

클라우드 도커

Cloud Docker



DOCKER

DOCKER

◆ 컨테이너 기술

- OS와 독립된 격리된 환경을 제공하는 운영체제 수준의 가상화 기술
 - 리눅스 커널을 공유하면서 프로세스를 격리된 환경에서 실행하는 기술
 - Jail 명칭에서 Container로 변경
-
- 2000년 처음 등장한 FreeBSD Jail
 - FreeBSD 시스템을 여러 하위 시스템 또는 Jail로 분할할 수 있도록 하는 기술
 - Jail은 시스템 관리자가 조직 내외부의 여러 사용자와 공유할 수 있는 안전한 환경으로 개발
 - 2001년, Jacques Gélinas의 VServer 프로젝트 통해 Linux에 격리된 환경 구현
 - 2013년 Kernel 3.8부터 Container 사용 가능해짐

DOCKER

◆ 컨테이너 기술 특징

- 운영체제 수준의 가상화

별도의 하드웨어 에뮬레이션 없이 리눅스 커널 공유해 컨테이너를 실행
게스트OS 관리가 필요하지 않음

- 빠른 속도와 효율성

하드웨어 에뮬레이션이 없기 때문에 컨테이너는 아주 빠르게 실행
프로세스 격리 위해 아주 약간 오버헤드 있지만 일반적인 프로세스 실행하는 것과 거의 차이 없음
또한 하나의 머신에서 프로세스만큼 많이 실행하는 것이 가능

DOCKER

◆ 컨테이너 기술 특징

- 높은 이식성 **portability**:

모든 컨테이너는 독자적인 실행 환경 가지며 환경은 파일들로 구성되며, 이미지 형식으로 공유
리눅스 커널 + 같은 컨테이너 런타임 사용할 경우 → 컨테이너 실행 환경 공유 / 손쉽게 재현 가능

- 상태를 가지지 않음 **stateless**:

컨테이너가 실행되는 환경은 독립적이기 때문에, 다른 컨테이너에게 영향을 주지 않음
도커와 같이 이미지 기반으로 컨테이너를 실행하는 경우 특정 실행 환경을 쉽게 재사용 가능

DOCKER

◆ 컨테이너 기술 종류

- 시스템 컨테이너

컨테이너 기술들 사용해 **운영체제 위에 하드웨어 가상화 없이 운영체제 실행하는 컨테이너**
일반적인 리눅스처럼 init 프로세스 등을 사용해서 **다수의 프로세스가 같은 환경 공유하는 것 목표**
대표: LXC(Linux Containers), LXD(Linux Container Daemon)

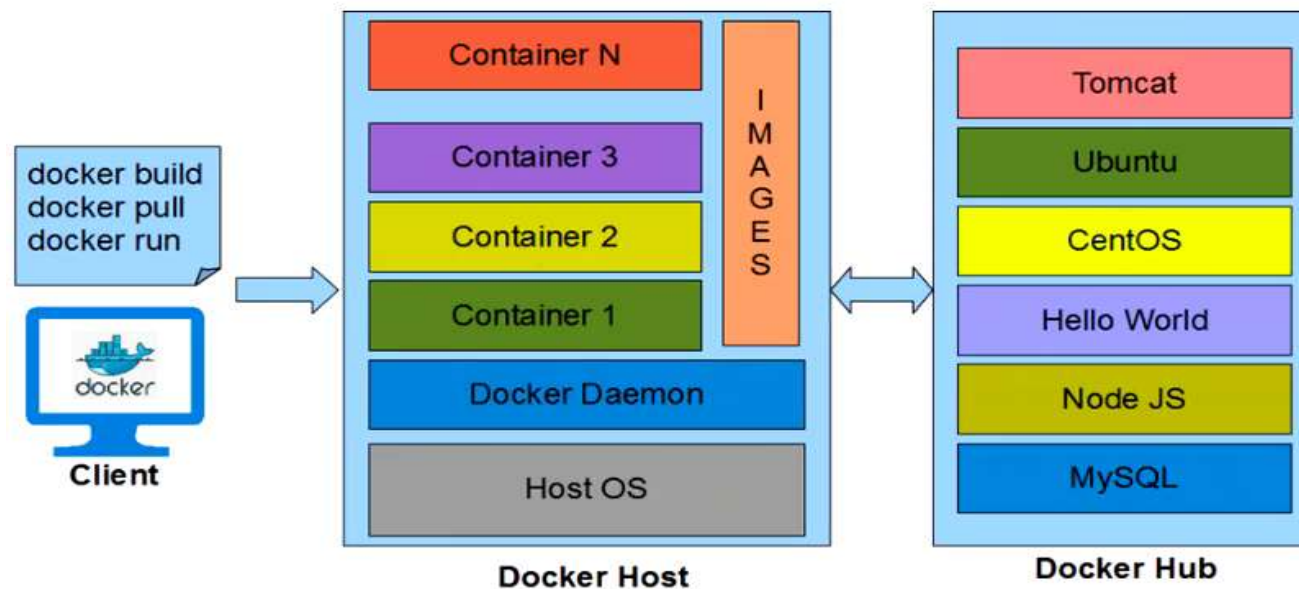
- 애플리케이션 컨테이너

컨테이너 기술 활용해 **하나의 애플리케이션(프로세스) 실행하는 것 목표**
독립적인 환경을 가진다는 점에서는 시스템 컨테이너와 동일
단 하나의 프로세스만 실행한다는 점에서 확장이 쉽고 관리 요소가 거의
대표: Docker

DOCKER

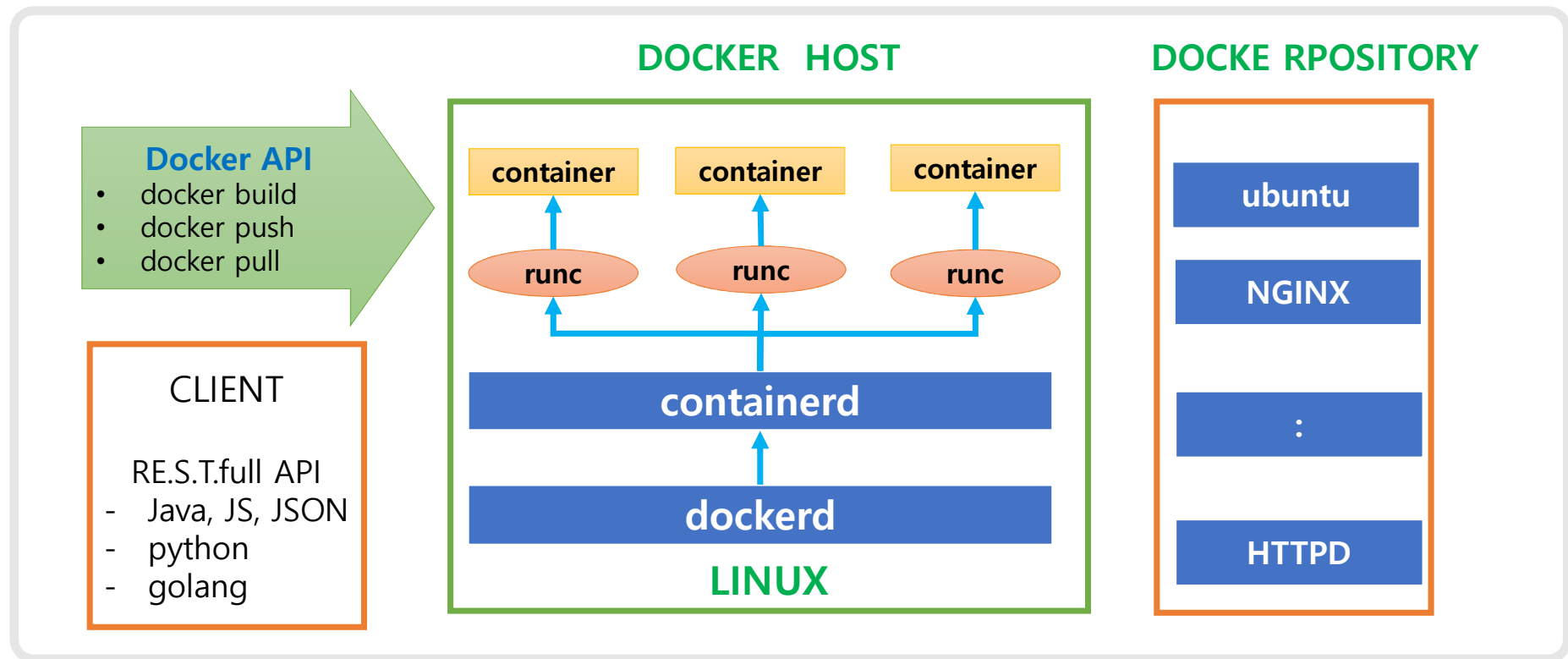
◆ Docker 란

- Container 기반으로 애플리케이션을 배포하는 소프트웨어/플랫폼



DOCKER

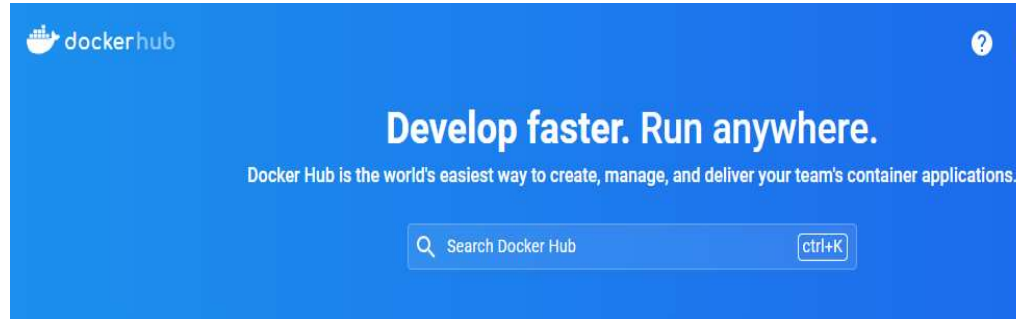
◆ Docker 구성



DOCKER

◆ Docker Hub

- 도커 회사에서 제공하는 Docker Repository/Registry
- 접속 UI : <https://hub.docker.com/>



REPOSITORY

ubuntu

NGINX

:

HTTPD

DOCKER

◆ Docker Image

- 필요한 여러 가지 파일들을 패키징한 파일
 - **읽기 전용**으로 스냅샷이라고도 하며, 특정 시점의 애플리케이션과 가상환경 의미
 - 소스 코드, 라이브러리, 종속성, 도구, 응용 프로그램 등 실행 위한 파일 포함하는 불변 파일
 - **Dockerfile 빌드로 생성**
-
- **시작 또는 실행 안됨** → 컨테이너 통해 실행!
 - **도커 이미지 1개** → 여러 개 **도커 컨테이너** 생성 가능

DOCKER

◆ Docker Container & Image

