쿠버네티스는 컨테이너를 pod 이라는 개념으로 감싸서 씀

개발자 입장에서 짜는 프로그램을 만든다 하면 if 문을 써서 어떤 노드가 있고, 빈 공간이 있는지 체크해서 cpu 메모리가 적은 곳에 할당한다 이럴텐데

스토리

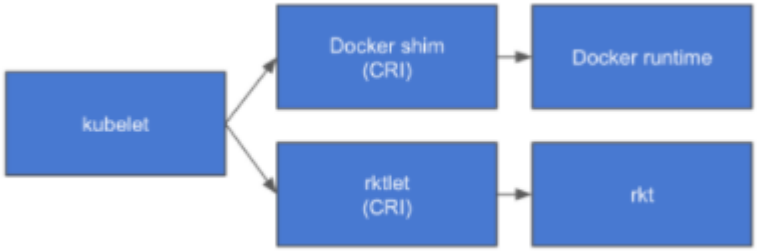
쿠버네티스는 pod 을 생성 ⇒ 생성 요청된 것을 감시하다가 있으면 할당 ⇒ 실제 노드 서버에 컨테이너를 띄우는 할당이 아니라 이 pod 은 3 번 노드에 띄우겠다 정보만 올림 ⇒ 할당은 되었지만 실행은 안됨 ⇒ 3 번 노드 입장에서 자신에게 할당되었지만 실행 안된거가 있는지 확인 ⇒ 있다면 도커로 띄우고 ⇒ Pod 의 상태를 알려주고 ⇒ 아 이제 다 할당이 되었구나! 이런 흐름

도커심 모듈은 쿠버네티스에서 도커를 지원하기 위한 '컨테이너 런타임 인터페이스(CRI)'다.

도커는 컨테이너 내의 자원할당 관련

쿠버네티스는 노드에 파드를 만들고 그 pod안에 컨테이너가 떠있음

그러므로 쿠버네티스가 파드에게 자원을 할당해주면 그 안에서 컨테이너가 돌아가면서 도커가 거기 안에 한정된 자원으로 알아서 할당하는 것이라 생각



Pod

쿠버네티스에서 생성하고 관리할 수 있는 배포 가능한 가장 작은 컴퓨팅 단위

하나 이상의 컨테이너 그룹 -> 같은 pod에 속해 있으면 컨테이너끼리 네트워크 및 스토리지 공유, 항상 함께 배치 및 스케쥴된다.

Pod는 Deployment 또는 job과 같은 워크로드 리소스를 이용해 생성

Pod 상태 추적은 statefulset 리소스를 고려한다.

각 파드는 특정 애플리케이션의 단일 인스턴스를 실행하기 위한 것이다. 더 많은 인스턴스를 실행하여 더 많은 전체 리소스를 제공하기 위해 애플리케이션을 수평적으로 확장하려면, 각 인스턴스에 하나씩, 여러 파드를 사용해야 한다. 쿠버네티스에서는 이를 일반적으로 *레플리케이션* 이라고 한다. 복제된 파드는 일반적으로 워크로드 리소스와 해당 컨트롤러에 의해 그룹으로 생성되고 관리된다.

- ➔ HPA가 연관있는 파트
- ➔ 1 pod 1 컨테이너가 흔히 사용되는 케이스, 1 pod N 컨테이너도 있긴 하다(서비스단위).
- ➔ 어플리케이션의 단일 인스턴스를 실해하기 위해 주로 하나의 pod만 배정함
- ➔ 어플리케이션의 리소스를 확장시켜주고 싶다면 pod를 복제해서 늘린다.
- ➔ 이게 HPA

파드의 컨테이너는 클러스터의 동일한 물리 또는 가상 머신에서 자동으로 같은 위치에 배치되고 함께 스케줄된다. 컨테이너는 리소스와 의존성을 공유하고, 서로 통신하고, 종료 시기와 방법을 조정할 수 있다.

일부 파드에는 앱 컨테이너 뿐만 아니라 초기화 컨테이너를 갖고 있다. 초기화 컨테이너는 앱 컨테이너가 시작되기 전에 실행되고 완료된다.

파드는 기본적으로 파드에 속한 컨테이너에 네트워킹과 스토리지라는 두 가지 종류의 공유 리소스를 제공한다

파드의 생성 및 관리는 컨트롤러에 의해 이루어진다.

디플로이먼트 – 클러스터에서 복제된 어플리케이션을 관리

스테이트풀셋 – 내구성이 있는 스토리지와 파드별로 지속성 식별자를 사용해서 파드 집합의 디플

로이먼트와 스케일링 관리

데몬셋 – 파드의 복제본을 클러스터 노드 집합에서 동작하게 한다.

Pod는 pod에 속한 컨테이너 간의 데이터 공유와 통신을 지원한다.

- Pod의 모든 컨테이너는 공유 볼륨에 접근 가능 즉 데이터를 공유할 수 있다.
- 파드의 컨테이너들은 IP주소, 네트워크 포트 공유

-