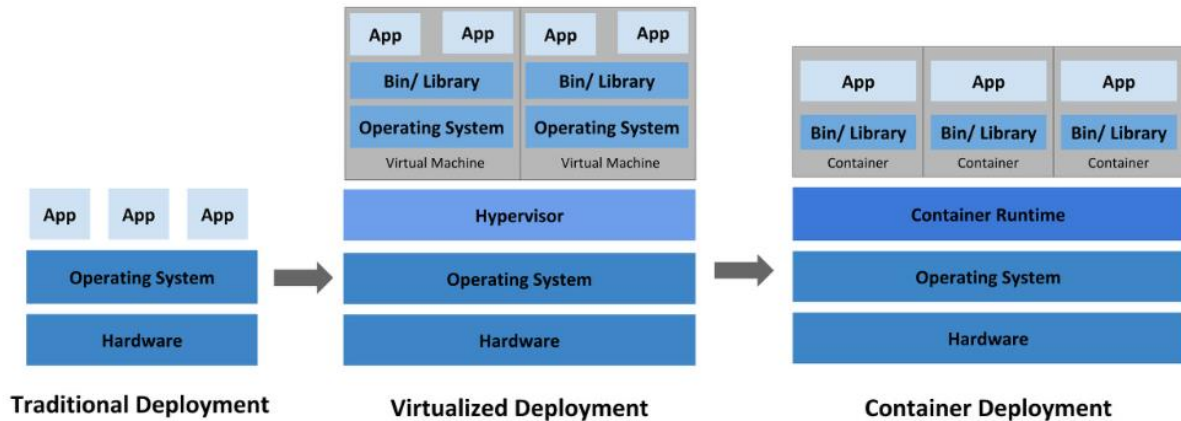


< Kubernetes Architecture >

쿠버네티스는 애플리케이션 간에 OS 를 공유해 VM 을 사용하는 것보다 가볍고, 또한 추상화 레벨이 높아 리소스 관리가 효율적입니다.



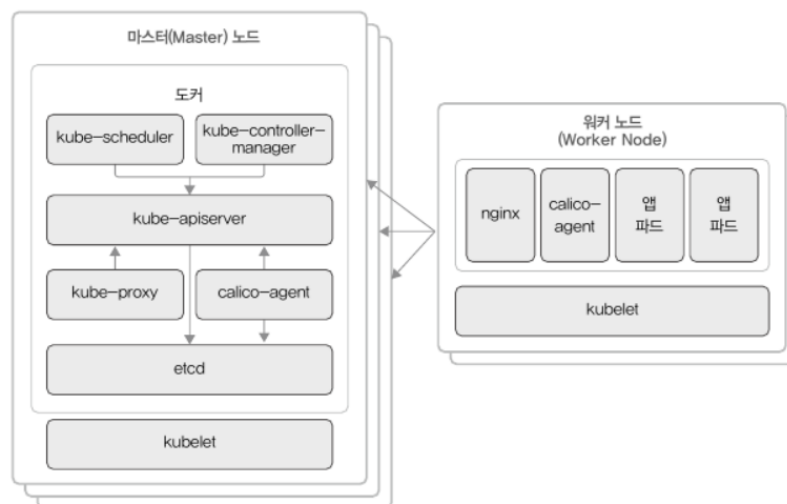
1. 쿠버네티스가 제공하는 것

- **서비스 디스커버리와 로드 밸런싱** : Service Discovery는 컨테이너 내부에서 실행되는 서비스의 생성을 자동으로 감지하고 등록하여 프로세스를 재 시작하게 하는 기능입니다. 이를 통해 관리자가 별도로 설정을 해 줄 필요 없이 외부에서 바로 서비스에 접근 할 수 있게 됩니다. DNS 이름이나 IP 주소를 이용해 컨테이너를 노출 할 수 있습니다. 트래픽 관리를 위한 로드 밸런싱도 제공합니다.
- **스토리지 오케스트레이션** : 로컬 저장소/공용 클라우드 공급자 등이 원하는 저장소 시스템을 자동으로 탑재할 수 있습니다.
- **자동화된 롤아웃과 롤백** : 이미지의 배포 버전이 업, 다운됨에 따라 Rollout, Rollback 을 자동적, 그리고 일괄적으로 처리하여 버전 관리를 수행 할 수 있습니다. 버전이 업그레이드 되면 일괄적 Rollout 을, 만약 새 버전에 문제가 발생해서 복구해야 한다면 이전 버전으로 Rollback 합니다.
- **자동화된 빈 패킹(bin packing)** : 컨테이너 작업을 사용할 수 있도록 쿠버네티스는 클러스터 노드를 제공한다. 컨테이너에 필요한 리소스 (CPU, RAM 등)를 쿠버네티스에 지시하면 노드에 맞추어 리소스를 잘 활용할 수 있도록 설정해 줄 수 있습니다.

- **자동화된 복구(self-healing)** : 실패한 컨테이너를 다시 시작하고, 컨테이너를 교체합니다.
- **시크릿과 구성 관리** : 암호, OAuth 토큰/SSH 와 같은 중요한 정보를 저장하고 관리 할 수 있습니다.

2. 쿠버네티스 컴포넌트

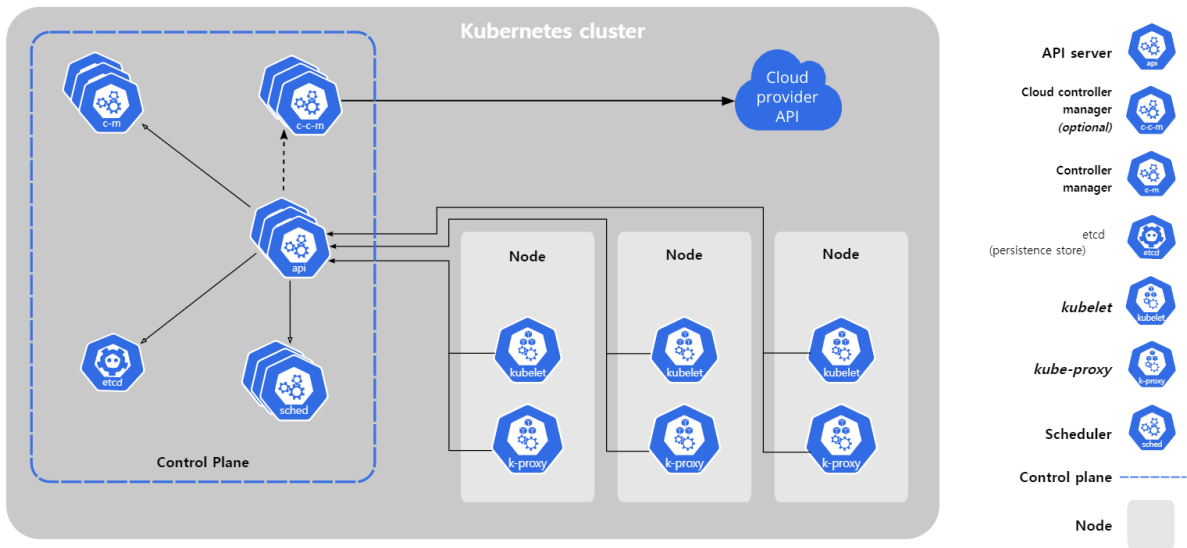
쿠버네티스 클러스터는 컨테이너화 된 애플리케이션을 실행하는 노드라고 하는 워커 머신의 집합입니다. 모든 클러스터는 최소 한 개의 워커 노드를 가집니다.



워커 노드는 애플리케이션의 구성요소인 파드를 호스트합니다. 컨트롤 플레인은 워커 노드와 클러스터 내 파드를 관리합니다. 프로덕션 환경에서는 일반적으로 컨트롤 플레인이 여러 컴퓨터에 걸쳐 실행되고, 클러스터는 일반적으로 여러 노드를 실행하므로 내결함성과 고가용성이 제공됩니다.

쿠버네티스 클러스터 밖에서 외부 통신하는 것을 아웃클러스터 라고 합니다. 쿠버네티스 클러스터 밖에서 kubeconfig 파일을 들고 kubectl 명령어로 통신합니다.

쿠버네티스에 내장되어있는 kubeconfig를 기반으로 컴포넌트 끼리 통신하는 것을 인클러스터 라고 합니다.



3. 컨트롤 플레인 컴포넌트

컨트롤 플레인 컴포넌트는 클러스터에 관한 전반적인 결정(예를 들어, 스케줄링)을 수행하고 클러스터 이벤트(예를 들어, 디플로이먼트의 replicas 필드에 대한 요구 조건이 충족되지 않을 경우 새로운 파드를 구동 시키는 것)를 감지하고 반응합니다.

컨트롤 플레인 컴포넌트는 클러스터 내 어떠한 머신에서든지 동작할 수 있습니다. 그러나 간결성을 위하여, 구성 스크립트는 보통 동일 머신 상에 모든 컨트롤 플레인 컴포넌트를 구동 시키고, 사용자 컨테이너는 해당 머신 상에 동작 시키지 않습니다. 만약 kubelet 이 마스터 노드에 존재한다면 master 겸 worker 노드로 사용되는 것 입니다.

- **kube-apiserver** : API 서버는 쿠버네티스 API 를 노출하는 쿠버네티스 컨트롤 플레인 컴포넌트
- **etcd** : 모든 클러스터 데이터를 담는 쿠버네티스 뒷단의 저장소로 사용되는 일관성/고가용성 키-값 저장소
- **kube-scheduler** : 노드가 배정되지 않은 새로 생성된 파드를 감지, 실행할 노드를 선택
- **kube-controller-manager** : 컨트롤러 프로세스를 실행. (노드, 레플리케이션, 엔드포인트, 서비스&토큰)
- **cloud-controller-manager** : 클라우드변 컨트롤 로직을 포함

4. 노드 컴포넌트

노드 컴포넌트는 동작 중인 파드를 유지시키고 쿠버네티스 런타임 환경을 제공하며, 모든 노드 상에서 동작합니다.

- **kubelet** : 클러스터의 각 노드에서 실행되는 에이전트
- **kube-proxy** : 클러스터의 각 노드에서 실행되는 네트워크 프록시로, 쿠버네티스의 서비스 개념의 구현부
- **컨테이너 런타임** : 컨테이너 실행을 담당하는 소프트웨어(Docker, containerd, CRI-O, Kubernetes CRI)

5. 애드온

애드온은 쿠버네티스 리소스를 이용해 클러스터 기능을 구현합니다. 이들은 클러스터 단위의 기능을 제공하기 때문에 애드온에 대한 네임스페이스 리소스는 kube-system 네임스페이스에 속합니다. (아래는 일부 애드온의 예시)

- 네트워크 정책: Calico, Canal, Flannel
- 서비스 검색: CoreDns(클러스터 내부에서 동작하는 DNS Server)

Reference

1. 쿠버네티스 기능 - <https://kubernetes.io/ko/docs/concepts/overview/what-is-kubernetes/>
2. 쿠버네티스 사용법 - <https://www.redhat.com/ko/topics/containers/what-is-kubernetes>
3. 쿠버네티스 아키텍처 - <https://kubernetes.io/ko/docs/concepts/architecture/>

4. 애드온 - <https://kubernetes.io/ko/docs/concepts/cluster-administration/addons/>

5. 쿠버네티스 컴포넌트 - <https://koreanfoodie.me/650>