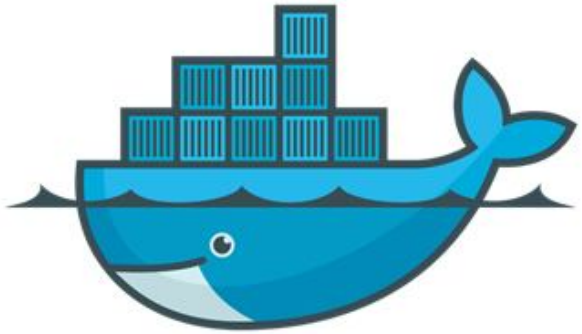
Docker Image & Container



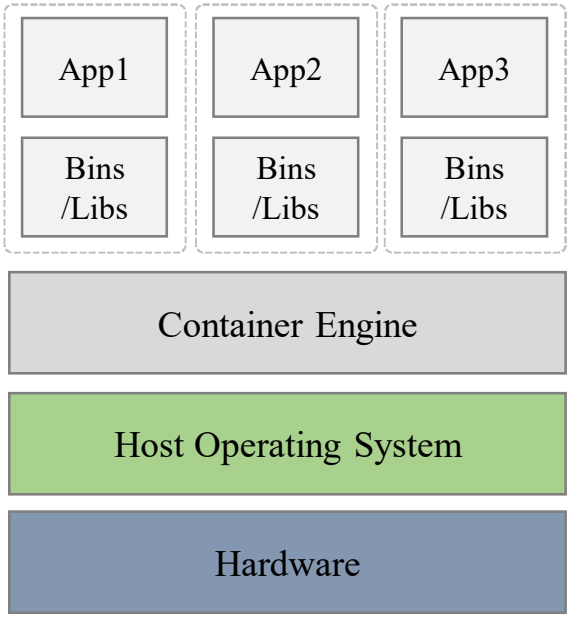
Docker

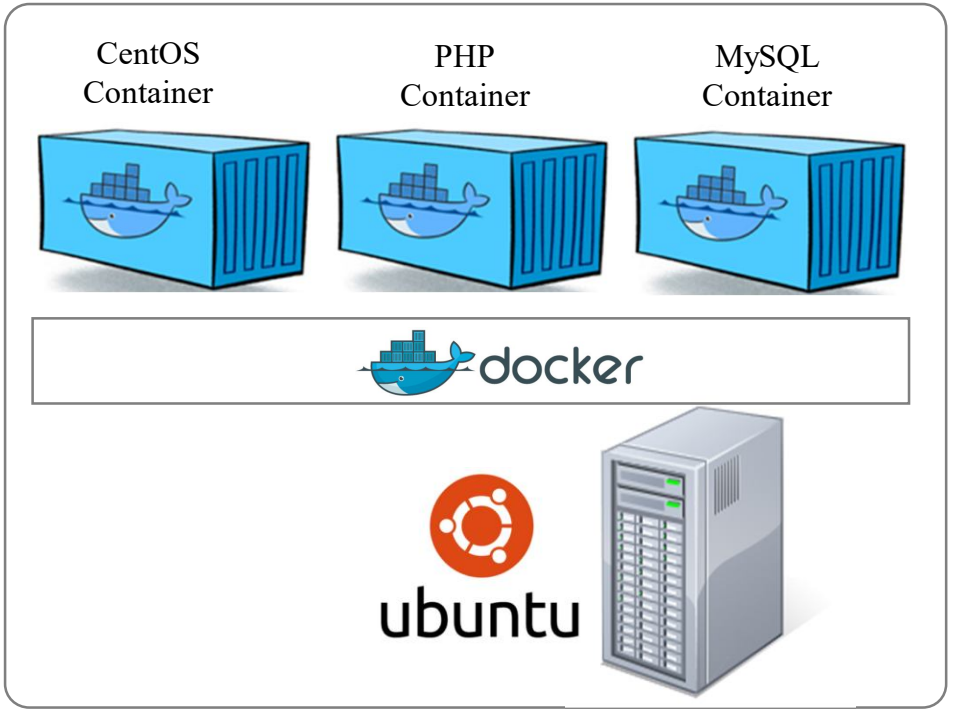
- Linux 상에서 동작하는 가상화 기술
- 가상화 환경에서 운영되는 애플리케이션의 시스템 구축 관리 도구
- 애플리케이션들의 실행 환경을 생성하고 관리하는 도구 (오픈 소스 플랫폼)

http://www.docker.com/

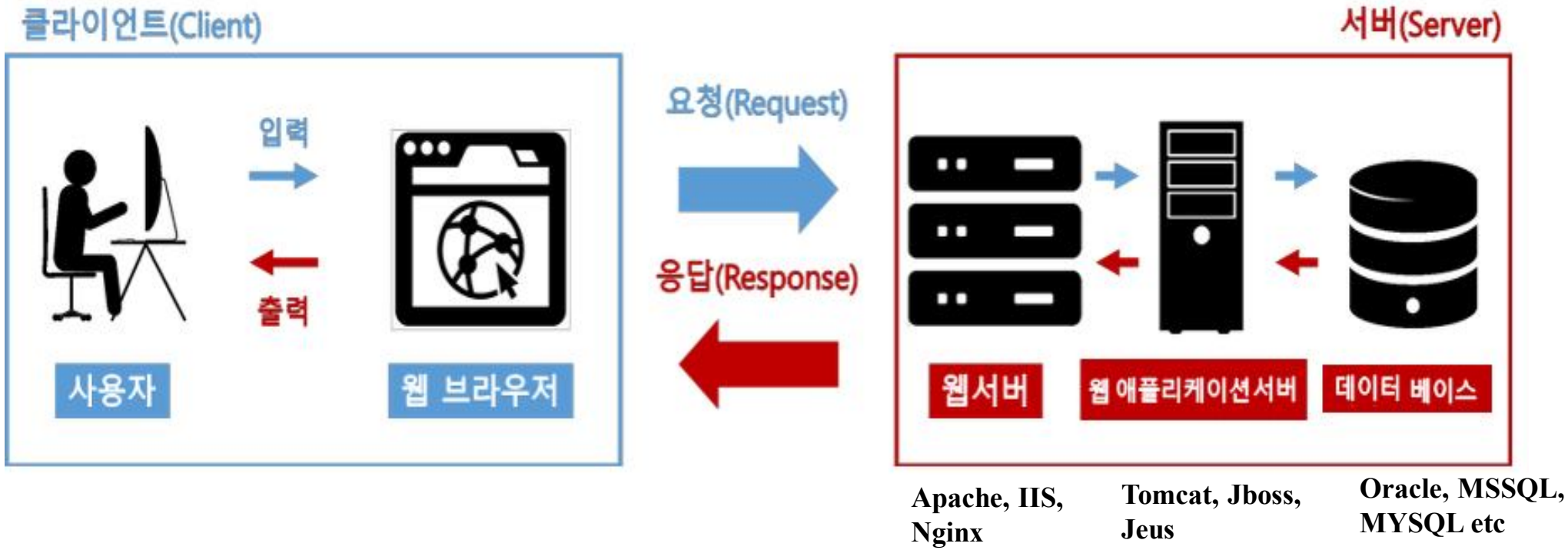
Docker Container

- 하나의 Application(실행파일)과 운영 환경이 모두 들어 있는 독립된 공간
- Container = Applications + Middlewares(라이브러리 파일)





Web System Architecture

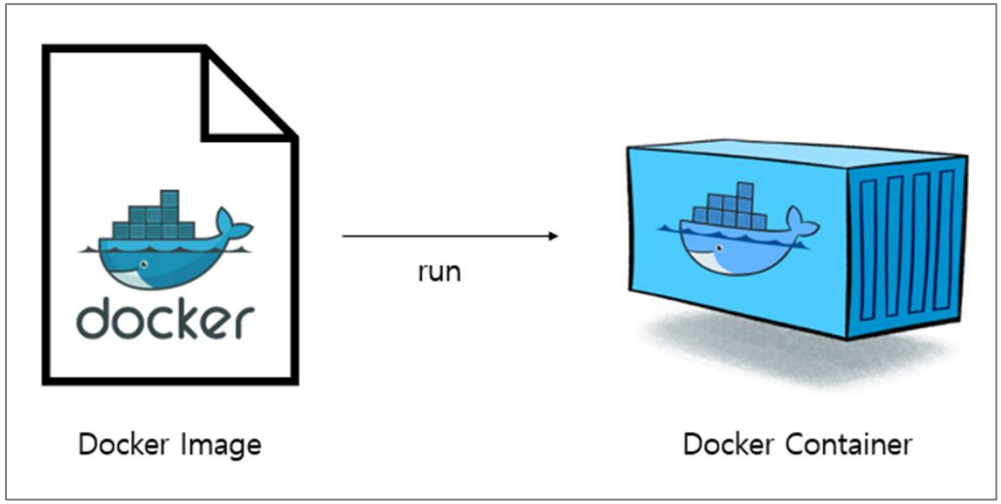


Docker Image

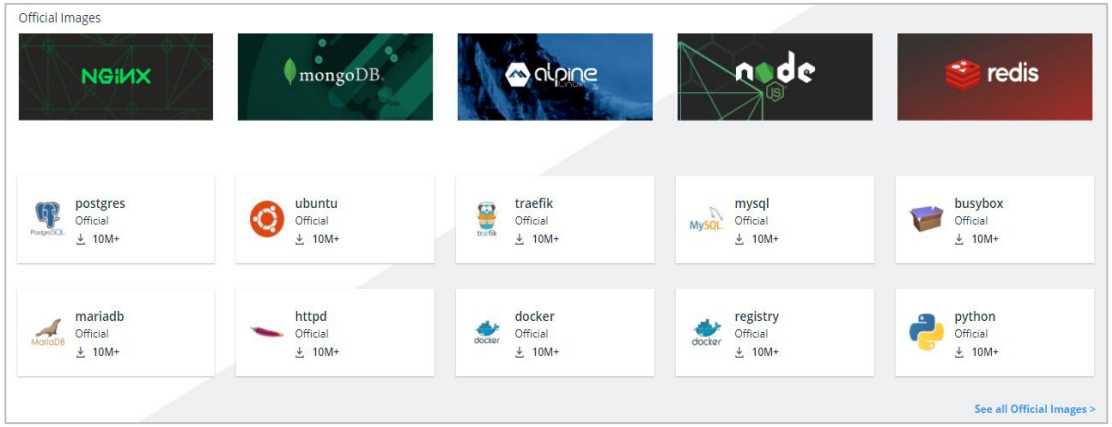
- 도커 서비스 운영에 필요한 서버 프로그램, 소스코드 및 라이브러리, 컴파일된 실행 파일을 묶는 형태
- 컨테이너 생성에 필요한 파일과 환경 설정 값을 가진 상태 파일

Docker Container

- Docker image를 실행(run)한 상태
- Docker image를 기반으로 container가 생성
- 컨테이너 생성에 필요한 파일과 환경설정값을 지닌 것



Docker Hub

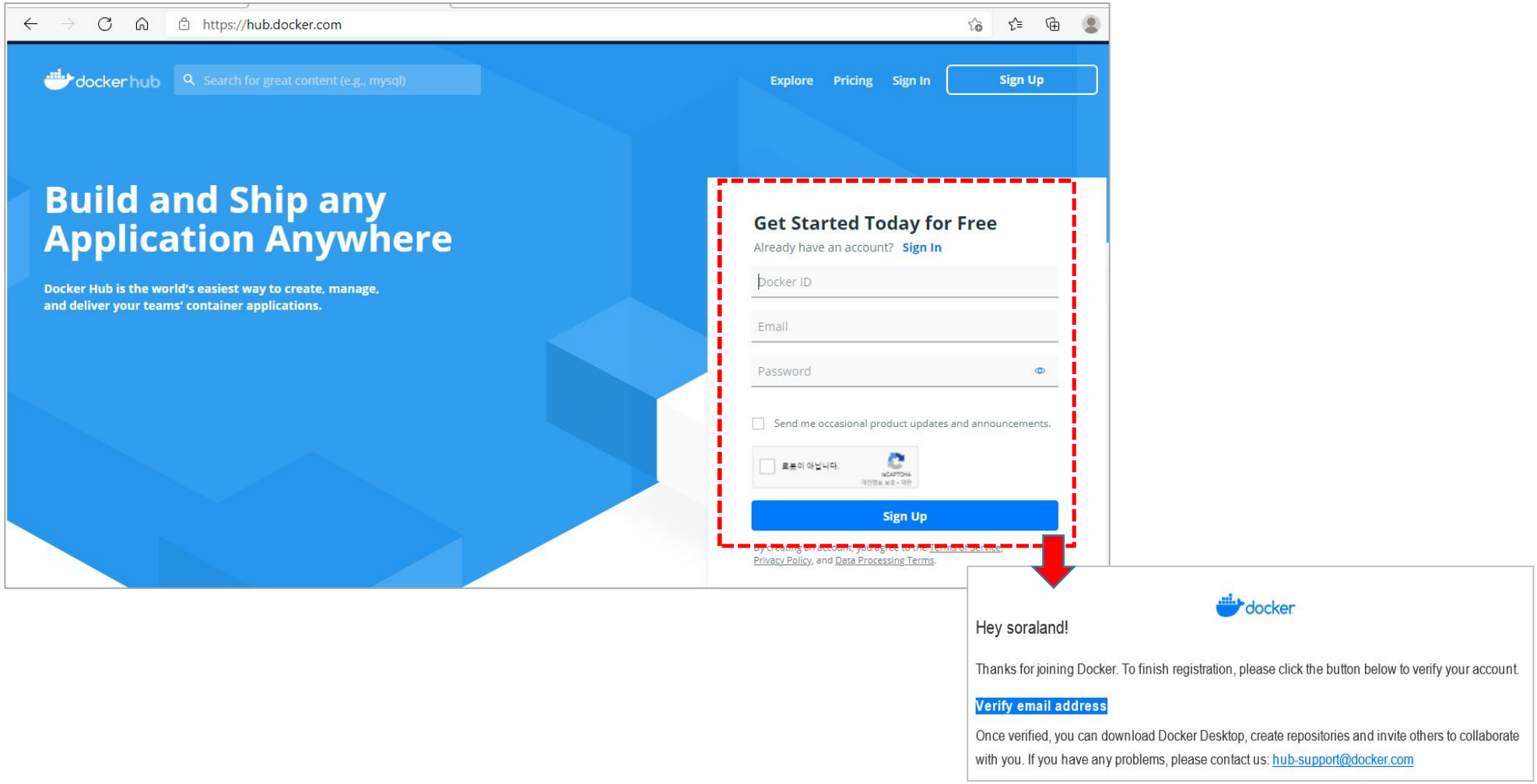


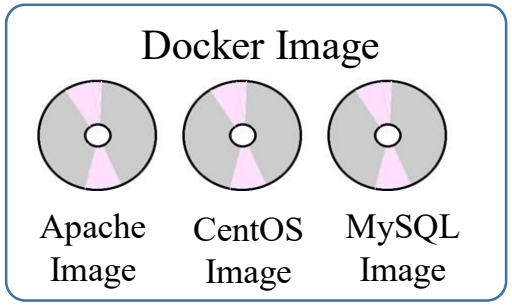
http://hub.docker.com

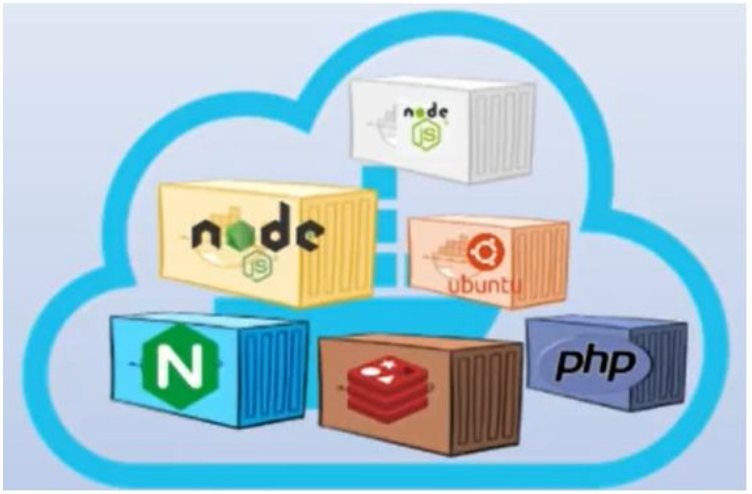
- Docker image를 저장하는 원격 스토리지 ➔ Image Registry
- Docker Hub는 Docker 에서 운영하는 Docker 이미지 저장소 서비스

  ➔ Docker의 공식 Image Registry
- 수 많은 Docker image가 업로드되어 있으며 Docker client와 기본적으로 연동 되어 있음
- 공식 Image 외에도 개인이 개발한 Image를 Docker Hub에서 자유롭게 공개하여 공유 할 수 있음
- 요건에 맞는 이미지를 자유롭게 다운 받을 수 있음

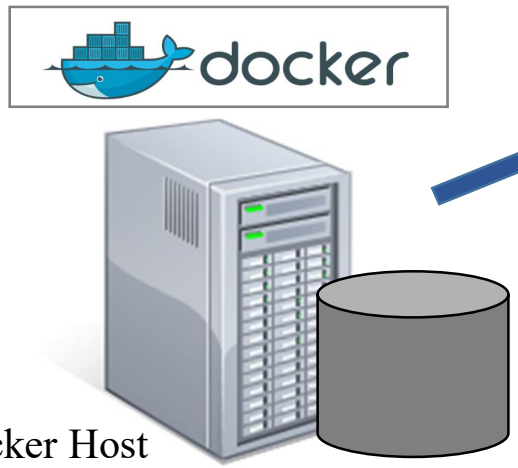
hub.docker.com 에 계정 생성

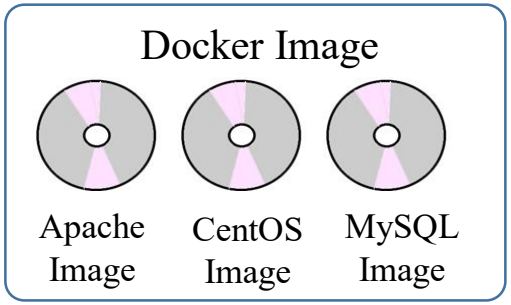






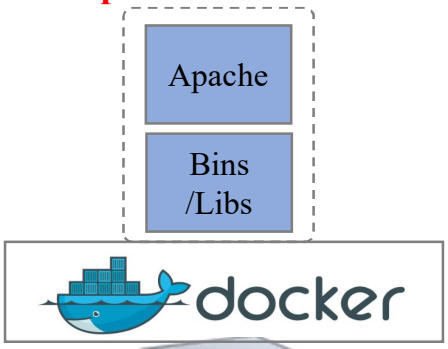
hub.docker.com



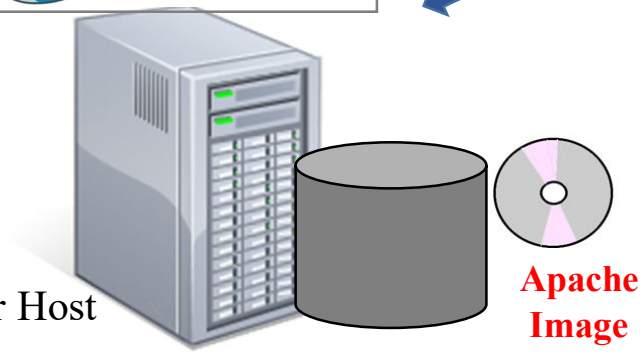


hub.docker.com

Apache Container

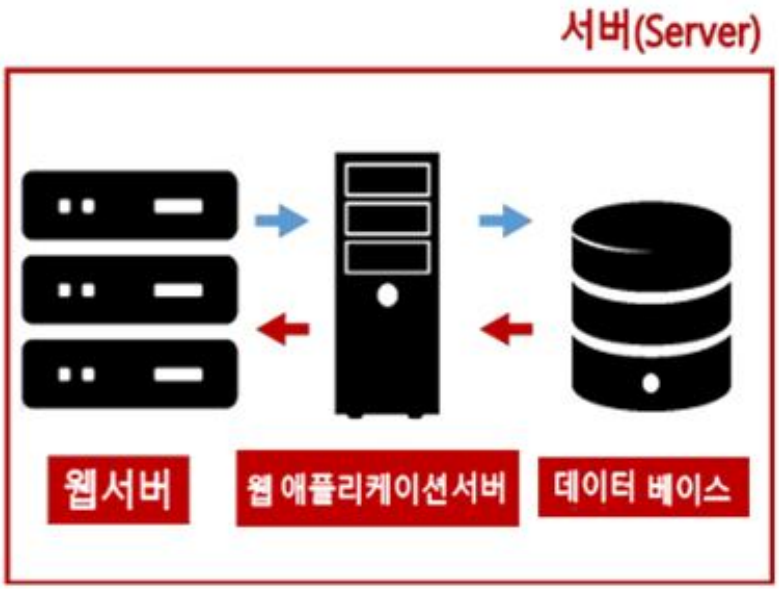


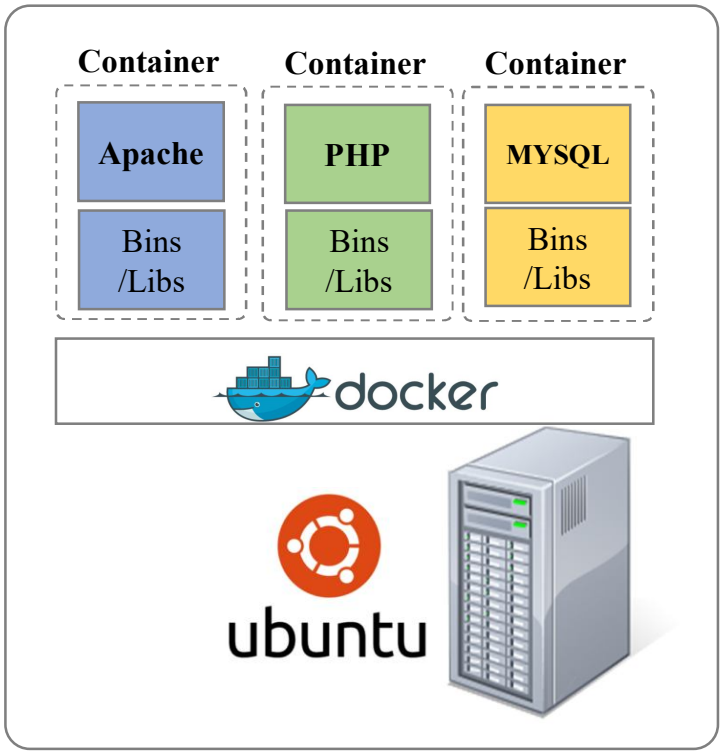
Apache Image Download



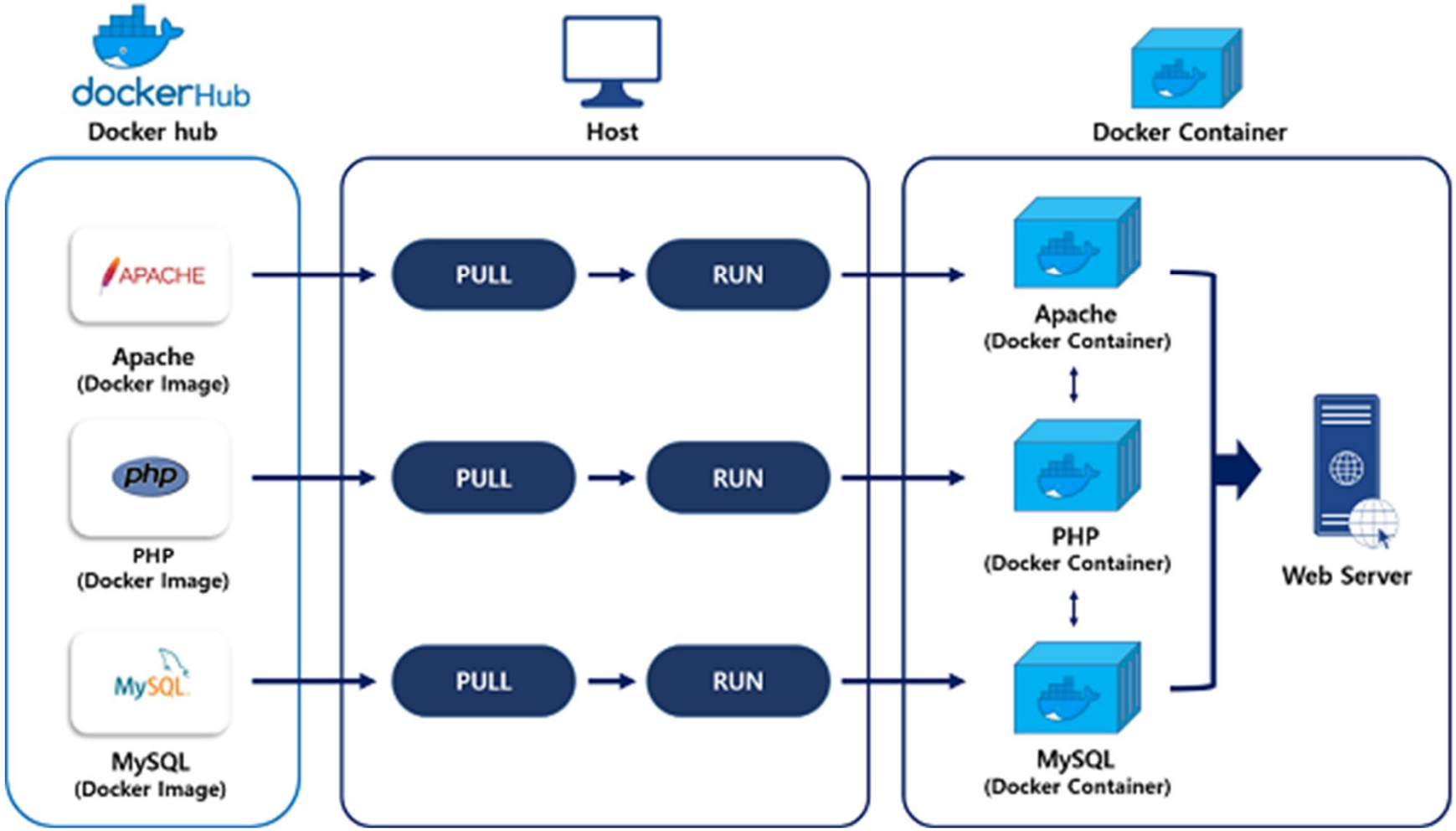
Docker Host

Apache Image



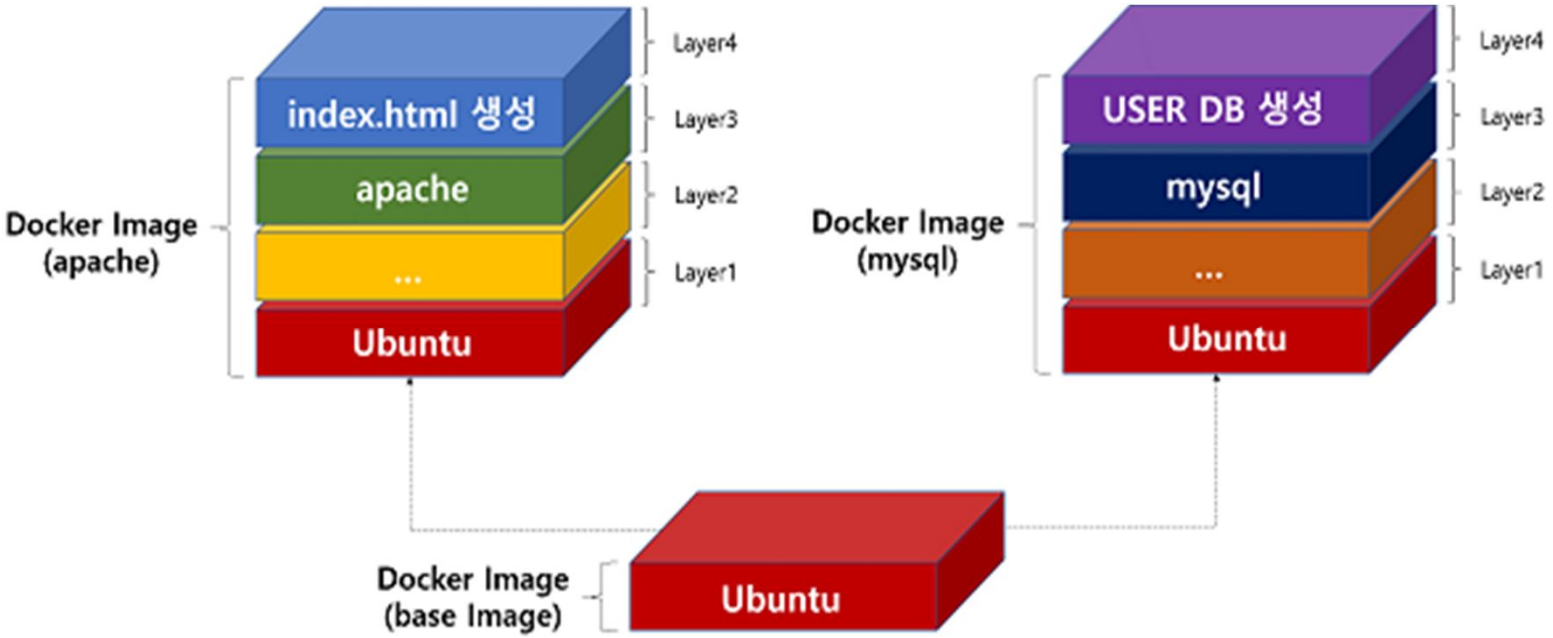


* Docker Host ; Docker 데몬이 동작되고 있는 리눅스 시스템



- 도커 이미지는 Docker hub에 등록하거나 Docker Registry 저장소를 직접 만들어 관리

Docker Image Layer



- Docker image는 여러 개의 레이어를 나누어 구성됨

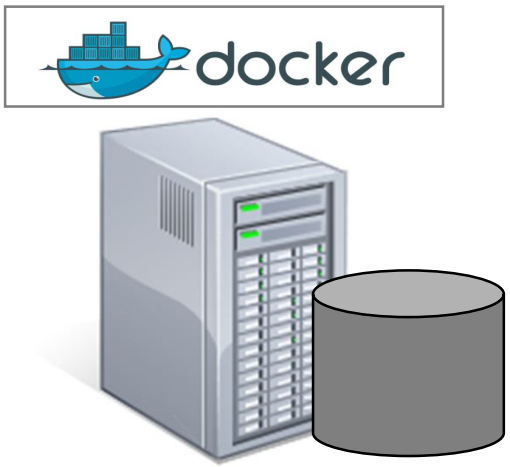
  (예) Apache Image = Base Image + Apache Image

- Layer 방식을 사용하면 기존의 image layer를 이용해 다양한 이미지를 새로 만들 수 있다.

Docker Container 생성과 삭제 실습

- Docker 데몬 확인
- Docker Hub에서 Image 검색
- Docker Image 다운로드
- Docker Container 생성하기
- Docker Image와 Container 삭제

❶ Docker Host 정상 운영 상태확인과 저장된 이미지 검색



Docker Host

- Docker 데몬 확인

  #systemctl status docker

  #docker version

- 로컬 저장소에 저장된 images 검색

  #docker images

❷ Docker hub에 저장되어 있는 image들 검색

hub.docker.com

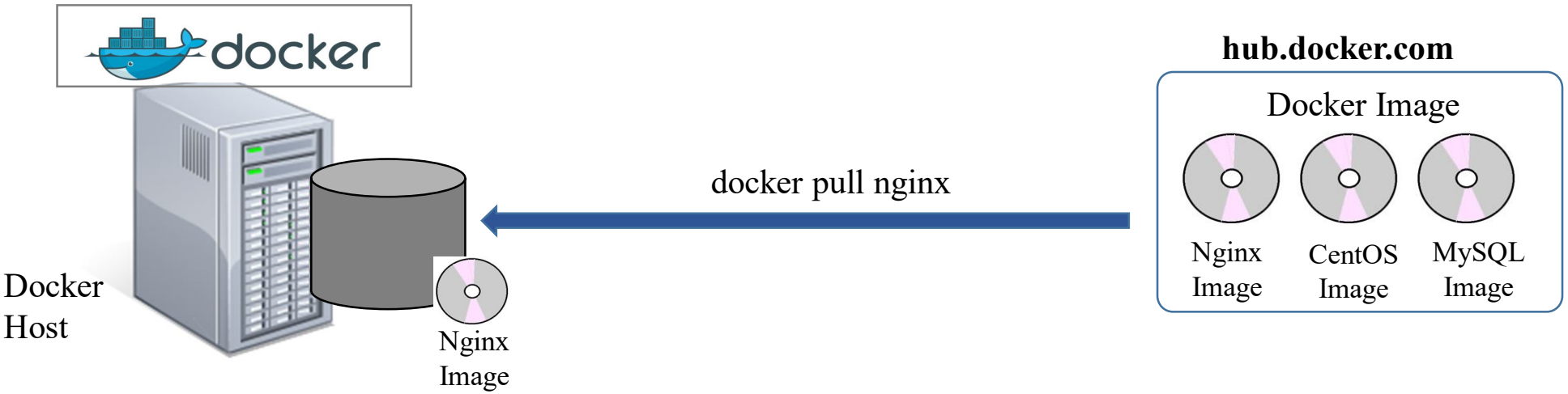
#docker search nginx

❸ Docker Hub에서 Image download

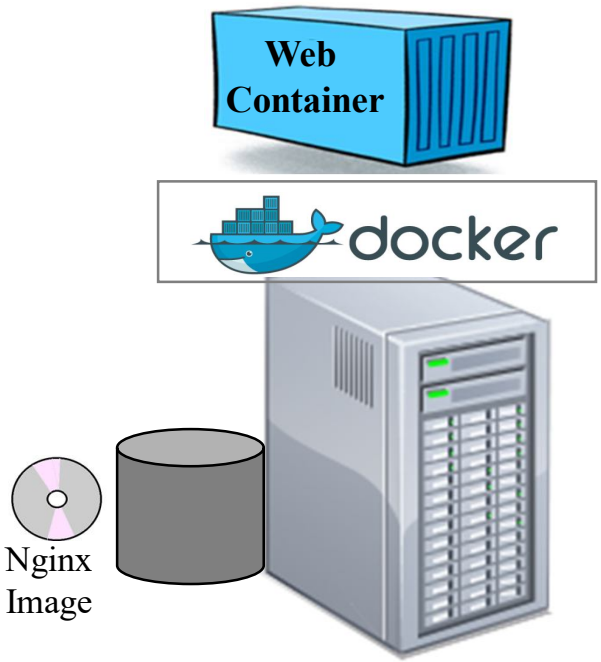
#cd /var/lib/docker/overlay2 ==> Images layer가 들어가 있는 디렉터리

#docker pull nginx

#docker images



❹ 컨테이너 생성

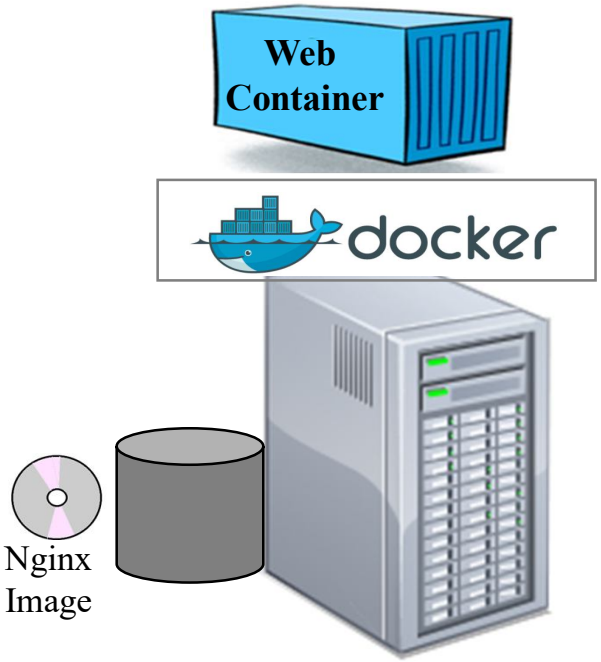


#docker run --name web -d -p 80:80 nginx

#docker ps

#curl localhost:80

❺ Container 삭제

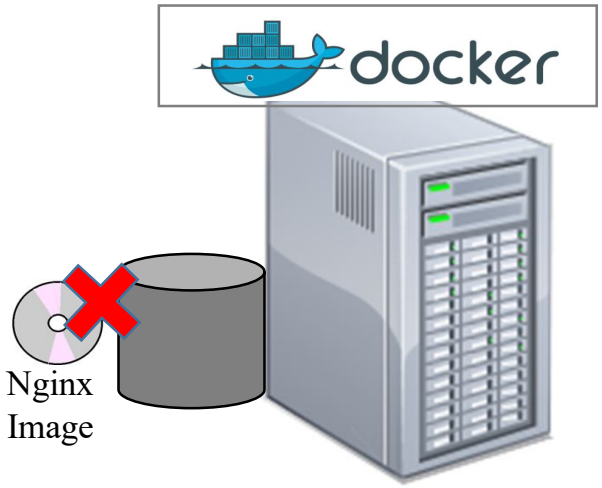


#docker ps

#docker stop web

#doceker rm web

❻ Image 삭제



#docker images

#docker rmi nginx