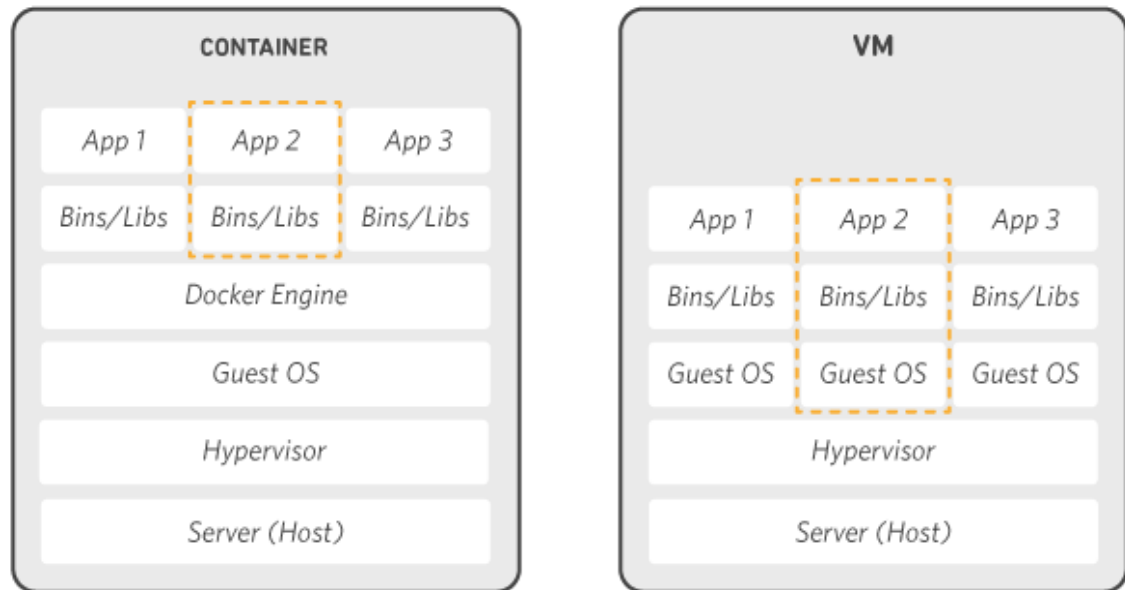


0xFF : Docker

Docker란?

Docker는 애플리케이션을 신속하게 구축, 테스트 및 배포할 수 있는 소프트웨어 플랫폼으로 일반적으로 사람들이 알고 있는 Oracle Virtual Box, VMware workstation과는 다르게 **Hypervisor**를 사용하는 것이 아닌 **Container**라는 개념을 사용한다.



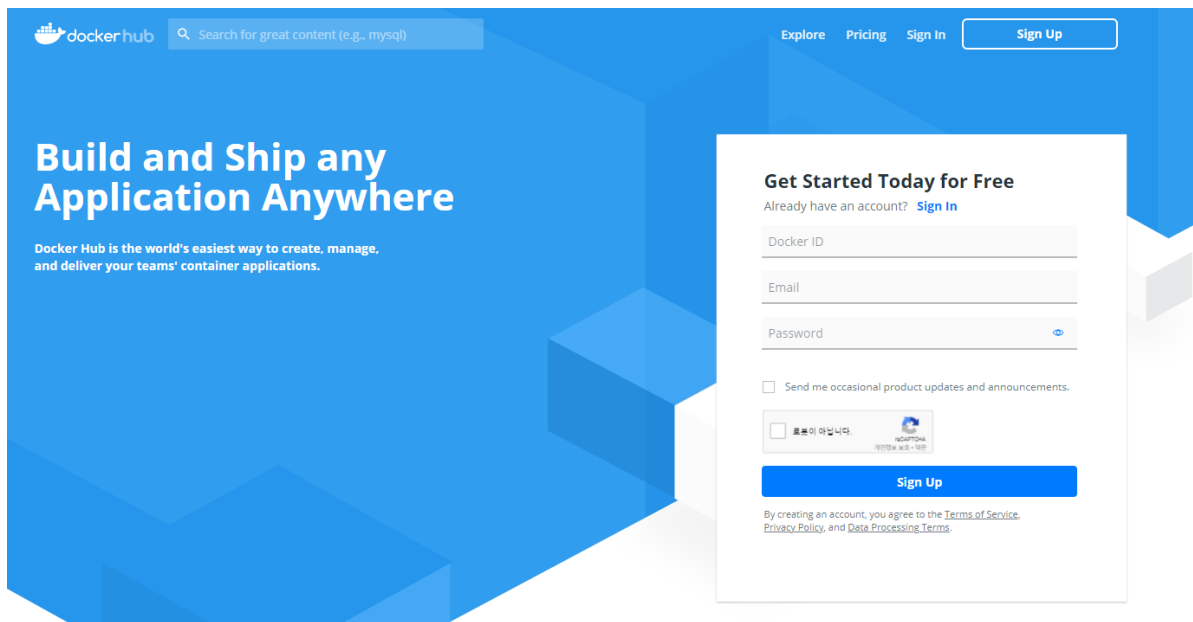
Container는 위 사진에서 확인할 수 있듯이 **Hypervisor**위에 바로 Guest OS 여러개가 실행되는 것이 아니라 **Docker Engine**이라는 것 위에 컨테이너가 올라가서 일반 VM보다 훨씬 더 활용성이 높고, 세팅하기도 편하며, 삭제, 보관이 용이하게 된다.

Container 개념에 대해서 깊게 들어가게 되면 매우 복잡한 글이 되므로 여기서 Docker에 대한 소개를 마치도록 하겠다.

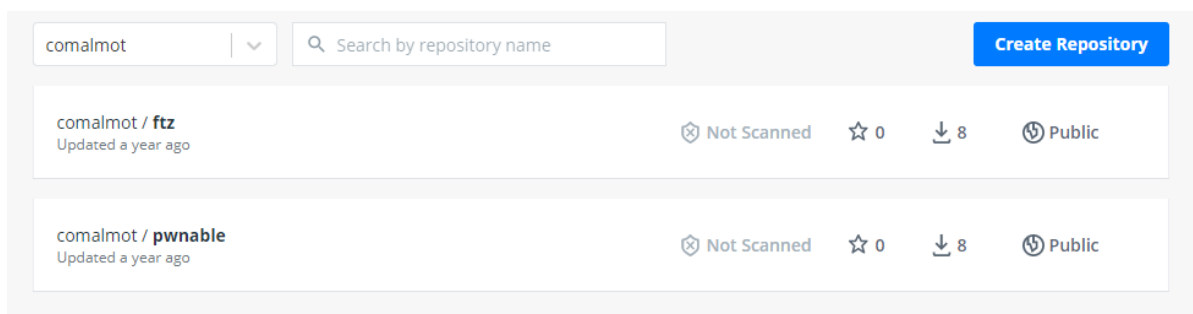
Docker Hub

Docker Hub는 Docker에서 제공하는 공식 이미지, 예를 들어 ubuntu, CentOS, Arch 등의 리눅스 이미지나 사용자가 만든 User Created 이미지도 만날 수 있다.

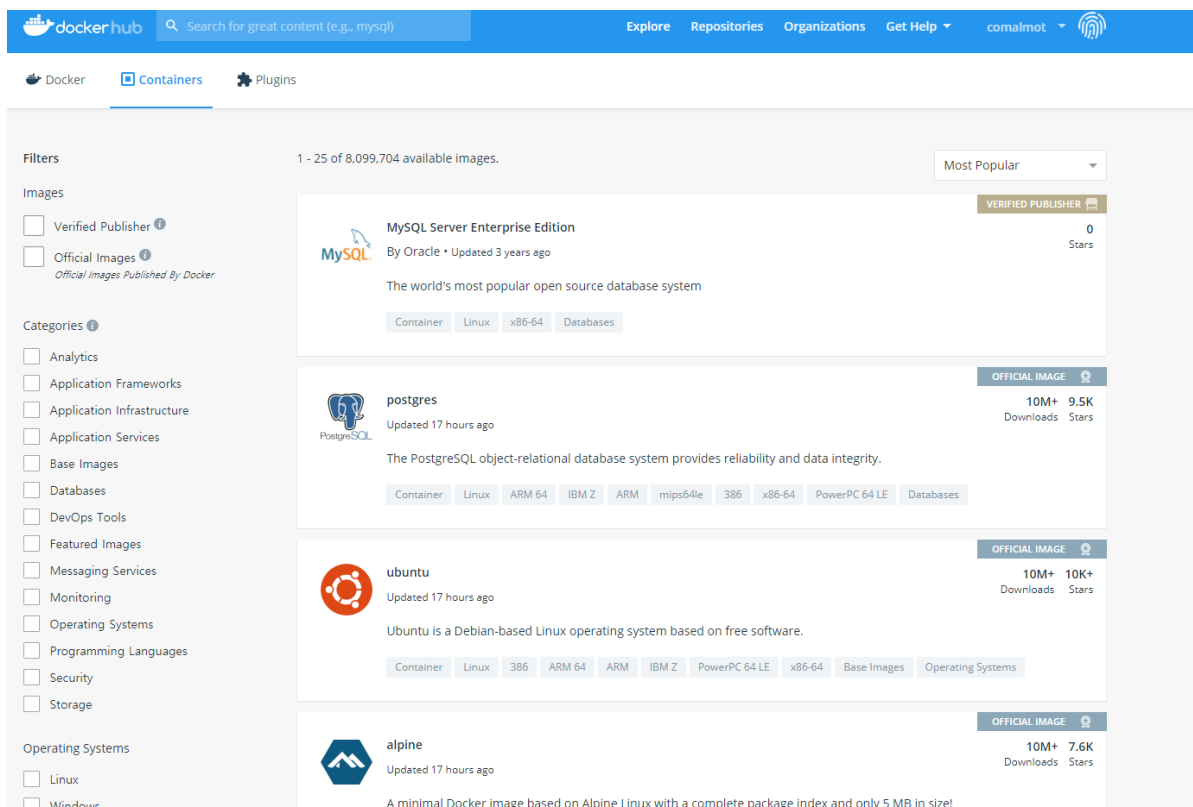
다만 사용자가 Docker에서 제공하는 공식 이미지처럼 새로운 배포판을 **Container**로 만들어 Docker Hub로 업로드할 수는 없으며, 기존에 올라와 있는 이미지를 가져다 용도에 따라 자신만의 세팅을 가미하여 (예를 들어 인공지능 서비스 전용 컨테이너, 웹서비스 전용 컨테이너 등) Docker Hub에 올려 버전 관리를 하는 것이 보통이다.



Docker Hub 메인 화면.



이렇게 레포지토리를 만들 수 있다 - 필자는 Ubuntu 18.04 LTS를 기반으로 pwnable 컨테이너를 구축, peda, pwndbg 등을 설치하여 포너블 전용 컨테이너를 제작하였다.



이렇게 여러가지 이미지들을 볼 수 있다.

Docker Container Command Line에서 설치 및 실행하기

Docker를 설치하기 위해서는 Ubuntu 기준으로

```
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io
```

를 하면 되겠다.

아래 명령은 dockerimage 중 nginx라는 이미지의 latest, 가장 최신 버전을 현재 Guest OS의 8080포트와 컨테이너의 80포트를 연결하여 실행시킨다는 것이다.

아래와 같이 실행을 하면 되겠다.

```
docker run -p 8080:80 nginx:latest
```

Dockerfile로 나만의 Docker Container 만들기

Dockerfile 작성 예이다.

```
# 1. ubuntu 설치 (패키지 업데이트 + 만든사람 표시)
FROM      ubuntu:16.04
MAINTAINER subicura@subicura.com
RUN       apt-get -y update

# 2. ruby 설치
RUN apt-get -y install ruby
RUN gem install bundler

# 3. 소스 복사
COPY . /usr/src/app

# 4. Gem 패키지 설치 (실행 디렉토리 설정)
WORKDIR /usr/src/app
RUN       bundle install

# 5. Sinatra 서버 실행 (Listen 포트 정의)
EXPOSE 4567
CMD       bundle exec ruby app.rb -o 0.0.0.0
```

이제 이 파일을 만든 디렉토리로 이동하여 이미지를 만들면 된다.

```
docker build -t app .
```

