

Modern PHP on Docker

www.rust-tutorials.com

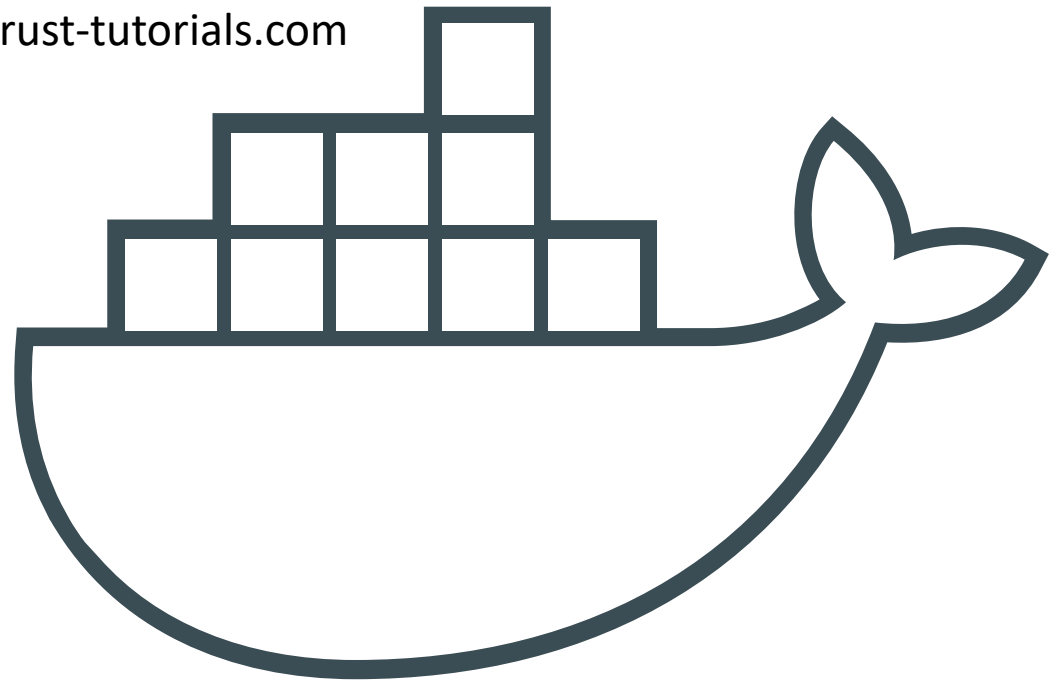


Table of Contents

이 튜토리얼의 목적

도커에 대해서

컨테이너 VS 가상머신

얼마전까지의 개발 환경

현재의 개발 환경

과거의 개발환경 VS 현재의 개발환경

이미지와 컨테이너

도커 명령어 익숙해지기

Dockerfile 익숙해지기

SSH 키 연결하기

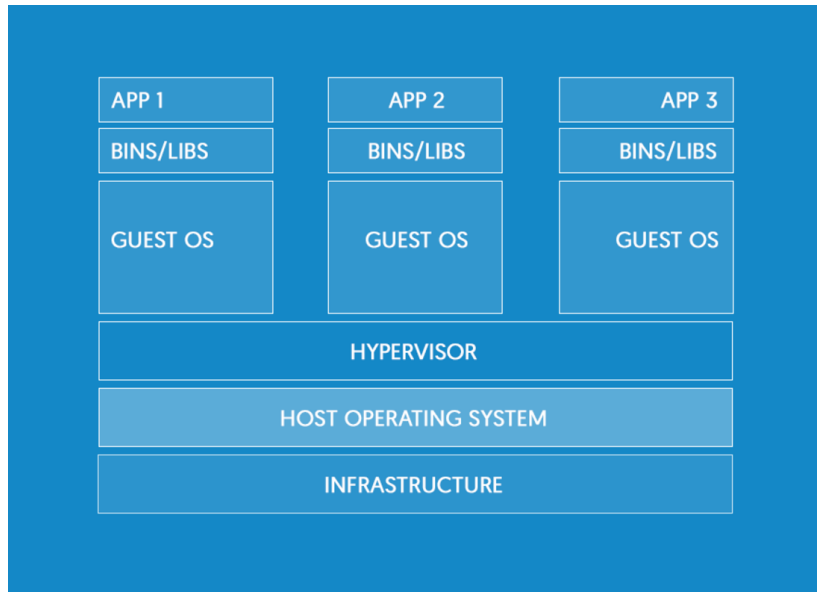
튜토리얼 목적

- 실제로 개발환경 구축에 필요한 도커 파악
 - 도커로 무얼 할 수 있는지를 알아봅니다.
 - 도커를 이용해서 개발 환경을 구축 할때 어떤 이점이 있는지 알아봅니다.
- Docker 이미지와 컨테이너 이해
 - 이미지와 컨테이너의 관계를 알아봅니다.
 - 이미지를 만드는 방법에 대해서 알아봅니다.
 - 이미지를 이용해서 컨테이너를 만드는 방법을 알아 봅니다.
- 간단한 웹 개발 환경 구축
 - 라라벨을 이용하여 간단한 웹 애플리케이션을 만들어 봅니다.

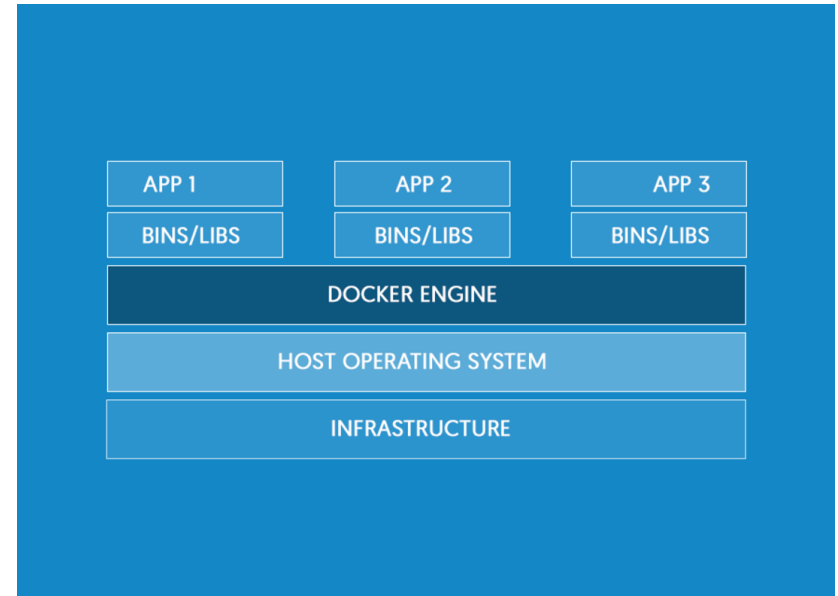
도커에 대해서

- 도커는 컨테이너를 손쉽게 관리할 수 있게 도와주는 도구
 - 밑에 저변에 깔려있는 기술은 복잡하나,
 - 실제로 몇개의 명령어(CLI)와 이미지와 컨테이너 개념만 익혀도 당장 개발환경으로 사용하기에 문제가 없습니다.
 - 기본적인 리눅스 명령어만 익혀도 괜찮아요.

가상머신 VS 컨테이너



가상머신



컨테이너

몇달전에 저의 개발환경



패러럴즈

+



vagrant

+



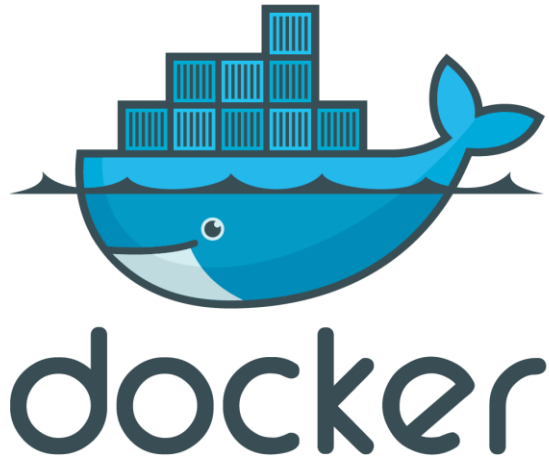
홈스테드

```
$ vagrant up
```

```
$ vagrant ssh
```

```
$ cd my_project_directory
```

현재 제가 사용하고 있는 개발 환경

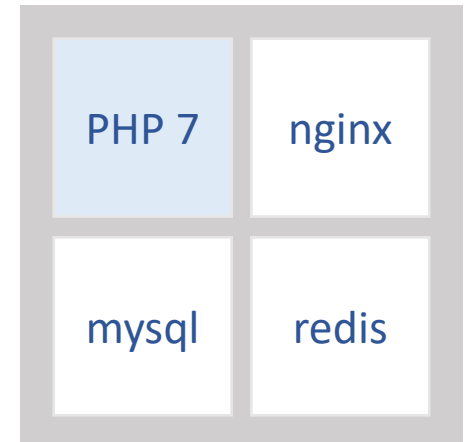
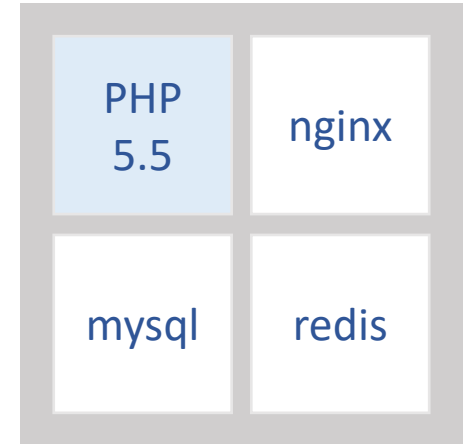


```
$ docker-compose up -d nginx mysql
```

```
$ docker-compose exec workspace bash
```

과거의 개발환경 VS 현재의 개발환경

- 프로젝트 별로 독립된 개발 환경을 구축
 - 애플리케이션마다 필요한 컨테이너를 올리면 된다.
 - 로컬 머신이 오염되지 않아서 좋다.



이미지와 컨테이너

- 이미지는 컨테이너를 만들기 위한 일종의 템플릿
 - 이미지를 이용해서 컨테이너를 생성합니다.
 - 도커 명령어를 이용해서 이미지를 로컬머신에 다운로드 받습니다.
 - `$ Docker pull <이미지이름>`
 - 이미지 이름은 <https://hub.docker.com/>에 찾아보자.
- 컨테이너는 이미지를 이용해서 만들어진 인스턴스 이자 독립된 공간이다
 - 컨테이너를 생성하기 위해서는 이미지가 필요하다.
 - `$ docker images`로 명령어로 호스트 머신에 설치된 이미지를 확인 할 수 있다.
 - 도커 명령어를 이용해서 이미지를 컨테이너로 생성하자.
 - `$ docker run <이미지이름>`

도커 명령어 익숙해지기

명령어	설명
\$ docker pull	이미지를 로컬 머신에서 가져온다.
\$ docker images	모든 이미지를 보여준다.
\$ docker ps	현재 실행중인 컨테이너를 보여준다.
\$ docker rm	컨테이너를 삭제한다
\$ docker run	컨테이너를 생성한다.
\$ docker stop	컨테이너를 정지시킨다.