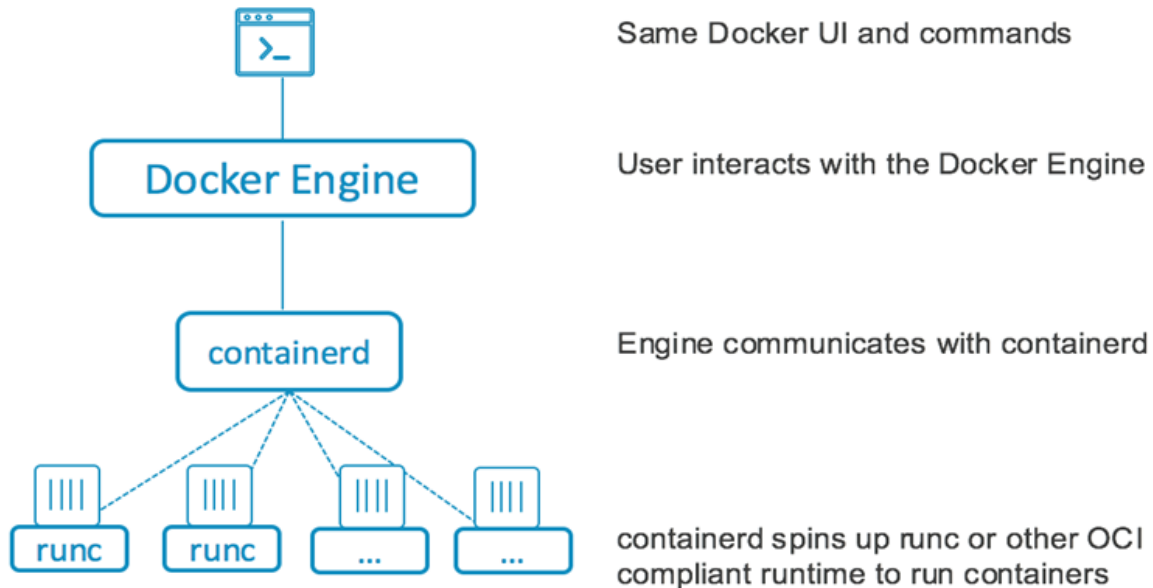


Docker 개요



Docker의 시작

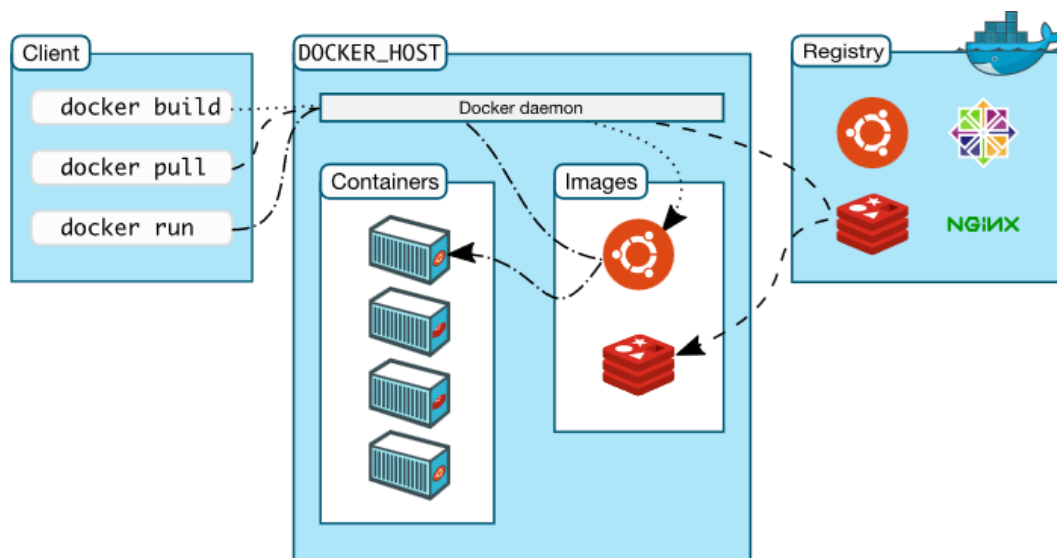
- 2013년 Docker사에서 Go 언어로 개발한 오픈소스 고수준 컨테이너 런타임
- 저수준 컨테이너 런타임을 제어하기 위한 RESTful API와 CLI를 제공
- 저수준 컨테이너 런타임의 변경
 - 초기에는 저수준 컨테이너 런타임으로 LXC를 사용
 - 0.9버전 부터 컨테이너 구동을 위한 라이브러리를 libcontainer로 대체
 - libcontainer는 이후 OCI(Open Container Initiative)에 기부되어 runC로 이름이 변경

Docker 장점

- 격리된 환경 제공

- Docker는 이미지(image)를 사용하여 컨테이너를 구동
- 컨테이너는 격리된 환경으로 하나의 애플리케이션을 실행하도록 구동
- 서버를 코드 형태로 정의 (IaC)
 - Docker 컨테이너에서 사용되는 이미지를 코드 형태로 정의하여 작성 가능(Dockerfile)
 - 이미지를 코드 형태로 관리할 경우 일관성을 유지할 수 있으며, 버전 관리에 용이함
- 이식성
 - Docker 컨테이너는 어떤 Docker 시스템에서도 동일하게 동작함
 - 개발 – 테스트 – 운영의 각 단계의 환경이 다르더라도 동일하게 실행
- 상호운용성
 - Google, Amazon, IBM, Microsoft, RedHat 등 다양한 벤더의 지원
다양한 오픈소스와 연계하여 사용

Docker의 기본 구조



- Client
 - docker 명령을 사용하여 Docker 데몬(dockerd)과 상호작용하는 구성요소

- Host
 - dockerd을 구동하며, Client로부터 명령을 전달받아 수행
 - 실제 컨테이너 구동위치
 - docker를 사용하여 구동한 컨테이너의 이미지를 저장
- Registry
 - Docker 이미지를 저장하고 있는 구성요소
 - 인터넷이 연결되어 있을 경우 기본 Registry로 Docker Hub 사용가능